Jürgen Richter-Gebert, TU München & Ulrich H. Kortenkamp, FU Berlin

Dynamische Geometrie: Grundlagen und Möglichkeiten

1. Was ist „Dynamische Geometrie“?

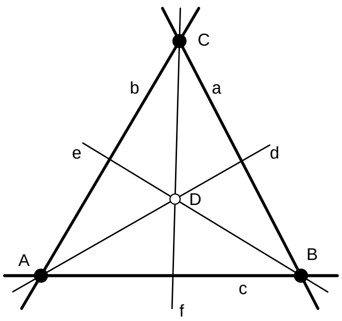
1.1. Die Grundidee

Eine Seite in einem Geometriekapitel eines Mathematikschulbuchs enthält in aller Regel Bilder, welche einen bestimmten geometrischen Sachverhalt visualisieren sollen. Häufig handelt es sich um einfache geometrische Lehrsätze, wie zum Beispiel die Feststellung, dass die Winkelhalbierenden im Dreieck sich in einem Punkt schneiden. Während der geometrische Lehrsatz für beliebige Dreiecke gilt, gibt die Zeichnung nur ein einziges, selten auch mehrere, Beispiele wieder.

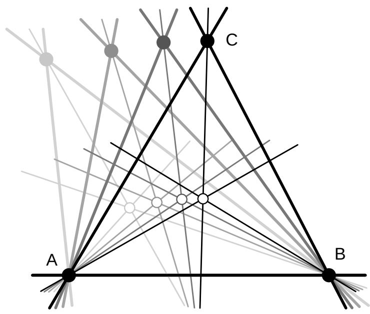
Dynamische Geometrie ermöglicht es nun mit Computerunterstützung den geometrischen „Erfahrungshorizont“ zu erweitern und durch das Bewegung einer Zeichnung Allgemeingültigkeiten zu erkennen. In dynamischer Geometrie ist eine Zeichnung nicht nur ein statisches Bild, sondern wird als Abfolge von Konstruktionsschritten beschrieben, die es ermöglichen, ein neues Bild aus den Koordinaten von Anfangspunkten zu berechnen.

```
1: A=FreierPunkt;
2: B=FreierPunkt;
3: C=FreierPunkt;
4: a=Gerade(B,C);
5: b=Gerade(A,C);
6: c=Gerade(A,B);
7: d=Winkelhalbierende(b,c);
8: e=Winkelhalbierende(a,c);
9: f=Winkelhalbierende(a,b);
10: D=Schnittpunkt(d,e);
```

Konstruktionssequenz



Statisches Bild



Idee des dynamischen Bildes

Hierdurch ist es möglich, nachträglich die Anfangspunkte einer Konstruktion mit der Maus zu „greifen“ und sie in einem so genannten *Zugmodus* zu bewegen, wobei sich die gesamte Konstruktion konsistent mitbewegt. Das Beispiel im obigen Bild stellt eine statische Visualisierung des Winkelhalbierendensatzes der Idee einer dynamischen Visualisierung gegenüber. Die angegebene Konstruktionssequenz beschreibt, wie man ausgehend von den Punkten A, B und C

das Bild des Winkelhalbierendensatzes konstruiert. Liegt eine solche Konstruktionssequenz vor, so kann man die frei wählbaren Punkte nachträglich mit der Maus verschieben. Der Computer berechnet für jede neue Position ein neues Bild und zeigt es an; dadurch entsteht der Eindruck einer fließenden Bewegung.

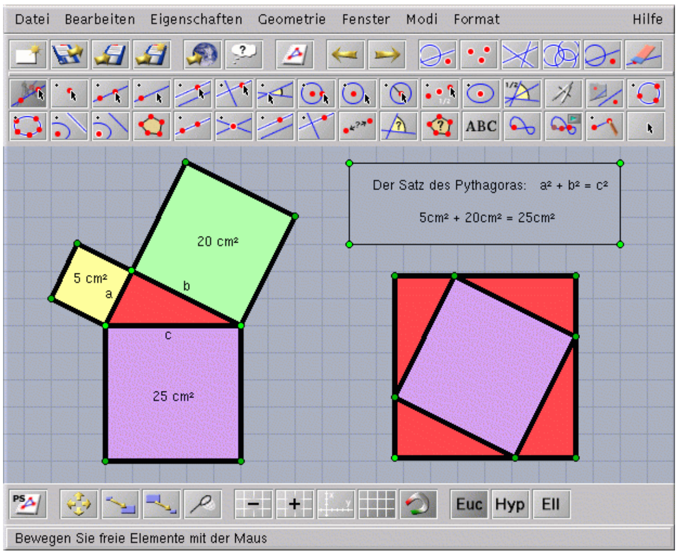
Ein Dynamisches Geometrieprogramm verfügt neben dieser Grundfunktionalität des Zugmodus natürlich auch über Möglichkeiten, Konstruktionen interaktiv zu *erstellen*. Hierzu steht eine Palette von geometrischen Werkzeugen zur Verfügung, die es erlauben, eine bestehende Konstruktion um weitere Elemente zu ergänzen. Zur Grundausstattung der meisten dynamischen Geometrieprogramme gehören hierbei Werkzeuge für Schnittpunkte, Verbindungsgeraden, Parallelen, Senkrechte, Winkelhalbierende und diverse Kreiskonstruktionen. Fortgeschrittenere Programme erlauben auch den direkten Umgang mit Kegelschnitten, Spiegelungen und geometrischen Transformationen.

1.2. Cinderella

Es ist keine neue Idee, geometrische Sachverhalte mit dem Computer „in Bewegung“ zu visualisieren. Bereits Mitte bis Ende der achtziger Jahre erschienen die ersten Programme zur dynamischen Geometrie (*Geometer’s Sketchpad* [2] und *Cabri Geométrè* [4]). Heutzutage gibt es weltweit mehr als 50 Programme mit vergleichbarer Zielsetzung. Bei einer solchen Masse an verfügbaren Programmen liegt die Vermutung nahe, dass längst alle Probleme in diesem Bereich sowohl in mathematischer als auch informatischer Hinsicht gelöst und keine prinzipiell neuen Entwicklungen zu erwarten sind.

