

# Structural and Syntactic Methods in Line Drawing Analysis: To Which Extent Do They Work?

Karl Tombre

INRIA Lorraine & CRIN/CNRS

Bâtiment LORIA,

615, rue du jardin botanique

B.P. 101

54602 Villers-lès-Nancy CEDEX

France

*E-mail:* Karl.Tombre@loria.fr

*URL:* <http://www.loria.fr/~tombre/>

## 1 Introduction

Syntactic methods in pattern recognition have been around for more than 20 years now [22, 23], and numerous applications have been done with these basic techniques. Structural pattern recognition is an even vaster field [47] (syntactic methods can actually be considered to belong to it), where many methods exist and have been used for various practical applications.

These methods have been extensively used in the application domain of character recognition, document analysis and graphics recognition, as appears from our review in § 2. It may therefore seem strange to ask today if these methods really work. But it has been our experience that when designing larger graphics recognition systems, we come to a number of what can be called “border problems”, which at first glance appear to be suitable precisely for this family of recognition methods, but where expected and more or less unexpected hindrances make life difficult for the recognition methodology. We present some of these problems in § 3.

As people use more or less strict definitions of what structural and syntactic pattern recognition is, let us first introduce our own understanding of these concepts. *Structural methods* are based on the relational organization of low-level features into higher-level structures. These structures are then matched with model structures. These methods are usually associated with graph matching techniques and structural organization of simple features into perceptual groups, based on a number of relationships between the features. We summarize the main idea by citing Suetens *et al.* [54]:

Relational matching overcomes the major inadequacies of pattern recognition by providing a representation for relational constraints. Objects or scenes are represented as relational structures whose nodes are sub-parts and whose arcs are relationships between the nodes they connect. The problem of matching relational structures is representable as one of optimizing some objective function. Heuristic search can be used to prune the search tree and reduce the computation time, at the possible risk of finding a non-optimal solution.

Graph matching techniques have been extensively studied. One typical problem is that of finding an instance of a known structural model in a larger graph, which usually represents the features and their relations, extracted from the input data to be analyzed. This problem is called subgraph isomorphism. The basic algorithm for solving it was proposed by Ullmann [59]. Many methods have been proposed to speed up the recognition process, through various heuristics, and to extend it to error-tolerant methods. For a recent good survey of these methods and presentation of some of the best algorithms for efficient and error-tolerant subgraph matching, we refer the reader to Bunke and Messmer [7].

*Syntactic pattern recognition* comes as a natural complement (one may also say as a specialization) of these basic matching methods. In his well-known book [22], K. S. Fu introduces the main idea as follows:

In order to represent the hierarchical (treelike) structural information of each pattern, that is, a pattern described in terms of simpler subpatterns and each simpler subpattern again described in terms of even simpler subpatterns, etc., the syntactic or structural approach has been proposed. This approach draws an analogy between the (hierarchical, or treelike) structure of patterns and the syntax of languages. Patterns are specified as being built up out of subpatterns in various ways of composition, just as phrases and sentences are built up by concatenating words, and words are built up by concatenating characters. [. . .] The syntactic approach to pattern recognition provides a capability for describing a large set of complex patterns by using small sets of simple pattern primitives and of grammatical rules.

K. S. Fu goes on to define a rigorous framework for syntactic pattern recognition, and many researchers have worked on formal grammars for patterns, proposing languages such as PDL [51] or Plex languages [21], and applying them to many domains. We have the tendency to be less strict in our use of the concept of “syntactic” methods, and we extend it to pattern recognition methods which use some kind of syntactic description of the possible pattern combinations, even if they don’t explicitly use a grammar and a parser. This trend was already noted by Bunke and Sanfeliu [8]:

There is no doubt that there will be a continuous convergence between the disciplines of pattern recognition, image understanding and waveform analysis on the one hand, and artificial intelligence on the other. We feel that syntactic methods can already be considered to be a first link between these disciplines. There is considerable evidence supporting this statement. Generally, a syntactic pattern recognition system can be considered a particular instance of a knowledge based system with a formal grammar as knowledge base and a parser as inference engine.

In particular, we consider the methods we used in our CELESSTIN III system [61], for instance, to be of syntactic nature, although we used a blackboard-based expert system instead of a grammar and a parser.

