



HAL
open science

Interactive interpretation of structured documents: Application to the recognition of handwritten architectural plans

Achraf Ghorbel, Aurélie Lemaitre, Eric Anquetil, Sylvain Fleury, Eric Jamet

► To cite this version:

Achraf Ghorbel, Aurélie Lemaitre, Eric Anquetil, Sylvain Fleury, Eric Jamet. Interactive interpretation of structured documents: Application to the recognition of handwritten architectural plans. *Pattern Recognition*, 2015, 48 (8), 10.1016/j.patcog.2015.01.028 . hal-01238056

HAL Id: hal-01238056

<https://inria.hal.science/hal-01238056>

Submitted on 4 Dec 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Interactive interpretation of structured documents: application to the recognition of handwritten architectural plans

Achraf Ghorbel

*Université Rennes 1
UMR IRISA, Campus de Beaulieu, F-35042 Rennes*

Aurélie Lemaitre

*Université Rennes 2
UMR IRISA, Campus de Beaulieu, F-35042 Rennes*

Eric Anquetil

*INSA de Rennes
UMR IRISA, Campus de Beaulieu, F-35042 Rennes*

Sylvain Fleury

*CRPCC, Université Rennes
2, Place du recteur Henri Le Moal, CS 24 307, 35043 Rennes Cedex
Université Européenne de Bretagne, France*

Eric Jamet

*CRPCC, Université Rennes
2, Place du recteur Henri Le Moal, CS 24 307, 35043 Rennes Cedex
Université Européenne de Bretagne, France*

Abstract

This paper addresses a whole architecture, including the IMISketch method. IMISketch method incorporates two aspects: document analysis and interactivity. This paper describes a global vision of all the parts of the project. IMISketch is a generic method for an interactive interpretation of handwritten sketches. The analysis of complex documents requires the management of uncertainty. While, in practice the similar methods often induce a large combinatorics, IMISketch method presents several optimization strategies to reduce the combinatorics. The goal of these optimizations

is to have a time analysis compatible with user expectations. The decision process is able to solicit the user in the case of strong ambiguity: when it is not sure to make the right decision, the user explicitly validates the right decision to avoid a fastidious a posteriori verification phase due to propagation of errors.

This interaction requires solving two major problems: how interpretation results will be presented to the user, and how the user will interact with analysis process. We propose to study the effects of those two aspects. The experiments demonstrate that (i) a progressive presentation of the analysis results, (ii) user interventions during it and (iii) the user solicitation by the analysis process are an efficient strategy for the recognition of complex off-line documents.

To validate this interactive analysis method, several experiments are reported on off-line handwritten 2D architectural floor plans.

Key words: Structured document recognition, interactive recognition, two-dimensional grammars, uses tests, solicitation user, architectural floor plan

1 Introduction

The interpretation of structured documents consists in recognizing its constituents. These components are all the symbols that we can find in a structured document. Unlike the interpretation of isolated symbols that consists in graphically recognizing the symbol, the interpretation of symbols in a structured document requires both the graphical recognition and the structural recognition, i.e. recognizing relations between instances of symbols.

Nowadays, digital documents are becoming more and more omnipresent in our life. Many reasons, such as the flexibility provided by digital processing, have led to transform handwritten documents to digital ones.

To edit documents already drawn, we have two possibilities: either redraw the document using of specific software - unfortunately this hypothesis can become a tedious task especially in the case of numerous documents - or automatically recognize the document, to edit it later. In this paper we focus on a new approach to interactively recognize the document.

Two types of interpretation are present in the literature: *eager interpretation* [33] [28] that consists in trying to understand the structure of the docu-

Email addresses: achraf.ghorbel@irisa.fr (Achraf Ghorbel),
aurelie.lemaitre@irisa.fr (Aur lie Lemaitre), eric.anquetil@irisa.fr
(Eric Anquetil), sylvain.fleury@uhb.fr (Sylvain Fleury), eric.jamet@uhb.fr
(Eric Jamet).

ment as well as its elements during its composition, more precisely after each input stroke, and *lazy interpretation* [39] [22] that recognizes the document when its composition is finished. Our approach is an original lazy interpretation method called IMISketch. Contrary to classical methods that can require a fastidious a posteriori verification phase, IMISketch¹ system attempts to avoid this phase by integrating the user during the analysis process. As shown on Figure 1, the input of this system is a scanned image of handwritten architectural plan and after interpretation the output is its digital version. This version is able to be edited 1.

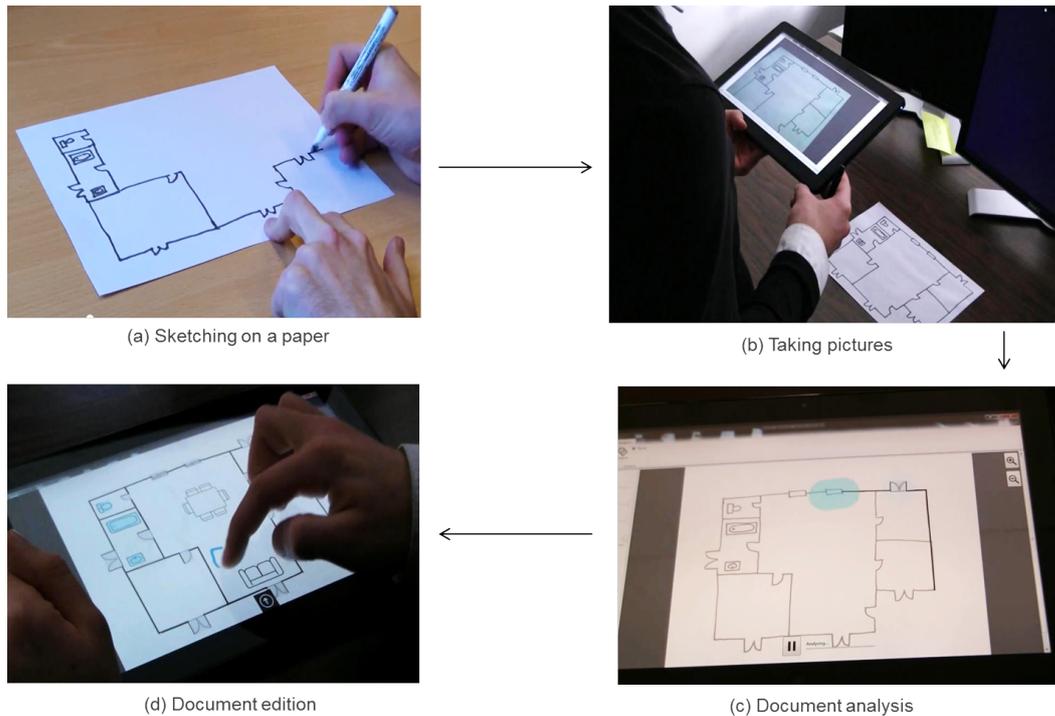


Fig. 1. IMISketch : a continuum between a technical paper and the same document in its digital interpreted form

IMISketch is based on the following characteristics:

- a generic method able to interpret structured documents of different fields (architectural floor plan, UML...) and types (handwritten, printed document...);
- an interactive method: the analyzer is able to solicit the user during the analysis;
- a hybrid method by modeling the document through two-dimensional grammars and incorporating the uncertainty through the statistics;
- a hybrid exploration by combining breadth-first and depth-first exploration according to the context.

¹ Interactive Method for Interpretation of Sketches

The IMISketch analyzer is based on a top-down analysis. The top-down analysis consists in predicting the presence of primitives in the structured document based on a priori knowledge, and then verifying their presence.

Thanks to the interactivity, the user can be solicited, if needed, by the analyzer to raise ambiguities of recognition [19] i.e. to choose between two or more possible hypotheses or to enrich the a priori knowledge of the system [17]. In fact, the user participation has a great impact to avoid error accumulation during the analysis step. This interactivity has been the topic of several studies [7]. Several questions need to be answered. We will give response of two questions: how interpretation results will be presented to the user, and how the user will interact with analysis process.

Note that the IMISketch is the result of an important work for a period of four years within a big project called "MobiSketch"²³ funded by the National Research Agency.

This paper aims at describing a global vision of all the parts of the IMISketch method. Several parts of our approach have been described in other papers [17–19]. The experiments already presented in these papers illustrate the unit validations for each part of the system. In this paper we give for the first time a complete description of the IMISketch approach and we validate the complete system considering the interpretation of complex architectural plans drawing by hand.

The remainder of this paper is organized as follows. Section 2 discusses previous work on the processing of structured document recognition. In the section 3, we introduce the architecture and the basic principles of IMISketch method. The implementation of the method is shown in section 4. The human computer interaction (HCI) is described in section 5. Experimental results are reported in section 6 and finally, conclusions are drawn in Section 7.

2 Related work

In this section, we focus on positioning our method compared to other methods of recognition based on the characteristics of IMISketch.

