# Robust Symbol Recognition using a Structural Approach

Mathieu Delalandre
CVC (Barcelona, Spain)
mathieu@cvc.uab.es

Simon Bernard
LITIS (Rouen, France)
simon.bernard@univ-rouen.fr

Jean-Marc Ogier
L3i (La Rochelle, France)
jean-marc.ogier@univ-lr.fr

Karell Bertet
L3i (La Rochelle, France)
karell.bertet@univ-lr.fr

Mickael Coustaty
L3i (La Rochelle, France)
mickael.coustaty@univ-lr.fr

## Abstract

*In this paper we present a robust system of symbol recognition using a structural approach. Our key objective here is to provide a system, equaling the statistical ones in robustness concerning the recognition, to apply next to localization. To do it we have investigated two particular structural methods: the straight line detection using Hough Transform and the vector templates matching. Experiments done on the GREC2003 database show how their combination allows to obtain high recognition results.*

## 1 Introduction

Symbol recognition is a particular application of pattern recognition. It aims to localize/recognize symbols within document images in regard to specific application domains (architectural, electrical, etc.). It was an active topic in the field of graphics recognition during the years 90's. In the last ten years, there has been a noticeable shift of attention towards the problem of performance evaluation of symbol recognition systems [2]. This has resulted in the organization of several international contests, held during the conferences ICPR 2000 and GREC 2003, 2005 and 2007. However, all these contests have been focussed on recognition of isolated symbols. They have not considered the problem of symbol localization in real documents, composed of multiple objects constrained by spatial relations. The main reason for that is the few of ready-to-use systems working in localization. In addition, the performance evaluation task is made harder. It is difficult to obtain groundtruth of whole documents, evaluation metrics must be also reformulated to evaluate localization/recognition in complete documents.

This particular situation results today in a mixed result. In one hand, the work done in performance evaluation has highlighted robust methods of isolated symbol recognition. In the other hand, none of the proposed methods can deal with the localization of symbols on real documents. Indeed, to gain robustness the authors have focussed their systems on statistical approaches. Feature vectors are computed globally to images, and next compared using statistical classification technics. Thus, to locate symbols a filtering must applied to whole document images. Such a filtering requires large computation times, despite the possible use of heuristics to limit the areas to explore. In addition large amounts of data are used to train these systems. Such experimental conditions can't be obtained with a real recognition case.

The system of [10], winer of the GREC2005 contest, is an example of that. It uses 2D kernel densities to represent symbols, computed from sample points issued of a skeleton. The comparison between two symbol is achieved using the Kullback-Leibler divergence as similarity measure. During this contest, provided databases were composed respectively of 50% for training and testing.

The open question today is how to define robust symbol recognition systems applicable to localization. The key point to do it are the structural methods. They are based on the relational organization of low-level primitives (straight-lines, arcs, etc.) into higher-level structures (graphs, signatures, etc.). Systems could rely on this primitives decomposition, to locate and recognize symbols at the same time, using closed-loop methodologies. In addition, structural matching methods require usually a few of data for training, at minimum ideal models of symbols. However, these methods are well-know for their noise sensitivity.

In this paper we propose a new structural system for symbol recognition. Our key objective here is to provide a system equaling the statistical ones in robustness concerning the recognition, to apply next to localization. Localization and recognition are both huge issues, in this paper we will investigate the structural recognition part. We will give in perspectives some considerations to use of this method for localization. To make robust our system we have investigated two particular structural methods for symbol recognition: straight line detection using Hough Transform, and vector template matching. In the rest of the paper we will present both of these methods in sections 2 and 3. In section 4, we will present results and experiments done with our system on the GREC2003 database. Finally, section 5 will give our conclusions and perspectives about this work.

## 2 Straight line detection

First step of any structural recognition system, is to extract graphical primitives (straight-lines, arcs, etc.) from images. During the last twenty years, different types of extraction methods have been proposed in literature including skeletonization, contouring, line tracking, etc. Among them, the ones based on the Hough Transform (HT) present good properties of robustness. It has been applied with success in the past to line drawing documents [6].

