CV ONLINE

paper submitted: october 25th, 2000

Josep Lladós, Enric Martí
Computer Vision Center - Computer Science Department
Universitat Autònoma de Barcelona
Catalonia - Spain

Jaime López-Krahe
Département Informatique, GRAII, Lab. ai/mime
Université Paris 8
France

Architectural Floor Plan Analysis

1 Introduction

Graphics recognition is a pattern recognition field that closes the loop between paper and electronic documents. Thus, while all the graphic design systems are able to accurately edit and print out their electronic diagrams, the reverse step, i.e. the automatic conversion from a paper-based document to an electronic CAD-readable format, is still a challenge. Architectural floor plan analysis is one of the application domains of the graphics recognition field.

The interpretation of architectural plans is a recent research activity [2,3,10,12,14,16,22]. It has been focused on two kinds of printed plans and hand drawn sketches. the goal of the first activity is the conversion of printed plans into a CAD-readable format for storage and edit purposes. Efficient vectorization engines are at the heart of this kind of systems. The interpretation of hand drawn sketches represents an alternative CAD-input technique and conveys distortion-tolerant symbol recognition processes.

According to the general framework of any graphics recognition system, an architectural floor plan analysis system is organized in three stages (see Fig. 1). Firstly, the *lexical level* is concerned about the extraction of the graphic primitives that compose the drawing, such as lines, arcs and regions. Usually, the lexical level involves a preprocessing step in which operations as binarization, noise removal, thinning, etc. are also performed. Discrimination of text from graphics is also carried out in this step. The second stage is the *syntactic level* which establishes structural relations between primitives grouping them to provide a symbolic representation of the input document. Finally, the high-level processing is the *semantic level*. The aim of the semantic phase is to understand the document, that is analyzing relationships among symbolic mid-level structures and assembling them into high level entities with a particular meaning depending on the domain of the specific application. In the case of architecture, these entities can be rooms, electrical network elements, furniture, etc.

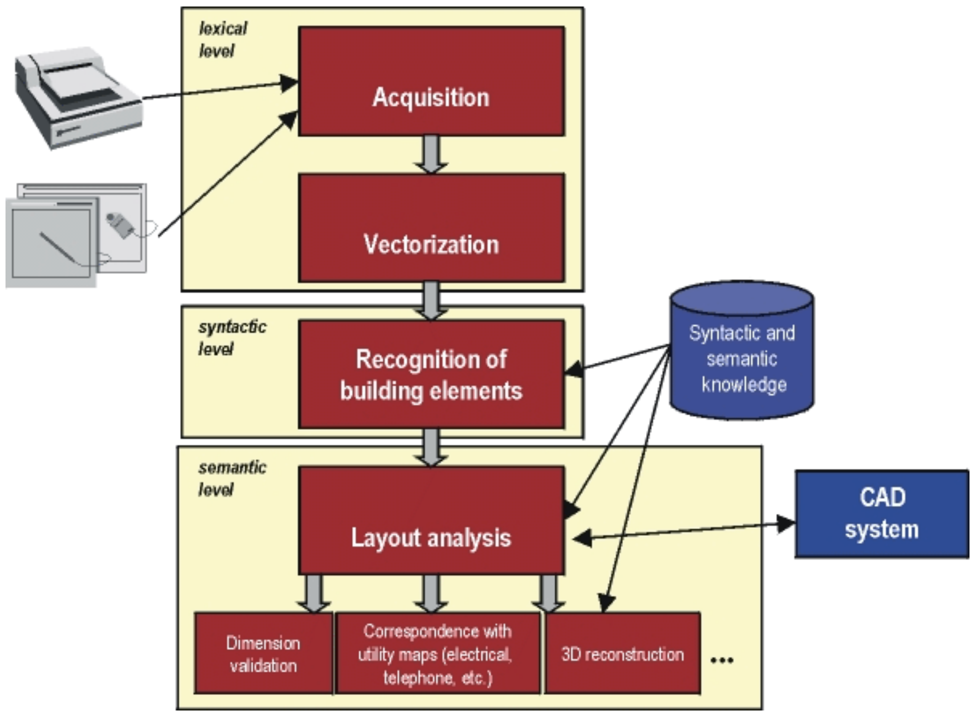

Figure 1: *General scheme for a floor plan analysis system.*

Two activities highlight from the former schema: vectorization and symbol recognition. In the following sections we overview these two steps.