The fact that we limit ourselves in this paper to structural and syntactic methods does of course in no ways mean that we despise or reject the statistical approach

to pattern recognition. Actually, the complementarity of these two families has already been extensively proved [24], attributed and stochastic grammars have been proposed [36, 58] and used in various applications [53, 55], probabilistic relaxation is often used in connection with structural matching [9], etc. But in this paper, we concentrate on the use of structural and syntactic pattern recognition methods when dealing with the analysis of graphics and line drawings.

## 2 Structural and syntactic methods in document and graphics analysis

*Document analysis* in general, and line drawings interpretation in particular, is a domain where the structural and syntactic approach is well suited and has been used in many applications. Some of these applications are especially well adapted to a model description through syntactic means:

- Two-dimensional mathematical notation is typically organized according to strict rules which can be represented syntactically [2]. For example, one of the systems [4] localizes a main operator in the formula, and then recursively parses the two parts of the formula separated by this operator. Recognition of the individual characters is performed through a structural method.
- Character recognition systems have themselves taken syntactic methods into use. For instance, Wolberg [64] describes the morphology of the characters in a structural way, and uses PDL to define the constraints and relationships between the low-level structural features. Duerr *et al.* [19] associate statistical and syntactic methods for the recognition of handwritten numerals. Similar approaches can be used to segment strings of numerals [52].
- In the analysis of printed documents, syntactic methods have been tried to recognize the logical structure. Viswanathan *et al.* [43, 63] have developed a system where the layout rules for a given scientific journal are described using a YACC grammar. Green and Krishnamoorthy use a similar approach to represent the model of tables in printed documents [25]. These systems yield interesting results. In each case, however, it seems difficult to extend the models to larger categories of documents.
- The rules for representing printed music can be modelled with some syntactic approach. Couâsson [13, 12] and Baumann [3] have made some promising experiments in that area.
- Some investigations have also been made into the use of syntactic information in postal sorting, i.e. the interpretation of postal addresses [33].
- A lot of work on the analysis of diagrams uses structural symbol recognition techniques, often associated with a statistical classification step [26, 35]. More seldom, syntactic rules are used in post-processing [20].
- In engineering drawings, it has been proved that the dimensioning rules can be described by a grammar [17]. Dov Dori's group has implemented such a system for analyzing dimension sets [14, 15, 18]. In our group, Suzanne Collin implemented a system which analyzes ANSI dimensions, by using a PLEX-grammar formalism [10, 11].

- Various other prototypes have tried to represent the “syntax” of engineering drawings and to use this representation to perform various interpretation tasks. Joseph *et al.* represent the building up of higher-level structures with a YACC grammar in their ANON system [32]. It should be noted that this system also includes a higher-level frame-based knowledge representation. Bottoni *et al.* interpret a drawing in terms of form-features, using a set of rewriting rules to represent the decomposition rules for these form-features [5]. We will come back to our own experiments in § 3.

All these systems proved to be successful on the (often limited) category of documents for which they were designed. But we have in each case a specialized system, and few generalization rules emerge from this work.

Only very few systems aimed at being more general. MIRABELLE, developed in our laboratory, was designed to deal with free-hand, simple drawings [37, 41, 42]. It is a typical example showing the power of syntactic methods (it recognizes free-hand drawings of many kinds) and its limitations (it becomes increasingly harder to write an appropriate grammar when the drawings become more complicated, as is the case with engineering drawings, for instance, and when noise pervades all the data). Another recent research work is that of Pasternak in Hamburg [45, 44]. This is probably one of the first tries at designing a general mechanism for representing taxonomies of technical objects and a generic recognition methodology. The results demonstrated in the PhD thesis are quite impressive. However, it is unclear how easy it is to design new and larger knowledge bases with representation rules for more complicated structures.

### 3 Border problems

We have used structural and syntactic methods in various research prototypes for line drawings analysis. This has led us to encountering several problems. Some of these were quite expected; for instance, many methods are sensitive to noise and errors in the input data. Other problems were progressively discovered when we built up our systems.