The HT is a well-known method introduced in years 60's by P.V. Hough. Its key idea is to project pixels of a given image into an parametric space $(\rho, \theta)$, where shapes can be represented in a compact way Fig. 1. The classical Hough transform was concerned with the identification of straight lines. It has been extended next to other shapes like circles or ellipses. In our system we have limited our detection to straight-lines. Usual process consists of three main steps:

1. **Point selection:** Points of interest are selected first using some pre-processings, allowing to reduce processing times of next steps. These points must characterize properly the shapes to detect, in order to not loose relevant information.

2. **Point accumulation and peak detection:** Points of interest previously selected are mapped to the Hough space. An accumulator array is employed to record the number of sine curves going trough a given point. Peak detection consists in the identification of points in the accumulator, for which the number of sine curves is important enough. Several methods have been proposed in the literature to accelerate the accumulation and to make more accurate the peaks detection (randomized HT, probabilistic HT, etc.).

3. **Segment detection:** The detected lines in Hough space are mapped with the document image, in order to detect the begin-end points composing segments. Areas of lines are scanned, in order to check the connectivity of neighboring pixels along. Details about the used methods to achieve this step, and their comparison, are seldom addressed in the literature.

Among the existing methods, we have made the following choices for our system. Our point selection results of a

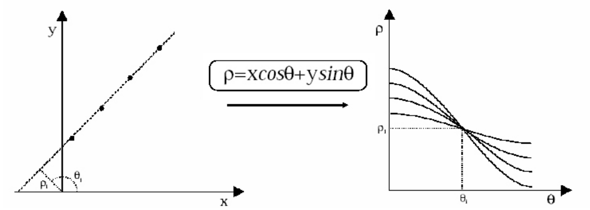



**Figure 1. Hough Transform**

skeletonization process. We have employed the algorithm of [1], based on a medial axis transform. Such a skeletonization approach is known to be less sensitive to noise and large thicknesses. The algorithm of [1] is also particulary adapted to multi-orientation cases due to the use of the 3-4 distance. It constitutes a good choice to process line drawing images. We apply next a point accumulation and peak detection as proposed by [7]. This method finds local maxima within the accumulator using a sliding window. The setting of the window size could impact a lot the detection, which constitutes a main drawback of the method. However, this method is much more faster than automatic ones based on iterative accumulations. This speed is mandatory when processing large size documents [6]. In a last step, we detect segments. Our method is based on computation of a dissemination criteria of meet pixels along the scan lines. This dissemination criteria corresponds to the mean of Euclidean distances, between the meet pixels along scanned lines. Post-processing steps are also employed to merge and to delete the near and small segments, using pre-defined thresholds.

## 3 Vector Templates Matching

### 3.1 Introduction

The graphical primitives must be exploited next during a matching process, to achieve the symbol recognition. It exists different approaches in literature to perform such a matching: signatures based classification, matching of ARG[1], etc. In this work we have investigated the vector templates matching.

Vector templates matching has been introduced in years 90's by J.R. Parker. It has been initially applied to character recognition, and next extended to symbols [4]. The key idea of this approach is to apply template matching technics to vectorial representations. Like this, it takes benefit of robustness of template matching and flexibility of vectorial representation (straight-lines, arcs, etc.). Vectors can be scaled and rotated easily without any distortion, it is then possible to render the matching invariant.

[1] Attributed Relational Graph

In the system proposed by J.R. Parker [4], images of any arbitrary dimensions are first converted (detection of vectors, rotation and scaling) into vector templates of regular square sizes. The templates are next matched together. For that purpose, they are redrawn into raster templates by respecting the original widths of vectors. The final match score is obtained by comparing the pixels of rasters. An Euclidean distance is computed for each pixel, to the nearest pixel of a same value in a raster template to compare. The global distance corresponds to the average.

We propose here a revisited method for vector templates matching. Indeed, the original method proposed by [4] raises problems due to the use of raster templates for comparison. This distorts the vectors in order to map them into pixels, especially when using small templates. In addition, this approach can't be used in a localization perspective. The computation of distance involves several accesses to rasters to look for the nearest pixels, impossible to apply with large images. To avoid these problems a solution could be to match at a vector level. Initial information without any distortion could be used, moreover vectors correspond to small subsets of points in regard to rasters. However, template matching must be be redefined to be apply at a vector level. We propose here an algorithm to achieve such a matching, detailed in the next subsections 3.2 and 3.3.

