Flexible Structural Signature for Symbol Recognition using a Concept Lattice Classifier

Mickaël Coustaty, Stéphanie Guillas, Muriel Visani, Karell Bertet and Jean-Marc Ogier

L3I, University of La Rochelle, av M. Crépeau, 17042 La Rochelle Cedex 1, France,
E-mail: {mcoustat, sguillas, mvisani, kbertet, jmogier}@univ-lr.fr

Abstract

In this paper, we propose a new approach for symbol recognition using structural signatures and a Galois Lattice as classifier. The structural signatures are based on topological graphs computed from segments which are extracted from the symbol images by using an adapted Hough transform. These structural signatures, that can be seen as dynamic paths which carry high level information, are robust towards various transformations. They are classified by using a Galois Lattice as a classifier. The performances of the proposed approach are evaluated on the GREC03 symbol database and the experimental results we obtain are encouraging.
Keywords: Symbol recognition, Concept lattice, Structural signature, Hough transform, Topological relation

1 Introduction

This paper deals with the symbol recognition problem. The literature is very abundant in this domain [1, 4, 10, 12]. Symbol recognition can be basically defined as a two-step process: signature extraction and classification. Signature extraction can be achieved by using statistical-based methods or syntactic/structural approaches. Most of the statistical-based methods use the pixel distribution. They are generally coupled with probabilistic or connexionist classifiers. On the other hand, syntactic and structural approaches are generally based on a characterization of elementary primitives that are extracted from the symbols (basic description, relations, spatial organization, ...) In this paper, a new approach for symbol recognition is introduced. It is based on the use of Galois lattices (also called concept lattices) [3] as classifier. The combined use of statistical-based signatures and Galois lattices has already been introduced by Guillas *et al.* in [6]. Our proposed approach is based on the use of structural signatures inspired by the work of Geibel *et al.* [4] and a Galois lattice classifier. The paper is organized as follows. Section 2 describes the proposed technique. Section 3 gives experimental results. Section 4 provides a conclusion and presents our future work.

2 Description of the Approach

The technique that is introduced in this paper is based on the combined use of structural signatures and of a Galois lattice classifier. The elementary primitives are segments which are extracted by using the Hough transform. For each symbol, we compute a topological graph. The signatures, which are constructed from the topological graph, are classified using a Galois Lattice classifier.

2.1 Segments Extraction

The structural primitives we use for symbol description are segments. The segments extraction method we have implemented is an adaptation of the Hough transform, initially defined for line extraction [7].
During the last thirty years different types of extraction methods have been proposed in the literature [13]: skeletonization, contouring, tracking, run length encoding.... In this paper, we consider Hough Transform (HT) based approach. Indeed, among the existing methods, these ones are known for their robustness property [13], especially in the context of very noisy images. The HT has been introduced in years 60's by [7]. Its key idea is to project pixels of a given image into an parametric space where the shapes can be represented in a compact way. This space is used to find curves that can be parameterized like straight lines, polynomials,

circles,.... Thus, the line segmentation problem in a given image can be considered as a peak detection inside the Hough space. Since the HT has been widely studied, a large number of papers is available [9]. It has been applied to different purposes in image processing like image comparison, filtering ....
For our work we are especially interested in the detection of straight lines. The Figure 1 shows how pixels of an image, represented with their (x, y) coordinates, can be mapped in the Hough space where any straight line of the image is represented by a couple (ρ_i, θ_i). This couple corresponds then to the polar coordinates of a line obtained by its normal parameterization, defined by: $\rho = x \times \cos\theta + y \times \sin\theta$.

