2011 International Conference on Document Analysis and Recognition

A Contour-based Progressive Technique for Shape Recognition

Stefano Ferilli and Teresa M.A. Basile and Floriana Esposito
Computer Science Dept.
University of Bari
Bari, Italy
Email: lastname@di.uniba.it

Marenglen Biba
Computer Science Dept.
University of New York, Tirana
Tirana, Albania
Email: marenglenbiba@unyt.edu.al

***Abstract*—Information Retrieval in large digital document repositories is at the same time a hard and crucial task. While the primary type of information available in documents is usually text, images play a very important role because they pictorially describe concepts that are dealt with in the document. Unfortunately, the semantic gap separating such a visual content from the underlying meaning is very wide. Additionally image processing techniques are usually very demanding in computational resources. Hence, only recently the area of Content-Based Image Retrieval has gained more attention. In this paper we describe a new technique to identify known objects in a picture based on a comparison of the shapes to known models. The comparison works by progressive approximations to save computational resources, and relies on novel algorithmic and representational solutions to improve preliminary shape extraction.**

***Keywords*-Shape Recognition; Information Retrieval; Document Processing; Digital Libraries;**

I. Introduction

Pictorial information is a precious source of information to understand, index and retrieve documents in a digital library based on their content. Indeed, while much effort was devoted in last decades to extract information about the document content from textual components, more recently significant attention has been paid towards images, as well. Understanding an image does not mean just being able to retrieve images in a database that are pictorially similar to a query image; it also involves recognizing what that image is about, including (or starting from) the objects it contains. Computer Vision deals with the analysis of digital images by computers, in order to discover and understand what is represented therein, and where. Raster images pose the additional problem that no high-level information is available about shapes and other geometrical elements, and each pixel is syntactically (although, clearly, not semantically) unrelated from all the others. In particular, an important sub-field of Computer Vision is Object Recognition (*OR*) [1], having many applications in automation processes. Recognizing an object means being able to distinguish it from a set of other objects. OR techniques usually classify objects based on distinguishing characteristics of the class they belong to, extracted from the image through a sequence of steps. This requires to preliminarily analyze a set of objects of a known class to acquire the most relevant information to be exploited subsequently.

This work aims at developing a method for Object Recognition in raster images that tries to understand an image by looking for known shapes in it, and relies on a combination of existing and novel image processing techniques, as a preliminary step to describe images using higher-level, human-understandable concepts and relationships among them. In particular, this paper will focus on the identification of potential objects in the image, on their representation and storage in suitable data structures and, lastly, on the definition of a suitable matching algorithm that allows to detect known objects in new images. After recalling some background notions and related work in next Section, the proposed technique will be described and evaluated in Sections III and IV, respectively. Lastly, Section V will conclude the paper and outline future work issues.

II. Background and Related Work

Although the techniques and algorithms to perform automatic Object Recognition are very different, depending on the operating environment, they all rely on a common background made up of image processing techniques, and follow a general workflow made up of three steps [2]:

1) Image Processing: a fundamental step that transforms the source image in another image more suitable for running subsequent steps and reaching the objectives;
2) Feature Detection: applies methods aimed at extracting characterizing elements of an image that are more significant than single pixels;
3) Recognition: exploits the features extracted in previous steps to first define classes of objects and then retrieve objects belonging to those classes.

A digital image consists of a set of primitive numeric items (pixels) that in isolation provide little significant information to understand the meaning of the whole picture. Several pixels, taken together, may make up more significant items such as lines, contours, blobs, textures. To be able to extract such a kind of information, often the image must be properly pre-processed using particular *filters*, i.e. functions operating on pixels that enhance some important details and/or dim other, less significant ones, such as the noise

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.151

introduced by the acquisition means or by the representation format (if lossy).

