Projet ANR-11-IS02-001

MEX-CULTURE/ Multimedia libraries indexing for the preservation and dissemination of the Mexican Culture

Deliverable

MID-term report on scalable visual descriptors

Programme Blanc International II- 2011 Edition

A IDENTIFICATION

Project acronym	MEX-CULTURE
Project title	Multimedia libraries indexing for the preservation and dissemination of the Mexican Culture
Coordinator of the French part of the project (company/organization)	Centre d'Etude et de Recherche en Informatique et Communications – Conservatoire National des Arts et Métiers
Coordinator of the Mexican part of the project (company/organization)	Centro de Investigación y Desarrollo de Tecnología Digital – Instituto Politécnico Nacional
Project coordinator (if applicable)	Michel Crucianu : France Mireya Saraí García-Vázquez : México
Project start date Project end date	01/01/2012* 31/12/2014
Competitiveness cluster labels and contacts (cluster, name and e-mail of contact)	Cap Digital Paris-Région Philippe Roy Philippe.Roy@capdigital.com
Project website if applicable	http://mxcprj.hopto.org/

** The Mexican partners are only financed since November 2012.*

Coordinator of this report	
Title, first name, surname	*Mireya Saraí García-Vázquez*
Telephone	*(+52) 664 2 12 60 08*
E-mail	*freemgarciav@gmail.com*
Date of writing	*20/01/2014*

Redactors :	Mireya Saraí García Vázquez (CITEDI-IPN) Alejandro Ramírez Acosta (CITEDI-IPN) Mariko Nakano Miyatake (ESIME-IPN), Jesús Yaljá Montiel Pérez (ESCOM-IPN) Manuel Cedillo Hernández (DPSFI-UNAM) Francisco García Ugalde (DPSFI-UNAM) Jenny Benois-Pineau (LaBRI) Michel Crucianu (CEDRIC) Stoian Andrei (CEDRIC)

B INTRODUCTION

B.1 MOTIVATION

Given the importance of cultural heritage content in promoting diversity in a globalized world, making this content quickly available to a broad audience is a critical issue. Large volumes of such content must be indexed and users must be provided with means for a fast and easy access to the multimedia information, making them able to browse (according to multiple criteria) and visualize desirable content stored in the archives. One of the research topics in this kind of processing multimedia information is information retrieval and its use in target application areas such as digital libraries and archives. Indeed, this topic requires exploring methods and tools for (semi-)automatic indexing and semantic annotation of non-textual objects (music, speech, images, videos). Given the size of the databases envolved, such operations must rely on *automatic* content-based indexing, as well as on scalable summarization and content-based retrieval. As audiovisual content itself is used as basis for indexing, then the content should be processed in the compressed form in order to save the computational cost of decompressing the videos.

At the beginning of multimedia indexing in 1990, methods for compressed stream analysis were not largely recognized. It is nowadays that the community finally understands the importance of the Rough Indexing paradigm. New compression standards (specifically JPEG2000 thanks to the transform used) are scalable, which enables indexing and retrieval of the content at various resolutions; and various scenarios for indexing simultaneously with compression developed by leading R&D actors (Philips Research) show that this is a way to follow. Nevertheless, the robustness of extractors and, consequently, of extracted features is a challenge to address in the framework of the Mex-Culture project.

The Content Based Image Retrieval CBIR has appeared in 90 years. It represents each image by a set of visual low-level features such as color, texture, shape and movement. These visual features are calculated automatically and then exploited for the system to compare and retrieve images. So, in the context of the visual characteristics of search for images/videos, we will develop this report.

B.2 AUDIO-VISUAL CORPORA

The methods devised in the project will be applied to the large databases (100,000 hours) provided by TVUNAM of the UNAM (National Autonomous University of Mexico), DL-INAH (Department of Linguistics - National Institute of Anthropology and History), audio-only content by the FONOTECA NACIONAL (National Sound Archive of Mexico), part of CONACULTA (National Council for Culture and the Arts of Mexico), VOD TVUNAM canal youtube and VOD Canal11IPN canal youtube. Since these databases will only become progressively available during the project, a large video database provided by INA (sub-contractor of Cnam) will be employed for the early evaluation of the methods devised in this project.

B.3 General Framework for scalable description of visual encoded content

One of our goals in this project is to devise and evaluate new automated methods for large-scale processing and indexing of multimedia content. This methods extract effective local and global spatio-temporal descriptors from JPEG2000 compressed flow. To achieve object-based indexing, it is necessary to be able to extract objects of interest from the image. The objects to be extracted are sets of connected components that are homogeneous in color and/or in texture with respect to a given criterion and that have a motion different from the global motion of the scene. Generally, authors try to mix color and motion information. Other way to make moving object extraction is to combine the results of color segmentation and motion segmentation.

In order to achieve the objectives mentioned in the previous paragraph, we have executed several works based mainly on the study and implementation of feature extraction techniques: how they are extracted, how low-level features are mapped to their high-level correspondences, whether retrieval in each modality (other than text) can be effectively improved.
The implemented works as well as the achieved results and the scope of these within the project are described in the following sections.

C State-of-the-art in Scalable description of Visual Encoded Content

This section presents a brief overview of the works developed for extract the local and global descriptors representing the characteristics of the content of the images and video.

C.1 Cross-media description of visual content

Cross-media description of visual content is the process of generating visuals descriptors by exploiting the media streams (images, videos), belonging to a single document.
The methods for obtaining visual descriptors are based on the analysis of information from the visual content of the images and videos. This analysis generates features of color, texture, shape and motion, or a combination of these.

C.2 Mexican Culture Video Retrieval System Based on Object Matching and Local Descriptor

Content Based Video Retrieval Systems (CBVR) select extracted features from video content for selecting, indexing and ranking according to the potential interest to the user. We propose a fast CBVR technique which involves the combination of a local descriptor obtained from the Speeded-Up Robust Feature (SURF) algorithm together with an effective and fast object matching operation (see section D.2 for details of the implementation). Before computing the SURF descriptor, the key frames are partially extracted from codified video: video frames are decoded from DCT to spatial domain using a fast inter-transformation between block DCTs and sub-block DCTs, then down-sampling frames are obtained by replacing each sub-blocks DCT of 2x2 pixels with half of the corresponding DC coefficient.

This strategy can significantly save highly computational cost in comparison with the conventional method. To measure the performance of the proposed technique the precision and recall metrics are used. The experimental results show the accuracy of the proposed fast CBVR technique applied to a database of Mexican Culture videos which are captured in highly varied environmental conditions and with different acquisition devices.

C.3 Global-Local color Descriptor for CBIR

In Content-Based Image Retrieval CBIR systems, the image descriptor is a very important element because is responsible for assessing the similarities among images and can be classified depending on the image property analyzed such as color, texture or shape [C.3.1]. The objective of CBIR systems is to retrieve images from a database that are more similar to a user query image. This is done by extracting the visual feature from images and store them in image descriptors which are compared in order to obtain the similarities.

For retrieving color images from multimedia database, low level features and especially the color feature, have been widely used in this regard [C.3.2], this is because color has the following characteristics: is the basic composition element of image content and compared with the other visual features is the most distinguishable feature as well as is insensitive to image translation, scale and rotation [C.3.3, C.3.2, C.3.4]. Color-based image descriptors are classified into two categories: global color descriptors and local color descriptors. Global color descriptors consider the whole image to obtain color features, there is no partitioning or pre-processing stage during feature extraction process. Local color descriptors obtain their visual features by partitioning the image into either fixed or different size regions [C.3.2].

We proposed a scheme which extracts global and local color features obtaining the global distribution of colors. We analyzed the whole image in order to obtain global color features using the Dominant Color Descriptor (DCD) proposed by MPEG-7 [C.3.4, C.3.5]. DCD calculate the distribution of colors, thus, we have the information of what colors contain the images, as well as the percentage of each color. We also partitioned the image into blocks in order to be analyzed one by one. In each block the correlation of the pixels and their neighborhood is computed using color autocorrelogram. Color autocorrelogram obtain the information of how the pixels are spatially distributed in the image. Thus, we managed to combine global and local color feature in one algorithm, making the proposed scheme a powerful tool for content-based image retrieval. (see section D.3 for details of the implementation).

C.4 Extraction Frames

The work description is being developed for extracting video frames representative described in section D.4.

C.5 Descriptors for Content Based Image Retrieval

Nowadays, it is essential to have sophisticated mathematical techniques to locate the relevant information in the media content for different applications. These techniques make

use of large-scale multimedia databases. These methods are based on the content analysis of images and video for obtaining multimedia content descriptors that allow to get characteristic patterns from it. With this, relevant information from the visual content is then labeled. The visual content information is obtained through visual descriptors with features of color, texture, shape and motion or a combination of these [C.5.1]. The characteristic patterns of visual content are correlated with subjective visual content through statistical similarity measures in the context of Content Based Image Retrieval [C.5.2]. In Section D.5, six mathematical techniques are described to obtain the characteristic patterns of color and texture from image content. The objective is to develop an effective mechanism to obtain the characteristic patterns that describe the features of color and texture from image. The analysis of these techniques we will allow to extrapolate its implementation and optimization in the compressed media of the JPEG2000 standard for the next phase of Mex-Culture project.

D PROPOSED METHODS FOR SCALABLE DESCRIPTION OF VISUAL ENCODED CONTENT

D.1 GENERAL ARCHITECTURE OF THE PROJECT

Indexing and retrieval of multimedia information is based on architecture that we called architecture for scalable search. This architecture will allow to store and organize multimedia information in a scalable manner, allowing a multimedia resource search in large databases scale easily and quickly.

The feature of this architecture is that the indexing and retrieval of information is performed in the compressed domain of multimedia information [D.1.1, D.1.2]. Figure D1.1 shows the proposed architecture.

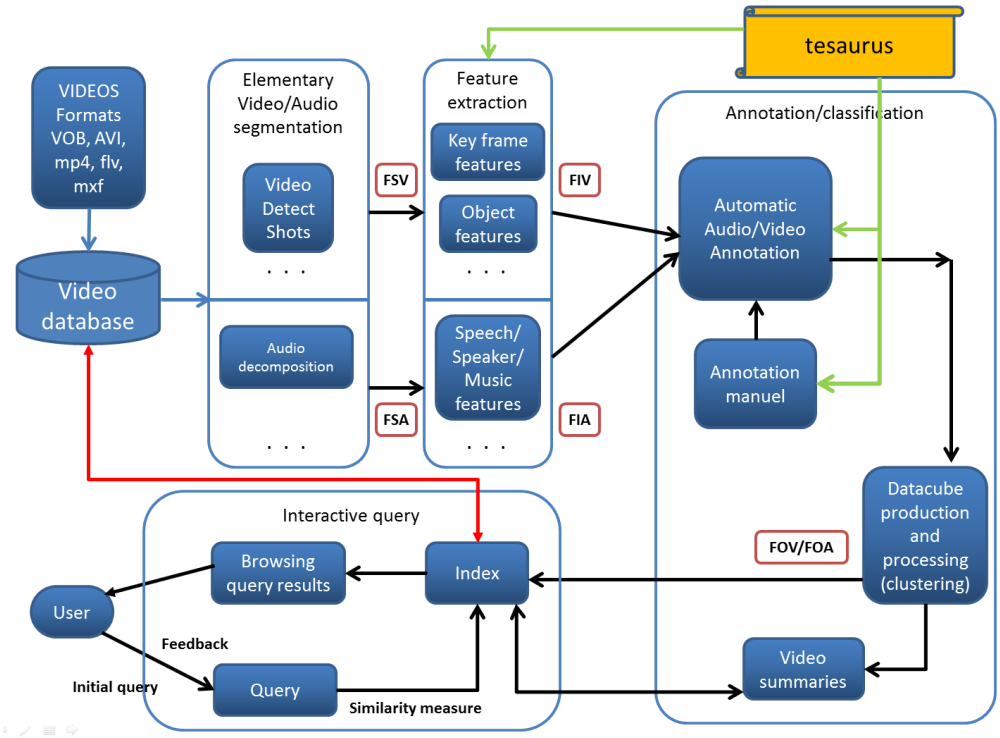


Figure D1.1. Architecture for content-based audiovisual scalable indexing and retrieval.

In Figure D2.2 we can see that the decomposition tasks are reflected in this scalable search architecture. It shows then the interrelationship between the different activities of each partner:

Task 1: Scalable description of visual encoded content. Extract effective local and global spatio-temporal descriptors from JPEG2000 compressed flow.

Task 2: Description of speech/audio content. Speech/audio signal segmentation, description and classification of sound events, Mexican native language speech recognition.

Task 3: Audiovisual summaries and scalable retrieval. Develop scalable methods for structuring the audiovisual database and for interactive content-based multimodal retrieval.

Task 4: Software development of multi-modal algorithms. Make the databases available for the project and develop the software platform.

Tasks 1, 2 and 3 all develop software components (for extracting content descriptions, for performing content summarization and for scalable search). All software integration activities were grouped in Task 4.

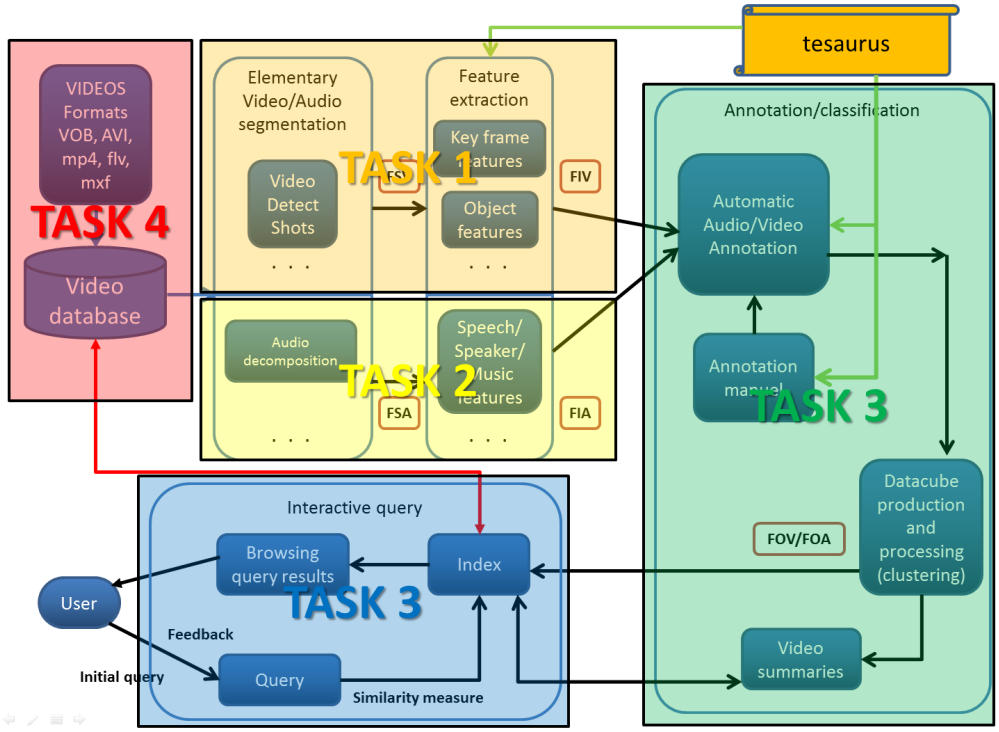


Figure D2.2. Task in the architecture for content-based audiovisual scalable indexing and retrieval.

