



Real-time interpretation of hand-drawn sketches with extended hierarchical bi-dimensional grammar

Omar Krichen, Nathalie Girard, Eric Anquetil, Simon Corbillé, Mickaël Renault

► To cite this version:

Omar Krichen, Nathalie Girard, Eric Anquetil, Simon Corbillé, Mickaël Renault. Real-time interpretation of hand-drawn sketches with extended hierarchical bi-dimensional grammar. The 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), Aug 2018, NIAGARA FALLS, United States. pp.273-278. hal-01917888

HAL Id: hal-01917888

<https://inria.hal.science/hal-01917888v1>

Submitted on 9 Nov 2018

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.

Real-time interpretation of hand-drawn sketches with extended hierarchical bi-dimensional grammar

Omar Krichen, Nathalie Girard, Eric Anquetil, Simon Corbillé and Mickaël Renault

Univ Rennes, CNRS, IRISA

F-35000 Rennes, France

Email: {firstname.lastname}@irisa.fr

Abstract—This paper presents a generic method for eager interpretation of online hand-drawn sketches. This work takes place within the ACTIF project, with the objective of designing an e-learning system for geometry lessons in middle-schools. The goal is to analyze on the fly the child's inputs (using a pen-based interaction on tablet), and to give real-time feedback (visual, corrective, or guidance ones). We are therefore faced with the constraint of real-time strokes interpretation. We propose to use an incremental analysis process enabling to decrease the search space size. Nevertheless, we encounter combinatorial problems given the application field is complex. For example, create sub-figures from an existing one means the parser has to consider all possible combinations of connected segments that are contained in the original shape. To tackle this issue, we extend the definition and the parsing of bi-dimensional grammars by formalizing a hierarchy between production rules. Therefore, we propose alternative exploration strategies and reduce the search space of applicable production rules at any given level of the analysis. Thus, the parser is able to dynamically switch between the exploration strategies in the analysis process. The system has been tested with different drawing scenarios of geometric figures sketching. It reached interesting real-time performance, without losing the generic aspect of the approach.

Index Terms—Online recognition of handwriting, Bi-dimensional structured document interpretation, Pen-based interaction, Digital learning

I. INTRODUCTION

This work is in the context of ACTIF, an innovative project that aims to promote active learning in middle schools. The objective is to design an intelligent tutoring system for geometry lessons, based on a stylus and thumb interaction on tablets. The popular tools used in middle schools, such as Geogebra [12], which belong to the family of Dynamic Geometry Software, mostly rely on a drag-and-drop approach, *i.e.* the child has to choose geometric objects from a graphical panel and then place their components on the drawing zone. One limitation is that these software require a training phase for the pupils since their approach is different from the traditional pen and paper setup. Another limitation is that dragging and dropping tend to limit the fluidity of the drawing process, and therefore the creative liberty of the user. In [9], Fiorella and Mayer demonstrate that "generative drawing", *i.e.* learning by drawing, has a positive impact on students learning abilities. This paper presents a new pen-based interactive system for geometry learning. To overcome the limitations explained before, our approach simulates the traditional pen and paper figure sketching which allows to avoid the training phase.

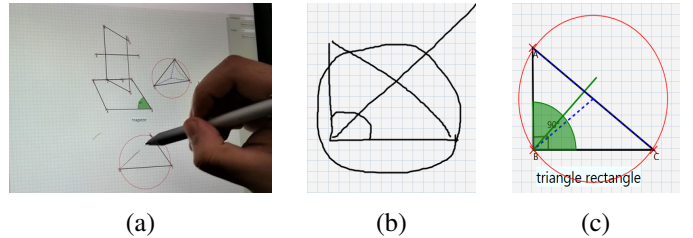


Fig. 1: Overview of the system: Prototype interface (a), handwritten figure (b), interpreted and beautified figure (c)