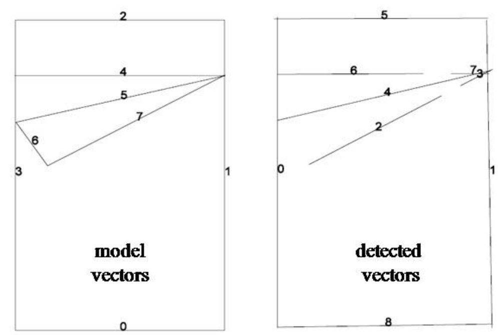
### 3.2 Building of Templates

The first step of our algorithm is to build the vector templates. A bounding box is computed from detected vector (i.e. straight-lines), using min-max methods. We rescale next these vectors in order to place their bounding box within a template of $128^2$. In a last step, they are shifted in order to center the bounding box to the middle of template. Our system represent the vectors at a sub-pixel precision of $10^{-6}$, making like this few distortions.

### 3.3 Matching of Templates

The key problem next is to define a templates matching algorithm able to work at a vector level. In fact, such a problematic has been already studied in the performance evaluation field for vectorization processes. Such an evaluation aims to compare a groundtruth with some vectorization results. Thus, proposed algorithms could be extended to vector templates matching. In our system we have taken benefit of methods proposed by [5, 9].

As proposed by [5], we build a match score matrix between a model and a built templates (Fig. 2). This matrix gives the best match scores between vectors of two templates. The global match score could be then obtained by meaning the matrix. We build it in three steps, employing a method closed to [5]: overlapping test, ranking process and computation of overlapping distances. Our overlapping test Fig. 3 (a) aims to determine possible overlappings between vectors of two templates. It is applied between extremities of two vectors to compare. When all the overlapping relations have been identified, we compute for each positive case an Euclidean distance between the corresponding vectors. We use these distances to rank the vectors from near to far ones. At last, we match the nearest ones using the overlapping distance proposed by [5] Fig. 3 (b), and report them into the matrix (Fig. 2).




detected vectors

| model vectors | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | | | | | | | | | 1,4 |
| 1 | | 0,4 | | | | | | | |
| 2 | | | | | | 1,1 | | | |
| 3 | 0,1 | | | | | | | | |
| 4 | | | | | | | 15 | 41 | |
| 5 | | | | | 1,1 | | | | |
| 6 | 43 | | | | | | | | |
| 7 | | | 12 | 41 | | | | | |

**Figure 2. Match Score Matrix [0-100]**

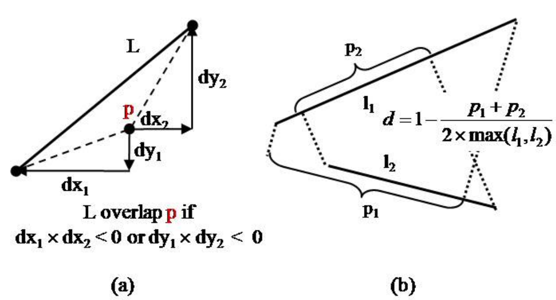
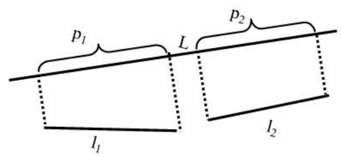



**Figure 3. Overlapping (a) test (b) distance**

However, some fragmentation cases could appear within the matrix. These cases correspond to vectors not well detected, like the cases (7) to (2,3), or (4) to (6,7) in Fig. 2. They could impact a lot the global match score, and must be processed consequently. In [5], these fragmentation cases are just detected and counted in an edit-cost, corresponding to hand-made corrections to apply to vectorized drawings.

Such a criterion is not adapted to the goal of vector templates matching, match scores are not recomputed. For that purpose we have extended the initial approach of [5] with fragmentation quality method proposed by [9] Fig. 4 (a). This one allows to match a vector with a set of fragmented vectors. It is defined as the average of lengths of fragmented vectors projected to a full one.

