IJDAR (2015) 18:15–30
DOI 10.1007/s10032-014-0236-5

ORIGINAL PAPER

CVC-FP and SGT: a new database for structural floor plan analysis and its groundtruthing tool

Lluís-Pere de las Heras · Oriol Ramos Terrades · Sergi Robles · Gemma Sánchez

Received: 16 April 2014 / Revised: 16 December 2014 / Accepted: 19 December 2014 / Published online: 10 January 2015

Abstract Recent results on structured learning methods have shown the impact of structural information in a wide range of pattern recognition tasks. In the field of document image analysis, there is a long experience on structural methods for the analysis and information extraction of multiple types of documents. Yet, the lack of conveniently annotated and free access databases has not benefited the progress in some areas such as technical drawing understanding. In this paper, we present a floor plan database, named CVC-FP, that is annotated for the architectural objects and their structural relations. To construct this database, we have implemented a groundtruthing tool, the SGT tool, that allows to make specific this sort of information in a natural manner. This tool has been made for general purpose groundtruthing: It allows to define own object classes and properties, multiple labeling options are possible, grants the cooperative work, and provides user and version control. We finally have collected some of the recent work on floor plan interpretation and present a quantitative benchmark for this database. Both CVC-FP database and the SGT tool are freely released to the research community to ease comparisons between methods and boost reproducible research.

L.-P. de las Heras (✉) · O. R. Terrades · S. Robles · G. Sánchez
Departament de Ciències de la Coputació, Computer Vision Center,
Universitat Autònoma de Barcelona, Catalonia, Spain
e-mail: lpheras@cvc.uab.cat

O. R. Terrades
e-mail: oriolrt@cvc.uab.cat

S. Robles
e-mail: srobles@cvc.uab.cat

G. Sánchez
e-mail: gemma@cvc.uab.cat

1 Introduction

Current advances on structured learning methods in many pattern recognition tasks have driven to the development of new methods encoding structural information. In the field of document image analysis, there is a long experience on structural methods for information extraction and analysis of multiple types of documents [21,29,37]. With all, these systems usually need conveniently annotated databases to extract and learn the structural interrelations between objects. The lack of such available databases may constrain the research advances in some domains, which is for instance, the case of automatic floor plan understanding.

Indeed, automatic floor plan analysis is a hot topic. Despite having a main architectural design purpose, nowadays floor plans are spreading their usability into different areas. New tools help non-expert users to virtually create and modify their own house by simply drawing its floor plan on an on-line application, such as Autodesk Homestyler[1] and Floorplanner.[2] These tools can automatically generate the 3D view of a building to get an idea of how it would finally look like. More recently, Google has introduced more than 10,000 indoor floor plans in Google Maps Indoor to facilitate the mobile user navigation inside large buildings, usually airports, stations, and malls [1]. In addition, state agents with large number of properties may index floor plans by some structural information extracted from them, as individual room size of each building. This kind of indexing system would be of a great help when customers ask for specific requirements, like holding a conference or organizing musical shows.

These new applications combined with current architectural re-utilization of old designs in order to cut designing

[1] http://www.homestyler.com.

[2] http://en.floorplanner.com/.

costs have dramatically incremented interest on automatic floor plan analysis systems. Thus, a wide research has been recently undergone in this topic: Floor plans structure has been extracted from hand-made images [8,28] and computer-generated scanned images [5,12,31,40]. 3D reconstruction from 2D printed documents has been studied in [19,30,33]. Moreover, the automatic analysis contributed to other specific applications such as structural floor plan retrieval [6], and emergency evacuation simulation from complex buildings [48].

Despite the great effort of the community, automatic floor plan interpretation is far from being solved. Floor plans contain multiple structural elements, symbols, color, and text. Moreover, there is not a common notation standard to draw floor plans, which lead the styles to highly vary from architect to architect. With all, the task of creating a system able to cope with all possible notations is very challenging. Indeed, the vast majority of the recent systems only adapt well to a limited number of notations, since most of them are tested on own private collections, precluding the comparison to other systems.

Therefore, there is a current need of public collections that favor the research on floor plan understanding. These databases should preferably be representative for the existing graphical notations on building modeling. Moreover, they must grant the extraction of structural interrelations between the architectural elements. This would let the systems to learn how these elements are structurally arranged and thus to trigger better interpretation less dependent on the graphical representation of the documents.

Yet, the creation of databases entails another difficulty: the image labeling. Even though it is a straightforward procedure, the creation of groundtruth (GT) is, for the most part, tedious and slow. Thus, tools allowing complex GT generation in an efficient way are highly required to speed up this procedure and make it as lighter as possible.

In order to give solution to the problems mentioned, this paper presents the following contributions:

1. We present a new database of real floor plans (CVC-FP) and the groundtruthing tool (SGT tool) that we have used to generate its structural GT.
2. A benchmark on *wall segmentation* and *room detection* tasks is presented in the paper.

The CVC-FP database, the SGT tool, the complete benchmark results, and the evaluation scripts presented in this paper are available at the CVC-FP web page.[3] Our intention is to ease and promote the researchers to test and compare their own interpretation methods.

We have organized the paper as follows. In Sect. 2 we review existing related databases and groundtruthing tools. Then, we start by introducing the SGT tool in Sect. 3. This will allow us to explain in detail in Sect. 4 the structural content and format of the groundtruth generated. Section 5 is devoted to present the images of the four datasets that conform the CVC-FP database. We finally present benchmark in Sect. 6 and conclude the paper in Sect. 7.

2 Related work

In order to put our work into context, we briefly explain the existing databases related to floor plans analysis tasks. We subsequently overview the characteristics of the available annotation tools to generate GT in documents.

2.1 Floor plan databases

Everyday, the amount of available datasets for research purposes is increasing thanks to the collaborative work of the community. Technical committees, research centers, and universities are highly contributing by updating, maintaining, and sharing their resources [2]. Yet, we are still far away from having a wide range of representative benchmark datasets for the different scenarios in document analysis. Testing and comparing different approaches in distinct domains is limited to few well-known labeled collections. This fact sometimes can favor ad hoc systems that fit very well in the existing datasets over those ones that better fill in the large variability of the real world. For these reasons, new annotated datasets, well structured and detailed, that fill empty spaces in any research domain are always welcome.

In our area of interest, graphics recognition in documents, multiple available databases have been incorporated for the different sub-areas that it covers. These datasets can be created either by means of synthetic data generation or by real document annotation. On the one hand, synthetic databases consist of data generated by varying a predefined set of parameters to model different degrees of distortion, noise, and degradation than real documents may suffer. The generation of these sort of collections tends to be much faster than the annotated ones. In return, the model has to be closed enough to the reality to allow strong conclusions when using them. On the other hand, the annotated databases of real documents reflect the real variability of the world. However, collecting and manually groundtruthing the images can be very time demanding. This issue can be relaxed by semi-automatic annotation procedures [32].

One example of synthetic database is the GREC’2003 [44]. It was conceived in the IAPR International Workshop on Graphics Recognition in 2003 to set up a common evaluation strategy for symbol recognition. This challenge dataset

[3] http://dag.cvc.uab.es/resources/floorplans.

contains 50 cropped models from architectural and electrical documents. The primitives of these symbols are lines and arcs, which are subjected to different levels of noise, shape distortions, and linear transformations. Lately, the GREC'2011 dataset [43] was created not only as an extension of GREC'2003 in terms of recognition, but also included a symbol spotting contest in both architectural and electrical documents.

One of the most used databases for symbol recognition related tasks is the SESYD database [17]. It is a collection of labeled synthetic images. They include architectural and electrical documents for symbol spotting, recognition, and retrieval. Additional datasets for text-graphic separation and text segmentation are included. Regarding its floor plan collections, they are specifically generated for detection purposes, leaving aside the semantic assembly between symbols and the building structure.

The FPLAN-POLY database [38] is, to our best knowledge, the only available collection of annotated real floor plans. Nevertheless, it aims for symbol spotting tasks. It contains 38 symbol models in a collection consisting of 48 vectorized images.

Although there is not any floor plan database for complete analysis purposes, on other structured drawings such as flowchart diagrams, several work has been pursued on structural and semantic understanding. Thereby, the CLEF-IP initiative investigates information retrieval techniques on patent documents. One of the goals of that challenge consists of extracting the structural information from patent flowcharts in order to be queried semantically a posteriori. This process entails not only the detection and recognition of the elements participating in the diagrams (nodes, text, arrows), but also the structural assembly between them and their semantic meaning [35,39].

2.2 Groundtruthing tools

In the document analysis domain, we can find a large set of tools developed for the generation of GT. We analyze them by describing their functionality and limitations.

Most of the existing groundtruthing tools for document analysis related tasks are oriented to deal with textual documents. On the one hand, some of them address the evaluation of logical and physical layout methods, e.g., Aletheia [13], GEDI [18], TRUEVIZ [24], PinkPanther [47], and GiDoc.[4] Here, entities are represented by rectangular or polygonal regions by both physical and logical information. Physical information usually belongs to *textual* regions, *pictures*, *figures*, *tables*, etc., while logical information usually denote the semantic meaning of each physical entity in the document context, e.g., *headers*, *title*, *footnote*, etc. On the other hand, some tools focus on performance evaluation at pixel level. These tools aim at a very accurate pixel annotation and include semi-automatic labeling tools to improve the groundtruthing efficiency. Examples of these tools are the multi-platform based on Java^{TM} PixLabeler [41], and the very recent web-based tools WebGT [10] and APEP-te [26].

The specific focus of the previously cited tools hinders their usability on other document analysis tasks, e.g., graphics recognition. Some of them only allow to label rectangular segments [22,24,46]. Others delimit the definition of object categories into a small set of predefined classes [10,41]. Moreover, the definition of object dependencies usually relies on hierarchical information [26,47] and limited structural concepts [13,18], e.g., reading order and relative location. Furthermore, to our best knowledge, only [13] has a multi-layer representation that permits the labeling of fully overlapped objects.

