

Geometry and Education in the Internet Age

Ulrich H. Kortenkamp and Jürgen Richter-Gebert

1. WHAT IS INTERACTIVE GEOMETRY?

In all modern secondary school curriculae geometry plays a major role. Besides its drawing skill training purposes it is used as a tool for teaching mathematical and logical reasoning. Unluckily, the second, much more important goal, is sacrificed for the first, since accurate drawings are needed for correct reasoning.

As an example, let us consider the well-known statement that the three altitudes of *any* triangle meet in a point. In order to convince himself of the validity of this simple statement a student has to draw an example of it. Unfortunately, in most practical cases the three lines will *not* meet in a single point. Due to little drawing inaccuracies the altitudes will miss each other slightly. This is not very instructive; even worse, it confuses instead.

A high level of discipline and practice is necessary to benefit from the understanding a correct drawing can serve. A computer drawing program that supports access to *geometric constructions* can help. It can serve as a *geometric pocket calculator* and relief the student from the routine work. Such a program helps to achieve the second goal that was mentioned above: the thought-stimulating effect of geometry.

2. KEY FEATURES OF INTERACTIVE GEOMETRY SYSTEMS

Interactive Geometry Systems (IGSs) differ from usual computer aided drawing (CAD) software. In contrast to CAD Systems (which are well suited to produce a single static picture) Interactive Geometry Systems consider a drawing as a dynamic entity. They provide not only access to geometric objects like the intersections of lines or perpendiculars to given lines. They, in addition, memorize all *construction steps* that led to a final picture.

As a first application you can use an IGS to produce drawings that are exact with respect to the restrictions a pixel oriented display or printer bears. But in fact you gain a lot more than exactness: Since the IGS keeps track of the whole construction sequence, it can redo the construction for another set of starting points, so you can actually move single points and watch the influence on the rest of your drawing. Going back to our example you are able to verify the fact that the three altitudes meet in any triangle not only for one, but for a large set of triangles. Let us summarize the key features of IGSs.

2.1. Dragging Objects. The “draggability” of complete constructions stresses the most important, because most interactive part of Interactive Geometry Systems. This is more than a tool for getting a “feel for geometry” — which is already significant —, this is even a method for *proving* geometric theorems. A conjectured theorem (like the altitude theorem for triangles) can be explored by moving the free objects (in our case the vertices of the triangle). If the theorem “seems to be true” (that is, it is true for every single drawing one creates), it is generally true with an extremely high probability. Here “high” means, that it is much more likely that the computer fails during execution of the program (due to hardware or software failures), than that the theorem is false. This theoretical background of this method is currently under development by the authors and uses tools from randomized polynomial checking [RGKb, IM83, Sch80]. In addition, under certain conditions a low number of examples (a so called *test set*) is already sufficient to prove the conjectured theorem in general, without depending on randomized proving [Hon86b, Hon86a, DYZ90].

2.2. Macro Operations. In order to facilitate constructions IGSs offer the definition of macro operations. Repetitive tasks like constructing the perpendicular bisector using two circles and their intersections can be bound to a macro that issues the necessary steps automatically. This leads to concepts like *geometric programming languages*. Unfortunately, this also brings up new theoretical problems inherent to this, like

the ambiguity of *the* intersection of two circles [Lab97], or, even worse, the intersection of two conics, which has not been resolved satisfactorily yet [Kor98].

2.3. Loci. While macro operations are a constructional tool, tracing the movement (*Locus*) of a point while moving a free object is a more investigative tool. How Loci are offered to the users differs a lot among the currently available IGSs¹. The spectrum ranges from simple point position recording, which gets erased whenever a new point is moved, along more sophisticated techniques that offer connect-the-dot-algorithms and are able to be viewed under different additional motions, up to almost fully automatic, high-speed drawing routines, that try to find the complete range of positions a dependent object can take. As in the case of macros operations, several ambiguities arise, and to be resolved they require a solid mathematical theory.

2.4. Additional requirements for IGSs. The described features alone are not sufficient for a geometric tool to be applicable in education and research. Let us mention a few additional requirements, which should make a good IGS, but are not available throughout all IGSs.

To use the constructions in publications one needs high-quality output in any resolution (e.g. Postscript) and the possibility to use different colors and styles for the construction elements. When used with presentations (for example with a beamer as a “geometric blackboard”), the screen output must be configurable to give a good visualization of the construction (points must be big enough to be seen!).

