Pattern Recognition Letters 35 (2014) 91–100

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Automatic analysis and sketch-based retrieval of architectural floor plans

Sheraz Ahmed [a,b,*], Markus Weber [a], Marcus Liwicki [a], Christoph Langenhan [c], Andreas Dengel [a,b], Frank Petzold [c]

[a] German Research Center for Artificial Intelligence (DFKI) GmbH, Trippstadter Straße 122, 67663 Kaiserslautern, Germany
[b] Knowledge-Based Systems Group, Department of Computer Science, University of Kaiserslautern, P.O. Box 3049, 67653 Kaiserslautern, Germany
[c] Chair of Architectural Informatics, Faculty of Architecture Technical University of Munich, Arcisstrasse 21, 80333 Munich, Germany

A R T I C L E I N F O

Article history:
Available online 19 April 2013

Keywords:
Floor plan analysis
Structure analysis
Architecture
Wall detection
Symbol spotting
Room detection

A B S T R A C T

The contribution of this article is twofold. First, we propose a sketch-based system, namely the a.SCatch system, for querying a floor plan repository. Second, for generating such a repository, a novel complete system for floor plan analysis is presented. The latter system extracts the semantics from existing floor plans. We introduce novel preprocessing methods, e.g., the differentiation between thick, medium, and thin lines and the removal of components outside the convex hull of the outer walls. Especially, the use of the convex hull increases the performance of the final system. The a.SCatch system enables the user to easily access knowledge from past projects. The user searches for semantically similar floor plans just by drawing parts of the new plan. An algorithm extracts the semantic structure sketched by the architect on DFKI's *Touch* & *Write* table. Finally, the extracted structures are compared using the graph matching, and the most similar one is retrieved. In our experiments for floor plan analysis on a reference data set we compare our approach to other approaches available in the literature. We show that our floor plan analysis system outperforms previous systems. Also the performance of the sketch recognition system is quite high. Overall, the performance of the floor plan analysis, as well as of the retrieval, is already good for the use in practice.

1. Introduction

During design process, architects use existing and already designed buildings as reference. These reference drawings are used to guide solutions for similar architectural situations. Whenever an architect has to solve a new architectural problem, his first task will be to search for similar projects. By studying one or several previous reference projects the architect tries to derive a solution for his current problem. This is a common approach in architecture, knowing about to use reference projects during the design process and the knowledge of reference projects is an essential skill of an architect.

These days electronic search in architecture is realized by searching for textual annotations. However describing architectural work just using textual information is not sufficient, as verbal descriptions are subjective and often imprecise. Thus a pure textual annotation of floor plans is too fuzzy for an efficient retrieval.

* Corresponding author at: German Research Center for Artificial Intelligence (DFKI) GmbH, Trippstadter Straße 122, 67663 Kaiserslautern, Germany.
E-mail address: sheraz.ahmed@dfki.de (S. Ahmed).

Hence, in order to support an architect in his early design phase, there is a need for a search tool which enables him or her to find similar projects. As already stated, a textual search approach is not sufficient and too fuzzy. Furthermore, as we are dealing with visual information it is more intuitive to formalize a query employing a visual query language. The language should be an abstraction of an original floor plan symbolism, in order to have an intuitive way for architects to formalize a search query.

Langenhan proposes a semantic structure to describe the content of a floor plan based on functional and spatial relations of different structural entities. This formalization results in a graph representation which can be used for the retrieval process. The a.SCatch system enables the user to easily access knowledge from past projects. The user searches for semantically similar floor plans just by drawing parts of the new plan.

However, it is a well known problem that it is difficult to generate the semantic database for already existing reference paper floor plans, also known as the bootstrapping problem. Therefore, our second major contribution is an automatic floor plan recognition system which analyzes the floor plans and finally retrieves the corresponding semantic information. Usually, floor plan analysis systems consist of information segmentation, followed by

0167-8655/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2013.04.005

structural analysis and finally, semantic information extraction and alignment. The retrieved structural and semantic information can be saved in a repository for later access during retrieval.

Note that this article is an extended version of the conference contribution by Weber et al. (2010), where the a.SCatch system was initially proposed. While Weber et al. (2010) provided only a short description of the retrieval system, this article presents more details on all involved processes. Furthermore, the floor plan recognition methods introduced by Ahmed et al. (2011a) are elaborated and evaluated in this paper as well. Finally, we propose novel post-processing techniques for the semantic floor plan analysis and report on results of the floor plan recognition as well as sketch recognition and floor plan retrieval.

The novel key techniques of our approach are: (i) the segmentation into thick, *medium*, and thin walls to improve preprocessing techniques; (ii) the extraction of the building boundary removing many false positives during text extraction; (iii) detection of wall edges using convex-concave hypothesis; (iv) use of patch based approach coupled with Speeded Up Robust Features (SURF) for finding doors locations in the given floor plan.

The rest of this paper is organized as follows. First, Section 2 gives an overview of the related work. Subsequently, Section 3 introduces the concepts of the a.SCatch system and discusses the details of the floor plan analysis system. Section 4 shows the performed experiments. Finally, Section 5 concludes the work.

2. Related work

This section provides an overview of related work to the different techniques used in this paper. Related work can be categorized into architectural background, sketch-based interfaces, symbol spotting, complete floor plan analysis systems, and graph matching.

2.1. Architectural background

Since the middle of the 1990s the approach of applying Case-Based Reasoning (CBR) to design and architectural tasks has been known as Case-Based Design (CBD). The case-base contains information on buildings that have already been built or designed, enabling the computer to adapt solutions accordingly, on its own or with help from the architects. Table 1 provides a brief overview of some CBD systems based on two studies published by Heylighen and Neuckermans (2001) and by Richter et al. (2007) regarding the proposed approach. The marked fields show whether the appropriate feature was realized in the concept.

The study by Richter et al. (2007) identifies an acquisition bottleneck in putting complete case descriptions (problem and solution) into the case-base. We assume this is due to a lack of adequate input strategies, indexing methods and knowledge management procedures. First of all, a user interface should support the graphical sketch-based workflow of architects combined with textual, schematic and tabular input strategies. Secondly, a lightweight indexing strategy is needed in contrast to the overall data storage method used. Thirdly, the problem and solution descriptions need to be stored according to the CBR paradigm. Most of the CBD prototypes do not properly implement this fundamental CBR attribute.

2.2. Sketch-based interfaces

Sketches are widely used in engineering and architectural fields as they are a familiar, efficient and natural way of expressing certain kinds of ideas. Feng et al. (2008), proposed an 2D dynamic programming approach for analyzing hand-drawn electronic circuits. Sezgin et al. (2001) introduced a system that combines multiple sources of knowledge to provide robust early processing for freehand sketching.

Sim-U-Sketch is a sketch-based interface for Simulink[1] (Kara and Stahovich, 2004) where users can construct functional Simulink models simply by drawing sketches on a computer screen. To support iterative design, Sim-U-Sketch allows users to interact with their sketches in real time to modify existing objects and add new ones.

The *COMIC* system (Os and Boves, 2003) is a large European project that studies multi-modal interactions in design applications using pen and speech. In multi-modal system methods, such as mode detection by Willems et al. (2005), are sufficient to improve the usability of these systems.