Finally, it is worth to mention that the current tendency is to design multiuser tools that foster real-time groundtruthing cooperation either by version control [10,26] or following crowdsourcing strategies [7,22]. Moreover, the vast majority of the recent tools use slight variations based on XML for GT specification,e.g., the PAGE format [36]. This fact permits to easily adapt the existing platforms to parse GT files generated by other applications. Yet, none of the existing web-based tools use the SVG format to naturally display the GT at the web browser interfaces.

3 The structural groundtruthing tool

The SGT tool is thought to perform general purpose groundtruthing, not restricted only to one specific domain as most of the existing tools are. It grants full flexibility since the database owners can create, modify, and erase their own object classes. Additionally, it is possible to define and declare n-ary properties for the labeled objects. Thus, the groundtruth can be seen as an attributed graph representation where nodes are objects and edges are relations between them. In Fig. 1 we can see a scheme of the SGT tool architecture. The SGT tool is user-friendly, it allows two different labeling options (bounding box and polygonal), and the output is in the standard Scalable Vector Graphics (SVG). The tool is a cross-platform running on a web service, which enforces co-working without sacrificing security. It has been implemented in php5 and HTML5, and the collections are stored in a relational database like MySQL [3].

In this section, we overview the SGT tool. For a further detailed explanation we encourage the users to read the *user guide*, available in the project CVC-FP web page.

[4] https://prhlt.iti.upv.es/page/projects/multimodal/idoc/gidoc.

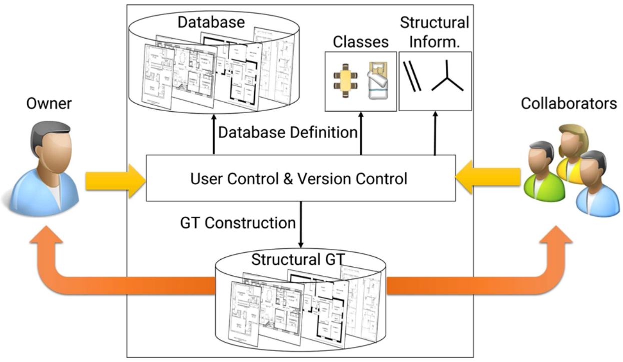
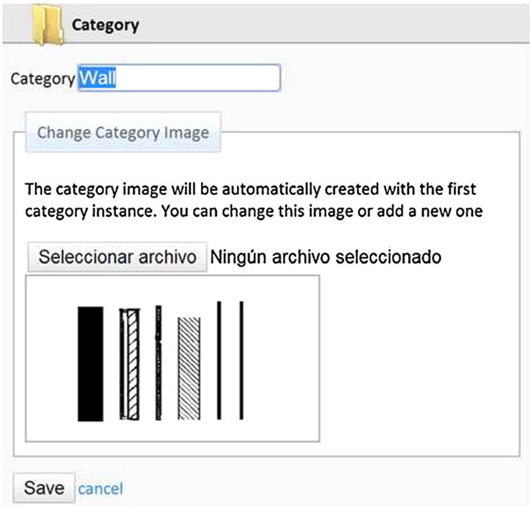


Fig. 1 Overview of SGT tool architecture. The multiple users access through a version and user control module to define and generate the GT


Category

Category Wall

Change Category Image

The category image will be automatically created with the first category instance. You can change this image or add a new one

Seleccionar archivo Ningún archivo seleccionado

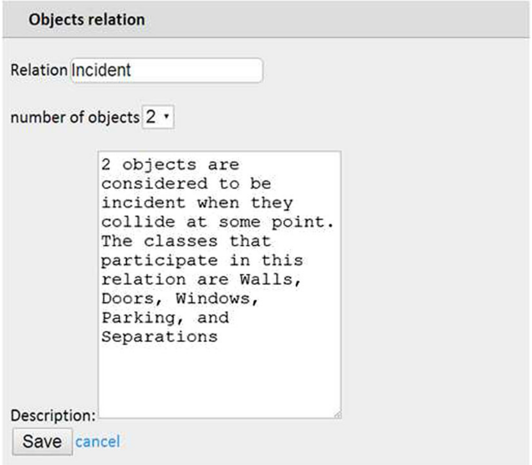
Save cancel

Fig. 2 Window for new category creation

3.1 Classes and structural relations: definitions and labeling

The SGT tool can be used in multiple domains since it allows the user to define their own object classes. For example, in the floor plan interpretation framework that we are interested in, we define object classes as *Wall*, *Room*, and *Door*. Contrarily, for symbol spotting we would rather define *Bed-type1*, *Bed-type2*, and *Shower-bath*, and for textual document layout analysis *Title*, *Legend*, and *Graphic*. The classes are defined at dataset level in a *Class Management* window, where the user can define, modify, and delete their own classes. When a new class is created an example image of the object can be added into its definition, see Fig. 2. This image is shown at labeling time to help unexperienced users in the cooperative groundtruthing task.

Object properties not only allow to define attributes for the different elements, but they also permit to declare structural and semantic dependences among multiple object instances. They are similarly defined and administered as classes at the *Relation Management* window. At definition time, the user can define the *arity* of the property: They can be specific for a single object or relating n. A brief description to help users can be also written in their definition, see Fig. 3. Their labeling is done by selecting first the desired property and then by picking those labeled objects that participate in it. The SGT tool ensures that the *arity* declared agrees with the property definition.


Objects relation

Relation Incident

number of objects 2

```
2 objects are
considered to be
incident when they
collide at some point.
The classes that
participate in this
relation are Walls,
Doors, Windows,
Parking, and
Separations
```

Description:

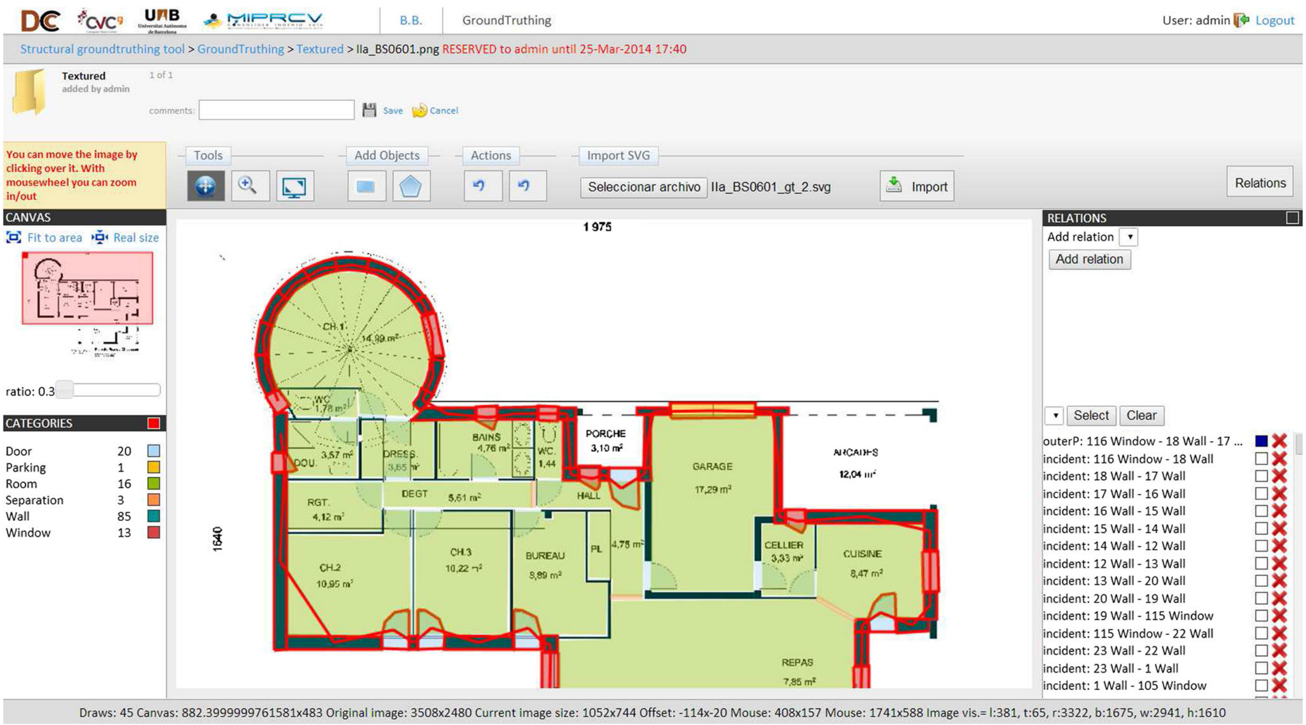
Save cancel

Fig. 3 Window for new relation creation

SGT tool facilitates the user the labeling procedure with a clear interface (see Fig. 4). Objects can be labeled either by drawing their bounding box through selecting just two corners or by drawing their polygon through a sequence of clicks. Moreover, it allows to make local zooming to ease the labeling of tiny objects. In other GT tools, the visualization and selection of the desired objects can become a challenging task in crowded images with multiple overlapping objects. Since SGT tool uses a multilayer representation for each object category, the users can display or hide object annotations at their convenience. This functionality extrapolates to object properties.

3.2 Creation and version control of a database

A registered user that uploads a collection of images to the SGT tool is its owner. Once uploaded, any registered user can participate in the groundtruthing task. They have only to select one image and start the annotation. Then, the tool will automatically avoid concurrent edition by controlling the access to the *in use* documents. For each of them, the new GT version associated with its author is stored by the versioning control system. Thereby, the database owner can track and control the whole groundtruthing procedure.



