

DYNAMISCHE GEOMETRIE: GRUNDLAGEN UND MÖGLICHKEITEN

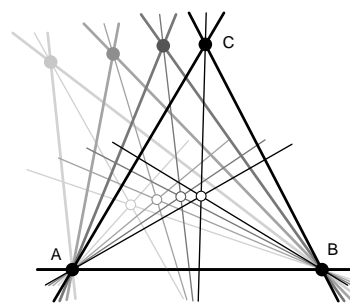
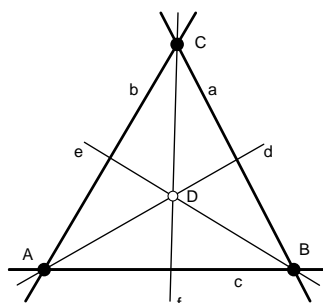
JÜRGEN RICHTER-GEBERT & ULRICH H. KORTENKAMP

1. WAS IST „DYNAMISCHE GEOMETRIE“

1.1. Die Grundidee. Stellen Sie sich eine Seite in einem Geometriekapitel eines Mathematikschulbuchs vor. In aller Regel enthält eine solche Seite Bilder, welche einen bestimmten geometrischen Sachverhalt visualisieren sollen. Bei solchen Sachverhalten handelt es sich zum Beispiel um einfache geometrische Lehrsätze, wie die Tatsache, dass sich die Winkelhalbierenden im Dreieck schneiden. Während der geometrische Lehrsatz für beliebige Dreiecke gilt, gibt die Zeichnung zumeist nur ein einziges Beispiel wieder.

Dynamische Geometrie ermöglicht es nun, mit Computerunterstützung den geometrischen „Erfahrungshorizont“ zu erweitern und durch Bewegungen einer Zeichnung Allgemeingültigkeiten zu erkennen. In dynamischer Geometrie existiert eine Zeichnung nicht als statisches Bild, sondern wird intern als Abfolge von Konstruktionsschritten beschrieben, die es ermöglicht, 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 die gesamte Konstruktion sich 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, wobei der Computer für jede neue Position ein neues Bild berechnet und anzeigt. Es entsteht der Eindruck einer fließenden Bewegung.

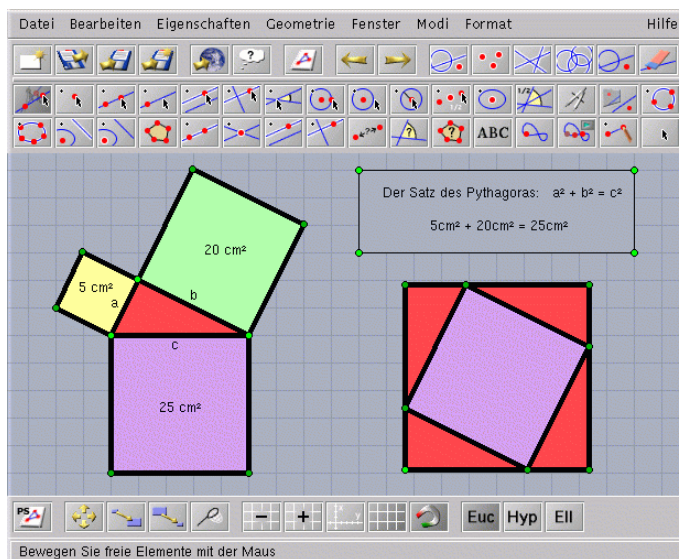
Ein Dynamisches Geometrieprogramm verfügt in aller Regel neben dieser Grundfunktionalität des Zugmodus auch über Möglichkeiten, Konstruktionen interaktiv zu erstellen. Hierzu steht eine Palette von geometrischen Werkzeugen zur Verfügung, die es erlauben, eine bestehende Konstruktion um weitere Elemente zu ergänzen. Zur Grundausstattung

der meisten dynamischen Geometrieprogramme gehören hierbei Werkzeuge für Schnittpunkte, Verbindungsgeraden, Parallele, Senkrechte, Winkelhalbierende und diverse Kreis-konstruktionen. Fortgeschrittenere Programme erlauben auch den direkten Umgang mit Kegelschnitten, Spiegelungen und geometrischen Transformationen.

1.2. **Cinderella.** Die Idee Computer zu nutzen, um geometrische Sachverhalte „in Bewegung“ zu visualisieren, ist keineswegs neu. Bereits Mitte der achtziger Jahre erschienen die ersten kommerziellen Programme zur dynamischen Geometrie (*Geometer's Sketchpad* [3] und *Cabri Géométrie* [5]). Heutzutage gibt es weltweit schätzungsweise 45 Programme mit vergleichbarer Zielsetzung. Man kann sich leicht fragen, ob bei einer solchen Masse an verfügbaren Programmen nicht längst alle Probleme in diesem Bereich sowohl in mathematischer als auch informatischer Hinsicht gelöst sein sollten und keine prinzipiell neuen Entwicklungen zu erwarten seien.

