Symbols Recognition by Global-Local Structural Approaches, Based on the Scenarios Use, and with a XML Representation of Data

Delalandre Mathieu[1], Nicolas Stéphane[1], Trupin Eric[1], Ogier Jean-Marc[2]

[1] *PSI Laboratory, University of Rouen, 76 821 Mont Saint Aignan, France*
[2] *L3I Laboratory, University of La Rochelle, 17042 La Rochelle, France*
[1]*{mathieu.delalandre;nicolas.stephane;eric.trupin}@univ-rouen.fr ;* [2] *jmogier@univ-lr.fr*

Abstract

This paper deals with the structural recognition of symbols on the documents. We have based our system on a combination of local and global structural approaches. The global approach groups the connected components together according to some closeness and connection constraints. The local approach splits up each connected component into a graph of geometrical objects (vectors, arcs, curves). The extracted graphs are matched thanks to a structural classifier, which permits graph-subgraph and exact-inexact matching. The system adaptability is obtained thanks to the scenarios use. A XML data representation is used, allowing the data manipulations and the graphic representations of results.

1. Introduction

This paper deals with the structural recognition of symbols. We present here a system based on the combination of global and local structural approaches. The objective being to obtain an adaptable system, we have based our system on the scenarios use. In the following of this paper, we present in section (2) the general problem of the structural recognition of symbols. The following sections are dedicated to three main parts of our system: the structural classification in section (3), the global structural approach in section (4), and the local structural approach in section (5). In section (6), we give a short presentation of scenarios construction. In section (7), we present experiments and results on a recognition application of technical symbols. In section (8), we present the use of XML in the system. Finally, in section (9), we conclude and give some perspectives.

2. Structural recognition of symbols

Classically, a recognition process is decomposed into two main steps: a data extraction step, and a classification step. In the case of the structural recognition of symbols, the data extraction step is an image-processing one. This step transforms the image into a graphs base, recognized during the classification step. An overview on the symbols recognition in general can be found in [1].

The structural classification step is a graph-matching problem, which depends on two criteria: graph-subgraph and exact-inexact matching. A subgraph is a subset of nodes and edges of a larger graph. The problem of the subgraph-matching is to recognize a model subgraph into a candidate graph. If the classification fits with a strict matching, the problem is known as exact graph-matching. Unfortunately, in image applications, graphs are often noisy and this problem becomes an inexact graph-matching one. The major parts of graph-matching problems are known to be NP (No Polynomial) difficult. An overview of graph-matching applied to the pattern recognition can be found in [2].

The image-processing step extracts graphs from the image. These graphs are no directed, inexact, and rely on a graph or/and subgraph matching problem. In this paper, we propose a classification of this image-processing step into local and global approaches. The boundary between the two approaches is the connected component. Figure 1 gives an example. The local approach (b) decomposes the connected component (a) into arc, junction, and vector objects. The global approach (c) groups together the connected components (a) according to some closeness and connection constraints.
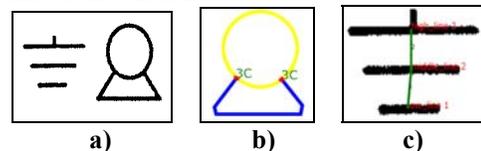

a)   b)   c)

Figure 1: Local and global approaches

The global approach is based on the analysis of spatial relations between the connected components. Classical spatial constraints (Euclidian distance, alignments, and so on) can be used but other approaches as the Voronoi diagrams can be employed too [3]. The global approach is generally used for the structural decomposition of administrative forms, the reconstruction of characters strings, the symbols grouping. The local approach employs different techniques issuing from the symbols recognition and the vectorisation. Generally, there are two steps: an objects graphs extraction step, and an approximation step of these objects. The object type extracted in first step constitutes a good criterion for the classification of local approaches. We have three types of objects graphs extracted: the pixel graph, the run graph,

and the region graph. An Overview on the local structural approaches can be found in [4].

The local and global approaches are two complementary ways for the structural recognition of symbols. In the following of this paper, we use the both approaches with a structural classification for the structural recognition of symbols.