Structural groundtruthing tool > GroundTruthing > Textured > lla_BS0601.png RESERVED to admin until 25-Mar-2014 17:40

Textured
added by admin
1 of 1

comments:

You can move the image by clicking over it. With mousewheel you can zoom in/out

lla_BS0601_gt_2.svg

ratio: 0.3

Door	20	🟦
Parking	1	🟨
Room	16	🟩
Separation	3	🟧
Wall	85	🟦
Window	13	🟥

outerP: 116 Window - 18 Wall - 17 ...	■	✗
incident: 116 Window - 18 Wall	□	✗
incident: 18 Wall - 17 Wall	□	✗
incident: 17 Wall - 16 Wall	□	✗
incident: 16 Wall - 15 Wall	□	✗
incident: 15 Wall - 14 Wall	□	✗
incident: 14 Wall - 12 Wall	□	✗
incident: 12 Wall - 13 Wall	□	✗
incident: 13 Wall - 20 Wall	□	✗
incident: 20 Wall - 19 Wall	□	✗
incident: 19 Wall - 115 Window	□	✗
incident: 115 Window - 22 Wall	□	✗
incident: 23 Wall - 22 Wall	□	✗
incident: 23 Wall - 1 Wall	□	✗
incident: 1 Wall - 105 Window	□	✗

Draws: 45 Canvas: 882.3999999761581x483 Original image: 3508x2480 Current image size: 1052x744 Offset: -114x-20 Mouse: 408x157 Mouse: 1741x588 Image vis.= l:381, t:65, r:3322, b:1675, w:2941, h:1610

Fig. 4 View of the *editing* page. Among other functionalities, the user can import an existing GT, choose the labeling procedure, label objects and structural information, and select those objects and relations want to show/hide

3.3 Input images and groundtruth SVG files

Concerning the input documents, our application accepts the most common types of image formats: PNG, JPG, and TIFF. When an image is uploaded, it is stored by its file name and indexed locally in its database. The SGT tool has been implemented to support heavy files; it behaves smoothly with images around 20 mega pixels.

To make easier the exchange of classes and relations between databases, the SGT tool incorporates importing/exporting tools. For a given image called X, the tool generates an extended version of a Scalable Vector Graphics (SVG) file. We have chosen SVG for formatting our GT mainly because of three reasons: It uses a well-structured format XML-based language, it is a recommendation of W3C,[5] which ensures evolution and maintenance and, finally, allows to describe 2-dimensional vectorial graphics that are displayable in most of the Internet browsers. It is worth to notice that, since the SGT tool is web-based, the user interface is displayed at the Internet browser. Therefore, the use of SVG format permits to adapt to different browser preferences while maintaining the same labeling visualization. Obviously, the tool also allows to import external SVG files to update the GT.

The format of an extended SVG file includes own metadata information and is defined as follows. Firstly, the general information regarding the GT is specified. It includes the image dimensions (width and height) in pixels, the number of different instances labeled, the number of classes appearing in the document, and the name of all the classes that appear in the dataset. Secondly, it contains the list of the elements in the image. Each text-line describes one object by its label, its document-unique identity number, and its polygon composed by the extremity points selected by the user. Finally, the document describes the relations between the objects. Each relation is identified by its type and the identities of the elements involved.

4 The floor plan groundtruth

In this section, we review in detail the GT for the CVC-FP database constructed using the SGT tool. This GT not only contains the location of the architectural elements, but also those structural relations that we have considered to be of the interest for floor plan analysis systems. In the definition of this database we have taken into account several considerations. We have contacted a team of architects to address their needs in automatic interpretation applications. We experienced several cooperations with research and private companies aiming for different applications related to floor plan interpretation. We have considered other floor plan definitions in the literature that entail some sort of structural

[5] http://www.w3.org/Graphics/SVG/.

understanding, such is the case of [48] for evacuation building simulation, and [45] for structural floor plan retrieval. Additionally, we have also been inspired by the relevance of the structural information for high-level understanding in graphical documents, e.g., flowchart interpretation in patent documents [35]. Still, this is our own definition of the GT and it will vary for other applications, images, and experts. Obviously, since the SGT tool is shared freely, the GT data can be modified or upgraded agreeing to every system and person requirements.

Nine people working on distinct areas of graphics recognition have participated in the generation of this GT. Thanks to the version and user control of the SGT tool, the creation of the GT has been parallelized for the complete collection of images. Once the annotation has been completed, one single person has checked the correctness and consistency of the data according to the definitions settled a priori. This task has been pursued to correct different subjective perceptions for the distinct users that have participated. Since the SGT tool is designed in a way that every category and relation can be displayed with independence to the rest, this process has been easily attended. We firstly review the convention followed in the object labeling and secondly the relations instantiation.

4.1 Element labels

Let us explain how we performed the labeling of the structural symbols. These are rooms, walls, doors, windows, parking doors, and separations. The labeling of each object has been pursued by selecting that polygon that maximizes the overlapping of its area; this is by selecting each of the extremities of the object.

- *Walls* work mainly to bear the structure of buildings, to isolate, and to delimit room space. Aiming for simplicity, they are usually rectangular-shaped, generating corners at their intersections, and gaps to locate doors and windows. However, with the lack of additional architectural information, it is not clear how wall-instances should be separated. We have followed our own convention trying to stick to their structural purpose. We split walls when they have different thickness, and when they intersect at some point generating a L-shaped corner. In Fig. 5 we show a detailed example to clarify our strategy.
- The labeling of *doors*, *windows*, and *parking doors* has been much easier since their boundaries are well defined. Yet, to label those objects with curved shapes (doors and windows), we have followed a trade-off between an accurate adjustment to the boundaries and object representation simplicity. A few examples are shown in Fig. 5.
- The labeling of *rooms* sometimes encloses ambiguity as their limits are not clearly defined. An example is shown in Fig. 6, where thanks to the text and the structural shape of the building we can presume the separation between the dining room (*repas*) and the kitchen (*cuisine*), the separation between the *salon* and the *hall*, and the separation of this latter and the corridor (*degt.*), although none of them they are physically separated. On the contrary, the text also instantiates the *salon* and the *repas* to be separated habitations. This time yet, the lack of furniture and the building structure are not helping on presuming the hypothetical separation between these two rooms. Therefore, the labeling becomes subjective to the expert perception. Due to the difficulty of creating a clear convention on these situations and given the lack of additional information, each room annotation has been examined in detail a posteriori by a single person trying to keep an agreement in the whole collection of images.
- *Separations* are rectangular abstract elements that separate two neighbor rooms without physical frontiers, see Fig. 6. These elements aim to make clear the accessibility area between these rooms.

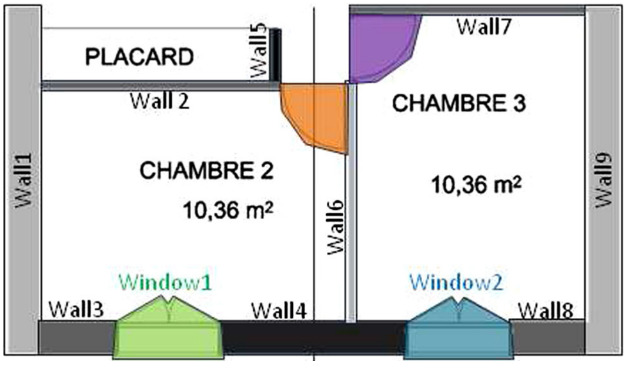


Fig. 5 Wall, door, and window labeling

4.2 Structural relations

Similar to the object properties in ontologies, the SGT tool permits the definition of relations between object instances. In other words, the SGT tool allows to define attributed graphs to enclose the mutual dependences among the labeled elements. In these graphs, the annotated elements are the nodes, whereas the contextual relations among the different objects are defined by attributed edges. This fact enriches the expressiveness our GT and allows systems to learn complex features and affinities between elements. We have defined 5 following relations:

- *Incident* Two elements are called to be *incident* when they intersect or collide at some point. An example on incident relation is shown in Fig. 7a. The elements with incident relations are *walls*, *doors*, *windows*, and *separations*.

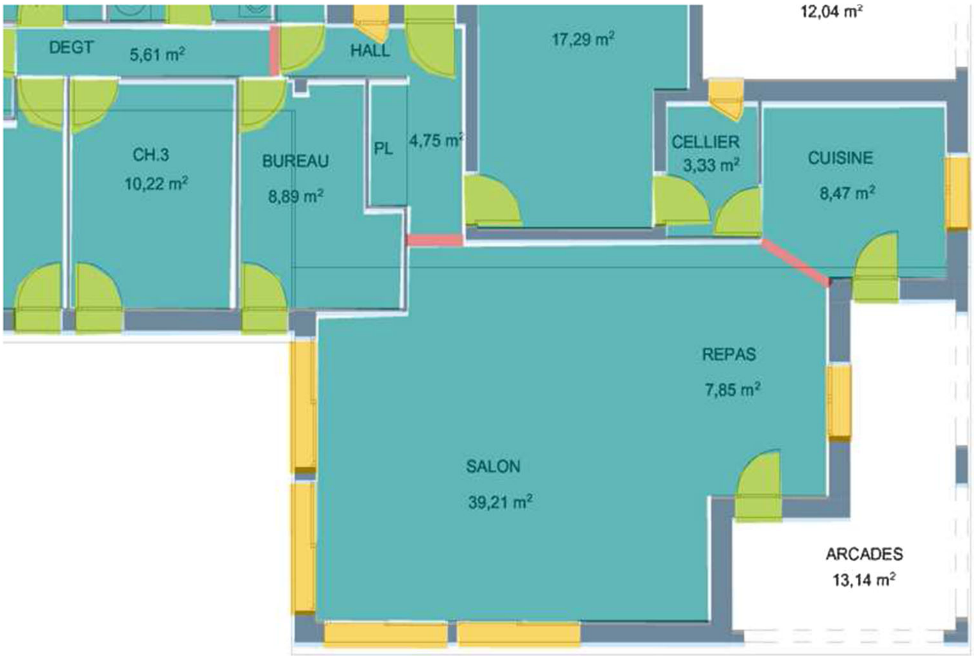


Fig. 6 Rooms labeling. *Rooms* are drawn in *turquoise* and *separations* in *red* (color figure online)

