Noname manuscript No.
(will be inserted by the editor)

Simple object recognition based on spatial relations and visual features represented using irregular pyramids

Annette Morales-González ·
Edel García-Reyes

Received: date / Accepted: date

Abstract Spatial relations among objects and object parts play a fundamental role in the human perception and understanding of images, thus becoming very relevant in the computational fields of object recognition, scene understanding and content-based image retrieval. In this work we propose a graph matching scheme that involves color, texture and shape features along with spatial descriptors to represent topological and orientation/directional relationships - which are obtained by means of combinatorial pyramids - in order to identify similar objects from a database. We also suggest a method for deciding which are the more useful levels in the hierarchy of segmentation for the recognition process. Our main objective is to prove that the combination of visual and spatial features is a promising road in order to improve the object recognition task. We performed experiments on two well known databases, COIL-100 and ETH-80 image sets, in order to evaluate the expressiveness of the proposed representation. These sets introduce challenges for simple object recognition in terms of view-point changes, and our results were comparable or superior than other state-of-the-art methods.

Keywords object recognition · spatial relations · topological relations · structure matching · graph matching

1 Introduction

A variety of representations used in computer vision consist of unordered sets of features. This is the case of the bag-of-features approach [47] and the pyramid matching kernel [11], which, in general terms, compute features from

Annette Morales-González · Edel García-Reyes
Pattern Recognition Department, CENATAV, Cuba
Tel.: (53-7) 272 -0422, 272-1670, 272-1676
Fax: (53-7) 273-0045
E-mail: {amorales,egarcia}@cenatav.co.cu

image patches and the frequency of occurrence of the modeled features in each image is used for classification. In these approaches, objects are usually characterized by distinct appearances, textures, shapes or parts, disregarding the spatial information existing among features and parts.

Spatial relations between objects of a scene have received much attention in the field of image analysis and retrieval, due to the fact that they can reveal important properties of the scene being analyzed. Moreover, it has been stated that structural relations among image components are fundamental in the human process of similarity comparison [28].

In general, spatial relations can be classified into three major categories [12]: (1) Topological relations, which remain invariant under transformations such as translation, scaling and rotating. (2) Direction (orientation) relations, which specify the absolute or relative spatial locations of objects. (3) Metric relations, which deal with sizes of objects or the distance between them.

Within this context, there have been some approaches which intend to add spatial information to image descriptions by means of visual features [13][19][24] and points of interest methods. They try to capture the spatial distribution of visual features in the image, but they do not identify regions and their relations. In order to solve this, the region-based approaches have been presented. Nevertheless, there are many works related to region-based representation of images that do not use spatial information among regions, or they do it poorly [44]. Also, there are methods that only use direction relations [37][35][30], or topological relations [18][26], and others that combine them together [17][43][16]. Most of these representations consider that each object is ideally identified or deals with its bounding box to compute the spatial relation descriptors. Nevertheless, this cannot be applied for automatic image segmentation where objects are often arbitrarily over-segmented, or the cases where bounding boxes overlap. For the case of [43], the spatial relations are defined for the context of sport scenes, where they model several orientation relations but just one topological relation (*overlap*) is taken into account. In [16] spatial relations are used to improve automatic image annotation, and although several orientation relations are used, in the topological set only the *adjacent* relation is employed. An attempt to express complex spatial relations by means of elementary spatial relations is proposed by [17], and this information is used to check semantic consistency of tasks such as segmentation. The relationships are expressed in a qualitative way and it might be difficult to compute similarities between them. In [8], a deformable part-based model is proposed for object detection, where they use a spatial model that reflects the cost of placing the center of a part at different locations relative to the whole object. Although this approach is very interesting, the spatial relations represented in this way are limited.

One explicit representation of spatial relations among regions is the region adjacency graph (RAG) [4]. This defines a simple graph from a given partition of the image, by associating one vertex to each region and creating an edge between two vertices if the associated regions are adjacent. However, the unique notion of adjacency is too poor to describe complex spatial organization of the different parts of an object, and does not provide enough information to differentiate an adjacency relationship from a *contains* or *inside* one [4]. In [34], a visual graph model is used for scene recognition. They employ a directed graph that can be seen as a RAG, where the vertices represent visual concepts obtained from regular regions of the image and the edges represent only two spatial relations: *left of* and *top of*.

Irregular graph pyramids [2] can overcome these drawbacks by using dual graphs to determine important edges in the pyramid construction. In this case, each level will be an extended RAG, where parallel edges and self-loops encode important relations between two regions (relevant parallel edges represent several common boundaries and self-loops represent a *contains* relation). Besides the spatial relations representation, the hierarchy of partitions provided by the irregular pyramid is an important source of information at different levels, that can be very helpful for tasks such as object recognition, image annotation and retrieval, etc. Moreover, the use of hierarchical segmentation algorithms reduces the influence of the over/under segmentation problem and thus increases the number and the quality of the matches.

Within the context of medical image retrieval, the use of hierarchical RAGs is proposed in [42] and [9]. They use a segmentation hierarchy to build one graph representing regions of an image. This graph encodes adjacency relations among nodes and hierarchical relations among different levels. Nevertheless, the method proposed to build the hierarchy, and therefore, the underlying graph, is too computationally expensive to be applied in general-purpose images. Another aspect to notice is that the contribution of the spatial relations in the graph matching scheme is reduced to the use of very simple relations (adjacency and hierarchy).

One of the contributions of this work is the proposal of a spatial descriptor which is easy to build, store and manipulate, and can be employed to explicitly represent many possible spatial configurations between pairs of image regions, taking into account several basic orientation and topological relationships. This spatial descriptor is used to label the edges of the irregular pyramid's graphs.

Using image segmentation for image recognition and classification purposes is useful, since it may provide precise localization of regions and objects in a scene, but even for images where the same object is shown, segmented regions might be significantly different due to varied object's directions and lighting conditions. We believe that using a hierarchy of partitions may help to overcome this problem, but also the use of spatial relations will play an important role. For this reason, given a segmentation hierarchy of an image, we decided to use several levels (and their underlying graphs) in the representation, in contrast to [9], which combines all the levels in a single graph.

The present work is an extension of our previous approach presented in [29]. In this paper we use the combinatorial pyramid framework [21] to obtain a hierarchy of partitions from an image and to determine the spatial relations between the regions found at each level. We introduce a representation to compute a spatial relations descriptor, taking into account topological and orientation relations, and a similarity measure for this descriptor is proposed. We describe a new graph matching scheme which combines visual, spatial and shape information in order to identify simple objects from a database, and we suggest a method for deciding which are the more useful levels in the segmentation hierarchy for the recognition process.

Section 2 of this paper explores the main characteristics of irregular pyramids. Section 3 explains the visual representation used for describing the images and the similarity measures selected for comparison. In Section 4 we present our novel spatial descriptor and a way for computing the similarity using this representation. Section 5 describes the proposed graph matching scheme and the selection of the pyramid levels to be used in it. Experimental results are presented in Section 6 and Section 7 concludes the paper.

