Real-time interpretation of geometric shapes for digital learning

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

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

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

Mickaël Renault
Univ Rennes, CNRS, IRISA
F-35000 Rennes
mickael.renault@irisa.fr

***Abstract*—In the context of the *ACTIF* project that aims for active and collaborative learning promotion, this paper presents a pattern recognition and analysis system for Geometry learning in middle school. The goal is to allow students to draw geometric shapes on a touch-tablet, given a teacher's instruction. To make the student active, the system have to recognize and analyze on the fly the student's productions in order to produce real-time visual, corrective, and guidance feedback. We base our work on the visual grammar CD-CMG [1] (Context Driven Constraints Multi-set Grammar), to model the domain prior knowledge and interpret the hand-drawn sketches on the fly. Our first contribution lies in adapting this grammar to the Geometry domain to cover the geometric objects taught in middle school curriculum. Although being expressive enough to model this large scope, the formalism could not cope with the exigence of real-time analysis, given that the multiple interactions between geometric objects generate combinatorial issues. Our second contribution lies in extending the formalism which resulted in having an acceptable performance for a real-time user interaction system. The first experiments show that the proposed approach allows complexity and interpretation time reduction.**

***Index Terms*—On-line Recognition, Hand-drawn stroke analysis , Digital learning**

I. Introduction

Our work is in the context of *ACTIF* project, which aims to use pen-based tablets in an educational context, mainly in French middle schools, to foster active learning [14]. In this paper, we focus on learning Geometry by drawing freely on a touch-tablet. Dynamic Geometry Software products are now an important part of teaching geometry. Their goal is to make geometric concepts understanding easier for the student by graphical construction, manipulation and visualization of figures. To our knowledge, the tools used in middle schools, such as Geogebra [2], rely on a drag-and-drop approach to manipulate geometric objects. Indeed, in order to compose a figure, a student must choose from a graphical panel the object he wants to create then has to place its components in the interface. This tends to limit the creative process of the user. In [13], Fiorella and Mayer demonstrate that "generative drawing", *i.e* learning by drawing, has a positive impact on students learning abilities in the classroom. In [15], Kluger and DeNisi show the impact of feedback intervention on learning performance. These two points represent the pedagogical foundation of our project. We propose a pen-based system that simulates the traditional pen and paper figure sketching and enriches it by real-time visual, corrective and guidance feedback. This paper presents the first works done in the project, and focuses on the online recognition method of the system. In the literature several works have been done for hand-drawn sketches recognition, the following Sec. II presents an overview of existing approaches. Based on this overview and the application domain, we introduce the formalism and the modelling of the geometric knowledge in Sec. III. Since the system has to recognize hand-drawn sketches in real-time, some optimizations are needed, Sec. IV describes the formalism extension and its impact on the analysis process. Sec. V presents our experiments and results while conclusion and perspectives are given in Sec. VI.

II. Related works

In this work, we are interested in on-line recognition of handwritten structured documents. We distinguish between two types of handwritten documents interpretations methods: lazy [8] and eager [10]. **Lazy interpretation** means that the analysis process begins after completion of the user's production. **Eager interpretation** means that the handwritten strokes are analyzed on the fly, which is more relevant to our objective of having real-time corrective and guidance feedback to prevent error propagation. There are two major approaches for document analysis: statistical and structural. **Statistical approaches** [3] rely on learning on large labelled databases to discriminate between symbols and are well suited for isolated shapes recognition. However, they do not allow the modelling of the document's structure. Since we are in Geometry context, the system has to recognize not only the geometric shapes, but also the structural relations between the objects. **Structural approaches** consider a symbol in terms of its constituents, the graphical primitives, and the structural relations between them. For instance, a triangle is considered as three segments related by spacial relations. Structural approaches rely on modelling prior domain knowledge by visual grammars. We distinguish between two classes of structural recognition methods. The former is based on **graph grammars**. For example, Zannibi *et al.* [4] use labelled graphs to recognize handwritten mathematical expressions. One problem with graph-based methods is that they are complex to manipulate for the designer, especially if the production rules number is high. The latter is based on **bi-dimensional grammars**. For example, in [5] Hammond