Spatial-Query-by-Sketch proposed by Egenhofer (1996) describes a visual spatial query language for geometric information systems. Yaner and Goel (2006) examines the retrieval and mapping tasks of visual analogy as constraint satisfaction problems.

2.3. Symbol spotting

A main issue in the floor plan analysis is symbol spotting. Therefore, we list some related work in symbol spotting in this section.

In the past, different pattern recognition techniques have been applied to symbol spotting. Belkasim et al. (1991), Li and Shen (1991), Adam et al. (2000) and Arajo and Kim (2007) used feature based description for the purpose of symbol spotting. Similarly symbol spotting based on structural representation of documents has been used by Lladoós et al. (2001) and Yan and Wenyin (2003). Furthermore, Tabbone et al. (2003) performed symbol spotting based on image segmentation. However, segmentation itself leads to errors, which are then propagated to the recognition.

Another idea to address symbol spotting is to use a vectorial image to spot the symbol rather than using a raster image. Messmer and Bunke (1996) and Rusiñol and Lladós (2006) used vectorized image for symbol spotting. In addition to vectorization, Rusiñol et al. (2010b)used indexing techniques to increase the efficiency of symbol spotting techniques. To further increase the scalability of the method Rusiñol et al. (2010a) used relational indexing of vectorial primitives. Dutta et al. (2011) proposed a method using hashing the shape descriptors of graph paths.

Nayef and Breuel (2010) used geometric primitives as feature points. These feature points are then searched using geometric matching algorithm. To further increase the performance as well as the accuracy of the method, Nayef and Breuel (2011) used statistical grouping for segmenting parts from line drawings. After grouping, symbol spotting is performed using the method by Nayef and Breuel (2010).

2.4. Floor plan analysis

In past, floor plan analysis has been performed for different purposes. Aoki et al. (1996) and Lladós et al. (1997) analyzed hand sketched floor plan and generated respective CAD representation. Whereas, Dosch and Masini (1999), Dosch et al. (2000), Lu et al. (2007) and Or et al. (2005) focused on analysis of 2D diagrams of floor plans so that their respective 3D model can be regenerated.

Wessel et al. (2008) proposed a method for extracting the room connectivity graphs from 3D architectural models. Based on this

[1] Simulink is an environment for multidomain simulation and Model-Based Design for dynamic and embedded systems. http://www.mathworks.com/products/simulink/: Last accessed 04/02/2010

Table 1
Overview Case-Based Design systems.

	Storage			Input			Output						
	Floor plans + text	Abstraction	Topology	Graphic	Verbal	Adaption	Reference projects	Applying solutions	Graphical Information	Learning	Subproblems	Semantic net	Analogy
Archie-II Domeshek et al. (1994)	X	X			X		X		X		X	X	
CADRE Hua et al. (1996)	X	X	X	X	X	X		X	X	X	X		
FABEL Voss (1997)	X	X	X	X	X	X	X	X	X		X		X
IDIOM Lottaz et al. (1998)			X	X	X	X		X	X				
PREC. Oxman and Oxman (1993)	X	X			X		X		X		X	X	
SEED L. Flemming (1994)		X			X	X	X	X	X		X		
SL_CB Lee et al. (2002)	X	X			X	X	X	X	X				
TRACE Mubarak (2004)		X	X	X	X		X	X	X				
CaseBook Inanc (2000)	X		X	X	X		X						
MONEO Taha (2006)	X	X			X		X		X				X
CBA Lin and Chiu (2003)	X	X			X		X					X	
DYNAMO Heylighen and Neuckermans (2001)	X	X		X	X		X		X	X		X	

connectivity graph a fast and efficient shape retrieval from an architectural database can be achieved. Macé et al. (2010) proposed a method to detect rooms in the architectural floor plan images. The method is based on a recursive decomposition of the images until convex regions are found.

Heras et al. (2011) performed the segmentation of walls from architectural floor plans. The segmented walls can be used for different purposes like, 3D reconstruction or building boundary construction.

In this paper, a novel method for automatic analysis of floor plans is introduced. It uses similar ideas as those proposed by Macé et al. (2010) and furthermore introduces new ideas like wall edges extraction, boundary detection. This general method can also be used for 3D reconstruction and generating the CAD format. However, the current focus is to extract the structure and the semantic information of the floor plan.

2.5. Graph matching

As for the retrieval algorithm, subgraph matching approaches seem to be promising. Given the problem that a query graph is formalized at run time and a database of model graphs exists a priori, techniques dealing with these conditions are of interest. Furthermore, the algorithm has to handle the fact that there might not be an exact match in the database.

Graph matching is challenging in presence of large databases (Bunke, 1997, 2000). Consequently, methods for indexing and preprocessing are essential methods. The main idea of the graph filtering is to use simple features to reduce to number of feasible candidates. Another concept clustering is used for grouping similar graphs. In principle, given a similarity (or dissimilarity) measure, such as GED (Bunke and Shearer, 1998), clustering algorithms can be applied.

Messmer and Bunke (1999) proposed a decision tree approach for subgraph matching. They are using the permutated adjacency matrix from a graph to build a decision tree. This technique is quite efficient during run time, as a decision tree is generated beforehand which contains all model graphs. Unfortunately the method has to determine all permutations of the adjacency matrices. Thus, as discussed in their experiments, the method is practically limited to graphs with a maximum 19 vertices.

3. Concept of a.SCatch

Based on the results of Langenhan and Petzold (2010), the aim of a.SCatch is to implement a system which takes advantage of the semantical information. A semantic search will be realized by sketching a concept of an architectural problem and triggering a search for similar projects of the past. Therefore several subtasks have to be solved:

1. Automatic extraction of the semantic structure of older projects,
2. extraction of the semantic structure from the sketch of the architect,
3. retrieval of similar floor plans from the repository,
4. visualization of the results and the interaction with the user interface.

The following sections will discuss the further details of subtasks of the a.SCatch system. Note that we tried to avoid using many parameters but still few exists. These parameters are optimized on validation set which include 10 images from original floor plans dataset. The test set contains the remaining 80 images which are used for evaluation of the proposed method.

3.1. Automatic extraction of the semantic structure from floor plans

The input data of our system is available in binary format.[2] First, segmentation algorithms are applied to separate the various types of

[2] The actual image size is 2479×3508. For making the analysis process more efficient, isotropic down scaling to 1413×2000 has been applied.

