Ten Years of Research in the Analysis of Graphics Documents: Achievements and Open Problems

Karl Tombre

Graphics Documents: Achievements and Open Problems

Karl Tombre

Loria–Inria
Campus scientifique, B.P. 239
54506 Vandœuvre-lès-Nancy CEDEX, France
`http://www.loria.fr/~tombre/`

Abstract: *Our research group has been investigating various aspects of graphics recognition techniques for more than ten years. We have worked on map analysis, symbol recognition, dimension analysis and the conversion of engineering drawings to* CAD *models. Lately, we are also conducting research on the interpretation of architectural drawings. In addition, we have built up a software platform of generic tool for graphics document image analysis, and we have participated in many international activities around the topic of graphics recognition.*

In this paper, we present some of our achievements and results from these ten years, and we propose a number of open problems, which we think are good challenges in the coming years, for ourselves and for other teams.

1. INTRODUCTION

Document image analysis is an exciting field of pattern recognition, where lots of activities are going on [7, 8, 23]. A special area in document image analysis is that of *graphics recognition* [18, 30] which includes work on raster-to-vector techniques, recognition of graphical primitives, analysis and interpretation of engineering drawings, logic diagrams, maps, diagrams, charts, etc.

Commercial graphics recognition system do exist, and they usually offer satisfactory implementations of low- and medium-level tasks, such as vectorization, text/graphics separation, and symbol recognition through template matching techniques. However, their performances are limited in different areas [31]:

- the text/graphics separation techniques perform very poorly as soon as some text touches graphics;
- a conversion system limited to vectorization is of limited interest, as a pure vector representation is too poor, in many cases;
- although these systems perform recognition of some graphics features and symbols, most of the time they use too low abstraction levels.

Due to these limitations, many systems include a vector editor for manual correction of the results of the raster-to-vector conversion. This manual editing has often been deemed too time-consuming by customers of commercial systems. To overcome this, another class of commercial solutions has been proposed. These are semi-automated systems, where the user, guided by the scanned raster image of the document and by a set of simple tools, such as line tracking, inputs the CAD model by himself/herself. The amount of domain-dependent knowledge in these systems is at best very limited. They usually stop short of applying such knowledge, which could have raised them beyond the basic recognition level.

This is the context in which our research group has been active for more than ten years. We have worked on map analysis, symbol recognition, dimension analysis and the conversion of engineering drawings to CAD models. Lately, we are also conducting research on the interpretation of architectural drawings. In addition, we have built up a software platform of generic tool for graphics document image analysis, and we have participated in many international activities around the topic of graphics recognition. In this paper, we present some of our achievements and results from these ten years, and we propose a number of open problems, which we think are good challenges in the coming years, for ourselves and for other teams.

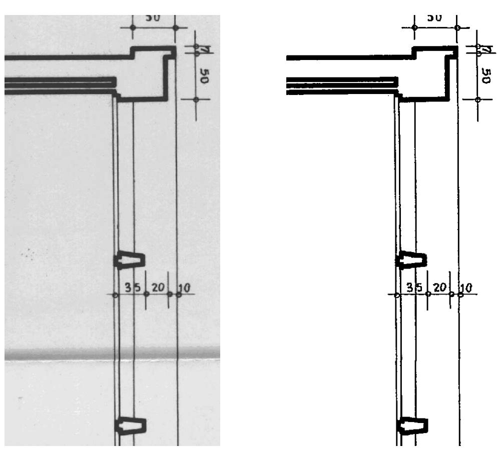
2. LOW-LEVEL TOOLS

The first stages of graphics recognition are typical image processing problems, with the specificities of graphics documents. We give a list here of our contributions, developments and choices in this area:

Binarization — Many scanners come with some built-in binarization, either in hardware, or in the accompanying driver software. For clean documents, this is sufficient, and binarization is not really a problem. But in some graphics recognition problems, we have to deal with blueprints, large drawings which have been folded and stored away for a long period, etc. In that case, we may have to use some adaptive binarization method, computationally costlier than the built-in tools, but necessary to avoid false objects due to folds, or to the merging of lines close to each other on a poor-quality blueprint.

