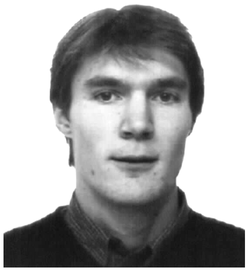
Converting CAD Drawings to Product Models

Robert NOACK
B.Sc. Civil Engineering
Ph.D. Student
Royal Inst. of Technology,
Construction Management
and Economics
Stockholm, Sweden
noack@recm.kth.se



Robert started at KTH in 1996 working with software prototypes in product model related research projects such as Condor and Concur. He started with the licentiate project in 1998 and is now working part time as an IT-consultant at Eurostep.

Summary

Methods to automatically recognize building elements in CAD drawings for use in product model or object based AEC systems are investigated. The need for this is shown by the fact that none of the existing commercial model based systems are able to read drawings produced by earlier vector based systems. Although the scope of this study is limited to use information stored in layer separated vector databases, much like the main bulk of drawings produced in the 20th century, it is believed that the methods can be applied to scanned drawings too. The methods presented are described and followed by a discussion of its application to different types of geometric representations. Prototype implementations have shown that it is possible to acquire good results of recognition on a specific subset of symbols, predetermined by the domain of interest.

Keywords: Shape recognition, shape interpretation, product models

1. Introduction

Design information has been communicated in drawings since the renaissance. The drawing language is now an accepted standard representation for building design, which follows the logic of architecture [1]. However, this language was designed for humans to interpret, or more precisely: it requires a trained expert to fully understand every detail of the drawing [2]. This works because of the tremendous speed a human can process an image on the retina in the eye, and the inductive and associative ability of learning and understanding symbolic language. This is beyond the capacity of any existing computer of today [3]; they require the information to be expressed in a semantically rich format.

It was during the 1990’s that architects started to use CAD almost exclusively in Sweden, which has produced a large set of vector drawings that are to be used today primarily in facility management. The problem is that modern CAD systems are not able to read and understand paper drawings or even drawings in vector format produced by earlier CAD tools [4]. They have been forced to reproduce them manually in a new system, a very costly and time consuming process.

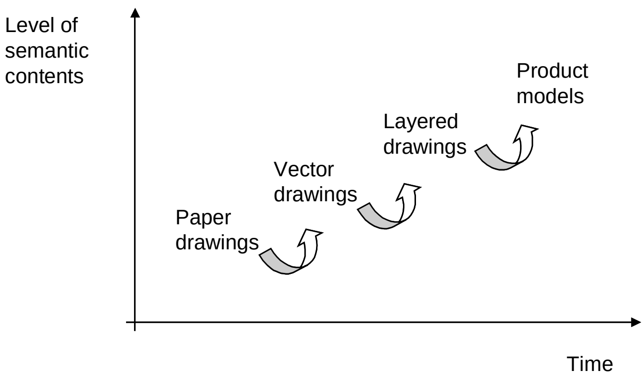


Fig. 1 Major changes of drawing formats

The evolution of CAD depicted in Fig. 1 shows the major steps of technology changes. Each step significantly increases the level of semantic that a drawing can contain, although the main difference between a paper and vector based drawing is the medium. With semantic content is meant the information structures and usability, it is not the form of the shapes that has changed.

Construction Informatics Digital Library http://itc.scix.net/
paper ecce-2001-30.content

1.1 Vector drawings

A vector drawing consists of shapes constructed by a few geometrical primitives such as curves and line segments. There are commercial tools for vectorization (recognizing lines and other graphics) of raster images, which use edge-detection etc to trace the primitives in the image data. The use of vector drawings within facility management is studied in [5]. They argue that vectorization without object recognition results in unstructured vector data with many interpretation errors, and also conclude that such recognizer would be desirable for a selected set of object types.

Layering is a method to classify the information content of a vector drawing, such as walls or annotations, and is heavily used by today's CAD systems. It can be used to filter information that is for the moment irrelevant. The naming conventions used typically derive from classification systems, but not until recently have there been any standard codes for layers. The ISO 13567 [6] provides a framework for structuring of layer names, but has only just been implemented in CAD systems.