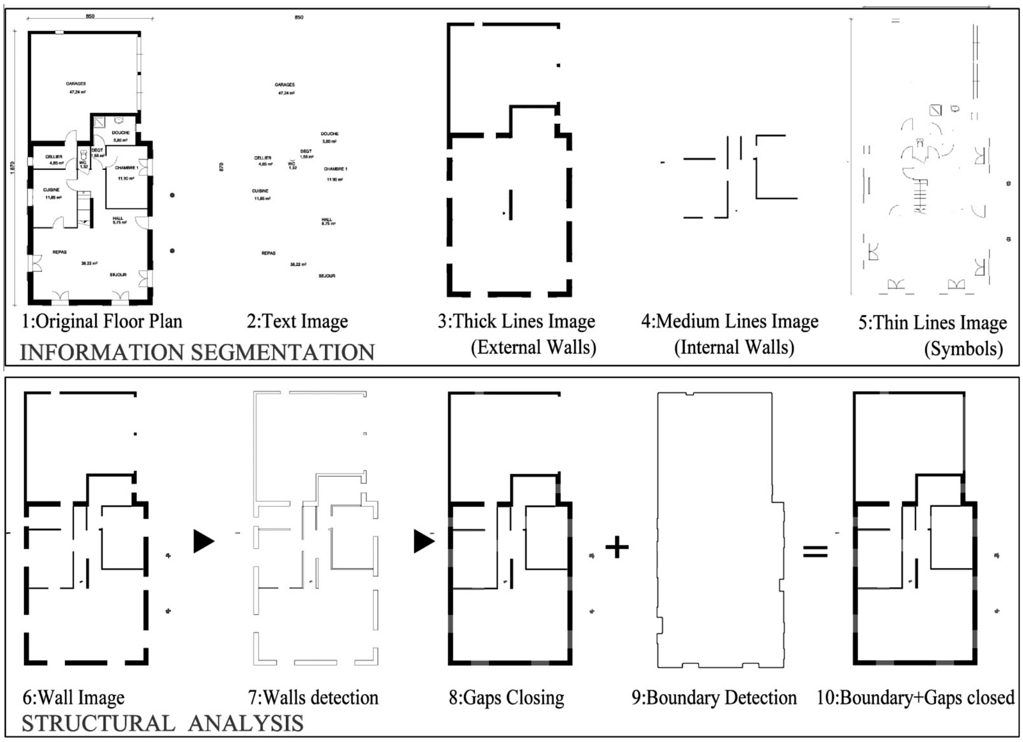


Fig. 1. Information Segmentation and Structural Analysis.

information from one another (see Section 3.1.1). Second, the structure of the extracted information is analyzed to retrieve the structure of the rooms (see Section 3.1.2). Finally, a semantic analysis is applied to retrieve the functions of the rooms, respectively (see Section 3.1.3).

3.1.1. Information segmentation

Floor plans contain information that collectively help an architect to express the actual dynamics of the building. During floor plan analysis, different types of information need to be interpreted at different points of time. Based on the divide and conquer strategy a process of information segmentation is performed. One of the key points of the proposed method is its fine segmentation of different types of information available in floor plans, e.g., walls, symbols, text etc. This is required because information, which is not needed for a specific step, is just noise and might lead to incorrect results.

First, text/graphics segmentation is performed using the methods presented by Ahmed et al. (2011b). This method is based on method by Tombre et al. (2002) with number of improvements specifically for the floor plans. Text/graphics segmentation separate the text from the graphics in the floor plan image. Text image (see Fig. 1.2) is later used in semantic analysis for extraction of room functions.

The graphics image is further segmented into walls and other building elements like door, windows, etc. Dosch et al. (2000) and Macé et al. (2010) used a thick/thin line separation algorithm for the separation of walls from the symbols. This algorithm separates the image into two images, i.e., a *thick lines image* containing the walls (both external and internal walls) and a *thin lines image* containing the symbols. We have enhanced this method by adding a third kind of lines, i.e., *medium lines*. In the proposed method, thick (external walls), medium (internal walls), and thin (symbols) lines image can be achieved by sequentially performing erosions followed by dilations. To get thick lines image, it is performed n times to remove everything but the thick walls, where n is computed empirically based on thickness of external walls from training set.

The thick line image is then subtracted from graphics image, to obtain an intermediate image which contains medium and thin lines. On the resulting image, it is performed only one time to get medium lines image. Finally, subtracting thick and medium lines image from graphic image give the thin lines images. Fig. 1 show the thick, medium, and thin line image extracted from graphics image. This fine segmentation of walls is very useful especially when it is required to save floor plan in CAD format or to do 3D reconstruction of the building. Thick lines can be used to construct the boundary of the building (see Section 3.1.2) or 3D reconstruction of the external view of building. Medium lines represents internal structure and partition of the building and can be used to reconstruct the internal structure of the building. Thin lines represent different building elements, e.g., doors, windows, sofas, etc. However, the overall building structure is represented by the walls, therefore thick and medium lines are grouped together to get the walls image (see Fig. 1.6).

3.1.2. Structural analysis

The aim of structural analysis is to extract as much structural information as possible using previously segmented information. Structural analysis begins with the detection of the walls from the wall image as mentioned above. To vectorize the wall image the following approach is used, i.e., first, contours of the wall image are extracted using the method proposed by Suzuki and Abe (1985). Second, a polygonal approximation is performed on the extracted contours using simple chaining. It is based on compressing horizontal, vertical, and diagonal lines from detected contours so that only endpoints of these lines are left, which results in the vectorial representation for each contour (where a contour correspond to a polygon). These points are later considered as corner points of the approximated polygon for each contour. Finally, the polygon is constructed for each wall using these corner points. Each polygon represents a wall in the wall image.

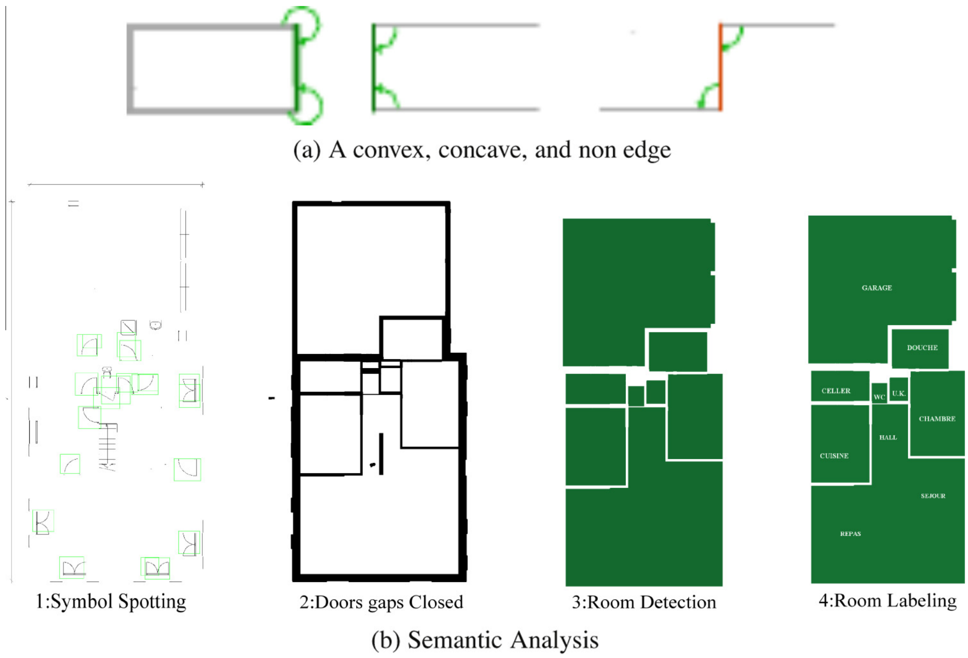


Fig. 2. Convex-Concave hypothesis, and Semantic Analysis.