(a)




$$d\left(L, \bigcup_{k=1}^{s} p_i\right) = 1 - \min\left(L, \sum_{k=1}^{s} p_i\right) \Big/ \max\left(L, \sum_{k=1}^{s} p_i\right)$$

(b)

$$D = \frac{\sum_{i=0}^{n} \sum_{j=0}^{m} \left(c(i,j) \times w(i,j)\right) + False + Misses}{Size}$$

**Figure 4. score (a) fragmentation (b) global**

To apply this method, it is necessary to detect the fragmentation cases first. It is achieved in our algorithm by creating a weighted matrix (Fig. 5) of the match score one. This matrix identifies the fragmentation cases, and provides a weight for each match score depending of the detected case. These weights will be used next in combination to the match score matrix, to compute the global distance. To build this matrix, we count all exiting values in the match score matrix. A count $n > 1$ corresponds to a fragmentation case, either one vector model to many (line), or either many to one (column). We put then weights of $\frac{1}{n}$ values in the weighted matrix at same places. The non detection cases correspond to missed vectors or false alarms.

detected vectors

| model vectors | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | | | | | | | | | 1,0 |
| 1 | | 1,0 | | | | | | | |
| 2 | | | | | | 1,0 | | | |
| 3 | 0,5 | | | | | | | | |
| 4 | | | | | | | 0,5 | 0,5 | |
| 5 | | | | | 1,0 | | | | |
| 6 | 0,5 | | | | | | | | |
| 7 | | | 0,5 | 0,5 | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| One to one | 4 | One to many | 2 | Many to one | 1 |
| Miss | 0 | False | 0 | Size | 7 |

**Figure 5. Weight Score Matrix [0-1]**

We process next the detected fragmentation cases with the fragmentation quality method of [9] Fig. 4 (a). The obtained results are reported in the match score matrix Fig. 6. We use next this modified match score matrix with its associated weighted matrix, to compute the global match score Fig. 4 (b). This global match score is obtained by multiplication of each weight $w_{i,j}$ with their corresponding match score $c_{i,j}$. It takes too into account the false alarm and missed cases. The Fig. 6 gives the global match score before (Score 1) and after (Score 2) the process of fragmentation cases, as soon as the gained delta between them.

detected vectors

| model vectors | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | | | | | | | | | 1,4 |
| 1 | | 0,4 | | | | | | | |
| 2 | | | | | | 1,1 | | | |
| 3 | 0.1 | | | | | | | | |
| 4 | | | | | | | 12 | 12 | |
| 5 | | | | | 1,1 | | | | |
| 6 | 0.1 | | | | | | | | |
| 7 | | | 7 | 7 | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| Score 1 | 11.4 | Score 2 | 5.1 | Delta | 6.4 |

**Figure 6. Modified Match Score Matrix [0-100]**

## 4 Experiments and results

In this section we present experiments and results about our system. We have tested it on the database of international contest of symbol recognition GREC2003[2] [8]. This database deals with the recognition of segmented architectural and electrical symbols. Because it has been developed for the $1^{st}$ GREC contest, it is well known today and widely used for comparisons. Different tests are available according to the number of class, the type and level of noise, the geometrical transformations (scale and/or orientation), etc. Each test is provided with groundtruth to allow performance evaluation. Among these tests we have considered those of binary degradation. They have been generated using the *Kanungo* method with different settings. We have retained only the tests with 20 symbol classes. Indeed, our system can't detect arcs, that constitutes a strong limitation to test it to a higher recognition problem. At the end, we have used 9 tests of 100 images each.

The Fig. 7 gives our recognition results. We have obtained a global recognition rate of $97.9\%$. This rate is just slightly lower than those of GREC2003 and GREC2005 winers (both based on a statistical approach) on the same database, respectively of $99.7\%$ [8] and $100.0\%$ [10]. Our results start from a lower rate of around $92\%$ to perfect rates of $100\%$. During the recognition process, we have also computed the mean match score for each test. The Fig. 7 gives the corresponding curve. These match scores are

[2] http://www.cvc.uab.es/grec2003/

ranked from 0.02 to 0.07. The higher values are obtained for tests 3 to 5, corresponding to the lower recognition rates.

