

# Computer-Generated Geometry Proofs in a Learning Context



Pedro Quaresma and Vanda Santos

## 1 Introduction

Given its formal, logical, and spatial properties, geometry is well suited to teaching environments that include dynamic geometry systems (DGSs), geometry automated theorem provers (GATPs), and repositories of geometric problems. These tools enable students to explore existing knowledge in addition to creating new constructions and testing new conjectures. With the help of a DGS, students can visualise geometric objects and link the formal, axiomatic nature of geometry (e.g., Euclidean geometry) with its standard models and corresponding illustrations (e.g., the Cartesian model). With the help of GATPs, students can check the soundness of a construction (e.g., if two given lines are parallel) and also create formal proofs of geometric conjectures. Supported by repositories of geometric knowledge, these tools provide teachers and students with a framework and a large set of geometric constructions and conjectures for doing experiments.

In this chapter, we trace the evolution of current automatic proving technologies, how these technologies are beginning to be used by geometry practitioners in general to validate geometric conjectures and generate proofs with natural language and visual rendering, and foresee their evolution and applicability in an educational setting. Following Hanna's (2000, p. 8) argument that "the best proof is one that also helps understand the meaning of the theorem being proved: to see not only that it is true, but also why it is true," and the large number of articles on proof and proving in mathematics education from the ICMI Study 19 Conference (Lin, Hsieh, Hanna, & de Villiers, 2009a, 2009b), we focus our attention on practices of verification, explanation, and discovery in the teaching and learning of geometry.

In the classroom, the fundamental question a proof must address is "why?" In this context, then, it is only natural to view proofs first and foremost as explanations and, as

---

P. Quaresma (✉) · V. Santos  
University of Coimbra, 3001-501 Coimbra, Portugal  
e-mail: [pedro@mat.uc.pt](mailto:pedro@mat.uc.pt)

V. Santos  
e-mail: [vsantos7@gmail.com](mailto:vsantos7@gmail.com)

a consequence, to give more value to those that provide a better explanation. Dynamic geometry systems encourage both exploration and proof because they make it so easy to pose and test conjectures. The feature that preserves manipulations allows students to explore “visual proofs” of geometric conjectures. Such a powerful feature gives them strong evidence that a theorem is true and reinforces the value of exploration by giving them confidence in a theorem.

The challenge facing classroom teachers is how to use the excitement and enjoyment of exploration to motivate students while also explaining that visual exploration is not a proof. Visual exploration is a useful aid, but is still only the exploration of a finite number of cases. One reason for giving students a formal proof is that exploration does not reflect the need for rigour in mathematics. Indeed, mathematicians aspire to a degree of certainty that can only be achieved by a proof. A second reason is that students should come to understand the first reason. As most mathematics educators would agree, students need to be taught that exploration, useful as it may be in formulating and testing conjectures, does not constitute a proof (Hanna, 2000; Hanna & Sidoli, 2007). A proof is a means of obtaining certainty about the validity of a conjecture (proof as a validation tool) and a strategy to further understand a formulated conjecture (proof as an instrument of understanding).

Geometry automated theorem provers open the possibility of formally validating properties of geometric constructions. For example: *Cinderella*<sup>1</sup> (Richter-Gebert & Kortenkamp, 1999) has a randomised theorem checker; *Geometry Constructions LaTeX Converter (GCLC)*<sup>2</sup> (Janičić, 2006) and *GeoGebra*<sup>3</sup> (version 5) (Hohenwarter, 2002) incorporate a number of automated theorem provers that provide a formal answer to a given validation question (Botana et al., 2015; Janičić & Quaresma, 2007).

By means of the deductive database method (Ye, Chou, & Gao, 2010b), GATPs also enable students to explore new knowledge and discover new results and theorems (e.g., the algebraic formula of a loci (Abánades, Botana, Kovács, Recio, & Solyom-Gecse, 2016; Recio & Vélez, 2012)). An important addition to any learning environment would be a GATP with the ability to produce human readable formal proofs with, eventually, visual counterparts (Chou, Gao, & Zhang, 1996a, 1996b; Janičić, Narboux, & Quaresma, 2012; Kovács, 2015; Stojanović, Pavlović, & Janičić, 2011; Stojanović, Narboux, Bezem, & Janičić, 2014).

In this chapter, we focus on the subjects of verification and explanation. Section 2 describes the past, present, and foreseeable future of geometry automated theorem proving. Section 3 uses examples to introduce the use of GATPs in the formal validation of a conjecture. In Sect. 4, we explore GATPs that produce readable formal proofs. In Sect. 5, we extend this exploration to GATPs that produce formal proofs with the addition of natural language and visual rendering. In Sect. 6, we draw conclusions and anticipate future avenues of research.

---

<sup>1</sup><https://cinderella.de>.

<sup>2</sup><http://poincare.matf.bg.ac.rs/~janicic/gclc/>.

<sup>3</sup><https://www.geogebra.org/>.

## 2 Formal Geometric Proofs

Given the cognitive complexity of geometric activities, learning geometry is not always immediate or easy. However, visualisation can be used to understand proofs and help mathematical reasoning. Visual elements attract students' attention, raise their awareness, and help them make connections between subjects and concepts. Visualization contributes to a better understanding of the results obtained from a search for different approaches to solving a problem or proving a theorem. It promotes new forms of reasoning that can inspire simple and elegant solutions to similar problems, allowing students to respond to more complex problems or prove new theorems (de Villiers, 2006).

Introducing DGSs in the learning process enables greater experimentation than previous tools did. DGSs allow free geometric objects to be continuously modified, thereby keeping geometric properties unchanged. Although these manipulations are not formal proofs because only a finite set of positions is considered and because visualisation can be misleading (when the image, for example, refers to a particular case), they provide a first clue to the truthfulness of a given geometric conjecture (Hoyles & Jones, 1998; Nelsen, 1993; de Villiers, 2006). It can be said that DGSs provide an initial non-formal link between theories and models of geometry (Janičić & Quaresma, 2007; Ruthven, Hennessy, & Deaney, 2008).

Geometry automated theorem provers similarly enhance learning. First, they can be used to validate a geometric construction such as: are two given lines parallel? The answer—yes!/no!/cannot be established—is a “black box” approach (it does not answer the “why”), but can be useful in situations where formal proofs are not being considered. Second, some GATPs are able to provide formal proofs—not only the answer to a question, but also the “why.” As said above, this constitutes the “best proof.”

Unlike DGSs that are similar in their approach and capabilities, GATPs vary significantly in terms of the method they use (e.g., synthetic vs. algebraic), the implementation (e.g., area vs. full-angle), and the output (formal proof vs. only a yes/no answer) (Chou & Gao, 2001; Chou, Gao, & Zhang, 1994; Quaresma, 2017; Wang & Su, 2015; Wu, 1984). According to Jiang and Zhang (2012), consideration needs to be given to the somehow opposite goals of efficiency and readability. Efficiency is important because, in a learning situation, it is not viable to wait more than a couple of seconds to get an answer. Readability is important given that, without it, the “why” would be lost.