On the user interface side the creation of constructions and macros has to be very simple. A *magnetic mode* that automatically detects incidences between points and lines or conics should support the user. The interface has to be configurable; if a teacher wants to allow circle and ruler only as construction tools, nothing else should be available.

And finally, a text-based output and input should be available. This gives the opportunity to check a construction and to fully understand it. The value of this text-based I/O is indisputable, but nevertheless there is no support for it in most IGSs.

3. CINDERELLA’S CAFÉ

Cinderella’s Café is a new IGS we would like to present in this context. It features all the necessary components of an IGS mentioned above, and adds some other features that have not been available before.

3.1. Using most general approaches to geometry. A key component in every geometry system is the mathematics used, e.g. how the dependent objects are calculated. By using the most general approaches possible (as shown below) Cinderella’s Café becomes a more general, simpler and consistent tool for doing geometry on a computer. A general approach to powerful mathematical concepts is the systematic elimination of special cases. This opens the possibility of more user interaction, makes programs more reliable and robust by eliminating possibilities of programming mistakes (“bugs”), and facilitates higher execution speed. We would like to present two examples of this.

3.1.1. Complex Numbers. Consider the following situation: You draw two circles that intersect each other in two points. Next you join these two intersection points by a line (see 3.1), the so-called *radical axis* of the two circles. By moving the circles you can move this line. It will always be perpendicular to the line connecting the centers of the circles and at a distance to each center that reflects the ratio between the radii. But: What should happen in an IGS if you move the circles apart? The two intersections disappear, and so does the radical axis? Or, even worse: You are not allowed to move the circles in this way? *Check it out with your IGS of choice, and find out which approach it takes.*

From a mathematical point of view, one would rather expect the following behavior: *The radical axis, which is still defined for non-crossing circles [CG67], moves consistently and continuously the whole time.* This could be achieved by defining a basic operation “Radical Axis” which does a computation independent from the crossings. However, the desired behavior is inherent to the described construction. An IGS should

¹As of now, there are two major systems offering Loci, *Cabri II* [LB] and *Geometer’s Sketchpad* [Jac], as well as some minor ones.



Figure 1: Correct behavior of the radical axis: The line defined by the two intersections of two circles is still real, even if the two intersections are complex (“vanish”).

be able to detect this without introducing additional concepts. Therefore Cinderella’s Café takes the following approach: Recall that the two crossings are the common solutions of two quadratic polynomials, which happen to be real when there are two *visible* intersection points, and which are complex otherwise. Thus carrying out the calculations in the field of complex numbers always gives us two *complex intersections*, whose connecting line happens to be real (since the two solutions are conjugates of each other), and this is exactly the radical axis.

This small example shows that the introduction of complex number calculations greatly simplifies geometric constructions by eliminating “vanishing” intersections.

3.1.2. Geometric Calculations in Projective Geometry. In the Euclidean Plane almost all pairs of lines have an intersection. More exactly, two lines intersect if and only if they are not parallel. This becomes an issue when one is doing interactive geometry. Two lines which had an intersection might become parallel, when some objects of the construction are dragged. In Euclidean Geometry this introduces another problem: Whenever one calculates the intersection of two lines one has to check whether they are skew or not. But what happens to another line which is defined by this intersection and another point? The line should become a third parallel, but instead (if we stick to the paradigms of euclidean geometry) it disappears, because its defining points are not defined anymore. How can we avoid this dilemma?

Projective Geometry, a powerful theory that has been developed already a hundred years ago, helps us to solve the problem. By embedding the Euclidean Plane into the Projective Plane we add the “points at infinity” lying on the “line at infinity”. First of all this eliminates the necessity of a special treatment parallels. In projective geometry *every* pair of lines has an intersection. When lines become parallel, their intersection lies on the introduced line at infinity. Moreover, the calculations that are needed to obtain the intersection are greatly simplified and unified (and not vice versa, as one could expect). For this we use homogeneous coordinates, i.e. we model the points of the Projective Plane by the one-dimensional linear subspaces of \mathbb{R}^3 , represented by a spanning vector, and we model the lines by two-dimensional linear subspaces, represented by their normals. Note that scalar multiples of these representing vectors are identified (see Figure 3.1.2). Now the calculation of the intersection of two lines reduces to calculating the cross product (or vector product) of two representing vectors in \mathbb{R}^3 . The same holds for the calculation of the line connecting two points [Sto91, Cox92].

