Toward Automatic Floor Plan Interpretation

Stefano Ferilli

Department of Computer Science
University of Bari, Bari, Italy
stefano.ferilli@uniba.it

Abstract Beyond exploitation and management of document collections based on the syntactic level, approaching the semantic level can open new perspectives for the collection users. This is particularly challenging in technical documents, where most relevant information is implicit in the graphic. This paper deals with architectural floorplans, proposing an approach based on formal representation and reasoning for their understanding and interpretation. The results of our study show that it is a viable and promising line of research.

1 Introduction

Understanding the logical and semantic structure of documents is key to carry out many high-level tasks, both on the single documents and on document collections. E.g., indexing the information for efficient retrieval and browsing [17], or answering questions about their content, or even providing interpretations and insights that help their users in spotting relevant content or errors. The availability of effective and efficient approaches to some of these tasks may have a positive impact on organizational and even economic performance of the organizations handling those documents.

Especially challenging is the case of technical documents (such as systems schemas, maps, etc.), where the information needed to support the above tasks is often implicit in the graphics. Nowadays they are often designed using software that produces a vector-graphics representation, in which the visual elements are symbols representing relevant object categories for the domain. While some source formats of the documents (e.g., an annotated vectorial format), as produced by these tools, may explicitly include metadata that report (at least part of) this information, often the documents lack any high-level information. Even worse, in practice, most often only a raster version of the final artifact is available as the starting point. Indeed, the source representation is rasterized for printing or publication into digital media, or, even worse, a scanned image is available. This causes the loss of its structure and makes interpretation more complex, posing an additional burden to recover the high-level information from a plain grid of pixels.

The branch of Artificial Intelligence aimed at dealing with this kind of issues is (Document) Image Analysis. Image analysis and image understanding, in turn, heavily rely on pattern recognition solutions. However, most approaches in this area can only recover the syntactic level of the document's content, or when they aim at semantics they just scratch the surface, by providing interpretations (labels) of single (simple or aggregate) components of the documents. We believe that a full support to the document users should be provided, including a high-level interpretation of the document and its components, that may allow advanced automatic document handling. For this, an approach that reproduces the inferences that an expert would make on the document, is required. This paper proposes a possible research direction toward this goal, based on formal (First-Order Logic, or FOL) representation and reasoning.

As a sample application domain, in this paper we focus on floorplan analysis, as a special case of image analysis and understanding. Architectural floorplans define indoor spaces as scaled bi-dimensional diagrams representing an intuitive and detailed graphical description of an entire building, a floor of a building, or a single room, using structural symbols (rooms, walls, doors, windows, parking doors, and room separations) and non-structural ones (bed, toilet, bath-tub, shower, sink). Among technical documents, floorplans are interesting because they can be handled also by non-technical people. However, not all kinds of inferences can be made by lay people on a plain floorplan image. Some require expert interpretation, and even experts may need some support to carry out their tasks. The goal of floorplan analysis is to extract different structural and semantic aspects of a building by analyzing the 2D image of its floorplan. As the sample task, we propose the interpretation of the kind of building based on its shape and furniture organization. To this aim, identifying the presence of locations, such as a dining room, a bedroom or a classroom, based on the type and organization of the furniture objects on the map, is a preliminary step. For example, the detected rooms can identify an apartment, a school or a clinic.

This paper is organized as follows. After discussing related works in the next section, in Section 3 we report about the image pre-processing steps, and in Section 4 about our object classification approach. Then, Section 5 describes our approach to floorplan interpretation, including the conceptualization we defined and a sample execution. Finally, the last section concludes the paper and outlines future work issues.

2 Related Work

Floorplan processing is an active research area, as witnessed by the development of several floorplan datasets for research purposes in the last decade. The CVC-FP collection [9] comprises 122 scanned floorplan documents of different qualities, resolutions and modeling styles. The dataset proposed in [14] contains 90 floorplans provided by a single architectural firm, and thus they contain only few objects and have low variability in the set of symbols. Since floorplan drawings are usually copyrighted, [5] proposed an approach to generate synthetic floorplans, and developed the SESYD dataset. It contains only 10 floorplans and is not realistic. Another solution is downloading floorplans freely available on the Internet [21, 4].