Basically, adaptive thresholding methods can be divided into two classes: methods based on the computation of a local threshold from measures such as jects. Basing ourselves on Trier and Jain's evaluation [33], we implemented one method from each of these two categories: Niblack's method [22] with Yanowitz and Bruckstein's post-processing step [40] for the local average approach, and Trier and Taxt's improvement on a method originally proposed by White and Rohrer for the contour approach [34]. Probably because of the special nature of graphical documents, the latter yields much better results. We therefore chose to implement a variation of Trier and Taxt's method. Our main changes to their algorithm are that instead of using *ad-hoc* filters such as the Sobel gradient, we use Gaussian filtering, which has become standard in edge detection. Details of the method are given in [29].

There are three thresholds in this method; our experiments show that the most important is the width of the Gaussian used, i.e. σ. It must be chosen such that the convolution masks have approximately the same width as the thickest lines in the image. As we want to have robust methods, let us stress that whenever the document is clean, it is better to use the built-in binarization coming with the scanner software! The other method is only useful when the degradations make this binarization useless. Fig. 1 shows a typical graphics image, with a fold and three thick lines very close to each other, and its binarization with this method.



(a) Original image (b) Binarization

Figure 1. Example of binarization using Trier and Taxt's method.

Text/graphics segmentation — Most published methods on text/graphics separation are variations on the principle of analyzing the connected components. One of the best explained methods in literature is that of Fletcher and Kasturi [14]. We therefore strongly suggest that, instead of spending a lot of time on reinventing some new method, which most of the time does not give any real improvements on the known ties of the documents to be processed.

In our case, we chose to implement this method with minor adaptations. As they designed their method for mixed text/graphics documents, some of the thresholds must be adapted to the new situation. We also added an absolute threshold on the size of a text component. Thus, we end up having three thresholds, but their interpretation is straightforward, and they have proven to be very stable for a family of graphics documents. As proposed by Fletcher and Kasturi, this is followed by string grouping using the Hough transform.

To perform further segmentation of the graphics layer, we separate thin lines from thick lines using morphological filtering:

- according to the limit we want to set between thin and thick lines, set a size n and perform an erosion $J = I \ominus B_n$, B_n being in our case a $(2n+1) \times (2n+1)$ square;
- retrieve the thick lines through partial geodesic reconstruction ($n+1$ iterations): for $i = 1 \ldots n+1$

$$K_0 = J; K_i = (K_{i-1} \oplus B_1) \cap I$$

- This yields two images $I_{thick} = K_{n+1}$ and $I_{thin} = I - I_{thick}$.

Fig. 2 illustrates the three corresponding layers after segmentation of an architectural drawing.

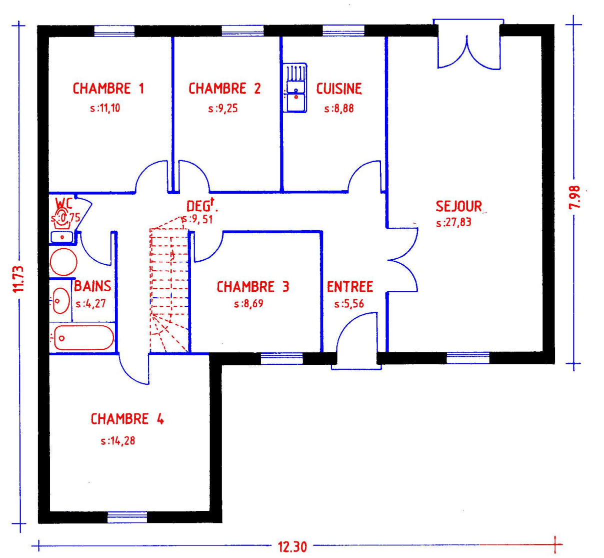


Figure 2. Segmentation: Thick lines in black, thin lines in blue, text in red.

3. VECTORIZATION

Vectorization, i.e. raster-to-graphics conversion, has been given a lot of attention, and many algorithms have been state of the art on that topic is quite paradoxical. Many of the proposed methods do work satisfactorily, but none of them is perfect. Most methods are based on some kind of skeletonization, followed by some kind of polygonal approximation. Other methods are also available, including various sparse-pixel approaches, run-based algorithms and approaches directly working on the image or on the distance transform. Although these methods yield good results, they all have their specific weaknesses, so that we cannot say that perfect raster-to-vector conversion is available. However, the quality is good enough for using the result as input data for higher-level recognition and analysis methods, so we tend to think that this field has matured. Various interesting post-processing steps have also been proposed, to enhance the quality of the vector description.