Genau das Gegenteil ist der Fall: praktisch alle Programme leiden an ähnlichen konzeptionellen Schwachpunkten, die unweigerlich in gewissen Situationen zu mathematisch unmotivierten oder inkonsistentem Verhalten führen. Erst in jüngster Zeit gelang eine eingehende mathematische Analyse und Erklärung dieser Effekte (siehe z.B. [8, 3, 7]), welche ein Überdenken der bisherigen Implementierungsparadigmen angebracht erscheinen lässt. In manchen Aspekten steht somit das Feld der dynamischen Geometrie mitten in einer „Neudefinition“, bei der grundlegende Fragen der mathematischen Modellierung im Vordergrund stehen.

Noch ein anderer Aspekt hat in den letzten Jahren eine weitere Veränderung des Mainstreams im Bereich dynamische Geometrie bewirkt. Das Internet und die allgegenwärtige Erreichbarkeit von Computern eröffnet die Möglichkeit, dynamische Zeichnungen zur *Kommunikation* von Geometrie einzusetzen. Während noch vor wenigen Jahren proprietäre Datenformate dominierten und Publikationen, die sich dynamischer Visualisierungsmöglichkeiten bedienten, meist mit Begleitdisketten für spezielle Computersysteme ausgeliefert werden mussten, geht heutzutage der Trend in Richtung nahtloser WWW-Integration und standardisierter Dateiformate.



Screenshot einer Cinderella Sitzung

Ausgehend von dem Anspruch ein dynamisches Geometrieprogramm zu schreiben, welches ein Maximum an mathematischer Konsistenz gewährleistet, begannen wir im August 1996 mit der Entwicklung des Programmes *Cinderella* [5, 6]. In einer engen Verflechtung von Grundlagenforschung und Implementierungsarbeit war das Ziel, die Grenzen dynamischer Geometrie auszuloten. Es stellte sich heraus, dass eine mathematisch zufriedenstellende Implementierung Konzepte sowohl aus der *projektiven Geometrie* als auch der *komplexen Analysis (Funktionentheorie)* einbeziehen musste. Diese Theorien müssen in der Implementation berücksichtigt werden, der Benutzer muss aber nicht damit vertraut sein; vielmehr arbeitet die Mathematik im Hintergrund und gewährt einen reibungslosen Ablauf des Zugmodus.

Eine weitere selbst gesteckte Anforderung war eine möglichst nahtlose Internetintegration. *Cinderella* ist daher komplett in Java geschrieben und erlaubt jedem die einfache Einbindung von Zeichnungen als *Applets* in HTML-Seiten, die man ohne weitere Hilfsmittel mit einem gängigen Browser (z.B. *Netscape* oder *Internet Explorer*) betrachten kann. Zusammen ergeben sich ganz neue didaktische Perspektiven: so kann man zum Beispiel Geometrieaufgaben mit vollautomatischer Lernkontrolle anbieten. Die Lernkontrolle wird hierbei vom einem automatischen Beweiser nach Vorgaben des Aufgaben-Autors durchgeführt. Hier ergeben sich neue Einsatzmöglichkeiten im Teleteaching bzw. Distance Learning, aber auch für Lernsoftware im sogenannten Nachmittagsmarkt.

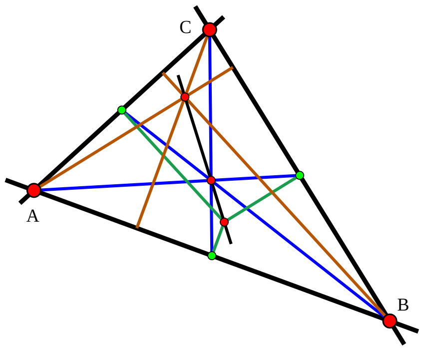
Dieser Artikel soll einen Einblick in die grundlegenden Probleme und Lösungsansätze geben, sowie Anwendungen, die sich aus der Verknüpfung von Geometrie und Internet ergeben, skizzieren.

2. Einsatzszenarien

Zunächst möchten wir die drei klassischen Einsatzfelder dynamischer Geometrie vorstellen. Diese Einsatzmöglichkeiten beschränken sich hierbei nicht allein auf den Bereich des Geometrieunterrichts.

2.1. Erstellen von Zeichnungen

Angenommen, Sie schreiben an einer Publikation wie dieser und wollen sie mit geometrischen Bildern anreichern. Dabei kommt es Ihnen darauf an, einen bestimmten geometrischen Sachverhalt möglichst prägnant in einem statischen (!) Bild darzustellen. Nehmen wir als Beispiel die Tatsache, dass Höhenschnittpunkt, Schnittpunkt der Mittelsenkrechten und Schnittpunkt der Seitenhalbierenden auf einer gemeinsamen Geraden liegen, der *Eulergeraden*.

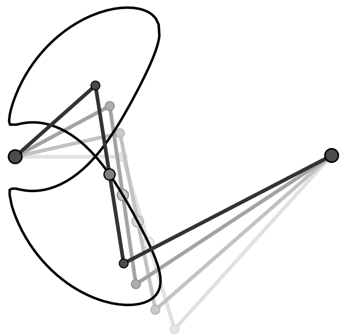


Die Eulergerade: Höhenschnittpunkt, Schnittpunkt der Mittelsenkrechten und Schnittpunkt der Seitenhalbierenden liegen auf einer Geraden.

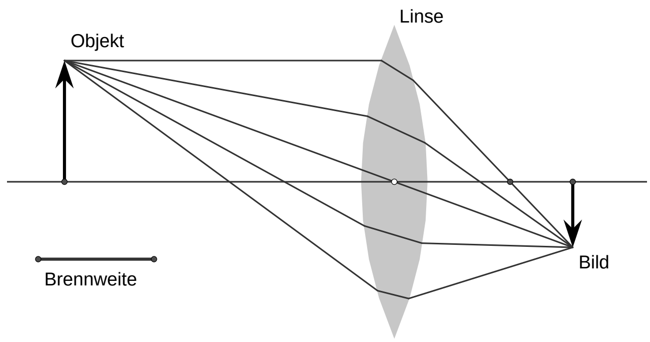
Mit der Wahl der Eckpunkte des Dreiecks liegt das komplette Bild dieser Konstruktion fest. Es erfordert aber einiges an Erfahrung oder Ausprobieren, diese Eckpunkte so zu wählen, dass die Zeichnung zufriedenstellend wird. Bei willkürlicher Wahl der Eckpunkte liegen meist entweder Punkte zu nahe beieinander, Geraden fallen beinahe zusammen oder sind fast parallel, so dass sich ihr Schnittpunkt nicht mehr auf dem Zeichenblatt befindet, etc. Der Einsatz eines dynamischen Geometrieprogramms bietet hier einen großen Vorteil: In einem ersten Arbeitsschritt kann man die Konstruktion erstellen, ohne dabei auf das konkrete Aussehen Rücksicht zu nehmen. Man gewährleistet lediglich, dass die gewünschten geometrischen Sachverhalte durch die Konstruktion fixiert werden. Danach, in einem zweiten Arbeitsschritt sucht man interaktiv nach einer geeigneten Position für die freien Elemente, die insgesamt einen optisch stimmigen Eindruck ergibt. Tatsächlich greifen diese beiden Arbeitsgänge meist ineinander über.