In gewisser Weise ist genau das Gegenteil 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. [4, 8]), auf deren Grundlage sich vollkommen neuartige Implementierungsparadigmen entwickeln lassen (die eine weitestgehende Vermeidung solcher Inkonsistenzen gewährleisten). In manchen Aspekten steht somit das Feld der dynamischen Geometrie mitten in einer „Neudefinition“, bei der grundlegende Fragen der mathematischen Modellierung im Vordergrund stehen.

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 zunehmend zur *Kommunikation* von Geometrie einzusetzen. Während noch vor wenigen Jahren proprietäre Datenformate dominierten und Publikationen, die sich dynamischer Visualisierungsmöglichkeiten bedienten, meist mit Begleitdisketten ausgeliefert wurden, geht heutzutage der Trend in Richtung nahtloser WWW Integration.



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* [6, 7]. Ziel war es hierbei in einer

engen Verflechtung von Grundlagenforschung und Implementierungsarbeit die Grenzen dynamischer Geometrie auszuloten. Es stellte sich heraus, dass eine mathematisch zufriedenstellende Implementierung Konzepte sowohl der *projektiven Geometrie* als auch der *komplexen Analysis* einbeziehen musste. Hierbei ist es wichtig zu beachten, dass diese Theorien in der Implementation zwar berücksichtigt werden müssen, dass aber der Benutzer keineswegs damit vertraut sein muss. Vielmehr arbeitet die Mathematik im Hintergrund und gewährt einen reibungslosen Ablauf des Zugmodus (ähnlich wie man als Autofahrer nicht wissen muss wie das ABS der Bremsen funktioniert).

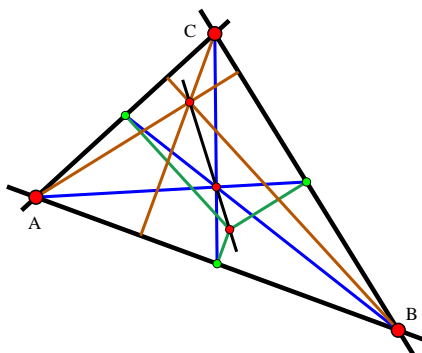
Eine weitere Anforderung, die wir uns stellten, war von Anfang an auf eine möglichst nahtlose Internetintegration hinzuarbeiten. *Cinderella* ist von daher komplett in Java geschrieben und erlaubt die einfache Einbindung von Zeichnungen als Applets in HTML-Seiten, die man ohne weitere Hilfsmittel mit einem gängigen Browser (z.B. *Netscape* oder *Internet Explorer*) betrachten kann. Weiterhin haben wir versucht, neue Möglichkeiten die sich durch das Anbieten dynamischer Geometrie im Internet ergeben, auszuschöpfen. So ist es z.B. möglich, mit Hilfe des Programms Geometrie Aufgaben mit vollautomatischer Lernkontrolle anzubieten. Die Lernkontrolle wird hierbei von einem *Automatischen* Beweiser durchgeführt, welcher wiederum entscheidend auf der mathematischen Theorie basiert. Insbesondere ergeben sich hierbei neue Möglichkeiten für den Bereich des Teleteaching bzw. Distance Learning.

Die folgenden Seiten sollen einen Einblick in die grundlegenden Probleme und Lösungsansätze geben, sowie Möglichkeiten die sich aus der Verknüpfung von Geometrie und Internet ergeben aufzeigen.

2. EINSATZSZENARIEN

Bevor wir uns den auftretenden Problemen beim Schreiben eines solchen Programmes widmen, wollen wir kurz auf drei klassische Einsatzfelder dynamischer Geometrie eingehen. Diese Einsatzmöglichkeiten beschränken sich hierbei nicht allein auf den Bereich des Geometrieunterrichts.

