PUBLISHING HOUSE
OF THE ROMANIAN ACADEMY

PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A,
Volume 4, Number 3/2003, pp.000 - 000

AN APPROACH TO IMAGE OBJECTS RECOGNITION AND INTERPRETATION

Tudor BARBU

Institute of Computer Science, Iaºi, Romania
E-mail: tudcont@iit.iit.tuiasi.ro

We have focused on a set of problems related to image pattern recognition, presenting an approach for the graphical objects' detection problem. First, in the introduction, we describe the general aspects of the computer vision system. Then, we provide approaches for image's objects identification, by composing the previously detected textured and non--textured regions, for the shape classification of the obtained objects and for a structure-based clustering of those image objects. Finally, after completing the steps of extraction and recognition, an objects' interpretation method is proposed. Our methods' results could be further applied in image retrieval and image understanding domains.

Key words: pattern recognition, clustering, graphical objects, shape recognition, structure classification, image understanding

1. INTRODUCTION

The objective of this paper is to present methods for image objects pattern recognition and understanding. We consider a graphical object as a connected set of pixels of an image, therefore an image region. From this point of view, an image may contain a great number of objects but most of them are just sintactic regions, having no special importance for us. Our task is to detect the semantic objects from a given image. Extracting the regions with a given meaning is very useful in image understanding (interpretation).

A color image, or a grayscale one, contains a finite set of uniform or (and) texture regions. We consider these textured and non--textured regions as simple objects or primitives (basic units) in our approach. Any complex image object consists of a set of connected simple objects. Usually, there are a lot of such complex objects in an image, many of them being meaningless. We consider a computer vision system, completing the following main steps:

1. A preprocessing stage ([2])
2. A uniformity--based or a texture--based image segmentation
3. Complex image objects identification by composing the segments obtained in the 2*th* step
4. A similarity--based grouping of the objects of each order in appropriate clusters
5. A semantic-based classification of previously obtained clusters

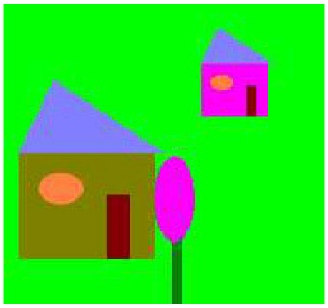

Fig. 1

The first part (first two steps) of the system is treated in our previous work, entitled ***A pattern recognition approach to image segmentation*** (see [1]). In Figure 1 the enhancement and segmentation results over a color textured image are presented, the resulted image being now our input image with 11 clusters. In last three steps our work has important contributions. First we provide a proper method for describing the complex graphical objects, based on shape, structure and primitives' neighboring relations. Then we provide a recursive algorithm for building these graphical objects. In the pattern recognition stage ([1], [3], [5]) we propose structure clustering methods and provide a moment-based shape invariant feature different from the known Hu moments ([3]). An object interpretation method is also proposed. We have implemented these approaches and obtained several graphical results.

Recommended by Horia-Nicolai TEODORESCU,
Member of the Romanian Academy

2. AN IMAGE OBJECTS IDENTIFICATION APPROACH

In this section we present methods for extraction and classification of the objects of an input image. The segments obtained as the clusterization result will become the primitives from which the complex objects are composed. As a primitive, a cluster is not characterized by its texture or uniformity anymore, the shape being its only characteristic. We are interested foreground objects only, so each image segment will be treated as a simple object if it is not the image's background. We have proposed the following reasoning for excluding the background: each cluster it is analysed and if the area of the smallest convex polygon which include that cluster is equal or greater than 3/4 of the total image's area, then we consider the cluster as background and do not include it in the set of primitives. In Figure 1 we have 10 primitives.

An n - order image object is composed of n such units (primitives). If it contains only one unit it is called first - order object. A complex object is characterized not only by shape but also by its structure which is related to the types of its components. The proposed recognition approach is based on the following property: 2 objects are considered similar if they have similar shapes and similar structures, uniformities or textures being not relevant. A given image I could have a great number of n - order objects $(n = 1,2,...)$. The extraction consists in determining the objects related to each level n. We have implemented a recursive method for objects' extraction.