The architecture for content-based audiovisual scalable indexing and retrieval includes the following:

a) **Video database**: the information of audiovisual content is in the compressed domain (audio format: MP4; video format: JPEG2000).

b) **Elementary video/audio segmentation**: the audiovisual content structure analysis aims at segmenting an audiovisual content into a number of structural elements that have semantic contents.

c) **Feature extraction**: the extraction of features of the audiovisual content in the structural elements, represents the base for scalable indexing and retrieval.
   - For visual encoded content we consider a scalable method of signal representation such as a spatio-temporal motion compensated wavelet-based solution. The objects to be extracted are sets of connected components that are homogeneous in color and/or in texture with respect to a given criterion and that have a motion different from the global motion of the scene [D.1.3, D.1.4, D.1.5, D.1.6].
   - The features of the audio encoded content are obtained with the automatic extraction of the descriptors. These descriptors are made up by different parameters that represent sound events of the speech/audio signals and by parameters that help the speech recognition performance with identification tasks [D.1.7, D.1.8].

d) **Annotation/classification**: these rely heavily on audiovisual structure analysis and the extracted audiovisual features. The annotation is the basis for the detection of audiovisual semantic concepts and the construction of semantic indices for audiovisual content [D.1.9]. The classification is to find rules using extracted features and then assign the audiovisual content into predefined categories [D.1.10]. The scalable wavelet-based descriptors (Task1) will be combined with audio descriptors (Task2) in order to use them in a global data-cube model which will represent each audiovisual document. To the constitution of visual summaries, we apply data summarization techniques coming from information retrieval domain [D.1.11].

e) **Interactive query**: The scalability challenge is significantly reinforced by the use of partial queries. The signal descriptions issued from tasks 1 and 2 will be employed; these descriptions are well-adapted to partial queries. It is will extend the Locality Sensitive Hashing (LSH) approach in two directions: improve the balance between effectiveness and efficiency for the types of content concerned by this project, and devise LSH-based methods for scalable retrieval with relevance feedback RF [D.1.12].

D.2 Proposed Method Based on Object Matching and Local Descriptor

With the rapid growing of multimedia data and networking technologies, especially bandwidth, the demand of video retrieval systems has been increased due to user's shift from text to content based retrieval systems. Content Based Video Retrieval System (CBVR) is defined as the search which retrieves video from a database, based on contents. [D.2.1] In this context, content is defined as some feature vectors which can be obtained directly from video such as color, shapes or textures [D.2.2], [D.2.3], [D.2.4]. The selection of extracted features from video content are intended for selecting, indexing and ranking according to their potential interest to the user [D.2.5]. However, as a drawback of the low-level features, is the loss of information details of the frames, when the frame that contains the same object is watching with different viewpoints. In recent years, the feature point detectors and descriptors [D.2.6] has been employed in several CBVR techniques to solve the above

mentioned drawback [D.2.7]. Thus, several visual descriptors have been proposed in the literature, which in many cases are dependent of the application as well as of the multimedia database where are applied. Hence, it is clear that there is not yet been able to solve the problematic of which descriptor is better or worse in a particular application [D.2.8]. In this way, challenging applications consist of retrieve videos with the same object which are captured by different devices in a highly variety of environmental conditions. We propose a fast CBVR technique which involves the combination of a local descriptor obtained from the Speeded-Up Robust Feature (SURF) algorithm [D.2.9] together with an effective and fast object matching operation.

The general procedure of the proposed CBVR technique is shown in Fig. D2.1. The proposed CBVR technique contains three stages: 1) The key frames are extracted from codified video, which are still images that best represent the content of video; 2) Feature extraction is done, where a feature vector is generated to accurately represent the content of each video in the descriptors database; 3) The retrieval stage is performed to retrieve the "closest" videos.

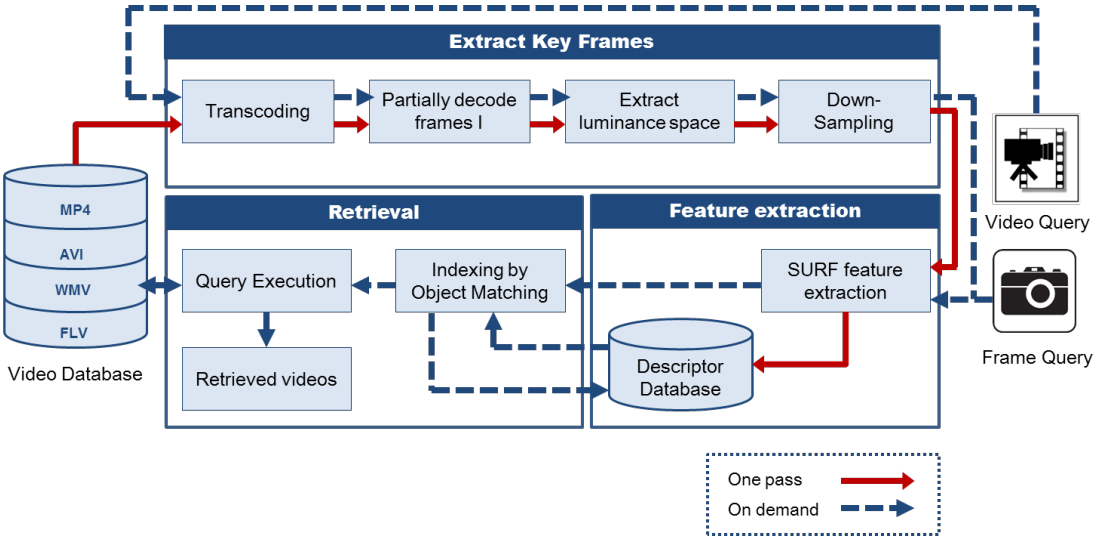


Figure D2.1. Proposed CBVR technique.

Given a video, read from the video database or as a video query, the first stage consists to extract the key frames from the codified video which best represent the content of video. The process of this stage is described as follows: 1) Video transcoding is performed changing the bit rate to 4 Mbps, MPEG-2 compression standard, 15 frames/seconds as temporal resolution and Common Intermediate Format (CIF-352x288) as spatial resolution. This task ensures the performance of the proposed CBVR technique independently of the video formats from the video database. 2) The video is partially decoded in order to obtain the DCT blocks of the codified *I*-frames. 3) The luminance DCT coefficients of every *I*-frame are isolated because this component has the highest frame information. 4) Calculate the 2x2 block DCT coefficients from the *I*-luminance block DCT frame by using Eq. (1) twice [D.2.10].

$$C_{b(i,j)} = A_1^{-1} \times \begin{bmatrix} C_{sb(1,1)} & C_{sb(1,2)} \\ C_{sb(2,1)} & C_{sb(2,2)} \end{bmatrix} \times A_1 \qquad (1)$$

Then, obtain the down-sampled version of the 2x2 block DCT *I*-luminance frame via replacing each 2x2 block with half of its DC coefficients [D.2.11]. The fast inter-transformation between block DCT and sub-block DCT is four times faster than the conventional method which accomplishes the same task via inverse DCT (IDCT) [D.2.12]. The Speeded Up Robust Feature (SURF) is a scale and rotation invariant detector and descriptor algorithm proposed by Bay et al. [D.2.9], that can be used in computer vision tasks such as object recognition. SURF algorithm is similar to the SIFT algorithm proposed by Lowe [D.2.3], although it presents notable differences such as considerably higher calculation's speed without causing a loss of performance and major robustness against different types of geometric and photometric transformations. The second stage of the proposed CBVR technique consists to extract the SURF descriptor, to each down-sampled *I*-luminance frame, then every SURF descriptor of *N-th* down-sampled *I*-luminance frame of *i-th* video is store it in the descriptor DB as $d_{N,i}$. It is important to note that the computational cost of the SURF algorithm is dependent of the size image, so according to the previous procedure, the SURF algorithm works with a frame of 176 x 144 pixels (QCIF format), which is twice faster than CIF format. First and second stages are performed in order to build the whole descriptors database; this is a one-pass and off-line process (solid line in Fig. D2.1). The proposed technique can accept images and video as query multimedia content: if the query content is a video first we need to extract the down-sampled I-luminance frames and then extracting their respective SURF descriptor. If instead a video the query content is an image, only the feature extraction stage is performed (dotted line in Fig. D2.1). Once that the SURF descriptor is obtained, the indexing by object matching is carried out as follows: 1) The object match operation between the query frame and each frame descriptor in the database is performed using the Euclidean distance, defined by (2):

$$Ed = \sqrt{\sum_{i=1}^{K} (d_{N,i} - dq)^2} \tag{2}$$

where $d_{N,i}$ is the descriptor of *N-th* down-sampled *I*-luminance frame of *i-th* video in the descriptors database, *dq* is the SURF descriptor of the query frame and *K* is the length of the SURF descriptor, which is 64 [D.2.2]. Once that the Euclidean distances *Ed* are obtained, those are allowed and sorted from lowest to highest order in an array denoted as *E*. 2) To know if the reference frame is related with the content of the query frame, the first twenty minimum Euclidean distances from the array *E* should satisfy the follow condition given by (3):

$$E(1) < Th \,\&\, E(2) < Th \ldots \,\&\, E(20) < Th \tag{3}$$

where *Th* is a pre-defined threshold and the symbol *&* denotes the logical "and" operation. If the reference frame satisfies the above condition, its index is stored in an array *I*, otherwise, the reference frame is discarded. At least one reference frame among the *N* down-sampled *I*-luminance frames of *i-th* video must satisfies the above condition in order to be considered as part of the retrieval videos. 3) Finally, the related content videos are queried and retrieved using the indexes I_r from the video database.

To evaluate the performance of the proposed CBVR technique, we use a video database composed of 100 video extracted randomly from YouTube web site with Creative Commons licensing [D.2.13]. These videos contain four groups related to the following representative Mexican Culture sites: Teotihuacan (Pyramid of the Sun), Teotihuacan (Avenue of the Dead), La Venta (Head Olmeca) and Templo Mayor (Stone of the Sun). Each group is composed of 10 videos with the same Mexican Culture site captured in highly diverse environmental conditions, and with different acquisition devices. As query multimedia content we choose two query frames (Fig. D2.2(a) and Fig. D2.2(b)) related to Teotihuacan (Pyramid of the Sun and Avenue of the Dead) and two query videos (Fig. D2.2(c) and Fig. D2.2(d)) related to La Venta (Head Olmeca) and Templo Mayor (Stone of the Sun). The rest of the videos (96) are related with other content types, such as landscapes of rivers, mountains, beaches, etc., as well as celebrity's portraits, cartoons, architectonical structures, flags, etc. Our experiments are carried out on a personal computer running win7© with an Intel© Corei5 processor (2.50 GHz) and 6GB RAM while the CBVR technique is implemented on Matlab© R2013a. The SURF parameters used in this work are Hessian response threshold = 0.0001, a number of octaves = 5 and a number of filters per octave = 2.

Figure D2.2. Mexican Culture multimedia content used in the experiments: (a) Frame of Teotihuacan (Pyramid of the Sun), (b) Frame of Teotihuacan (Avenue of the Dead), (c) First frame of video sequence La Venta (Head Olmeca) and (d) First frame of video sequence Templo Mayor (Stone of the Sun).

The performance measurement is based on the quantity of video retrieved successfully given a query frame. In this context the most commonly measurements used in the literature are precision and recall, which are given by (4) and (5), respectively [D.2.7], [D.2.8]:

$$Precision = \frac{Number\ of\ relevant\ videos\ retrieved}{Total\ number\ of\ videos\ retrieved} \tag{4}$$

$$Recall = \frac{Number\ of\ relevant\ videos\ retrieved}{Total\ number\ of\ relevant\ videos\ in\ database} \tag{5}$$

where the total number of relevant videos in the video database for each group is 10. In order to obtain the most adequate threshold *Th* for each Mexican Culture site, we obtain the

precision and recall measurements varying the *Th* values from 0.05 to 0.15. Fig. D2.3 shows the precision and recall values for each content type with different threshold values.

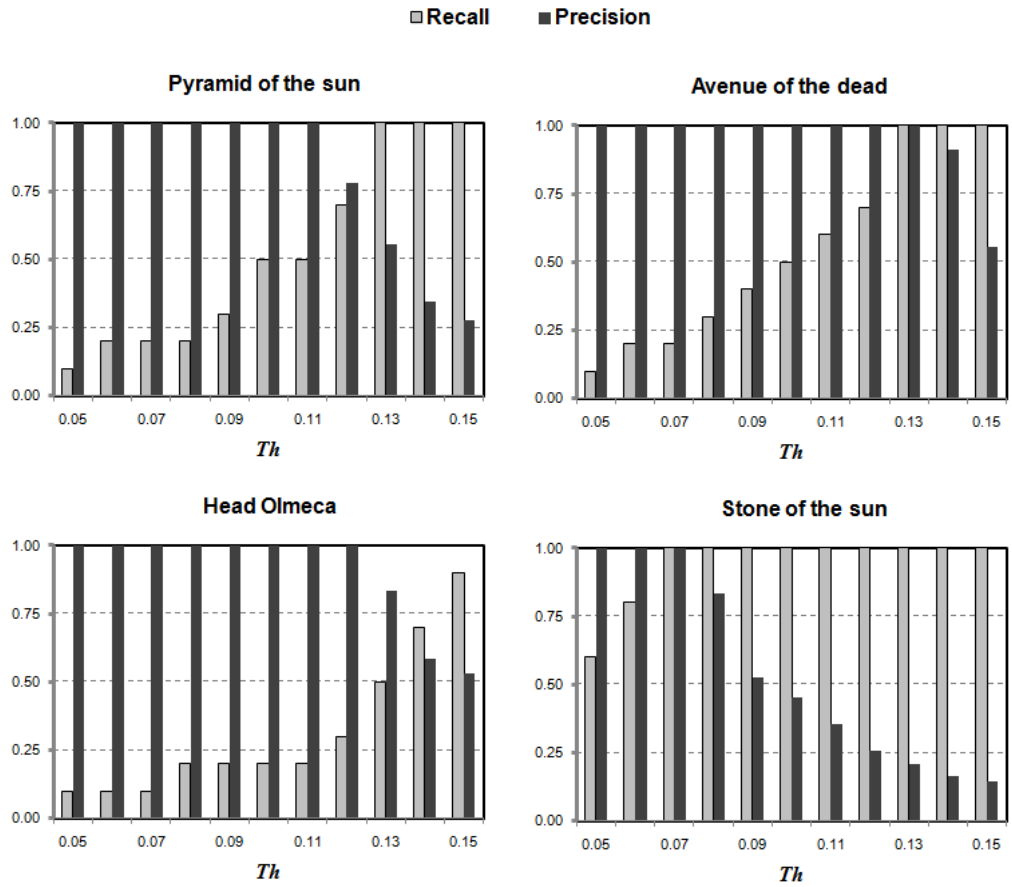


Figure D2.3. Precision and Recall values for (a) Teotihuacan (Pyramid of the Sun), (b) Teotihuacan (Avenue of the Dead), (c) La Venta (Head Olmeca) and (d) Templo Mayor (Stone of the Sun).

With the aim to obtain the most accuracy in precision terms and at the same time avoid as much as possible the false object matches (videos that are not related to the content of the query content), from Fig. D2.3 (a) we can observe how the value *Th*=0.124 offers the major precision for video retrieval of Teotihuacan (Pyramid of the Sun). On the other hand, from Fig. D2.3 (b) we calculate the value *Th*=0.130 which provides the best performance for video retrieval (recall and precision equals to one) of Teotihuacan (Avenue of the Dead). From Fig. D2.3(c) the value *Th*=0.138 provides the best balance between recall and precision for La Venta (Head Olmeca) and finally, from Fig. D2.3 (d), a value *Th*=0.070 offers again the best performance for video retrieval of Templo Mayor (Stone of the sun). Note that the above values of *Th* provide the most exact results as much as possible, however, from Fig. D2.3 we can observe how the value of the *Th* may be adjusted in order to build other filter that includes several results sorted by most relevant content. Figures D2.4 and D2.5 show the retrieved videos for Mexican Culture videos Teotihuacan (Pyramid of the Sun) and Templo Mayor (Stone of the sun), respectively.