2.2. Geometrisches Experimentierfeld

Der Einsatz dynamischer Geometrie ermöglicht es auch, Zusammenhänge zu erforschen und neue Erkenntnisse zu gewinnen. Ein Geometrieprogramm wird dann als eine Art geometrischer Taschenrechner benutzt, welcher es gestattet, qualitative und quantitative Aussagen über geometrische Sachverhalte zu erlangen. Hierbei muss es sich nicht notwendigerweise um rein mathematische Fragestellungen handeln; auch weite Bereiche der Physik (insbesondere der Mechanik und Optik) lassen sich adäquat auf geometrische Modelle abbilden. So kann man beispielsweise relativ einfach die Kinematik bestimmter Gelenkmechanismen explorieren. Schülern kann man eine vorgefertigte Konstruktion als *virtuelles Labor* zur Verfügung stellen.



Gelenkmechanismus



Strahlengang in einer Linse

2.3. Lernsoftware

In der Vermittlung von Wissen hat sich dynamische Geometriesoftware bereits weitgehend etabliert. Es gibt viel didaktisch aufbereitetes Material, welches sich am Curriculum orientiert, aber auch gänzlich neue Ansätze. Im Abschnitt 4 werden wir auf spezielle Situationen im Detail eingehen.

3. Die Mathematik

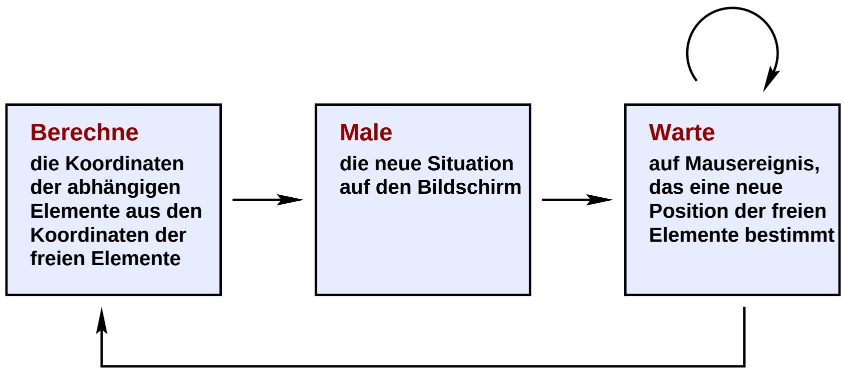
Auf den ersten Blick scheint es keine außerordentlich schwierige Aufgabe zu sein, ein Geometrieprogramm zu schreiben. Der übliche Ansatz sieht in etwa folgendermaßen aus.

1. Man repräsentiert die Elemente durch geeignete Koordinaten (z.B. Punkte durch (x,y)

Koordinaten, Geraden durch die Parameter (a,b,c) einer Geradengleichung $ax+bx+c=0$ und Kreise durch Mittelpunkt (x,y) und Radius r).

2. Man schreibt für jede geometrische Primitivoperation eine Prozedur, die die Parameter eines Ausgabe-Objektes aus den Parametern der Eingabeobjekte berechnet (z.B. errechnet sich die Verbindungsgerade zweier Punkte (x_1,y_1) und (x_2,y_2) aus $a=y_1-y_2$, $b=x_2-x_1$, $c=x_1y_2-x_2y_1$). Dies ist meistens mit Mitteln der Schulmathematik machbar.

3. Man organisiert eine Kontrollschleife, die auf Mausereignisse wartet, feststellt, wenn ein freier Punkt bewegt wird, dessen neue Position bestimmt, dann der Reihe nach die Koordinaten aller abhängigen Elemente berechnet und diese schließlich auf den Bildschirm bringt.



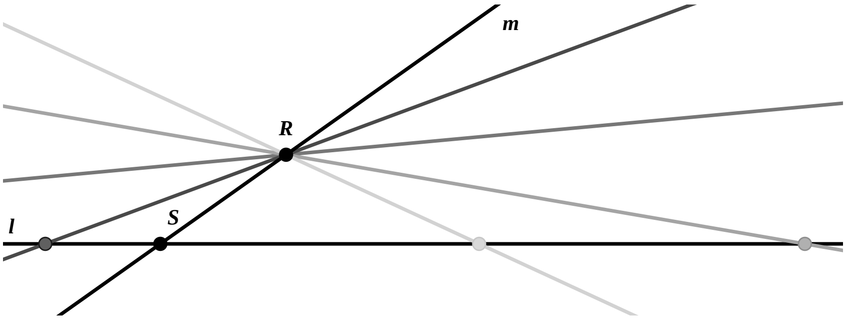
Ablaufdiagramm der Standard-Implementierungsstrategie

Erstaunlicherweise stößt man auf bei einer solchen Implementierung auf größere mathematische Probleme. Nicht nur die Koordinatisierung der Elemente ist komplizierter als oben dargestellt, sondern auch die Berechnung der abhängigen Elemente ist *konzeptionell* wesentlich schwieriger. Tatsächlich kann man nicht einfach „der Reihe nach" aus den Eingabe-Elementen die Ausgabe-Elemente berechnen, wie im übernächsten Absatz erläutert wird.

3.1. Punkte im Unendlichen

Fertigt man auf einem Blatt Papier eine Zeichnung an, so steht nur eine begrenzte Fläche dafür zur Verfügung. Will man den Schnittpunkt zweier Geraden zeichnen, die sich auf dem Blatt nicht treffen, so hat man „Pech" und muss von vorne beginnen. Jeder tatsächlich ausführbare Konstruktionsschritt findet im begrenzten Gebiet des Blattes statt. Bei dynamischer Geometrie ist die Situation grundlegend anders. Es spricht nichts dagegen, einen Schnittpunkt zu berechnen, der außerhalb des sichtbaren Zeichenbereichs liegt. Mehr noch, selbst in einer einfachen Konstruktion, die zunächst komplett innerhalb des Zeichenfensters durchgeführt wurde, können die freien

Elemente nachträglich (!) so verschoben werden, so dass Schnittpunkte beliebig weit draussen liegen.



