Extension of a bi-dimensional grammar for online interpretation of structured documents: application on architecture plans and geometry domains

Omar Krichen, Nathalie Girard, Eric Anquetil

▶ To cite this version:

HAL Id: hal-02929753

https://hal.archives-ouvertes.fr/hal-02929753

Submitted on 3 Sep 2020

Extension of a bi-dimensional grammar for online interpretation of structured documents: application on architecture plans and geometry domains

Omar Krichen
Univ Rennes, CNRS, IRISA
F-35000 Rennes, France
omar.krichen@irisa.fr

Nathalie Girard
Univ Rennes, CNRS, IRISA
F-35000 Rennes, France
nathalie.girard@irisa.fr

Eric Anquetil
Univ Rennes, CNRS, IRISA
F-35000 Rennes, France
eric.anquetil@irisa.fr

Abstract—In this paper, we present an extension of the bi-dimensional grammar called LCD-CMG (Layered Context Driven Constraint Multiset Grammar), a generic formalism able to model and analyse different types of structured documents. We place ourselves in the context of eager interpretation of the user's strokes, on pen-based tablets. The main constraint is therefore the real-time user interaction. LCD-CMG context sensitiveness reduces the combinatory associated to the analysis of the document structure (spacial/geometric relations between elements, *e.g.* the relation between a door and its associated wall). However, when we introduce complex semantic notions in the document (rooms in the architecture domain, polygons, quadrilaterals, etc in geometry), we are confronted with a combinatorial explosion. To deal with this issue, we extend the grammar parser with a semantic context analyser. This analyser models a global vision of the document, *i.e.* high-level relations between the elements, whereas the LCD-CMG parser models a more local vision of the document structure. The coupling of this semantic context analyser with the CD-CMG parser considerably reduces the analysis complexity, enhances the expressiveness of the formalism, and can be generalized to different types of structured documents. Experiments on two application domains, the creation of architecture plans and that of geometric shapes, show the relevance and the improvement of the extension.

Index Terms—Online recognition of handwritten strokes, Bi-dimensional structured document interpretation, Pen-based interaction

I. Introduction

This work is part of the ACTIF project, which aims at enhancing middle-school pupils performance in geometry learning. To achieve this goal, we designed an intelligent tutoring system that enables the user to draw freely with the stylus on a pen-based tablet. The tutor has also to provide prompt feedback on the pupil production. Therefore, the system must interpret the hand-drawn sketches in real-time for the feedback to be instant and adaptive. In this paper, we focus on the recognition aspect of our tutor. A lot of works have been realised for the online interpretation of structured documents in a digital learning context, such as chemical drawings [8], truss diagrams in engineering [5], or UML diagrams [9]. The type of interpretation suitable for real-time user interaction is called *eager* interpretation, and consists in trying to analyse the document during its composition. In this paper, we present an eager interpretation approach that could be applied to all types of structured documents, since it is based on a generic bi-dimensional formalism. We can identify in the literature two ways of approaching the document analysis problem, statistical approaches [2] [10] [11], well adapted for isolated symbols recognition, and syntactic approaches [3] [4] [6] [9] [12], that are able to model structural interactions between elements. There are also hybrid methods that combine the two approaches, by defining visual languages with enough expressiveness to model the prior domain knowledge as well as to exploit classifiers in the recognition process [1]. In [7], the authors define a new class of visual grammars, CMG (Constraint Multiset Grammar). This formalism introduces the notion of spacial constraints to model the prior domain knowledge and recognise hand-written gestures (*e.g.* a transition is an arrow that is connected to two states in a state-transition diagram, *c.f.* fig. 1, 2). The parser associated to this grammar is based on a bottom-up approach, such as it checks all combinations of circles and arrows in order to apply the correct production rule (in this example, (s1, s2, arrow), (s1, s3, arrow), (s2, s4, arrow),...). The complexity here is exponential, which is a limitation with respect to the real-time analysis constraint. In [1], the authors define Context-Driven CMG, to solve the limitations of CMG. The principle here is to externalise the contextual knowledge to drive the analysis by context, *i.e.* only the elements that satisfy specific contextual conditions are triggered/checked (spacial relations, *e.g. connected(arrow, circle)*). This context sensitiveness reduces highly the complexity of the analysis, however it is limited to a local vision of the document structure, and is not suited to deal with the recognition of semantically complex
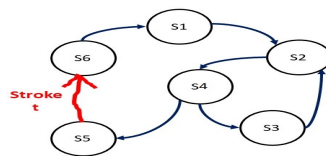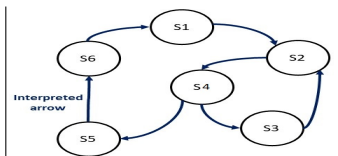

