



**HAL**  
open science

## **IMISketch: an interactive method for sketch recognition**

Achraf Ghorbel, Eric Anquetil, Jean Camillerapp, Aurélie Lemaitre

### ► **To cite this version:**

Achraf Ghorbel, Eric Anquetil, Jean Camillerapp, Aurélie Lemaitre. IMISketch: an interactive method for sketch recognition. *Pattern Recognition Letters*, 2014, 35 (78-90), pp.13. hal-00933697

**HAL Id: hal-00933697**

**<https://hal.science/hal-00933697>**

Submitted on 20 Jan 2014

**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.

# IMISketch: an interactive method for sketch recognition

Achraf Ghorbel<sup>a</sup>, Eric Anquetil<sup>a</sup>, Jean Camillerapp<sup>a</sup>, Aurélie Lemaitre<sup>b</sup>

<sup>a</sup>*INSA de Rennes, UMR IRISA, Campus de Beaulieu, F-35042 Rennes  
Université Européenne de Bretagne, France*

<sup>b</sup>*UMR IRISA, Campus de Beaulieu, F-35042 Rennes  
Université de Rennes 2, France*

---

## Abstract

In this paper, we present a new generic method for an interactive interpretation of sketches to avoid a fastidious verification phase. After a preprocessing phase in which we extract a set of primitives, the interpretation process consists of an interactive analysis. The analyzer is based on a competitive breadth-first exploration of the analysis tree. As opposed to well known structural approaches, this method allows to evaluate simultaneously several possible hypotheses of recognition in a dynamic local context of document. 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. While, in practice these approaches often induce a large combinatory, this paper presents optimization strategies to reduce the combinatory. The goal of these optimizations is to have time analysis compatible with user expectations. These strategies have been integrated into both preprocessing and analysis phases. To validate this interactive analysis method, several experiments are reported in this paper on off-line handwritten 2D architectural floor plans.

---

*Email addresses:* [achraf.ghorbel@irisa.fr](mailto:achraf.ghorbel@irisa.fr) (Achraf Ghorbel),  
[eric.anquetil@irisa.fr](mailto:eric.anquetil@irisa.fr) (Eric Anquetil), [jean.camillerapp@irisa.fr](mailto:jean.camillerapp@irisa.fr) (Jean Camillerapp), [aurelie.lemaitre@irisa.fr](mailto:aurelie.lemaitre@irisa.fr) (Aurélie Lemaitre)  
*Preprint submitted to Pattern Recognition, September 2, 2013*

25 *Keywords:* sketch recognition, breadth-first exploration, interactive  
26 recognition, 2D architectural floor plans, Kalman filter

---

## 27 **1. Introduction**

28 Nowadays, digital documents are becoming more and more omnipresent in  
29 our life. Many reasons, such as the flexibility provided by digital processing,  
30 have led to transform handwritten documents to digital ones. In this context,  
31 people are working on mapping technical paper documents, like architectural  
32 floor plans, to digital ones. We aim at offering a complete and homogeneous  
33 solution to unify paper document recognition and pen-based composition (for  
34 instance: with Tablet PC). We present the IMISktech system : Interactive  
35 Method for Interpretation of Sketches. The input of this system is a scanned  
36 image of handwritten architectural plan and after interpretation the output  
37 is its digital version. This method is the result of four years of research  
38 that leads to several scientific publication [14, 16, 17, 15]. This paper is a  
39 synthesis of this method. We focus on the extraction of primitives that feed  
40 the analyzer to optimize the management of combinatory.

41 We have identified two major approaches for document analysis: syntactic  
42 and statistical approaches. Choosing one of these two approaches depends  
43 on the document type.

44 The syntactic approaches [6, 8, 29, 12, 21, 22] lean on prior knowledge  
45 of the document structure to drive the analysis. They are often based on  
46 visual languages for describing this knowledge and generating the analyzer.  
47 However, syntactic methods have difficulties to incorporate the uncertainty.

48 The statistical approaches [25, 30, 32] provide a better ability to incor-

49 porate uncertainty and usually lack the ability to convey the hierarchical  
50 structure of the document. The use of statistical approaches needs a wide  
51 learning on a homogeneous and labeled base. Each type of approach has  
52 advantages and drawbacks. The interpretation of handwritten structured  
53 documents needs on the one hand an approach that retains its structure, ie  
54 a syntactic approach, and on the other hand an approach that provides a  
55 better ability to incorporate uncertainty, ie a statistical approach.

56 In this work, we design a complete system for the sketch interpretation:  
57 IMISketch<sup>1</sup>. One of the main originalities of IMISketch is to avoid an *a pos-*  
58 *teriori* verification phase by soliciting the user. After a preprocessing phase  
59 in which primitives of the structured document are extracted, the system  
60 is characterized by an interactive analysis phase. The analyzer (referred as  
61 IMISketch) uses a new syntactic approach based on an interactive and lazy  
62 interpretation of the document. Unlike the classical syntactic approaches,  
63 IMISketch does not always select the first or the best found hypothesis. The  
64 associated analysis process is able to take into account the uncertainty.

65 Thanks to the interactivity, the user can be solicited, if needed, by the  
66 analyzer to raise ambiguities of recognition [17] i.e. to choose between two  
67 or more possible hypotheses or to enrich the *a priori* knowledge of the sys-  
68 tem [14]. In fact, the user participation has a great impact to avoid error  
69 accumulation during the analysis step. To detect ambiguities, we should  
70 adopt a method based on breadth-first exploration. Like all the analysis  
71 methods based in breadth-first exploration, this approach can induce a large

---

<sup>1</sup>Interactive Method for Interpretation of Sketch

72 combinatorics. This combinatorics mainly depends on the quality of prim-  
73 itives extracted from the image and the manner to analyze them. In this  
74 paper we propose some optimizations to reduce it, by addressing these two  
75 points. These optimizations are introduced from the phase of segmentation  
76 to the analysis. They will lead to the new system IMISketch+.

77 The complete system can be applied to off-line documents (image), as  
78 illustrated in this paper (Figure 9(a)), as well as on-line or vectored docu-  
79 ments.

80 In the state of the art, one interesting generic approach is the LAD-  
81 DER [19] [18] system which has been proposed by Hammond and Davis  
82 for interpreting *a posteriori* or on the fly on-line handwritten documents.  
83 LADDER language has been exploited for the design of various systems of  
84 interpretation of structured documents, such as UML [18], electrical dia-  
85 grams [2] or complex graphs [20]. Also Plimmer proposed InkKit that is a  
86 framework and a toolkit to recognize complex components [31] [13]. In addi-  
87 tion VR Sketchpad [27] is a pen-based computing environment for inputting  
88 and locating 3D objects in a virtual world.

89 Unlike these methods, our method interprets off-line handwritten struc-  
90 tured documents. It has been tested on 2D architectural floor plans. The  
91 specific task of floor plan analysis has been addressed for more than twenty  
92 years. Lladós [26] proposed a method for understanding hand drawn floor  
93 plans using subgraph isomorphism and Hough transform. Aoki [3] proposed  
94 also a method for interpreting a hand-sketched floor plan. This method fo-  
95 cuses on understanding the hand sketched floor plan and converting it into  
96 a CAD representation. Also, Ahmed [33] proposed an analysis method spec-

97 ified in printed architectural floor plans.

98       Contrary to these methods that can require a fastidious *a posteriori* veri-  
99 fication phase, IMISketch system attempts to avoid this phase by integrating  
100 the user during the analysis process.

101       The recognition of a structured document using a structural approach  
102 needs an a priori description. The modeling of structured documents differs  
103 from one type to another. Several techniques allow the document descrip-  
104 tion. Yamamoto [35] and Bunke [5] use classical one-dimensional grammars.  
105 Other techniques are used to model two-dimensional documents. Fahmy [11]  
106 and Bunke [4] offer grammar graphs. These grammars have been widely used  
107 in the various communities for interpreting off-line documents such as math-  
108 ematical formulas. Despite graph grammars offer a very expressive mecha-  
109 nism for pattern recognition, these grammars have their limitations. They  
110 are expensive to implement and difficult to handle by the developer, espe-  
111 cially when the productions become numerous. These graphs are also poorly  
112 adapted to deal with uncertainty.

113       Our goal is to analyze documents of different kinds such as handwritten  
114 documents. To overcome this problem, we adopt context-driven constraint  
115 multi-set grammars (CD-CMG), designed for *on-line* recognition [28] asso-  
116 ciated with a scoring approach based on the fuzzy logic theory. The main  
117 contribution of our work is to modify and to extend this formalism to design  
118 an interactive analyser for *off-line* recognition. This strategy allows to solicit  
119 the user in the case of strong ambiguity and avoids the fastidious a posteriori  
120 verification task to find and correct the remaining interpretation errors.

