Structure, Syntaxe et Sémantique en Interprétation de Documents Techniques

Structure, Syntax and Semantics in Technical Document Recognition

Karl Tombre* and Pascal Vaxivière°

RÉSUMÉ -

Dans cet article, nous montrons qu'il y a deux types de connaissance exploitables en interprétation de documents techniques : la connaissance structurelle et syntaxique, relative aux règles de représentation des objets, et ce que nous avons appelé la connaissance sémantique, qui fait référence à la connaissance propre aux objets du monde réel dans un contexte d'ingénierie. Nous détaillons d'abord cette distinction et étudions l'emploi des différentes sources de connaissance dans les systèmes d'interprétation de documents techniques ; ensuite, nous illustrons l'interaction de la sémantique d'une part et de la structure et syntaxe de l'autre dans le cadre de notre travail avec le système CELESSTIN.

Mots clés : connaissance, structure et syntaxe, sémantique, documents techniques, génie mécanique, interprétation

ABSTRACT -

In this paper, we show that there are two distinct kinds of knowledge in technical document analysis: structural and syntactical knowledge, which corresponds to the representation rules of objects, and what we have called semantical knowledge, which refers to knowledge about the real-world objects themselves in an engineering context. After giving an overview of what we mean by this distinction and reviewing the state of the art in the use of knowledge for technical document interpretation, we illustrate how semantics and structure/syntax interact by describing our work on the CELESSTIN system.

Keywords : knowledge, structure and syntax, semantics, technical drawings, mechanical engineering, interpretation

*INRIA Lorraine - CRIN, Campus scientifique, B.P. 239, 54506 Vandœuvre CEDEX, France

°ESSTIN, Parc Robert Bentz, 54500 Vandœuvre, France

1 Introduction — Different Levels of Knowledge

In automated analysis of technical documents, the main objective is to be able to retrieve the information described by paper documents in order to convert it to a format suitable for CAD systems. Many methods have been proposed for performing *vectorization* of technical drawings and are commercially available; however, we strongly believe that even if there is a market at this moment for vectorization systems, much higher-level analysis is necessary in order to really retrieve and make use of the information conveyed by a technical drawing. As CAD systems work with high-level entities having a technological meaning[1], it would be very useful to be able to recognize such entities in the scanned drawing, i.e. to really perform *knowledge-based interpretation* of the document.

In the last years, our group has conducted research in several directions in this area, for various kinds of documents: city maps, electricity and phone wiring schemas, mechanical engineering drawings, etc. Although we have been ourselves interested in vectorization, our primary aim is to build systems which perform high-level interpretation of the document, for a suitable conversion to a description in terms of CAD library entities. All this work has progressively given rise to more general considerations, among others about the way to exploit *a priori* knowledge [11]. It has become clearer and clearer for us that this knowledge can be of various kinds and that the interpretation strategies have to take into account at which level the document has to be recognized.

The most important distinction which has to be made is between the *representation* of objects in a drawing and the objects themselves. Technical drawings often obey strict representation standards, which determine basic entities or *structures* to use in the drawing, and a *syntax* for combining these basic structures into more elaborate ones. This structural and syntactical knowledge can be used to analyze the drawing; however, we must not forget that in this way we can only "understand" it from a representation point of view. Nearly all existing interpretation systems are limited to this kind of knowledge.

But in the real world of engineering, another large set of expertise is available, namely that of the engineers themselves, who deal with the real objects, in their real environment. This expertise goes from the basic laws of physics to well-known technologies for manufacturing, assembling, disassembling and using the objects. In the following, we call this kind of knowledge the *semantics* of technical drawings, although we are aware that this terminology may be a bit controversial.

This distinction does not imply that *any* interpretation system has to take into account all these levels of knowledge. In some cases, the analysis can be completely based on the structure of the drawing. In other cases, representation rules are more complex, but there are standards for the way to represent various entities and technical systems. This leads to a more or less syntactical approach: drawing rules can be represented as a two-dimensional grammar or as a set of production rules leading to the progressive assembly of complex structures from simpler ones.