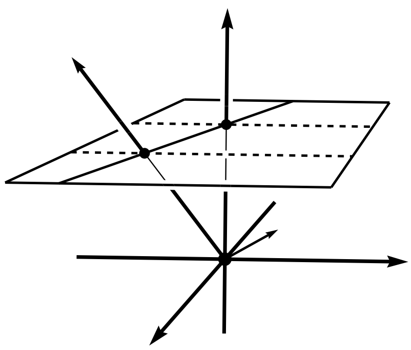
Punkte können durch Drehen der Geraden *m* beliebig weit „weg“ wandern

Betrachten wir den Schnittpunkt S zweier Geraden l und m, wobei die Gerade l festgehalten werden soll und die Gerade m frei um einen Punkt R rotieren darf. Rotieren wir m, so kommen wir an einer bestimmten Stelle zu der Situation, wo m und l fast parallel sind. Der Schnittpunkt befindet sich dann sehr weit außerhalb unseres Zeichenbereiches. Bewegen wir m nun weiter und so dass die Parallelsituation überstrichen wird, so kommt der Schnittpunkt von der anderen Seite zurück. Wie ist die Parallelensituation selbst zu behandeln? In vielen Systemen wird ein solcher „Schnittpunkt“ zweier paralleler Geraden für „ungültig“ erklärt, da man ihn ja ohnehin nicht sieht und eine Berechnung zu einer Division durch Null führt. Was aber, wenn an diesen Punkt weitere Konstruktionselemente geknüpft sind (z.B. eine Gerade durch S und einen weiteren Punkt A)? Sollen diese dann auch für ungültig erklärt werden? Oftmals kann es vorkommen, dass S nur die Rolle eines Hilfspunktes einer Konfiguration spielt. In speziellen Situationen würden somit ganze Teile einer Konstruktion für ungültig erklärt. So wird es notwendig, eine große Menge von Spezialfällen zu behandeln (und vorherzusehen!).

Es gibt hierfür einen mathematisch befriedigenderen Ansatz: Man erweitert die Ebene um „Punkte im Unendlichen“. Zu jeder Parallelenschar führt man genau einen solchen Punkt ein, welcher per Definition auf allen Geraden dieser Schar liegt. Auf diese Weise haben zwei Geraden immer einen Schnittpunkt. Sind die Geraden parallel, so ist deren Schnittpunkt der zu diesen Parallelen gehörige Punkt im Unendlichen. In genau der gleichen Weise erreicht man, dass je zwei Punkte immer eine Verbindungsgerade haben. Man muss dazu lediglich noch eine weitere *Gerade im Unendlichen* zulassen, auf der alle unendlich fernen Punkte liegen. Die Verbindungsgerade zweier endlicher Punkte ist nach wie vor deren (klassische) Verbindungsgerade. Die Verbindungsgerade eines endlichen und eines unendlichen Punktes ist die Gerade durch den endlichen Punkt, die in Richtung der zum unendlichen Punkt gehörigen Parallelen zeigt. Die Verbindungsgerade zweier unendlicher Punkte ist die Gerade im Unendlichen. Dies ist das System der *Projektiven Geometrie*, welches in der ersten Hälfte des letzten Jahrhunderts von Geometern wie Plücker, Monge und Poncelet entwickelt wurde.

Durch die Erweiterung der Geometrie um unendlich ferne Elemente gelingt es, die Geometrie von vielen lästigen Spezialfällen zu befreien, die ansonsten gesondert behandelt werden müssten. Mehr noch, es ist sogar möglich, diese so erweiterte Geometrie rechnerisch sehr einfach zu behandeln, so dass insgesamt die darauf basierenden Programme kürzer, allgemeiner und stabiler werden. Auch für Kreise und Kegelschnitte kann man entsprechende projektive Darstellungen einführen und mit ihnen rechnen.

Wir wollen kurz beschreiben, wie die rechnerische Darstellung konkret aussieht. Statt einen ebenen Punkt durch zwei Koordinaten (x,y) zu beschreiben, stellen wir ihn durch drei Koordinaten (x,y,z) dar, wobei Ausdrücke die sich nur um ein Vielfaches unterscheiden den selben Punkt darstellen sollen. Also repräsentieren $(2,3,1)$ und $(6,9,3)$ denselben Punkt. Alle Vektoren, die den gleichen Punkt repräsentieren, liegen auf einer Geraden durch den Koordinatenursprung. Die alten Punkte (x,y) entsprechen in der neuen Darstellung den Vektoren $(x,y,1)$. Vektoren der Form $(x,y,0)$ spielen die Rolle der Punkte im Unendlichen.



Einbettung der euklidischen Ebene in den Raum

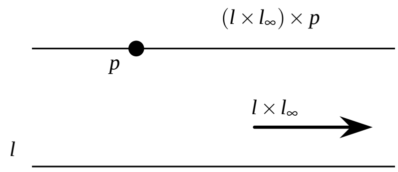
Die gewöhnliche euklidische Zeichenebene wird somit räumlich als $z=1$-Ebene eingebettet. Ein Punkt in dieser Ebene wird als Verbindungsgerade des Punktes mit dem Koordinatenursprung dargestellt. Wandert ein Punkt weit nach draussen, wird diese Gerade zunehmend horizontal (mit Bezug zur xy-Ebene). Die im Grenzfall entstehenden horizontalen Geraden entsprechen genau den unendlich fernen Punkten.

Geraden werden weiterhin durch die Parameter (a,b,c) der Geradengleichung $ax+by+c=0$ dargestellt. In diesem Bild liegt also ein Punkt (x,y,z) auf einer Geraden (a,b,c) genau dann, wenn $ax+by+cz=0$ gilt. Die Rolle der Geraden im Unendlichen spielt der Vektor $(0,0,1)$. Will man die Verbindungsgerade zweier Punkte $p_1=(x_1,y_1,z_1)$ und $p_2=(x_2,y_2,z_2)$ bestimmen, so berechnet man einfach deren Kreuzprodukt $p_1\times p_2=(y_1z_2-z_1y_2,z_1x_2-x_1z_2,x_1y_2-y_1x_2)$. Dies geht so, weil für das Kreuzprodukt $(a,b,c)=p_1\times p_2$ die Beziehungen $ax_1+by_1+cz_1=0$ und $ax_2+by_2+cz_2=0$ gelten.