Several authors proposed methods of interpretation of sketches. They are usually dedicated to the interpretation of a unique type of document. Lank [28]

² The general concept of this project is illustrated in <http://youtu.be/HIV6dQHgbuw> and http://youtu.be/7divT_r7E10.

³ (<http://mobisketch.irisa.fr/>)

proposed a method to recognize the online UML diagrams. This method requires a limited number of shapes to recognize. On-line diagram recognition systems exist for a number of notations other than UML Diagram, including mathematical formulas [9], engineering drawings [26] and architecture diagrams [20]. Unlike these methods that are designed to a specific domain, IMISketch method is generic, i.e. it is able to interpret many kinds of structured documents.

In the state of the art, one interesting generic approach is the LADDER [22] [21] system which has been proposed by Hammond and Davis for interpreting a posteriori or on the fly on-line handwritten documents. LADDER language has been exploited for the design of various systems of interpretation of structured documents, such as UML [21], electrical diagrams [5] or complex graphs [23].

All the cited methods are interpretation methods of on-line structured documents. Notowidigdo [39] proposed an off-line sketch interpretation. Unlike these methods that are specific to a particular signal (on-line or off-line), our method interprets off-line handwritten structured documents as on-line documents.

We usually identify two major kinds of approaches for document analysis: syntactic and statistical approaches. Choosing one of these two approaches often depends on the document type. The syntactic approaches [8] [10] [13] [34] lean on prior knowledge of the document structure to drive the analysis. They are often based on visual languages for describing this knowledge and generating the analyzer. However, syntactic methods have difficulties to incorporate the uncertainty. Mas [35] uses a syntactic approach to describe and interpret sketched diagrams. This method is able to cope with the freedom in the drawing order of the input primitives and to cope with the distortion inherent in sketches. The statistical approaches [30] [37] provide a better ability to incorporate uncertainty but usually lack the ability to convey the hierarchical structure of the document. Several applications have applied classical pattern recognition techniques including Bayesian Networks [4] and Hidden Markov Model [12] to recognize more complex shapes. The use of statistical approaches needs a wide learning on a homogeneous and labeled base. Each type of approach has advantages and drawbacks.

The interpretation of handwritten structured documents needs on the one hand an approach that retains its structure, a syntactic approach, and on the other hand an approach that provides a better ability to incorporate uncertainty, a statistical approach. IMISketch is a hybrid method that can describe the document structure through two-dimensional grammars, and manage uncertainty through statistical formalism and a solicitation of the user.

The tests of IMISketch method are performed on 2D architectural floor plans.

The specific task of floor plan analysis has been addressed for more than twenty years. Lladós [31] proposed a method for understanding hand drawn floor plans using subgraph isomorphism and Hough transform. Aoki [6] proposed also a method for interpreting a hand-sketched floor plan. This method focuses on understanding the hand sketched floor plan and converting it into a CAD representation. Ahmed [40], also, proposed an analysis method specialized in printed architectural floor plans. Unlike these approaches which generates error propagation and thus a tedious verification phase, IMISketch method solicits the user if necessary to limit the verification phase.

3 IMISketch : an interactive analysis process

In this section, we describe the different parts of the interactive method IMISketch. IMISketch solicits the user when necessary to reduce the verification phase. IMISketch method consists of four major blocks shown in Figure 2:

- a primitive extraction block which is designed to extract the primitives of the document to be analyzed;
- a block of modelling the a priori knowledge associated with documents to recognize;
- a block of analysis tree construction which allows the exploration of possible interpretations;
- a decision process block that validates the correct interpretation either implicitly or explicitly by soliciting the user.

The following sections will detail the four main blocks of IMISketch process. The primitive extraction block is the only block that is independent of the other blocks. The block of the a priori knowledge is called by the analysis tree construction block and decision making. In fact, the analysis tree construction corresponds to the production rules modelled by the grammar, and the decision making calculates the scores based on a priori knowledge (grammar and classifier). The tree construction block is strongly dependent on making decision block. Indeed, each node created by the analysis trees block has a score determined from making decision block.

3.1 *Primitive extraction phase*

3.1.1 *Primitive choice*

The primitives are the basic elements that will power the system. The choice of these primitives depends on the type of document to recognize.

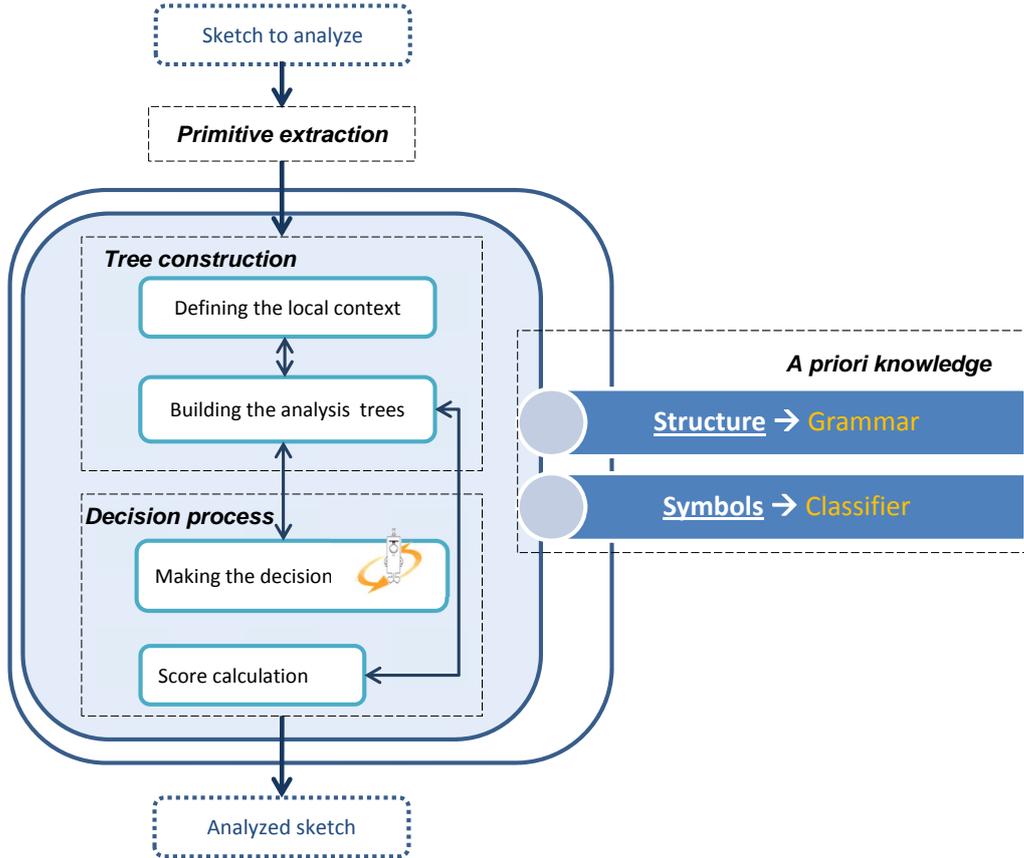


Fig. 2. IMISketch Processing

In the literature, there are methods that consider the segments and connected components [10] as primitive. Hammond [24], Lank [28] and Freeman [16] choose geometric shapes (rectangle, arc, etc.). Ablameyko [1] gives more details on the primitives and considers that segments, dotted segments, arcs, and shaded areas are sufficient to have the necessary information on the images. Zheng [42] defines the primitive form of segments, arcs and circles. Huang [25] proposes the polygons and circles to form the sets of image primitives. Messmer [36] and Notowodgodo are based on a set of segments as input for off-line recognition. Shio [41] gives more specificity to divide them into segments and thick segments.

To keep a generic nature and to give more flexibility to IMISketch, we have chosen to work only with line-segments, which represent the basic input primitives of our analysis.

3.1.2 Primitive extraction step

The first step consists in extracting the necessary information from the structured document. This phase is generic and off-line and does not depend on

the type of document to interpret.

We adopt the Kalman filter to extract these primitives [29]. The presence of curved lines generates many small line segments in a reduced zone. To reduce this large amount of primitives, we have developed a technique that allows both to keep all the extracted line segments and to retain the knowledge of connection between line-segments of the same curve. This technique involves two representations.

The rough representation, illustrated in Figure 3(a), is used to replace the chained small line-segments by the line-segment joining the ends to reduce the combinatorics: the number of line-segment in the local context decreases. This representation is used for the structural recognition of symbols. The fine representation, illustrated in Figure 3(b), gives more precision for the interaction with a classifier. This representation is used for the graphical recognition of symbols.

