Musings on Symbol Recognition

Karl Tombre[1], Salvatore Tabbone[2], and Philippe Dosch[2]

[1] LORIA-INPL, École des Mines de Nancy, Parc de Saurupt,
CS 14234, 54042 Nancy CEDEX, France
Karl.Tombre@loria.fr

[2] LORIA-Université Nancy 2, Campus scientifique,
B.P. 239, 54506 Vandœuvre-lès-Nancy CEDEX, France
{Salvatore.Tabbone, Philippe.Dosch}@loria.fr

Abstract. In this paper, we review some ideas which emerged in the early years of research on symbol recognition and we show how these ideas evolved into a large variety of contributions. We then propose some interesting challenges for symbol recognition research in the present years, including symbol spotting methods, recognition procedures for complex symbols, and a systematic approach to performance evaluation of symbol recognition methods.

1 Introduction

Symbol recognition is a field within graphics recognition to which a lot of efforts have already been devoted. However, a document analysis expert who is more familiar with OCR might rightfully wonder what exactly we call a symbol and how symbol recognition differs from basic character recognition.

Our feeling is that the problem is very different because of the much higher number and variety of symbols to be recognized. Except in strongly context-dependent applications, it is impossible to provide a database of all possible symbols. It is also in many cases impossible to assume that symbol recognition can be performed on clearly segmented instances of symbols, as symbols are very often connected to other graphics and/or associated with text. The well-known paradox therefore appears: in order to correctly recognize the symbols, we should be able to segment the input data, but in order to correctly segment them, we need the symbols to be recognized!

This in turn means that it is usually not possible to assume that a reliable segmentation process is available, that the symbols have been clearly extracted, normalized, etc. It is hence not reasonable to assume that a vector of general-purpose features can be computed on the segmented areas deemed to be potential symbols, in such a way that the vector can be classified by some appropriate statistical pattern recognition method. The most common approach in symbol recognition therefore relies on structural methods able to capture the spatial and topological relationships between graphical primitives; these methods are sometimes complemented by a classification step, once the candidate symbol has been segmented or spotted.

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 23–34, 2006.

This paper does not pretend to be yet another survey on symbol recognition methods, as several excellent surveys already exist [4, 6, 21]. We will rather try to take a step back, look at the main efforts done in that area throughout the years and propose some interesting directions to investigate.

2 A Quick Historical Overview

As previously said, the early specific work on symbol recognition, as opposed to character recognition, emphasized the use of structural pattern recognition techniques, as usual statistical classification techniques were not suitable. Early efforts included template matching techniques [13], grammar-based matching techniques [8] and recognition techniques based on structural features [11] and dynamic programming [24].

When dealing with specific families of symbols, techniques similar to OCR could be used; this is the case for symbols having all a loop [25] or for music recognition [31]. However these techniques have their own limitations, in terms of computational complexity and of discrimination power.

Very early, people therefore became aware that graph matching techniques are especially suited to specificities of symbol recognition. Twenty years ago, Kuner proposed the search for graph or subgraph isomorphisms as a way for matching symbols with models [14]. Groen et al. [9] analyzed electrical wiring diagrams by representing symbol models by graphs, and using probabilistic matching techniques to recognize symbols. Lin et al. [17] similarly matched symbols to be recognized with model graphs using graph distance computations.

Although simple, this basic idea of graph matching suffers from a number of drawbacks. In its basic principle, it is sensitive to errors and noise; as we usually cannot assume that segmentation is perfect nor reliable, this means that the graphs to be processed can also have a number of extra or missing nodes and vertices. Very early, authors dealt with the general problem of inexact graph matching [34]. In later years, seminal work by Horst Bunke's team has brought to evidence that it is possible to design error-tolerant subgraph isomorphism algorithms [3, 23]. Another possible approach is to make statistical assumptions on the noise present in the image [26].

Another problem with graph matching is the computational complexity of subgraph isomorphism methods. A lot of efforts have therefore been devoted to optimizing the matching process through continuous optimization [15] or constraint propagation techniques to perform discrete [10, 44] or probabilistic [5] relaxation.