Von den oben aufgeführten drei Implementierungsschritten betrifft die projektive Denkweise insbesondere die ersten beiden. Punkte, Geraden und Kreise sollten in speziellen, auf die projektive Denkweise abgestimmten Koordinatenrepräsentationen vorliegen. Die Berechnung der Grundoperationen muss an diese Darstellung entsprechend angeglichen werden. Unter Verwendung der projektiven Denkweise gestalten sich solche Operationen oftmals wesentlich einfacher. Als weiteres Beispiel sei die Berechnung einer Parallelen q zu einer Geraden l durch einen Punkt P genannt: Es sei zunächst $l_\infty = (0,0,1)$ die Gerade im Unendlichen. Dann gilt

$$q = (l \times l_\infty) \times p.$$

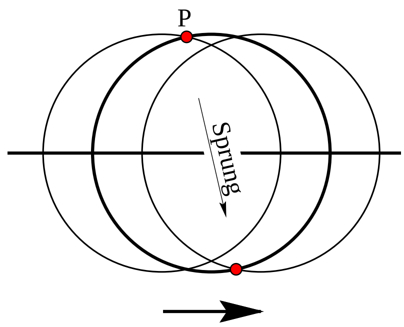
Mittels $l \times l_\infty$ wird zuerst der Fernpunkt in Richtung l bestimmt (der Schnitt von l mit der Geraden im Unendlichen). Sodann wird die Verbindungsgerade von diesem Fernpunkt mit Punkt p bestimmt. Dies ist die gesuchte Parallele.



Berechnung der Parallelen zu l durch p

3.2. Springende Punkte

Die Verwendung projektiver Ansätze und die Einbeziehung unendlich ferner Punkte wird außer bei Cinderella auch bei einigen anderen Programmen eingesetzt (z.B. bei Cabri Geométrè). Es gibt allerdings noch einen weiteren, mathematisch unmotivierten und die Benutzung der Software erschwerenden, und somit unerwünschten Effekt, der speziell dynamischer Natur ist. Betrachtet man in einem herkömmlichen dynamischen Geometrieprogramm eine etwas umfangreichere Konstruktion so kann man oftmals beobachten, dass während man einen der freien Punkte im Zugmodus bewegt, plötzlich ganze Teile der Konstruktion unerwartet von einer Position an eine andere „springen“. Oftmals repräsentiert die Zeichnung nach einem solchen Sprung nicht mehr die gewünschte Konstruktion.

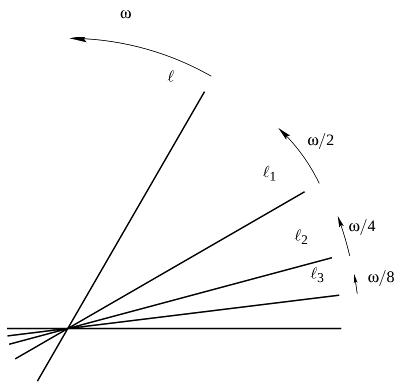


Springendes Verhalten in herkömmlichen Systemen

Die folgende einfache Konstruktion kann als Indikator für solches Verhalten dienen: Man zeichnet zunächst zwei entlang einer Geraden verschiebbare Punkte, um die man zwei Kreise mit gleichem Radius schlägt. Nun markiert man einen der beiden Schnittpunkte der beiden Kreise. Bewegt man den Mittelpunkt des einen Kreises entlang der Geraden durch den Mittelpunkt des anderen Kreises, so springt der Schnittpunkt von der einen auf die andere Seite der Gerade. Wären an den Schnittpunkt noch weitere Konstruktionselemente gebunden, so würden diese ebenfalls mit dem Punkt spontan die Position wechseln.

Die Ursache für dieses Verhalten ist erstaunlich tiefliegend und erfordert Methoden der *komplexen Analysis* zu deren Behebung. Das beschriebene Sprungverhalten kann immer dann auftreten, wenn eine Operation mehrdeutige Interpretationen zulässt. So haben in unserem Beispiel die beiden Kreise nicht nur einen Schnittpunkt sondern deren zwei. Für jedes neu zu berechnende Bild muss das Programm (während Schritt 2 der obigen Liste) entscheiden, welcher der beiden Punkte nun gemeint sein soll. Fällt es diese Entscheidung falsch, so treten Sprünge auf.

Wir wollen uns unter Vernachlässigung aller formaler Details kurz klarmachen, warum in jedem System das naiv der oben angegebenen Implementierungsstrategie folgt, unvermeidlich solche Sprünge auftreten müssen. Laut Schritt 2 und 3 muss das Programm zu einer Neuberechnung des Bildes lediglich die Koordinaten der freien Elemente der Reihe nach heranziehen. Also führen unter diesem Paradigma identische Positionen der freien Elemente immer zu identischen Bildern. Betrachten wir nun die folgende Konstruktion: Wir beginnen mit einer Geraden ℓ, die frei um den Koordinatenursprung rotieren kann, und konstruieren die Winkelhalbierende ℓ_1 mit der x-Achse. Sodann konstruieren wir die Winkelhalbierende ℓ_2 von ℓ_1 mit der x-Achse, und so fort.



Iterierte Winkelhalbierende

Bewegen wir nun im Zugmodus die Gerade ℓ mit der Maus und rotieren diese mit Drehgeschwindigkeit ω um den Ursprung. Die Gerade ℓ_1 läuft dann mit Geschwindigkeit $\omega/2$, die Gerade ℓ_2 mit Geschwindigkeit $\omega/4$, usw. Dies bedeutet aber, dass wenn ℓ eine volle Drehung gemacht hat und wieder an ihre ursprüngliche Position zurückgekehrt ist, die Gerade ℓ_2 nur eine Vierteldrehung vollführt haben kann. Mit anderen Worten: wollen wir in diesem Beispiel Sprünge vermeiden, so darf es nicht sein, dass die gleichen Koordinaten für die freien Elemente immer zum gleichen Bild führen. Auch bei der Winkelhalbierenden handelt es sich um eine mehrdeutige Operation, denn zu zwei Geraden gibt es zwei (senkrecht aufeinander stehende) Winkelhalbierende.