While recovering the high-level information from a rasterized floorplan image is complex, the task of architectural floorplan analysis is not new. Different objectives were pursued, e.g., generating 3D models [6, 10] or corresponding CAD format [20]. An interesting application area is the retrieval of similar floorplans [18, 3]. [2, 18] deal with room detection. [15] proposes a way to partition open plan regions. [14, 19] aim at detecting rooms and their connectivity topology. [20, 11] propose methods to interpret and understand hand-sketched floorplans. [10] focuses on the detection of walls. [7] presents a complete system for the analysis of architectural diagrams in order to recognize the basic primitives, also involving human feedback in the analysis phase.

Object detection is a crucial, but particularly hard, task in floorplan image processing. Most existing algorithms focus solely on the symbol recognition task [16]. [13] proposed a technique based on graph matching to extract the symbols. [1] proposed a neural network based solution to identify the doors and windows in the floorplan. Deep learning techniques, in particular, are interesting because they allow to localize and recognize symbols in a single step. [12, 21, 4]. While deep learning-based approaches have been successful in recognizing and detecting objects in real scene images, detecting objects in floorplan images is a very different task. Also, they are hardly applicable to small training sets.

After identifying the building elements in the floorplan, semantic analysis aims at interpreting them with respect to their context. However, to the best of our knowledge, existing approaches are not very ambitious, and far from a real high-level understanding and interpretation of the drawings. [16] reports 'Architectural Floorplan Image Understanding' in the title, but mainly tackles the problem of generating 3D model description of a building solely from its 2D floorplan obtained by digitizing a plan hardcopy. Their contribution primarily lies in the 3D model authoring process. They consider only the lines representing the construct, ignoring text and furniture objects. They also require manual intervention to remove some symbols which are not intended to be reconstructed. [3] builds on [2, 14], focusing on 'semantic analysis' to find the function of the detected rooms (e.g., WC, Living room, etc.). However, this is done by reading the text in the boundary of the room and checking it against a dictionary. Rooms with many function labels but no physical partition are then split into several sub-regions based on the detected labels, until all rooms have one label. This kind of splitting is very subjective. Finally, the regions which do not have any room label are merged with a neighboring room which is aligned with it.

Like [7], in our prototype we involve the user in the recognition phase to ensure a better quality of the input to the reasoning step. Our main contribution lies in semantic analysis, where we aim at demonstrating the potential of logic-based representation and inference for high-level interpretation of the floorplans.

3 Overall Pipeline & Pre-processing

The prototype we developed carries out the following pipeline of steps[1]:

Walls identification Walls are identified, and used for a first splitting of the floorplan into rooms.

Floorplan segmentation The image is segmented to obtain furniture objects, and their related bounding boxes.

Symbol recognition The extracted regions are given as input to a classifier for identifying the type of furniture.

Floorplan description Based on the detected rooms and furniture objects, a FOL description of the document is generated.

Reasoning The formal floorplan description is used in Prolog-based inference engine to understand what is represented in the floorplan.

While complex and tailored solutions exist in the literature for the floorplan pre-processing steps, we used standard approaches readily available in a library, so as to quickly obtain a working system. Indeed, the focus of this paper is on proving the usefulness of the formal reasoning apprach, not ensuring top effectiveness or efficiency of the pre-processing steps. Since the effectiveness of the reasoning approach is obviously related to the quality of the pre-processing, we allowed manual correction of the pre-processing results by the user.

The remainder of this section describes the image pre-processing techniques we used, up to the segmentation step.

Image binarization The next steps require a binary copy of the original image (unless it is already binary), that we obtain applying the Otsu's method [94] for automatic image thresholding. It returns a threshold intensity that separates pixels into foreground and background classes, determined by maximizing the variance of each class. Optionally, to highlight the edges in the image, it is possible to perform a logical AND on white pixels, between the original image filtered with Otsu's method, and a negative copy of the original image filtered first, with Sobel's operator for edge detection and then with Otsu's method.