Still, another problem remains: that of the scaling of such structural methods to encompass a large number of candidate symbols. It remains to be proven that a symbol recognition method based on graph matching can successfully scale to a large number of model symbols. Also, it is seldom feasible to directly search for subgraph isomorphisms on a whole drawing or document, without some kind of segmentation or pre-segmentation.

Therefore, although there have been a number of successful complete symbol recognition systems, these are mostly within areas with relatively few kinds of symbols to discriminate and within areas where it is easy to localize or pre-segment potential symbols. This includes electrical wiring diagrams [8, 9, 16, 17] and flowcharts [24], typical areas where pre-segmentation can be performed quite easily through separation on the graphical layer between connecting lines and complex areas which are assumed to be symbols. Some attempts have also been made at recognition in areas where pre-segmentation is not easy; this includes work in our own group on recognition of architectural symbols by propagating basic graphical features through a network of nodes representing structural and geometrical constraints on how these features are assembled into symbols [1]. This approach makes it possible to group the information represented by each structural symbol model into a single network, but it remains prohibitively expensive and complex in terms of memory use when the number of model symbols grows. In addition, the fact that the system has to work with noisy data leads to using a number of local rules for inexact matching, and when this propagates through the network there is a real danger of recognizing everything everywhere!

3 Challenges and Research Directions

On the basis of the capabilities and limitations of structural symbol recognition methods, as surveyed above, we discuss in this section a number of interesting challenges and research directions in which our group is currently working.

3.1 The Right Information in the Right Place

Despite their limitations, structural recognition methods provide powerful tools for dealing with complex information. This stems from the large representational power of a graph, as a structure to capture pieces of information and the relationships between these pieces. Attributed relational graphs (ARG) are especially suitable for supporting the structural representation of symbols [26].

But a first challenge is to *put the correct information into the graph.* A typical natural, but often simplistic, and sometimes even wrong way of proceeding is to use the result of some raster-to-vector process to build a graph where the vertices would be the vectors and the nodes the junctions between the vectors. This leads to representing a symbol as a set of graphical features and the spatial relations between these features, represented by relational attributes. Of course, we are aware that it is not enough to have good features in the right place of the graph; the matching method also has to be robust to noise [2].

Adding higher-level topological, geometrical and relational information to the nodes and vertices of the graph can open up new possibilities in recognition problems. When some pre-segmentation methods can divide the image to be analyzed into homogeneous regions, region adjacency graphs are a good candidate as they capture a lot of interesting information [19]. When this is not possible, it may make sense to start with extracting simple graphical features which can be reliably found without prior segmentation: vectors, arcs, basic shapes, and to use a graph where these basic features are attributes of the nodes and the vertices convey information about topological and geometrical relationships between these features (inside, above, at-right-of, touching, etc.) A good example of such use of spatial relations for symbol recognition purposes is the system built by Liu Wenyin's team in Hong Kong [18, 46].

3.2 Symbol Spotting

A way to avoid the dilemma of needing segmentation to perform recognition, and vice-versa, is to try to localize the symbols in a complex drawing without necessarily going all the way through complete recognition. This gives first pieces of information on the subareas in which to apply recognition methods which may be more computationally expensive.

In order to overcome the segmentation vs. recognition paradox, we have worked in the last years on *symbol spotting* methods, i.e. ways to efficiently localize possible symbols and limit the computational complexity, without using full recognition methods. This is a promising approach, based on the use of signatures computed on the input data. We have worked on signatures based on force histograms [38] and on the Radon transform [36, 37], which enable us to localize and recognize complex symbols in line-drawings. We are currently working on extending the Radon signature to take into account photometric information, in order to improve the results when retrieving similar symbols in graphical documents [39]. By using a higher-dimensional signature, we are able to include both the shape of the object and its photometric variations into a common formalism. More precisely, the signature is computed on the symbol, at several grey levels. Thus, the effects of noise on the boundary of the processed object become insignificant, relatively to the whole shape.