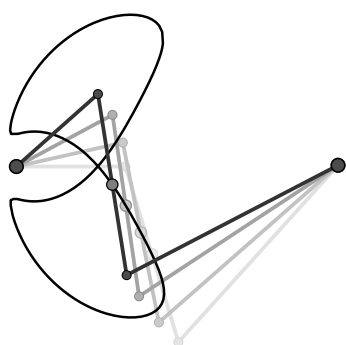
2.1. Erstellen von Zeichnungen. Angenommen, Sie schreiben an einer Publikation wie dieser und wollen sie mit geometrischen Bildern anreichern. Womöglich kommt es Ihnen sogar noch darauf an, einen bestimmten geometrischen Sachverhalt möglichst prägnant in einem statischen (!) Bild darzustellen. Nehmen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



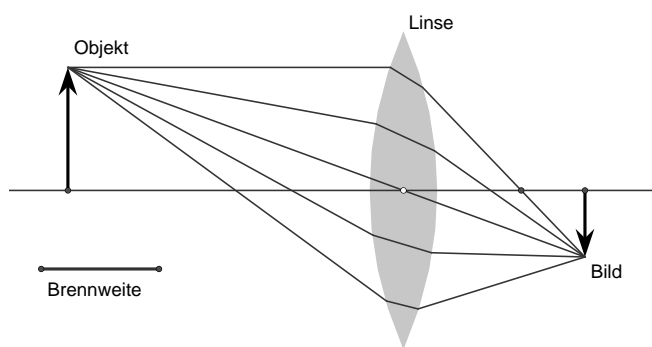
Die Eulergerade: Höhenschnittpunkt, Schnittpunkt der Mittelsenkrechten und Schnittpunkt der Seitenhalbierenden liegen auf einer Geraden.

Mit der Wahl der Eckpunkte des Dreiecks liegt das komplette Bild dieser Konstruktion fest. Es erfordert einiges an Erfahrung oder Ausprobieren, diese Eckpunkte so zu wählen, dass die Zeichnung zufriedenstellend wird. Bei einer mehr oder weniger zufälligen Wahl der Eckpunkte kommen sich entweder Punkte zu nahe, Geraden fallen beinahe zusammen oder werden fast parallel, so dass sich deren 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. Fast alle Zeichnungen in dieser Publikation wurden auf diese Weise erstellt.

2.2. Geometrisches Experimentierfeld. Der Einsatz dynamischer Geometrie ermöglicht auch das Gewinnen neuer Erkenntnisse. Man kann das Programm einfach als eine Art geometrischer Taschenrechner benutzen, 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 (insb. der Mechanik und Optik) lassen sich adäquat auf geometrische Modelle abbilden. So kann man z.B. relativ einfach die Kinematik bestimmter Gelenkmechanismen explorieren oder auch z.B. einem Schüler eine vorgefertigte Konstruktion quasi als virtuelles Labor zur Verfügung stellen.



Gelenkmechanismus



Strahlengang in einer Linse

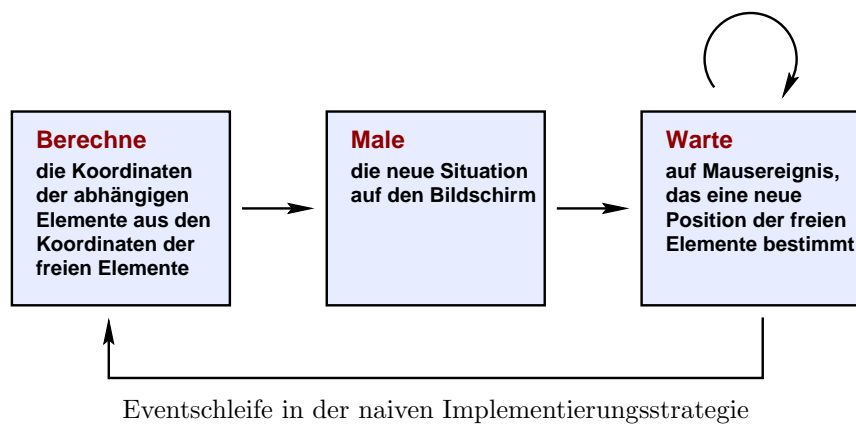
Tatsächlich werden oftmals dynamische Geometriesysteme auch direkt in der Forschung und Entwicklung eingesetzt, um Zusammenhänge zu explorieren.

2.3. Vermitteln von Wissen. Auf diesen naheliegenden Aspekt soll hier nur ganz kurz eingegangen werden. Die Möglichkeiten dynamischer Geometrie bei der Vermittlung von Wissen sind überaus vielfältig. Sie reichen vom Bereitstellen vorgefertigter Konstruktionen über das interaktive Lösen von Aufgaben bis hin zur Benutzung als Präsentationswerkzeug. In einem späteren Kapitel werden wir nochmals speziell auf Lehr- und Lernsituationen in Internetszenarien eingehen.