1.2 Product Models

Product models, or object models, carries the highest level of semantic. This type of model can contain information not only related to geometry, but also regarding material, time schedules etc. A product model is a logical representation of a building, as opposed to geometrical or shape driven representation. This is illustrated in Fig. 2, where a door is described with its attributes and relations to other objects, compared with the explicit geometrical representation of a CAD drawing.

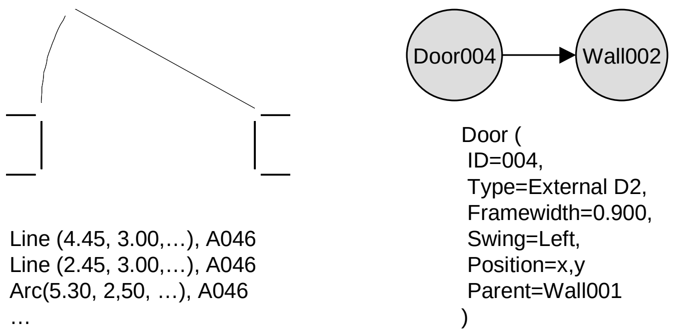


Fig. 2 The representation of a door is illustrated to the left as in a layered CAD drawing and to the right as a product model.

Comprehensive reviews of the available product models for the construction industry is given in [7] and [8]. Common to all of the approaches is the aim for developing a semantically rich logical model to be able to communicate with well-defined and structured data.

It has taken a lot of time to produce standard model, which is one of the reasons that the leading CAD vendors formed the International Alliance for Interoperability to produce the Industrial Foundation Classes (IFC) [9], which was used in this project.

1.3 Architectural drawing interpretation

Although there is much research done on the fundamental technologies for shape recognition, there are not many studies of their application to the interpretation of drawings in the construction domain. However, a library of geometric manipulation functions for querying a vector database is developed at Georgia Tech [10]. The approach resembles much of a relational database language; by making queries on the data set and using filters for sorting out for instance parallel line segments. Subsequent queries can then be made on the result of the initial query. Some initial results show that it is possible to extract information from scanned and vectorized drawings using their library.

Another very extensive research on the subject can be found in [1]. They have built a system that is based on syntactic pattern recognition consisting of four parts: semantics, syntax, geometry and context. They have defined a very thorough system that seems to work at least on their test drawing, although they make a comment on the robustness if it is used with commercial drawings. The system does not expect layered vector drawings, so the question is how well it performs on drawings with much more geometry and noise.

1.4 Scope and objectivities

The overall aim for this study is to determine if it is possible to convert CAD drawings to product models by recognizing building elements. The approach is to test recognition methods in prototype implementations applied to architectural floor plans. The criteria for a successful method is a combination of factors, such as ease of implementation, stability and sensitivity for noise and how specialized the algorithm has to be to recognize a certain type of symbol. The recognition process use these four methods:

Shape Identification. The aim is to find the geometric primitives that compose the shape. This can be difficult since shapes may be intersecting or totally inside of another shape. The approach in paper is to use layers to make the search space smaller.

Shape Classification is done to determine the type of object the shape represents. The problem here is that the shape can be rotated and scaled. It is not always certain of kind of object it represents even if the shape is marked with a layer code. In this paper an artificial neural net classifier are applied to classify symbols in drawings.

Shape Interpretation. Some aspects of an object can be found based on knowledge of how it is represented. For example, if a shape is classified as a window, we might be looking for its length and thickness or the number of glass panes. A wall recognition algorithm is presented here.

Create Relationships. The final step is to examine the relationships between the objects, such as enclosing or connecting objects. One of the key features of an object model is to be able to analyze the structure that the objects are organized in. A method for searching for rooms enclosed by building elements is tested in this paper.

2. Recognition Methods

2.1 Shape Identification

In order to recognize an object in a drawing its shape representation first has to be found. This can be compared with recognizing text, where the characters have to be grouped together based on their position and distance to other characters to form a meaningful word.

Fig. 3 illustrates the shapes that were used in this study, a number of commonly used symbols, together with their class code. Symbols are usually drawn with a tool that inserts a copy from a template library, but it is very common that they are manually edited afterwards, thus they become ungrouped. This means that the algorithm must search for geometry that is connected. To find intersecting or touching lines and curves the only way is to test every geometric primitive against each other.