When it comes to spotting target symbols, structural approaches are powerful in terms of their representational capabilities. Therefore, we use a simple structural representation of symbols to introduce a hybrid approach for processing symbols connected to other graphical information. For this, we compute a skeleton and we organize its junction points into a graph where the graph edges describe the link between junction points (see Fig. 1). From this representation, candidate symbols are selected and validated with the signature descriptor. Fig. 2 illustrates the working of the system: when a candidate symbol is selected in the document, a number of candidate regions are retrieved.

3.3 Measuring the Progress: Performance Evaluation

As in many other areas within pattern recognition, performance evaluation has become a crucial part of our work on symbol recognition, in order to be able to compare different methods on standard datasets with metrics agreed upon by everyone. Our team co-organized the two first international contests on symbol recognition, held at GREC'03 [40, 41] and at GREC'05. The basic principles are as follows:

– The *datasets* include both real scanned symbols and synthetic data, i.e. symbols stemming from CAD systems which were degraded using a combination

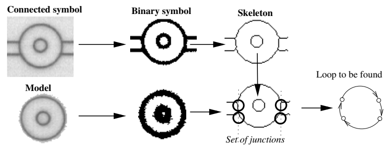

Fig. 1. Example of graph organization based on the junction points(from [39])

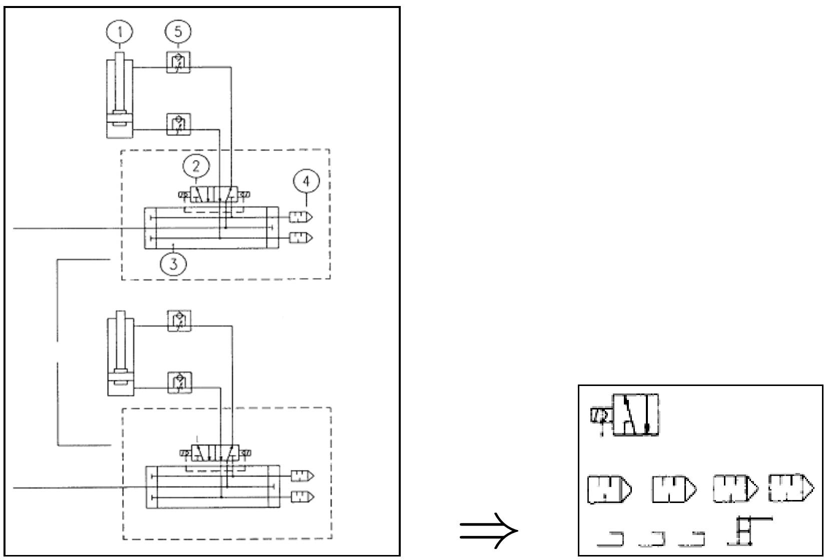

Fig. 2. Example of symbol spotting on an engineering drawing, from [48]. The user delineates a symbol (left) and a number of candidates are retrieved (right).

of an image degradation model [12] and of vectorial degradation [42]. Other basic transformations, such as scaling and rotation, are also used.

There are two types of datasets: isolated symbols (pre-segmented) for which the task is to recognize a symbol among n possible models, with various measures for an increasing n and an increasing degradation of the data, and symbols in their context (without segmentation) where there is a double task of spotting/localizing the symbol, and then recognizing it. Note that although most of the framework was in place, we finally decided not to run the symbol localization part at the second contest.

Managing a great number of heterogeneous data may be confusing for participant methods, sometimes designed for a specific purpose, as well as for post-recognition analysis steps, that could be irrelevant if the results are themselves too heterogeneous. Therefore, all datasets are classified according to several properties, increasing the readability in both cases. Basically, these properties are defined either from a technological point of view (bitmap/vectorial representation, graphical primitives used...) or from an application point of view (architecture/electronic...)