Walls Removal Floorplans are basically images with white background, and walls and furniture objects as foreground. Walls are typically continuous black lines which divide the represented space to describe the planimetry. Before applying object detection, walls must be removed. Using the line removal strategy described above might alter or anyway affect the furniture objects, requiring additional operations to fix them. Since wall removal is not the focus of this paper, here we adopted a quick-and-dirty technique for this step. The walls typically correspond to the largest (in number of pixels) connected components in the

[1] In the following we will refer to raster floorplan images. Processing of vector formats is expected to be easier (because they carry more explicit information), and will be the subject of future work. At the moment, vectorial images would be transformed into raster images before processing.

floorplan image (compared to the furniture objects). In our dataset, we selected floorplans in which walls were represented as thick, solid black lines[2]. This allowed us to adopt our quick-and-dirty approach on our dataset. So, we extract all the components whose size is above the average of the connected components in the floorplan image, and use it as a binary mask having the walls pixels as foreground. Since there can be furniture objects attached to the walls, depicted with thin lines (compared to the thick lines of the walls), we first erode the mask, so that these object lines disappear, and then dilate it twice. Then, the resulting mask is removed from the original image. While not directly used to identify rooms (which happens in the floorplan interpretation phase), walls detection might be usefully during actual room identification by checking that all valid rooms identified must not span across walls.

Image Segmentation Extraction of objects requires a segmentation step. We used a segmentation method based on region growing. Region growing are spatial segmentation methods, in fact they rely mainly on the assumption that the neighboring pixels within one region have similar values. The common procedure is to compare one pixel with its neighbors. If a similarity criterion is satisfied, the pixel can be set to belong to the same cluster as one or more of its neighbors. Compared to other methods, we think it is more suitable for extracting objects in floorplan images, because it is less dependent on the image size and it does not require a segmentation threshold. While our algorithm can work on color images, in the following we describe its version for black&white images, where 'white' is interpreted as the background, and 'black' as the foreground. Its only requirement for a correct segmentation is that the object has a continuous external contour made up of foreground pixels only. It works in 3 phases:

Background Expansion This phase is needed because many objects are solid shapes, but they contain background pixels. The input image is scanned row-wise starting from the top-left pixel, searching for a background pixel. Once such pixel is found, all its 4-connected neighbor pixels are expanded and added to the homogeneous region, if they are background or have not yet been considered for the expansion. This procedure continues until no further expansion is possible.

Foreground Expansion The pixels not belonging to the expanded background create a large heterogeneous region containing all pixels belonging to different objects. The aim of this phase is to correctly split such a region in many homogeneous regions, each corresponding to a different object. All 4-connected neighbor pixels of the first foreground pixel retrieved from the heterogeneous region are expanded, added to the current homogeneous region and removed from the heterogeneous region, always checking that they are foreground pixels or have not yet been expanded. When the homogeneous region under construction cannot be further be expanded, it is removed from the foreground and another pixel is picked from the remaining foreground to detect

[2] In other representations, walls are represented as pairs of parallel thin lines with background pixels in between

another heterogeneous region. The procedure continues until no pixels are left in the heterogeneous region.

Object Extraction Each homogeneous region detected in the previous phase corresponds to an object. This phase identifies the corresponding bounding box, and extracts them into new images, each obtained by copying the content of the corresponding bounding box in the original image. As an additional noise reduction, we discarded all images whose bounding boxes have a size less than 3×3.

4 Furniture Object Classification

We cast the object detection task as a symbol recognition problem. So, each candidate object extracted in the segmentation step is fed to the recognition method, in order to assign it to a known class. For this purpose, we used Artificial Neural Networks as the classification model, as described in the following.

Feature vector extraction Since Artificial Neural Networks work with a numeric representation of the input, the input image must be translated into a numeric *feature vector*. For a hopefully better performance, in this step we use all available information in the original image, and thus we consider its color version.

The extraction of the features takes place in two steps:

1. **resizing**, where the image is normalized to a size of 100×100 pixels (if larger or smaller than that size)
2. **translation towards the center**, where the image is positioned in the center with respect to the smaller size

These steps serve to make the recognition invariant with respect to the size of the symbols. Then, the image is partitioned into 16 regions of size 25×25. Then for each region we perform:

- a horizontal partition: we compute the average of the red (M_r) green (M_g) and blue (M_g) values of the pixels in each row in the region. For each row, the three values are then combined as $M_r \cdot 10^6 + M_g \cdot 10^3 + M_b$.
- a vertical partition: obtained as for the horizontal partition, but considering columns.