and Davis proposed Ladder, a generic description language, and applied it for the interpretation of Truss diagrams in a digital learning context [6]. In [7], a bi-dimensional extension to the Stochastic Context Free Grammar is proposed for handwritten mathematical expressions analysis. In this work, to model the geometry domain knowledge, we prefer to use Context Driven Constraints Multi-set Grammar (**CD-CMG**) [1], a generic formalism for eager interpretation of hand drawn documents. Indeed, in this grammar, the context is explicitly specified in the production rules, which reduces the search space. Moreover, this formalism is the combination between a statistical approach (to locally recognize a shape) and a structural approach (to model the global structure of the document). Finally, CD-CMG has been applied on various types of documents such as architectural plans [11] or electrical sketches [1]. All these features show these formalism is well adapted for our purpose. Thereafter, our contribution lies in two folds: adapting this grammar to the Geometry domain for e-education (see Section III), and extending this formalism to match the constraint of real-time analysis of geometric productions (see Section IV).

III. Geometry Domain Modelling

In this section, we present the formalism, and illustrate it through its adaptation to the Geometry domain.

A. Context Driven Constraints Multi-Set Grammar

As an extension of the well-known grammar CMG [12], 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= $\{stroke\}$;*
- *S: the first symbol, or axiom;*
- *P: the set of production rules.*

And where a production rule $p \in P$ is composed of three blocks allowing different levels of vision on the document. The **precondition** and the **postcondition** blocks stand for the global vision of the document while the **constraint** block stands for the local vision of the analyzed strokes. Therefore, a production rule p is denoted as follows:

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

Preconditions and postconditions are based on the concept of *Document Structural Context*, which models a zone in the document and the awaited elements in it, defined as follows:

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 symbol that has to intersect the zone.*

The **preconditions** are a set of DSC that have to be satisfied and represent the context in which β can be replaced by α. The **postconditions** are a set of DSC that represent the objects that can be created from the new reduced elements α. 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

B. Adaptation of CD-CMG to Geometry

We consider the main geometric objects taught in French middle schools: segments, arcs, circles, angles, all the types of triangles and quadrilaterals. We defined around 20 productions rules to model these objects productions as well as the interactions between them (*e.g* intersection and orthogonality). Let's illustrate this with two production rules.
Fig. 1 presents a part of a **segment production rule**, while Fig. 2 and Fig. 3 illustrate a segment composition. In this example, the red stroke in Fig. 2 is transformed into a segment if the **precondition** block and the **constraint** block are satisfied. The **postcondition** DSC (green rectangle in Fig. 1) models the fact that a bisector production rule will be triggered if a straight d intersects the center zone of the new created segment (res).

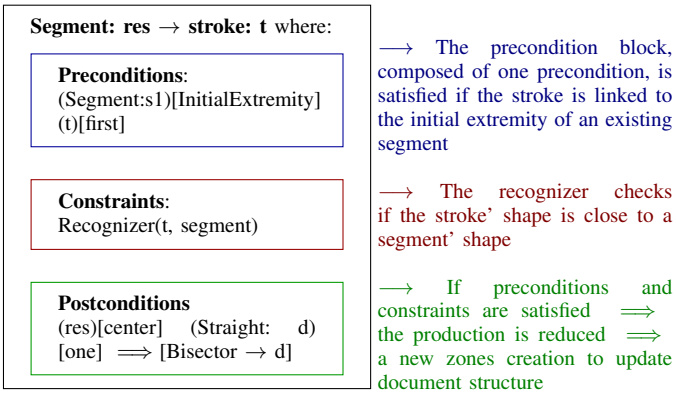


Fig. 1: Segment production rule in CD-CMG

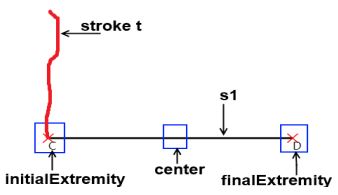


Fig. 2: Drawn stroke (in red)

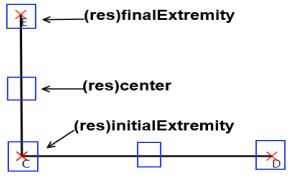


Fig. 3: Interpreted segment

Fig. 4 illustrates **the production rule of a scalene triangle**. The precondition block is composed of one precondition which is a conjunction of three DSC modelling the fact that each segment intersects a zone related to the other two. The constraint here is structural, such that it verifies that the three

Triangle: res → segment: s1, s2, s3 with:
Preconditions:
(S1) [Zone] (s2) [one] & (S2) [Zone] (s3) [one] & (S3) [Zone] (s1) [one]
Constraints:
LinkedSegments(s1, s2, s3)

Fig. 4: Triangle production rule

segments are linked by their extremities. Since the structural context is the same for all types of triangles (and quadrilaterals as well), **we established a hierarchy between production rules**, from general to specific, in order to prune the search space and speed up the analysis process. For example, a triangle can be reduced into an isosceles triangle if two of its sides are equal.

