Contour-Based Progressive Identification of Known Shapes in Images

Stefano Ferilli[1], Floriana Esposito[1], Domenico Grieco[1], and Marenglen Biba[2]

[1] Dipartimento di Informatica, LACAM Laboratory
Università degli Studi di Bari "Aldo Moro"
{stefano.ferilli,floriana.esposito}@uniba.it
[2] Computer Science Department
University of New York, Tirana (Albania)
Marenglenbiba@unyt.edu.al

Abstract. Information Retrieval in digital libraries is at the same time a hard task and a crucial issue. While the primary type of information available in digital 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, and 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. It is based on shape contours, and works by progressive approximations to save computational resources and to improve preliminary shape extraction. Small (controlled) and more extensive experiments are illustrated, yielding interesting results.

Keywords: Shape Recognition, Information Retrieval, Document Processing, Digital Libraries.

1 Introduction

Graphical components are a precious source of information to understand, index and retrieve documents in a digital library based on their content. Indeed, the power of modern technology allowed to efficiently and effectively store documents that are not just made up of text, but include (often a significant amount of) pictorial content whose relevance to the document cannot be ignored. Accordingly, while much effort was devoted in the past decades to information extraction from textual components, more recently significant attention has been paid towards images, as well.

Computer Vision deals with the analysis of digital images by computers, in order to discover and understand what is represented therein, and where. While vectorial images explicitly represent shapes and other geometrical elements, raster images pose the additional problem that no high-level information is available therein, and each pixel is syntactically (although, clearly, not semantically) unrelated from all the others. An important sub-field of Computer

T. Catarci, N. Ferro, and A. Poggi (Eds.): IRCDL 2013, CCIS 385, pp. 17–28, 2014.

Vision is Object Recognition (*OR*) [5], that has 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 pre-processing steps. This requires to preliminarily analyze a set of objects of a known class to acquire the most relevant information to be subsequently exploited. However, understanding an image does not mean just being able to retrieve other images in a database that are pictorially similar to it; it also involves recognizing what that image is about, including (or starting from) the objects it contains. Content-Based Image Retrieval (CBIR) [2] focuses on image content, rather than on their overall features.

This work concerns Object Recognition aimed at understanding raster images by looking for known shapes in them. It proposes 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. Such descriptions might be exploited as metadata to be added to the documents where the image appears, or be input to standard text processing techniques in order to index the documents based also on their pictorial content, or be fed to relational Machine Learning systems to infer models of image classes when the depicted information is too complex for standard propositional and statistical techniques. Although a full semantic understanding of the image meaning is still to come, this might nevertheless bring many advantages, among which the possibility of retrieving and relating documents in a digital library according to the images they contain, and providing explicitly otherwise implicit and latent information. This will also allow to perform queries using visual information such as images in addition to standard textual search techniques (e.g., by providing a sample image expressing the concept to be searched for).

The focus of this paper is on the overall technique and on its performance, rather than on the details of its single steps. In particular, here we present novel experimental results on the technique originally presented in [4]. After recalling some background notions and related work in next Section, the proposed technique will be described and evaluated in Sections 3 and 4, respectively. Lastly, Section 5 will conclude the paper and outline future work issues.

2 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 consisting of three steps [7]:

1. Image Processing: 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.

Concerning step 1, a raster digital image consists of a set of primitive numeric items (pixels) that in isolation provide little significant information, just like a single element of a puzzle does not allow 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 introduced by the acquisition means or by the representation format (if lossy).

Step 2 identifies and extracts 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.

As regards step 3, 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 Object Recognition system to be effective and flexible, several properties are desirable. Here, we focus on the following ones, deemed as very important [1]:

- 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 in contrast to effectiveness).

While several proposals are present in the literature to face these problems (e.g., [6]), here we refer to the technique described in [4]. In that work, the identification of potential objects in the image, their representation and storage in suitable data structures and a corresponding matching algorithm that allows to detect known objects in new images were first introduced and described.