3. Structural classification

The structural toolbox that we use includes methods for the exact-inexact and graph-subgraph matching. The reference link on the exact-inexact graph matching work can be found in [5]. Recently, several methods for exact-inexact subgraph isomorphism have been added to our toolbox, including the "standard" Ullman's method, the Clique based detection method, the exact VF method [6] and the Messmer's decomposition methods (exact and error tolerant) [7]. All these detection methods have been completed with a method for factorization of detected subgraphs. The aim of factorization process is to reduce iteratively the size of input graphs. Thereafter, we use these graph-subgraph matching methods with the global and local approaches results for the symbols recognition.

4. Global structural approach

In [5] we have presented a structural and statistical approach for the symbols recognition. This approach is based on a global structural treatment of the loops image, and a statistical recognition of these loops using mathematical transforms. Figure 2 gives an example of statistical-structural recognition result. On this application, this is the loops extraction step (b) that allows us to use a global approach (c) on these symbols (a).
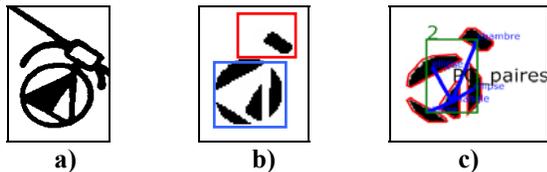

| a) | b) | c) |

Figure 2: Statistical and structural recognition

The global structural tool that is used builds graphs from image using connection constraints and a closeness constraint. We create all connections between all the loops on image, and select the shortest connections according to a Euclidian distance constraint. The connection constraints define the maximum possible connections number by loop label. The statistical training base used defines these labels. So, these connections constraints are very sensitive to statistical results, and must be defined consequently. We can change the constraints order to apply a distance constraint before and the connection constraints after and vice-versa. These constraints permit to control the allowable inter-connection between the graphs corresponding to symbols. So, it is possible to follow two strategies:

i. A graphs construction by segmentation strategy isolates each symbol on image (Figure 2 c).

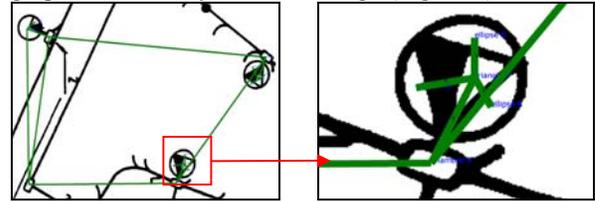ii. A graphs construction by grouping strategy creates a graph for all the symbols on image (Figure 3).

Figure 3: Grouping strategy

Obviously, there are some problems in the case of two nearest symbols, because they are constructed as a single graph. In Figure 2 b, we have an example of two symbols reconstructed as a unique graph (Figure 2 c), the top rectangle represents the first one (1 loop), and the bottom rectangle represents the second one (4 loops). A way to resolve this problem is to use a local approach in order to complete the global approach results.

5. Local structural approach

Our local structural approach is based on the pixels lists graphs extraction (contours detection, skeleton), and the mathematical approximation (vectorisation, curves and arcs fitting). For the objective of our scenario use, we have decomposed the classical vectorisation chain in granular treatments. Our system is decomposed in three levels: an image processing level, a structured-data processing level, and a boundary level between the image data and the structured-data. Each level is composed of different granular treatments (Table 1).

Image Level	Skeletonisation Adaptation Contours Detection, Skeletonisation
Boundary Level	Objects Simplification, Pixels Lists Extraction
Structured Data Level	Degrees Segmentation, Post Corrections Junctions and Graphs Construction, Vectorisation, Curves and Arcs Fitting

Table 1: Treatments levels in local approach

The image processing level uses classical image treatments for the contours detection and the skeletonisation. For the skeletonisation, we use two standard algorithms: Di Baja [8] and Taconet [9]. These skeletonisation algorithms are based on the medial axis transform. For contours detection, we use a classical neighboring test method with a 3*3 basic configuration of the structuring element ("X" and "+"). Figure 4 a gives an example of overlapped results of the skeletonisation and contours detection. The main problem of the skeletonisation is the noise, this noise depends on the

image lines thickness. In order to adapt the image for the skeletonisation, we use the 3-4 distance transform of Di Baja [8] in combination with a classical image-reducing tool. Figure 4 b gives an example of the impact of the skeletonisation without any reduction (middle) and with a reduction (right).
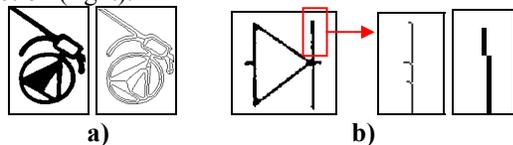

