Table of Content

List of Figures

Content-Based Image Retrieval Using Topological Descriptors

Annette Morales González-Quevedo and Edel García Reyes

Dpto. Reconocimiento de Patrones, Centro de Aplicaciones de Tecnología de Avanzada (CENATAV),
Ciudad de La Habana, Cuba
{amorales, egarcia}@cenatav.co.c

RT_039, Serie Azul, CENATAV
Aceptado: 24 de septiembre de 2010

Abstract. Although lot of research have been dedicated to the topic of Content-Base Image Retrieval (CBIR), existent approaches still have ill-defined issues and the well-known “semantic gap” between low-level computational representation of images and high-level semantics required by humans, remains unbridged. In this work we make a state-of-the-art of several major areas that are involved in the CBIR process. These are the visual low-level features for describing images, spatial relationship representations of regions in images, current approaches to endow images with high-level semantics, such as automatic image annotation and retrieval methods which also intend to make semantic retrievals from image databases. Some current CBIR systems are also described.

Keywords: content-based image retrieval, CBIR, automatic image annotation, relevance feedback, topological relationships, irregular pyramids, combinatorial maps.

Resumen. Aunque se han llevado a cabo muchas investigaciones sobre el tema de Recuperación de Imágenes por Contenido, los enfoques existentes aún tienen problemas sin resolver y la “brecha semántica” entre la representación computacional de bajo nivel de las imágenes y la semántica de alto nivel que requieren los humanos, aún se mantiene entre los principales puntos a solucionar. En este trabajo se realiza un estado del arte de las principales áreas que se involucran en el proceso de la recuperación de imágenes por contenido. Entre estas se encuentran el proceso de descripción de la imagen mediante rasgos de bajo nivel y la representación de relaciones espaciales entre las regiones de las imágenes. También son abordados los enfoques actuales para dotar las imágenes con semántica de nivel superior, tales como la anotación automática de imágenes y métodos de recuperación que plantean realizar recuperaciones semánticas en bases de datos de imágenes.

Palabras clave: recuperación de imágenes por contenido, CBIR, anotación automática de imágenes, retroalimentación por relevancia, relaciones topológicas, pirámides irregulares, mapas combinatorios.

1 Introduction

Due to the advances in digital photography, storage capacity and networks speed, storing large amounts of high quality images has been made possible. Digital images are used in a wide range of applications such as medical, virtual museums, military and security purposes, and personal photo albums. However, users have difficulties in organizing and searching large numbers of images in databases, as the current commercial database systems are designed for text data and not well suited for digital images. Therefore, an efficient way for image retrieval is desired. The current most desirable image retrieval feature is retrieving images based on their semantic content.

Content-based image retrieval is an important aspect for any world, society or organization relying on digital images as an important source of information because the ability to produce pictorial information exceeds the ability to retrieve this information by far.

Object class recognition, automatic image annotation, and object retrieval are strongly related tasks. In object class recognition, the aim is to identify whether a certain object is contained in an image; in automatic image annotation, the aim is to create a textual description of a given image; and in object retrieval, images containing certain objects or object classes have to be retrieved out of a large set of images. Each of these techniques is important to allow for semantic retrieval from image collections.

A common limitation of the existing retrieval systems is not considering the spatial relation of image objects (or components) which may reveal important properties of the scene being analyzed. Some studies from the visual cognition community concluded that structural relations between image components play a central role in the human similarity comparison process. An earlier finding by Lowe states that the similarity between two groups of objects does not equal the sum of similarities between the individual objects. [1]

We believe that spatial location of objects or regions as well as inter relationships between regions within an image provide vital clues for searching image databases.

Objects that are mapped into the image plane induce spatial relations among each other and between their parts. Geometrical measurements derived from a digital image are very sensitive to errors due to noise, discrete sampling and motion inaccuracies. However these structural and topological relations are inherent to the objects and their arrangement in the image and mostly do not depend on the particular imaging situation. This is the background of several recent contributions describing spatial/structural representations and transformations preserving existing topological relations in the image plane.

Image characterization should emulate human perception as possible. Therefore, our perception works in a multi-resolution fashion. For some visual tasks, human eyes may select coarse filters to obtain coarse image features; for others, they select finer features. An image search engine thus must have the flexibility to model subjective perceptions and to fulfill a variety of search tasks. [2] This is the principle of many hierarchical approaches which represent image features in successively reduced resolution levels.

This is also the base of the irregular dual pyramid framework proposed by Kropatsch, which not only captures important topological relationships between regions, but it is also capable of generating several levels of resolution preserving these relationships. We believe that this framework can be very useful for adding rich semantics to images.

The retrieval principle of CBIR systems is based on visual features such as color, texture, and shape or the semantic meaning of the images, which can be added using image annotations. Nevertheless, in most cases, the user needs are not fulfilled. This problem stems from the fact that visual similarity measures, such as color histograms, in general do not necessarily match *semantics* of images human *subjectivity*. To make the problem even worse, people often have different semantic interpretations of the same image. Even the same person may have different perception about the same image at different times.

In this work, we perform an overview about the current approaches involved in Content-Base Image Retrieval, with both their advantages and problems.

The rest of the report is organized as follows. In Section 2 we briefly review various low-level image features used in high-level semantic-based CBIR systems. Image similarity measure is also discussed in Section 2. In Section 3, we present some methods to characterize topological relationships between regions and Section 4 provides an overview of some developed approaches which intend to encode topological relations among objects or regions. In Section 5 we described some methods regarding automatic image annotation for adding semantics to images and Section 6 describes proposed methods for semantically retrieving images from databases. In Section 7, a description of some current CBIR systems and application's areas is provided. Section 8 shows a discussion about the open problems in CBIR. Sections 9, 10 and 11 are devoted to the development of this activity in the world, showing important databases, scientific conferences, institutions and investigation groups that are related to this topic. Finally, Section 12 concludes the report.

2 Visual Features of Images

The first step in a CBIR system is defining how images will be represented regarding their visual content, since a request in CBIR is not posed to the image itself but to a representation of features of the image. Features are often represented by a feature vector. Quantifiable attributes such as histograms of color, grey scale or texture features are entries of the feature vector [3].

For any object in an image, there are many features which are interesting points on the object that can be extracted to provide a feature description of the object. This description can then be used to identify the object when attempting to locate the object in another image containing many objects.

Most object recognition systems tend to use either global image features, which describe an image as a whole, or local features, which represent image patches. Global features have the ability to generalize an entire object with a single vector. Consequently, their use in standard classification techniques is straightforward. Local features, on the other hand, are computed at multiple points in the image and are consequently more robust to occlusion and clutter. However, they may require specialized classification algorithms to handle cases in which there are a variable number of feature vectors per image [4].

Basically, there are two kinds of features that can be extracted from an image: visual features and spatial relationship features. Visual features, such as texture, shape, and color, could potentially be used as a basis for coarse-grained image similarity retrieval (e.g. "find images which are predominantly green"). If these visual features are extracted for individual objects within images, the queries could be more detailed (e.g. "find images with a yellow box").When the spatial relationships are introduced, the queries can be even more precise (e.g. "find a yellow box over a white table") [5].

In a CBIR system, it is very challenging to find a set of features that can model the user's perception of images in the database.

Measuring meaningful image similarity is a problem that rests on two elements: finding a set of features which adequately encodes the characteristics that we intend to measure and finding a suitable metric for the feature space. Since the same feature space can be endowed with infinity of metrics, the two problems are by no means equivalent nor does the first subsume the second [6].

Features Descriptors

When an image is segmented in several regions representing potential objects, the remaining question is which is the most appropriate descriptor to characterize the regions? There are a large number of possible descriptors and associated distance measures which emphasize different image properties like pixel intensities, color, texture, edges etc. [7]

In [7], an evaluation of the descriptors is performed in the context of matching and recognition of the same scene or object observed under different viewing conditions.

The earliest work on appearance-based object recognition has mainly utilized global descriptions such as color or texture histograms. The main drawback of such methods is their sensitivity to real-world sources of variability such as viewpoint and lighting changes, clutter and occlusions. For this reason, global methods were gradually supplanted over the last decade by part-based methods, which became one of the dominant paradigms in the object recognition community. Part-based object models combine appearance descriptors of local features with a representation of their spatial relations.

Initially, part-based methods relied on simple Harris interest points, which only provided translation invariance. Subsequently, local features with higher degrees of invariance were used to obtain robustness against scaling changes and affine deformations. While part-based models offer an intellectually satisfying way of representing many real-world objects, learning and inference problems for spatial relations remain extremely complex and computationally intensive, especially in a weakly supervised setting where the location of the object in a training image has not been marked by hand [8].

Recent work has shown that local features invariant to common image transformations (e.g., SIFT) are a powerful representation for recognition, because the features can be reliably detected and matched across instances of the same object or scene under different viewpoints, poses, or lighting conditions. Most researchers, however, have done recognition with local feature representations using nearest-neighbor or voting-based classifiers followed by an alignment step; both may be impractical for large training sets, since their classification times increase with the number of training examples.

Many different descriptors have been presented in the literature (see [7] for an overview). In [8] they use three different descriptors: SIFT, SPIN and RIFT. The SIFT descriptor has been shown to outperform a set of existing descriptors [7], while SPIN and RIFT, have achieved good performance in the context of texture classification.

2.1.1 Color Feature Descriptors

Color is one of the most important image indexing features employed in CBIR. Color is probably the most important feature that users can specify when they create image queries. In addition, proper color measures can be reliable even in the presence of changes in illumination, view angle, and scale [9].

Color features include the conventional color histogram (CCH), the fuzzy color histogram (FCH), the color correlogram (CC) and a more recent color-shape-based feature. The extraction of the color-based features follows a similar progression in each of the four methods [10]:

- Selection of the color space,
- quantization of the color space,
- extraction of the color feature,
- derivation of an appropriate distance function.

Conventional Color Histogram. The conventional color histogram of an image indicates the frequency of occurrence of every color in an image. From a probabilistic perspective, it refers to the probability mass function of the image intensities. It captures the joint probabilities of the intensities of the color channels (R, G and B in the RGB color-space, or H, S and V in the HSV color-space, and similarly for other color spaces) [10].

Computationally, the color histogram is constructed by counting the number of pixels of each color (in the quantized color space). The appealing aspect of the color histogram is its simplicity and ease of computation. There are however, several drawbacks associated with it. The first is the high dimensionality of the color histogram, even after the quantization of the color space. Another downside of the color histogram is that it does not take into consideration color similarity across different bins. Also, the color histogram is a global image feature that does not encode any color-spatial information.

Fuzzy Color Histogram. In the fuzzy color histogram approach, a pixel color belongs to all histogram bins with different degrees of memberships to each bin. The primary advantage of this histogram is that it encodes the degree of similarity of each pixel color to all other histogram bins through a fuzzy-set membership function. Taking into account color pixel similarity makes the fuzzy color histogram more robust to quantization errors as well as to changes in light intensity. However, it still embeds several drawbacks. Like the conventional color histogram, the fuzzy color histogram delineates only the global color properties of the image, and its features dimensionality is as high as that of the color histogram. In addition, the fuzzy color histogram approach introduces the additional challenge of computing the appropriate fuzzy membership function [11].

Color correlogram. The color correlogram expresses how the spatial correlation of pairs of colors changes with distance. A color correlogram for an image is defined as a table indexed by color pairs, where the *d*th entry for row (*i*,*j*) specifies the probability of finding a pixel of color *j* at a distance *d* from a pixel of color *i* in the image. In general, since local correlations between different colors are more significant than global correlations in an image, a small value of d is sufficient to capture special correlations. Important characteristics of the color correlogram method are that it encodes local as well as global spatial information, and it has been shown to work well for coarse color images. The major drawback of this method is the high dimensionality of the feature space [12].

The Color-Shape Based Method. The color-shape based method (CSBM) [13] is based on color, area and perimeter-intercepted lengths of segmented objects in an image. The algorithm starts by clustering image pixels into *K* clusters according to the K-means algorithm. The mean value of each cluster is regarded as a representative color for the cluster. A quantized color image *I'* is obtained from the original image I by quantizing pixel colors in the original image into *K* colors. A connected region having pixels of identical color is regarded as an object. The area of each object is encoded as the number of pixels in the object. Further, the shape of an object is characterized by "perimeter-intercepted lengths" (PILs), obtained by intercepting the object perimeter with eight line segments having eight different orientations and passing through the object center. The PILs have been shown to be a good characterization of object shapes. The main advantage of this method is that it encodes object shapes as well as colors. The drawback on the other hand, is more involved computation, and the need to determine appropriate color thresholds for the quantization of the colors. Another drawback of CSBM is its impressionability to contrast and noise variation.

Though all these methods provide good characterization of color, they have the problem of high-dimensionality. This leads to more computational time, inefficient indexing, and low performance. The global color-based methods suffer from problems of non-invariance and large storage requirements. Hence other features like shape, texture, and spatial location are added to the feature space to enhance the retrieval efficiency and effectiveness [14].

2.1.2 Texture Feature Descriptors

Texture generally refers to the presence of a spatial pattern that has some properties of homogeneity [10]. Texture manifests itself as variations of the image intensity within a given region. Directional features are extracted to capture image texture information. Texture feature extraction methods include the steerable pyramid, the contourlet transform, the Gabor wavelet transform and the complex directional filter bank (CDFB).

Texture features usually contain important information about the structural arrangement of surfaces and their relationship to the surrounding environment [15]. Typical texture measures used in image retrieval systems are coarseness, contrast and directionality. Coarseness measures the scale of the texture (pebbles versus boulders), contrast describes its vividness, and directionality describes whether it has a favored direction (like grass) or not (like a smooth object). A good texture discrimination is not needed in image retrieval, but more important is the perceptual similarity of textures [9].

Co-occurrence matrices. Co-occurrence matrices are based on second-order statistics of pairs of intensity values of pixels in an image. A co-occurrence matrix counts how often pairs of grey levels of pixels, separated by a certain distance and lying along certain direction, occur in an image. This characterization of texture is not very effective for classification and retrieval. In addition, these features are expensive to compute; hence, co-occurrence matrices are rarely used in image database applications [15]. Instead of computing the texture features in the spatial domain, an attractive alternative is to use transform domain features. The discrete Fourier transform (DFT), the discrete cosine transform (DCT), and the discrete wavelet transforms (DWT) have been quite extensively used for texture classification [15].

The Steerable Pyramid. The steerable pyramid generates a multi-scale, multi-directional representation of the image. The basic filters are translations and rotations of a single function. The image is decomposed into one decimated low-pass sub-band and a set of undecimated directional sub-bands. The decomposition is iterated in the low-pass sub-band. Because the directional sub-bands are undecimated, there are 4K/3 times as many coefficients in the representation as the original image, where K is the number of orientations [16].

The Contourlet Transform. The contourlet transform provides a multi-scale, multi-directional decomposition of an image. It is a combination of a Laplacian pyramid and a directional filter bank (DFB). Bandpass images from the Laplacian pyramid are fed into the DFB so that directional information can be captured. The low frequency components are separated from the directional components. After decimation, the decomposition is iterated using the same DFB. Its redundancy ratio is less than 4/3 because the directional sub-bands are also decimated [16].

The Gabor Wavelet Transform. To obtain a Gabor filter bank with K orientations and S scales, the two-dimensional Gabor function is dilated and rotated appropriately by setting the parameters of the Gabor function (thus obtaining K*S Gabor functions). The image is then convolved with each of the obtained Gabor functions. It has been shown that the Gabor Transform for texture image retrieval yields the highest texture retrieval results. However, it results in an over-complete representation of the original image with a redundant ratio of K*S [16].

The Complex Directional Filter Bank. The shift-invariant complex directional filter bank (CDFB) consists of a Laplacian pyramid and a pair of directional filter banks, designated as primal and dual filter banks. The filters of these filer banks are designed to have special phase functions so that the overall filter is the Hilbert transform of the primal filter bank. A multi-resolution representation is obtained by reiterating the decomposition at the low-pass branch. The attractive features of the CDFB are its shift invariance, its comparably high texture retrieval performance and its relatively low redundancy ratio. The over-complete ratio of the CDFB is bounded by 8/3, whereas those of the Gabor transform and steerable pyramid increase linearly with the number of directional sub-bands [16].

Autoregressive and random field texture models. As described in [15], one can think of a textured image as a two-dimensional array of random numbers. Then, the pixel intensity at each location is a random variable. One can model the image as a function f (r, ω), where r is the position vector representing the pixel location in the 2D space and ω is a random parameter. For a given value of r, f (r, ω) is a random variable (because ω is a random variable). Once we select a specific texture ω, f (r, ω) is an image, namely, a function over the two-dimensional grid indexed by r. f (r, ω) is called a random field. Thus, one can think of a texture-intensity distribution as a realization of a random field. Random field models (also referred to as spatial-interaction models) impose assumptions on the intensity distribution.

A special case of the Markov random field (MRF) that has received much attention in the image retrieval community is the Simultaneous Autoregressive Model (SAR). In order to define an appropriate SAR model, it has to be determined the size of the neighborhood. This is a nontrivial problem, and often, a fixed-size neighborhood does not represent all texture variations very well. In order to address this issue, the Multiresolution Simultaneous Autoregressive (MRSAR) model has been proposed. The MRSAR model tries to account for the variability of texture primitives by defining the SAR model at different resolutions of a Gaussian pyramid. Thus, three levels of the Gaussian pyramid, together with a second-order symmetric model, require 15(3×5) parameters to specify the texture [15].

2.1.3 Shape Feature Descriptors

Of all the different primitive features, shape is used the least in content-based image retrieval. Shape can be characterized fairly easy (shape is the outline of a structure) but it is difficult to represent shape in such a way that perceptually similar shapes are close to each other based on some distance metric in feature space. Nevertheless, shape has long been recognized as an important feature for describing and differentiating objects in pictures [3]. Two major steps are involved in shape feature extraction. They are object segmentation and shape representation. Only approximate representations of shape are practically usable for image retrieval. [15].