2 Irregular Pyramids Overview

Irregular graph pyramids are formed by a region adjacency graph (RAG) per level. In these graphs $G = (V, E)$ the vertices (V) represent the cells or regions, and the edges (E) represent the neighborhood relations of the regions. When we build an irregular pyramid from an image, each level represents a partition of the pixel set into cells, i.e. connected subsets of pixels. On the base level (level 0) of the image pyramid, the cells represent single pixels and the neighborhood of the cells is defined by the 4-connectivity of the pixels. A cell on level $k+1$ (parent) is a union of neighboring cells on level k (children) [14]. Descriptions of the regions and their relationships can be 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). 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. In this way, each surviving vertex represents all the non surviving vertices mapped to it and becomes their father [22]. This parent-child relationship may be iterated down to the base level and the set of children of one vertex in the base level is named its receptive field (RF). The steps for building the pyramid are roughly depicted in Figure 1.

The selection of the surviving vertices (white vertices in Figure 1c) can be performed in different ways. One of them is to find a *maximal independent set (MIS)* which fulfills the constraints that each non-surviving vertex must be adjacent to at least a surviving one, and that two adjacent vertices cannot survive. More information regarding the selection of surviving vertices can be found in [2][23].

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. The lack of self-loops does not allow to differentiate inclusion from adjacency relationship. The lack of parallel edges prevent from having

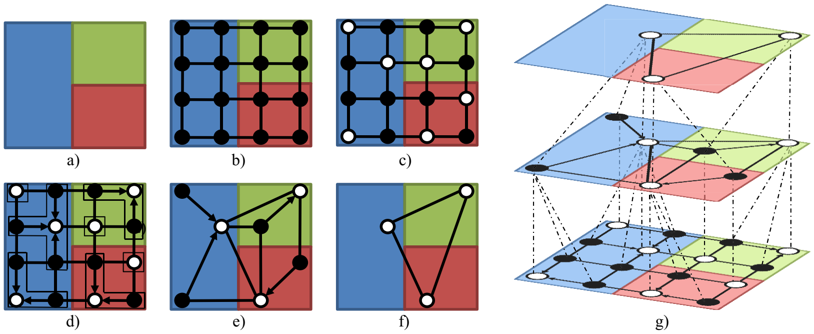


Fig. 1 Building an irregular pyramid from an image. a) original image, b) base level graph (G_0), c) white nodes represent the nodes that will survive to the next level, d) contraction kernels (CK) for each surviving node (the arrows indicate the surviving node corresponding to each non-surviving node), e) level G_1 built from G_0, f) level G_2 built from G_1. In g) the level hierarchy is shown (levels G_0, G_1 and G_0 are presented from bottom to top)

information regarding multiple common boundaries between two adjacent regions.

To overcome these problems, dual graph pyramids are introduced. In order to correctly represent the embedding of the graph in the image plane, the dual graph $\overline{G} = (\overline{V}, \overline{E})$ 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 [22].

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. This set is called a Contraction Kernel (CK) [4]. Figure 1d) shows an example of the contraction kernels in one level of the pyramid. 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. The contraction process must follow two steps [2]:

1. A set of edge contractions on G_K encoded by the CK. The dual of the contracted graph G_{K+1} is computed from $\overline{G_K}$ by removing the dual of the edges contained in the CK.
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.

The spatial relations encoded by the pyramid can be seen in Figure 2. At a given level of the pyramid, each edge represents an adjacency relation between two vertices (regions). In Figure 2a this is represented by the edges connecting region A with region D, and region D with region B. Also, it is possible to have parallel edges between two vertices, representing multiple adjacency, which is the case of region A with region B. The inclusion relation is represented by a simple edge denoting adjacency and a self-loop, surrounding the region that is inside. This configuration can be seen between region A and region C.

One of the drawbacks of the irregular pyramid codification is that it can encode the presence of an inclusion relationship, but using graphs it is not possible to know which region is inside and which one is outside just by having a self-loop [4]. This can be seen in Figure 2b, where the self-loop can be either surrounding region A or region C.

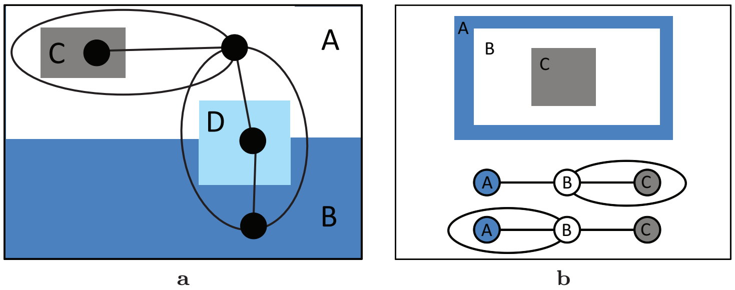


Fig. 2 Spatial relations encoding within the irregular pyramid framework. a) Representation of simple adjacency, multiple adjacency and contains/inside relationship. b)Example case where the encoding of the inclusion relationship can not distinguish which region is outside and which one is inside.

Combinatorial pyramids [4] are introduced in order to properly characterize the inclusion relationship. In this case, the edges orientation around a vertex is needed. 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 $d1$ and $d2$, each dart belonging to only one vertex), α is the reverse permutation which maps $d1$ to $d2$ and $d2$ to $d1$ and σ is the successor permutation which encodes the sequence of darts encountered when moving around a vertex [4]. This can be seen in Figure 3.

The dual of a CM 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 [4].

A combinatorial pyramid is then a stack of successively reduced combinatorial maps, having the advantages that each CM explicitly encodes the orientation of darts around each vertex and the dual is defined on the same set of darts by the permutations φ, therefore, only one data structure has to be encoded and maintained along the pyramid [3].

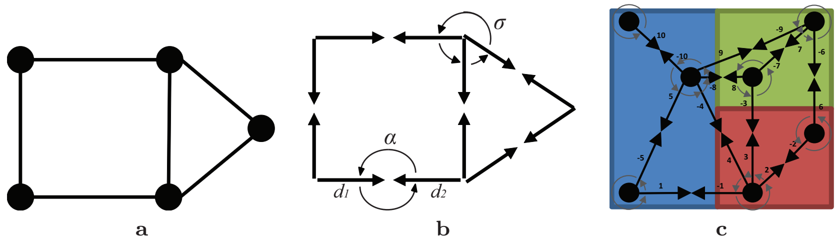


Fig. 3 Equivalence between a graph and a combinatorial map. a) Example graph, b) combinatorial map equivalent to the graph in a, c) combinatorial map equivalent to the graph in Figure 1e. In c, darts are represented by numbered segments, σ by gray arrows (example $\sigma(1)$=-5) and two darts linked by α are drawn consecutively and share the same number, but with different signs (example $\alpha(1)$=-1).

3 Visual Description of Regions

Beyond using combinatorial pyramids to obtain a stack of image segmentations at different levels, this structure can be used to store and represent important information regarding images. If we properly characterize each region and its relationships with its neighborhood, it could be exploited for other tasks such as object recognition.

Every node in the combinatorial pyramid represents a region in the image. These regions can be characterized using low-level features, so they can successfully represent distinctive parts of the objects. There is a large number of image features that have been proposed in the literature to accomplish this. In the present work we use color and texture to represent image segments, but other features can be included in the future. Shape features are used to represent a group of image segments.

Color features are widely used for object recognition and image retrieval. That's why one of the visual features selected to represent these regions is their color histogram in RGB space. The three dimensional values of the RGB color space make the discriminative power of this representation superior with respect to the one dimensional values of the grayscale images. For this reason, we built a histogram with 16 bins per channel, yielding a total of 48 bins.

