

WinCAG

Lehr- und Lernsoftware zur (Computer-)Geometrie

Karl-Heinz Brakhage

Institut für Geometrie und Praktische Mathematik, RWTH Aachen

Insbesondere in der Geometrie spielt Lehr- und Lernsoftware eine immer größere Rolle. Es bedarf aber wohldurchdachter Konzepte, um die Themen mittels Computeranimationen wirklich besser vermitteln zu können. Einen großen Aspekt spielt hier die Einheit von Lehr- und Lernsoftware. Der Unterrichtsstoff sollte mit dem System Schritt für Schritt nachgearbeitet werden können. WinCAG ist so konzipiert, dass man sowohl simple Konstruktionen als auch komplizierte Zwangsbewegungen einfach animieren kann. Dies kommt, richtig eingesetzt, speziell der Schulung des räumlichen Vorstellungsvermögens zugute. Erst durch die Online-Modifikation der Objekte und/oder der Blickrichtung tritt die Räumlichkeit plastisch hervor. Das Programmpaket WinCAG ist ursprünglich speziell für die Lehre der Darstellenden Geometrie entwickelt worden. So enthält es etwa spezielle Tools, die es ermöglichen, auf sehr einfache Art der Darstellung in Zweitafelprojektion eine anschauliche Axonometrie hinzuzufügen. Mit Hilfe der Layertechnik können auch sehr komplexe, vorkonstruierte Sachverhalte Schritt für Schritt gezeigt und von den Studenten zu Hause nachgearbeitet werden. Dies sind nur zwei typische Punkte, die WinCAG von anderen dynamischen Geometrieprogrammen unterscheidet. In dieser Arbeit soll ein Überblick des Umfangs und der Einsatzmöglichkeiten von WinCAG gegeben werden.

1 Einleitung

WinCAG (**Windows** version of **Computer Aided Geometry**) hieß ursprünglich CAG und wurde bereits ab 1985 als DOS-Version entwickelt. Ausgangspunkt war der Bedarf eines leistungsfähigen Systems zum Erstellen von Zeichnungen und Übungsaufgaben zur Darstellenden Geometrie. Schon damals gab es zahlreiche zwei- und dreidimensionale CAD-Systeme. Die meisten speicherten aber nur die Objekte oder das, was davon gezeichnet werden sollte. So sind im Moment der Erstellung der Konstruktionen zwar Operationen wie Schnitte, Tangenten usw. möglich, aber nachträgliche Änderungen der in die Konstruktion eingehenden Objekte werden nicht berücksichtigt, können es auch nicht mehr, denn die dazu nötige Information (Konstruktionsbefehl) fehlt. Hier lag der Ansatzpunkt des Systems CAG.

Bereits 1987 lag eine recht leistungsfähige interaktive Version vor. Die zu Grunde liegenden Objekte waren Punkte, Geraden, Kreise, Ellipsen und Splines. Für sie waren sehr umfangreiche Operationen inklusive aller möglichen Schnitte und Tangenten untereinander implementiert. Objekte konnten zu *Polylinien* zusammengefasst, schraffiert, verschoben, gedreht und beschriftet werden. Benutzer waren eine Technische Zeichnerin, wissenschaftliche Mitarbeiter und studentische Hilfskräfte. Man konnte nun auch die Maße so lange ändern, bis die Gesamtkonstruktion papierfüllend war (optimale Größe), und die einzelnen Konstruktionsteile sich nicht überlagerten (Übersichtlichkeit). Das erste Ziel war erreicht und wurde in [1] an Beispielen

dokumentiert. Ab dem Wintersemester 1987/88 wurde das System auch zur Ausbildung von Studenten (Maschinenbauer und Bauingenieure) eingesetzt. In [2] ist die DOS-Version detailliert beschrieben. Die Entwicklung war Ende 1987 im Prinzip abgeschlossen und wurde für ca. 10 Jahre eingestellt. Es sei hier nur noch erwähnt, dass wegen der nicht vorhandenen Leistungsfähigkeit der Rechner die Modifikationen erst auf Abruf und nicht *on the fly* wirksam wurden.