So, each region is represented by a vector of 50 elements. The 16 vectors are merged to form the final features vector, consisting of $16 \cdot 50 = 800$ values.

Training Set Examples in the training set are pairs (I, O), where I is the feature vector of the given example, and O is the one-hot vector of the expected output, containing as many elements as the target classes, each associated to a different class, and value 1 in the position corresponding to the expected class of membership and value 0 in all the other positions. Since the target classes are numbered 1 to 200, the size of the expected output vector is 200.

Neural Network architecture We used a Multi-Layer Perceptron architecture, consisting of 3 layers:

input layer (800 neurons), the elements of the feature vector;
hidden layer (500 neurons);
output layer (200 neurons), the one-hot vector of the target classes.

For input neurons the transfer function is an identity function while for hidden and output neurons we used the sigmoid function; therefore, the input and output values of neurons range from 0 to 1. It is then necessary to normalize the features vector in such range, before it is passed in input to the respective neurons of the input layer. The normalization of the features vector is done by dividing each element by $256 \cdot 10^6$.

We trained this neural network for 500 epochs, or until errors falls below a given threshold. The loss function for the backpropagation algorithm takes the following form:

$$E(\omega) = \frac{1}{2} \sum_{X \in T} \sum_{i=1}^{N} (t_X(i) - o_X(i))^2$$

where t_X and o_X are respectively the network output for example X and the expected output for that example. For weights update we used the Resilient Backpropagation algorithm.

Neural networks approximate the input example with the closest known example, even if the input example is very different from each known example. For this reason, the example is actually recognized only if the maximum value in the output vector produced by the network is greater than 0.9. Otherwise, the output class for the example is 0.

The neural network was trained just once on about 10% of the furniture objects extracted from the floorplans in our dataset. This amounts to 47 items, each appearing 4 times, one for each orientation (North, South, East, West), for a total of 188 images. While small, this training set was representative of the objects to be classified, since all of them were drawn from a homogeneous set of floorplans. So, it was acceptable for our preliminary experiment. Applied to the remaining 90% furniture objects, the learned model was able to correctly label 96% thereof. Then, wrong classifications were manually fixed by the user.

We use a database to store information on the classes and instances of symbols to be recognized and their instances. In the interactive use of the system, the floorplan image is displayed with the recognized symbols in a green bounding box, and those not recognized in a red bounding box. The user may label unrecognized objects, or re-label recognized ones to fix recognition errors, assigning them to other classes. Objects (re-)labeled by the user are displayed in a blue bounding box. The new labels are stored in the database, and the user may retrain the classifier on the whole database, in order to improve its future performance. Our system stores all furniture objects, both those recognized and those not recognized. Since neural networks are not learned incrementally, the instances in the database may not be aligned with the examples used to train the current network. The alignment is run offline, by retraining the network from scratch on all the objects in the database.

5 Floorplan Interpretation

In order to enable formal reasoning on the content of a floorplan, it must be formally described according to a conceptualization. We adopt a First-Order Logic formalism, allowing the most flexibility in representing and handling the available knowledge, and for which many reasoning tools are available in the literature and practice.

As said, a floorplan is a drawing to scale, showing a view from above, of the rooms, spaces, and other physical features for one level of a structure. Given a floorplan, we describe both the objects it contains, with their type and attributes, and, most important, several kinds of relationships among them. Based on this representation, many inferences can be made, both structural (typically, by experts) and functional (typically, by the intended users of the construction). In this paper, for demonstration purposes, we focus on the basic goal of understanding the type of building represented in the floorplan, based only on its shape, on the furniture objects identified into it, and organization, and neglecting possible text in the floorplan. Our strategy (and thus the organization of our knowledge base) is based on 3 main high-level concepts:

Physical objects : all the furniture (sofa, bed, table, . . .) or structural objects (doors, windows, lighting point, . . .) that can be represented on a floorplan, and can be useful to understand it.

Location : a space or a room on the floorplan. It is described in terms of the physical objects it contains. For example, a ‘bedroom’ could be defined as including a ‘bed’ and a ‘wardrobe’, or a ‘dining room’ could consist of a ‘table’ and some ‘chairs’.

Plan : the kind of building depicted by the diagram. It is expressed in terms of detected locations. For instance, an ‘apartment’ could be defined as including a ‘bedroom’, a ‘dining room’, a ‘kitchen’ and a ‘bathroom’. In this paper we assume that each diagram describes a single type of plan.