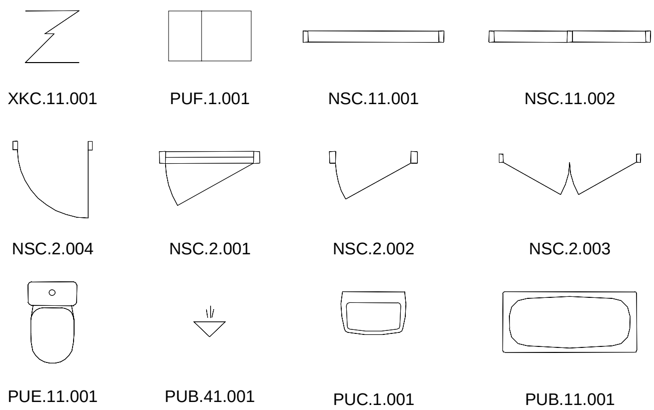


Fig. 3 Shapes part of the symbol library

There are also cases where some details are inside the shape without connections, for example the flush handle of a toilet. A simple algorithm, called parity testing, for determining if a point is inside a closed polygon can be helpful for this purpose.

2.2 Shape Classification

Classification of a shape as belonging to some category of building element is done by looking at some specific features in the geometry. A feature is an attribute of the shape that characterizes it, for example, door symbols usually have rectangular frames on the sides. A problem is to define a set of features that any shape can have, and it can be difficult to obtain these from the shape since it can be rotated and scaled in the drawing. This means that a feature cannot be high level concept, such as has ‘an arc above the base line’ any meaning if the shape happens to be upside-down.

In this study an artificial neural network which given a number of input signals, corresponding to the features of a shape, can calculate the probability for the class that the shape [11]. Furthermore, the neural net can be used with imperfect data and still be able to calculate the likelihood with another shape. This made possible by generalizing of the classes when the net is trained with sample data [3].

Table 1 Example of the feature vectors

Class	F1	F2	F3	F4	F5	F6	F7	F8
NSC.2	4	2	1	0	3	0	11	0
NSC.2	1	2	1	0	0	0	3	0
PUB.11	6	0	10	0	6	0	12	0
PUE.11	5	0	11	1	2	0	3	0
XKC.11	5	0	0	0	3	0	4	2

The feature vectors shown in Table 1 describe of some of the symbol shapes in Fig. 3. The number of features must be equal for all shapes, but the value can be zero if the shape does not have that feature. The proposed method utilizes a vector with 8 features: the number of the geometric primitive types and their relationships. These are all quite easy to determine, they are low level concept that can be applied to all shapes and are not dependent on the transformation of the shape.

2.3 Shape Interpretation

The interpretation of a shape can be considered as the process that lifts the semantic content of the drawing to a higher level. The objects are conceptually defined in a product model, IFC in this case, which means that these definitions will steer the design of the interpretation algorithms. This can only be done by understanding what the concept of an attribute of an object is, how it is represented in the shape and finding a way to calculate it.

Walls and other free form shapes, such as pipes and ducts, can be very difficult to recognize. This is mainly due to they are represented as parallel contour lines with many interruptions. The definition of a wall segment in this study is the shortest centerline between two connection points. The wall recognition algorithm takes lines organized into parallel subsets as input, and produces interconnected wall segments, as can be seen in Fig. 4. It can be started either from a corner of the building or from the centerline of a door or window. It will then recursively search for other walls that are connected to that segment.

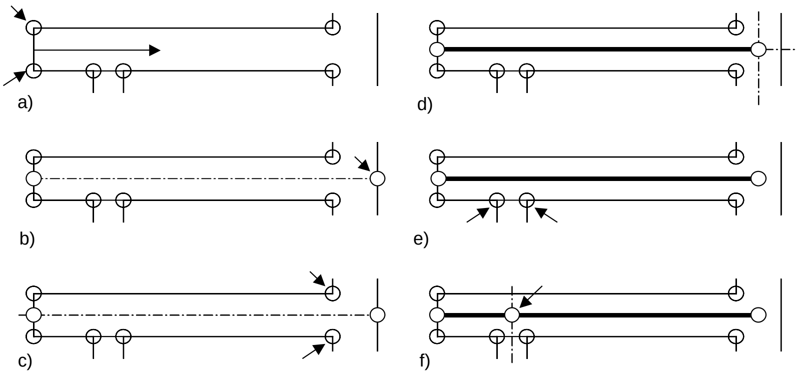


Fig. 4 An example of creating wall segments

This process must run in parallel with the one of linking meeting wall segments together to avoid creating duplicate connection points. This makes it necessary to keep track of what contour lines and intersections are already used and what connections are available.