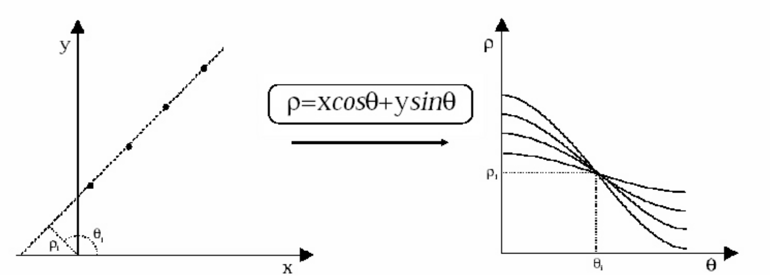

Figure 1: Straight Line Hough Transform (SLHT)

From a theoretical point of view the SLHT seems easy to apply to segment detection. Obviously, its practical use raises different problems [9]. First of all the HT is of quadratic complexity, it is then necessary to use a pre-processing step in order to decrease the number of pixels to map during the transform. Next, on real-life images, the mapped points produce heterogenous sine curves in the Hough space and multiple crossing points can appear. So, a peak detection algorithm is needed in order to group these crossing points and to detect their corresponding mean line. At last, the end points of detected lines cannot be known from the analysis of the Hough space. So, it is necessary to map the lines detected in the Hough space on their corresponding document image in order to achieve the detection process. Based on these considerations an HT-based segment detection system can be divided into four main steps:

1. **Characteristic point selection:** Some characteristic points are to be selected here, before performing the HT, in order to reduce the number of pixels to map, and so the time processing. In our method, we just use a mean filtering in combination with a skeletonization processing [7].

2. **Hough Transform:** Each point previously selected is mapped on the Hough space. This step corresponds to the process shown in Figure 1. An accumulator array is commonly used during this step in order to record the number of sine curve for a given point in the Hough space. We use the initial HT implementation of [7].

3. **Peak detection:** It consists in identifying the points in the accumulator for which the number of sine curves is important enough. Our peak detection algorithm is based on the analysis of the gravity centers of the line sets.

4. **Segments extraction:** The lines detected in the Hough space are mapped on their corresponding document image in order to extract segments (the begin and end points). It consists in detecting sequence of strictly adjacent pixels along the scanned line. This is realized using the Euclidean distances $d(p_i, L)$ between the line L and the crossing points of the image. During this detection stage we also perform a gap verification. If a gap between two consecutive sequences is too small we merge them. Finally we check the length of every line previously obtained. If a length is too small, we consider the line as being produced by noise and we delete it. This step aims at avoiding false alarms (*ie.* lines that are detected but do not exist), which are mainly met at the crossing areas of shapes.

Our algorithm performs robust extraction of maximal segments. An example of the obtained results is shown in Figure 2. The maximal length of the segments implies a reduction of the possible junctions between adjacent segments.

2.2 Topological Graph Computation

2.2.1 Description

Once the segments are extracted, each topological relation between two segments s and s' is described by the following triplet of information:

$$< \textbf{\textit{relation type}}, \textbf{\textit{relation value}}, \textbf{\textit{length ratio}} > \qquad (1)$$

- ***relation type***: We use the finite set of relations types X, Y, V, P, O as in [2, 10, 1, 8] to fully describe the possible relations between pairs of segments (see Table 1).

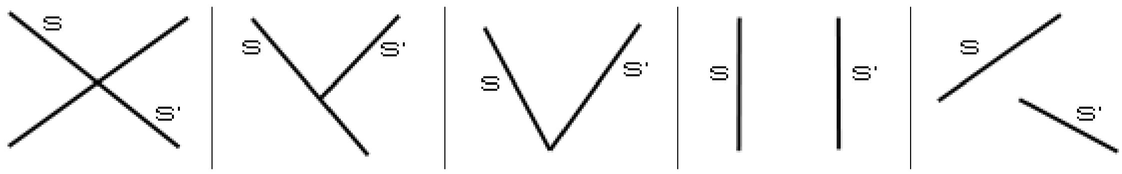

Table 1: The different types of relations we consider (from left to right: X, Y, V, P, O).

- ***relation value***: To be more exhaustive and to discriminate more precisely the relations, we add a value to the relation. This value aims at precising topological relations between segments, such as angle between intersecting segments (available for X, Y, V and O), or distance for parallel segments (relation P).
- ***length ratio***: The last value of each triplet is a ratio between the length of the longest segment and the shortest segment of each pair.