3. DIE MATHEMATIK

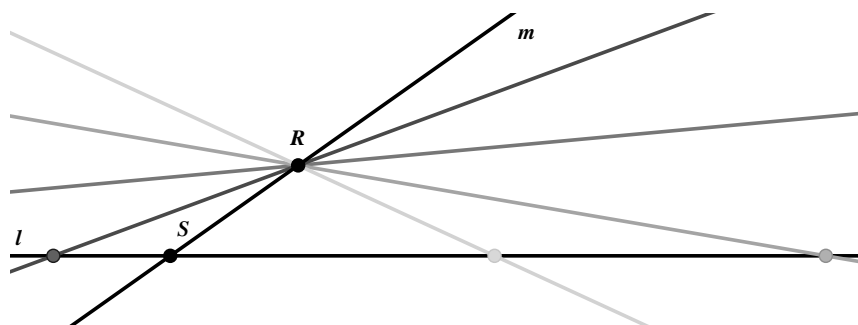
Auf den ersten Blick scheint das Schreiben eines Geometrieprogramms keine ausserordentlich schwierige Aufgabe zu sein. Der übliche erste Ansatz sieht in etwa folgendermassen aus.

- (1) Man repräsentiert die Elemente intern durch geeignete Koordinaten (z.B. Punkte durch (x, y) Koordinaten, Geraden durch die Parameter (a, b, c) einer Geradengleichung $ax + by + c = 0$ 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. Dies ist meistens mit Mitteln der Schulmathematik machbar. (Z.B. ergibt sich die Verbindungsgerade zweier Punkte (x_1, y_1) und (x_2, y_2) zu $a = y_1 - y_2$, $b = x_2 - x_1$, $c = x_1y_2 - x_2y_1$.)
- (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 malt.



In der Tat folgen die meisten der auf dem Markt erhältlichen dynamischen Geometrieprogramme mehr oder weniger diesen drei Implementierungscharakteristika. Im Detail erweisen sich allerdings alle drei Punkte als mathematisch unzulänglich, dies soll im Folgenden kurz dargelegt werden.

3.1. Punkte im Unendlichen. Fertigt man auf einem Blatt Papier eine Zeichnung an, so steht einem nur eine begrenzte Fläche zur Verfügung. Will man den Schnittpunkt zweier Geraden zeichnen, die sich auf dem Blatt nicht treffen, so hatte man „Pech“ und muss schlimmstenfalls von vorne beginnen. Jeder tatsächlich ausführbare Konstruktionsschritt findet im begrenzten Gebiet des Blattes statt. Bei dynamischer Geometrie ist die Situation grundlegend anders. Erstens spricht nichts dagegen, einen Schnittpunkt zu berechnen, der ausserhalb des sichtbaren Zeichenbereichs liegt. Mehr noch, selbst in einer einfachen Konstruktion können die freien Elemente nachträglich (!) so verschoben werden, dass Schnittpunkte beliebig weit draussen liegen.



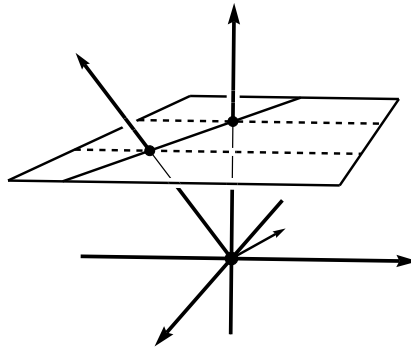
Punkte können durch Verschieben beliebig weit weg wandern

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 überstreichen wir die Parallelsituation, so kommt der Schnittpunkt von der anderen Seite zurück. Wie ist die Parallelsituation selbst zu behandeln? In vielen Systemen wird ein solcher „Schnittpunkt“ zweier paralleler Geraden für einfach „ungültig“ erklärt, da man ihn ja ohnehin nicht sieht. Was aber, wenn an diesem 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. Oftmals führt dies zur Notwendigkeit der Behandlung von allen möglichen Spezialfällen.

In der Tat gibt es hierfür einen mathematisch befriedigenderen Ansatz. Man kann die Ebene um so genannte „Punkte im Unendlichen“ erweitern. Zu jeder Parallelschar führt man genau einen solchen Punkt ein, welcher auf allen Geraden dieser Schar liegen soll. 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 kann man auch erreichen, 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. Man gelangt somit zum System der so genannten *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 $(6, 9, 3)$ den selben 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 $(x, y, 1)$. Punkte der Form $(x, y, 0)$ spielen die Rolle der Punkte im Unendlichen.