2.4 Create relationships

Logical relationships or implicit objects are created when linking the structure together. Rooms and spaces are not usually indicated other than with a room label, but can be derived from the surrounding walls. The method used in this study for recognizing rooms is based on the idea that every room in the building has at least one door. The algorithm starts searching for doors from the outside of the building, goes through every door it finds and traces the wall centerlines clockwise, creating a closed polygon. For every door on these walls, the algorithm is repeated recursively.

However, some rooms may have several doors leading to it. All doors are therefore marked as closed from the beginning and they are opened before the algorithm goes through it. In addition, if a wall terminates in a dead end the algorithm must trace back one or more steps until it finds a connecting wall that leads away from the end. This requires a well-designed program since there is a risk that it goes into an infinitive loop, or traces back all the way to the start point.

3. Experimental results

3.1 Test of shape identification

To be able to find out what building elements are represented on what layers a matching table had to be made between the layer codes and the building elements found in IFC. Some, but not many, building elements and layer codes could not be matched.

The shape identification algorithm is very time consuming since it checks every geometric entity for a certain spatial relationship to any other entity in the current layer. Simple symbols could be identified if the geometry was connected end to end. Touching relationships was found the most useful, since it includes the end-to-end relationship and any entities along the geometry. Intersections in general resulted in too many entities that actually belonged to different shapes.

3.2 Test of shape classification

The shape classification method was found very usable. The drawings used for testing the method contained a lot symbols but only a few different types. This fact was used for speeding up the process by reducing the number of queries to the neural network. A simple hash table indexed the different types uniquely, and each of these was then classified.

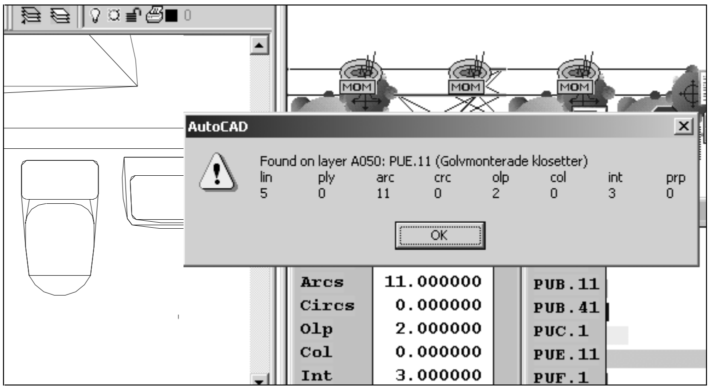


Fig. 5 Result of classification of a toilet symbol

As can be seen in Fig. 5 the prototype has successfully classified a toilet symbol by communicating with the network in NeuroSolutions [12]. The data is sent through the network and result in a vector containing the probability for what type of building element the symbol represents. The most plausible class of these is then displayed in a message box.

The question of what probability is required to safely say what class a shape belongs to depend on the number and similarity of shapes the network can differentiate. A reasonably high probability, 0.7, could be used in the prototype since there where quite few shapes.

3.3 Test of shape interpretation

The wall recognition algorithm was found to work only in almost perfect conditions. There were many situations however where it failed to trace correctly, which had consequence on the following walls. They became corrupt since the algorithm could not calculate the start points or search direction, or completely or unrecognized. This happened for example when a door was connected not to the end of a wall, but to the side, then the wall was traced perpendicular to the correct direction, or when columns were integrated with walls.

3.4 Test of creating relationships

Logical relationships were created after each step of the shape interpretation algorithms. The objectified intersection relationship was found difficult to use if more than two lines meet at a point. This situation will produce three relationships at that point, which confused the algorithms.

The room recognition algorithm is heavily depending on the correctness of the relationships and that no building elements are missing. There were also a few exceptions where the algorithm would not work, for example where the only access to a room is through a vertical opening.