For texture representation we chose the local binary patterns (LBP) histogram of regions [40]. The LBP operator codes a local window pattern from a texture patch, and its histogram is often treated as texture feature in classification problems. Among the advantages of LBP are its invariance to any monotonic change in gray level and its computational simplicity. Its efficiency originates from the detection of different micro patterns (edges, points, constant areas etc.). Furthermore, some studies have shown that local binary patterns can produce good texture discrimination [41][15]. We extract the local binary patterns of a local circular window to represent the feature of the center pixel and the LBP distribution of one region is approximated by a LBP histogram.

The structure of the combinatorial pyramid is perfect for computing statistical features, such as histograms. The computation of each region's histogram can be performed during the construction of the pyramid very easily, updating each level from the data of the level below. Given an image obtained by computing the LBPs from the original image, it is possible to update each region's histogram at each level by using the following equation:

$$H(R)_j = \sum_{i=1}^{n} H(i)_{j-1} \tag{1}$$

Where n is the number of regions merged into the current region R, and j is the level of the pyramid. The same applies for the construction of the color histograms.

Shape is a very important feature when it comes to recognizing objects. Two objects may have similar colors and textures, and the only way to differentiate them is by analyzing their shape. Let's take for example, the comparison between apples and pears, where the main difference between these two object categories is given by their shapes.

In this case, for representing shapes, we chose the Legendre moments [5]. According to the comparison performed by [6], the Legendre moments show very good results for handwritten character recognition. They also have been successfully used for object classification [1] and image retrieval [36]. The Legendre moments for a $(N \text{ X } N)$ digital image is given by [5]:

$$L_{pq} = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} P_p(m_N) P_q(n_N) f(m,n), \tag{2}$$

where p and q are integers between $(0, \infty)$, being $(p+q)$ the order of the Legendre moment computed, P_p and P_q are the Legendre polynomials, and

$$m_N = \frac{2m - N + 1}{N - 1} \tag{3}$$

The shape descriptor is then the concatenation of the Legendre moments into a single vector.

3.1 Computing visual similarity

Once the visual features to be used are defined, one important step is to select the similarity measures for them. Since our main contribution is not in the topic of visual similarity, we chose two well-known similarity measures for our features.

The LBP histograms and the color histograms of the pyramid regions are normalized, since the different sizes of regions produce uneven histograms. For histogram similarity we use the Bhattacharyya distance, which is then transformed into a similarity measure.

For combining the similarity values obtained for the LBP histograms and the color histograms, we add two weights, ω_C and ω_H, in order to give different importance to color and texture, and to have a final value of visual similarity between two regions r_1 and r_2:

$$S_V(r_1, r_2) = \omega_C * S_C(r_1, r_2) + \omega_H * S_H(r_1, r_2), \qquad \text{where } S_V(r_1, r_2) \in [0, 1] \tag{4}$$

S_C and S_H are the color histogram similarity and LBP histogram similarity respectively.

For comparing the Legendre moments vectors we use Euclidean distance, which gives a dissimilarity value between the vectors. Nevertheless, shape is not included in the previous combination, because S_V is intended to represent the similarity between two regions of two different objects. Shape (dis)similarity must be computed over the whole object silhouette, that's why the inclusion of this dissimilarity in the comparison process will be explained in further detail in Section 5.

4 Spatial Relations in 2D Images

There have been several models proposed for representing spatial relations among regions. For topological relations, the 4IM and 9IM [7] are well known. In these models, and for the case of 2D images, eight topological relations are described: *disjoint*, *contains*, *inside*, *equal*, *meet*, *covers*, *covered by* and *overlap*. The main drawback of these models is their inability to represent complex topological relations (i.e. when two regions have more than one boundary in common).

For the case of 2D images, eight relations are unnecessary since some of them will never be present (i.e the *overlap* relation). In 2D images, we certainly can have occlusion (two objects overlapped), but at the time of segmentation we will be unable to establish a difference between this and a simple adjacency relation, since we will have only a boundary in common. We selected from these eight relations, three that will be representative for 2D images. These relations can be seen in Figure 4.

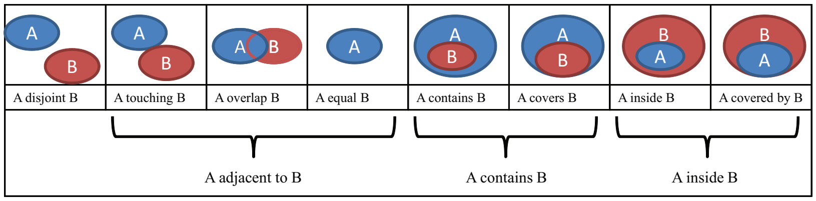


Fig. 4 Topological relations between 2D regions and the selection for 2D images.

Although these relationships are implicitly encoded in the local neighborhood of the combinatorial pyramid, retrieving them is not always an easy task and in some cases it may involve several steps [4]. The graphs in the irregular pyramid are not simple, i.e they contain parallel edges and self-loops, therefore, performing matching operations on this kind of structures is not straightforward. This is why we chose to create a spatial descriptor that explicitly encodes the spatial configuration between two regions.

We consider that orientation relations between regions can also provide important information, therefore the new spatial descriptor should take both types of relations into account. Accordingly, we decided to use the relations *left of*, *right of*, *top of*, *bottom of*, *horizontally aligned* and *vertically aligned*, somehow similar to the order relations proposed in [16]. These relations will be computed based on the spatial disposition of the centroids for every pair of regions.

4.1 Spatial Descriptor

Our spatial description proposal consists of a binary vector that will encode both topological and orientation relations. The vector will have 9 elements, each representing one basic spatial relation, as shown in Figure 5. For every position, we put a 1 if the two regions share that spatial relation and 0 otherwise. These basic relations are split into three categories: (1) Topological relations - *adjacent*, *contains* and *inside*, (2) Alignment relations - *horizontally aligned* and *vertically aligned*, (3) Orientation relations - *left of*, *right of*, *top of* and *bottom of.*

H	V	L	R	T	B	A	C	I

H – Horizontally aligned
V – Vertically aligned

L – Left of
R – Right of

T – Top of
B – Bottom of

A – Adjacent
C – Contains
I – Inside

Fig. 5 Spatial descriptor combining topological and orientation relations.

We also store, for every pair of related regions, the number of common boundary segments, which will be a descriptor of the adjacency between them.

For computational purposes, each value of the descriptor will be stored as bits. This leads us to a 9 bit (2 bytes with 7 unused bits) representation, which is very simple, compact and easy to use.

4.2 Spatial Relationship Similarity

In order to compute the similarity between two spatial relations, we need to find out how many basic relations they share, this is why we chose a similarity measure that can be used with binary vectors. We are proposing to use the Sokal-Michener measure [38] since it treats positive and negative matches equally. It has shown good performance compared with other measures reported in [46] and it is very easy to compute. Let X and Y be binary vectors of the same length d and let S_{ij} $(i, j \in \{0, 1\})$ denote the number of occurrences of matches with i in X and j in Y at the corresponding positions. The Sokal-Michener measure for two spatial descriptors sp_1 and sp_2, which annotate edges e_1 and e_2 respectively, can be computed as:

$$S_{SD}(sp_1, sp_2) = \frac{S_{11} + S_{00}}{d} \tag{5}$$

The term S_{11} denotes the positive matches (i.e. the number of 1 bits that matched between X and Y) and the term S_{00} denotes the negative matches (i.e. the number of 0 bits that matched between X and Y).

