

Analysis of Engineering Drawings: State of the Art and Challenges

Karl Tombre

▶ **To cite this version:**

Karl Tombre. Analysis of Engineering Drawings: State of the Art and Challenges. Tombre, Karl and Chhabra, Atul K. Graphics Recognition – Algorithms and Systems, Springer Verlag, pp.257-264, 1998, Lecture Notes in Computer Science. inria-00107516

HAL Id: inria-00107516
https://inria.hal.science/inria-00107516

Submitted on 19 Oct 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Analysis of Engineering Drawings: State of the Art and Challenges

Karl Tombre

Loria–Inria Lorraine, 615 rue du jardin botanique, B.P. 101, 54602 Villers-lès-Nancy CEDEX, France

Abstract. In this paper, we analyze the state of the art in interpretation of engineering drawings, both from a methodological point of view and from the perspective of the applications. We try to emphasize where techniques are mature, where they need further maturing, and where we still have open challenges. Special attention is given to the progress in the last two years, i.e. to the evolution in the field since GREC'95.

1 Introduction

The problem of converting engineering drawings to CAD representations is not new. A lot of research and development effort has already been devoted to it, for many years. At a graphics recognition workshop, questions therefore quite naturally arise: What has been achieved? Has there been some true progress, or are we more or less reinventing the wheel? Are we coming closer to real-life applications?

Among the conclusions of the GREC'95, the following items, relative to engineering drawings, were mentioned [12]:

- As large quantities of scanned images of documents become readily available, many hybrid raster/vector interactive document data management systems are also become available commercially or are in development. In such systems, semiautomatic conversion is generally accepted as a compromise with simple, routine tasks done by machine. Such systems provide a means for integrating graphics recognition algorithms as they become mature while putting digital images to immediate use...
- Migration from electronic files of 2-D CAD to 3-D CAD systems has created new opportunities for extending the usefulness of Graphics Recognition algorithms beyond paper drawing conversion...
- Graphics recognition is also useful as a human-computer interface in situations where a human would draw sketches and the computer would interpret them. This method is particularly useful for inputting complex parts into CAD systems...
- Search and retrieval systems based on graphical content will experience strong demand in digital library projects and image database systems. Graphical search and indexing capabilities would be crucial for the success of such systems...
- Methods for representation of knowledge and tools for flexible representation and use of such knowledge are still ad-hoc. A formal engineering approach is needed.

In this paper, we propose to examine how things have evolved in the last years to come closer to working, large-scale solutions. Are we still hindered by the same problems? Do we have a set of methods which are mature or approaching maturity? Have we made significant progress on the most challenging problems?

There are two ways to make a state of the art in such a domain. We can either list the different *methods* and *algorithms*, i.e. the "building stones" with which we design a document analysis system, and try to establish the degree of maturity of each of them. We can also be more application-driven, and try to answer the following question: What kind of real-world problems are we able to solve?

We will try to briefly present this state of the art following both of these criteria. Then, we propose a list of interesting challenges, for further research in the area.

2 State of the art: the methods

Although everybody agrees that a good analysis system cannot be *linear* but has to provide some *backtracking* capabilities, most systems remain largely linear! However, in this article, we are also bound by the linearity of paper and time, so we will list the methods in the "usual" order. We are aware that at least the lower-level methods are not specific to engineering drawings analysis, so we may repeat (with our own colors) things already mentioned in other chapters of this book, for different application domains.

2.1 Segmentation

Various segmentation steps can be performed on an engineering drawing. We discuss here the recognition of physical layers; logical layers can only be recognized by higher-level analysis methods [22].

The first segmentation which comes to mind is *text/graphics separation*. When text is not touching the graphics, simple methods, based on the analysis of connected components, usually work well. First, small components are considered as being characters. Various recognition steps gather small components which actually belong to a dotted or dashed line, for instance, and find the groupings of characters into words, in different directions. One of the reference algorithms is that of Fletcher and Kasturi [9]. This can be said to be a mature method, with the limitation that it does not handle well text touching graphics.

There is also often the need to separate thin lines from thick lines. This can be efficiently performed using simple morphological operations, such as an erosion followed by partial geodesic reconstruction. Directional mathematical morphology also offers promising tools for finer-grained segmentation, such as the separation of text strings touching graphics [1, 27].

