

# Dreidimensionale Computergrafik und Analytische Geometrie

## Vorschläge für den Mathematikunterricht in der S II

von

Andreas Filler, Berlin

**Zusammenfassung:** Die dreidimensionale Computergrafik übt auf viele Jugendliche einen großen Reiz aus und weist zudem eine Reihe interessanter Bezüge zu den Inhalten des Mathematikunterrichts in der Sekundarstufe II auf. Auf der Grundlage einer Diskussion von Zielen und Defiziten des Unterrichts im Stoffgebiet Analytische Geometrie sowie einer zusammenfassenden Darstellung der mathematischen Grundlagen der 3D-Computergrafik werden in diesem Beitrag Vorschläge für deren Einbeziehung in den Unterricht unterbreitet.

**Summary:** Applications of 3D computer graphics are very attractive for a lot of young people. There are close connections between the mathematical foundations of this topic and classical contents of mathematics education in grammar schools. Based on the discussion of aims and problems of instruction in analytic geometry and a comprising description of mathematical foundations of 3D computer graphics, proposals for using the potential of computer graphics in mathematics education are offered in this article.

### 1 Gründe für die Einbeziehung von Elementen der 3D-Computergrafik in das Stoffgebiet Analytische Geometrie

In diesem Kapitel werden einige Positionen zu Problemen des Unterrichts im Stoffgebiet Analytische Geometrie skizziert sowie prinzipielle Möglichkeiten und Ziele bei der Einbeziehung der 3D-Computergrafik in dieses Stoffgebiet umrissen.

#### 1.1 Zu gegenwärtigen Problemen des Unterrichts in Analytischer Geometrie

Die Diskussion der letzten Jahre über die Weiterentwicklung des Mathematikunterrichts in der Sekundarstufe II ist von dem Ansatz geprägt, den allgemeinbildenden Charakter des Unterrichts zu stärken, wobei sich die Bestimmung des Begriffs Allgemeinbildung in Bezug auf den Mathematikunterricht meist auf die von WINTER in [24] gegebene Charakterisierung stützt. Der allgemeinbildende Wert der Analytischen Geometrie wird dabei (u. a. in [3] und [6]) in „ihren mächtigen Methoden und interessanten Objekten zur Erschließung des uns umgebenden Raumes“ gesehen. Jedoch konstatiert [3] schwer wiegende Defizite des Mathematikunterrichts

der S II in Hinblick auf den Bezug des Unterrichts zu relevanten Problemen sowie hinsichtlich des Erwerbs heuristischer Fähigkeiten durch die Schüler (S. 79). Ein weiteres wesentliches Defizit des Unterrichts im Stoffgebiet Analytische Geometrie besteht darin, dass Schüler (vor allem in Grundkursen) oft nur noch lineare Objekte kennen lernen. Diese Formenarmut beklagt u. a. SCHUPP in [17]. Über die „Aufgabe des Geometrieunterrichts, seine Gegenstände als Vehikel der Welter-schließung zu erarbeiten und anzuwenden“ schreibt er: „Am wenigsten tut er das in der SII. Lineare Gebilde und artifizielle Graphen sind nicht nur inner-, sondern auch außermathematisch entweder bloße Hilfen oder bedeutungsarm.“(S. 52).

Der weitaus größte Teil der für den Unterricht im Stoffgebiet Analytische Geometrie vorgesehenen Zeit ist derzeit nach den meisten Rahmenplänen für die Behandlung der Geraden und Ebenen im Raum vorgesehen, wobei deren Beschreibung durch Gleichungen, Aufgaben zu Lagebeziehungen und Schnittverhalten sowie Abstands- und Winkelberechnungen im Mittelpunkt stehen. Damit ergibt sich eine oftmals recht einseitige Orientierung des Unterrichts auf das Einüben von Kalkülen und das Lösen von Routineaufgaben. So kritisiert SCHUPP die „Überwucherung durch Beschreibungs- und Methodendrill“ und fordert, „die Methodendemonstration (die auf Dauer unweigerlich mit Objektverarmung verbunden ist) weit häufiger als bisher ab(zu)lösen, zumindest (zu) unterbrechen durch Objektexploration“ ([17], S. 57). Diese Forderung erscheint auch insofern bedeutsam, als die Leistungsfähigkeit von Abstraktionen (wie z. B. des Vektorbegriffs), Beschreibungen (wie impliziten Gleichungen und Parameterdarstellungen) sowie Verfahren (wie der Zurückführung von Schnittmengen auf Lösungsmengen von Gleichungssystemen) für die Schüler nicht genügend nachvollziehbar wird, wenn sie diese nur an wenigen (und zudem recht uninteressanten) Objekten erfahren und anwenden.

Zusammengefasst ist es für die Weiterentwicklung des Unterrichts im Stoffgebiet Analytische Geometrie notwendig, in stärkerem Maße *Anknüpfungspunkte aus der Lebensumgebung der Schüler zu berücksichtigen und heuristischen Arbeitsweisen eine größere Bedeutung beizumessen*. Der Unterricht bedarf *einer größeren Vielfalt zu untersuchender interessanter geometrischer Objekte* und einer *flexibleren Herangehensweise bei der Auswahl von Beschreibungen und Methoden*. Die Aufnahme von Elementen der 3D-Computergrafik und dadurch ermöglichte visuelle und experimentelle Vorgehensweisen können, wie in diesem Beitrag aufgezeigt werden soll, für die Erfüllung dieser Ziele wertvolle Beiträge leisten.

## **1.2 Zum Einsatz von Computern im Unterricht der Analytischen Geometrie unter besonderer Berücksichtigung computergrafischer Anwendungen**

Fragen des Computereinsatzes im Mathematikunterricht nehmen in der didaktischen Diskussion der vergangenen zwei Jahrzehnte eine wichtige Stellung ein. Im Stoffgebiet Analytische Geometrie überwiegt dabei die Nutzung von Computeralgebrasystemen (CAS) für das Lösen von Gleichungssystemen sowie zur Visuali-

sierung geometrischer Objekte und Lagebeziehungen in der Ebene und (teilweise) im Raum.<sup>1</sup> Weiter gehende Vorschläge beziehen sich auf die Arbeit mit Matrizen und berücksichtigen dabei teilweise deren Anwendungen in der Computergrafik.<sup>2</sup>

Ausgehend von der Darstellung der mathematischen Grundlagen der Computergrafik untersuchte GIEDING 1988 in [9] systematisch die Möglichkeiten für deren Einbeziehung in den Mathematikunterricht. Die Realisierung seiner Vorschläge erforderte entweder die Programmierung der verwendeten Prozeduren in einer Programmiersprache wie Pascal oder blieb theoretischer Natur. Um jedoch Inhalte der Analytischen Geometrie mit deren Anwendungen in der Computergrafik überzeugend zu motivieren, ist es sinnvoll, dass die Schüler mit computergrafischen Anwendungen arbeiten und die Nützlichkeit des Erkennens ihrer „mathematischen Funktionsweise“ bei dieser Arbeit erleben können. Ein wichtiger Aspekt der motivierenden Wirkung der Computergrafik besteht ohne Zweifel im ästhetischen Reiz von Bildern, welche die Schüler selbst erstellen. Die technischen Voraussetzungen dafür waren Ende der achtziger Jahre nur in sehr geringem Umfang gegeben.

Da die Bedeutung computergrafischer Anwendungen in den vergangenen Jahren stark gewachsen ist und Computergrafik zu einem Teil der „Lebenswirklichkeit“ von Jugendlichen wurde,<sup>3</sup> erscheint es geboten, ausgehend vom aktuellen Stand ihrer Entwicklung und den inzwischen recht guten hard- und softwareseitigen Voraussetzungen, eine Analyse der Inhalte der Computergrafik hinsichtlich ihrer Einbeziehung in den Unterricht vorzunehmen und geeignete Unterrichtskonzepte hierfür zu entwickeln. So forderte SCHUPP in seinem bereits erwähnten Beitrag [17]:

*„Die Schule ... sollte versuchen, die enormen visuellen Potenzen der neuen Medien zu enkultivieren. Geometrieunterricht kann und muß dazu einen wichtigen Beitrag leisten, nämlich das verweilende Arbeiten am Computerbild inszenieren, seine Konstruktion, Fokussierung, Anreicherung, Exploration, Dynamisierung, Variation ermöglichen und einfordern. Mit dem bloßen Visualisieren anders und anderwärts geklärter Sachverhalte ... ist noch nicht viel erreicht, zumal wir heute wissen, daß Bilder nicht so sehr als methodische Hilfe denn als eine wesentliche Komponente des Wissens selbst aufgefasst werden müssen.“* ([17], S. 54)

Für die Weiterentwicklung des Unterrichts der Analytischen Geometrie und die Überwindung wesentlicher eingangs dieses Beitrags aufgeführter Probleme des Unterrichts in diesem Stoffgebiet bietet die Einbeziehung von Elementen der 3D-Computergrafik eine Reihe wertvoller Möglichkeiten:

- 
- <sup>1</sup> Zum Teil wurden für Visualisierungen einfacher geometrischer Objekte des Raumes auch spezielle Programme entwickelt, siehe u. a. [1], [15] und [23].
  - <sup>2</sup> Siehe z. B. [13], [14] und [25].
  - <sup>3</sup> Erwähnt seien hier nur Titelseiten von Zeitschriften, Werbespots jeder Art, MTV, Spielfilme, ja sogar der Wetterbericht des Fernsehens und natürlich Computerspiele.

- Durch die Ergänzung der algebraischen Untersuchung geometrischer Objekte und Relationen durch computergrafische Visualisierungen kann eine „Geometrisierung“ des Unterrichts in Analytischer Geometrie erreicht werden.
- Die Verwendung computergrafischer Software ermöglicht heuristische und experimentelle Wege der Untersuchung geometrischer Objekte und damit die verstärkte Einbeziehung dieser Arbeitsweisen in den Unterricht.
- Durch den mithilfe von 3D-Grafiksoftware möglichen visuellen und experimentellen Zugang eröffnen sich Möglichkeiten, interessante geometrische Objekte zu betrachten und damit eine größere Formenvielfalt zu erreichen.
- Die Schüler erfahren die enorme Anwendungsrelevanz der Analytischen Geometrie auf einem Gebiet, das für sie sehr attraktiv ist. Sie können den Nutzen ihres mathematischen Wissens bei der Gestaltung von Computerbildern erleben, indem sie vom „Spielen“ zu einem systematischen Vorgehen finden.
- Das Herstellen stoffgebiets- und fächerübergreifender Bezüge (u. a. zur Analysis, Physik, Informatik und Kunst) wird erleichtert oder erst ermöglicht.
- Nicht zuletzt kann der ästhetische Reiz dreidimensionaler computergrafischer Darstellungen eine Motivierung der Schüler für den Unterricht bewirken.

Im Gegensatz zu anderen Einsatzgebieten des Computers im Mathematikunterricht kann die 3D-Computergrafik nicht nur als wertvolles *Hilfsmittel* für den Unterricht dienen, sondern auch als *Unterrichtsgegenstand* den Schülern die Bedeutung und Nützlichkeit grundlegender Inhalte und Arbeitsweisen der Analytischen Geometrie an einem praxisrelevanten und interessanten Gegenstand verdeutlichen. Eine Konzeption für die Einbeziehung der 3D-Computergrafik in den Unterricht sollte daher so aufgebaut sein, dass sich theoretische Überlegungen und die praktische Arbeit mit computergrafischer Software gegenseitig anregen und befruchten. Dafür bieten sich folgende Unterrichtsszenarien an:

- Die Schüler entdecken durch visuelles Experimentieren „neue“ geometrische Objekte (z. B. als Schnittmengen), die sie anschließend mathematisch beschreiben. Durch anhand von Gleichungen gewonnene Verallgemeinerungen modellieren sie weitere Formen, veranschaulichen und untersuchen diese.
- Durch die Arbeit mit der Software werden Fragen nach deren Funktionsweise aufgeworfen, die zunächst praktisch motiviert sind: Welche Parameter sind wie zu setzen, um bestimmte Effekte zu erzielen? Davon ausgehend werden mathematische Modellbildungen erarbeitet und praktisch überprüft.

Für beide Szenarien werden im dritten Teil dieses Artikels Vorschläge unterbreitet. Zunächst soll jedoch ein Überblick über mathematische Grundlagen und wichtige Vorgehensweisen der 3D-Computergrafik deren inhaltliche Bezüge zur Analytischen Geometrie verdeutlichen.

## 2 Mathematische Grundlagen der 3D-Computergrafik und ihre Umsetzung in 3D-Grafiksoftware

In diesem Abschnitt werden die wichtigsten mathematischen Grundlagen der dreidimensionalen Computergrafik überblicksartig dargestellt. Dabei wird der Schwerpunkt auf die geometrischen und optischen Aspekte gelegt, während die programmiertechnische Umsetzung nur am Rande Erwähnung findet, da diese für den Mathematikunterricht nur von untergeordneter Bedeutung bleiben kann. Ergänzt werden die Grundlagen der 3D-Computergrafik um Belange der praktischen Anwendung. Dazu werden Beispiele anhand aktueller 3D-Grafiksoftware dargestellt.<sup>4</sup>

Der 3D-Computergrafik liegt die Vorstellung einer virtuellen Kamera zugrunde, durch die ein Ausschnitt des dreidimensionalen Raumes „fotografiert“, d. h. in eine Ebene projiziert wird. Das in Abb. 1 dargestellte, sehr einfache Beispiel wurde mit POV-Ray durch die Eingabe der folgenden Szenenbeschreibung erzeugt:<sup>5</sup>

```
background {White}
camera
  {location <0,10,-20>
  look_at <0,0,0>
  angle 10}
light_source {<-70,0,-50> color White}
light_source {<80,100,-50> color White}
sphere { <1,2,-1> 1
  texture{pigment {color Blue}} }
```

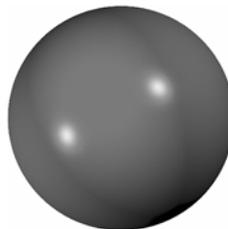


Abb. 1

Die Anweisung **background** legt in dieser Szenenbeschreibung die Farbe des Hintergrundes fest. Durch **camera** werden die Koordinaten und die Blickrichtung sowie der Öffnungswinkel der Kamera beschrieben, durch welche die Abbildung der Szene in eine Projektionsebene erfolgt (siehe Abschnitt 2.4). Jede Lichtquelle (**light\_source**), welche die Szene beleuchtet, muss durch ihre Koordinaten und die Farbe des abgegebenen Lichtes beschrieben werden. Schließlich wird durch **sphere** eine Kugel mit den Mittelpunktskoordinaten (1;2;-1) und dem Radius 1 erzeugt. Die Anweisung **texture** (siehe dazu Abschnitt 2.5) weist dieser Kugel Oberflächeneigenschaften (hier lediglich eine Farbe) zu.