We first distinguish the different locations based on objects’ position, orientation, bounding box and mutual relationships. This information allows to describe many patterns of possible locations (e.g., “if a coffee table is in between a sofa and a TV, a living room is identified”). Then, we use some typical organizations of the detected locations to distinguish the type of building described in a plan (e.g., locations living room, kitchen, bathroom and bedroom can describe an apartment). So, we are interested in representing spatial and compositional relationships between the various elements.

In the following we report the conceptualization we used in our experiments. In addition to satisfactorily supporting the demonstration purposes of this paper, it (with its spatial and composition relationships) is already able to satisfactorily express a wide range of situations. Moreover, it can be easily extended to handle more complex floorplans or objectives, or to improve effectiveness or efficiency.

Predicates The following basic predicates are used to express the information coming from the pre-processing and object recognition steps, concerning position and type of the furniture objects:

object_type(O, T) : furniture object O is of type T (e.g. sofa, armchair, table, ...);
direction(O, D) : object O points to direction D (north, south, east or west);
pos_center(O, X, Y) : the center of the bounding box of furniture object O has coordinates (X, Y);
pos_box(O, X', Y', X'', Y'') : the bounding box of object O is delimited horizontally by coordinates (X', X'') and vertically by coordinates (Y', Y'').
pos_center(O, X, Y) : furniture object O has a rectangular bounding box with center at coordinates (X, Y)
max_distance(X) : X is the maximum distance to consider two objects close to each other.

Since a floorplan may include many physical objects of the same kind, each object is assigned a unique integer identifier. The type, bounding box and orientation of each object are known from the object recognition phase. Some types of objects may have more than one direction, in which case many facts are generated. For example, a 'table' with identifier 1 pointing in the north direction will also point to south; so it will generate facts direction(1,north) and direction(1,south). For identifying locations that are not delimited by walls, a proximity-based approach is used. It is based on a parameter representing the maximum distance (in pixels), such that two furniture objects can be considered sufficiently close so as to belong to the same location. Setting this value is crucial, because too large or too small values may prevent correct recognition of locations. The value may be specified by the user, or computed automatically from statistics about the distances of objects in the current floorplan.

Given a floorplan to interpret, after labeling the objects of interest, its description using the above formalism is created and added to the working memory of the reasoner. It consists of facts describing the floorplan based on the identified objects, generated according to the above conceptualization.

Rules Further predicates express higher-level concepts, directly or indirectly derivable from the basic ones using the rules in the knowledge base. Different groups of rules in the knowledge base are devoted to defining various concepts, and concepts in some groups may concur in defining higher-level concepts in other groups. The following groups of rules are defined in the current prototype (by increasing concept level):

- Rules to understand when two objects are sufficiently close to each other so as to allow supposing that they might belong to the same location.
- Rules to infer the mutual position of the objects from the floorplan's reader perspective.
- Rules to describe the spatial organization of the objects.

- Rules to describe aggregates of objects of the same type, close each other. This is useful to describe certain locations characterized by the presence of object aggregates (e.g., an aggregate of student desks, may identify a classroom).
- Rules, usually used besides aggregate, in order to identify a single object from which the aggregate can be recognized, when no other references are available.
- Rules to describe possible location patterns, considering the spatial relationships among the furniture objects contained in them. For each location one or more patterns have been defined.
- Rules to detect the type of plan, starting from the identified locations (the current prototype knowledge base models 5 floorplan types, with related locations: apartment, classroom, clinic, restaurant and cinema).

