Reconstruction of the 3D Structure of a Building from the 2D Drawings of its Floors

Philippe Dosch, Gérald Masini

HAL Id: inria-00098758

https://hal.inria.fr/inria-00098758

Submitted on 26 Sep 2006

Reconstruction of the 3D Structure of a Building
from the 2D Drawings of its Floors*

Philippe Dosch, Gérald Masini
LORIA–CNRS–INRIA–UHP, B.P. 239
54506 Vandœuvre-lès-Nancy Cedex, France
Philippe.Dosch@loria.fr

Abstract

This article presents our first results in reconstructing the 3D structure of a building using primitives delivered by the analysis of the 2D architectural drawings of its floors. A set of features is extracted from the set of architectural symbols available for each floor. A graph of matching hypothesis between the most pertinent features of two consecutive floors is then constructed. A maximal clique detection algorithm supplies the best set of matches. The geometrical transformation allowing each floor to be aligned with respect to its lower floor is computed from the pairs of matching features. The final 3D model of the building is obtained by heaping the 3D models of the consecutive floors.

1. Introduction

We are currently working on a project partially funded by CNET, the French National Research Center for Telecommunication. The purpose is the automatic generation of a 3D geometric model of a given building, as complete as possible, from the architectural drawings of its different floors. The way the Hertzian waves propagate through the building can then be simulated with respect to the materials used for walls, covers, doors, etc.

A 3D reconstruction typically requires five steps. After binarization, the image of each floor map is separated into its textual part and its graphical part. The latter is vectorized, *i.e.* segmented into a set of attributed primitives, like straight line segments and arcs of circle. These primitives are grouped into graphical symbols representing doors, windows, staircases, and so on. Finally, pertinent symbols extracted from two consecutive floors are used as features to match the floors and find the location of the upper floor over the lower floor.

This article focuses on the last step. The organization of the software system (§ 2) and the processings involved in the first four steps are very briefly described. The matching is then explained in details (§ 3): The categories of features used for the matching, the selection of pertinent features for a particular matching, the construction of a compatibility graph and the extraction of the largest maximal clique. The first experimental results (§ 3.6) are presented before concluding (§ 4).

2. An overview of the system

Our system globally includes three layers. The first one is a set of image and graphics basic tools, designed as C++ classes and grouped into a library named ISADORA [6]. The different tools originate from the work of investigation that our research group has been leading about graphics recognition techniques during the last ten years [11]. The second layer includes useful so-called (graphics recognition) *applications*, combining low-level processings from the previous layer into higher-level ones, for example vectorization or symbol extraction. The third layer is a sophisticated user interface, named MICA [6], encapsulating the two other layers. It allows a human assistance to be applied at each step of analysis and it gives a visual feedback on the global working of the system.

Each sheet of a set of drawings describing the different levels of a same building has first to be transformed into its geometrical 2D structure, expressed in terms of "symbols": Dividing and bearing walls, doors, windows, staircases, etc. As lots of methods to extract and combine the information we need for that are now available in the literature, we did not try and imagine quite new techniques for each (sub)problem we had to deal with. On the contrary, whenever it was possible, we rather adapted notorious and efficient methods appearing to be best suited to our purpose.

Extracting symbols takes four steps. The image of a drawing is binarized [12] and texts are separated from

graphics [7]. The elements of the graphics part are then classified into thin lines (dividing walls and architectural symbols) and thick lines (bearing walls). The vectorization extracts attributed primitives, *i.e.* straight line segments and arcs of circle [5], from the graphical part, classically using a skeletonization [4] followed by a polygonal approximation [9]. All the primitives provided by the previous steps are grouped into symbols representing building components like doors, windows, pipes [1], or staircases [10]. The final result gives the geometrical 2D structure of the drawing.

3. Matching floors

3.1. The features

The way how two consecutive floors fit is determined by matching features selected in their 2D geometrical description. Various architectural drawings have been first studied in order to select the kind of features to be taken into account. On the one hand, it appears that symbols like staircases and pipes can be directly used as features because they are "transverse" symbols: Their location is generally invariant between two consecutive levels, and, if they are present on a given level, they are present on the next lower or upper one. On the other hand, the matching cannot be exclusively based on such symbols as they are generally not so many or, like pipes, are often missing. In addition, the shape of staircases may change from one level to the other and thus does not provide great accuracy for the matching.