<sup>4</sup> Die meisten der in diesem Artikel enthaltenen Beispiele wurden mit der Software POV-Ray erstellt (siehe [27]). POV-Ray ist ein hochwertiges, frei verfügbares 3D-Grafikprogramm, das sich sehr gut für die Verdeutlichung der mathematischen Grundlagen der 3D-Computergrafik eignet, da Grafiken durch eine Szenenbeschreibungssprache erzeugt und geometrische Objekte dabei durch ihre Koordinaten bzw. Gleichungen beschrieben werden müssen. Grundsätzlich sind jedoch alle hier dargestellten Verfahren auch auf andere (mausgesteuerte) 3D-Grafikprogramme übertragbar.

<sup>5</sup> Alle in diesem Artikel vorgestellten Beispieldateien sowie farbige Versionen der Abbildungen sind auf der Internetseite [26] zugänglich.

Wie dieses einfache Beispiel zeigt, sind für die Erstellung dreidimensionaler computergrafischer Darstellungen drei Aspekte maßgeblich:

1. Repräsentation, Manipulation und gegebenenfalls Animation von Körpern des Raumes durch Koordinaten bzw. Gleichungen in räumlichen Koordinaten.
2. Festlegung von Oberflächeneigenschaften der Objekte und Beleuchtung der Szene (Definition von Lichtquellen).
3. Abbildung (Projektion) der Szene in eine Ebene, die dann auf dem Bildschirm dargestellt wird. 3D-Grafikprogramme berechnen unter Berücksichtigung optischer Gegebenheiten möglichst „fotorealistische“ Abbildungen.

## 2.1 Beschreibung von Punkten durch Koordinaten im Raum

Punkte des dreidimensionalen Raumes werden in der euklidischen Geometrie durch Koordinatentripel bezüglich eines kartesischen Koordinatensystems beschrieben. In der Computergrafik ist auch die Verwendung der aus der projektiven Geometrie bekannten homogenen Koordinaten gebräuchlich. Dabei werden Punkte des Raumes durch Koordinatenquadrupel  $(x;y;z:w)$  dargestellt, wobei den homogenen Koordinaten  $(x;y;z:w)$  die kartesischen Koordinaten  $\left(\frac{x}{w}; \frac{y}{w}; \frac{z}{w}\right)$  entsprechen

(für  $w=0$  werden uneigentliche, d. h. „unendlich ferne“ Punkte beschrieben). Homogene Koordinaten werden für interne Berechnungen in Computergrafik-Programmen genutzt, weil alle Transformationen des Raumes durch  $4 \times 4$ -Matrizen beschrieben werden können (die Darstellung von Verschiebungen durch  $3 \times 3$ -Matrizen bezüglich kartesischer Koordinaten ist nicht möglich) und Zentral- und Parallelprojektionen bezüglich homogener Koordinaten durch einheitliche Matrizen darstellbar sind. Des Weiteren bietet die Verwendung homogener Koordinaten numerische Vorteile, da durch die freie Wahl von  $w$  beliebige Punkte mit hinreichender Genauigkeit durch Ganzzahlen beschrieben werden können, deren Verarbeitung effizienter erfolgt als die von Gleitkommazahlen. Auf der Benutzerebene (für die Eingabe von Objekten in 3D-Grafiksoftware) werden jedoch kartesische Koordinaten verwendet, die Umsetzung in homogene Koordinaten erfolgt Software-intern. Die folgenden Betrachtungen zur Beschreibung dreidimensionaler Objekte werden daher auf der Grundlage kartesischer Koordinaten geführt.

## 2.2 Modellierung – Mathematische Beschreibung von Körpern und Flächen

Während Körper und Flächen aus mathematischer Sicht wohlunterschiedene Klassen von Objekten sind, ist diese Unterscheidung aus der Sicht der Computergrafik oftmals nicht wesentlich, da Lichtstrahlen i. Allg. mit *Körperoberflächen* interagieren und andererseits durch die Zusammensetzung von Flächen Körper beschrieben werden können. Daher sind sowohl die Beschreibung von geometrischen Körpern

als auch die von Flächen für die Modellierung in der Computergrafik bedeutsam. Im Folgenden werden einige wichtige Darstellungsweisen kurz beschrieben.<sup>6</sup>

### Beschreibung von Grundkörpern („primitives“)

Eine Reihe geometrischer Körper kann in 3D-Software durch die (numerische oder mausgestützte) Eingabe weniger Punkte und Maße beschrieben werden:<sup>7</sup>

- Kugel: `sphere{<x,y,z>,r}` (Koordinaten des Mittelpunktes und Radius),
- Quader: `box{<x1,y1,z1>,<x2,y2,z2>}` (Koordinaten zweier diagonal gegenüberliegender Eckpunkte),
- Gerader Kreiszylinder: `cylinder{<x1,y1,z1>,<x2,y2,z2>,r}` (Koordinaten der Mittelpunkte von Grund- und Deckfläche, Radius),
- Gerader Kreiskegelstumpf: `cone{<x1,y1,z1>,r1,<x2,y2,z2>,r2}` (Koordinaten der Mittelpunkte sowie Radien von Grund- und Deckfläche).

Neben den hier aufgeführten können weitere Körper (wie z. B. Prismen und Tori) in analoger Weise beschrieben und dargestellt werden. Des Weiteren ist es in der 3D-Computergrafik gebräuchlich, Körper durch *Boolesche Operationen* (Vereinigung, Differenz und Durchschnitt) aus Grundkörpern zu erstellen, wodurch bereits eine Vielzahl von Formen modelliert werden kann. Als Beispiel zeigt Abb. 2 die Differenz aus dem Durchschnitt zweier Zylinder und der Vereinigung zweier Zylinder:

```
difference {
  object {intersection {
    object { cylinder{<-2,0,0>,<2,0,0>,2} }
    object { cylinder{<0,-2,0>,<0,2,0>,2} }
  } }
  object {merge {
    object { cylinder{<-2,0,0>,<2,0,0>,1} }
    object { cylinder{<0,-2,0>,<0,2,0>,1} }
  } } }8
```



Abb. 2

### Extrusionskörper

Eine weitere Möglichkeit der Modellierung räumlicher Objekte besteht darin, Körper durch die „Extrusion“ von (meist geschlossenen) Kurven zu erzeugen, hierbei wird eine Kurve an einer anderen Kurve (der Extrusionskurve) entlang „gezogen“.

<sup>6</sup> Ausführliche Darstellungen dieser Thematik finden sich u. a. in [2], [7], [8] und [22].

<sup>7</sup> Die angegebene Beschreibung verwendet die Syntax der 3D-Grafiksoftware POV-Ray.

<sup>8</sup> Dabei steht **difference** für die Differenz, **intersection** für den Durchschnitt und **merge** für die Vereinigung zweier Objekte.

Ein Beispiel hierfür sind die Rotationskörper, bei denen eine gegebene Kurve entlang eines Kreises extrudiert wird. Kommt als Extrusionskurve eine Gerade (bzw. Strecke) zur Anwendung, entsteht ein allgemeiner Zylinder. Abb. 3 zeigt als weiteres Beispiel die Extrusion eines Kreises entlang einer Bézierkurve.

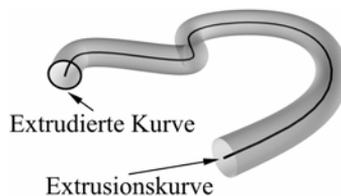


Abb. 3

### Flächendarstellung durch Funktionsgleichungen und implizite Gleichungen

Flächen des Raumes, die als Grafen von Funktionen zweier Variabler darstellbar sind, können durch Gleichungen der Form

$$z = f(x, y) \quad (1)$$

beschrieben werden. Basierend auf der Überlegung, welche Schnittkurven zwischen einer darzustellenden Fläche und der  $x$ - $z$ - bzw. der  $y$ - $z$ -Ebene entstehen müssen, um eine Fläche mit gewünschter Gestalt zu erhalten, ist es möglich, durch Funktionsgleichungen eine Reihe von Formen zu modellieren.

Als Beispiel zeigt Abb. 4 eine

Fläche, die mittels der Gleichung

$$z = \sin x \cdot \sin y \quad (x, y \in [-10; 10])$$

in POV-Ray erzeugt wurde.



Abb. 4

Die Vielfalt der durch Funktionsgleichungen darstellbaren Flächen ist durch die notwendige Eindeutigkeit der Zuordnung  $(x, y) \rightarrow z$  beschränkt. Diese Beschränkung wird durch die Verwendung impliziter Gleichungen

$$f(x, y, z) = 0 \quad (2)$$

vermieden. Ist dabei  $f$  eine lineare Funktion in den Variablen  $x$ ,  $y$  und  $z$ , so beschreibt die Gleichung (2) eine Ebene, für eine quadratische Funktion  $f$  eine Fläche zweiter Ordnung. Als weiteres Beispiel zeigt Abb. 5 die mittels der Gleichung

$$\frac{\sin(10x)}{8} + \sqrt{z^2 + y^2} - 0,5 = 0$$

erzeugte Rotationsfläche:

`isosurface`<sup>9</sup>

```
{ function {sin(10*x) / 8
+ ((sqrt(z*z+y*y))-0.5)}
contained_by {box{-2,2}} }
```

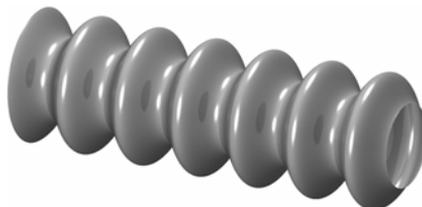


Abb. 5

<sup>9</sup> Die Anweisung `isosurface` wird in POV-Ray für die Erzeugung von Flächen durch implizite Gleichungen verwendet.

## Darstellung von Flächen durch Parameterdarstellungen

Die flexibelste Form der geschlossenen mathematischen Darstellung von Flächen ergibt sich durch die Einführung zweier Parameter  $(u, v)$  und die Darstellung der drei Raumkoordinaten als Funktionen

$$x = f_x(u, v), \quad y = f_y(u, v) \quad \text{sowie} \quad z = f_z(u, v) \quad (3)$$

der beiden Parameter. Auch diese Beschreibung beinhaltet die Darstellung von Flächen durch Funktionsgleichungen als Spezialfall (z. B. mit  $u = x, v = y$ ).

Sind  $f_x, f_y$  und  $f_z$  lineare Funktionen, so werden durch (3) i. Allg. Ebenen beschrieben. Bei Verwendung nichtlinearer Funktionen lässt sich eine große Vielfalt von Flächen modellieren. Als Beispiel für eine durch eine Parameterdarstellung beschriebene Fläche zeigt Abb. 6 die durch

$$x = u \cdot v \cdot \sin(15v); \quad y = u \cdot v \cdot \cos(15v); \quad z = v$$

(mit  $u \in [0;1], v \in [-1;1]$ ) gegebene konische Spirale.



Abb. 6

## Freiformflächen

Besonders flexible Möglichkeiten der Modellierung bieten Flächen, die aus vorgegebenen Punkten approximiert werden. Außer durch die Punkte selbst kann die Gestalt derartiger Flächen durch vorgegebene Tangenten beeinflusst werden. Realisiert werden Freiformflächen i. Allg. durch die stetig differenzierbare Zusammensetzung bikubischer Flächenstücke<sup>10</sup>, also von Flächenstücken, die sich als Funktionen 3. Grades in den Parametern  $u$  und  $v$  darstellen lassen:

$$x(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 a_{i,j}^{(x)} u^i v^j, \quad y(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 a_{i,j}^{(y)} u^i v^j, \quad z(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 a_{i,j}^{(z)} u^i v^j.$$

Dabei sind  $A_{i,j} = (a_{i,j}^{(x)}; a_{i,j}^{(y)}; a_{i,j}^{(z)})$  (mit  $i, j = 0, \dots, 3$ ) die 16 Koeffizientenvektoren des Flächenstücks. Da im Grafikdesign zumeist nicht die Aufgabe besteht, Flächen zu entwerfen, die exakt durch vorgegebene Punkte verlaufen, sondern vielmehr beabsichtigt wird, gegebene Punkte durch möglichst „glatte“ Flächen anzunähern und dabei an vorgegebenen Stellen bestimmte Richtungen einzuhalten, werden die Koeffizientenvektoren durch Kontrollpunkte und ggf. zugehörige Tangenten gesteuert. So werden *Bézierflächen* durch Kontrollpunkte und vorgegebene Tangenten erzeugt. Sie stellen die Übertragung der Bézierkurven auf Flächen des Raumes dar.

<sup>10</sup> Die Verwendung bikubischer Funktionen hat sich in der Computergrafik durchgesetzt, da biquadratische Funktionen keine genügend flexible Annäherung an vorgegebene Punkte ermöglichen und Funktionen höherer Ordnung zu sehr komplizierten Interpolationsalgorithmen führen.

In der 3D-Computergrafik haben in den letzten Jahren die *NURBS-Flächen* (Non-Uniform Rational B-Splines) größte Bedeutung erlangt, die durch Kontrollpunkte und zugeordnete Gewichte beschrieben werden (siehe u. a. [2]).