Some of these problems with syntactic and structural methods were summarized by Sanfeliu [48]:

The formal grammar pattern representation [...] has some restrictions which have to be taken into account in practical applications:

1. The descriptive power of the string and tree grammars [...] is limited. This implies that some complex patterns can not be generated and, in other cases, it is not trivial to find the grammar. . .
2. The design of a pattern analysis system implies the consideration of hundreds and thousands of prototype patterns. Since there are not yet available inference procedures for generating the optimal grammar, sometimes it becomes almost impossible to create such a grammar. . .

The use of prototype pattern representation has also some issues which have to be taken into account for practical applications:

1. The design of a pattern analysis system implies the generation of all the prototype patterns which must be used to analyze the pattern class. If the number of patterns is high or almost not enumerable, which is not rare in some practical cases, it is not possible to generate all patterns. . .
2. Some applications (for example document image analysis) require the use of fixed data structures, [ . . . ] but contain substructures which can differ slightly on the information, position and orientation. . .

We will examine these different problems, in the practical context of line drawings analysis.

### 3.1 Sensitivity to noise and errors

Parsers as well as graph matching methods tend to be very sensitive to noise and errors in the input data. And as everybody knows, there are always noise and errors in real-life problems. . .

Many methods have been proposed to deal with this: inexact graph matching using some kind of distance [50], probabilistic relaxation [6], simulated annealing [31], etc. Messmer and Bunke have recently proposed a general algorithm for error-tolerant subgraph isomorphism [38]. We also experimented in our group with the possibility to find matches even in presence of missing or extraneous lines [27]. In order to master the complexity of the matching algorithm, we use an approach based on labeling and propagation of geometric and topological constraints [30]. The addition of labels for “missing edge” and “extraneous edge” helped solving the problem.

Syntactic parsing is also sensitive to noise. However, partial parsings can be interpreted as possible matches with lower confidence level [10]. Another solution in this context is to use attributed and stochastic grammars.

We can therefore say that in principle, error-tolerant methods are available if we consider the pattern recognition problem as a standalone module. However, real-life applications are more complex than that, and that’s where we come to the limits of existing methods:

**Interaction with the segmentation process** — If we go back to the origin of noise and errors, we find several causes. First, the usual noise associated with scanning, discretization, and image processing is of course present in line drawings analysis as it is in all scene analysis problems. But in addition, we often have to deal with information which is represented clearly enough *per se*, but which consists of several *information layers* which are superimposed in the document image. For instance, text strings can touch lines. This is not really signal noise, but as bottom-up methods usually perform some string segmentation based on the analysis of connected components, followed by vectorization by some thinning and polygonal approximation, touching characters tend to be considered as mere noise by the latter. In this case, it is not possible to only rely on the robustness to noise of the pattern recognition method. We need to actually retrieve the lost information, through some kind of interaction between segmentation and recognition. The most recent methods proposed to achieve this are more or less based on “multi-scale” vectorization:

find the evident lines and the evident characters, then try to extend the strings at the neighborhood of lines, performing in this neighborhood some higher-precision analysis of the lines.

The exact specification and performance of structural methods in such a context has still to be thoroughly studied.

**Selection of region of interest** — Graph matching methods are often proposed to solve the problem of recognizing symbols in diagrams. These methods have also proved to be quite efficient in matching an unknown symbol with the right model. But when analyzing a complete diagram, we also need an efficient method to extract potential candidate subgraphs to be matched with the models. Without such a method, the system would waste its time trying to match what is actually intersections between lines, too large areas of the diagram, or even subgraphs containing only parts of a symbol.

Here again, the problem is to let the recognition module interact correctly and efficiently with its surrounding modules. For instance, in our group, Anja Habacha proposes a method which relies on text annotations, small white loops and *a priori* size information, and which follows the lines from already recognized symbols to find the most probable next symbol locations [29, 28].

It appears clearly that these limitations can be overcome only if the pattern recognition module is defined and inserted in its *context*, and not as a standalone tool or a “black box”. For instance, syntactic pattern recognition can obviously be used to recognize various kinds of perceptual groups, but this recognition step must be properly inserted in the whole process, i.e. we need to *interact* both with the low-level image segmentation and with some higher-level analysis, which may use AI-like interpretation strategies.

### 3.2 Genericity

One of the main drawbacks of the numerous methods proposed in our area is that they have most of the time been designed and implemented for specific problems. As soon as we shift the problem to a new area, or if we try to widen up the scope in which the method can be used, we end up having unexpected problems. This is not a limitation *per se* of syntactic methods, which are intrinsically generic, but it shows that genericity in practical problems is hard to attain.