Mit der erhöhten Leistungsfähigkeit der Rechner ergaben sich dann 10 Jahre später neue Ziele: Dynamisches Anpassen bei Modifikationen (entspricht dem *Zug-Modus* bei Cinderella [3]) und automatischer Ablauf vorbereiteter Änderungen in komplexen Zeichnungen zum Einsatz in den Vorlesungen. Zu diesem Zweck wurde CAG zunächst objektorientiert umgeschrieben und auf Windows portiert. Danach wurden dann die Modifikationsmöglichkeiten erweitert. Um auch räumliche Darstellung einfach und schnell konstruieren zu können, wurde ein *pseudo 3d* Modul integriert. Umfangreiche Zeichnungen und deren Details wurden durch Layertechniken handhabbar. Mit dem so erweiterten System WinCAG wurden zu den Übungen und Vorlesungen zur Darstellenden Geometrie für Architekten nach [4] zahlreiche Vorlagen zur visuellen Unterstützung des räumlichen Vorstellungsvermögens und des Verständnisses der Konstruktionen erstellt. Einen Auszug hiervon findet man in [5]. Wesentlich sind hier die folgenden Aspekte: Schrittweise Entwicklung einer Zeichnung, gleichzeitige Visualisierung verschiedener Projektionen, Veränderung der Parameter der Objekte, Veränderung der räumlichen Lage der Objekte sowie die Veränderung der Parameter der Abbildung. In [5] findet man auch Beispiele für die Visualisierung komplexer Zusammenhänge. Das Programm WinCAG steht in der dort vorgestellten Version unter www.dg-ac.de/wincag für Lehr- und Lernzwecke kostenlos zur Verfügung.

Bevor in den folgenden Abschnitten genauer auf den Umfang und die Einsatzmöglichkeiten von WinCAG eingegangen wird, folgt hier noch eine kurze Zusammenfassung der wesentlichen Merkmale. Die Objekte des Systems sind Punkte, Geraden, Kreise, Kegelschnitte und Splines. Die Operationen zur Objekterzeugung lassen sich in vier Klassen zusammenfassen:

1. *Definitionen* — Ein Punkt durch seine Koordinaten, eine Gerade durch zwei Punkte usw.
2. *Schnitte* — alle möglichen Schnitte beliebiger Objekte.
3. *Tangenten* — jede mögliche Tangente kann erzeugt werden.
4. Operationen wie Mittelpunkt, Parallele, Orthogonale usw.

Da die Konstruktionsbefehle gespeichert werden, bei Modifikationen nicht die Objektdaten, sondern die erzeugenden Kommandos geändert werden, und die Rekonstruktion der Zeichnung nur mittels der Kommandos geschieht, werden auch die von den Änderungen betroffenen Objekte korrekt modifiziert. Die Modifikationen werden normalerweise interaktiv vorgenommen. Man kann die Konstruktion aber auch als Textdatei speichern, diese editieren und dann diese laden. Zu diesem Zweck sind alle Befehle so aufgebaut, dass sie zu einer eindeutigen Lösung führen.

Das System interpretiert sämtliche Objekte als Hilfsobjekte. Soll ein Teil des Objektes (oder das ganze Objekt) auch bei der Ausgabe auf einen Drucker oder in eine (Postscript- oder BMP-)

Datei sichtbar werden, so muss es vorher entsprechend eingezeichnet werden. Dabei wird für Postscriptdateien selbstverständlich das Prinzip der Vektorgrafik verwendet. Es können verschiedene Farben, Linientypen und -stärken verwendet werden. Mit dem System können auch anspruchsvolle Schraffuren und Beschriftungen durchgeführt werden. Dazu ist die Schriftgröße und -richtung wählbar, und es stehen verschiedene, auch mathematische Fonts zur Verfügung.

2 Objekte — Erzeugung und Modifikation

Wir wollen hier im Wesentlichen nur die Operationen besprechen, die für den automatischen Ablauf von Modifizierungen relevant sind. Diese sogenannten *Demos* können nämlich so vorbereitet werden, dass sie in der Vorlesung mittels eines simplen Mausklicks ablaufen (siehe auch [5]).

Um das Prinzip der Modifikationen genauer zu verstehen, widmen wir uns exemplarisch den Punkten. Wir unterscheiden drei Arten von Punkten.