### 2.3 Aufbereitung der modellierten Körper und Flächen für die Weiterverarbeitung in 3D-Grafiksoftware

Grafikprogramme können nicht alle der im vorangegangenen Abschnitt beschriebenen Darstellungsformen unmittelbar für die Bildberechnung (Rendering) verarbeiten. Intern nehmen sie daher meist eine Umwandlung der eingegebenen Flächen (bzw. Körper) in Polygonnetze vor,<sup>11</sup> wobei sich die verschiedenen 3D-Grafikprogramme darin unterscheiden, inwiefern auch einfache Grundkörper (wie z. B. Kugeln) als Polyeder repräsentiert werden.<sup>12</sup>

Für die Umwandlung einer Fläche in ein Gitternetz werden die Intervalle, auf denen die Parameterdarstellung der Fläche gegeben ist, in Teilintervalle zerlegt und für deren Randpunkte die Koordinaten der Flächenpunkte berechnet. Diese Methode wird auch bei der Flächendarstellung von Computeralgebrasytemen sichtbar. So wird in Mathematica die Feinheit der Intervalle mit der Anweisung „Plot-Points“ festgelegt; Abb. 7 zeigt die bereits in Abb. 6 dargestellte (dort mit POV-Ray erzeugte) konische Spirale:

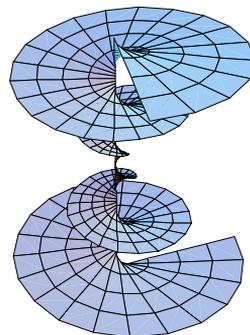


Abb. 7

```
ParametricPlot3D[
  {u*v*SIN[15*v] , u*v*COS[15*v] , v},
  {u,0,1},{v,-1,1},PlotPoints->{5,100}]
```

Für den Parameter  $v$  muss eine recht feine Unterteilung gewählt werden, damit die Gestalt der Fläche sichtbar wird, während für  $u$  eine gröbere Einteilung ausreicht. Genau betrachtet ist in Abb. 7 nicht die durch die Parameterdarstellung beschriebene Fläche dargestellt, sondern ein Polygonnetz mit 500 Eckpunkten.

Die meisten Grafikprogramme bereiten Flächen vor der Bildberechnung als Mengen von Dreiecken auf, die Polygonnetze werden also „trianguliert“. Die Ursache hierfür liegt darin, dass für die Bildberechnung die Oberflächennormalen eine

<sup>11</sup> Neben der Repräsentation durch Polygonnetze existieren noch andere Möglichkeiten, z. B. durch eine Diskretisierung des Raumes (Voxelrepräsentation, siehe u. a. [21]), die jedoch keine so breite Anwendung gefunden haben.

<sup>12</sup> Viele Grafikpakete betrachten Kugeln intern als Polyeder; bei der Konstruktion ist die Anzahl der Segmente einzugeben. Das hier betrachtete Programm POV-Ray stellt die Grundkörper hingegen exakt anhand ihrer Gleichungen dar. Parametrisierte Flächen sowie andere Objekte werden jedoch auch durch POV-Ray in Gitternetze umgewandelt.

wichtige Bedeutung haben (wie noch beschrieben wird) und sich Dreiecken eindeutig Ebenen und somit Normaleneinheitsvektoren zuordnen lassen.

Mittels eines von MAEDER (vgl. [11]) entwickelten Zuzsatzes wurde die in Abb. 7 dargestellte Fläche in eine für die Weiterverarbeitung in POV-Ray geeignete Menge von Dreiecken konvertiert; sie wird dabei durch die Vereinigung von 792 Dreiecken mit der Syntax

```
triangle {< $x_1, y_1, z_1$ >, < $x_2, y_2, z_2$ >, < $x_3, y_3, z_3$ >}
```

repräsentiert. Abb. 8 zeigt ausschnittsweise das von POV-Ray berechnete Bild dieser Repräsentation, wobei harte Farbübergänge zwischen den Dreiecken auffallen.



Abb. 8: Flat-Shading

Die Ursache für diesen, meist unerwünschten, Effekt besteht darin, dass die Schattierung (Ausleuchtung) einer Facette mittels ihrer Normalen erfolgt.<sup>13</sup> Da sich beim Übergang von einem Dreieck zu einem anderen der Normalenvektor abrupt ändert, entstehen harte Farbübergänge. Um dies zu vermeiden, wurde von GOURAUD<sup>14</sup> ein Verfahren entwickelt, die Helligkeits- bzw. Farbwerte innerhalb der einzelnen Facetten zu interpolieren. Dazu wird jedem Eckpunkt eines Dreiecks ein Einheitsvektor zugeordnet, der durch Berechnung der Mittelwerte der Normaleneinheitsvektoren aller an diesem Punkt anliegenden Dreiecke entsteht. Ein Dreieck wird somit durch die drei Eckpunkte und die diesen Eckpunkten zugeordneten Normaleneinheitsvektoren beschrieben:<sup>15</sup>

```
smooth_triangle {
  < $x_1, y_1, z_1$ >, < $xn_1, yn_1, zn_1$ >,
  < $x_2, y_2, z_2$ >, < $xn_2, yn_2, zn_2$ >,
  < $x_3, y_3, z_3$ >, < $xn_3, yn_3, zn_3$ > }.
```

Bei der Berechnung der Farbe eines Punktes wird nun ein (in Abhängigkeit der Entfernung dieses Punktes von den Eckpunkten des Dreiecks) gewichteter Mittelwert aus den Eckpunktnormalen verwendet, wodurch sich allmähliche Helligkeitsverläufe ergeben (Abb. 9).

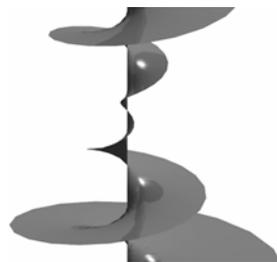


Abb. 9: Phong-Shading

<sup>13</sup> Die Helligkeitswerte hängen (wie weiter unten beschrieben wird) von den Winkeln zwischen den Flächennormalen und den Verbindungsvektoren der betrachteten Punkte zu den Lichtquellen sowie zum Beobachter (Kamera) ab.

<sup>14</sup> Dieses als Gouraud-Shading bekannte Verfahren wurde erstmals 1971 in [10] beschrieben und 1975 von PHONG weiterentwickelt (siehe auch u. a. [5], [7], [8], [21] und [22]).

<sup>15</sup> Die Berechnung dieser Vektoren wird durch Computergrafikprogramme intern vorgenommen. Für das CAS Mathematica ist in [11] das Paket SurfaceGraphics3D beschrieben, welches die Eckpunktnormalen erzeugt und „geglättete Dreiecke“ exportiert.

## 2.4 Transformationen

Geometrische Transformationen und ihre Beschreibung, vor allem durch Matrizen, sind in der 3D-Computergrafik in mehreren Zusammenhängen von Bedeutung:

- Oft ist es sinnvoll, Position und Größe geometrischer Objekte nach ihrer Erstellung zu verändern, da die Wahl einer speziellen Lage gegenüber dem Koordinatensystem die Beschreibung durch Koordinaten oder Gleichungen häufig vereinfacht. Durch geeignete Transformationen können die Objekte dann in die gewünschte Lage und Größe gebracht werden.
- Für die Erstellung von Animationen (Filmen) müssen vielfältige zeitabhängige Transformationen durchgeführt werden.
- Zur letztendlichen Bilderzeugung erfolgt die Projektion des dreidimensionalen Raumes in eine Ebene, wobei Zentralprojektionen, in Ausnahmefällen Parallelprojektionen, vorgenommen werden.
- Um Objekte mit Oberflächen zu versehen (zu texturieren), werden darauf Bilder projiziert. Dabei sind ebene, sphärische und zylindrische Projektionen, vor allem aber das sogenannte  $u$ - $v$ -Mapping von Bedeutung, bei dem die Koordinaten des zweidimensionalen Bildes durch geeignete Vorschriften den Parametern der Fläche zugeordnet werden, auf die das Bild projiziert wird.<sup>16</sup>

### Affine Abbildungen des Raumes auf sich, Koordinatentransformationen

Während affine Abbildungen jedem Punkt des Raumes einen Bildpunkt zuordnen, interessieren in der Computergrafik oftmals nur die Bildpunkte eines einzelnen Objektes, auf das eine Transformation angewendet wird. Objekte können in 3D-Grafiksoftware verschoben, gedreht, gespiegelt und skaliert (gestreckt) werden.

Abb. 10 b) zeigt die Wirkung dieser Transformationen anhand eines Torus (der in Abb. 10 a) in seiner Ausgangslage abgebildet ist):

```
torus { 1, 0.2
  scale 1.5
  rotate <60,30,0>
  translate<-1,1,1>
}
```

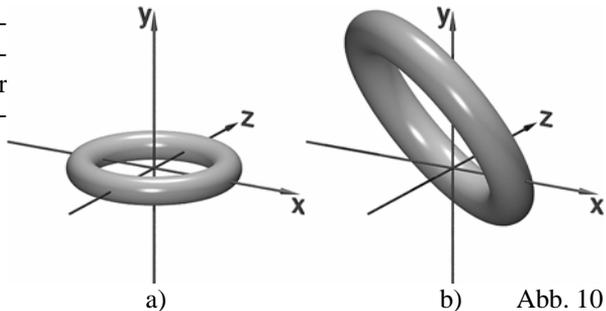


Abb. 10

Die drei Argumente in **rotate**  $\langle \phi_x, \phi_y, \phi_z \rangle$  geben dabei die Drehwinkel um die drei Koordinatenachsen, das Argument von **scale** den Streckungsfaktor einer

<sup>16</sup> Auf den Aspekt der Texturprojektionen wird in diesem Beitrag nicht näher eingegangen, siehe hierzu u. a. [5], [8], [21] und [22].

zentriscen Streckung mit dem Koordinatenursprung als Zentrum und die Argumente von **translate** die Komponenten des Verschiebungsvektors an. Neben dieser elementaren Beschreibung von Transformationen kann auch die wesentlich flexiblere Matrizendarstellung genutzt werden, welche die Software intern ohnehin für die Berechnung der Bildkoordinaten verwendet.

Alle affinen Abbildungen des Raumes können durch Abbildungsgleichungen der Form

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}$$

beschrieben werden. Dabei sind  $x, y, z$  die Koordinaten eines beliebigen Punktes,  $x', y', z'$  die Koordinaten des zugehörigen

Bildpunktes,  $\begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}$  der Translationsvektor und  $\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$  die Matrix der zugehörigen linearen Abbildung.

Alternativ zu dieser Darstellung können affine Abbildungen durch  $4 \times 4$ -Matrizen der Form

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & t_x \\ a_{21} & a_{22} & a_{23} & t_y \\ a_{31} & a_{32} & a_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{mit} \quad \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = A \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

beschrieben werden, wobei der Vorteil besteht, dass die Nacheinanderausführung beliebiger affiner Abbildungen durch das Produkt der zugehörigen Matrizen repräsentiert wird. In 3D-Grafiksoftware werden Berechnungen daher mit  $4 \times 4$ -Matrizen durchgeführt. In POV-Ray lassen sich Transformationsmatrizen mittels

```
matrix < a11, a12, a13 , a21, a22, a23 , a31, a32, a33 , tx, ty, tz >
```

auf beliebige Objekte anwenden.

Bewegungen lassen sich als Nacheinanderausführungen von Drehungen, Translationen und Spiegelungen durch die Transformationsgleichung

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} + \begin{pmatrix} \cos \phi_z & -\sin \phi_z & 0 \\ \sin \phi_z & \cos \phi_z & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \phi_y & 0 & -\sin \phi_y \\ 0 & 1 & 0 \\ \sin \phi_y & 0 & \cos \phi_y \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi_x & -\sin \phi_x \\ 0 & \sin \phi_x & \cos \phi_x \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \pm 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

bzw. durch  $4 \times 4$ -Matrizen der Form

$$\begin{pmatrix} \cos \phi_y \cos \phi_z & -\cos \phi_z \sin \phi_x \sin \phi_y & -\cos \phi_x \sin \phi_z & \mp \cos \phi_z \cos \phi_x \sin \phi_y \pm \sin \phi_x \sin \phi_z & t_x \\ \cos \phi_y \sin \phi_z & \cos \phi_z \cos \phi_x - \sin \phi_x \sin \phi_y \sin \phi_z & \mp \cos \phi_z \sin \phi_x \mp \cos \phi_x \sin \phi_y \sin \phi_z & \mp \cos \phi_z \sin \phi_x \mp \cos \phi_x \sin \phi_y \sin \phi_z & t_y \\ \sin \phi_y & \cos \phi_y \sin \phi_x & \pm \cos \phi_x \cos \phi_y & \pm \cos \phi_x \cos \phi_y & t_z \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

darstellen. Dabei sind  $\phi_x, \phi_y$  und  $\phi_z$  die Drehwinkel um die Koordinatenachsen (Eulersche Winkel). Für „+“ handelt es sich dabei um eigentliche (orientierungserhaltende), für „-“ um uneigentliche Bewegungen, also Bewegungen, in denen eine Spiegelung enthalten ist.

Streckungen entlang der Koordinatenachsen werden durch Matrizen der Form

$$S = \begin{pmatrix} k_x & 0 & 0 & t_x \\ 0 & k_y & 0 & t_y \\ 0 & 0 & k_z & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

beschrieben, wobei  $k_x$ ,  $k_y$  und  $k_z$  die Streckungsfaktoren

entlang der drei Achsen sind; für  $k_x = k_y = k_z$  ergeben sich zentrische Streckungen.

Als Fixpunkt bei Drehungen und Streckungszentrum bei Skalierungen wird stets der Koordinatenursprung verwendet. Dabei ist es in der Computergrafik üblich, verschiedene Koordinatensysteme (Weltkoordinatensystem, lokales bzw. Objektkoordinatensystem, Kamera- bzw. Ansichtskoordinatensystem) zu verwenden und Transformationen dieser Koordinatensysteme anstelle von Transformationen der Objekte vorzunehmen.<sup>17</sup> Durch Verschiebung der Objektkoordinatensysteme gegenüber den zugehörigen Objekten können deren Drehpunkte und Streckungszentren flexibel angepasst werden. Eine praktische Anwendung von Koordinatentransformationen auf Benutzerebene besteht darin, die Szenenbeschreibung nicht bezüglich des in der Computergrafik häufig verwendeten linkshändigen Koordinatensystems (siehe z. B. Abb. 19) vorzunehmen, sondern das u. a. in der Schule gebräuchliche rechtshändige Koordinatensystem zu nutzen und die gesamte Szene dann mittels

der Transformationsmatrix  $T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$  in das (u. a. von POV-Ray für die

Eingabe erwartete) linkshändige System zu transformieren.