Ein wesentliches Element, um dieses Problem aufzulösen, besteht in der Einbeziehung der Vorgeschichte einer Konfiguration. Man kann nicht allein aus den Koordinaten der freien Punkte das Bild berechnen; man muss die vorherigen Positionen mitberücksichtigen. Weiterhin verlangt die Auflösung einen sensiblen Umgang mit verschwindenden Schnitten, die z.B. auftreten, wenn zwei Kreise, deren Schnitt gebildet wurde, voneinander weg bewegt werden. Verschwindende Schnitte kann man durch die Einbeziehung *komplexer Zahlen* weiterverarbeiten. Die dritte, schwere, Hürde ist die korrekte Behandlung von *Singularitäten*, die dann auftreten, wenn die Mehrfachlösungen einer Berechnung aufeinanderfallen (zum Beispiel die Schnittpunkte einer Geraden mit einem Kreis wenn die Gerade in die Tangentialsituation bewegt wird).

4. Dynamische Geometrie und das Internet

Wir wollen nun auf die speziellen Möglichkeiten, die der Einsatz interaktiver Geometrie in Verbindung mit Internettechnologie bietet, eingehen. Diese Szenarien wurden teilweise bereits in *Cinderella* umgesetzt, teilweise befinden sie erst sich im Erprobungsstadium.

4.1. Kommunikation

In der jetzigen Form gestattet *Cinderella* einen extrem einfachen Export dynamischer Zeichnungen und Animationen aufs Internet: Eine Konstruktion kann als interaktive HTML-Seite[1] abgespeichert werden. Das Java-basierte und streng modulare Design von *Cinderella* ermöglicht es nun, dass im Internet-Browser die Zeichnung als vollwertige dynamische Konstruktion dargestellt wird. Der Betrachter kann mit der Maus die Konstruktion auf der Seite im Zugmodus bewegen. Auf der Seite kann man bei Bedarf zusätzlichen Text zur Erklärung der Zeichnung einbauen. In der Praxis wird dies oftmals ausgenutzt um geometrische Zusammenhänge distanzunabhängig zu kommunizieren oder zu dokumentieren. Die schnelle Kommunikation komplexer Zusammenhänge hat sich hierbei oftmals als sehr fruchtbar erwiesen – ein Bild sagt mehr als tausend Worte, ein bewegliches Bild sagt noch mehr.

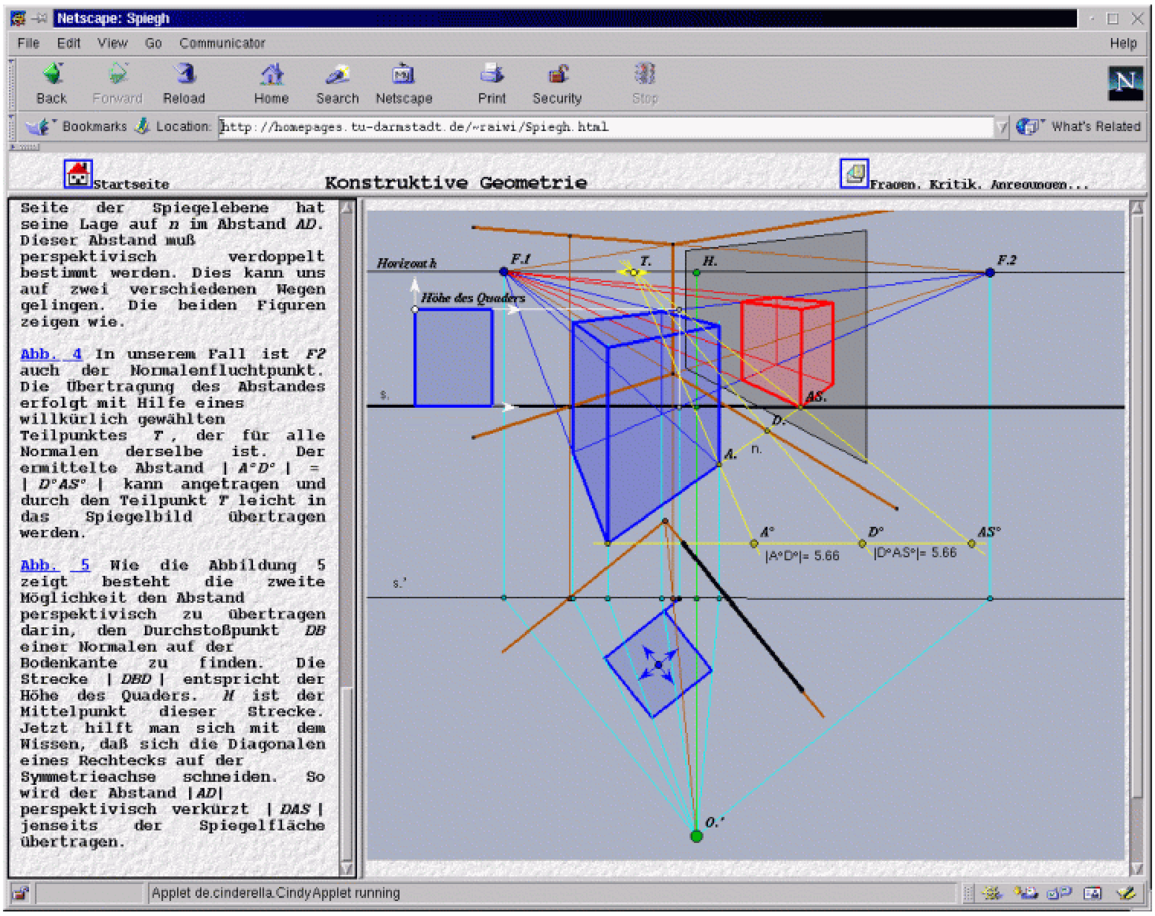
Diese Art der Kommunikation entspricht der Kommunikation durch Brief oder FAX. Es gibt weder einen Rückkanal noch kann die bereits „verschickte“ Konstruktion nachträglich verändert werden (sie kann nur durch eine neue Konstruktion ersetzt werden). Ein aktuelles Weiterentwicklungsprojekt von Cinderella ermöglicht eine telefonartige Kommunikation zwischen zwei Cinderella-Benutzern: Hier können zwei Benutzer gemeinsam an einer Konstruktion arbeiten, und die jeweiligen Änderungen werden synchron über ein Netzwerk ausgetauscht. Als Ergänzung dienen Audio- oder Videoverbindungen über das gleiche Netzwerk.