4. Conclusions

The tests of the methods on a number of have shown results of varying degree of success. They fail mainly due to unexpected variations of the shape representations or due to errors in the drawings.

There is a balance between grouping to many primitives and to few when identifying the elements of a shape. If two separate shapes are positioned adjacently there is always a risk that they will be identified together as one shape. In addition, if the geometric entities of a shape are not intersecting at all, as the shower symbol used here, there is no way to know that they belong together.

The most important finding in this study is perhaps the idea of using a shape classifier for symbols. Depending on the usage of the resulting object model, an appropriate recognizer can then interpret certain objects of interest, or it could even be enough to have the knowledge of what kind of building element is located in a certain room without having to identify its properties.

The wall recognition algorithm had several limitations. This affected in turn the room recognition algorithm, which was based on perfect interpretation of the walls. What would make these kinds of recognizers still useful is if they were able to ask the user if it becomes insecure. Unfortunately, the only way to recognize something is to distinguish it from everything else, which means that a recognizer can never tell if it found the right shape or one similar that it does not know about. This leads to the conclusion that the more shapes the system can differentiate between the better, and that fewer errors will occur if the shapes are described in greater detail.

The final conclusion is that the key to design a working CAD-drawing converter for use in practice is robustness, i.e. a high level of error-tolerance. It should be able to detect an error and either ask the user for guidance or select another method for getting around that obstacle. Cross-validation of the results should also be done by alternative algorithms to ensure a result that can be used for further processing.

4.1 Further research

For further research is the obvious need the industry has for recognizing scanned paper drawings. Although much poorer result can be expected from such system, it would be interesting to see what a symbol recognizer could do so that free form recognizers etc can work more undisturbed.

If the product model should be stored with 3D geometry the altitudes and heights must be specified some way. One method is by combining the floor plan drawings with elevation or façade drawings, which requires that the same object can be identified from two perspectives. This technique is developed for engineering drawings and could be used for this purpose.

Acknowledgements

This research was funded by the Swedish research-program “IT in Construction and Facility Management 2002” and was supervised by both professor Bo-Christer Björk and later professor Örjan Wikforss. The author also whish to thank professor Chuck Eastman and Olubi Babalola for the hospitality at Georgia Tech and finally the reference group for their support and advices.

References

[1] Mitchell J. M., *The Logic of Architecture: Design, Computation and Cognition*, MIT Press, Cambridge, ISBN 0-262-13238-9, 1990

[2] Cherneff J., Logcher M., Connor J. and Patrikalakis N., Knowledge Based Interpretation of Architectural Drawings, *Research in Engineering Design*, Springer-Verlag, New York, 1992

[3] Russell J. S. and Norvig P., *Artificial Intelligence - A Modern Approach*, Prentice-Hall International Inc., ISBN 0-13-360124-2, 1995

[4] Dori D. and Tombre K., From engineering drawings to 3D CAD models: are we ready now?, *Computer-Aided Design*, Elsevier Science Ltd., Vol. 27, No. 4, 1997, pp. 243-254

[5] Svensson K., Hunhammar M. and Zabielski L., Are scanned drawings sufficient for facility management work, *CIB/W78 Workshop on Computer Integrated Construction*, Helsinki, 1994

[6] Björk B-C., Löwnertz K. and Kiviniemi A., ISO DIS 13567 - The proposed international standard for structuring in computer aided building design, http://itcon.org, vol 2, 1997

[7] Eastman M. C., Building Product Models – Computer environment supporting design and construction, CRC Press LLC, ISBN 0-8493-0259-5, 1999

[8] Tarandi V., *Neutral Intelligent CAD Communication*, PhD thesis, Royal Institute of Technology, Stockholm, 1998

[9] IAI, Official website and documentation of IFC 2.0, http://iaiweb.lbl.gov, 2000

[10] Babalola O. and Eastman M. C., Working draft paper and discussions at the Collage of Architecture, Georgia Tech, Atlanta, U.S.A, 2000

[11] Holst A., *The Use of a Bayesian Neural Network Model for Classification Tasks*, PhD thesis, Dept of Numerical Analysis and Computing Science, KTH, Sweden, 1997

[12] NeuroDimension, Inc., *The Neural Network Simulation Environment*, Manual, 2000