### Zentral- und Parallelprojektionen

Um aus einer dreidimensionalen Szene ein Bild zu erzeugen, muss diese Szene in eine Ebene projiziert werden. Dabei werden (entsprechend der bereits erwähnten Analogie zwischen 3D-Computergrafik und Fotografie) meist Zentralprojektionen, mitunter aber auch Parallelprojektionen durchgeführt.<sup>18</sup> Bei einer Zentralprojektion mit dem Projektionszentrum  $Z$  wird jeder Punkt  $P$  des Raumes auf den Schnittpunkt  $P'$  der Geraden  $ZP$  mit der Projektionsebene abgebildet (siehe Abb. 11).

<sup>17</sup> Sämtliche geometrischen Transformationen der sogenannten „Grafik-Pipeline“, d. h. aller Schritte der Computergrafik von der Objektmodellierung bis hin zur Abbildung in der Monitorebene, können als Transformationen von Koordinatensystemen aufgefasst werden, siehe u. a. [8] und [22].

<sup>18</sup> Im Folgenden werden nur die Zentralprojektionen betrachtet, da diese für die Erzeugung fotorealistischer 3D-Darstellungen von weitaus größerer Bedeutung sind. Parallelprojektionen werden (wegen des geringeren Rechenaufwandes) vor allem für Echtzeit-3D-Darstellungen verwendet. Dabei wird die Szene (bzw. das Koordinatensystem) so transformiert, dass die Projektions- (Bildschirm-)ebene mit der  $x$ - $y$ -Ebene übereinstimmt, die  $z$ -Koordinate gibt dann die „Tiefe“ von Punkten an und entscheidet bei zwei Punkten mit gleichen  $x$ - und  $y$ -Koordinaten, welcher Punkt von dem anderen verdeckt wird.

Vor der eigentlichen Projektion wird i. Allg. zunächst eine Koordinatentransformation der gesamten Szene vorgenommen, in deren Ergebnis die Projektionsebene parallel zur  $x$ - $y$ -Ebene des Koordinatensystems ist und das Projektionszentrum in dessen Ursprung liegt (Sichtkoordinatensystem).

Nach den Strahlensätzen ergibt sich für die Koordinaten  $x'$ ,  $y'$  des Bildpunktes  $P'$  eines Punktes  $P(x; y; z)$ :

$$x' = \frac{d \cdot x}{z} \quad \text{und} \quad y' = \frac{d \cdot y}{z},$$

wobei  $d$  der Abstand des Projektionszentrums von der Projektionsebene ist. Besonders elegant lässt sich die Zentralprojektion in homogenen Koordinaten darstellen, hierbei gilt

$$(x'; y'; z'; w') = \left( x; y; z; \frac{z}{d} \right)$$

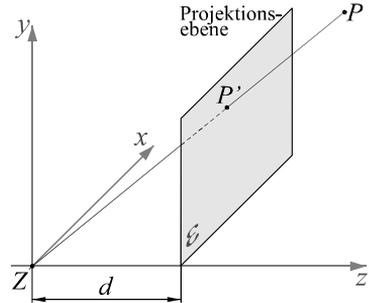


Abb. 11

für die Bildkoordinaten eines (vorher normierten) Punktes  $P(x; y; z; 1)$  bzw.

$$\begin{pmatrix} x' \\ y' \\ z' \\ w' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}.$$

Während die Bildberechnung Software-intern erfolgt, ist die Wahl von Projektionszentrum und -ebene durch den Benutzer vorzunehmen und bestimmt den Inhalt des berechneten Bildes. In 3D-Grafiksoftware werden nicht Projektionszentrum und -ebene angegeben, sondern die Position der Kamera (in POV-Ray: **location**, entspricht dem Projektionszentrum  $Z$ ) und ein Vektor  $\vec{d}$ , der ihre Richtung beschreibt (**direction**) oder ein Zielpunkt  $L$ , auf den die Kamera gerichtet ist (**look\_at**).

Die Projektionsebene wird dann (bis auf Verschiebung) als eine zu dem „Direction“-Vektor  $\vec{d}$  (bzw. dem dazu kollinearen Vektor  $\vec{ZL}$ ) orthogonale Ebene festgelegt. Da bei computergrafischen Darstellungen nicht der gesamte Raum projiziert, sondern lediglich ein Teil des Raumes auf einen rechteckigen Ausschnitt der Projektionsebene abgebildet wird, können zusätzlich Angaben zum Bildausschnitt (**up** und **right**) oder zum Öffnungswinkel der Kamera (**angle**) gemacht werden, durch die dann die Größe des Bildausschnittes festgelegt wird (siehe Abb. 12).

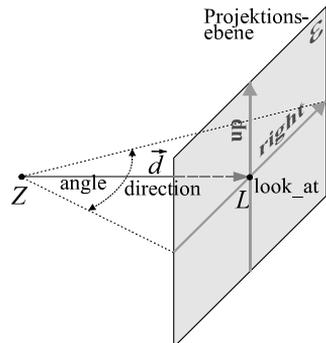


Abb. 12

## 2.5 Das Raytracing-Verfahren zur Erzeugung von 3D-Darstellungen

Von den vielfältigen Verfahren zur Bildberechnung in der 3D-Computergrafik hat das Raytracing- (Strahlverfolgungs-) Verfahren bei der Erzeugung hochqualitativer dreidimensionaler Darstellungen eine besondere Bedeutung erlangt.<sup>19</sup> Es wird von der Mehrzahl der aktuellen 3D-Softwarepakete angewendet; die hier vorrangig betrachtete Software POV-Ray basiert auf diesem Verfahren. Da sich Raytracing eng an den Gesetzen der geometrischen Optik orientiert, ist seine Funktionsweise im Vergleich zu der anderer Verfahren zudem recht einfach nachvollziehbar.

Bei der Bildberechnung durch Raytracing werden vom „Auge des Betrachters“ bzw. der Kamera ausgehende Lichtstrahlen – über möglicherweise mehrere Spiegelungen an Körpern und Durchdringungen transparenter Körper – zu den Lichtquellen betrachtet (Abb. 13). Es erfolgt also eine Umkehrung der Zentralprojektion, indem – ausgehend vom Projektionszentrum – nach Objekten „gesucht“ wird, die auf den betrachteten Ausschnitt der Projektionsebene abgebildet werden. Dabei kommt ein elementares Gesetz der geometrischen Optik, das Gesetz von der Umkehrbarkeit des Lichtweges, zur Anwendung.

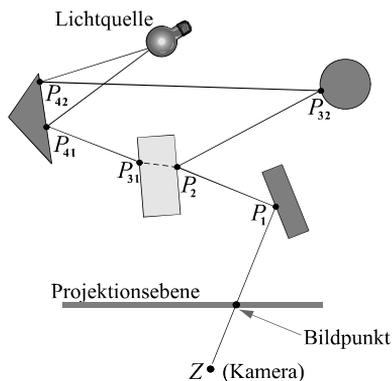


Abb. 13

Die Anzahl der zu verfolgenden Lichtstrahlen hängt von der gewünschten Auflösung des Bildes ab. Soll diese z. B.  $800 \times 600$  Pixel (Bildpunkte) betragen, so müssen 480 000 Lichtstrahlen verfolgt werden.<sup>20</sup> Für jeden der zu verfolgenden Lichtstrahlen sind folgende Schritte durchzuführen:

1. Bestimmung des vom Ausgangspunkt des Lichtstrahls aus nächstgelegenen Schnittpunktes des Strahls mit einer Objektoberfläche. Ein derartiger Schnittpunkt wird *Knoten des Sichtstrahlenbaumes* genannt.
2. Für den Lichtstrahl werden in dem unter 1. ermittelten Schnittpunkt unter Berücksichtigung der geometrischen und optischen Eigenschaften des Objekts

<sup>19</sup> In Bereichen, bei denen es auf eine sehr schnelle Bilderzeugung ankommt (z. B. Computerspiele) werden allerdings andere Verfahren angewendet, da Raytracing hierfür viel zu rechenintensiv ist. Im Bereich der hochqualitativen Computergrafik hingegen wird das Raytracing-Verfahren zunehmend um (teilweise noch aufwändigere) Verfahren der globalen Beleuchtung (Radiosity) ergänzt, die zu einer Steigerung des Fotorealismus computergrafischer Darstellungen führen, siehe u. a. [5] und [22].

<sup>20</sup> Diese Zahl steigt noch, wenn im Interesse der Qualitätsverbesserung (Kantenglättung) Interpolationsmechanismen (Antialiasing) zum Einsatz kommen.

die Richtungen sowie farbabhängigen Intensitäten reflektierter und (bei transparenten Objekten) transmittierter Lichtstrahlen berechnet.

3. Für jeden reflektierten oder transmittierten Lichtstrahl werden erneut die Schritte 1. und 2. durchgeführt, wodurch neue Knoten und Lichtstrahlen entstehen, für die wiederum 1. und 2. auszuführen sind. Da dieser Prozess unendlich lange dauern könnte, wird das Verfahren nach einer festgelegten Zahl von Reflexionen oder einer festgelegten Abschwächung, die der Strahl durch Reflexionen bzw. Durchdringungen von Körpern erfahren hat, abgebrochen.<sup>21</sup> Abb. 13 zeigt die von einem einzigen Bildpunkt ausgehenden Lichtstrahlen und Knoten (bei einer Strahlverfolgungstiefe von 4) für eine einfache Szene, wobei zur Vereinfachung angenommen wird, dass an den Innenflächen des teiltransparenten Körpers keine Reflexionen auftreten.
4. Für jeden Knoten des Sichtstrahlenbaums wird ein Helligkeitswert errechnet, der sich aus dahinter liegenden Reflexionen und der unmittelbaren Beleuchtung durch die Lichtquellen ergibt. Im Sichtstrahlenbaum weiter hinten liegende Knoten haben damit Einfluss auf das Erscheinungsbild weiter vorn liegender Knoten, da diese das Licht der hinten liegenden Knoten zur Kamera hin reflektieren. Die Helligkeits- bzw. Farbwerte der von der Kamera aus ersten Knoten werden den Pixeln des Bildes zugeordnet.<sup>22</sup> Zur Simulation von atmosphärischen Einflüssen (wie Nebel) kann dabei noch eine entfernungsabhängige Abschwächung vorgenommen werden.

Entscheidend für das Aussehen des berechneten Bildes ist die Berechnung der Lichtintensität in jedem Punkt, in dem ein verfolgter Lichtstrahl auf eine Oberfläche eines Körpers trifft. Dieser Helligkeitswert setzt sich aus verschiedenen Beleuchtungsanteilen zusammen, von denen die für eine realistische Bildberechnung wichtigsten der *ambiente*, der *diffuse* und der *Phongsche Beleuchtungsanteil* sowie der Anteil *der direkten Reflexion* sind.

### Ambiente Beleuchtung (Umgebungslicht)

Die *ambiente* (richtungsunabhängige) Beleuchtung wird in der Natur durch vielfältige Lichtstreuungen hervorgerufen und tritt insbesondere bei Nebel stark auf. Die Intensität  $I$  der ambienten Beleuchtung hängt nur von der globalen Intensität des

---

<sup>21</sup> In der Praxis werden Lichtstrahlen meist über 5 bis 10 Reflexionen bzw. Durchdringungen hinweg verfolgt, danach ist die Lichtintensität i. Allg. derart abgeschwächt, dass ein Abbrechen der Strahlverfolgung kaum noch Einfluss auf das Ergebnis hat. In POV-Ray lässt sich die Tiefe der Strahlverfolgung mittels `max_trace_level` einstellen.

<sup>22</sup> In der 3D-Computergrafik kommt fast ausschließlich das RGB-Farbmodell zum Einsatz, bei dem jede Farbe aus den drei Grundfarben Rot, Grün und Blau zusammengesetzt wird. Alle im folgenden beschriebenen Helligkeitsberechnungen müssen also jeweils dreifach, nämlich für jede der drei Grundfarben, durchgeführt werden.

ambienten Lichtes ( $I_a$ ) und einem Faktor  $k_a$  ab, der angibt, wie stark die Oberfläche des betrachteten Körper ambientes Licht wiedergibt:

$$I = I_a \cdot k_a \text{ mit } 0 \leq k_a \leq 1.$$

Die ambiente Beleuchtung ist somit völlig unabhängig von der gegenseitigen Lage von Objekt, Kamera und Lichtquellen. Ein räumlicher Eindruck entsteht bei ausschließlich ambierter Beleuchtung nicht (siehe Abb. 17 a).

### Diffuse Reflexion (Streulicht)

Der diffuse Anteil des direkt von der Lichtquelle auf die Oberfläche fallenden Lichtes ist abhängig von der Lichteinfallrichtung, aber unabhängig von der Richtung zum Betrachter. Dieser Ansatz ist dadurch gerechtfertigt, dass bei rauhen Oberflächen die Lichtstrahlen unterschiedlich weit eindringen und in verschiedene Richtungen reflektiert werden (Abb. 14). Nach dem Lambertschen Kosinusetz ist der Anteil des diffus reflektierten Lichts, das von einer Lichtquelle der Intensität  $I_j$  ausgeht, proportional zum Skalarprodukt des (normierten) Lichteinfallsvektors  $\vec{l}$  mit dem Normaleneinheitsvektor  $\vec{n}$  im betrachteten Flächenpunkt:

$$I = I_j \cdot k_d \cdot \max(0, \langle \vec{n}, \vec{l} \rangle) \text{ mit } 0 \leq k_d \leq 1.$$

Dabei ist  $k_d$  eine Materialkonstante, die angibt, wie stark der Anteil des diffus reflektierten Lichts für eine Oberfläche ist. Die Berücksichtigung der diffusen Beleuchtung erzeugt bereits einen gewissen räumlichen Eindruck (siehe Abb. 17 b).

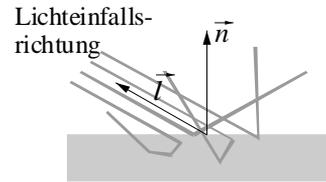


Abb. 14

### Direkte (spiegelnde) Reflexion

Direkte Reflexion tritt auf, wenn die Flächennormale  $\vec{n}$  im betrachteten Punkt mit den Verbindungsvektoren  $\vec{l}$  zur Lichtquelle und  $\vec{b}$  zum Auge gleiche Winkel einschließt und alle drei Vektoren komplanar sind (Reflexionsgesetz):

$$I = \begin{cases} I_{\text{ein}} \cdot k_s, & \text{falls } \vec{n} = \frac{\vec{b} + \vec{l}}{|\vec{b} + \vec{l}|} \\ 0, & \text{sonst} \end{cases}$$

Dabei gibt  $k_s$  an, wie stark ein Körper Lichtstrahlen direkt reflektiert, für ideale Spiegel gilt  $k_s = 1$ .