121       The remaining of the paper is organized as follows. In the section 3(a),

122 we introduce the architecture and the basic principles of IMISketch method.  
123 The phase of primitive extraction is described in section 3. Section 4 presents  
124 the concepts that are linked with an interactive breadth-first analysis. In sec-  
125 tion 5 implementation and optimization of IMISketch analyzer are presented.  
126 Experimental results on interpretation of images of 2D handwritten architec-  
127 tural floor plans are reported in section 6 and finally, section 7 concludes the  
128 paper.

## 129 **2. Interactive analysis stages**

130 In this section, we summarize the different steps of treatment to ensure the  
131 recognition of a handwritten structured document (cf. Figure 1). The first  
132 step is the segmentation process. This step is purely off-line (i.e. without user  
133 interaction). The aim of this phase is to extract all the basic primitives that  
134 will be used to analyze the document. In the context of sketch recognition,  
135 the segmentation process consists in extracting handwritten strokes as a set  
136 of segments. This part is detailed in section 3. The second main step is to  
137 analyze these primitives and to compose them according to their structural  
138 arrangement in the document to identify the symbols. Our analyzer is made  
139 of two associated key processes: the recognition of the document structure  
140 is managed by **the grammatical analyzer** that drives the calls of **symbol**  
141 **classifiers** to evaluate a fuzzy scoring for each hypotheses. For instance, in  
142 architectural plans, structural recognition detects walls, opening, etc., and  
143 the fuzzy classifiers identify the opening to window, door, etc.

144 The user can be solicited during the analysis process in case of ambiguity  
145 detection. This approach is generic and needs some *a priori* knowledge at

146 the structural level by defining a grammar that will describe the document  
 147 structure and at the classifier level to identify the symbols. This knowledge  
 148 corresponds to the specific part of the analysis system.

149 We detail in the following sections the preprocessing (section 3) and the  
 150 analysis phases (section 4).

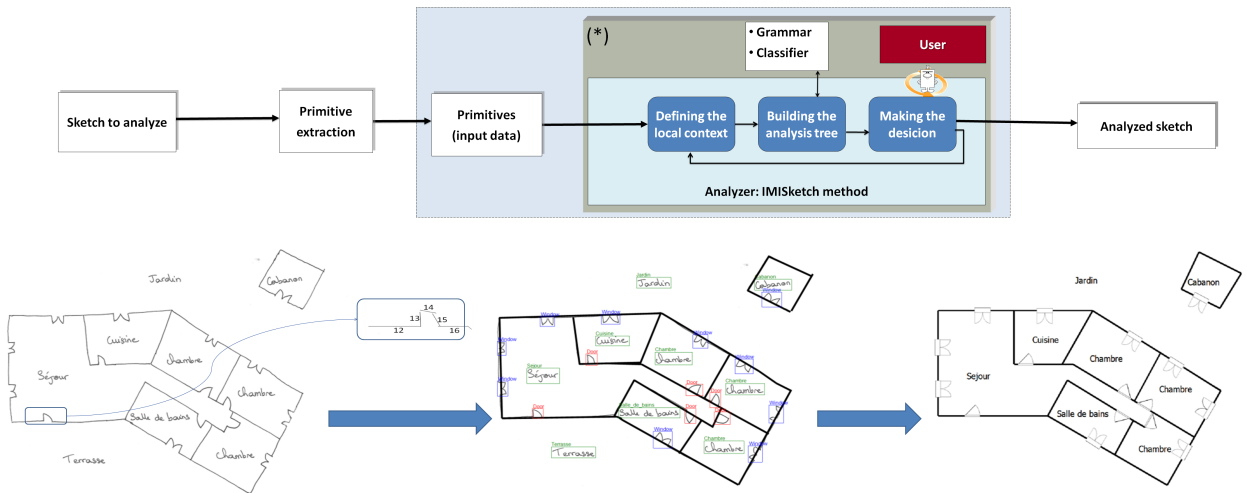


Figure 1: Synthetic scheme for structured documents interpretation

### 151 3. Preprocessing phase: primitive extraction

#### 152 3.1. Related works

153 The recognition of architectural plans has already been studied in par-  
 154 ticular by Dosch et al [10]. In this kind of plans, segments, representing the  
 155 walls, are primitives widely used. In these works, most of the analyzed plans  
 156 have been drawn with a ruler (or CAD software), consequently, segments are  
 157 really straight. In [23], Hilaire proposed a method that improves the method



158 of Dosch. The originality of his work comes from the segmentation process  
159 of the skeleton. This aspect is very important in the context in which draw-  
160 ings are crossing. His paper contains an important bibliography. Hilaire, as  
161 Dosch, works on printed documents, but our aim is to process handwritten  
162 documents. In such documents, walls are not straight lines and the repre-  
163 sentation of doors and windows use arcs of a circle which are sometimes very  
164 roughly drawn.

165 In [7], Chang proposes to carry out the vectorization of handwritten doc-  
166 uments by Bezier curves. This approach has been designed for the vector-  
167 ization of cartoons, but it does not lead to an easy representation to be used  
168 for a later interpretation of symbols like pieces, doors, windows, etc.

169 In [9] de Brucq uses a Kalman filter to decompose handwritten strokes  
170 into straight lines and arcs of circle. These primitives are well adapted for  
171 architectural plan interpretations, but the system works with an on-line rep-  
172 resentation of the drawing.

### 173 *3.2. Our method*

174 To extract lines in off-line sketch documents, we propose to adapt the  
175 method we presented in [24]. It uses a Kalman filter.

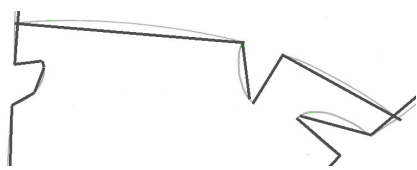
176 A Kalman filter is a prediction verification process that provides an es-  
177 timate of a model from observations. In the case of the drawing extraction,  
178 our model is based on three values: the thickness, the position and the local  
179 slope of the line. When the drawing is relatively regular, the Kalman filter  
180 can use these three values to predict the next position and update the model  
181 from the pixels of the image. In areas that include intersections, the observed  
182 thickness is not consistent with the predicted thickness, but it is possible to

183 use the prediction ability of the filter to pass through these regions. In a  
184 region where the direction of the line changes rapidly, it is not possible to  
185 find black pixels in the predicted position and the follow-up of the drawing  
186 stops. Equations and more details can be found in [24].

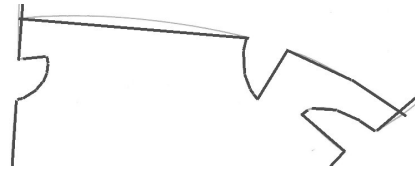
187 For sketch documents the most important parameter for the Kalman filter  
188 is the one that defines the variability of the slope. With a large value (0.1, i.e.  
189 a 10% variation of the slope) a curved section of drawing is detected as a single  
190 segment joining its two ends (Figure 2(a)). With a very low value (0.005)  
191 the model allows a very little variation of the slope, and thus many small  
192 segments are detected in the same curved section (Figure 2(b)). We note, in  
193 this example, that the symbols representing opening (door or window) are  
194 more precisely represented in Figure 2(b) (precise decomposition) than in  
195 Figure 2(a) (rough decomposition). A precise decomposition increases the  
196 number of segments extracted from the image, however this decomposition  
197 can be very useful when using a classifier to recognize symbols.

198 To fix ideas, the full plan from which is extracted the Figure 2 is repre-  
199 sented by 144 segments for the rough decomposition and 177 segments for the  
200 precise one. This increase of the number of segments is mostly due to small  
201 segments in the opening regions. We must notice that a little region with  
202 many small segments will produce a combinatorics explosion of the analyze  
203 time. This effect is pointed out later in the section 6 and Figure 9(c).

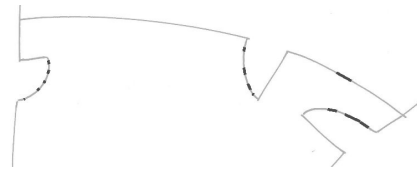
204 To overcome this dilemma, i.e. a precise representation but a low com-  
205 binatorics, we propose to extract only a precise decomposition and to build  
206 spatial relationships between the segments. Thus, it is important to re-  
207 member if a segment is a curvilinear extension of another one. During the



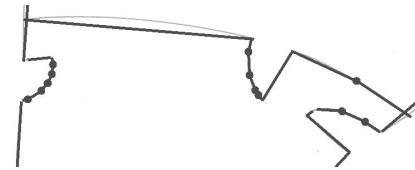
(a) Rough decomposition of sketch



(b) Fine decomposition of sketch



(c) Common part of two segment is shown dark



(d) Representation with segments and lines (circle denotes connection between consecutive segments)

Figure 2: Extraction of primitives: Original drawing in light gray, primitives shown in dark black

208 detection of segments by the Kalman filter, it is possible to detect that a  
209 part of the drawing belongs to two different segments (Figure 2(c)). In this  
210 case, we remove this common part from one of the two segments and mem-  
211 orize that there is a curvilinear link between the extremities of these two  
212 segments. The small circles in Figure 2(d) represent a connection between  
213 two segments. Each set of segments attached to one another corresponds to  
214 a polygonal approximation of the curved parts of the drawing (for instance  
215 an opening).