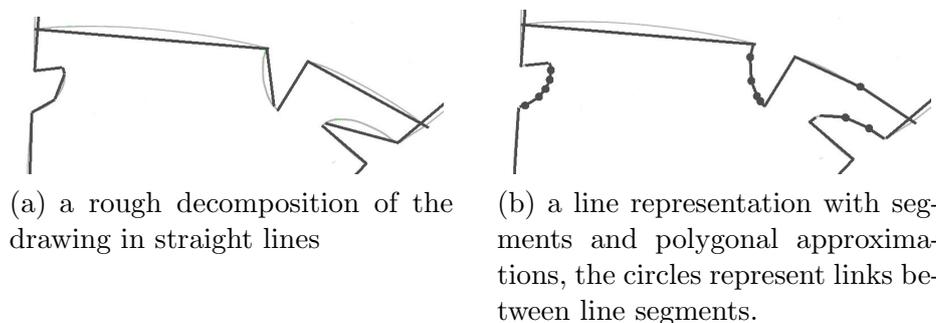


Fig. 3. Extraction of primitives. The original drawing is in light gray, and the extracted primitives are in black.

3.2 Modelling a priori knowledge

IMISketch is characterized by the use of two types of knowledge: structural and statistical.

Structural knowledge is modelled by context-driven constraint multi-set grammars (CD-CMG) [32] (cf. section 3.2.1). The objective of structural knowledge is to drive the analyzer in the two-dimensional structure of the document. Structural knowledge will also allow to call classifiers. The fusion of these two complementary types of knowledge (structural and statistical) will help to establish a robust decision making process.

3.2.1 CD-CMG grammars

In this section, we briefly describe the grammar and we illustrate the use of this grammar in our IMISketch system.

3.2.1.a CD-CMG : Definition and syntax

We have chosen CD-CMG grammars [32] designed for eager interpretation of on-line hand-drawn structured document. This grammatical formalism has not been a priori designed for lazy interpretation of off-line document.

A CD-CMG production rule consists of three blocks: preconditions, constraints and postconditions. This three blocks model a coupling of a global vision of the document (the preconditions and postconditions) with a local vision of the analyzed elements (the constraints). The general idea is to externalize the contextual knowledge that will help interpreting the document and drive the analysis process. These rules are based on the concept of document structural context (DSC). A DSC is a specific constraint modeling both a location in the document and elements that are awaited in it, now or in the sequel of the composition. A DSC is presented as follows:

$$\gamma[position]\delta[condition]$$

where :

- γ is a set of references;
- $[position]$ is a position, relatively to the references;
- δ is a set of awaited symbols;
- $[condition]$ is a subset of pixels from the elements in δ (e.g. all their pixels, their first pixel, their left pixel, any of their point, etc.); if these points are in this specified location, then the constraint succeeds.

Definition of postconditions

In CD-CMG, when a production reduction occurs, it means that a multiset of elements has been replaced by another one. This reduction has also impacts on the modeling of the document. The syntax of postcondition DSC is the one presented as follows :

$$\{\gamma[position]\delta[condition] \Rightarrow [\alpha \rightarrow \beta]\}^q$$

where :

- γ is symbol instance;
- $[position]$ is a position, relatively to the references;
- δ is symbol class;
- α, β are symbol classes or instances;
- q is the number of times that this DSC can be used.

Definition of preconditions

CD-CMG preconditions model the DSC, created in some production post-conditions, which have to be satisfied. Informally, preconditions are a set of constraints ensuring that the elements are, from a document point of view, in a consistent structural context with the production.

Definition of constraints

CD-CMG constraints model a local vision of the β elements. Constraints can have two main purposes: on the one hand, check if it is pertinent to reduce β into α (these are semantic constraints), and, on the other hand, decide if the shape of the β elements is consistent with the production (these are recognition constraints).

3.2.1.b Example of implementation of CD-CMG in IMISketch

The figure 4(a) illustrates an example of a rule for recognizing furniture. The *FurniturePart*, which is a set of primitive too close, of furniture is transformed into furniture if and only if:

- The preconditions bloc is validated. In this case, all the furniture is in the document.
- All the constraints are validated. In this example the constraints are the results of the classifier on the part of the furniture.

In the same manner, the figures 4(b) and 4(c) present the rules used to create a part of furniture.

Therefore, adaptations and improvements were necessary to adapt this formalism to the off-line lazy recognition. These improvements focus on solving problems of combinatorial explosion. Indeed, in on-fly the interpretation, the analyzer interprets a primitive (stroke) in a already interpreted document. In lazy interpretation, we have several primitives to interpret in the context of a

```

Rules : FurnitureCreation
Furniture: FurRes → FurniturePart: fp
Preconditions:
  {Document[in] fp [all]}
Constraints:
  ClassifierFurniture(fp)

```

(a) Example of CD-CMG grammar for creating furniture

```

Rules : SequenceFurniturePartCreation
FurniturePart: FurPartRes → FurniturePart: fp, primitive: p
Preconditions:
Or{
  {fp[ExtendedRelativePosition] p [one]}
  {fp[ReducedRelativePosition] p [one]}
}
Constraints :
  SizeFurniture(p)
Postconditions:
  {FurPartRes[ExtendedRelativePosition] (primitive: pr1)}[one]
  → [FurPartRes → FurPartRes, pr1]
  {FurPartRes[ReducedRelativePosition] (primitive: pr1)}[one]
  → [FurPartRes → FurPartRes, pr1]

```

(b) A grammar rule that transforms a primitive into a furniture part

```

Rules : SequenceFurniturePartCreation
FurniturePart: FurPartRes → FurniturePart: fp, primitive: p
Preconditions:
Or{
  {fp[ExtendedRelativePosition] p [one]}
  {fp[ReducedRelativePosition] p [one]}
}
Constraints :
  SizeFurniture(p)
Postconditions:
  {FurPartRes[ExtendedRelativePosition] (primitive: pr1)}[one]
  → [FurPartRes → FurPartRes, pr1]
  {FurPartRes[ReducedRelativePosition] (primitive: pr1)}[one]
  → [FurPartRes → FurPartRes, pr1]

```

(c) A grammar rule that transforms a primitive and a furniture part into a furniture part

Fig. 4. Grammar rules creating furniture

partially recognized document. All the primitives can be interpreted in several ways which generates a very large combinatorial.

The first adaptation of CD-CMG is to control the type of exploration [18]. We propose a dynamic strategy to switch between a breadth-first exploration and a depth-first exploration to reduce the combinatorics. We improve the use of

the existing CD-CMG grammar to drive this new strategy analysis. In this hybrid strategy, CD-CMG is not only used to the modelling of the document but also used for the choice of the exploration strategy: either breadth-first exploration or depth-first exploration.

- *depth-first exploration*: this exploration strategy is chosen if the production rule applied at the root of the analysis tree generates only one principal way to interpret the other (interconnected primitives).
- *breath-first exploration*: if the interpretation of the root of the analysis tree generates several ways to interpret the primitives.

As we have said previously the depth-first exploration may not generate all the hypotheses. Consequently, we propose to reduce the risks by limiting the possible zone of application of the depth-first analysis. We implement this idea using the concept of *relative position*. The relative position is the search zone that is created after the interpretation of each element in order to continue the analysis. In this analyzer, we combine two kinds of relative positions : *reduced relative position* and *extended relative position*(Figure 5).

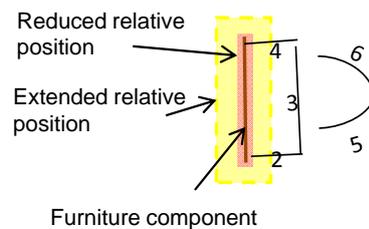


Fig. 5. Example of two relative positions for a furniture component

- *Extended relative position*: when this position is activated, the analyzer adopts the classical breadth-first exploration.
- *Reduced relative position*: this search zone is smaller and enables to adopt the depth-first exploration. This position is intended to collect interconnected primitives those are very close.

In the grammatical description, each interpreted element can create the both kinds of relative positions. Thanks to these positions, the hybrid exploration is totally led by the grammatical description and can be adapted for the description of each element present in the document. If a created element is associated to two positions, the reduced position is used in priority. The transition from reduced position to extended position is established only after no rule of production is applicable. The figure 6 shows the same production rule described in figure 4(a), but after improving the type of exploration. The designer assumes that the rule which transforms a part of furniture into furniture should be in competition with other hypotheses. For this, the rule is labeled *breadth-first*.

Rules : FurnitureCreation Furniture: FurRes \rightarrow FurniturePart: <i>fp</i> Preconditions: {Document[in] <i>fp</i> [all]} Constraints : ClassifierFurniture(<i>p</i>) Postconditions:

Fig. 6. Example of optimization of CD-CMG grammar for creating furniture

3.2.2 Classifier

A CD-CMG production rule can call an external classifier to recognize the symbols. This classification system is based on first-order Takagi-Sugeno (TS) fuzzy inference system [2]. This classifier takes as input a set of primitive points and associates a label to each symbol. The classifier is able to reject out-layer or confused input. Each recognition is associated to a score of confidence.