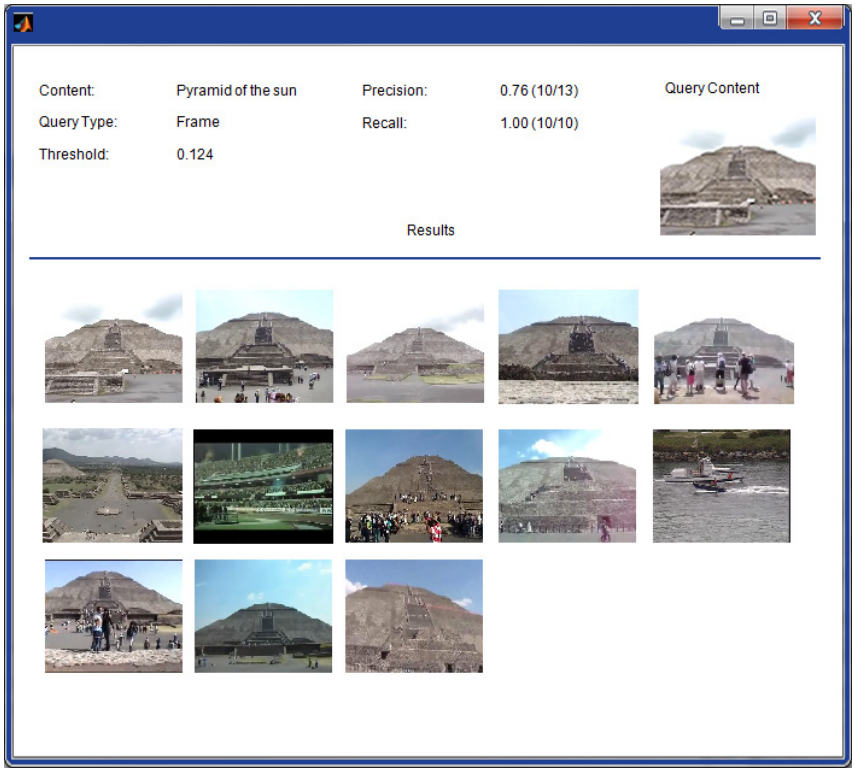


Figure D2.4. Retrieved videos for Teotihuacan (Pyramid of the sun).

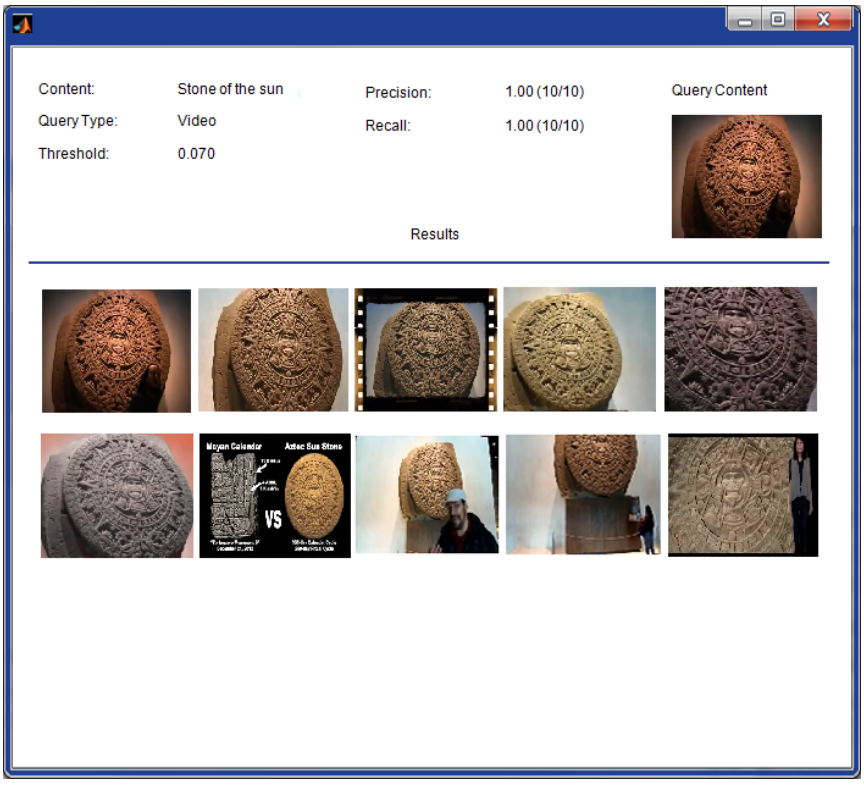


Figure D2.5. Retrieved videos for Templo Mayor (Stone of the sun).

D.3 PROPOSED METHOD BASED ON GLOBAL-LOCAL DESCRIPTOR

Nowadays, the amount of multimedia data has increased rapidly, this is because of many reasons, and one could be digital cameras and mobile phones. One easily can take a picture or capture a video and due to the high-speed Internet connections and a bunch of social networks, we can easily upload our multimedia data and share them with friends. That means that image and video databases has increased considerably making difficult to index and classify multimedia data. The first generation of image retrieval systems developed in late 1970's, was mainly linked to text retrieval [D.3.1]. This traditional manner of image retrieval heavily relies on manual labor to label images with keywords, which unfortunately can hardly describe the diversity and ambiguity of image contents [C.3.3]. The keywords which labeled the multimedia data is called metadata and is subjective because depends on the person who create it. The problem of using metadata is that different users may use different keywords for the same multimedia content [D.3.2].

We proposed a descriptor which combines the global and local color features in order to obtain better results on image retrieval. The figure D3.1 shows the diagram block of the proposed descriptor.

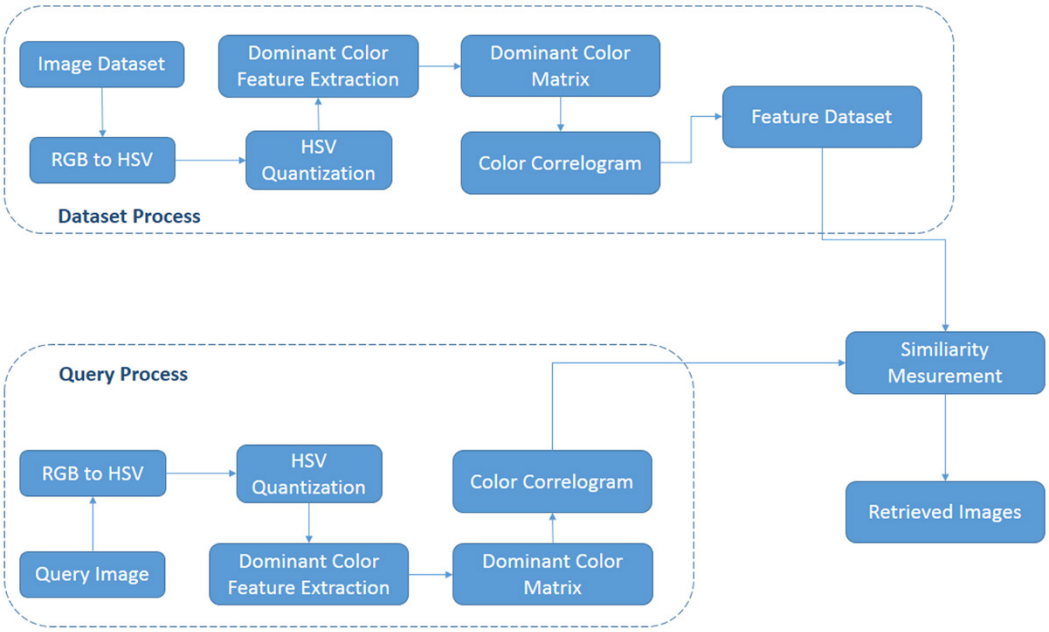


Figure D3.1. Proposed Image Retrieval Scheme.

In order to evaluate the proposed method for content-based image retrieval, we use the metrics Average Retrieval Rate (ARR) which determines if the algorithm is good or not, taking into account the images successfully retrieved. Average Retrieval Precision (ARP) is another metric used and is similar to ARR. One last metric is Average Normalized Modified Retrieval Rank (ANMRR) which takes into account the images successfully retrieved as well as its rank, and, the images that were not successfully retrieved are penalized, thus, the evaluation of the algorithm is very strict because it is taken into account the hits and the misses. The results shows that the proposed scheme has a better retrieval performance than the algorithms studied from the literature.

One the most used scheme for image retrieval is the Histogram Intersection, which was proposed by Swain and Ballard [D.3.3]. This scheme works with color histograms, which are calculated defining n bins and once the color histograms are calculated the histogram intersection is the minimum value between the color histogram of the query image and the color histogram of the data base images. This scheme is also used as a similarity measurement [D.3.4]. Dominant Color Descriptor (DCD) is another global color descriptor [C.3.2, C.3.4, C.3.5] which provides an effective, compact, and intuitive salient color representation it also describes the color distribution in an image [C.3.5] and contains two main components: representative colors and the percentage of each color [C.3.3]. DCD is not consistent with human perception and in [C.3.3, C.3.5] author used a new and efficient scheme to address this problem: The RGB (Red Green Blue) color space is divided into 8 coarse partitions where all the colors located in the same partition will be consider as one and as a result, it is obtained a set of dominant colors, and the final number of dominant colors is constrained to 4-5 on average. This algorithm proposed by [C.3.5] is called Linear Block Algorithm (LBA). This LBA algorithm is a dynamic method of quantization, because the final set of dominant colors depend of color distribution of the image. In [C.3.4] author used a static algorithm to quantize the images, which consists of converting the images from RGB color space to HSV color space, then, each component of HSV is quantized statically: the H component is divided into 8 intervals, each of which correspond to one of the main hues (red, blue yellow, green, purple, orange, dark blue and violet) and S and were divided into 3 intervals each (3 levels for saturation and 3 levels for value). As a result, it is obtained 8x3x3=72 dominant colors. As we mentioned before, this is a static method of quantization and the number of dominant colors will be always 72 because it does not depend on the image as in LBA algorithm. Histogram intersection and Dominant Color Descriptor are both global color descriptors. The advantage of these descriptors is the low computational cost but the disadvantage is they lack of spatial information.

Color correlogram is a local color descriptor and compute the correlation between pairs of color. It was proposed by [D.3.5] and is defined as: Given any pixel of color c_i in the image, color correlogram gives de probability that a pixel at distance k away from the given pixel is of color c_j. The autocorrelogram of an image captures spatial correlation between identical colors only [D.3.5]. The computational cost of color correlogram is high compared to histogram intersection and dominant color descriptor. Color Layout Descriptor (CLD) is a local color descriptor too. CLD is a numeric quantity that describes a color feature of an image and is a very compact and resolution-invariant representation of color for high-speed image retrieval and was designed to efficiently represent the spatial distribution of colors [D.3.6]. The extraction of the descriptor consists of four stages: image partitioning, dominant color selection, DCT transform and non-linear quantization of the zigzag-scanned DCT coefficients [D.3.7].

D.3.1 RGB TO HSV COLOR SPACE CONVERSION

The color space of a given input image is converted into HSV color space. If the original color space is the common RGB, it can be transformed by:

$$H = \begin{cases} 60° \times \left(\dfrac{G-B}{\Delta} (\text{mod } 6) \right), & \text{if } C_{\max} = R \\ 60° \times \left(\dfrac{B-R}{\Delta} + 2 \right), & \text{if } C_{\max} = G \\ 60° \times \left(\dfrac{R-G}{\Delta} + 4 \right), & \text{if } C_{\max} = B \end{cases} \quad (1)$$

$$S = \begin{cases} 0, & \text{if } \Delta = 0 \\ \dfrac{\Delta}{C_{\max}}, & \text{if } \Delta \neq 0 \end{cases} \quad (2)$$

$$V = C_{\max} \quad (3)$$

where (R,G,B) is the normalized RGB color components,

$$\Delta = C_{\max} - C_{\min}, C_{\max} = \max(R,G,B) \text{ and } C_{\min} = \min(R,G,B).$$

We chose the HSV color space because it is very similar in the way humans perceive colors. The first thing we perceive is the Hue (H), in other words, when we see a red car, the first thing we perceive is the red hue. The second thing we perceive is the saturation, which means that if the car is pure red or pastel red. The last thing we distinguish is the brightness, the red car is brighter in the day and dark in the night.

D.3.2 HSV Quantization

Once the image is converted from RGB to HSV color space, it is necessary to quantize the colors in order to minimize the amount of information, thus, the computational cost is lower. As in [C.3.4, D.3.8], we used a non-interval quantization algorithm of HSV color space. The algorithm is as follows:

1. From HSV, we divided H into 8 shares, S and V are divided into 3 shares each.
2. For H (Hue), we divided into 8 shares, the 8 main hues as follows:

$$H = \begin{cases} 0 \; if \; h & \in [316,20) \\ 1 \; if \; h & \in [20,40) \\ 2 \; if \; h & \in [40,75) \\ 3 \; if \; h & \in [75,155) \\ 4 \; if \; h & \in [155,190) \\ 5 \; if \; h & \in [190,270) \\ 6 \; if \; h & \in [270,295) \\ 7 \; if \; h & \in [295,316) \end{cases} \quad (4)$$

Each interval of H represent a hue, for example, the interval [316, 20) represents the red range. The other intervals represent the orange, yellow, green, blue, dark blue, purple and violet [D.3.9] according to HSV model.

3. The S and V component of HSV color space are divided into 3 shares. As we mentioned before, S is the saturation of the color and V the brightness. These components are divided into 3 levels only.

$$S = \begin{cases} 0 \; if \; s \; \in [0,0.2] \\ 1 \; if \; s \; \in (0.2,0.7] \\ 2 \; if \; s \; \in (0.7,] \end{cases} \quad (5)$$

$$\mathrm{V} = \begin{cases} 0 \; if \; v \; \in [0,0.2] \\ 1 \; if \; v \; \in (0.2,0.7] \\ 2 \; if \; v \; \in (0.7,] \end{cases} \quad (6)$$

4. Once the 3 components of HSV color space are divided into shares then they are combined as follows:

$$C = 9 \times H + 3 \times S + V \quad (7)$$

5. As a result, we obtained 8x3x3=72 different colors only.

D.3.3 Dominant Color Feature Extraction

The dominant color descriptor (DCD) in MPEG-7 is defined as:

$$F = \{\{c_i, p_i\}, i = 1, \dots, N\} \quad (8)$$

where N is the total number of dominant colors for an image, c_i is a 3-D dominant color vector, p_i is the percentage for each dominant color, and the sum of p_i is equal to 1 [C.3.3, C.3.2, C.3.4, C.3.5]. As in [C.3.4], the algorithm for dominant color feature extraction is as follows:

1. It is calculated a color histogram of the vector $C = 9 \times H + 3 \times S + V$ with $P_i(i=0,1,\dots,71)$ expressing proportion of color vector i.
2. A new vector Q_j is defined expressing the percentage in descending order to P_i.
3. The M first dominant colors are considered only:

$$P_i = \begin{cases} P_i & P_i = Q_j \\ 0 & P_i \neq Q_j \end{cases} \quad (9)$$

$$i = 0,1, \dots, 71$$
$$j = 0,1, \dots, M-1$$

4. The first M dominant colors are then normalized:

$$P'_i = \frac{P_i}{\sum_{j=0}^{M-1} Q_j} \quad (10)$$
$$j = 0,1, \dots, M-1$$

5. As a result, we obtain M dominant colors and their percentage:

$$F = \{C_i, P_i\} \quad i = 0,1,\ldots,71 \tag{11}$$

D.3.4 Dominant Color Matrix

The Dominant Color Matrix (DCM) is a 2-D vector which contains the dominant colors only instead of all the colors as in [D.3.8, D.3.9]. The algorithm is a follows:

1. Once we know what the first M dominant colors are, we considered only those colors for generating the Dominant Color Matrix.
2. We generate the Dominant Color Matrix as follows:

$$DCM_{i,j} = \begin{cases} C_{i,j} & If\ C_{i,j}\ is\ a\ dominant\ color \\ -1 & If\ C_{i,j}\ is\ a\ dominant\ color \end{cases} \tag{12}$$

   DCM is a mxn matrix, where mxn are the dimension of the image.