Fig. 1: Hand-drawn stroke

Fig. 2: Recognized arrow

elements in real-time. For the example in Fig. 1, CD-CMG local sensitiveness is not sufficient to control the complexity of cycles detection. In [13], a first extension to CD-CMG is presented which consists in the definition of multiple layers of interpretation and the formalisation of hierarchy between production rules (Layered CD-GMG). **In this paper, we present an extension to the LCD-CMG parser, called the semantic context analyser**. This analyser models a global vision of the document's structure, and is responsible of finding particular patterns within a set of related elements. Just like LCD-CMG externalises the local contextual knowledge (*e.g.* spacial relations between two segments) to drive the analysis search, the semantic context analyser externalises the global contextual knowledge (*e.g. n* connected walls constitute a particular room). In this paper, we show that the combination of LCD-CMG local context vision and the semantic context analyser global context vision reduces heavily the analysis complexity, while preserving the genericity of the formalism. The paper is organized as follows. LCD-CMG is explained in Sec. II. Sec. III describes the formalism parser extension. To demonstrate the genericity and applicability on different types of structured documents, experiments and results on two application domains, namely architecture plans design, and geometric figures constructions, are presented in Sec. IV. Conclusion and future works are given in Sec. V.

II. LCD-CMG

We describe formally in this section the LCD-CMG formalism, from its origin, CMG, to its last extension.

A. Constraint Multiset Grammar

CMG is formally defined as follows:

Definition 1. *CMG is a tuple G=(V_N, V_T, S, P) with:*
- *V_N: the set of non terminal symbols = symbol classes;*
- *V_T: the alphabet;*
- *S: the first symbol, or axiom;*
- *P: the set of production rules.*

where V_T, the alphabet, is a set of graphical primitives such as lines, circles, and arcs, and where a production rule $p \in P$ is defined as follows:

$$\alpha \rightarrow \beta \left\{\text{Constraints}\right\} \mid \alpha \in V_N{}^+, \beta \in (V_T \cup V_N)^*$$

which means a mutliset of elements β are transformed into a multiset of elements α if a set of constraints are satisfied (these constraints can define the spacial relations).

B. Context Driven Constraint Multiset Grammar

In CD-CMG, two additions are made to CMG, the first consists in the redefinition of V_T, *i.e.* $V_T = \{stroke\}$, which means there is no need for systematic segmentation of the user's handwritten input. The second extension concerns the production format, where $p \in P$ is defined as follows:

$$\alpha \rightarrow \beta \left\{\begin{array}{l}\text{Preconditions}\\\text{Constraints}\\\text{Postconditions}\end{array}\right\} \mid \alpha \in V_N{}^+, \beta \in (V_T \cup V_N)^+$$

Door: d → stroke t with:
Layer: 1
Preconditions:
(Wall: W1) [LengthSegment] (t) [first] & (Wall: W1) [LengthSegment] (t) [last]
Constraints:
Recognizer(d, Door)
Postconditions:
(d)[In] (Stroke: t) [First] $\implies$ [invertedDoor → d]

Fig. 3: Door production rule

Here, two blocks are added in the production rule. The preconditions are a set of contextual constraints that have to be satisfied in order to trigger the rule. The postconditions block models new contextual information. Thereby, the contextual knowledge is externalized from the constraints block, into the Preconditions and Postconditions blocks, which are based on the concept of *Document Structural Context* (DSC).