2 Vectorization

Vectorization, or raster-to-vector conversion, consists in converting a scanned image to a vectorial format suitable for further analysis. It is not an specific step for architectural plan analysis, but one of the central interests of graphics recognition community. Two major vectorization approaches can be stated: those that are based on skeleton or medial axis computation, and others that do not thin the image but match opposite contours of lines. Tombre and Tabonne in [21] rencently compared these two families of vectorization methods.

Skeleton based methods consist in computing the medial axis with an iterative thinning or distance transform algorithm. Afterwards, the skeleton is approximated by straight lines, arcs or higher order primitives such as splines. Many skeleton-based vectorization methods have been proposed throughout the years (e.g. [1,5,18,20]), they key issues are the algorithm used to compute the skeleton and the polygonal approximation. The main drawback of these methods is that they tend to introduce distortions at line end points and junctions. However, their robustness when dealing with complex diagrams make them to be the most used method in vectorization packages.

Concerning to contour-matching approaches [11,13,19], they are based in finding the line contours of the plan, polygonally approximate them, match the corresponding vectors in terms of slope and distance criteria and compute the medial line from the pairs of matched vectors. They are more accurate in finding the junctions than skeleton-based methods. The difficulty of contour-matching methods appears in complex diagrams in which one-to-many and many-to-one matching hypotheses appear, for example in case of arcs approximated by polygons, and some heuristics have to guide the correspondence process. Although there are still several improvements to make, the field is mature enough to ask for a robust and universal vectorization method able to work on different domains and with the minimum set of parameters [20]. Probably, one of the improvements in the vectorization process depends on an efficient combination of both families of methods.

Vectorization is not enough to fully understand a plan and convert it to a CAD readable format for further edition. Vectors are only a set of primitives with geometrical and topological attributes. Subsequent intelligent processes are necessary to detect and classify symbolic structures that, according to diagrammatic notations given by the architects, represent doors, windows, walls, etc. In the next section we overview the key issues involved in the recognition of building elements.

3 Recognition of building elements

In architectural drawings, the recognition of higher level entities such as walls, doors, windows, furniture, stairs, etc. allows the interpretation of the document and, hence, its conversion to a CAD environment in which perform features as design modifications, validations, 3D visualization and virtual navigation inside the building. Building element recognition is a pattern recognition activity and, particularly, a symbol recognition problem. Symbol recognition is currently an intensive research activity. It is at the heart of most graphics recognition systems that have been developed. The reader is referred to recent surveys as [4,6,8] to further study on general methodologies for symbol recognition. Although many symbol recognition methods are available, it is difficult to find a general approach that covers different domains. Authors tend to develop techniques restricted to particular domains and costly to be reused in other domains. In the field of architectural plan interpretation, the most outstanding contributions use structural pattern recognition approaches.

Two major symbol structures can be categorized in architectural drawings: prototype-based symbols and texture-based symbols (see Fig. 2). Examples of symbols characterized by a prototype are doors and windows. The recognition is performed by a pattern matching procedure. On the other hand, architectural drawings often contain symbols represented by an structured texture as hatching or tiling. Such structures represent walls, floors or stairs.