Moreover, our system recognizes on the fly the child's productions, which gives them more creative liberty (see Fig. 1.a). More broadly, beyond the geometry problematic, our objective is real time recognition of handwritten sketches for different types of structured documents. That is why **we propose a generic method for eager interpretation of online hand-drawn strokes**. We distinguish between two types of handwritten documents interpretations methods: lazy [7] and eager [8]. *Lazy interpretation* consists in analyzing the user's drawing after its completion. *Eager interpretation* consists in analyzing the handwritten strokes on the fly, which is more suited to our goal of having real-time feedback interventions to detect the child's errors, if any, and to guide him to answer the teacher's instruction. There are two major approaches for symbol recognition: statistical and structural. *Statistical approaches* [2] rely on classifiers and are well suited for isolated shape recognition. However, they cannot model the document structure. Since we are in Geometry context, the system has to recognize not only the geometric shapes, but also to model the spatial relations between them. *Structural approaches* model a symbol as a set of graphic primitives with determined structural relations. For instance, a triangle is considered as three segments linked by spacial relations. In this context, several works have been done in mathematical expressions recognition. For examples, in [3] labelled graphs are used for recognition, while in [6] a bi-dimensional extension to the Stochastic Context Free Grammar is used for mathematical expressions analysis. Another interesting approach is Ladder [4], a generic description language, which was applied in a digital learning context for truss diagrams interpretation [5]. Structural approaches main limitation is the difficulty to include uncertainty, which is the advantage of the statistical approaches. In this work, to model the geometry knowledge,

we use Context Driven Constraints Multi-set Grammar (CD-CMG) [1], a generic formalism for eager interpretation of hand-drawn documents. Actually, the advantage of CD-CMG is that this formalism is **a combination between a statistical approach** (to locally recognize a shape) **and a structural approach** (to model the spacial relations and the global document structure). The methodology is based on the compilation of the formalized knowledge in CD-CMG, the generation of the associated symbol recognition engines, and the exploitation of the compiled knowledge with a generic parser. CD-CMG has been applied on various types of documents such as architectural plans [7] or electrical sketches [1]. These application fields are close to geometric field, so this approach is well suited for our objective. However, for the creation of more complex sketches, the existent version of CD-CMG could not cope with the constraint of real-time analysis. **To tackle this issue, our contribution consists in extending this formalism by defining a hierarchy between different layers of interpretation, which allows to have alternative exploration strategies in order to reduce the search space size.** The paper is organized as follows. The formalism and the knowledge is explained in Sec. II. Sec. III describes the formalism extension towards a hierarchical grammar to cope with the constraint of real-time analysis. Experiments and results are presented in Sec. IV. Conclusion and future works are given in Sec. V.

II. KNOWLEDGE MODELLING

This section presents the CD-CMG formalism, and illustrates it through its adaptation to the Geometry field.

A. Context Driven Constraints Multiset Grammar

CD-CMG is formally defined as follows:

Definition 1. A CD-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, here $V_T = \{\text{stroke}\}$;
- S : the first symbol, or axiom;
- P : the set of production rules.

And where a production rule $p \in P$ is denoted as follows:

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

Preconditions and postconditions are based on the concept of *Document Structural Context* (DSC) defined hereafter, and which models a zone in the document and the awaited elements in it.

Definition 2. A DSC is defined by $(\lambda)[\text{position}](\gamma)[\text{part}]$ 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 symbol that has to intersect the zone.

Triangle: $\text{res} \rightarrow \text{segment: s1, segment: s2, segment: s3}$ where:

Preconditions:

(s1) [TotalLengthSegment] (s2) (s3) [extremity] &
(s2) [TotalLengthSegment] (s1) (s3) [extremity] &
(s3) [TotalLengthSegment] (s1) (s2) [extremity]

Constraints:

LinkedSegments(s1, s2, s3)

Postconditions

(res)[TriangleZone] (Circle: c) [All] \Rightarrow [Inscribed-Circle \rightarrow c]