To describe the symbol, we rely on the HBF49 characteristics [11]. HBF49 is a unique set of features for the representation of hand-drawn symbols to be used as a reference for evaluation of symbol recognition systems. It is characterized by its ability to describe unconstrained pen based input (number of strokes, writing order, direction). Also, HBF 49 shows a high performance with a limited size (reasonably low number of 49 features).

In our application context, we use two classifiers. The first allows the recognition of the types of opening (e.g. door, window...). The second is used to recognize furniture (bed, couch...).

3.3 Tree construction

The construction phase is to look for possible hypotheses to interpret a document element. To reduce the interpretation search space, we limit the exploration of the context for the interpretation of a primitive to an area called the local context of the document search.

3.3.1 Defining the local context

The primitive interpretation depends on its neighborhood in structured documents: the structured document analysis requires a two-dimensional context. The analyzer begins by defining a spatial contextual focus that aims to limit the combinatorial exploration due to the hybrid exploration of the analysis tree. This two-dimensional local context is defined for an analysis tree as the maximum distance between the elements of the root and the elements of any

leaves. The choice of the size of the local context depends on the application domain. For example, to interpret an architectural plan, we suggest a local context with a size corresponding to the maximum size of an entity in the document (Figure 7).

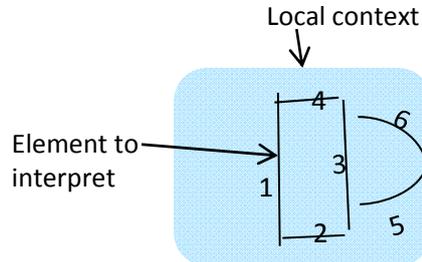


Fig. 7. Local context to interpret the primitive '1'

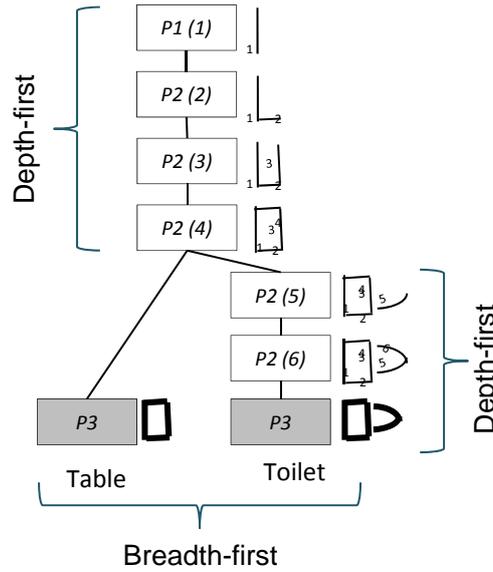
3.3.2 Building the analysis trees

Once the local context is defined, the process builds the analysis trees. Indeed, the analyzer explores all the possible hypotheses of interpretation in the spatial context using a set of two-dimensional rules that describe the structure of the document. Each primitive can be interpreted in several ways which led to the construction of an analysis tree. In the building of the analysis tree, the analyzer explores all the possible hypotheses of interpretation using hybrid exploration in the spatial context with the algorithm described in [18]. Each root is the production rule that would consume this primitive. Each node or leaf is the application of a production rule deduced from the previous node. The number of analysis trees corresponds to the number of possible interpretations for the current primitive. Figure 8 shows a subset of the analysis trees to interpret the primitive 1 illustrated in Figure 7. This figure shows that the primitive 1 can be a part of table or part of toilet. The interpretation of the primitive 1 into a part of furniture is in competition with the interpretation of the same primitive into a wall.

3.4 Decision process

3.4.1 Score calculation

Each branch of the tree (section 3.3.2) is a possible hypothesis. The uncertainty is formalized by the attribution of scores to each hypothesis. Every leaf or node of the tree has a score calculated from both its local score and the score obtained from the preceding nodes. Every score determines the adequacy degree to validate a production. It is calculated from each rule. The production



- **P1 (i)** : Rule « BeginFurniturePartCreation » is applied on primitive i
- **P2 (i)** : Rule « SequenceFurniturePartCreation » is applied on primitive i
- **P3** : Rule « furnitureCreation » is applied

Fig. 8. Hybrid exploration for analysis trees. Competition between two hypotheses: toilet and table

score can also be deduced from a classifier. Unlike other methods, the CD-CMG grammar allows the fusion of classifier and structural (preconditions and constraints) scores.

The equation 1 defines the manner the score is calculated for each production. The use of the square root is a normalization using a geometric average. The adequacy measure of a production is simply defined as a fuzzy combination of the membership degrees of its precondition ($\mu_{preconditions}$) and postcondition ($\mu_{constraints}$).

$$\rho_P = \sqrt{\mu_{preconditions} \cdot \mu_{constraints}} \quad (1)$$

Each branch (hypothesis) is characterized by a score. The equation 2 determines the degree of adequacy (score) of a hypothesis. $|PS|$ is the number of production in the considered branch (referred PS).

$$\rho_{PS} = \left(\prod_{P_i \in PS} \rho_{P_i} \right)^{\frac{1}{|PS|}} \quad (2)$$

A production rule can call an external classifier to recognize the symbols.

3.4.2 Making the decision

Once the tree is well constructed, we start the decision making phase. The goal of the decision process is to validate the right hypothesis among a set of competing hypotheses generated with a descending hybrid analysis. It is a structural decision. The decision process also validates the recognition of symbol shapes. Sometimes the decision process is not sure to make the right decision by validating the best hypothesis (because it has a too low score or it goes into confusion with the other hypotheses). In this case, the analysis process solicits the user and the user validates the right hypothesis.

3.4.2.a Structural decision

The structural decision aims to validate the structural interpretation inside the set of possible hypotheses. The structural decision process researches the ambiguous hypotheses from competing hypotheses constructed in the analysis trees. In practice, an ambiguity is detected if the difference between the branch with the highest score and another branch is below a threshold, called threshold of ambiguity and these branches are contradictory (at least one joined primitive is not used by the same rule production). The equations 3 and 4 describe the adopted algorithm to detect the ambiguous hypotheses.

$$\textit{AmbiguousHypotheses} = \tag{3}$$

$$\{\textit{BestHypothesis}\} \cup \{\textit{AmbiguousAlternativeHypotheses}\}$$

$$\textit{AmbiguousAlternativeHypotheses} = \tag{4}$$

$$\{\textit{hypothesis}_{i \in n}\} / \textit{Score}_{\textit{besthypothesis}} - \textit{Score}_{\textit{hypothesis}_i} \leq \textit{ambiguitythreshold}$$

where n is the number of alternative hypotheses. Two cases may occur:

- If $|\textit{AmbiguousHypotheses}| = 1$: *an implicit validation*. The analyzer is confident enough to choose the right root without asking the user. It implicitly validates the root of the branch that has the highest score.
- If $|\textit{AmbiguousHypotheses}| > 1$: *an explicit validation*. The decision process is not sure to take the right decision, it is a case of ambiguity. Therefore, it solicits the user to make the right decision. The analyzer presents, using a graphical interface, all the ambiguous hypotheses and the user chooses the right hypothesis.

Algorithm 1. Decision algorithm

```
1: procedure MAKING THE DECISION(right hypothesis: list of nodes)
2:   validated-nodes : list of nodes;
3:   validated-nodes.add(root of the right hypothesis);
4:   successor  $\leftarrow$  validated-nodes.lastElement.successor;
5:   while Number of successor == 1 do
6:     validated-nodes.add(validated-nodes.lastElement);
7:     successor  $\leftarrow$  validated-nodes.lastElement.successor;
8:   end while
9: end procedure
10: return validated-nodes
```

The decision is not limited to validate the right root, but can also be used to validate a part of the branch (hypothesis), to accelerate the analysis. In general, if the direct son of a node is unique, the validation of this node automatically means the validation of its direct son (Algorithm 1).

3.4.2.b Shape decision

Once a symbol is structurally defined (opening, furniture), it will be labeled by a classifier. Sometimes the classifier hesitates between two or more labels for the same symbol. The same principles of section 3.4.2.a are applied. The only difference is the type of hypotheses. The structural hypotheses will be replaced by labeling hypotheses. In this case, the analyzer throws a form ambiguity and solicits the user to choose the right label. The interaction between the analysis system and the user requires a study to interact with the user in the best way.

Choice of ambiguity threshold

In IMISketch method, the determination of ambiguity threshold is important and sensitive. Indeed too large threshold generates too many user interactions, but a too low threshold allows to have too many recognition errors. Another advantage of having a significant number of solicitations is to avoid the "out-of-the-loop performance problem" [27]. This problem is the consequence of automation, without the operator having direct control. This situation can have harmful consequences like vigilance decrements or complacency. To avoid this problem, Norman [38] proposes to provide feedback to the operator on the automated task and the possibility to take control in case of failure. Rejection of complex data that are hard to recognized by the classifier hence allow to inform the user of system difficulties and to explicitly ask him to take control and correct the system.

In addition, researcher in experimental psychology have shown that in semi-automated system where human and machine cooperates, it is important to dose this interaction to avoid such phenomena of under-confidence or over-confidence of the user for the system [14].