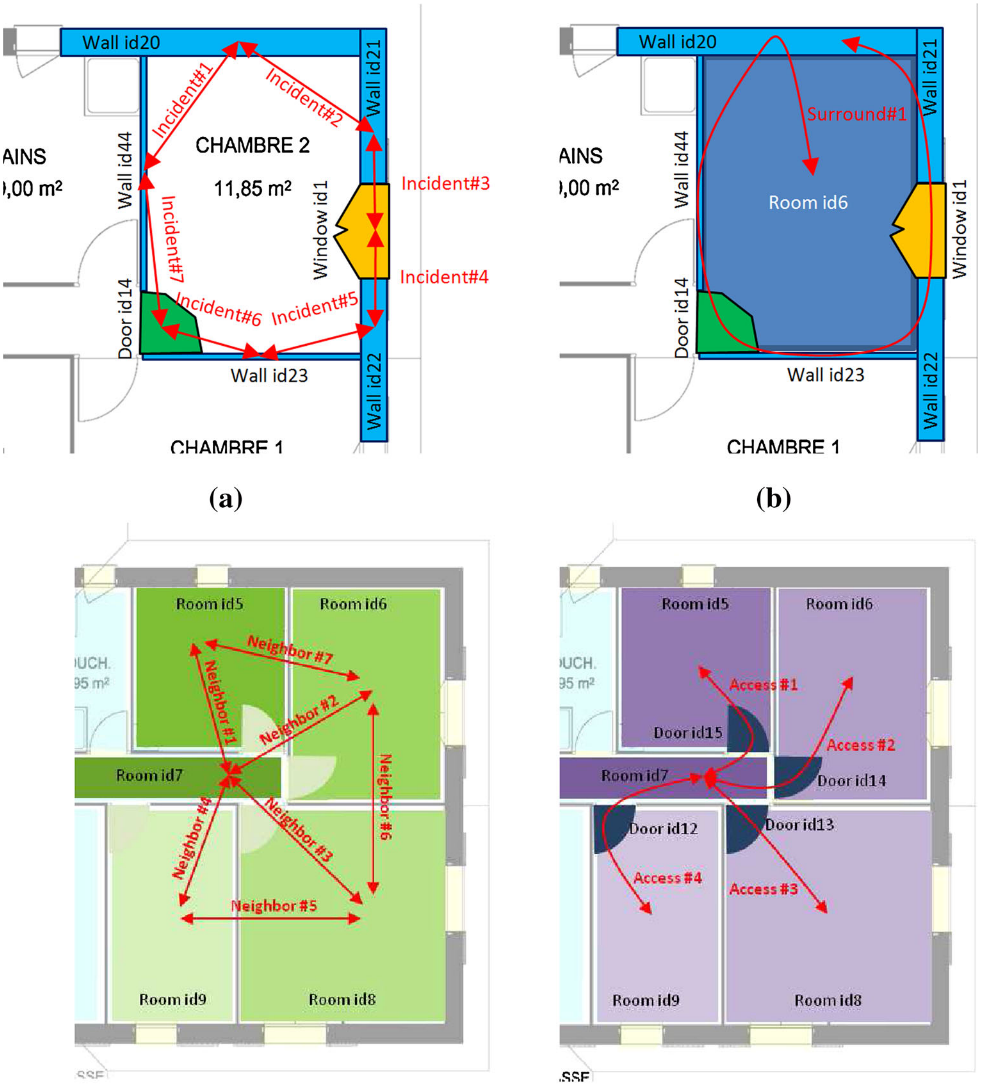
- *Surround* Several *walls doors*, *windows*, *parking entrances*, and *separations* can delimit the space of a room. The surrounding relation, as it can be seen at Fig. 7b, creates a graph of these elements connected with the room they encircle.
- *Neighborhood* Two rooms are called to be *neighbors* when they share at least one *wall*, one *door*, one *window*, or one *separation* in their surrounding perimeters. Figure 7c shows the *neighbor* graph that generates a little part of a plan.
- *Access* This relation put in correspondence two rooms that are accessible from each other through a door or a separation. It is also used for defining which rooms through which doors are possible entrances to the dwellings. Figure 7d shows the *access* graph that generates a little part of a plan.
- *Surrounding perimeter* It defines the exterior boundary of a building. It is composed by *walls*, *doors*, *windows*, *parking entrances*, and *separations*. Each isolated building only contains one *surrounding perimeter* relation.

5 The CVC-FP images

Let us now introduce the images in the CVC-FP database. This is a collection of real floor plan documents compiled and groundtruthed during the last recent years. It all started with the SCANPLAN[6] project in 2008, and still today the Document Analysis Group of the Computer Vision Centre is working on these graphical documents in multiple domains, such as structural analysis, semantic reasoning, and symbol spotting and recognition. The dataset is composed of 122 scanned documents and a partially groundtruthed version was presented in [14]. Nevertheless, these documents have been shared much before to foster the research in floor plan analysis [4,5,31].

The four sets have completely different drawing styles, image qualities and resolutions, and incorporate different sort of information. This is not an arbitrary fact; We have created a heterogeneous dataset to foster the creation of robust techniques that are able to deal with different image scenarios and graphical notations. It is important to take into account that different architects and architectural studios usually have their own graphical conventions. Therefore, there is a need of constructing systems that are able to learn each specific notation to be able to generalize for the existing architectural conventions. In addition to that, the different amount of images in each dataset permits to test the effectiveness of the proposed methodologies either when there is a large or a small set of documents available for learning purposes. We subsequently overview the characteristics of each subset separately, focusing on the structural information of the images, their symbolism, and the textual information.

[6] SCANPLAN PROJECT: http://www.eurekanetwork.org/project/-/id/4462.



(a) (b) (c) (d)

Fig. 7 Examples for the different structural relations between objects. **a** Incident relation. **b** Surround relation. **c** Neighbor relation. **d** Access relation

5.1 Black dataset

The name of this subset, as the rest does, references the graphical modeling of the walls, a thick black line as it can be seen in Fig. 8. It consists of 90 floor plan binary images of good quality. The size of these images is 2,480 × 3,508 or 3,508 × 2,480 pixels depending on the orientation of the building. These plans were conceived to sample the structural distributions of the buildings to possible customers, so they do not contain an excessive amount of technical information.

In this dataset, building drawings are centered and well oriented with respect to the document and most of the architectural lines are parallel to the horizontal and vertical axes. They model the ground floor of detached houses, usually including terraces, porches, and garages with cars. The drawing style is clear, with few elements crossing among them. Concerning the structural symbols, walls are mostly modeled by black lines of three different thicknesses whether they are main, interior, or exterior walls. Just in three plans, walls are modeled by parallel lines. Simple doors are drawn by a quarter of a circle arc, whereas building's main doors have an additional rectangular base of the size of their incident walls. Moreover, toilet doors are represented by a quarter circle arc, and double doors by two consecutive arcs centered in each of the wall limits and tangent in the center of the accessible area (see Fig. 8b). The window models can highly vary, see Fig. 8c. We can find full opened windows, partly opened windows, and sliding windows, all of them with different thicknesses. The last of the structural symbols that appear in some of these images are the stairs. They are