1. Völlig freie Punkte wie
 - a) Punkt durch Koordinatenangabe (Maus/Tastatur)
 - b) Punkt relativ zu einem anderen Punkte
2. Bedingt freie Punkte wie
 - a) Punkt auf Gerade, Kreis, Kegelschnitt, Spline
 - b) Punkt zwischen zwei Punkten (Teilungsverhältnis)
 - c) Punkt im gewissen Abstand und einer vorgegebenen Richtung (Höhe)
3. Fixierte Punkte wie
 - a) Schnittpunkte, Tangentenpunkte
 - b) Lotpunkte, Mittelpunkte

Punkte der ersten Art haben zwei Freiheitsgrade; die zweiter Art noch einen und die dritter Art gar keinen mehr. Beim Modifizieren können zwar z.B. Schnitte umdefiniert werden, aber zur Steuerung von Demos eignen sich nur die Definitionen erster und zweiter Art. Die Verbindungen zu anderen Objekten können nachträglich geändert werden — also etwa einen Punkt auf einen Kreis legen oder ihn von einem solchen entfernen. Während der Konstruktion erkennt WinCAG bei fast allen Befehlen automatisch die Inzidenz (Punkt auf einem anderen Objekt, Schnittpunkt, usw.) und setzt automatisch den zur Erzeugung erforderlichen Befehl ab.

Für 1.a) lautet der Befehl zur Erzeugung eines Punktes etwa **P=DefinePoint(x,y)**, wodurch ein Punkt **P** mit den Koordinaten **x** und **y** erzeugt wird. Der Befehl zum automatischen modifizieren sieht dann folgendermaßen aus:

ModifyPoint(P; x₀, x₁, dx; y₀, y₁, dy)

In Abbildung 1 sehen wir im linken Bild die Wirkung dieses Befehls. Der Punkt bewegt sich in dem durch $[\mathbf{x}_0, \mathbf{x}_1] \times [\mathbf{y}_0, \mathbf{y}_1]$ gegebenen Rechteck. Dabei nimmt er in x -Richtung die Schrittweite \mathbf{dx} und y -Richtung die Schrittweite \mathbf{dy} . An den Rechteckgrenzen prallt der Punkt nach dem Prinzip *Einfallswinkel gleich Ausfallswinkel* ab. Anwendung findet dies etwa in einer Demo wie wir sie im rechten Bild von Abbildung 1 sehen — der Punkt \mathbf{O}' wird dabei wie gerade für Punkt \mathbf{P} beschrieben modifiziert. Man sieht dann, wie der Standort des Beobachters sich in der Perspektive auswirkt. Mit einem Befehl gemäß 2.b) oder 2.c) lässt sich dann auch noch

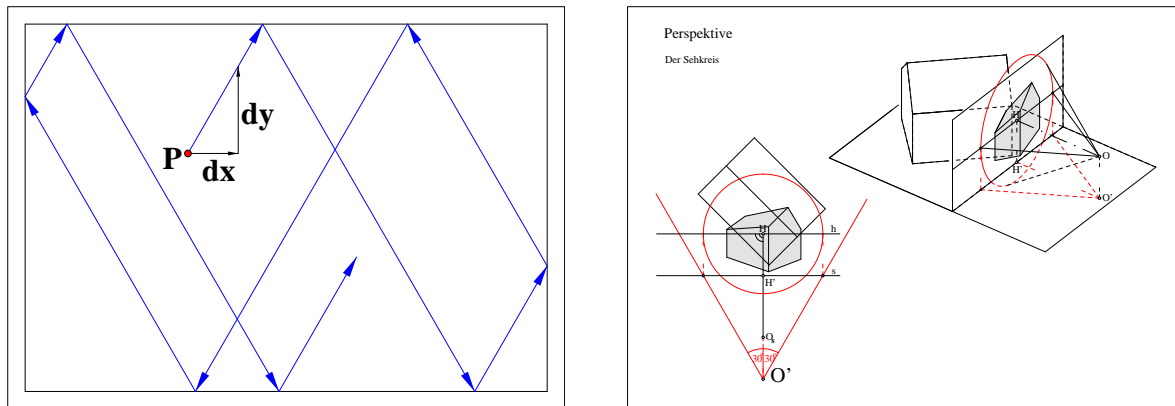


Abbildung 1: Automatische Modifikation von Punkten

einfach demonstrieren wie sich die *Höhe* des Beobachters auf das Bild auswirkt.