4.2. Interaktive Bücher

Natürlich braucht sich Kommunikation nicht auf einen Betrachter zu beschränken. Eine naheliegende Anwendung der Exportmöglichkeiten ist die Erstellung von interaktiven Büchern oder von interaktivem Buchbegleitmaterial. Tatsächlich wird diese Möglichkeit bereits von einigen *Cinderella*-Benutzern (und von Benutzern anderer exportfähiger Geometrieprogramme) rege genutzt. Ein sehr schönes Beispiel ist hier das von Rainer Witt aus Darmstadt erstellte Tutorial zur darstellenden Geometrie [9]. Dort werden mehrere komplexe Konstruktionen des perspektivischen Zeichnens schrittweise erläutert und durch entsprechende interaktive Beispiele illustriert. Der Anwendungsbereich interaktiver Bücher beschränkt sich nicht allein auf Geometrie, sondern erfasst auch Gebiete wie Optik, Mechanik oder Architektur.

Zusätzlich besteht die Möglichkeit, einem Benutzer auch auf einer HTML-Seite Konstruktionswerkzeuge zur Verfügung zu stellen und ihn Zeichnungen ergänzen zu lassen.

[1] HTML ist das Standardformat für WWW-Seiten und wird von Internet-Browsern verstanden

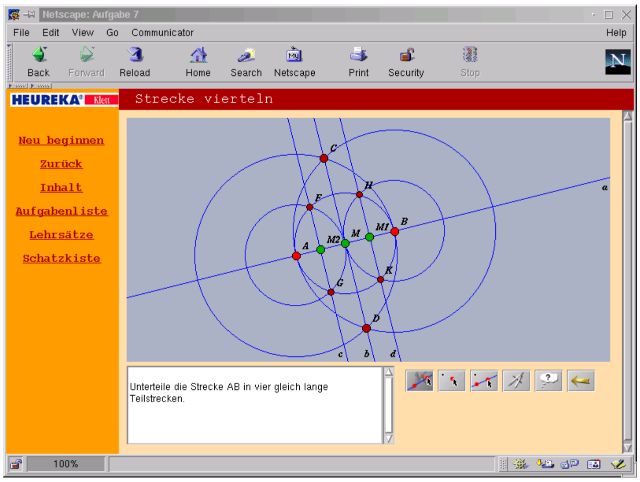


Seiten zur Darstellenden Geometrie von Rainer Witt [9]

4.3. Aufgaben mit Lernkontrolle

Mit der aktuellen Version von *Cinderella* können (Geometrie-)Aufgaben erstellt werden, die ohne direkte Lehrerpräsenz bearbeitet werden können. *Cinderella* verfügt über einen Mechanismus, welcher es ermöglicht, automatisch und zuverlässig zu prüfen, ob zwei auf verschiedene Weise konstruierte Elemente einander identisch sind. Damit kann man zum Beispiel geometrischer Lehrsätze automatisch überprüfen lassen. Dies basiert wesentlich auf der vorher erwähnten mathematischen Theorie, welche das Springen von Elementen verhindert.

Man kann diesen Mechanismus aber auch dazu verwenden, Aufgaben mit automatischer Lernkontrolle zu exportieren. Hierbei stellt der Autor eine Aufgabe, beispielsweise das Konstruieren der Mittelsenkrechte zweier Punkte. Er konstruiert zunächst eine Musterlösung für diese Aufgabe. In einem speziellen, in Aufgabeneditor markiert er die Startelemente und die Lösungselemente der Aufgabe, schreibt Begleittexte, legt die Werkzeuge fest, mit denen der Schüler arbeiten darf und exportiert die Aufgabe.



Gelöste Aufgabe „Strecke Vierteln“ mit Lernkontrolle

Der Schüler erhält ein HTML-basiertes Arbeitsblatt, welches sowohl die Zeichenfläche mit den Startelementen zeigt, als auch den Aufgabentext und die zur Verfügung stehenden Werkzeuge. Die Musterlösung ist vollkommen verborgen. Der Schüler beginnt nun mit Hilfe der Werkzeuge die Aufgabe zu lösen. Hat er das Zielelement der Aufgabe (in unserem Beispiel die Mittelsenkrechte) konstruiert, wird er vom Programm mit einem entsprechendem Text „belohnt“. Hierbei muss der Schüler keineswegs der Musterlösung des Lehrers folgen. Der automatische Beweiser stellt sicher, dass solche Konstruktionen, die immer (!) mit dem Zielelement der Musterlösung übereinstimmen, automatisch als richtig erkannt werden. Es kommt also bei der Aufgabe letztlich nicht auf den Lösungsweg, sondern lediglich auf die Korrektheit des Ergebnisses an. Der Aufgabeneditor gestattet darüber hinaus auch, eine Aufgabe mit Hinweistexten zu versehen, die dem Schüler eine Orientierung entlang eines bestimmten Lösungsweges bieten.

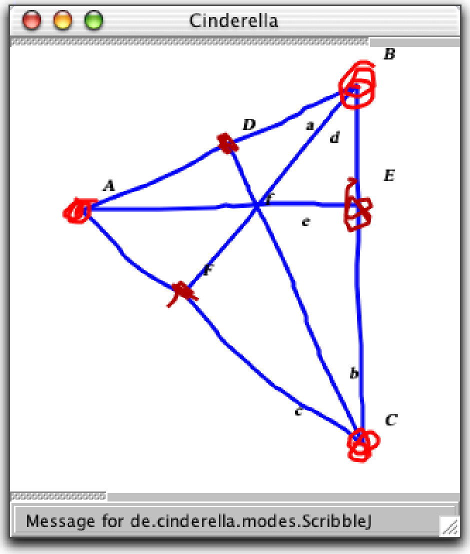
Das in *Cinderella* umgesetzte Aufgabenkonzept wurde bereits in Schul-versuchen eingesetzt und stiess dabei immer wieder auf eine sehr positive Resonanz von Seiten der Schüler. Die Schulversion [6] ist bereits mit einer Sammlung von interaktiven Aufgaben auf CD versehen. Der britische Lehrer Bryan Dye gestaltete unabhängig davon eine online abrufbare Aufgabensammlung zum Thema Rotation, Spiegelung, Translation [1].