Today these thresholds are defined in an empirical manner on a small user panel. For experimental validation, we should extend these experiments with many users but this is very expensive.

4 Application on architectural plans

In this section, we describe the implementation of our interactive analysis method IMISketch and illustrate it on 2D handwritten architectural plans.

4.1 Grammatical rule description

The grammar allows to describe the physical and logical structure of documents. The use of grammars CD-CMG is used to describe both a global vision of a symbol by focusing on its position relative to its neighbors and a local vision modeled by constraints.

The developed application aims to interpret architectural plans containing walls, openings and furniture. The interpretation of the architectural plan components takes into account the specificity of description, such as:

- furniture can be connected to walls;
- furniture are inside the architectural plan;
- furniture are an interconnected set of primitives or a set of very close primitives;
- an opening is a set of primitives interconnected or very close, which stands on a support, generally, two collinear walls on one side and the other of the opening;

Our objective is to introduce the grammatical rules for interpreting the primitives (segments and polygons) extracted from the architectural plans as follows:

- a wall is a primitive;
- an opening is a set of primitives;
- a piece of furniture is a set of primitives.

To overcome the difficulties mentioned above, we consider that, at the end of

a wall, we can find another wall, a door or furniture.

We also consider initializing the process that the longest primitive is part of a wall or furniture.

We apply the depth-first exploration to interpret a piece of furniture in an architectural plan. Indeed, if the primitives are very close, we consider that they belong to the same furniture. Breadth-first exploration is used to recognize the walls, the openings and the furniture.

4.2 Dimension of spatial contextual focus

The spatial contextual focus aims at limiting the combinatorial exploration due to the size of the analysis tree. A too small spatial contextual focus may decrease the rate of recognition of the document. A large spatial contextual focus creates a combinatorial explosion. Indeed, the presence of all the primitives of symbol in the spatial contextual focus enables the symbol recognition. The choice of the size of the local context depends on the application domain. In the case of architectural plans, the context size corresponds to the size of the largest opening.

4.3 Cases of ambiguity

In this section, we show two examples of ambiguity that require interacting with the user. The first example describes a structural ambiguity and the second shows a shape recognition ambiguity.

4.3.1 Structural ambiguity

We illustrate an example of structural ambiguity which requires prompting the user. The objective is to interpret all the primitives extracted from an architectural plan.

At the step illustrated in Figure 9, the decision process decides to entrust the decision to the user because the two competing branches are contradictory and the difference between the two scores is below the ambiguity threshold. In fact, there are two ways to interpret the primitives (Figure 9): a window between two walls or three walls and two doors. The figure 9 illustrates the two hypotheses. The user selects the right hypothesis (a window between two walls). This hypothesis will be applied in the document.

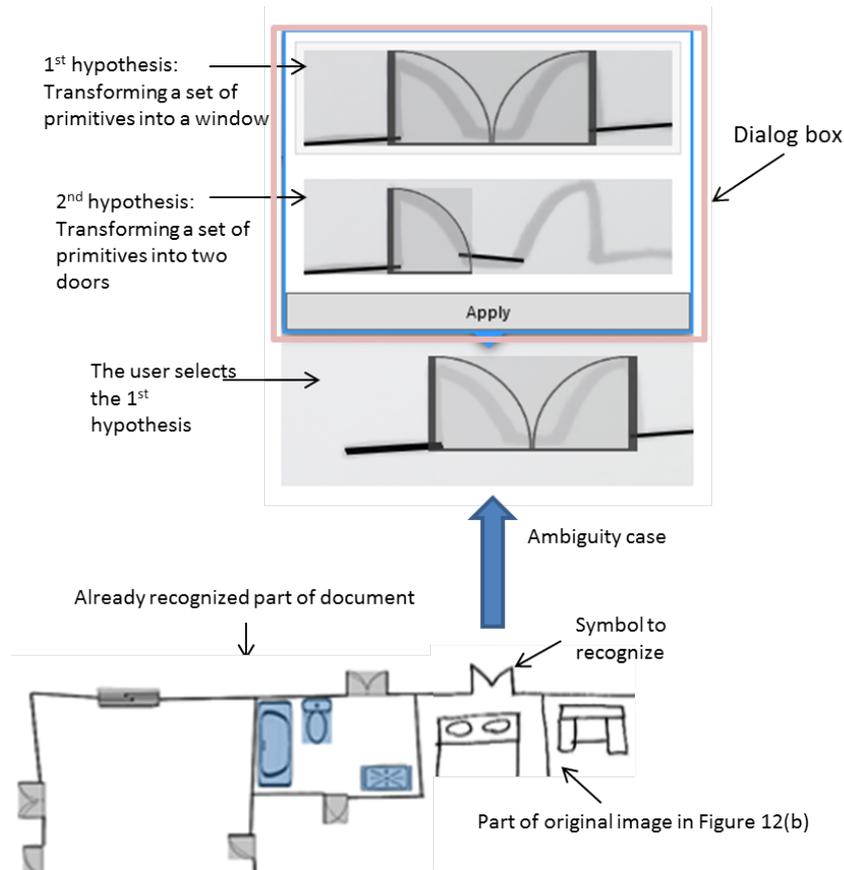


Fig. 9. Structural ambiguity: Two possible hypotheses to interpret the primitives: two doors or window.

The analyzer starts to interpret the first primitive in a wall, then it combines the primitives to form the opening.

4.3.2 Shape recognition ambiguity

In this section, we present the interaction in shape recognition, offered by the integration of a classifier. This classification system used in our method is an evolving system [3,2]. The incremental learning algorithms are used to train evolving classifiers. In incremental learning algorithms, new instances from existing classes can be progressively introduced to the system to improve its performance. Moreover, new unseen classes can be added to the system at any time by the incoming data.

In particular, we show how and when the user can interact with this classifier.

When raising an ambiguity, the user is then in front of four possibilities (see. Figure 10):

- The user validates the hypothesis proposed by the classifier in spite of the

low degree of confidence given by the classifier. The classifier will enhance the model of this class.

- The user associates the symbol to recognize to another existing class in the classifier. The classifier will reduce the confusion between two classes.
- The user associates the symbol to a new class: the user considers that the symbol does not belong to an existing class. With this new information, the classifier will start to learn a new class of symbols.
- The user ignores the symbol to recognize: it is the rejection case. The user considers that the recognized symbol is an outlier (noise in the image). No action is done by the classifier.

With this interaction process, the classifier continuously learns in order to improve its interpretations. The more the analysis goes on, the more the classifier is accurate, the less the user is solicited. This incremental learning is able to deal with the recognition of new classes of symbols. It is a key point to accommodate the great variability of symbols that can occur in a sketch.

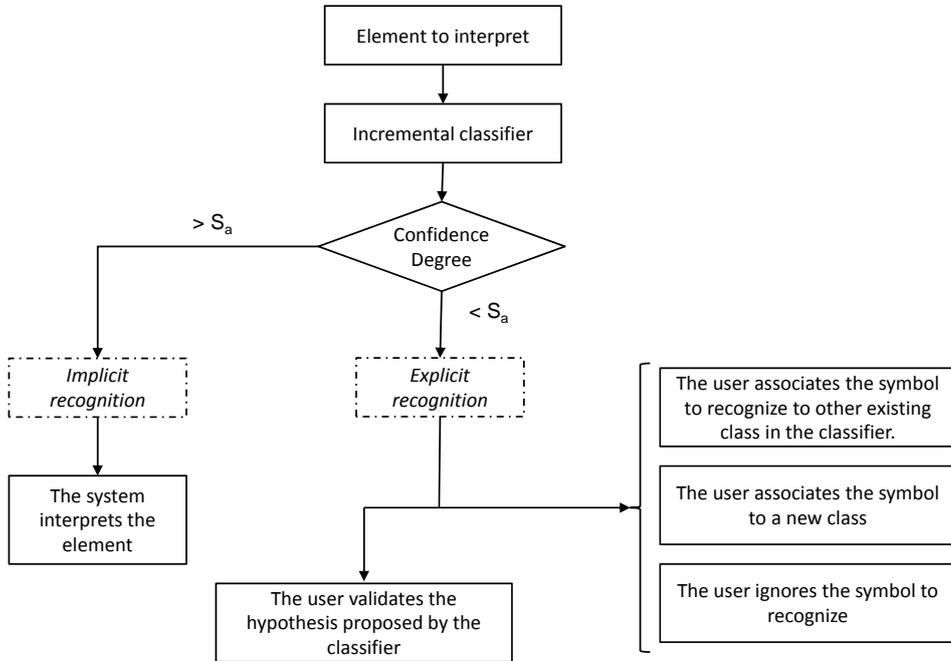


Fig. 10. Interaction scheme of symbol recognition

5 Human Computer Interaction in IMISketch method

To ensure the most appropriate interaction in our method, we adopt use tests with researchers in cognitive psychology and ergonomics of the *Loustic* plat-

form ⁴ with a user centered development method. The aim of these tests is to answer two questions: how interpretation results will be presented to the user, and how the user will interact with analysis process [15].