Shape has been represented using a variety of descriptors such as moments, Fourier descriptors (FD), geometric and algebraic invariants, polygons, polynomials, splines, strings, deformable templates, skeletons, and so on. Each of these representations aims at capturing specific perceptually salient dimensions of the qualitative aspects of shape. Because of the heterogeneous nature of the aspects captured, it is not possible to compare different descriptors outside the context of very specific applications [15].

Most shape features used by CBIR systems are circularity, eccentricity, major axis orientation and algebraic moment. Sometimes differences between objects of the same type are due to changes in viewing geometry or they are due to physical deformation. One object, for example, can be a stretched, bent, tapered or dented version of the other. To describe these deformations, therefore, it is reasonable to model the physics by which real objects deform, and then to use that information to guide the matching process. In general, most CBIR systems using shape-based similarity assume that objects are simple, for example they are composed of only one homogeneous part [9].

Boundary vs. Interior. Two large categories of shape descriptors can be identified: those capturing the boundary (or contour appearance) and those characterizing the interior region. Boundary representations emphasize the closed curve that surrounds the shape. This curve has been described by numerous models, including chain codes, polygons, circular arcs, splines, explicit and implicit polynomials, and boundary Fourier descriptors. Alternately, a boundary can be described by its features, for example, curvature extrema and inflection points [15].

Interior descriptions of shape, on the other hand, emphasize the “material” within the closed boundary. The interior has been modeled in a variety of ways, including collections of primitives (rectangles, disks, superquadrics, etc.), deformable templates, by modes of resonance, skeletal models, or simply as a set of points (as in mathematical morphology) [15].

Global vs. Local. Shape can also be viewed either from a local or from a global perspective. Many early models in indexing by shape content used features such as moments, eccentricity, area, and so on, which are typically based on the entire shape and are thus global. Similarly, Fourier descriptors of two-dimensional shape are global descriptors. On the other hand, local representations restrict computations to small neighborhoods of the shape. For example, a representation based on curvature extrema and inflection points of the boundary is local [15]. Purely global representations are affected by variations, such as partial occlusion and articulation, whereas purely local representations are sensitive to noise [15].

Composition of Parts vs. Deformation. Shape can also be viewed either as the composition of simpler, elementary parts, or as the deformation of simpler shapes. In the “part-based view,” shapes are composed of simple components (e.g. a hand is seen as four fingers and one thumb attached to a palm). Superquadrics represent a rich space of shape primitives from which to choose.

Shape can also be decomposed into parts based on “local” evidence. Properties of the boundary belong to this category. For example, the boundary can be decomposed into codons along negative minima of its curvature or by taking into account regional properties, such as good continuation of tangents. The latter approach has been shown to produce parts that are perceptually meaningful [15].

Topology-preserving shape representation. In [17], an interesting approach has been proposed for extracting contours with topological control. The topological approach consists of decomposing an image as a cell complex, handling each cell of the decomposition by topological operators. They employ the triangular cell decomposition (simplicial complex), as it allows a better contour definition. An advantage of handling an image as a cell complex is that regions of interest in the image can be represented by a set of cell, enabling the contour extraction from the boundary edges of the cells. Another advantage is that the topology of the regions can be controlled in a straightforward way, as topological operators can be employed to glue the cells that comprise the region.

The contour extraction starts by modeling an object according to user input parameters that define a target threshold range, the number of components and the number of holes. The threshold range identifies which triangles of *T* will belong to the final object *O*, and the number of components and holes defines the topology of the object *O*. Constructing the mesh using the Morse operators enables the system to keep track of the exact number of connected components and holes after each triangle insertion [17].

Once the object is modeled, its contours can be extracted by walking along the boundary edges, i.e., edges which are shared by only one triangle [17].

2.1.4 Other Feature Descriptors

Distribution-based descriptors. These techniques use histograms to represent different characteristics of appearance or shape. A simple descriptor is the distribution of the pixel intensities represented by a histogram.

An approach robust to illumination changes relies on histograms of ordering and reciprocal relations between pixel intensities which are more robust than raw pixel intensities. The binary relations between intensities of several neighboring pixels are encoded by binary strings and a distribution of all possible combinations is represented by histograms. This descriptor is suitable for texture representation but a large number of dimensions is required to build a reliable descriptor.

The scale invariant feature transform (SIFT) combines a scale invariant region detector and a descriptor based on the gradient distribution in the detected regions [7]. The SIFT descriptor computes a gradient orientation histogram within the support region. For each of 8 orientation planes, the gradient image is sampled over a 4×4 grid of locations, thus resulting in a 4×4×8 = 128-dimensional feature vector for each region. A Gaussian window function is used to assign a weight to the magnitude of each sample point. This makes the descriptor less sensitive to the small changes in the position of the support region and puts more emphasis on the gradients that are near the center of the region. [8]

Gradient location-orientation histogram (GLOH) is proposed in [7] and is an extension of the SIFT descriptor, designed to increase its robustness and distinctiveness. They compute the SIFT descriptor for a log-polar location grid with 3 bins in radial direction (the radius set to 6, 11 and 15) and 8 in angular direction, which results 17 location bins. The central bin is not divided in angular directions. The gradient orientations are quantized in 16 bins. This gives a 272 bin histogram. The size of this descriptor is reduced with PCA. The covariance matrix for PCA is estimated on 47 000 image patches collected from various images. The 128 largest eigenvectors are used for description.

Geometric histogram and shape context descriptors implement the same idea and are very similar to the SIFT descriptor. Both methods compute a 3D histogram of location and orientation for edge points where all the edge points have equal contribution in the histogram [7].

PCA-SIFT descriptor is a vector of image gradients in x and y direction computed within the support region. The gradient region is sampled at 39x39 locations therefore the vector is of dimension 3042. The dimension is reduced to 36 with PCA. These descriptors were successfully used, for example, for shape recognition of drawings for which edges are reliable features. [7]

The SPIN descriptor, based on *spin images* used for matching range data, is a rotation-invariant two-dimensional histogram of intensities within an image region. The two dimensions of the histogram are *d*, the distance of the center, and *i*, the intensity value. The entry at (*d*,*i*) is simply the probability of the occurrence of pixels with intensity value *i* at a fixed distance *d* from the center of the patch. [8]

The RIFT descriptor is a rotation-invariant version of SIFT. An image region is divided into concentric rings of equal width, and a gradient orientation histogram is computed within each ring. To obtain rotation invariance, gradient orientation is measured at each point relative to the direction pointing outward from the center. [8]

In [8], to obtain robustness to illumination changes, the descriptors were made invariant to affine illumination transformations of the form $aI(x)+b$. For SPIN and RIFT descriptors each support region is normalized with the mean and standard deviation of the region intensities. For SIFT descriptors the norm of each descriptor is scaled to one.

In [18], is proposed the SURF descriptor. They construct a square region centered around the interest point and split it up regularly into smaller 4 × 4 square sub-regions. For each sub-region, they compute a few simple features at 5×5 regularly spaced sample points. The Haar wavelet response in horizontal direction d_x and the Haar wavelet response in vertical direction d_y are summed up over each sub-region and form a first set of entries to the feature vector. In order to bring in information about the polarity of the intensity changes, they also extract the sum of the absolute values of the responses, $|d_x|$ and $|d_y|$. Hence, each sub-region has a four-dimensional descriptor vector for its underlying intensity structure, which results in a descriptor vector for all 4×4 sub-regions of length 64. The wavelet responses are invariant to a bias in illumination (offset). Invariance to contrast (a scale factor) is achieved by turning the descriptor into a unit vector.

Spatial-frequency techniques. Many techniques describe the frequency content of an image. The Fourier transform decomposes the image content into the basis functions. However, in this representation the spatial relations between points are not explicit and the basis functions are infinite, therefore is difficult to adapt to a local approach. The Gabor transform overcomes these problems, but a large number of Gabor filters is required to capture small changes in frequency and orientation. Gabor filters and wavelets are frequently explored in the context of texture classification [7].

Differential Descriptors. A set of image derivatives computed up to a given order approximates a point neighborhood. Differential invariants combine components of the local jet to obtain rotation invariance. Steerable filters steer derivatives in a particular direction given the components of the local jet. Steering derivatives in the direction of the gradient makes them invariant to rotation. Steerable filters and differential invariants use derivatives computed by convolution with Gaussian derivatives of σ=6.7 for an image patch of size 41. Changing the orientation of derivatives gives equivalent results to computing the local jet on rotated image patches [7].

Moment invariants. Generalized moment invariants have been introduced to describe the multi-spectral nature of the image data. The moments characterize shape and intensity distribution in a region. They are independent and can be easily computed for any order and degree. However, the moments of high order and degree are sensitive to small geometric and photometric distortions. Computing the invariants reduces the number of dimensions. These descriptors are therefore more suitable for color images where the invariants can be computed for each color channel and between the channels [7].

Multiresolution histograms. The multiresolution decomposition of an image is computed with Gaussian filtering. The image at each resolution gives a different histogram. The multiresolution histogram, H, is the set of intensity histograms of an image at multiple image resolutions. In [19], the multiresolution decomposition of an image is implemented with a pyramid for efficiency. The multiresolution histogram can be computed and stored efficiently also. Moreover, it can also be matched very fast using the L1 norm. The multiresolution histogram not only combines intensity with spatial information, but it also preserves the efficiency, simplicity, and robustness of the plain histogram. The feature proposed in [19] uses the histogram of the original image together with the differences between histograms of consecutive image resolutions.

2.1.5 Comparison of some feature descriptors

In the evaluation performed by [7], the GLOH descriptor obtained the best results in most of the tests, closely followed by SIFT; however, GLOH is computationally more expensive. These results can be explained by the fact that they capture a substantial amount of information about the spatial intensity patterns, while at the same time being robust to small deformations or localization errors [18]. This shows the robustness and the distinctive character of the region-based SIFT descriptor. Shape context also shows a high performance. However, for textured scenes or when edges are not reliable its score is lower.

According to [7], the best low dimensional descriptors are gradient moments and steerable filters. They can be considered as an alternative when the high dimensionality of the histogram-based descriptors is an issue. Differential invariants give significantly worse results than steerable filters, which is surprising as they are based on the same basic components (Gaussian derivatives). The multiplication of derivatives necessary to obtain rotation invariance increases the instability.

Cross correlation gives unstable results. The performance depends on the accuracy of interest point and region detection, which decreases for significant geometric transformations. Cross correlation is more sensitive to these errors than other high dimensional descriptors. [7]

According to [8]**,** SIFT performs better than SPIN and RIFT, while the performance rank between SPIN and RIFT depends on dataset. It is not surprising that RIFT performs worse than SIFT, since it averages gradient orientations over a ring-shaped region and therefore loses important spatial information.

Combining SIFT with SPIN and RIFT with SPIN boosts the overall performance because the two descriptors capture different kinds of information (gradients vs. intensity values). [8]

For content based image retrieval good response times are required and this is hard to achieve using the huge amount of data obtained by local features. The dimension of the SIFT descriptor is extremely high because the size of the key-point descriptor is 128 dimensional vector. In [20], to reduce the dimensionality, they use histograms of local features. With this approach the amount of data is reduced by estimating the distribution of local features for every image. The creation of these histograms is a three step procedure. First, the key-points are extracted from all database images, where a key-point is described with a 128 vector of numerical values. The key-points are then clustered in 2000 clusters. Afterwards, for each key-point they discard all information except the identifier of the most similar cluster center. A histogram of the occurring patch-cluster identifiers is created for each image. This results in a 2000 dimensional histogram per image [20].

Similarity Measures

After having selected the right set of features and having characterized an image as a point in a suitable vector space, we can make some uncritical and unwarranted assumptions about the metric of the space. Typically, the feature space is assumed to be Euclidean. A similarity function is a mapping between pairs of feature vectors and a positive real-valued number, which is chosen to be representative of the visual similarity between two images [15].

A number of similarity measures proposed in the literature explain similarity as a distance in some suitable feature space that is assumed to be a metric space [6].

Several researchers have designed similarity measures that operate on sets of unordered features. One proposition is a kernel that averages over the similarities of the best matching feature found for each feature member within the other set [21]. Another similar kernel also considers all possible matchings between features but measures overall similarity with a different bias, by raising the similarity between each pair of features to a given power. Both approaches have a computational complexity that is squared in the number of features. Furthermore, both match each feature in a set independently, ignoring potentially useful co-occurrence information.

2.1.6 Common Similarity Measures

Feature vector histograms can be compared using various similarity measures. Well-known measures are based on the L_1 norm distance, L_2 norm distance, Standard Euclidean (SE) distance, Cosine (Cos) distance and Correlation (Cor) distance. For two histograms $H_i(b)$ and $H_j(b)$, with bins numbered as b=1,2,. . .,L, the similarity measures are defined as follows [22].

L_1 and L_2 based measures.

$$L_1(i,j) = \sum_{b=1}^{L} \left|H_i(b) - H_j(b)\right| . \tag{2.1}$$

$$L_2(i,j) = \sum_{b=1}^{L} (H_i(b) - H_j(b))^2 . \tag{2.2}$$

L_1 norm gives overall best performance. The possible reason is that the nature of L_1 norm fits well with the intrinsic property of the optimization method in which histograms bins are ordered and their length adjusted to provide best performance. The L_1 norm also has lower computational complexity and good accuracy. [22]

Standard Euclidean measure.

$$SE(i,j) = \sum_{b=1}^{L} \left(\frac{H_i(b) - H_j(b)}{\sigma(b)}\right)^2 \tag{2.3}$$

Where $\sigma(b)$ represents the variance of all histograms at b-th bin.

Cosine distance.

$$Cos(i,j) = \sum_{b=1}^{L} \frac{H_i(b)H_j(b)}{\sqrt{\|H_i\|\|H_j\|}} \tag{2.4}$$

Where $\|H_i\|$ and $\|H_j\|$ are histogram length in the L_2 norm.

Correlation distance.

$$Cor(i,j) = \sum_{b=1}^{L} \frac{(H_i(b) - \overline{H_\iota})(H_j(b) - \overline{H_J})}{\sqrt{\|H_i - \overline{H_\iota}\|\|H_j - \overline{H_J}\|}} \tag{2.5}$$

Earth Mover's Distance (EMD). Given the signature of an image as a set of descriptors $= \{(\boldsymbol{p_1}, \boldsymbol{u_1}), \dots, (\boldsymbol{p_m}, \boldsymbol{u_m})\}$, where *m* is the number of image clusters, p_i is the center of the *i*-th cluster, and u_i is the relative size of the cluster (the number of descriptors in the cluster divided by the total number of descriptors extracted from the image), the Earth Mover's Distance between two signatures $\boldsymbol{S_1} = \{(\boldsymbol{p_1}, \boldsymbol{u_1}), \dots, (\boldsymbol{p_m}, \boldsymbol{u_m})\}$ and $\boldsymbol{S_2} = \{(\boldsymbol{q_1}, \boldsymbol{w_1}), \dots, (\boldsymbol{q_m}, \boldsymbol{w_m})\}$ is defined as follows:

$$D(S_1, S_2) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d(p_i, q_i)}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}} \qquad 2.6$$

Where f_{ij} is a flow value that can be determined by solving a linear programming problem and $d(p_i, q_j)$ is the ground distance (i.e. Euclidean distance) between cluster centers p_i and q_j.

Intuitively, EMD measures of the amount of work necessary to transform one weighted point set into another. EMD is a cross-bin dissimilarity measure and can handle variable-length representation of distributions, i.e., m and n do not have to be the same. [8] EMD was first used in vision to measure the distance between intensity images. More recently EMD has been used for global color- or texture-based similarity, and for comparing vector-quantized signatures of affine invariant features in texture images. [23]. The time complexity of EMD is larger than $O(N^3)$ where *N* is the number of histogram bins. [24]

$\boldsymbol{\chi^2}$ **distance.** To compare two histograms $\boldsymbol{S_1} = (\boldsymbol{u_1}, \dots, \boldsymbol{u_m})$ and $\boldsymbol{S_2} = (\boldsymbol{w_1}, \dots, \boldsymbol{w_m})$ the $\boldsymbol{\chi^2}$ distance is defined as **[8]**:

$$D(S_1, S_2) = \frac{1}{2} \sum_{i=1}^{m} \frac{(u_i - w_i)^2}{u_i + w_i} \qquad 2.7$$

Comments regarding these measures. The most often used bin-to-bin distances between histogram-based local descriptors (e.g. χ^2 statistics, L_2 distance and Kullback-Leibler divergence) assume that the histograms are already aligned, so that a bin in one histogram is only compared to the corresponding bin in the other histogram. These methods are sensitive to distortions in histogram-based local descriptors as well as quantization effects. Cross-bin distances, such as the *Earth Mover's Distance* (EMD), allow bins at different locations to be (partially) matched and therefore alleviate the quantization effect [24].

The *L1* norm has the lowest computational complexity. However, it could produce false negatives (not all similar images are retrieved). The *L2* norm (i.e., the Euclidean distance) is probably the most widely used metric. However, it can result in false positives (dissimilar images are retrieved) [15].

These norms can be applied to measure similarity for many kinds of descriptors, but in the case of shape descriptors, the situation is different. Ideally, the representation should map each shape to a vector of numbers in such a way that the Euclidean distance between pairs of vectors indicates shape dissimilarity. Unfortunately, shape comparison is inherently complex and current representations do not allow for such a mapping. Thus, the role of the matching process is to define such a metric [15].

Boundary-based and region-based representations inherently lead to different matching procedures. Boundary-based techniques are typically accompanied by curve-based comparisons. In these approaches two curves are compared based on their properties, such as curvature, resulting in a single similarity measure [15].

Some approaches include using line primitives to describe the curve and then using the length and absolute orientation of the primitives to measure curve similarity, others have proposed an approach to measuring curve similarity based on an initial alignment [15].

Region-based representations typically involve trees and have relied on such methods as graph-tree matching, string edit distance, graduated assignment, tree edit distance, eigenvalue decomposition, Bayesian matching, containment tree matching, and so on. Other region-based techniques, such as modal matching and deformable prototypes, allow for a global to local ordering of shape deformations [15].