Step 2 consists in identifying and extracting significant information from the pre-processed image resulting from (a combination of) the aforementioned techniques. The information obtained in this way allows for a higher-level interpretation of the image. Depending on the kind of features to be extracted, several techniques are available, and often specific features are exploited for particular objectives. Features can be distinguished according to their morphology:

Keypoint. Provide point-level information, that is robust to occlusions and scale invariance (e.g., [3]);

Edges. Concern the contour lines of objects in images, usually corresponding to zones where a change of color, intensity or texture occurs [4];

Region segments. A segmented region is made up of a set of pixels that 'go together' according to some logic (e.g., being part of a same object).

Each element identified in the image can be compared to previously stored models in order to check possible correspondences. This is done by different algorithms, considering different kinds of information. Limitations in applying Computer Vision systems come from the difficulty in extracting information from images. For an OR system to be effective and flexible, several properties are desirable. Here, we focus on the following ones, deemed as very important [5]:

- Scale invariance.
- Translation invariance (the position of the object to be recognized cannot be assumed to be fixed in the acquired image).
- Robustness to change in intrinsic variables of the image (even in controlled environments, small changes in color, luminance or contrast can take place).
- Rotation invariance. Unfortunately, rotating a 3D object usually results in completely different shapes depending on the perspective; nevertheless, making the system robust at least to 2D rotation already ensures a noteworthy degree of reliability.
- Efficiency (usually opposite to effectiveness).

III. Object Recognition Technique

The object recognition technique we propose works in different steps on the input image. A graphical summarization of the various steps, applied to the original image in the top-left, is provided in Figure 1.

A. Pre-processing

We would like to blur the image within objects, so that they can be considered as single blobs by the segmentation step, but without blurring (and possibly even sharpening) their contours also, otherwise the resulting shape would be meaningless. Any blurring and edge-enhancing technique might be used in this step (of course, the outcome would

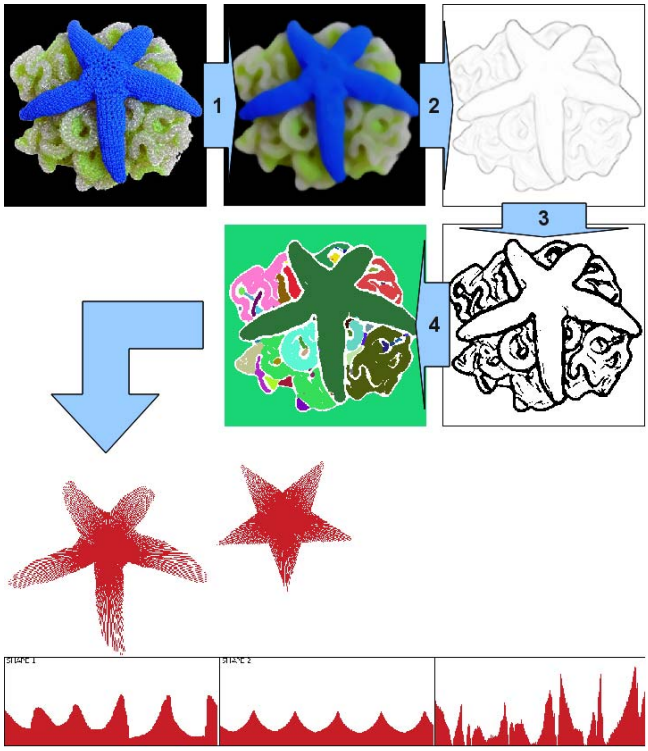

Figure 1. Processing steps on a sample image

be different). The experiments in this paper were obtained using two novel filters (due to space limitations, and to the focus of this paper on the overall strategy, here we must drop their detailed specification) we purposely developed for reaching the above objective (see the top row of Figure 1):

1) **selectiveBlur**: to smooth the image while preserving the contours; the output pixel is assigned a color determined by a weighted average based on color similarity of a selection of the 8-neighbors.
2) **extractContour**: returns a gray-scale image with enhanced object contours (darker zones correspond to sharper contours in the original image). Basically, a pixel is considered as belonging to a contour if it is placed in a point where the image color changes: the more the change, the more the importance of that pixel as a contour one.