But in the most general cases, even this is far from being enough; structural entities themselves, even if they are perfectly legal according to the representation rules, may have no direct technical interpretation; they only make sense if they correspond to valid objects or systems, in the context of the appropriate technical domain. Therefore, a useful interpretation system in these cases must take into account the knowledge about the technical domain itself, not only about the representation standards. This is what we have called the semantics of technical documents.

In this paper, we try to make more explicit the specificities of these different kinds of knowledge and the interpretation results which can be achieved in each case. We start with a review of different systems and methods in the light of this "classification" of knowledge. This review does not pretend to be exhaustive with respect to existing methods, but we want to underline that the problems to be solved in document interpretation systems cannot always be presented in a linear way if the kind of knowledge is not explicited; we must take into account the kind of knowledge

---

[1] A vector in itself has no "semantics" in a mechanical engineering drawing, for instance!

used and the level of "expertise" we want to put into the system. After this review, we present some of our ongoing work, which illustrates what kind of structural, syntactical and semantic knowledge can be used.

2 Structure and syntax

Actually, anyone interested in writing interpretation systems for technical documents will probably have to deal with their structure, at least the lowest-level structure: it is commonly accepted that the basic structural entity in graphics is the *line vector*; hence the numerous vectorization techniques developed in the field. Vectorization is already a structural interpretation, although a very low-level one, as the only kind of knowledge used is that a technical document is to a large extent made of line graphics; hence one can assume that some kind of raster-to-vector conversion is a good way to describe the document in a more pertinent way as that of being a bitmap of black and white pixels.

However, as previously noted, a line vector in itself has no intrinsic meaning in most cases, i.e. it does not convey any technological information *per se*. In addition, the conversion of technical documents into a set of vectors can be interesting if the user wants to perform vector edition on the retrieved document but is insufficient if his aim is to use a CAD system at the level of its library of technical components, as such a system deals with much higher-level entities. Therefore, we believe that pure vectorization systems are only really sufficient in two cases:

1. when the stated purpose of the user is to do vector (or graphics) edition and only that on the document;
2. for the purpose of document *archiving*, although vectorization is competing in this field with good image compression algorithms.

To enhance the quality and structural level of the vectorization process, several other basic structures are sometimes looked for: cross-hatched areas, dotted lines, circular arcs and circles, etc. Finding such structures yields a richer description of the document, but this description is still basically at the same abstraction level as vectors, i.e. only well suited for *graphics edition* of the interpretation result.

Other structures may also be looked for. For instance, several text–graphics segmentation methods adapted to technical documents have been proposed, so that text parts can be processed apart. Text can thus be considered as a separate *layer*; more generally, a technical document can be seen as the superposition of several layers, each containing a specific kind of information. One possible interpretation strategy is then to extract these layers in the most appropriate order and to analyze them individually, combining the results of interpretation of several layers when necessary. This is typically a completely structural approach to document analysis. In addition to the text parts, cross-hatching can be considered as a separate layer, which can be detected as such and then removed from the image of the document.

This approach was chosen in our group in the context of interpretation of French city maps, where buildings are cross-hatched polygons, which are enclosed in larger polygons representing the parcels of land, which in turn are bordered by streets, etc. All the entities are represented by simple structures; thus, the induced interpretation process can be based on the localization and recognition of these structures in the drawing. The choice in this case was to represent the knowledge as a procedural network, i.e. a graph where each structural entity model is connected to its neighbours by topographic relationships and where a procedure at each node looks for matches between the model and actual parts of the vectorized map [1].

As vectorization tends to smooth out small details, it may also be necessary to extract a layer of small elements directly on the binary image, before vectorization. Of course, we then must use the domain knowledge to know which small elements to look for; this can be arrowheads in the case of dimensions in engineering drawings, small symbols for vegetation in maps, etc.

Another typical example of analyzing the document in terms of higher-level structures, based on domain knowledge, is that of the minimum closed block drawn in thick lines, which we defined as the basic structural component of mechanical engineering drawings and which is the basic "building stone" of the further interpretation phases of the CELESSTIN system (cf. § 4).