3 Object Recognition Technique

The object recognition technique proposed in [4] aims at identifying regions of an image that correspond to objects, and at recognizing the class of these objects

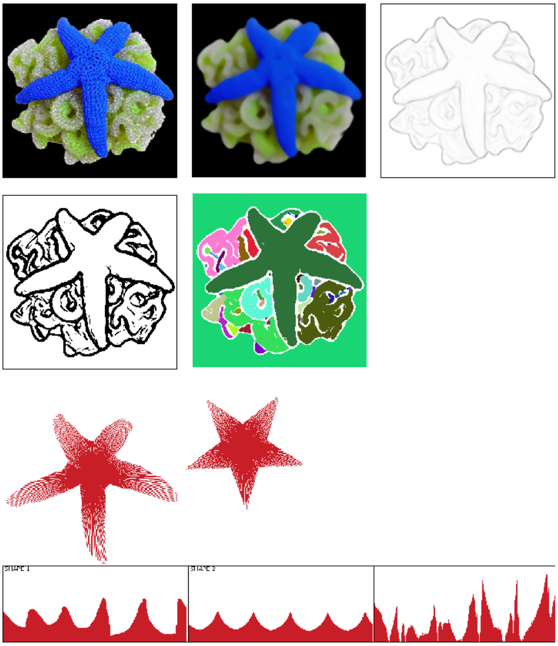

Fig. 1. Processing steps on a sample image

in a simple and effective way. While regions are determined on the grounds of color homogeneity, class recognition exploits the region contours. A graphical summarization of the various steps, applied by this technique to the original image in the top-left, is provided in Figure 1, while next subsections provide a description and discussion of these steps.

3.1 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. Then, we need to extract the single blobs by preserving their contour peculiarities. Although any blurring and edge detection technique available in the literature can be used in this step, we purposely developed two filters that better reach these objectives, whose results are shown in the top row of Figure 1.

The image segmentation step, shown in the second row from the top in Figure 1, aims at identifying the candidate objects represented in the image. In this case we used standard binarization and region growing. As to the former, we used tresholding (247 was empirically found to be an effective threshold on average). The blobs surrounded by the resulting contour areas, determined by filling the white areas by the region growing algorithm, are considered as candidate objects in the image. The allowed length of this paper does not allow us to provide further details on our pre-processing algorithms. However, also other algorithms can be used interchangeably.

3.2 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). Blob border was found to be a very indicative feature for object recognition [8]. Specifically, Fourier descriptors based on distance from the centroid of the shape to its contour points proved to be very effective. Thus, we were inspired by this indicator to set up our approach. Differently from Fourier descriptors, however, we consider the distance from the centroid for contour pixels, and pick just those intersecting selected radian lines at pre-defined angles, originated in the shape centroid. More precisely, a shape is described by a histogram of n sampled values, each normalized to the integer interval $[0 \dots k]$, 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 using both radians (above) and the corresponding 'unrolled' histogram (below).

A first question is how to choose the representation range to represent the single sampled values. Normalizing all the sampled values of a shape to a fixed k ensures scale invariance. We empirically found that a scale of 256 integer values provides a sensible tradeoff between accuracy and tolerance to noise in the blob contours (while requiring just a single byte in memory). So, we associate the centroid to value 0 and the largest sample in a shape to value $k = 255$. Another crucial decision concerns the number n of samples to be taken, in order to have a sufficiently accurate representation without excessively burdening the system. 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 we set such a parameter, and why. This representation 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).

3.3 Shape Matching

Once the information about the candidate shapes in an image has been extracted, provided a database of sample relevant shapes of interest ('models') is available, the extracted shapes can be compared to those models for possible matching in order to determine the shape class it belongs to. In the bottom row of Figure 1, the left part refers to a query shape extracted from the image, while the center refers to a model shape in the database. Note that each class may be associated to many sample models in the database, and that the matching is performed against the models, not the classes. So, even if the number of classes is kept constant, increasing the number of models in the database also increases the recognition effort. The expected outcome of the matching between a query shape and a model 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 (rightmost 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.

3.4 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 be carried on to a next matching step including more comparisons, until a single model neatly wins or the maximum number of comparisons has been reached.

