Assessing the Role of Spatial Relations for the Object Recognition Task

Annette Morales-González and Edel García-Reyes

Advanced Technologies Application Center. 7a # 21812 b/ 218 and 222,
Rpto. Siboney, Playa, P.C. 12200, La Habana, Cuba
{amorales,egarcia}@cenatav.co.cu

Abstract. It has been proved that spatial relations among objects and object's parts play a fundamental role in the human perception and understanding of images, thus becoming very relevant in the computational fields of object recognition and content-based image retrieval. In this work we propose a spatial descriptor to represent topological and orientation/directional relationships, which are obtained by means of combinatorial pyramids. A combination of visual and spatial features is performed to improve the object recognition task. We ran an experiment to evaluate the expressiveness of this representation and it has shown promising results. It was performed on the benchmark ETH-80 Image Set database and we compare our approach with a state-of-the-art method recently published.

Keywords: object recognition, spatial relations, topological relations.

1 Introduction

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.

In general, spatial relations can be classified into three major categories [1]: (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 deals with sizes of objects or the distance between them.

Within this context, there are many works related to region-based representation of images that do not use the spatial information between regions, or they do it poorly. Also, there are methods that only use direction relations [2][3], only topological relations [4][5], and others that combine them together [6][7][8]. Most of these representations consider that each object is ideally identified or deals with their bounding box to compute the spatial relations descriptors. Yet, this does not match the case in a segmented image where objects are often arbitrarily over-segmented, or the cases when bounding boxes overlap.

I. Bloch and R.M. Cesar, Jr. (Eds.): CIARP 2010, LNCS 6419, pp. 549–556, 2010.

One explicit representation of spatial relations among regions is the region adjacency graph (RAG)[9]. 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[9].

Irregular graph pyramids [10] 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).

In this work we use the combinatorial pyramid framework [11] to obtain a hierarchy of partitions from an image and to determine the spatial relations between the regions found at each level. We propose a new representation to compute a spatial relations descriptor, taking into account topological and orientation relations. A similarity measure for this descriptor is proposed and a graph matching algorithm is used to identify similar images from a database. The spatial description of regions relationships is combined with visual descriptions of them to make more robust the recognition task.

Section 2 of this paper explains the visual representation used for describing the images and the similarity measures selected for comparison. In Section 3 we present our novel spatial descriptor and a way for computing the similarity using this representation. Finally Section 4 provides the results of an experiment to evaluate the proposed representation.

2 Visual Description of Images

Graph pyramids and combinatorial pyramids are built from bottom (each vertex is a pixel in the image) to top (each vertex is a group of pixels forming a region), and all levels in between form partitions of the image at different scales. To build a new level, a series of topology-preserving edge contractions are performed from the previous level, following some criteria [10]. The criteria for combining pixels into regions may vary.

In the present case, we are using only the color value in RGB color space of the image pixels. The difference between the color value of pixels is computed, and if it falls beneath a threshold, these pixels are merged into a region, which will survive to the next level and its color value will be the average of the pixels that were combined. That is why, one of the features selected for similarity purposes is the average color of each region.

For texture representation we chose the locally binary patterns (LBP) histogram of regions [12]. 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.

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.

2.1 Computing Visual Similarity

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

For computing visual similarity between two pairs of regions regarding color value, we will compute the Euclidean distance in RGB space. Since this distance will yield a dissimilarity value, we will turn it into a similarity value S_C.

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

For combining these similarity values, we add two weights, ω_C and ω_H, in order to give different importance to the features and to have a final value of visual similarity between two regions:

$$S_V = \omega_C * S_C + \omega_H * S_H, \qquad \text{where } S_V \in [0, 1] \tag{2}$$

3 Our Proposed Spatial Descriptor

There have been several models proposed for representing spatial relations among regions. For topological relations, the 4IM and 9IM [13] 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 of them that will be representative for 2D images. These relations can be seen in Figure 1.

We consider that orientation relations between regions can also provide important information, this is why we choose to create a spatial descriptor that

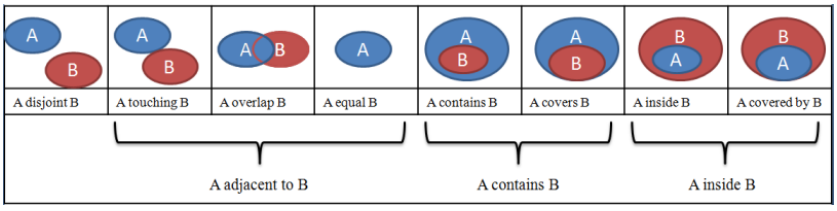

Fig. 1. Topological relations between 2D regions and the selection for 2D images

would take both types of relations into account. For this matter 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 [8]. These relations will be computed based on the spatial disposition of the centroids for every pair of regions.