Actually, this last case illustrates a well-known property: when some basic structures are defined, it is often possible to create higher-level ones by using "syntactical" composition rules. Thus, *syntax* is tightly related to structure, as syntactical rules lead to the definition of higher-level structures. Technical documents very often follow strict representation rules, which give a standardized way of drawing various elements. These standards can then be represented by a grammar or by a set of production rules, thus enabling the recognition of complex structures (non-terminals) from the most basic ones (terminals). The "knowledge engineering" process revolves around the question: "When reading a drawing, how do you decompose it into more and more elementary structures, or inversely, what are the representation rules followed by the draughtsman for creating higher-level objects from basic structures?".

Several systems developed in the field of technical document analysis have this kind of formalization:

- Dov Dori has shown that dimensioning rules can be described by a grammar [4]; the dimensions can therefore be recognized by progressive grouping of basic structures such as arrowheads, thin lines and text [2].
- We have also tested this approach: the assembly of blocks along an axis leads to the recognition of entities such as screws, shafts, and from them ball bearings, gears, etc. A set of simple assembly rules allows the progressive assembly of larger objects from the minimal closed block drawn in thick line [13].
- ANON [6] is based on a structural description of engineering drawings in terms of frames representing different components (lines, curves, words, dimensions...) and relations between these components. The interpretation itself follows strategy rules written in the yacc syntax; the parsing allows the recognition of entities such as dimensions or broken lines.
- MIRABELLE is aimed at recognizing hand-drawn line figures [8] [9]. It uses geometric primitives and a description language for decomposing larger objects into subshapes and finally into a combination of several primitives. The assembly of subshapes into more complex shapes uses combination and topographic operators. Recognition is achieved by purely syntactic parsing.

In fact, structural and syntactical interpretation is probably the most common approach in technical document analysis. All kinds of wiring schemes in electricity or electronics are also well suited to a structural representation of knowledge. They are basically made of a set of symbols connected by a lines. Symbol recognition (the problem of matching a symbol with a model) can be seen as a structural pattern recognition problem [7] [5]. However, as we will see in § 3, higher-level interpretation, for detection of anomalies for instance, requires the introduction of domain knowledge.

Although such systems give interesting results, they have limitations. A purely syntactic description of shapes may be convenient for simple figures but is often completely inadequate for complete technical drawings, as such a description only captures a part of the available knowledge. In fact, in a general interpretation system, syntactic parsing should rather be considered as a tool among others, which may be useful for recognizing components of the drawing (such as the dimensioning for instance), while being driven by a higher-level knowledge-based system. We discuss in the next section what this higher-level knowledge can be.

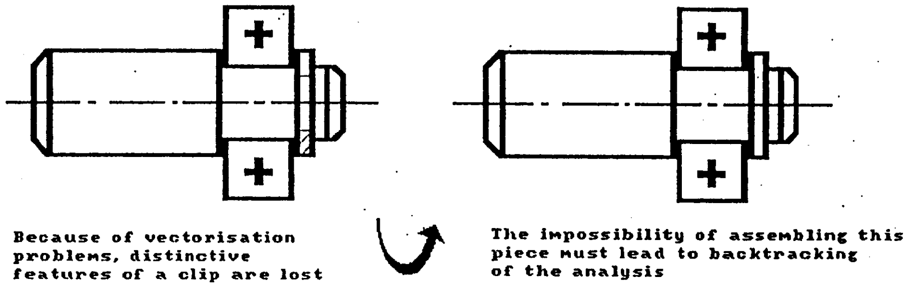
3 Semantics

Whereas all the knowledge rules shown until now apply to the drawing, i.e. the symbolic representation of a real-world object, semantic rules deal with knowledge about the object itself. Semantics is probably the most difficult part of knowledge to formalize, but also the most rewarding. To describe this kind of knowledge, the questions which have to be asked are: "How does it work, how can I manufacture the object, what use can it have, when does a schema correspond to a consistent and valid technical solution?". The drawing must not only be a syntactically valid assembly of structural entities but it must describe a technically meaningful and working system.