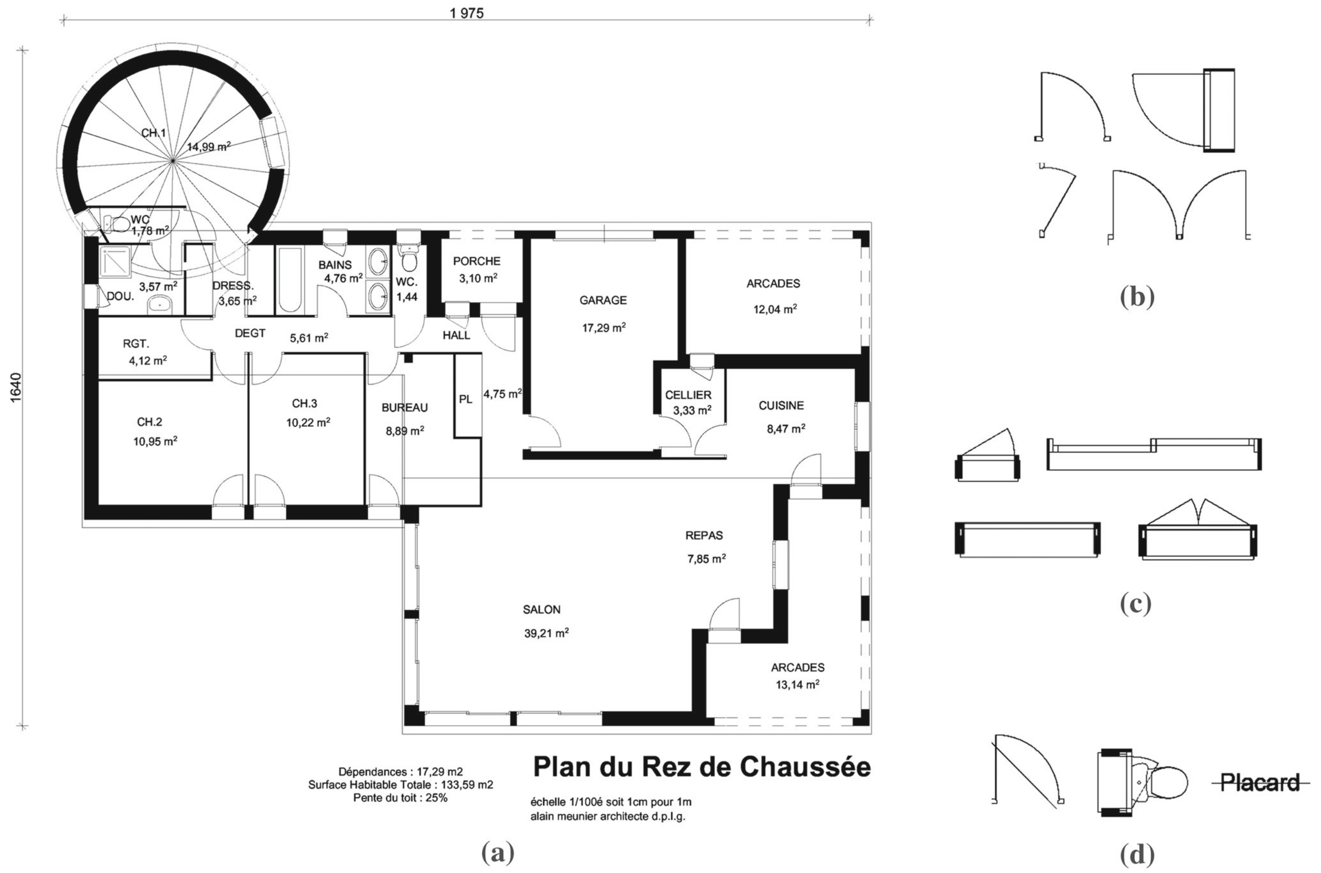


Fig. 8 Black dataset. **a** We show a sample image from this dataset. **b** We show the different types of doors, **c** the window models, and **d** some of the difficulties of the dataset

modeled by consecutive parallel rectangles. In terms of non-structural symbols, the floor plans contain mostly symbols making reference to bath utilities. Different kind of sinks, toilets, shower baths, and bathtubs are the only ones repeated in all the images. In addition to that, occasionally in some images, we can find living room furniture and in buildings delighting of a terrace or a porch may include a garden table with four chairs.

Text can be found in these documents. Each floor plan has a title with big bold letters that it can be read "Plan du Rez de Chauseé", in English "plan of the ground floor". As a subtitle we find the scale of the model (always 1/100, 1 cm is 1 m) and information about the architectural studio. In some plans, next to the title we can find information about the surface area of the dependencies, the building utile area, and the slope of the roof. Less frequently, information of the surface and the orientation of the windows is included in the subtitle. Moreover, each room encloses the text describing its functionality and area—in squared meters. Finally, each plan has two dimensions measuring in meters the rectangular surface of the building. They are located in the limits of the building perimeter.

5.2 Textured dataset

This is the second floor plan dataset compiled by the authors. It consists of ten poor quality and grayscale images whose resolutions can vary from 1,098 × 905 pixels the smallest to 2,218 × 2,227 the largest, see Fig. 9a. They are computer drawings of detached houses containing not only structural symbols but also furniture, several dimension quotes, and textual information.

Here walls are modeled by two parallel lines with a diagonal line pattern in between for the exteriors, and a heterogeneous gray-dotted pattern for the interiors. The notation of doors and stairs is exactly the same of Black Dataset. Contrarily, all the windows follow a rectangular pattern of different breadths, which can be seen in Fig. 9b. In this dataset, terraces are indicated by a repetitive pattern of squares. Regarding non-structural symbols, we mainly can find sofas, tables, and bath and kitchen utilities such as sinks, baths, and ovens. Furthermore, most of the buildings have a garage with the drawing of a car in it.

This dataset contains textual information, most of it belonging to numbers of dimension measurements. All the

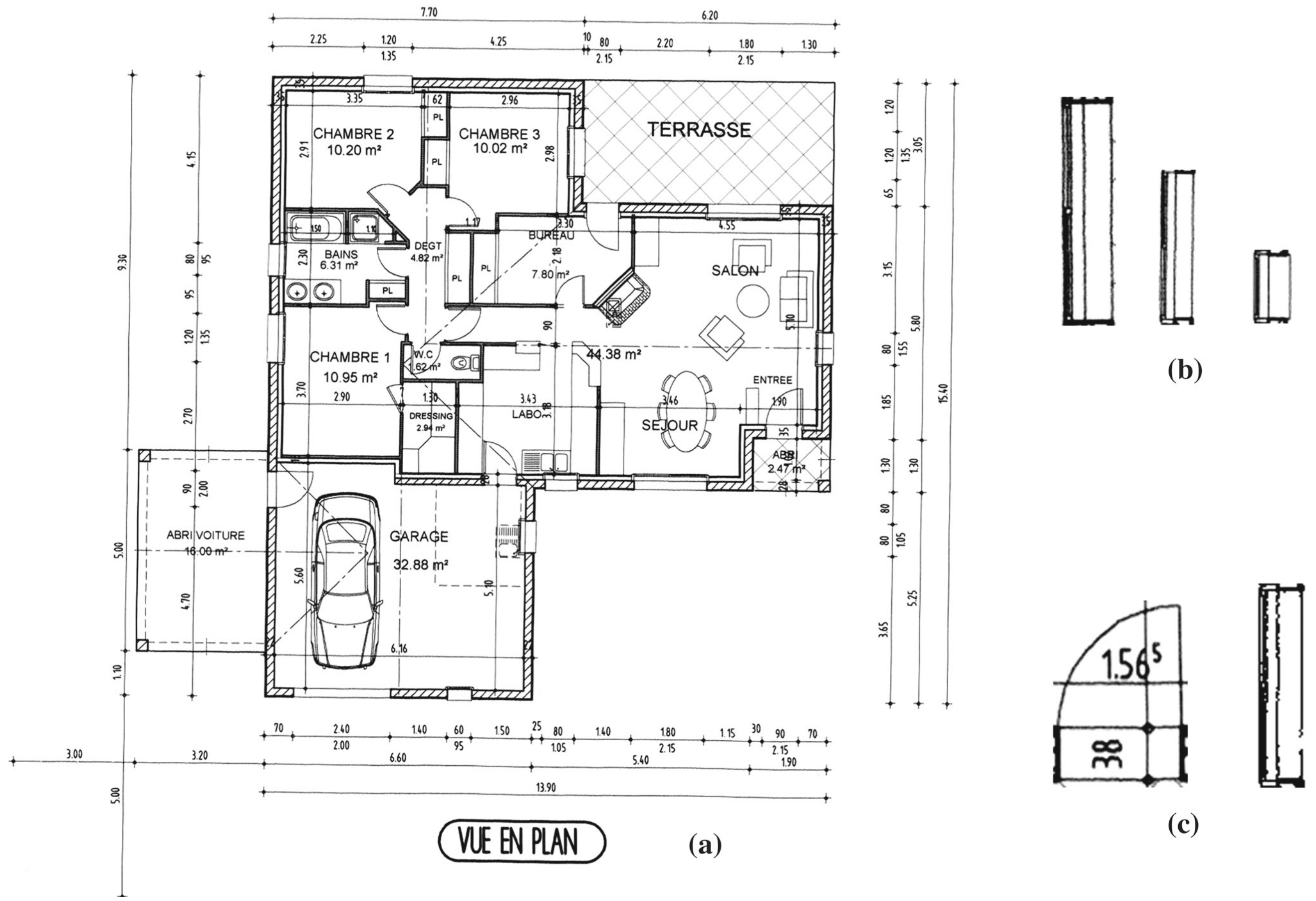


Fig. 9 Textured dataset. **a** We show a sample image from this dataset. **b** We show three different window symbols. **c** We show some difficulties in the dataset: the multiple intersection of symbols and text to the *left*, and the side effects of binarizing in poor quality plans to the *right*

rooms are labeled with their name and their area—in squared meters. Some plans have also a big bold text at the bottom of the image that says "vue en plan", in English "floor plan's view". Additionally, some extra structural information is written in barely readable text.

5.3 Textured2 dataset

The Textured2 dataset is composed by 18 images of 7,383 × 5,671 pixels collected from a local architectural project in Barcelona. The singularity of this dataset is that the 18 floor plans belong to a single building of six floors. The first image, which is shown in Fig. 10a, contains the drawings of two different floors: the one corresponding to the ground floor, and just beside the overlapping of the first, the second, and the third floors, which are identical. Similar to the first, the second image contains the plans of the basement and the 4th floor. The rest of the images contain the same drawings but not exactly located and with different sorts of information. The two first contain general structural information; this is for instance the habitable area of each floor, the area of living rooms, and the area of the sleeping rooms. The second couple of images contain the detailed architectural dimensions. The third couple includes the information of the surface materials: whether the ground is made of parked or marble and the walls are covered by either plastering or natural stone. The fourth pair of images shows the distribution of the dropped or suspended ceiling. The fifth shows the plumbing distribution, whereas the sixth displays the waste plumbing distribution. In the seventh the building's electrical installation is detailed. The eighth shows the gas installation and finally, the ninth is for heating installation.