The first methods proposed in the 1950s adapted general reasoning approaches from the field of artificial intelligence to the automation of traditional geometric proving processes. In order to avoid combinatorial explosion while applying postulates, many suitable heuristics were used; for example, adding auxiliary elements to geometric configurations, and using the corresponding geometric constructions as a source of counterexamples to help the reasoning procedure choose the right path. Although these methods produced readable proofs that could be used in learning, they were very narrow in scope and inefficient (Wu, 1984). However, some recent

results using this approach in education focus on developing GATPs scoped for a given set of problems (Paneque, Cobo, Fortuny, & Richard, 2016; Richard, Oller Marcén, & Meavilla Seguí, 2016). Still, its usefulness in education has yet to be established.

The next step in creating more generic, powerful, and efficient provers was to integrate algebraic methods such as the characteristic set method (also known as Wu’s method) (Chou, 1985; Wu, 1984), the polynomial elimination method (Wang, 1995), the Gröbner based method (Kapur, 1986), and the Clifford algebra approach (Li, 2000). These methods reduced the complexity of logical inferences by computing relations between coordinates of geometric entities. The corresponding implementations could successfully solve many complicated geometric problems and discover new theorems. However, human readability of the proofs was lost, and the algebraic proofs were complex and not related to geometric reasoning (Chou et al., 1994; Wu, 1984). Nonetheless, there are many recent implementations of these methods (Botana et al., 2015; Janičić et al., 2012; Janičić & Quaresma, 2006; Kovács, 2015; Ye, Chou, & Gao, 2011) and, in Sect. 3, we explore the possible use of these GATPs in education.

In an effort to combine the readability of synthetic methods and the efficiency of algebraic methods, some approaches, such as the area method (Chou et al., 1996a; Janičić et al., 2012) and the full-angle method (Chou et al., 1996b), represent geometric knowledge with respect to geometric invariants. These methods and some of their implementations are capable of proving a large number of complex geometric theorems and, in many cases, rendering the proofs in readable natural language. In Sects. 4 and 5, we describe this innovation in length.

### 3 Verification of the Truth of a Geometric Statement

Dynamic geometry programs give users an initial visual validation of a geometric property. Instead of producing a fixed example, these programs produce a large set of examples that reinforce confidence in the truth of a statement. However, given that not all possible cases can be covered, this can be misleading.

*GeoGebra* has had, since an early version, a validation tool, the “ $a \stackrel{?}{=} b$ ” tool. This tool performs a numerical verification instead of a formal proof, and suffers from the fact that numerical errors may lead to erroneous conclusions. However, since version 5, *GeoGebra* has been enhanced with the support of GATPs. This makes it possible to ask for a formal validation of a given geometric statement (Botana et al., 2015) such as the Midpoint Theorem.

**Theorem 1** (Midpoint Theorem) *Let  $ABC$  be a triangle, and let  $D$  and  $E$  be the midpoints of  $AB$  and  $BC$  respectively. Then the line  $DE$  is parallel to the base  $AC$ .*

Having made the geometric construction (see Fig. 1), a *GeoGebra* user can check if the conjecture is indeed true. Entering the command “Prove (AreParallel(b, f))” produces the answer “d=true” (see Fig. 1). This is a formal validation generated by *GeoGebra*’s built-in GATPs.

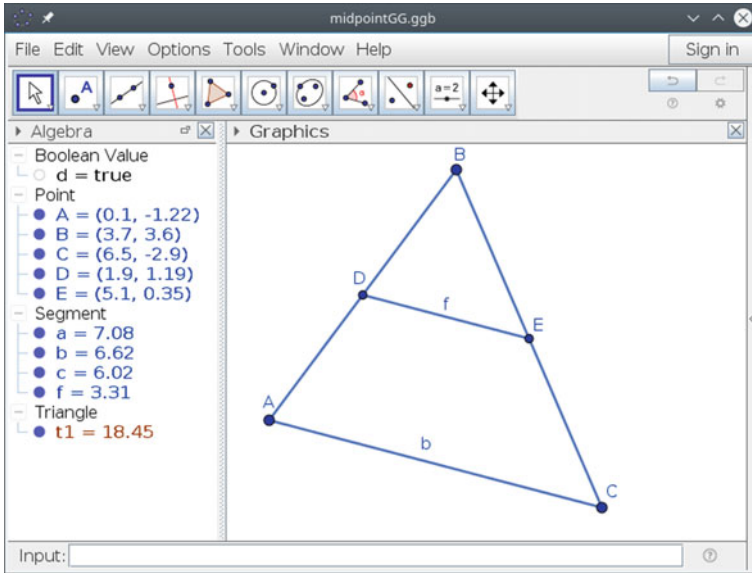


Fig. 1 GeoGebra construction validation

Other DGSs also have integrated GATPs. For example, *GCLC* incorporates *GCLCprover*, which gives the following response to a similar situation (Janičić & Quaresma, 2007):

Deduction check invoked: the property that led to the error will be tested for validity.

Once the conjecture is successfully proved, the critical property always holds. The prover output is written in the file `error-proof.tex`.

A similar approach is used by the DGS/GATP *Java Geometry Expert (JGEX)*<sup>4</sup> (Ye et al., 2011). It calculates fix-points during the construction; that is, all the properties that can be inferred from the construction to a certain point. When a user tries to perform an illegal construction, the tool says why it is not possible to perform that construction (see Fig. 2).

If we jump to “explanation”—providing insight into why a given statement is true—we need to consider proofs. In the next two sections, we explore the capabilities of current GATPs to produce readable proofs with a natural language and even visual rendering.

<sup>4</sup><http://www.cs.wichita.edu/~ye/>.

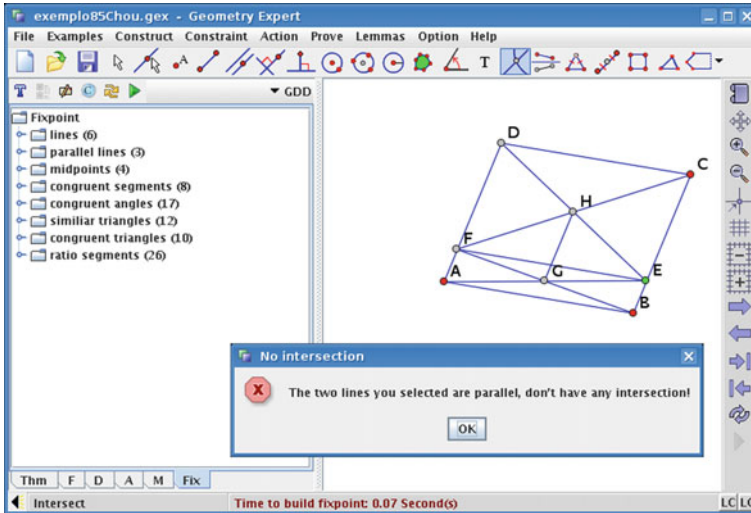


Fig. 2 *JGEX* construction validation,  $\overline{FE} \parallel \overline{AB}$

## 4 Proofs with a Natural Language Rendering

Some of the methods used for geometric automated theorem proving, especially algebraic methods, are not useful in terms of the proofs produced. After a synthetic geometric representation is converted to an algebraic representation, the proofs are all done using complex algebraic reasoning without any connection to the geometric construction. The algorithms implemented in *GeoGebra* are of this type. They produce a time efficient answer to a verification problem, but are useless if the proof itself is the focus.