When computing the spatial similarity between two pairs of regions, we consider that all the basic relations should not contribute in the same way to the final result. We think that topological relations are more reliable than the others in the present case, since they are invariant to transformations such as scaling, translating and rotation. Therefore, they must have a bigger weight in the decision of whether two spatial relations are similar or not. In the same way, we consider the alignment relations to be more important than the orientation relations. For this reason we decided to use three weights ω_T, ω_A and ω_O for topological, alignment and orientation relations respectively, following the criteria $\omega_T > \omega_A > \omega_O$. These weights will be applied to every element's match/mismatch in the computation of the Sokal-Michener measure, using the weight corresponding to the basic spatial relation represented by the element in each case.

5 Matching strategy

In order to take into account the spatial relations between object's parts, we chose to implement a graph matching algorithm, since this makes it possible to compute similarity between images. In the present case, we're not interested in finding the similarity between two images, but in finding similarities between the objects of each image, so we are talking about a subgraph matching problem.

One important advantage of using irregular pyramids for this process is that they provide a hierarchy of partitions of one image. Having several segmentation levels instead of only one can be very useful in the recognition process, since different partitions of the same image will yield diverse and useful information. But we still have the problem that not all the pyramid levels will make such contribution, since we can find levels so over-segmented or under-segmented, that they will not provide helpful representations. These levels will be a burden in the matching process, making it slower without adding meaningful information. Therefore, it would be desirable to dismiss the unwanted partitions.

5.1 Selecting the Pyramid Levels for the Matching Process

Having the entire hierarchy of partitions, it is easy for a person to manually select one level or several levels that one might find ”better” segmented according to some criteria. Problems emerge when we want to do the same thing automatically. In this case we confront issues like which level(s) of the pyramid we should use or if we were to select one level, can we be sure that the main object parts are well represented in such partition?

For this reason, we decided to evaluate the levels of the pyramid in order to decide which levels are the best ones (according to the measure defined). In [39], a method for simplifying segmentation hierarchies is proposed. The selection of the most semantic levels in the hierarchy is done by using spectral graph theory, which in our case is not applicable, since our graphs contain multiple edges to represent inclusion and multiple adjacency.

We believe that image edges can be an important criteria to evaluate the partitions. When a partition does not preserve all relevant edges in the image, it usually means that several regions from different objects or background were merged into one single region, thus losing very useful information. Moreover, even a partition that segments the object as a whole silhouette may not be the best one, since we are more interested in finding object parts and their relations, in order to provide discriminative information to the object recognition algorithm.

Since we do not have *apriori* knowledge regarding the image, we choose a Canny filter to determine relevant edges, and use the resulting edges mask as reference to evaluate the segmentation at each level. The Canny edge detector presupposes a notion of continuity by using thresholding with hysteresis for the detection. In general, this edge detector shows good performance and is very fast, although it presents some drawbacks such as three parameters to adjust and the Y-junctions problem. In the future, more sophisticated edge detectors can be tested to improve this process. Before applying the Canny detector, the images are smoothed to reduce the influence of noise. For evaluating each partition of the pyramid we propose the following measures:

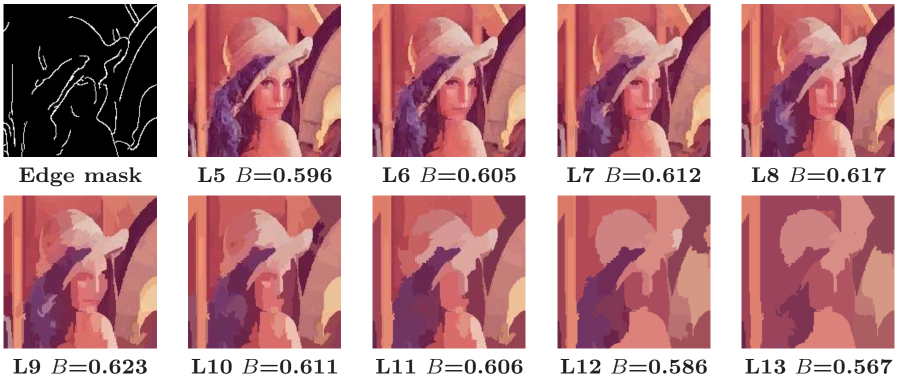
$$B_G = \frac{|P \cap R|}{|R|} \tag{6}$$

$$B_B = 1 - \frac{|P \backslash R|}{n} \tag{7}$$

where P is the set of all edge pixels from the partition being evaluated, R is the set of edge pixels in the Canny mask image and n is the total number of pixels in the image. $|.|$ is the cardinality of set. Measure 6 evaluates how well the partition edges matched those of the Canny mask, and measure 7 evaluates how many border pixels in the partition are not present in the Canny mask. Thus, measure 6 tends to favor over-segmented partitions while measure 7 does the opposite, and penalizes partitions with more edges than those present in the mask, so these measures are combined into a global measure B using two weights ω_G and ω_B.

$$B = \omega_G * B_G + \omega_B * B_B \tag{8}$$

Some sample results of the level evaluation using the B measure can be seen in Figure 6. In this example, the 9th level of the hierarchy obtained the best evaluation. We can see in level 10 that some edges of Lena's face were lost, thus mixing a portion of the face with a background region. Also some edges of the background objects were not kept in this partition. This is why the B value starts to decrease from level 10 onwards.


Edge mask **L5 B=0.596** **L6 B=0.605** **L7 B=0.612** **L8 B=0.617**

L9 B=0.623 **L10 B=0.611** **L11 B=0.606** **L12 B=0.586** **L13 B=0.567**

Fig. 6 Example level evaluation of the pyramid.

5.2 Matching Process

We use a greedy algorithm to find matchings between structures and, in order to prune the search space,we use the visual similarity measure (Eq. 4) and the spatial similarity measure (Eq. 5) proposed previously to discriminate nodes and edges that are too different to be taken into account. Also, in order to reduce the matching time, we choose for the comparison the best three levels of each pyramid (according to Eq. 8).

In a nutshell, the algorithm takes an input graph G (which in this case is the best level evaluated in the pyramid) that must be compared with a number of selected levels of some pyramid. For each graph (selected level) in the pyramid we find all the similar structures to the input graph. We take every node in the input graph and we compare it to each node in a level of the pyramid, and if they are visually similar, according to equation 4, then we try to expand the structure - in a greedy way - by testing the node's edges using the weighted S_{SD} measure in equation 5. If they are spatially similar, we repeat the process for every node they connect. This matching strategy is based on the algorithm proposed in [20].

Once we have a substructure T that matched with the input graph G, the corresponding matching nodes (representing regions) $\{t_1, t_2 .. t_n\}$ and $\{g_1, g_2 .. g_n\}$, and the corresponding matching edges (representing spatial relations) $\{r_1, r_2 .. r_m\}$ and $\{e_1, e_2 .. e_m\}$, we compute an overall visual similarity value as:

$$VS(T,G) = \sum_{i=1}^{n} S_V(t_i, g_i) \tag{9}$$

and an overall spatial similarity value as:

$$SS(T,G) = \sum_{i=1}^{m} S_{SD}(r_i, e_i) \tag{10}$$

The contour associated with substructure T, $C(T)$, is given by:

$$C(T) = C(\bigcup_{i=0}^{n} RF(t_i)) \tag{11}$$

where $C(\cdot)$ is the set of pixels belonging to the boundary of the corresponding image region and $RF(t_i)$ is the receptive field of the node t_i (See Section 2).

The Legendre moments vector of this contour is compared with the corresponding vector of the input graph's contour, as explained in Section 3, resulting in the shape distance $ShD(T,G)$.