We build a topological graph per symbol where nodes are segments and edges are relations (see Figure 3). The topological graph we obtain is a complete graph where each pair of segments is uniquely described.

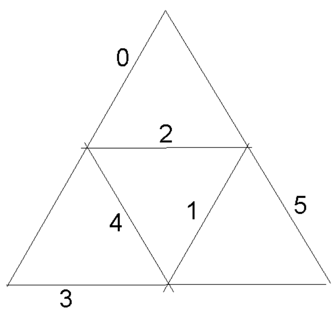

Figure 2: Example of extracted segments

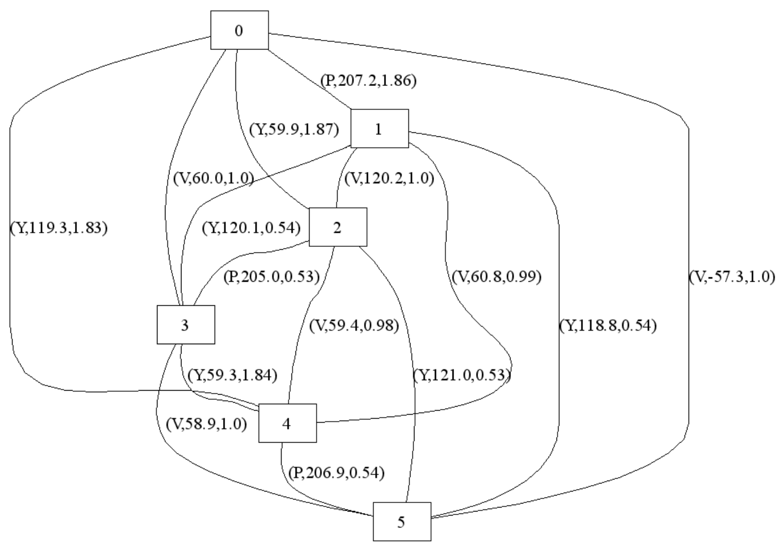

Figure 3: Associated topological graph

Restrictions In order to reduce the number of possible triplets (see Eq. 1), one may discretize them. After performing a statistical analysis of the symbol shapes, we choose to limit the set of possible values for the angles of junctions X, Y and V to the following set: $\{30°, 45°, 60°, 90°\}$ (possibly, a relation value may be assigned to the closest value in that set). It is also possible to discretize the distances between parallel segments in groups (colinear, near, spaced and far for example). The length ratios can be separated into three groups (equal, sligthly different or very different). We can also consider only the type of relation (or any of the pairs <relation type, relation value> or <relation type, length value>), or reduce the set of types of relations we consider.

2.2.2 Discussion

For each symbol, we obtain a set of triplets which fully describes the structural organization of the segments (*eg.*, the relation type differentiates a cross from a rhombus, the relation value a rhombus from a rectangle and the length ratio a rectangle from a square). Moreover, the use of this triplet-based representation has three main advantages:

- each pair of segments is described by one unique triplet;
- each symbol is characterized by one unique and complete graph;
- this description is invariant towards rotation, scale and vectorial distortion.

However, this representation has some drawbacks:

- It does not consider arcs
- We may need a lot of triplets to characterize one symbol (at most n^2, where n is the number of segments)

2.3 Computation of the Structural Signatures

2.3.1 Description

Once the triplets are extracted from each pair of segments, they could characterize the paths of length 1 which are equivalently described by the graph in Figure 3 or its associated adjacency matrix int Table 2, as in [1, 8]. However, we aim at characterizing the symbols by descriptors that discriminate different types of structures, such as regular shapes (square, rectangle, triangle,. . .).

	0	1	2	3	4	5
0		P	Y	V	Y	V
1	P		V	Y	V	O
2	Y	V		P	V	Y
3	V	Y	P		Y	V
4	Y	V	V	Y		P
5	V	O	Y	V	P	

Table 2: Adjacency matrix (M) associated to the graph of Figure 2.