In (2010a), Ye et al. list the features they feel important to the dynamic visualisation of proofs in geometry. The first of these features is that “The proof text created by the program should be readable, similar to proofs in geometry textbooks or books.”

The semi-synthetic methods like the area method, the full-angle method or the deductive database method, along with the coherent logic based method, are able to produce readable geometric proofs. Implementations like the *GCLC* (area method) (Janičić et al., 2012; Quaresma, Janičić, Tomašević, Vujošević-Janičić, & Tošić, 2008), the *JGEX* (area method, full-angle method, and deductive databases) (Ye et al., 2010a, 2010b, 2011) and the *ArgoCLP*<sup>5</sup> (coherent logic) (Stojanović et al., 2011) are examples of systems with such capabilities.

**The Area Method** The area method is a decision procedure for a fragment of Euclidean plane geometry. It deals with problems as sequences of specific geometric construction steps, using a set of specific *geometric quantities* to define relations (Janičić et al., 2012).

<sup>5</sup><http://argo.matf.bg.ac.rs/?content=downloads>.

**Table 1** Expressing geometry predicates in terms of area method geometric quantities

Property	Area method geometric quantities
Points $A$ and $B$ are identical	$\mathcal{P}_{ABA} = 0$
Points $A, B, C$ are collinear	$\mathcal{S}_{ABC} = 0$
$AB$ is perpendicular to $CD$	$\mathcal{P}_{ABA} \neq 0 \wedge \mathcal{P}_{CDC} \neq 0 \wedge \mathcal{P}_{ACD} = \mathcal{P}_{BCD}$
$AB$ is parallel to $CD$	$\mathcal{P}_{ABA} \neq 0 \wedge \mathcal{P}_{CDC} \neq 0 \wedge \mathcal{S}_{ACD} = \mathcal{S}_{BCD}$
$O$ is the midpoint of $AB$	$\mathcal{S}_{ABO} = 0 \wedge \mathcal{P}_{ABA} \neq 0 \wedge \frac{\overline{AO}}{\overline{AB}} = \frac{1}{2}$
$AB$ has the same length as $CD$	$\mathcal{P}_{ABA} = \mathcal{P}_{CDC}$
Points $A, B, C, D$ are harmonic	$\mathcal{S}_{ABC} = 0 \wedge \mathcal{S}_{ABD} = 0 \wedge \mathcal{P}_{BCB} \neq 0 \wedge \mathcal{P}_{BDB} \neq 0 \wedge \frac{\overline{AC}}{\overline{CB}} = \frac{\overline{DA}}{\overline{DB}}$
Angle $ABC$ has the same measure as $DEF$	$\mathcal{P}_{ABA} \neq 0 \wedge \mathcal{P}_{ACA} \neq 0 \wedge \mathcal{P}_{BCB} \neq 0 \wedge \mathcal{P}_{DED} \neq 0 \wedge \mathcal{P}_{DFD} \neq 0 \wedge \mathcal{P}_{EFE} \neq 0 \wedge \mathcal{S}_{ABC} \cdot \mathcal{P}_{DEF} = \mathcal{S}_{DEF} \cdot \mathcal{P}_{ABC}$
$A$ and $B$ belong to the same circle arc $CD$	$\mathcal{S}_{ACD} \neq 0 \wedge \mathcal{S}_{BCD} \neq 0 \wedge \mathcal{S}_{CAD} \cdot \mathcal{P}_{CBD} = \mathcal{S}_{CBD} \cdot \mathcal{P}_{CAD}$

- *Ratio of parallel directed segments*, denoted  $\overline{AB}/\overline{CD}$ . If the points  $A, B, C$ , and  $D$  are collinear,  $\overline{AB}/\overline{CD}$  is the ratio between the lengths of directed segments  $AB$  and  $CD$ . If the points  $A, B, C$ , and  $D$  are not collinear, and it holds  $AB \parallel CD$ , there is a parallelogram  $ABPQ$  such that  $P, Q, C$ , and  $D$  are collinear and then  $\frac{\overline{AB}}{\overline{CD}} = \frac{\overline{QP}}{\overline{CD}}$ .
- *Signed area* for a triangle  $ABC$ , denoted  $\mathcal{S}_{ABC}$  is the area of the triangle  $ABC$ , negated if  $ABC$  has a negative orientation.
- *Pythagoras difference*,<sup>6</sup> denoted  $\mathcal{P}_{ABC}$ , for the points  $A, B, C$ , defined as  $\mathcal{P}_{ABC} = \overline{AB}^2 + \overline{CB}^2 - \overline{AC}^2$ .

These three geometric quantities allow for expressing (in the form of equalities) geometry properties such as the collinearity of three points, the parallelism of two lines, the equality of two points, the perpendicularity of two lines, and so forth (see Table 1).

The basic objects in an area method conjecture are the points—free points if they are freely placed in the Euclidean plane, and constructed points if they are obtained as the result of a given geometric construction. For example, in the Midpoint Theorem (see Theorem 1) the points  $A, B$  and  $C$  are free points not defined by construction steps. The constructed points  $D$  and  $E$  are the midpoints of  $\overline{AB}$  and  $\overline{BC}$  respectively.

The proof of a conjecture is based on eliminating all the constructed points in reverse order until equality is reached in only the free points. If the equality is provable, then the original conjecture is also a theorem. For Theorem 1, the proof automatically found by *GCLCprover* (see Fig. 3) shows not only the steps leading to

<sup>6</sup>The *Pythagoras difference* is a generalisation of the Pythagorean equality regarding the three sides of a right triangle, to an expression applicable to any triangle (for a triangle  $ABC$  with the right angle at  $B$ , it holds that  $\mathcal{P}_{ABC} = 0$ ).