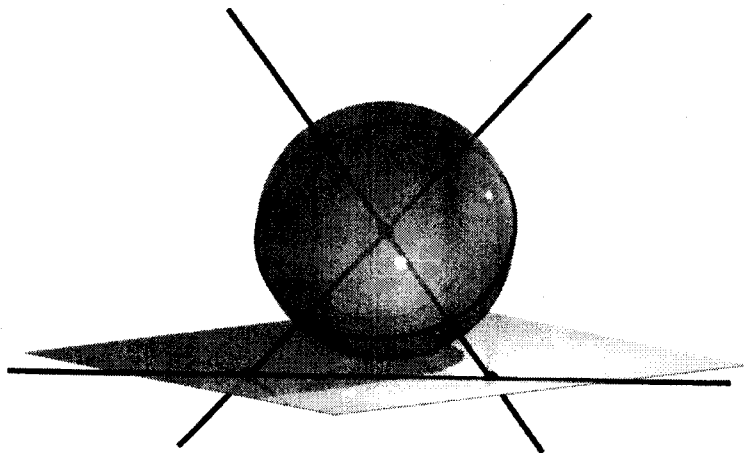


Figure 2: Central projection of points on the sphere onto the plane, giving a double-covering of the sphere by the projective plane. The preimages of points in the plane are pairs of points, while the preimages of lines are great circles.



Figure 3: Exploring conic sections: Screenshot 1 shows a hyperbola projected onto the sphere, together with its asymptotes. Notice the two intersections of the asymptotes at infinity (which is the outermost circle). Screenshot 2 shows the same hyperbola, but now the sphere has been rotated to show that the hyperbola is projectively equivalent to an ellipsoid. For your convenience the line at infinity has been added to the picture.

3.2. Multiple Views. We just sketched how to carry out calculations in the projective plane. How can one visualize the new, non-euclidean objects, or even input them? For this, Cinderella's Café offers the possibility to view a construction in a multitude of different Windows. So it is possible to have an overview of a scene and a detailed view of the same scene (zoom in) at the same time in different Windows. All windows are fully functional, that is, you can move or add elements in either window, and all windows are synchronized, that is, you can see the effects of one operation immediately in all windows.

Since it is not possible to zoom out so much that you can see the elements at infinity, another approach has been taken. It is possible to view a construction in a completely different model, for example, in the “spherical double-covering model of the projective plane”. Here the Euclidean Plane can be imagined as being shifted to the $\{z = 1\}$ -plane in 3-space. Then the complete scene is projected onto the unit sphere.