Um im Zugmodus oder bei den automatischen Modifikationen konsistentes Verhalten zu erzielen, sind in WinCAG zwei sich ergänzende, unterschiedliche Strategien implementiert. Diverse Befehle, z.B. der Schnitt zweier Kreise (quadratische Gleichung), haben mehr als eine Lösung. Die erste Strategie ist die, dass stetige Änderungen der Parameter auch zu stetigen Änderungen der Lösung führen. Dies lässt sich bei einer Begrenzung der Schrittweite in vielen Situationen durch den Vergleich zur vorherigen Lösung realisieren.

Nun können aber auch Situationen auftreten, wo keine reellen Lösungen existieren (die Kreise schneiden sich reell nicht) bzw. die Lösung liegt im *Unendlichen* (parallele Geraden). In [6] und [3] wird beschrieben, wie man hier mit Mitteln der *komplexen Analysis* und *projektiven Geometrie* weiterkommt. Ein wesentlicher Knackpunkt bleibt aber bestehen. Es gibt Situationen (z.B. sich berührende Kreise), in denen Lösungen zusammenfallen. Bewegt man sich in eine solche Lage hinein und wieder heraus, so ist kontextfrei i.a. nicht zu entscheiden, welches die *korrekte* Lösung ist. Der Benutzer kann aber fast immer die gesuchte Lösung spezifizieren — etwa durch Angabe eines Referenzobjektes. So kann etwa angegeben werden, dass das System den Schnittpunkt nehmen soll, der einem vorgegeben Punkt am nächsten liegt. So ist gewährleistet, dass keine Vertauschungen passieren können. In WinCAG kann man diese Referenzobjekte (auch nachträglich) angeben.

WinCAG bietet noch eine weitere Variante für den Fall nicht existierender reeller Lösungen. Dies sind die *Konstruktionen mit Alternative*. Das Prinzip wollen wir uns an der klassischen Konstruktion eines Dreiecks, von dem die drei Seiten gegeben sind, klar machen. Im linken Bild von Abbildung 2 ist die Konstruktion illustriert. Im rechten Bild sieht man, dass für Sei-

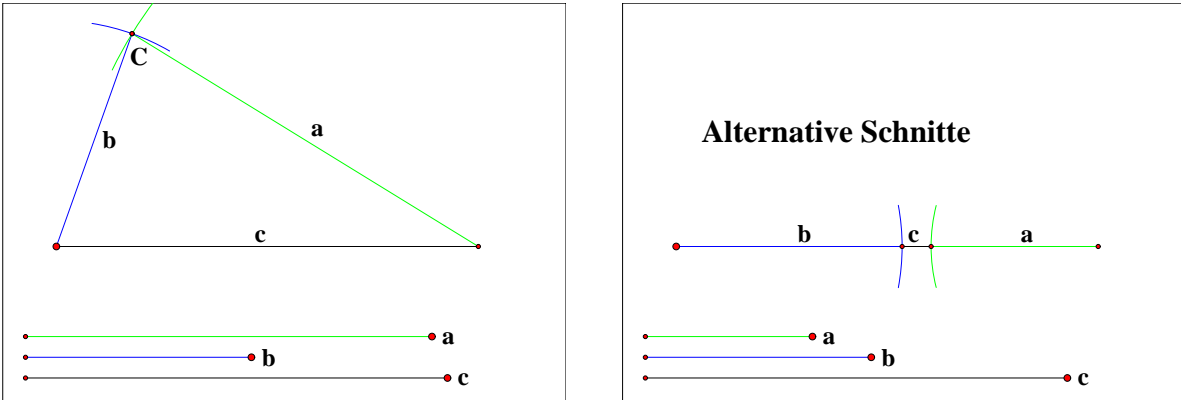


Abbildung 2: Alternative Schnitte

tenlängen $a + b < c$ kein Dreieck existiert. Soll nun der Übergang der rechten Konstellation in die linke durch Verkürzen der Seite a animiert werden, so liegt ab einer bestimmten Länge von a kein (reeller) Schnittpunkt der Kreise mehr vor. In WinCAG kann man angeben, dass als Alternative für den Schnitt des *linken Kreises* K_l mit dem *rechten Kreis* K_r der Schnitt von K_l mit c und statt des Schnittes von K_r mit K_l der Schnitt von K_r mit c genommen wird. Der Punkt C wird also in Wirklichkeit durch zwei Punkte repräsentiert. Dies ist beim Zeichnen der Strecken a und b sowie der angedeuteten Kreise zu berücksichtigen.