216 Thus, there are two types of primitives extracted from an image: sim-  
217 ple segments and polygons. We transmit the connection information so that  
218 the interpretation process can directly use simple segments and polygons as  
219 primitive, or can decide to split a polygon into segments according to some  
220 structural information like the size of the polygon or criterion of collinearity.  
221 The use of this dual representation on one hand allows a more precise char-  
222 acterization of curve sections and on the other hand limits the combinatorics  
223 during the interpretation process.

#### 224 **4. Interactive breadth-first exploration**

225 In this section, we present the analyzer by first describing its main char-  
226 acteristics. Then, we detail the different steps of the internal analysis. The  
227 description of this method is followed by a concrete example in the next  
228 section (section 5).

##### 229 *4.1. Analyzer characteristics*

230 The interpretation strategy of structured documents is driven by the *a*  
231 *priori* knowledge we have on the structural rules of the application domain.

232 These structural information are formalised by a visual language from which  
233 we can automatically develop a breadth-first interactive analyser. For hand-  
234 written architectural plans, the interpretation of a primitive takes into ac-  
235 count the neighbouring objects. Moreover, as we work in a two-dimensional  
236 context (images), it can induce a large combinatorics. To overcome this com-  
237 plexity problem, we propose to guide our breadth-first exploration using a  
238 spatial contextual focus.

239 This idea is close to the well known LL(k) analysis, where reading the next  
240 k tokens enables to choose without ambiguities which rule must be applied.  
241 In the same manner, in our two-dimensional analysis, we have to limit the  
242 number of token k to explore, i.e. the depth of the analysis. Consequently,  
243 contrary to the LL(k) analysis, the exploration of the following tokens does  
244 not allow to take a unique decision on the rule to apply, because we volun-  
245 tarily limit the value of k. Moreover, sometimes, the grammar is not LL(k)  
246 for any k and the analyser meets ambiguities. In those two cases, the process  
247 can not be sure to take itself the decision, and may hesitate between several  
248 hypotheses. In order to validate the right decision, we propose an analysis  
249 process that can, through its decision process, take the right decision or so-  
250 licit the user in case of ambiguities. This analyzer is based on the following  
251 characteristics:

- 252 • the expression of a priori structural knowledge of the document through  
253 a visual language based on production rules;
- 254 • a two-dimensional descending breadth first analysis;
- 255 • a spatial contextual focus of the exploration to limit the combinatorics;

- 256 • the formalization of uncertainty by the attribution of scores to each  
257 hypothesis, represented by a tree analysis branch;
- 258 • some user sollicitaions: if the ambiguities can not be resolved in the  
259 local context in an automatic manner, the user will be solicited by the  
260 analyzer to resolve the ambiguity.

261 These characteristics have been defined to ensure a good interaction between  
262 the process of analysis and the user. This interactivity allows in particular  
263 to avoid an *a posteriori* verification phase, which can become fastidious on  
264 complex documents. Indeed, the user participation, on the critical phases of  
265 the analysis of the document, has a great impact to avoid error accumulation  
266 during the analysis step and overcomes the combinatorics due to the sketch  
267 complexity.

#### 268 *4.2. Steps of the analysis*

269 The analyzer tries to match the set of primitives contained in the doc-  
270 ument with the structure model defined by the two-dimensional production  
271 rules. The interactive breadth-first analysis process consists of three stages:  
272 1) defining the local context, 2) building the analysis tree and 3) making the  
273 decision. The inner part(\*) of Figure 1 illustrates these three phases and the  
274 relationships between them.

##### 275 *4.2.1. Defining the local context*

276 The recognition of a given primitive depends on its neighbourhood in  
277 structured documents. The analyzer begins by defining a spatial contextual

278 focus that aims to limit the combinatorial exploration due to the breadth-  
 279 first exploration of tree analysis. The structured document requires a two-  
 280 dimensional context. This two-dimensional local context is defined for an  
 281 analysis tree as the maximum distance between the elements of the root and  
 282 the elements of any leaves. The choice of the size of the local context depends  
 283 on the application domain. For example, to interpret an architectural plan,  
 284 we suggest a local context with a size corresponding to the maximum size of  
 285 an opening (door, window, etc.). Figures 3(a) and 3(b) show the shifting of  
 286 the local context in two consecutive steps. Once the local context is set, we  
 287 go to the building of the analysis tree stage.

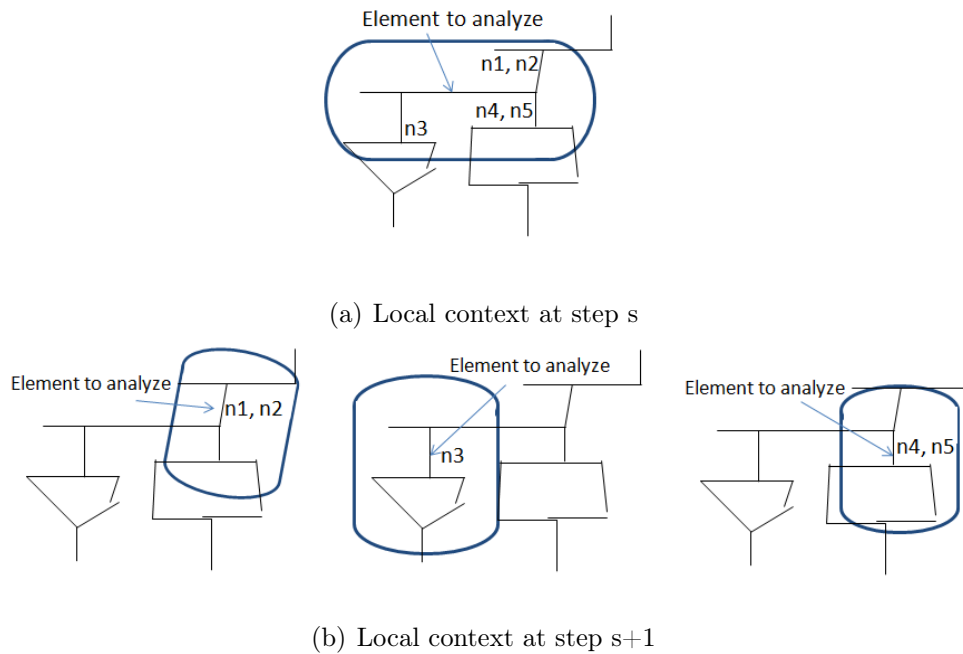


Figure 3: Dynamic adaptation of local context around the element to analyse

#### 288 4.2.2. *Building the analysis trees*

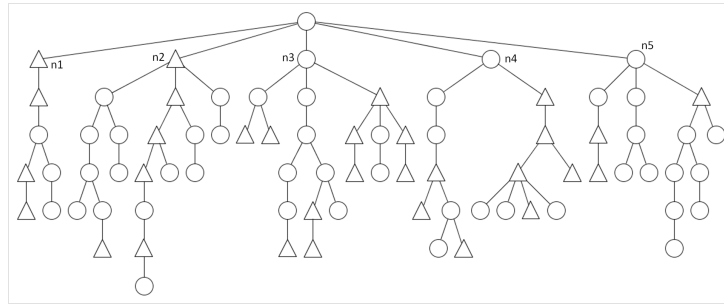
289 In this stage, the analyzer explores all the possible hypotheses of inter-  
290 pretation in the spatial context using a set of two-dimensional rules that  
291 describe the structure of the document. Each primitive can be interpreted  
292 in several ways. Each node or leaf is the application of a production rule  
293 deduced from the previous node. Every leaf or node of the tree has a score  
294 calculated from both its local score and the score obtained from the preceding  
295 nodes. Every score determines the adequacy degree to validate a production.  
296 It is calculated from each rule. The production score can also be deduced  
297 from a classifier. Each branch (hypothesis) is characterized by a score.