Einbettung der euklidischen Ebene in den Raum

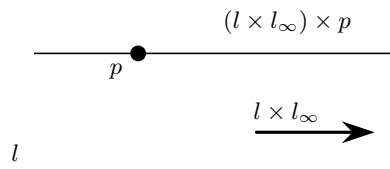
Die gewöhnliche euklidische Zeichenebene wird somit praktisch räumlich als $z = 1$ -Ebene eingebettet. Ein Punkt in dieser Ebene wird als Verbindungsgerade des Punktes mit dem Koordinatenursprung dargestellt. Wandert ein Punkt weit nach draussen, wird diese Gerade zunehmend horizontal. Die im Grenzfall 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 so genanntes Kreuzprodukt $p_1 \times p_2 = (y_1 z_2 - z_1 y_2, z_1 x_2 - x_1 z_2, x_1 y_2 - y_1 x_2)$. Dies ist der Fall, da 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 Punkten 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. Betrachten wir z.B. die Berechnung einer Parallelen q zu einer Geraden l durch einen Punkt P : Es sei zunächst $l_\infty = (0, 0, 1)$ die Gerade im Unendlichen. Die Berechnung von q kann dann wie folgt durchgeführt werden

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

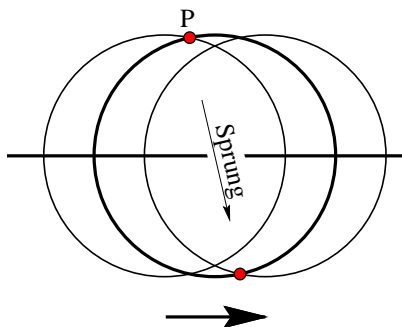
Mittels $l \times l_\infty$ wird zuerst der Fernpunkt in Richtung l bestimmt (der Schnitt von l mit der Geraden im Unendlichen). Sodann wird die Verbindungsgerade von diesem Fernpunkt mit Punkt p bestimmt. Dies ist die gesuchte Parallele.



Berechnung der Parallelen zu l durch p

3.2. Vermeidung Springender Punkte. Die Verwendung projektiver Ansätze und die Einbeziehung unendlich ferner Punkte wird ausser bei Cinderella auch bei einigen anderen Programmen eingesetzt (z.B. Cabri Géomètre). Es gibt allerdings noch einen weiteren unerwünschten (mathematisch unmotivierten) Effekt, der speziell dynamischer Natur ist. Betrachtet man in einem herkömmlichen dynamischen Geometrieprogramm eine etwas

umfangreichere Konstruktion so kann man oftmals das folgende Verhalten feststellen: Während man einen der freien Punkte im Zugmodus bewegt, springen plötzlich ganze Teile der Konstruktion unerwartet von einer Position an eine andere. Oftmals repräsentiert die Zeichnung nach einem solchen Sprung nicht mehr die gewünschte Konstruktion.

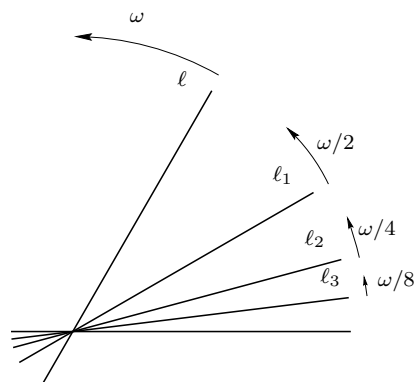


Springendes Verhalten in herkömmlichen Systemen

Die folgende einfache Konstruktion weist in allen uns bekannten Systemen (ausser *Cinderella*) dieses Verhalten auf: 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, springt spontan der Schnittpunkt von der einen auf die andere Möglichkeit. Würden an den Schnittpunkt noch weitere Konstruktionsschritte gebunden, so würden diese ebenfalls spontan mit dem Punkt die Position wechseln.

Die Ursache für ein solches Verhalten ist erstaunlich tief liegend 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 eben nicht nur einen Schnittpunkt sondern zwei. Für jedes neu zu berechnende Bild muss das Programm 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 dem dritten Punkt der oben angegebenen Implementierungsstrategie folgt, unvermeidlich solche Sprünge auftreten müssen. Der dritte Punkt der „naiven Implementierungsstrategie“ besagt, dass man zu einer Neuberechnung des Bildes lediglich die Koordinaten der freien Elemente heranziehen muss. Insbesondere 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 bilden die Winkelhalbierende ℓ_1 mit der x -Achse. Sodann bilden wir die Winkelhalbierende ℓ_2 von ℓ_1 mit der x -Achse, und so fort.