3. The coefficients of DCM are the first M dominant colors and the non-dominant colors are represented as -1 [D.3.9].

D.3.5 Color Auto-Correlogram Calculation

So far, the global features has been calculated from section D.3.1 to D.3.4. In this section, the local color features are computed using color auto-correlogram. Color correlogram is defined as:
Given any pixel of color c_i, color correlogram gives the probability that a pixel at distance k away from the given pixel is of color c_j. In this work, we computed auto-correlogram, which means that both pixels, c_i and c_j are of the same color. Color correlogram is define by the equation 13:

$$\gamma_{c_i c_j}^{(k)}(I) \triangleq Pr_{p_1 \in I_{c_i}, p_2 \in I_{c_j}}[p_2 \in I_{c_j} || p_1 - p_2| = k] \tag{13}$$

The algorithm is a follows:

1. From the DCM matrix, one pixel is taken as the given pixel as in figure D3.2.
2. At *k* distance away from the given pixel, the neighbors pixels are scanned in order to find a pixel p_2 of the same color of p_1. In case of *k=1*, there are 8 neighbors pixels.
3. We moved through the pixels of the DCM matrix in order to find a new pixel p_1 of the same color of the last pixel p_1 and again, around it, we try to find a new pixel p_2 of the same color as in figure D3.3.
4. The color auto-correlogram of the first dominant color is calculated by the equation 14:

-1	4	-1	2	3	3	4	3	2	2	-1	-1	2	2	2
2	6	3	2	2	3	3	4	6	4	5	6	3	2	2
3	5	4	-1	4	5	5	5	5	7	7	7	5	3	2
2	3	3	2	3	3	4	4	3	6	6	5	5	4	4
2	3	4	-1	0	3	4	4	2	4	4	4	4	3	3
2	3	4	-1	0	0	2	2	0	2	2	2	2	2	2
3	0	3	2	-1	-1	2	2	3	0	3	3	3	2	2
3	2	3	2	-1	-1	3	3	4	4	4	3	3	3	2
3	2	3	3	2	2	2	2	4	5	5	5	5	2	3
4	2	3	3	2	2	2	2	3	2	3	5	5	3	3
4	3	2	2	2	0	2	2	3	-1	3	5	4	4	3
4	3	2	2	2	-1	2	2	3	0	3	5	4	4	4
5	3	2	-1	0	-1	2	2	3	-1	3	6	4	5	4
6	7	7	-1	0	-1	2	2	3	2	3	6	4	5	4
6	7	7	-1	-1	-1	2	2	3	2	4	6	5	5	5

Figure D3.2. Selection of the first pixel p_1 of color c_i=4. In the figure a pixel p_1 of color c_i=4 is taken as sample, and around it at k=1 distance there are not pixels of the same color, which means there are not any pixel p_2.

-1	4	-1	2	3	3	4	3	2	2	-1	-1	2	2	2
2	6	3	2	2	3	3	4	6	4	5	6	3	2	2
3	5	4	-1	4	5	5	5	5	7	7	7	5	3	2
2	3	3	2	3	3	4	4	3	6	6	5	5	4	4
2	3	4	-1	0	3	4	4	2	4	4	4	4	3	3
2	3	4	-1	0	0	2	2	0	2	2	2	2	2	2
3	0	3	2	-1	-1	2	2	3	0	3	3	3	2	2
3	2	3	2	-1	-1	3	3	4	4	4	3	3	3	2
3	2	3	3	2	2	2	2	4	5	5	5	5	2	3
4	2	3	3	2	2	2	2	3	2	3	5	5	3	3
4	3	2	2	2	0	2	2	3	-1	3	5	4	4	3
4	3	2	2	2	-1	2	2	3	0	3	5	4	4	4
5	3	2	-1	0	-1	2	2	3	-1	3	6	4	5	4
6	7	7	-1	0	-1	2	2	3	2	3	6	4	5	4
6	7	7	-1	-1	-1	2	2	3	2	4	6	5	5	5

Figure D3.3. Scanning of DCM matrix. The DCM is scanned in order to find how many pixels p_1 exists in the image as well as the pixels p_2.

$$Color\ Autocorrelogram = \frac{Total\ number\ of\ pixels\ p_1}{Total\ number\ of\ pixels\ p_2 \times Number\ of\ Neighbors \times k} \qquad (14)$$

From figure D3.2 and D3.3, we have:

$$Color\ Autocorrelogram = \frac{35}{66 \times 8 \times 1} = 0.0663 = 6.63\ \%$$

Which means that the probability of finding a pixel p_2 of color c_j=4 at distance k=1 away from a pixel p_1 of color c_i=4 is 6.63%.

5. We did the same for each dominant color.

6. We did not consider the pixels of color -1 because they represent the non-dominant colors.
7. As a result, it is obtained a vector of length 72 (the total number of dominant colors).

D.3.6 Similarity Measurement

A similarity measurement is normally defined as a metric distance [D.3.4]. In order to know which images are more similar to the query image, it is necessary to measure the distance between the color descriptors. A large distance means that images are different and a short distance indicates the images are similar.

D.3.7 Minkowsky-form Distance

The Minkowsky-form distance is defined based on the L_p norm [D.3.4]:

$$d_p(\boldsymbol{Q},\boldsymbol{T}) = \left(\sum_{i=0}^{N-1}(Q_i - T_i)^p\right)^{\frac{1}{p}} \quad (15)$$

where $\boldsymbol{Q} = \{Q_0, Q_1, \dots, Q_{N-1}\}$ and $\boldsymbol{T} = \{T_0, T_1, \dots, T_{N-1}\}$ are the query and target feature vectors respectively. When $p=1$, $d_1(Q,T)$ is the city block distance or Manhattan distance (L_1):

$$d_1(\boldsymbol{Q},\boldsymbol{T}) = \sum_{i=0}^{N-1}|Q_i - T_i| \quad (16)$$

We used the normalized L_1 distance in order to generalize all type of images. The Normalized L_1 is defines as:

$$d_1(\boldsymbol{Q},\boldsymbol{T}) = \sum_{i=0}^{N-1}\frac{|Q_i - T_i|}{1+Q_i+T_i} \quad (17)$$

D.3.8 Results

We used the proposed color descriptor in order to retrieve the most similar image to an image query from a dataset. We used two databases, one of them (dataset 1) is a dataset composed by 500 images from COREL dataset divided into 25 categories with 20 ground truth images per category. The categories are: Gorilla, owl, tiger, wolf_snow, wolf_lawn, horse, eagle, drink, fight, firework, dolphin, leaves, balloon, cards, roses, car, food, bus, clouds, ships, dino, duck, texture 1, egg and sunset

The second dataset (dataset 2) is composed by 1000 images from COREL dataset divided into 20 categories with 50 ground truth images per category. The categories are: Fireworks, lion, horse, eagle, clouds, fight, leaves, sunset, bonsai, rock form, mineral, cards, flags, roses, door, egg, dino, duck, food and texture 2.

D.3.9 Performace measurement

We evaluated the proposed color descriptor as well as the ones from the literature as Histogram Intersection, Color Correlogram, Dominant Color Descriptor and Color Layout Descriptor. The evaluation was done using performance measure metrics as ARR (Average Retieval Rate), ARP (Average Retrieval Precision) and ANMRR (Average Normalized Modified Retrieval Rank). These metrics are described below:

D.3.10 ARR (AVERAGE RETRIEVAL RATE)

The Average Retrieval Rate (ARR) [C.3.2] can be computed as:

$$ARR = \frac{1}{NQ}\sum_{q=1}^{NQ} RR(q) \leq 1 \quad (18)$$

where $N_Q$represents the number of queries that are used for the purpose of verifying the descriptor in certain dataset. RR is the retrieval rate of a single query. RR can be calculated as:

$$RR(q) = \frac{N_R(\alpha,q)}{N_G(q)} \leq 1 \quad (19)$$

where $N_G(q)$ is the number of ground truth images of a query q. $N_R(q)$ indicates the number of the relevant images found in the first $\alpha \times N_G(q)$ images. α Should be more than or equal to 1. In this paper we set $\alpha = 1$ and $\alpha = 2$. The number of ground truth for the dataset 1 is 20. There is something important to take into account, the number of queries should be at least 1% of the dataset size. In the case of the dataset 1, we used 13 images as queries which is the 2.6%. The results for ARR are shown in table D3.1. For the dataset 2, we used 20 images as queries, which is the 2% of the dataset size. The best value for ARR is 1. The results for ARR are shown in the table D3.2

Table D3.1. Evaluation of retrieval performance using the dataset 1 (500 images).

Algorithm	**ARR**	
	$\boldsymbol{\alpha = 1}$	$\boldsymbol{\alpha = 2}$
Color Auto-Correlogram	0.5923	0.7577
Histogram Intersection	0.6269	0.8115
Dominant Color Descriptor (using non-interval algorithm)	0.6154	0.8154
Dominant Color Descriptor (LBA)	0.5642	0.6808
Color Layout Descriptor	0.5731	0.7385
Proposed Color Descriptor	0.7000	0.8269

Table D3.2. Evaluation of retrieval performance using the dataset 2 (1000 images).

Algorithm	**ARR**	
	$\boldsymbol{\alpha = 1}$	$\boldsymbol{\alpha = 2}$
Color Auto-Correlogram	0.5870	0.7200
Histogram Intersection	0.5760	0.7610
Dominant Color Descriptor (using non-interval algorithm)	0.5590	0.7420
Dominant Color Descriptor (LBA)	0.5500	0.7320
Color Layout Descriptor	0.5740	0.7620
Proposed Color Descriptor	0.5920	0.7580

D.3.11 AVERAGE RETRIEVAL PRECISION (ARP)

The ARP metric is defined as:

$$ARP = \frac{1}{N_Q}\sum_{q=1}^{N_Q} RP(q) \quad (20)$$

where N_Q represent the number of queries and $RP(q)$ indicates the precision of image retrieval of a single query q and is defined as:

$$RP(q) = \frac{N_s(q)}{N_R(\alpha,q)} \quad (21)$$

where $N_R(\alpha, q)$ indicates the number of retrieved images and it is calculated as:

$$\alpha \times N_G(q) \quad (22)$$

where $N_G(q)$ is the number of ground truth images and α indicates a rate between the number of retrieved images and $N_G(q)$. Here, α takes value of [0.25, 0.5, 1] and finally $N_s(q)$ is the number relevant inages retrieved in $N_R(q)$. The best value for ARP is 1. The results of ARP for dataset 1 and dataset 2 are shown in the table D3.3 and D3.4 respectively.

Table D3.3. Evaluation of retrieval performance using the dataset 1 (500 images).

Algorithm	**ARP**		
	$\boldsymbol{\alpha = 1}$	$\boldsymbol{\alpha = 0.5}$	$\boldsymbol{\alpha = 0.25}$
Color Auto-Correlogram	0.5923	0.7846	0.8923
Histogram Intersection	0.6229	0.7923	0.9077
Dominant Color Descriptor (using non-interval algorithm)	0.6154	0.7846	0.8615
Dominant Color Descriptor (LBA)	0.5642	0.7154	0.8154
Color Layout Descriptor	0.5731	0.7154	0.8000
Proposed Color Descriptor	0.7000	0.8538	0.9385

Table D3.4. Evaluation of retrieval performance using the dataset 2 (1000 images).

Algorithm	**ARP**		
	$\boldsymbol{\alpha = 1}$	$\boldsymbol{\alpha = 0.5}$	$\boldsymbol{\alpha = 0.25}$
Color Auto-Correlogram	0.5870	0.7620	0.8538
Histogram Intersection	0.5760	0.7380	0.8308
Dominant Color Descriptor (using non-interval algorithm)	0.5590	0.6920	0.8154
Dominant Color Descriptor (LBA)	0.5500	0.7040	0.8077
Color Layout Descriptor	0.5740	0.7280	0.8038
Proposed Color Descriptor	0.5920	0.7400	0.8692

D.3.12 AVERAGE NORMALIZED MODIFIED RETRIEVAL RANK

This is the last metric used here in order to evaluate the proposed color descriptor as well as the ones from the literature. The Average Normalized Modified Retrieval Rank is considered as one of the most accurate metrics used in CBIR, because it combines many conventional

metrics, which are hit-miss counters, precision-recall and ranking information [C.3.2]. ANMRR can be calculated as follows:

$$ANMRR = \frac{1}{N_Q}\sum_{q=1}^{N_Q} NMRR(q) \leq 1 \quad (23)$$

where NMRR is computed as:

$$NMRR = \frac{2\times AVR - N_G(q) - 1}{2\times W - N_G(q) + 1} \quad (24)$$

where $N_G(q)$is the number of ground truth images, W is a window of retrieved images, we set $W = 2 \times N_G(q)$ [C.3.5] that means that we retrieved W images only. AVR (Average Rank) is computed as follows:

$$AVR(q) = \frac{1}{N_G(q)}\sum_{k=1}^{N_G(q)} R(k) \quad (25)$$

where $R(k)$ is the rank of each ground truth image of a query q in the first W images. The relevant images that do not appear in the window W are penalized with $R(k) = W + 1$ as follows

$$R(k) = \begin{cases} R(k) & If\ the\ ground\ truth\ image\ appears\ in\ W \\ W+1 & If\ the\ ground\ truth\ image\ does\ not\ appear\ in\ W \end{cases} \quad (26)$$

The best value for ANMRR is 0. The results of ANMRR for dataset 1 and dataset 2 are shown in the table D3.5 and D3.6 respectively.

Table D3.5. Evaluation of retrieval performance using the dataset 1 (500 images).

Algorithm	ANMRR
Color Auto-Correlogram	0.3126
Histogram Intersection	0.2507
Dominant Color Descriptor (using non-interval algorithm)	0.2576
Dominant Color Descriptor (LBA)	0.3579
Color Layout Descriptor	0.3378
Proposed Color Descriptor	0.2129

Table D3.6. Evaluation of retrieval performance using the dataset 2 (1000 images).

Algorithm	ANMRR
Color Auto-Correlogram	0.3228
Histogram Intersection	0.3174
Dominant Color Descriptor (using non-interval algorithm)	0.3384
Dominant Color Descriptor (LBA)	0.3478
Color Layout Descriptor	0.3194
Proposed Color Descriptor	0.3094

In the figures D3.4, D3.5, D3.6, D3.7, D3.8 and D3.9 there are 15 retrieved images from dataset 1 using the algorithms from the literature and the proposed scheme are shown, and it is observed that the proposed algorithm improve the image retrieval. The same happened for the second dataset, the image retrieval task is improve combining the local and global color features. The results for the dataset 2 is shown in the figures D3.10, D3.11, D3.12, D3.13, D3.14 and D3.15.

Figure D3.4. Retrieved images by Color Auto-correlogram from dataset 1. The first 15 retrieved images from dataset 1.

Figure D3.5. Retrieved images by Histogram Intersection from dataset 1. The first 15 retrieved images from dataset 1.

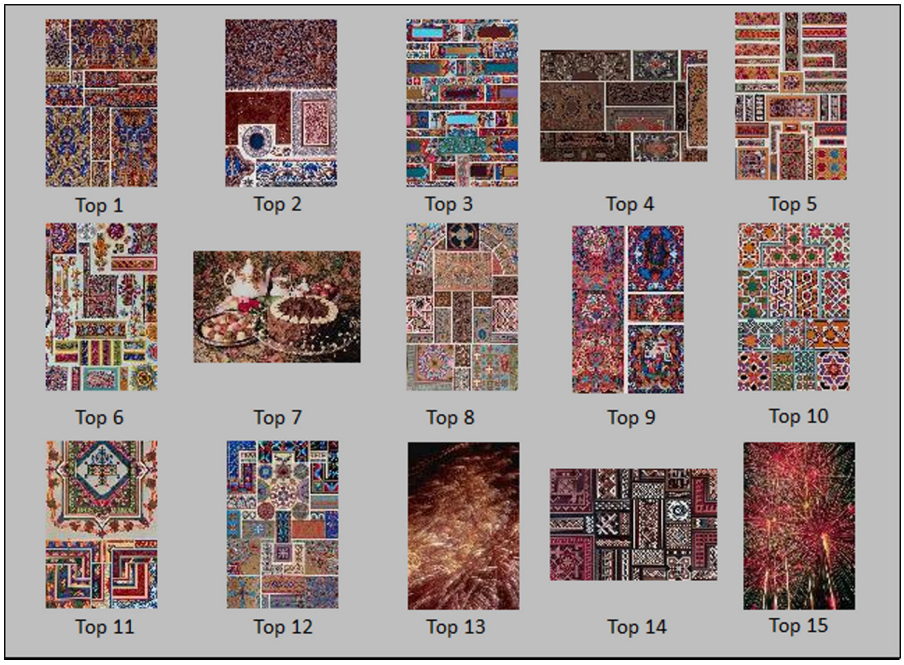


Figure D3.6. Retrieved images by Dominant Color Descriptor using non-interval algorithm from dataset 1. The first 15 retrieved images from dataset 1.