2.1.7 Classification Kernels

Pyramid Matching Kernel. The spatial pyramid matching method proposed in [21] works by placing a sequence of increasingly coarser grids over the feature space and taking a weighted sum of the number of matches that occur at each level of resolution. At any fixed resolution, two points are said to match if they fall into the same cell of the grid; matches found at finer resolutions are weighted more highly than matches found at coarser resolutions.

More specifically, let X and Y be two sets of vectors in a d-dimensional feature space and let us construct a sequence of grids at resolutions $0 \ldots L$, such that the grid at level ℓ has 2^{ℓ} cells along each dimension, for a total of $D = 2^{d\ell}$cells. Let H_x^{ℓ} and H_y^{ℓ} denote the histograms of X and Y at this resolution, so that $H_x^{\ell}(i)$ and $H_y^{\ell}(i)$ are the numbers of points from X and Y that fall into the i-th cell of the grid. Then the number of matches at level ℓ is given by the *histogram intersection* function:

$$I\left(H_X^l, H_Y^l\right) = \sum_{i=1}^{D} \min\left(H_X^l(i), H_Y^l(i)\right) . \qquad 2.8$$

The number of matches found at level ℓ also includes all the matches found at the finer level $\ell + 1$. So, the number of new matches found at level ℓ is given by $I^{\ell} - I^{\ell+1}$ for $\ell = 0, \ldots, L - 1$. The weight associated with a level is inversely proportional to cell width at that level, to penalize matches found in larger cells, since they involve increasingly dissimilar features. A pyramid match determines a partial correspondence by matching points once they fall into the same histogram bin.

The similarity between the two sets is measured by the sum of the weighted number of new matches found at each level in the pyramid:

$$\widetilde{K_{\Delta}}\left(\Psi(y), \Psi(z)\right) = \sum_{i=0}^{l} \frac{1}{2^i}\left(I\left(H_i(y), H_i(z)\right) - I\left(H_{i-1}(y), H_{i-1}(z)\right)\right) . \qquad 2.9$$

By construction, the pyramid match offers an approximation of the optimal correspondence-based matching between two feature sets, in which the overall similarity between corresponding points is maximized. The pyramid match kernel allows precise matching of two collections of features in a high dimensional appearance space, but discards all spatial information [25] [21].

In [25], the approach of the pyramid matching kernel is used but instead of performing the matching in the high dimensional feature space, it does it in the two-dimensional image space, and use traditional clustering techniques in feature space. In this approach they quantify all feature vectors into M discrete types, and make the assumption that only features of the same type can be matched to one another. Each channel m defines two sets of two-dimensional vectors, X_m and Y_m, representing the coordinates of features of type m found in the respective images. The final kernel is the sum of the separate channel kernels:

$$K^L(X, Y) = \sum_{m=1}^{M} k^L(X_m, Y_m) . \qquad 2.10$$

They implement K^{L} as a single histogram intersection of “long” vectors formed by concatenating the appropriately weighted histograms of all channels at all resolutions.

This approach maintains continuity with the "visual vocabulary" paradigm [25].

EMD and χ^2 Kernels. In [8] they use SVM for classification, with the pair-wise coupling method, which trains a classifier for each possible pair of classes. To incorporate EMD or $\boldsymbol{\chi^2}$ distance into the SVM framework, they use the extended Gaussian Kernels:

$$K(S_1, S_2) = \exp\left(-\frac{1}{A}\right) D(S_1, S_2)\,. \qquad 2.11$$

Where$D(S_i, S_j)$is EMD (or χ^2 distance) if S_i and $S_j$are image signatures (or vocabulary-histograms). The resulting kernel is the EMD kernel (or χ^2kernel). A is a scaling parameter that can be determined through cross-validation.

3 Topological Relationships between Regions

Representing spatial relationships seems to be a very promising approach in CBIR systems, especially when we are dealing with multi-object images. It is important to model how these objects are distributed and related within the image, since this can be a source of important information in the retrieval process.

So far, there are two kinds of representation for spatial relationship features [5]:

- Topological relationships, which are invariant under topological transformation of the referenced objects, such as translation, rotation and scaling.
- Orientation relationships, which concern partial and total orientation relationships among objects. Basically they describe where objects are placed relative to one another.

Topological relationship has been intensively studied; in particular, it has been applied in GIS due to its invariance under transformation [5]. Topological relations between objects are those relations that remain invariant under continuous transformations (i.e. translation, rotation, scaling, bending), so this kind of spatial information can be very important for developing descriptors invariants to these transformations.

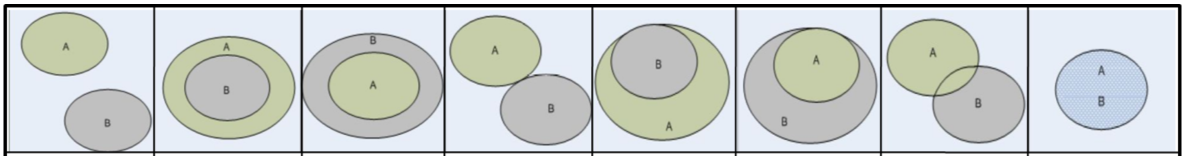
Several models have been proposed to represent topological relationships between regions. The best known models are the 4-Interesection Model and 9-Interesection Model.

4-Intersection Model

Binary topological relations between two objects, *A* and *B*, are defined in terms of the 4 intersections of *A*'s boundary (*∂A*) and interior (*A*) with *B*'s boundary (*∂B*) and interior (*B*). This model is concisely represented by a 2 X 2 matrix, called the 4-intersections [26]:

$$\Im_4(A, B) = \begin{bmatrix} A \cap B & A \cap \partial B \\ \partial A \cap B & \partial A \cap \partial B \end{bmatrix} \qquad 3.1$$

By considering the values empty (∅) and non-empty (¬∅) for the 4 intersections, one can distinguish 2^4=16 binary topological relations. Eight of these sixteen relations can be realized for homogeneously 2-dimensional objects with connected boundaries, called regions if the objects are embedded in $\mathbb{R}^2$. This can be seen in Fig. 1**:** Eight topological relations between two regions *A* and *B.*



A and B are disjoint	A contains B	A is inside of B	A is touching B	A covers B	A is covered by B	A and B are overlaped	A and B are equal
$\begin{bmatrix}0&0&1\\0&0&1\\1&1&1\end{bmatrix}$	$\begin{bmatrix}1&1&1\\0&0&1\\0&0&1\end{bmatrix}$	$\begin{bmatrix}1&0&0\\1&0&0\\1&1&1\end{bmatrix}$	$\begin{bmatrix}0&0&1\\0&1&1\\1&1&1\end{bmatrix}$	$\begin{bmatrix}1&1&1\\0&1&1\\0&0&1\end{bmatrix}$	$\begin{bmatrix}1&0&0\\1&1&0\\1&1&1\end{bmatrix}$	$\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$	$\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}$

Fig. 1: Eight topological relations between two regions A and B

This model has the advantage of being simple and is well accepted, but has some disadvantages [26]:

- Does not distinguish between conceptually different situations (different situations may correspond to the same matrix)
- Considers only region/region relations.

9-Intersection Model

The 4-intersection model is extended by considering the location of each interior and boundary with respect to the other object's exterior, therefore , the binary topological relation between two objects, A and B, in $\mathbb{R}^2$ is based upon the intersection of A's interior (A), boundary (∂A) and exterior (A^-) with B's interior (B), boundary (∂B) and exterior (B^-). The nine intersections between the six object parts describe a topological relation and can be concisely represented by a 3 X 3 matrix [26]:

$$\mathfrak{I}_9(A,B) = \begin{bmatrix} A \cap B & A \cap \partial B & A \cap B^- \\ \partial A \cap B & \partial A \cap \partial B & \partial A \cap B^- \\ A^- \cap B & A^- \cap \partial B & A^- \cap B^- \end{bmatrix} \quad 3.2$$

In analogy to the 4-intersection, each intersection will be characterized by a value empty ($\emptyset$) and non-empty ($\neg\emptyset$), which allows one to distinguish 2^9=512 different configurations. Only a small subset of them can be realized between two objects in $\mathbb{R}^2$.

This model is potentially more expressive than the 4-intersection model and considers topological relations between generic sets of spatial entities (not just region/region relations) and relationships between region and embedding space [26]. The 9-intersection model is reduced to the 4-intersection model relations in 2 cases:

- If two objects are simply connected, their boundaries form Jordan Curves and the objects have co-dimension 0.
- If two objects are simply connected, each boundary forms a separation and the objects have co-dimension 0.

Disadvantages of the 9-intersection model [27]:

- It fails to distinguish certain disjoint relations, i.e. when the infinitive exterior of the objects cannot play roles in distinguishing disjoint relations.
- It fails to identify the topological relations between two objects with holes (this model can only deal with simple entities such as homogeneous, 2 dimensional and connected areas, and lines with exactly two end points)
- It is difficult or impossible to compute the intersections with an entity's complement since the complements are infinite

Voronoi-Based 9-Intersection Model

The 9-intersection model has problems both in theory and in practice, for example, difficulty in distinguish several disjoint relations and relations with complex entities with holes; and difficulty or impossibility of computing the intersections with an entity's complement, since the complements are infinite. This approach is intended to deal with these handicaps by using Voronoi's regions. The Voronoi region of an entity is defined as the area containing all locations closer to itself than to any other [27].

By replacing the complement of an object with its Voronoi region, the new 3 X 3 matrix can be formulated as:

$$\begin{bmatrix} A \cap B & A \cap \partial B & A \cap B^V \\ \partial A \cap B & \partial A \cap \partial B & \partial A \cap B^V \\ A^V \cap B & A^V \cap \partial B & A^V \cap B^V \end{bmatrix} \qquad 3.3$$

Where Av is object A's Voronoi region and Bv is object B's Voronoi region.

Here, each object has limited neighbors instead of having relations with all other objects. The disjoint relationships and relationships between complexes objects can be distinguished. It is possible and easier to compute the five exterior-based intersections since the Voronoi regions of each object can be generated and manipulated. [27]

4 Models for Representing Spatial Relationships and Topology.

Many vision problems may be formulated in an abstract setting with solid theoretical foundations from graph theory. Additionally, graphs allow abstracting the exact shape and positioning of the objects under study and thus improve the potential applications and the efficiency of many graph based vision algorithms. The use of graphs within the computer vision framework is not new. However, there is a growing interest toward an explicit formulation of vision problems as graph problems. The recent trends in this field concern: the graph partitioning, graph indexing, graph matching and clustering and the graph generalization problems. The potential applications of such problems are respectively the segmentation, the image data base retrieval, the object recognition and classification and the object recognition and modeling [28].

However, graphs used in vision may be quite big and many graph algorithms have a high computational cost. For example, a simple graph encoding a 512×512 regular grid is defined by O(512^2) vertices and edges. In the same way the brute force approach of graph matching requires a computational cost of O(*n*!) where *n* is the number of vertices [28].

Curvature Tree

In [1], the authors propose an image representation based in a curvature tree (**CT**). The CT is a rooted, directed, unordered, and acyclic graph *T=(V, E),* where *V* is a set of nodes and *E* is a set of edges. The CT has a single root node at level 0, representing the background of the image, the external contour of the primary objects are stored at the first level nodes, and the contours of possible holes are at the second level nodes, and so on. An example is shown in Fig. 2**:** Curvature tree representation (b) of a multi-object image (a).

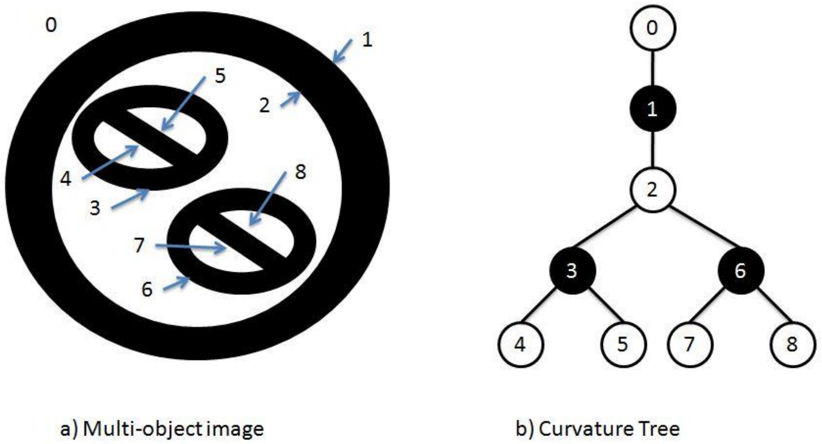


Fig. 2: Curvature tree representation (b) of a multi-object image (a)

Triangle-area representation (TAR) of each closed boundary of an object or hole is stored at the corresponding node. Although this tree hierarchy reflects the inclusion relationships between the objects and holes, the adjacency relationships are not represented.

Region Adjacency Graph (RAG)

This representation is defined from a given partition of the image, by associating one vertex to each region and by creating an edge between two vertices if the associated regions share a common boundary. This corresponds to a simple graph without any double edge between vertices nor self-loops.

The RAG model is usually applied as a merging step to overcome the over-segmentation produced by a previous splitting algorithm. In this case, the edge information may be interpreted as a possibility to merge the two regions identified by the vertices incident to the edge. Such a merge operation implies to collapse the two vertices incident to the edge into one vertex and to remove this edge together with any double edge between the newly created vertex and the remaining vertices.

The RAG model encodes only the existence of a common edge between two regions, and this does not provide enough information to differentiate a *meets* relationship from a *contains* or *inside* one [29].

The edges of the RAG implicitly encode the adjacency relationship between regions, but in some approaches, such as [30], the edges have been explicitly labeled with the relationships between regions. In this case, they consider topological relations (adjacent and disjoint) and order relations (beside, horizontally align, vertical aligned, above and below), grouped in 3 sets: topological relations, horizontal relations and vertical relations. Using this encoding, an edge may represent several relationships, for example “adjacent, above”, instead of the single adjacency relationship encoded by the traditional RAG.

Regular Pyramids

Regular image pyramids are a sequence of images with decreasing resolution. Each image of this sequence is called a level of the pyramid. Using the neighborhood relationships defined on each image, the reduction window relates each pixel of the pyramid with a set of pixels defined in the level below [31]. An example regular pyramid representation can be seen in Fig. 3**:** Regular pyramid representation.

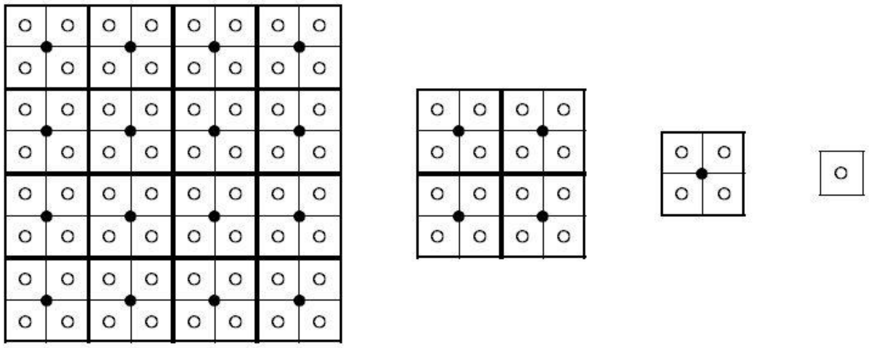


Fig. 3: Regular pyramid representation

Regular pyramids have been widely used in image segmentation, shape analysis, surface reconstruction and motion analysis. However, the rigidity of the vertical structure of regular pyramids induces several drawbacks, such as the shift-dependence and scale-dependence problem, and the limited number of regions encoded at a given level of the pyramid. The fixed size and shape of the reduction window together with the fix value of the reduction factor induce a poor ability of regular pyramids to adapt their structure to the data [31].

Irregular Graph Pyramids

Not using the regular pyramid approach implicates that the horizontal and vertical neighborhood relations need to be explicitly represented. This can be achieved by using a region adjacency graph (RAG). In these graphs $G = (V, E)$ the vertices represent the cells or regions, and the edges represent the neighborhood relations of the regions. The graph content is stored in attributes attached to both vertices and edges (i.e. color, size, gray values of the pixels, a weight measuring the difference between the two end points).

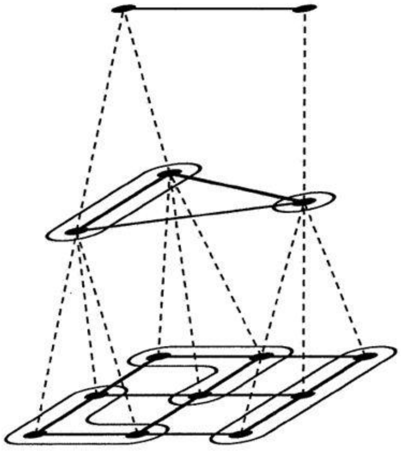


Fig. 4: Irregular pyramid representation

The irregular graph pyramid is then a stack of successively reduced graphs (being the base level the high resolution input image). Each graph is built from the graph below by selecting a set of vertices named surviving vertices and mapping each non surviving vertex to a surviving one. Therefore each non-surviving vertex is the child of a surviving one which represents all the non-surviving vertices mapped to it and becomes their father (See Fig. 4: Irregular pyramid representation) [32]. The models based on the irregular pyramid framework encode naturally the *composed of* relationship [29].

Roughly speaking, the decimation process to build the levels can be summarized in the following steps [31]:

1 The selection of a set S of surviving vertices
2 The definition of a set of edges N linking each non-surviving vertex to its father.
3 The contraction of the set of edges N
4 The removal of all multiple edges and self-loops

The definitions of the sets S and N vary according to the considered method.

Using simple graphs (graphs without multiple edges and self-loops) as the levels of the pyramid, the encoding of the spatial structure of the image might not be accurate. By not allowing multiple edges, the presence of several disconnected boundaries between two regions cannot be represented (See Fig. 5**:** Inadequacy of simple graphs for representing certain topological configurations).