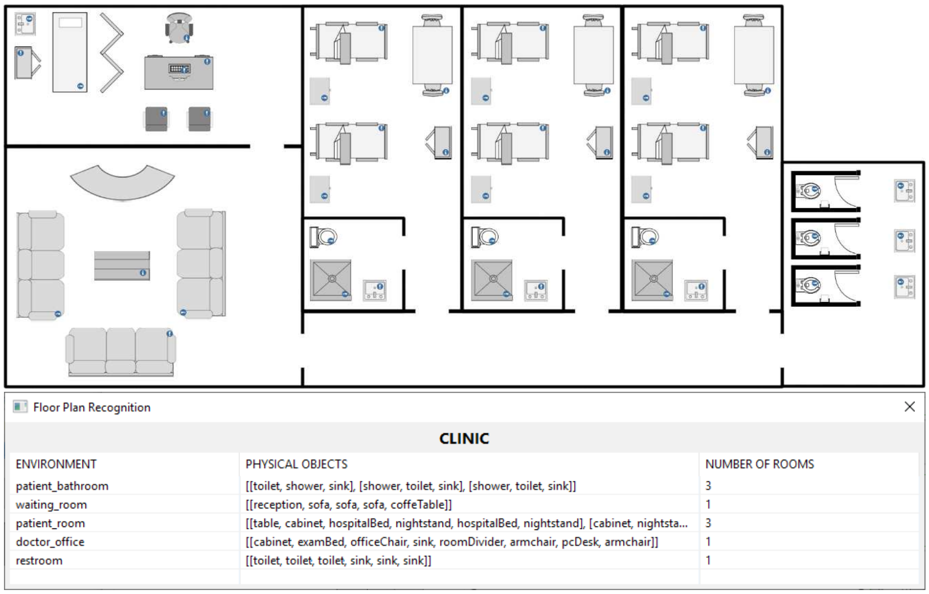
Overall, the knowledge base includes 75 rules. What's really important, these rules, and especially those for location and plan types, which are the most complex and difficult to develop, can be automatically learned using systems such as the one in [8].

The output consists of the type of floorplan detected and information about the location with their respective furniture objects. It is displayed in a table whose label on the top specifies the type of floorplan detected, and whose content is organized in 3 columns, reporting respectively: the locations with associated type, the list of object types in the location and the number of locations of that type. If no known floorplan type is detected, an alert message is displayed. Figure 1 shows a sample floorplan (on the top), and the corresponding interpretation (on the bottom). The system can also explain its classifications, by reporting an explicit account of the reasoning steps used to reach its conclusions.

Since our aim was proving the effectiveness of logic-based interpretation of floorplans, we did not run a thorough quantitative evaluation of our approach. As to the floorplan interpretation step, we ran it on 30 floorplans of various type collected from the Internet (due to property rights they cannot be shared), and the current version of the knowledge base could correctly recognize 26 of them. Using the explanation feature, we could understand the deficiencies of the knowledge base for which it misrecognized the other 4 floorplans (mainly missing required furniture of different furniture layout). This again proves the usefulness of a logic-based approach, and suggests it may provide many additional and more interesting insights.

6 Conclusions and Future Work

Beyond exploitation and management of document collections based on the syntactic level, approaching the semantic level can open new perspectives for the collection users. This is particularly challenging in technical documents, where most relevant information is implicit in the graphic. In this paper, we focused on architectural floorplans. While automated floorplan processing is not new, most works in the literature focused on the extraction of syntatic features only,



CLINIC

ENVIRONMENT	PHYSICAL OBJECTS	NUMBER OF ROOMS
patient_bathroom	[[toilet, shower, sink], [shower, toilet, sink], [shower, toilet, sink]]	3
waiting_room	[[reception, sofa, sofa, sofa, coffeTable]]	1
patient_room	[[table, cabinet, hospitalBed, nightstand, hospitalBed, nightstand], [cabinet, nightsta...	3
doctor_office	[[cabinet, examBed, officeChair, sink, roomDivider, armchair, pcDesk, armchair]]	1
restroom	[[toilet, toilet, toilet, sink, sink, sink]]	1

Fig. 1. Sample floorplan interpretation output

or on limited semantic interpretations. Here we proposed an approach based on formal representation and reasoning for their understanding and interpretation.

A prototype implementation of our approach obtained interesting results, showing how it can provide useful support to both professional and non-technical users. In future work, we plan to expand our knowledge base, so as to provide additional insight in the floorplans, and to support decision making on them. Also, we will investigate the possibility of automatically learning the knowledge base, so as to overcome the well-known ‘knowledge acqiusition bottleneck’.