Fig. 2: Triangle production rule

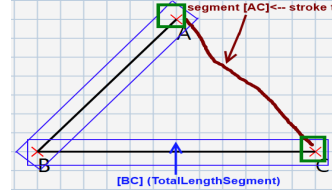


Fig. 3: Preconditions role

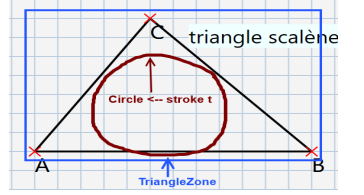


Fig. 4: Postconditions role

The **postconditions** are a set of DSCs created after the reduction of the production, to update the document's structure. This formalization enables to *drive the analysis process by the context*. Indeed, the preconditions represent the **verification step** while the postconditions represent the **prediction step**. The **constraints** model a local vision on the analyzed elements β . They have two purposes: checking that the shape of β is consistent with the production, and decide if it is pertinent to reduce β into α . Fig. 2 presents a production rule example in the Geometry field, i.e. the triangle production rule. Fig. 3 and Fig. 4 illustrate the roles of the preconditions and postconditions blocks. In Fig. 3, a new stroke is drawn, with its extremities linked with two contact zones (in green in Fig. 3). This will result in reducing this stroke to a new segment [AC] and creating its postcondition DSCs. The fact that at least one postcondition DSC relative to triangle creation is activated (blue zones in Fig. 3 + new segment DSC) will trigger the triangle production rule. This is the **prediction step**. The system will then check that all the precondition DSC are satisfied for the production, to verify that the context is coherent for triangle creation. This is the **verification step**. The constraints block (red rectangle in Fig. 2) contains a structural constraint which verifies that the segments intersect in three different points. Note that it is also possible to call a shape recognizer in the constraints block. The postcondition DSC (blue zone in Fig. 3) models the fact that an inscribed circle production will be triggered if all the points of a circle c are located in *TriangleZone*.

B. Existent parser diagnostic

The analysis process, extensively explained in [1], is a combination of a bottom-up strategy (guided by the reduced elements) and a top-down strategy (guided by the postcondi-

tions DSCs). For each new element, the parser searches the DSC it satisfies and vice versa. Consequently, a production is triggered if its β elements contain at least a new element and its precondition block contains at least a new DSC. Fig. 5 illustrates a scene composed of a new stroke t and 3 existing segments. The analysis of t leads to the construction of the analysis (or derivation) tree, presented in Fig. 6. The root represents the stroke t . The nodes and the leaves represent the triggered rules, while the blue dotted path is the sequence of reduced production rules, *i.e* the analysis result.

For the decision making process (see detail in [1]), each

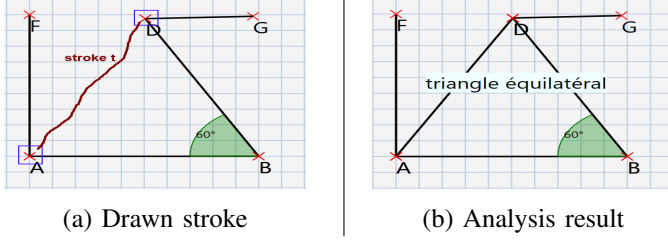


Fig. 5: Analysis process

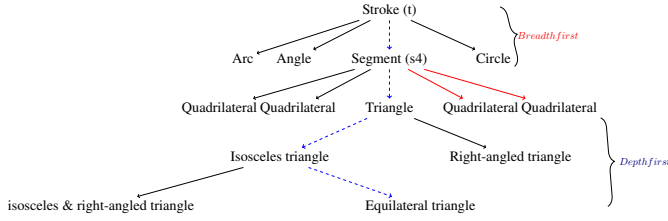


Fig. 6: Stroke analysis tree of the drawn stroke t in Fig. 5 (a)

production rule is associated to a score which depends on the context coherence and constraint validity. The score of a production, denoted ρ_p and defined in eq. 1, is a fuzzy combination between the membership degree of its preconditions ($p^{\text{preconditions}}$ for context coherence), and the membership degree of its constraints ($p^{\text{constraints}}$ for constraint validity). A production is applicable if its score is higher than a given threshold.