3.1 The 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 2. 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	L	R	V	T	B	A	C	I

H – Horizontally aligned V – Vertically aligned A – Adjacent
L – Left of T – Top of C – Contains
R – Right of B – Bottom of I - Inside

Fig. 2. 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.

3.2 The 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 [14] since it treats positive and negative matches equally. Let X and Y be binary vectors of the same length d and let x_i denote the ith value which is either 0 or 1. The Sokal-Michener measure can be computed as:

$$S_{SD} = \frac{xy + \overline{xy}}{d} \tag{3}$$

The term xy denotes the positive matches (i.e. the number of 1 bits that matched between X and Y) and the term $\overline{xy}$ denotes the negative matches (i.e. the number of 0 bits that matched between X and Y).

We believe that, when computing the spatial similarity between two pairs of regions, all the basic relations should not contribute in the same way in the final result. We consider that topological relations are more relevant than the others, 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.

4 Experiments

For validating this representation, we chose to implement a graph matching algorithm since this makes possible to compute similarity between images. In the present case, we're not interested in finding the similarity between two images, but to find similarities between the objects of each image, so we are talking about a subgraph matching problem.

4.1 Matching Strategy

We are using a greedy algorithm to find matchings between structures but, in order to avoid the high complexity of this kind of algorithm, we used the visual similarity measure and the spatial similarity measure proposed previously to discriminate nodes and edges that are too different to be taken into account.

In a nutshell, the algorithm takes an input graph that must be compared to an irregular pyramid of graphs. For each graph (level) in the pyramid we find all the similar structures to the input graph. We take every node in the input graph and compare it to each node in a level of the pyramid, and if they are visually similar, according to equation 2, then we try to expand the structure by testing the node's edges using the weighted S_{SD} measure in equation 3. If they are spatially similar, we repeat the process for every node they connect. This matching strategy is based on the algorithm proposed in [15], please refer to this

work for further details in its implementation. We compute the final similarity between the structures as a combination of the average of spatial similarities of the matched edges and the average of visual similarities of the matched nodes.

4.2 Experiment Description and Results

We carry out the experiments using the ETH-80 Image Set database [16] 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 3).

Fig. 3. Example images from the ETH-80 Image Set database

For this experiment we used 6 categories. For each category we took 4 objects and for each object we took 10 different views, leaving a total of 240 images in the database. From the remaining images we took 60 per category (15 views per object) to be used as the examples to be classified. The main goal was to recognize similar objects in the database, then we found 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.

The combinatorial pyramids for the images of this database 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. The level selected for representing the example images has between 40 and 50 nodes, and about 130 edges.

For selecting the sub-graph that will represent the image for the matching process, we draw a square box having the same center of the image, and we get all the regions (nodes) that this box touches at a given level of the pyramid. A global view of the structure matching process can be seen in Figure 4.

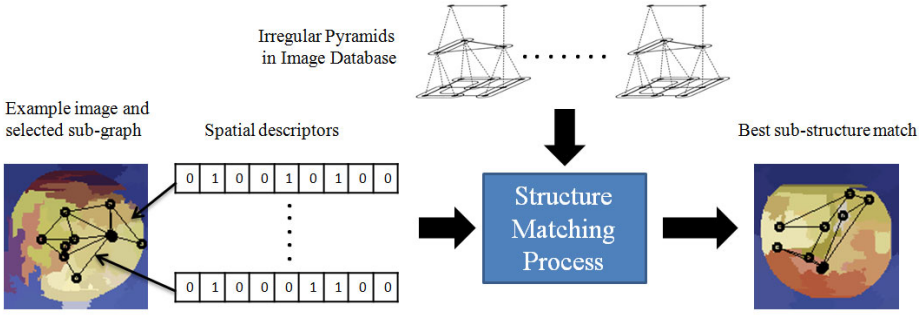

Fig. 4. Sub-structure matching process

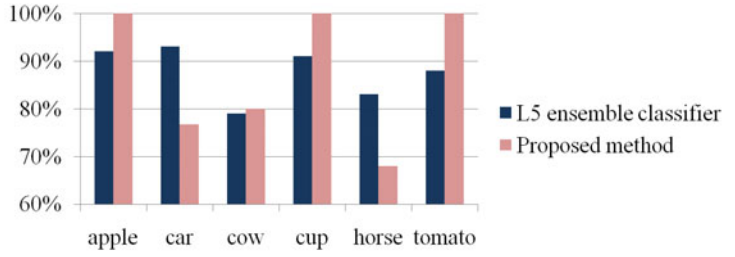

Fig. 5. Recognition accuracy for the L5 ensemble classifier and the proposed method

We compared our results with those obtained in [17]. They proposed a collaborative ensemble learning model where they construct 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 5.