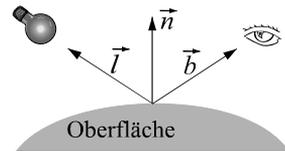


Abb. 15

Im Gegensatz zur diffusen und der (unten beschriebenen) Phongschen Reflexion werden für die Berechnung der direkten Reflexion nicht nur Lichtstrahlen berücksichtigt, die unmittelbar von Lichtquellen ausgehen, sondern auch von anderen Objekten der Szene reflektierte Strahlen; in Abb. 17 c) ist deutlich erkennbar, wie sich die Schachbrettebene in der Kugel spiegelt.

### Das Phong'sche Beleuchtungsmodell

Da reale Oberflächen keine idealen Spiegel sind, entspricht das einfache Modell der spiegelnden Reflexion nicht der Realität. Vielmehr werden Leuchtflecken („Highlights“) auf Körperoberflächen wahrgenommen, die mit der Position des Betrachters „wandern“. Da aber Lichtquellen i. Allg. als punktförmig betrachtet werden, würden derartige Leuchtflecken durch die bisher beschriebenen Komponenten nicht erzeugt werden. Durch das Phong'sche Reflexionsmodell werden nun mehr oder weniger scharf abgegrenzte Leuchtbereiche in der Nähe der Punkte, in denen die Lichtquellen (mit der Lichtintensität  $I_j$ ) reflektiert werden, hinzugefügt:

$$I = I_j \cdot k_p \cdot \max\left(0, \langle \vec{n}, \vec{h} \rangle\right)^m \quad \text{mit } 0 \leq k_p \leq 1 \quad \text{und} \quad \vec{h} = \frac{\vec{b} + \vec{l}}{|\vec{b} + \vec{l}|}.$$

Dabei gibt der *Highlight-Vektor*  $\vec{h}$  die Richtung der Winkelhalbierenden der Verbindungsvektoren des Oberflächenpunktes zur Lichtquelle und zum Beobachter an. Je geringer der Winkel dieses Vektors zum Normaleneinheitsvektor ist, desto stärker wird die Lichtquelle in dem betrachteten Punkt „gespiegelt“.

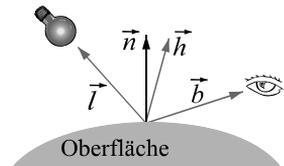


Abb. 16

Die Stärke der Abnahme der Lichtintensität bei wachsendem Winkel zwischen  $\vec{n}$  und  $\vec{h}$  wird durch den Exponenten  $m$  (*Rauhigkeits-Index*) bestimmt. Je geringer  $m$  ist, desto größer sind die Leuchtflecken. Oberflächen aus Metall weisen kleine und stark hervortretende Glanzpunkte auf; sie zeichnen sich somit durch hohe  $m$ -Werte aus. In den Abbildungen 17 d) und e) sind Beispiele für verschiedene Rauhigkeits-Indizes dargestellt, gebräuchliche Werte liegen zwischen 10 und 50.

### Zusammenfassung der Lichtkomponenten

Die vier diskutierten Beleuchtungskomponenten lassen sich (unter Hinzunahme einer Komponente für transmittiertes Licht bei transparenten Körpern) zu folgender Gleichung für die Gesamtlichtintensität  $I$  in einem Knoten des Sichtstrahlenbaums zusammenfassen:

$$I = k_a I_a + k_d \cdot \sum_{j=1}^n I_j \cdot \max\left(0, \langle \vec{n}, \vec{l}_j \rangle\right) \cdot O + k_p \cdot \sum_{j=1}^n I_j \cdot \max\left(0, \langle \vec{n}, \vec{h}_j \rangle\right)^m + k_s I_r + k_t I_t.$$

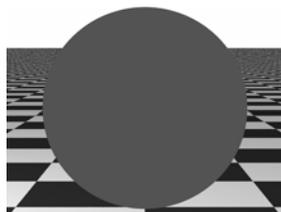
Dabei ist  $n$  die Anzahl der Lichtquellen in der Szene mit den Intensitäten  $I_1, \dots, I_n$ ,  $\vec{n}$  der Normaleneinheitsvektor im betrachteten Flächenpunkt,  $\vec{l}_j$  der normierte Verbindungsvektor zur  $j$ -ten Lichtquelle und  $\vec{h}_j$  der Highlight-Vektor in Bezug auf diese Lichtquelle.  $I_a$ ,  $I_r$  und  $I_t$  sind die Intensitäten des ambienten Lichtes in der Szene und des aus Reflexions- und Transmissionsrichtung auf den betrachteten Punkt einfallenden Lichtes. Durch die Faktoren  $k_a$  (**ambient**),  $k_d$  (**diffuse**),  $k_p$  (**phong**),  $k_s$  (**reflection**),  $k_t$  (Transparenz) und den Rauhigkeits-Index  $m$

(`phong_size`)<sup>23</sup> wird die Beschaffenheit der Oberfläche beschrieben. Der Faktor  $O$  beschreibt die Farbe der Oberfläche. Es handelt sich bei diesem Faktor um die der betrachteten Grundfarbe entsprechende Komponente des Farbvektors  $\langle r, g, b \rangle$ , der aus den Komponenten Rot, Grün und Blau besteht. Wie bereits erwähnt, muss die Lichtintensität für jede dieser drei Grundfarben berechnet werden.

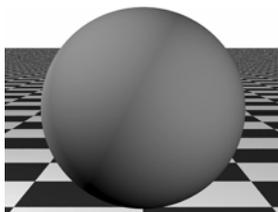
Durch das Verhältnis der beschriebenen Beleuchtungskomponenten und die Festlegung der Farbe lässt sich das Aussehen der Oberfläche eines Körpers in weiten Grenzen bestimmen.<sup>24</sup> In POV-Ray wird dazu dem entsprechenden Körper durch die folgende Anweisung eine Oberfläche (Textur) zugewiesen:

```
texture { pigment { color rgbt <r,g,b,k_t> } }
        finish {ambient k_a diffuse k_d
                phong k_p phong_size m reflection k_s } }
```

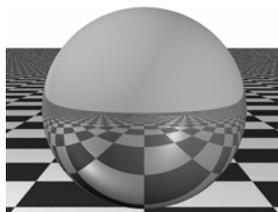
Abb. 17 zeigt eine Kugel, deren Oberfläche mit dieser Definition und den angegebenen Werten für die Parameter  $k_a$ ,  $k_d$ ,  $k_p$ ,  $k_s$ ,  $k_t$  und  $m$  erzeugt wurde.



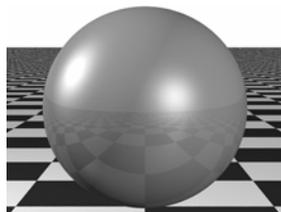
a)  $k_a = 0,9$ ;  $k_d = 0$ ;  $k_s = 0$   
 $k_p = 0$ ;  $k_t = 0$



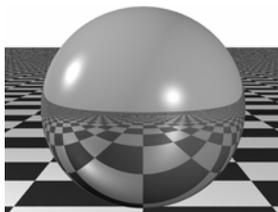
b)  $k_a = 0,1$ ;  $k_d = 1$ ;  $k_s = 0$   
 $k_p = 0$ ;  $k_t = 0$



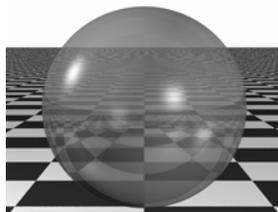
c)  $k_a = 0,1$ ;  $k_d = 0,7$ ;  $k_s = 0,3$   
 $k_p = 0$ ;  $k_t = 0$



d)  $k_a = 0,1$ ;  $k_d = 0,7$ ;  $k_s = 0,1$   
 $k_p = 0,5$ ;  $m = 12$ ;  $k_t = 0$



e)  $k_a = 0,1$ ;  $k_d = 0,7$ ;  $k_s = 0,3$   
 $k_p = 0,5$ ;  $m = 50$ ;  $k_t = 0$



f)  $k_a = 0,1$ ;  $k_d = 0,7$ ;  $k_s = 0,3$   
 $k_p = 0,5$ ;  $m = 50$ ;  $k_t = 0,4$

Abb. 17

<sup>23</sup> Durch die in Klammern angegebenen Schlüsselwörter können die entsprechenden Werte in POV-Ray festgelegt werden.

<sup>24</sup> Natürlich werden auch weiter gehende Methoden zur Oberflächengestaltung verwendet, z. B. die Projektion von Bildern auf Körper (Texture-Maps) und die Simulation von Vertiefungen (Bump-Maps). Die hier beschriebenen Parameter lassen jedoch bereits eine recht weit reichende Gestaltung gleichmäßiger, einfarbiger Oberflächen zu.

### 3 Vorschläge für die Einbeziehung von Elementen der 3D-Computergrafik in den Unterricht der Analytischen Geometrie

#### 3.1 Modellierung einfacher Körper, Arbeit mit dem 3D-Koordinatensystem

Um die Möglichkeiten, welche die Nutzung einer 3D-Grafiksoftware im Stoffgebiet Analytische Geometrie bietet, umfassend nutzen zu können, erscheint es sinnvoll, die Schüler gleich am Anfang dieses Stoffgebietes an die Arbeit mit einer solchen Software<sup>25</sup> heranzuführen. Dafür spricht auch, dass der wichtigste Unterrichtsinhalt in dieser Anfangsphase die Arbeit mit einem räumlichen Koordinatensystem ist.<sup>26</sup> Die Orientierung in diesem ist notwendig, um mit POV-Ray Grafiken zu erstellen und wird gleichzeitig durch den Umgang mit der Software geschult.

Die Beschreibung einfacher Körper und ihre Positionierung im Raum durch Koordinaten bietet einen guten Einstieg in die Arbeit mit dem Programm, und die Schüler machen sich dabei gleichzeitig mit dem Koordinatensystem vertraut.<sup>27</sup> So könnte ihnen das Bild einer einfachen Szene (wie z. B. Abb. 18) mit der Aufforderung vorgelegt werden, die Objekte der Szene, bei denen es sich um einfache Grundkörper handelt, durch Koordinaten zu beschreiben.

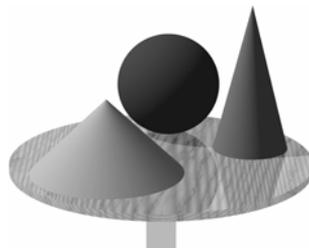


Abb. 18

Um den Schülern den Einstieg in die Arbeit mit der Software zu erleichtern, empfiehlt es sich, ihnen anfangs vorbereitete Dateien zur Verfügung zu stellen, welche die benötigten Makropakete laden und die Kamera sowie die Lichtquellen festlegen. Eine solche Vorgabedatei (Template) sollte die folgenden Zeilen enthalten:

- 
- <sup>25</sup> Wie bereits erwähnt, wird hier die Verwendung von POV-Ray ([27]) vorgeschlagen.
- <sup>26</sup> Obwohl – wie in Abschnitt 2.1 ausgeführt – für die Computergrafik die homogenen Koordinaten sehr bedeutsam sind, erscheint deren Behandlung in der Schule weniger sinnvoll, da bereits die Orientierung im dreidimensionalen kartesischen Koordinatensystem für die Schüler mit Problemen verbunden ist. Für das inhaltliche, geometrische Verständnis der Computergrafik ist die Arbeit mit homogenen Koordinaten auch nicht unbedingt notwendig, da für die geometrische Modellierung kartesische Koordinaten verwendet und diese erst software-intern in homogene Koordinaten transformiert werden.
- <sup>27</sup> Wie bereits erwähnt, liegt POV-Ray ein linkshändiges Koordinatensystem zugrunde (siehe Abb. 19). Soll im Unterricht weiterhin das rechtshändige System genutzt werden, müssen die Schüler jeweils die  $z$ - und  $y$ -Koordinaten vertauschen oder eine Koordinatentransformation vornehmen. Allerdings wäre die dazu einzufügende Matrix (siehe Abschnitt 2.4) eine „Black-Box“ für die Schüler. Dieses Problem lässt sich dadurch umgehen, dass im gesamten Unterricht das linkshändige System verwendet wird. Dagegen spricht, dass viele deutsche Schulbücher rechtshändige Koordinatensysteme verwenden.

```

#version 3.5;          (Gibt an, welche Programmversion verwendet wird.)
#include "colors.inc"  (Lädt eine Datei, in der Farben definiert sind.)
#include "textures.inc" (Lädt eine Datei, in der Oberflächen definiert sind.)
background {White}   (Legt die Farbe des Hintergrundes fest.)
camera { location <0,6,-20> angle 12 look_at <0,-0.5,0> }
light_source { <-30,30,0> color White }

```

In dieser vorbereiteten Datei müssen die Schüler nun noch die geometrischen Objekte der Szene beschreiben:<sup>28</sup>

```

sphere { <0.1,-0.3,1.2> 0.7 texture{pigment{color Blue } } }
cone { <1.2,-1,0.5>, 0.5, <1.2,1,0.5>, 0
      texture { pigment { color Red } } }
cone { <-0.7,-1,-0.5>, 1, <-0.7,0,-0.5>, 0.0
      texture { pigment{ color Green } } }
cylinder { <0,-1.1,0>, <0,-1,0>, 2 texture { DMFWood6 } }
cylinder { <0,-10,0>, <0,-1,0>, 0.2 texture { Aluminum } }

```

Dabei ist zu erarbeiten, wie Kugeln, Zylinder und Kegel im Raum beschrieben werden können; die Syntax von POV-Ray ist dann leicht verständlich. Die Erzeugung z. B. eines Kegelstumpfes durch `cone{< $x_1, y_1, z_1$ >,  $r_1$ , < $x_2, y_2, z_2$ >,  $r_2$ }` erscheint nahe liegend, wenn herausgearbeitet wurde, dass dieser durch die Mittelpunkte und Radien seiner Grund- und Deckfläche beschrieben werden kann.