Very few systems include this kind of knowledge. The work of Murase and Wakahara [10] deals with symbols found in flowcharts and in logic circuit diagrams. Simple semantic rules are used to recover from erroneous symbol recognition; for instance, basic knowledge about the "meaning" of different flowchart symbols leads to rejection of some inconsistent combinations, such as a *terminal* symbol located at a branch of a process flow. These inconsistencies are defined from knowledge about valid and invalid algorithms.

Benjamin *et al.* use several levels of knowledge in their system for interpretation of telephone outside plant drawings [3]: syntactical and structural rules describe the various kinds of symbols and relations between them, whereas higher-level rules describe the "meaning" of these symbols and the consistencies which have to exist between the corresponding entities in the real-world application.

In engineering drawings, it would be very interesting to be able to analyze the drawing from its functional scheme or from the understanding of how an object can be assembled or disassembled. For instance, Fig. 1 illustrates a case where thin lines lost during vectorization can be guessed after the failure of a semantic analysis: as the object without these features *cannot* be assembled, the analysis can backtrack and infer the existence of the missing items.



Because of vectorisation problems, distinctive features of a clip are lost

The impossibility of assembling this piece must lead to backtracking of the analysis

Figure 1: The importance of semantics in drawing analysis.

For some kinds of technical documents, really useful interpretation can only be achieved at this level, as representation rules are not followed rigorously and more generally the draughtsman, the engineer and the manufacturer share a common technical expertise which enables them to understand drawings even when they are syntactically ambiguous.

4 CELESSTIN

We will illustrate the basic ideas described in the previous sections by describing our work on the CELESSTIN system. We have been working for four years on the recognition and conversion to CAD of mechanical engineering drawings, with a group of circa 15 students each year, at the ESSTIN engineering school[2]. The pedagogical aim is to build a reusable software, each new student group starting with the result of the previous year's project. Technically, the aim is to design a

---

[2] This work was partially financed by a grant from IBM France to ESSTIN.

system having as input a scanned mechanical engineering drawing and as output its analysis in terms of entities described in the CAD library[3].

After having built a decent vectorization module [12], first experiments in matching between a model and the vectors showed us that the vector as the basic structural element is too simple and too local for achieving good results in interpretation of the drawing. For example, we separated the lines into two classes: thick lines and thin lines, the latter being further divided into several classes, such as dashed lines, axis lines and cross-hatching. This approach allowed us to solve some of the ambiguities of the drawing, by use of simple syntactical rules. Cross-hatching can represent a section through a solid part or the state of a surface; the draughtsman may limit the cross-hatching to a part of the surface when there is no possible confusion; however, the hatching actually refers to the whole surface. Hence, the following kind of rule is given:

> **Rule:** *When a cross-hatching family is recognized, it is extended to the whole thick-line contour.*