$$\rho_p = p^{\text{constraints}} \cdot p^{\text{preconditions}} \quad (1)$$

The score of an hypothesis (or branch), which is a sequence of productions, is the fuzzy combination of the involved production scores. The exploration strategy (*breadth-first* or *depth-first*) is also explicit in CD-CMG and specific for each production rule. By default, it is breadth-first. The *FirstRule* operator, set on a rule for depth-first exploration, allows dynamic switching between strategies during the analysis process. For *depth-first* exploration, the parser will reduce the first applicable production rule and stop the search. The resulting elements and DSCs (from the production reduction) will be the starting points of the analysis rerun. For *breadth-first* strategy, the parser explores all the possible interpretations and chooses the hypothesis with the highest score. As shown in Fig. 6, for the production rules that constitute an ambiguity

risk, such as segment and arc, we adopt a breadth-first strategy. For polygons rules we adopt a depth-first strategy: once a production is applicable, it is reduced (the red paths in Fig. 6 are productions that are not tested because of the type of exploration). We establish a natural hierarchy between the productions, *e.g* a triangle can be reduced into isosceles triangle.

C. Limits of CD-CMG

As we can see in Fig. 6, the quadrilateral production rule is triggered even if there is no coherent context for creating a quadrilateral in this scene. This is due to the fact that this production contains a new element (AD) and one of their precondition DSCs is satisfied. The impact on the combinatorics is not important here, but when the document is complex, the analysis becomes costly. Even though the formalism is generic and expressive enough to model the prior geometry knowledge, the multiple possible interactions between geometric objects, *e.g* creating sub-figures from existing ones, also generate combinatorial problems in the analysis process. Plus, one specificity of this field is the fact that there are multiple layers of interpretation: the basic symbol recognition layer, detecting symbol connections (such as intersections or segment divisions), and another semantic layer which recognizes geometric figures such as triangles or quadrilaterals. With CD-CMG, it is only possible to define two layers of interpretations, such that one layer stands for breadth-first type rules, and the other stands for depth-first type rule. This is not sufficient to control properly the analysis complexity. To tackle this issue, we extend the formalism and modify its associated parser such that it includes different layers of interpretation by introducing the concept of hierarchy between interpretation layers.

III. TOWARDS A HIERARCHICAL GRAMMAR

This section introduces the complexity issue of the analysis process and our proposed solution, establishing different levels of hierarchy between production rules and precondition contexts, to control the search space size for the analysis process.

A. Context hierarchization

There can be multiple coherent contexts to reduce a production rule, as shown in Fig. 7, which illustrates the segment creation production rule. The precondition block is a dis-junction of three preconditions, modeling the fact a stroke can be linked to two segments - ①, one segment - ②, or to nothing - ③. We can see from Fig. 5 that the three preconditions are valid for segment creation. The context search space is similar to the rule search space. This means that the parser will consider these preconditions as three *independent segment production rules*, expand the tree in a breadth-first fashion, and choose the one with the highest score. The existent *FirstPrecondition* operator is similar to the *FirstRule* operator (*c.f* Sec. II), such that it specifies that the exploration strategy for context search is depth-first, *i.e* the parser will choose the first valid precondition without exploring the others (see Fig. 8). Within the chosen precondition, the search space is constituted of the elements that satisfy the specified precondition. In this

Segment: $res \rightarrow stroke\ t$ with:
Preconditions: *FirstPrecondition* *FirstContext*
 @ (Segment:S1) [Zone] (t) [first] & (Segment:S2) [Zone] (t) [last]
 or ① (Segment: S1) [Zone] (t) [extremity]
 or ② (Document) [in] (t) [all]
Constraints:
 Recognizer(t, Segment)
Postconditions:
 (res)[LengthSegment] (Stroke: t) [extremity] \Rightarrow [angle \rightarrow t]
 (res)[TotalLengthSegment] (Segment: s1, s2) [extremity] \Rightarrow [Triangle \rightarrow res, s1, s2]