Differently from the previous step, the image segmentation step (second row from the top in Figure 1) exploits two standard techniques to find candidate objects:

3) **binarization**: standard tresholding (247 was empirically found to be an effective threshold on average) on the outcome of extractContour.
4) **region growing**: by contraposition, the blobs surrounded by such contour areas are considered as candidate objects in the image, and are determined by filling the white areas.

B. Feature Extraction / Representation

Given a blob, the associated *shape* is a more refined description ready to be compared to the available models (expressed in the form of shapes, as well). In particular, we focus on the blob border, that was found to be a very indicative feature for object recognition [6]. Specifically, Fourier descriptors based on distance from centroid of the contour points of the shape proved to be very effective. Thus, we adopted this indicator, but embedded in a novel approach. Indeed, we do not consider the distance from the centroid for all contour pixels, but just of those intersecting selected radian lines at pre-defined angles, originated in the shape centroid.

A first, crucial issue is determining the number of samples to be taken, in order to have a sufficiently accurate representation without burdening the system more than needed. Clearly, the proper tradeoff also depends on the size of the database of models to be matched, and on the kind of objects the system is intended to handle. Next subsections will explain how and why we set such a parameter. Another question is how to represent the single sampled values. We obtain scale invariance by normalizing all the sampled values of a shape to the largest sampled value in that shape. Moreover, we empirically found that a scale of 256 integer values provides a sensible tradeoff between sufficient accuracy and tolerance to noise in the blob contours (requiring just a single byte).

Summing up, a shape is described by a histogram of n sampled values, each normalized to the integer interval $[0 \dots 255]$, taken at equally spaced angles from the positive X axis in a coordinate system centered in the blob centroid. The bottom-left of Figure 1 shows a graphical representation of two shapes (one extracted from an image, and a model shape) using both radians (above) and the corresponding 'unrolled' histogram (below). This choice ensures invariance with respect to translation (no information on spacial placement is stored), scale (that does not affect the data structure, but just the values it contains), and intrinsic variables of the images such as luminance and color (completely ignored by the representation, although more refined techniques are to be included in future work). It is also robust to 2D-rotation (by rotating the histogram) and mirroring (by mirroring the histogram).

C. Shape Matching

Although basing object recognition on the shape only is clearly limiting, because it represents just a part of the whole matter, especially in 3D images, nevertheless a method that is invariant to translation, scale, 2D rotation and color is often sufficient.

Now, once the information about the candidate shapes in an image has been extracted, provided a base of sample relevant shapes of interest ('models') is available, the extracted shapes can be compared to those models for possible matching. The expected outcome of the matching is a similarity/distance value among the two compared elements. We compare their histograms, representing the distance from the centroid of the blob border in each of the radian directions, according to the intuition that, the more deformed is an object with respect to the model, the more different they are. Specifically, we proceed by overlapping them and summing the absolute pairwise differences of corresponding bars to obtain the overall evaluation (in this case, representing a distance). Another option might be using the statistical measure of variance, but since in our case both the number of values and the values are normalized, a simple summation provides the same results with much less effort. Moreover, for rotation invariance, one such comparison for each displacement of the histogram to be classified over that of the model (considering the histograms as if the last bar were immediately followed by the first one) is needed, displacing each time the histogram by 1 degree to the left, for a total of comparisons equal to the number of bars considered, and then the best case (i.e., the minimum distance value) is taken. The outcome is shown in the bottom-right of Figure 1, where the model shape has been rotated to the best-matching position, and the rightmost histogram shows the pairwise differences among the bars of the shape and model histograms on the left for such an alignment. Overall, if there are n bars to be compared, the effort consists of $c = n \cdot n$ comparisons (subtractions). For mirroring-independence one must double the effort, repeating the above procedure and proceeding in the opposite directions when rotating the histogram (from left to right in one case, and from right to left in the other).