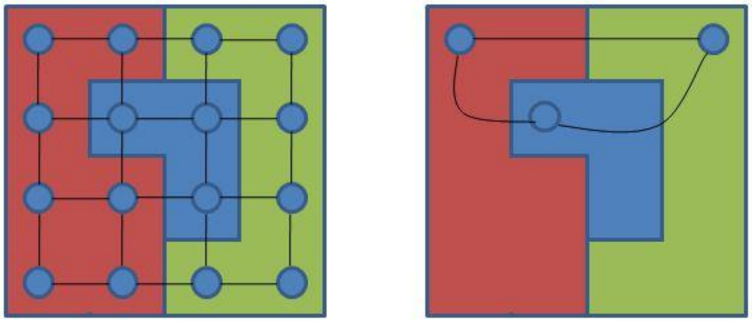


Fig. 5: Inadequacy of simple graphs for representing certain topological configurations

The left and right regions of the left image in Fig. 5**:** Inadequacy of simple graphs for representing certain topological configurations share two distinct boundaries. A reduction process such as the one defined in the stochastic and adaptative pyramids frameworks provides a final graph encoding these multiple boundaries by a single edge [31]. The lack of self-loops does not allow to differentiate inclusions from adjacencies relationships. The existence of a common edge between two vertices does not provide enough information to differentiate a *meets* relationship from a *contains* or *inside* one (See Fig. 6**:** In the ideal segmentation of these images, they are represented by the same *RAG*.). In this case, two different configurations may be encoded by the same RAG.

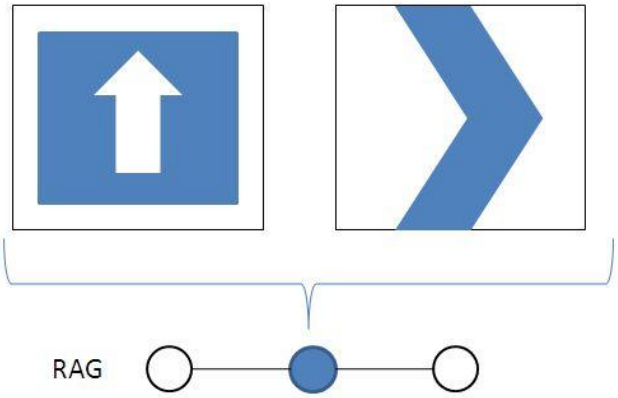


Fig. 6: In the ideal segmentation of these images, they are represented by the same *RAG*.

Dual Graph Pyramids

To overcome these problems, the dual graph pyramids are introduced. In order to correctly represent the embedding of the graph in the image plane, the dual graph $\bar{G} = (\bar{V}, \bar{E})$ of the RAG is additionally stored at each level. The RAG is also replaced by a RAG+ (enhanced region adjacency graph), which is a RAG that includes non-redundant self-loops or parallel edges [32]. Using the dual graph, the meaningful self-loops and parallel edges can be determined.

Given an initial planar graph G, the vertices of its dual $\bar{G}$ are located inside every face of G. The edges of $\bar{G}$ connect those dual vertices of which the corresponding faces are adjacent. The levels of the pyramid are represented as dual pairs $(G_k, \bar{G}_K)$. In this case, the edges of the dual graph represent the borders of the cells in each level k and the vertices represent meeting points of at least three edges from the original RAG [31]. The construction of the dual graph induces a one to one mapping between the edges of both graphs.

4.1.1 Contraction Kernels

Within the dual graph pyramid framework the reduction process is performed by a set of edge contractions. The edge contraction collapses two adjacent vertices into one vertex and removes the edge. Many edges except self-loops can be contracted independently of each other and also in parallel. This set is called a Contraction Kernel (CK) [29].

A CK is defined on a graph $G = (V, E)$ by a set of surviving vertices S and a set of non-surviving edges N such that [31]:

- (V, N) is a spanning forest of G
- Each tree of (V, N) is rooted by a vertex of S

The contraction of the graph reduces the number of vertices while maintaining the connections to other vertices. As a consequence, the decimation of a graph by a CK may induce the creation of some redundant edges. In [32], the authors propose using the Minimum Spanning Tree method (See Annex 1) to find the CKs (al the nodes that will be contracted into a single node) in each level.

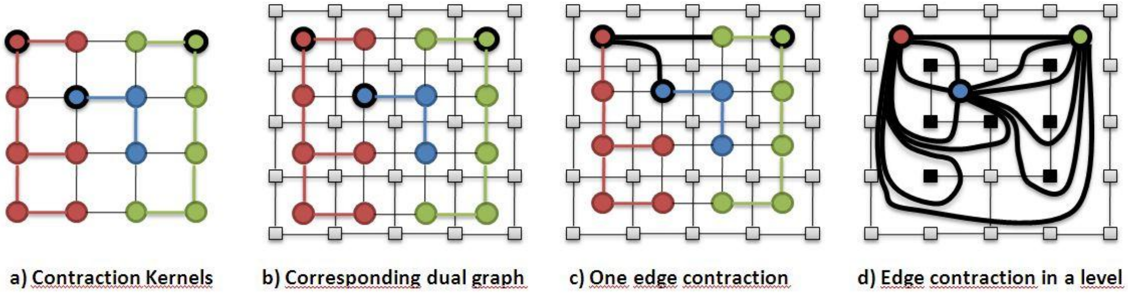

a) Contraction Kernels b) Corresponding dual graph c) One edge contraction d) Edge contraction in a level

Fig. 7: Edge contraction step. In (a), the surviving nodes of the primal graph are marked with a black border. In (b), the corresponding dual graph is shown. In (c), one edge has been contracted. In (d), all possible edge contractions are made; the dual nodes marked in black are the ones encoding redundant information.

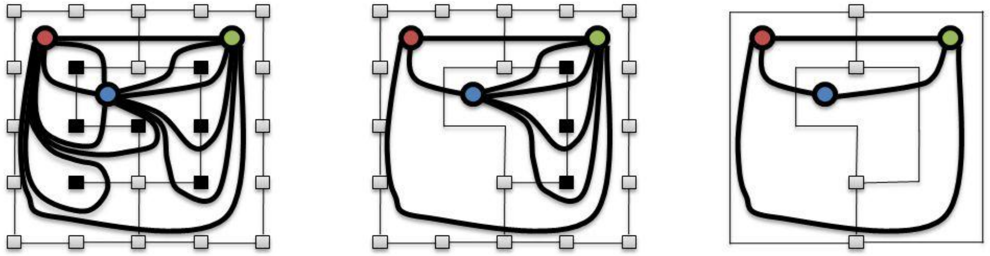
The contraction process must follow two steps [31]:

1 A set of edge contractions on G_0 encoded by the CK(S, N). The dual of the contracted graph G_1 is computed from $\overline{G}\ _0$ by removing the dual of the edges contained in N (See Fig. 7**:** Edge contraction step. In (a), the surviving nodes of the primal graph are marked with a black border. In (b), the corresponding dual graph is shown. In (c), one edge has been contracted. In (d), all possible edge contractions are made; the dual nodes marked in black are the ones encoding redundant information.).

2 The removal of redundant edges encoded by a CK applied on the dual graph. The edge contractions performed in the dual graph has to be followed by edge removals in the initial one in order to preserve the duality between the reduced graphs (See Fig. 8**:** Edge removal step. In (a) is shown the final state of the edge contraction step on the primal graph. In (b) the dual edges have been contracted and the corresponding primal edges have been removed. (c) represents the final state of the process. Meaningful double edges are maintained.).

The characterization of these edges requires a better description of the topological relationships between the objects described by the graph. The redundant edges belong to one of the following categories [29]:

- Redundant double edge: These edges encode multiple adjacency relationships between two vertices and define degree 2 faces. They can be characterized in the dual graph as degree 2 dual vertices. In terms of partition's encoding, these edges correspond to an artificial split of one boundary between two regions.
- Empty self-loop: These edges correspond to a self-loop with an empty inside. These edges define degree one faces and are characterized in the dual graph as degree one vertices. Such edges encode artificial inner boundaries of regions.



a) Edge contraction in a level. The dual vertices marked in black are the ones that correspond to redundant vertices in the primal graph. They must be contracted.

b) Dual edge contractions and primal edge removals.

c) Dual edges characterizing meaningful edges of the primal graph

Fig. 8: Edge removal step. In (a) is shown the final state of the edge contraction step on the primal graph. In (b) the dual edges have been contracted and the corresponding primal edges have been removed. (c) represents the final state of the process. Meaningful double edges are maintained.

The preservation of the meaningful edges by the two step strategy defined above induces a one to one mapping between the edges of the graph and the region boundaries [31].

Although the problem of having two regions with distinct meeting boundaries can be solved using the dual graph contraction method, the problem with the *contains* / *inside* relationships distinction remains unsolved. The encoding of the adjacency between two regions one inside the other can be encoded by two edges: one encoding the common border between the two regions and one self-loop incident to the vertex encoding the surrounded region (See Fig. 9**:** The dual graph does not represent properly the inclusion relationship) [29].

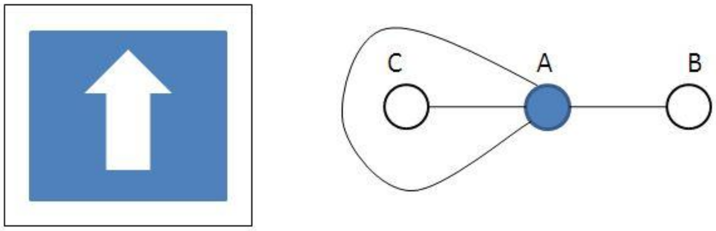


Fig. 9: The dual graph does not represent properly the inclusion relationship

However, one may exchange the surrounded vertex without modifying the incidence relationships between both vertices and faces. This last remark shows that the *contains*/*inside* relationship cannot be characterized locally within the dual graph [29].

4.1.2 Dual Graph Pyramids Drawbacks

Dual graph pyramids present some drawbacks that limit the relationships that can be represented and introduce problems for its computational implementation. These are:

- The contains/inside relationship cannot be encoded properly using the dual graph method.
- This approach needs an explicit encoding of the dual graph, therefore, two data structures must be encoded and maintained through the pyramid levels.

Combinatorial Maps and Combinatorial Pyramids

A Combinatorial Map (CM) may be understood as a planar graph encoding explicitly the orientation of edges called darts, each dart having its origin at the vertex it is attached to. A CM can be defined as $G = (D, \sigma, \alpha)$, where D is a set of darts (an edge connecting two vertices is composed of two darts d_1 and d_2, each dart belonging to only one vertex), α is the reverse permutation which maps d_1 to d_2 and d_2 to d_1 and σ is the successor permutation which encodes the sequence of darts encountered when turning around a vertex [29]. In Fig. 10**:** From a plane graph to a combinatorial map this transformation is shown.

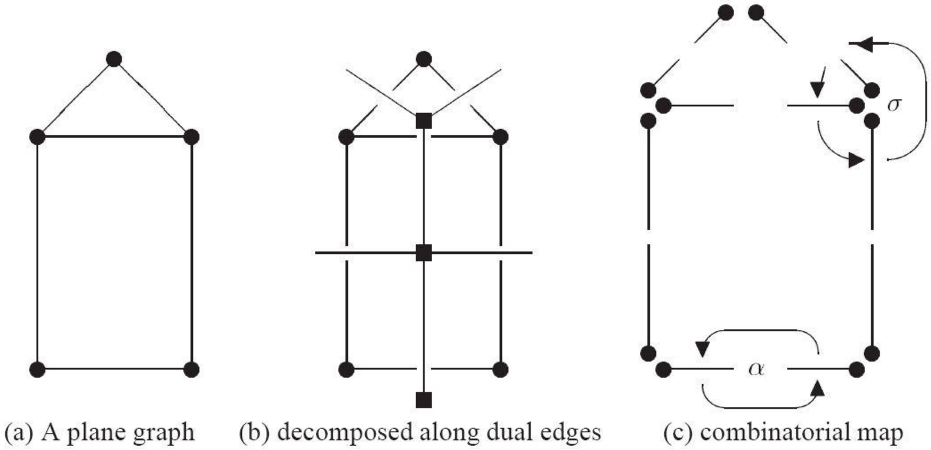
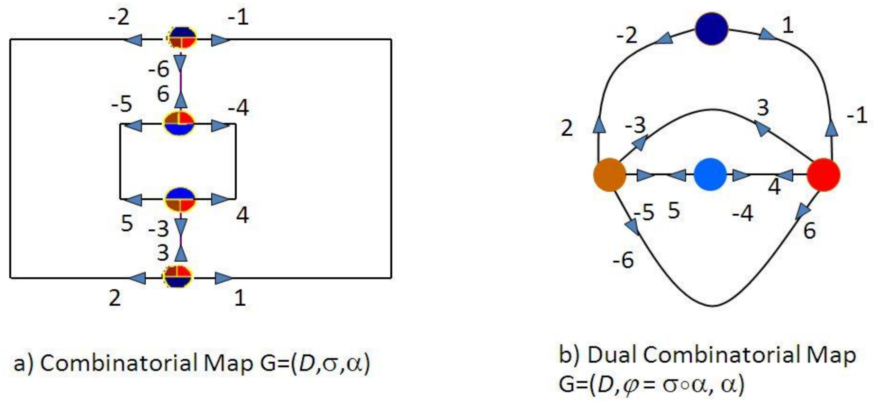


Fig. 10: From a plane graph to a combinatorial map

Given a combinatorial map $G = (D, \sigma, \alpha)$ its dual is defined by $G = (D, \varphi, \alpha)$ with $\varphi = \sigma \circ \alpha$. The cycles of the permutation φ encode the set of darts encountered when turning around a face of G (See Fig. 11: Combinatorial map and its corresponding dual map). The σ, α and φ cycles of a dart may be respectively understood as elements of dimensions 0, 1 and 2. One of the major differences between a combinatorial map and a usual graph encoding a partition is that, although a combinatorial map may be seen as a planar graph with a set of vertices (the cycles of σ or σ^*) connected by edges (the cycles of α or α^*), the combinatorial map encodes additionally the local orientation of edges around each vertex thanks to the order defined within each cycle of σ [29].



a) Combinatorial Map G=(D,σ,α)

b) Dual Combinatorial Map G=(D,φ= σ∘α, α)

$$\varphi = (1,-2)(-1,3,4,6)(-6,-5,-3,2)(-4,5)$$

Fig. 11: Combinatorial map and its corresponding dual map

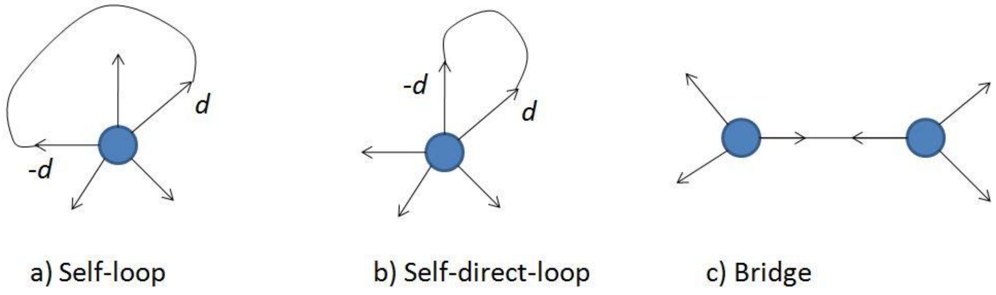
4.1.3 Contraction and Removal

Before explaining the contraction and removal process, a few definitions given by [33] are needed (See Fig. 12**:** Definitions within the combinatorial map framework for an illustration of each definition):

Self-loop: An edge $\alpha^*(d)$ is called a self-loop iff: $-d \in \sigma^*(d)$, which means that its corresponding dart in the α permutation is also part of the σ orbit around the vertex.

Self-direct-loop: A dart is called a self-direct-loop iff: $(d) = -d$, which means that the dart and its α permutation must be consecutive in the σ orbit.

Bridge: The edge $\alpha^*(d)$ is called a bridge iff: $-d \in \varphi^*(d)$, which means that the edge $\alpha^*(d)$ connects to different components of the graph.



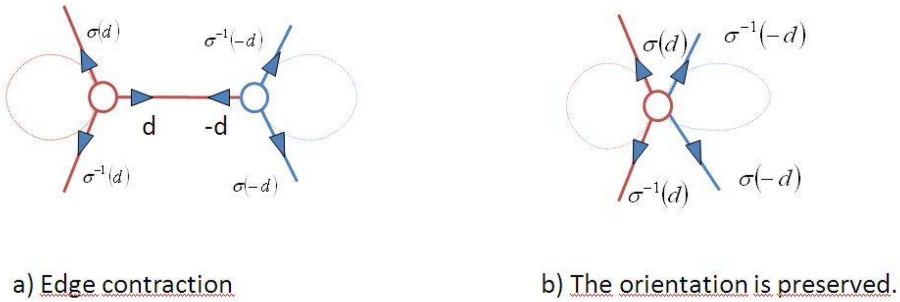
a) Self-loop

b) Self-direct-loop

c) Bridge

Fig. 12: Definitions within the combinatorial map framework

Contraction operations are controlled by contraction kernels (CK) such as the ones explain for the dual graph pyramid, and may induce the creation of some redundant edges (See Fig. 13: Edge contraction step in the combinatorial map). Using the dual graph framework, such redundant edges are characterized by their associated dual edges, which are incident to dual vertices with a degree lower than 3. Using the CM framework one of the darts of a redundant edge $\alpha^*(d)$ is either a self-direct-loop or belongs to an orbit of φ with cardinal equals to 2. This last condition characterizes dual vertices with a degree 2 [31].



a) Edge contraction b) The orientation is preserved.

Fig. 13: Edge contraction step in the combinatorial map

The removal of redundant edges is performed as in the dual graph reduction scheme by a removal kernel. This kernel is however decomposed in two sub-kernels: A removal kernel of empty self-loops which contains all darts incident to a degree 1 dual vertex and a removal kernel of empty double edges which contains all darts incident to a degree 2 dual vertex [29].