Damit der Punkt C beim Verlängern von a nicht *nach unten* wandert, ist auch hier ein für den Schnitt der Kreise als *nächst gelegen* deklarierter Referenzpunkt genügend weit *oberhalb* von c angegeben.

Ein weiterer wichtiger Aspekt für den Ablauf von Demos ist die Layertechnik. Während der Konstruktion mit WinCAG werden vom Prinzip nur die Objekte erzeugt. Man kann Optionen setzen, um Punkte oder Linen automatisch zu zeichnen. Dies ändert aber nichts am obigen Prinzip. Es werden dann nur zusätzliche Befehle abgesetzt. Das Gezeichnete kann man dann Layer(n) (auch mehreren) zuordnen bzw. in Layer kopieren. Beim Ablauf der Demo werden dann die gewünschten Layer selektiert. So lässt sich etwa die schrittweise Entwicklung einer Zeichnung leicht nachvollziehen.

3 Das Abbildungsmodul

Eine große Hilfe für räumliche Darstellungen und deren Animation bietet in WinCAG das Abbildungsmodul. Dem linken Bild von Abbildung 3 ist leicht zu entnehmen, dass in einer Zweifelfprojektion durch Angabe eines Punktes $O' = O''$ ein Koordinatensystem induziert wird. Wählt man also einen Punkt durch seine Lage P' im Grund- und P'' im Aufriss aus, so sind WinCAG damit 3d-Koordinaten gegeben. Durch Angabe einer (beliebigen) Abbildung und der Lage von O können also Bildpunkte berechnet werden. Es sei noch erwähnt, dass, ist man nur an dem anschaulichen Bild interessiert, man den Aufriss nicht komplett konstruieren muss. WinCAG benutzt den Punkt P'' nur zur Bestimmung der Höhe (z -Komponente). Diese müssen also nur an einer Stelle abgetragen werden. Das rechte Bild zeigt eine typische Anwendung.

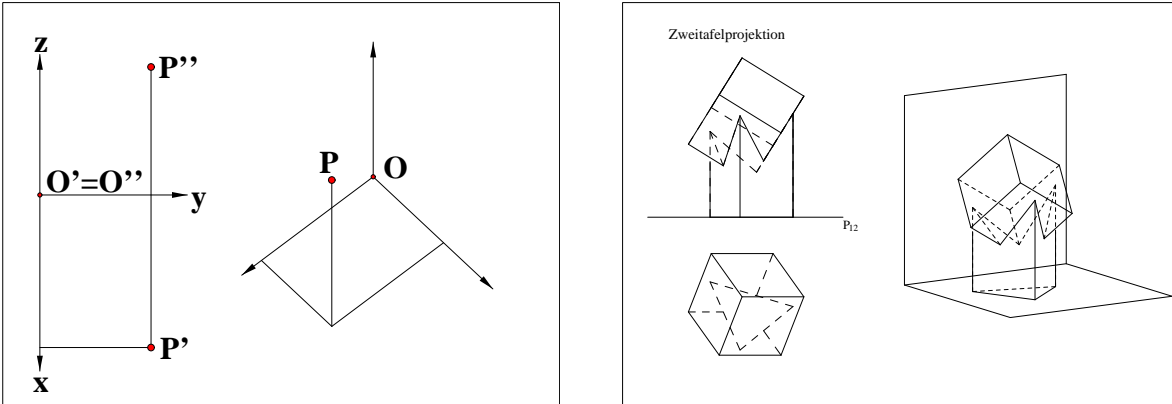


Abbildung 3: Das Abbildungsmodul

Die eigentliche Konstruktion erfolgt in der Zweifafelprojektion. Zur Visualisierung bildet man dann die erforderlichen Punkte wie oben beschrieben ab. Da das System die Objekte — hier Prisma und Würfel — aber nicht kennt, muss man die erforderlichen Linien oder Schraffuren hinzufügen.