Dividing walls are not interesting features as their configuration is not guaranteed to be stable through the different floors. This is not the case with bearing walls whenever the outer shape of the building is relatively similar for each floor. They are considered as pertinent features, especially at *corners* where two walls meet, as the corresponding intersection points allow a very precise location of the features. When corners and pipes are missing, only bearing walls themselves, *i.e.* simple straight line segments, are available. Then, the matching cannot obviously be precise, since such segments are not provided with characteristic points to be used as "anchors" for the correspondence.

Four categories of features have been finally selected. *Corners* are pertinent features whenever the shape of the floors is quite stable. They are determined using a method described by Jiang and Bunke [8]. *Staircases* infallibly are available on maps of multi-level buildings, although corresponding ones may have quite different shapes. The shape of *pipes* is always invariant but is the most often symmetric (square or circle) and can be consequently matched in several different ways. By definition, the location of *bearing walls* generally is invariant. However, there are not reliable features, in particular because the nature and disposition of

pieces of joinering (doors and windows) may differ from one level to the other.

3.2. The matching method

The choice of the algorithm to match the 2D features depends on several criteria implied by the previous observations. As maps may be drawn using different scales and are scanned from separate sheets, matching features requires the computation of the transformation (a combination of a translation, a rotation and a scaling) that aligns a map with the other. The sets of features of the different floors are generally very dissimilar. Only the so-called transverse features are stable. Their (relative) locations do not change, even when the global shapes of the floors differ from each other. Relative locations are also invariant to translation, rotation and scaling. It thus seems appropriate to represent a floor as a relational model: A feature is characterized by the relative locations of a fixed minimum number $\mathcal{N}_{\min}$ of neighboring stable features.

As there are relatively few features to match and as each feature is associated with a small number of simple attributes, a sophisticated matching algorithm would be useless. In fact, our problem appears to be very close to that presented in [3]. It convinced us of adopting a similar approach, based on a largest maximal clique detection method.

3.3. Feature selection

A reliable match between a pair of given floors, $\mathcal{F}_1$ and $\mathcal{F}_2$, must rely on a minimum number of robust features. A priority order is defined on the categories of features, depending on the invariance of their locations and shapes, as explained in § 3.1: Pipes, staircases, corners and bearing walls, in decreasing order. Location stability prevails shape stability since the former allows more precise matches, and thus a more precise computation of the transformation relating $\mathcal{F}_1$ and $\mathcal{F}_2$.

Let $f_{\min}$ be the minimal number of features required for a reliable match. Its value is experimentally defined. Let $c_{\min}^i$ be the minimum of the numbers of features from category $\mathcal{C}_i$ available in $\mathcal{F}_1$ and in $\mathcal{F}_2$. It corresponds to the maximum number of consistent matches that can be performed for this category. The categories $\mathcal{C}_i$ are successively considered in their priority order, summing the corresponding $c_{\min}^i$. When the total reaches $f_{\min}$, the possibly remaining categories are ignored and the others are selected for the global matching process.

Such principles ensure that noisy or non reliable features are not used whenever the set of pertinent features is large enough. They subsequently do not disturb the matching.

3.4. The compatibility graph

A match hypothesis between a selected feature from $\mathcal{F}_1$ and a selected feature from $\mathcal{F}_2$ constitute a node in a compatibility graph. Each feature is assimilated to a point: The gravity center of the polygon representing the contours of a staircase or a pipe, the intersection of the two segments forming a corner, the middle of the segment representing a bearing wall. The relative orientation of two features is computed as the direction of the segment joining their two representative points.

Two features $f_1 \in \mathcal{F}_1$ and $f_2 \in \mathcal{F}_2$ are associated in a match hypothesis on condition that confidence rate $\Delta(f_1, f_2)$ is lower than a fixed threshold:

$$\Delta(f_1, f_2) = \sum_{i=1}^{i=\mathcal{N}_{\min}} \min_{j=1}^{j=\mathcal{N}_{\min}} \delta(f_{1.i}, f_{2.j})$$

where $f_{p.q}$ represent the q-*th* neighbor of feature f_p. δ expresses the difference between the relative orientations of the two considered neighboring features. If these features do not belong to the same category, δ returns a value which makes any match between the two features impossible.