We have ourselves experimented with several algorithms [5, 37], having interesting properties and yielding good results. But because of the stableness and robustness criteria, we have ended up coming back to what most people use: skeletonization followed by polygonal approximation. This stems from the fact that this approach is the one requiring the lowest number of parameters to be set; with the other approaches, we often had to fine-tune our parameters for each new family of documents.

We have chosen to use a skeleton based on the 3–4 distance transform, by implementing a method first proposed by Arcelli and Sanniti di Baja [6, 11]. This is followed by implementations of the following steps:

- post-processing of the vectors to better position the junction points, using a method proposed by Janssen [17];
- recognition of arcs, by adapting Rosin and West's method [26] to our vectorization;
- addition of geometric constraints to the vectorization and arc recognition, using an method proposed by Röösli and Monagan [25].

4. SYMBOL RECOGNITION

The recognition of graphical symbols is a well-known problem, for which many methods have been proposed [9]. Many methods have been proposed to deal with this: inexact graph matching using some kind of distance, probabilistic relaxation, simulated annealing, etc.

Messmer and Bunke have proposed a general algorithm for error-tolerant subgraph isomorphism [20]. We also experimented in our group with the possibility to find matches even in presence of missing or extraneous lines [15]. In order to master the complexity of the matching algorithm, we use an approach based on labeling and propagation of geometric and topological constraints [16]. The addition of labels for "missing edge" and "extraneous edge" helped solving the problem.

But in our quest for a good recognition method, we felt the need for *flexibility* and *genericity*. As architectural drafting as doors or windows are represented. We therefore *cannot* build an *a priori* set of models and decide that these are the only symbols we will recognize. We must be able to incrementally add new models to the knowledge base, with minimal computational overhead at recognition time.

A first system which inspired us was that of Pasternak [24]. In his ADIK kernel system, he uses graphical specifications of the symbols, based on a number of predicates and on constraints between parts of the same geometric composition object. To take into account the fact that a model can match with an object in the document with different poses, he also introduces the concept of views or aspects of each basic feature. The whole knowledge base is represented as a structural/geometric taxonomy, to allow for efficient specialization of symbols.

Keeping some of these ideas, especially the hierarchical modelling and the constraint-based description of models, we turned to another method for the efficient management of the set of models. Continuing their work, Messmer and Bunke [21] proposed a method which allows for model pre-compilation through the use of a network, where all model descriptions are gathered at once; the features are the input to this network and "trickle down" until one of the terminal nodes—i.e. one of the model symbols—is activated.

This work is based on graph isomorphism; in our case, we use constraint propagation, more or less in Pasternak's spirit, but we adapt the network concept to these constraints [4]. The main idea of our system is to build a network of all constraints by progressive learning, based on a syntactic description of the model symbols and factorization of common constraints. After the learning phase, recognition is performed by having all graphics features (segments and arcs) "trickle" through this network (Fig. 3).

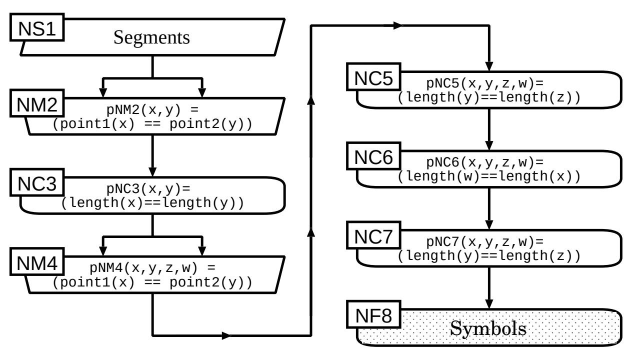
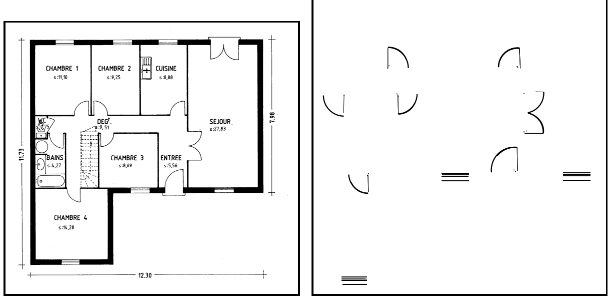


Figure 3. Principle of the constraint network.