2.2 Vectorization

Vectorization, i.e. raster-to-graphics conversion, has been given a lot of attention, and many algorithms have been proposed. There are also a number of commercial packages

which perform some kind of vectorization. The state of the art on that topic is quite paradoxical. We claim that:

1. none of these methods works,
2. vectorization is nevertheless a mature field.

Actually, the methods do work, 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 [24], run-based algorithms [29] and approaches directly working on the image [19] or on the distance transform [4]. 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 to use 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 [3, 10].

2.3 Recognition of basic graphical features

Algorithms exist for recognizing various graphical features which are often used in engineering drawings. Some of them are quite mature. For instance, there are good and validated ideas on the way to recognize dashed lines [1, 7], arrowheads [18], circles and arcs [5, 21]. Others are still more experimental, such as the recognition of hatching, and must be considered as maturing techniques.

However, some problems remain more or less open. For instance, it is difficult to design an algorithm which recognizes simultaneously arcs and segments and connects them in a seamless, correct and unambiguous way.

Moreover, in new emerging applications such as graphical indexing for information retrieval from large document databases (cf. § 4), the usual graphical features are not necessarily the best. Here, new challenges appear, as we must find the most useful information for indexing purposes.

2.4 Symbol recognition

Although engineering drawings do not contain as many symbols as charts and diagrams, we still have to recognize a number of them. Symbol recognition in general is a vast domain, and we refer the reader to other articles in this book. One of the main problems with symbol recognition methods is that of their *adaptability* to new symbols or to variants of existing symbols. Recent results by Messmer and Bunke [16] show one possibility for including automatic learning in the process, and for taking noise into account when performing structural matching. But there is still room for lots of improvements, both from a theoretical point of view and with respect to the complexity of the real-world drawings which the methods can handle.

It may also be interesting to continue work on the *genericity* of the recognition processes and on a better framework for representing (possibly as a structural hierarchy) the symbols to be recognized [17].

2.5 3-D reconstruction

Algorithms for geometric reconstruction of 3-D objects from 2-D views have been available for a long time [6, 26, 28]. But they have several limitations, and hence they have not yet been widely used for the practical purpose of converting 2-D drawings to 3-D CAD models, in real-life applications:

- The complexity of the general algorithms is high, so we need good heuristics or more efficient algorithms for large and complex drawings.
- The algorithms are purely geometric, whereas actual drawings also contain symbolic information.
- The fact that we work with scanned documents implies that we have *errors* at every stage; these errors must be taken into account and/or corrected in the reconstruction process.

Some partial solutions are given to these problems in other chapters of this book. However, it is clear that although existing algorithms do exist, true 3-D CAD reconstruction must still be considered as a challenge (cf. § 4).

2.6 Representation of engineering knowledge

In order to achieve higher-level analysis, such as the conversion to *feature-based* CAD, we need efficient and robust ways to represent the knowledge describing the drafting and engineering rules followed by the analyzed drawings. Admittedly, this is a hard problem, especially if we want not only to show the feasibility on a very limited scope of drawings, as we did in the CELESSTIN system [23], but also to design a working system for real-life applications. Despite some scientific successes, such as the TECNOS system [2], which uses conditional attributed rewriting systems to recognize a set of assembly and manufacturing form features, or the recent results published by Langrana *et al.* [13], this must still be considered as a real challenge (cf. § 4). We do not have the impression that there has been any major breakthrough in that area during the last years.

3 State of the art: the applications

If we now take the point of view of the applications for which it may be interesting to perform analysis of engineering drawings, let us first clearly state one point: In our opinion, mere vectorization of scanned engineering drawings is not especially interesting. A 2-D vector has no specific semantics *per se* in CAD, so if the only purpose is to archive documents, it may be sufficient to store the compressed images, or maybe to segment the document image to retrieve the text strings for indexing, leaving the graphical part in image format.

Several applications nevertheless include more or less advanced analysis methods. Here, we very naturally find again the topics mentioned in the GREC'95 conclusions:

Semi-automated conversion: In many practical applications, it makes sense to "put man in the loop", i.e. to build a system where an operator converts the drawing to a CAD representation, with on-line "helper" applications for all the tedious and repetitive tasks. In such a context, the man–machine interface must be carefully designed, with an easy access to the document analysis methods, gathered in some kind of integrated toolbox.