Figure D3.7. Retrieved images by Dominant Color Descriptor using LBA algorithm from dataset 1. The first 15 retrieved images from dataset 1.

Figure D3.8. Retrieved images by Color Layout Descriptor from dataset 1. The first 15 retrieved images from dataset 1.

Figure D3.9. Retrieved images by The Proposed Algorithm from dataset 1. The first 15 retrieved images from dataset 1.

Figure D3.10. Retrieved images by Color Auto-correlogram Algorithm from dataset 2. The first 15 retrieved images from dataset 2.

Figure D3.11. Retrieved images by Histogram Intersection from dataset 2. The first 15 retrieved images from dataset 2.

Figure D3.12. Retrieved images by Dominant Color Descriptor using non-interval algorithm from dataset 2. The first 15 retrieved images from dataset 2.

Figure D3.13. Retrieved images by Dominant Color Descriptor using LBA algorithm from dataset 2. The first 15 retrieved images from dataset 2.

Figure D3.14. Retrieved images by Color Layout Descriptor from dataset 2. The first 15 retrieved images from dataset 2.

Figure D3.15. Retrieved images by The Proposed Algorithm from dataset 2. The first 15 retrieved images from dataset 2.

D.4 Proposed Method for Segmentation in video scenes and Representative video Frames

The algorithm used is the analysis of fractal binary partition partitioned into random segments.

The algorithm consists of:

- Perform binary partition.
- Given a time duration T video and a range r. A table T / 2 (t1), one in T / 4 (t2) and one in 3T / 4 (t3) is taken, taking into account the ranges given by r. These ranges are between t1 - r and t1 + r. It happens the same case with t2 and t3.
- Comparison pictures. t1 to t2 and t3 to t1, through such images PSNR.
- If the PSNR is greater than a threshold, then a binary partition is performed and analyzed, making the duration T t1 in the case of being t2, and t1 being the starting time t3 should be.
- Recursively algorithm repeats.

The range r, is determined by a random percentage between 1% and 5% of the length of the video. The format of this video is MPEG-4 (mp4).

This algorithm gives us the segmentation in video scenes and representative video frames are obtained.
Was taken as a basis for comparison to the FFMPEG library, having the option to process video and obtain representative scenes video.
The Figure D4.1 shows the processing time of each of the algorithms to obtain the representative video frames. The base consists of 17 test videos with different durations.

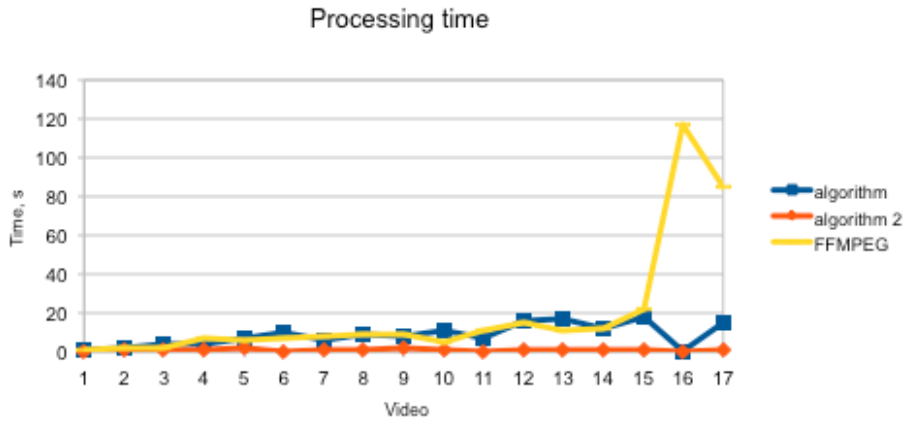


Figure D4.1. Processing time for obtaining representative frames of the 17 test videos.

The graph in Figure D4.1 represents the processing time performance of the original algorithm (algorithm); this algorithm performs processing under the scheme of memory management data on the hard disk of the computer. An optimization algorithm to realize this by making the handling of information in RAM (Algorithm 2).

The graph in Figure D4.2 shows the number of representative frames of video obtained by each algorithm.

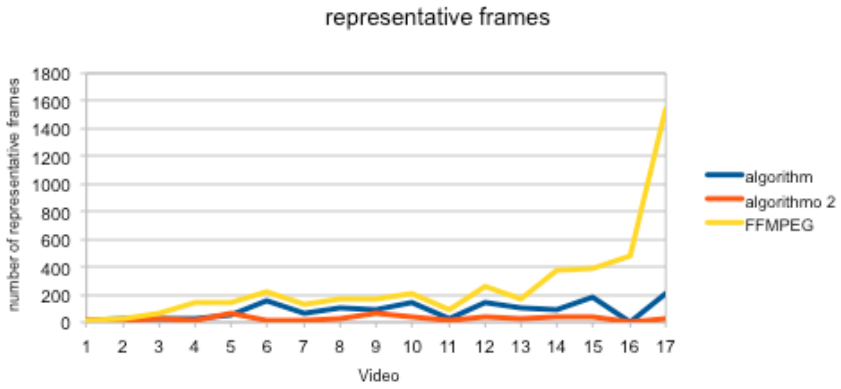


Figure D4.2. Representative video frames.

The graph in Figure D4.3 shows the number of repeated frames or very similar in content to have for each video on the analysis of each of the algorithms.

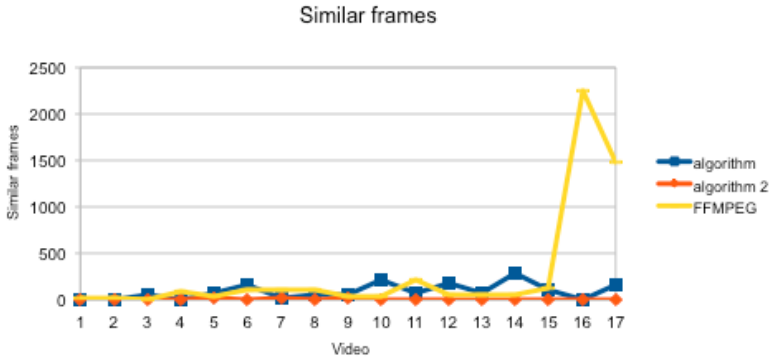


Figure D4.3. Repeated similar in content or pictures.

D.5 Color and Texture Descriptors for CBIR

The figure D5.1 shows the CBIR system diagram based on descriptors of color and texture.

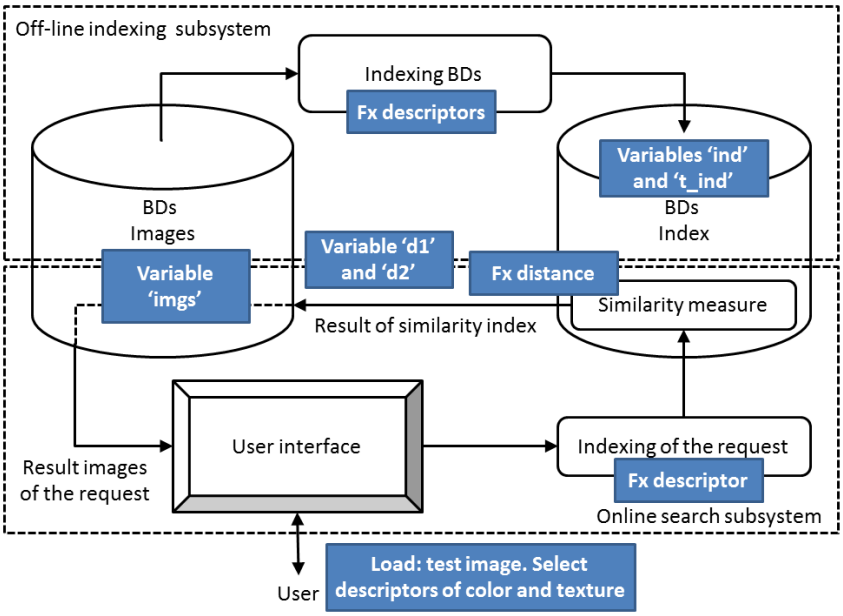


Figure D5.1. CBIR System.

In a offline phase of the system, the descriptors of color and texture of the database (images) are computed. Then, a index base structure which contains the feature vectors of the database of the images visual content is generated. The figure D5.2 shows the implemented system interface. Through the interface, a test image is selected and visual descriptors of color and texture are chosen to find similar images. The similarity measure between descriptors is based on normalized Euclidean distance. The tests of this implementation were performed with a very small database. The main objective is to perform the implementation of the descriptors and view its performance. Those descriptors with the best performing will be implemented on the of Mex-Culture platform.

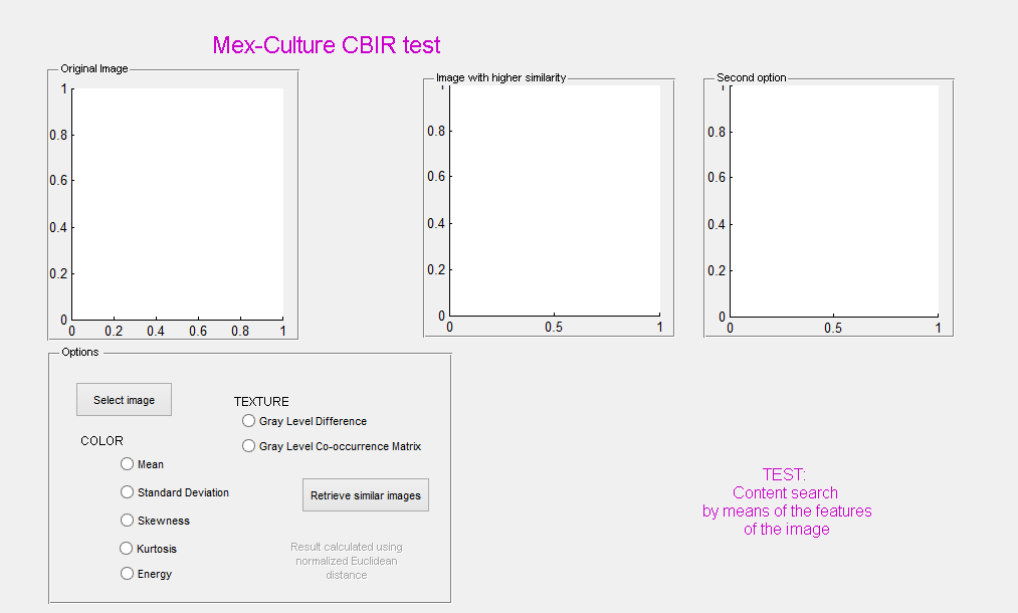


Figure D5.2. CBIR test Interface.

D.5.1 COLOR DESCRIPTORS

The color retrieving images on a search system based on the content of the images, the characteristics of color are the most commonly used. Indeed, the color characteristics present a robustness to the complex backgrounds and they are invariant with respect to rotation, traslation and image size. A color indexing is based on two elections: the colorimetric model and the color representation mode on this model. The color descriptor the most used is the color histogram [D.5.1]. Stricker et al [D.5.2] use color moments to overcome the effects of quantization of the color histogram. Any color distribution can be characterized by its moments, taking the most information concentrated in the lower order moments. Based on the above, we implemented as visual descriptors with color features the moment of first order (mean), moment of second order (Standard Deviation), moment of third order (skewness), moment of fourth order (Kurtosis) and the energy. The Euclidean distance is used as a measure of similarity between two color moments. The RGB (Red, Green, Blue) model is the color model used in this implementation. The color distributions of the R, G and B components of an image are represented by its color moments.

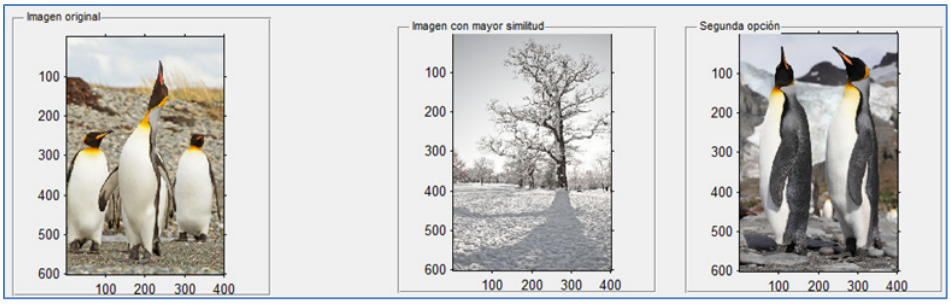
D.5.1.1 First color moment (Mean)

The first color moment of the i-th color component (i=1,2,3) is defined by

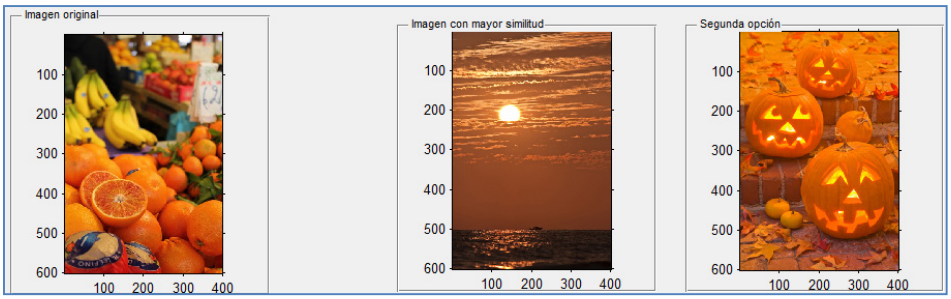
$$\mu_i = M_i^1 = \frac{1}{N}\sum_{j=1}^{N} I_{i,j} \quad (1)$$

where $I_{i,j}$ is the color value of the i-th color component of the j-th image pixel and N is the total number of pixels in the image.

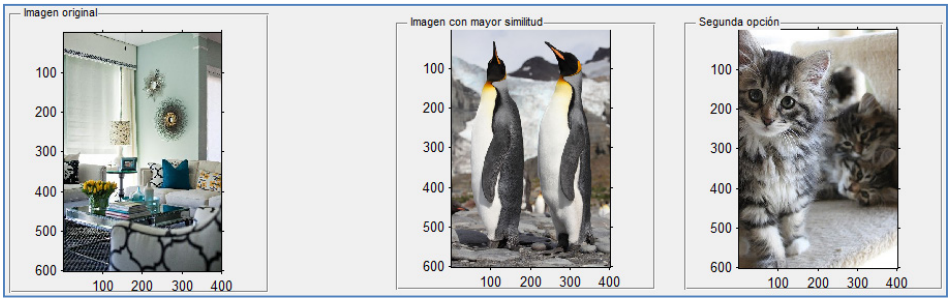
The moment of order one represents the average color of the image. This for each component of color model. The Figure D5.3 a) and b) show the good performance of the mean descriptor. This is due primarily to that the majority of the color information is more concentrated in one or at most two of the components of the color model (R, G, B). In the figure D5.3 c), there are multiple objects with different color; this makes that the distribution of color information is distributed almost equally on the color model components, and the obtained color distributions are very close. The images with best similarity do not necessarily present a visual content very similar to the original image.



a)



b)



c)

Figure D5.3. Retrieval of images by the mean descriptor.

D.5.1.2 Second color moment (Standard Deviation)

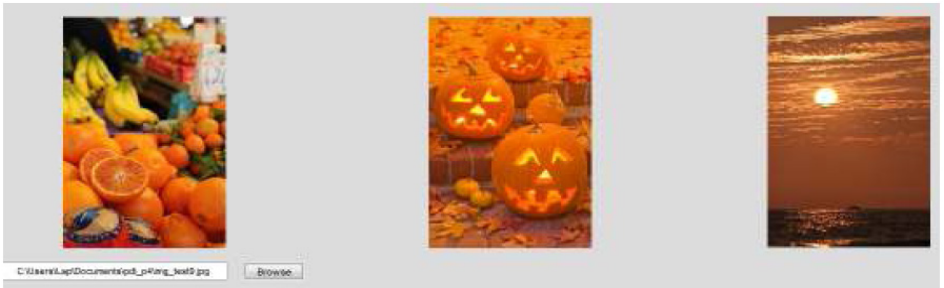
The second color moment of the i-th color component (i=1,2,3) is defined by

$$\sigma_i = M_i^2 = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(I_{i,j} - \mu_i\right)^2} \tag{2}$$

where $I_{i,j}$ is the color value of the i-th color component of the j-th image pixel, μ_i is the mean of image and N is the total number of pixels in the image.