- (1)  $S_{DEA} = S_{DEC}$ , by the statement
- (2)  $S_{ADE} = S_{CDE}$ , by geometric simplifications
- (3)  $\left(S_{ADC} + \left(\frac{1}{2} \cdot (S_{ADB} + (-1) \cdot S_{ADC})\right)\right) = S_{CDE}$ , by Lemma 29 (point  $E$  eliminated)
- (4)  $\left(\left(\frac{1}{2} \cdot S_{ADC}\right) + \left(\frac{1}{2} \cdot S_{ADB}\right)\right) = S_{CDE}$ , by algebraic simplifications
- (5)  $\left(\left(\frac{1}{2} \cdot S_{ADC}\right) + \left(\frac{1}{2} \cdot S_{ADB}\right)\right) = \left(S_{CDC} + \left(\frac{1}{2} \cdot (S_{CDB} + (-1) \cdot S_{CDC})\right)\right)$ , by Lemma 29 (point  $E$  eliminated)
- (6)  $\left(\left(\frac{1}{2} \cdot S_{CAD}\right) + \left(\frac{1}{2} \cdot S_{BAD}\right)\right) = \left(0 + \left(\frac{1}{2} \cdot (S_{BCD} + (-1) \cdot 0)\right)\right)$ , by geometric simplifications
- (7)  $(S_{CAD} + S_{BAD}) = S_{BCD}$ , by algebraic simplifications
- (8)  $\left(\left(\left(S_{CAA} + \left(\frac{1}{2} \cdot (S_{CAB} + (-1) \cdot S_{CAA})\right)\right)\right) + S_{BAD}\right) = S_{BCD}$ , by Lemma 29 (point  $D$  eliminated)
- (9)  $\left(\left(0 + \left(\frac{1}{2} \cdot (S_{CAB} + (-1) \cdot 0)\right)\right) + S_{BAD}\right) = S_{BCD}$ , by geometric simplifications
- (10)  $\left(\left(\frac{1}{2} \cdot S_{CAB}\right) + S_{BAD}\right) = S_{BCD}$ , by algebraic simplifications
- (11)  $\left(\left(\frac{1}{2} \cdot S_{CAB}\right) + \left(S_{BAA} + \left(\frac{1}{2} \cdot (S_{BAB} + (-1) \cdot S_{BAA})\right)\right)\right) = S_{BCD}$ , by Lemma 29 (point  $D$  eliminated)
- (12)  $\left(\left(\frac{1}{2} \cdot S_{CAB}\right) + \left(0 + \left(\frac{1}{2} \cdot (0 + (-1) \cdot 0)\right)\right)\right) = S_{BCD}$ , by geometric simplifications
- (13)  $\left(\frac{1}{2} \cdot S_{CAB}\right) = S_{BCD}$ , by algebraic simplifications
- (14)  $\left(\frac{1}{2} \cdot S_{CAB}\right) = \left(S_{BCA} + \left(\frac{1}{2} \cdot (S_{BCB} + (-1) \cdot S_{BCA})\right)\right)$ , by Lemma 29 (point  $D$  eliminated)
- (15)  $\left(\frac{1}{2} \cdot S_{CAB}\right) = \left(S_{CAB} + \left(\frac{1}{2} \cdot (0 + (-1) \cdot S_{CAB})\right)\right)$ , by geometric simplifications
- (16)  $0 = 0$ , by algebraic simplifications

**Fig. 3** *GCLCprover* output—Proof of Midpoint Theorem

the solution, but also the justification for those steps. *GCLCprover* uses the axioms and rules of inference of the area method to reach the conclusion in 16 steps, of which only 5 are applications of lemmas of the method (e.g., Lemma 29).

The area method implementation in *GCLC* took 0.001 s to prove the Midpoint Theorem. It also found that there are no non-degenerated conditions to this result; that is, it is true in any configuration.

Given that the area method does not follow the normal chain of geometric reasoning used in primary and secondary schools, the resulting proofs are not directly usable by students. However, with the help of teachers, the area method (the full-angle method is similar) could be used at secondary and college levels. This claim needs validation by case studies (see Sect. 6).



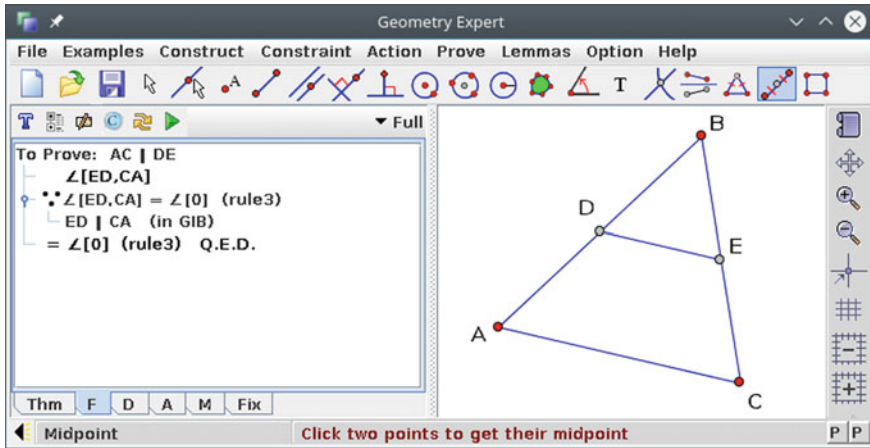


Fig. 4 JGEX screen capture—Proof of Midpoint Theorem

**The Full-Angle Method** The full-angle method is similar to the area method, but introduces the full-angle as another geometric quantity. Intuitively, a full-angle,  $\angle[u, v]$ , is the angle from line  $u$  to line  $v$  (not between segments).

Formally, full-angles can also be defined using the signed area and the Pythagoras difference. A full-angle is defined as an ordered pair of lines which satisfies a set of axioms and rules of inference that constitute the base of the method. The proofs are similar to those using the area method (Chou et al., 1996a).

The JGEX system implements the full-angle method and the deductive database method (based on full-angle rules). Using the full-angle prover, the Midpoint Theorem (see Theorem 1) can be proved (see Fig. 4) in a very concise form. However, the JGEX system does not produce a PDF output file.

Like the area method proofs, the use of full-angle method proofs in primary and secondary levels is limited by the axioms and rules of inference used. JGEX does not produce a proof with a natural language rendering. Unless the reader is familiar with the axiom system and corresponding inference rules, the proofs are somehow difficult to follow.

**Coherent Logic Provers** A different approach is given by the *ArgoCLP*, a coherent logic-based theorem prover. Coherent logic is a fragment of first-order logic where the conjectures can be proved directly (i.e., not by refutation). Proofs in coherent logic are natural and intuitive, and reasoning is constructive (Bezem & Coquand, 2005). Coherent logic is therefore a suitable framework for producing both readable and formal proofs. The *ArgoCLP* automatically and simultaneously generates traditional human readable proofs and formal proofs of geometry theorems (for various axiom systems). The generated step-by-step proofs are very similar to the proofs given in standard geometry textbooks. Unfortunately, however, efficiency issues prevent its use in education (Stojanović et al., 2011, pp. 13–14).

## 5 Proofs with Natural Language and Visual Rendering

Given the current implementations in geometry automated theorem proving, it is now possible to begin work on a new approach: proofs that interrelate geometric elements in the proof text and the construction.

As Ye et al. (2010a) show in their features of visually dynamic proofs in geometry, effective connection between the visual rendering and textual rendering is key:

The displays of the proof text and the geometry elements in the construction are separated, but should be internally related. By clicking a step or a part of a step of the proof’s text, the corresponding geometric elements in the construction should be highlighted using various dynamic visual effects.