Various software packages already exist for that purpose, but it makes sense to integrate additional methods, whenever they become mature.

On-line sketching: An interesting application is that of the online sketching software, which enables an engineer to interactively make a first sketch, which can then be transferred to a CAD system for further refinements [8]. There certainly is a potential niche for such systems. However, they must still be considered as very experimental.

Technical documentation: An emerging application, with many practical uses, is that of the large technical documentation database, constructed—at least partially—from scanned engineering drawings and other documents. The purpose is not so much to convert to CAD, but rather to be able to browse such a database in a convenient way, and to retrieve the right information through textual *and* graphical indexing. In this context, the emphasis is not really on complete 3-D functional CAD reconstruction, but rather on the recognition of the right graphical features which prove to be useful and efficient for retrieval purposes.

This application domain is growing very rapidly. There are huge needs, and even limited capabilities—in terms of the level of complexity of recognized parts—can lead to useful indexing and browsing tools.

"Dumb" CAD to functional CAD: A large number of CAD files are stored in rather low-level formats, typically vectorial representations. There is a lot of interest in converting these data into the representations based on form features, used by modern CAD systems [11, 14, 15, 20]. Here, the specific problems of document image analysis are not necessarily present anymore, but the geometric reasoning and recognition processes are very similar.

In our opinion, our community is probably not enough aware of this application field. We ought to have closer contacts with the research community in CAD, which has research work in that area.

What about paper-to-CAD? It may come as a surprise that we do not explicitely mention the well-known problem of converting a scanned engineering drawing into a high-level CAD model. We clearly believe that, although there are lots of interesting problems to solve, this does not correspond to a market need anymore. The big companies which needed this kind of conversion have already performed it manually or semi-automatically!

4 Challenges

So, what are the challenges we propose for the coming years? If we cross the methodological aspects with the applications we have presented, we come up with the following propositions:

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.

Recognition of form features: 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.

3-D reconstruction: We have to continue the 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. It must be noted that 3-D reconstruction is a much more general problem (think for instance of medical imaging), so there may be interesting methods to find in neighboring areas.

Thresholds, thresholds, thresholds: 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, and so on, *ad nauseum*. In most cases, these thresholds are fixed in a very empirical way. We need methods to characterize the behavior of our algorithms, to determine as automatically as possible most thresholds, and to analyze the influence one threshold has on the others.

Precision, robustness: 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.

Performance characterization and evaluation: We have to continue ongoing work on the characterization and evaluation of the performances, for the methods we design [25]. We also need to progress towards true systems engineering considerations, to be able to evaluate the performance of a complete system, for a given application.

5 Conclusion

In our opinion, a researcher working on something as concrete as the analysis of engineering drawings should not be satisfied if only his/her prototype yields interesting results on two or three test images, or if other scientists find the results "beautiful". We should ask ourselves "where the market is", without becoming completely market-driven, on the other hand!

The applicative needs are not necessarily where we tend to think they are. For instance, we may have stored away interesting pattern recognition algorithms, which are not sufficient to convert a drawing to a full CAD model, but which still can yield useful indexes for documentation browsing.

We may also have stored away algorithms and software modules which deserve to be better characterized and evaluated. It may seem more intellectually satisfying to

design a “brand new method” for performing vectorization, segmentation or whatever graphics recognition method you happen to think about. But much too often, these “brand new methods” end up being only minor variations on old melodies, and they are rarely breakthroughs when it comes to solving real-life problems.

Let us rather stress the complete maturing of existing methods, so that we can build up a toolbox of robust algorithms and software modules, even if they are not perfect—what is perfection, anyway? From such a basis, we may then be able to reach out to more complex problems, step by step.