The wall edges are then extracted from the detected walls to close the gaps between the walls. These gaps occur at elements like doors, windows, or sometime at gates. The process extracts all edges where those elements are likely to be found. To extract these edges we introduced convex/concave hypothesis. According to this hypothesis, a line of polygon is selected as wall edge if it is short and is either convex or concave (see Fig. 2a).

$$Walledge = \begin{cases} Convex & \text{both angles are} > 90^\circ \\ Concave & \text{both angles are} < 90^\circ \end{cases}$$

Fig. 1.7 shows the extracted wall edges from the walls image according to the convex/concave hypothesis. As a next step the gaps between the extracted edges are closed. Note, that here focus is to close only those gaps where windows or doors are likely to be found based on empirically defined thresholds T_{merge}. T_{merge} is selected based on the size of doors and windows which are inside the building and is optimized on validation set. However, gaps at the outer walls are often larger than gaps occurring at doors inside the building. In order to merge even those larger gaps we compute the outer wall image and use boundary image of the building.

Extraction of the building boundary is another key point of the proposed approach. The extracted building boundary is used to close the large gaps which are normally due to gates and can be used to get the external structure of a building.

To extract the building boundary a convex hull of the wall image is created, and the portion of the floor plan image which is inside the convex hull is extracted, neglecting everything outside. Horizontal and vertical smearing is performed on extracted image to fill the gaps between the lines corresponding to windows and gates. To remove all the lines which are not part of the building structure (often they correspond to measurements), erosion and dilation is performed on the smeared image. After removal of these lines we can directly extract the external contours of the image. These contours approximate the building boundary, i.e., the external walls. In our experiments described in Section 4.1 we show the influence of this particular processing step.

3.1.3. Semantic analysis

The aim of semantic analysis is to extract the semantic information of the floor plan. While it is easy for a human to gather this information, its automation involves a high complexity. Semantic analysis spots different building elements in the floor plan and interprets them with respect to their context. In this paper we use the speeded up robust features (SURF) (Bay et al., 2008), which is a robust, translation, rotation, and scale invariant representation method. It first extracts the key points/points of interest from the image. Then each key point is represented by a discriminative descriptor. A standard door image serves as a reference template for SURF. Mainly arc is detected by SURF, therefore it able to detect both left and right doors.

Note that some erroneous symbols have been extracted by our approach. At a later step these symbol positions are matched with the gaps found during wall edge detection. Only those results which overlap with gates are taken into account as actual doors. Fig. 2b.2 shows the image where the gaps at the doors are closed.

To detect the actual bounds of rooms, the image with the closed gaps is inverted and connected component analysis is performed on it. All of the very small connected components are removed, where as each of the remaining connected component is referred as room. The detected rooms can be found in Fig. 2b.3. After detection of rooms the next step is to define there functions like WC, Living room etc. In order to find the function of each room, the text layer from the information segmentation as well as the connected component of the room is used. In particular, all text components which lie in the boundary of a room are taken into account. After extraction of the room text, horizontal and vertical smearing is performed on the extracted text to merge the neighboring characters, resulting in the bounds for words. Using the bounding boxes all the words are rotated to a horizontal direction and OCR[3] is performed on them. The OCR result is then compared to rooms title dictionary and the closest title according to the Levenshtein distance is assigned to the room. Note that before applying the dictionary, all the digits and special characters are removed from the OCR result. The rooms which do not have any physical partition contain more than one label.

[3] Tessarect has been used for performing OCR.

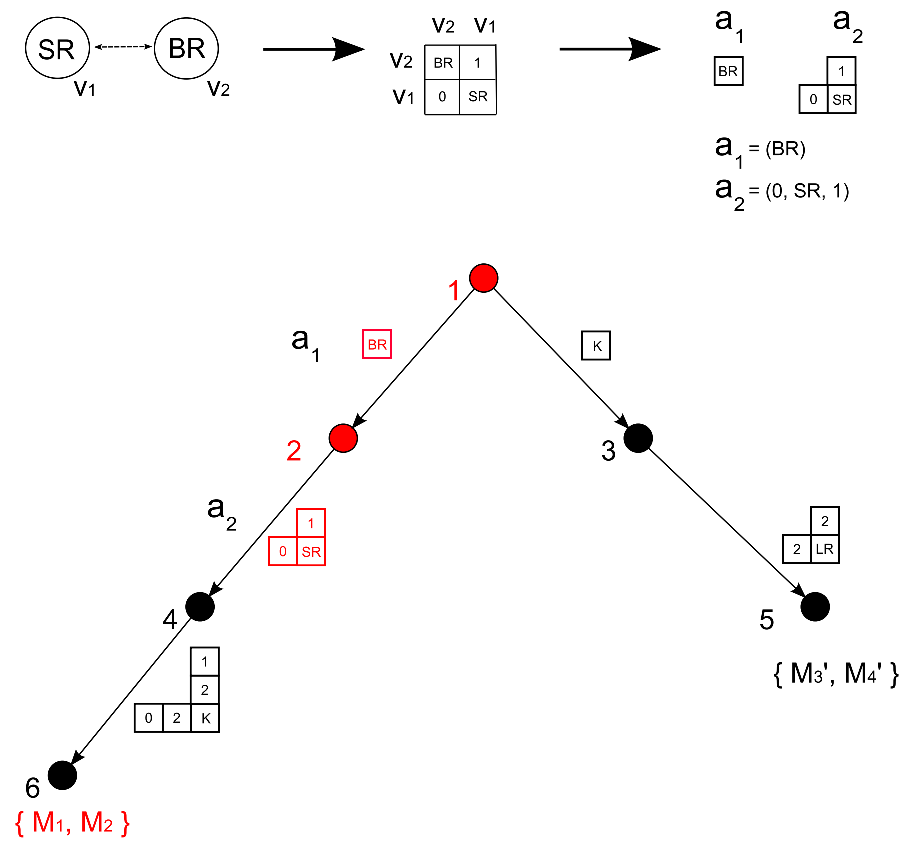


Fig. 3. The query graph with its two vertices v_1 and v_2 is represented using an adjacency matrix. The ordering of the vertices has to fulfill the well-founded order. The matrix is the split into its row column vectors a_1, a_2. These vectors will be used to find the path in the decision tree which is used for retrieval.

3.2. Sketch-based retrieval

As we are dealing with visual information, and an exclusive textual description of floor plans is too fuzzy, we propose a simplified visual query language. Whenever the architect is searching the repository, he formalizes his query as a sketch, similar to the fundamental concepts of Spatial-Query-by-Sketch proposed by Egenhofer (1996). Initially, the architect sketches a floor plan with the associated rooms, zones and units (see Fig. 6b). The respective online data of the pen device used to detect the geometrical shapes (see Fig. 6c) representing the concepts and gestures, which indicate the connection type.

Sketch recognition is performed using the following procedure. During sketching the online pen data is cached. As soon as the user stops drawing for a certain period of time, the recognition of the previous components starts. For the shape detection we used the Vision Objects shape detection.[4]

Currently we use the following visual query language:

- **Rectangles** represent **structural entities**,
- **Enclosings** imply **part-of relation**,
- **Single lines** indicate **adjacent connections**,
- **Two parallel lines** indicate **direct connections**.