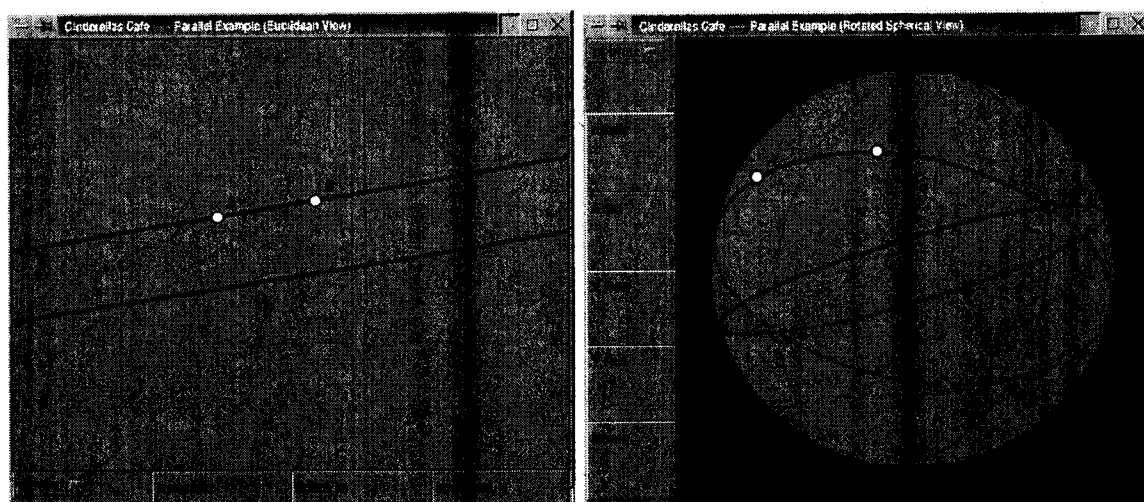


Figure 4: Constructing parallel lines: In the (rotated) spherical view you can see how the parallel in the left picture has been constructed. The sphere has been rotated to show the (thin) line at infinity. The intersection *a.dir* of the line connecting *A* and *B* and the line at infinity has been constructed automatically by Cinderella's Café.

Thus the points at infinity may be found at the equatorial circle at $z = 0$ on the sphere². Cinderella's Café allows all the operations of the euclidean view also in the spherical view. The additional possibility of rotating the view of the sphere simplifies the exploration of geometric objects and constructions (see Fig. 3.1.2).

3.3. Euclidean Geometry support. Since most IGSs are to be used in secondary school teaching it is desirable to have a strong focus on Euclidean Geometry. In the last section we described how to avoid the pitfalls of Euclidean Geometry by using complex Projective Geometry. One might expect that by the introduction of the line at infinity and its associated points we lost the special role of infinity that is needed for “proper” Euclidean Geometry. This is not the case.

We just keep in mind (or in memory) that there is a special line — which does not need any special treatment —, and we can use it to do euclidean constructions. Let us take parallels as an example. When we like to construct a parallel ℓ_P to a given line ℓ we proceed as follows: Take the line at infinity ℓ_∞ , calculate the meet of ℓ with ℓ_∞ , and let ℓ_P be a line through this intersection, chosen specifically to the needed other properties of ℓ_P (see Figure 3.3). We can carry out all the calculations in homogeneous coordinates without any care for special situations. The user is not bothered with these technical details, since all internal operations that are needed to construct the parallel are encapsulated within a macro.

Another example are circles. Since circles depend on euclidean measurements, one might expect that it is not possible to find a projective description of them. However a circle is a special conic, which can be defined by five points. A classical observation of Poincaré shows that all circles have two points in common — which have complex (!) coordinates. So our two mathematical concepts (Complex Coordinates and Projective Geometry) just do fine in modeling Euclidean Geometry [Kle28].

3.4. Hyperbolic Geometry and beyond. Currently, most IGSs provide support for the Poincaré-model of Hyperbolic Geometry only via a sets of macro operations. The multi-view approach of Cinderella's Café allows for a “native” support of Hyperbolic Geometry. In the same way as with spherical views one can add a hyperbolic view that provides full functionality (see Figure 3.3).

By this we can avoid the usual pitfalls in designing macros [Kor98, Lab97]. But we are also given the opportunity to create new views, either traditional ones, or completely new ones. All views share a common computational kernel, and one can concentrate on adding just the translation from abstract objects

²This is the same as identifying the linear subspaces by their intersections with the unit sphere