Figure 4: a) Skeleton and contouring b) Adaptation of skeleton

Our boundary level uses two tools: the first one allows to convert the image into structured-data and the other exploits the structured-data on the image. On the one hand, we use a pixels lists extraction tool allowing to convert the skeleton and contours images into structured-data. This tool is based on the connected pixels destruction. First, all 3-connected pixels are destructed. After, we chain the connected pixels into pixels lists. Each list is composed of 1-connected pixels (extremities) and 2-connected pixels. Figure 5 a gives an example of structured-data extracted. On the other hand, we use an objects simplification approach to reduce the complexity of the treated image during the process [10]. Currently, this tool is only running with the arc objects. Figure 5 b gives an example of arc object erasing result.
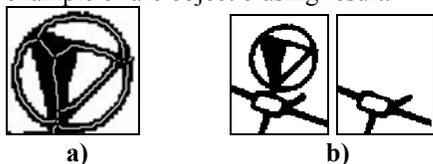

Figure 5: a) Lists extraction b) Object simplification

The structured data level is the main part of the vectorisation scenarios use. Indeed, all treatments use the same input/output data format. In practice, we can call the treatments in any order, but in theory, some treatments depend on a minimum structuring degree of data (for example the arcs fitting depends of vectorisation results). Our data format represents geometrical objects graphs base composed of arcs, curves, polylines, pixels lists, and junctions. The aim of vectorisation scenarios process is to add semantic information to basic graph obtained by the pixels lists extraction tool. For that, we use different granular treatments (see Table 1). For the interiority degrees segmentation, we apply a thickness segmentation threshold based on a simple test of thickness variation. Information of the pixels interiority degrees is obtained by the successive calls of skeletonisation/pixels-lists-extraction tools. Figure 6 a gives an example of interiority degrees segmentation, with the original image (left), the graphic representation of pixels lists before segmentation (middle) and after segmentation (right). For the transformation of pixels lists into vectors lists "polylines",

we use the "standard" lines approximation method of Ramer, with the "standard" Split & Merge fusion procedure of Palvidis. For each polyline, we compute some attributes, like the vectors thickness (from original pixels lists), the vectors lengths, and the angular relations between two consecutive vectors according to the polyline tracing. Our arcs fitting algorithm is a basic tool only based on the test of angular and length relations inside a polyline. We use here a standard geometrical property: a perfect circle can be approximated by a regular (in length and angle) polygon. Figure 6 b gives an example of polygonisation and arcs fitting applied on a contours image. Recently, we extended the geometrical objects extraction with the standard Bernstein curves approximation of pixels lists. Figure 6 c gives an example of Bernstein approximation of contours detection of the Figure 5 b.

Figure 6: a) Interiority degrees segmentation b) Vectorisation and arc fitting c) Curve fitting

An important treatment of the local approach is the junctions detection. We base our approach on a junctions reconstruction algorithm. This algorithm constructs all the connections between the objects extremities using a Euclidian distance constraint. We analyze all the connections to find the inter-connections (group of joined extremities). Each inter-connection constitutes a junction. During this research, we forbid the interconnections between two extremities of a same object. With the information on the junctions, we use a graphs construction algorithm to construct the geometrical objects graphs composed of: pixels, arcs, curves, and polylines lists. We exploit the information on the junctions for different standard treatments in the vectorisation process: pruning and merging. Figure 7 gives a junctions reconstruction example (a, b) with successive pruning (b, c) and merging (c, d) treatments.
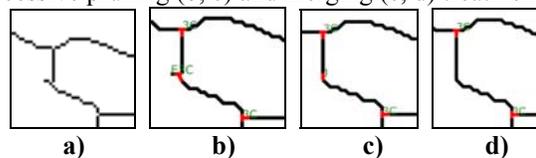

Figure 7: Pruning & merging

6. Scenarios construction

With all the treatments of the local approach, the global approach, and the structural classification, we can construct different treatments chains "scenarios". We use an interface for the scenarios construction and execution. This interface can save and load scenarios from a scenarios base. So, we can test different scenarios according to the recognition goals. A description this interface can be found in [11].