Die 3d-Information wird nur für die Abbildung selbst verwendet. Man muss also danach wieder konstruktive Kenntnisse einsetzen, um die Objekte korrekt zu zeichnen. WinCAG stellt aber spezielle Befehle zur Verfügung, die auch hier zu einfachen Konstruktionen führen und bei der Sichtbarkeitsklärung hilfreich sind (s.u.).

Auch die Abbildungen können verändert und animiert werden. Die Abbildungsart selbst (orthogonale Axonometrie, Grundrissaxonometrie, Perspektive, usw.) ist ein Parameter, der für die Demos eingestellt werden kann. Auch alle anderen Bestimmungsgrößen/Parameter der Abbildung können wie für Punkte beschrieben automatisch modifiziert werden. Für die Konstellation im rechten Bild von Abbildung 3 kann man z.B. die Blickrichtung für die Axonometrie (*Längen- und Breitengrad*) völlig analog zu **ModifyPoint** behandeln. Dadurch tritt die Räumlichkeit des Objektes deutlich hervor. Auch die Änderungen, die das Objekt selbst betreffen und in der Zweifafelprojektion vollzogen werden, wirken sich unmittelbar im axonometrischen Bild aus.

4 Spezielle Befehle

Da WinCAG eine spezielle Ausrichtung auf die Darstellende Geometrie hat, sind hierfür spezielle Konstruktions- und Zeichenbefehle integriert. Abbildung 4 zeigt zwei typische Befehle dieser Art. Sollen Kreise oder Halbkreise abgebildet werden, so muss man nur das berandende Rechteck abbilden. Da die zugehörige Berechnung auch für Ellipsen gültig ist, kann man damit etwa auch den Schnitt eines zylindrischen Vordaches mit einem ebenen Hauptdach sehr einfach konstruieren. Im rechten Bild sieht man die Wirkung eines Befehls, der zur Unterstützung der Sichtbarkeitsklärung und beim Einzeichnen von *Horizonten* in Perspektiven konzipiert wurde. Eine Linie kann durch mehrere andere Objekte verdeckt werden. Es werden nur die Teile

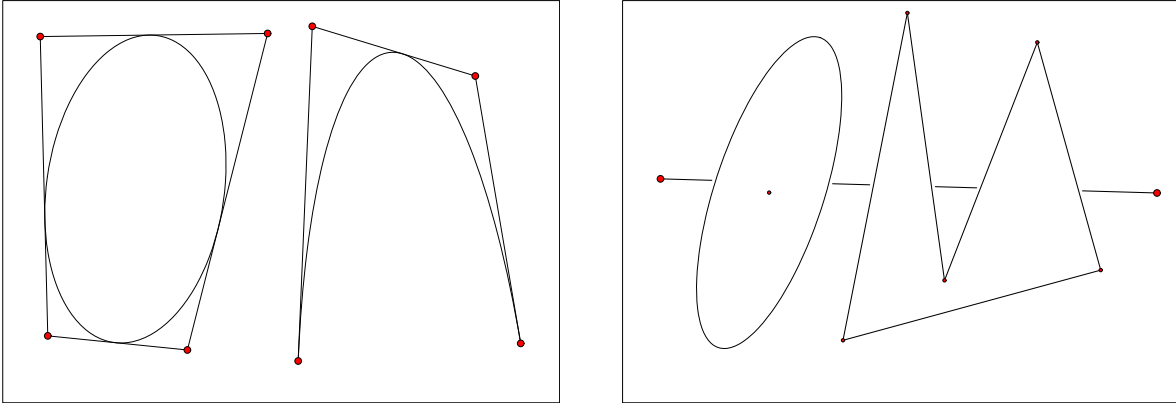


Abbildung 4: Spezielle Konstruktionen und Zeichenbefehle

gezeichnet, die nicht verdeckt sind.