Definition 2. *A DSC is defined by $(\lambda)[position](\gamma)[part]$ where:*
- *λ is a set of reference elements;*
- *position is a zone (i.e a position) related to λ;*
- *γ is a set of awaited symbols in this zone;*
- *part is a part of the awaited symbols.*

The **preconditions** are DSCs that have to be satisfied for the production to be triggered. The **postconditions** are DSCs that are created after the production reduction, and model the update of the document structural context. The **constraints** block checks that the shape of β is consistent with the production (*e.g* by calling a classifier), and decides if it is relevant to reduce β into α.

C. Layered Context Driven Constraint Multiset Grammar

The latest extension of the formalism consists in defining a hierarchy between different layers of interpretation, in order to control the search space size [13]. For example, a first layer would model the prior knowledge related to the definition of graphical primitives such as segments, circles, or angles in the geometry domain. A second layer would model the structural interactions between the elements (intersection, perpendicularity, parallelism), and a third layer would model the creation of semantic elements such as trapezoids or triangles. A production rule in LCD-CMG is defined as follows:

$$\alpha \rightarrow \beta \left\{\begin{array}{l}\text{Layer}\\\text{Preconditions}\\\text{Constraints}\\\text{Postconditions}\end{array}\right\} \mid \alpha \in V_N{}^+, \beta \in (V_T \cup V_N)^+$$

We place ourselves now in the context of architecture plans design. Fig. 3 presents the modeling of the creation of a door, in the architecture domain, by a LCD-CMG production rule. Since the parameter is a stroke *t*, this rule is in **Layer 1**, *i.e.* it will compete with other productions of the first layer, such as Wall → t, or Window → t. The verification step is modelled by the preconditions block: the production is
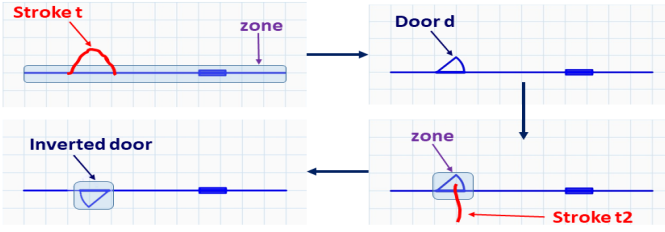
Fig. 4: Composition and eager interpretation in LCD-CMG

Fig. 5: Drawn stroke

Fig. 6: Analysis result

Fig. 7: Part of analysis process

triggered only if the context is valid(*i.e.* t is linked by its extremities to the wall zone (*c.f.* Fig. 4). After checking the context, the parser checks the shape of t in the constraints block by calling a statistical classifier. If the constraints are satisfied, the production is reduced and new DSC or zones will be created in the postconditions block. This block models the prediction phase, and the production rule of door inversion will be triggered if a stroke t2 intersect the newly created zone (*e.g.* Fig 4).

D. LCD-CMG parser

The analysis process in LCD-CMG is hierarchical and driven by the context at the same time. It is driven by the context in the sense that it combines a bottom-up approach (searches for preconditions DSCs that are satisfied by a new element), and top-down approach (searches for elements that satisfy each newly created DSC). This way, only the rules that are contextually valid are triggered by the parser, which limits the search space size. In another hand, the analysis process is hierarchical in the sense that the parser begins its search at the first layer (at stroke level), and moves on the next only if there is no more applicable rules belonging to this layer. Fig. 5 and 6 illustrate the composition of a new wall in a plan composed of two rooms. Fig. 7 represents a part of the parser analysis process related to this composition. In this domain, four layers are defined, and this hierarchy appears clearly through the analysis tree. The nodes of this tree represent the triggered rules, while the blue dotted path represents the reduced productions. We can see for example that in the second layer, only divisions productions are triggered (w6,w7,w8←w2,w5 means the new wall w5 will divide the old one w2 in two which creates three new walls). In the third layer, which is a semantic layer, the triggered rules are graph creation rules (g2,w8←g1,w8). g1 represents the state of connected walls before the new wall is interpreted. Since the division productions modify the structure, the graph g1 is replaced by a new graph g2. The final layer consists in finding new rooms by finding cycles in the updated graphs. The reduction of the rule g2, room r←g2,w8 is repeated until there are no more new cycles involving w8 in g2.

