MAKING THE MOVE: THE NEXT VERSION OF *CINDERELLA*

ULRICH KORTENKAMP
Institut für Informatik, Freie Universität Berlin, Takustr. 9, 14195 Berlin, Germany
E-mail: kortenkamp@inf.fu-berlin.de

JÜRGEN RICHTER-GEBERT
Lehrstuhl Geometrie und Visualisierung, Technische Universität München, Germany
E-mail: richter@ma.tum.de

Cinderella is a software package for interactive or *dynamic* geometry. Its first version was published in 1999 and was the first geometry software to be based on the theory of complex tracing[3], thus avoiding mathematical inconsistencies and unmotivated discontinuities (modulo numerical errors). At the ICMS 2002 we will introduce the next major version of Cinderella and highlight its new features and new concepts.

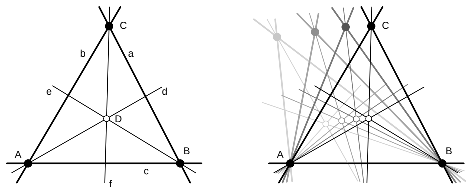
1 The Interactive Geometry Software Cinderella

Let us briefly discuss the main features of the first version of Cinderella[12]. With Cinderella, you can easily create *constructions*, figures consisting of points, lines, circles, conics and other geometric elements, together with relations that describe the mathematical connections between the elements.[5] A standard example is a triangle, consisting of three points and three segments connecting each pair of points, and the altitudes in that triangle. The representation of the construction within the software makes it possible to move the vertices of the triangle (using the mouse) and get a new drawing for each position while the segments and the altitudes are updated accordingly.

The mathematical core of Cinderella is based on projective geometry and Cayley-Klein geometries for maximal generality. All geometric constructions are designed to work in all meaningful situations and to have as few special cases as possible. At the same time it is possible to have different, simultaneous, interactive views of a geometric construction, like its 2D representation, a spherical projection, polarized versions of this or special views like the Poincaré disk for hyperbolic geometry.

As soon as multi-valued geometric constructions, like the intersections of two circles or conics, or angle bisectors, are used, it is necessary to handle these with special care to avoid mathematical inconsistencies. Using a continuation technique on associated Riemann surfaces[4] Cinderella can guarantee – up to

```
1:  A = FreePoint;
2:  B = FreePoint;
3:  C = FreePoint;
4:  a = Line(B,C);
5:  b = Line(A,C);
6:  c = Line(A,B);
7:  d = AngleBisector(b,c);
8:  e = AngleBisector(a,c);
9:  f = AngleBisector(a,b);
10: D = Intersection(d,e);
```



internal representation of the construction sequence | static drawing | idea of the dynamic construction

Figure 1. The basic principle of dynamic geometry software: A construction consists of movable, independent objects (here: the free points A, B, and C, and dependent objects, that are defined using already existing elements. Whenever the position of a free element is changed, the dependent elements are updated accordingly.

numerical errors – that continuous changes of parameters (i.e. free points) will cause continuous changes of the construction (within its realization space). As a consequence we do have a notion of "dynamic theorems", consisting of a sequence of geometric construction steps and an element of the "fiber" over a given assignment of free values as a "germ." The theory behind this gives rise to several mathematical questions that are interesting on their own,[6] but are not covered here.

The intuitive user interface of Cinderella hides most of the mathematical details but exposes the mathematical content of geometry and makes a proper handling of geometry available not only for academic research, but also for teaching on all levels, from K-12 to university. An integrated randomized theorem checking engine is used for computer guided exercises which can be authored with Cinderella. As Cinderella is written in Java, these can be integrated in web pages to create interactive learning environments.

Although Cinderella is a fairly large software package with a lot of functionality, it lacks some essential features that other programs[2,?] offer. Some of them were not implemented because the authors did not have the time or just did not think of them as being necessary, some others are missing because they did not fit into the mathematical framework. For example: As Cinderella is based on complex numbers internally, and even paths of free points within the real numbers are deformed into complex paths, there is no easy way to introduce orientations – it is not possible to order the complex numbers. It was clear that a next version will not be done by just adding missing features, but there had to be new concepts, both mathematically and from a software development point of view.

2 New Features And Concepts

We want to give a rough overview of the new features in the upcoming version of Cinderella – Cinderella 2.0 – and point out some of the difficulties or benefits that come with each of these.

2.1 Transformations

Transformations are at the core of geometry. In fact, the very concept of geometric invariant theory in the sense of Felix Klein is based essentially on the concept of transformations. Cinderella 2.0 introduces transformations not only as operations on geometric objects, but also as geometric objects themselves. It is possible to define translations (with respect to different geometries, e.g. hyperbolic translations), similarities, affine and projective transformations, rotations and others by associating points to their images. These transformations can be composed or inverted – it is possible to work within the full transformation group. All interaction with the transformations is done visually on screen. Any of these can then be easily applied to other geometric objects, e.g. circles. Of course the transformations are themselves dynamic: If their defining objects change, they change, too, as do the images from their application. Even more: the transformations are fully integrated in the automatic proving engine that recognizes whether the same transformation is created in two different ways. This makes it possible to perform calculations in transformation groups.