298 The breadth-first exploration using a local context of the IMISketch  
299 method is illustrated in Figures 4(a) and 4(b). This exploration generates  
300 some combinatorics. Consequently, we describe in section 4.2.2.a a new al-  
301 gorithm for constructing analysis trees for reducing the combinatorics.

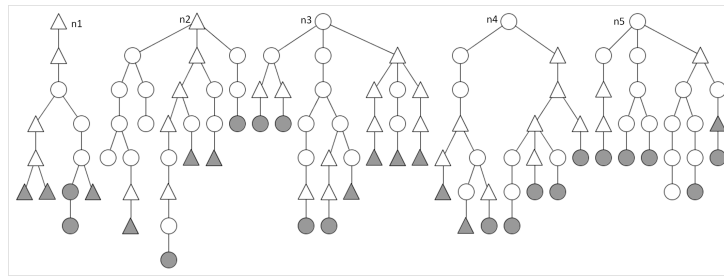
##### 302 **4.2.2.a. Our new IMISketch+ method of tree construction**

303 Each analysis tree characterizes the elements to interpret in the defined local  
304 context. Each root is the production rule that would consume this primitive.  
305 The number of analysis trees corresponds to the number of possible inter-  
306 pretations for the current primitive. The analysis tree then contains a set of  
307 complete or incomplete objects. An object is called complete if and only if  
308 this object can be found in the final result of the document interpretation.  
309 For example, in the case of the architectural plans, the complete objects can  
310 be walls, doors, windows... An incomplete object is an object that is not  
311 complete, but rather a part of complete object, i.e, we do not find it in the

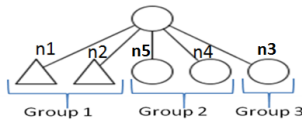




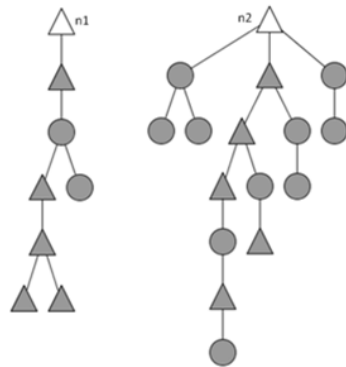
(a) IMISketch method at step  $s$



(b) IMISketch method at step  $s+1$



(c) IMISketch+ at step  $s$ . The direct son of the root are grouped by consumed elements



(d) IMISketch+ at step  $s+1$ . Only the roots ( $n1, n2$ ), belonging to the same group (Group 1) are developed

Figure 4: IMISketch Vs IMISketch+: transition between two successive analysis trees. Nodes modeled by circles refer to complete object. Nodes modeled by triangles refer to incomplete objects. The grayed leaves indicate the new applied production rules after movement of the local context.

312 final result of the interpretation. These objects can lead to complete objects  
313 (furniture, openings).

314 Despite the use of a local context, the problem of combinatory and there-  
315 fore computing time, is an important problem that must be addressed to meet  
316 the criteria of acceptability and usability of this type of interactive analy-  
317 sis with a user. The combinatory is mainly due to the number of applied  
318 production rules in the local context.

319 In fact, the development of certain nodes in tree analysis is useless. That  
320 is why, we present a new approach to build the tree analysis in order to only  
321 develop the useful nodes for making the right decision [16]. This algorithm  
322 is described below:

- 323 • If the number of analysis tree roots is equal to 1: (Figure 4(c))
  - 324 1. limit the development of direct sons of the root. If the root to de-  
325 velop is unique, we can say that only one interpretation is possible  
326 for the consumed element. The decision process is then sure to  
327 validate the right root. In this case, since we know the root to be  
328 validated, we consider unnecessary to build all the tree analysis,  
329 and we can limit the tree development to the direct sons. After  
330 validation of this root, these direct sons will be the new roots to  
331 build.
- 332 • If the number of analysis tree roots is greater than 1: (Figure 4(d))
  - 333 1. regroup the roots by consumed elements. We will not build all  
334 roots but only those that share the same elements. We want

- 335 to find the right interpretation in order to consume the element  
336 within the root.
- 337 2. order these groups by their scores. Each group has a score de-  
338 rived from the roots within it. This score is the score of the best  
339 hypothesis (branch) located in the group.
  - 340 3. develop only roots belonging to the group having the best score.
  - 341 4. build analysis tree as long as the following conditions are verified:
    - 342 – the newly consumed element is in the local context of research;
    - 343 – the number of consumed elements in each hypothesis (branch)  
344 is below a threshold. In the architectural plans, we note that  
345 the number of primitive of an opening can not exceed 10 prim-  
346 itives. For this we fix this threshold to 10.
    - 347 – the number of complete elements belonging to a branch is less  
348 than a second threshold. We can fix a threshold to 3 for the  
349 architectural plans.

350 Figures 4(d) and 4(c) illustrate an example of building the analysis trees  
351 with our new method (IMISketch+). The local context is not only limited to  
352 the distance between primitives but also to the number of complete elements  
353 in hypothesis. This optimization can generate a lack of information on hy-  
354 potheses and therefore a possible ambiguity. But thanks to the interactivity,  
355 this insufficiency does not influence on the final result as far as the user may  
356 be solicited to validate the right hypothesis.

#### 357 4.2.2.b. IMISketch+ Vs IMISketch

358 The aim of this section is to show the improvements deduced from the new  
359 algorithm (IMISketch+) for building analysis trees compared to the classic  
360 exploration method IMISketch (in which all possible branches are explored).  
361 To facilitate this comparison, we present an example of an artificial tree  
362 analysis. The goal is to compare, for each step, the number of developed  
363 nodes according to the placement of the local context depending on the  
364 elements to be analyzed. The transition of an analysis tree to the next  
365 analysis tree involves a shift of the local context. This movement allows  
366 applying the other productions. In fact, the building of the new analysis  
367 trees is not the rebuilding of the whole branches but only the new found  
368 productions by shifting the context.

369 Figure 4(a) shows the result of the exploration according to the IMISketch  
370 method. The number of interpretation (nodes) is equal to 80 interpretations.  
371 The tree construction based on the IMISketch+ algorithm generates only 6  
372 interpretations (Figure 4(c)). Figures 4(b) et 4(d) illustrate the new analysis  
373 trees by moving the local context. After two successive construction steps  
374 of analysis trees, we went from 111 interpretations (IMISketch method) to  
375 28 interpretations (IMISketch+), this corresponds to a gain of about 75% of  
376 computation time. These optimizations are generic and do not depend on  
377 the category of structured document to analyze.

#### 378 4.2.3. Making the decision

379 Once the tree is well constructed, we start the decision phase. The role  
380 of the decision process is to validate the right hypothesis among a set of

381 competing hypotheses generated with a descending breadth first analysis.  
382 Two cases are presented (section 4.2.2): case of a single root and case of  
383 several roots.

- 384 • Case of a single root: this root is validated.
  
- 385 • Case of several roots: the hypothesis (branch) having the highest leaf  
386 score in each tree is considered. These branches are sorted by the score  
387 of each leaf. Then we compare the obtained branches. Two cases may  
388 appear:
  - 389 – Implicit validation: when the analyzer is confident enough to  
390 choose the correct root without requesting the user. The ana-  
391 lyzer implicitly validates the root which has the branch having  
392 the highest score.
  - 393 – Explicit validation: when the decision process requires the user to  
394 validate the right decision. In practice, if the difference between  
395 the branch with the highest score and another branch is below a  
396 threshold, called *threshold of confidence* and these two branches  
397 are contradictory (at least one joint primitive is not consumed by  
398 the same rule production), the user intervention is required.

399 The decision is not limited only to validate the correct root but can also vali-  
400 date directly a part of the branch (hypothesis), for accelerating the analysis.  
401 In general, if the direct son of a node is unique, the validation of this node  
402 generates automatically the validation of its direct son. The decision making  
403 is summarize in Algorithm 1.

```

Function Making the decision( right hypothesis : list of nodes) : list
of nodes
    validated-nodes : list of nodes;
    validated-nodes.add(root of the right hypothesis) ;
    successor ← validated-nodes.lastElement.successor;
    While (Number of successor == 1) do
        validated-nodes.add(validated-nodes.lastElement);
        successor ← validated-nodes.lastElement.successor
    done
    return validated-nodes;
End

```

Algorithm 1: Decision algorithm

404 When the correct root is validated, other roots are put on hold and the  
405 new roots are either the sons of this root if exists, or the waiting roots  
406 otherwise and the analyzer goes back to the first step (defining the local  
407 context step). The analysis is complete when no more production rule is  
408 applicable.