E. LCD-CMG limitations

We can see that for a relatively simple example (*c.f.* Fig. 7), the analysis process is quite tedious. As the document becomes more complex, the combinatory becomes too much for the parser to handle, and the real-time interpretation constraint
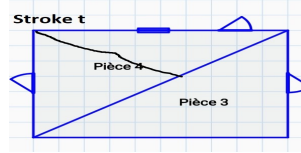
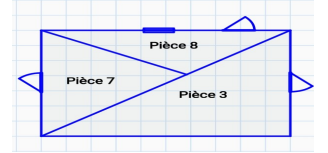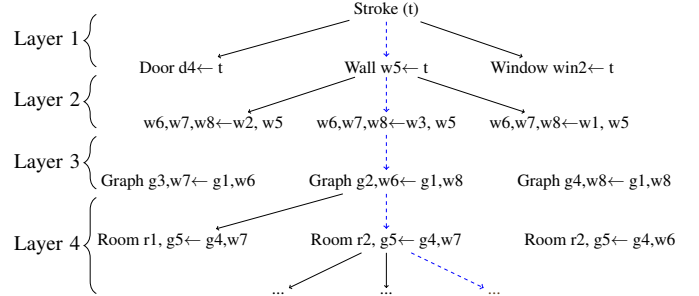is no longer satisfied. This is due to the fact that the global semantic context (in this case, the rooms) is handled by the rules in such as manner that to create n rooms from n cycles, the room production rule has to be reduced n times. Ideally, this rule should be triggered only one time to create n rooms. Unfortunately, since we model prior knowledge in the grammar, the number of α elements resulting from the reduction of β should be known in advance. Another limitation is the formalism should be easily manipulable by a designer/user. If an expert/designer wants to use LCD-CMG to model the prior knowledge of his domain of expertise, the definition of the rules should be fairly simple, such as room r $\leftarrow$ wall$_1$, wall$_2$,...wall$_n$. As it stands, the formalism, although expressive, is out of the reach of the ordinary user. To tackle these limits, we extend the grammar parser by externalizing the global context, with the introduction of a semantic context analyser, which will be presented in detail in the next section.

III. SEMANTIC CONTEXT ANALYSER

A. Automated generation from user definition

The idea is to simplify the designer task, by replacing the definition of complex production rules by a simple definition of the *connection* elements, and the *connection* DSCs (*c.f.* fig. 8). The designer also specifies the desired pattern to find between these elements (a cycle in this case). From the user specification, the Semantic Context Analyser, or C.S.A, is generated automatically, alongside the LCD-CMG parser.

Pattern: Cycle
Connection elements type: wall
Connection DSC: (wall: w_1) [Extremity] (wall: w_2) [extremity]
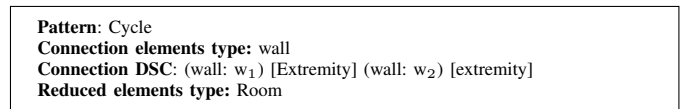**Reduced elements type:** Room

Fig. 8: Required specification for analyser generation and room modelling

The first benefit is the simplicity for the user, compared to the definition of complex rules within the grammar (*c.f.* fig. 7). The second benefit is the genericity, since in all types of structured documents, patterns such as cycles could be of interest in the knowledge modelling. For example, the same specification illustrated in fig. 8 can be used for the geometry domain, by replacing the reduced elements type (Room) by polygons.