Let $f_1^a{:}f_2^i$ and $f_1^b{:}f_2^j$ be two match hypothesis. An edge relates the two corresponding nodes in the graph if the pair of nodes represent a consistent hypothesis, that is to say $a \neq b$ and $i \neq j$ (a same feature from $\mathcal{F}_1$ cannot obviously match two different features from $\mathcal{F}_2$, and vice versa) and $\theta(f_1^a) - \theta(f_2^i) \approx \theta(f_1^b) - \theta(f_2^j)$. θ represents the absolute direction (in the coordinate system of the image plane) of a corner or a wall: The direction of the bisectrix of the angle formed by the corner, and the direction of the segment representing the wall, respectively.

3.5. The largest maximal clique

The largest maximal clique, *i.e.* the largest completely connected subgraph, that can be extracted from the compatibility graph represents the best match between the two current floors. As we say in § 3.2, we do not need using any sophisticated method since, in particular, the size of our compatibility graph generally is relatively small. A simple and straightforward algorithm, like the one described in [3] represents a good compromise solution.

The resulting clique is used to compute the transformation[1] that aligns floor $\mathcal{F}_1$ to floor $\mathcal{F}_2$. A match $f_1^a{:}f_2^i$ relates feature f_1^a, whose characteristic point is (x_1^a, y_1^a), to feature f_2^i, whose characteristic point is (x_2^i, y_2^i). The coordinates of the latter point are expressed according to the coordinates of the former point in the following way:

$$x_2^i = S\cos\theta x_1^a - S\sin\theta y_1^a + T_x$$
$$y_2^i = S\sin\theta x_1^a + S\cos\theta y_1^a + T_y$$

[1]We would like to thank Marie-Odile Berger for her help for this point.

where T is the translation, θ the angle of the rotation and S the scaling factor. The coordinates of the n points of floor $\mathcal{F}_2$ are given by the system:

$$\underbrace{\begin{bmatrix} x_2^1 \\ y_2^1 \\ \vdots \\ x_2^n \\ y_2^n \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} x_1^1 & -y_1^1 & 1 & 0 \\ y_1^1 & x_1^1 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^n & -y_1^n & 1 & 0 \\ y_1^n & x_1^n & 0 & 1 \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} kc = S\cos\theta \\ ks = S\sin\theta \\ T_x \\ T_y \end{bmatrix}}_{\phi}$$

This system corresponds to the least square minimization of the error on the transformation. The precision of the solution increases with the number of available matched points. Matrix ϕ can be computed in the following way [2]: $B^t A = (B^t B)\phi \Rightarrow \phi = (B^t B)^{-1} B^t A$.

The translation factors along the X and Y axis are T_x and T_y, respectively. The rotation angle θ is given by $\arctan\frac{ks}{kc}$ and the scaling factor S by $\sqrt{kc^2 + ks^2}$.

3.6. Results

Figure 1 shows the first experimental results obtained with a small building, a private house two stories high. The 3D reconstruction has been generated from the architectural drawings of the first and second floors. The different construction components have been correctly recognized and precisely located (Fig. 1.a and 1.b).

The upper floor has been aligned to the lower floor thanks to the transformation computed after the floors have been matched. The result does not seem very precise: The upper floor is slightly misplaced on the lower floor. The matching itself is not concerned, as junction points delivered by the vectorization process we are using for the moment are distorted. They are wrongly located, and thus the points characterizing the corners are wrongly located too. The alignment cannot be reliably performed.
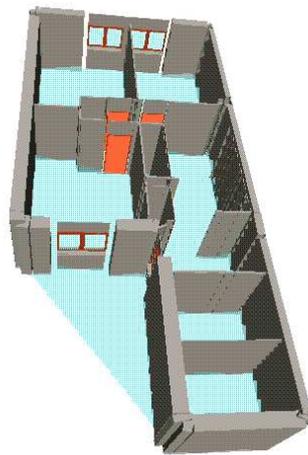
The first floor contains 79 architectural symbols and the second 125, from which have been respectively selected 53 and 51 features. A compatibility graph with 15 nodes and 36 edges has given a largest maximal clique of 5 nodes. The whole process, from the determination of the features to the generation of the full 3D structure, takes less than 0.3 s on a Sun Ultra 1 workstation.
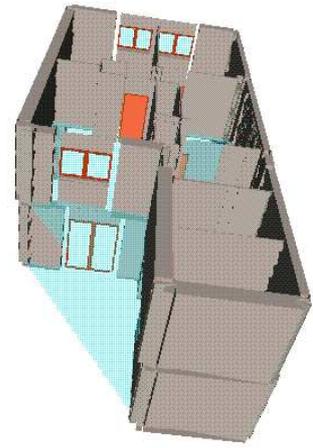
4. Conclusion