Abbildung 5 zeigt eine weitere Anwendung. Hier wurde zunächst ein quadratischer Turm mit aufgesetzter Pyramide konstruiert. Dieser wurde dann dupliziert und verschoben und mit der oben beschriebenen Methode zu einem zylindrischen Turm mit aufgesetztem Kegel umgearbeitet. Hier wurden auch die Befehle *Tangente-Ellipse-Ellipse* und *Tangente-Punkt-Ellipse* verwendet. Weitere Beispiele zur Sichtbarkeitsklärung und zu Horizonten findet man in [5].

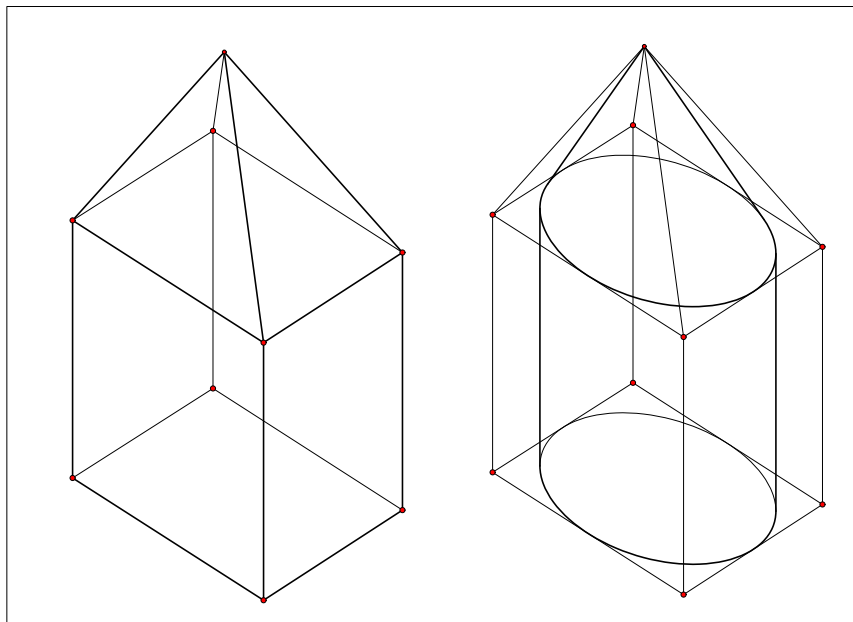


Abbildung 5: Spezielle Konstruktionen und Zeichenbefehle

5 Abschließende Bemerkungen

In dieser Arbeit konnten nur wenige, wichtige Aspekte angerissen werden. Den besten Eindruck eines solchen Systems bekommt man, wenn man sich Demos dazu ansieht. Wie bereits oben erwähnt kann das Programm WinCAG in der in [5] vorgestellten Version unter www.dg-ac.de/wincag für Lehr- und Lernzwecke kostenlos heruntergeladen werden. Wir vervollständigen das Angebot an Demos ständig. Weiterhin arbeiten wir an einer automatischen Anpassung der Geschwindigkeiten an die Rechnerleistung.

Literatur

- [1] Brakhage, Karl–Heinz, Nitschke, Martin *Ein zweidimensionales CAG-System zur Ingenieurausbildung in Darstellender Geometrie*, Institutsbericht Nr. 47, Institut für Geometrie und Praktische Mathematik der RWTH Aachen (1987)
- [2] Brakhage, Karl–Heinz, *Ein menugesteuertes, intelligentes System zur zwei- und dreidimensionalen Computergeometrie*, Fortschrittsberichte, VDI Reihe 20, Nr. 26, VDI Verlag (1990)
- [3] Kortenkamp, Ulrich, *Foundations of Dynamic Geometry*, Ph.D. thesis, ETH Zürich (1999)
- [4] Pütz, Claus *Arbeitsblätter zur Darstellenden Geometrie für Architekten*, ISBN 3-8265-6292-5, Shaker Verlag, Aachen (2002)
- [5] Pütz, Claus, *Einsatz des dynamischen Geometrieprogramms WinCAG in der Vorlesung zu DG für Architekten*, this proceedings
- [6] Richter–Gebert, Jürgen, Kortenkamp, Ulrich, *Dynamische Geometrie: Grundlagen und Möglichkeiten*, <http://www.inf.ethz.ch/~richter>

Karl-Heinz Brakhage
Institut für Geometrie und Praktische Mathematik
RWTH Aachen
Templergraben 55
D–52056 Aachen