B. Analysis process

Fig. 9 illustrates the new analysis process with the integration of the Semantic Context Analyser. Just like for the LCD-CMG parser, the combination of connection elements and connection DSCs triggers the creation and update of semantic connection graphs. At each creation/update of a graph, the semantic context analyser searches for new particular pattern that includes new connection elements. In the fig. 8 example, the creation of a new wall that satisfies one connection DSC will trigger the update of one of the graphs, as well as the search for new cycles to create new semantic elements such as rooms or polygons. These new cycles include necessarily a new wall, such as the starting point of the search is the wall's first extremity and the ending point is its last extremity. The hierarchy of the process is maintained in the sense that the semantic context analyser plays its role only when the semantic layer, which is the last interpretation layer, is activated. When the new semantic elements are created, they are fed to the CD-CMG parser. The interpretation layer is reset to the first one, and the analysis process is repeated until there is no more applicable productions. In the context of geometric figures production, a polygon, created by the semantic context analyser, could be transformed into a trapezoid or a pentagon by the CD-CMG parser.

C. Impact on analysis complexity

If we consider architecture plans design and geometric figures production, the pattern of interest consist in finding all the cycles in the document to create rooms or polygons. The first impact of externalizing the global contextual knowledge from the CD-CMG parser is the fact that one search of all
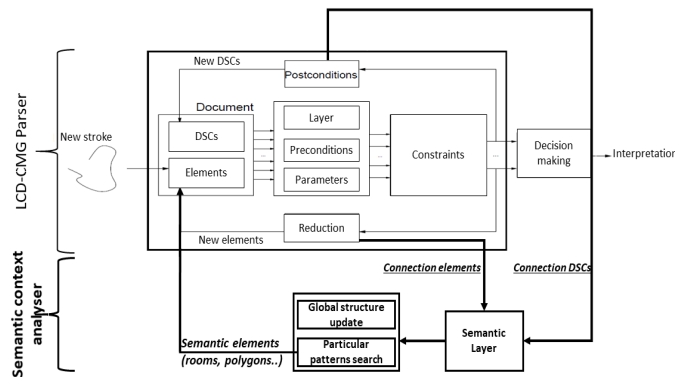

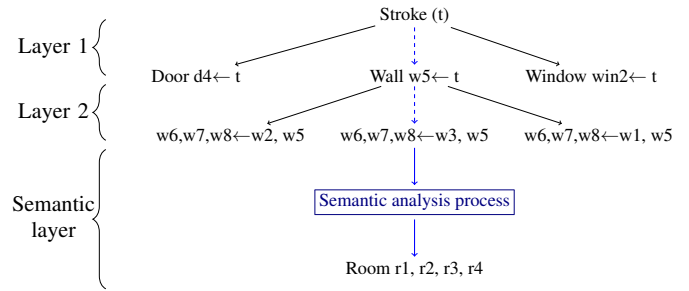Fig. 9: Interaction between LCD-CMG parser and semantic context analyser

Fig. 10: Impact on analysis tree

new cycles in the connection graph is sufficient to create the n rooms with no impact on the parser search space size. Without this externalization, n productions rules need to be reduced while search space size increasing leading to a combinatorial explosion. Fig. 10 illustrates the impact of the semantic context analyser on the analysis tree (see drawing scenario in fig. 6). The search space size is considerably reduced, especially when dealing with the creation of new rooms. Four rooms are created, corresponding to the "explicit" rooms (pièce 7 and pièce 8 in fig. 6) and "implicit" rooms. "Implicit" refers to the combinations of adjacent rooms (pièce 7 + pièce 3, pièce 8 + pièce 3) (pièce means room in French). There is a need for this "implicit" notion in order to maintain created rooms after deleting a wall that is adjacent to two of them. Even if the search space size is drastically decreased with the integration of the semantic context analyser, the fact that all cycles have to be computed is problematic when the document gets complex. We will deal with this issue in the next subsection.