Figure 2 shows the sensitivity of the proposed technique to different geometrical transformations for a sample image (left shape) and corresponding modifications (right shape). For each comparison, the best-matching alignment of histograms is shown, along with the corresponding difference histogram (right-most histograms). Invariance to translation trivially holds. Invariance to rotation (top case) is proved, since the difference between the shapes is so close to zero that the bars are not visible in the difference histogram. As to scaling (middle case), the difference is visible, but nevertheless small. Also changing the image colors, in this case by considering the negative of the image (bottom case) has a slight effect on the comparison, due to the different outcome of the segmentation step.

D. Progressive Approach

Since the matching effort is quadratic in the number of bars to be compared, the basic version of the technique described above might turn out to be inefficient as long as the database size grows. Our solution to tackle this problem consists in a progressive matching procedure, that starts with a few comparisons, and repeatedly selects the most similar models only to carry on to a next matching step including

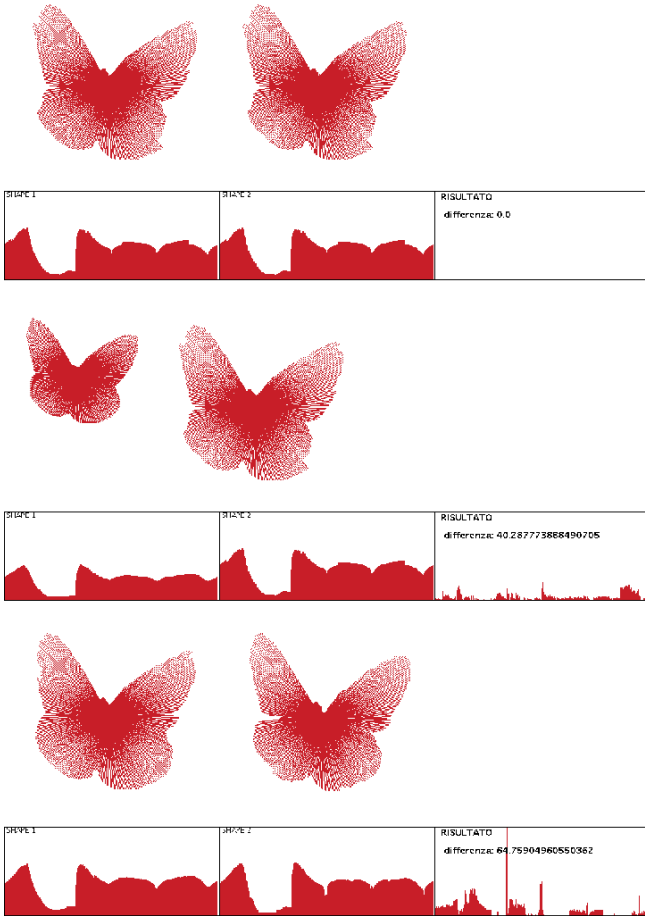

Figure 2. Check of invariance on a sample image

more comparisons, until a single model neatly wins or the maximum number of comparisons has been reached.

A simple and straightforward way for increasing the number of comparisons at each matching stage is doubling it, which would make more comfortable the use of powers of 2. In fact, the binary system for angle measurement divides the round angle into 256 degrees, called *brads* (from Binary RADianS). Thus, a straight angle consists of 64 brads, and angles can be comfortably represented using a single byte (more in general, an integer number of bytes — but in our case 2 bytes would be already too much).

The first stage in the matching algorithm compares just 16 values (less comparisons would be too limited to provide a sensible indication on the actual shape), sampled at $16 \cdot i$ brads ($i = 0, \ldots, 15$) along the raw shape, to the 256 values representing a model, for a total of just $16 \cdot 256 = 4096$ comparisons for each shape in the database. Due to the doubled sampling frequency technique, the samples considered at each next step are a superset of those in the previous one, and hence the number of new comparisons per shape is, respectively, 4096, 8192, 16384 and 32768 in the last step.