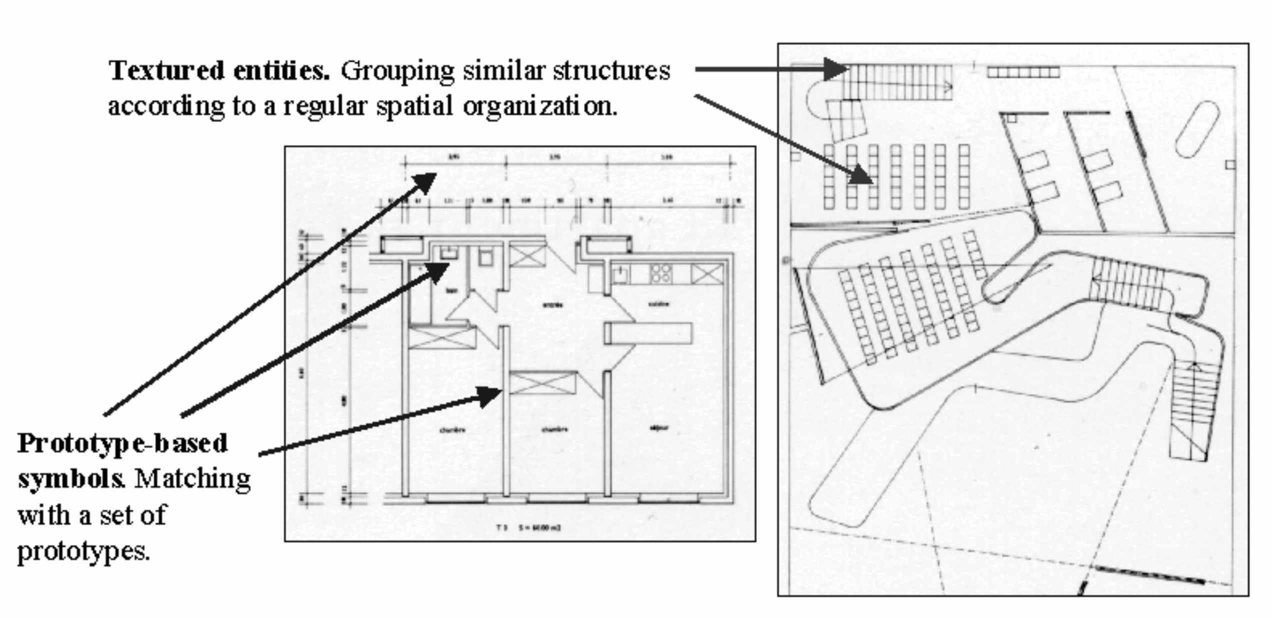
Textured entities. Grouping similar structures according to a regular spatial organization.
Prototype-based symbols. Matching with a set of prototypes.

Figure 2: *Types of structures appearing in an architectural floor plan.*

The more widespread structural method for symbol recognition in architectural plan analysis is graph matching [2,16]. Both, vectorized plans and model symbols are represented by attributed graph structures. Since noisy input documents, and errors in preprocessing and vectorization steps introduce distortions in structures, an error tolerant graph matching procedure is required. Graph-edit procedures and rule-based languages are the most common techniques to solve the problem. Thus, an inexact subgraph isomorphism algorithm finds the minimum cost transformation, in terms of elemental graph edit operations (node and edge insertion, deletion and substitution), from a graph representing the symbol prototype to a graph representing the input document. This approach, in addition to distortion tolerance, does not require a presegmentation step, i.e. symbol recognition is performed directly on the input document and hence, involves symbol segmentation. This is specially useful in architectural plan analysis because symbols are not easily segmentable. However, the computation of the graph edit distance is computationally expensive, specially with complex symbols.

Other approaches work on presegmented symbols. Syntactic and deformable template matching are well known recognition methods when symbols are able to be segmented. Syntactic methods represent symbols by a grammar, usually a graph or a web grammar, and perform the recognition by a parsing procedure. Such syntactic approaches are often used to recognize the dimensioning symbols [7,9,17]. Deformable template matching was used in [22] to recognize presegmented hand drawn architectural symbols. To modelize the distortion involved in hand drawn sketches, the recognition is formulated as an energy minimization problem in a bayessian framework. The energy is defined through the combination of internal energy, which measures fidelity to ideal shape, and external energy which measures the degree of fitness between the deformed symbol and the image.