In order to preserve the number of connected components of the original CM, bridges must be excluded from removal operations. Furthermore, self-direct-loops are excluded from this general operation, and are treated as special cases [29].

The two dual points of view on merging regions are performed by two dual operations on the combinatorial map and its dual. Thus many particular cases of one operation may be retrieved thanks to the particular cases of the other. For example, since bridges are forbidden for removal operation the dual of a bridge, i.e. a self-loop, is forbidden for contraction [34].

As in the dual graph framework, an *inside* relationship between two regions is encoded by two edges: one edge encodes the common border between the two regions while the other encodes a self-loop incident to the vertex associated to the surrounding region. Considering the example already used for dual graph pyramids (See Fig. 6**:** In the ideal segmentation of these images, they are represented by the same *RAG*.), one can exchange the surrounded vertex without changing the order of the darts around the vertex $\sigma^*(1)$. Therefore, the two drawings shown in this example are encoded by the same combinatorial map. One cannot determine from the formally specified combinatorial map which part is inside and which is contained (See Fig. 14**:** The encoding of an ideal segmentation of the image (a) by a combinatorial map may be drawn using either (b) or (c)) [29].

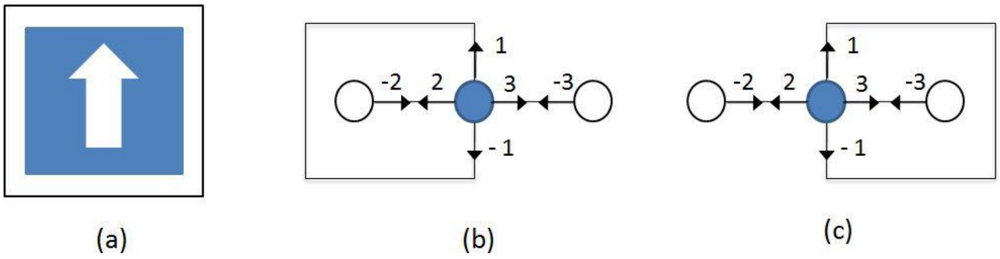


Fig. 14: The encoding of an ideal segmentation of the image (a) by a combinatorial map may be drawn using either (b) or (c).

This ambiguity in the location of the self-loop is related to the fact that the two darts of a self-loop play a symmetric role in the cycle σ to which they belong [29].

The determination of the *contains* and *inside* relationships requires thus to define a criterion in order to differentiate the two darts of a self-loop. This criterion is based on the orientation explicitly encoded by combinatorial maps. The starting dart of a self-loop is computed taking into account the orientations of all the darts in the σ orbit. It is said that d_j is the starting dart of the loop if the sequence of darts encoding the inside connected component is enclosed between d_j and $d_k = \alpha_i(d_j)$. Given a vertex incident to a self-loop, this last characterization allows to determine the regions *inside* the region encoded by this vertex [29].

4.1.4 Advantages of Combinatorial Pyramids

1 CMs explicitly encode the orientation of darts around one vertex. This information should be helpful to differentiate some configurations within the graph matching and clustering frameworks. This information is not encoded by RAGs nor explicitly available in dual graph data structure [28].
2 Given a CM, its dual is defined on the same set of darts by the permutations $\varphi = \sigma \circ \alpha$ and α. The efficiency of this transformation avoids an explicit encoding of the dual graph. Therefore, only one data structure has to be encoded and maintained along the pyramid [28].
3 CM may be defined in any dimensions [28].
4 Since a surviving dart is always linked to its vertex, which must survive by definition, redefinition and renaming of the surviving darts is no needed. Hence, the equivalent CKs can be expressed by simple subset relations [34].

4.1.5 Drawbacks of Combinatorial Pyramids

1 Certain structural entities are not represented explicitly, i.e. vertices and faces are implicitly defined and need extra processes to be identified, or to receive further attributes like in attributed relational graphs [34].
2 The *composed of* relationship is not encoded in this framework [29].

5 Bridging the Semantic Gap

In early stages of the development of CBIR, research was primarily focused on exploring various feature representations, hoping to find a “best” representation for each feature. In these systems, users first select some visual features of interest and then specify the weight for each representation. This burdens the user by requiring a comprehensive knowledge of low-level feature representations in the retrieval system. There are two more important reasons why these systems are limited: the difficulty of representing semantics by means of low-level features and the subjectivity of the human visual system.

One approach to solve this problem is by means of manual image annotation, but this is a tedious task and often it is difficult to make accurate annotations on images. There are many annotation tools available but human input is still needed to supervise the process. So, there should be a way to minimize the human input by making the annotation process semi or fully automatic. In the latter case, although there is much research on automatic image annotation, the results often do not really satisfy the retrieval requirements because of the flexibility and variety of user needs [35].

Algorithms providing global annotations, such as distinguishing between city and landscape images or between images acquired indoors and outdoors, have a higher success rate than algorithms attempting to detect specific objects, such as cars, cows and sunglasses. Automatic recognition of activities, events and abstract or emotive qualities in images currently performs rather poorly [36].

Many approaches have been proposed to reduce the gap between high-level image semantics and low-level image features.

Automatic Image Annotation (AIA)

Applied to image retrieval, the semantic annotation of images creates a conceptual understanding of the domains that the image represents, enabling software agents, i.e. search engines, to make more intelligent decisions about the relevance of the image to a particular user query [37].

Annotations are most frequently assigned at the global level and even when assigned locally the extraction of relational descriptors is often neglected. However, current annotation system might recognize and identify a beach and an ocean in an image but fail to represent the fact that they are next to each other. Therefore, to enrich the semantic description of the visual information, it is important to capture such relations [35].

Different machine learning methods for image annotation model the association between words and images or image regions. These include translation models, classification approaches and relevance models. While most models have used the co-occurrence of image regions and words, few have explored the dependence of annotation words on image regions [38].

There are two approaches to associating textual information with images described in the computer vision literature: *annotation* and *categorization*. In annotation, keywords or detailed text descriptions are associated with an image, whereas in categorization, each image is assigned to one of a number of predefined categories [36].

5.1.1 Classification Models

In the classification models, each annotated word is treated as an independent class and one semantic keyword corresponds to one classifier. The representative works are automatic linguistic index for pictures, content-based annotation method with Support Vector Machine (SVM) and Bayes Point Machine, estimating the visual feature distributions associated with each keyword. [39]

In [40] an approach of automatic linguistic indexing of pictures is presented, where categories of images, each corresponding to a concept, are profiled by statistical models, in particular, the two-dimensional multi-resolution hidden Markov model (2D MHMM). The pictorial information of each image is summarized by a collection of feature vectors extracted at multiple resolutions and spatially arranged on a pyramid grid. An element in an image is a block instead of a pixel. Features computed from one block at a particular resolution form a feature vector and are treated as multivariate data in the 2D MHMM. The 2D MHMM aims at describing statistical properties of the feature vectors and their spatial dependence. The 2D MHMM fitted to each image category plays the role of extracting representative information about the category. In particular, a 2D MHMM summarizes two types of information: clusters of feature vectors at multiple resolutions and the spatial relation between the clusters, both across and within resolutions. As a 2D MHMM is estimated separately for each category, a new category of images added to the database can be profiled without repeating computation involved with learning from the existing categories. Since each image category in the training set is manually annotated, a mapping between profiling 2D MHMMs and sets of words can be established. For a test image, feature vectors on the pyramid grid are computed. Consider the collection of the feature vectors as an instance of a spatial statistical model. The likelihood of this instance being generated by each profiling 2D MHMM is computed. To annotate the image, words are selected from those in the text description of the categories yielding highest likelihoods.

In [41], the SVM method approach is described as follows: the images are processed by taking for each pixel a fixed number of partially overlapping image subdivisions (tiles) that contain it, each of which is then independently classified by a multi-class Support Vector Machine (SVM) constructed according to “one per class” strategy. Each SVM is thus trained to discriminate between one class and the others. Before submitting a tile to a classifier, in the training or in the testing phase, a description of it is computed in terms of low-level features. They used joint histograms as feature vectors. After the satisfactory training of a classifier, they designed a strategy for annotating whole images. In order to label each pixel of the image as belonging to one of the classes, the tiles are sampled at fixed intervals.

Since several tiles overlap, every pixel of the image is found in a given number of tiles. Each tile is independently classified, and the pixel's final label is decided by majority vote.

An approach to *soft* annotation, using Bayes Point machines, is to give images a confidence level for each trained semantic label. It has been explored in [2]: Content-based Soft Annotation (CBSA). This vector of confidence labels can then be exploited to rank relevant images in case of a keyword search. In this method, each training image is manually labeled with one of the K pre-selected semantic labels. The labeling scheme is simple since it does not involve segmenting of images, or manual annotating of an image with multiple keywords and probabilities. Using the labeled instances, they train an ensemble of K binary classifiers (using Bayes Point Machines (BPMs)). Each classifier assumes the task of determining the confidence score for a semantic label. They automatically annotate images using the classifiers: each image is classified by the K classifiers and assigned a confidence score for the label that each classifier is attempting to predict. As a result, a K-nary label-vector consisting of K-class membership is generated for each image. CBSA performs annotation using global features.

5.1.2 Probabilistic Models

One well-known approach is to model image annotation as translating visual representation of concepts in an image to their textual representation.

In the translation model used in [38], images are segmented to images regions using the N-cut algorithm. Feature vectors capturing the image attributes of a region are clustered using the K-means clustering method to generate the visual vocabulary or the lexicon for image representation. Each image region is mapped to an element of this lexicon referred to as a blob. The Expectation-Maximization (EM) is used to estimate the translation probabilities. They propose a method to use a hierarchy defined on the annotation words derived from a text ontology to improve automatic image annotation and retrieval. Specifically, the hierarchy is used in the context of generating a visual vocabulary for representing images and as a framework for the hierarchical classification approach for automatic image annotation. In the hierarchy of annotation words that include other concepts related to the annotation words, images regions are grouped under each node in the hierarchy. An image region *r* in image *I* is placed under a concept *w* if either

- *w* is an annotation word for the image *I*, or,
- One of its hyponyms (descendants in the hierarchy) annotates the image, *I*.

While the translation models capture the correlation between blobs and words, the dependencies between blobs or words are not captured.

Graphical models of increasing sophistication have been proposed for capturing the dependence between words and image regions: Gaussian mixture models (GM-Mixture), Gaussian-Multinomial LDA (GM-LDA) and correspondence Latent Dirichlet Allocation (LDA) for the image annotation problem. These models capture the co-occurrence information by introducing latent variables to link image features with keywords. [38]

GM-mixture assumes a low-dimensional topology, leading to a fully-parametric model where 200 or so "latent aspects" are estimated using the EM algorithm [42]. Here, a single discrete latent variable *z* is used to represent a joint clustering of an image and its caption. An image/caption is assumed to be generated by first choosing a value of *z*, and then repeatedly sampling N region descriptions and M caption words conditional on the chosen value of *z*. Given a fixed number of factors K and a corpus of images/captions, the parameters of a GM-Mixture model can be estimated by the EM algorithm. This yields K Gaussian distributions over features and K multinomial distributions over words which together describe a clustering of the images/captions. Since each image and its caption are assumed to have been generated conditional on the same factor, the resulting multinomial and Gaussian parameters will correspond. An image with high probability under a certain factor will likely contain a caption with high probability in the same factor. Computing a region-specific distribution over words is beyond the scope of the GM-Mixture model. Conditional on the latent factor variable z, regions and words are generated independently, and the correspondence between specific regions and specific words is necessarily ignored. [43]

The latent Dirichlet allocation (LDA) model is a latent variable model that allows factors to be allocated repeatedly within a given document or image. Thus, different words in a document or different regions in an image can come from different underlying factors, and the document or image as a whole can be viewed as containing multiple topics [43].

The correspondence LDA (Corr-LDA) is a model that combines the flexibility of GM-LDA with the associability of GM-Mixture. With this model, they achieve simultaneous dimensionality reduction in the representation of region descriptions and words, while also modeling the conditional correspondence between their respective reduced representations. Corr-LDA can be viewed in terms of a generative process that first generates the region descriptions and subsequently generates the caption words [43].

The independence assumptions of the Corr-LDA model are a compromise between the extreme correspondence enforced by the GM-Mixture model, where the entire image and caption are conditional on the same factor, and the lack of correspondence in the GM-LDA model, where the image regions and caption words can conceivably be conditional on two disparate sets of factors. Under the Corr-LDA model, the regions of the image can be conditional on any ensemble of factors but the words of the caption must be conditional on factors which are present in the image [43].

While GM-LDA models better the joint distribution of words and regions, it fails to model the relationship between them. Corr-LDA finds much better predictive distributions of words than either GM-LDA or GM-Mixture. It provides as flexible a joint distribution as GM-LDA but guarantees that the latent factors in the conditional Gaussian (for image regions) correspond with the latent factors in the conditional multinomial (for caption words). Furthermore, by allowing caption words to be allocated to different factors, the Corr-LDA model achieves superior performance to the GM-Mixture which is constrained to associating the entire image/caption to a single factor [43].

The GM-Mixture predicts completely incorrect words if the average features do not easily correspond to a common theme. For example, the background of fish, reefs, water is not the usual blue and GM-Mixture predicts words like “fungus", “tree", and “flowers."

According to [43] the Corr-LDA model gives the best performance and correctly labels most of the example pictures.

Relevance models for image annotation have shown significant performance improvements. The model learns the joint probability of associating words to image features from training set and uses it to generate the probability of associating a word to a given query image [38].

One approach assumes that image annotation could be viewed as analogous to the cross-lingual retrieval problem and proposed a cross-media relevance model (CMRM) [42]**.**

Another approach proposes continuous-space relevance model (CRM) which assumed that every image is divided into regions and each region is described as a continuous-valued feature vector. Given a training set of images with annotations, a joint probabilistic model of image features and words is estimated. The feature vectors are based on automatic segmentation of the target image into regions and are modeled using a kernel-based probability density function. The annotation words are modeled with a multinomial distribution. Then the probability of generating a word given the image regions can be predicted. The CRM directly models continuous features, so it does not rely on clustering and consequently avoids the granularity issues. It does not make any assumptions about correspondence of annotation words to image regions. It is meant to reflect the prominence of words in a given annotation [42].

Another relevance model **(**MBRM**)** in which a Multiple Bernoulli model is used to generate words instead of the multinomial one as in CRM has been proposed. This model explicitly focuses on presence or absence of words in the annotation, rather than on their prominence, and it labels the entire picture and not specific image regions in a picture. MBRM adopts multiple Bernoulli distribution to replace the multinomial distribution in CRM. Actually, it is because that MBRM provides a solution based on the multi-label learning instead of a multi-class one as CRM. This method divides each image with a fixed-size rectangular grid. Given a new (un-annotated) image they split it into regions r_A, compute feature vectors $g_1, \dots, g_n$ for each region and then use the probability of a joint observation $\{r_A, w_B\}$ to determine what subset of vocabulary w* is most likely to co-occur with the set of feature vectors [42].

In image annotation, a multinomial would split the probability mass between multiple words. For example, if an image was annotated with "person, grass", with perfect annotation, the probability for each word would be equal to 0.5. On the other hand another image which has just one annotation "person" would have a probability of 1.0 with perfect annotation. If we want to find images of people, when rank ordering these images by probability the second image would be preferred to the first although there is no reason for preferring one image over another. The problem can be made much worse when the annotation lengths for different images differ substantially. A similar effect occurs when annotations are hierarchical. The Bernoulli model avoids this problem by making decisions about each annotation independent of the other words [42].

While the above models predicted the probability, $P(w|I)$ of an annotation word w given an image I, one is interested in generating a set of annotation words $\{w\}$.

While annotation of certain length can be selected by ranking the probabilities $P(w|I)$, the Coherent Language Model has been proposed. It predicts the annotation words $\{w\}$, by relaxing the estimation of $P(\{w\}|I)$ to estimate the probability, $P(\theta_w|I)$ of a language model θ_w to generate the annotation words for the image I. It takes into account word-to-word correlation. Instead of predicating each annotated word independently for a given image, this approach estimates a coherent language model for the image [38].

Except for the correspondence LDA model that captures the dependence between image regions and word annotations, and Coherent Language Model that captures the correlation between annotated words of an image, all the above models assume independence of word and image element events in their generative models for image annotation [38].

5.1.3 Ontology-Based Models

An ontology defines a set of representational terms called concepts. An ontology can be constructed in two ways: domain-dependent or generic. Generic ontologies are definitions of concepts in general; such as WordNet, which defines the meaning and interrelationships of English words. A domain-dependent ontology generally provides concepts in a specific domain, which focuses on the knowledge in the limited area, while generic ontologies provide concepts more comprehensively [44].

Ontology contains concepts (entities) and their relationships and rules. Adding a hierarchical structure to a collection of keywords produces a *taxonomy*, which is an ontology as it encodes the relationship "is a" (i.e. a dog is an animal). An ontology can solve the problem that some keywords are ambiguous. For example, a "leopard" could be a large cat, a tank, a gecko or a Mac operating system. Ontologies are important for the Semantic Web, and hence a number of languages exist for their formalization, such as OWL and RDF. [36]

From a computing science point of view, an ontology represents an area of knowledge that is used by people, databases, and applications that need to share domain information. Ontologies include computer-usable definitions of basic concepts in the domain and the relationships among them. [37]

The implementation of an ontology is generally taxonomy of concepts and corresponding relations. In an ontology, concepts are the fundamental units for specification, and provide a foundation for information description. In general, each concept has three basic components: terms, attributes and relations. Terms are the names used to refer to a specific concept, and can include a set of synonyms that specify the same concepts. Attributes are features of a concept that describe the concept in more detail. Finally relations are used to represent relationships among different concepts and to provide a general structure to the ontology [44].

The use of text ontologies as a basis for defining visual vocabulary or as a framework for automatic image annotation increases the number of concepts an image annotation system can recognize for a given image [38].