The schematic abstraction is now interpreted and translated into the proposed semantic structure by Langenhan and Petzold (2010). Finally, when the user triggers the search, the scene which is composed of rectangles and lines will be analyzed. The type of the drawn entity or connection currently is defined by the user by using handwritten annotations on the rectangles. Further work is to interpret the natural symbolism of architects instead of the simplified query language.

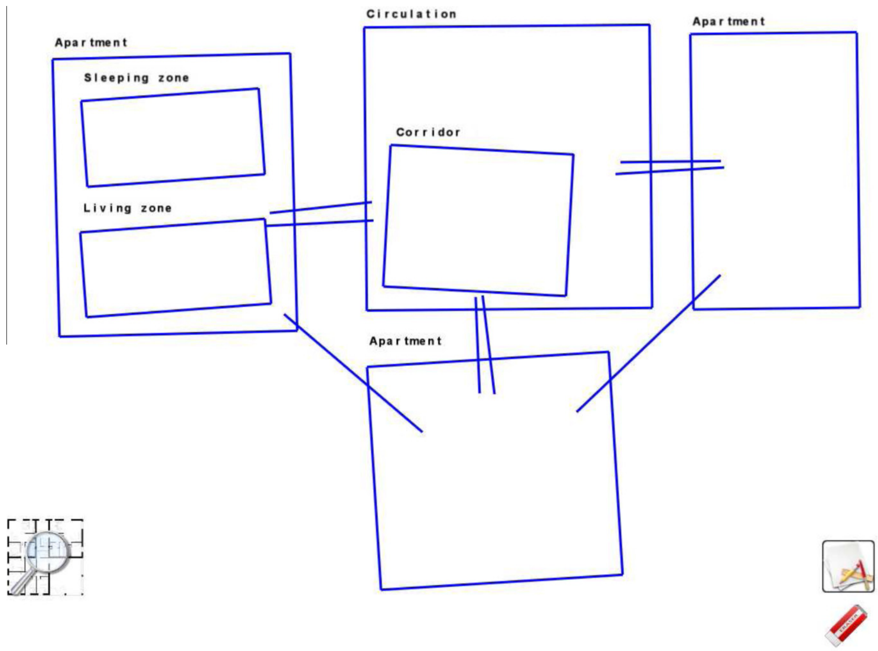
3.3. Graph structure

In this section, we discuss how the extracted graph structure can be used for the retrieval. We propose our concept for the retrieval. The extracted semantics are represented as a graph $G = (V, E)$. The vertices V have a type T_{vertex} reflecting a level, unit, zone or room. The edges E also have different types T_{edge} indicating if the vertices are connected directly or are just adjacent, both of these relations are symmetric. The *part_of* relation indicates which vertex adheres to a vertex of a superior type T_{vertex}, for instance a sleeping room which is part of a sleeping zone.

The retrieval algorithm is based on a modified version of Messmer and Bunke (1999) algorithm, where the row column vectors of the adjacency matrices are arranged in a decision tree. The modified method Weber et al. (2011) introduces a well-founded total order on the graph labels. In the original method Messmer and Bunke (1999) all permutations of the adjacency matrix, representing the graph, have to be determined. Whereas for the proposed method the permutations which violate the introduced order are not allowed. Hence the number of possible permutations is limited. The row-column vectors for each permutation of the adjacency matrix are used to compile the decision tree. The decision tree contains multiple graphs and acts as an index for real-time sub-graph matching.

So, during run-time the decision tree is loaded into memory and by traversing the decision tree, the corresponding subgraph matri-

[4] http://www.visionobjects.com/: Last accessed 10/25/2011.



(a) a.SCatch retrieval interface.

Fig. 4. a.SCatch retrieval interface.

Table 2
Room Detection results.

	Macé et al. (2010)	Proposed	No boundary det.
Detection rate (%)	85	94.88	77.47
Rec. accuracy (%)	69	81.3	86.88
One to many count	2	1.48	1.01
Many to one count	0.76	2.14	1.58

ces are classified. For the query graph Q the adjacency matrix M is determined following the constraints defined by ordering. Afterwards the adjacency matrix is split up into row-column vectors a_i. For each level i of the decision tree the corresponding row-column vector a_i is used to find the next node in the decision tree using an index structure. Fig. 3 provides an illustration of a sample query, where a sleeping room (SR) should be directly connected to a bath room (BR). Further details are presented in Weber et al. (2011) as well as experiments on graphs with up to 30 vertices.

Our current work focuses on researching approximative approaches for solving the subgraph matching problem. Furthermore, we are researching techniques to cluster the graph in our repository, by comparing floor plan graphs among each other and group similar ones together. Thus, if a query is not similar to a graph of a cluster it might also be not similar to the other graphs in the cluster.

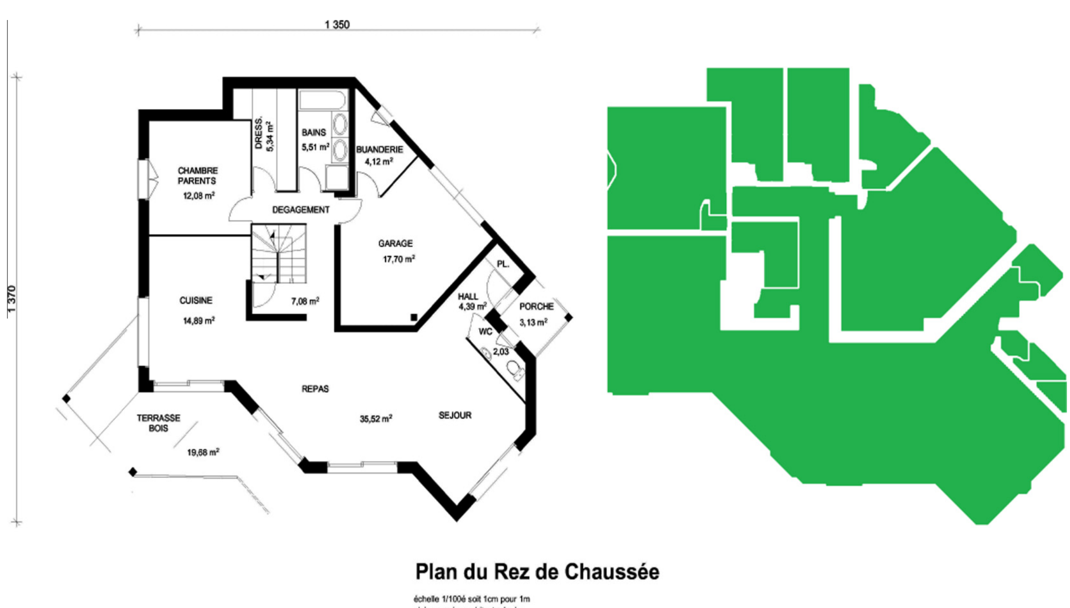


Fig. 5. Room detection result.

Table 3
Complexity and Detection rate for each query.