Let $p_1, p_2,...p_N$ be the image units. An n - order object x of I can be therefore described as $x = \{p_{i_1}, p_{i_2},...,p_{i_n}\}$, where $i_1, i_2,...,i_n \in \{1,...,N\}$. An m - order subobject of x is an m - order object $y = \{p_{j_1}, p_{j_2},...,p_{j_m}\}$where $j_1,..., j_m \in \{i_1,...,i_n\}$. For x we can define its shape, $Shape(x)$, as the set of coordonates (X, Y) of the pixels x, therefore we have the relation:

$$\text{Shape}(x) = \text{Shape}(p_{i_1}) \cup \text{Shape}(p_{i_2}) \cup ...\text{Shape}(p_{i_n}) \tag{1}$$

Also, we can express the structure of x, $\text{Struct}(x)$ as the pair:

$$\text{Struct}(x) = (S(x), R(x)) \tag{2}$$

where $S(x) = \{i_1, i_2,...,i_n\}$ and $R(x) = R(p_{i_1},..., p_{i_n})$ represents a relationship between the primitives related to the way of combining them in the body of x. There are many ways to express R, each of them describing the relations of neighborhood in the set $\{p_{i_1},..., p_{i_n}\}$. The basis form of R is that of a neighborhoods matrix $n \times n$, $neigh_x^n$, given by:

$$\text{neigh}_x^n(a,b) = \begin{cases} 1, \text{ if } p_{i_a} \text{ and } p_{i_b} \text{ are neighbors} \\ 0, \ p_{i_a} \text{ and } p_{i_b} \text{ are not connected each other.} \end{cases} \tag{3}$$

To obtain easier implementations of these matrices, $neigh_x^n$ for each x, we build the general neighborhoods' matrix of the image, defining it in a similary with that described by (3):

$$\text{neigh}_I(i, j) = \begin{cases} 1, \text{ if } p_i \text{ and } p_i \text{ are connected} \\ 0, \text{ if } p_{i_a} \text{ and } p_{i_b} \text{ are not connected.} \end{cases} \tag{4}$$

This matrix is computed considering each pair of different simple objects, $(p_i, p_j), i \neq j$. These primitives are neighbors if there is at least one pair of connected pixels such as a pixel belongs to p_i and the other to p_j. We use neighborhoods of type 4 in this work.

For each $(p_i, p_j), i < j$ and $i, j \in \{1,\ldots,N\}$ all coordonate elements from $\text{Shape}(p_i)$ and $\text{Shape}(p_j)$ are searched until a pair (C_1,C_2), $C_1 \in \text{Shape}(p_i)$ and $C_2 \in \text{Shape}(p_j)$, is found having the property $d(C_1, C_2) = 1$. In this case p_i and p_j are neighbors, otherwise they are not.

Extracting an object x from the input image I means computing $\text{Shape}(x)$, $S(x)$ and $R(x)$. The following recursive algorithm is computing $S(x)$ only, which is enough because the other two characteristics are then computed from (1) and (3). The idea of our method consists in extracting the $n-1$- order objects first, then, for each of them adding to their structure the primitives from their neighborhood, the n- order complex objects being obtained this way.

```
1. If $n = 1$ then the $n$- objects are $p_i, i = \overline{1,N}$ and $S(p_i) = \{i\}, \forall i = \overline{1,N}$.
Else $\{n > 1\}$
2. Find the ($n-1$)- order objects recursively: $x_i$ and $S(x_i), k < N, i = \overline{1,K}$
  For $i = 1$ to $K$ do
     For each $p_s$ with $s \notin S(x_i)$
        If $\exists t \in S(x_i)$ such that $\text{neigh}_I(s,t) = 1$
        3. $S(x_i) = S(x_i) \cup \{s\}$;
        End
     End
  End
End
```