The walls are modeled similar to the Textured Dataset, this time with a higher frequency diagonal pattern between the two parallel lines. Doors are drawn by 90° arcs and windows follow the same model. Mostly all the utilities and furniture symbols are drawn in the first couple of images: sinks, toilets, bathtubs, ovens, beds, tables, and wardrobes. Meanwhile the rest of the images enclose the different types of symbols agreeing to their architectural purpose, see Fig. 10b. The types of suspended ceilings are represented by different textural patterns, and water, electrical, gas, and heating

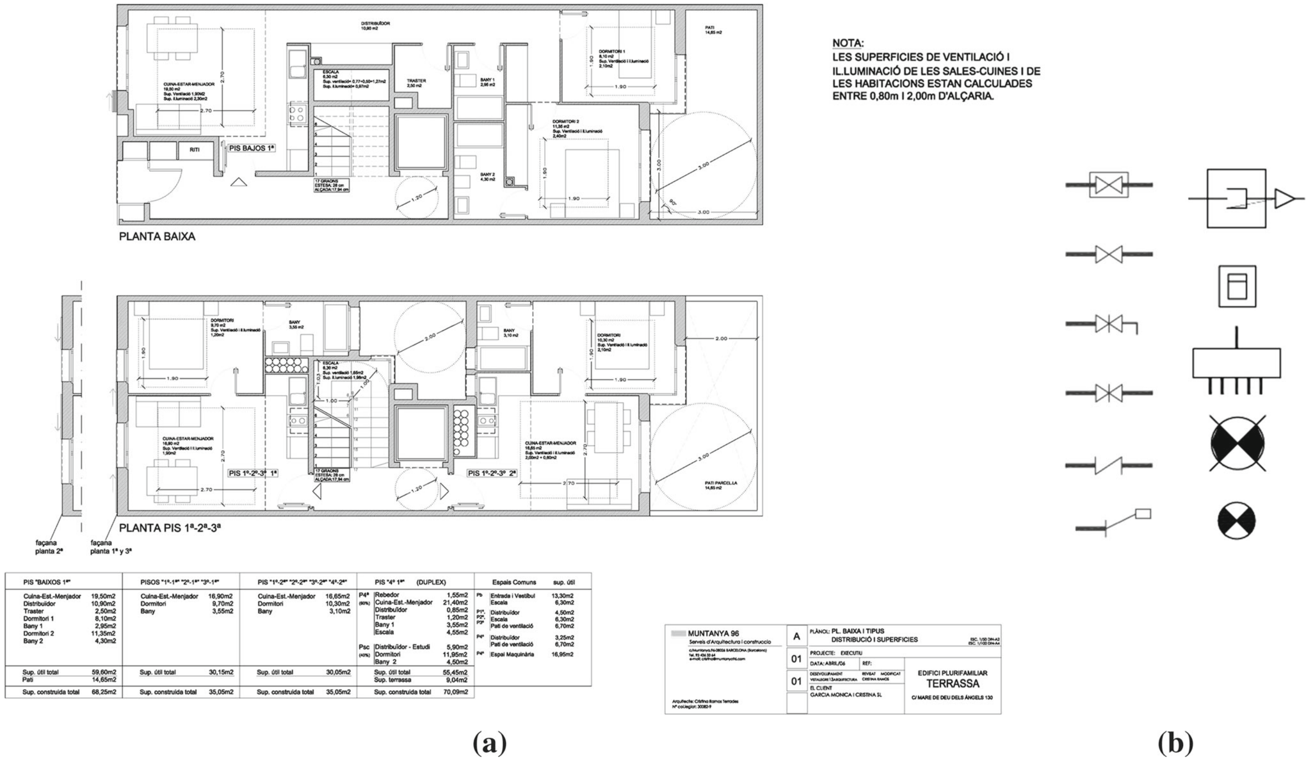

PLANTA BAIXA
PLANTA PIS 1ª-2ª-3ª
NOTA:
LES SUPERFICIES DE VENTILACIÓ I
IL.LUMINACIÓ DE LES SALES-CUINES I DE
LES HABITACIONS ESTAN CALCULADES
ENTRE 0,80m I 2,00m D'ALÇARIA.
MUNTANYA 96
EDIFICI PLURIFAMILIAR
TERRASSA
(a)
(b)

Fig. 10 Textured2 dataset. **a** We show the structural distribution of all the floors in the flat. **b** We show the different types symbols: from *left* to *right* water and electrical symbols

symbolism is specified in their respective legends. Meanwhile, textual information is omnipresent in all the images. Firstly, a text-table situated at the bottom right corner of each image specifies the information regarding the architectural studio, the project, and the document. Secondly, each document except for those enclosing the architectural dimensions contains a legend detailing the semantic meaning of the symbol encountered in the plan. Finally, every document has specific text in key positions to help its interpretation. This text includes for instance room's naming, dimensions, floor statements, walls' height, and facade orientation.

5.4 Parallel dataset

This last collection is composed by only four images and was added to perform wall segmentation on walls drawn by simple parallel lines. They are extracted from Google Images[7] using the keyword *floor plan* and are created by one single architectural studio to display the building distribution of two detached houses for sale. An instance of this dataset is shown in Fig. 11.

The binary images are of good quality and high resolution (2,550 × 3,300 pixels). As mentioned, walls are modeled by simple parallel lines, doors by a 90° arcs, and windows following a rectangular model. Some house utilities are drawn, as the usual from bath and kitchen. Moreover, since the buildings delight of a laundry room, washing and drying machines symbols can be found. Text also appears in these images. Each room has its functionality written in its perimeter. In addition, two of them, those belonging to the ground floor, have a text-table with the characteristics of the different surface areas—in squared feet.

6 Experiments

Several floor plan analysis strategies have already been applied on parts of CVC-FP while it was under construction. They have mainly been centered on wall segmentation and room detection tasks. On the one hand, rooms define the structure of the buildings, so their detection has been the main objective of these systems. On the other hand, the correct segmentation of walls usually has lead to better room detection in most of the floor plan interpretation techniques.

This section, which is slight extension of the evaluation section in [14], has been included in this paper to give completeness to the database explanation. It is divided in two main parts: *the wall segmentation task* and *the room detection task*. Both evaluations have been performed at pixel level. Since the GT is at polygon level, we have used a

[7] http://www.google.com/imghp.

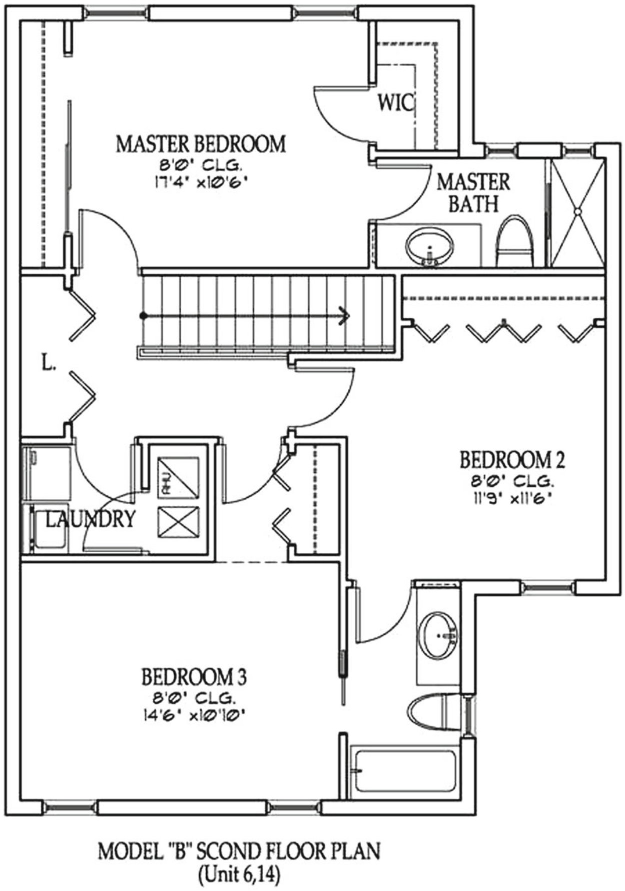


Fig. 11 Parallel dataset image

JAVA$^{\mathrm{TM}}$ script to convert every SVG file into several binary images (PNG format)—one for each object category. Additionally, we have performed a 1-way analysis of variance (ANOVA) to each task in order to asses the significant difference on the performance behavior for the different floorplan datasets [11]. This test allows to reject the hypothesis that the reference method behaves similarly on every dataset when the output p value is lower than a fixed significance value α (usually $\alpha = 0.01$ or $\alpha = 0.05$). The evaluation scripts, the *JAVA*$^{\mathrm{TM}}$ software to convert the SVG files into PNG format, and the evaluation results for every image are publicly accessible at the project website.

6.1 Wall segmentation task

6.1.1 Wall segmentation evaluation protocol

We evaluate the performance on wall segmentation using the Jaccard Index (JI). The JI score is used in PASCAL VOC competitions for object segmentation [20], and it counts the mislabeled pixels in the image. In this evaluation, *true positives* (TP), *false positives* (FP), and *false negatives* (FN) are calculated only on the black pixels of the original images binarized as Table 1 specifies. The reason is that only black pixels covey relevant information on walls. The JI is calculated as follows:

Table 1 Definition of true positives (TP), false positives (FP), and false negatives (FN) for wall segmentation evaluation

	Original image	GT image	Output image
TP	1	1	1
FP	1	0	1
FN	1	1	0

Black pixels in an image are considered as 1 and white ones as 0

$$\mathrm{JI} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}. \tag{1}$$

6.1.2 Base-line results on wall segmentation

The first attempt to detect walls on the *Black* dataset was presented in [31]. They firstly spot lines using the Hough transform on vectorized images. The lines of interest are those that are longer over a threshold or fit alignment heuristics. Then, wall hypothesis is generated by seeking parallel lines. The ones totally filled by a black texture in between are considered as walls of the floor plan.

Very differently, a much more straightforward strategy by minding wall thickness with respect to the rest of the lines was used in [4–6]. They divide the image lines in three different layers regarding their thickness using mathematic morphology. Lines on the medium and large layer correspond to walls and those in the large layer are part of the building boundary.

Finally, the authors of this paper presented two differentiated methods for segmenting walls independently to the graphical notation. In [15], a bag-of-patches approach is able to learn the graphical modeling of walls from few annotated images. Contrarily in [14], some general structural knowledge for walls modeling is used to segment possible instances. After that, the walls graphical modeling is learned out of these positive instances to spot the rest of the unrecognized walls.