For example, we designed the CELESSTIN system, which bases its analysis on the “syntax” for mechanical engineering drawings of objects such as gearboxes. The results were very interesting, but it proved to be unexpectedly difficult to extend our methodology to a larger subset of mechanical engineering drawings [62], or to another domain such as architecture [56].

Various reasons can be mentioned for these shortcomings:

- “Bright ideas” don’t occur every other week. To our knowledge, there is no well-founded and well-specified methodology for writing down the syntax of a given technical domain. We therefore have to rely on the good ideas we can have. For instance, in CELESSTIN, we had the idea, after discussing with a mechanical engineer, to describe the syntax of the drawing by looking at what is located along the

symmetry axes, which can be detected by looking for dot-dashed lines. But where should we start the analysis for drawings of foundry models, for instance?

- The large variability of objects in a given class makes it often difficult to come up with a generic description of this class. For instance, in architecture, it already takes a large combination of components and rules to represent the design of a very simple, modular construction, that called the *primitive hut* [40, 46]. How should we then expect to find a generic syntax for a large office building?
- Syntactic representations are intrinsically hierarchical with respect to the PART-OF relationship, but are less successful in taking into account the ISA (specialization) relationship. This tends to “flatten” down the knowledge available to describe a given world. However, there is a real need for a *taxonomy* of concepts to master the entities used during the analysis process. Several research teams have proposed this kind of approach in the knowledge representation: frames are used in the ANON system [32], Dov Dori combines knowledge and interpretation process in a unified object-process model [16], Boris Pasternak proposes a double hierarchy, taking into account both the structural/geometric taxonomy and the functional taxonomy [44]. But there is still a lot of work to do to really master the relations between hierarchical representations and recognition processes.

The most thorough study, to our knowledge, of a possible *generic* methodology for drawing interpretation is the ADIK kernel proposed by Pasternak [44]. The system also includes capabilities for model learning. However, it is still unclear how much effort should be made to extend this prototype to a larger domain. The syntactic and structural rules used in the system have to deal with very low-level features, and an open question is how much a human must adapt the system in an *ad hoc* way to use it in a real-life application. Still, this work is certainly an interesting step in the right direction.

### 3.3 Model inference

Another point, which is closely related to the previous one, is that of model inference. As we have seen that it is often very difficult or even impossible to write down “from scratch” the syntax of a given world, it naturally comes to mind that it may be possible to automatically *infer* these models by looking at examples.

In graph matching, some techniques have been proposed for that [39, 49]. Grammatical inference, on the other hand, has been studied theoretically, but only some first experiments have been made on document analysis applications. Akindele proposed an inference method for the analysis of printed document structures [1]. Satoh *et al.* have a rule-based system for analyzing maps, and they include in their system some learning capabilities [49].

These are only first tries, and it is certainly a challenge to come up with robust learning capabilities. There is actually a lot of theoretical work on learning in knowledge representation. Our application domain is obviously an interesting challenge for this field.

### 3.4 Descriptive power

As noted by Sanfeliu, the descriptive power of grammars is limited. This is a general problem in knowledge representation: as there are several knowledge levels, the *context* in which some of the knowledge is represented must be taken into account, and this is not an easy task [57].

This limitation of descriptive power is especially obvious when we look at worlds such as those of engineering drawings. The draftsman uses lots of conventions, abbreviations, etc. which are understood by all those who handle the document. For instance, if several threaded holes are located along a symmetry line, there may be only one which is fully represented; the others are abbreviated (by a cross or something like that) and the reader understands from the context what is meant to be at this location. This kind of abbreviation is difficult to describe non-ambiguously with some syntactical approach. There is also a lot of information which is intrinsically of *functional* nature. We made some preliminary experiments with functional analysis in CELESSTIN IV [60], but once again this was based on a couple of “bright ideas” and we lack the general formalism to introduce this kind of knowledge in a generic way.