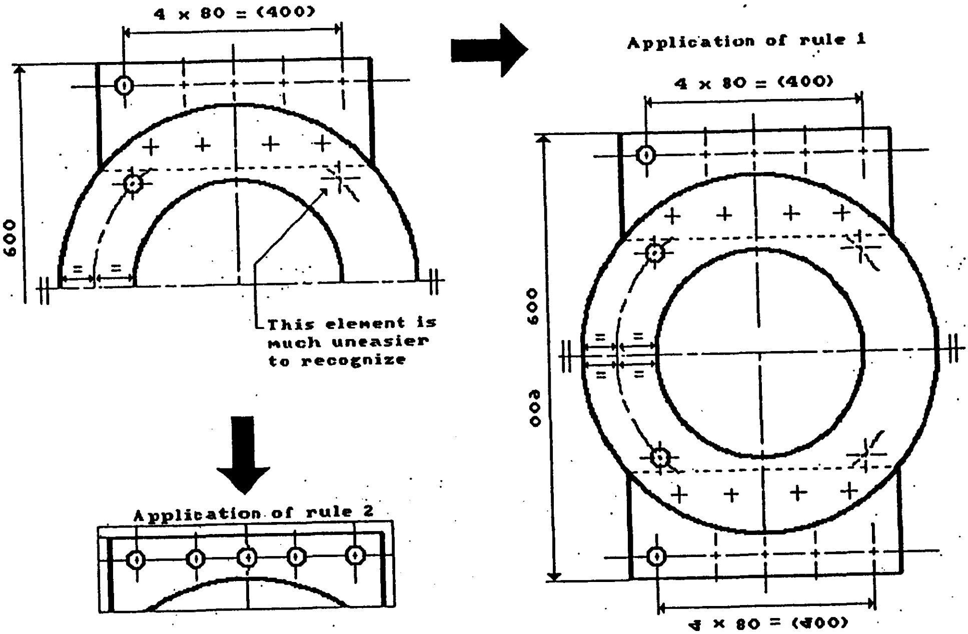


Figure 2: Symmetries around axis lines.

However, these kinds of rules rapidly become very complex, as the semantics associated with the technical drawing is not always easy to represent without knowledge about the context or with only the limited context of the combination of neighboring vectors. We then moved on to basing our analysis on the *axis line*, which symbolizes symmetry and conveys more information. Fig. 2 illustrates two rules applying to this kind of line:

> **Rule 1:** *When a drawing stops on an axis line bearing the || sign at both extremities, the figure is symmetric with respect to this axis line.*
>
> This symmetry should not usually apply to dimensions.
>
> **Rule 2:** *In an overview drawing, the repetition of the same technical element is represented by its symmetry axis or its center.*
>
> In the example given in Fig. 2, the drilling holes are represented by a thin-line cross showing the center of the hole.

---

[3] We use the CATIA CAD system from Dassault Systemes.

The latter rule is typically related to the technical meaning of the object more than to symmetry relations: it is "evident" that this kind of cover is held by screws and that the crosses represent the location of the drillings.

In addition, by trying to solve ambiguities such as the one illustrated by Fig. 3, we evidenced that in a technical drawing, symmetry lines actually represent a "near-symmetry".

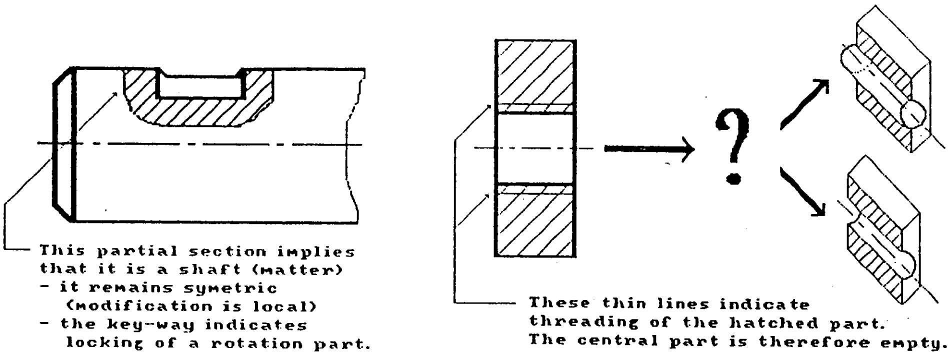


Figure 3: Typical problems with lines.

Although this yielded interesting results, we had to conclude that a line vector in itself is far too weak as structure element in order to be able to associate coherent semantics to it. Its context is too local and the neighboring relation between two segments does not convey enough information. For instance, it did not allow us to represent a shaft, which is an element with lots of variations around a basic concept, as a single entity referring to a precise object in the CAD library.

We therefore looked for a higher-level element, containing more appropriate information than the line vector. One of our ideas was that a technical drawing represents a set of components ordered in such a way that the object as a whole can be used for precise technical aims. One part of the semantics is therefore that it must be possible to assemble and disassemble a machine.