Concerning to entities characterized by an structured texture, they appear in architectural drawings representing walls, stairs, tiles, roofs, etc. The most usual patterns are hatching and tiling. Hatched pattern detection is based on clustering parallel lines having the same slope angle and sorting them along a direction normal to their slope angle. Besides hatched patterns, textures based on regular repetitions of polygonal shapes are eventually used. To modelize them, syntactic approaches such as array or web grammars are suitable tools. The presence of this type of textures can be found in architectural drawings to represent a tiled floor. In our architectural plan interpretation system [15], we have developed a Hough-based method to detect hatched patterns representing walls, and a graph clustering approach based on a polygon similarity criterion, for tiled patterns.

4 An example of architectural floor plan interpretation

In Fig. 3 we show an example of the interpretation of a hand drawn architectural floor plan. Figure 3(a) shows the scanned document, A skeleton-based algorithm gives the vectors of Fig. 3(b). Afterwards, the building elements are recognized in two steps, namely, structures characterized by a hatched pattern (Fig. 3(c)) and prototype-based symbols (Fig. 3(d)). Finally, the reconstructed plan is shown in Fig. 3(e).

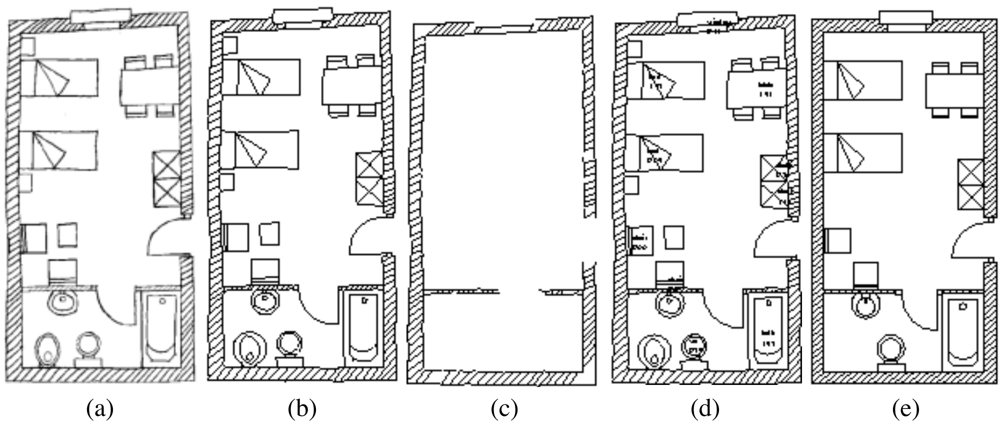

Figure 3: Example: (a) Input image, (b) Vectorization, (c) Hatched pattern detection (walls), (d) Prototype-based symbol detection, (e) Plan reconstruction.

The most recent version of this article is here.

References

[1]
S. Ablameiko, V. Bereishik, O. Frantskevich, M. Khomenko, E. Melnik, N. Paramonova, O. Patsko, and O. Okun. A system for automatic vectorization and interpretation of graphic images. *Pattern Recognition and Image Analysis*, 3(1):39-52, 1992.

[2]
C. Ah-Soon. A constraint network for symbol detection in architectural drawings. In K. Tombre and A. Chhabra, editors, *Graphics Recognition: Algorithms and Systems, Selected Papers from Second International Workshop on Graphics Recognition, GREC'97*, pages 81-90. Springer, Berlin, 1998. Volume 1389 of Lecture Notes in Computer Science.

[3]
Y. Aoki, A. Shio, H. Arai, and K. Odaka. A prototype system for interpreting hand-sketched floor plans. In *Proceedings of 13th. International Conference on Pattern Recognition*, pages 747-751, Aug 1996. Vienna, Austria.

[4]
D. Blonstein. General diagram-recognition methodologies. In R. Kasturi and K. Tombre, editors, *Graphics Recognition: Methods and Applications, Selected Papers from First International Workshop on Graphics Recognition, 1995*, pages 106-122. Springer, Berlin, 1996. Volume 1072 of Lecture Notes in Computer Science.

[5]
Y. Chen, N. Langrana, and A. Das. Perfecting vectorized mechanical drawings. *Computer Vision and Image Understanding*, 63(2):273-286, March 1996.