Fig. 7: Part of segment production rule

example, this search space is the couples of segments (S1, S2) that are linked to the stroke t by their extremities. A limit to the existent formalism is that there is no alternative exploration strategy in this second layer of the context search. In this example, within the first valid precondition - @, t can be linked to the couples (AB, BD), (AB, DG), (AF, BD) or (AF, DG) (c.f. Fig. 5). Since there is only one exploration strategy in this context search layer (breadth-first), the parser will consider these four hypotheses as independent segment production rules, therefore multiplying the analysis complexity by four. In order to have coherent strategies at all levels of the search (at production level, at precondition level, and context level), we extend the formalism by defining a new operator **FirstContext**. This operator allows a depth-first search exploration strategy for context research (c.f. Fig. 9) within the same precondition. Therefore, the combination of *FirstPrecondition* and **FirstContext** forces the parser to stop the context search at the first valid hypothesis within the first valid precondition (i.e. with (AB, BD) in the example). The impact on the analysis process is illustrated in Fig. 9. **This will drastically reduce complexity, without losing information about connections.** In rare cases, choosing the first context,

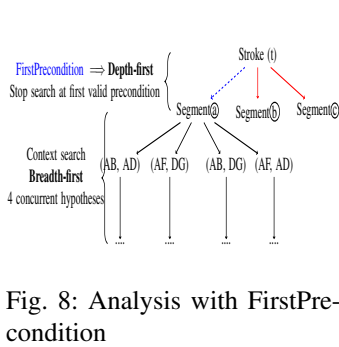


Fig. 8: Analysis with FirstPrecondition

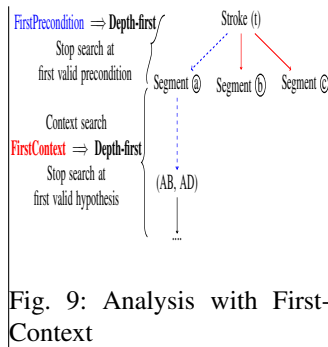


Fig. 9: Analysis with FirstContext

which is not necessarily the best, can generate errors. This limit is compensated by the fact that the user can either implicitly validate the interpretation, or delete the last stroke and redraw.

B. Production hierarchization

The existent interpretation logic considers that there are two strategies for two types of production rules: the first type of rules is for basic symbol recognition, and the corresponding strategy is breadth-first to ensure a valid interpretation.

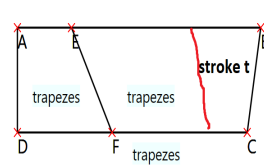


Fig. 10: Drawn stroke

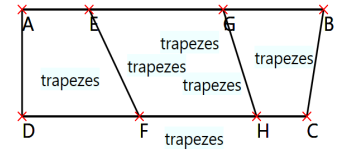


Fig. 11: Result: 6 segments, 3 trapezoids

The second type is low risk productions, such as segment intersections or edition rules, and the corresponding strategy is depth-first, to reduce the analysis complexity. Depth-first type production rules are prioritized, such that parser switches to breadth-first exploration if there is no applicable low risk production. This logic is valid when the application field is simple, e.g. having a breadth-first strategy for graphical primitives recognition and a depth-first strategy for symbols connections. However, when considering other semantic layers, such as figure and sub-figure detection, we face the existent logic limits. Indeed, we consider polygons creation as low-risk production rules, therefore setting the *FirstRule* operator on these productions to specify the type of exploration is depth-first. The problem is that the parser can take time before finding the valid production to reduce, since the production rule search space size increases as the application field becomes more complex. Therefore, depth-first strategy for low risk productions is not sufficient to cope with real-time analysis. **To solve this issue, we extend the CD-CMG grammar in order to include the notion of hierarchy between production rules, defining as much hierarchy levels as there are semantic interpretation layers for the application field.** We define a new operator **RuleLayer**, such that for each production, it will specify the corresponding semantic layer. This means that at a particular level of the analysis tree, the parser will only trigger productions belonging to the same hierarchy level. Let's explain the impact of this extension for the geometry field. We establish three hierarchy layers in this case:

- **RuleLayer=1: {"Classic" productions}**: to recognize a basic shape, such as segment, arc, angle or a circle. The associated strategy is breadth-first, to avoid invalid shape recognition (ambiguity between segment and arc, for example);
- **RuleLayer=2: {Connection productions}**: when the intersection between segments generates sub-segments, hence updating the document structure. The associated exploration strategy is depth-first.
- **RuleLayer=3: {Polygons productions}**: for figure and sub-figure detection. The associated strategy here is depth-first.

Fig. 10 and Fig. 11 illustrate the analysis of a drawn stroke t . At the stroke level, only **classic productions** are explored in breadth-first manner. This stroke interpretation as a segment led to the creation of 6 sub-segments by the division of [AB], [DC], [EB], and [FC]. Three new trapezoids were also created due to the new connections generated by these segments. It

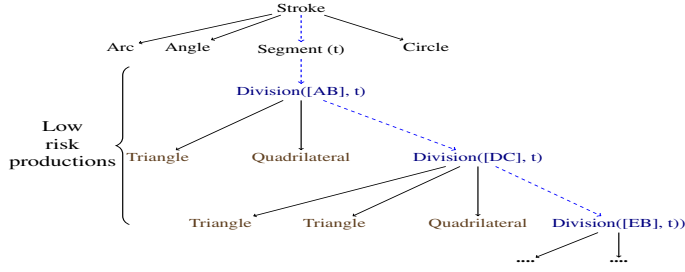


Fig. 12: Part of analysis tree without hierarchy

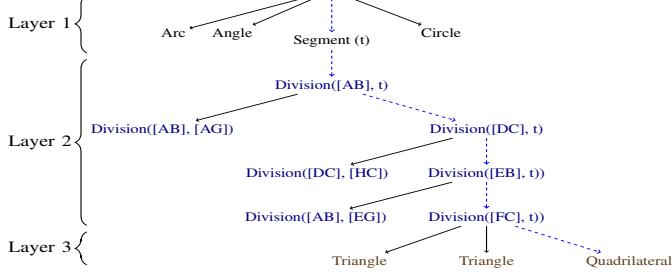


Fig. 13: Impact of hierarchy on the analysis tree

is clear then that the **polygon productions** depend strongly on the connections between the segments, therefore on the **connection productions** (here the division). Without our established hierarchy, the DSCs created by each division will trigger polygons rules, which adds unnecessary complexity to the analysis process, since the connections links between the segments are not completely updated yet. Since each segment creates 21 DSCs (to manage all possible interactions with other objects), and since the division does not replace the original segment, we are rapidly confronted with a combinatorial problem when the document is complex. The analysis tree without hierarchy is illustrated in Fig. 12. By establishing our hierarchy, only the productions belonging to the same layer will be triggered, therefore polygon productions are triggered only if there is no more applicable connections productions. The prioritization of connection rules will therefore reduce the analysis complexity by reducing the number of triggered rules. The hierarchization impact on the analysis tree is illustrated in Fig. 13. By integrating the presented operators, a production rule will be defined as follows:

$$(\text{RuleLayer}, \text{FirstRule}) \alpha \rightarrow \beta \left\{ \begin{array}{l} \text{Preconditions (FirstPrecondition, FirstContext)} \\ \text{Constraints} \\ \text{Postconditions} \end{array} \right\}$$

IV. EXPERIMENTS AND RESULTS

A. Quantitative study

To evaluate the impact of our contributions on the system performance, the following criteria are used:

- Iterations: number of reduced productions rules,
- Interpretations: number of branches in the analysis tree;
- Time: analysis time;
- Triggered: number of triggered rules.