The datasets are further divided into training data made available to the participants beforehand, and test data used during the contest itself.
- The *ground-truth* definition for symbols in their context is simple and readable. It is basically based on the manual definition of bounding-boxes around each symbol of the test data, labelled by the model symbol.
- The *performance measures* for isolated symbols include the number of false positives and missing symbols, the confidence rates (when provided by the recognition method), computation time (which gives an implicit measure of the complexity of the method) and scalability, i.e. a measure of the way the performances decrease when the number of symbols increases.
  The performance evaluation for symbols in their context is based on two measures. The first is unitary and is related to each symbol. It is based on the overlapping of a ground-truth bounding-box and a bounding-box supplied by a participant method, in the case where both symbol labels are the same. The second measure allows us to compose all unitary measures for a test data, and is based on the well-known notions of precision and recall ratios. Again, computation time is used to qualify the scalability of the participant method.
- Finally, the *results analysis* is led from the data point of view (data based), as well as from the methods point of view (methods based). Indeed, if it is interesting to understand which methods give good results with a lot of data, it is also interesting to understand which data are difficult to recognize with respect to the several recognition approaches. The interest of a performance evaluation campaign is guided by these two points of view.
- The general framework provides online access to training data and description of the metrics used.

In addition, our team is leading a project financed by the French government but open to international teams, on the performance evaluation of symbol recognition and logo recognition (see `http://www.epeires.org/`). The purpose of this project is to build a complete environment providing tools and resources for performance evaluation of symbol recognition and localization methods. This environment is intended to be used by the largest possible community. A test campaign, opened to all registered participants, will be organized during its final step. In addition to providing the general framework for organizing benchmarks and contests on a more stable basis, our goal is to make available for the community a complete environment including online collaborative ground-truthing tools, reference datasets, results of already published methods on these datasets, and performance metrics which can be used for research teams throughout the world to compare their own work on symbol localization and/or recognition with the state of the art.

3.4 Complex Symbols

In many cases, a symbol is not only a set of segments and arcs, but a complex entity associating a graphical representation, a number of connection points and text annotations. Symbol recognition should be able to deal with such *complex symbols* in order to be of practical use in a number of areas.

Figure 3 gives some examples of complex symbols from the area of aeronautics (wiring diagrams of an Airbus plane). The challenge here is to be able to

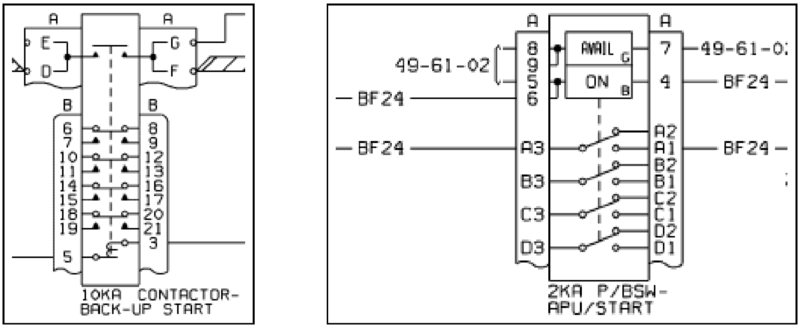

Fig. 3. Examples of complex symbols

discriminate between symbols which may differ not by their graphical shape, nor by their topology, but simply by the number of connectors or by the type of textual annotations. As an example, Fig. 4 illustrates two complex symbols from the area of electrical design which differ only by slight variations in the shape of their upper constituent sub-symbols.

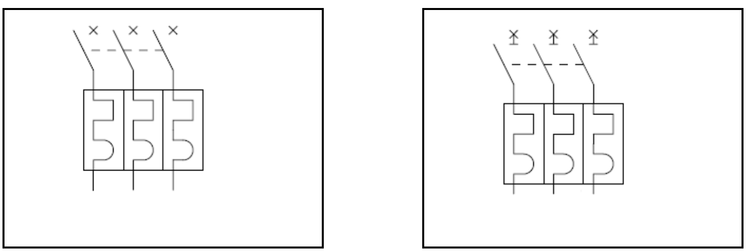

Fig. 4. Example of very similar symbols (courtesy Algo'tech Informatique)