Having these three measures, the overall similarity value between T and G is given by:

$$S(T,G) = \frac{VS(T,G) * SS(T,G)}{ShD(T,G)} \tag{12}$$

The structure with the highest S value will be the best match for the input graph. This matching strategy is roughly depicted in Figure 7.

Analyzing the time complexity of this algorithm, we find that testing every vertex of one graph against all vertices in the other graph takes $O(n^2)$ operations, where n is the number of vertices in one image graph. Once we are standing in any arbitrary pair of vertices, the substructure expansion between them is performed in $O(n^2)$ operations, since the maximum vertex degree is $n-1$, and we don't visit twice a vertex or edge that has already been matched in a given expansion iteration (the similarities between edges are computed only once and stored for future iterations). This gives an overall time complexity of $O(n^4)$. Nevertheless, this time complexity is given for the worst case scenario, which is very unlikely to happen. In fact, the pruning performed with the visual and spatial similarity measures plays a crucial role in the performance of the algorithm. We ran some empirical tests to evaluate how much these measures are able to prune the search space and we obtained the results

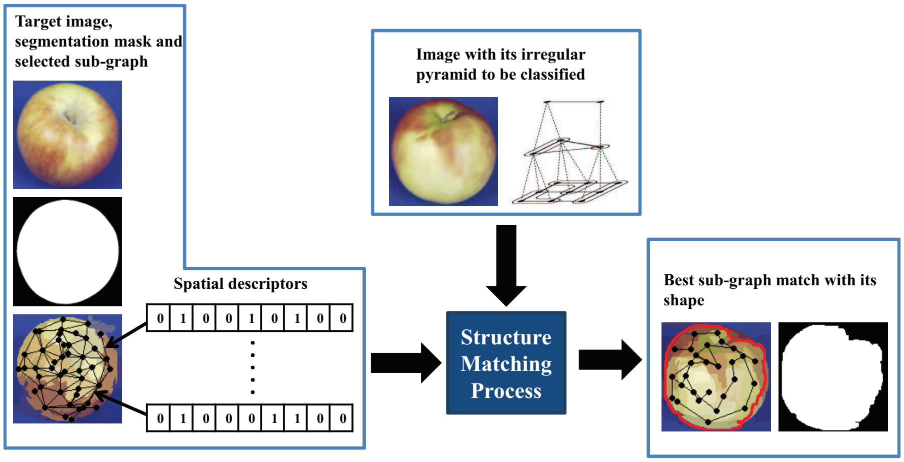

Target image, segmentation mask and selected sub-graph
Spatial descriptors
0 1 0 0 1 0 1 0 0
0 1 0 0 0 1 1 0 0
Image with its irregular pyramid to be classified
Structure Matching Process
Best sub-graph match with its shape

Fig. 7 Sub-structure matching process.

presented in Table 1. The test was performed for the case where random images were compared (first column), where only similar images were compared (second column) and where only dissimilar images were compared (third column). The spatial pruning is performed over the filtered output of the visual pruning. Logically, the pruning for similar images is smaller than the pruning for dissimilar ones, since it is expected for similar images to find more matchings. This analysis gives the notion that, although the theoretical complexity might seem high, we are able to prune approximately 96% of the possible branches involved in the expansion process.

Table 1 Evaluating how much the visual and spatial measures can prune the search space.

	random images	similar images	dissimilar images
Spatial pruning	16.3%	9.4%	52.4%
Visual pruning	97.7%	93.5%	99.8%
Total pruning	96.5%	91.3%	99.7%

Since image segmentation is crucial for our method, it is also important to be aware of the disadvantages that it may introduce. In the segmentation step (performed during the construction of the pyramid), even for images where the same object is shown, segmented regions might be significantly different due to varied object's directions and lighting conditions. Consequently, pyramids of graphs are different and it may seem difficult to appropriately calculate similarities. Nevertheless, the use of the level hierarchy helps find levels where partitions are relatively similar. By computing similarity through the regions in each level and their spatial relationships, it is possible to overcome the problem of lighting or viewing direction variations, since the proposed matching scheme allows to obtain partial matchings, and is flexible to slight variations of color and texture of the regions being compared. Take for example the apples shown in Figure 7. Although the apple is a whole object it is represented by several nodes, because it is not completely homogeneous. Therefore, patches of the apple where the color/texture is different, due to illumination effects or because of the nature of the object itself, are represented by different nodes that will encode the features of each patch. The next time we find an apple with a similar distribution, in different positions, we will be able to match them using this strategy.

The experiments described in Section 6 aim to test the robustness of this approach under such variations.

6 Experiments Description and Results

The Pascal VOC Challenges[1]are among the most recognized benchmarks to test object detection and classification algorithms, and are widely acknowledged as difficult testbeds for these tasks. The goal of these challenges is to recognize objects from a number of visual object classes in realistic scenes (i.e. not pre-segmented objects). Nevertheless, our method focuses on the recognition of simple objects (which in the future may be extended to detection tasks, but is not the main goal of the present work), therefore we rather test its performance using simple object databases, that introduce challenges in terms of object viewpoints variations.

We chose two well known databases: COIL-100 [31] and ETH-80 [25] image sets. Both databases contain images of simple objects taken from different viewpoints. We also count with ground truth segmentation masks that precisely delimit the object in each image.

For both databases, the experiment setup is the same. Having the training and testing sets defined, the images of the training set are represented by a single graph, which belongs to the best segmented level (according to Eq. 8) of each pyramid. For selecting the sub-graph that will represent each object, we used the segmentation mask and we get all the regions (nodes) that are totally inside of this mask. The test images are represented by the whole pyramid, but in the matching process only the best three evaluated levels are involved.

The combinatorial pyramids of these images have an average of 16 levels. The base level contains 16385 nodes and 33020 edges, while the uppermost level usually has 2 nodes and 1 edge. In most cases, the level selected by the evaluation process for representing the database images has between 40 and 50 nodes, and about 130 edges.

The main goal of the experiment is to recognize similar objects in the database and, to accomplish that, we find the nearest neighbor of each example image among the images in the database. We consider a positive match if the nearest neighbor of the example image belongs to its category.

[1] http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2010/

6.1 Weights Computation

For this experiment, the weights involved in the different measures previously presented were set empirically. For computing the visual similarity in Eq. 4, we considered that color and texture should contribute equally to the final result, thus we decided that $\omega_C = \omega_H = 0.5$. In the tested images, the texture patterns were not relevant enough to think that texture should have bigger weight than color. In the same way, in many cases, color is not the most prominent difference between images, therefore it should not have a bigger weight than texture. It is the balance between them what makes the greater discrimination.

Other parameters to tune are the weights involved in the spatial similarity computation (ω_T - for topological relations, ω_A - for alignment relations and ω_O for orientation relations). We tried to model how the human observers evaluate the similarity between two spatial relations. We prepared a set of visual configuration of binary spatial relations. An example of this can be seen in Figure 8.

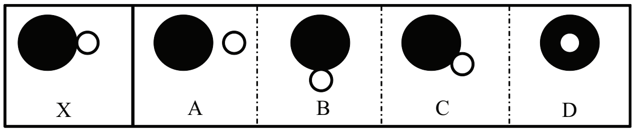


Fig. 8 Example relations used to test how humans perceive spatial similarity. According to our model, the relation of the black region with respect to the white region in each case is described as: Relation X - Horizontally aligned, Left of, Adjacent; Relation A - Horizontally aligned, Left of; Relation B - Vertical aligned, Top of, Adjacent; Relation C - Left of, Adjacent; Relation D - Contains.