C. Analysis process associated to CD-CMG

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 postconditions 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. Let's consider the scene illustrated in Fig. 5 composed of a new stroke *t* (in red) and 3 segments (s1, s2, s3).

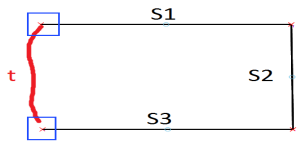


Fig. 5: Stroke analysis process

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 path is the sequence of reduced production rules, *i.e* the analysis result. As shown in Fig. 6, *t* is first reduced into a segment, denoted thereafter s4. Then, several rules are tested among which the production rules that led to the correct interpretation of the user's drawing.

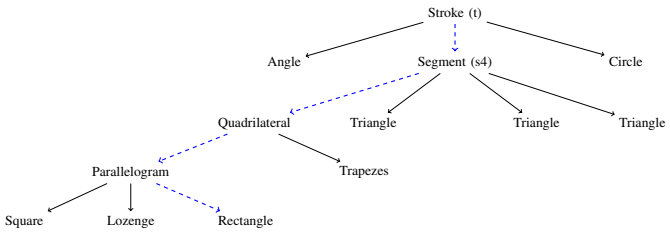
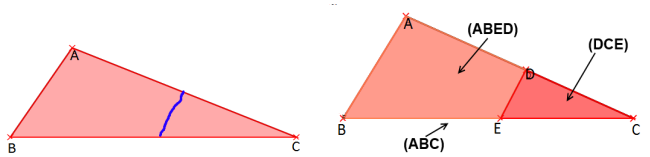


Fig. 6: Analysis tree

D. Limits of CD-CMG in geometry

As we can see in Fig. 6, the triangle production rule is triggered three times for each possible combination of segments: (s4, s1, s2), (s4, s1, s3) and (s4, s2, s3), even if there is no coherent context for creating a triangle in this scene. This is due to the fact that these productions contain a new element (s4) and one of their preconditions DSC is satisfied (*c.f* Fig. 5). Only the preconditions block is checked in this case, since not all DSC are 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 (*c.f* Fig. 7), also generate combinatorics problems in the analysis process.



(a) Drawn stroke (in blue)

(b) 2 Triangles and 1 quadrilateral

Fig. 7: Sub-figures creation

Let's consider the triangle production rule (*c.f* Fig. 4). A direct consequence of adapting CD-GMC to the geometry domain is that the β elements (here the 3 segments) are not really replaced by the α elements (here the triangle). They contribute to create the triangle but they remain considered in the analysis process in order to create other new elements. This has a big impact on the applicable rules search space size. We distinguish two factors producing the combinatorics explosions: **the format of the DSCs**, and **the computation of equivalent interpretations**. We will explicit these factors and our proposed solutions in the next section.

IV. Revision and Formalism Extension

In this section, we present the problems we faced in terms of analysis process complexity and our proposed solutions.

A. The DSC problematic

The expressivity of the formalism in terms of describing the document structure with the DSC allows to formalize that all the components of a polygon are linked by their extremities. Unfortunately, as we have seen in Section III.C, the fact that a production can be triggered even if only one of its preconditions DSC is validated generates a combinatory problem. Indeed, the more segments a polygon contains, the more DSC there are in the polygon production. Fig. 8 illustrates the composition of a new stroke t in the context of three already interpreted segments. t will be recognized as a new segment called thereafter s2. The fact that s2 is linked to [AB] (blue zone in Fig. 8) will activate the DSC:
[AB] [InitialExtremity] s2 [one] $\Longrightarrow$ *triangle* $\rightarrow$ *[AB],s2,s3.*