5.1 Presentation of interpretation on the screen

During this first experiment, we asked participants to compare an original plan with its digital format interpreted and to detect errors. 54 volunteers (19 men and 35 women) participated in this experiment. Each participant successively compared three pairs of plans in one of three experimental conditions: separated, integrated and sequential [15].

5.1.1 Separated condition

The original architectural plan is displayed on the screen and no interaction is possible as long as the analysis phase is in progress. Then, the interpreted plan appears next the original document. Participants can then surround the errors.

5.1.2 Integrated condition

The integrated condition is similar to the separated condition because only the original document appears on the screen for the analysis duration. However, the interpreted document appears above the original document at the end of the analysis process.

5.1.3 Sequential condition

The sequential condition consists in displaying the interpreted document in a progressive manner and above the original document. Therefore the interpretation process is shown in real time to the participants. An example of implementing this interface condition is shown in Figure 11.

Discussion

The experiments using these three conditions show that the participants performed the task of detecting errors in a shorter time with superposed architectural plans (integrated and sequential conditions) relative to separate plans

⁴ Loustic is a platform located in Rennes (France) for multidisciplinary research on user-centered design methods

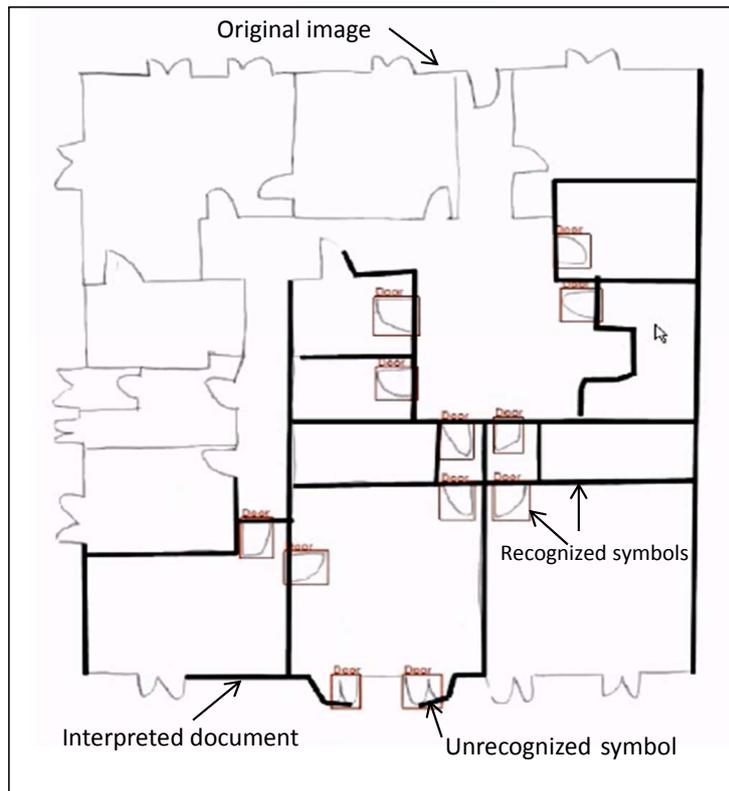


Fig. 11. Sequential condition : the interpreted document appears above the original document in a progressive manner

(separated condition). In the separate condition, the participants encode a visuo-spatial information from the manuscript plan to match it with the interpretation plan. The results suggest that the superposition of plans removes these steps of visual search. In the integrate condition, when the participant looks at a particular point in the interpreted plan, he has sufficient information to identify an error if there is one.

The experiment showed that only 35% of people were able to identify any error in the integrated condition and 47% in the separated condition (the difference is not statistically significant) This gain could be due to an effect of intentional guidance offering to participants. We choose then to use the sequential condition for our IMISketch method, so the analysis will appear over the analyzed plan along its interpretation.

5.2 How the user will interact with analysis process

Two ways are possible to ensure interaction between the system and the user: interruption of interpretation by the user and interruption of interpretation by the system.

A first test [15] was conducted to assess the impact of interruptions of interpretation by users on the interaction. In this test, 36 volunteers (10 males and 26 females) aged 18 to 33 years were asked to surround the misinterpretations of three successive architectural plans that were synthetically interpreted by IMISketch method. The principle of interruption by the system was never mentioned but peripheral aspects were always here. A second user test study [15] was realized with 18 volunteer students (12 females and 6 males) aged 18 to 25 years, concludes that this functionality was thought well of by users, but it requires some improvements.

For example, it is likely that the impact of this solicitation on interaction is strongly related to the number of requests on a real error, and also to the number on errors not specified by this device.

An ongoing study seeks to assess the effects of an interruption made by the system on task performance, its duration and its error detection accuracy. Nevertheless, it seems to be preferable to allow users to intervene on live when they identify an error. The interruption of the system by users increases the interaction efficiency. One may wonder whether an interruption that would be performed by the system itself could also generate a performance improvement.

We believe that in order to ensure the best way human computer interaction, the best strategy is to progressively present the result of the interpretation and keep the two types of interaction: user interventions during the analysis process and the user solicitation by the analysis process if needed.

6 Experimental results

Our method is tested and validated on architectural floor plans sketches database containing walls, openings and furniture. To our knowledge, the existing methods are not tested on the same type of floor plan. Consequently, we are not able to propose a convenient performance comparison with other methods. However the results will demonstrate the interesting properties of our approach. For this, and for comparison with future interactive methods, we propose to publish our database and make it accessible in www.irisa.fr/intuidoc/ArchPlanDB.html.

In this section we report different results obtained with the complete interactive recognition system. These experiments focus on the contribution of interactivity in a lazy (a posteriori) recognition method.

6.1 Architectural floor plan database

To evaluate our method, we created an architectural plan database. Each architectural plan consists of dozen furniture types (toilet, table, bed...), 3 types for openings (door, window and sliding window) and walls (more details in Table 1). Figure 12 shows some examples of architectural plans.

Number of architectural plans	24
Number of walls	961
Number of openings	414
Number of furniture	523

Table 1
Architectural plan database

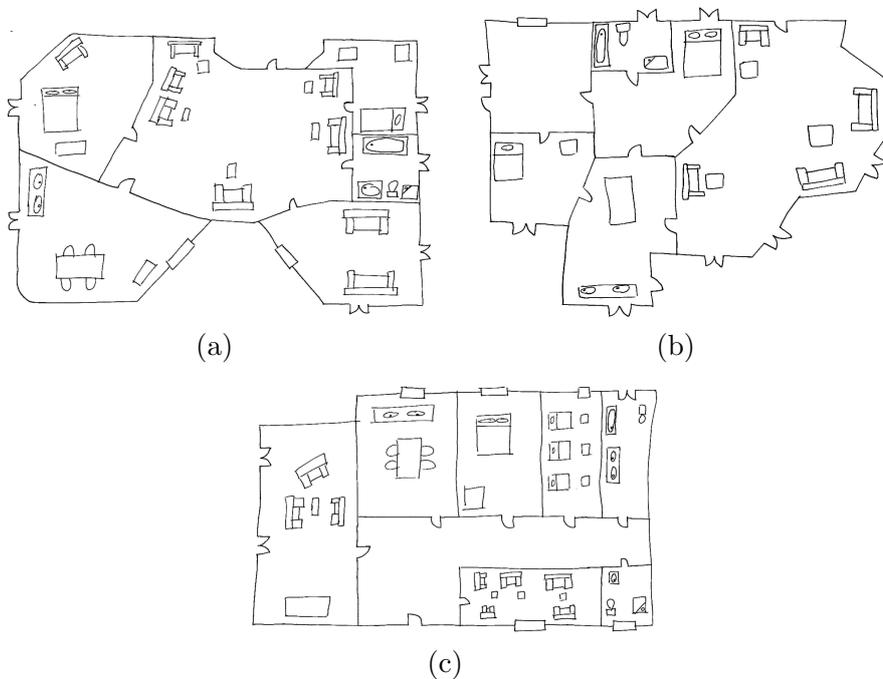


Fig. 12. Example of architectural plans

6.2 Overall Evaluation of IMISketch method

In this section, we present an overall evaluation of the method. In this experiment, we first determine the size of the spatial context focus. This size corresponds to the size of the largest symbol in the document. We also fix the threshold of ambiguity that ensures the best compromise recognition/solicitation [18]. We test our interactive method on architectural plans (Figure 12).

We divide the dataset into two subsets:

- Initial learning subset: used to train the classifier in full-supervised manner, i.e. the label of each sample is given by the user. This subset contains 9 architectural plans.
- Evaluation subset: used to evaluate IMISketch method. This subset consists of 15 architectural plans with an average of 84 symbols (walls, openings and furniture) by architectural plan. The primitive extraction step gives an average of 302 primitives (segments and polygons) per plan.