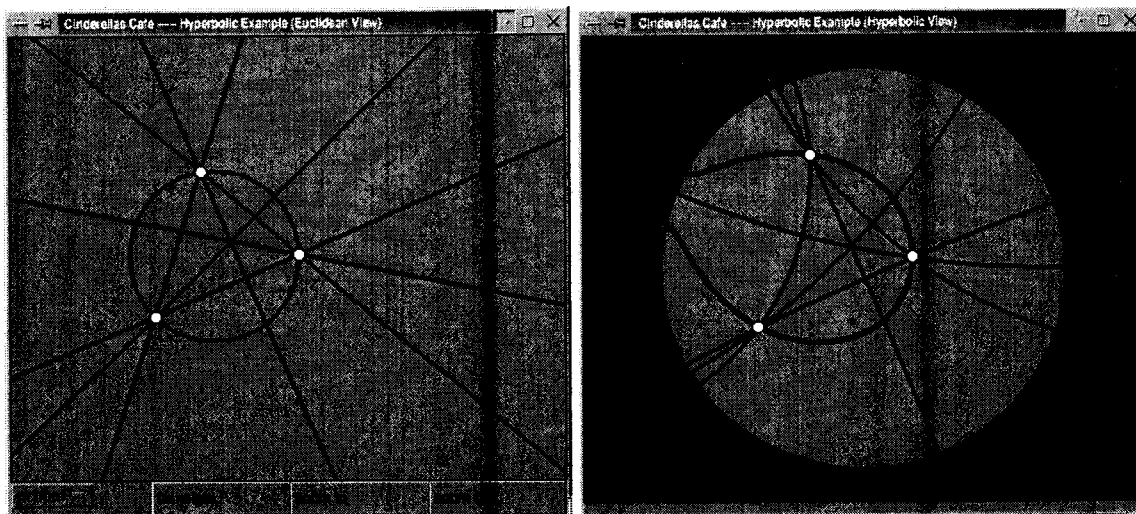


Figure 5: Hyperbolic geometry in Cinderella's Café: Observe that the statement about the heights in a triangle meeting in a point still holds for hyperbolic perpendiculars, while euclidean circles become strange.

to their visualizations. The interface for this will be made public, so it is possible to share new views in the community.

3.5. Measurement of Angles and Distances. Since we mentioned circles, we have to discuss measurements as well. Despite the fact that the educational value of doing measurements within IGS is questioned [Kor98], it is a feature desired by current curriculae.

Any measurement has to be carried out with respect to some scale. Here the point is not that it is possible to zoom into drawings while still having the correct distances. The situation is much worse: The definition of distance and angle is bound to the type of geometry we are using. Since projective geometry is a common basis for doing euclidean, spherical, and hyperbolic geometry (among others), we do not want to restrict ourselves to euclidean measurements. The unified approach to different measurements is Cayley-Klein geometry, as featured by Cinderella's Café.

Cayley-Klein geometry defines angles and distances as multiples of logarithms of certain cross ratios. These cross ratios are calculated with respect to certain fundamental conics, which are the defining conics of the geometry used [Cox92]. So we have a "pluggable" measurement, which supports all our needs (and even more).

The same fundamental conics play a role when we define circles by a center and a point on the circle, or when we define perpendiculars. Both these objects come in different flavors, since they are heavily dependent on the measurement chosen.

While Cinderella's Café supports simple calculations to be performed with the measurements taken, it does not allow to mix different geometric units. So it is not possible to add an angle and a distance, or to assign the area of a triangle to the length of a segment. This is an intended restriction, since it would lead to results which do not reflect any geometric situation.

4. USING THE INTERNET IN EDUCATION

These days there is a great effort to connect schools to the Internet. There are various arguments for that, but there are few high-quality Internet-aware educational tools. Having this in mind, Cinderella's Café has been designed to fill this gap.

4.1. Choosing the right language. Portability and simplicity have been the reasons for choosing JAVA as the development language of Cinderella's Café.