For an even deeper understanding of the structural behavior of transformations it is possible to create non-deterministic iterated function systems (IFS) by assigning probabilities to them. Using different colors schemes one can visualize the self-similarities and fixpoints.

2.2 Logical Operations

Over the complex numbers two circles always have two intersection points (counted with multiplicity) or they coincide. This is a great way to construct the perpendicular bisector of a segment: take any two circles of the same radius around the endpoints of the segment and draw the line through their intersections. This line will always be a real line, because the two points of intersection are complex conjugates in case they are complex solutions. Thus even if the two circles are "too small" and do not intersect we have a valid construction of the perpendicular bisector (the radical axis). Unfortunately, this is not a desired behavior in K-12 classroom use, as the students rarely know complex numbers, and it might cause more confusion than delight.

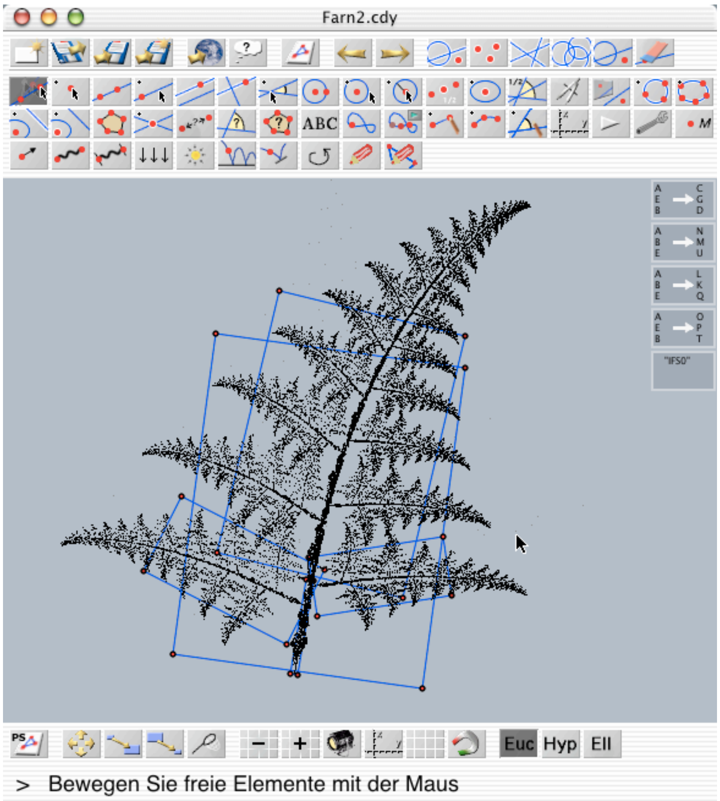


Figure 2. Iterated function systems (IFS) create self-similar structures. Many processes found in nature can be created with IFS, like this fern.

Another situation where elements should vanish although they are valid and real has to be considered with the intersection of line segments. In the former release of Cinderella line segments are “clipped” lines, and as any two distinct lines do have a point of intersection in the projective plane. In order to introduce conditionally vanishing points we had to think of a way to decide whether a point of intersection should appear on the display or not, while still having it accessible for the mathematical kernel – otherwise our concept of theorem could not work.

We ended up with introducing the possibility of showing an element based on a given condition, like “lies on segment S” or “is right of ℓ”. The natural generalization was furthermore to introduce boolean objects that can be composed using logical operations (AND, OR, NOT), making it possible to show parts of construction conditionally in a controlled way. The arbitrary boolean formulas are very flexible and they will prove useful for complex visualizations in education and research.

2.3 Functions

A common request for enhancement is the addition of functions to Cinderella: Based on values gained from measurements or as coordinates of geometric elements it should be possible to calculate new values and feed them back into Cinderella as number objects or as coordinates of other geometric elements.

Here it is very important which functions are asked for. If we could restrict ourselves to polynomials, rational functions, or even single-valued complex functions like sin, cos, exp, this were very easy to add. If we also want to allow multi-valued functions – and we do want to do this, because $\sqrt{\cdot}$ is such a multi-valued function – we can do the same complex tracing as we do with the other multi-valued geometric constructions. This does not cause a major problem, as well.

The theory on which the tracing strategy of Cinderella is based implies that every value is not only continuous in the input parameters, but has to be analytic. So we must not allow functions like the absolute value function, as it is not differentiable in 0. Even complex conjugation, seemingly harmless and trivial to implement, has to be forbidden. Although this has a very sound mathematical explanation, it is a unsatisfying situation for both the users and the authors of the software.