3. GRAPHICAL OBJECTS PATTERN RECOGNITION

Having all the image's objects, we can focus on their clustering problem now. Every object x is given by the pair $(\text{Shape}(x), \text{Struct}(x)) = (\text{Shape}(x), (S(x), R(x)))$ therefore the clustering task for x has to be divided in two parts: classification of $\text{Shape}(x)$ and classification of $(S(x), R(x))$. Obviously two objects of different orders cannot belong to the same class, even if they have similar shapes, because of their structures. Therefore the classification make sense for same order objects only. We could consider each n as determining a category of image objects: the class of n- order objects (let name it n - class). Each n - class is then divided into a new set of classes on the principles of shape and structure similarity. We have the following situations:

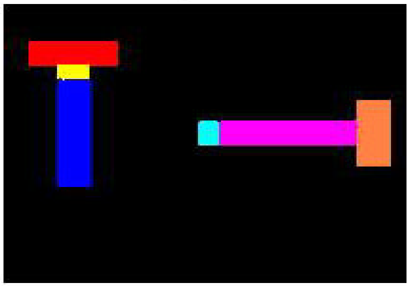

Fig. 2

1. $n = 1$: We do a shape classification only, $\text{Shape}(x)$ being the measure to be discriminated.
2. $n = 2$: First a shape classification is done. For each Shape - class a structure clustering is then done, therefore every class of shapes will contain a set of subclasses determined by $S(x)$.
3. $n \geq 3$: This case has a higher complexity because it involves $R(x)$ characteristic.

A shape classification is done first. Then, for each shape class an S- classification is performed over it. Finally, every S- class will be divided into subclasses determined by R- similarity. The reason we use an R- classification for $n \geq 3$ is exemplified by Figure 2: the two 3-order objects are not similar, although they have similar shapes and primitives, because of different R relationships. We can use $R(x) = \text{neigh}^n_x$, $n \geq 3$, or we can obtain new forms for R by using the 2 - order subobjects of x. From the relation $\text{neigh}^n_x(a,b) = 1$ if p_{i_a} and p_{i_b} are connected the equivalence between neigh^n_x and the set:

$$\left\{\left(p_{i_a}, p_{i_b}\right) \middle| a, b \in \{1,\dots,n\}, a \neq b \middle| p_{i_a}, p_{i_b} \text{ being neighbors} \right\}$$

results.

This set is equivalent with :

$$\left\{(i_a, i_b) \middle| p_{i_a} \text{ connected with } p_{i_b},\ i_a, i_b \in S(x)\right\} = \left\{S(x_2) \middle| x_2 \subset x,\ x_2 \text{ is 2 - order object}\right\},$$

in this way the 2-order subobjects of an n - order object are identified. These x_2 subobjects are then classified by shape and structure by the method of the case $n = 2$.

Let now describe our approach for shape recognition problem. Having a given shape, a similar new one can be obtained through geometric transforms, such as translation, rotation, scaling and reflection, or smooth deformations, such as stretching, squeezing and dilation. For characterizing a shape we need an invariant measure to these transformations. There are many shape pattern recognition methods, most known being Fourier descriptors ([4]) and Hu moments.

We have used a moment--based approach for shape analysis, obtaining similar feature vectors and results for area and perimeter moments. We present the area moments case only, when all the pixels of a pattern are used in computation. For an object o we consider all the elements of $Shape(o)$ the (x, y) coordonate paris corresponding to the pixels of o, and the image function :

$$f(x, y) = \begin{cases} 1, & (x,y) \in \text{Shape}(o) \\ 0, & (x,y) \notin \text{Shape}(o). \end{cases} \quad (5)$$

For this f, the $p + q$ - order moment of o become:

$$m_{p,q} = \sum_x \sum_y x^p y^q f(x, y) = \sum_x \sum_y x^p y^q,\ (x, y) \in \text{Shape}(o) \quad (6)$$

The center of gravity (centroid) of the object o has the coordonate pair $(\bar{x}, \bar{y})$ given by:

$$\bar{x} = \frac{m_{10}}{m_{00}},\ \bar{y} = \frac{m_{01}}{m_{00}}. \quad (7)$$

The central moment of o is obtained computing the pq moment with respect to the centroid:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q = \sum_x \sum_y \left(x - \frac{m_{10}}{m_{00}}\right)^p \left(y - \frac{m_{01}}{m_{00}}\right)^q \quad (8)$$