The moment of second order represents the contrast of an image. If the value of the color variance is high, the image has more contrast.

The figure D5.4 a) and b) shows the functionality of standard deviation descriptor.



a)

b)

Figure D5.4. Image retrieval by the standard deviation descriptor.

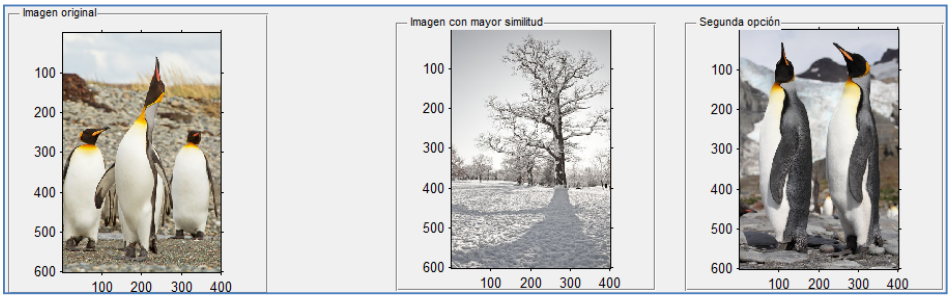
D.5.1.3 Third color moment (Skewness)

The third color moment of the i-th color component (i=1,2,3) is defined by

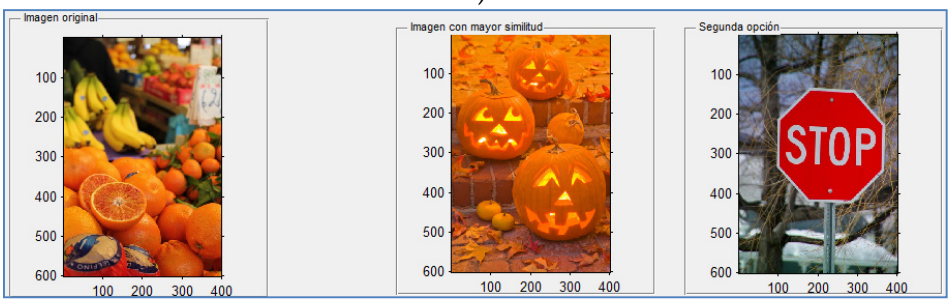
$$s_i = M_i^3 = \frac{\sum_{j=1}^{N}(I_{i,j}-\mu_i)^3}{(N-1)\sigma_i^3} \quad (3)$$

where $I_{i,j}$ is the color value of the i-th color component of the j-th image pixel, μ_i is the mean of image, σ_i is the standard deviation and N is the total number of pixels in the image.

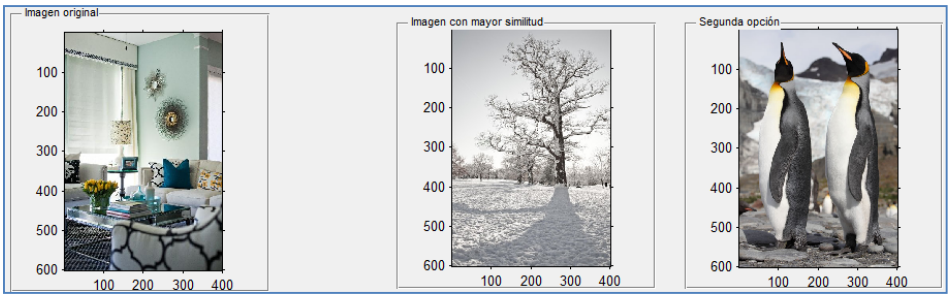
The third order moment characterizes the amount of light in an image. An image with a positive skewness value tends to be dark and very bright compared to a similar image with a value of less skewness.



a)



b)



c)

Figure D5.5. Image retrieval by the skewness descriptor.

The figure D5.5 a), b) and c) shows the functioning of the descriptor of skewness. It is noted that the two images most similar to the test have a different content.
The justification of the 3 tests that we performed, it is shown in the following graphics. The figure D5.6 shows the histogram of the test images in grayscale; the figure D5.7 shows the variety of histograms for the images of the database, where we can find the 'most similar' images according to the descriptor implemented. The histograms are directly related to the probability density function (pdf) found for each image.

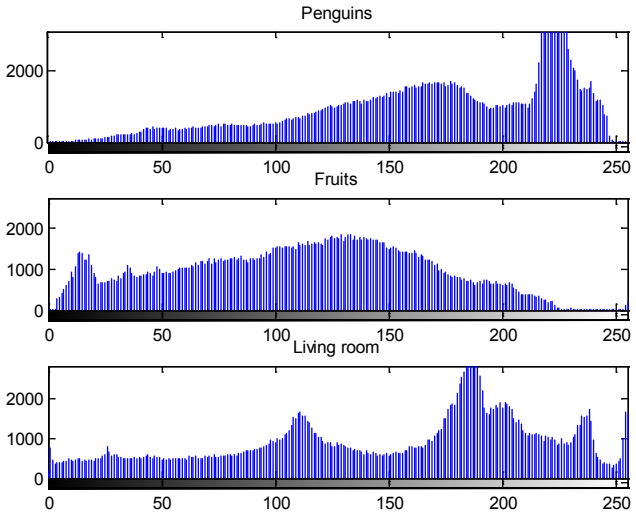


Figure D5.6. Grayscale histograms.

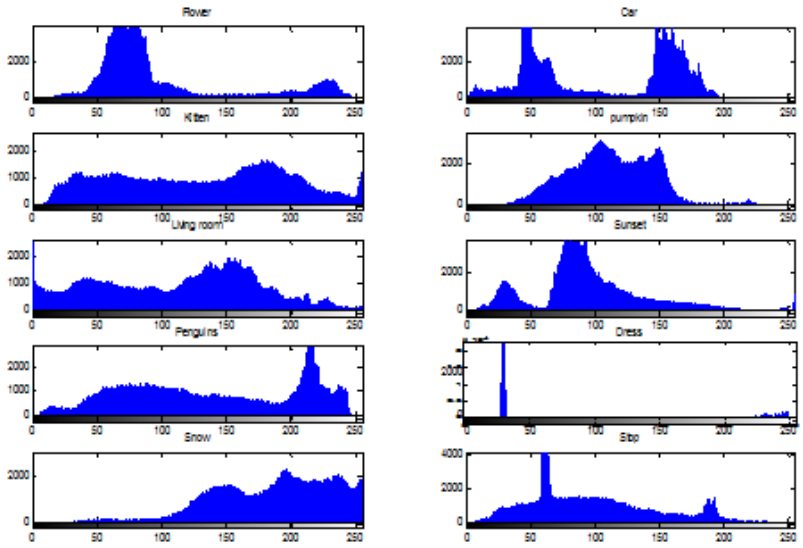


Figure D5.7. Histograms in grayscale of the database.

D.5.1.4 Fourth color moment (Kurtosis)

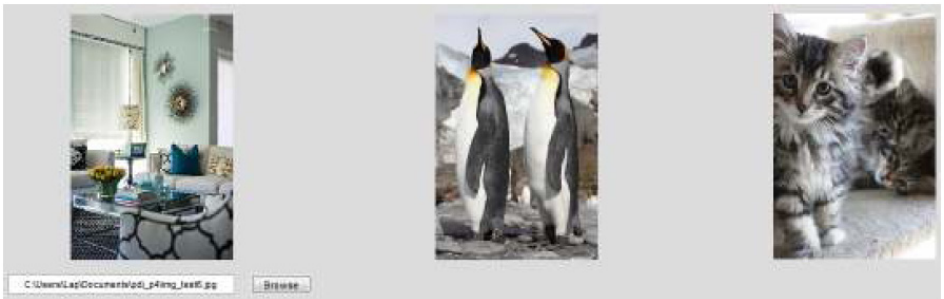
The fourth color moment of the i-th color component (i=1,2,3) is defined by

$$k_i = M_i^4 = \sqrt[4]{\frac{1}{N}\sum_{j=1}^{N}\left(I_{i,j} - \mu_i\right)^4} \tag{4}$$

where $I_{i,j}$ is the color value of the i-th color component of the j-th image pixel, μ_i is the mean of image and N is the total number of pixels in the image.
Kurtosis is the fourth color moment, and, similarly to skewness, it provides information about the shape of the color distribution. More specifically, kurtosis is a measure of how flat or tall the distribution is in comparison to normal distribution. A high value of kurtosis means that most of the values of the variance are extreme values and infrequent.

The figure D5.8 a) and b) shows the functionality of kurtosis descriptor.

a)



b)

Figure D5.8. Image retrieval by the kurtosis descriptor.

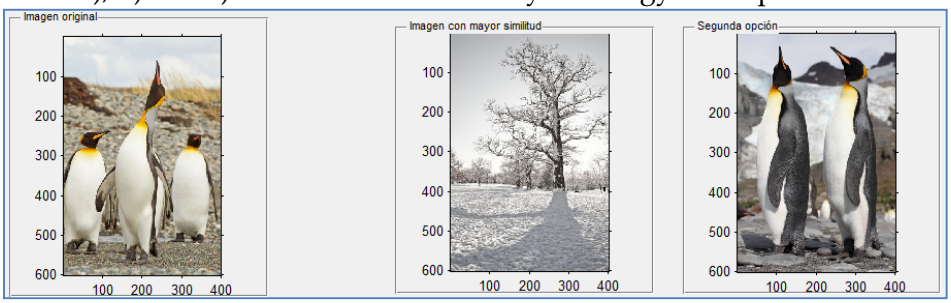
D.5.1.5 Energy descriptor

The energy of the i-th color component (i=1,2,3) is defined by

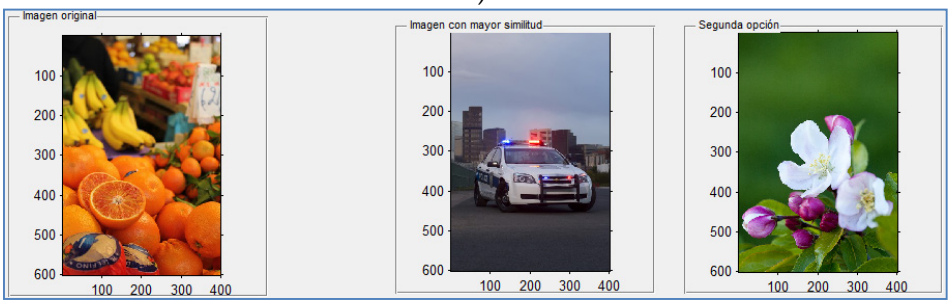
$$E_i = \sum_{j=1}^{N} \left(PDF_{i,j}\right)^2 \tag{5}$$

where $PDF_{i,j}$ is the probability density function of the i-th color component of the j-th image pixel and N is the total number of pixels in the image.

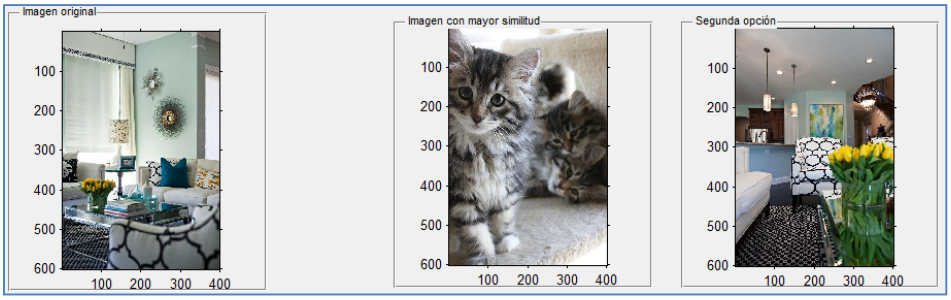
The figure D5.9 a), b) and c) shows the functionality of energy descriptor.



a)



b)



c)

Figure D5.9. Image retrieval by the energy descriptor.

D.5.2 Texture Descriptors

The texture is the second visual attribute widely used in the content-based image search. The texture has better response that color, especially when the color distributions are very close. The texture can be seen as a set of pixels (gray levels) spatially distributed by a number of spatial relationships, creating a homogeneous region. Research on texture modeling have focused on characterizing these spatial relationships, numerous studies and models have been proposed for texture characterization [D.5.3, D.5.4]. A texture is usually described as smooth or rough, soft or hard, coarse of fine, matt or glossy, and etc. Haralick considers a texture as an "organised area phenomenon" which can be decomposed into "primitives"

having specific spatial distributions [D.5.5]. This definition, also known as *structural approach*, comes directly from human visual experience of textures.

D.5.2.1 Gray Level Difference Method (GLDM)

The gray level difference method is a statistical method. The GLDM allows to calculate the number of occurrences of a difference of gray levels given. This amounts to compute the parameters of an image of difference between an original image and a translated image d [D.5.6]. GLDM gives the appearance of the texture by the difference in gray levels between pixels. This difference of gray levels is defined for each pixel of a region given by:

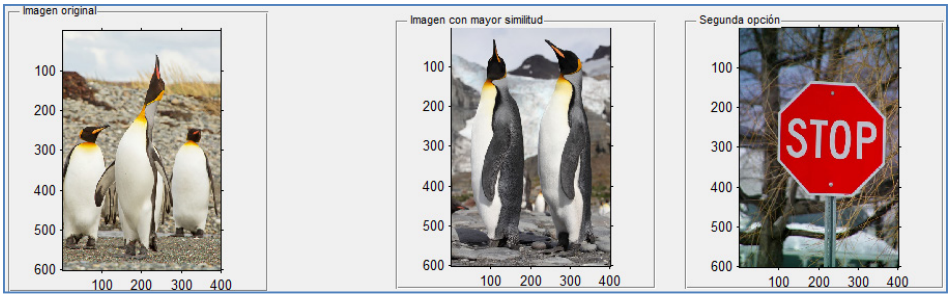
$$g = |I(i,j) - I(i+di, j+dj)| \tag{6}$$

where I(i,j) is the gray level of the coordinate point (i,j), and the coordinates of the displacement vector are described by (di,dj).
In this technique, we assume that the distribution values taken by g to the set of pixels belonging to the object are characterizing the texture.

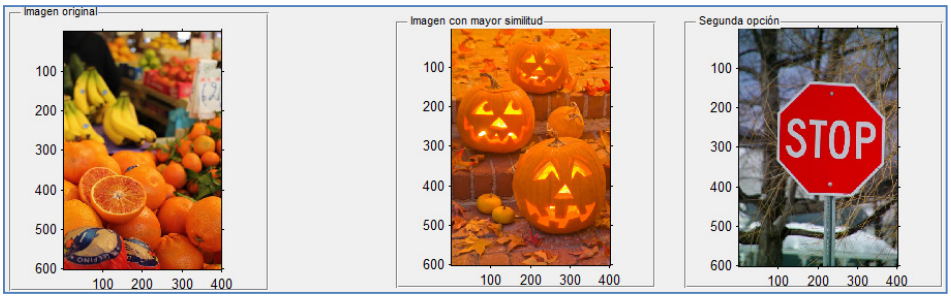
For GLDM method four images of differences related to four directions (north-east, north-west, south-east and south-west) are generated.These difference images are based on changes in intensity of the pixels in each of these directions. Then the histogram of this images is calculated followed by the probability density function (pdf). As processing is in grayscale, it has a length of 256 values by address. Finally the 4 directions are concatenated into a single vector and it is used as the texture descriptor. The distance between pixels to make a difference can vary, the 4 directions are defined as follow:

$$g = |I(i,j) - I(i, j+dj)| \quad \text{north-east direction} \tag{7}$$
$$g = |I(i,j) - I(i-di, j+dj)| \quad \text{north-west direction} \tag{8}$$
$$g = |I(i,j) - I(i+di, j)| \quad \text{south-east direction} \tag{9}$$
$$g = |I(i,j) - I(i-di, j-dj)| \quad \text{south-west direction} \tag{10}$$

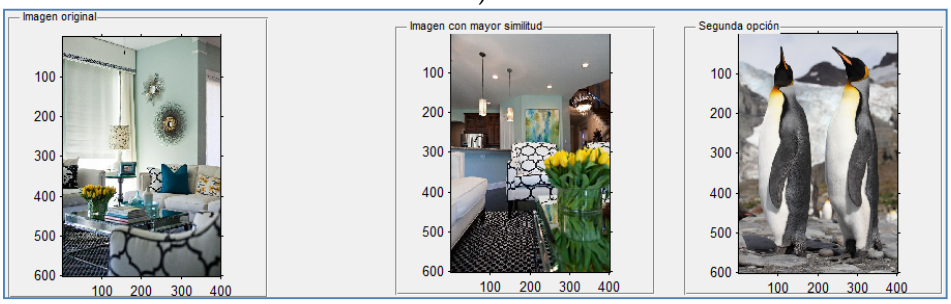
The figure D5.10 a), b) and c) shows the functionality of GLDM texture descriptor.



a)



b)



c)

Figure D5.10. Image retrieval by the GLDM texture descriptor.