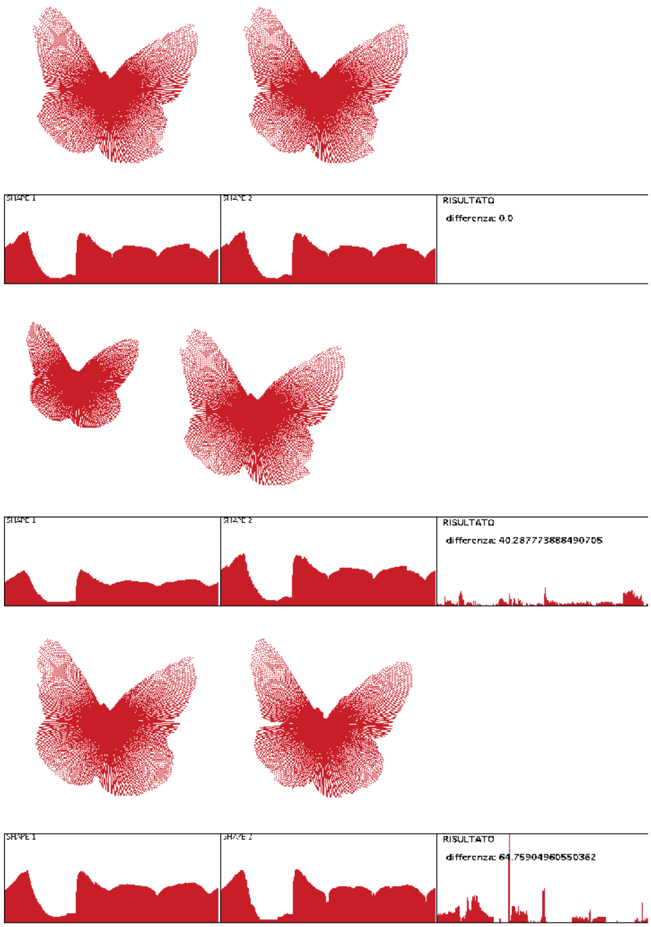

Fig. 2. Check of invariance on a sample image

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. an angle displacement of 1.40625 degrees between consecutive radians). 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,

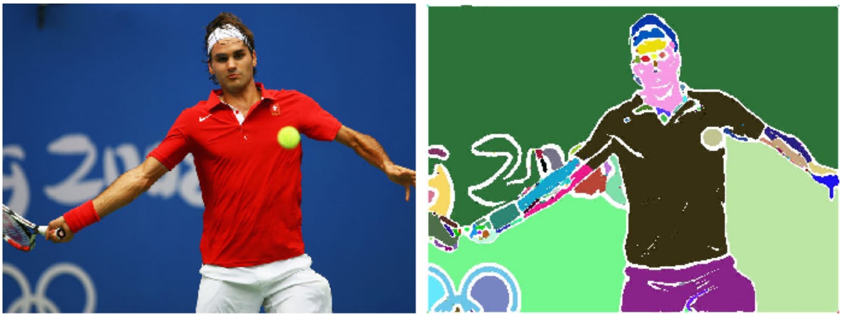

Fig. 3. A picture involving a combination of several shapes

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.

3.5 Prospects

The final aim of this kind of processing, that we intend to develop as future work, is to be able to understand the content of a whole picture based on the shapes recognized in it and to the particular spatial relationships existing among them. E.g., the picture in Figure 3 might be classified as ‘sports’ or ‘tennis’ due to the presence of shirt, shorts, face, ball and racket shapes, where the face is just above the shirt that, in turn, is just above the shorts. To do this, First-Order Logic descriptions are needed, to be able to express relationships among shapes. In the above example, the description might run as:

contains(p,f), face(f), contains(p,t),shirt(t), contains(p,s), shorts(s),
contains(p,b), ball(b), above(f,t), above(t,s), above(f,b), above(b,s), overlap(b,t)

The classification rules might even be learned using ILP systems. In particular, due to the incremental behavior of the shape-learning technique, an incremental ILP technique should be exploited as well, such as those proposed in [3].

4 Evaluation

The preliminary controlled experiments reported in [4] show that the proposed technique is a viable solution, that can efficiently and effectively recognize known shapes in new images. The reported performance refers to a PC endowed with a Dual Core processor at 2GHz and running Windows Vista. A database of 50 selected model shapes was set up, and used to answer pictorial queries consisting of new images (not used to build the database).