Häufig fällt die anschauliche Orientierung im dreidimensionalen Koordinatensystem und somit die Erstellung von Grafiken leichter, wenn auf den Bildern ein Achsenkreuz mit dargestellt wird. Ein solches können die Schüler aus Grundkörpern (Zylinder, Kegel, Kugeln und Textobjekte) selbst erstellen. Sollte die zur Verfügung stehende Zeit hierfür nicht ausreichen, können sie mittels der `#include`-Anweisung auch eine bereits vorbereitete Datei einbinden.<sup>29</sup>

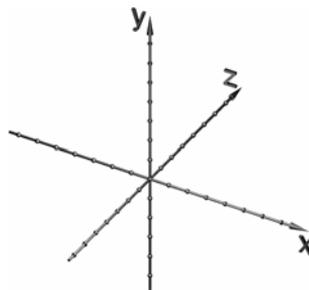


Abb. 19

<sup>28</sup> Dazu sollte den Schülern der Koordinatenbereich mitgeteilt werden, der von der vorbereiteten Kamera erfasst wird. Weiterhin sollten sie die `texture`-Anweisung kennen, um den Körpern Oberflächeneigenschaften zuzuweisen. Eine Vertiefung dieser Thematik bietet sich jedoch eher im Zusammenhang mit der Behandlung der Normalenvektoren an. Erfahrungsgemäß werden allerdings einige Schüler bereits an dieser Stelle mit Texturen experimentieren, da sich hiermit interessante Effekte erzielen lassen.

<sup>29</sup> Ein vorbereitetes Koordinatensystem steht unter [26] zur Verfügung. Da das Arbeits tempo der Schüler bei Computeranwendungen meist sehr unterschiedlich ist, erscheint auch ein arbeitsteiliges Vorgehen sinnvoll, wobei einzelne Schüler derartige Dateien erstellen, die dann von den Mitschülern genutzt werden. Andere Schüler könnten sich währenddessen z. B. mit der Erzeugung von Videos oder mit Texturen beschäftigen.

Eine wichtige Eigenschaft der 3D-Computergrafik besteht darin, dass sich Szenen aus beliebigen Richtungen betrachten lassen. Nachdem die Schüler eine einfache Szene erstellt haben, können sie mehrere „Schnappschüsse“ dieser Szene anfertigen, indem sie die Lage der Kamera ( $\text{camera}\{\text{location}\{<math>\langle x, y, z \rangle</math>\}$ ) verändern.

Durch das Berechnen einer Sequenz von Bildern mit jeweils gering voneinander abweichenden Kamerapositionen können die Schüler kleine Videoclips mit „Kameraflügen“ anfertigen. Dazu muss die Position der Kamera in Abhängigkeit von einem zeitlich veränderbaren Parameter dargestellt werden:

$$x = f_1(t), y = f_2(t), z = f_3(t).$$

Mathematische Grundlagen der Erstellung von Kameraflügen bilden somit die Parameterdarstellungen von Raumkurven, die allerdings (mit Ausnahme von Geraden) nicht Gegenstand des gegenwärtigen Unterrichts sind.

Jedoch erfordert beispielsweise die Beschreibung eines Kreises in spezieller Lage durch eine Parametergleichung lediglich die Erinnerung an die Einführung der trigonometrischen Funktionen am Einheitskreis (siehe Abb. 20). Davon ausgehend kann für den in der Ebene mit der Gleichung  $y = h$  gelegenen Kreis mit dem Mittelpunkt  $(0; h; 0)$  und dem Radius  $r$  die Parameterdarstellung

$$x(t) = r \cdot \cos \alpha, y(t) = h, z(t) = r \cdot \sin \alpha$$

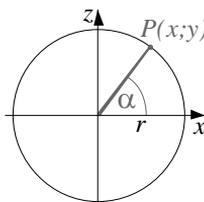


Abb. 20

erarbeitet werden. Für die Erstellung eines Videos, bei dem die Kamera auf dem Kreis entlang fährt, ist dann  $\alpha$  in Abhängigkeit von dem Parameter  $\text{clock}$  auszudrücken und die Kameraposition mit  $(r \cos \alpha; h; r \sin \alpha)$  festzulegen.<sup>30</sup>

Zusammengefasst können die Schüler durch die Darstellung einfacher Körper, die Änderung ihrer Lage im Raum und das Variieren der Kamerakoordinaten einen Einstieg in die Beschreibung geometrischer Objekte durch Koordinaten finden und mit dem Koordinatensystem vertraut werden. Derartige Tätigkeiten eignen sich daher gut für den Beginn des Stoffgebietes Analytische Geometrie. Die vorgeschlagene Beschreibung geometrischer Körper anhand ihrer konkreten Eigenschaften durch Koordinaten lehnt sich eng an elementargeometrische Arbeitsweisen an und stellt somit für die Schüler Bezüge zwischen der Elementargeometrie und der Analytischen Geometrie her. Zudem können die sich dabei zeigenden Grenzen bezüglich der Modellierbarkeit von Formen für die Motivierung der späteren Behandlung von impliziten Gleichungen und Parameterdarstellungen genutzt werden.

<sup>30</sup> In POV-Ray ist  $\text{clock}$  als zeitabhängiger Parameter vorgesehen; für diesen Wert wird ein Intervall  $[0; n]$  natürlicher Zahlen festgelegt, wobei dann  $n + 1$  Einzelbilder („Frames“) für ein Video berechnet werden. Technische Details zur Erstellung von Videos werden auf der Internetseite [26] beschrieben; auch das fertige Video ist dort verfügbar.

### 3.2 Zur Einführung und Verwendung von Vektoren

Aus der Sicht der Verwendung des Begriffes „Vektor“ in der 3D-Computergrafik (wie überhaupt in der Informatik und ihren Anwendungen) liegt eine arithmetische Auffassung dieses Begriffes als  $n$ -Tupel (meist Tripel) reeller Zahlen nahe. Sowohl in POV-Ray als auch in Computeralgebrasystemen und Programmiersprachen werden Vektoren als (Daten-)Strukturen, die aus mehreren Zahlen bestehen, aufgefasst.<sup>31</sup> Eine geometrische Bedeutung erhalten Vektoren dabei erst aus dem Kontext heraus, in dem sie angewendet werden.

Sicherlich ist die Auffassung von Vektoren als  $n$ -Tupel in der Informatik kein hinreichender Grund, das Vektorraummodell  $\mathbf{R}^n$  (bzw. speziell  $\mathbf{R}^3$ ) auch dem Mathematikunterricht zugrunde zu legen. Jedoch sind auch in der mathematikdidaktischen Diskussion m. E. wichtige Argumente für die Verwendung dieses Modells vorgebracht worden. Neben seiner Einfachheit werden in [4] die „freie Beweglichkeit zwischen verschiedenen Deutungen“ des Vektorbegriffs (S. 177) sowie die Möglichkeit der Übertragung von Konzepten zwischen dem  $\mathbf{R}^n$  und der Geometrie (S. 179) hervorgehoben. TIETZE nennt u. a. den Vorteil, dass sich die Struktur der  $n$ -Tupel „über die Fälle  $n = 2, 3$  hinaus gut verallgemeinern lässt und zugleich ein zentrales Mathematisierungsmuster darstellt“ ([19], S. 79). Den „zwanglosen Wechsel zwischen dem algebraischen Modell und geometrischen Modellen sowie zwischen den geometrischen Modellen untereinander“ sieht er als ein „wichtiges heuristisches Hilfsmittel“ und die  $n$ -Tupel als Bindeglied zwischen geometrischen Inhalten und Themen einer algorithmisch orientierten linearen Algebra – die Funktion als Bindeglied zur Informatik kann m. E. aus heutiger Sicht hinzugefügt werden. Andererseits gelangte G. WITTMANN nach der Analyse von Schülerkonzepten zu der Auffassung, dass sich die „Einführung von Vektoren als  $n$ -Tupel ... nicht als unmittelbare Vorbereitung für eine Analytische Geometrie des Anschauungsraumes“ eignet ([20], S. 140), wobei er sich allerdings auf die Situation bezieht, dass ein arithmetisch eingeführter Vektorbegriff danach ausschließlich für geometrische Anwendungen genutzt wird. Da auch die Behandlung anderer Vektormodelle (wie z. B. der Pfeilklassen) mit erheblichen Schwierigkeiten verbunden ist und die axiomatische Einführung des Vektorraumbegriffs im Unterricht als gescheitert angesehen werden kann, möchte ich hier für eine flexible Verwendung des Vektorbegriffs plädieren, die sich an dem jeweiligen Kontext orientiert, in dem Vektoren betrachtet werden.<sup>32</sup>

---

<sup>31</sup> In der Hilfe zu POV-Ray heißt es dazu: „A vector is a set of related float values.“

<sup>32</sup> Die Thematisierung des Vektorbegriffs unmittelbar am Anfang des Stoffgebietes Analytische Geometrie erscheint allerdings m. E. ohnehin nicht empfehlenswert, da die Schüler anhand der Beschreibung von Punkten und einfachen geometrischen Objekten durch Koordinaten einen anschaulicheren und einfacheren Einstieg in die Analytische Geo-

Das erste Vektormodell, das die Schüler kennen lernen, wenn sie mit einem 3D-Grafikprogramm arbeiten, ist dabei (auch wenn dies im Unterricht zunächst nicht explizit angesprochen werden muss) das der  $n$ -Tupel. Die wichtigsten Zusammenhänge, in denen diese in der Computergrafik auftreten, sind:

- *Die Beschreibung von Punkten des Raumes durch Koordinatentripel.*<sup>33</sup>
- *Die Beschreibung geometrischer Transformationen.* Neben Translationen (**translate**  $\langle t_x, t_y, t_z \rangle$ ) können auch Drehungen (**rotate**  $\langle \phi_x, \phi_y, \phi_z \rangle$ ; die Komponenten des „Drehvektors“ sind die Eulerschen Winkel der Drehung) sowie Streckungen (**scale**  $\langle s_x, s_y, s_z \rangle$ ; die Komponenten dieses Vektors geben die Streckungsfaktoren in Richtung der drei Koordinatenachsen an) durch Vektoren beschrieben werden.
- *Die Beschreibung von Farben.* Wie bereits erwähnt, wird im RGB-Farbmodell jede Farbe durch Intensitätswerte der drei Grundfarben rot, grün und blau beschrieben (**color rgb**  $\langle r, g, b \rangle$ ).<sup>34</sup> Durch Erweiterung auf Quadrupel wird dieses Modell des Farbvektors um die Beschreibung der Transparenz eines Objektes erweitert: **color rgbt**  $\langle r, g, b, t \rangle$ .

Die zunächst völlig „ungeometrische“ Beschreibung von Farben durch Vektoren stellt unter den genannten Beispielen eine Besonderheit dar. Interessant wird die Beschreibung von Farben durch Tripel und die anschließende Interpretation dieser „Farbvektoren“ als Ortsvektoren dadurch, dass ein zunächst nicht geometrischer Inhalt geometrisch veranschaulicht werden kann. Dies geschieht mit dem bekannten „Farbwürfel“ bzw. „RGB-Würfel“, bei dem alle Farben als Punkte eines Würfels dargestellt werden (Abb. 21).<sup>35</sup>

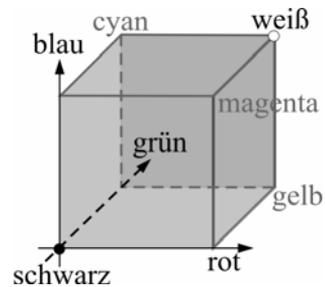


Abb. 21

metrie finden dürften, als wenn gleich am Anfang der vergleichsweise abstrakte Vektorbegriff gewissermaßen „auf Vorrat“ eingeführt würde.

- <sup>33</sup> Die dadurch implizierte Identifizierung von Punkten und Vektoren stellt allerdings ein didaktisches Problem dar (siehe z. B. [20], S. 138), das spätestens bei der Behandlung der Parameterdarstellungen von Geraden und Ebenen zum Tragen kommt. Hier wird es notwendig, dass die Schüler zwischen den verschiedenen Deutungen des Begriffs Vektor unterscheiden; Hinweise hierzu werden u. a. in [4] gegeben.
- <sup>34</sup> Diese Farbvektoren dürften allerdings streng genommen nicht als Vektoren bezeichnet werden, denn da die  $r$ -,  $g$ - und  $b$ -Werte jeweils zwischen 0 und 1 liegen, bilden sie keinen Vektorraum. Auf diese Tatsache sollten die Schüler zumindest hingewiesen werden.
- <sup>35</sup> Auch die Vektoraddition und die Multiplikationen mit Skalaren sind – innerhalb des Definitionsbereiches  $[0;1]$  für die drei Komponenten – sinnvolle Operationen mit Farbvektoren. Das RGB-Modell wird auch als „additives Farbmodell“ bezeichnet.