## 409 5. Implementation of IMISketch

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

412 5.1. Use of CD-CMG

413 Given that architectural plans are two-dimensional structured documents,  
414 for modeling our interactive analyzer, we adopt the two-dimensional gram-  
415 mars. We use the context-driven constraint multiset grammars (CD-CMG)  
416 (detailed in [28]). The analysis process is driven by the context that involves  
417 a significant decrease in the complexity of the analysis process. The score  
418 calculated by each production (node) is due to preconditions and constraints  
419 of the rule production [28]. Equation 1 defines the manner that we calculate  
420 the score for each production. The use of the square root is a normalization  
421 using a geometric average.

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

422 Deducing  $\mu_{preconditions}$ : the computing of  $\mu_{preconditions}$  is simply the fuzzy  
423 application of the precondition block. Each DSC <sup>2</sup> is evaluated and the  
424 corresponding membership degrees are merged: a fuzzy conjunction (t-norm)  
425 is used for an '&', whereas a fuzzy disjunction (t-conorm) is used for an '|'.  
426 Once again, the resulting degree is normalized to avoid giving an advantage  
427 to productions with less DSC.

428 Deducing  $\mu_{constraints}$ : the computing of  $\mu_{constraints}$  is based on the same  
429 principle than the computing of  $\mu_{preconditions}$ . It means that each constraint  
430 (structural or statistical) must return a membership degree. In practice,  
431 defining fuzzy structural constraints is often straightforward and recognition  
432 systems we use are based on fuzzy inference systems from which we obtain

---

<sup>2</sup>DSC is a specific constraint modeling both a location in the document and elements that are awaited in it.

433 such degrees [34].

434 Equation 2 determines the degree of adequacy (score) of a hypothesis.  
435  $|PS|$  is the number of production in the considered branch (referred PS).  $\rho_{P_i}$   
436 is calculated by Equation 1.

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

437 A production rule can call an external classifier to recognize the symbols.  
438 In our application context, the classifier is used for the recognition of the  
439 types of opening (eg door, window, etc.). Each recognition is associated to a  
440 score. This classification system is based on first-order Takagi-Sugeno (TS)  
441 fuzzy inference system [1].

## 442 5.2. Impact of the primitives

443 To demonstrate the interest of using the polygons (cf. section 3) to reduce  
444 the combinatorics, we compare a first set of production rules for interpreting  
445 primitives composed only with segments (Table 1(Ps-P5)) and a second set  
446 of rules dealing with two kinds of primitives : the segments and polygons.  
447 This second set consists of the same rules enriched by three new rules that  
448 allow to analyze polygons (Table 1(P6-P8)).

449 By applying the production rules described in Table 1 to a set of segments  
450 extracted from an architectural plan (Figure 5(a)), we obtain the analysis  
451 tree illustrated in Figure 5(c). In the next step, the local context is shifted  
452 (Figure 5(b)). We then obtain tree analysis described in Figure 5(d). Despite  
453 this example is quite simple, these two analysis trees contain 28 nodes.

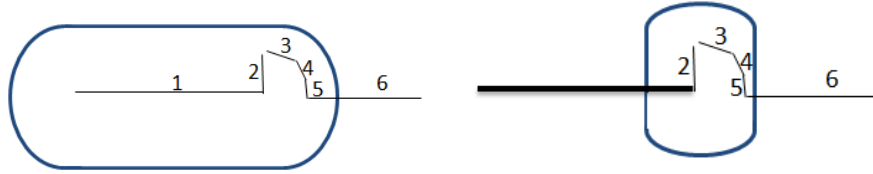
454 Now, we integrate the new polygon primitive. We add production rules  
455 illustrated in Table 1(P6-P8). To interpret the segment 1, the analyzer de-



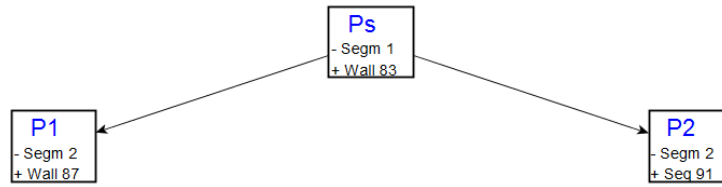
Table 1: Example of production rules used for the architectural plans using **segments** (Ps-P5) and using **polygons**(P6-P8)

Name	Created elements	Consumed elements
P <sub>s</sub>	Starting wall	longest <b>segment</b> in the document
P1	Wall	<b>segment</b> at the end of an other <i>wall</i>
P2	Sequence	<b>segment</b> at the end of an interpreted <i>wall</i> or <i>sequence</i>
P3	Opening and wall	a <i>sequence</i> and two collinear <i>walls</i> or a <i>sequence</i> and <i>wall</i> and a <b>segment</b> which are collinear
P4	Door	an <i>opening</i>
P5	Window	an <i>opening</i>

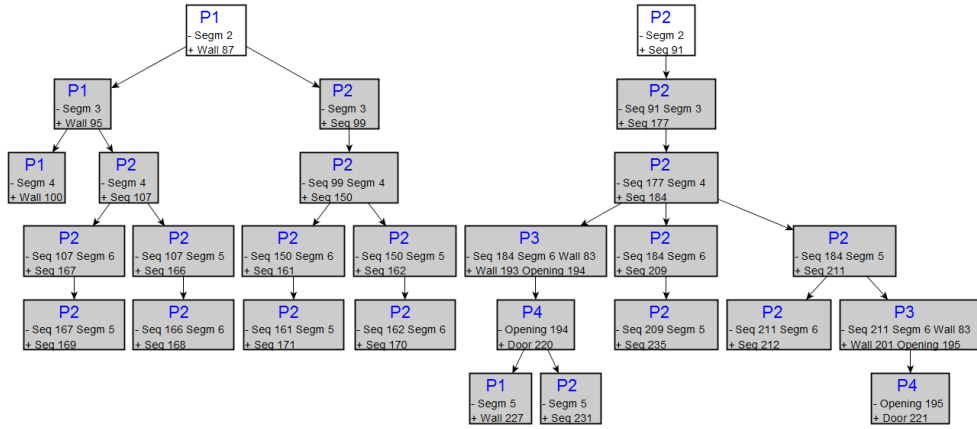
Name	Created elements	Consumed elements
P6	Wall	<b>polygon</b> at the end of an other <i>wall</i>
P7	Sequence	<b>polygon</b> at the end of an interpreted <i>wall</i> or <i>sequence</i>
P8	Opening and wall	a <i>sequence</i> and <i>wall</i> and a <b>polygon</b> which are collinear



(a) Local context at step  $n$  (centered on the segment 1)    (b) Local context at step  $n+1$  (centered on the segment 2)



(c) Analysis tree at step  $n$



(d) Analysis tree at step  $n+1$ . The grayed leaves indicate the new applied production rules after shifting the local context.

Figure 5: Position of the local context (box) during the analysis and the associated analysis trees. The primitives only consist of segments.

456 termines the local context (Figure 6(a)) and builds the tree analysis shown  
457 in Figure 6(c). In the next step, the segment 2 will be interpreted, the local  
458 context associated with this segment is illustrated in Figure 6(b) and the  
459 new tree analysis is described in Figure 6(d). Thanks to the chaining of seg-  
460 ments in polygons, we can reduce the combinatorics to 50%. We went from  
461 28 nodes to 14 nodes.

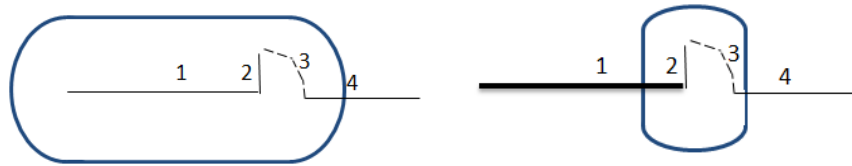
462 We have shown in this section, the impact of using the chaining of seg-  
463 ments on building of the analysis tree. This improvement induces optimiza-  
464 tions in the decision process. Moreover, using polygons increases the chances  
465 to validate a set of nodes. (cf. section 4.2.3).

466 The use of polygons as a primitive not only reduces the combinatorics  
467 for structural recognition but also guarantees a better shape recognition for  
468 obtained symbols through a more accurate representation of its constituents.  
469 Indeed, once the symbol is well recognized structurally, all the segments  
470 forming the symbol (the segments that are primitive and chained segments  
471 forming the polygon) will be forwarded to the classifier to label the symbol.  
472 Indeed, once the symbol is well structurally recognized, all the segments of  
473 symbol (the segments that are primitive and chained segments forming the  
474 polygon) will be transmitted to the classifier to label this symbol.