In one experiment, the query image contained a single (unknown) shape to be compared against a subset of 16 models chosen at random from the database:

tree (2 models), flower (2 models), shirt, pear, face, star, fish, man, butterfly, fence, moon, sea star, banana, glasses. 3 steps (up to 64 comparisons) were needed before finding a neat winner model for classification. Less than 5 msec were taken for each matching in the first stage, no more than 30 msec for the third stage. Each step discarded all surviving models whose similarity was below the average similarity among the models in the previous step: 5 shapes were discarded in the first step, 3 more in the second one, and finally 7 in the third, which determined the winner.

Table 1. Matching performance for a single shape

Initial models	step 1	step 2	step 3	step 4
16	5/11	3/8	-	-
25	10/15	6/9	-	-
35	14/21	7/14	6/8	-
50	21/29	12/17	7/10	-
Comparisons	16	32	64	128

Then, in the second experiment the subset of models was increased from 16 to 25, then to 35 and finally to 50 shapes. The number of filtered/surviving models at each step for each size is reported in Table 1, showing that the larger the database, the more shapes are cut off at each step. While for the 16- and 25-shape sizes 64 comparisons were sufficient to complete recognition, for databases including 35 and 50 models one more step (128 comparison) is needed. Interestingly, the system never required to run the last step (256 comparisons).

The third experiment evaluated the effort required to process pictures involving several shapes. Specifically, a picture including 5 shapes was used as a query against the same subsets of the model shape database, of increasingly larger size, as in the second experiment. As expected, the time needed to process the whole picture is linear in the number of shapes to be processed (taking about 1 sec per shape). Much more interesting is the fact that the size of the database seems to very marginally affect the effort: the gap in shape recognition time between the 16 models case and the 50 models case is just 0.2 sec, and the time curves for the databases sized 25 and 35 in fact overlap.

Now, we present a novel experiment aimed at assessing the convergence performance of the algorithm for several kinds of shapes having different peculiarities. We considered a set of heterogeneous images including complex shapes in different contexts, positions and orientations, and image composition. The dataset included 250 images of 15 different shapes, as summarized in Table 2 (first and second column). It was collected from various repositories on the Internet[1]. Sample images for each shape type are shown in Figure 4. Specifically, a separate experiment was run for each shape class, and an incremental approach was adopted for each experiment. The models database was initialized with all

[1] The shape dataset can be downloaded at `http://lacam.di.uniba.it/~ferilli/ufficiale/res/shapes_dataset.zip`.

Table 2. Dataset composition

Category	Images	Recognized	Last failure
Red Cross images in various scenarios	35	23 (68%)	35
Human hands	20	13 (68%)	13
Fighter aircrafts	30	18 (62%)	16
Simple geometric shapes in various orientations	10	9 (100%)	–
Trifoil	10	9 (100%)	–
Bat shape	20	17 (89%)	3
Ferrari horse	10	9 (100%)	–
W letter	20	18 (95%)	3
S Letter	20	19 (100%)	–
Machine gun, AK-47	10	7 (78%)	4
Mozart	15	13 (93%)	7
Hen	10	9 (100%)	–
House	20	15 (83%)	5
Key	10	8 (89%)	2
Italy Map	10	9 (100%)	–
Average		88%	

the images in the other classes; then the available images for the class under consideration were progressively submitted to the system, and whenever an image was not correctly recognized, it was added to the database before submitting the next images. We disabled mirroring tolerance in this experiment, to check whether and how it affects recognition performance.

As expected, the recognition performance improves with the growing number of images (i.e., models) in the database (remember that many models may be associated to one class): as more models are available in the repository, the number of recognized images grows constantly. Table 2 shows in the third column the number and percentage of recognized images (denoting accuracy), and in the fourth column the number of the last non-recognized image that causes an addition to the models database (denoting how quick the convergence is). The figures are computed ignoring the first image, that (being a shape not yet present in the repository) is obviously not recognized. In most cases, the initial performance depends on the images already added to the database, which explains why at the beginning some images are not recognized. Then gradually, by enriching the database incrementally, the sequence of images that are correctly recognized grows steadily. Of course, even after many images have been added to the database, there may be cases that are not correctly recognized and this is due to the peculiarities of these images. For instance, in the 'Red Cross' class the last image is not recognized. However, in most cases normal shapes already inserted in the database are correctly identified.