The total analysis of 15 architectural plans shows that the total recognition rate reached 93.4% at symbol level. A symbol is considered well recognized when its bounding box is correct and its class label is correct. The obtained errors are related to the classifier labeling, i.e the symbol is structurally recognized but poorly recognized by the classifier (mislabeled).

The user intervenes an average of 5 times per architectural plan to solve *structural ambiguity*, and in 25% of cases he does not validate the best hypothesis found by the system (the hypothesis with the best score). To solve the *symbol ambiguity*, the classifier also solicits the user when it is not sure to give the right label symbol to symbol. The test, on 15 architectural plans, gives an average of 5 solicitations per plan, 50% are useful interventions, i.e. the user does not choose the label proposed by the classifier. Table 2 illustrates the results.

Number of architectural plans	15
Number of symbols	961
Recognition rate	93.4%
Average structural solicitations per plan	5
Percentage useful structural solicitations	25%
Average classifier solicitations per plan	5
Percentage useful classifier solicitations	49%

Table 2
Recognition rate for architectural plan

An example of a plan to interpret is illustrated in Figure 12(a). Figure 13 illustrates the final result of the interpretation.

In Figure 13(a), we note two symbols that are not recognized for reasons of lack of necessary primitives (problem of extracting primitive). We also have a mislabeled symbol (problem of classifier).

Figure 13(b) is well recognized. It identifies walls, openings and furniture with the right labels. The system solicits the user 3 times to resolve the ambiguity.

These ambiguities are mainly due to the presence of several primitive too close, but belonging to different symbols (Table 3).

In Figure 13(c), the document is well recognized except one symbol that is mislabeled. IMISketch system solicits one time the user (Table 3).

The ambiguity cases of Figures 12(b) and 12(c) are detailed in the Table 3. In the first two cases, the IMISketch system presents to the user two possible hypotheses: either merge all primitives in the same symbols (a couch, wrong hypothesis) or assign them in different symbols (right hypothesis). The ambiguity in the other two cases is due to the presence of several collinear primitives.

Note that the symbol rotation is not yet integrated in the current version of IMISketch.

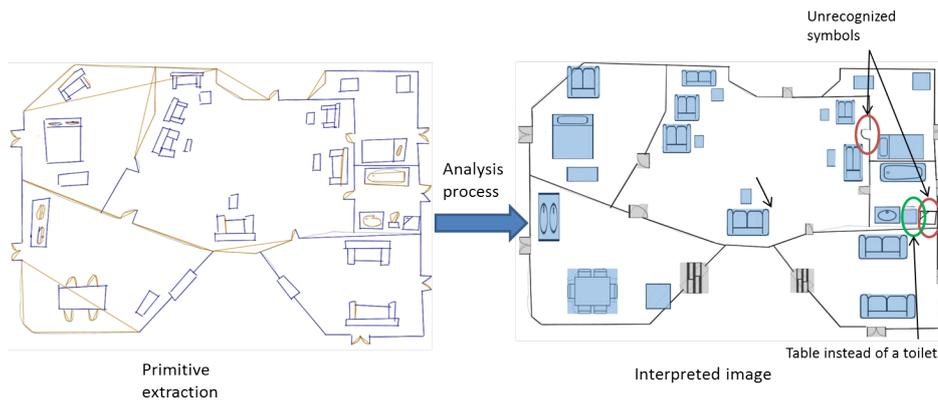
Primitives	 Figure 12(b)	 Figure 12(b)	 Figure 12(b)	 Figure 12(c)
Possible hypotheses	<ul style="list-style-type: none"> • A couch • A door and two pieces of furniture (table and bed) 	<ul style="list-style-type: none"> • A couch • A couch and table 	<ul style="list-style-type: none"> • Several collinear primitive can lead several hypotheses of opening 	<ul style="list-style-type: none"> • Several collinear primitive can lead several hypotheses of opening

Table 3
Example of ambiguity cases

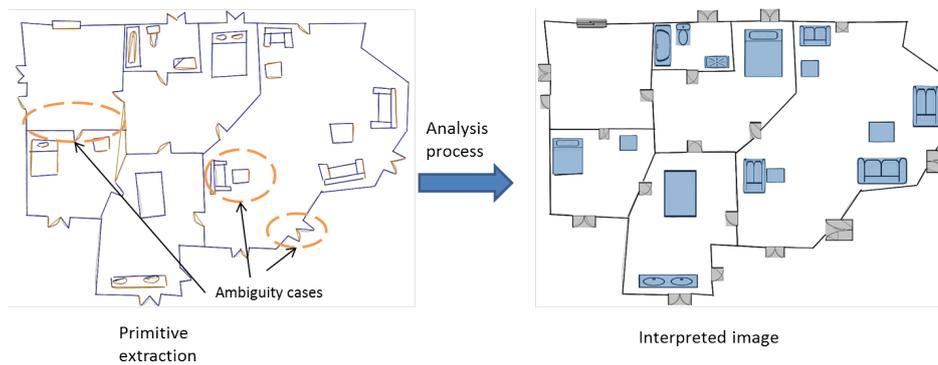
The experimental results are very encouraging. They suggest that it is possible to introduce a breadth-first exploration avoiding the combinatorial problem. This reinforces the interest of designing an interactive system for the recognition of documents. Soliciting the user guarantees to obtain very high recognition rates even in the case of complex documents. The use of the polygonal primitive does not have a negative impact on the structural recognition rate, further, it reduces the number of user intervention during the analysis and also speeds up the calculation.

Moreover to illustrate the usability of the presented system, we present two videos which show the general concept of IMISketch: <http://youtu.be/HIV6dQHgbuw> and http://youtu.be/7divT_r7El0.

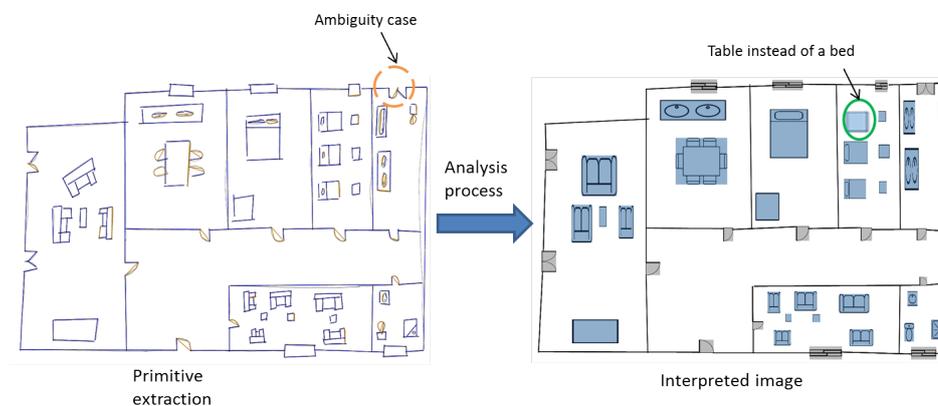
IMISketch was validated on offline handwritten architectural floor plan. Several extension of our approach are now possible. The first extension will be to adapt IMISketch for interactive online document recognition (using tablet). For this, we simply need to adjust the primitive extractor to online document. We consider that the polygon extraction of online document is simpler than offline document. The extension of IMISketch to online document interpretation therefore should not be too complicated. Another extension is



(a) Interpretation of image in Figure 12(a)



(b) Interpretation of image in Figure 12(b)



(c) Interpretation of image in Figure 12(c)

Fig. 13. Example of architectural plans

to use IMISketch approach to off-line recognition of printed documents. The complexity of this extension will be related to the complexity of considered printed plans. It could be direct for similar handwritten plans already validated but much more complex when the architectural plan is composed of many layers of information.

7 Conclusion

In this paper, we have presented our IMISketch method. IMISketch method is generic and interactive. The analyzer is based on a competitive hybrid exploration of the analysis tree according to a dynamical local context of the document. The choice between breadth-first and depth-first exploration is described in the production rules. This addition enhances the two-dimensional grammars CD-CMG originally designed for the on-fly online analysis.

The decision process is able to solicit the user in the case of strong ambiguity.

We validated the criteria of acceptability and usability of the system by doing usage tests. The aim of these tests is to determine the best way to present the results to the user interpretation and the best manner to interact with the user. We have shown that displaying the interpretation result of documents in a progressive manner is most appreciated by the participants.

The tests of this interactive analyzer have been made on 2D handwritten architectural floor plans. Integrating the user in the analysis process is, in our view, a key point to address complex off-line sketch recognition and to avoid an a posteriori verification phase.

Future work will focus on extending the experimental results on large image databases containing printed and vectored document architectural plans and other types of documents such as the circuit diagram.

Acknowledgement

The authors would like to thank all the people who took part in the experiments. This work partially benefits from the financial support of the ANR Project Mobisketch (<http://mobisketch.irisa.fr/>).