We are still working on the appropriate strategy to deal with this kind of recognition problems. One of our ideas is to compile, from the set of reference symbols, a number of basic shapes which can be considered as the basic building blocks for drawing such symbols: rectangles, triangles, squares, disks, horizontal and vertical segments, other straight segments, arcs, etc. Some of these shapes may be filled and are thus represented by their contour. Then, very simple recognition agents would localize in the drawing all instances of these simple shapes, and progressively remove them from the drawing, to simplify it, following the basic principle applied by Song et al. to the vectorization problem [35]. Complex symbols can then be represented by rules for assembling these basic shapes,

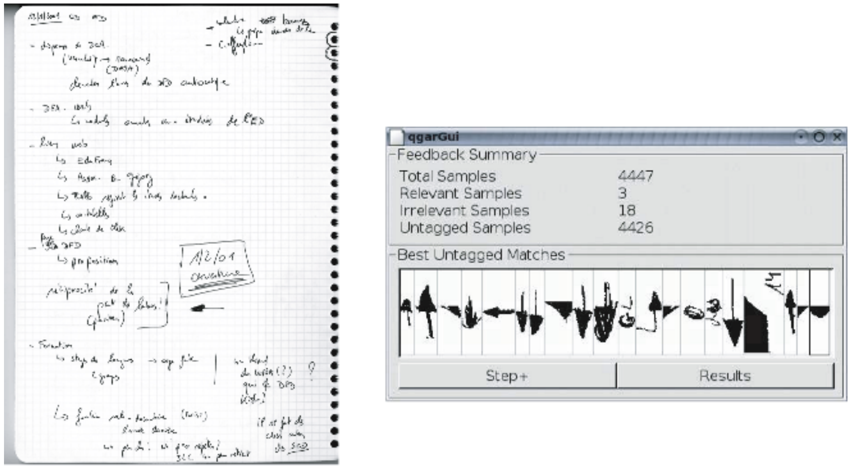

Fig. 5. Working of first prototype for dynamic recognition on scanned handwritten notebooks such as the example on the left side (from [30]). The idea in this example is to retrieve the arrows written by the user.

the annotations present in the text layer, the connection information from the vectorized connecting lines, and other spatial information.

Pasternak was one of the first to experiment with this kind of recognition strategy, with a system combining a number of simple agents triggering assembly rules for recognizing higher-level symbols [27, 28]. However, his system remained complex to adapt and to use in practical applications. We have started to work on this problem and we plan to use structural/syntactic methods such as graph grammars [7, 20, 29, 32] to describe the combination rules leading from the simple shapes and the annotations to complex symbols.

3.5 Dynamic Recognition

Until now, we have addressed the problem of recognizing a symbol among a set of known models. But there are situations, especially when browsing an open set of documentation, where nobody is able to build a library of model symbols or even to predict which symbols the user may be interested in. In that case, we have to rely on what we have called dynamic symbol recognition. The idea is that the user interactively selects an area or a region of a document which (s)he calls a symbol. The challenge is then to retrieve other instances of this symbol in the same document or in other documents available in the digital library.

The system can of course include some relevance feedback mechanism allowing the user to validate or invalidate the results of a first symbol spotting phase and then restart the whole process.

To achieve this, one of our ideas is to rely on a set of simple features which can be pre-computed on the digital library. Each document image can be divided into small images, using either a simple meshing method, or some rough document image segmentation technique. On each subimage obtained through this subdvision or segmentation, one or several signatures, based on the Radon transform or on other generic descriptors [22, 45, 47], can be used to characterize the subimage. When the user selects a part of the image, the descriptors of this part are computed and some distance can be used to find the regions of interest having the closest descriptors. Relevance feedback allows the user to validate or invalidate the different symbols spotted in this way, and the mechanism can be iterated until the user is satisfied with the result.

The scenario sketched above only represents some preliminary ideas on this matter of dynamic recognition, which is ongoing work in our team. Figure 5 illustrates the working of our first prototype, presented in [30].

4 Conclusion

In this paper, we have reviewed some ideas which emerged in the early years of research on symbol recognition and have tried to show how these ideas evolved into a large variety of contributions, which for many of them are based on the same structural recognition paradigm. We have then proposed some challenges for symbol recognition research in the present and coming years.