As expected, the most problematic cases are those in which the target object is taken in different perspectives and contexts, which results in significant changes in their possible shapes. Indeed, for the 'Red Cross', 'Hands' and 'Aircrafts' subsets, looking at the specific images not recognized one can see that they

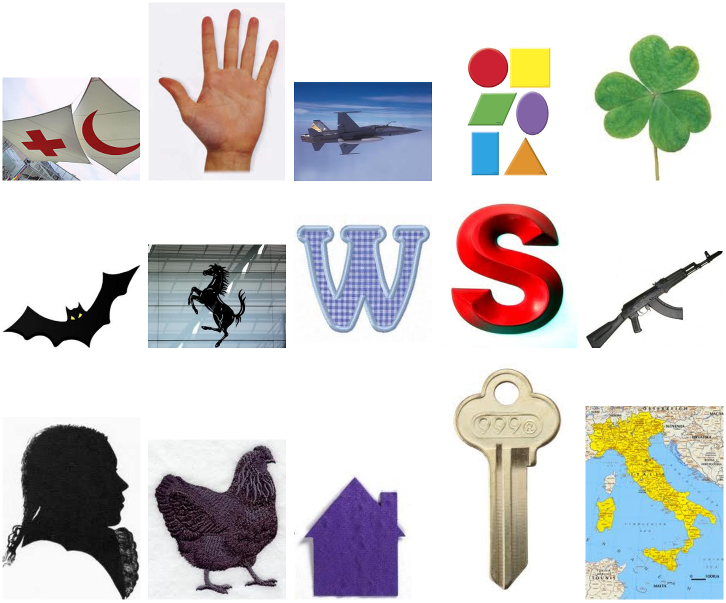

Fig. 4. Sample shapes from the test dataset

differ from the already inserted images mainly in their orientation and in the configuration of the background. Conversely, for more standard shapes, such as ‘Geometric shapes’, ‘Trifoil’, ‘Ferrari’, letter S and Italy, the addition of the very first shape is sufficient to completely and correctly learn its model. Also, the good performance on letters might indicate that the proposed technique can be a valuable tool for recognizing printed symbols in general. This raises an additional question, concerning how many images the technique needs in the database in order to have a stable performance. Differently from the cases just discussed, where just one tagged image is sufficient for the technique to subsequently recognize that shape in all future images, for some other shapes it takes more images to make the technique start recognizing shapes steadily. From our observations we found that, in most cases, it is necessary and sufficient to insert in the database images that represent distinguishing features of the shape orientation or direction. For example, for the Mozart silhouette, the two images that were added to the model database are those depicting Mozart in left and right orientation, respectively. After the latter was added to the database, the system recognized successfully all the following images that represent both left and right orientation. This confirms that the shape orientation is important in order for the system to behave correctly, and that in some cases mirroring tolerance is necessary. It is a topic of future work assessing the tradeoff between accuracy and computational cost due to the introduction of mirroring tolerance.

5 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 improve image indexing and retrieval in digital libraries. 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, a larger evaluation on benchmark datasets and an assessment of what kinds of images can profitably make use of this technique.

References

1. Brause, R., Arlt, B., Tratar, E.: Project semacode: A scale-invariant object recognition system for content-based queries in images databases. Technical Report 11/99 (FB20), Johann Wolfgang Goethe University, Computer Science Dept., Frankfurt/Main (1999)
2. Chen, Y., Li, J., Wang, J.Z.: Machine Learning and Statistical Modeling Approaches to Image Retrieval. Information Retrieval, vol. 14. Kluwer (2004)
3. Ferilli, S., Basile, T.M.A., Biba, M., Di Mauro, N., Esposito, F.: A general similarity framework for horn clause logic. Fundamenta Informaticae 90, 43–66 (2009)
4. Ferilli, S., Basile, T.M.A., Esposito, F., Biba, M.: A contour-based progressive technique for shape recognition. In: Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR 2011), vol. 1, pp. 723–727. IEEE Computer Society (2011)
5. Hogendoorn, H.: The state of the art in visual object recognition (2006)
6. Shu, X., Wu, X.-J.: A novel contour descriptor for 2d shape matching and its application to image retrieval. Image and Vision Computing 29(4), 286–294 (2011)
7. Szeliski, R.: Computer Vision: Algorithms and Applications. Springer (2011)
8. Zhang, D., Lu, G.: A comparative study of curvature scale space and fourier descriptors. Journal of Visual Communication and Image Representation 14(1), 41–60 (2003)