The evaluation is done on two complex benchmarks, illustrated in Fig. 14, 15, and Fig. 16, 17. The performance is

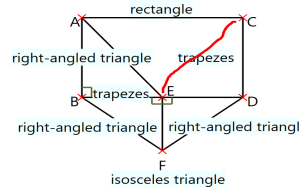


Fig. 14: Drawn stroke

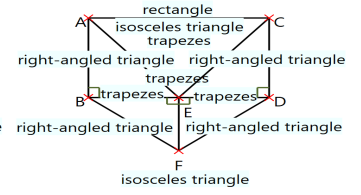


Fig. 15: Analysis result

compared with the existing approach based on CD-CMG [1]. The experiments were realized on a windows 64bits, Intel i7 at 2.90 GHz with 16 GB RAM.

B. Impact of context hierarchization

The scene (Fig. 14) illustrates a drawn stroke that will produce an analysis process. The stroke will be first interpreted as segment [DG]. This segment will trigger the creation of two trapezoids, an isosceles triangle and one right-angled triangle (Fig. 15). TABLE I presents the comparative results between CD-CMG and our formalism extension. The classic CD-CMG

TABLE I: First benchmark analysis results

| Approach | Iterations | Interpretations | Time | Triggered |
|-----------------------------------|------------|-----------------|--------|-----------|
| (1): CD-CMG+ <i>FirstRule</i> | 135 | 15 | 2.65 s | 2526 |
| (2): 1 + <i>FirstPrecondition</i> | 72 | 8 | 1.91 s | 1350 |
| (3): 2 + <i>FirstContext</i> | 9 | 1 | 0.35 s | 289 |

cannot cope with the constraint of real-time analysis (2.65 seconds). The addition of *FirstPrecondition* operator allows to control the complexity but the analysis time is still not transparent for real-time user interaction since it takes up to 1.9 seconds. This is due to the high number of triggered rules (1350), and the high number of iterations (72) which is caused by the computation of 8 equivalent interpretations at the context level. By switching the exploration strategy for context search to depth-first, *FirstContext* operator forces the parser to consider only the first valid interpretation, therefore reducing the analysis time down to 0.35 seconds.

C. Impact of rules hierarchization

The scene (Fig. 16) illustrates a drawn stroke that will produce an analysis process. The stroke will be first interpreted as segment [HI]. This segment will trigger 6 divisions which will create 8 new sub-segments. These new elements will trigger the creation of a right-angled triangle, two trapezoids, and one quadrilateral (Fig. 17). TABLE II presents the impact of our established hierarchy on the system performance. The prioritization of connection rules over polygon rules allows to

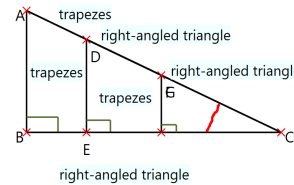


Fig. 16: Drawn stroke

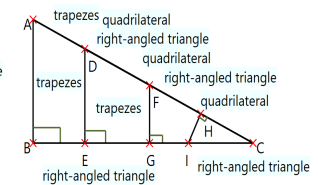


Fig. 17: Analysis results

TABLE II: Second benchmark analysis results

| Approach | Iterations | Interpretations | Time | Triggered |
|--------------------------------------|------------|-----------------|--------|-----------|
| 3 | 14 | 1 | 2.26 s | 1997 |
| 3 + three levels of hierarchy | 14 | 1 | 1.02 s | 1484 |

decrease the analysis time from 2.26s to 1.02s. The number of triggered productions decreases from 1997 to 1484. Therefore, we can say that our extension of CD-CMG formalism allows the design of a system with an acceptable performance given the constraint of real-time interaction. Beyond these improvements, the user interaction conditions have an important role in our system, the following section presents the characteristics of our prototype.

D. Qualitative study

1) *Edition mode*: In the traditional pen and paper setup, we often use symbols or codes to explicit geometric properties such as orthogonality or equality. We extend this notion by designing command gestures that not only display properties, but also modify physically the objects, with respect to these properties. These gestures are **integrated in the grammar** as strokes triggering production rules, therefore modifying the geometric objects. The orthogonality command gesture is illustrated in Fig. 18.