Fig. 4 illustrates the result of this symbol recognition process.

5. ENGINEERING DRAWINGS

From 1989 to 1991, our group worked on Celesstin, an integrated, blackboard-based prototype system which converts drawings into a CAD description [35]. The first ver-



(a) A drawing (b) Recognized symbols

Figure 4. Result of symbol recognition.

sions of this system were essentially based on structure and syntax to recognize entities such as shafts, screws, ball bearings or gears on a single view of a mechanical device. The system decomposes the vectorized document into a set of blocks having contextual attributes (hatching, threading, etc.) and analyzes these blocks by focussing on technical elements located along the axis lines. In the last version, Celesstin IV, we experimented with semantic knowledge rules. By focussing on a specific area of mechanical engineering, we were able to show that it is possible to analyze a single view of a drawing at the level of technological functionalities. We designed two "experts", one focussing on *disassembling*, based on the assumption that it *must* be possible to disassemble a mechanical setup, and the other dealing with the *kinematics* of the whole setup. The kinematics expert determines the functionalities of various entities from their behavior when a rotation motion is applied around the identified axes in the drawing [36].

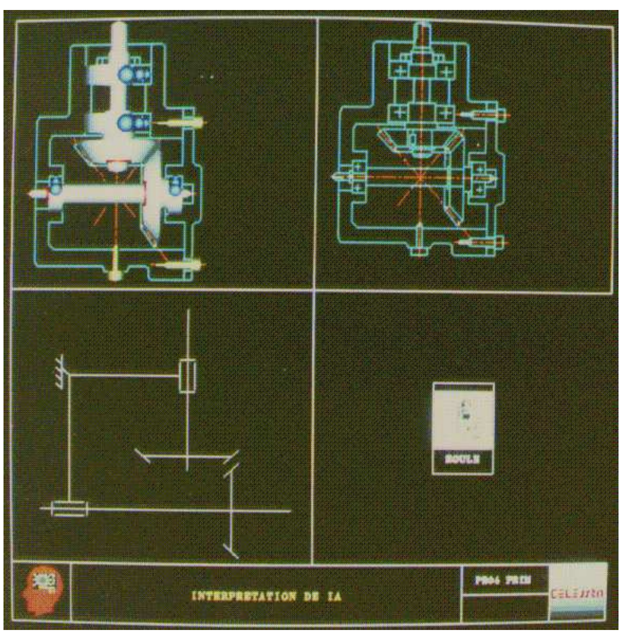


Figure 5. Functional analysis in Celesstin.

Fig. 5 illustrates the results of this functional analysis. Al-
ing all possible functional interpretations, we believe that this work suggests a possible methodology for extracting functional information from technical drawings.

We also worked on the analysis of dimensioning, which can be described by a grammar [12]. Our system analyzes ANSI dimensions, by using a PLEX-grammar formalism [10].

But the analysis of engineering drawings would not be complete if it was limited to 2-D. We should also be able to reconstruct a 3-D model from different views. This need arises not only when starting from a paper drawing, but also in the numerous cases where the available data are a set of geometrical 2-D views constructed by some computer-based drafting system, that needs to be converted into a real 3-D CAD model.

Many purely geometric methods have been designed for combining several orthogonal views into a 3-D model [13]. We chose to implement a variation of the algorithm known as the "fleshing out projections" concept, which was first formalized by Wesley and Markowsky [39]. We also investigated the possibility to add symbolic information to this process [2].

6. ARCHITECTURAL DRAWINGS

Surprisingly, few teams have been dealing with architectural drawings. There are probably two main reasons for that. Firstly, there has been less demand from the application field than in other domains for systems capable of analyzing paper drawings and yielding a 2-D or 3-D CAD description of the represented building. Secondly, architectural design is more or less at the crossroads between engineering and art, which makes precise analysis and reconstruction more difficult.

But these last years, our team has started investigations in this field. A first incentive was internal: we have defined a research project for the next years with the scientific objective of studying the problems of image analysis and computer graphics in the context of virtual reality and augmented reality applications, our particular application domain being that of architectural and urban environments. For instance, in order to compute augmented reality images of such environments, we need input data. Architectural drawings have the potential of yielding a lot of such data about buildings, as well those which already exist as those which don't exist anymore (historical simulations) or which don't exit yet (urban planning, architectural simulation, etc.). External needs have given additional incentives: various applications require large amounts of architectural and urban data for simulation and planning. A typical example is that of mobile telecommunications.