The Ontology Working Language (OWL) has become the de-facto standard for expressing ontologies. It adds extensive vocabulary to describe properties and classes and express relations between them (such as disjointness), cardinality (for example, "exactly one"), equality, richer typing of properties, and characteristics of properties (such as symmetry). OWL is designed for use by applications that need to process the content of information rather than just present information to humans [37].

All semantic models use two types of properties to build relationships between individuals (classes): Datatype properties and Object properties. When assigning properties to a class, all its sub-classes inherit their parent class properties.

As stated in [37], deciding on the appropriate type of property to use is not a trivial task. Every use of these properties has to be done thoughtfully, whereas object properties link individuals of different classes together, datatype properties can just point to immediate values (e.g. text strings), which are meaningless to a reasoning software, except for performing a string-based search. For instance, allocating datatype properties to the person class in order to give each new instance a first name, a last name is a correct use of datatype properties, because they cannot be reused by another individual. On the other hand, object properties are required to assign someone a nationality. Indeed, a country is more than a mere string. A country can have properties such as a currency, a capital city, many towns and villages, a language, a national flag and anthem, etc. A country needs thus to be an instance. Furthermore, such useful classes might be defined in already existing ontologies which enables their reusability [37].

Furthermore, in order to increase the automation of available data, inverse properties are used. Therefore, there is no privileged way to reason; all the properties are added in a dynamic manner (i.e. Object Properties such as hasPlayer can have inverse property isPlayerOf) [37].

At present, semantic annotation is implemented by some markup language such as XML based on a shared ontology definition. The markup language provides a mechanism for describing information in a structural way. The shared ontology definition provides a standard repository of concepts, which are used to describe all the entities that may be related to the image content [44].

While a number of ontologies and vocabularies are available, they tend to suffer from at least one of the following disadvantages listed in [36]:

- The vocabularies or ontologies developed for commercial purposes, such as those belonging to CORBIS and Getty Images, are proprietary competitive tools and are not available for public use.
- The vocabularies or ontologies developed for specific areas of application, such as the Iconclass ontology, while containing a wealth of terms, are concentrated on too narrow a domain to be useful for annotating general collections of images.

The main disadvantage of using ontologies is that the domain ontology is usually incomplete, because it will not include everything a user might want to say about an image.

5.1.4 Comparison between Some of the Annotation Methods

As these algorithms seem to be so different from each other, it is not easy to answer such questions as which models are better, what the connections among them are, and how they should be utilized.

In [45] an overall comparison has been made between several methods.

MBRM and CLP: They both provide the solutions within the context of the multi-label learning, while CLP employs a more sophisticated form. In the label propagation, CLP give more chance to rare word and relatively weaken the bias to common words, while MBRM suffer from the bias. Accordingly, CLP achieves wider coverage of correctly annotated words and recall than MBRM [45].

CMRM and CRM: Both methods adopt the probabilistic relevance model between images and words to perform basic image annotation. However, they differ in the representation of visual feature. CRM uses the continuous region features to calculate the image similarity, while CMRM uses blob histograms. Because the blobs are obtained by clustering the region features, much information has been lost. Therefore, CRM can better reflect the image relation than CMRM [45].

Better performance of MBRM and CLP than CRM implies that the formulation of image annotation as a multi-label learning problem is really preferable to as a multi-class learning problem [45].

Considering each word correlation individually, their different roles on the performance improvement can be observed. First, the statistical correlation by co-occurrence, i.e. SC, gains obvious improvement on the measure of NumWord, but it losses on the average precision. This indicates that the method is capable of connecting more words through the statistical information, but the connections cannot ensure the relatedness on the semantic level [45].

The combination of SC and CCS achieves the best performance. It shares the advantages from both correlations and gives a relatively precise and comprehensive representation of word semantic relatedness [45].

CMRM and CLM: Both models share blob-based image features, while CLM designs a new language model to represent the image-to-word relation and adopts the EM algorithm to update its image-to-image relation and the language model. Simply, CLM utilize a more sophisticated learning model to improve the image-to-image relation and the image-to-word relation. Accordingly, it boosts its performance [45].

There are three significant differences between MBRM and CMRM. First, CMRM is a discrete model and cannot take advantage of continuous features. In order to use CMRM for image annotation we have to quantize continuous feature vectors into a discrete vocabulary (similarly to the translation [5] models). MBRM, on the other hand, directly models continuous features. The second difference is that CMRM relies on *clustering* of the feature vectors into *blobs*. Annotation quality of the CMRM is very sensitive to clustering errors, and depends on being able to a-priori select the right cluster granularity: too many clusters will result in extreme sparseness of the space, while too few will lead us to confuse different objects in the images. MBRM does not rely on clustering and consequently does not suffer from the granularity issues. Finally, CMRM also models words using a multinomial process. MBRM performs significantly better than all previously proposed models on the tasks of image annotation and retrieval [42].

5.1.5 Current Approaches for Image Retrieval

The task of object retrieval is to classify objects found in images. This means to find objects in an image that are similar to sample objects in the pre-classified images. There are two problems with this task: the first is how we model objects. The second is how we measure similarity of objects.

When text annotation is nonexistent and incomplete content-based method must be used. Retrieval accuracy can be improved by content-based methods.

The semantic gap makes it difficult for the user to formulate queries against the image library. The integrated use of relevance feedback and object-based retrieval can overcome the problem of the semantic gap. Content-based Image Retrieval (CBIR) systems automatically extract image contents based on image features, i.e. color, texture, and shape. Relevance feedback methods are applied to CBIR to integrate users' perceptions and reduce the gap between high-level image semantics and low-level image features [9].

In CBIR systems, image processing techniques are used to extract visual features such as color, texture and shape from images. Therefore, images are represented as a vector of extracted visual features instead of just pure textual annotations. An object model is defined to represent images based on visual features. A user formulates a query by providing examples of images similar to the ones he wishes to retrieve. The system uses a query model to convert the image into an internal representation of query, based on features extracted from input images. A retrieval model performs image retrieval by computing similarities between images in object and the query representations, and the results are ranked based on the computed similarity values. Overall similarity (distance) between an object and the image query is computed as a weighted summation of similarities (distances) over the feature set. The object, query, and retrieval models together define a CBIR model [9].

The key idea is that we should incorporate human perception subjectivity into the retrieval process and provide users opportunities to evaluate retrieval results and automatically refine queries on the basis of those evaluations.

There are two different modes of user interactions involved in image retrieval systems. In one case, the user types in a list of keywords representing the semantic contents of the desired images. In the other case, the user provides a set of examples images as the input and the retrieval system will try to retrieve other similar images. In most image retrieval systems, these two modes of interaction are mutually exclusive. In [46], the authors argue that combining these two approaches and allow them to benefit from each other yields a great deal of advantage in terms of both retrieval accuracy and ease of use of the system.

Region-Based Retrieval

Region based approaches are based on the fact that semantic meaning of an image is often contained in object level in the image. User is often interested in objects in images, not the whole. For example, user might be interested in an image of a tiger on grass only because of the tiger, but not care about the grass. Representing images on region level gives the possibility of inferring in object level [47].

Region-based image retrieval methods attempt to overcome the drawback of global features by representing images at object-level, which is intended to be close to the perception of human visual system. The NeTra and Blobworld are two earlier region-based image retrieval systems. During retrieval, a user is provided with segmented regions of the query image, and is required to assign several properties, such as the regions to be matched, the features of the regions, and even the weights of different features [48].

A region-based retrieval system applies image segmentation to decompose an image into regions, which correspond to objects if the decomposition is ideal and retrieves images based on the similarity between regions [9].

In [48], the authors proposed a region-based image retrieval method using relevance feedback. They segment the images into regions and describe those regions with the color moment. Enlightened by the idea of relevance feedback, they designed a scheme that uses users' feedback information, i.e., positive and negative examples, to estimate the region importance of all positive images. Their basic assumption is that important regions should appear more times in the positive images than unimportant regions. They defined the similarity measure between two images, each one represented by regions. To compute the similarity of two images, they first match all regions in the two images. Then they assign the matching between two regions, with a significance credit, that indicates the importance of the matching for determining similarity between images.

In [47], they propose a retrieval method using relevance feedback information both on image level and on region level. They propose an optimization based semi-supervised learning approach to fuse information of the two levels together. The algorithm first predicts region score based on the labeled information and prior, then calculates image score, based on which a rank of images is given.

Score of unlabeled regions are predicted according to labeled regions on a graph $G = (V,E)$ with regions as nodes $x_i \in V$ and similarity between regions i, j as weight w_{ij} of edge $e_{ij} \in E$. Weights are collected in an affinity matrix W representing similarities between any pair of nodes. A region is represented by different sets of features (color histogram, texture coarseness). Similarity between two regions can be calculated with a distance measure which can be χ^2, Euclidian, or EMD distance depending on the type of the feature. They use three priors for assigning the labels: The first term is the smoothness prior requiring that labels for similar regions should be similar. The second term expresses the requirement that predicted labels for the labeled regions should be as close to the given label as possible and the third states that for positive feedback on an image, they do not know the exact label for all its regions, but they know there is at least one region in the image with high score [47].

Relevance Feedback

Relevance feedback is another approach to reduce the gap between high-level image concepts and low-level image features by involving the user's perception of images in the retrieval process. This approach gradually refines the original image query based on the feedbacks the user provides on the images retrieved at each iteration [9].

Since all image low-level features cannot capture high-level semantic concepts, most retrieval methods have tried to find an optimum set of feature weights to model the user's perception based on image features (feature weighting). Some CBIR systems ask the user to set the feature weights; however, there are several shortcomings to such approaches. Users may find it difficult to express their query appropriately in terms of the provided features since they do not initially have a clear idea of the information needed. Furthermore, there may be a mismatch between the users' perception of the visual properties and the feature representations that are actually used for retrieval [9].

The most crucial issues in relevance feedback of CBIR systems are the following [46]:

1 How to learn effectively from small sets of feedback samples;
2 How to accumulate knowledge learned from feedback;
3 How to integrate low-level visual and high-level semantic features in query and feedbacks. An effective relevance feedback system should provide effective solutions to address these three issues, in addition many others.

Relevance feedback approaches have been successfully applied in the information retrieval area. In such approaches, the user needs to provide the retrieval system with positive examples, negative examples or both. In a CBIR system, positive examples are images that are similar to the images the user is looking for, and negative examples are those that are not similar to user's query. In each retrieval iteration, the system uses relevance feedback data to modify feature weights in order to create a more accurate query model. Studies shows using only positive example lead to more improvement than only negative examples. However, best improvement in retrieval is obtained by using positive and negative examples together [9].

Recent image retrieval approaches have been proposed based on long-term learning from previous feedbacks as well as short-term learning from feedbacks in the current query session. Different approaches have been proposed based on Collaborative Filtering (a technique used in recommendation systems), Support Vector Machines (a learning and classification method), machine learning methods, and probabilistic methods [9].

5.1.6 Query Refinement framework

The Query Refinement framework can be split in two branches: Query Modification and Query Weighting [46]. Query Modification allows users to refine the query representation. A user may start from a query object that approximately captures his information need. In each iteration of feedback, the system modifies the representation of the query to a more suitable representation. Query weighting changes the relative weights of different features in the query representation. The re-weighting mechanism allows the system to learn the user's interpretation of similarity/distance function [9]. The basic idea behind the re-weighting method is to enhance the importance of the dimensions of a feature that help in retrieving the relevant images and reduce the importance of those dimensions that hinder this process [46].

Query Modification can be achieved using either of two approaches: query expansion and query point movement. In the query point movement approach, a query is represented by a single point in a feature space and refinement process attempts to move that point toward the direction where relevant points were located. On the other hand, query expansion does not assume that a query is represented as a point in a multidimensional space. Instead, it modifies the query by selectively adding new relevant objects to the query representation [9].

In [9] is stated that experimental evaluation have shown that query expansion outperforms query point movement in retrieval effectiveness. Another advantage of query expansion over query point movement is that query expansion can be coupled with existing information systems without requiring any modification being made to them [9].

Short-Term and Long-Term Learning

In relevance feedback-based approaches, a CBIR system learns from feedbacks provided by the user. In short-term learning, only the feedbacks for the current search session are used in the learning algorithm, and image features are the primary source of data. The main challenge in this approach is to find the best combination of image features that presents the user's query. Such optimum set of features can include features that capture similarities between positive images, or features that discriminate positive examples from negative ones. Therefore, feature weighting, discriminant analysis, SVM, and instant learning methods are widely used in short-term learning [9].

Long-term learning approaches utilize the feedbacks collected during prior search transactions. Accumulated feedbacks are stored in a search history matrix (See Table 1: An example of a search history). A search history matrix, denoted by $H_{N.M}$, stores the labels provided by the user for image x_i, i=1,…,M in transaction t_k, k=1,..,N. A transaction is the set of feedbacks collected form a user during relevance feedback iterations of a search session. It is assumed that the user does not change the query image he has in his mind during the relevance feedback iterations. Therefore, each transaction corresponds to a semantic and can be represented by labeled images in an L-dimensional space where L is the number of images labeled in the transaction.

t	X1	X2	X3	X4	X5	X6
Transaction 1	+	+	-	-	-	-
Transaction 2	+	+	-	-	-	-
Transaction 3	-	-	+	+	-	-
Transaction 4	-	-	-	-	+	+

Table 1: An example of a search history

The first step in a long-term learning approach is detecting the number of semantic classes, which is the number of concepts presented in a search history matrix, and creating the semantic space by defining each semantic class. Then, each image should be assigned to its corresponding semantic class. The size of search history matrix is large, statistical models and approaches such as principal component analysis and latent semantic analysis are popular in long-term learning approaches [9].

Feature Weighting

In CBIR systems, the distance between two images is computed as a weighted summary of their feature distances:

$$D(i_1, i_2) = \sum_{j=1}^{m} w_j d(f_{i_1,j}, f_{i_2,j}) \ . \qquad 5.1$$

Where, w_j is the weight for feature j and d is a distance function. Popular distance functions are Manhattan, Euclidean, and Cosine distances [9].

Relevance feedback data are used to modify the query representation in order to capture user perception by updating feature weights. Updated feature weights modify the pair-wise image distances; therefore, level of similarity between the query and images in the database are changed for the next retrieval iteration.

Discriminant Analysis

The objective of discriminant analysis is to find the most discriminant features of data (x_i) in the original high-dimensional space, and map data points to a projected low-dimensional space in a way that discriminant features are preserved. Linear discriminant analysis is a popular method in CBIR area. Linear discriminant analysis tries to find the transformation matrix W that maximizes the separation between different classes while minimizing within-class scatters in the new subspace.

When there are only two classes, the process is known as Fisher Discriminant Analysis (FDA). A significant problem with FDA is its assumption that negative examples are drawn from the same distribution, which is not usually true in the case of image data. Another choice is Multiple Discriminant Analysis (MDA) that considers each negative example as a different class and creates a (N_N+1)-class discriminant analysis problem where N_N is the number of negative examples. Again, this assumption may not be true and some of negative examples do belong to the same distribution. Biased Discriminant Analysis (BDA) keeps negative examples away from positive examples, and clusters only positive examples. This means that all positive examples should be located closely in the same area in the feature space. However, semantically similar images may not be close to each other in the feature space, especially when their relations are defined based on high levels of semantic concepts [9].

In [9] is stated that empirical experiments with synthesized data have shown that when the number of positive examples (N_P) is much higher than the number of negative examples (N_N), compacting negative examples, and discriminating negative examples from positive examples is the most efficient strategy. On the other hand, when $N_N >> N_P$, it would be better to compact positive examples and keep them away from the mean of negative points. The reason is when the number of positive examples is much higher, it would be a heavy burden to compact them or discriminate them from negative examples. It would be the same for negative examples when their number is much higher than positive points.

Support Vector Machines (SVM)

Support vector machines are a core machine learning technology. In the area of image retrieval, SVMs have been used for feature weighting.

In image retrieval by relevance feedback, SVMs can be applied to the image features space. Data points are images which are labeled as positive (+1) or negative (-1) [46]. The task of SVMs is to create a hyper-plain to separate all images in the database to two group of relevant (+1) and irrelevant (-1) images. During the relevance feedback process, an SVM is constructed in each dimension of the feature space and the generalization error is computed and features with smaller generalization error are assigned larger weights. Generalization error measures how good a classifier can classify training data.

In another SVM method, weights are assigned to each types of feature rather than each dimension of the features so that only a few weights need to be estimated which may have less risk in relevance feedback problems with high dimensionality on the features and small size of training samples.

In relevance feedback problems, positive examples can be assumed to belong to one class. However, negative examples are different from the query in many different ways and may not belong to one class [9].

Probabilistic Models

Probabilistic models have also been applied in image retrieval by relevance feedback to find the probability that the user selects each image. Bayesian models are widely used to solve such probabilistic models. In a learning approach, a Discrimination version of Expectation-Maximization (D-EM) algorithm is proposed to use data from relevance feedback to cluster images. In image retrieval by relevance feedback, users label only a small ratio of images in the database and EM algorithm has been proven to be suitable for problems with small size of labeled data. The algorithm iterates in two steps until no specific improvement is achieved. In the first step, cluster centers are estimated based on the labeled data, and in the second step, unlabeled images are labeled using the cluster centers computed in the first step [9].

The relevance feedback problem has been also studied as an optimization problem. It is a query point movement approach to construct a point as the new query for next relevance feedback iteration such that it minimizes the distances of currently labeled images from the current query [9].

Clustering

The task of a clustering algorithm is to partition a data set into subgroups such that those in each particular group are more similar to each other (inter-similarities) than to those of other groups (intra-dissimilarities). A clustering algorithm can be agglomerative or divisive. An agglomerative approach begins with each data point as a cluster, and successively merges clusters together until a stopping criterion is satisfied. A divisive method begins with all data points in a single cluster and performs splitting until a stopping criterion is met. Stopping criteria can be defined by validation rules. A validation rule measures some characteristics of created clusters such as compactness and separateness. Compactness measures how close are the data point to each other in a cluster, and separateness measure how far clusters are located from each other [9].