The idea to circumvent this restriction may sound trivial, but it is the key to many other improvements: Instead of doing the calculation for a non-analytically dependent element directly, we move the element to its new position *as if it were done by the user interactively.* By splitting up the chain of dependencies we can allow any – even discontinuous – function to be applied to dependent elements. The only negative effect is that the theorem checking engine will not be able to find theorems that stretch across non-analytic functions. Note however that our notion of dynamic theorem does not even include any statement about such situations; the theory does not cover this at all. So it is quite comforting that we do not find theorems.

2.4 Communication

An issue that has gained attention among math software developers[7] is the ability to exchange mathematical content. There are ways to communicate mathematical content based on its representation; TeX and LaTeX are de facto standards when it comes to encoding math on a computer such that another mathematician can read it. MathML[14] by now has become an accepted standard format for mathematical material, as it is a well-structured format (being based on XML), and its industry support. The main advantage of MathML is that is can be parsed easily to recreate and thus can be used for computer-computer communication protocols. However, MathML is still and will be forever a format for the representation of math, the typesetting, the syntax. Currently, the only way to capture semantic information is the upcoming OpenMath standard as defined by the OpenMath consortium[11].

Cinderella 2.0 does support exchanging OpenMath content, both for functions and also for the upcoming new content dictionaries for plane geometry[1]. This makes it also possible to use Cinderella as a server side extension for web servers, rendering OpenMath to 2D constructions.

Two other built in ways of communication should be mentioned: Cinderella offers a TCP/IP based communication model that can be used to synchronize two constructions via the internet. Two or more people can work on different machines and manipulate the same construction simultaneously. This creates exciting new perspectives for distant teaching and remote research. Also new is a Java API that can be used for custom extensions. Using Cinderella as a visualization and manipulation device it is easy to write Java code demonstrating for example certain algorithms from computational geometry or computer graphics.

The Java API can also be used to interface directly with Mathematica[10] using J/Link or other computer algebra software that is able to execute Java code. The communication is not bound to have Cinderella as visualization component only, but the mathematical kernel and the complex tracing strategy can also be used from the outside, for example for the visualization of Riemann surfaces[4].

2.5 Scripting and Macros

With the former versions of Cinderella there is no easy way to create "macros" or "scripts" – subroutines that automate repeating tasks. This is not only annoying if one wants to create complex constructions consisting of many similar subconstruction, but also from an educational point of view: The proper identification of substructures is a key skill to be learned in mathematics. The next version of Cinderella supports the automatic creation of scripts from a selected part of a construction.

There is also an alternative way of building scripts: A contributed parser[9]

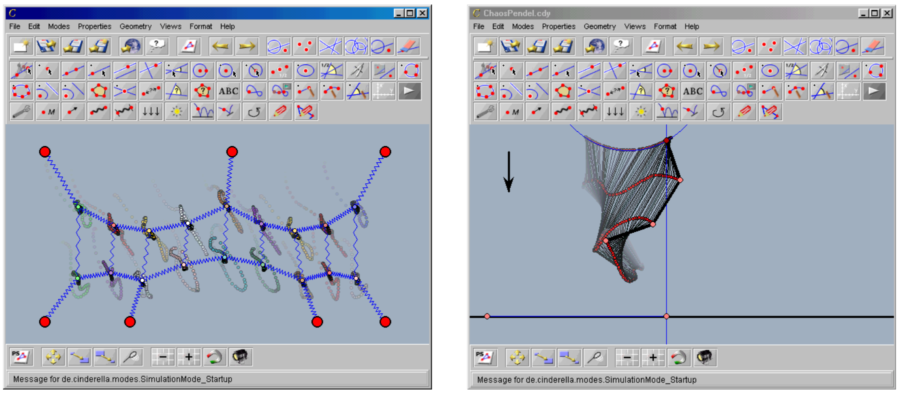


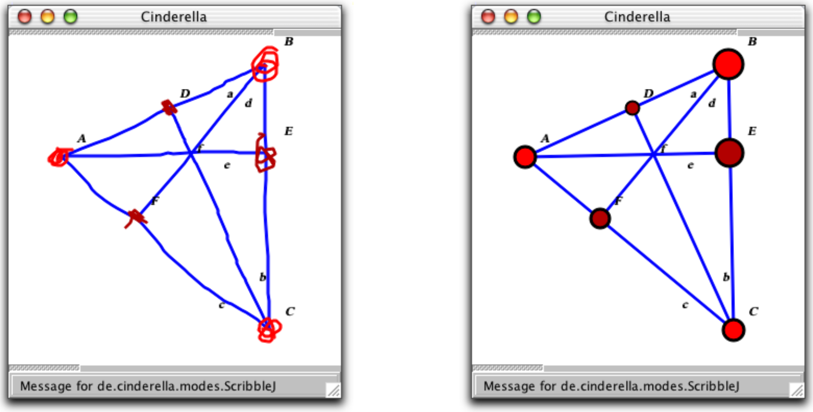
Figure 3. The new physics simulation features of Cinderella enable users to work with springs and other physical objects. On the left you see a network of springs attached to some fixed points, on the right there is a snapshot of a triple pendulum showing chaotic behavior.

is able to "understand" natural language descriptions of geometric constructions and transforms them into dynamic constructions within Cinderella. Currently this works for German language only, but probably the concepts can be extended to other languages as well, with varying success.