4.4. Vernetzter Klassenraum

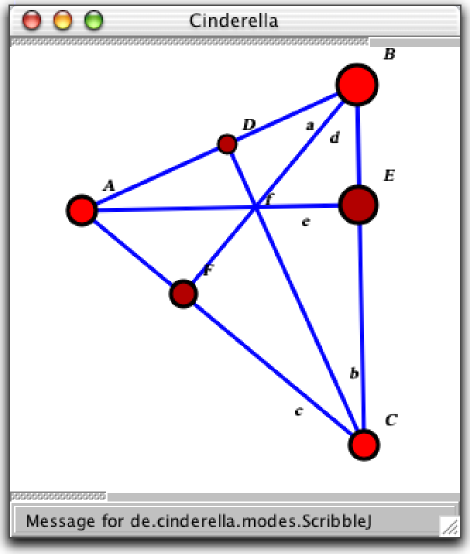
Neue Hardware schafft derzeit die Voraussetzungen für ein Projekt, welches vor wenigen Jahren noch nicht durchführbar gewesen wäre. Funknetzwerke (Wireless Local Area Networks, WLAN) können mit kostengünstiger Technologie aufgebaut werden und somit eine Verbindung zwischen Schüler- und Lehrerrechnern herstellen. Die andere Innovation stellen elektronisch abgetastete

Tafeln, die zusammen mit einer Computerprojektion die Steuerung der Maus übernehmen, dar: Damit kann ein Lehrer für alle sichtbar und nachvollziehbar – da der Zusammenhang zwischen der Bewegung des Lehres und der Maus hergestellt wird – Computerprogramme bedienen.

Die bereits erwähnte „Telefonversion“ von Cinderella lässt es zu, Zeichnungen der Schüler auf der Tafel darzustellen und weiterzuverabeiten. Schüler können auch direkt vom Platz aus ihre Ergebnisse vorführen.



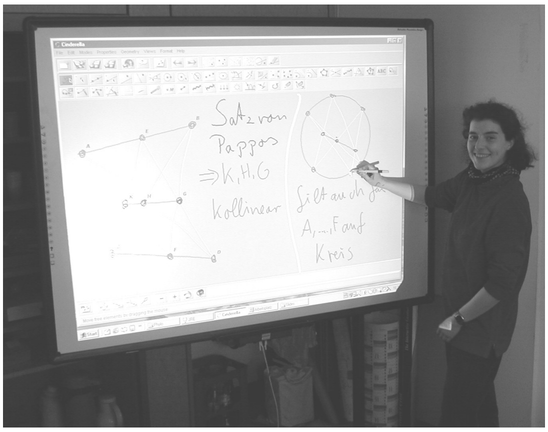
Skizze des Höhensatz



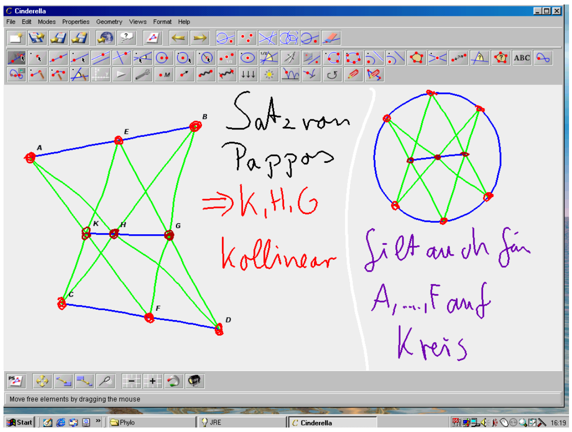
Automatische Umsetzung in eine Konstruktion

Diese Situation stellt auch neue Anforderungen an das Userinterface für dynamische Geometrie: Das Arbeiten an der Tafel soll für den Lehrer oder die Lehrerein (und für Schüler) keine Umstellung zur herkömmlichen Arbeitsweise bedeuten. Eine Zeichnung samt ihrer geometrischen Information soll möglichst in einem Arbeitsgang erstellt werden können, ohne dass es nötig ist, die „Werkzeuge“ zu wechseln. Ein Beispiel: Eine Skizze eines Dreiecks mit seinem Umkreis soll *automatisch* in eine Konstruktionssequenz umgesetzt werden, die drei Punkte, die 3 Verbindungssegmente und den durch die drei Punkte definierten Kreis enthält. Danach soll es möglich sein, einen der Punkte zu verschieben, und die Skizze wird an die neue Position angepasst.

An der elektronischen Tafel ist mit einem solchen Userinterface ein intuiveres Arbeiten möglich; es hat aber auch entscheidende Vorteile für die Schülerrechner. Anstatt die Schüler mit eigenen, teuren und empfindlichen Computern auszustatten, kann man moderne Kleincomputer (PDAs) einsetzen. Die geringe Größe und Auflösung des Bildschirms (typischerweise 240$\times$320 Bildpunkte) kann mit einer „Skizziereingabe“ voll ausgenutzt werden, ohne dass wertvoller Platz für Bedienelemente verbraucht wird. Ein solches Lehr-/Lernszenario wird dieses Jahr erstmals auf der CeBit in Hannover vorgestellt.



Arbeiten an der interaktiven Tafel



Screenshot des Tafelbildes

Literatur

[1] B. Dye, *Transformations*, `http://www.anglia.co.uk/education/mathsnet/campus01/topindex.html`, AngliaCampus, 1999.

[2] N. Jackiw, *The Geometer's Sketchpad*, Key Curriculum Press, Berkeley, 1991–1995.

[3] U. Kortenkamp, *Foundations of Dynamic Geometry*, Dissertation, ETH Zürich, 1999, `http://www.inf.fu-berlin.de/~kortenka/Papers/diss.pdf`.

[4] J.-M. Laborde und F. Bellemain, *Cabri-Geometry II*, Texas Instruments, 1993–1998.

[5] J. Richter-Gebert & U. Kortenkamp, *Cinderella - The interactive geometry software*, Springer 1999; siehe auch `http://www.cinderella.de`.

[6] J. Richter-Gebert & U. Kortenkamp, *Cinderella - Die interaktive Geometriesoftware*, HEUREKA Klett, 2000.

[7] J. Richter-Gebert & U. Kortenkamp, *Complexity issues in Dynamic Geometry.* In: *Proceedings of the Smale Fest 2000*, Hongkong 2001. Auch erschienen als TR-22/00, Institut für Informatik, FU Berlin.

[8] J. Richter-Gebert & U. Kortenkamp, *Grundlagen Dynamischer Geometrie.* In: *Zeichnung – Figur – Zugfigur*, herausgegeben von H.-J. Elschenbroich, Th. Gawlick, H.-W. Henn, Franzbecker, Hildesheim 2001.

[9] Rainer Witt, *Konstruktive Geometrie*, `http://homepages.tu-darmstadt.de/~raiwi/`, TU Darmstadt, 1999.