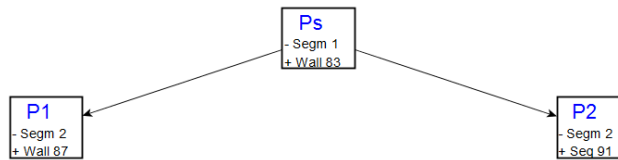
475 Once the tree analysis is explored, the analyzer starts the final phase: the  
476 decision making. In this phase, illustrated in the next section, the user may  
477 be solicited for removing ambiguities.

### 478 *5.3. Ambiguity cases*

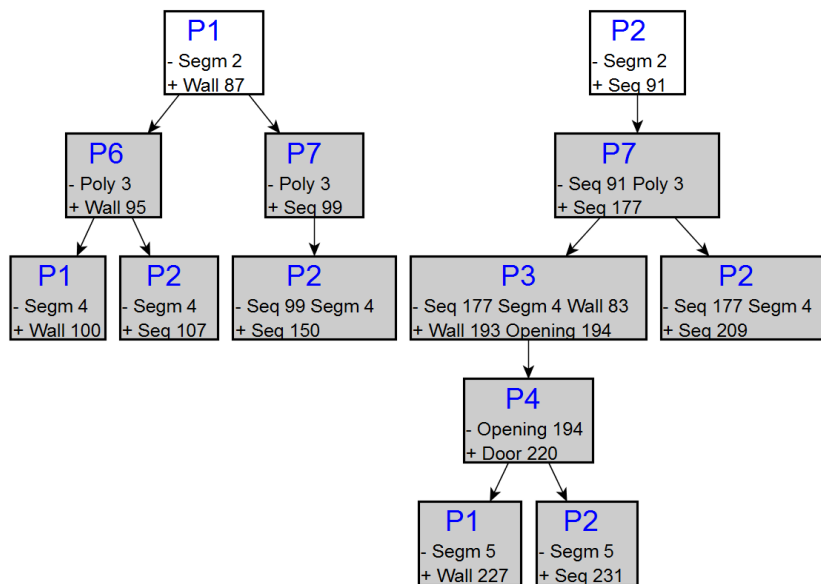
479 In this section, we show an example of ambiguity which requires a user  
480 solicitation. The aim is to analyze the set of primitives extracted from an



(a) Local context at step n (centered on the segment 1) (b) Local context at step n+1 (centered on the segment 2)



(c) Analysis tree at step n



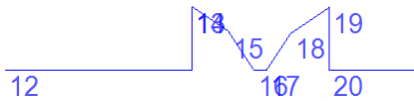
(d) Analysis tree at step n+1. The grayed leaves indicate the new applied production rules after shifting the local context.

Figure 6: Position of the local context (box) during the analysis and the associated analysis trees. The primitives consist of segments and polygons. The dotted line (primitive 3) means the polygon.

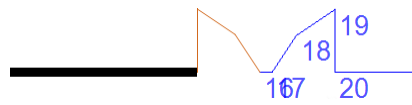
481 architectural plan (Figure 7(a)). At the step illustrated in Figure 7(b), the  
482 decision process decides to report the decision on the user because the two  
483 branches in competition are contradictory and the difference between the  
484 two scores is lower than the threshold of confidence. The decision process  
485 launches an interaction with the user (Figure 8), and proposes to validate the  
486 first hypothesis. If the user validates this hypothesis, the document will be  
487 recognized as two doors (Figure 7(c)). If the user declines the first hypothesis,  
488 the system implicitly validates the second hypothesis (Figure 7(d)) and the  
489 interpreted document will be made of one window. We note that the interface  
490 is basic for the first version; we have not focused on the HCI design but on  
491 the fact to interact with user. We focus on "*when*" the interpretation process  
492 solicits the user. "*How*" to interact will be the subject of our future work.

## 493 6. Experimental results

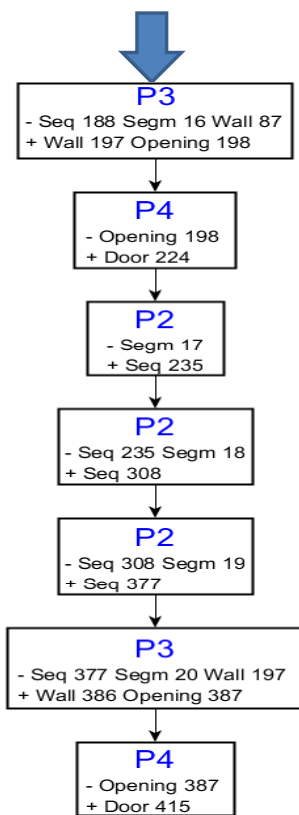
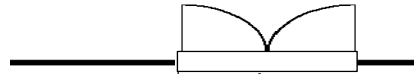
494 In this section we report different results obtained with the complete  
495 interactive recognition system. The results are focused on the impact of the  
496 presented optimizations : the use of the polygon primitive and the building  
497 strategy of tree analysis in terms of recognition rate and of computing time.  
498 For this reason, we propose three versions of IMISketch (Figure 10). The first  
499 one (referred as IMISketch) explores all the hypotheses (branches) of the tree  
500 analysis in a local context with a set of primitives that contains only segments  
501 (without polygons). The second version (referred as IMISketch+(Seg)) takes  
502 into account the new algorithm to build the analysis tree keeping the same  
503 set of primitives (only segments) as in the first version. The last version  
504 (referred as IMISketch+(Seg&poly)) incorporates the both optimizations:



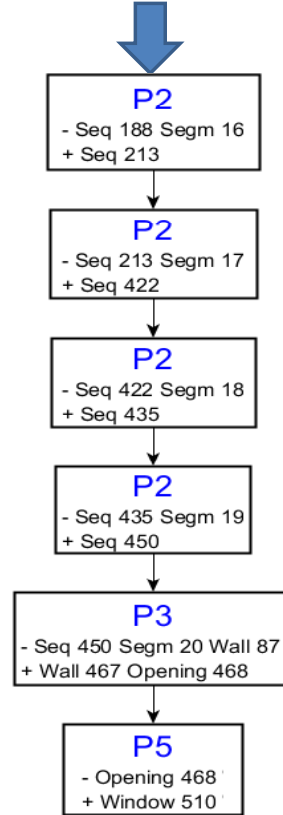
(a) Primitives of the document to analyze



(b) Interface step



(c) The first hypothesis: transforming a set of primitives into a wall and two doors



(d) The second hypothesis: transforming a set of primitives into a wall and a window

Figure 7: Example of ambiguity between two hypotheses

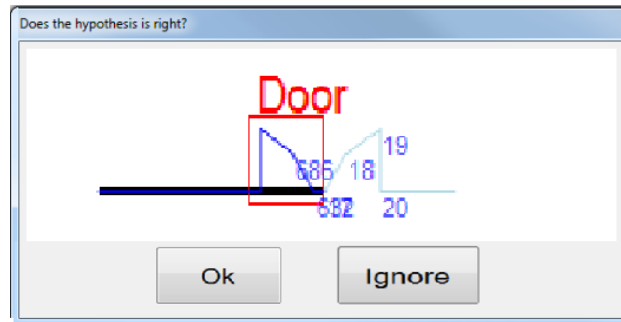


Figure 8: User interface: the analyzer proposes a user to validate the hypothesis described in Figure 7(c). If the user don't validate this hypothesis, the analyzer will validate the hypothesis implicitly described in Figure 7(d)

505 the use of the polygon primitive and the new algorithm to build the analysis  
 506 tree (Figure 10).

507 The experiments were carried on 69 handwritten architectural plans of  
 508 varying complexity drawn by around ten different people. Each architec-  
 509 tural plan is composed of a set of walls, doors, windows and sliding win-  
 510 dows. The databases contains 2641 walls, 555 doors, 377 windows and 401  
 511 sliding windows. Some examples of architectural plans are illustrated in Fig-  
 512 ure 9(a). For each architectural plan, we compare, for the three versions  
 513 (IMISketch, IMISketch+(Seg), IMISketch+(Seg&poly)), the final interpre-  
 514 tation result and the obtained computing time (Figure 9).

515 Concerning the recognition rates (cf. Figure 9(b)), we note that the  
 516 optimizations performed to IMISketch do not affect system performance. In-  
 517 deed, we observe a very light difference between IMISketch, IMISketch+(Seg)  
 518 and IMISketch+(Seg&poly) in terms of recognition. With the user solicita-  
 519 tion, during the analysis, we obtain 97.96% with IMISketch, 98.13% with

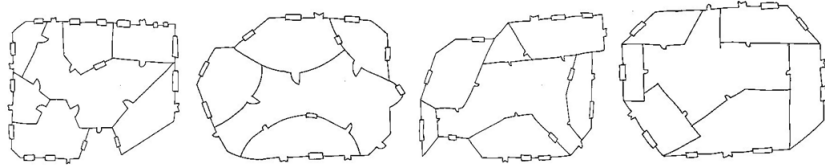
520 IMISketch+(Seg) and 97.93% with IMISketch+(Seg&poly) (Figure 9(b)).