We are aware that there are a number of other issues which we have not dealt with in this paper. Let us just mention the necessity of combining various approaches to achieve better global recognition results. This includes combining structural and statistical methods, but also combining various descriptors in a better way than simply putting them into a vector where each feature computed is assumed to play the same role and have the same weight. Some first results have been obtained in our group using the Choquet integral to aggregate various descriptors for better symbol spotting and recognition [33, 43].

Symbol recognition has been a research topic for many years already, and spectacular achievements have been obtained. Still, a number of issues remain open and lead to a number of research challenges for the coming years.

References

1. C. Ah-Soon and K. Tombre. Architectural Symbol Recognition Using a Network of Constraints. *Pattern Recognition Letters*, 22(2):231–248, February 2001.
2. H. Bunke. Error-Tolerant Graph Matching: A Formal Framework and Algorithms. In A. Amin, D. Dori, P. Pudil, and H. Freeman, editors, *Advances in Pattern Recognition (Proceedings of Joint IAPR Workshops SSPR'98 and SPR'98, Sydney, Australia)*, volume 1451 of *Lecture Notes in Computer Science*, pages 1–14, August 1998.
3. H. Bunke. Error Correcting Graph Matching: On the Influence of the Underlying Cost Function. *IEEE Transactions on PAMI*, 21(9):917–922, September 1999.
4. A. K. Chhabra. Graphic Symbol Recognition: An Overview. In K. Tombre and A. K. Chhabra, editors, *Graphics Recognition—Algorithms and Systems*, volume 1389 of *Lecture Notes in Computer Science*, pages 68–79. Springer-Verlag, April 1998.

5. W. J. Christmas, J. Kittler, and M. Petrou. Structural Matching in Computer Vision Using Probabilistic Relaxation. *IEEE Transactions on PAMI*, 17(8): 749–764, August 1995.
6. L. P. Cordella and M. Vento. Symbol recognition in documents: a collection of techniques? *International Journal on Document Analysis and Recognition*, 3(2): 73–88, December 2000.
7. H. Fahmy and D. Blostein. A Survey of Graph Grammars: Theory and Applications. In *Proceedings of 11th International Conference on Pattern Recognition, Den Haag (The Netherlands)*, volume 2, pages 294–298, 1992.
8. C. S. Fahn, J. F. Wang, and J. Y. Lee. A Topology-Based Component Extractor for Understanding Electronic Circuit Diagrams. *Computer Vision, Graphics and Image Processing*, 44:119–138, 1988.
9. F. C. A. Groen, A. C. Sanderson, and J. F. Schlag. Symbol Recognition in Electrical Diagrams Using Probabilistic Graph Matching. *Pattern Recognition Letters*, 3: 343–350, 1985.
10. A. H. Habacha. A New System for the Analysis of Schematic Diagrams. In *Proceedings of 2nd International Conference on Document Analysis and Recognition, Tsukuba (Japan)*, pages 369–372, 1993.
11. E. Hansen and K. P. Villanger. A Combined Thinning and Contour Tracing Approach to the Recognition of Engineering Drawing Symbols. In *Proceedings of International Seminar on Symbol Recognition, Oslo (Norway)*, pages 82–100, 1985.
12. T. Kanungo, R. M. Haralick, H. S. Baird, W. Stuezle, and D. Madigan. A statistical, nonparametric methodology for document degradation model validation. *IEEE Transactions on PAMI*, 22(11):1209–1223, November 2000.
13. W. Kikkawa, M. Kitayama, K. Miyazaki, H. Arai, and S. Arato. Automatic Digitizing System for PWB Drawings. In *Proceedings of 7th International Conference on Pattern Recognition, Montréal (Canada)*, volume 2, pages 1306–1309, 1984.
14. P. Kuner. Efficient Techniques to Solve the Subgraph Isomorphism Problem for Pattern Recognition in Line Images. In *Proceedings of 4th Scandinavian Conference on Image Analysis, Trondheim (Norway)*, pages 333–340, 1985.
15. P. Kuner and B. Ueberreiter. Pattern Recognition by Graph Matching — Combinatorial versus Continuous Optimization. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(3):527–542, 1988.
16. S.-W. Lee. Recognizing Hand-Drawn Electrical Circuit Symbols with Attributed Graph Matching. In H. S. Baird, H. Bunke, and K. Yamamoto, editors, *Structured Document Image Analysis*, pages 340–358. Springer-Verlag, Heidelberg, 1992.
17. X. Lin, S. Shimotsuji, M. Minoh, and T. Sakai. Efficient Diagram Understanding with Characteristic Pattern Detection. *Computer Vision, Graphics and Image Processing*, 30:84–106, 1985.
18. Y. Liu, L. Wenyin, and C. Jiang. A Structural Approach to Recognizing Incomplete Graphic Objects. In *Proceedings of the 17th International Conference on Pattern Recognition, Cambridge (UK)*, August 2004.
19. J. Lladós, E. Martí, and J. J. Villanueva. Symbol Recognition by Error-Tolerant Subgraph Matching Between Region Adjacency Graphs. *IEEE Transactions on PAMI*, 23(10):1137–1143, 2001.
20. J. Lladós and G. Sánchez. Graph matching versus graph parsing in graphics recognition — A combined approach. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(3):455–473, 2004.