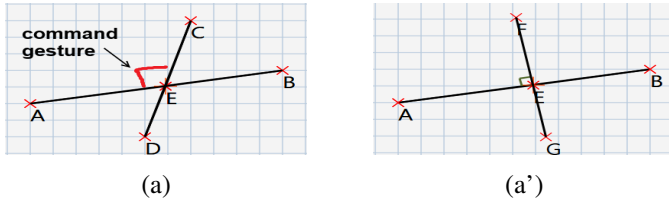


Fig. 18: Including geometry codes as command gestures: Orthogonality command (a), result (a')

2) *Virtual tools*: Pen interaction with the dominant hand is considered as better for actions that demand a high precision level, while Touch input is well suited for zooming, or object manipulation. Our **virtual ruler** is inspired from the one developed by Microsoft (e.g "Windows Maps" application). The user can freely move the virtual ruler on the drawing surface by dragging it with one or two fingers. The ruler will therefore follow the fingers movement (rotation, translation) (c.f Fig. 19). To manipulate the **compass**, the user first chooses a stationary point with the pen. Then, he sets the compass length, and finally he draws his arc on the virtual compass circle (c.f Fig. 20). These tools are going to be tested in pilot middle schools in order to evolve towards suited tools for the children.

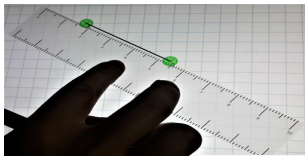


Fig. 19: Ruler tool

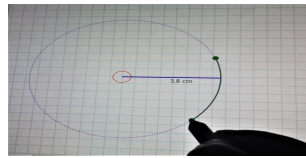


Fig. 20: Compass tool

V. CONCLUSION AND PERSPECTIVES

In this paper, we propose an e-learning system for geometry learning on tablets in middle-school. Our engine is based on the extension of CD-CMG formalism to analyze the user's productions in real-time. Adding the notion of hierarchy in the grammar to cope with the multiple interpretation layers had a significant impact on the system's performance, enabling to control the analysis complexity. Our future work consists in designing a teacher mode, where the instruction and all the possible solutions will be generated from the teacher's drawing. The objective is to detect the child's intentions and guide him through his realization of the exercise based on teacher's instruction. This will end in having a personalized teaching tool that will help make the child autonomous. The versions of this prototype will be tested in pilot middle schools and the works of our partners in ergonomics and cognitive science will help us to design well suited feedback to enhance learning performance.

ACKNOWLEDGMENT

"ACTIF" is funded by the French government project e-fran (BPI) and the region of Brittany. The authors would like to thank the Caisse des Dépôts/GIP-Far, as well as the academic partners, the LP3C and LOUSTIC laboratories.

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. Ruiz, 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] A. Ghorbel, A. Lemaître, E. Anquetil, S. Fleury, E. Jamet. "Interactive interpretation of structured documents : application to the recognition of handwritten architectural plans", *Pattern recognition*, 48(8), 2015, pp. 2446-2458.
- [8] E. Lank, J. Thorley, S. Chen, D. Blostein, "On-line recognition of UML diagrams". *Document Analysis and Recognition*, 2001. Proceedings. Sixth International Conference on Document Analysis and Recognition, 2001, pp. 356–360.
- [9] L. Fiorella, R.E Mayer. "Learning as a generative activity: Eight learning strategies that promote understanding". Cambridge University Press, 2015.
- [10] A. W. Chickering, S. C. Ehrmann. "Implementing the seven principles: Technology as lever". *AAHE Bulletin*, 49, 1996, pp. 3–6.
- [11] A. N. Kluger, A. DeNisi. "The effects of feedback interventions on performance: A historical review, a meta-analysis, and a preliminary feedback intervention theory". *Psychological Bulletin*, 119, 1996, pp. 254-284.
- [12] K. Bhagat, C-Y. Chang, "Incorporating GeoGebra into geometry learning – A lesson from India", *Eurasia Journal of Mathematics, Science & Technology Education*, 11 (1), 2015, pp. 77–86