D. Optimization strategy

1) Minimal cycles search: It is clear that the increase of the analysis search space size is caused by the implicit rooms computation. To control the analysis complexity, the obvious starting point is to compute only the minimal cycles that include a new element. In the example illustrated in Fig. 6, only explicit rooms would be created. However, we loose the semantics of the implicit rooms, and therefore, we loose the semantics of the whole document. If the user deletes the wall between Pièce 7 and Pièce 3, the scene will consist only in pièce 8, since no implicit room containing Pièce 7 and 3 had been created beforehand. This means that computing only the minimal cycles renders the system unusable by its target population (housing technicians). To tackle this issue, we introduce the concept of **suppression triggered analysis**.

2) suppression triggered analysis: LCD-CMG, as a visual grammar, is generative, *i.e.* the analysis is triggered by each new hand-drawn stroke. Since the deletion of a wall causes an update in the number of cycles in its containing graph, a new analysis process should take place. Fig. 11 illustrates this process. Unlike the classical stroke led analysis, this process begins at the semantic layer level. The graph that contained the deleted wall is updated, and the new cycles (combination of the deleted rooms that were connected by
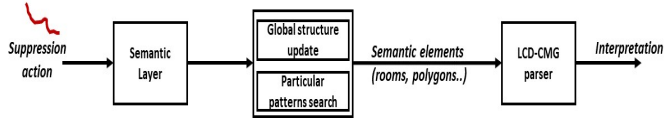
Fig. 11: Suppression triggers new analysis

that wall) are extracted. Finally, they are fed to LCD-CMG parser for further analysis. This optimization has a big impact in terms of analysis: the eager interpretation is in compliance with the real-time interaction constraint. We will evaluate the impact of the integration of the new semantic context analyser, as well as the later optimization, in the next section.

IV. EXPERIMENTS AND RESULTS

Our contribution consists in decreasing the analysis process complexity in order to enable real-time user interaction. Therefore, we will not present results in terms of shape recognition in this section. The base formalism has proven its performance in this aspect. For more details, see [1]. We present the impact of our extension in terms of analysis time, on two domains: architecture plans design, and geometric figures production for e-learning. The experiments where realized on a windows 64bits, Intel i7 at 2.90 GHz with 16 GB RAM. We will compare three approaches: **LCD-CMG base** refers to the basic formalism, **LCD-CMG SCA** refers to the extension with the Semantic Context Analyser, **LCD-CMG SCA + Opt** refers to the suppression triggered analysis optimization.

A. Architecture plans design

The scene (Fig. 12) illustrates a drawn stroke that will lead to an analysis process. TABLE I illustrates the impact of the extension and the optimization. As we explained earlier, for the LCD-CMG formalism, n iterations (reductions) are needed in order to create n rooms. The number 788 refers to the creation of 779 implicit and explicit rooms, plus the production rules belonging to other layers. The extension of the formalism parser with the semantic context analyser improves greatly analysis performance. The number of iterations becomes 12, and the analysis time goes from nearly two minutes to 8 seconds. The number of triggered rules also decreases heavily, from 5034 to 1356. However, in terms of real-time interaction, 8 seconds is still too much in the context of designing plans on the fly. We can see that the final optimization (minimal
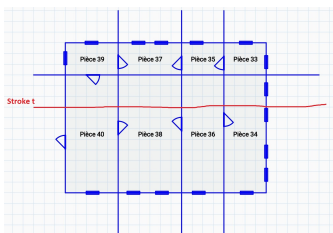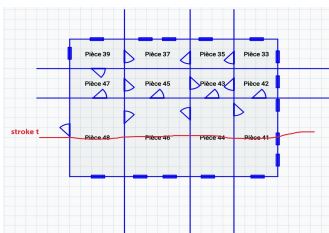

Fig. 12: First scenario   Fig. 13: Second scenario

TABLE I: First scenario analysis results (based on Fig. 12)

Approach	Iterations	Time	Triggered
LCD-CMG base	788	1min: 54s	5034
LCD-CMG SCA	12	8s	1356
LCD-CMG SCA + opt	12	0.35s	407

TABLE II: Second scenario analysis results (based on Fig. 13)

Approach	Iterations	Time	Triggered
LCD-CMG base	6442	5h 0min: 03s	305151
LCD-CMG SCA	12	2min: 58s	6987
LCD-CMG SCA + opt	12	0.37s	544

cycles and suppression triggered analysis), enables the system to have an acceptable performance (0.35s of analysis time, 407 triggered rules). In Fig. 13, we illustrate a scenario that demonstrates the combinatorial explosion when the document becomes complex. We study the analysis results in TABLE II.