A comparison for the latest wall segmentation strategies is shown in Table 2. At a glance, all the methods behave robustly well for the *Black* dataset. However, for the rest of the datasets, which contain more complex notations, their performance drops to 70 %. This is also validated by the ANOVA test, which returns a p value $p \approx 0 < \alpha$ and states that the performance of the method is significantly different regarding the dataset.

Table 2 Mean *JI* score on the *wall detection* task and *p* value of 1-way ANOVA

	[4]	[15]	[14]
BlackSet	0.90	0.97	0.97
TexturedSet	–	0.83	0.86
Textured2Set	–	0.81	0.82
ParallelSet	–	0.70	0.71
p value	–	–	≈0

6.2 Room detection task

6.2.1 Room detection evaluation protocol

We based the performance evaluation for room detection on the protocol of Phillips and Chhabra [34], which was first introduced in [31]. This protocol searches for the best alignment between the rooms segmented and the ones in the GT and allows to report the *exact* and *partial* matches.

First of all we create a *match_score* table where rows represent the rooms segmented by the system and columns are the rooms in the GT. Each table position (i, j) specifies the overlapping between the segmented room i and the groundtruthed room j. It is calculated as:

$$match\_score(i, j) = \frac{area(d[i] \bigcap g[j])}{\max(area(d[i]), area(g[j]))} \quad (2)$$

In the *match_score* table, a *one2one*/exact match is given when the overlapping score in (i, j) overcomes an *acceptance threshold*, and the rest of the row and column are below a *rejection threshold*. This means that the room segment i matches with groundtruth room j and does not match with any other. Both thresholds are set as in [5,6,31] to perform fair comparison: This is to 0.5 and 0.1, respectively. Then, the partial matches are calculated as it is described in [34] and they are divided into the following categories:

- *g_one2many*: A room in the groundtruth overlaps with more than one detected room.
- *g_many2one*: More than one room in the groundtruth overlaps with a detected room.
- *d_one2many*: A detected room overlaps with more than one room in the groundtruth.
- *d_many2one*: More then one detected room overlaps with a room in the groundtruth.

Finally, the detection rate (DR), the recognition accuracy (RA), and the *one2onerate* are calculated as follows:

$$\mathrm{DR} = \frac{one2one}{N} + \frac{g\_one2many}{N} + \frac{g\_many2one}{N}, \quad (3)$$

$$\mathrm{RA} = \frac{one2one}{M} + \frac{d\_one2many}{M} + \frac{d\_many2one}{M}, \quad (4)$$

$$one2one\ rate = \frac{one2one}{N}, \quad (5)$$

where N and M are the total number of groundtruth and detected rooms, respectively.

6.2.2 Base-line results on room detection

Rooms in [31] are detected following a shape partitioning strategy. This top-down strategy progressively splits the geometric space into convex polygons using the method proposed in [27]. Room boundaries are potentially undetected walls, doors, or windows. A postprocessing step [23] is applied at the end to reduce oversegmentation issues.

In [4–6], room detection is divided into three different steps: information segmentation, structural analysis, and semantic analysis. The first segmentation steps involve the separation between text and graphic information, and the segmentation of walls. In the structural analysis, they approximate the wall polygons out of the border segments extracted by Suzuki and Be [42]. Then, the gaps between walls from potential doors, windows, and gates are closed by horizontal and vertical smearing. Finally, in the semantic analysis, doors, windows, and rooms are extracted from the images. Here, doors and windows—which are represented by arcs—are detected using the SURF detector [9]. Rooms are closed white connected components within the dwelling boundary. They are eventually labeled by applying an OCR on the textual layer.

The authors in [14] divide the room detection into two sequential steps: (i) a statistical wall, window, and door segmentation and (ii) a structural wall and room recognition. In (i) a bag-of-patches approach based on [15,16] learns the graphical appearance and detects the potential areas where walls, doors, and windows are likely to be. Then, in (ii) it uses an own adaptation of the A* algorithm that searches well-aligned individuals in the vectorized original image. Finally, it recognizes the rooms of the building by finding closed regions using [25].

We compare the existing room detection strategies in Table 3. To do so, we have extended the results in [14] to the *Textured2* and *Parallel* datasets. According to this table, the different strategies tend to perform better in terms of RA for the *Black* and *Textured* datasets. This is caused because in the *Textured2* and *Parallel* datasets, the system [14] falls into overdetection—the *one2one* rate is lower and *one2many* count is much higher. This overdetection

Table 3 Results on the *room detection* task

	BlackSet			*TexturedSet*	*Textured2Set*	*ParallelSet*	*p* value
	[31]	[5,6]	[14]	[14]	[14]	[14]	
DR (%)	85	94.88	94.76	90.74	99.44	100	0.026
RA (%)	69	81.30	94.29	85.65	40.38	65.31	≈0
one2onerate (%)	–	–	57.68	46.84	13.51	36.80	–
one2many count	2	1.48	1.34	1.4	10.44	4.75	–
many2one count	0.76	2.14	2.24	3.4	0.16	0	–

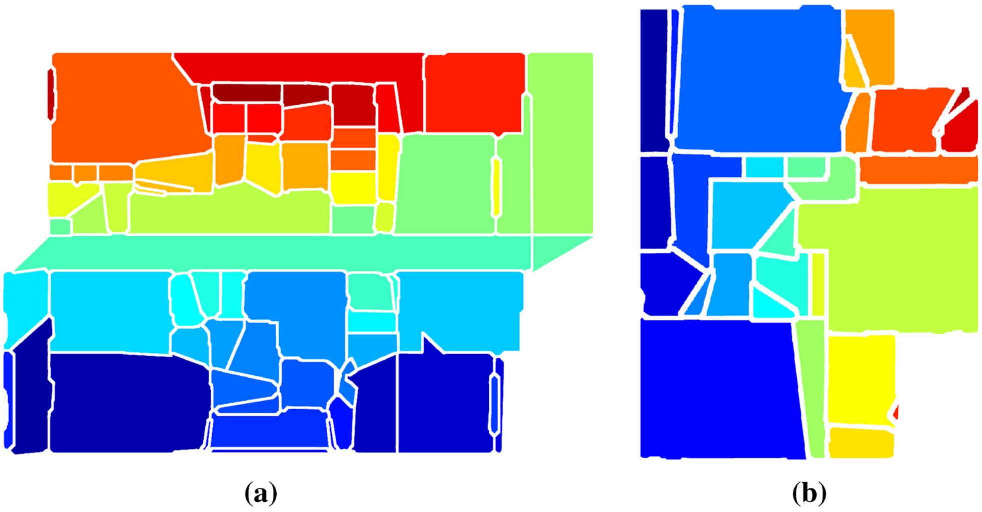


Fig. 12 Room detection for *Textured2* and *Parallel* datasets. **a** The room detection for the image shown Fig. 10a. **b** The room detection for Fig. 11. **a** *Textured2* room detection. **b** *Parallel* room detection

is mainly produced because the system relies on a simple heuristic based on relative orientation angles and distances to decide which walls, doors, and windows are correctly aligned. Moreover, in the *Textured2* dataset, this overdetection is aggravated by the fact that the images always contain two aligned flats of the building. The system enforces considering the two flats as one single building and thus tries to align possible rooms in the empty gap between the two drawings. This issue is shown in Fig. 12a. Differently, in the *Parallel* dataset, the oversegmentation of walls produces multiple possible room alignments, as it can be seen in Fig. 12b. With all, the high scores on DR for all the methods demonstrate that almost all of the rooms are detected for all the datasets. This conclusion is also supported by the ANOVA test. The *p* value obtained for the RA measure rejects the hypothesis that the method behavior is similar on every dataset. Contrarily, the *p* value for the DR measure is $p = 0.026$. Depending on the significance level considered, this hypothesis cannot be rejected.

7 Conclusion and future work

Recent results on structured learning methods have shown the impact of structural information in the performance of a wide range of pattern recognition tasks. Yet, these techniques usually need conveniently annotated databases to learn the interrelation between the objects of interest. In this paper, we have presented the CVC-FP database. It is composed of real floor plan documents that are fully annotated for architectural symbols and make specific their structural interrelations. This sort of information will let floor plan analysis systems to learn directly from observable data how the elements are structurally arranged and thus to trigger better interpretation. We have additionally presented a quantitative benchmark on the CVC-FP for two different tasks: wall segmentation and room detection.

The groundtruthing tool used to generate this database, the so-called SGT tool, is a general purpose groundtruthing tool. It is a web-based service that permits to create own objects classes and relations using a very intuitive user interface.

This tool fosters the collaboration by allowing standalone and multi-user handling, including user and version control. Thus, the SGT tool is suitable for the creation, the upgrade, and the maintenance of databases in domains where making specific additional structural information can be of great interest.

The CVC-FP database, the SGT tool, and the evaluation scripts are freely released to the research community to ease comparisons and boosting reproducible research.

Regarding future work, in a short-middle term we are planning to upgrade the SGT tool to allow the arrangement of object classes and relations in a taxonomic way. The aim is to organize and facilitate complex labeling procedures and to foster the reutilization by defining formally the groundtruth domain. This taxonomy will be defined by either creating a class and relation hierarchy in the SGT tool, or uploading a formal ontology definition. The SGT tool will be able to export the GT in an ontological framework for further semantic reasoning, e.g., to check the groundtruth consistency according its definition. In a longer term, we plan to include this ontological functionality to the tool. Hence, the SGT tool will be able to help correct and make suggestions to the user at real time.

Acknowledgments This work has been partially supported by the Spanish project TIN2012-37475-C02-02 and the research grant of the Universitat Autònoma de Barcelona 471-02-1/2010.