The obtained shape feature μ_{pq} is a translation invariant moment. Using the standard deviations :

$$\sigma_x = \sqrt{\frac{\text{ì}_{20}}{m_{00}}},\ \sigma_y = \sqrt{\frac{\text{ì}_{02}}{m_{00}}} \quad (9)$$

a stretching, squeezing and dilation invariant moment is computed by:

$$\hat{\text{î}}_{pq} = \sum_x \sum_y \left(\frac{x - \bar{x}}{\sigma_x}\right)^p \left(\frac{y - \bar{y}}{\sigma_y}\right)^q,\ (x, y) \in \text{Shape}(o.) \quad (10)$$

To make it scaling invariant, this new central moment must be normalized as below:

$$\text{ç}_{pq} = \frac{\hat{\text{î}}_{pq}}{\text{ì}_{00}^{\gamma}},\ \gamma = \frac{p + q}{2} + 1 \quad (11)$$

Using the ς_{pq} moments, we propose a feature vector $v(o) = (f_1, f_2)$ where:

$$f_1 = (\varsigma_{30} + \varsigma_{12})^2 + (\varsigma_{03} + \varsigma_{21})^2, \tag{12}$$

a Hu moment used in expressing the object's eccentricity, rotation--reflection invariant, and :

$$f_1 = \eta_{24} + \eta_{42} \tag{13}$$

a rotation--reflection invariant moment--based feature too. We have tested both $v(o) = (f_1, f_2)$ and also scalar form $v(o) = f_1 + f_2$ and obtained similar results in classification stage.

After the feature extraction, the next stage of shape pattern recognition is shape classification. The general forms of the VQ k - means supervised algorithm and of the region-growing unsupervised algorithm, proposed in our previous work ([1]), are available for shape clustering. Let us present now the final form of objects recognition approach:

1. Perform the order classification operation computing the n - classes of image I.
2. For each n - class do the following steps:
3. If $n = 1$ then the image's objects are classified by shape in K categories. The results are displayed in Figure 3: 5 classes marked by different colors. Then for each primitive p_i, its class value is associated: $Class\,1(i)$ = the number, between 1 and K, of the class of p_i.
4. If $n = 2$, the objects are detected and classified by shape. For each 2 - order object x, given by $S(x) = \{i_1, i_2\}$, $\text{Shape}(x) = \text{Shape}(p_i) \cup \text{Shape}(p_2)$ is computed and shape recognition is done. The resulted 2 - shapes are represented in Figure 4.

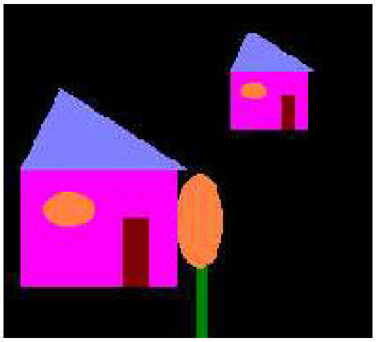

Fig. 3

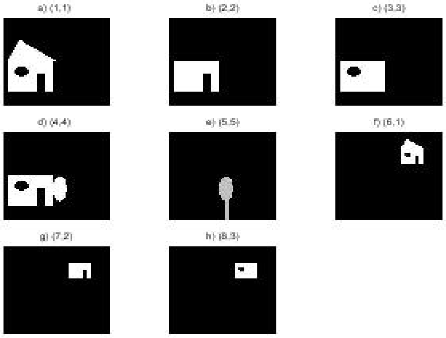

Fig. 4

Each image a) - h), is labeled with a pair (2-order object, shape class). For example, a) (1,1) means: the object 1 belongs to shape class 1. For each shape class of 2-order objects an S - classification is performed. For each x belonging to a shape class and given by $S(x) = \{i_1, i_2\}$, the unordered set $\{\text{Class1}(i_1), \text{Class1}(i_2)\}$ is considered. If no other object from classes processed before x has this set, x is inserted in a new S - class. Otherwise x is associated to the S - class whose objects have the unordered set $\{\text{Class1}(i_1), \text{Class1}(i_2)\}$. For each 2-order object, its S - class value is associated: $\text{Class2}(i)$ = the class number of the 2-order object with number i. In our case the shape classes are no more divided by S - classification, therefore we get similar S - classes, as shown in Figure 5. For each image, a pair (2 - order object, S - class) has been associated. We have the following results:

$$\text{Class2}(1)=1,\ \text{Class2}(2)=2,\ \text{Class2}(3)=3,\ \text{Class2}(4)=4,$$
$$\text{Class2}(5)=5,\ \text{Class2}(6)=1, \text{Class2}(7)=2,\ \text{Class2}(8)=3.$$

5. If $n \geq 3$, the n - order objects are extracted then classified by shape. For each shape class a 2-order objects based classification is done. For each object x of a shape class, its 2-order subobjects are detected. We will consider for it the set of the indices of its 2-order subobjects, $S_2(x) = \{j_1,\ldots,j_t\}, t$ being the number of the 2-order subobjects of x. Then we compute the set of S-classes $\{\mathrm{Class2}(j_1),\ldots,\mathrm{Class2}(j_t)\}$ as a feature vector for x. We perform a classification by this feature vectors, in a similar way with the case $n = 2$, each new vector of classes being inserted in a new cluster. If consider $n = 4, K = 7$ we get 7 shape classes as shown in Figure 6, the first class, {1,8}, being the only one with more than 1 object. Applying the 2-order-objects-based analysis for it, we get $S_2(1) = \{1,2,3\}$ and $S_2(8) = \{6,7,8\}$. The feature set of object 1 will be $\{\mathrm{Class2}(1),\mathrm{Class2}(2),\mathrm{Class2}(3)\}=\{1,2,3\}$. The feature set of object 8 will be $\{\mathrm{Class2}(6),\mathrm{Class2}(7),\mathrm{Class2}(8)\}=\{1,2,3\}$ which means the two objects have the same set and are included in the same cluster (shape class not divided anymore). See final classes, corresponding to the shape classes, in Figure 7.

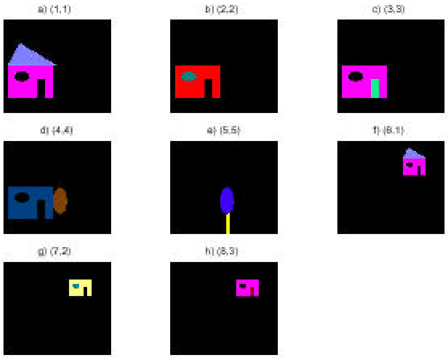

Fig. 5

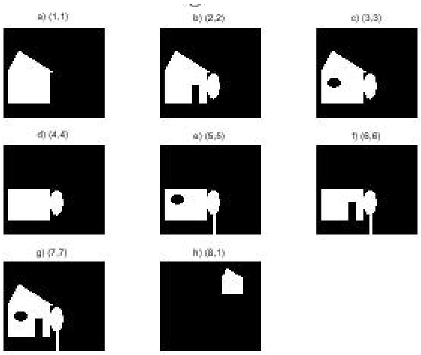

Fig. 6

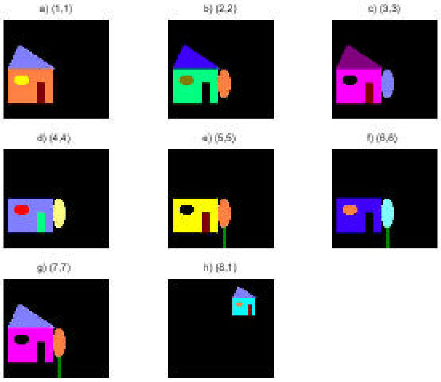

Fig. 7

Applying the 2-order-objects-based analysis for it, we get $S_2(1) = \{1,2,3\}$ and $S_2(8) = \{6,7,8\}$. The feature set of object 1 will be $\{\mathrm{Class2}(1),\mathrm{Class2}(2),\mathrm{Class2}(3)\}=\{1,2,3\}$ The feature set of object 8 will be $\{\mathrm{Class2}(6),\mathrm{Class2}(7),\mathrm{Class2}(8)\}=\{1,2,3\}$ which means the two objects have the same set and are included in the same cluster (shape class not divided anymore). See final classes, corresponding to the shape classes, in Figure 7.