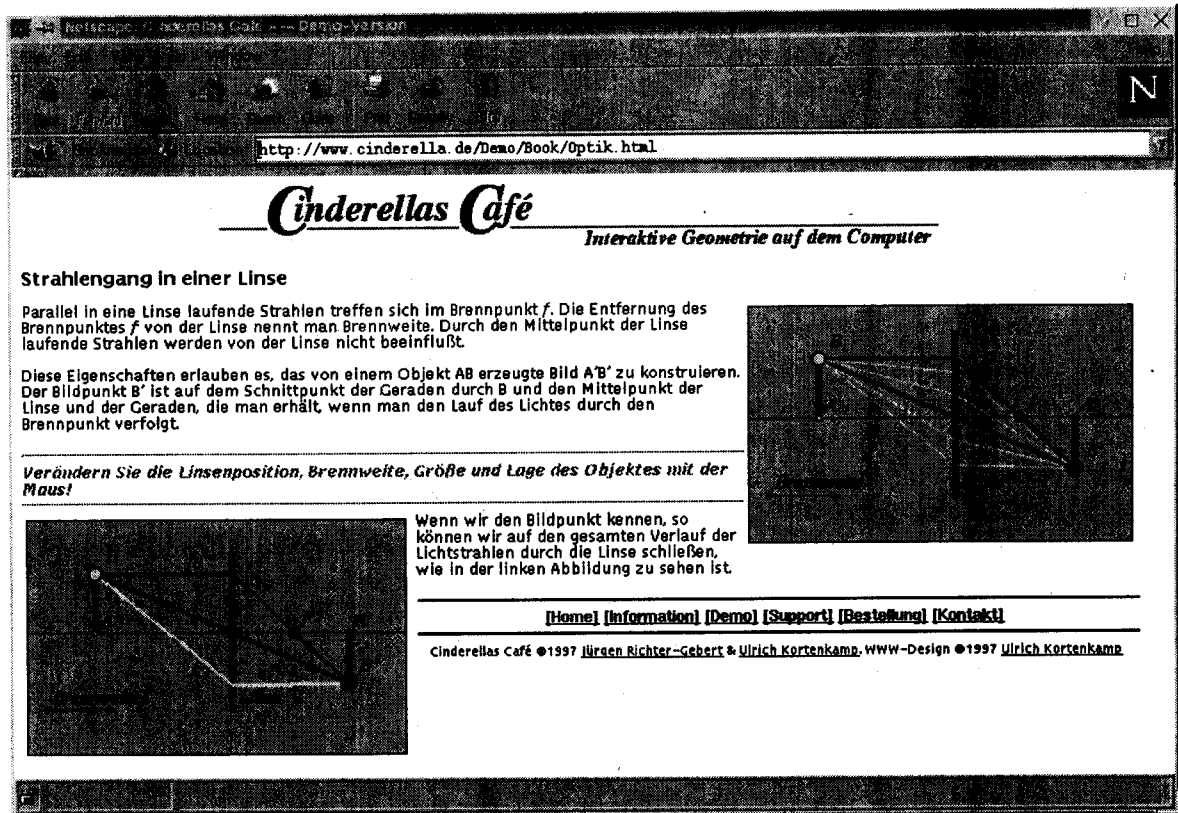


Figure 6

The program has to be highly portable across platforms in order to make it accessible on a wide range of systems, since it should be possible to use it with all the computer systems currently installed in schools, academic institutions, and at home. This has been made possible by using the language JAVA, developed by Sun Microsystems/Javasoftware. This language is available on all major platforms, e.g. Windows 95, Mac OS, Solaris, and Linux.

The execution speed of JAVA was a major concern, but benchmarks showed that the performance is satisfactory even on 486-based systems.

4.2. Instant availability. As an additional benefit we get instant availability of Cinderella's Café via the World-Wide-Web. It is possible to enhance webpages with interactive geometric content without having the user to install any software besides a JAVA-enabled browser. At our website [KRG] you may find some examples (see Figure 4.1) as well as a complete demo-version.

Users of Cinderella's Café are able to create examples and export them directly into their own webpages. So everybody might use Cinderella's Café to enhance his webpages, which may be (interactively) viewed by anyone having access to the Internet. The visitor does not have to install any software for this, and there is no royalty fee.

4.3. Example databases. Currently there is work in progress to build up example databases which reflect the different curriculae. Thus in the near future a teacher or student may benefit from ready-made constructions. This underlines that it makes sense to provide connectivity to schools and other academic institutions, at least if you have software like Cinderella's Café that uses the additional possibilities one encounters.

4.4. Interactive exercises. Another new and exciting feature of Cinderella's Café is the possibility to prepare exercises for geometric constructions. The teacher can construct a sample configuration and mark the set of starting objects and the desired object to be constructed. Then he or she can export the exercise together

with a restricted set of tools. For instance, one can imagine to give the exercise to construct the perpendicular bisector with a ruler and a compass. You only need one construction with Cinderella's Café in order to create an interactive exercise sheet.

The solution of the construction exercise is checked automatically by Cinderella's Café. This non-trivial task, which has not been available in any IGS before, is done via a new randomized theorem prover developed by the authors [RGKb, RG95]. So the student is not bound to the particular construction the teacher had in mind, but is free to find any construction that leads to the desired result. So the program can be used as an educational tool without restrictions on the creativity of the student.

5. CONCLUSION

We presented Cinderella's Café, which is a new IGS with a *high mathematical background*. It supports *highly consistent interactive manipulation* of geometric constructions. This is achieved using the *most general mathematical models* whenever possible, in order to avoid the necessity of special treatment of degenerate cases.

The software package is very well applicable in education and presentation since a lot of features not known to other IGSs is included. Since the program is written entirely in JAVA, it is *highly portable* and *Internet-aware*. The additional possibilities of the network are used to enhance the software by offering *direct export into webpages* and the creation of *interactive exercises*. Thus it gives the Internet the necessary content to make it suited for educational purposes.