In order to integrate these different structures, as in [4], we compute the paths of different lengths by using the adjacency matrix and its powers (see Tables 2 and 3). Let us denote the adjacency matrix as M. As M conveys information about paths of length 1, M^3 corresponds to 3-length paths (useful to describle triangles), M^4 to 4-length paths (squares and rectangles),. . .
The adjacency matrices we work with are not boolean, so we generalize the usual product of boolean matrices (see Eq. 2) to the union of string concatenation (see Eq. 3):

$$\forall(i,j) \in [0,L]^2,\ (A \times B)_{ij} = \sum_{k=1}^{L} (a_{ik} \times b_{kj}) \quad (2) \qquad \forall(i,j) \in [0,L]^2\,;(A \times B)_{ij} = (\bigcup_{k=1}^{L} (a_{ik} + b_{kj})) \quad (3)$$

where L is the size of the matrix and $+$ is the string concatenation operator. This product is restricted to elementary paths only and the symmetric paths are grouped. For instance, two equivalent paths XV are grouped as $2\times$XV and the symmetric paths POV and VOP are grouped as $2\times$POV. The matrix M^2 corresponding to the square of the matrix M (given in Table 2) is shown in Table 3.

	0	1	2	3	4	5
0		4YV	2PV 2YV	2PY 1YY 1VV	2PV 2YV	2PY 1YY 1VV
1	4YV		2PY 1VV 1YY	2PV 2VY	2PY 1VV 1YY	2PV 2VY
2	2PV 2VY	2PY 1VV 1YY		4VY	1YY 1VV 2PY	2VY 2PV
3	2PY 1YY 1VV	2PV 2VY	4VY		2VY 2PV	1VV 1YY 2PY
4	1VY 1YV 2PV	1VV 1YY 2PY	1YY 1VV 2PY	2VY 2PV		4VY
5	1YY 1VV 2PY	2VY 2PV	2VY 2PV	1VV 1YY 2PY	4VY	

Table 3: Matrix M^2 (where M is given in Table 2).

Once all the power matrices are computed, a set of paths (features) of different lengths is obtained. As in [10], we organize the features in a hierarchical way to compute the signature. So for each symbol image, we compute its structural signature by concatenating all the paths and their number of occurrences. Indeed, the presence of a 4-length path is more discriminative than the presence of a 1-length path, but the distortions affect more the longest paths. For each symbol image, we compute its structural signature by concatenating the type of path and its number of occurrences in the topological graph associated to that symbol.

Restrictions There may be a lot of paths and therefore the signatures may be huge and contain much redondant information. That is why we only consider paths of length inferior or equal to 4. In order to improve the efficiency of this approach, we may also apply a dimensionality reduction method or a selection technique to the lengths of paths to consider or to the paths directly (selection criterion associated to triplets).

2.3.2 Discussion

The structural signatures we obtain are not based on the search for predefined shape templates. Instead, we dynamically compute the shapes observed from our sample images, which confers genericity to our approach. Our method is inspired of the work of Geibel *et al.* [4] and is different on many points. First, we use a Galois lattice instead of a decision tree. Secondly, we do not use the same set of topological relations. Finally, our method is based on a Hough-based segments extraction method from images of symbols where [4] used datasets of chemical compounds and do not use any primitive extractor.

2.4 Classification

We developped a recognition system named NAVIGALA (NAVIgation into GAlois LAttice), dedicated to noisy symbol recognition [11]. As denoted by its name, this system is based on the use of a Galois lattice as classifier. A Galois lattice is a graph which represents, in a structural way, the correspondences between a set of symbols and a set of attributes. These correspondences are given by a binary table (see Figure 4 where each attribute correspond to an interval of occurences for a given path) where crosses are membership relations. In the Galois lattice, nodes are denoted concepts and contain a subset of symbols and a corresponding subset of attributes and edges represent an inclusion relation between the nodes (see Figure 5). The principle of classification is to navigate through the lattice from the top of the graph to the bottom by validating attributes and thus reduce the possibilities of matching symbols. This navigation is similar to the one used for classification with a decision tree. However, in the Galois lattice, several ways are proposed to reach the same node of the graph. We noticed

	X [1]	PP [1]	V [3-4]	VV [3]	VV [4]
×	×				
□		×	×		×
⊠	×	×	×		×
Δ			×	×	

Figure 4: Example of binary table used for lattice construction

that this property is interesting for noisy symbols because, experimentally, concept lattice is more effective than decision tree in the presence of noise.