Iterierte Winkelhalbierende

Fassen wir nun im Zugmodus die Gerade ℓ mit der Maus an und rotieren diese mit konstanter 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 seine ursprüngliche Position zurückgekehrt ist, die Gerade ℓ_2 nur eine Vierteldrehung vollführt hat. Mit anderen Worten: wollen wir in diesem Beispiel Sprünge vermeiden, so kann 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 nämlich zwei (senkrecht aufeinander stehende) Winkelhalbierende.

Ein wesentliches Element, um dieses Problem aufzulösen, besteht in der Einbeziehung der Vorgeschichte einer Konfiguration. Man kann eben 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.

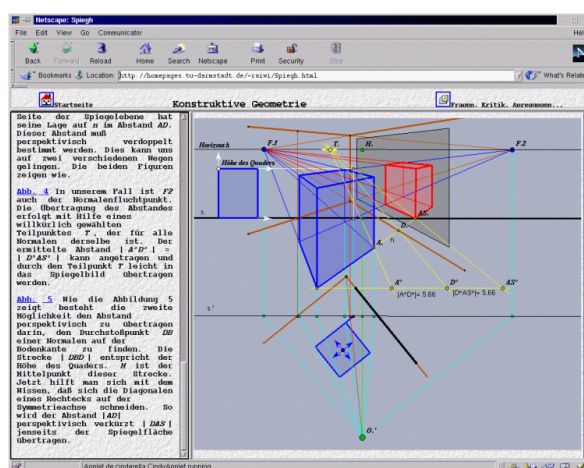
4. DYNAMISCHE GEOMETRIE UND DAS INTERNET

Abschliessend wollen wir noch auf die speziellen Möglichkeiten, die der Einsatz interaktiver Geometrie in Verbindung mit Internettechnologie bietet, eingehen. Hierbei wollen wir einerseits die in *Cinderella* bereits umgesetzten, aber auch einige visionäre Szenarien beschreiben.

4.1. Kommunikation. In der jetzigen Form gestattet *Cinderella* einen extrem einfachen Export dynamischer Zeichnungen und Animationen aufs Internet. Eine einmal gemachte Konstruktion wird hierzu abgespeichert und *Cinderella* erstellt eine zugehörige HTML-Seite. Das Java basierte und streng modulare Design von *Cinderella* ermöglicht es nun, dass im Browser die Zeichnung als vollwertige dynamische Konstruktion vorhanden ist. Der Betrachter kann mit der Maus auf der HTML-Seite in das Bild gehen und die Konstruktion 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).

Ein in diesem Zusammenhang geplantes Projekt ist die Kommunikation „telefonartig“ zu gestalten. Gemeint ist damit, dass zwei (oder mehr) Benutzer simultan über das Internet an einer Zeichnung arbeiten. Auf diese Weise ergäbe sich praktisch die Möglichkeit einer „geometrischen Telekonferenz“.

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



Seiten zur Darstellenden Geometrie

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

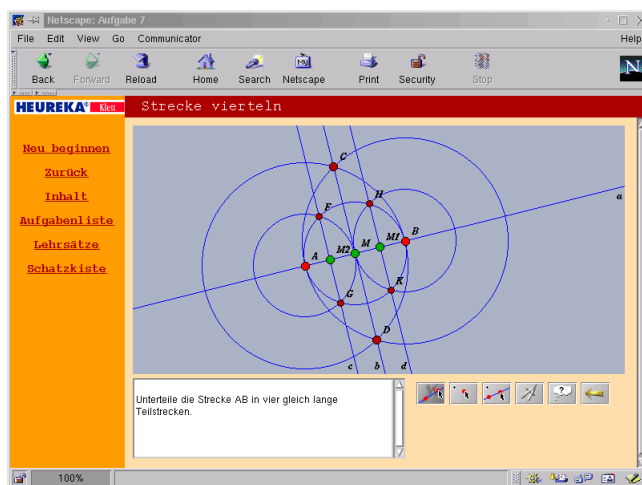
4.3. Vernetzter Klassenraum. Ein bislang noch nicht umgesetztes Szenario, welches die Möglichkeiten des WWW sehr schön ausnutzen würde, wäre der Einsatz in einem vernetzten Klassenraum. Denkbar wäre hierbei, dass der Lehrer ein solches Programm einerseits als Demonstrationswerkzeug an der interaktiven Tafel (Smartboard) einsetzt und entweder anhand Vorbereiteter multimedialer HTML-basierter Unterrichtsmaterialien oder mittels der Standaloneversion vorträgt.

Danach könnten Schüler an ihren Bildschirmarbeitsplätzen Aufgaben selbst lösen. Mittels der oben angesprochenen Telekonferenzfunktionalität könnte ein Lehrer direkt in die Konstruktionen der Schüler Einsicht nehmen und Hinweise geben. Umgekehrt könnten auch Schüler in die für alle sichtbare Tafelarbeit des Lehrers eingreifen und von ihren Arbeitsplätzen aus selbst für alle an der Tafel sichtbar Lösungen präsentieren.