The texture descriptor performed better than the mean descriptor to find similar images in the database. It is important to mention that in the figure D5.10 c) the most similar image is the same scene with a different angle of view.

D.5.2.2 Gray Level Co-occurrence Matrix (GLCM)

The gray level co-occurrence matrix method is a statistical technique [D.5.7]. The GLCM is the joint probability occurrence of grey levels i and j for two pixels with a defined spatial relationship in an image. The spatial relationship is defined in terms of distance d and angle θ. The matrix describes the observed regularities of the pixels gray levels in a region. We implement the cases in wich the pixel orientation is at 0, and 90 degrees. The distance between two analyzed pixels can be any value, but of course not more than the size of the whole image, but for each of the 2 directions considered earlier, we have chosen a distance of 0. The fact that the considered distance is 0, is favorable, because this way all the pixels in the processed image will be analyzed. Despues las matrices son normalizadas por la formula siguiente:

$$P_{d,\theta}(i,j) = \frac{p_{d,\theta}(i,j)}{MxN} \tag{11}$$

where $p_{d,\theta}(i,j)$ is the element of the co-occurrence matrix that defines the frequency of occurrence of pairs of gray levels for i and j pairs separated by a distance d and a direction θ. M and N pixels are the size of the image. The 256 gray leves were reduced to 8. From each co-occurrence matrix, two statistical measures are extracted [D.5.7].

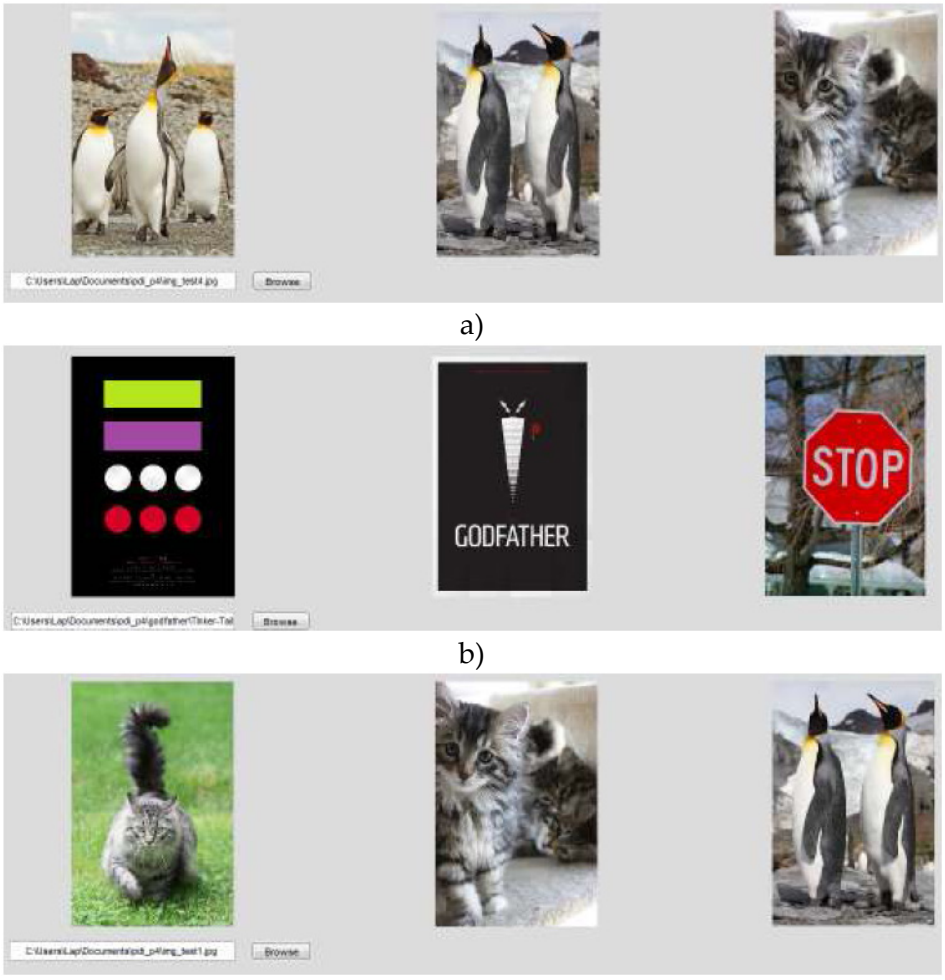
Entropy: The value of the entropy is low if the same pair of pixels apears frequantly and high if all the grayscale levels are faintly represented. This is an indicator of disorder that charaterises the texture.

$$H = -\sum_i \sum_j P_{d,\theta}(i,j) \log P_{d,\theta}(i,j) \quad (12)$$

Inverse Different Moment: This parameter measures the homogeneity of the image. It is correlated with a linear combination of the energy and contrast variables.

$$IDM = -\sum_i \sum_j \frac{1}{1+(i-j)^2} p_{d,\theta}(i,j) \quad (13)$$

The figure D5.11 a), b) and c) shows the functionality of GLCM texture descriptor .



a)

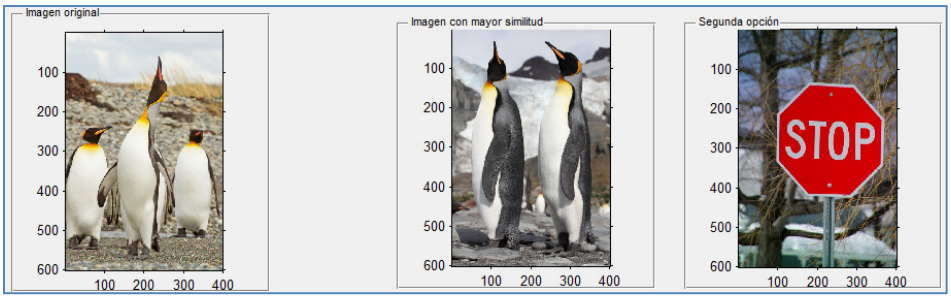
b)

c)

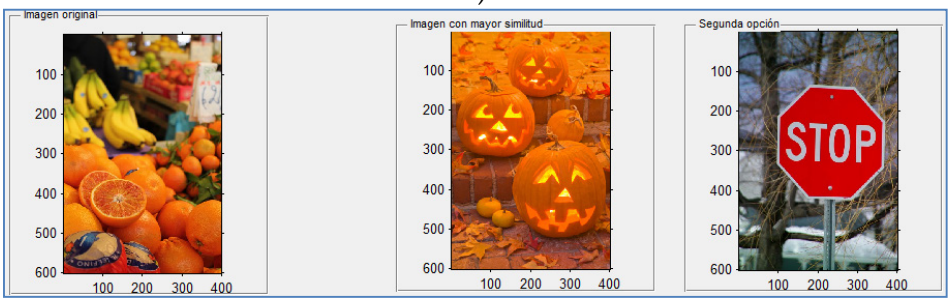
Figure D5.11. Image retrieval by GLCM texture descriptor.

D.5.3 TEST WITH THE SELECTION: MEAN DESCRIPTOR, SKEWNESS DESCRIPTOR, ENERGY DESCRIPTOR AND GLDM DESCRIPTOR.

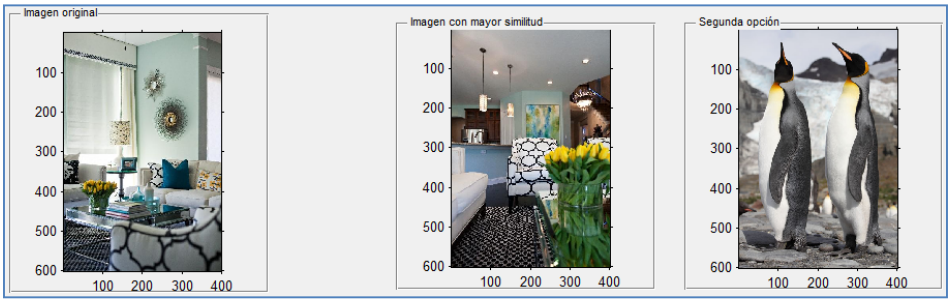
The figure D5.12 a), b) and c) shows the results when the descriptors of color and texture are selected: **mean, skewness, energy and GLDM**.



a)



b)



c)

Figure D5.12. Image retrieval by the color and texture descriptors.

When the descriptors are selected, the descriptors that have more weight in the decision are the mean and GLDM texture descriptors respectively. The skewness and energy descriptors have very small values despite that it used a normalized distance.

D.6 Local Features Based on JPEG2000

In the current state-of-the-art, the scalability of media description is not connected to scalable retrieval. When defining the project, we saw two connections between these two concepts of scalability. Consider a scalable, hierarchical representation of content such as Daubechies wavelets (employed in JPEG2000). First, interactive retrieval with relevance feedback can be made more scalable by iteratively increasing, during consecutive relevance feedback loops, the resolution on which content descriptors have been extracted. Second, we believed that a hierarchical representation of content would allow to design new multi-level LSH solutions that also support scalable retrieval. Optimism was justified by the results obtained in defining *global* features based on Daubechies wavelet pyramids (see [D.6.1]). However, our attempts (during the first few months of the project) to build *local* features on the basis of Daubechies wavelet pyramids did not result in competitive detectors and descriptors. Indeed, the performances of these detectors and descriptors were significantly lower than the performance of well-known state-of-the-art local features computed on the reconstructed

images. While other types of wavelets may, in principle, alow to obtain more competitive local features, these wavelets were not retained for scalable video coding standards, so we did not continue in this direction. Wavelet-based local features will not be included in the final system.

D.7 Feature Selection for Improved Scalability

For cultural video databases it is important to scalably index the content at a semantic level, ideally supporting several types of queries. One such query type is based on human action retrieval, which can be seen as Content Based Video Retrieval (CBVR), where the query is a semantic model.

Since we are interested in both the quality and the scalability of action detection, we propose a two level cascade (Fig. D.7.1) where the inexpensive first level serves to filter out a maximum of irrelevant video segments. The more expensive second level, using the GA kernel, only processes what the first level considers as potentially relevant. To further improve the quality and speed of detection with the GA kernel, we also introduce a feature selection method adapted to sparse multidimensional time series.

The size of the feature vectors employed for video content description is typically large and this has a significant impact on the complexity of sequence matching. Feature selection can both improve the quality of the results (by removing noise features) and reduce the complexity of matching (by significantly reducing the size of the feature vectors). “Scalability” can be understood as the ability of the system to "scale" to a very large amount of data. Thus, reducing computation time by one or more orders of magnitude is essential. Since content descriptors are usually of high dimension, feature selection methods can be considered as a solution to this problem if they preserve (or improve) the quality of the similarity computation using fewer features.

Video description is based on ‘tracklets’: points are sampled on a regular grid in each frame and tracked across 15 frames. Tracking is done by motion estimation between consecutive frames, based on the optical flow. The trajectory of a point consists of the coordinates of the point in consecutive frames and the patches around the trajectories are described with histograms of gradient, optical flow and optical flow gradient.

We devised a new method that is adapted to sequences of sparse, high-dimensional feature vectors. The presentation of this method was included in a recently submitted paper. In the experiments performed, this feature selection method strongly reduced the number of features (from 4,000 to 30-150), while the quality of detection increased.

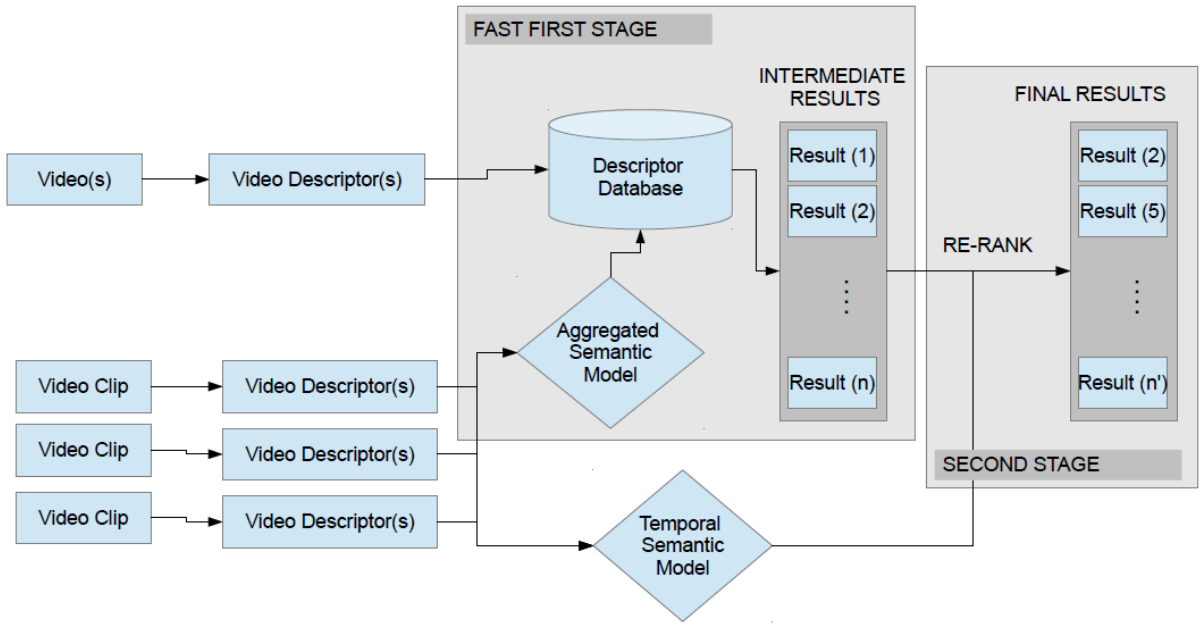


Figure D.7.1. Block diagram of the two-stage detection method.

Videos usually contain background motion in addition to the actions of interest, thus, because of the dense sampling, tracklet features are noisy and spurious trajectories are abundant. Many of the features (visual words) describing video frames will then act as noise. Furthermore, the Global Alignment (GA) kernel employs a summation over all the warping paths, using the L_2 distance between frames as an atomic dissimilarity measure. Consequently, the use of all the features may strongly impair the discrimination ability of the GA kernel. It is then important to use feature selection in the second stage before prior to the application of the GA kernel. This also has a positive impact on the computation cost.

Feature selection methods following the filter approach, like mRMR [D.7.1], aim to maximise the mutual information between the selected features and the classes, while minimising feature redundancy. Such methods retain just as well features (visual words) that are present in the positive examples and absent from the negative ones as features that are present in the negative examples and absent from the positive ones. However, the negative examples for one class include here not only examples from the other classes but also 'background' sequences, i.e. video sequences not showing any of the actions. Such background sequences from the training data are not representative for other videos. Consequently, features that are present in such sequences and absent from the positive examples will act as noise.