While we could accept that in some extremely rare cases, the analysis time takes one minute or so, we can see here that a pen-based system based on LCD-CMG is unusable. The analysis time here is in the order of 5 hours with 305151 triggered in order to create 6429 polygons. The need for the extension and optimization is obvious in this case. The parser extension reduces greatly the time to 3 minutes, and the suppression triggered analysis optimization renders the system performance acceptable. Let's note that a pen-based system based on this work has been commercialized and is now being used by housing technicians for on the fly architecture plans design.

B. Geometric figures composition

As stated in the introduction, the main goal of our work is to take advantage of the introduction of pen-based tablets in middle-schools for geometry learning support. The target population here are teachers and pupils, so the digital support should be transparent to the user. Since we simulate the pen and paper traditional approach, there is no need for a tedious training phase for the pupil to master the system. Since we demonstrated in the last subsection that a standard LCD-CMG based system is not usable in practice, we focus here on the necessity of the suppression triggered analysis optimization. To provide prompt feedback, our system should be able to interpret the hand-drawn figures in less than one second. Fig. 14 illustrates the composition of a complex geometric figure. The difference with the architecture domain is that the polygons created by the semantic context analyser are then fed to the parser and can be reduced into all sorts of shapes: trapezoids, equilateral triangles, pentagons...

TABLE III: Third scenario analysis results (based on Fig. 14)

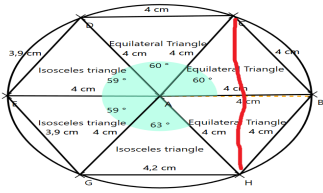| Approach	Iterations	Time	Triggered
LCD-CMG SCA	15	1.26s	515
LCD-CMG SCA + opt	9	0.37s	157

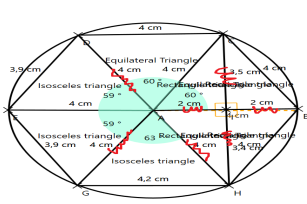Fig. 14: Complex figure composition, new stroke in red
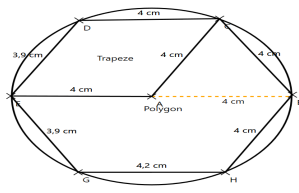

Fig. 15: Segments suppression

Fig. 16: Suppression analysis result

We can see from TABLE III that the optimization is needed with the extension to cope with the real-time analysis constraint. The decreasing number of iterations (from 9 to 15) is due to the fact that less figures are created by the semantic context analyser. However, as stated before, the semantics of the figure are not lost. We can see in fig. 15 and 16 that the suppression action of the user leads to the creation of a polygon, and a trapezoid. The real-time user drawings interpretation and the pen and touch interactions open up a lot of possibilities. Using planning techniques, we can analyse dynamically the pupil resolution state and generate guidance and correction feedback that is relevant and adapted to the learner [14]. An e-learning tutor based on the extended LCD-CMG formalism was tested in pilot middle-schools. Positive results in terms of learning transferability from the digital support to the traditional one will be discussed in future works.