According to these results, our algorithm outperforms the recognition of *apples*, *cups*, *tomatoes* and *cows* having a 100% of recognition accuracy for the first three of them. The categories of *cars* and *horses* did not show improvements in the recognition accuracy compared to the L5 ensemble classifier. We believe that this may be due to the form of selecting the initial graph for comparing, since the square box used to select the regions of the graph may incorporate several background regions in the case of horses and cars, due to their elongated shape.

The overall recognition accuracy for the L5 ensemble classifier is of 87.6%, while our method yields 87.5%. It is important to notice that, although we achieved a very similar global accuracy to the one obtain with the L5 ensemble classifier, the visual description that we used is much simpler, thus showing the relevance of the spatial relations.

5 Conclusions

In this work we have proposed a new approach for describing spatial relations between regions of images based on the partitions provided by combinatorial pyramids and we proposed a spatial similarity measure to test the similarity between this kind of features. We performed an experiment that proved that the object recognition accuracy can be improved by taking into account the spatial distribution of object's parts, even when the visual description of the image regions is very simple. In future works we plan to study the selection of the sub-graph in the example images, and to find optimal levels for comparison in the irregular pyramids in order to increase the performance of the method.

References

1. Guting, R.H., Iv, P.I., Hagen, F.: An introduction to spatial database systems. VLDB Journal 3, 357–399 (1994)
2. Skiadopoulos, S., Koubarakis, M.: Composing cardinal direction relations. Artif. Intell. 152(2), 143–171 (2004)
3. 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)
4. 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)
5. 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)
6. Hodé, Y., Deruyver, A.: Qualitative spatial relationships for image interpretation by using semantic graph. In: Escolano, F., Vento, M. (eds.) GbRPR. LNCS, vol. 4538, pp. 240–250. Springer, Heidelberg (2007)
7. Tsapatsoulis, N., Petridis, S.: Classifying images from athletics based on spatial relations. In: International Workshop on Semantic Media Adaptation and Personalization, pp. 92–97 (2007)
8. Hernández-Gracidas, C., Sucar, L.E.: Markov random fields and spatial information to improve automatic image annotation. In: Mery, D., Rueda, L. (eds.) PSIVT 2007. LNCS, vol. 4872, pp. 879–892. Springer, Heidelberg (2007)
9. Brun, L., Kropatsch, W.: Contains and inside relationships within combinatorial pyramids. Pattern Recogn. 39(4), 515–526 (2006)
10. Brun, L., Kropatsch, W.: Introduction to combinatorial pyramids, pp. 108–128 (2001)
11. Illetschko, T., Ion, A., Haxhimusa, Y., Kropatsch, W.G.: Effective programming of combinatorial maps using coma - a c++ framework for combinatorial maps. Technical Report PRIP-TR-106, PRIP, TU Wien (2006)
12. Pietikainen, M., Ojala, T., Harwood, D.: A comparative study of texture measures with classification based on featured distribution. Pattern Recognition 29(1), 51–59 (1996)
13. Egenhofer, M.J., Sharma, J., Mark, D.M.: A critical comparison of the 4-intersection and 9-intersection models for spatial relations: Formal analysis. Autocarto 11, 1–11 (1993)
14. Sokal, R.R., Michener, C.: A statistical method for evaluating systematic relationships. University of Kansas Science Bulletin 38, 1409–1438 (1958)
15. Iglesias-Ham, M., Bazán-Pereira, Y., García-Reyes, E.B.: A multiple substructure matching algorithm for fingerprint verification. In: Rueda, L., Mery, D., Kittler, J. (eds.) CIARP 2007. LNCS, vol. 4756, pp. 172–181. Springer, Heidelberg (2007)
16. Leibe, B., Schiele, B.: Analyzing appearance and contour based methods for object categorization. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2003), pp. 409–415 (2003)
17. Nomiya, H., Uehara, K.: Data Mining and Knowledge Discovery in Real Life Applications. IN-TECH, ch. 9, pp. 157–166 (2009)