Query	Quadrangles		Adjacent C.		Direct C.	
1	#	Corr	#	Corr	#	Corr
1	8	0.96	2	0.95	2	0.8
2	5	0.96	2	09	3	0.9
3	8	0.93	2	0.95	5	0.7
4	6	0.95	3	0.93	1	0.7
5	3	1.0	2	1.0	1	1.0
6	9	0.99	3	0.97	5	0.96
7	6	0.98	2	0.95	4	0.65
8	9	0.99	4	0.93	4	0.9
9	10	0.97	1	0.8	4	0.9
10	3	0.97	1	0.9	2	0.9
Overall		0.97		0.93		0.86

3.4. User interface

As the pen and touch paradigm is more intuitive for the work of an architect, the prototype is implemented for the *Touch* & *Write* table. The *Touch* & *Write* table (Liwicki et al., 2010) combines the paradigm of multi-touch input and pen input. Architects prefer to sketch in their initial design phase. A pen gives them more freedom than using a mouse with Computer Aided Design (CAD) software. Using the *Touch* & *Write* pen device to draw in a digital environment allows more immediate interaction and the architects immediately benefit from the digital representation of their drawings.

The pen is an adequate tool to sketch the current architectural problem. The results of the semantic search will be represented as graphical information and the touch interaction is an intuitive metaphor for interacting with the displayed information. For example, the architect is able interact with the graphical information using simple and intuitive gestures to zoom or navigate within the floor plan.

The a.SCatch system provides tools for two purposes. First, an input interface offers the architect a possibility to edit and correct the results of the automatic room and interconnection detection discussed in Section 3.1. Here the pen device is used to frame rooms, zones or units in a floor plan.

Second, the architect can sketch a query by using the discussed visual query language. The results are displayed as images ordered according to the calculated similarity measure. The interaction with the result is done by using touch gestures (see Fig. 4).

4. Experiments

4.1. Floorplan analysis evaluation

Our system is evaluated using a data set containing original floor plan images. This data set was introduced by Macé et al. (2010) and contains the floor plan images from the period of more the ten years. The size of each floor plan image in the data set is 2479×3508. All floor plans are binarized to ensure that only structural information of the floor plans is used for the analysis (and not the color information).

In order to report the accuracy of our system, we use the protocol introduced by Phillips et al. (1999). It allows reporting *exact match* (one to one) as well as *partial matches* (one to many and many to one). For further details refer to (Phillips et al., 1999).

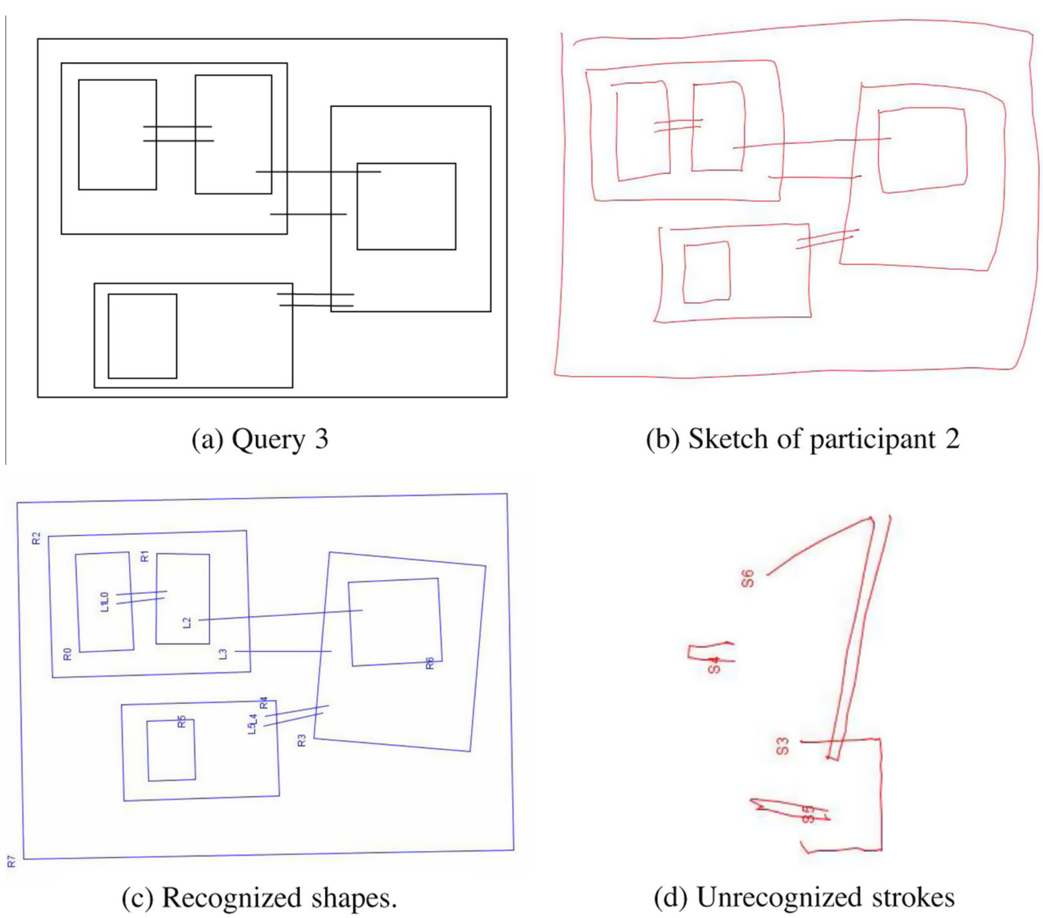


Fig. 6. Visualization of the results.

Table 4
Detection rate for each participant.

Proband	Quadrangles	Adjacent C.	Direct C.
1	1.0	1.0	0.97
2	0.96	0.82	0.81
3	0.93	0.82	0.77
4	0.91	0.95	0.74
5	0.99	1.0	0.87
6	1.0	0.9	0.97
7	0.97	0.95	0.83
8	0.97	0.95	0.84
9	0.99	0.95	0.94
10	0.97	0.91	0.84

Table 2 shows the results of rooms detection over the floor plan images dataset. The overall detection rate is 94.88%, which is approximately 10% higher than the 85% achieved in the reference system by Macé et al. (2010). More remarkably, the recognition accuracy has been improved by 12.30%. For around 20% of the images we received the recognition accuracy and detection rate both greater than 90%. In the worst case, the recognition accuracy and detection rate of our system were still 50% and 66.66% respectively. A further analysis shows the influence of the boundary detection (last column in Table 2) which was introduced in this paper. The detection rate is significantly improved.

The analysis of results in Table 2 reveals that our system has a good recognition accuracy and detection rate, along with less one to many count on average. This is because, a region is split into sub-regions only when a door or physical partition is found, in contrast to Macé et al. (2010) where regions are divided based on polygonal partitioning. To further reduce over segmentation, gap closing on the location where doors are detected need to be improved.

If there is no physical partition in the detected room no division is performed, which results in reduced segmentation overhead. Fig. 5 shows a very difficult example where large regions which do not contain any physical partition are marked as a single room. To further split these regions detailed semantic analysis can be done, in order to split large regions based on the measurement and text information available in the floor plan. This trend can also be seen in Table 2, where the *many to one count* is higher.