5.1.7 Hierarchical clustering

A hierarchical algorithm yields a nested grouping of data points, and similarity levels at which groupings change. Most hierarchical clustering algorithms are variants of the single-link, complete-link, and minimum-variance algorithms. These algorithms differ in the way they characterize the similarity between a pair of clusters. In the single-link method, the distance between two clusters is the minimum of the distances between all pairs of data points drawn from the two clusters (one pattern from the first cluster, the other from the second).

In the complete-link algorithm, the distance between two clusters is the maximum of all pair-wise distances between patterns in the two clusters. In either case, two clusters are merged to form a larger cluster based on minimum distance criteria [9].

5.1.8 Partitional clustering

A partitional clustering algorithm obtains a single partition of the data instead of a clustering structure. Partitional methods have advantages in applications involving large data sets for which the construction of a hierarchical structure is computationally prohibitive. A problem accompanying the use of a partitional algorithm is the choice of the number of desired output clusters. Thus, partitional techniques usually produce clusters by optimizing a criterion function. In practice, the algorithm is typically run multiple times with different starting states, and the best configuration obtained from all of the runs is used as the output clustering.

The most intuitive and frequently used criterion function in partitional clustering techniques is the squared error criterion, which tends to work well with isolated and compact clusters. The k-means is the simplest and most commonly used algorithm employing a squared error criterion. It starts with a random initial partition and keeps reassigning data points to clusters based on the similarity between the data point and the cluster centers until a convergence criterion is met. The k-means algorithm is popular because it is easy to implement, and its time complexity is O(n), where n is the number of data points. A major problem with this algorithm is that it is sensitive to the selection of the initial partition and may converge to a local minimum of the criterion function value if the initial partition is not properly chosen [9].

5.1.9 Nearest Neighbor Clustering

Nearest neighbor distances can be used in clustering procedures. In an iterative procedure, data points are assigned to the cluster of its nearest labeled neighbor data point, if the distance to that labeled neighbor is below a threshold. The process continues until all data points are assigned [9].

Late-Fusion

Late fusion of independent retrieval methods is one of the simplest and most widely used approaches to combine visual and textual information for Multimedia Image Retrieval (MIR). The approach consists of building several retrieval systems (i. e. independent retrieval models, hereafter IRMs) based on different information from the same collection of documents. At querying time, each IRM returns a list of documents relevant to a given query. The output of the different IRMs is combined to obtain a single list of ranked documents. In [49] it was considered the combination of multiple heterogeneous IRMs through the late fusion approach (i.e. LFHM). Heterogeneousness in IRMs has proved being important to improve the fusion results by providing complementary and diverse, yet redundant, lists of documents to the fusion; the inclusion of many IRMs contributed in the same directions as well, although mostly in redundancy. The lists of ranked documents are combined by assigning a score W to each document d_j as follows:

$$W(d_j) = \left(\sum_{i=1}^{N} 1d_j \in L_i\right) \times \sum_{i=1}^{N} \left(\alpha_i \times \frac{1}{\psi(\mathrm{d_j}, \mathrm{L_i})}\right) \qquad 5.2$$

Where i indexes the N available lists of documents $L_{\{1;\ldots;N\}}$; $\psi(x;H)$ is the position of document x in ranked list H; 1_a is an indicator function that takes the unit value when a is true and $\alpha\left(\sum_{k=1}^{N} \alpha_k = 1\right)$ is the relevance weighting for IRM i, when using hierarchical LFHM. Each list L_i is the output of one of the IRMs we considered, these are shown in Table 1. Documents are re-ranked in descending order of this score, and the top-x documents are kept [49].

Combining Semantic Annotations and Visual Features for Retrieval

5.1.10 Relevance Feedback and Query Expansion

In [46], the authors proposed a CBIR framework with integrated relevance feedback and query expansion, in which the semantic-based index and relevance feedback are seamlessly integrated with those based on low-level feature vectors. The proposed relevance feedback framework consists of a semantic network which links images to semantic annotations in a database, a similarity measure that integrates both semantic features and image features, and a machine learning algorithm to iteratively update the semantic network and to improve the system's performance over time.

The framework supports both query by keyword and query by image example through semantic network and low-level feature indexing. Cross-modality query expansion is supported. That is, the retrieved images based on keyword search are considered as the positive examples, based on which the query is expanded by features of these images. In this way, the system extends a keyword-based query into feature-based queries to expand the search range.

For query by image example, similar procedure takes effect to extend the retrieval from feature space to semantic space. In this way, user input information is utilized as much as possible to improve the retrieval performance. More importantly, there is a progressive learning process to propagate the keyword annotations from the labeled images to un-labeled ones during the feedback. In this way, more and more images are implicitly labeled by keywords by the semantic propagation process. In addition to supporting keyword search, this annotation propagation process also the retrieval system accumulated users' feedback information so that it will be improve performance of future retrieval requests [46].

5.1.11 Annotation-Based Expansion

In [49] the authors proposed an approach intended to represent documents by considering both their high level (manual annotations) and low-level semantics (labels automatically assigned). It consists of expanding manual annotations with labels generated by automatic annotation methods.

They segment the images using the normalized nuts algorithm and several features are extracted from each region. Using a subset of annotated regions and a classifier, all the regions in the segmented collection are automatically labeled. For each image, the generated labels are included as an expansion of the original annotations. The expanded annotation is considered a textual document, and then, a text-based retrieval model is used for indexing the documents. They selected as retrieval engine a vector space model (VSM) with a combination of augmented-normalized term-frequency and entropy for indexing/weighting documents [49].

A knn classifier was used for generating the annotations and they improved the results of the annotation methods by using a Markov Random Field that uses spatial relationships between connected regions for maximizing the annotation coherence for each image [49].

Ontology-Based Methods

Ontology can improve user understanding and bridge the semantic gap. Ontology can be described as the objects and their relationships to other objects. Hence general information requirements can be satisfied by the ontology even without exact matches to provide keywords.

Image retrieval queries can be formalized based on the shared ontology. It is possible to design the query interface to use only concepts defined in the ontology. Together with the concepts, users should also be allowed to propose some constraints on the attributes for corresponding concepts. The system can then construct an XML file to specify the retrieval query. Using the formalizations for both the images and a retrieval query, we can compare the semantic similarity between each image and the user's query [44].

The Image Retrieval algorithm proposed in [37] uses a tree comparison that traverses the ontology classes in order to find a path that represents the "nearest neighbor" to the query. In order to allow a dialog between the query and the annotated files, a semantic description generator needs to be created [37]. The semantic description generator transforms his query into an OWL query, which determines a first set of images on which the Image Retrieval Algorithm can be applied. This algorithm uses a weightings system previously set by the user in order to fine-tune the final results in accordance to the user queries. The last stage consists in displaying the results [37]. The nearest neighbor matchmaking algorithm continues traversing back to the upper class of the ontology and matching instances until there are no super classes in the class hierarchy, i.e. the leaf node for the tree is reached, giving degree of match equal to 0 [37].

In [44], the authors propose a semantic similarity between images and retrieval queries by calculating their similarity probability. For each combined concept entity in a retrieval query, a satisfaction probability is introduced. This probability specifies what proportion of each combined concept entities in the retrieval query is satisfied in each image's set of combined concept entities. In addition to the satisfaction probability, each combined concept entity in a retrieval query also has a weight related to it, which allows the user to express the retrieval priority for a concept. They represent the similarity between an image and a retrieval query using the satisfaction probability and retrieval weight.

6 Current Systems and Application Areas

Current Systems

The most widely known image retrieval system is IBM's **QBIC** (Query by Image Content) system. In QBIC, the user is allowed to specify certain characteristics of the image they want to find. The results are returned in descending order score of textual relevance to the query. Recent versions of QBIC contain simple automated region segmentation functionality. In other previous systems, color histograms have been used and proved to be helpful, although the use of such global features as a point of query has provided little information about how that color is distributed spatially about the image [9].

SIMPLICity is another system that incorporates the properties of all the segmented regions so that information about an image can be fully used. To segment an image, the system partitions the image into blocks and extracts a feature vector for each block. The k-means algorithm is used to cluster the feature vectors into several classes with every class corresponding to one region in the segmented image. Six features are used for segmentation. Three of them are color components (LUV color space), and the other three represent energy in high frequency bands of the wavelet transform. A significance credit is assigned to the regions to be used in distance function. The significant factor can be uniform (all regions are equally important), based on the area percentage, or location of the region [9].

SIMPLICity uses Integrated Region Matching (IRM) to account for all regions in two images by calculating the distance between them as weighted average of the distance between any two region pairs from the two images. By considering distance between all regions, this method tends to under-estimate the distance. For example, two images with large region of grass will have small distance no matter it is a horse or a tiger on the grass. Also, user relevance feedback is only on image level, and the system cannot get the information about the real interested region in the image [47].

Blobword is another CBIR system that is based on segmenting the image into regions and querying the image database using features of those regions instead of basing the query on global properties. Blobworld recognizes images as collections of objects that are in a spatial relationship to one another. Using the Expectation-Maximization algorithm to estimate the parameters of this model, the resulting pixel-cluster memberships provide a segmentation of the image. Once the image is segmented, features of the different segments are produced, such as color and texture. While querying, the user is allowed to access the segments directly to determine which features of the image are important to his/her query. When results are returned, the user also sees the Blobworld representation of the image, which is used to refine the user's query [9].

In the NeTra system retrieval is based on segmented image regions. The segmentation scheme requires user supervision for parameter tuning and segmentation corrections. Furthermore, a one-to-one region matching is proposed after region selection by the user.

However, among many region based systems, including Netra and Blobworld, user can only specify an interested region on a mask which indicates pre-computed segmentation. The interface is not easy to use, since the user needs to look at both the mask and the image. It also depends highly on segmentation accuracy. If one object is segmented into many regions, it is difficult to correct this error. Limited user knowledge about the image is incorporated into the system through selecting single region [47].

In VisualSeek, each image is decomposed into regions of equally dominant colors. For each region, feature properties and spatial properties are retained for the subsequent queries. A query consists of finding the images that contain the most similar arrangements of similar regions. The color region extraction uses the back-projection technique. To start a query, the user sketches a number of regions, positions them on a grid, and selects a color for each region. To find the matches of a query image with a single region, queries on color set, region absolute location, area and spatial extent are first done independently. The results of these queries are intersected and from the obtained candidate set, the best matching images are taken by minimizing a total distance given by the weighted sum of the four distances mentioned. If the query image consists of a number of regions, in absolute or relative location, then for each region positioned in absolute location, a query like that described above is made, and for regions positioned by relative location individual queries on all attributes except location are performed [9].

For the intersection of all this query results, the relative spatial relations specified by the user are evaluated using 2D string representation

MARS is the pioneer of CBIR systems in implementing relevance feedback techniques. Queries in MARS can be a combination of low-level features (color, texture, shape) and textual descriptions. There is a tree associated with each query. In a query tree, the leaves represent the feature vectors (the terms of the boolean expression defining the query) while the internal nodes correspond to boolean operators or more complex terms indicating a query by object. The tree is evaluated bottom-up, each internal node receives from each child a list of ranked images and combines these lists according to the weights on the parent-child links. In MARS, color is represented by a 2D histogram over the HS coordinates of the HSV space and the similarity distance between two color histograms is computed by histogram intersection. Texture is represented by two histograms, one measuring the coarseness and the other one the directionality of the image, and one scalar defining the contrast. In order to extract the color/texture layout, the image is divided into 5 × 5 sub images and for each sub image, features are extracted. The object in an image is segmented out in two phases. First, a k-means clustering method in the color-texture space is applied, then the detected regions are grouped by an attraction based method. A number of attractor regions are defined and each region is associated with the attractor that has the largest attraction to it. The attraction between two regions, *i* and *j*, is defined as $F_{ij}=M_i M_j / d_{ij\ 2}$, where M_i, M_j are the sizes of the two regions and d_{ij} is the Euclidean distance between the two regions in the spatial-color-texture space. The Euclidean distance between the vector representations is used to compute the texture similarity between two sub-images. A weighted sum of the 5 × 5 color/texture similarities is used to compute the color/texture layout distance between two images. The shape of the boundary of the extracted object is represented by means of Fourier Descriptors. The similarity between two textures of the whole image is determined by a weighted sum of the Euclidean distance between contrasts and the histogram intersection distances of the other two components. The user can also choose a set of desired features from a list when querying the system [9].

Photobook is another CBIR system which implements three different approaches to constructing image representations for querying purposes, each for a specific type of image content: faces, 2D shapes and texture images. The first two representations are similar in the way that they offer a description relative to an average of a few prototypes by using the eigenvectors of a covariance matrix as an orthogonal coordinate system of the image space. First a preprocessing step is done in order to normalize the input image for position, scale and orientation. In a texture description, an image is viewed as a homogeneous 2D discrete random field, which is expressed as the sum of three orthogonal components. These components correspond to periodicity, directionality and randomness. In creating a shape description, first a silhouette is extracted and a number of feature points on this are chosen (such as corners and high-curvature points). This feature points are then used as nodes in building a finite element model of the shape. To perform a query, the user selects some images from the grid of still images displayed and/or enters an annotation filter. From the images displayed, the user can select another query images and reiterate the search [50].

Application Areas

CBIR counts with many application areas, among them are the following:

- In Search Engine: This can be implemented in the search engine through which we can find similar image from the whole world.
- Crime Prevention: Automatic face recognition systems, used by police forces.
- Security Check: Finger print or retina scanning for access privileges
- Medical Diagnosis: Using CBIR in a medical database of medical images to aid diagnosis by identifying similar past cases.
- Intellectual Property: Trademark image registration, where a new candidate mark is compared with existing marks to ensure no risk of confusing property ownership.

7 Open Problems

A significant problem in Content-based Image Retrieval (CBIR) systems is the gap between high-level semantics in human minds and low-level features computable by machines. The semantic gap continues to be one of the major issues, along with scalability, benchmarking and search speed.

In current CBIR systems a large improvement has been made to represent visual content of images. However, the relation between image content and image information representation is currently not understood satisfactorily. Representations automatically generated from an image do not possess enough expressive power to accommodate user input when he searches for images with a specific content. Hence, using low-level features alone may not be effective in representing users' feedbacks and in describing their intentions.

From the representational point of view, it seems that by using spatial information, CBIR may automatically improve its results. But questions such as: "How to do it?" or "Which of all the possible relations can be useful?" are still unanswered. Also, for representing spatial relationships between regions, it is assumed that the regions have been segmented out of the images reliably. But in most cases this is not a reasonable factor, since reliable segmentation of regions poses a problem itself.

Regarding Automatic Image Annotation, the main problem that arises is that the developed vocabularies cannot be complete, that is because it will not include everything a user might want to say about an image.

In most of the CBIR systems, relevance feedback is provided in the form of positive and negative examples. However, when images or query concepts are semantically rich, it is not convenient for the users to transfer the degrees of relevancy they have in their minds through binary feedbacks to the system; therefore, the quality of the system input is reduced and learning performance plunges [9]. Also, the ranking methods to show the searched results are a problem to consider.

8 Available Databases

Some of the benchmark databases used for performing experiments related to Content-Based Image Retrieval are listed below:

- PASCAL VOC databases: These databases are freely available on the PASCAL web-page[1] and consists of several databases, five of them are fully annotated, three are partially annotated and there is one unannotated.
- IAPR TC-12 dataset: The IAPR TC-12 Benchmark database consists of 20,000 still images taken from locations around the world and comprising an assorted cross-section of still images which might for example be found in a personal photo collection. It includes pictures of different sports and actions, photographs of people, animals, cities, landscapes and many other aspects of contemporary life. This data is also strongly annotated using textual descriptions of the images and various meta-data. [36]
- Corel database: The Corel data set consists of 5000 images from 50 Corel Stock Photo cds. Each cd includes 100 images on the same topic, and each image is also associated with 1-5 keywords. Overall there are 371 keywords in the dataset [42].
- Caltech-101 database: This database contains from 31 to 800 images per category. Most images are medium resolution, i.e., about 300 × 300 pixels. Caltech-101 is probably the most diverse object database available today, though it is not without shortcomings. Namely, most images feature relatively little clutter, and the objects are centered and occupy most of the image. In addition, a number of categories, such as minaret, are affected by "corner" artifacts resulting from artificial

[1] http://www.pascal-network.org/challenges/VOC/databases.html

image rotation. Though these artifacts are semantically irrelevant, they can provide stable cues resulting in misleadingly high recognition rates [25].

- Graz dataset: This dataset is characterized by high intra-class variation. This dataset has two object classes, bikes (373 images) and persons (460 images), and a background class (270 images). The image resolution is 640 × 480, and the range of scales and poses at which exemplars are presented is very diverse, e.g., a "person" image may show a pedestrian in the distance, a side view of a complete body, or just a closeup of a head [25].
- MPEG-7 CE-Shape-1 database: It is widely used for testing the performance of shape description and matching algorithms. It consists of 1400 silhouette images which are grouped into 70 classes with 20 objects per class [51].
- KIMIA99 database: Consists of 99 shapes which can be grouped into 9 classes, each consisting of 11 objects [51].
- Empics database: Empics is a Nottingham-based company which is part of the Press Association Photo Group Company. As well as owning a huge image database in excess of 4 million annotated images which date back to the early 1900's, the company processes a colossal amount of images each day from varying events ranging from sport to politics and entertainment. The company also receives annotated images from a number of partners that rely on a different photo indexing schema.

Two datasets have been released by Microsoft Research in Cambridge. The "Database of thousands of weakly labeled, high-res images" contains images divided into 23 categories. Some of these are divided into sub-classes, such as different views of cars. The "Pixel-wise labeled image database" contains 591 images in which regions are manually labeled using the 23 labels. Combining the keyword lists results in 33 unique keywords [36].

9 Scientific Conferences

ICPR [2] is the conference of the International Association for Pattern Recognition (IAPR). ICPR is an international forum for discussions on recent advances in the fields of Computer Vision; Pattern Recognition and Machine Learning; Signal, Speech, Image and Video Processing; Biometrics and Human Computer Interaction; Multimedia and Document Analysis, Processing and Retrieval; Bioinformatics and Biomedical Applications.