References

[1] Ah-Soon, C.: A constraint network for symbol detection in architectural drawings. In: Graphics Recognition – Algorithms and Systems, Lecture Notes in Computer Science, vol. 1389, pp. 80–90 (1997)

[2] Ahmed, S., Liwicki, M., Weber, M., Dengel, A.: Improved automatic analysis of architectural floor plans. In: 11th International Conference on Document Analysis and Recognition. pp. 864–869. IEEE (2011)

[3] Ahmed, S., Liwicki, M., Weber, M., Dengel, A.: Automatic room detection and room labeling from architectural floor plans. In: 2012 10th IAPR International Workshop on Document Analysis Systems. pp. 339–343 (2012)

[4] Capobianco, S.: Deep Learning Methods for Document Image Understanding. Ph.D. thesis (2020)

[5] Delalandre, M., Valveny, E., Pridmore, T., Karatzas, D.: Generation of synthetic documents for performance evaluation of symbol recognition & spotting systems.

International Journal on Document Analysis and Recognition (IJDAR) **13**, 187–207 (2010)

[6] Dosch, P., Masini, G.: Reconstruction of the 3d structure of a building from the 2d drawings of its floors. In: International Conference on Document Analysis and Recognition. vol. 5, p. 487 (1999)

[7] Dosch, P., Tombre, K., Ah-Soon, C., Masini, G.: A complete system for analysis of architectural drawings. International Journal on Document Analysis and Recognition **3**, 102–116 (2000)

[8] Esposito, F., Semeraro, G., Fanizzi, N., Ferilli, S.: Multistrategy theory revision: Induction and abduction in inthelex. Machine Learning **38**, 133–156 (2000)

[9] de las Heras, L.P., Terrades, O., Robles, S., Sánchez, G.: Cvc-fp and sgt: a new database for structural floor plan analysis and its groundtruthing tool. International Journal on Document Analysis and Recognition (IJDAR) **18**, 15–30 (2015)

[10] de las Heras, L.P., Mas, J., Sánchez, G., Valveny, E.: Wall patch-based segmentation in architectural floorplans. In: 11th International Conference on Document Analysis and Recognition. pp. 1270–1274 (2011)

[11] J. Llados, J.L.K., Marti, E.: A system to understand hand-drawn floor plans using subgraph isomorphism and hough transform. Machine Vision and Applications **10**, 150–158 (1997)

[12] Liu, C., Wu, J., Kohli, P., Furukawa, Y.: Raster-to-vector: Revisiting floorplan transformation. In: IEEE International Conference on Computer Vision, ICCV 2017. pp. 2214–2222 (2017)

[13] Llados, J., Suchez, G., Mart, E.: A string based method to recognize symbols and structural textures in architectural plans. In: Graphics Recognition – Algorithms and Systems, Lecture Notes in Computer Science, vol. 1389, pp. 91–103 (1997)

[14] Macé, S., Locteau, H., Valveny, E., Tabbone, S.: A system to detect rooms in architectural floor plan images. In: Proceedings of the 9th IAPR International Workshop onDocument Analysis Systems (DAS '10). pp. 167–174. ACM (2010)

[15] Madugalla, A., Marriott, K., , Marinai, S.: Partitioning open plan areas in floor plans. In: 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), volume 1. pp. 47–52. IEEE (2017)

[16] Or, S., Wong, K., Yu, Y., Chang, M.: Highly automatic approach to architectural floorplan image understanding & model generation (2008)

[17] Tombre, K., Lamiroy, B.: Graphics recognition - from re-engineering to retrieval. In: Proceedings of the 7th International Conference on Document Analysis and Recognition - Volume 1, ICDAR '03. p. 148. IEEE Computer Society (2003)

[18] Weber, M., Liwicki, M., Dengel, A.: a.SCAtch – a sketch-based retrieval for architectural floor plans. In: 12th International Conference on Frontiers of Handwriting Recognition. pp. 289–294 (2010)

[19] Wessel, R., Blümel, I., Klein, R.: The room connectivity graph: Shape retrieval in the architectural domain. In: 16th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2008. UNION Agency-Science Press (Feb 2008)

[20] Y. Aoki, A. Shio, H.A., Odaka, K.: A prototype system for interpreting hand-sketched floor plans. In: 13th International Conference on Pattern Recognition (ICPR 1996). vol. 3, pp. 747–751 (Aug 1996)

[21] Ziran, Z., Marinai, S.: Object detection in floor plan images. In: ANNPR, 2018. pp. 383–394. LNCS (2018)