V. Conclusion and perspectives

In this paper, we present an extension to the LCD-CMG parser in order to control the analysis complexity of the eager interpretation of structured documents. The global contextual knowledge is externalized from the formalism and is managed by a semantic context analyser. The later is responsible for the update of the global document structure, represented by semantic graphs that model the relations between the elements in the documents. Externalizing the search of particular patterns from the prior knowledge definition to the semantic context analyser has two objectives. The first one is to facilitate the manipulation and adaptation of the LCD-CMG grammar by users/designers. The second one is to overcome the parser limitation in terms of finding these complex patterns (n iterations needed to create n rooms). The experiments show the positive impact on two different application domains, architecture plans design and geometric figures construction. Our future works consist in diversifying the possible patterns that a designer can specify in the formalism. Cycles are sufficient for geometry and architecture plans, or electricity grids design, but the concept could be generalized to other structured documents, such as chemistry for example. The idea is that the user would draw his/her pattern, the system would learn it and integrate it in its knowledge base of particular patterns. The recognition of these learned patterns would be reformulated as a graph matching problem. As mentioned in the last section, an e-learning geometry tutor has been developed to assist pupils in their production. Other future works consists in trying to generalize this tutoring concept to other e-learning or professional domains.

References

[1] S. Macé, E. Anquetil, "Eager interpretation of on-line hand-drawn structured documents: The DALI methodology", Pattern Recognition 42, 2009, pp. 3202–3214.

[2] A. Delaye, E. Anquetil. "Hbf49 feature set : A first unified baseline for online symbol recognition", Pattern Recognition 46(1), 2013, pp. 117-130.

[3] R. Zanibbi, H. Mouchère, C. Viard-Gaudin. "Evaluating structural pattern recognition for handwritten math via primitive label graphs", Document Recognition and Retrieval XX, 2013, pp. 1–11.

[4] T. Hammond, R.Davis, "LADDER, A Sketching Language for User Interface Developers", Computers & Graphics 29, 2005, pp. 518-532.

[5] O. Atilola, S. Valentine, H.H. Kim, D. Turner, E. McTigue, T. Hammond, J. Linsey, "Mechanix : A natural sketch interface tool for teaching truss analysis and free-body diagrams", Artificial Intelligence for Engineering Design, Analysis and Manufacturing 28, 2014, pp. 169-192.

[6] Á. Muñoz, F.S. Peiró, J.B. Ruiz, "Recognition of on-line handwritten mathematical expressions using 2D stochastic context-free grammars 48 and hidden Markov models", Pattern Recognition Letters 35, 2014, pp. 58-67.

[7] K. Marriott. "Constraint multiset grammars". In Proceedings of the IEEE Symposium on Visual Languages, (VL'94), 1994, pp. 118–125.

[8] T. Y. Ouyang, R. Davis. "Recognition of Hand Drawn Chemical Diagrams". In Proceedings of the 22Nd National Conference on Artificial Intelligence, 2007, pp. 846–851.

[9] G. Costagliola, V. Vincenzo, M. Risi. "A multi-layer parsing strategy for on-line recognition of hand-drawn diagrams". In Proceedings of the Visual Languages and Human-Centric Computing (VL/HCC'06), Brighton, UK, 4–8 September 2006, pp. 103–110.

[10] F.D. Julca-Aguilar, N. S. T. Hirata. "Symbol detection in online handwritten graph-ics using faster R-CNN". In Proceedings of the 13th International Workshop on Document Analysis Systems, 2018, pp. 151–156.

[11] R. K. Sarvadevabhatla, J. Kundu, and R. Venkatesh Babu."Enabling My Robot To Play Pictionary: Recurrent Neural Networks For Sketch Recognition". In Proceedings of the ACMMM, 2016, pp. 247–251.

[12] F. Julca-Aguilar, H. Mouchère, C. Viard-Gaudin, N. Hirata. "A General Framework for the Recognition of Online Handwritten Graphics". International Journal on Document Analysis and Recognition (IJDAR), 2017.

[13] O. Krichen, N. Girard, E. Anquetil, S. Corbillé and M. Renault, "Real-Time Interpretation of Hand-Drawn Sketches with Extended Hierarchical bi-Dimensional Grammar". 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), Niagara Falls, NY, 2018, pp. 273-278.

[14] O. Krichen, E. Anquetil, N. Girard. "IntuiGeo: Interactive tutor for online geometry problems resolution on pen-based tablets". European Conference on Artificial Intelligence (ECAI) 2020, In Press, Aug 2020, Santiago de compostela, Spain.