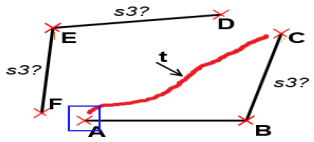


Fig. 8: Strict DSC

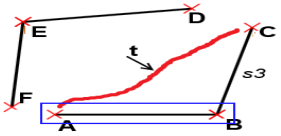


Fig. 9: Refined DSC

The parser will search the third segment (*i.e* s3) that completes the triangle rule with [AB] and s2. There is no contextual information in this DSC about the segment [BC] that completes the triangle since it is not concerned by the zone [AB] [InitialExtremity]. In consequence, for this scene composed of 3 segments besides [AB] and s2, the triangle production rule will be tested three times (for s3=[BC], s3=[ED] and s3=[EF]) instead of once. Thus, the analysis time can be very long, especially if the document is complex. In fact, this issue relies on a CD-CMG limitation. The formalism does not allow to have more than one zone in a DSC, which would enable positioning many awaited elements in relation to one reference element.

To resolve this problem, we propose to **refine the constraints on the zones** such that a zone can cover all the awaited symbols in the same DSC. The DSC related to the triangle production rules will then be:

[AB] [TotalLengthSegment] s2, s3 [one],

where $TotalLengthSegment$ is the zone that covers the length of [AB] (in blue in Fig. 9). This formulation allows to have a contextual information on all the segments composing a triangle. The loss in focus of the zone (from covering an extremity to covering all the segment) is balanced in the $Constraint$ block by verifying that the segments are structurally linked by their extremities. For the scene illustrated in Fig. 9, the triangle production rule is triggered only once, which reduces the analysis complexity.

B. Equivalent interpretations problematic and formalism extension

Since we are in Geometry learning context, it is important know the dependence links between the elements, *e.g* the connections between several segments. These links are modeled in the preconditions DSCs. Fig. 10 presents a focus on the precondition block of the segment production rule. It is composed of a dis-junction of three preconditions. They model the fact that there are three possible contexts for a segment creation. The stroke can be linked to two existing segments by their extremities, or linked to the extremity of only one segment, or not linked to anything (in the document).

The $FirstPrecondition$ operator, introduced in [11], establishes an order between the preconditions, *e.g* from specific to general, and forces the parser to stop the context research at the first valid precondition.

Segment: res $\rightarrow$ **stroke t** with:
Preconditions:
$FirstPrecondition$:
(Segment:S1) [Zone] (t) [first]& (Segment:S2) [Zone] (t) [last] **or**
(Segment: S1) [Zone] (t) [extremity] **or**
(Document) [in] (t) [all]
...

Fig. 10: Precondition block for segment production

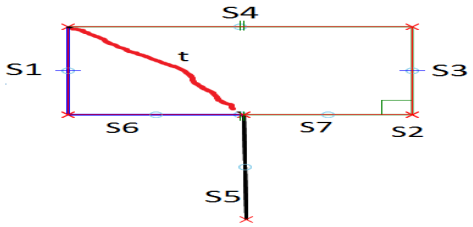


Fig. 11: FirstPrecondition limits

Fig.11 illustrates a scene in which a stroke t is recognized as a segment. $FirstPrecondition$ forces the parser to consider the stroke as linked to two segments, *considering* only the first precondition. Without this operator, the three preconditions, which are valid in this case, will be tested as hypotheses and the parser will choose the one with the highest score.

A limit to this operator is that the verification of the first precondition can also be complex. Indeed, in this example, there are six contextually valid hypotheses: t can be linked to the couples (s1, s5), (s1,s6), (s1, s7), (s4, s5), (s4, s6), (s4, s7). Hence, six equivalent branches will be created in the analysis tree multiplying the analysis complexity by six. To tackle this problem, we propose to extend the formalism by creating a new operator $FirstContext$. This operator forces the parser not only to stop the research at the first valid precondition, but also **at the first valid context** within a precondition. That means that the search is stopped when the first reference elements that are coherent with the precondition are found. In the example (Fig. 11), the parser will choose the first couple of segments that satisfies the DSCs of the preconditions, *e.g* it will choose the couple (s1, s7) without considering the other combinations. This will drastically reduce complexity, without losing information about connections. In use, we have noticed a limit to this new operator, which occurs when the segments are not exactly connected, but have overlapping zones (see the example in Fig. 12) . Without the $FirstContext$ operator, the system computes the membership degree of the stroke's extremity in each zone of the segments to choose the best possible interpretation. With $FirstContext$ operator, it has to