21. J. Lladós, E. Valveny, G. Sánchez, and E. Martí. Symbol Recognition: Current Advances and Perspectives. In D. Blostein and Y.-B. Kwon, editors, *Graphics Recognition – Algorithms and Applications*, volume 2390 of *Lecture Notes in Computer Science*, pages 104–127. Springer-Verlag, 2002.
22. S. Loncaric. A Survey of Shape Analysis Techniques. *Pattern Recognition*, 31(8):983–1001, 1998.
23. B. T. Messmer and H. Bunke. A New Algorithm for Error-Tolerant Subgraph Isomorphism Detection. *IEEE Transactions on PAMI*, 20(5):493–504, May 1998.
24. H. Murase and T. Wakahara. Online Hand-Sketched Figure Recognition. *Pattern Recognition*, 19(2):147–160, 1986.
25. A. Okazaki, T. Kondo, K. Mori, S. Tsunekawa, and E. Kawamoto. An Automatic Circuit Diagram Reader with Loop-Structure-Based Symbol Recognition. *IEEE Transactions on PAMI*, 10(3):331–341, 1988.
26. B. G. Park, K. M. Lee, S. U. Lee, and J. H. Lee. Recognition of partially occluded objects using probabilistic ARG (attributed relational graph)-based matching. *Computer Vision and Image Understanding*, 90:217–241, 2003.
27. B. Pasternak. The Role of Taxonomy in Drawing Interpretation. In *Proceedings of 3rd International Conference on Document Analysis and Recognition, Montréal (Canada)*, pages 799–802, August 1995.
28. B. Pasternak. *Adaptierbares Kernsystem zur Interpretation von Zeichnungen*. Dissertation zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften (Dr. rer. nat.), Universität Hamburg, April 1996.
29. J. L. Pfaltz and A. Rosenfeld. Web Grammars. In *Proceedings of 1st International Joint Conference on Artificial Intelligence*, pages 609–619, 1969.
30. J. Rendek, B. Lamiroy, and K. Tombre. A Few Step Towards On-the-Fly Symbol Recognition with Relevance Feedback. In H. Bunke and A. L. Spitz, editors, *Document Analysis Systems VII: Proceedings of 7th International Workshop on Document Analysis Systems, Nelson (New Zealand)*, volume 3872 of *Lecture Notes in Computer Science*, pages 604–615, February 2006.
31. J. W. Roach and J. E. Tatem. Using Domain Knowledge in Low-level Visual Processing to Interpret Handwritten Music: An Experiment. *Pattern Recognition*, 21(1):33–44, 1988.
32. A. Rosenfeld. Array, Tree and Graph Grammars. In H. Bunke and A. Sanfeliu, editors, *Syntactic and Structural Pattern Recognition: Theory and Applications*, chapter 4, pages 85–115. World Scientific, 1990.
33. J.-P. Salmon, L. Wendling, and S. Tabbone. Automatical Definition of Measures from the Combination of Shape Descriptors. In *Proceedings of 8th International Conference on Document Analysis and Recognition, Seoul (Korea)*, September 2005.
34. L. G. Shapiro and R. Haralick. Structural Description and Inexact Matching. *IEEE Transactions on PAMI*, 3(5):504–519, 1981.
35. J. Song, F. Su, C.-L. Tai, and S. Cai. An Object-Oriented Progressive-Simplification Based Vectorization System for Engineering Drawings: Model, Algorithm, and Performance. *IEEE Transactions on PAMI*, 24(8):1048–1060, August 2002.
36. S. Tabbone and L. Wendling. Technical Symbols Recognition Using the Two-dimensional Radon Transform. In *Proceedings of the 16th International Conference on Pattern Recognition, Qubec (Canada)*, volume 2, pages 200–203, August 2002.