In many senses, architectural drawings are similar to engineering drawings, as they typically represent orthogonal projections of the walls and construction elements. But a first difference is that during architectural design and construction, a number of different representation scales are used [32]:

its general outlines and major characteristics. The sketch's precision level is very variable from one architect to the other, but generally, its highly semantic content makes it understandable only by the designer, and not at all by a hypothetic automated process on a computer. Actually, the sketch can be said to belong more to "art" than to "engineering"!

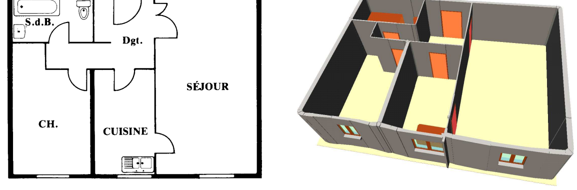
- On the contrary, in the *design phase*, a set of drawings is made up, going from the least to the most detailed, and including plans, elevations and cross sections. The building's architecture appears then as an arrangement of volumes, some solid and some open, of opaque and transparent surfaces, but the documents also illustrate the layout of passages, the choice of technology types, the architectural composition of the façades. The most precise of these drawings have a scale of 1/50 and contain sufficiently precise information for a 3-D representation to be built.
- In the last phase, the architect and the specialized engineers (construction technologies, thermo-analysis, acoustics, lighting) design the *detailed workplans* of the buildings, typically at a scale of 1/20. These documents give the exact dimensions of the building and specify the construction techniques and the materials to be used. But for the purpose of 3-D reconstruction, they contain so many details that the resulting 3-D representation would rapidly become geometrically too complex to be useful.

In the present work, we therefore have chosen to analyze the intermediate design phase drawings [3]. An additional advantage of these drawings is that as they correspond to those submitted to the authorities for the building permission, they are widely available, before, during and after the construction itself. Ultimately, we want to build a 3-D model by combining the different views contained in such drawings. But in the first phase of our work, we have concentrated on the top view, which is richest in semantic information.

After the usual low- and intermediate-level image processing and vectorization phases, we designed a spatial analysis method, based on the idea that architectural design is about arranging spaces. This led us to the concept of analyzing the large white "loops", which are candidates for representing rooms, and of propagating the analysis from these rooms to the walls.

Fig. 6 illustrates the kind of reconstruction we get currently.

As no process is 100% perfect, we also integrate all these tools in a man–machine interface, to allow for easy interaction with the user at any step of the analysis [1]. This work is still in progress. We are currently adding dimension analysis, a module for stairway and tiling recognition [19], and a matching process between the different floor plans, to "pile up" the levels of the building. In the next two years, we also plan to integrate matching with the



(a) An architectural drawing (top view)

(b) Result of spatial analysis

Figure 6. Spatial analysis of architectural drawings.

front view and true 3-D reconstruction, so that we finally get a complete 3-D model, in which it will be possible to navigate and/or to perform various simulations.

7. CHALLENGES

In this section, we list a number of open problems and challenges [27, 28]. These stem from our own experience, but we hope that they will also be addressed by other groups.

Complete annotation analysis: As well for 3-D reconstruction as for indexing documentation databases, we have to recognize all the annotations on a drawing: dimensioning, form feature annotations, tolerances, references to the nomenclature, etc. A number of teams have already worked on dimension analysis, but we need to integrate this into a larger system for analyzing all the textual information of the drawing, and the graphical parts it refers to.

Functional and 3-D CAD conversion: Good work has been done on the recognition of basic graphical objects. But we have to put this into the perspective of CAD, and to recognize form features, which are useful both for CAD conversion and for indexing. Results such as those of our Celesstin system are still much too limited and on a too small scale. We also have to continue work on complete, geometric *and* functional reconstruction of a 3-D model from several views. An efficient way to represent and integrate higher-level knowledge must probably be found for that purpose.