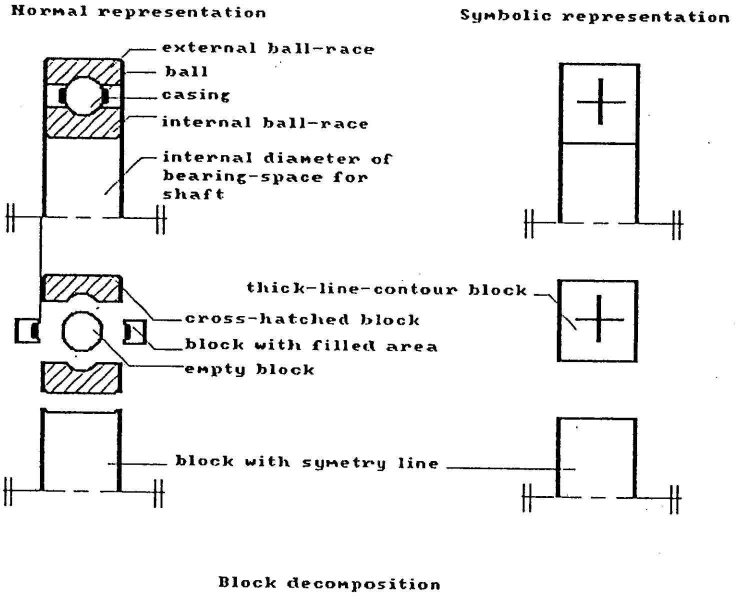


Figure 4: The example of a ball bearing.

Actually, what is a technical component? It is a setup having a mechanical functionality;

it may be used for sliding, rolling or locking. In some cases, people are not interested in the individual parts of such a component; in other cases, these details are important and must be represented. For instance, ball bearings are sometimes represented in a completely symbolic way, when the draughtsman concentrates on their position in the whole assembly; on the contrary, if it is important to underline their functionality, all the details of the bearings will be drawn (see Fig. 4).

By considering the assembly of a the real-world object *ball bearing*, we find that it is the ordering of several materials: a ring, balls and another ring for instance; each *block* of matter is represented by a thick-line contour, with a thin-line *context* representing the characteristics of the material: cross-hatching for a section of matter, white for emptiness of for matter which should not be sectioned (balls, shafts...), dot-dashed line illustrating a symmetry.

Thus, this input of semantical knowledge led us to the definition of a higher-level structure, the thick-line minimal closed contour, called the *block* [13]; from there, we can define new neighborhood relations between blocks. Once again, knowledge about the semantics of mechanical engineering led us to associate a new syntax to this new structure, thus allowing for interpretations such as that of Fig. 5: it is the white, empty block which is threaded, not the cross-hatched parts, even if they actually contain the threading lines.

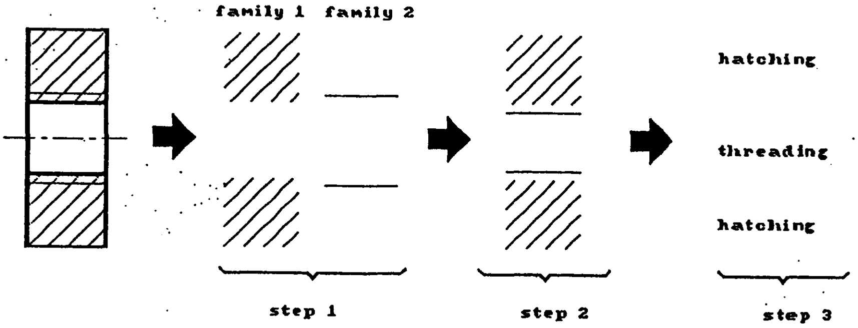


Figure 5: Syntactic analysis on the block structure.

From this new structure and syntax, we can further expand our knowledge about the drawing, by parsing along the axis lines, with the closed blocks as basic structural elements, thus allowing the recognition of entities such as ball bearings, screws, clips, pins, gear wheels and shafts, as we published in [13]. With appropriate syntactical rules, we were able to put forward hypotheses about the nature of a block's direct neighbours or of its symmetric with respect to a dot-dashed line.

But this kind of structuring became too weak when it was necessary to apply rules such as:

- any mechanical device can be disassembled;
- if a shaft has been recognized, this implies a motion, and hence a coherent cinematic scheme.