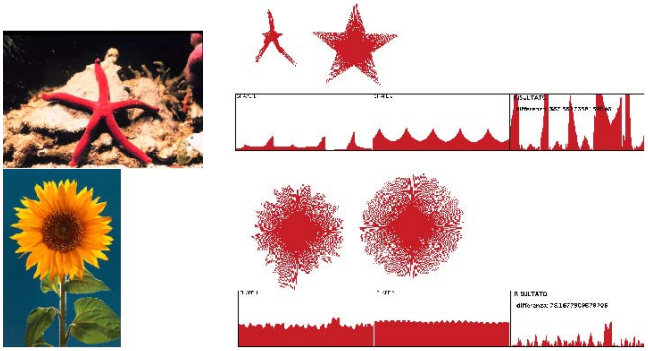

Figure 3. Sample comparisons

IV. Evaluation

Let us start with a quick evaluation of the effectiveness of the proposed approach, by comparing noteworty pairs of objects, or objects taken from real pictures to artificial models stored in a database. For this evaluation, we exploited a small database of shapes, and ran the system to recognize shapes in new pictures (not exploited to build the database). The system results are sensible even when dealing with very low-quality contours (as for the top case in Figure 3). It is also noteworthy the successful recognition of a sunflower (bottom of Figure 3), despite of the different number and orientation of the petals.

As to efficiency, interestingly the system never required to run the last step (256 comparisons), but always returned a solution with at most 128 comparisons. Often 16 comparisons only were sufficient to identify the correct shape, and in almost all cases of shapes not included in the database it actually returned no classification. Let us show the performance evolution under different parameters, referring to a PC endowed with a Dual Core processor at 2GHz and running Windows Vista.

First, we compared a single shape to 16 models chosen at random from the database. The progressive technique required 3 steps only (up to 64 comparisons) to find a neat winner model for classification (top graphic in Figure 4, where the lines report the times for matching the shape to each model at each step). Less than 5 msec were taken for each matching in the first stage, up to less than 30 msec for the third stage. Carrying on each time to the next step all models whose similarity exceeds the average similarity among the models in the previous step, 5 shapes are discarded in the first step, and 3 more in the second one; then, among the 8 survivors, the third step determines the winner. The bottom graphic in Figure 4 reports the surviving models at each step for increasing size (16, 25, 35 and 50) of the database. The larger the database, the more shapes are cut off at each step; for databases including 35 and 50 models one more step (128 comparisons) is needed.