References

- [1] Ablameyko, S.: An introduction to interpretation of graphic images. SPIE International Society for Optical Engineering (1997)
- [2] Almaksour, A., Anquetil, E.: Improving premise structure in evolving takagi-sugeno neuro-fuzzy classifiers. *Evolving Systems* 2(1), 25-33 (2011)
- [3] Almaksour, A., Anquetil, E.: Iclass: Error-driven antecedent learning for evolving takagi-sugeno classification systems. *Applied Soft Computing* (0), (2013).

- [4] Alvarado, C., Davis, R.: Dynamically constructed bayes nets for multi-domain sketch understanding. In: ACM SIGGRAPH 2006 Courses, p. 32. ACM (2006)
- [5] Alvarado, C., Davis, R.: Sketchread: a multi-domain sketch recognition engine. In: ACM SIGGRAPH 2007 courses, pp. 34-es. ACM (2007)
- [6] Aoki, Y., Shio, A., Arai, H., Odaka, K.: A prototype system for interpreting hand-sketched floor plans. In: Pattern Recognition, 1996., Proceedings of the 13th International Conference on, vol. 3, pp. 747-751. IEEE (1996)
- [7] Brennan, S., Hulteen, E.: Interaction and feedback in a spoken language system: A theoretical framework. Knowledge-Based Systems 8(2), 143-151 (1995)
- [8] Chan, K., Yeung, D.: An efficient syntactic approach to structural analysis of on-line handwritten mathematical expressions. Pattern Recognition 33(3),375-384 (2000)
- [9] Chen, L., Yin, P.: A system for on-line recognition of handwritten mathematical expressions. Computer Processing of Chinese and Oriental Languages 6(1), 19- 39 (1992)
- [10] Coüasnon, B.: Dmos, a generic document recognition method: Application to table structure analysis in a general and in a specific way. IJDAR 2006 8(2), 111-122 (2006)
- [11] Delaye, A., Anquetil, E.: Hbf49 feature set: A first unified baseline for online symbol recognition. Pattern Recognition 46(1), 117-130 (2013)
- [12] Feng, G., Viard-Gaudin, C., Sun, Z.: On-line hand-drawn electric circuit diagram recognition using 2d dynamic programming. Pattern Recognition 42(12), 3215-3223 (2009)
- [13] Fitzgerald, J., Geiselbrechtner, F., Kechadi, T.: Mathpad: A fuzzy logic-based recognition system for handwritten mathematics. In: ICDAR 2007, vol. 2, pp. 694-698 (2007). DOI 10.1109/ICDAR.2007.4377004
- [14] Fleury, S.: Le rôle de l'utilisateur dans les systèmes de traitements automatiques. Ph.D. thesis, Université européenne de Bretagne (2014)
- [15] Fleury, S., Ghorbel, A., Lemaitre, A., Anquetil, E., Jamet, E.: User-centered design of an interactive off-line handwritten architectural floor plan recognition. In: ICDAR, pp. 1073-1077 (2013)
- [16] Freeman, I.J., Plimmer, B.: Connector semantics for sketched diagram recognition. In: Proceedings of the eight Australasian conference on User interface - Volume 64, AUIC '07, pp. 71-78. Australian Computer Society, Inc., Darlinghurst, Australia, Australia (2007).
- [17] Ghorbel, A., Almaksour, A., Lemaitre, A., Anquetil, E.: Incremental learning for interactive for sketch reconition. Ninth IAPR International Workshop on Graphics RECOgnition - GREC 2011 (2011)

- [18] Ghorbel, A., Lemaitre, A., Anquetil, E.: Competitive hybrid exploration for off-line sketches structure recognition. In: *Frontiers in Handwriting Recognition (ICFHR)*, 2012 International Conference on, pp. 571-576 (2012)
- [19] Ghorbel, A., Macé, S., Lemaitre, A., Anquetil, E.: Interactive competitive breadth-first exploration for sketch interpretation. *ICDAR 2011* pp. 1195-1199 (2011)
- [20] Gross, M.: The electronic cocktail napkin computational environment for working with design diagrams. *Design Studies* 17(1), 53-69 (1996)
- [21] Hammond, T., Davis, R.: Ladder: A language to describe drawing, display, and editing in sketch recognition (2003)
- [22] Hammond, T., Davis, R.: Ladder, a sketching language for user interface developers. *Computers & Graphics* 29(4), 518-532 (2005)
- [23] Hammond, T., O’Sullivan, B.: Recognizing free-form hand-sketched constraint network diagrams by combining geometry and context. *Proceedings of the Eurographics Ireland 2007* (2007)
- [24] Hammond, T., Paulson, B.: Recognizing sketched multistroke primitives. *ACM Trans. Interact. Intell. Syst.* 1(1), 4:1-4:34 (2011).
- [25] Huang, G., Zhang, W., Wenyan, L.: A discriminative representation for symbolic image similarity evaluation. *Graphics Recognition. Recent Advances and New Opportunities* pp. 71-79 (2008)
- [26] Hutton, G., Cripps, M., Elliman, D., Higgins, C.: A strategy for on-line interpretation of sketched engineering drawings. In: *Document Analysis and Recognition, 1997., Proceedings of the Fourth International Conference on*, vol. 2, pp. 771-775. IEEE (1997)
- [27] Kaber, D.B., Endsley, M.R.: Out-of-the-loop performance problems and the use of intermediate levels of automation for improved control system functioning and safety. *Process Safety Progress* 16(3), 126-131 (1997).
- [28] Lank, E., Thorley, J., Chen, S., Blostein, D.: On-line recognition of UML diagrams. In: *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*, pp. 356-360. IEEE (2001)
- [29] Lemaitre, A., Camillerapp, J.: Text line extraction in handwritten document with kalman filter applied on low resolution image. *Document Image Analysis for Libraries, International Workshop on* 0, 38-45 (2006).
- [30] Lemaitre, M., Grosicki, E., Geoffrois, E., Preteux, F.: Preliminary experiments in layout analysis of handwritten letters based on textural and spatial information and a 2d markovian approach. In: *ICDAR 2007*, vol. 2, pp. 1023-1027 (2007). DOI 10.1109/ICDAR.2007.4377070
- [31] Lladós, J., López-Krahe, J., Martí, E.: A system to understand hand-drawn floor plans using subgraph isomorphism and hough transform. *Machine Vision and Applications* 10(3), 150-158 (1997)

- [32] Macé, S., Anquetil, E.: A generic method for eager interpretation of on-line handwritten structured documents. In: Pattern Recognition, 2006. ICPR 2006. 18th International Conference on, vol. 2, pp. 1106-1109. IEEE (2006)
- [33] Macé, S., Anquetil, E.: Eager interpretation of on-line hand-drawn structured documents: The dali methodology. Pattern Recognition 42(12), 3202-3214 (2009)
- [34] Mao, S., Rosenfeld, A., Kanungo, T.: Document structure analysis algorithms: a literature survey. In: Proc. SPIE Electronic Imaging, vol. 5010, pp. 197-207 (2003)
- [35] Mas, J., Lladós, J., Sánchez, G., Jorge, J.: A syntactic approach based on distortion-tolerant adjacency grammars and a spatial-directed parser to interpret sketched diagrams. Pattern Recognition 43(12), 4148-4164 (2010)
- [36] Messmer, B., Bunke, H.: Automatic learning and recognition of graphical symbols in engineering drawings. Graphics Recognition Methods and Applications pp. 123-134 (1996)
- [37] Montreuil, F., Grosicki, E., Heutte, L., Nicolas, S.: Unconstrained handwritten document layout extraction using 2d conditional random fields. ICDAR 2009 0, 853-857 (2009).
- [38] Norman, D.A.: The problem with automation: inappropriate feedback and interaction, not 'over-automation'. Philosophical Transactions of the Royal Society of London. B, Biological Sciences 327(1241), 585-593 (1990)
- [39] Notowidigdo, M., Miller, R.: Off-line sketch interpretation. In: AAAI Fall Symposium, pp. 120-126 (2004)
- [40] Sheraz Ahmed, a.M.L., Weber, M., Dengel, A.: Improved automatic analysis of architectural floor plans. ICDAR 2011 pp. 864-868 (2011)
- [41] Shio, A., Aoki, Y.: Sketch plan: A prototype system for interpreting handsketched floor plans. Systems and Computers in Japan 31(6), 10-18 (2000)
- [42] Zhang, W., Liu, W.: A new vectorial signature for quick symbol indexing, filtering and recognition. In: Proceedings of the Ninth International Conference on Document Analysis and Recognition - Volume 01, ICDAR '07, pp. 536-540. IEEE Computer Society, Washington, DC, USA (2007)

Author Biography

I am Achraf Ghorbel. I am currently holding an assistant professor position at the university of Rennes (France). I am doing my researches at [IRISA](#) laboratory, as member of [Intuidoc](#) team. I took my thesis in 2012.