Die für dieses Szenario benötigten Ergänzungen zu der aktuellen *Cinderella*-Version bestehen im Wesentlichen aus den schon angesprochenen Telekonferenzmodulen und der Schaffung eines Kontrollmechanismus, welcher die lokalen Zugriffsrechte administriert.

4.4. **Aufgaben mit Lernkontrolle.** Mit der jetzigen Version ist aber auch ein Bearbeiten von Aufgaben ohne direkte Lehrerpräsenz möglich. *Cinderella* verfügt über einen Mechanismus, welcher es ermöglicht, automatisch und zuverlässig zu prüfen, ob zwei auf verschiedene Weise konstruierte Elemente einander identisch sind. Dieser Mechanismus kann z.B. zum automatischen Überprüfen geometrischer Lehrsätze verwendet werden. Er 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 in *Cinderella* zunächst eine Musterlösung für diese Aufgabe. In einem speziellen, in *Cinderella* integrierten 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

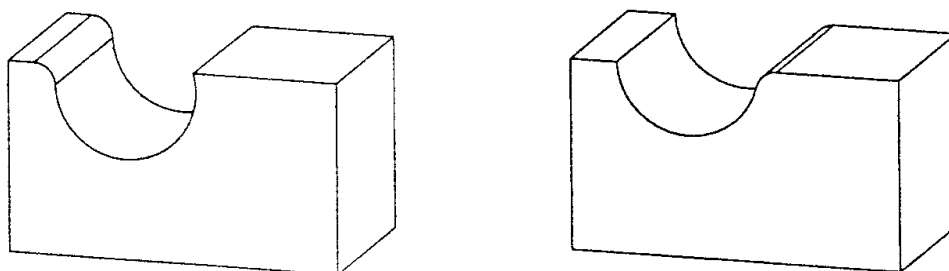
Der Schüler erhält ein HTML-basiertes Arbeitsblatt, welches sowohl die Zeichenfläche mit den Startelementen zeigt, als auch den Aufgabentext und die zur Verfügung stehenden Werkzeuge. Die Musterlösung ist vollkommen verborgen. Der Schüler beginnt nun mit Hilfe der Werkzeuge die Aufgabe zu lösen. Hat 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 Hinweis- und Belohntexten zu versehen, die den Schüler entlang eines bestimmten Lösungsweges leiten.

Das in *Cinderella* umgesetzte Aufgabenkonzept wurde bereits mehrfach in Schul-Feldversuchen eingesetzt und stieß dabei immer wieder auf eine sehr positive Resonanz von Seiten der Schüler. Die bei HEUREKA Klett [7] erschienene Ausgabe 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] und gewann damit den diesjährigen Europäischen Wettbewerb *Mathplets* über Java basierte Mathematik Applets.

5. WEITERGEHENDE ANWENDUNGEN

Die in *Cinderella* umgesetzten mathematischen Theorien haben auch Anwendungen, die über den direkten Bereich der dynamischen Geometrie hinausgehen. Diese sind als zukünftige Forschungsprojekte geplant und sollen im folgenden kurz skizziert werden.

5.1. Parametrisches CAD. Im Bereich von *Computer Aided Design* hat man es im Wesentlichen auch mit geometrischen Konstruktionen zu tun. Ein Spezialgebiet dort ist das so genannte *parametrische CAD*. Hierbei ist man daran interessiert, genau wie in der dynamischen Geometrie nicht allein eine einzige Konstruktion zu beschreiben, sondern eine ganze Klasse von Konstruktionen. Beispielsweise einen Tisch mit abgerundeten Ecken, bei dem man nachträglich noch die Dicke der Tischplatte, den Radius der Rundung, die Höhe der Beine etc. variieren kann.