REFERENCES

- [Bla47] Wilhelm Blaschke, *Projektive geometrie*, Wolfenbütteler Verlags-Anstalt, 1947, Notdruck.
- [CG67] H.S.M. Coxeter and S.L. Greitzer, *Geometry revisited*, New Mathematical Library, no. 19, The Mathematical Association of America, 1967.
- [Cox87] H.S.M. Coxeter, *Projective geometry*, Springer-Verlag, 1987.
- [Cox92] H.S.M. Coxeter, *The real projective plane*, Springer-Verlag, 1992.
- [CRG94] Henry Crapo and Jürgen Richter-Gebert, *Automatic proving of geometric theorems*, Invariant Methods in Discrete and Computational Geometry (Dordrecht) (Neil L. White, ed.), Caribbean Mathematics Foundation, Kluwer Academic Publishers, 1994, pp. 167–196.
- [DYZ90] Mike Deng, Lu Yang, and Jingzhong Zhang, *The parallel numerical method of proving the constructive geometric theorem*, Theoretical Computer Science (1990), no. 74, 253–271.
- [Gre94] Marvin Jay Greenberg, *Euclidean and non-euclidean geometries*, W. H. Freeman, 1994.
- [Höl94] Reinhard Hölzl, *Im Zugmodus der Cabri-Geometrie*, Ph.D. thesis, Universität Augsburg, 1994.
- [Hon86a] J. Hong, *Can we prove geometry theorems by computing an example?*, Sci. Sinica (1986), no. 29, 824–834.
- [Hon86b] J. Hong, *Proving by example and gap theorems*, Proc. 27th Ann. Symp. Foundations Comput. Sci., 1986, pp. 107–116.
- [HS82] J. Heintz and C.-P. Schnorr, *Testing polynomials which are easy to compute*, Logic and Algorithmic: An international Symposium held in honor of Ernst Specker, Monogr. de l'Enseign. Math., no. 30, 1982, pp. 237–254.
- [IM83] Oscar H. Ibarra and Shlomo Moran, *Probabilistic algorithms for deciding equivalence of straight-line programs*, Journal of the Association for Computing Machinery 30 (1983), no. 1, 217–228.
- [Jac] Nicolas Jackiw, *The Geometer's Sketchpad*, Key Curriculum Press, Berkeley, CA.
- [Kle28] Felix Klein, *Vorlesungen über nicht-euklidische geometrie*, Die Grundlehren der mathematischen Wissenschaften in Einzeldarstellungen, vol. 26, Springer, Berlin, 1928.
- [Kor98] Ulrich H. Kortenkamp, *Algorithmic foundations of interactive geometry*, Ph.D. thesis, TU Berlin, 1998, in preparation.
- [KRG] Ulrich H. Kortenkamp and Jürgen Richter-Gebert, *Cinderella's Café*, <http://www.cinderella.de/>.
- [Lab97] Jean-Marie Laborde, *Exploring non-euclidean geometry in a dynamic geometry environment like Cabri-géomètre*, Geometry Turned On (James King and Doris Schattschneider, eds.), MAA Notes, no. 41, The Mathematical Association of America, 1997, pp. 185–191.
- [LB] Jean-Marie Laborde and F. Bellemain, *Cabri-Geometry II ®*, Texas Instruments, Dallas, TX.
- [RG95] Jürgen Richter-Gebert, *Mechanical theorem proving in projective geometry*, Annals of Mathematics and Artificial Intelligence (1995), no. 13, 139–172.
- [RGKa] Jürgen Richter-Gebert and Ulrich H. Kortenkamp, *On continuity problems in real projective geometry*, in preparation.
- [RGKb] Jürgen Richter-Gebert and Ulrich H. Kortenkamp, *Proving geometric theorems by random examples*, in preparation.
- [Sch80] J.T. Schwartz, *Fast probabilistic algorithms for verification of polynomial identities*, Journal of the Association for Computing Machinery 27 (1980), no. 4, 701–717.
- [Sto91] Jorge Stolfi, *Oriented projective geometry: a framework for geometric computations*, Academic Press, Inc., San Diego, CA, 1991.