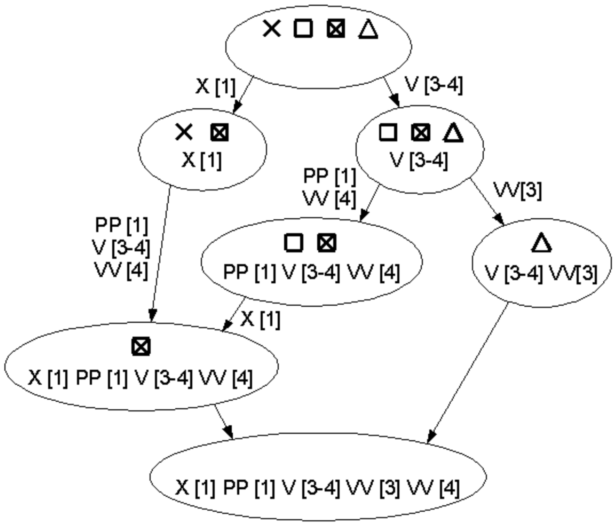

Figure 5: Example of a concept lattice used for classification.

3 Experimental Results

We perform our experiments on the GREC03 database of symbol images [5]. We evaluate the effectiveness of the proposed approach on symbols extracted from 8 classes (see Figure 6) and 9 levels of deterioration (see Figure 7). We use the original symbol, more one symbol per level of deterioration (*ie.* 10 symbols per class) for training. The recognition results are computed from 72 deteriorated query symbol images per class. Tables 4 and 5 provide the recognition rates we obtain by using a) only the relation types and not the full triplet given in (1) (Table 4) and b) the full triplet (Table 5).

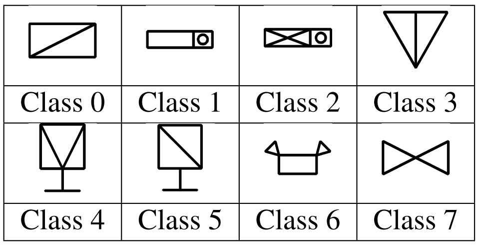

Figure 6: 8 classes of symbol models used for tests.

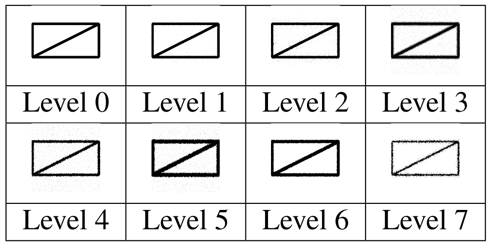

Figure 7: Different levels of noise for class 0.

Lengths of paths	1 and 2	2 and 3	3 and 4	1, 2 and 3
Recognition rate	84%	84,3%	85,6%	83,8%
Number of paths	7	28	161	759

Table 4: Experimental results using partial triplets.

Lengths of paths	1	2	3	4	23	1, 2 and 3	1, 2, 3 and 4
Recognition rate	94,4%	81,9%	82,6%	80,6%	80,4%	90,6%	92%
Number of paths	38	407	3293	16656	3700	3738	20388
Number of attributs	20	20	20	26	20	18	16
Number of concepts	452	685	672	6450	689	262	172

Table 5: Experimental results using full triplets.