References

1. Google maps indoor. https://maps.google.com/help/maps/indoormaps/index.html
2. Technical committee on graphics recognition. http://iapr-tc10.univ-lr.fr/
3. Mysql: The world's most popular open source database (2014). https://dev.mysql.com/
4. Ahmed, S., Liwicki, M., Weber, M., Dengel, A.: Improved automatic analysis of architectural floor plans. In: International Conference on Document Analysis and Recognition (2011)
5. Ahmed, S., Liwicki, M., Weber, M., Dengel, A.: Improved automatic analysis of architectural floor plans. In: International Conference on Document Analysis and Recognition (2011)
6. Ahmed, S., Weber, M., Liwicki, M., Langenhan, C., Dengel, A., Petzold, F.: Automatic analysis and sketch-based retrieval of architectural floor plans. Pattern Recognit. Lett. **35**(0), 91–100 (2014). doi:10.1016/j.patrec.2013.04.005. http://www.sciencedirect.com/science/article/pii/S0167865513001554. Frontiers in Handwriting Processing
7. Amato, A., Sappa, A.D., Fornés, A., Lumbreras, F., Lladós, J.: Divide and conquer: atomizing and parallelizing a task in a mobile crowdsourcing platform. In: Proceedings of the 2nd ACM International Workshop on Crowdsourcing for Multimedia, pp. 21–22 (2013)
8. Aoki, Y., Shio, A., Arai, H., Odaka, K.: A prototype system for interpreting hand-sketched floor plans. In: Proceedings of the 13th International Conference on Pattern Recognition, vol. 3, pp. 747–751 (1996)
9. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf). Comput. Vis. Image Underst. **110**, 346–359 (2008)
10. Biller, O., Asi, A., Kedem, K., El-Sana, J., Dinstein, I.: Webgt: an interactive web-based system for historical document ground truth generation. In: 12th International Conference on Document Analysis and Recognition 0, pp. 305–308 (2013). doi:10.1109/ICDAR.2013.68
11. Casella, G., Berger, R.: Statistical Inference. Duxbury Press, Pacific Grove, CA (1990)
12. Cherneff, J., Logcher, R., Connor, J., Patrikalakis, N.: Knowledge-based interpretation of architectural drawings. Res. Eng. Des. **3**, 195–210 (1992)
13. Clausner, C., Pletschacher, S., Antonacopoulos, A.: Aletheia—an advanced document layout and text ground-truthing system for production environments. In: Document Analysis and Recognition (ICDAR), 2011 International Conference on, pp. 48–52 (2011)
14. de las Heras, L.P., Ahmed, S., Liwicki, M., Valveny, E., Sánchez, G.: Statistical segmentation and structural recognition for floor plan interpretation. Int. J. Doc. Anal. Recognit. **17**, 221–237 (2014)
15. de las Heras, L.P., Mas, J., Sanchez, G., Valveny, E.: Wall patch-based segmentation in architectural floorplans. In: 11th International Conference on Document Analysis and Recognition, pp. 1270–1274 (2011)
16. de las Heras, L.P., Mas, J., Sánchez, G., Valveny, E.: Notation-invariant patch-based wall detector in architectural floor plans. In: Graphic Recognition, Lecture Notes in Computer Science, vol. 7423, pp. 79–88 (2012)
17. Delalandre, M., Pridmore, T., Valveny, E., Locteau, H., Trupin, E.: Building synthetic graphical documents for performance evaluation. In: Liu, W. , Lladós, J. , Ogier J.-M. (eds.) Graphics Recognition. Recent Advances and New Opportunities, pp. 288–298. Springer, Berlin (2008)
18. Doermann, D., Zotkina, E., Li, H.: Gedi—a groundtruthing environment for document images. In: Ninth IAPR International Workshop on Document Analysis Systems (DAS 2010) (2010)
19. Dosch, P., Masini, G.: Reconstruction of the 3d structure of a building from the 2d drawings of its floors. In: Proceedings of the International Conference on Document Analysis and Recognition, pp. 487–490 (1999)
20. Everingham, M., Van Gool, L., Williams, C., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. Int. J. Comput. Vis. **88**, 303–338 (2010)
21. Fernández, D., Marinai, S., Lladós, J., Fornés, A.: Contextual word spotting in historical manuscripts using markov logic networks. In: Proceedings of the 2nd International Workshop on Historical Document Imaging and Processing, HIP '13, pp. 36–43. ACM, New York (2013). doi:10.1145/2501115.2501119
22. Fornés, A., Lladós, J., Mas, J., Pujades, J.M., Cabré, A.: A bimodal crowdsourcing platform for demographic historical manuscripts. In: Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage, DATeCH '14, pp. 103–108 (2014)
23. Freeman, H., Shapira, R.: Determining the minimum-area encasing rectangle for an arbitrary closed curve. Commun. ACM **18**(7), 409–413 (1975). doi:10.1145/360881.360919
24. Ha Lee, C., Kanungo, T.: The architecture of TRUEVIZ: a groundTRUth/metadata Editing and VIsualizing toolkit. Tech. rep., LAMP (2001)
25. Jiang, X., Bunke, H.: An optimal algorithm for extracting the regions of a plane graph. Pattern Recognit. Lett. **14**(7), 553–558 (1993)
26. Karatzas, D., Robles, S., Gomez, L.: An online platform for ground truthing and performance evaluation of text extraction systems. In: Proceedings of the International Workshop on Document Analysis Systems (2014)

27. Lien, J.M., Amato, N.M.: Approximate convex decomposition of polygons. In: Proceedings 20th Annual ACM Symposium Computation Geom. (SoCG), pp. 17–26 (2004)
28. Lladós, J., López-Krahe, J., Martí, E.: A system to understand hand-drawn floor plans using subgraph isomorphism and hough transform. Mach. Vis. Appl. **10**, 150–158 (1997)
29. Lladós, J., Martí, E., Villanueva, J.: Symbol recognition by error-tolerant subgraph matching between region adjacency graphs. IEEE Trans. Pattern Anal. Mach. Intell. **23**(10), 1137–1143 (2001)
30. Lu, T., Yang, H., Yang, R., Cai, S.: Automatic analysis and integration of architectural drawings. Int. J. Doc. Anal. Recognit. **9**, 31–47 (2007)
31. 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 on Document Analysis Systems, DAS '10, pp. 167–174 (2010)
32. Nakagawa, K., Fujiyoshi, A., Suzuki, M.: Ground-truthed dataset of chemical structure images in japanese published patent applications. In: Proceedings of the 9th IAPR International Workshop on Document Analysis Systems, DAS '10, pp. 455–462. ACM, New York (2010). doi:10.1145/1815330.1815389
33. Or, S.H., Wong, K.H., Yu, Y.K., Chang, M.M.Y.: Highly automatic approach to architectural floorplan image understanding & model generation. In: Proceedings of the Vision, Modeling, and Visualization, pp. 25–32 (2005)
34. Phillips, I., Chhabra, A.: Empirical performance evaluation of graphics recognition systems. Pattern Anal. Mach. Intell. IEEE Trans. **21**(9), 849–870 (1999)
35. Piroi, F., Lupu, M., Hanbury, A., Sexton, A., Magdy, W., Filippov, I.: Clef-ip 2012: retrieval experiments in the intellectual property domain. CLEF 2012 evaluation labs and workshop (Online Working Notes) (2012)
36. Pletschacher, S., Antonacopoulos, A.: The page (page analysis and ground-truth elements) format framework. In: Proceedings of the 2010 20th International Conference on Pattern Recognition, ICPR ' 10, pp. 257–260. IEEE Computer Society, Washington, DC, USA (2010). doi:10.1109/ICPR.2010.72
37. Richardson, M., Domingos, P.: Markov logic networks. Mach. Learn. **62**(1–2), 107–136 (2006). doi:10.1007/s10994-006-5833-1
38. Rusiñol, M., Borràs, A., Lladós, J.: Relational indexing of vectorial primitives for symbol spotting in line-drawing images. Pattern Recognit. Lett. **31**(3), 188–201 (2010)
39. Rusiñol, M., de las Heras, L.P., Terrades, O.R.: Flowchart recognition for non-textual information retrieval in patent search. Inf. Retr., 1–18 (2013). doi:10.1007/s10791-013-9234-3
40. Ryall, K., Shieber, S., Marks, J., Mazer, M.: Semi-automatic delineation of regions in floor plans. In: Proceedings of the Third International Conference on Document Analysis and Recognition, pp. 964–983 (1995)
41. Saund, E., Lin, J., Sarkar, P.: Pixlabeler: user interface for pixel-level labeling of elements in document images. In: ICDAR, pp. 646–650. IEEE Computer Society (2009)
42. Suzuki, S., Be, K.: Topological structural analysis of digitized binary images by border following. Comput. Vis. Graphics Image Process. **3**0(1), 32–46 (1985). doi:10.1016/0734-189X(85)90016-7. http://www.sciencedirect.com/science/article/pii/0734189X85900167
43. Valveny, E., Delalandre, M., Raveaux, R., Lamiroy, B.: Report on the symbol recognition and spotting contest. In: Graphics Recognition. New Trends and Challenges, Lecture Notes in Computer Science, vol. 7423, pp. 198–207 (2013)
44. Valveny, E., Dosch, P.: Symbol recognition contest: a synthesis. In: Graphics Recognition. Recent Advances and Perspectives, Lecture Notes in Computer Science, vol. 3088, pp. 368–385 (2004)
45. 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)
46. Yacoub, S., Saxena, V., Sami, S.: Perfectdoc: a ground truthing environment for complex documents. In: Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on, vol. 1, pp. 452–456 (2005)
47. Yanikoglu, B., Vincent, L.: Pink panther: a complete environment for ground-truthing and benchmarking document page segmentation. Pattern Recognit. **31**, 1191–1204 (1998)
48. Zhi, G., Lo, S., Fang, Z.: A graph-based algorithm for extracting units and loops from architectural floor plans for a building evacuation model. Comput. Aided Des. **35**(1), 1–14 (2003)