4.2. Retrieval evaluation

The query generation is a crucial part of the a.SCatch system, thus the first experiments focus on the visual language. We defined ten example queries covering different complexity levels (see Table 3) and and asked ten participants to copy these sketches, resulting in a total of 100 sketches. The participants were male and female students in the age between 23 and 29 years. All sketches were drawn on the *Touch* & *Write* and the handwritten strokes were recorded. To assess the pure recognition performance, we did not give a direct feedback of the recognized shapes. In order to measure the accuracy of the detection algorithm we count the correctly detected quadrangles and connections.

An example of a query taken for the evaluation is given in Fig. 6a. Furthermore Fig. 6b shows the recorded sketch of a participant. The detected shapes of the shape detection algorithm are illustrated in Fig. 6c.

For the evaluation we distinguished between detection rates for quadrangles, adjacent and direct connections. Table 3 shows the detection rate for each query. As can be seen, the detection rates for the quadrangles are very promising. Most of the other errors are due to missing quadrangles. Note that whenever a quadrangle is not detected all corresponding connections with this quadrangle will also be wrongly detected. Considering Query 2, 5, and 10, one can see that a high recognition rate of the quadrangles corresponds to a high recognition rate of the connections.

The detection rate for each participant is given in Table 4. An interesting observation is that there are persons who have a very nice way of drawing which is easily to be recognize. Other users tend to produce gaps in-between the strokes or draw very rough quadrangles, resulting in wrong recognition of the quadrangles and all corresponding connections, as illustrated in Fig. 6d.

As the interactive system acknowledges correct detected shapes, the user of the system has the chance to correct misinterpreted drawings before triggering a search. It is a good result that only one out of twenty rectangles needs to be corrected, making the sketch recognition already practically useful.

We have also performed experiments querying for reference floor plans using our graph search algorithm described in Section 3.3. The main complexity is the generation of the decision tree. Our optimization of the algorithm lead into a decrease of the tree nodes from 3.10×10^{31} to 1.09×10^{13}. More details are reported in Weber et al. (2011).

5. Conclusion and future work

In this paper we have presented an intuitive system for searching floor plans by using a sketch-based interface. Furthermore, a complete system for automatic floor plan analysis is presented which is able to extract structural and semantic content of floor plan from given image. To represent the content of floor plan a graph-based semantic structure and associated visual query language is also presented in this paper. The recognition rates of sketch recognition as well as floor plan analysis are already good for the use in practice.

Our floor plan analysis system has been evaluated on a database from the literature. We outperform previous state-of-the-art methods and achieve a perfect recognition rate on several documents. Our experiments have shown that the proposed method works very well on a large corpus of 90 floor plans. However, in practice more different types of floor plans exist. We will adopt our methods to other types of plans in our future work.

To improve the detection rates of sketches, a dynamic programming approach to combine strokes (Feng et al., 2008) or using a combination of an online and offline detection can be used. Further work for the interactive system will be to offer the architect the freedom to choose between the proposed visual query language or let him sketch an initial floor plan with his symbolism and extract the semantic graph structure directly from this sketch.

Possible improvements for floor plan analysis consist of normalization of the contours and removing graphical elements outside the outer walls. Furthermore, a weak point of our approach is that it is only able to find the physical existent rooms. This means that if there is a large region and there is no wall with in this region, it will be marked as single room. However, architects tend to divide those rooms still into several functional rooms. This division can be achieved by a more sophisticated semantic analysis based on room labeling results.

In summary, the a.SCatch system is a successful approach integrating state-of-the-art offline image processing and online sketch recognition technologies. Together with the use of recent progress in knowledge management, it provides a novel powerful tool for sketch-based document work which can be applied to other application areas as well.

Acknowledgment

The authors of this paper like to thank Joseph Llados and the members of the CVC for providing us the reference data set. This work was financially supported by the ADIWAs project.

References

Adam, S., Ogier, J., Cariou, C., Mullot, R., Labiche, J., Gardes, J., 2000. Symbol and character recognition: application to engineering drawings. IJDAR 3, 89–101.

Ahmed, S., Liwicki, M., Weber, M., Dengel, A., 2011a. Improved automatic analysis of architectural floor plans. In: ICDAR-2011 Beijing, China.

Ahmed, S., Weber, M., Liwicki, M., Dengel, A., 2011b. Text/graphics segmentation in architectural floor plans. In: ICDAR-2011 Beijing, China.

Aoki, Y., Shio, A., Arai, H., Odaka, K., Aug. 1996. A prototype system for interpreting hand-sketched floor plans. In: 13th ICPR, vol. 3, pp. 747–751.

Arajo, S.A. de, Kim, H.Y., 2007. Rotation, scale and translation-invariant segmentation-free shape gray scale recognition using mathematical morphology. In: 8th International Symposium on Mathematical Morphology, vol. 2, pp. 61–62.

Bay, H., Ess, A., Tuytelaars, T., Van Gool, L., 2008. Speeded-up robust features (surf). Computer Vision and Image Understanding 110, 346–359.

Belkasim, S., Shridhar, M., Ahmadi, M., 1991. Pattern recognition with moment invariants: a comparative study and new results. Pattern Recognition 24 (12), 1117–1138.

Bunke, H., 1997. On a relation between graph edit distance and maximum common subgraph. Pattern Recognition Letters 18 (8), 689–694.

Bunke, H., 2000. Recent developments in graph matching. In: ICPR 2, pp. 2117.

Bunke, H., Shearer, K., 1998. A graph distance metric based on the maximal common subgraph. Pattern Recognition Letters, 255–259.

Domeshek, E.A., Kolodner, J.L., Zimring, C.M., 1994. The design of a tool kit for case-based design aids. In: Third International Conference on Artificial Intelligence and Design.

Dosch, P., Masini, G., 1999. Reconstruction of the 3d structure of a building from the 2d drawings of its floors. In: ICDAR, pp. 487.

Dosch, P., Tombre, K., Ah-Soon, C., Masini, G., 2000. A complete system for the analysis of architectural drawings. IJDAR 3, 102–116.

Dutta, A., Llados, J., Pal, U., 2011. Symbol spotting in line drawings through graph paths hashing. In: ICDAR-2011 Beijing, China.

Egenhofer, M.J., 1996. Spatial-query-by-sketch. Visual languages. In: IEEE Symposium on Visual Languages, pp. 60.

Feng, G., Sun, Z., Viard-Gaudin, C., 2008. Hand-drawn electric circuit diagram understanding using 2D dynamic programming. In: Proceedings of the 11th ICFHR, pp. 493–498.

Flemming, U., 1994. Case-based design in the seed system. Automation in Construction 3 (2–3), 123–133 (Special Issue: Knowledge-based, Computer-aided Architectural Design).

Heras, L.-P. de las, Mas, J., Snchez, G., Valveny, E., 2011. Wall patch-based segmentation in architectural floorplans. In: ICDAR-2011 Beijing, China.

Heylighen, A., Neuckermans, H., 2000. Dynamo: a dynamic architectural memory on-line. Educational Technology & Society 3 (2).