## 4 Conclusion

We don't pretend in this paper to come up with any radically new solution, or to prove either that structural and syntactic methods should be scrapped or that they are the panacea. We have rather tried to review their potential uses and their limitations in practical applications such as line drawings analysis of various kinds. Some of these limitations are intrinsic to the methods, and should urge people not to consider these recognition tools as simple “black boxes” which can be simply put into a processing chain, but rather as a piece in a complex puzzle, where the interactions with the other pieces must be carefully specified and designed. Some other limitations may probably be overcome by more fundamental and applied research. We think especially of genericity and automated model inference.

## Acknowledgments

Critical comments by Gérald Masini helped the author improve this paper. The work done by Suzanne Collin, Anja Habacha-Hamada and Pascal Vaxivière during their PhD theses is the basis of our group's investigation into syntactic and structural methods for line drawings analysis.

## References

1. O. T. Akindede. *Vers un système de construction automatique de modèles génériques de structures de documents*. Thèse de doctorat, Université Henri Poincaré Nancy 1, January 1995.
2. R. H. Anderson. Syntax directed recognition of hand-printed two-dimensional mathematics. In M. Klerer and J. Reinfelds, editors, *Interactive Systems for Experimental Applied Mathematics*. Academic Press, New York, 1968.
3. S. Baumann. A Simplified Attributed Graph Grammar for High-Level Music Recognition. In *Proceedings of Third International Conference on Document Analysis and Recognition, Montréal (Canada)*, pages 1080–1083, August 1995.
4. A. Belaïd and J.-P. Haton. A Syntactic Approach for Handwritten Mathematical Formula Recognition. *IEEE Transactions on PAMI*, 6(1):105–111, 1984.
5. P. Bottoni, U. Cugini, P. Mussio, C. Papetti, and M. Protti. A system for form-feature based interpretation of technical drawings. *Machine Vision and Applications*, 8:326–335, 1995.
6. H. Bunke and G. Allermann. Probabilistic Relaxation for the Interpretation of Electrical Schematics. In *International Conference on Pattern Recognition*, pages 438–440, 1981.
7. H. Bunke and B. T. Messmer. Efficient Attributed Graph Matching and its Application to Image Analysis. In C. Braccini, L. DeFloriani, and G. Vernazza, editors, *Proceedings of 8th International Conference on Image Analysis and Processing, San Remo, Italy*, volume 974 of *Lecture Notes in Computer Science*, pages 45–55, September 1995.
8. H. Bunke and A. Sanfeliu. General Remarks on Syntactic Pattern Recognition. *Pattern Recognition*, 19(4):249–254, 1986.
9. 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.
10. S. Collin. *Interprétation de la cotation des dessins techniques par analyse syntaxique*. Thèse de doctorat, Institut National Polytechnique de Lorraine, Vandœuvre-lès-Nancy, January 1992.
11. S. Collin and D. Colnet. Syntactic Analysis of Technical Drawing Dimensions. *International Journal of Pattern Recognition and Artificial Intelligence*, 8(5):1131–1148, 1994.
12. B. Coüasnon. *Segmentation et reconnaissance de documents guidées par la connaissance a priori : application aux partitions musicales*. Thèse de doctorat, Université de Rennes I, January 1996.
13. B. Coüasnon and J. Camillerapp. Using Grammars to Segment and Recognize Music Scores. In *Proceedings of IAPR Workshop on Document Analysis Systems, Kaiserslautern (Germany)*, pages 15–27, 1994.
14. D. Dori. A Syntactic/Geometric Approach to Recognition of Dimensions in Engineering Drawings. *Computer Vision, Graphics and Image Processing*, 47:271–291, 1989.
15. D. Dori. Dimensioning Analysis: Toward Automatic Understanding of Engineering Drawings. *Communications of the ACM*, 35(10):92–103, October 1992.
16. D. Dori. Representing pattern recognition-embedded systems through object-process diagrams: the case of the machine drawing understanding system. *Pattern Recognition Letters*, 16:377–384, April 1995.
17. D. Dori and A. Pnueli. The Grammar of Dimensions in Machine Drawings. *Computer Vision, Graphics and Image Processing*, 42:1–18, 1988.
18. D. Dori, Y. Velkovitch, and L. Wenyin. Object-Process Based Segmentation and Recognition of ANSI and ISO Standard Dimensioning Texts. In Kasturi and Tombre [34], pages 212–232.