Auch wenn die Schüler Vektoren zunächst als  $n$ -Tupel kennen lernen, wird ihre Eigenschaft, eine Richtung anzugeben, spätestens bei der Behandlung der Parameterdarstellungen von Geraden und Ebenen in den Vordergrund treten und ihre Deutung durch Pfeile erfordern.<sup>36</sup> Hierbei kann die Verwendung einer Grafiksoftware bei der Veranschaulichung von Vektoren und einfachen Vektoroperationen hilfreich sein. Da in POV-Ray die Visualisierung von Vektoren als Pfeile nicht vorgesehen ist, müssen hierfür geeignete Makros erstellt bzw. vorhandene Makros eingebunden werden.<sup>37</sup>

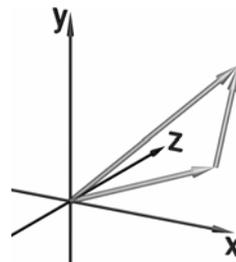


Bild 22

### 3.3 Visualisierung von Ebenen und ihren Lagebeziehungen, Übergang von linearen zu nichtlinearen Objekten

Wie bereits im zweiten Kapitel dieses Artikels beschrieben wurde, lassen sich mithilfe von POV-Ray Flächen, die durch implizite Gleichungen oder Parametergleichungen gegeben sind, darstellen (**isosurface**- bzw. **parametric**-Objekte, siehe Abschnitt 2.2). Hiermit können natürlich auch Ebenen (i. Allg. durch rechteckige Ausschnitte) veranschaulicht werden. Die Möglichkeit, diese transparent darzustellen, erleichtert es, die Lage einer Ebene im Raum zu verdeutlichen, Schnittgeraden zweier Ebenen zu visualisieren und die gegenseitige Lage mehrerer Ebenen des Raumes zu untersuchen.<sup>38</sup>

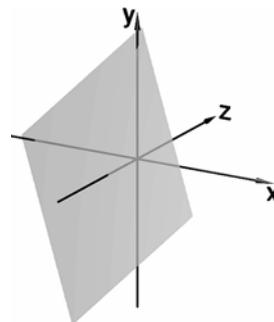


Bild 23

Die im Unterricht sehr verbreiteten (rechnerisch zu lösenden) Aufgaben zu Lagebeziehungen können somit durch Visualisierungen ergänzt werden. Allerdings rechtfertigt die Darstellung linearer Objekte allein nicht den Aufwand einer Be-

<sup>36</sup> Dabei ist die Behandlung des recht abstrakten und für die Schüler oft schwer verständlichen Begriffs „Pfeilklass“ nicht unbedingt erforderlich, vielmehr genügt (z. B. nach [19], S. 79) „die Vorstellung von einem Pfeil, den man frei verschieben kann.“

<sup>37</sup> „Vektoren“ (als Pfeile) können durch Zylinder und Kegel dargestellt werden, die in Abhängigkeit von den Anfangs- und Endpunktkoordinaten des darzustellenden Pfeiles zu positionieren sind. Unter [26] steht hierfür das Makropaket „ana-geo.inc“ zur Verfügung, das die einfache Erzeugung von Orts- und Verbindungsvektoren, Strecken, Ebenen und Koordinatensystemen ermöglicht. Um diese Objekte zu nutzen, ist das Makropaket in die entsprechende POV-Ray-Datei einzubinden: `#include "ana-geo.inc"`.

<sup>38</sup> Eine recht komplexe Anwendung der Visualisierung von Ebenen, die durch implizite Gleichungen gegeben sind, ist die auf der Internetseite [26] gezeigte Veranschaulichung des Gauss-Algorithmus.

schäftigung mit einem komplexen Programm wie POV-Ray.<sup>39</sup> Soll der Computer nur für die Visualisierung von Vektoren, Geraden und Ebenen sowie ihrer Lagebeziehungen genutzt werden, empfiehlt sich der Einsatz einer Software mit einem geringeren Funktionsumfang, die speziell für diese Belange entwickelt wurde.<sup>40</sup>

Wird aber die 3D-Computergrafik, wie hier vorgeschlagen, an mehreren Stellen im Unterricht thematisiert und eine Grafiksoftware wie POV-Ray ohnehin genutzt, so liegt auch ihre Verwendung für die Darstellung von Geraden und Ebenen nahe. Im Sinne der eingangs angestellten Überlegungen zur Formenarmut des Unterrichts erscheint es jedoch wünschenswert, nicht bei der Behandlung linearer Objekte stehen zu bleiben. Indem sie in impliziten Gleichungen und Parameterdarstellungen von Ebenen lineare durch nichtlineare Terme ersetzen, können die Schüler erste Experimente mit der Darstellung von Flächen durchführen und auch mit recht einfachen Gleichungen attraktive Ergebnisse erzielen. Natürlich sollten sich derartige Experimente nicht auf das reine „Spielen“ mit Gleichungen und deren visuelle Betrachtung beschränken, sondern durch Überlegungen ergänzt werden, wie sich Veränderungen an den Gleichungen auf die Gestalt der Flächen auswirken. Eine wichtige Strategie bei derartigen Überlegungen besteht darin, die Schnittkurven der betrachteten Flächen mit den Koordinatenebenen zu untersuchen.

Schnittfiguren haben eine wichtige Bedeutung für die Modellierung von Formen in der 3D-Computergrafik; gleichzeitig wird ihre Untersuchung durch das Veranschaulichungspotenzial von 3D-Grafiksoftware erleichtert und z. T. erst ermöglicht. Es erscheint daher sinnvoll, im Unterricht nicht ausschließlich die (geometrisch recht langweiligen) Schnittgeraden von Ebenen zu behandeln. Die Möglichkeit der Visualisierung erleichtert es, auch Schnitte von Ebenen mit beispielsweise Kugeln und Zylindern bzw. dieser Objekte miteinander zu betrachten.<sup>41</sup> Besonders interessant sind dabei die Schnittfiguren von Ebenen und Kegeln, anhand derer einige der angedeuteten Vorgehensweisen exemplarisch erläutert werden sollen.

---

<sup>39</sup> Zudem benötigt POV-Ray für die Darstellung von Ebenen, die durch parametrische oder implizite Gleichungen gegeben sind, recht lange Rechenzeiten, da durch die Anweisungen `parametric` und `isosurface` beliebige Flächen qualitativ hochwertig dargestellt werden können. Die Software erfasst bei der Eingabe von Ebenengleichungen nicht, dass es sich um recht triviale Spezialfälle handelt, und verwendet wesentlich komplexere Algorithmen als für die Darstellung von Grundkörpern.

<sup>40</sup> Ein hierfür geeignetes Programm ist z. B. DreiDGeo (siehe [1]). Falls die Schüler bereits mit der Nutzung eines CAS vertraut sind, kann natürlich auch dieses für Visualisierungszwecke in der Analytischen Geometrie genutzt werden.

<sup>41</sup> Betrachtet man statt der ebenen Schnitte bzw. der Schnitte von Körperoberflächen die Durchschnitte der „Körperinneren“, so gelangt man zu den Booleschen Operationen (siehe Abschnitt 2.2), einem der wichtigsten Modellierungsprinzipien der 3D-Computergrafik, das auch als CSG (Constructive Solid Geometry) bekannt ist.

### 3.4 Nutzung computergrafischer Darstellungen für die Untersuchung von Kegelschnitten und Flächen 2. Ordnung

Die Nutzung von Computervisualisierungen erleichtert wesentlich die Behandlung der Kegelschnitte im eigentlichen Sinne ihres Namens: als Schnittkurven einer Ebene und eines Kreiskegels.

Dazu können die Schüler Kegel und Ebenen in verschiedenen gegenseitigen Lagen darstellen (siehe z. B. Abb. 24)<sup>42</sup> und zu Vermutungen über mögliche Schnittfiguren gelangen.<sup>43</sup> Dazu müssen sie Kegel zunächst nicht durch Gleichungen beschreiben, da diese auch als **cone**-Objekt zur Verfügung stehen (vgl. Abschnitt 2.2).

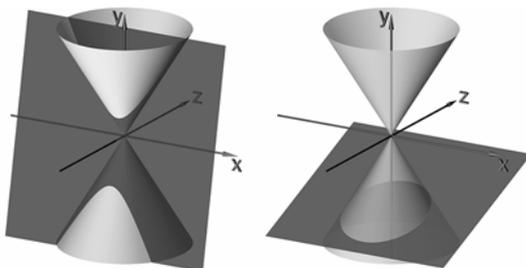


Abb. 24

Nach dieser „Experimentierphase“ und nachdem die Schüler Vermutungen bezüglich der entstehenden Schnittkurven herausgearbeitet haben, können für spezielle Kegel und Ebenen die Gleichungen der Schnittkurven hergeleitet werden, wozu nun die Kegelgleichung benötigt wird. Die Gleichung  $x^2 - y^2 + z^2 = 0$  für einen „Einheitskegel“ in Mittelpunktslage lässt sich durch die Überlegung gewinnen, dass Schnittkurven dieses Kegels mit zur  $x$ - $z$ -Ebene parallelen Ebenen Kreise sind, und sich beim Schnitt mit der  $x$ - $y$ -Ebene sowie der  $y$ - $z$ -Ebene jeweils Geradenpaare ergeben. Für die Schnittfiguren dieses Kegels mit z. B. den Ebenen  $z = 1$ ,  $y = 1$  und  $y = z - 1$  können die Schüler nun die Gleichungen

$$x^2 - y^2 = 1, \quad x^2 + z^2 = 1 \quad \text{und} \quad x^2 + 2z = 1$$

ermitteln und somit Gleichungen für spezielle Hyperbeln, Ellipsen und Parabeln herleiten (wobei herausgearbeitet werden sollte, dass auch „echte“ Ellipsen, also nicht nur Kreise, entstehen können).<sup>44</sup>

Nach diesen Betrachtungen, bei denen die Schüler Kegelschnitte als Schnittfiguren räumlicher Objekte visuell betrachtet und durch Gleichungen beschrieben haben, können nun deren Eigenschaften als ebene Objekte untersucht werden. Eine andere

<sup>42</sup> Unter [26] steht darüber hinaus ein Video zur Verfügung, das sichtbar macht, wie die einzelnen Schnittfiguren bei Veränderung der Ebene ineinander übergehen.

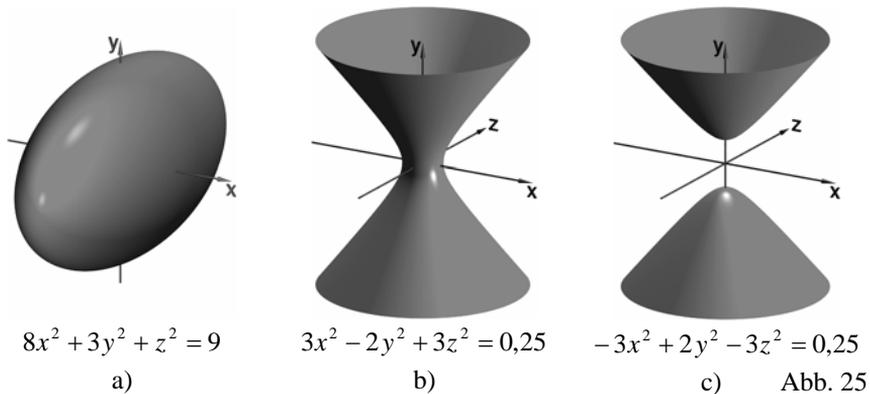
<sup>43</sup> Hierbei können die Schüler insbesondere erkennen, dass die entstehenden Ellipsen bezüglich beider Achsen symmetrisch sind, es sich also nicht um „eiförmige“ Kurven handelt, was aufgrund der Verjüngung des Kegels oft vermutet wird.

<sup>44</sup> Leistungsstarke Schüler können auch, gegebenenfalls mit einigen Hilfen, die allgemeine Scheitelfgleichung der Kegelschnitte herleiten (siehe z. B. [16], S. 242f.).

lohnende Anknüpfungsmöglichkeit besteht zu den Flächen zweiter Ordnung. Die für die Untersuchung der Kegelschnitte beschriebene Vorgehensweise von der Visualisierung zur mathematischen Beschreibung wird dabei umgekehrt, indem Gleichungen aufgestellt und die durch sie beschriebenen Objekte mithilfe der Software dargestellt und exploriert werden. Ausgangspunkte für das Aufstellen von Gleichungen können sowohl die Gleichungen der Kegelschnitte, die durch das Einfügen von Termen für die dritte Koordinate zu Flächen des Raumes erweitert werden, als auch Kugeln und Kegel sein. Indem die Schüler z. B. in den Gleichungen

$$x^2 - y^2 + z^2 = 0 \text{ und } x^2 + y^2 + z^2 = 1$$

für einen Einheitskegel in Mittelpunktslage sowie die Einheitskugel Vorzeichen, Koeffizienten und absolute Glieder variieren sowie ggf. lineare Glieder einfügen, stellen sie Gleichungen für unterschiedliche Flächen zweiter Ordnung auf, die sie anschließend visuell darstellen (siehe z. B. Abb. 25 und 26). Dabei können die Schüler das Ellipsoid (siehe z. B. Abb. 25 a), das ein- und zweischalige Hyperboloid (Abb. 25 b, c) sowie das elliptische und das hyperbolische Paraboloid (Abb. 26 a, b) entdecken und anschließend weiter untersuchen.<sup>45</sup> Dabei sollten sie vom einfachen Verändern der Gleichungen zu einem systematischeren Vorgehen gelangen und die Frage nach dem Einfluss von Gleichungsänderungen auf die Gestalt der Flächen sowie die umgekehrte Frage, welche Änderungen in den Gleichungen notwendig sind, wenn bestimmte Formen modelliert werden sollen, diskutieren.



Eine wichtige Bedeutung bei der Untersuchung sowie gleichungsmäßigen und verbalen Beschreibung der Flächen zweiter Ordnung haben wiederum Überlegungen zu den Schnittkurven der entsprechenden Flächen mit den Koordinatenebenen; damit erfolgt eine Rückbesinnung auf die Kegelschnitte.

<sup>45</sup> Die vollständige Betrachtung oder gar Klassifizierung der Flächen zweiter Ordnung ist dabei m. E. nicht notwendig, da die angestrebten Ziele durchaus anhand exemplarischer Untersuchungen einzelner Flächen erreicht werden können.

Das wichtigste Ziel der vorgeschlagenen Untersuchungen von Flächen zweiter Ordnung wie auch von anderen Flächen des Raumes besteht darin, dass die Schüler exemplarisch den Zusammenhang zwischen der mathematischen Beschreibung und dem geometrischen Objekt anhand interessanter Körper und Flächen durch eigene Experimente entdecken und dadurch Fähigkeiten der Modellierung entwickeln. Die computerunterstützte Visualisierung kann dabei eine wertvolle Hilfe sein, soll aber den Schülern vor allem dabei helfen, eigene anschauliche Vorstellungen herauszubilden und die Fähigkeit ausprägen, Visualisierungen durch Gleichungen beschriebener Objekte auch „im Kopf“ herzustellen sowie umgekehrt zu überlegen, wie Gleichungsterme variiert werden müssen, um gewünschte Formen darzustellen. Die Betrachtung von Schnittfiguren sowie die damit verbundene Transformation ebener in räumliche Sachverhalte und umgekehrt erweist sich dabei als zentrale Strategie.