References

1. G. Agam, H. Luo, and I. Dinstein. Morphological Approach for Dashed Lines Detection. In Kasturi and Tombre [12], pages 92–105.
2. 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.
3. Y. Chen, N. A. Langrana, and A. K. Das. Perfecting Vectorized Mechanical Drawings. *Computer Vision and Image Understanding*, 63(2):273–286, March 1996.
4. J. E. den Hartog. *A Framework for Knowledge-based Map Interpretation*. PhD thesis, Technische Universiteit Delft, September 1995.
5. D. Dori. Vector-based Arc Segmentation in the Machine Drawing Understanding System Environment. *IEEE Transactions on PAMI*, 17(11):1057–1068, November 1995.
6. 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.
7. D. Dori, L. Wenyin, and M. Peleg. How to Win a Dashed Line Detection Contest. In Kasturi and Tombre [12], pages 286–300.
8. L. Eggli, C.-Y. Hsu, B. D. Brüderlin, and G. Elber. Inferring 3D models from freehand sketches and constraints. *Computer-Aided Design*, 29(2):101–112, February 1997.
9. 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.
10. 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.
11. Q. Ji and M. M. Marefat. Bayesian approach for extracting and identifying features. *Computer-Aided Design*, 27(6):435–454, June 1995.
12. R. Kasturi and K. Tombre, editors. *Graphics Recognition—Methods and Applications*, volume 1072 of *Lecture Notes in Computer Science*. Springer-Verlag, May 1996.
13. N. A. Langrana, Y. Chen, and A. K. Das. Feature Identification from Vectorized Mechanical Drawings. *Computer Vision and Image Understanding*, 68(2):127–145, November 1997.
14. T. De Martino and F. Giannini. The Role of Feature Recognition in Future CAD Systems. In *Proceedings of IFIP international conference on Feature modeling and recognition in advanced CAD/CAM systems, Valenciennes (France)*, volume 1, pages 343–356, May 1994.
15. S. Meeran and M. J. Pratt. Automated feature recognition from 2D drawings. *Computer-Aided Design*, 25(1):7–17, January 1993.
16. B. T. Messmer and H. Bunke. Automatic Learning and Recognition of Graphical Symbols in Engineering Drawings. In Kasturi and Tombre [12], pages 123–134.
17. 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.
18. G. Priestnall, R. E. Marston, and D. G. Elliman. Arrowhead recognition during automated data capture. *Pattern Recognition Letters*, 17(3):277–286, March 1996.

19. J.-Y. Ramel. *Interprétation automatique de dessins — méthodes d'analyse et de reconnaissance — application aux plans cinématiques*. Thèse de doctorat, Institut National des Sciences Appliquées, Lyon, November 1996.
20. A. A. G. Requicha. Geometric Reasoning for Intelligent Manufacturing. *Communications of the ACM*, 39(2):71–76, February 1996.
21. P. L. Rosin and G. A. West. Segmentation of Edges into Lines and Arcs. *Image and Vision Computing*, 7(2):109–114, May 1989.
22. K. Tombre. Graphics Recognition —General Context and Challenges. *Pattern Recognition Letters*, 16:883–891, September 1995. IAPR pages.
23. 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.
24. L. Wenyin and D. Dori. Sparse Pixel Tracking: A Fast Vectorization Algorithm Applied to Engineering Drawings. In *Proceedings of the 13th International Conference on Pattern Recognition, Vienna (Austria)*, volume 3, pages 808–812, August 1996.
25. L. Wenyin and D. Dori. A protocol for performance evaluation of line detection algorithms. *Machine Vision and Applications*, 9(5/6):240–250, 1997.
26. M. A. Wesley and G. Markowsky. Fleshing Out Projections. *IBM Journal of Research and Development*, 25(6):934–954, November 1981.
27. H. Yamada, K. Yamamoto, and K. Hosokawa. Directional Mathematical Morphology and Reformalized Hough Transformation for the Analysis of Topographic Maps. *IEEE Transactions on PAMI*, 15(4):380–387, April 1993.
28. H. Yoshiura, K. Fujimura, and T. L. Kunii. Top-Down Construction of 3-D Mechanical Object Shapes from Engineering Drawings. *IEEE COMPUTER Magazine*, 17(12):32–40, December 1984.
29. S. Di Zenzo, L. Cinque, and S. Levialdi. Run-Based Algorithms for Binary Image Analysis and Processing. *IEEE Transactions on PAMI*, 18(1):83–89, January 1996.