7. Experimentation and results

We present here an application of global-local recognition applied on technical symbols of utility maps (Figure 3). Our system is adaptable to new applications; we have also tested it on the meteorological symbols recognition [5]. On the application of technical symbols, we have 1-4 nodes by graph corresponding to a symbol, and 3 classes of symbols. The major problem of this application is the nearest symbols (see 4). We have tested different scenarios according to five strategies. So, we have tested these strategies on utility maps composed of 50 symbols and 110 loops. We don't deal here with statistical recognition step; this work has been presented and referenced in [5].

The first and the second strategies use a global approach by segmentation, with a graph-matching approach (first strategy), and a subgraph-matching approach (second strategy). We have already presented the first strategy in [5]. In this first strategy, we have considered the nearest symbols as a new unique symbol in the graphs training base (3 to 5 graph classes). We have tested these strategies with two different statistical results, perfect and imperfect, and two construction rules for the global approach by segmentation (distance constraint then connection constraints, and distance constraint alone). So we have obtained a graphs candidate base of 250 graphs (1-5 nodes by graph) composed of 840 nodes. Table 2 gives the results of these strategies (perfect statistical results are noted "0% Noise" and imperfect results "49% of recognition" are noted "51% Noise").

The third strategy uses a subgraph approach with a global approach by grouping. We have tested this strategy with the two different statistical results of the first and second strategies, and five construction rules (no constraint "totality interconnected graph", high distance constraint, three different connections constraints). So, we have obtained a graphs candidate base of 100 graphs (5-20 nodes by graph) composed of 1100 nodes. Table 2 gives the results of this strategy.

The fourth strategy used is a local approach combined with a global approach. The local approach detects the symbols arcs. Our intention being to erase the arcs from symbols, so we reduce the complexity of the global approach. Obviously, with this strategy, we may erase the nearest symbols. Figure 8 gives an example of result. (a) and (b) are the loops extraction result of the Figure 5 b "left" and "right", (b), (c) is the extraction result of the no common image between (a) and (b). With our basic circle-fitting tool we have recognized about 80% of symbols circles (18/22 symbols searched). Then, the following global approach is the first strategy, but without the erased symbols classes into the structural and statistical bases. Table 2 gives the results of this strategy.
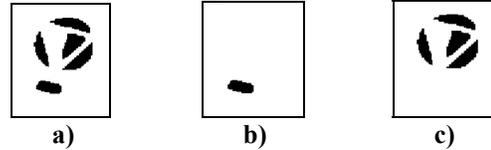

| a) | b) | c) |

Figure 8: Loops extraction after arc detection

Strategy	0% of Noise	51% of Noise
1	100%	86.6%
2	100%	46.8%
3	100%	Incoherent
4	100%	87.5%

Table 2: Strategies comparison

The graph-matching tests of Table 2 have been performed with all the graph-subgraph methods of our graph toolbox (see 3). If the graphs have no noise, we obtain perfect results with all the strategies, but the subgraph or local approaches (strategies 2,3,4) permit the treatment of nearest symbols. For a high level of noise, we obtain incoherent results with a grouping strategy (strategy 3). Indeed, the graphs of symbols classes are small sized, and a false statistical recognition of one node has important effects on a complete recognition of symbol graph. When there is a lot of noise, a segmentation strategy gives better results (strategies 1,2,4), and the graph approach (strategy 1) is better than subgraph (strategy 2). Obviously, we obtain a graph problem less complex when we can obtain one graph for one symbol. In any case, the global-local strategy (strategy 4) gives the most interesting results (good recognition rate, nearest symbols detections). So, the local approach corrects the statistical results. Perfect arcs detection permits a perfect segmentation, and gives a graph problem, which is less complex.