The human observers were asked to perform a ranking of similarities among relations A, B, C, and D with respect to the spatial relation X. In general, we found that people picked relations C and B more similar to relation X. Note for relation B, that the orientation and alignment relations are completely different to X. Nevertheless, the common topological relation (adjacency) has a bigger effect to evaluate the spatial similarity by the human observers. Most humans have ranked the relation D in the last position because the topological relation is different. Relation A is located often in penultimate place even though X and A share similar orientation and alignment relations, but the topological relation is different. Following this kind of reasoning, we assigned a higher weight to the topological relations over the alignment and orientation relations. We tested several weight configurations until the output ranking of the spatial similarity measure was consistent with the human perception. Finally, the three weights were set as $\omega_T = 4$, $\omega_A = 2$ and $\omega_O = 1$.

For the measures used to evaluate the pyramid levels (Eq. 8), we performed several tests using an image set that contained human-made ground-truth segmentations of them. We tested how well the best level evaluated by the

B measure matched the ground-truth boundaries, and we selected the weight configuration that yielded better results in this sense. The weights were set to $\omega_G = 0.4$ and $\omega_B = 0.6$, which indicates that we are trying to avoid mostly over-segmentation.

6.2 COIL-100 Image Set

We performed one experiment in the Columbia Object Image Library (COIL-100) image set [31], which is a database of color images of 100 objects. These images were taken at pose intervals of 5 degrees along one axis of rotation, yielding 72 poses per object. In Figure 9 some examples are shown.

Fig. 9 Example images from the COIL-100 Image Set database.

For this experiment we took 25 objects randomly selected. We extracted 11% of the images corresponding to these objects to build the training set and the remaining images were used as testing set. In this case, the experiments correspond to object identification. We compared our results with other methods presented in the literature and these have been summarized in Table 2. The first column of this table shows the algorithm name (or some alias that we used to name it), third and fourth columns present the number of training or testing images used in the experiments respectively. The fifth column shows whether or not the algorithm uses spatial relations and the last column exhibits the recognition rates obtained in each case.

Table 2 Global recognition accuracy on the COIL-100 database

Algorithm	Year	Training (views/object)	Testing (views/object)	Spatial Relations	Recognition Rate
DTROD-AdaBoost [45]	2006	4	68		84.5 %
RSW+Boosting [27]	2005	1	71		89.2 %
Sequential Patterns [30]	2008	8	64	X	89.8 %
Proposed method	2011	8	64	X	91.6 %
LAF [33]	2002	8	64		99.4 %

As it can be seen in Table 2 our method performs slightly better than the others presented, except for the LAF (Local Affine Frames) algorithm [33] which was specifically designed to deal with severe view changes of the same object, which makes it a good method for object identification, but probably not that good under generalization. However, taking into account the simple descriptors employed for representing regions, it is a good sign that we were able to obtain results comparable to the other methods. In [30], spatial relations are used in the form of sequential patterns, where foci-of-interest (FOI) present in the images are concatenated through paths extracted from a complete graph of the FOIs. Directional relations between these FOIs are introduced in the sequential patterns that represent images. Although our algorithm obtained a better score than [30], taking into account that the visual features employed in their approach are more sophisticated than ours, we believe that if we use more robust features to describe regions probably our results can be improved.

6.3 ETH-80 Image Set

The second experiment was performed on the ETH-80 Image Set database [25] which contains 80 objects from 8 categories (*apples*, *cars*, *cows*, *cups*, *dogs*, *horses*, *pears* and *tomatoes*). Each object is represented by 41 different views yielding a total of 3280 images (See Figure 10). This database is more challenging than the COIL-100 database in the sense of the viewpoint diversity.

Fig. 10 Example images from the ETH-80 Image Set database.

For this experiment we split up the database, leaving 24% of the images in the database and the remaining 76% were used as the examples to be classified. The goal of this experiment is to categorize each object.

We compared our results with those obtained in [32]. They proposed a collaborative ensemble learning model where they constructed four types of ensemble classifiers (L2 , L3 , L4 and L5) by integrating two, three, four and five base learners respectively. We compared our method with the L5 ensemble classifier, which showed the best results.

The comparison result regarding the recognition accuracy for each category can be seen in Figure 11.

According to these results, our algorithm outperforms L5 in the recognition of *apples*, *cups*, *tomatoes* and *cows*. The remaining categories did not show improvements in the recognition accuracy compared to the L5 ensemble classifier, although *pears* got very close. This shows that our method is better for discriminating objects with obvious differences in appearance and shape, and is worse with the opposite situation, which is the case of *dogs* and *horses*.

The resulting confusion matrix for the proposed method can be seen in Table 3. It shows that the most relevant misclassification problems occur within objects of the same type, for instance, animals (*dogs*, *cows* and *horses*), which

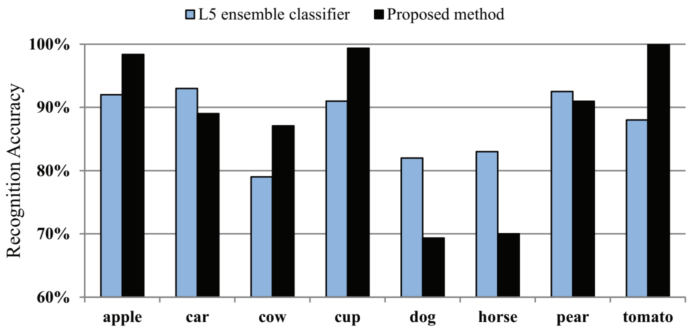


Fig. 11 Recognition accuracy for the proposed method compared with other reported methods in the ETH-80 dataset.

in several cases present similar texture, color and shape. Our method makes a better differentiation between *apples* and *tomatoes*, that share a very similar shape and color, but different textures. It also makes a good division between *apples* and *pears*, that share similar color and texture, but different shapes.

Table 3 Confusion matrix of the experiment performed in the ETH-80 dataset

	apple	car	cow	cup	dog	horse	pear	tomato	
apple	305	0	0	0	0	1	0	4	98.4%
car	1	276	17	5	5	1	1	4	89.0%
cow	2	9	270	1	4	22	2	0	87.1%
cup	1	1	0	308	0	0	0	0	99.4%
dog	5	7	45	2	215	30	6	0	69.4%
horse	3	6	51	4	29	217	0	0	70.0%
pear	25	1	2	0	0	0	282	0	91.0%
tomato	0	0	0	0	0	0	0	310	100%
	89.2%	92.0%	70.1%	96.3%	85.0%	80.1%	97.0%	97.5%	88.0%

We also compared these results with other methods in terms of overall recognition rates. This comparison can be seen in Table 4.

Table 4 Global recognition accuracy in the ETH-80 dataset

DTROD-AdaBoost [45]	RSW+Boosting [27]	Pyramid match kernel [10]	L5 ensemble classifier [32]	Proposed method
76.0%	79.6%	82.0%	87.2%	88.0%

Again, our method slightly outperforms the others. Nevertheless, given the small difference compared to other methods, we venture to say that the spatial relations are not enough to discriminate between objects, although it can provide important cues.

6.4 Assessing the role of spatial relations within our approach

In order to assess the relevance of the spatial relations in our approach, we decided to make an additional experiment where the spatial relations are not taken into consideration for the matching process.