521 This can be explained as the hypotheses proposed by the three methods  
522 are not the same. IMISketch produces more hypotheses than IMISketch+(Seg)  
523 and IMISketch+(Seg) generates more hypotheses than IMISketch+(Seg&poly).  
524 The number of competing hypotheses is more important in IMISketch than  
525 IMISketch+(Seg&poly). This might suggest that there are more chance to  
526 have the right hypothesis with IMISketch, but the generated confusions are  
527 also potentially more numerous. In the end, the results in terms of recogni-  
528 tion performance are very comparable.

529 Moreover, the user solicitation decreases from 6 interventions in IMISketch  
530 to about 4 interventions in IMISketch+(Seg&poly)(Figure 9(b)). This de-  
531 crease shows a reduction in the competing hypothesis that can lead to ambi-  
532 guities. This is due to the polygons which has reduced the number of nodes  
533 per branch and therefore the conflicts between hypotheses. The obtained  
534 errors (about 2%) are due either to poor calibration of the local context, or  
535 the badly drawing of certain symbols.

536 The details of computation time per image between the three versions  
537 (IMISketch, IMISketch+(Seg) and IMISketch+(Seg&poly)) are shown in Fig-  
538 ure 9(c). The images are ordered by the computing time according to  
539 IMISketch+(Seg&poly). We improve the computation time from about 11  
540 minutes of average computing time per image (with classical IMISketch) to  
541 about 4 minutes 43 seconds (with IMISketch+(Seg)), i.e. a gain of 57%.  
542 IMISketch+(Seg&poly) further reduces computation time to attain about 2  
543 minutes 31 seconds, which presents a total gain of 77%. Depending on the  
544 complexity of the plans, the gain of computing time can reach 90% per image

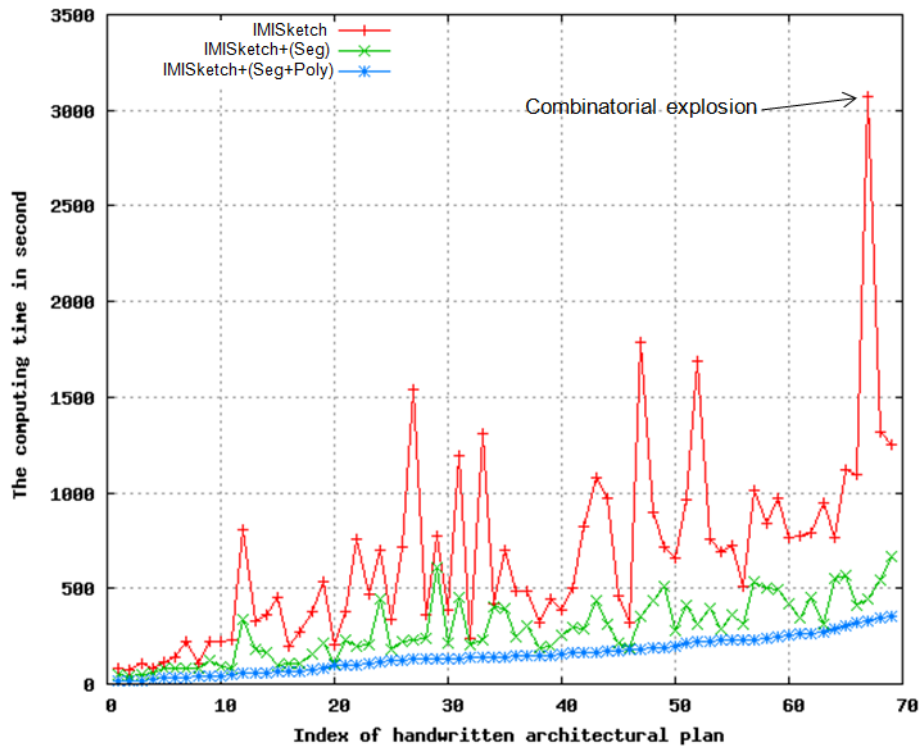




(a) Examples of architectural floor plans

	IMISketch	IMISketch+(Seg)	IMISketch+(Seg+poly)
Types of primitive	Segments		Segments and polygons
Building the analysis trees	All possible hypotheses in the local context (classical approach)		Using the new approach described in Section 4.2.2
Number of unrecognized symbols	81	74	82
Recognition rate	97.96%	98.13%	97.93%
Average computation time	11m03	4m43	2m31
Average user solicitation	6.07	5.17	3.59

(b) Characteristics of IMISketch versions and recognition rate on architectural plans per version



(c) Computing time per image showing the disappearance of combinatorial peak using IMISketch+

Figure 9: Experimental results

545 (image with index 67). We note, in Figure 9(c), that the peaks in the analysis  
546 time have completely disappeared with the optimized method. Thus, there is  
547 no risk for a long wait to interpret a more complex plan. These computation  
548 times are obtained in real conditions, i.e. in the presence of a user in the  
549 loop of analysis. Today, the average computation time is 2 minutes 31 per  
550 image. In a future work, we will try to reduce the computation time to attain  
551 around 1 min 30 per image: experimentally, it seems the ideal timing that  
552 allows the user to follow in real time the recognition process.

553 The experimental results are very encouraging. They suggest that it is  
554 possible to introduce a descending breadth first analysis by controlling the  
555 generated combinatorics. This supports the idea of conceiving an interac-  
556 tive systems for the document recognition. User solicitations, driven by the  
557 analyzer, guarantee the obtaining of very high rate of reliability even when  
558 considering the treatment of complex documents. the use of the polygon  
559 primitive does not have a negative impact on the structural recognition rate,  
560 in addition it reduces the number of user intervention during the analysis  
561 and also accelerates the computation time.

#### 562 *Current state of IMISketch system*

563 Optimizations of our IMISketch method were carried out to the inter-  
564 pretation of more complex architectural plans containing a dozen furniture  
565 types (toilet, table, bed...), 3 types for openings (door, window and sliding  
566 window) and walls. Indeed, some experiments on 24 architectural handwrit-  
567 ten plans (see Figure 10) show that the structural recognition rate increase  
568 from 91% without user solicitations to 96% with user solicitations.

569 We can notice that the best compromise (recognition rates/user solicita-



Figure 10: Interpretation of a complex architectural floor plans

570 tions) is obtained with 12 user solicitations per image: it means that 4% of  
 571 primitive interpretations require the user solicitation. 49% of user solicita-  
 572 tions are useful to take the right decision which is not the best hypothesis  
 573 proposed by the analyser (more details in [16]).

## 574 7. Conclusion

575 In this paper, we have presented a complete generic method to interpret  
 576 sketches such as 2D architectural floor plans. This method consists of a  
 577 preprocessing phase in which we extract useful primitives which constitute  
 578 the inputs of an interactive analyzer. This analyzer is based on a competi-  
 579 tive breadth-first exploration of the analysis tree according to a dynamical  
 580 local context of the document. The decision process is able to solicit the  
 581 user in the case of strong ambiguity. Generally, the competitive breadth-first  
 582 exploration generates combinatorics. Our interactive method integrates an  
 583 optimization strategy for solving combinatorics. This strategy concerns the

584 preprocessing phase and the analysis phase. The preprocessing phase avoids  
585 the agglomeration of primitives in reduced zones of document. At the anal-  
586 ysis phase, the strategy is based on a dynamic construction of analysis tree  
587 by controlling the depth of each branch following a set of criteria specified in  
588 the current local context. It may be noted that this strategy is generic and  
589 therefore it could be easily applied to other types of structured documents,  
590 and other analyzers characterized by a breadth-first exploration. The first  
591 tests of this interactive analyzer have been made on 2D handwritten archi-  
592 tectural floor plans. Integrating the user in the analysis process is, in our  
593 view, a key point to address complex off-line sketch recognition and to avoid  
594 an a posteriori verification phase.

595 Future work will focus on extending the experimental results on large  
596 image databases containing more complex architectural plans (integration of  
597 furniture, quotes, etc.). We will also validate the criteria of acceptability and  
598 usability of the system by doing usage tests that will be conducted in collabo-  
599 ration with experts from the Loustic laboratory (<http://www.loustic.net/rennes>).

## 600 **Acknowledgment**

601 The authors would like to thank all the people who took part in the  
602 experiments. This work partially benefits from the financial support of the  
603 ANR Project Mobisketch.

## 604 **References**

- 605 [1] Almaksour, A., Anquetil, E., 2011. Improving premise structure in  
606 evolving takagi-sugeno neuro-fuzzy classifiers. *Evolving Systems* 2, 25–

607

33.