4. AN IMAGE OBJECTS INTERPRETATION APPROACH

As mentioned in introduction, we are specially interested in obtaining the semantic objects. Dividing the n - order objects' set in a number of classes helps us in this task because it is obviously that each class has the following property: all its members are either semantic entities or non-semantic ones. We have to handle an interpretation-based classification task. For an image understanding problem we keep, for each n, only the semantic classes and exclude the meaningless ones. We can associate to each of them a text description of their meaning.

For example, in Figure 7 we distinguish semantic classes and meaningless ones. Therefore we can label the first class with the text "*House*", because it contains two objects representing houses. The 7^{th} class could be labeled with "*House without door and window near a tree*". The 4*th* class, for example, could be considered as having no meaning. In the case of 2-order objects displayed in Figure 5 we could distinguih semantic and non-semantic objects too. For example, to the 5*th* class, containing the 5*th* object, we can associate the description "*Tree*", while the 4*th* object (and class) has no meaning. Of course, the term *semantic* it is a relative one, depending also on what graphical entities are of interest for us. They become semantic objects and the others remain just sintactic ones. We could give the following method for automatic interpretation-based classification: a set of text-labeled image objects representing entities of interest are created first (houses, trees or humans, for example); their orders are then determined; each such labeled object is then compared with the input image's classes of objects having the same order and if it can be classified in such a class, that class will become semantic and it will be labeled with object's description; finally, all labeled classes are merged in a greater semantic class and the unlabeled ones in a non-semantic one, the classification being therefore done. It's obviously that this approach is information retrieval related.

5. CONCLUSIONS

Pattern recognition methods for image objects identification and recognition have been proposed and implemented and graphical results have been presented. As we have already seen, the object recognition results can be used in image understanding domanin. Also they can be succesfully applied in the information retrieval domain, more exactly in image retrieval problems. We can give several examples of tasks that could be solved by our approaches.

Given an image database, a texture retrieval problem can be formulated as follows: search in database and find all the images which contain a given texture *t*. For this task we segment by texture each image and compare *t* with the obtained textures. If we can associate *t* to a texture class of an image *i*, then is chosen. The some problem can be formulated for uniform regions too: find a given uniformity in a database.

More interesting are image objects retrieval problems. Given an *n* - order object as an image query, we have to solve the task of finding it in an image database. The output must be composed of all the images which contain that object. This problem could be formulated as a image understanding task too. Given a text query as input, the database is searched and the images which contain graphical objects having a closed description are retrieved. For solving this task we should extract, classify and text-label the complex objects of each image from analyzed database and compare the obtained labels with the input query.

REFERENCES

1. TUDOR BARBU, *A Pattern Recognition Approach to Image Segmentation*, Proceedings of the Romanian Academy, Series A,, **4**, *2*, 2003.
2. ANIL K. JAIN, *Fundamentals of Digital Image Processing*, Prentice--Hall International, Inc., Englewood Cliffs NJ, 1989.
3. HU, M.K., *Visual pattern recognition by moment invariants*, IRE Transactions on Information Theory, vol. IT-8, February 1962, pp. 179--187.
4. DENGSHENG ZHANG, GUOJUN LU, *Enhanced generic Fourier descriptors for object-based image retrieval*, International Conference on Acoustics, Speech and Signal Processing, 2002
5. RICHARD O. DUDA, *Pattern Recognition for HCI*, Department of Electrical Engineering San Jose State University, 1996-1997

Received March 11, 2003