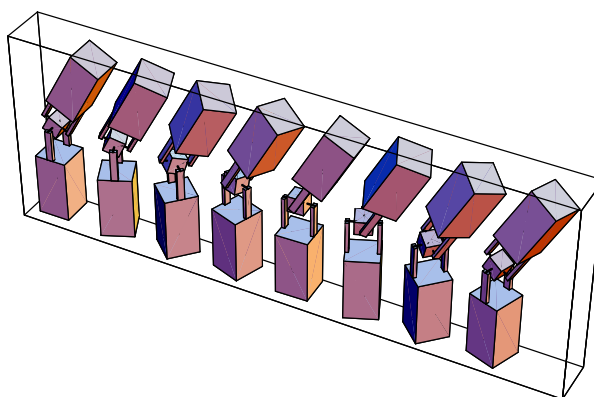


Springendes Verhalten im parametrischen CAD

Genau wie in der dynamischen Geometrie kann man auch im Bereich des parametrischen CAD bei praktisch allen Systemen Sprungeffekte beobachten. Das folgende Beispiel wurde von Ch.M. Hoffmann bei einem echten CAD System beobachtet [2]. Man nehme einen rechteckigen Block, bei dem man am Rand ein rundes Loch herausfräst und danach eine der Fräskanten abrundet. Bewegt man das Loch auf und ab, so springt die Fräskante von einer Seite auf die andere.

Die Gründe für das Springen sind haargenau die gleichen wie in der dynamischen Geometrie. Letztlich ist der Schnitt der Fräsung mit der Kante des Blocks eine mehrdeutige geometrische Operation (Schnitt Kreis/Gerade). Es ist ein von uns geplantes Forschungsprojekt, unsere Theorien auch für den CAD Bereich umzusetzen.

5.2. Computergestützte Kinematik. In der computergestützten Kinematik möchte man auf einem noch höheren Niveau Design betreiben. Man möchte in einem virtuellen Mechanikbaukasten vorgefertigte Teile zusammenstecken und deren Verhalten unter Kräfteinfluss und Bewegung studieren. Auch hier treten wieder die gleichen Probleme auf, denn typischerweise enthalten mechanische Konstruktionen sehr viele geometrische Mehrdeutigkeiten, die während einer Bewegung konsistent aufgelöst werden sollen.



Kinematische Simulation eines Kradangelenks

Die Probleme treten hierbei allerdings nicht isoliert auf, da oftmals durch die Einbeziehung weiterer gleichzeitig zur Verfügung stehender Freiheitsgrade Mehrdeutigkeiten viel höherer Ordnung entstehen können.

5.3. Virtuelle Welten. Noch komplexer werden Simulationsprobleme im Bereich der *virtual reality*. Dort hat man den Anspruch, gewisse Szenarien möglichst realitätsnah zu simulieren und bildlich darzustellen. Zumeist sind die Simulationsprobleme derart komplex, dass das Schwergewicht auf der bildlichen Darstellung gelegt wird und die eigentliche Simulation sich auf das Aufrufen vorgefertigter Sequenzen beschränkt.

In einem System zur „echten“ virtuellen Realität treten aber dann auch unweigerlich die Probleme auf, die bereits in dynamischer Geometrie zu Tage treten. So sollte es in einem solchen System z.B. möglich sein, vier virtuelle Metallstangen zu einem einfachen Gelenkmechanismus zusammensetzen und diesen dann in der virtuellen Hand zu bewegen. Die dann auftretenden Probleme sind wieder prinzipiell geometrischer Natur und erfordern letztlich die gleichen Verfahren wie sie für dynamische Geometrie notwendig sind.

LITERATUR

- [1] BRYAN DYE, *Transformations*, <http://www.anglia.co.uk/education/mathsnnet/campus01/topindex.html>, AngliaCampus, 1999.
- [2] CH. M. HOFFMANN, *Solid Modeling*, in: J.E. Goodman & J. O'Rourke. (eds.): *Handbook of Discrete and Computational Geometry*, Lecture Notes in Mathematics **1346**, CRC Press, Boca Raton, New York, 1997, 863–880.
- [3] NICHOLAS JACKIW, *The Geometer's Sketchpad*, Key Curriculum Press, Berkeley, 1991–1995.
- [4] U. KORTENKAMP, *Foundations of dynamic geometry*, PhD-thesis, ETH Zürich, 1999, <http://www.inf.fu-berlin.de/~kortenka/Papers/diss.pdf>.
- [5] JEAN-MARIE LABORDE AND FRANCK BELLEMAIN, *Cabri-Geometry II*, Texas Instruments, 1993–1998.
- [6] J. RICHTER-GEBERT & U. KORTENKAMP, *Cinderella - The interactive geometry software*, Springer 1999; see also <http://www.cinderella.de>.
- [7] J. RICHTER-GEBERT & U. KORTENKAMP, *Cinderella - Die interaktive Geometriesoftware*, HEUREKA Klett, 2000.
- [8] J. RICHTER-GEBERT & U. KORTENKAMP, *Complexity issues in dynamic geometry*, in preparation, manuskript available on request.
- [9] RAINER WITT, *Konstruktive Geometrie*, <http://homepages.tu-darmstadt.de/~raiwi/>, TU Darmstadt, 1999.