They propose visual effects such as translations, rotations, reflections, and scaling to show congruence and similarities between geometric elements. Colours and blinking effects can be used to show the relations between the two renderings of a proof.

As discussed above (see Sect. 4), natural language rendering is possible and usable with some limitations in different contexts. Is it possible to connect natural language rendering with a formal proof and visual rendering? In the following, we examine the strengths and challenges of different approaches.

### 5.1 Area Method Visual Rendering of Proofs

As mentioned above (see Sect. 4), the area method deals with problems stated as sequences of specific geometric construction steps, using a set of specific *geometric quantities* (Chou et al., 1996a, 1996b; Janičić et al., 2012). In the following, we establish a connection between the axioms and inference rules of this method. Using this connection makes it possible to synchronize a formal proof, a natural language rendering, and a visual rendering.

**Visual Counterparts of Geometric Quantities** The *ratio of parallel directed segments* and the *signed area* have a clear visual counterpart. The *Pythagoras difference* needs the support of the algebraic quantity  $\overline{AB}^2 + \overline{CB}^2 - \overline{AC}^2$  (see Fig. 5).

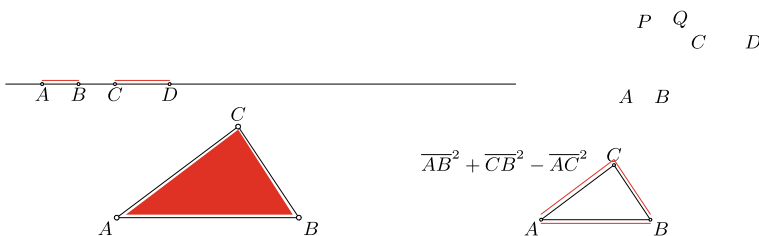
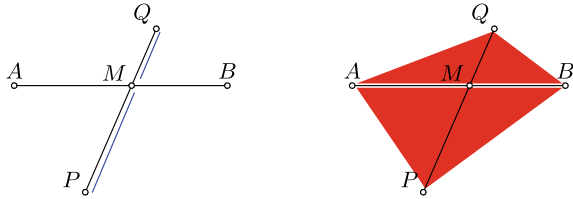


Fig. 5 Illustration for area method geometric quantities

**Fig. 6** Illustration for elimination Lemma 1:  
 $\frac{PM}{QM} = \frac{S_{PAB}}{S_{QAB}}$



**Axioms and Lemmas of the Area Method** The area method uses a large set of lemmas that characterise the geometric quantities and facilitate structuring the proofs (Chou et al., 1996a; Janičić et al., 2012; Quaresma & Janičić, 2009). Many of these lemmas are about technical issues related to the formal proofs; for example, how two triangles that differ only in the order of their vertices can be considered the same. The most important for area method proofs and also for their visualisation are the elimination lemmas.

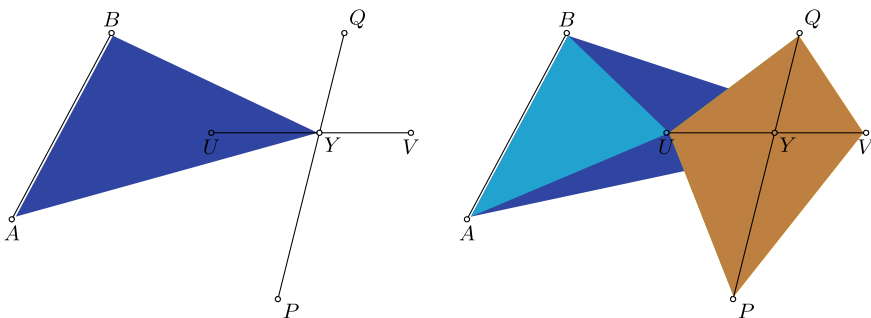
In the following, we introduce two of these lemmas and present their visual counterparts.

**Theorem 5.1** (Elimination Lemma 1—The Co-side Theorem) *Let M be the intersection of two non-parallel lines AB and PQ and  $Q \neq M$ . Then it holds that  $\frac{PM}{QM} = \frac{S_{PAB}}{S_{QAB}}$ ,  $\frac{PM}{PQ} = \frac{S_{PAB}}{S_{PAQB}}$ ,  $\frac{QM}{PQ} = \frac{S_{QAB}}{S_{PAQB}}$  (Fig. 6).*

**Theorem 5.2** (Elimination Lemma 3 (reduced version)) *Let  $S_{ABY}$  be the  $\Delta ABY$  signed area where point Y is the intersection of lines UV and PQ. Then it holds that (Fig. 7)*

$$S_{ABY} = \frac{S_{UPQ}S_{ABV} - S_{VPQ}S_{ABU}}{S_{UPVQ}}$$

**Area Method Visual Proof—Midpoint Theorem** Using the visual counterparts of the lemmas of the area method we can revisit the Midpoint Theorem (see Theorem 1).



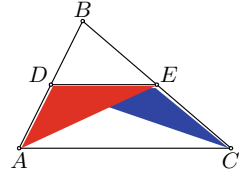
**Fig. 7** Illustration for elimination Lemma 3:  $S_{ABY} = \frac{S_{UPQ}S_{ABV} - S_{VPQ}S_{ABU}}{S_{UPVQ}}$

This time the *GCLCprover* area method proof has two parts: a natural language part and a visual counterpart.

(1–2) by the statement and geometric simplifications

$$S_{DEA} = S_{DEC}$$

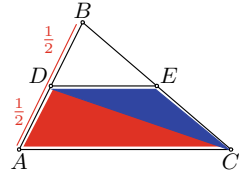
$$S_{ADE} = S_{CDE}$$



(3–4) by Lemma 29 (point *E* eliminated) and algebraic simplifications

$$S_{ADC} + \frac{1}{2}(S_{ADB} - S_{ADC}) = S_{CDE}$$

$$\frac{1}{2}S_{ADC} + \frac{1}{2}S_{ADB} = S_{CDE}$$

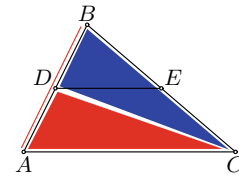


(5–7) by Lemma 29 (point *E* eliminated), geometric and algebraic simplifications

$$\frac{1}{2}S_{ADC} + \frac{1}{2}S_{ADB} = S_{CDC} + \frac{1}{2}(S_{CDB} - S_{CDC})$$

$$\frac{1}{2}S_{CAD} + \frac{1}{2}S_{BAD} = 0 + \frac{1}{2}(S_{BCD} - 0)$$

$$S_{CAD} + S_{BAD} = S_{BCD}$$

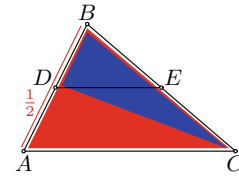


(8–10) by Lemma 29 (point *D* eliminated), geometric and algebraic simplifications