[6]
A. Chhabra. Graphic symbol recognition: An overview. In *Proceedings of Second IAPR Workshop on Graphics Recognition*, pages 244-252, Aug 1997. Nancy, France.

[7]
S. Collin and D. Colnet. Syntactic analysis of technical drawing dimensions. *International Journal of Pattern Recognition and Artificial Intelligence*, 8(5):1131-1148, 1994.

[8]

L. Cordella and M. Vento. Symbol and shape recognition. In *Proceedings of Third IAPR Workshop on Graphics Recognition*, pages 179-186, Sept 1999. Jaipur, India.

[9]
D. Dori. A syntactic/geometric approach to recognition of dimensions in engineering machine drawings. *Computer Vision, Graphics and Image Processing*, 47(3):271-291, 1989.

[10]
P. Dosch, C. Ah-Soon, G. Masini, G. Sánchez, and K. Tombre. Design of an integrated environment for the automated analysis of architectural drawings. In S.-W. Lee and Y. Nakano, editors, *Document Analysis Systems: Theory and Practice. Selected papers from DAS'98*, pages 295-309. Springer Verlag, 1999. Volume 1655 of Lecture Notes in Computer Science.

[11]
K. Fan, D. Chen, and M. Wen. Skeletonization of binary images with nonuniform width via block decomposition and contour vector matching. *Pattern Recognition*, 31(7):823-838, July 1998.

[12]
C. Faure and K. Chabani. Assistance interactive pour l'aménagement de pièces d'habitation à partir de dessins à main levée. In *CNED*, July 1996. Nantes.

[13]
C. Han and K. Fan. Skeleton generation of engineering drawings via contour matching. *Pattern Recognition*, 27(2):261-275, 1994.

[14]
A. Koutamanis and V. Mitossi. Automated recognition of architectural drawings. In *Proceedings of 11th. International Conference on Pattern Recognition*, pages 660-663, Sep 1992. The Hage, The Netherlands.

[15]
J. Lladós, J. López-Krahe, G. Sánchez, and E. Martí. Graph-based solutions for recognition of maps and plans. In *Proceedings of Rec. de Formes et Intell. Artificielle (RFIA'2000)*, pages 225-234, February 2000. Paris, France.

[16]
J. Lladós and E. Martí. A graph-edit algorithm for hand-drawn graphical document recognition and their automatic introduction into CAD systems. *Machine Graphics & Vision*, 8(2):195-211, 1999.

[17]
W. Min, Z. Tang, and L. Tang. Using web grammar to recognize dimensions in engineering drawings. *Pattern Recognition*, 26(9):1407-1916, 1993.

[18]
V. Nagasamy and N. Langrana. Engineering drawing processing and vectorisation system. *Computer Vision, Graphics and Image Processing*, 49:379-397, 1990.

[19]
J. Ramel, N. Vincent, and H. Emptoz. A coarse vectorisation as an initial representation for the understanding of line drawing images. In K. Tombre and A. Chhabra, editors, *Graphics Recognition: Algorithms and Systems, Selected Papers from Second International Workshop on Graphics Recognition, GREC'97*, pages 48-57. Springer, Berlin, 1998. Volume 1389 of Lecture Notes in Computer Science.

[20]
K. Tombre, C. Ah-Soon, P. Dosch, G. Masini, and S. Tabonne. Stable and robust vectorization: How to make the right choices. In *Proceedings of Third IAPR Workshop on Graphics Recognition*, pages 3-16, Sept 1999. Jaipur, India.

[21]
K. Tombre and S. Tabonne. Vectorization in graphics recognition: To thin or not to thin. In *Proceedings of 15th. International Conference on Pattern Recognition*, volume 2, pages 91-96, September 2000. Barcelona, Spain.

[22]
E. Valveny and E. Martí. Application of deformable template matching to symbol recognition in hand-written architectural drawings. In *Proceedings of 5th. International Conference on Document Analysis and Recognition*, pages 483-486, Sep 1999. Bangalore, India.

File translated from TEX by TTH, version 2.78.
On 25 Oct 2000, 18:17.