19. B. Duerr, W. Haettich, H. Tropsf, and G. Winkler. A Combination of Statistical and Syntactical Pattern Recognition applied to Classification of Unconstrained Handwritten Numerals. *Pattern Recognition*, 12:189–199, 1979.
20. 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.
21. J. Feder. Plex Languages. *Information Science*, 3:225–241, 1971.
22. K. S. Fu. *Syntactic Methods in Pattern Recognition*, volume 112 of *Mathematics in Science and Engineering*. Academic Press, New York, 1974.
23. K. S. Fu, editor. *Syntactic Pattern Recognition, Applications*, volume 14 of *Communication and Cybernetics*. Springer-Verlag, 1977.
24. K. S. Fu. A Step Towards Unification of Syntactic and Statistical Pattern Recognition. *IEEE Transactions on PAMI*, 5(2):200–205, 1983.
25. E. A. Green and M. S. Krishnamoorthy. Model-Based Analysis of Printed Tables. In Kasturi and Tombre [34], pages 80–91.
26. 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.
27. A. H. Habacha. Structural Recognition of Disturbed Symbols Using Discrete Relaxation. In *Proceedings of First International Conference on Document Analysis, Saint-Malo, France*, volume 1, pages 170–178, 1991.
28. 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.
29. A. H. Habacha. *Reconnaissance de symboles techniques et analyse contextuelle de schémas*. Thèse de doctorat, Institut National Polytechnique de Lorraine, Vandœuvre-lès-Nancy, June 1993.
30. A. H. Habacha and K. Tombre. Structural Symbol Recognition. In *Proceedings of 7th Scandinavian Conference on Image Analysis, Aalborg (Denmark)*, volume 1, pages 486–493, August 1991.
31. L. Hérault. *Reseaux de neurones récurrents pour l'optimisation combinatoire; application à la théorie des graphes et à la vision par ordinateur*. Thèse de doctorat, Institut National Polytechnique de Grenoble, February 1991.
32. S. H. Joseph and T. P. Pridmore. Knowledge-Directed Interpretation of Mechanical Engineering Drawings. *IEEE Transactions on PAMI*, 14(9):928–940, September 1992.
33. R. J. N. Kalberg, G. H. Quint, and H. Scholten. Automatic Interpretation of Dutch Addresses. In *Proceedings of 11th International Conference on Pattern Recognition, Den Haag (Netherlands)*, volume 2, pages 367–370, 1992.
34. R. Kasturi and K. Tombre, editors. *Graphics Recognition — Methods and Applications*, volume 1072 of *Lecture Notes in Computer Science*. Springer-Verlag, May 1996.
35. 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.
36. S. Y. Lu and K. S. Fu. Stochastic error-correcting Syntax Analysis for Recognition of Noisy Patterns. *IEEE Transactions on Computers*, 26, 1977.
37. G. Masini and R. Mohr. MIRABELLE: A System for Structural Analysis of Drawings. *Pattern Recognition*, 16(4):363–372, 1983.
38. B. T. Messmer and H. Bunke. Efficient error-tolerant subgraph isomorphism detection. In D. Dori and A. Bruckstein, editors, *Shape, Structure and Pattern Recognition (Post-proceedings of IAPR Workshop on Syntactic and Structural Pattern Recognition, Nahariya, Israel)*. World Scientific, 1995.