$$S_{CAA} + \frac{1}{2}(S_{CAB} - S_{CAA}) + S_{BAD} = S_{BCD}$$

$$0 + \frac{1}{2}(S_{CAB} - 0) + S_{BAD} = S_{BCD}$$

$$\frac{1}{2}S_{CAB} + S_{BAD} = S_{BCD}$$

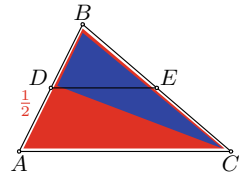


(11–13) by Lemma 29 (point *D* eliminated), geometric and algebraic simplifications

$$\frac{1}{2}S_{CAB} + S_{BAA} + \frac{1}{2}(S_{BAB} - S_{BAA}) = S_{BCD}$$

$$\frac{1}{2}S_{CAB} + 0 + \frac{1}{2}(0 - 0) = S_{BCD}$$

$$\frac{1}{2}S_{CAB} = S_{BCD}$$

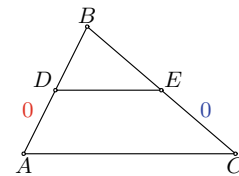


(14–16) by Lemma 29 (point *D* eliminated), geometric and algebraic simplifications

$$\frac{1}{2}S_{CAB} = S_{BCA} + \frac{1}{2}(S_{BCB} - S_{BCA})$$

$$\frac{1}{2}S_{CAB} = S_{CAB} + \frac{1}{2}(0 - S_{CAB})$$

$$0 = 0 \quad \text{Q.E.D.}$$



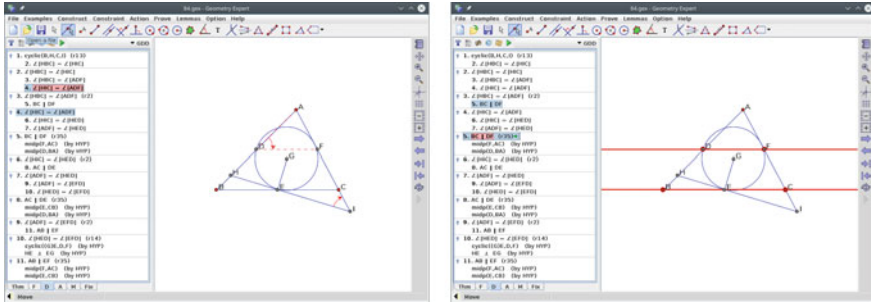


Fig. 8 JGEX—Example 84, Steps 2 and 5

The example illustrates how to express a problem using the given geometric quantities and how to prove it in a concise, easily understood way with a corresponding visual rendering.

As far as we know, there are no systems for the area method that provide such a connection. It is not yet a feature of *GCLC* (see Sect. 6). However, *JGEX* (Ye et al., 2010a, 2010b, 2011) does provide this connection for the full-angle method.

### 5.2 Full-Angle Method Visual Proofs

Using the *JGEX* system, we can build a given construction, state a conjecture about it, and then, using one of the built-in GATPs, prove it. Using the full-angle method based GATP we can produce examples where the formal proof has a visual counterpart.

Figure 8 was taken from the tool’s own set of examples. Clicking on a step of the formal proof produces a visual animation of the step on the construction. The related relations between objects on the construction—e.g. the angles between two lines in Fig. 8 (left)—initially “blink.” They then become fixed, but use colours to clearly show the corresponding relations in the formal proof.

In both situations, the drawback is that neither the area method nor the full-angle method use the usual set of axioms and rules of inference of primary and secondary school geometry. Instead, they use the geometric quantities *ratio of parallel directed segments*, *signed area*, *Pythagoras difference* and *full-angle*, and the axioms and rules of inference for these geometric quantities. Using these methods in secondary schooling could prove difficult.

## 6 Conclusions and Future Work

Geometry, with its very strong and appealing visual content and its formal axiomatic theory, is a privileged domain. It is an area where computational tools can significantly enhance the learning environment and make students active in building their knowledge.

Introducing dynamic geometry systems enhances exploration and proof by making it easy to posit and test conjectures. The ability to make, properties-preserving, manipulations enables students to explore “visual proofs” of geometric conjectures. DGSs significantly help students to acquire knowledge about geometric objects and, more generally, to acquire mathematical rigour.

Geometry automated theorem provers capable of construction validation and human readable proofs consolidate the knowledge acquired with the use of DGSs. If the GATP produces synthetic proofs, the proof of a conjecture or the proof of the soundness of a construction can be used as an object of study providing a logical explanation. With a DGS, students can visually explore constructions or check that certain conjectures are true. However, these systems do not provide mathematical arguments for the conclusions they produce. Instead of producing a “visual check,” a GATP can be used to draw accurate mathematical conclusions. Thus, we claim that GATPs can be used in the learning process (Janičić & Quaresma, 2007; Quaresma & Janičić, 2006; Santos & Quaresma, 2012, 2013).

The natural language rendering of a formal proof, especially if paired with a visual rendering, would allow for a wider application of GATPs in learning contexts. This is still an active area of research involving exploration of different methods, implementations, and renderings. Currently, we are exploring the construction of a controlled hybrid language for geometry; that is, a pair of controlled languages (natural and visual) with common semantics. By considering figures as sentences in a visual language sharing semantics with the natural language of geometric statements, we can achieve interaction between parts of text and corresponding figures. We can connect formal proofs to natural language and visual descriptions. This approach is more generic than the concrete cases described above. It is a line of research we are already pursuing.<sup>7</sup>

Given the availability of GATP technology and the work currently being done on the rendering of the proofs, it is important to begin integrating these advances in learning environments. As authors, we have already developed the *Web Geometry Laboratory* (WGL),<sup>8</sup>—an adaptive and collaborative blended-learning Web environment that integrates a dynamic geometry system (Quaresma, Santos, & Bouallegue, 2013; Quaresma, Santos, & Marić, 2018; Santos, Quaresma, Marić, & Campos, 2018). A short/medium term goal of the WGL is to connect with the *OpenGeoProver* (OGP)<sup>9</sup> (Baeta & Quaresma, 2013), an open library of GATPs that we are currently developing alongside with others researchers. The connection between WGL and OGP will provide the lab with the automatic deduction capabilities discussed in this chapter; namely, checking the validation of a construction and getting formal proofs with a human readable and visual rendering (Quaresma & Santos, 2016). This and other projects like the integration of GATPs in *GeoGebra* (Botana et al., 2015) aim to

---

<sup>7</sup>Haralambous, Yannis and Quaresma, Pedro, *Geometric Statements as Controlled Hybrid Language Sentences, an Example* in preparation.

<sup>8</sup><http://hilbert.mat.uc.pt/WebGeometryLab>.

<sup>9</sup><https://github.com/ivan-z-petrovic/open-geo-prover>.

include the power of automatic deduction in the learning and teaching of geometry. Ultimately, our goal is to help teachers and students answer the question “why.”