But one may wonder whether there is still a market for this. The big companies which needed conversion from paper to CAD have already performed it manually or semi-automatically! However, a large number of CAD files are stored in rather low-level formats, typically vectorial representations. Here, the specific problems of document image analysis are not necessarily present anymore, but the geometric reasoning and recognition processes necessary to convert these data to higher-level CAD representations are very similar.

Analysis of architectural drawings: We clearly believe that there are interesting problems and potential apother groups ...

Modelling of urban environments: Map analysis is an important problem, with lots of applications. A typical problem is to update a map or a GIS through the matching of existing cartographic information, aerial or satellite views, and elevation data. Although some work has already been done in this area, this largely remains a "hot" topic with lots of interesting research subjects: update the map information when new roads or intersections have been built, check the available elevation data through comparison between elevation maps and measurements, build a model of an urban area through merging map information with images taken from the air, etc. Many utilities companies also want to convert to GIS. In each case, there is an interesting, large-scale problem related to graphics recognition: extract from a large set of maps the relevant utilities information.

Performance characterization and evaluation: Many methods are still developed and tested on a limited number of drawings. The validation scope should be significantly extended, so that we can be sure that the methods we design are robust enough. The problem of precision is also crucial, especially in vectorization algorithms.

Another aspect of robustness is that, as in many other image analysis applications, we often end up with lots of *ad hoc* thresholds: What's a thick line and what's a thin line? What's the largest size for a connected component to be a character? What's the angular tolerance for two segments to be aligned? And so on ... In most cases, these thresholds are fixed in a very empirical way, and it would be interesting to have methods for characterizing the behavior of the algorithms, to determine as automatically as possible most thresholds, and to analyze the influence one threshold has on the others.

We also have to continue ongoing work on the characterization and evaluation of the performances, for the methods we design [38], and on the design of robust, generic software environments.

Acknowledgements

I am thankful to all the colleagues, PhD students and other students, who have contributed to our group's results throughout these years. Special thanks to my former PhD students Dominique Antoine, Suzanne Collin, Anja Habacha-Hamada, Pascal Vaxivière, and to my present PhD students Christian Ah-Soon and Philippe Dosch.

References

[1] C. Ah-Soon and Ph. Dosch. Analyse de plans architecturaux. In *Actes du 11ème congrès de reconnaissance des formes et intelligence artificielle, Clermont-Ferrand*, volume 3, pages 113–122, January 1998.

ing Drawings. In *Proceedings of Third International Conference on Document Analysis and Recognition, Montréal (Canada)*, pages 331–334, August 1995.

[3] C. Ah-Soon and K. Tombre. Variations on the Analysis of Architectural Drawings. In *Proceedings of Fourth International Conference on Document Analysis and Recognition, Ulm (Germany)*, pages 347–351, August 1997.

[4] C. Ah-Soon and K. Tombre. Network-Based Recognition of Architectural Symbols. In *Submitted to 7th International Workshop on Structural and Syntactic Pattern Recognition, Sydney, Australia*, August 1998.

[5] D. Antoine, S. Collin, and K. Tombre. Analysis of Technical Documents: The REDRAW System. In H. S. Baird, H. Bunke, and K. Yamamoto, editors, *Structured Document Image Analysis*, pages 385–402. Springer-Verlag, Berlin/Heidelberg, 1992.

[6] C. Arcelli and G. Sanniti di Baja. A Width-Independent Fast Thinning Algorithm. *IEEE Transactions on PAMI*, 7(4):463–474, 1985.

[7] H. S. Baird, H. Bunke, and K. Yamamoto, editors. *Structured Document Image Analysis*. Springer-Verlag, 1992.

[8] H. Bunke and P. S. P. Wang, editors. *Handbook of Character Recognition and Document Image Analysis*. World Scientific, 1997.

[9] A. K. Chhabra. Graphic Symbol Recognition: An Overview. In *Proceedings of Second International Workshop on Graphics Recognition, Nancy (France)*, pages 244–252, August 1997.

[10] S. Collin and D. Colnet. Syntactic Analysis of Technical Drawing Dimensions. *International Journal of Pattern Recognition and Artificial Intelligence*, 8(5):1131–1148, 1994.

[11] G. Sanniti di Baja. Well-Shaped, Stable, and Reversible Skeletons from the (3,4)-Distance Transform. *Journal of Visual Communication and Image Representation*, 5(1):107–115, March 1994.