2.6 Physics Simulation

During the development of the Mathematica integration (see Sec. 2.4) all examples we did came from physics. This raised the question whether the ability to have basic interaction simulating physical effects should not be included natively into Cinderella. It turned out to be relatively easy to have a qualitatively correct simulation of masses, springs, and gravity, in many cases, but it is hard to have quantitatively correct and a numerically stable simulation, especially for very stiff springs and other extreme values. However, the current system can be used for explorations in physics. It is also possible to use the physical elements for special effects, like creating projectively correct drawings of Schlegel diagrams for 3-dimensional polytopes.

As springs can change their length periodically and masses can rotate around other masses there is even more dynamism and more possibilities for animations than before. You can build apparatus that actually do things, or plug together "animals" like you do with sodaplay[7].



A hand-drawn sketch. . .   . . . and its automatic recognition

Figure 4. The new sketching recognition is useful with special hardware devices, like handhelds with touch screen or interactive whiteboards. Relations like perpendicularity or parallelism can be expressed using extra gestures.

2.7 Sketching

During the development of Cinderella we had the opportunity to test new hardware. First of all, we had access to an interactive whiteboard, a projection screen that is able to track movements of a special pen. It is possible to work on this screen with a computer display projection in a very natural way, the interaction is immediate and accurate. Still, we wanted to make working with Cinderella on this whiteboard be as easy as working with a conventional blackboard. This was our starting point for automatic sketch detection: A hand drawn sketch is interpreted and automatically converted into a construction, i.e. geometric elements that satisfy certain relations. Gestures are used for adding non-trivial content, for example, one can indicate that two lines are parallel by giving both of them a small mark, or two lines "are made" orthogonal by placing an orthogonality-mark at their intersection.

This gesture-based constructing is also very useful on another device, located at the other end of the scale. We were able to create versions of Cinderella that run on handheld devices, e.g. the Casio Cassiopeia E-125 or the SHARP Zaurus SL-5000 series. These handhelds typically have an approx. 3×4 inch touchscreen with a resolution of 240×320 pixels, operated with a pen. The small screen makes it necessary to have as few as possible user interface elements on screen. This makes gesture-based constructing the primary choice for these devices.

3 Further Information

More information about Cinderella and a demo version is available on the WWW at http://www.cinderella-geometry.com.

References

1. A. Cohen, H. Cuypers, U. Kortenkamp, *A content dictionary for plane geometry (tentative title)*, in preparation.
2. N. Jackiw, *The Geometer's Sketchpad*, Key Curriculum Press, Berkeley (1991–2002).
3. U. Kortenkamp, *Foundations of Dynamic Geometry*, Ph.D. thesis, ETH Zürich (1999).
4. U. Kortenkamp, *Visualizing Riemann Surfaces Using Cinderella*, in preparation for VisMath 2002.
5. U. Kortenkamp, J. Richter-Gebert, *A dynamic setup for elementary geometry*, Proceedings of MTCM 2000, Multimedia Tools for Communicating Mathematics, Lisboa, Springer-Verlag, Heidelberg (2002).
6. U. Kortenkamp, J. Richter-Gebert, *Complexity issues in Dynamic Geometry*, Proceedings of the Smale Fest 2000, Hongkong (accepted).
7. U. Kortenkamp, *The Future of Mathematical Software*, Proceedings of MTCM 2000, Multimedia Tools for Communicating Mathematics, Lisboa, Springer-Verlag, Heidelberg (2002).
8. J.-M. Laborde and F. Bellemain, *Cabri Geometry II*, Texas Instruments, (1993–1998).
9. D. Materlik and U. Kortenkamp *A Parser For Natural Language Descriptions of Geometric Constructions*, Technical Report, Freie Universität Berlin, to appear (2002).
10. Mathematica, Wolfram Research Inc., http://www.wolfram.com.
11. The OpenMath Consortium, http://www.openmath.org.
12. J. Richter-Gebert and U. Kortenkamp, *The Interactive Geometry Software Cinderella*, Springer-Verlag, Heidelberg (1999).
13. Sodaplay, http://www.sodaplay.com.
14. W3C Math Home, http://www.w3.org/Math.