## References

- Abánades, M., Botana, F., Kovács, Z., Recio, T., & Sólyom-Gecse, C. (2016). Towards the automatic discovery of theorems in GeoGebra. In G. M. Greuel, T. Koch, P. Paule, & A. Sommese (Eds.), *Mathematical Software—ICMS 2016* (pp. 37–42). Cham: Springer International Publishing.
- Baeta, N., & Quaresma, P. (2013). The full angle method on the OpenGeoProver. In C. Lange, D. Aspinall, J. Carette, J. Davenport, A. Kohlhase, M. Kohlhase, P. Libbrecht, P. Quaresma, F. Rabe, P. Sojka, I. Whiteside, & W. Windsteiger (Eds.), *MathUI, OpenMath, PLMMS and ThEdu Workshops and Work in Progress at the Conference on Intelligent Computer Mathematics*, no. 1010 in CEUR Workshop Proceedings. Aachen. <http://ceur-ws.org/Vol-1010/paper-08.pdf>.
- Bezém, M., & Coquand, T. (2005). Automating coherent logic. In G. Sutcliffe & A. Voronkov (Eds.), *Logic for programming, artificial intelligence, and reasoning*. Lecture Notes in Computer Science (Vol. 3835, pp. 246–260). Berlin/Heidelberg: Springer. [https://doi.org/10.1007/11591191\\_18](https://doi.org/10.1007/11591191_18).
- Botana, F., Hohenwarter, M., Janičić, P., Kovács, Z., Petrović, I., Recio, T., et al. (2015). Automated theorem proving in GeoGebra: Current achievements. *Journal of Automated Reasoning*, 55(1), 39–59. <https://doi.org/10.1007/s10817-015-9326-4>.
- Chou, S. (1985). Proving and discovering geometry theorems using Wu’s method. Ph.D. thesis, The University of Texas, Austin.
- Chou, S. C., & Gao, X. S. (2001). Automated reasoning in geometry. In J. A. Robinson & A. Voronkov (Eds.), *Handbook of automated reasoning* (pp. 707–749). Elsevier Science Publishers B.V.
- Chou, S. C., Gao, X. S., & Zhang, J. Z. (1994). *Machine proofs in geometry*. World Scientific.
- Chou, S. C., Gao, X. S., & Zhang, J. Z. (1996a). Automated generation of readable proofs with geometric invariants, I. Multiple and shortest proof generation. *Journal of Automated Reasoning*, 17(13), 325–347. <https://doi.org/10.1007/BF00283133>.
- Chou, S. C., Gao, X. S., & Zhang, J. Z. (1996b). Automated generation of readable proofs with geometric invariants, II. Theorem proving with full-angles. *Journal of Automated Reasoning*, 17(13), 349–370. <https://doi.org/10.1007/BF00283134>.
- Hanna, G. (2000). Proof, explanation and exploration: An overview. *Educational Studies in Mathematics*, 44(1–2), 5–23. <https://doi.org/10.1023/A:1012737223465>.
- Hanna, G., & Sidoli, N. (2007). Visualisation and proof: A brief survey of philosophical perspectives. *ZDM*, 39(1–2), 73–78. <https://doi.org/10.1007/s11858-006-0005-0>.
- Hohenwarter, M. (2002). Geogebra—A software system for dynamic geometry and algebra in the plane. Master’s thesis, University of Salzburg, Austria.
- Hoyle, C., & Jones, K. (1998). Proof in dynamic geometry contexts. In *Perspectives on the teaching of geometry for the 21st century* (pp. 121–128). Springer. <https://eprints.soton.ac.uk/41227/>.
- Jančić, P. (2006). GCLC—A tool for constructive Euclidean geometry and more than that. In A. Iglesias & N. Takayama (Eds.) *Mathematical Software—ICMS 2006*. Lecture Notes in Computer Science (Vol. 4151, pp. 58–73). Springer. [https://doi.org/10.1007/11832225\\_6](https://doi.org/10.1007/11832225_6).
- Jančić, P., Narboux, J., & Quaresma, P. (2012). The area method: A recapitulation. *Journal of Automated Reasoning*, 48(4), 489–532. <https://doi.org/10.1007/s10817-010-9209-7>.
- Jančić, P., & Quaresma, P. (2006). System description: GCLCprover + GeoThms. In U. Furbach, N. Shankar (Eds.), *Automated reasoning*. Lecture Notes in Computer Science (Vol. 4130, pp. 145–150). Springer. [https://doi.org/10.1007/11814771\\_13](https://doi.org/10.1007/11814771_13).
- Jančić, P., & Quaresma, P. (2007). Automatic verification of regular constructions in dynamic geometry systems. In F. Botana & T. Recio (Eds.), *Automated deduction in geometry*. Lecture