39. B. T. Messmer and H. Bunke. Automatic Learning and Recognition of Graphical Symbols in Engineering Drawings. In Kasturi and Tombre [34], pages 123–134.
40. W. J. Mitchell, R. S. Liggett, S. N. Pollalis, and M. Tan. Integrating Shape Grammars and Design Analysis. In *CAAD Futures '91*, Wiesbaden, Germany, 1991.
41. R. Mohr. Precompilation of Syntactical Descriptions and Knowledge Directed Analysis of Patterns. *Pattern Recognition*, 19(4):255–266, 1986.
42. R. Mohr. A General Purpose Line Drawing Analysis System. In H. Bunke and A. Sanfeliu, editors, *Syntactic and Structural Pattern Recognition: Theory and Applications*. World Scientific, 1990.
43. G. Nagy, S. Seth, and M. Viswanathan. A Prototype Document Image Analysis System for Technical Journals. *IEEE COMPUTER Magazine*, 25(7):10–22, July 1992.
44. 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.
45. B. Pasternak, G. Gabrielides, and R. Sprengel. Knowledge-based Drawing Interpretation. In J. Hoppe, editor, *Integrated Management of Technical Documentation — The System SPRITE*, Research Reports ESPRIT, chapter 15, pages 206–222. Springer-Verlag, 1992.
46. J.-C. Paul. Architectural Shapes Modeling and Design Analysis. In *Proceedings of ACADIA '90*, pages 76–92, Cambridge, Mass. (USA), March 1990.
47. T. Pavlidis. *Structural Pattern Recognition*. Springer-Verlag, New York, 1980.
48. A. Sanfeliu. Matching Methods. In *Proceedings of 8th Scandinavian Conference on Image Analysis, Tromsø (Norway)*, pages 353–370, 1993.
49. S. Satoh, H. Mo, and M. Sakauchi. Drawing Image Understanding System with Capability of Rule Learning. In *Proceedings of 2nd International Conference on Document Analysis and Recognition, Tsukuba (Japan)*, pages 119–124, 1993.
50. L. G. Shapiro and R. Haralick. Structural Description and Inexact Matching. *IEEE Transactions on PAMI*, 3(5):504–519, 1981.
51. A. C. Shaw. A Formal Picture Description Scheme as a Basis for Picture Processing Systems. *Information and Control*, 14(1):9–52, 1969.
52. M. Shridhar and A. Badreldin. Context-directed Segmentation Algorithm for Handwritten Numeral Strings. *Image and Vision Computing*, 5(1):3–9, 1987.
53. E. Skordalakis. ECG Analysis. In H. Bunke and A. Sanfeliu, editors, *Syntactic and Structural Pattern Recognition: Theory and Applications*, chapter 18, pages 500–532. World Scientific, 1990.
54. P. Suetens, P. Fua, and A. J. Hanson. Computational Strategies for Object Recognition. *ACM Computing Surveys*, 24(1):5–61, March 1992.
55. J. W. Tai and Y. J. Liu. Chinese Character Recognition. In H. Bunke and A. Sanfeliu, editors, *Syntactic and Structural Pattern Recognition: Theory and Applications*, chapter 15, pages 416–451. World Scientific, 1990.
56. K. Tombre and J.-C. Paul. Document Analysis: A Way to Integrate Existing Paper Information in Architectural Databases. In A. Koutamanis, H. Timmermans, and I. Vermeulen, editors, *Visual Databases in Architecture*, chapter 3, pages 43–52. Avebury, 1995.
57. G. T. Toussaint. The Use of Context in Pattern Recognition. *Pattern Recognition*, 10:189–204, 1978.
58. W. H. Tsai and K. S. Fu. Attributed Grammar: A Tool for Combining Syntactic and Statistical Approaches to Pattern Recognition. *IEEE Transactions on PAMI*, 10(12):873–885, 1980.
59. J. R. Ullmann. An Algorithm for Subgraph Isomorphism. *Journal of the ACM*, 23(1):31–42, 1976.

60. P. Vaxivière. *Interprétation de dessins techniques mécaniques*. Thèse de doctorat, Institut National Polytechnique de Lorraine, Vandœuvre-lès-Nancy, February 1995.
61. P. Vaxivière and K. Tombre. Celesstin: CAD Conversion of Mechanical Drawings. *IEEE COMPUTER Magazine*, 25(7):46–54, July 1992.
62. P. Vaxivière and K. Tombre. Knowledge Organization and Interpretation Process in Engineering Drawing Interpretation. In A. L. Spitz and A. Dengel, editors, *Document Analysis Systems*, pages 307–317. World Scientific, 1995. Slightly revised version of paper presented at DAS'94, Kaiserslautern, october 1994.
63. M. Viswanathan and M. S. Krishnamoorthy. A Syntactic Approach to Document Segmentation. In R. Mohr, T. Pavlidis, and A. Sanfeliu, editors, *Structural Pattern Analysis*, volume 19 of *Series in Computer Science*, pages 197–215. World Scientific, 1990.
64. G. Wolberg. A Syntactic Omni-font Character Recognition System. *International Journal of Pattern Recognition and Artificial Intelligence*, 1(3&4):303–322, 1987.