Lastly, we turn to evaluate the effort required to process

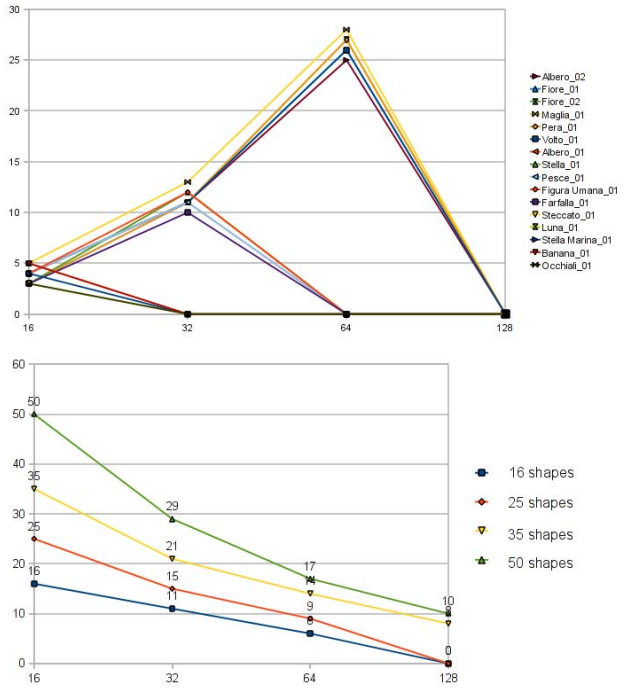

Figure 4. Matching performance for a single shape

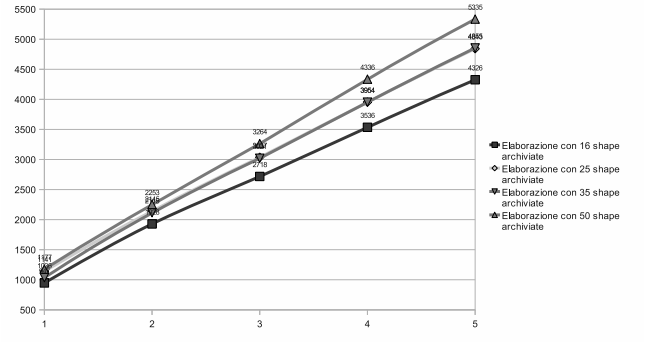

Figure 5. Matching performance for a 5-shape image and 4 databases of different size

a whole picture, with all the shapes it includes, against databases of different (increasingly larger) size. The detailed figures obtained when matching the previously mentioned 5-shape sample image against databases including 16, 25, 35 and 50 models, respectively, are graphically summarized in Figure 5. As expected, the time needed to process an entire picture, with all the shapes it includes, is linear in the number of shapes to be processed (taking the lowest line in Figure 5, nearly 4 secs were needed to process 5 shapes in the sample picture considered). Also the size of the database seems to marginally affect the effort: in the considered cases, the difference among a database including 50 models and one made up of just 16 models is about 1 sec, and, interestingly, the time for the databases sized 25 and 35 is in fact overlapping.

V. Conclusion

Information expressed by images can be hardly accessed, due to the *semantic gap* separating the raw set of pixels from their overall perceptual meaning. Nevertheless, images are very information-dense elements, and hence being able to understand their content would help to support several automatic tasks on documents. This work specifically focuses on Object Recognition, as a fundamental task towards a high-level description of the image content in terms of the objects contained and their inter-relationships. A progressive technique is proposed, that integrates and improves a set of existing representation and processing techniques for identifying objects belonging to known classes for which model shapes are available. A prototype implementation of the proposed approach suggests that effective recognition can take place, with reasonable efficiency in terms of time and space resources. It can recognize objects based on their shape, independently of scaling, translation, mirroring and (2D) rotation.

Future work will concern finding a mix of features that are sufficiently complementary to significantly improve recognition performance over application of shape recognition alone, while not increasing excessively the computational burden. Moreover, we are working on devising strategies for exploitation of the high-level description provided by this technique, both for document understanding and indexing. Other directions for investigation concern the improvement of the pre-processing step, for providing a better input to the recognition engine.

References

[1] H. Hogendoorn, "The state of the art in visual object recognition," 2006. [Online]. Available: http://www.ufonds.uu.nl/tmp/content/314/10565597684929.pdf

[2] R. Szeliski, *Computer Vision: Algorithms and Applications*. Springer, 2011.

[3] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[4] M. Heath, S. Sarkar, T. Sanocki, and K. Bowyer, "Robust visual method for assessing the relative performance of edge-detection algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 1338–1359, 1997.

[5] R. Brause, B. Arlt, and E. Tratar, "Project semacode: A scale-invariant object recognition system for content-based queries in images databases," Johann Wolfgang Goethe University, Computer Science Dept., Frankfurt/Main, Tech. Rep. 11/99 (FB20), 1999.

[6] D. Zhang and G. Lu, "A comparative study of curvature scale space and fourier descriptors," *Journal of Visual Communication and Image Representation*, vol. 14, no. 1, pp. 41–60, 2003.