This paper has presented a complete interactive system comprising a library of basic image and graphics processings, and a set of higher-level graphics recognition applications, linked to a user interface. The system is currently used to generate a 3D geometrical model of a building from its architectural drawings.
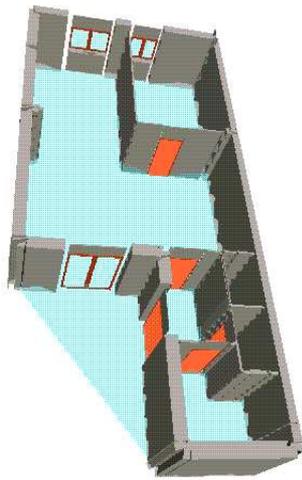
(a) 3D structure of the 1st floor.   (b) 3D structure of the 2nd floor.   (c) 3D structure of the whole building.

Figure 1. 3D reconstruction of a private house from the architectural drawings of its floors.

We have emphasized the criteria that have guided the design of the matching method: Simplicity and efficiency led us to use a maximal clique detection. For the moment, the method has just been experimented with small buildings (essentially private houses with two levels), but the first experimental results are very encouraging and demonstrate the appropriateness of our choices. They also clearly point out the main defects of our system.

The low-level processings, especially vectorization, are concerned: They do not supply information reliable enough to be able to precisely locate construction components and consequently to precisely align floors. We are currently working on the implementation of refined algorithms of skeletonization and polygonal approximation.

We are also integrating new tools into the system, to get more precise information. For example, the recognition of the dimensioning lines will provide the actual dimensions of the construction components and will help to align their match. At last, we are improving the detection of complex symbols, like staircases, in particular by the use of new methods of texture and hatching detection.

References

[1] C. Ah-Soon. Symbol Detection in Architectural Drawings. In *Proceedings of 2nd International Workshop on Graphics Recognition, Nancy (France)*, pages 280–286, Aug. 1997.

[2] D. Ballard and C. Brown. *Computer Vision*. Prentice-Hall, Englewood Cliffs (NJ), USA, 1982.

[3] R. C. Bolles and R. A. Cain. Recognizing and Locating Partially Visible Objects: The Local-Feature-Focus Method. *The International Journal of Robotics Research*, 1(3), 1982.

[4] G. S. 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, 1994.

[5] D. Dori. Vector-based Arc Segmentation in the Machine Drawing Understanding System Environment. In A. L. Spitz and A. Dengel, editors, *Document Analysis Systems*, pages 338–362. World Scientific, 1995.

[6] 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, *Proceedings of Third IAPR International Workshop on Document Analysis Systems (Nagano, Japan)*, pages 366–375, Nov. 1998.

[7] 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.

[8] X. Y. Jiang and H. Bunke. An Optimal Algorithm for Extracting the Regions of a Plane Graph. *Pattern Recognition Letters*, 14:553–558, 1993.

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

[10] G. Sánchez, J. Lladós, and E. Martí. Segmentation and Analysis of Linial Texture in Planes. In *Proceedings of 7th Spanish National Symposium on Pattern Recognition and Image Analysis, Barcelona, Spain*, volume 1, pages 401–406, 1997.

[11] K. Tombre, C. Ah-Soon, P. Dosch, A. Habed, and G. Masini. Stable, Robust and Off-the-Shelf Methods for Graphics Recognition. In *Proceedings of the 14th International Conference on Pattern Recognition, Brisbane (Australia)*, pages 406–408, Aug. 1998.

[12] Ø. D. Trier and T. Taxt. Improvement of 'Integrated Function Algorithm" for Binarization of Document Images. *Pattern Recognition Letters*, 16:277–283, Mar. 1995.