Currently, we test a fifth scenario for the direct structural recognition of loops, and for the zones detection of connected characters. After the circles erasing and the post-processing treatments (merging, pruning), we exploit the junctions and pixels lists graphs. For the direct symbols detection, we search cycles into the candidate graphs. For the zones detection of

connected characters, we construct graphs with the length information of pixels lists (two node labels: short and long), and search into the graphs base the groups of connected nodes, which are labeled short. Figure 9 gives an example of result. We have detected the symbol and the zone of connected character "3".
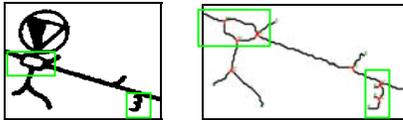

Figure 9: Direct structural recognition

8. XML use in system

Data manipulated by our system are represented with XML[1]. We use this description language for the structured output data of all the treatments, the structural-statistical training-candidate bases for the classifiers, and the scenarios construction results [11]. The use of this data description language offers several advantages. XML permits to use transforming processors. These processors transform a XML data stream with the help of a XSLT[2] script. It is possible, for example, to create easily different graphs with the same XML stream. We have created two different graphs for the direct structural recognition and the zones detection of connected characters (with and without length information), with the same XML stream and two different XSLT scripts (see Figure 9). We also used SVG[3] in the system for all our graphic representations presented in this paper, and XGMML[4] for the graphs description.

9. Conclusion and perspectives

In this paper we have presented a system for the structural recognition, using a combination of global and local approaches, and based on the scenarios use. With the global approach, a bad statistical recognition on small symbols implies a difficult inexact graph-matching task. The use of a local approach reduces the complexity of the global approach, and increases the recognition results. The scenarios use permits to test different strategies according to the recognition goals, and to adapt the system to new applications. For the perspectives, we will to use an automatic threshold method for the symbols construction in the global approach, and for the junctions reconstruction in the local approach. Currently, the system does not analyze the consistency between the strategies results. For that, we plan to use RuleML[5]. This will permit to use contextual knowledge (stocked in rules

base) depending on document applications. We will exploit XML-QL[6] too for the complete reconstruction of XML documents with all our results XML streams.

10. References

[1] J. Lladós, E. Valveny, G. Sánchez, E. Martí, "Symbol Recognition : Current Advances an Perspectives", *Graphics Recognition (GREC)*, 2001.

[2] E. Hancock, R. Wilson, "Graph-Based Methods for Vision: A Yorkist Manifesto", *Structural and Syntactical Pattern Recognition (SSPR)*, 2002.

[3] M. Burge, G. Monagan, "Using the Voronoi Tessellation for Grouping Words and Multi Part Symbols in Document", *Vision Geometry IV*, 1995.

[4] W. Liu, D. Dori, "From Raster to Vectors : Extracting Visual Information from Line Drawings", *Pattern Analysis and Applications (PAA)*, 2(2) : 10-21, 1999.

[5] M. Delalandre, P. Héroux, S. Adam, E. Trupin, et J.M. Ogier, "A Statistical and Structural Approach for Symbol Recognition Using XML Modelling", *Structural and Syntactical Pattern Recognition (SSPR)*, 2002.

[6] P. Foggia, C. Sansone, M. Vento, "An Improved Algorithm for Matching Large Graphs", *Graph Based Representation (GBR)*, 2001.

[7] B.T Messmer, "Thesis: Efficient Graph-matching Algorithms for Preprocessed Model Graphs", *Phd Thesis of Bern University, Switzerland*, 1995.

[8] G.B Di Baja, "Well shaped, Stable, and Reversible Skeletons from the 3-4 Distance Transform", *Journal of Visual Communication and Image Representation*, 5(1) : 107-115, 1992.

[9] B. Taconet, A. Zahour, S. Zhang, A. Faure, "Deux Algorithmes de Squelettisation", *Reconnaissance Automatique de l'Ecriture (RAE)*, 1990.

[10] J. Song, F. Su, C. Tai, S. Cai, "An Object-Oriented Progressive-Simplification Based Vectorisation System for Engineering Drawings: Model, Algorithm and Performance", *Pattern Analysis and Machine Intelligence (PAMI)*, in review.

[11] Y. Saidali, N. Beaudoin, E. Trupin, M. Holzen, J. Labiche, "An Interactive Application for Knowledge Acquisition in Image Processing", *International Conference on Artificial Intelligence and Applications (IASTED)*, 2002.

---

[1] eXtensible Markup Language: http://www.w3.org/
[2] eXtensible Stylesheet Transform Language
[3] Scalable Vector Graphics
[4] eXtensible Graph Markup and Modeling Language
[5] Rule Markup Language

---

[6] XML Query Language