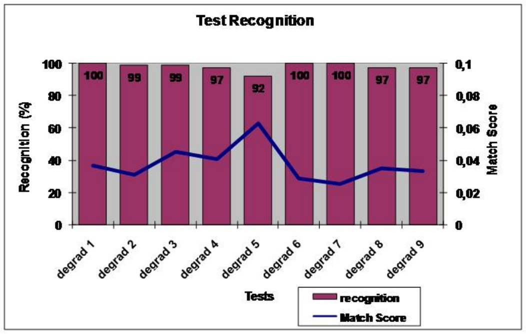



**Figure 7. Recognition results**

In order to interpret our recognition results, we have also computed the recognition rates of individual symbol classes, and the global confusion matrix. Fig. 8 presents a typical recognition error of our system. Such a symbol is often confounded. Indeed, it is composed of some vectors of small lengths, difficult to identify properly with our HT-based detection. The bottom parallel vector is too near to other ones, especially when noise makes connect them by filling the hole (like in test *degrad 5*). The two vectors are merged during the detection, and the symbol is matched to another representative model during the recognition step.

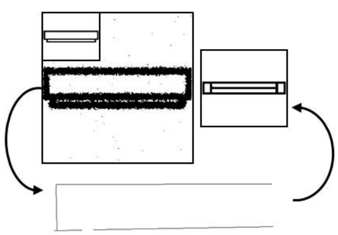


**Figure 8. Typical recognition error**

## 5 Conclusions and Perspectives

In this paper we have presented a robust structural system of symbol recognition. Our key objective here is to provide a system equaling the statistical ones in robustness concerning the recognition, to apply next to localization. To do it we have investigated two particular structural methods: the straight line detection using HT and the vector templates matching. Both of them presents interesting properties of robustness. We have shown in our experiments, how the combination of these two methods, permits to obtain high recognition results on the GREC2003 database.

Concerning the perspectives, the short term one will be to integrate arc and circle detection in our system. It will allow us to extend recognition experiments to other datasets, especially regarding scalability. Our long term perspective concerns the extension of our method to symbol localization. Indeed, we have re-defined our templates matching to work at the vector level. The use of this vectorial representation for matching, makes today possible the application of windows technics (sliding, framing, etc.) to match our templates. Experiments done in past contributions [3] show than localization could be achieved within reasonable times using such a technic.

## 6 Acknowledgement

This work is supported by the French ANR project MADONNE, the Spanish project TIN2006-15694-C02-02, and by the Spanish research programme Consolider Ingenio 2010: MIPRCV (CSD2007-00018).

## References

[1] G. D. Baja. Well-shaped, stable, and reversible skeletons from the (3,4)-distance transform. *Journal of Visual Communication and Image Representation*, 5(1):107–115, 1994.

[2] M. Delalandre, E. Valveny, and J. Lladós. Performance evaluation of symbol recognition and spotting systems: An overview. In *Workshop on Document Analysis Systems (DAS)*, pages 497–505, 2008.

[3] P. Dosch and J. Lladós. Vectorial signatures for symbol discrimination. In *Workshop on Graphics Recognition (GREC)*, volume 3088 of *Lecture Notes in Computer Science (LNCS)*, pages 154–165, 2004.

[4] J. Parker, J. Pivovarov, and D. Royko. Vector templates for symbol recognition. In *International Conference on Pattern Recognition (ICPR)*, volume 2, pages 602–605, 2000.

[5] I. Phillips and A. Chhabra. Empirical performance evaluation of graphics recognition systems. *Pattern Analysis and Machine Intelligence (PAMI)*, 21(9):849–870, 1999.

[6] J. Song, M. Cai, M. Lyu, and S. Cai. A new approach for line recognition in large-size images using hough transform. In *International Conference on Pattern Recognition (ICPR)*, volume 1, pages 33–36, 2002.

[7] P. Thrift and S. Dunn. Approximating point-set images by line segments using a variation of the hough transform. *Computer Vision, Graphics, and Image Processing (CVGIP)*, 21:383–394, 1983.

[8] E. Valveny and P. Dosch. Symbol recognition contest: A synthesis. In *Workshop on Graphics Recognition (GREC)*, volume 3088 of *Lecture Notes in Computer Science (LNCS)*, pages 368–386, 2004.

[9] L. Wenyin and D. Dori. A protocol for performance evaluation of line detection algorithms. *Machine Vision and Applications*, 9:240–250, 1997.

[10] W. Zhang, L. Wenyin, and K. Zhang. Symbol recognition with kernel density matching. *Pattern Analysis and Machine Intelligence (PAMI)*, 28(12):2020–2024, 2006.