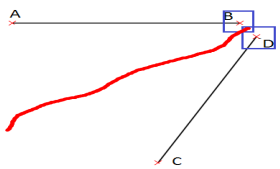


Fig. 12: Zones overlap problem

choose the first valid interpretation, which is not necessarily the best. However, the robustness of this extension lies in the interaction with the user since he has the possibility to implicitly validate the interpretation by continuing his drawings, or to delete the segment and redraw it more precisely. This is a trade-off between interpretation precision and analysis process. We will detail the impact of our contributions in the next section.

V. Experiments and results

A. Quantitative study

To evaluate the impact of our contributions on the system performance, we established several criteria:

- 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 realized on one complex drawing benchmark, illustrated in Fig. 13.

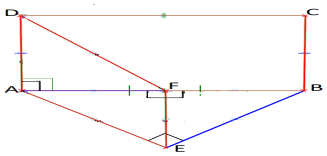
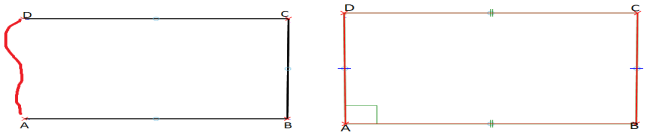


Fig. 13: Benchmark

We study the impact of our contributions on three critical steps of the drawing scenario of this figure, illustrated in Fig. 14, Fig. 15, and Fig. 16. We compare the performance of our system with **DALI**, the framework based on CD-CMG and its associated parser. In the following, the term *ZoneOpt* refers to the constraints refinement on the zones while $FirstContext$ refers to the formalism extension by the addition of the new operator.

1) First step of the scenario: The scene (Fig. 14) illustrates a drawn stroke that will produce an analysis process. The stroke will be first interpreted as segment [AD]. This segments will trigger the production of a rectangle.



Drawn stroke Analysis result

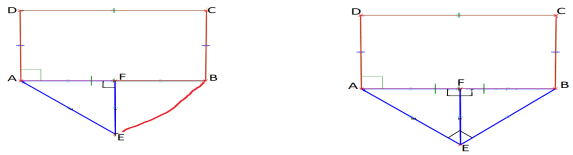
Fig. 14: First step of the scenario

TABLE I presents the analysis results for this step.
Since the document is still simple, the performance is good (around 0.20 seconds). However, our contributions have no real impact on the analysis time. With *ZoneOpt*, the number of triggered rules decreases from 43 to 23. $FirstContext$ operator has no effect since there is no equivalent interpretations to consider.

TABLE I: First step analysis result

Approach	Iterations	Interpretations	Time	Triggered
DALI	4	1	0.23 s	43
ZoneOpt	4	1	0.20 s	**23**
$FirstContext$	4	1	0.20 s	43
ZoneOpt+ $FirstContext$	4	1	0.19 s	23

2) Second step of the scenario: The scene (Fig. 15) illustrates a more complex production. The drawn stroke will be first interpreted as segment [EB]. This segments will trigger the production of two triangles, one being isosceles and the other rectangle.



Drawn stroke Analysis result

Fig. 15: Second step of the scenario

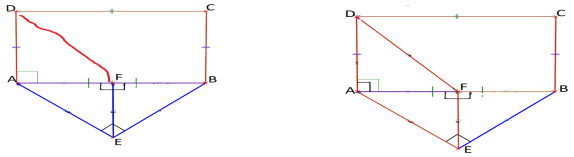
The impact of our contributions on this step is illustrated in TABLE II.

TABLE II: Second step analysis results

Approach	Iterations	Interpretations	Time	Triggered
DALI	15	3	6.4 s	1057
ZoneOpt	15	3	2 s	330
$FirstContext$	5	1	4.2 s	356
ZoneOpt + $FirstContext$	5	1	**0.62 s**	113

With the classic Dali methodology, the performance is not acceptable since analysis time takes 6.4 seconds. This is due to the number of triggered rules (1057) and equivalent interpretations (3). By modifying the format of the DSCs, *ZoneOpt* improves the analysis time (2s) by reducing the number of triggered rules. $FirstContext$ operator forces the parser to consider only one interpretation. In consequence, the number of triggered rules decreases from 1057 to 356, and the analysis time is down to 4.2 seconds, which is still not acceptable in a context of real-time interaction with a user. However, the coupling of Opt1 and Opt2 enables to have an analysis time of 0.62 which is acceptable.