[12] D. Dori and A. Pnueli. The Grammar of Dimensions in Machine Drawings. *Computer Vision, Graphics and Image Processing*, 42:1–18, 1988.

[13] D. Dori and K. Tombre. From Engineering Drawings to 3-D CAD Models: Are We Ready Now? *Computer-Aided Design*, 29(4):243–254, April 1995.

[14] L. A. Fletcher and R. Kasturi. A Robust Algorithm for Text String Separation from Mixed Text/Graphics Images. *IEEE Transactions on PAMI*, 10(6):910–918, 1988.

[15] 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.

[16] A. H. Habacha and K. Tombre. Structural Symbol

volume 1, pages 486–493, August 1991.

[17] R. D. T. Janssen and A. M. Vossepoel. Adaptive Vectorization of Line Drawing Images. *Computer Vision and Image Understanding*, 65(1):38–56, January 1997.

[18] R. Kasturi and K. Tombre, editors. *Graphics Recognition — Methods and Applications*, volume 1072 of *Lecture Notes in Computer Science*. Springer-Verlag, May 1996.

[19] J. Lladós, G. Sánchez, and E. Martí. A String Based Method to Recognize Symbols and Structural Textures in Architectural Plans. In *Proceedings of Second International Workshop on Graphics Recognition, Nancy (France)*, pages 287–294, August 1997.

[20] 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)*, pages 231–240. World Scientific, 1995.

[21] B. T. Messmer and H. Bunke. Automatic Learning and Recognition of Graphical Symbols in Engineering Drawings. In Kasturi and Tombre [18], pages 123–134.

[22] W. Niblack. *An Introduction to Digital Image Processing*, pages 115–116. Prentice Hall, 1986.

[23] L. O'Gorman and R. Kasturi. *Document Image Analysis*. IEEE Computer Society Press, 1995.

[24] 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.

[25] M. Röösli and G. Monagan. Adding Geometric Constraints to the Vectorization of Line Drawings. In Kasturi and Tombre [18], pages 49–56.

[26] P. L. Rosin and G. A. West. Segmentation of Edges into Lines and Arcs. *Image and Vision Computing*, 7(2):109–114, May 1989.

[27] K. Tombre. Graphics Recognition — General Context and Challenges. *Pattern Recognition Letters*, 16:883–891, September 1995. IAPR pages.

[28] K. Tombre. Analysis of Engineering Drawings: State of the Art and Challenges. In *Proceedings of Second International Workshop on Graphics Recognition, Nancy (France)*, pages 54–61, August 1997.

[29] K. Tombre, C. Ah-Soon, Ph. Dosch, A. Habed, and G. Masini. Stable, Robust and Off-the-Shelf Methods for Graphics Recognition. In *Submitted to 14th International Conference on Pattern Recognition, Brisbane, Australia*, August 1998.

[30] K. Tombre and A. K. Chhabra, editors. *Graphics Recognition—Algorithms and Systems*, volume 1389 of *Lecture Notes in Computer Science*. Springer Verlag, April 1998. To appear.

pages 457–484.

[32] 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.

[33] Ø. Due Trier and A. K. Jain. Goal-Directed Evaluation of Binarization Methods. *IEEE Transactions on PAMI*, 17(12):1191–1201, December 1995.

[34] Ø. Due Trier and T. Taxt. Improvement of "Integrated Function Algorithm" for Binarization of Document Images. *Pattern Recognition Letters*, 16:277–283, March 1995.

[35] P. Vaxivière and K. Tombre. Celesstin: CAD Conversion of Mechanical Drawings. *IEEE COMPUTER Magazine*, 25(7):46–54, July 1992.

[36] 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.

[37] P. Vaxivière and K. Tombre. Subsampling: A Structural Approach to Technical Document Vectorization. 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)*, pages 323–332. World Scientific, 1995.

[38] L. Wenyin and D. Dori. A protocol for performance evaluation of line detection algorithms. *Machine Vision and Applications*, 9(5/6):240–250, 1997.

[39] M. A. Wesley and G. Markowsky. Fleshing Out Projections. *IBM Journal of Research and Development*, 25(6):934–954, November 1981.

[40] S. D. Yanowitz and A. M. Bruckstein. A New Method for Image Segmentation. *Computer Vision, Graphics and Image Processing*, 46(1):82–95, April 1989.