37. S. Tabbone and L. Wendling. Binary shape normalization using the Radon transform. In *Proceedings of 11th International Conference on Discrete Geometry for Computer Imagery, Naples (Italy)*, volume 2886 of *Lecture Notes in Computer Science*, pages 184–193, November 2003.
38. S. Tabbone, L. Wendling, and K. Tombre. Matching of Graphical Symbols in Line-Drawing Images Using Angular Signature Information. *International Journal on Document Analysis and Recognition*, 6(2):115–125, 2003.
39. S. Tabbone, L. Wendling, and D. Zuwala. A Hybrid Approach to Detect Graphical Symbols in Documents. In *Proceedings of the 6th IAPR International Workshop on Document Analysis Systems, Florence, (Italy)*, volume 3163 of *Lecture Notes in Computer Science*, pages 342–353, September 2004.
40. E. Valveny and Ph. Dosch. Performance Evaluation of Symbol Recognition. In *Proceedings of the 6th IAPR International Workshop on Document Analysis Systems, Florence, (Italy)*, volume 3163 of *Lecture Notes in Computer Science*, pages 354–365, September 2004.
41. E. Valveny and Ph. Dosch. Symbol recognition contest: a synthesis. In J. Lladós and Y. B. Kwon, editors, *Graphics Recognition: Recent Advances and Perspectives – Selected papers from GREC'03*, volume 3088 of *Lecture Notes in Computer Science*, pages 368–385. Springer-Verlag, 2004.
42. E. Valveny and E. Martí. Deformable Template Matching within a Bayesian Framework for Hand-Written Graphic Symbol Recognition. In A. K. Chhabra and D. Dori, editors, *Graphics Recognition—Recent Advances*, volume 1941 of *Lecture Notes in Computer Science*, pages 193–208. Springer-Verlag, 2000.
43. L. Wendling and S. Tabbone. A New Way to Detect Arrows in Line Drawings. *IEEE Transactions on PAMI*, 26(7):935–941, July 2004.
44. R. C. Wilson and E. R. Hancock. Structural Matching by Discrete Relaxation. *IEEE Transactions on PAMI*, 19(6):634–648, June 1997.
45. J. Wood. Invariant Pattern Recognition: A Review. *Pattern Recognition*, 29(1): 1–17, 1996.
46. X. Xiaogang, S. Zhengxing, P. Binbin, J. Xiangyu, and L. Wenyin. An online composite graphics recognition approach based on matching of spatial relation graphs. *International Journal on Document Analysis and Recognition*, 7(1):44–55, September 2004.
47. S. Yang. Symbol Recognition via Statistical Integration of Pixel-Level Constraint Histograms: A New Descriptor. *IEEE Transactions on PAMI*, 27(2):278–281, February 2005.
48. D. Zuwala and S. Tabbone. A Method for Symbol Spotting in Graphical Documents. In H. Bunke and A. L. Spitz, editors, *Document Analysis Systems VII: Proceedings of 7th International Workshop on Document Analysis Systems, Nelson (New Zealand)*, volume 3872 of *Lecture Notes in Computer Science*, pages 518–528, February 2006.