CIARP[3] is the Iberoamerican Congress on Pattern Recognition supported by IAPR and sponsored by another five PR iberoamerican societies. This is a fruitful forum for the exchange of scientific results and experiences, as well as the sharing of new knowledge, and the increase of the co-operation between research groups in pattern recognition and related areas.

CIVR[4], the ACM (Association for Computing Machinery) International Conference on Content-based Image and Video Retrieval (CIVR) is a prominent technical conference in this field. After 5 editions, CIVR has now become an official ACM Conference and an IAPR co-sponsored event since 2007 in Amsterdam (thanks to the initiatives by 2007 co-chairs Marcel Worring and Nicu Sebe).

ImageCLEF[5] is the cross-language image retrieval track which is run every year as part of the Cross Language Evaluation Forum (CLEF). ImageCLEF has already seen participation from both academic and commercial research groups worldwide from communities including: Cross-Language Information Retrieval (CLIR), Content-Based Image Retrieval (CBIR) and user interaction. ImageCLEF was originally proposed by Mark Sanderson and Paul Clough from the Department of Information Studies,

[2] http://www.icpr2010.org/
[3] http://www.gdl.cinvestav.mx/ciarp2009/index.php
[4] http://www.iapr.org/members/newsletter/Newsletter09-02/index_files/Page361.htm
[5] http://www.imageclef.org/

University of Sheffield. However, today ImageCLEF is organized and run voluntarily by a much larger number of individuals and research groups.

VISIGRAPP[6] is the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications. The purpose of this conference is to bring together researchers and practitioners on the areas of computer vision, imaging, computer graphics and Information Visualization, interested in both theoretical advances and applications in these fields. The VISIGRAPP component conferences are specialized in the following topics: GRAPP is structured along four main tracks, covering different aspects related to Computer Graphics, from Modeling to Rendering, including Animation and Interactive Environments, IMAGAPP covers theory, applications and technologies related to image display, color coding, medical imaging, remote sensing, business document processing, digital fabrication, printing and electronic devices, VISAPP has also four main tracks, namely: Image Formation and Processing, Image Analysis, Image Understanding and Motion, Tracking and Stereo Vision and IVAPP structured along several topics related to Information Visualization.

ICIAP[7] is the International Conference on Image Analysis and Processing, organized every two years by the Italian group of researchers affiliated to IAPR (GIRPR), with the aim to bring together researchers in image processing and pattern recognition from around the world.

10 Institutions and Investigation Groups in the World

10.1.1 Pattern Recognition and Image Processing (PRIP) Group

The PRIP[8] is a group from the Institute of Computed Aided Automation, Vienna University of Technology, Austria. The main research areas of this group are Image Pyramid, 3D Vision, Video Analysis, Structure & Topology. The Head of this group is Walter G. Kropatsch. His major areas of interest are theoretical and practical research in digital image processing, pattern recognition, remote sensing, art history, archeology, industry, analysis of medical imagery, digital elevation models, expert vision systems, and image pyramids.

10.1.2 Learning and Recognition in Vision (LEAR)

LEAR[9] is a joint team of INRIA Grenoble - RhôneAlpes and the LJK laboratory, a joint research unit of the Centre National de Recherche Scientifique (CNRS), the Institut National Polytechnique de Grenoble (INPG), the Université Joseph Fourier (UJF) and Université Pierre-Mendès-France (UPMF). LEAR's main focus is learning based approaches to visual object recognition and scene interpretation, particularly for object category detection, image retrieval, video indexing and the analysis of humans and their movements. The Head of the group is Cordelia Schmid. Her major areas of interest are image and video description, object and category recognition and machine learning.

10.1.3 Texto+Imagenes+Aprendizaje (TIA)

TIA[10] was established on July 2006 with the goal of doing research that can help to bridge the semantic gap between low-level features (extracted from images) and high-level concepts (in which users are interested) for multimedia image retrieval. It belongs to the Department of Computer Science from INAOE (National Institute for Astrophysics, Optics and Electronics), Puebla, Mexico. Their research interests are on automatic segmentation, automatic image annotation, content-based, text-based and annotation-based image retrieval and benchmarking multimedia image retrieval.

[6] http://www.visigrapp.org/
[7] http://www.iciap2009.org/index.jsp?page=main.jsp
[8] http://www.prip.tuwien.ac.at/research
[9] http://lear.inrialpes.fr/index.php
[10] http://ccc.inaoep.mx/~tia/index.htm

11 Conclusions

Research in Content-Based Image Retrieval (CBIR) has been focused on image processing, low-level feature extraction, etc. Experiments on CBIR systems demonstrate that low-level image features cannot always describe high-level semantic concepts in the users' mind.

This report has reviewed many methods regarding Content-Based Image Retrieval, from the very basic image representation as visual features, to the retrievals methods to get images from the databases. We have identified two major approaches to add semantics to the images. One is automatic image annotation, which intends to endow image regions and images as a whole with text descriptions describing the concepts present in them. The other is by means of Relevance Feedback, where the user intention and evaluation is involved in the process of retrieval, by choosing the most relevant images according to his criterion. Nevertheless, current approaches still have drawbacks that prevent the fully semantic description of image content in CBIR systems.

It is believed that CBIR systems should provide maximum support in bridging the "semantic gap" between low-level visual features and the richness of human semantics by including the spatial relations between image regions, especially the topological relations, which are shown to be invariant to a number of transformations. Also, by using a hierarchical model, it will be possible to add semantics at different levels of abstraction, making possible to pose queries with very flexible information and at various levels of details.

Thus, our approach will be based on the following general tasks:

- Represent images by characterizing the topological relations between regions in a hierarchical framework. Tentatively, we can use the irregular dual pyramid approach.
- Describe regions according to their visual content by choosing a descriptor that best fit our needs.
- Create a descriptor that takes into account both visual features and topological relations between regions, and define a similarity measure for computing similarities between images.
- Perform automatic image annotation to semantically label all the regions represented by the hierarchical representation at each level.
- Perform the retrieval process combining text annotations and visual features, applying relevance feedback to obtain better responses according to the user's needs.

References

1. N. Alajlan, M. S. Kamel, and G. Freeman, "Multi-object image retrieval based on shape and topology," *Signal Processing: Image Communication*, vol. 21, 2006.
2. E. Chang, K. Goh, G. Sychay, and G. Wu, "CBSA: Content-based Soft Annotation for Multimodal Image Retrieval Using Bayes Point Machines," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 1, pp. 26-38, 2003.
3. K. D. Toennies, K. Boehm, C. Herrmann, and I. Schmitt, "The representation of shape for retrieval of pictures by semantic means," in *Proceedings of the International Workshop for Computational Visualistics, Media Informatics and Virtual Communities*, 2003.
4. D. A. Lisin, M. B. Mattar, M. B. Blaschko, M. C. Benfield, and E. G. Learned-Miller, "Combining Local and Global Image Features for Object Class Recognition," in *Proceedings of the 2005 Conference on Computer Vision and Pattern Recognition*, 2005, pp. 47--.
5. X. M. Zhou, C. H. Ang, and T. W. Ling, "Image Retrieval based on Object's Orientation Spatial Relationship," *Pattern Recognition Letters*, vol. 22, no. 5, pp. 469--477, 2001.
6. S. Santini and R. Jain, "Similarity Measures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, pp. 871--883, 1999.
7. K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615--1630, 2005.
8. J. Zhang, M. Marszalek, S. Lazebnik, and C. Schimd, "Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study," *International Journal of Computer Vision*, vol. 73, no. 2, 2007.
9. A. Shah-hosseini, "Semantic Image Retrieval using Relevance Feedback and Transaction Logs," Engineering Science, Louisiana State University, etd-07132007-091706, 2007.
10. R. Smith and S. F. Chang, "Automated Image Retrieval using Color and Texture," Columbia University, CU/CTR 408-95-14, 1995.
11. J. Han and K. Ma, "Fuzzy Color Histogram and Its Use in Color Image Retrieval," *IEEE Transactions On Image Processing*, vol. 11, pp. 944-952, 2002.
12. J. Huang, S. R. Kumar, M. Mitra, W. Zhu, and R. Zabih, "Image Indexing Using Color Correlograms," in *Proc. IEE Conf. on Computer Vision and Pattern Recognition*, 1997, pp. 762-768.
13. N. R. Howe and D. P. Huttenlocher, "Integrating Color, Texture and Geometry for Image Retrieval," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. II, 2000, pp. 239-246.
14. B. G. Prasad, K. K. Biswas, and S. K. Gupta, "Region-based image retrieval using integrated color, shape, and location index," *Computer Vision and Image Understanding*, vol. 94, no. 1-3, pp. 193--233, 2004.
15. Various Authors, *Image Databases: Search and Retrieval of Digital Imagery*.: John Wiley & Sons, Inc., 2002.
16. S. Oraintara and T. T. Nguyen, "Using Phase and Magnitude Information of the Complex Directional Filter Bank for Texture Image Retrieval," in *Proc. IEEE International Conference on Image Processing*, vol. 4, 2007, pp. 61-64.
17. O. Martinez Bruno, L. G. Nonato, M. A. Pazoti, and J. B. Neto, "Topological multi-contour decomposition for image analysis and image retrieval," *Pattern Recognition Letters*, vol. 29, no. 11, pp. 1675--1683, 2008.

18. H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features," in *In Proceedings of ECCV 2006*, 2006, pp. 404-417.

19. E. Hadjidemetriou, M. D. Grossberg, and S. K. Nayar, "Multiresolution Histograms and Their Use for Recognition," *IEEE Transactions on Pattern Analysis and Machne Intelligence*, vol. 26, no. 7, pp. 831--847, 2004.

20. I. Dimitrovski, D. Kocev, S. Loskovska, and S. Dzeroski, "ImageCLEF 2009 Medical Image Annotation Task: PCTs for Hierarchical Multi-Label Classification," in *Proceedings of the 10th international conference on Cross-language evaluation forum: multimedia experiments*, 2010, pp. 231--238.

21. K. Grauman and T. Darrell, "Pyramid Match Kernels: Discriminative Classification with Sets of Image Features," in *Proceedings of the Tenth IEEE International Conference on Computer Vision*, 2005, pp. 1458-1465.

22. D. Zhong and I. Defée, "Performance of similarity measures based on histograms of local image feature vectors," *Pattern Recognition Letters*, vol. 28, no. 15, pp. 2003--2010, 2007.

23. K. Grauman and T. Darrell, "Efficient Image Matching with Distributions of Local Invariant Features," in *In IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 627--634.

24. H. Ling and K. Okada, "Diffusion Distance for Histogram Comparison," in *In IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 246--253.

25. S. Lazebnik, C. Schmid, and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," in *Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition*, 2006, pp. 2169--2178.

26. M. J. Egenhofer, J. Sharma, and D. M. Mark, "A Critical Comparison of the 4-Intersection and 9-Intersection Models for Spatial Relations: Formal Analysis," in *Auto-Carto*, vol. 11, 1993.

27. J. Chen, Z. Li, C. Li, and C. M. Gold, "Describing topological relations with Voronoi-based 9-intersection model," *International Archives for Photogrammetry and Remote Sensing*, vol. 32, no. 4, pp. 99-104, 1998.

28. L. Brun and W. Kropatsch, "Contraction Kernels and Combinatorial Maps," *Pattern Recognition Letters*, vol. 24, no. 8, pp. 1051--1057, 2003.

29. L. Brun and W. Kropatsch, "Contains and Inside relationships within Combinatorial Pyramids," *Pattern Recognition*, vol. 39, no. 4, pp. 515--526, 2006.

30. C. Hernández-Gracidas and L. E. Sucar, "Markov Random Fields and Spatial Information to Improve Automatic Image Annotation," in *Proc. of the 2007 Pacific-Rim Symposium on Image and Video Technology*, 2007, pp. 879--892.

31. L. Brun and W. Kropatsch, "Introduction to Combinatorial Pyramids," in *Digital and image geometry*.: Springer-Verlag New York, Inc., 2001, pp. 108--128.

32. W. G. Kropatsch, Y. Haxhimusa, and P. Lienhardt, "Hierarchies relating Topology and Geometry," in *In Proceedings of Cognitive Vision Systems*, vol. 3948, 2006, pp. 199–220.

33. L. Brun and W. Kropatsch, "Dual Contraction of Combinatorial Map (Technical Report)," Pattern Recognition and Image Processing Group, Institute of Computer Aided Automation, Vienna, PRIP-TR-54, 1999.

34. W. G. Kropatsch and L. Brun, "Hierarchies of Combinatorial Maps," in *Proceedings of the Czech Pattern Recognition Workshop*, vol. Czech Pattern Recognition Workshop, 2000, pp. 131–137.

35. Z. Muda, "Ontological Description of Image Content Using Regions Relationships," in *In ESWC Phd Symposium, European Semantic Web Conference*, 2008, pp. 46-50.

36. A. Hanbury, "A survey of methods for Image Annotation," *Journal of Visual Language Computing*, vol. 19, no. 5, pp. 617--627, 2008.

37. T. Osman, D. Thakker, G. Schaefer, M. Leroy, and A. Fournier, "Semantic Annotation and Retrieval of Image Collections," in *Proceedings 21st European Conference on Modelling and Simulation*, 2007, pp. 324-329.

38. M. Srikanth, J. Varner, M. Bowden, and D. Moldovan, "Exploiting Ontologies for Automatic Image Annotation," in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, 2005, pp. 552--558.

39. J. Liu, M. Li, and W. Y. Ma, "An Adaptive Graph Model for Automatic Image Annotation," in *Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, 2006, pp. 61--70.

40. J. Li and Z. Wang, "Automatic Linguistic Indexing of Pictures by a Statistical Modeling Approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1075--1088, 2003.

41. C. Cusano, G. Ciocca, and R. Schettini, "Image annotation using SVM," in *Internet imaging V*, vol. 5304, 2004, pp. 330-338.

42. S. L. Feng, R. Manmatha, and V. Lavrenko, "Multiple Bernoulli Relevance Models for Image and Video Annotation," in *Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition*, 2004, pp. 1002--1009.

43. David M. Blei and Michael I. Jordan, "Modeling annotated data," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval.*, 2003, pp. 127-134.

44. W. Zheng, Y. Ouyang, J. Ford, and F. S. Makedon, "Ontology-based Image Retrieval," in *In 6th WSEAS Int. Conf. on Mathematical Methods and Computational Techniques in Electrical Engineering*, 2004.

45. J. Liu, B. Wang, H. Lu, and S. Ma, "A graph-based image annotation framework," *Pattern Recognition Letters*, vol. 29, no. 4, pp. 407--415, 2008.

46. Z. Su and H. Zhang, "Relevance Feedback in CBIR," in *In Proceedings of VDB'2002*, 2002, pp. 21-35.

47. J. Cui and C. Zhang, "Combining Stroke-based and Selection-based Relevance Feedback for Content-based Image Retrieval," in *Proceedings of the 15th international conference on Multimedia*, 2007, pp. 329--332.

48. F. Jing, B. Zhang, F. Lin, W. Y. Ma, and H. J. Zhang, "A Novel Region-Based Image Retrieval Method Using Relevance Feedback," in *Proceedings of the 2001 ACM workshops on Multimedia: multimedia information retrieval*, 2001, pp. 28--31.

49. H. Jair Escalante et al., "Annotation-Based Expansion and Late Fusion of Mixed Methods for Multimedia Image Retrieval," in *Proceedings of the 9th Cross-language evaluation forum conference on Evaluating systems for multilingual and multimodal information access*, 2009, pp. 669--676.

50. R. C. Veltkamp and M. Tanase, "Content-Based Image Retrieval Systems: A Survey," Department of Computing Science, Utrecht University, UU-CS-2000-34, 2002.

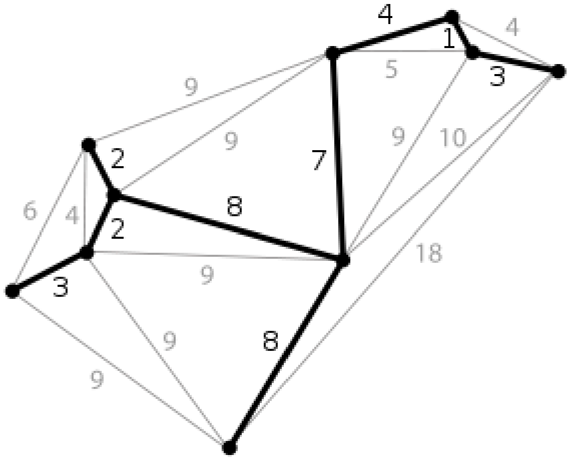
51. P. Kontschieder, M. Donoser, and H. Bischof, "Beyond Pairwise Shape Similarity Analysis," in *Proceedings of Asian Conference on Computer Vision (ACCV)*, 2009, pp. 655-666.

52. Y. Haxhimusa, A. Ion, and W. G. Kropatsch, "Comparing Hierarchies of Segmentations: Humans, Normalized Cut and Minimun Spanning Tree," in *Proceedings of 30th OEAGM Workshop*, 2006, pp. 95--103.

Annexes

Annex 1: Minimum Spanning Tree Segmentation Method

In this method, the image is transformed into an attributed graph representation G=(V,E,*w*), where vertices represent pixels and edges their neighborhood. Instead of cutting edges, the edges are added to connected components based on the minimum spanning tree principle.



In order to decide which component to merge, a function is defined that measures the difference along the boundary of two components relative to a measure of differences of components' internals differences.

In the pairwise comparison of neighboring vertices, the function returns true if the external contrast difference between two partitions is grater that the internal contrast differences within a partition, and this means that a border exists between this two partitions. It returns false in the opposite case, meaning that the border does not exist, and thus, these partitions are merged.

This method is able to produce a stack of graphs, for example, a hierarchy of partitions. The pyramid is built from bottom to top [52].