In this case, given two images having the proposed representation, we find similarities between each region of one image against every region in the other one, only using the visual similarity measure. An N x M matrix is created, where N is the number of regions in the first image and M is the number of regions in the second one, and the similarity between each pair of regions is stored. After this step, we use the Hungarian Algorithm to find the best matching configuration among the regions, disregarding this way all the spatial information between them. After the matching configuration is found, we sum up all the similarity values of the regions matched, as depicted in Equation 9 thus obtaining the global $VS(T,G)$ value for the images. The image having the highest score is considered to be the best match for the input image.

Using this approach, the first thing to notice is that the matched regions between two images are usually disconnected, as it can be seen in Figure 12. Therefore, finding the matched regions contour using Equation 11 and using the shape distance $ShD(T,G)$ does not seem to be a good idea in this case. Nevertheless, in order to be consistent with our original approach, we ran another experiment using the shape information extracted.

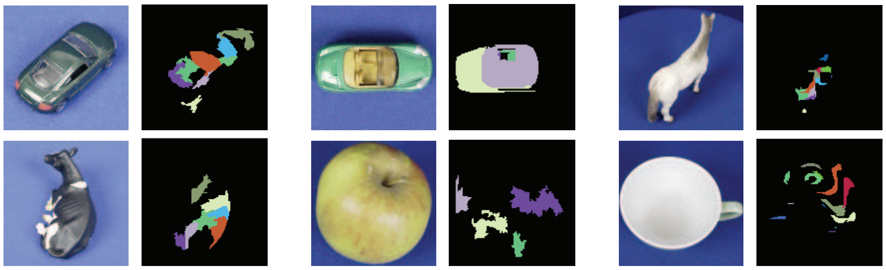


Fig. 12 Example of misclassified images using only visual similarity. In the top row the input images appear , and the bottom row shows the best match found in the database in each case. Even columns show the matched regions between the images, where the matches can be recognized by their color. (Best seen in color)

The results over the ETH-80 image set can be seen in Table 5. As it can be noticed, the results obtained without using spatial relations are far worse than those that use them. When the shape measure is introduced in the experiment, the results slightly improve but are still very distant from being good or close to the original approach.

Table 5 Comparison of the results obtained using spatial relations and not using spatial relations in terms of recognition rate. Method 1 stands for our proposed approach, method 2 represents this approach without using spatial relations (only visual similarity) and method 3 is the same as method 2, but the shape distance is included in the process.

Method	apple	car	cow	cup	dog	horse	pear	tomato	overall
1	98.4%	89.0%	87.1%	99.4%	69.4%	70.0%	91.0%	99.4%	88.0%
2	92.9%	54.3%	17.1%	90.0%	17.1%	0%	0%	95.7%	45.9%
3	97.1%	67.1%	21.4%	94.3%	15.7%	4.3%	48.6%	97.1%	55.7%

This shows that integrating spatial structure for matching different object parts is important, because even when there is a correspondence among the visual features, it is necessary to check the spatial consistency. This spatial consistency is also very important in order to recover the shape contour from the substructure, considering it as a connected component. Otherwise, the object may be recovered as a split shape.

That said, we still believe that, by using more sophisticated visual descriptors of the regions, our results can improve significantly, but then again, the spatial relations should not be discarded in this process.

7 Conclusions

In this work we have proposed a new method for representing and combining visual features of images (such as color, texture and shape) with spatial relations between regions, obtained from the partitions of combinatorial pyramids, in order to improve object recognition tasks. We proposed the use of a spatial similarity measure to test the similarity between spatial features, and also a graph matching scheme to compute the overall similarity between objects. We performed experiments that proved that the object recognition accuracy can be improved by taking into account the spatial distribution of object parts, even when the visual description of the image regions is simple, and especially when we do not have *apriori* knowledge regarding the objects present in it.

In future works we plan to improve the evaluation of the pyramid levels that will participate in the recognition process, since we are basing our evaluation only in the image edges - thus inheriting the problems related to edge detection - disregarding other visual cues that may help in the evaluation. Also, using another classifier , such as SVM, instead of the nearest neighbor classifier, may improve the results. We also plan to introduce and test other visual features to characterize the pyramid's regions in order to better discriminate among objects that share notable similarities regarding their shape and appearance.

References

1. Arif, T., Shaaban, Z., Krekor, L., Baba, S.: Object classification via geometrical, zernike and legendre moments. Journal of Theoretical and Applied Information Technology **7**(1), 31–37 (2009)
2. Brun, L., Kropatsch, W.: Introduction to combinatorial pyramids pp. 108–128 (2001)
3. Brun, L., Kropatsch, W.: Contraction kernels and combinatorial maps. Pattern Recogn. Lett. **24**(8), 1051–1057 (2003). DOI http://dx.doi.org/10.1016/S0167-8655(02)00251-9
4. Brun, L., Kropatsch, W.: Contains and inside relationships within combinatorial pyramids. Pattern Recogn. **39**(4), 515–526 (2006). DOI http://dx.doi.org/10.1016/j.patcog.2005.10.015
5. Cheriet, M., Kharma, N., Liu, C.l., Suen, C.: Character Recognition Systems: A Guide for Students and Practitioners. Wiley-Interscience (2007)
6. Duval, M.A., Vega-Pons, S., Llano, E.G.: Experimental comparison of orthogonal moments as feature extraction methods for character recognition. In: CIARP'10 Proceedings, pp. 394–401 (2010). DOI http://dx.doi.org/10.1007/978-3-642-16687-7_53
7. Egenhofer, M.J., Sharma, J., Mark, D.M.: A critical comparison of the 4-intersection and 9-intersection models for spatial relations: Formal analysis. In: Autocarto 11, pp. 1–11 (1993)
8. Felzenszwalb, P.F., Girshick, R.B., McAllester, D.A., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Anal. Mach. Intell. **32**(9), 1627–1645 (2010). DOI http://doi.ieeecomputersociety.org/10.1109/TPAMI.2009.167
9. Fischer, B., Thies, C., Gld, M.O., Lehmann, T.M.: Content-based image retrieval by matching hierarchical attributed region adjacency graphs. In: Proc. SPIEMedical Imaging: Image Processing, vol. 5370, pp. 598–606 (2004)
10. Grauman, K., Darrell, T.: Pyramid match kernels: Discriminative classification with sets of image features. Tech. Rep. MIT-CSAIL-TR-2005-017, Massachusetts Institute of Technology, Cambridge (2005)
11. Grauman, K., Darrell, T.: The pyramid match kernel: Efficient learning with sets of features. J. Mach. Learn. Res. **8**, 725–760 (2007)
12. Guting, R.H., Iv, P.I., Hagen, F.: An introduction to spatial database systems. VLDB Journal **3**, 357–399 (1994)
13. Hadjidemetriou, E., Grossberg, M., Nayar, S.: Spatial Information in Multi-resolution Histograms. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. I, pp. 702–709 (2001)
14. Haxhimusa, Y., Kropatsch, W.G.: Segmentation Graph Hierarchies. In: Fred, A., Caelli, T., Duin, R.P., Campilho, A., de Ridder, D. (eds.) Proceedings of Joint International Workshops on Structural, Syntactic, and Statistical Pattern Recognition S+SSPR 2004. Springer, Berlin Heidelberg, New York
15. Heikkilä, M., Pietikäinen, M., Schmid, C.: Description of interest regions with local binary patterns. Pattern Recognition **42**(3), 425–436 (2009). DOI http://dx.doi.org/10.1016/j.patcog.2008.08.014
16. Hernández-Gracidas, C., Sucar, L.E.: Markov random fields and spatial information to improve automatic image annotation. In: PSIVT, pp. 879–892 (2007). DOI http://dx.doi.org/10.1007/978-3-540-77129-6_74
17. Hodé, Y., Deruyver, A.: Qualitative spatial relationships for image interpretation by using semantic graph. In: GbRPR, pp. 240–250 (2007). DOI http://dx.doi.org/10.1007/978-3-540-72903-7_22
18. Hsieh, J.W., Grimson, W.E.L.: Spatial template extraction for image retrieval by region matching. IEEE Transactions on Image Processing **12**(11), 1404–1415 (2003). DOI http://dx.doi.org/10.1109/TIP.2003.816013
19. Hurtut, T., Gousseau, Y., Schmitt, F.: Adaptive image retrieval based on the spatial organization of colors. Comput. Vis. Image Underst. **112**(2), 101–113 (2008). DOI http://dx.doi.org/10.1016/j.cviu.2007.12.006
20. Iglesias-Ham, M., Bazán-Pereira, Y., García-Reyes, E.B.: A multiple substructure matching algorithm for fingerprint verification. In: CIARP'07 Proceedings, pp. 172–181. Springer-Verlag (2007)