We introduce here a feature selection method that takes this issue into account and also considers the fact that we compare sequences of vectors rather than simple vectors. A feature set F is considered present in a sequence $x = x_1, \ldots, x_L$ if *all* the frames in the sequence contain at least one feature of the set. A feature set F is considered absent if at least one frame contains *no* feature from the set. We aim to find a set of features (visual words) that is (1) maximally present in the positive examples P, and (2) maximally absent from the negative examples N. Presence in the positive examples is measured by $P^+(F)$ in eq. D.7.1 and absence from the negative examples by $A^-(F)$ in eq. D.7.2. Algorithm D.7.1 describes a greedy search method that jointly maximizes these two criteria. We called this method Presence in Positives and Absence from Negatives (PPAN).

$$P^{+}(F) = \frac{1}{|P|}\sum_{x\in P}\prod_{i=1}^{L} 1\left(\sum_{w\in F} x_{iw}\right) \quad \text{(D.7.1)}$$

$$A^{-}(F) = \frac{1}{|N|}\sum_{x\in N}\left[1 - 1\left(\prod_{i=1}^{L}\sum_{w\in F} x_{iw}\right)\right] \quad \text{(D.7.2)}$$

In eq. D.7.1 and D.7.2, x_{iw} is the value of word w at time i in the BoVW sequence x and $1(x) = \{1 \text{ if } x > 0, 0 \text{ otherwise}\}$. These criteria can be extended to other feature representations.

Algorithm D.7.1. PPAN feature selection.

```
Require: A positive set P and a negative set N of BoVW timeseries
1: Set F to ∅, GroupScore = 0, PrevScore = 0, W = {all visual words}
2: while P+(F) < 0.8 OR PrevScore <= GroupScore do
     3: Set PrevScore = GroupScore
     4: Find arg max_{w∈W} S(F ∪ {w}) where S(G) = P+(G) A-(G). See
        eq. D.7.1 and D.7.2
     5: Update F = F ∪ {w}, GroupScore = S(F ∪ {w}), W = W − {w}
6: end while
7: return F
```

The algorithm stops when the selected features are present in 80% of the positive examples or an upper limit is reached for this presence. Preliminary experiments allowed to validate this criterion.

Since the mexican culture dataset was not yet available, we test on two common action detection datasets: Smoking and Drinking and MSR2. The Smoking and Drinking dataset (3 hours) consists of three videos (resolution 720 x 576): 2 feature films, “Coffee and Cigarettes” (2002) and “Sea of Love” (1989), and one video consisting solely of drinking sequences. It is split into a training set, a validation set and a testing set. For the Drinking action there are 106 training, 16 validation and 38 test sequences. For the Smoking action there are 78 training, 12 validation and 42 test sequences. The MSR Action Dataset II has one hour of footage (resolution 320 x 240) split into 54 videos with cluttered background. It contains three actions selected among those of the KTH dataset: boxing (81 instances), [hand]clapping (51) and [hand]waving (71).

Retrieval performance is given in Average Precision (AP - %). Timings for each action query are also given in tables D.7.1 and D.7.2.

Table D.7.1. Performance comparison on MSR Action II.

Method	Clapping	Waving	Boxing	Search time
Cao *et al.* [1]	13.1%	36.7%	17.5%	–
B&B Search [21]	23.9%	43.0%	30.3%	24.7 s
Max Subarray [22]	36.1%	54.1%	31.7%	2.3 s
Our method	**39.7%**	**55.0%**	**39.6%**	78 s (avg)

Table D.7.2. Performance comparison on Smoking and Drinking.

Method	Drinking	Smoking
Laptev *et al.* [12]	49%	–
Gaidon *et al.* [5]	57%	31%
Klaser *et al.* [10]	59%	24%
Oneata *et al.* [13]	64%	**50%**
AP of our method	**65.5%**	45.1%
Search time of our method	178 s	240 s

E CONCLUSION AND PERSPECTIVES

Nowadays CBVR technique is a very important research field to manage and index multimedia databases. Thus, several visual descriptors has been proposed in the last years, which in many cases are dependent on the application as well as on the multimedia database where are applied. Hence, it is clear that there is not yet been able to solve the problematic of which descriptor is better or worse in a particular application [D.2.8]. In this way, a challenging application consists on retrieve multimedia content with the same object which is captured by different devices, and in distinct environmental conditions. In the work carried out, it is proposed a fast CBVR technique which involves the combination of a local descriptor obtained from the Speeded-Up Robust Feature (SURF) algorithm together with an effective and fast object matching operation (section D.2). Before computing the SURF descriptor, the key frames are extracted from codified video. The experimental results show that the proposed methodology provides good accuracy in precision terms giving values equals to or greater than 90%. Also, if the application requires, values of the threshold Th may be adjusted in order to build several filters that include results sorted by the most relevant content.
In the context of the project and considering the human resource formation two students from UNAM are involved in the project under the advise of Francisco Javier García-Ugalde: the doctoral student Fernando Castillo Flores, officially enrolled at "Posgrado en ciencia e ingeniería de la computació", PCIC, on August 5, 2013, and the master student Laura Reyes Ruíz, officially enrolled at "Posgrado en ingeniería: procesamiento digital de señales", PI, on January 27 2014.

In the other work, we managed to combine the global and local color features in one descriptor and as a result we obtained an efficient color descriptor which improves the image retrieval task (section D.3). The results of retrieving images were strictly evaluated by 3 metrics, ARR, ARP and ANMRR. The evaluation of the algorithms from the literature was compared to the evaluation of the proposed scheme and the results show that the proposed descriptor is better retrieving images than the other ones. The image retrieval result depends on the database which is used if it is used a global or a local color descriptor only, for instance, if the database contains images with many colors and many details, a local color descriptor should be used, and for images with few colors and with no details, a global color descriptor improves the image retrieval. Using the proposed scheme, the image retrieval result does not depend on the database images, because we are taking into account both color features, global and local.

In section D.4, for the analysis of video and get the metadata describing it, should be analyzed in different modules. As the first module is the algorithm that finds representative video frames. To then analyzing the frames and generate metadata describing the video.
Several algorithms and software for frames of a video, but not necessarily representative. In this first module an algorithm which was developed by a binary partition of time are a representative video frames and streamlines this search, both processing time and appropriate. The results are compared with a library, FFMPEG, and best results are obtained in very short processing times. Which enters the requirements for automatic video analysis: quickness mainly.

In section D.5, choosing a better set of visual descriptors promises a good characterization of color, texture and form concepts. For this, we are implementing other descriptors of color, texture, shape and movement. All these descriptors will be applied also in the compressed domain. In this way we can have the descriptor or combination of them that will better meet the content databases.

F REFERENCES

[D.1.1] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain, "Content-based multimedia information retrieval: State of the art and challenges," ACM Trans. Multimedia Comput., Commun. Appl., vol. 2, no. 1, pp. 1–19, Feb. 2006.

[D.1.2] C. Morand, J. Benois-Pineau, J.P. Domenger, "Scalable indexing of HD video," Proceeding of International Workshop on Content-based Multimedia Indexing. pp.417–424. CBMI08, 2008.

[D.1.3] J. Benois-Pineau, F. Morier, D. Barba, and H. Sanson, "Hierarchical segmentation of video sequences for content manipulation and adaptive coding," 66(2):181–201, April 1998.

[D.1.4] O. Brouard, F. Delannay, V. Ricordel, and D. Barba, "Spatio-temporal segmentation and regions tracking of high definition video sequences based on a markov random field model," Pages 1552–1555, 2008.

[D.1.5] Francesca Manerba, J. Benois-Pineau, and Riccardo Leonardi, "Extraction of foreground objects from an mpeg2 video stream in rough-indexing framework," volume 5307, pages 50–60. SPIE, 2004.

[D.1.6] P. Salembier and L. Garrido, "Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval," 9(4):561–576, April 2000.

[D.1.7] Pierre Hanna, Pascal Ferraro, and Matthias Robine, "On optimizing the editing algorithms for evaluating similarity between monophonic musical sequences," 2007.

[D.1.8] Hugo Meinedo and Joao Neto, "Automatic speech annotation and transcription in a broadcast news task," In in Proc. ISCA ITRW on Multilingual Spoken Document Retrieval, Hong Kong, 2003.

[D.1.9] Y.Wu,B. L. Tseng, and J.R. Smith, "Ontology-based multi-classification learning for video concept detection," in Proc. IEEE Int. Conf. Multimedia Expo., vol. 2, pp. 1003–1006. Jun. 2004.

[D.1.10] S. Fischer, R. Lienhart, and W. Effelsberg, "Automatic recognition of film genres," in Proc. ACM Int. Conf. Multimedia, pp. 367–368. 1995.

[D.1.11] Nicolas Hanusse, Sofian Maabout, and Radu Tofan, "Matérialisation partielle des cubes de données," In Actes de la conférence Bases de Données Avancées (BDA), pages 21–40, Namur (Belgique), octobre 2009.

[D.1.12] Yi Yu, Michel Crucianu, Vincent Oria, and Lei Chen, "Local summarization and multi-level LSH for retrieving multi-variant audio tracks," In MM'09: Proceedings of the 17th ACM international conference on Multimedia, pages 341–350, New York, NY, USA, 2009.

[D.2.1] H. B. Kekre, S. D. Thepade and S. Gupta, Content Based Video Retrieval in Transformed Domain using Fractional Coefficients, International Journal of Image Processing (IJIP), 7 (2013), 237-247.

[D.2.2] W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, Content-based image retrieval at the end of the early years, IEEE Trans. Pattern Analysis and Machine Intelligence, 22 (2000), 1349–1380.

[D.2.3] R. Datta, D. Joshi, J. Li, and J. Z. Wang, Image retrieval: Ideas, influences, and trends of the new age, ACM Comput. Surv., 40 (2008), 1–60.

[D.2.4] http://mpeg.chiariglione.org/standards/mpeg-7

[D.2.5] B. V. Patel and B. B. Meshram, Content Based Video Retrieval Systems, International Journal of UbiComp (IJU), 3 (2012), 13-30.

[D.2.6] W. Hu, N. Xie, L. Li, X. Zeng and A. Maybank, A survey on visual content-based video indexing and retrieval, IEEE Trans. on Systems, Man, and Cybernetics-Part C, 41 (2011), 797-819.

[D.2.7] K. Velmurugan, and S. Santhosh Baboo, Content-Based Image Retrieval using SURF and Color Moments, Global Journal of Computer Science and Technology, 11 (2011), 1-4.

[D.2.8] O. Boullosa-García, Estudio comparativo de descriptores visuales para la detección de escenas cuasi-duplicadas, M.Sc. Thesis in Video Processing and Understanding, Lab. Dpto. de Ingeniería Informática Escuela Politécnica Superior Universidad Autónoma de Madrid, 2011.

[D.2.9] H. Bay, A. Ess, T. Tuytelaars, and L Van Gool, SURF: Speeded Up Robust Features, Computer Vision and Image Understanding (CVIU), 110 (2008), 346-359.

[D.2.10] B. J. Davis and S. H. Nawab, The relationship of transform coefficients for differing transforms and/or differing sub block sizes, IEEE Transactions on Signal Processing 52 (2004), 1458–1461.

[D.2.11] W. Yulin and A. Pearmain, Blind MPEG-2 video watermarking robust against geometric attacks: a set of approaches in DCT domain, IEEE Transactions on Image Processing, 15 (2006), 1536–1543.

[D.2.12] J. Jianmin and F. Guocan, The spatial relationship of DCT coefficients between a block and its sub-blocks, IEEE Transactions on Signal Processing, 50 (2002), 1160–1169.

[D.2.13] http://www.youtube.com/yt/copyright/es/creative-commons.html

[C.3.1] Otávio A.B. Penatti, Eduardo Valle, Ricardo da S. Torres (2012) Comparative study of global color and texture descriptors for web image retrieval, Journal of Visual Communication & Image Representation 23: 359-380.

[C.3.2] Ahmed Talib, Massudi Mahmuddin, Husniza Husni, Loay E. George (2013) A weighted dominant color descriptor for content-based image retrieval, Journal of Visual Communication & Image Representation 24: 345-360.

[C.3.3] Xiang-Yang Wang, Yong-Jian Yu, Hong-Ying Yang (2004) An effective image retrieval scheme using color, texture and shape features, Computer Standards & Interfaces 33: 59-68.

[C.3.4] Hong Shao, Yueshu Wu, Wencheng Cui, Jinxia Zhang (2008) Image Retrieval Based on MPEG-7 Dominant Color Descriptor, IEEE, The 9th International Conference for Young Computer Scientist: 753-757.

[C.3.5] Nai-Chung Yang, Wei-Han Chang, Chung-Ming Kuo, Tsia-Hsing Li (2008) A fast MPEG-7 dominant color extraction with new similarity measure for image retrieval, Journal of Visual Communication & Image Representation 19: 92-105.

[D.3.1] Md. Mahmudur Rahman, Prabir Bhattacharya, Bipin C. Desai (2009) A unified retrieval framework on local visual and semantic concept-based feature spaces, Journal of Visual Communication & Image Representation 20: 450-462.

[D.3.2] Hock C. Chan, Yue Wang (2004) Human factors in color-based image retrieval: an empirical study on size estimate accuracies, Journal of Visual Communication & Image Representation 15: pp113-131.

[D.3.3] Michael J. Swain, Dana H. Ballard (1991) Color Indexing, International Journal of Computer Vision 7(1): 11-32.

[D.3.4] Dengsheng Zhang, Guojun Lu (2003) Evaluation of similarity measurement for image retrieval, IEEE International Conference of Neural Networks & Signal Processing: 928-931.

[D.3.5] Jing Huang, Kumar S. R., Mitra M., Wei-Jing Zhu, Zabih R (1997) Image Indexing Using Color Correlograms, Conference on Computer Vision and Pattern Recognition: 762-768.

[D.3.6] Hamid A. Jalab (2011) Image retrieval system based on color layout descriptor and Gabor filters, IEEE Conference on Open Systems: 32-35.

[D.3.7] Eiji Kasutani, Akio Yamada (2001) The MPEG-7 color layout descriptor: A compact image feature description for high-speed image/video segment retrieval, IEEE International Conference of Image Processing 1: 674-677.

[D.3.8] A. Fierro-Radilla, M. Nakano-Miyatake, M. García-Vázquez (2013) Evaluación de Descriptores Basados en Color para Búsqueda de Imágenes en Internet, Simposio Iberoamericano Multidisciplinario de Ciencias e Ingenierías

[D.3.9] Atoany N. Fierro-Radilla, Mariko Nakano-Miyatake, Héctor Pérez-Meana, Manuel Cedillo-Hernández, Francisco García-Ugalde (2013) An Efficient Color Descriptor Based on Global and Local Color Features for Image Retrieval, IEEE 10th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE): 233-238.

[C.5.1] C. Yeh and C. J. Kuo, “Iteration-free clustering algorithm for nonstationary image database,” in IEEE Trans. Multimedia, vol.5, No. 2, 2003.

[C.5.2] S. S. Ravela, “On multi-scale differential features and their representations for image retrieval and recognition,” in Ph.d University of Massachusetts Amherst, 2003.

[D.5.1] Wain, M. and Ballard, D., “Color indexing,” International Journal of Computer Vision, 1991, 7, (1), pp. 11-32.

[D.5.2] Stricker, M. and Orengo, M., “Similarity of Color Images,” in Proc. SPIE Storage and Retrieval for Still Image and Video Databases III, February 1995, San Jose, CA, USA, pp. 381-392.

[D.5.3] Majid Mirmehdi, Xianghua Xie, and Jasjit Suri., "Handbook of Texture Analysis," Imperial College Press, 2008.

[D.5.4] G. N. Srinivasan, and Shobha G., "Statistical Texture Analysis," Proceedings of world academy of science, engineering and technology vol.36 Dec. 2008.

[D.5.5] R. M. Haralick., "Statistical and structural approaches to texture," In Proc. IEEE, volume 67(5), pages 786-804, 1979.

[D.5.6] Conners RW, Harlow CA., “A theoretical comparison of texture algorithms,” IEEE Trans Pattern Anal 2:204–222, 1980.

[D.5.7] R. M. Haralick, K. Shanmugan, I. Dinstein., "Textural features for image classification," IEEE Trans. Systt. Man. Cybern. SMC-3(6): 610-621. 1973.

[D.6.1] C. Morand. Segmentation spatio-temporelle et indexation video dans le domaine des représentations hiérarchiques. Thèse de doctorat présentée à l’Université Bordeaux 1, 2009.

[D.7.1] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans. on PAMI, 27 (8):1226–1238, 2005.