For comparison, we perform tests on the same sets of symbols (for learning and recognition) with a method based on the use of statistical signatures (Radon Transform) and a Galois lattice as classifier [6]. The recognition rate we obtain is 98.9%. 14 attributes and 96 concepts were created in the lattice for recognition. We can see that the use of statistical signatures gives a better global recognition rate. But, for the symbols from class 6, it leads to confusions with classes 0 or 3. Using the structural signatures, we recognize symbols from class 6 without any ambiguity with classes 0 and 3 (structural signature induces confusion between classes 6 and 7 for 2 symbols among 81). We can infer that structural and statistical signatures are complementary in terms of recognition results and therefore the may be used jointly in order to improve the performances.

4 Conclusion and Future Work

In this paper, we propose a new structural signature dedicated to symbol recognition using a Galois lattice as classifier. The structural primitives are segments extracted by using an adapted Hough transform. For each symbol, we compute a topological graph from which we can extract the structural signature of that symbol. The signatures are further classified using a Galois Lattice classifier. The experiments we perform on the GREC03 database show the robustness of the proposed approach towards various sources of noise. The structural signatures we obtain are not based on the search for predefined shape templates. Instead, we dynamically compute the shapes observed from our sample images, which makes our approach generic.

In order to ameliorate this structural signature, we are further working on the extraction of circle/ellipse arcs and on their integration into our structural signatures. Next, we aim at evaluating the performances of the proposed approach not only on single symbols, but in real-life applications. Finally, a procedure based on an iterative combination of statistical and structural signatures may enhance the performances of the proposed approach.

References

[1] P. Dosch and J. Lladós. *Vectorial Signatures for Symbol Discrimination*, volume 3088 of *Lecture Notes in Computer Science*, pages 154–165. Springer Berlin / Heidelberg, 2004.

[2] A. Etemadi, J.-P. Schmidt, G. Matas, J. Illingworth, and J. Kittler. Low-Level Grouping of Straight Line Segments. *British Machine Vision Conference*, 1991.

[3] B. Ganter and R. Wille. *Formal Concept Analysis*. Springer Berlin / Heidelberg, 1999.

[4] P. Geibel and F. Wysotzki. Learning Relational Concepts with Decision Trees. In L. Saitta, editor, *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 166–174, San Fransisco, 1996. Morgan Kaufmann Publishers.

[5] GREC03 Database (Graphics RECognition). www.cvc.uab.es/grec2003/symreccontest/index.htm.

[6] S. Guillas, K. Bertet, and J.-M. Ogier. A Generic Description of the Concept Lattices' Classifier: Application to Symbol Recognition. In *Graphics Recognition: Ten Years Review and Future Perspectives - Selected papers from GREC'05*, 2006.

[7] P. Hough. Machine Analysis of Bubble Chamber Pictures. In *International Conference on High Energy Accelerators and Instrumentation*, pages 554–556, 1959.

[8] Q. Iqbal and J. Aggarwal. Retrieval by Classification of Images Containing Large Manmade Objects Using Perceptual Grouping. In *Pattern recognition*, volume 35, pages 1463–1479, 2002.

[9] V. F. Leavers. Survey: which hough transform. *Computer Vision and Image Understanding (CVIU)*, 58(2):250–264, 1993.

[10] M. Rusiñol and J. Lladós. Symbol Spotting in Technical Drawings Using Vectorial Signatures. In L. Wenyin and J. Lladós, editors, *Graphics Recognition: Ten years Review and Future Perspectives - Selected papers from GREC'05*, volume 3926, pages 35–45. Lecture Notes in Computer Science, 2006.

[11] K. B. S. Guillas and J.-M. Ogier. Reconnaissance de Symboles Bruités à l'Aide d'un Treillis de Galois. *Colloque International Francophone sur l'Ecrit et le Document, CIFED 2006*, pages 85–90, 18-21 septembre 2006.

[12] K. Tombre and B. Lamiroy. Graphics Recognition - from Re-engineering to Retrieval. *Proceedings of ICDAR*, 2003.

[13] L. Wenyin and D. Dori. From Raster to Vectors : Extracting Visual Information from Line Drawings. *Pattern Analysis and Applications (PAA)*, 2(2):10–21, 1999.