Heylighen, A., Neuckermans, H., 2001. A case base of case-based design tools for architecture. Computer-Aided Design 33 (14), 1111–1122.

Hua, K., Faltings, B., Smith, I., 1996. Cadre: case-based geometric design. Artificial Intelligence in Engineering 10, 171–183 (13).

Inanc, B.S., 2000. Casebook. an information retrieval system for housing floor plans. In: Fifth CAADRIA, pp. 389–398.

Kara, L.B., Stahovich, T.F., 2004. Sim-u-sketch: a sketch-based interface for simulink. In: AVI, pp. 354–357.

Langenhan, C., Petzold, F., 2010. The fingerprint of architecture: sketch-based design methods for researching building layouts through the semantic fingerprinting of floor plans. Architecture and Modern Information Technologies 4 (13).

Lee, J., James, P., Garrett, H., Stephen, P., Lee, R., 2002. Integrating housing design and case-based reasoning. Ph.D. thesis, Carnegie Mellon University.

Li, B.-C., Shen, J., 1991. Fast computation of moment invariants. Pattern Recognition 24 (8), 807–813.

Lin, C.J., Chiu, M.L., 2003. Smart semantic query of design information in a case library. digital design – research and practice. In: Proceedings of the 10th International Conference on Computer Aided Architectural Design, Futures, pp. 125–135.

Liwicki, M., El-Neklawy, S., Dengel, A., 2010. Touch & write – a multi-touch table with pen-input. In: DAS, pp. 479–484.

Lladoós, J., Martí, E., Villanueva, J.J., 2001. Symbol recognition by error-tolerant subgraph matching between region adjacency graphs. IEEE Transaction on PAMI 23, 1137–1143.

Lladós, J., López-Krahe, J., Martí, E., 1997. A system to understand hand-drawn floor plans using subgraph isomorphism and hough transform. Machine Vision and Applications 10, 150–158.

Lottaz, C., Stalker, R., Smith, I., 1998. Constraint solving and preference activation for interactive design. Artificial Intelligence for Engineering Design, Analysis and Manufacturing 12 (1), 13–27.

Lu, T., Yang, H., Yang, R., Cai, S., 2007. Automatic analysis and integration of architectural drawings. IJDAR 9, 31–47.

Macé, S., Locteau, H., Valveny, E., Tabbone, S., 2010. A system to detect rooms in architectural floor plan images. In: DAS. ACM, New York, NY, USA, pp. 167–174.

Messmer, B., Bunke, H., 1999. A decision tree approach to graph and subgraph isomorphism detection. Pattern Recognition 32, 1979–1998.

Messmer, B.T., Bunke, H., 1996. Automatic learning and recognition of graphical symbols in engineering drawings. In: Selected Papers from the 1st GREC. Springer-Verlag, London, UK, pp. 123–134.

Mubarak, K., 2004. Case based reasoning for design composition in architecture. Ph.D. thesis, School of Architecture, Carnegie Mellon University.

Nayef, N., Breuel, T.M., 2010. Graphical symbol retrieval using a branch and bound algorithm. In: ICIP, pp. 2153–2156.

Nayef, N., Breuel, T.M., 2011. Statistical grouping for segmenting symbols parts from line drawings, with application to symbol spotting. In: ICDAR-2011 Beijing, China.

Or, S.-H., hong Wong, K., kin Yu, Y., yuan Chang, M.M., 2005. Highly automatic approach to architectural floorplan image understanding & model generation. VMV.

Os, E.D., Boves, L., 2003. Towards ambient intelligence: Multimodal computers that understand our intentions. In: eChallenges, pp. 22–24.

Oxman, R., Oxman, R., 1993. Precedents: memory structure in design case libraries. In: Proceedings of the fifth international conference on Computer-aided architectural design futures, CAAD Futures '93. North-Holland Publishing Co., Amsterdam, The Netherlands, pp. 273–287.

Phillips, I., Chhabra, A., 1999. Empirical performance evaluation of graphics recognition systems. PAMI 21 (9), 849–870.

Richter, K., Heylighen, A., Donath, D., 2007. Looking back to the future – an updated case base of case-based design tools for architecture. In: Knowledge Modelling – eCAADe.

Rusiñol, M., Borràs, A., Lladós, J., 2010a. Relational indexing of vectorial primitives for symbol spotting in line-drawing images. Pattern Recognition Letters 31 (3), 188–201.

Rusiñol, M., Lladós, J., 2006. Symbol spotting in technical drawings using vectorial signatures. In: Graphics Recognition. Ten Years Review and Future Perspectives. Lecture Notes on Computer, Science, vol. 3926, pp. 35–46.

Rusiñol, M., Lladós, J., Sánchez, G., 2010b. Symbol spotting in vectorized technical drawings through a lookup table of region strings. Pattern Analysis and Applications 13 (3), 321–331.

Sezgin, T.M., Stahovich, T., Davis, R., 2001. Sketch based interfaces: early processing for sketch understanding. In: Proceedings of the PUI 2001. ACM, New York, NY, USA, pp. 1–8.

Suzuki, S., Abe, K., 1985. Topological structural analysis of digitized binary images by border following. Computer Vision, Graphics, and Image Processing 30 (1), 32–46.

Tabbone, S., Wendling, L., Tombre, K., 2003. Matching of graphical symbols in line-drawing images using angular signature information. IJDAR 6, 115–125.

Taha, D.S., 2006. Moneo, an architectural assistant system. Ph.D. thesis, University of Alexandria, Egypt.

Tombre, K., Tabbone, S., Pélissier, L., Lamiroy, B., Dosch, P., 2002. Text/graphics separation revisited. In: Lopresti, D., Hu, J., Kashi, R. (Eds.), DAS, vol. 2423. Springer, Berlin/Heidelberg, pp. 615–620.

Voss, A., 1997. Case design specialists in fabel. In: Issues and Applications of Case based Reasoning in Design, pp. 301–335.

Weber, M., Liwicki, M., Dengel, A., 2010. a.SCAtch - A Sketch-Based Retrieval for Architectural Floor Plans. In: 12th ICFHR.

Weber, M., Liwicki, M., Dengel, A., 2011. Indexing with well-founded total order for faster subgraph isomorphism detection. In: Jiang, X., Ferrer, M., Torsello, A. (Eds.), Graph-Based Representations in Pattern Recognition. Springer, pp. 185–194.

Wessel, R., Blümel, I., Klein, R., 2008. The room connectivity graph: shape retrieval in the architectural domain. In: Skala, V. (Ed.), WSCG. UNION Agency-Science Press.

Willems, D., Rossignol, S., Vuurpijl, L., 2005. Mode detection in on-line pen drawing and handwriting recognition. In: Proceedings of the ICDAR '05, Washington, DC, USA, pp. 31–35.

Yan, L., Wenyin, L., 2003. Engineering drawings recognition using a case-based approach. In: Proceedings of the 7th ICDAR, Washington, DC, USA, pp. 190–194.

Yaner, P.W., Goel, A.K., 2006. Visual analogy: viewing analogical retrieval and mapping as constraint satisfaction problems. Applied Intelligence 25 (1), 91–105.