21. Illetschko, T., Ion, A., Haxhimusa, Y., Kropatsch, W.G.: Effective programming of combinatorial maps using coma - a c++ framework for combinatorial maps. Tech. Rep. PRIP-TR-106, PRIP, TU Wien (2006)
22. Kropatsch, W.G., Haxhimusa, Y., Lienhardt, P.: Cognitive Vision Systems: Sampling the Spectrum of Approaches, chap. 13. Hierarchies relating Topology and Geometry. Lecture Notes in Computer Science. Springer, Berlin Heidelberg, Dagstuhl (2004)
23. Kropatsch, W.G., Haxhimusa, Y., Pizlo, Z., Langs, G.: Vision pyramids that do not grow too high. Pattern Recogn. Lett. **26**, 319–337 (2005)
24. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2169–2178. IEEE Computer Society (2006). DOI http://dx.doi.org/10.1109/CVPR.2006.68
25. Leibe, B., Schiele, B.: Analyzing appearance and contour based methods for object categorization. In: In IEEE Conference on Computer Vision and Pattern Recognition (CVPR'03), pp. 409–415 (2003)
26. Lin, P.L., Tan, W.H.: An efficient method for the retrieval of objects by topological relations in spatial database systems. Inf. Process. Manage. **39**(4), 543–559 (2003). DOI http://dx.doi.org/10.1016/S0306-4573(02)00034-1
27. Marée, R., Geurts, P., Piater, J., Wehenkel, L.: Decision trees and random subwindows for object recognition (2005)
28. Markman, A.B., Gentner, D.: Structure mapping in the comparison process. American Journal of Psychology **113**(4), 501–538 (2000)
29. Morales-González, A., García-Reyes, E.B.: Assessing the role of spatial relations for the object recognition task. In: CIARP'10 Proceedings, pp. 549–556 (2010). DOI 10.1007/978-3-642-16687-7_72
30. Morioka, N.: Learning object representations using sequential patterns. In: Proceedings of the 21st Australasian Joint Conference on Artificial Intelligence: Advances in Artificial Intelligence, AI '08, pp. 551–561 (2008)
31. Nene, S.A., Nayar, S.K., Murase, H.: Columbia Object Image Library (COIL-100). Tech. rep. (1996)
32. Nomiya, H., Uehara, K.: Data Mining and Knowledge Discovery in Real Life Applications, chap. 9, pp. 157–166. IN-TECH (2009)
33. Obdrzálek, S., Matas, J.: Object recognition using local affine frames on distinguished regions. In: Proceedings of the British Machine Vision Conference 2002. British Machine Vision Association (2002). DOI http://www.bmva.org/bmvc/2002/papers/134/full_134.pdf
34. Pham, T.T., Mulhem, P., Maisonnasse, L., Gaussier, E., Lim, J.H.: Visual graph modeling for scene recognition and mobile robot localization. Multimedia Tools and Applications pp. 1–23–23. DOI 10.1007/s11042-010-0598-8
35. Punitha, P., Guru, D.S.: An effective and efficient exact match retrieval scheme for symbolic image database systems based on spatial reasoning: A logarithmic search time approach. IEEE Trans. on Knowl. and Data Eng. **18**(10), 1368–1381 (2006). DOI http://dx.doi.org/10.1109/TKDE.2006.154
36. Rao, C., Kumar2, S., Mohan3, B.: Content based image retrieval using exact legendre moments and support vector machine. International Journal of Multimedia and Its Application **2**(2) (2010)
37. Skiadopoulos, S., Koubarakis, M.: Composing cardinal direction relations. Artif. Intell. **152**(2), 143–171 (2004). DOI http://dx.doi.org/10.1016/S0004-3702(03)00137-1
38. Sokal, R.R., Michener, C.: A statistical method for evaluating systematic relationships. University of Kansas Science Bulletin **38**, 1409–1438 (1958)
39. Song, Y.Z., Arbelaez, P., Hall, P., Li, C., Balikai, A.: Finding semantic structures in image hierarchies using laplacian graph energy. In: Proceedings of the 11th European conference on Computer vision: Part IV, ECCV'10, pp. 694–707. Springer-Verlag, Berlin, Heidelberg (2010). URL http://portal.acm.org/citation.cfm?id=1888089.1888142
40. T. Ojala, M.P., Harwood, D.: A comparative study of texture measures with classification based on featured distribution. Pattern Recognition **29**(1), 51–59 (1996)
41. Takala, V., Ahonen, T., Pietikäinen, M.: Block-based methods for image retrieval using local binary patterns. In: SCIA, Lecture Notes in Computer Science, vol. 3540, pp. 882–891 (2005). DOI http://dx.doi.org/10.1007/11499145_89

42. Thies, C., Malik, A., Keysers, D., Kohnen, M., Fischer, B., Lehmann, T.M.: Hierarchical feature clustering for content-based retrieval in medical image databases. In: Proc. Medical Imaging, Proc. SPIE, pp. 598–608. San Diego, CA (2003)
43. Tsapatsoulis, N., Petridis, S.: Classifying images from athletics based on spatial relations. Semantic Media Adaptation and Personalization, International Workshop on **0**, 92–97 (2007). DOI http://doi.ieeecomputersociety.org/10.1109/SMAP.2007.49
44. Vieux, R., Benois-Pineau, J., Domenger, J.P., Braquelaire, A.: Segmentation-based multi-class semantic object detection. Multimedia Tools and Applications pp. 1–22–22 (2010). DOI 10.1007/s11042-010-0611-2. URL http://dx.doi.org/10.1007/s11042-010-0611-2
45. Wang, Y., Gong, S.: Tensor discriminant analysis for view-based object recognition. In: Proceedings of the 18th International Conference on Pattern Recognition - Volume 03, ICPR '06, pp. 33–36 (2006)
46. Zhang, B., Srihari, S.N.: Binary vector dissimilarity measures for handwriting identification. In: DRR, SPIE Proceedings, vol. 5010, pp. 28–38 (2003)
47. Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: A comprehensive study. Int. J. Comput. Vision **73**, 213–238 (2007)