3) Third step of the scenario: In this final step of the scenario (illustrated in Fig. 16), the drawn stroke will be first interpreted as segment [DF]. This segment will trigger the production of rectangle triangle, a trapezes and a parallelogram.



Drawn stroke Analysis result

Fig. 16: Third step of the scenario

For this step of the scenario, the coupling of *ZoneOpt* and $FirstContext$ enables to decrease the analysis time from 30 seconds to only 1.5 seconds. As we can see from TABLE II and TABLE III, the more complex the scene gets, the greater the impact of our contributions on the performance gets. Thus, taking into account the desired **real-time user interaction**, the proposed optimizations allow the design of a system with acceptable performance.

TABLE III: Third step analysis results

Approach	Interpretations	Analysis time	Triggered rules
CD-CMG	6	30 s	2472
Extended zones	6	7 s	962
$FirstContext$	1	5 s	412
Extended zones + $FirstContext$	**1**	**1.5 s**	**167**

B. 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.**??**. Our system allows also to modify the length of a segment by dragging one of its extremities. The same principle is applied for angle modification.

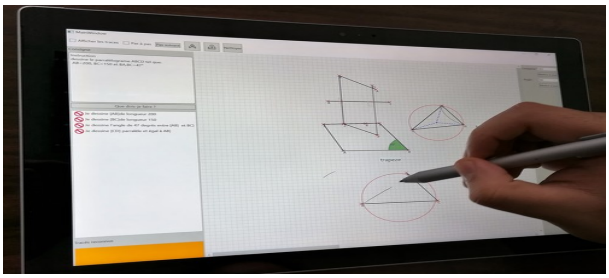


Fig. 17: Prototype interface

VI. Conclusion and perspectives

In this paper, we propose a pen-based system that interprets in real-time geometric figures in order to give visual and corrective feedback, given an instruction. We optimize and extend the CD-CMG formalism to adapt the DALI methodology to geometry. Our contributions have a consequent impact on the system performance which is now acceptable for real-time user-interaction. Our future work consist in improving even more the analysis time, one possible solution being the modification of the analysis process in terms of triggering rules. We will also work on an author mode, where the teacher can create customized exercises. The goal will be to generate automatically the solver procedure from the teacher's drawings. The system will also have to generate all the alternative procedures in order to give personalized corrective feedback to each student. The successive versions of our prototype will be tested in pilot middle schools, and we will benefit from the studies of LP3C and LOUSTIC laboratories in usage psychology and ergonomics to design visual and corrective feedbacks that are well suited for the students.

Acknowledgment

"ACTIF" project is funded by the French government and the region of Brittany. The authors would like to thank the academic partners and 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] 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
[3] A. Delaye, E. Anquetil. "Hbf49 feature set : A first unified baseline for online symbol recognition", Pattern Recognition 46(1), 2013, pp. 117-130.
[4] 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.
[5] T. Hammond, R.Davis, "LADDER, A Sketching Language for User Interface Developers", Computers & Graphics 29, 2005, pp. 518-532.
[6] 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.
[7] 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.
[8] 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.
[9] D. Py, "Environnements interactifs d'apprentissage et géométrie". Habilitation à diriger des recherches de l'université Rennes 1, 2001.
[10] 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.
[11] M. Pecot, S. Macé, E. Anquetil. "Interprétation interactive de plans d'architecture composés à main levée". Actes du XIème Colloque International Francophone sur l'Ecrit et le Document (CIFED'10), 2010, pp. 185-200.
[12] K. Marriott. "Constraint multiset grammars". In Proceedings of the IEEE Symposium on Visual Languages, (VL'94), 1994, pp. 118–125.
[13] L. Fiorella, R.E Mayer. "Learning as a generative activity: Eight learning strategies that promote understanding". Cambridge University Press, 2015.
[14] A. W. Chickering, S. C. Ehrmann. "Implementing the seven principles: Technology as lever". AAHE Bulletin, 49, 1996, pp. 3–6.
[15] 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.