608

[2] Alvarado, C., Davis, R., 2007. Sketchread: a multi-domain sketch recognition engine. In: ACM SIGGRAPH 2007 courses. ACM, pp. 34–es.

609

610

[3] Aoki, Y., Shio, A., Arai, H., Odaka, K., 1996. A prototype system for interpreting hand-sketched floor plans. In: Pattern Recognition, 1996., Proceedings of the 13th International Conference on. Vol. 3. IEEE, pp. 747–751.

611

612

613

614

[4] Bunke, H., nov. 1982. Attributed programmed graph grammars and their application to schematic diagram interpretation. Pattern Analysis and Machine Intelligence, IEEE Transactions on PAMI-4 (6), 574 –582.

615

616

617

[5] Bunke, H., 1990. String grammars for syntactic pattern recognition. Syntactic and Structural Pattern Recognition. Theory and Applications, 29–54.

618

619

620

[6] Chan, K., Yeung, D., 2000. An efficient syntactic approach to structural analysis of on-line handwritten mathematical expressions. Pattern Recognition 33 (3), 375–384.

621

622

623

[7] Chang, H.-H., Yan, H., 1998. Vectorization of hand-drawn image using piecewise cubic bézier curves fitting. Pattern Recognition 31 (11), 1747 – 1755.

624

625

626

[8] Coüasnon, B., 2006. 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.

627

628

- 629 [9] de Brucq, D., Amara, M., Courtellement, P., Wallon, P., Mesmin, C.,  
630 1996. A recursive estimation of parameters of straight lines and circles  
631 by an extended Kalman filtering. Application to the modeling of on-  
632 line Handwritten drawings. In: International Conference on Signals and  
633 Image Processing and Applications (SIPA'96). pp. 213–216.
- 634 [10] Dosch, P., Tombre, K., Ah-Soon, C., Masini, G., 2000. A complete sys-  
635 tem for analysis of architectural drawings. International Journal On Doc-  
636 ument Analysis and Recognition 3 (2), 102–116.
- 637 [11] Fahmy, H., Blostein, D., aug-3 sep 1992. A survey of graph grammars:  
638 theory and applications. In: Pattern Recognition, 1992. Vol.II. Confer-  
639 ence B: Pattern Recognition Methodology and Systems, Proceedings.,  
640 11th IAPR International Conference on. pp. 294 –298.
- 641 [12] Fitzgerald, J., Geiselbrechtinger, F., Kechadi, T., 2007. Mathpad: A  
642 fuzzy logic-based recognition system for handwritten mathematics. In:  
643 ICDAR 2007. Vol. 2. pp. 694 –698.
- 644 [13] Freeman, I. J., Plimmer, B., 2007. Connector semantics for sketched di-  
645 agram recognition. In: Proceedings of the eight Australasian conference  
646 on User interface - Volume 64. AUIC '07. Australian Computer Society,  
647 Inc., Darlinghurst, Australia, Australia, pp. 71–78.  
648 URL <http://dl.acm.org/citation.cfm?id=1273714.1273726>
- 649 [14] Ghorbel, A., Almaksour, A., Lemaitre, A., Anquetil, E., 2011. Incre-  
650 mental learning for interactive for sketch reconition. Ninth IAPR Inter-  
651 national Workshop on Graphics RECOgnition - GREC 2011.

- 652 [15] Ghorbel, A., Anquetil, E., Lemaitre, A., 2012. Optimization analysis  
653 based on a breadth-first exploration for a structural approach of sketches  
654 interpretation. In: Document Analysis Systems (DAS), 2012 10th IAPR  
655 International Workshop on. pp. 240–244.
- 656 [16] Ghorbel, A., Lemaitre, A., Anquetil, É., 2012. Competitive hybrid ex-  
657 ploration for off-line sketches structure recognition. In: ICFHR. pp. 571–  
658 576.
- 659 [17] Ghorbel, A., Macé, S., Lemaitre, A., Anquetil, E., 2011. Interactive  
660 competitive breadth-first exploration for sketch interpretation. ICDAR  
661 2011, 1195–1199.
- 662 [18] Hammond, T., Davis, R., 2003. Ladder: A language to describe drawing,  
663 display, and editing in sketch recognition.
- 664 [19] Hammond, T., Davis, R., 2005. Ladder, a sketching language for user  
665 interface developers. *Computers & Graphics* 29 (4), 518–532.
- 666 [20] Hammond, T., O’Sullivan, B., 2007. Recognizing free-form hand-  
667 sketched constraint network diagrams by combining geometry and con-  
668 text. *Proceedings of the Eurographics Ireland 2007*.
- 669 [21] Hammond, T., Paulson, B., Oct. 2011. Recognizing sketched multistroke  
670 primitives. *ACM Trans. Interact. Intell. Syst.* 1 (1), 4:1–4:34.  
671 URL <http://doi.acm.org/10.1145/2030365.2030369>
- 672 [22] Hammond, T. A., Davis, R., 2009. Recognizing interspersed sketches  
673 quickly. In: *Proceedings of Graphics Interface 2009*. pp. 157–166.

- 674 [23] Hilaire, X., Tombre, K., 2006. Robust and accurate vectorization of line  
675 drawings. *IEEE Transactions on Pattern Analysis and Machine Intelli-*  
676 *gence* 28, 890–904.
- 677 [24] Lemaitre, A., Camillerapp, J., 2006. Text line extraction in handwritten  
678 document with kalman filter applied on low resolution image. *Document*  
679 *Image Analysis for Libraries, International Workshop on* 0, 38–45.
- 680 [25] Lemaitre, M., Grosicki, E., Geoffrois, E., Preteux, F., 2007. Preliminary  
681 experiments in layout analysis of handwritten letters based on textural  
682 and spatial information and a 2d markovian approach. In: *ICDAR 2007.*  
683 *Vol. 2.* pp. 1023 –1027.
- 684 [26] Lladós, J., López-Krahe, J., Martí, E., 1997. A system to understand  
685 hand-drawn floor plans using subgraph isomorphism and hough trans-  
686 form. *Machine Vision and Applications* 10 (3), 150–158.
- 687 [27] luen Do, E. Y., 2001. Vr sketchpad, create instant 3d worlds by sketch-  
688 ing on a transparent window. In: *Proceedings of CAAD Futures 2001*  
689 *(Eindhoven. Kluwer Academic Publishers,* pp. 161–172.
- 690 [28] Macé, S., Anquetil, E., 2009. Eager interpretation of on-line hand-  
691 drawn structured documents: The dali methodology. *Pattern Recog-*  
692 *niton* 42 (12), 3202–3214.
- 693 [29] Mao, S., Rosenfeld, A., Kanungo, T., 2003. Document structure analysis  
694 algorithms: a literature survey. In: *Proc. SPIE Electronic Imaging. Vol.*  
695 *5010.* pp. 197–207.



- 696 [30] Montreuil, F., Grosicki, E., Heutte, L., Nicolas, S., 2009. Unconstrained  
697 handwritten document layout extraction using 2d conditional random  
698 fields. ICDAR 2009 0, 853–857.
- 699 [31] Plimmer, B., Freeman, I., 2007. A toolkit approach to sketched diagram  
700 recognition. In: Proceedings of the 21st British HCI Group Annual Con-  
701 ference on People and Computers: HCI...but not as we know it - Volume  
702 1. BCS-HCI '07. British Computer Society, Swinton, UK, UK, pp. 205–  
703 213.  
704 URL <http://dl.acm.org/citation.cfm?id=1531294.1531323>
- 705 [32] Sezgin, T. M., Davis, R., 2005. Hmm-based efficient sketch recognition.  
706 In: Proceedings of the 10th international conference on Intelligent user  
707 interfaces. IUI '05. ACM, New York, NY, USA, pp. 281–283.  
708 URL <http://doi.acm.org/10.1145/1040830.1040899>
- 709 [33] Sheraz Ahmed, a. M. L., Weber, M., Dengel, A., 2011. Improved auto-  
710 matic analysis of architectural floor plans. ICDAR 2011, 864–868.
- 711 [34] Takagi, T., Sugeno, M., 1985. Fuzzy identification of systems and its  
712 applications to modeling and control. Systems, Man and Cybernetics,  
713 IEEE Transactions on SMC-15 (1), 116–132.
- 714 [35] Yamamoto, R., Sako, S., Nishimoto, T., Sagayama, S., Oct. 2006.  
715 On-line recognition of handwritten mathematical expressions based on  
716 stroke-based stochastic context-free grammar. In: Tenth International  
717 Workshop on Frontiers in Handwriting Recognition. Suvisoft, La Baule  
718 (France).