In this case, we had to create a new structure, which we called the "entity", defined as the set of blocks symmetric with respect to a dot-dashed line, having the same shape and the same thin-line context. Thus, the ball bearing of Fig. 4 becomes a unique entity, which can be manipulated as such by the reasoning mechanism.

5 Conclusion

As shown by our example, the use of the semantics level of knowledge in technical document understanding is not just a top-level shell over lower-level knowledge. It must rather be seen as an *interaction* between some structure and syntax on the one hand, and the associated semantics on the other hand, which leads to introduction of higher-level structures, of a new syntax on these structures, of stronger semantics associated to this, and so on. This was illustrated by our progression from vectors to axis lines, then to "blocks" and finally to "entities". Of course, work is continuing on this system and hence, we may come once again to the necessity of adding a new structural level, because of a higher-level formalization of the syntax and semantics.

We do not pretend to give a thorough solution to the problem of knowledge representation in technical document interpretation, but we hope that this paper gives some hints about possible directions in the difficult task of building *useful* knowledge models and analysis systems.

References

[1] D. Antoine. Interprétation des plans cadastraux français à partir d'un modèle. Thèse de doctorat, Institut National Polytechnique de Lorraine, Vandœuvre-lès-Nancy, 1991.

[2] D. Antoine, S. Collin, and K. Tombre. Analysis of Technical Documents: The REDRAW System. In *Pre-proceedings of IAPR Workshop on Syntactic and Structural Pattern Recognition, Murray Hill, NJ (USA)*, pages 1–20, 1990.

[3] D. Benjamin, O. Forgues, E. Gulko, J.B. Massicotte, and C. Meubus. The Use of High-Level Knowledge for Enhanced Entry of Engineering Drawings. In *Proceedings of 9th International Conference on Pattern Recognition, Rome*, pages 119–124, 1988.

[4] D. Dori. A Syntactic/Geometric Approach to Recognition of Dimensions in Engineering Drawings. *Computer Vision, Graphics and Image Processing*, 47:271–291, 1989.

[5] A.H. Habacha and K. Tombre. Structural Symbol Recognition. In *The 7'th Scandinavian Conference on Image Analysis*, Aalborg University, Denmark, August 1991.

[6] S.H. Joseph and T.P. Pridmore. A System for the Interpretation of Images of Graphics. In *Pre-proceedings of IAPR Workshop on Syntactic and Structural Pattern Recognition, Murray Hill, NJ (USA)*, pages 154–164, 1990.

[7] S.W. Lee, J.H. Kim, and F.C.A. Groen. Translation-, Rotation-, and Scale-Invariant Recognition of Hand-Drawn Electrical Circuit Symbols with Attributed Graph Matching. In *Pre-proceedings of IAPR Workshop on Syntactic and Structural Pattern Recognition, Murray Hill, NJ (USA)*, pages 273–292, 1990.

[8] G. Masini and R. Mohr. MIRABELLE: A System for Structural Analysis of Drawings. *Pattern Recognition*, 16(4):363–372, 1983.

[9] R. Mohr. A General Purpose Line Drawing Analysis System. In H. Bunke, editor, *Syntactical and Structural Pattern Analysis*. Springer Verlag, 1987.

[10] H. Murase and T. Wakahara. Online Hand-sketched Figure Recognition. *Pattern Recognition*, 19(2):147–160, 1986.

[11] K. Tombre and D. Antoine. Analysis of Technical Documents using a priori Knowledge. In R. Mohr, T. Pavlidis, and A. Sanfeliu, editors, *Structural Pattern Analysis*, volume 19 of *Series in Computer Science*, pages 229–241. World Scientific, 1990.

[12] K. Tombre and P. Vaxivière. Paper to CAD conversion for mechanical engineering drawings. Rapport interne, Centre de Recherche en Informatique de Nancy, Vandœuvre-lès-Nancy, 1989.

[13] P. Vaxivière and K. Tombre. Interpretation of Mechanical Engineering Drawings for Paper–CAD Conversion. In *Proceedings of IAPR Workshop on Machine Vision Applications, Tokyo*, pages 203–206, 1990.