- Notes in Computer Science (Vol. 4869, pp. 39–51). Springer. [https://doi.org/10.1007/978-3-540-77356-6\\_3](https://doi.org/10.1007/978-3-540-77356-6_3).
- Jiang, J., & Zhang, J. (2012). A review and prospect of readable machine proofs for geometry theorems. *Journal of Systems Science and Complexity*, 25(4), 802–820. <https://doi.org/10.1007/s11424-012-2048-3>.
- Kapur, D. (1986). Using Gröbner bases to reason about geometry problems. *Journal of Symbolic Computation*, 2(4), 399–408. [https://doi.org/10.1016/S0747-7171\(86\)80007-4](https://doi.org/10.1016/S0747-7171(86)80007-4).
- Kovács, Z. (2015). Computer based conjectures and proofs in teaching Euclidean geometry. Ph.D. thesis, Universität Linz. urn:nbn:at:at-ubl:1-5034.
- Kovács, Z. (2015). *The relation tool in GeoGebra* (Vol. 5, pp. 53–71). Springer International Publishing. [https://doi.org/10.1007/978-3-319-21362-0\\_4](https://doi.org/10.1007/978-3-319-21362-0_4).
- Li, H. (2000). Clifford algebra approaches to mechanical geometry theorem proving. In X. S. Gao & D. Wang (Eds.), *Mathematics mechanization and applications* (pp. 205–299). San Diego, CA: Academic Press.
- Lin, F. L., Hsieh, F. J., Hanna, G., & de Villiers, M. (Eds.). (2009a). *Proceedings of the ICMI Study 19 Conference: Proof and Proving in Mathematics Education* (Vol. 1). The Department of Mathematics: National Taiwan Normal University.
- Lin, F. L., Hsieh, F. J., Hanna, G., & de Villiers, M. (Eds.). (2009b). *Proceedings of the ICMI Study 19 conference: Proof and Proving in Mathematics Education* (Vol. 2). The Department of Mathematics: National Taiwan Normal University.
- Nelsen, R. B. (1993). *Proofs without words: Exercises in visual thinking* (Vol. 1). MAA.
- Paneque, J., Cobo, P., Fortuny, J., & Richard, P. R. (2016). Argumentative effects of a geometric construction tutorial system in solving problems of proof. In: *Proceedings of the 4th International Workshop on Theorem Proving Components for Educational Software*, July 15, 2015, Washington, D.C., USA. CISUC Technical Reports (Vol. 2016-001, pp. 13–35). CISUC.
- Quaresma, P. (2017). Towards an intelligent and dynamic geometry book. *Mathematics in Computer Science*, 11(3), 427–437. <https://doi.org/10.1007/s11786-017-0302-8>.
- Quaresma, P., & Janičić, P. (2006). Integrating dynamic geometry software, deduction systems, and theorem repositories. In J. M. Borwein & W. M. Farmer (Eds.), *Mathematical knowledge management*. Lecture Notes in Computer Science (Vol. 4108, pp. 280–294). Berlin: Springer. [https://doi.org/10.1007/11812289\\_22](https://doi.org/10.1007/11812289_22).
- Quaresma, P., & Janičić, P. (2009). The area method, rigorous proofs of lemmas in Hilbert’s style axiom system. Tech. Rep. 2009/006, Centre for Informatics and Systems of the University of Coimbra.
- Quaresma, P., Janičić, P., Tomašević, J., Vujošević-Janičić, M., & Tošić, D. (2008). Communicating mathematics in the digital era. In *XML-bases format for descriptions of geometric constructions and proofs* (pp. 183–197). Wellesley, MA: A. K. Peters, Ltd.
- Quaresma, P., & Santos, V. (2016). Visual geometry proofs in a learning context. In W. Neuper & P. Quaresma (Eds.), *Proceedings of ThEdu’15, CISUC Technical Reports* (Vol. 2016001, pp. 1–6). CISUC. <https://www.cisuc.uc.pt/ckfinder/userfiles/files/TR2016-01.pdf>.
- Quaresma, P., Santos, V., & Bouallegue, S. (2013). The Web Geometry Laboratory project. In J. Carette, D. Aspinall, C. Lange, P. Sojka & W. Windsteiger (Eds.), *CICM 2013*. Lecture Notes in Computer Science (Vol. 7961, pp. 364–368). Springer. [https://doi.org/10.1007/978-3-642-39320-4\\_30](https://doi.org/10.1007/978-3-642-39320-4_30).
- Quaresma, P., Santos, V., & Marić, M. (2018). WGL, a web laboratory for geometry. *Education and Information Technologies*, 23(1), 237–252. <https://doi.org/10.1007/s10639-017-9597-y>.
- Recio, T., & Vélez, M. P. (2012). *An introduction to automated discovery in geometry through symbolic computation* (pp. 257–271). Vienna: Springer. [https://doi.org/10.1007/978-3-7091-0794-2\\_12](https://doi.org/10.1007/978-3-7091-0794-2_12).
- Richard, P. R., Oller Marcén, A. M., & Meavilla Seguí, V. (2016). The concept of proof in the light of mathematical work. *ZDM*, 48(6), 843–859. <https://doi.org/10.1007/s11858-016-0805-9>.
- Richter-Gebert, J., & Kortenkamp, U. (1999). *The interactive geometry software Cinderella*. Springer.



- Ruthven, K., Hennessy, S., & Deaney, R. (2008). Constructions of dynamic geometry: A study of the interpretative flexibility of educational software in classroom practice. *Computers & Education*, 51(1), 297–317.
- Santos, V., & Quaresma, P. (2012). Integrating DGSs and GATPs in an adaptative and collaborative blended-learning Web-environment. In *First Workshop on CTP Components for Educational Software (THedu'11), EPTCS* (Vol. 79, pp. 111–123). <https://doi.org/10.4204/EPTCS.79.7>.
- Santos, V., & Quaresma, P. (2013). Collaborative aspects of the WGL project. *Electronic Journal of Mathematics & Technology*, 7(6). Mathematics and Technology, LLC.
- Santos, V., Quaresma, P., Marić, M., & Campos, H. (2018). Web geometry laboratory: Case studies in Portugal and Serbia. *Interactive Learning Environments*, 26(1), 3–21. <https://doi.org/10.1080/10494820.2016.1258715>.
- Stojanović, S., Narboux, J., Bezem, M., & Janičić, P. (2014). A vernacular for coherent logic. In S. M. Watt, J. Davenport, A. Sexton, P. Sojka, & J. Urban (Eds.), *Intelligent computer mathematics. Lecture Notes in Computer Science* (Vol. 8543, pp. 388–403). Springer International Publishing. [https://doi.org/10.1007/978-3-319-08434-3\\_28](https://doi.org/10.1007/978-3-319-08434-3_28).
- Stojanović, S., Pavlović, V., & Janičić, P. (2011). A coherent logic based geometry theorem prover capable of producing formal and readable proofs. In P. Schreck, J. Narboux, & J. Richter-Gebert (Eds.), *Automated deduction in geometry. Lecture Notes in Computer Science* (Vol. 6877, pp. 201–220). Berlin, Heidelberg: Springer. [https://doi.org/10.1007/978-3-642-25070-5\\_12](https://doi.org/10.1007/978-3-642-25070-5_12).
- de Villiers, M. (2006). Some pitfalls of dynamic geometry software. *Learning and Teaching Mathematics*, 2006(4), 46–52.
- Wang, D. (1995). Reasoning about geometric problems using an elimination method. In J. Pfalzgraf & D. Wang (Eds.), *Automated practical reasoning* (pp. 147–185). New York: Springer.
- Wang, K., & Su, Z. (2015). Automated geometry theorem proving for human-readable proofs. In *Proceedings of the 24th International Conference on Artificial Intelligence, IJCAI'15* (pp. 1193–1199). AAAI Press. <http://dl.acm.org/citation.cfm?id=2832249.2832414>.
- Wu, W. (1984). On the decision problem and the mechanization of theorem proving in elementary geometry. In *Automated theorem proving: After 25 years* (Vol. 29, pp. 213–234). American Mathematical Society.
- Ye, Z., Chou, S. C., & Gao, X. S. (2010a). Visually dynamic presentation of proofs in plane geometry, Part 1. *Journal of Automated Reasoning*, 45, 213–241. <https://doi.org/10.1007/s10817-009-9162-5>.
- Ye, Z., Chou, S. C., & Gao, X. S. (2010b). Visually dynamic presentation of proofs in plane geometry, Part 2. *Journal of Automated Reasoning*, 45, 243–266. <https://doi.org/10.1007/s10817-009-9163-4>.
- Ye, Z., Chou, S. C., & Gao, X. S. (2011). An introduction to java geometry expert. In T. Sturm & C. Zengler (Eds.), *Automated deduction in geometry. Lecture Notes in Computer Science* (Vol. 6301, pp. 189–195). Berlin, Heidelberg: Springer. [https://doi.org/10.1007/978-3-642-21046-4\\_10](https://doi.org/10.1007/978-3-642-21046-4_10).