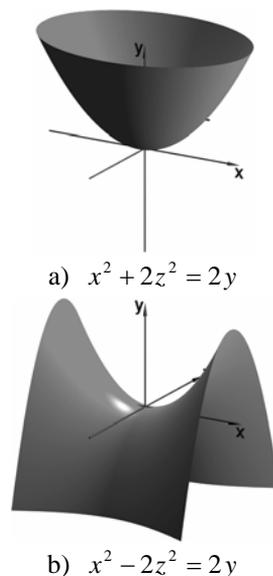


Abb. 26

### 3.5 Normalenvektoren und das Erscheinungsbild von Körperoberflächen

Bei der Darstellung der mathematischen Grundlagen der 3D-Computergrafik wurde bereits deutlich, dass die Normalen von Oberflächen bzw. Facetten maßgeblich deren Erscheinungsbild bestimmen (siehe Abschnitte 2.3 und 2.5). Diese Tatsache kann im Unterricht genutzt werden, um die Beschäftigung mit Normalen(einheits-)vektoren zu motivieren. Ausgangspunkte hierfür bilden Überlegungen zur Funktionsweise der Bildberechnung in 3D-Grafikprogrammen<sup>46</sup> sowie – damit im Zusammenhang stehend – die auch bei deren praktischer Anwendung wichtige Frage nach der Bedeutung der Parameter **ambient**, **diffuse**, **phong**, **phong\_size** und **reflection** sowie ihrer geeigneten Justierung in Abhängigkeit vom gewünschten Erscheinungsbild einer Oberfläche. Da leicht zu erkennen ist, dass für die optische Erscheinung von Körperoberflächen verschiedenartige Reflexionen von Lichtstrahlen maßgeblich sind, kann an das den Schülern bereits aus dem Anfangsunterricht in Physik bekannte Reflexionsgesetz angeknüpft werden.

<sup>46</sup> Das in Abschnitt 2.5 vorgestellte Raytracing-Verfahren eignet sich als Grundlage für diese Betrachtungen besonders gut, da es weit verbreitet und recht leicht verständlich ist. Zudem beruhen die wesentlichen Schritte der Bildberechnung bei diesem Verfahren auf Inhalten der Analytischen Geometrie, die nach den meisten heutigen Rahmenplänen Inhalte des Mathematikunterrichts der Sekundarstufe II sind.

Dieses besagt, dass Einfallswinkel und Reflexionswinkel maßgleich sind, wobei der Einfallswinkel ( $\alpha$ ) vom einfallenden Strahl und dem Einfallslot, der Reflexionswinkel ( $\beta$ ) vom reflektierten Strahl und dem Einfallslot gebildet wird.<sup>47</sup> Als Einfallslot wird die auf der spiegelnden Oberfläche im Auftreffpunkt des einfallenden Strahles errichtete Senkrechte verstanden.

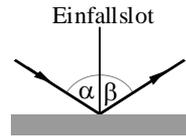


Abb. 27

Um das Reflexionsgesetz, das im Physikunterricht der SI in der Ebene betrachtet wird, auf den dreidimensionalen Fall zu übertragen, ist es notwendig, eine Entsprechung für das Einfallslot zu finden, also Geraden bzw. Vektoren anzugeben, die zu gegebenen Flächen in den Lichteinfallspunkten orthogonal sind.<sup>48</sup> Ausgehend von dieser Notwendigkeit können die Begriffe des Normalenvektors und des Normaleinheitsvektors eingeführt und Möglichkeiten ihrer Berechnung für Ebenen erarbeitet werden. Damit ist es dann möglich, Normaleneinheitsvektoren dreieckiger Facetten von Körperoberflächen zu bestimmen und das Reflexionsgesetz für den Raum (in vektorieller Formulierung<sup>49</sup>) zu erarbeiten.

Werden der Normaleneinheitsvektor mit  $\vec{n}$  und die normierten Richtungsvektoren des einfallenden bzw. reflektierten Lichtstrahls mit  $\vec{l}$  bzw.  $\vec{b}$  bezeichnet, so lässt sich das Reflexionsgesetz durch die folgenden Bedingungen formulieren:<sup>50</sup>

1.  $\vec{n}$ ,  $\vec{l}$  und  $\vec{b}$  sind komplanar,
2. es gilt  $\langle \vec{n}, \vec{l} \rangle = \langle \vec{n}, \vec{b} \rangle$ .

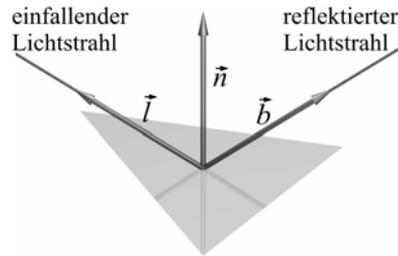


Abb. 28

An die Erarbeitung des Reflexionsgesetzes können sich Aufgaben zur Bestimmung der Richtung reflektierter Lichtstrahlen anschließen. Außerdem sollten die Schüler Objekte mit stark reflektierenden Oberflächen anhand einfacher Szenen (wie z. B.

<sup>47</sup> Teilweise wird in den Lehrbüchern hinzugefügt, dass die beiden Lichtstrahlen und das Einfallslot in einer Ebene liegen; ansonsten geht dies aus der zugehörigen Skizze hervor.

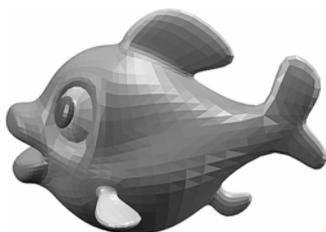
<sup>48</sup> Für Ebenen sind die Normalenvektoren natürlich unabhängig vom betrachteten Punkt. Für gekrümmte Flächen müsste der Exaktheit halber von Senkrechten zu den Tangentialebenen der Flächen in den betrachteten Punkten gesprochen werden. Bei speziellen Flächen (wie z. B. Kugeloberflächen) dürften die Schüler jedoch auch ohne ausführliche Beschäftigung mit den Tangentialebenen die Normalen erkennen.

<sup>49</sup> Dass die vektorielle Beschreibung der Normalenrichtung und der Richtungen der Lichtstrahlen hinsichtlich der Bestimmung der Winkel zwischen diesen Richtungen sinnvoller ist als die Beschreibung durch Geraden, kann leicht plausibel gemacht werden.

<sup>50</sup> In Abschnitt 2.5 wurde bereits eine andere Formulierung des räumlichen Reflexionsgesetzes angegeben, bei der entschieden wird, ob direkte Reflexion von einem vorgegebenen Punkt zu einem Beobachter mit ebenfalls vorgegebener Position erfolgt.

in Abb. 16) mithilfe der Software darstellen. Dabei stellt sich heraus, dass die alleinige Berücksichtigung der direkten Reflexion nur für sehr stark reflektierende Oberflächen realistische Ergebnisse liefert. Indem die Schüler reale Beleuchtungsverhältnisse von Körpern beobachten und diskutieren sowie Versuche mit den Parametern **ambient**, **diffuse** und **phong** anstellen, können sie weitere Komponenten des in Abschnitt 2.5 vorgestellten Beleuchtungsmodells erarbeiten.

Da die meisten geometrischen Objekte in der Computergrafik durch Facetten in Form von Dreiecken angenähert werden (siehe Abschnitt 2.3), die ihrerseits Ebenen vorgeben, ist die Berechnung der Normalenvektoren von Ebenen, die im gegenwärtigen Schulunterricht einen recht breiten Raum einnimmt, auch von zentraler Bedeutung für die Bildberechnung in der 3D-Computergrafik.<sup>51</sup> Dies können die Schüler anhand konkreter Beispiele selbst nachvollziehen, wobei neben Flächen, die mit einem CAS erzeugt und als 3D-Objekte exportiert wurden, auch Modelle von realen Objekten interessant sind. Geeignete Modelle stehen im Internet (zumeist im 3ds-Format des Programms 3D-Studio) frei zur Verfügung und lassen sich mit (ebenfalls frei erhältlichen) Konvertierungsprogrammen in POV-Ray-Dateien umwandeln.<sup>52</sup> Werden derartige Modelle in einfache Dreiecksnetze konvertiert, so fällt die kantige Darstellung der Objekte auf (siehe Abb. 29 a), die daraus resultiert, dass jedes dieser Dreiecke eine Normale besitzt, anhand derer die Helligkeitswerte für alle Punkte des Dreiecks berechnet werden. Somit ändern sich an den Übergängen zwischen den Dreiecken die Normalen abrupt. Durch Verwendung von „smooth-triangles“, bei denen jedem Eckpunkt ein Normalenvektor zugeordnet wird (vgl. Abschnitt 2.3), interpoliert die Software für jeden berechneten Punkt des Bildes einen Normalenvektor und erzeugt dadurch eine glatte Darstellung (siehe Abb. 29 b).



a)



b) Abb. 29

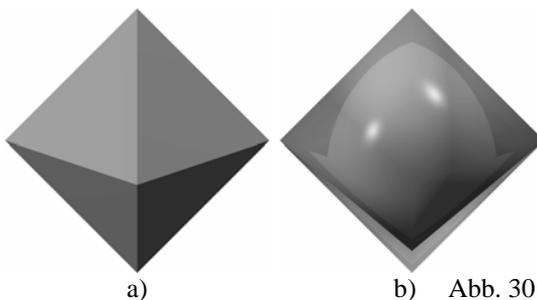
Anhand komplexer Objekte (wie des abgebildeten Fisches mit ca. 6000 Dreiecken) können die Schüler die Struktur der Objekte als Vereinigungsmengen von Drei-

<sup>51</sup> Dennoch sollten die Schüler anhand anderer Beispiele erfahren, dass Normalenvektoren nicht nur für Ebenen von Bedeutung sind. So können sie z. B. recht leicht Normalenvektoren von Kugeln und Zylindern beschreiben.

<sup>52</sup> Ein geeignetes Konvertierungsprogramm (3ds2pov) steht unter [26] zur Verfügung. Für die Beispiele in Abb. 29 a) und b) wurde eine 3ds-Datei unter Verwendung dieses Programms ohne bzw. mit Kantenglättung in POV-Ray-Dateien umgewandelt.

ecken nur durch die Analyse automatisch generierter Dateien erkennen. Indem sie aber sehr einfache Objekte als Mengen von Dreiecken selbst modellieren, besteht für sie die Möglichkeit, die Kantenglättung durch die Berechnung von Mittelwerten von Normaleneinheitsvektoren „manuell“ vorzunehmen. Ein mögliches Beispiel hierfür ist das Oktaeder mit den Eckpunkten  $(1;0;0)$ ,  $(0;1;0)$ ,  $(-1;0;0)$ ,  $(0;-1;0)$ ,  $(0;0;-1)$  und  $(0;0;1)$ , das die Schüler zunächst als Menge von 8 Dreiecken darstellen können (siehe Abb. 30 a). Indem sie die Normaleneinheitsvektoren der durch die 8 Dreiecke festgelegten Ebenen bestimmen und für jeden Eckpunkt den Mittelwert dieser Vektoren für die vier anliegenden Dreiecke berechnen (oder durch inhaltliche Überlegungen feststellen, dass diese Mittelwerte gerade mit den Koordinaten der entsprechenden Eckpunkte übereinstimmen), können sie nun ein Oktaeder aus „smooth-triangles“ erzeugen.

Auch in diesem extremen Fall erfolgt eine Kantenglättung (siehe Abb. 30 b). Die Software täuscht für einen großen Teil des Oktaeders eine glatte Oberfläche vor. Dass sich die geometrische Form des Oktaeders dabei nicht verändert hat, wird bei Betrachtung eines Videos deutlich, in dem sich dieses dreht.<sup>53</sup>



b) Abb. 30

#### 4 Abschließende Bemerkungen

Die hier skizzierten Beispiele decken natürlich bei weitem nicht alle Möglichkeiten ab, welche durch die Einbeziehung von Elementen der 3D-Computergrafik in den Unterricht des Stoffgebietes Analytische Geometrie eröffnet werden. Sie sollen diese lediglich exemplarisch verdeutlichen. Dabei wollte ich sowohl recht elementare Beispiele, die konkrete, „handwerkliche“ Zugänge zur Analytischen Geometrie repräsentieren, als auch vom mathematischen Inhalt her anspruchsvollere Themen darstellen. Nach meiner Auffassung kann die Einbeziehung von Elementen der Computergrafik sowohl in Grund- als auch in Leistungskursen dazu beitragen, die eingangs genannten Defizite des Unterrichts in diesem Stoffgebiet zu mildern, es attraktiver zu gestalten und zu „geometrisieren“. Auf das Unterbreiten von Unterrichtsvorschlägen, die in Bezug auf das mathematische Anspruchsniveau weit über die zur Zeit im Unterricht hauptsächlich behandelten Inhalte hinausgehen, habe ich verzichtet, auch wenn die Behandlung z. B. von Transformationsmatrizen oder Elementen der klassischen Differenzialgeometrie sehr reizvoll erschien und

<sup>53</sup> Dieses Video steht unter [26] einschließlich der POV-Ray-Quelldatei zur Verfügung.

viele Anknüpfungsmöglichkeiten an die Computergrafik böte. Jedoch benötigt die Einarbeitung der Schüler in ein 3D-Grafikprogramm und die Thematisierung von Grundlagen der 3D-Computergrafik Zeit und stellt selbst einen Unterrichtsschwerpunkt dar. Ob darüber und über die obligatorischen Inhalte des Unterrichts hinaus Zeit vorhanden ist, um weitergehende Themen zu behandeln, erscheint in der großen Mehrheit sowohl der Grund- als auch der Leistungskurse fraglich.

Es soll nicht verschwiegen werden, dass für den Einsatz von 3D-Grafiksoftware einige erschwerende Rahmenbedingungen bestehen. Dazu gehört die zeitliche Beschränkung des Lehrgangs, die dazu zwingt, die Neuaufnahme von Inhalten durch die Streichung bzw. Straffung bestehender Themen zu kompensieren.<sup>54</sup> Zudem sind die Voraussetzungen der Schüler hinsichtlich des Umgangs mit dem Computer oft sehr unterschiedlich; ein Teil der Schüler benötigt viel Einarbeitungszeit für die Bedienung der Software. Schließlich erschwert das z. T. schlecht ausgeprägte Raumvorstellungsvermögen vieler Schüler den Umgang mit 3D-Grafiksoftware.<sup>55</sup>

Diesen ernst zu nehmenden Problemen steht aber auch eine große Chance gegenüber, die sich aus der Einbeziehung der 3D-Computergrafik in den Unterricht ergibt: Die Attraktivität der dreidimensionalen Computergrafik ruft erfahrungsgemäß zumindest bei einigen Schülern eine große Begeisterung hervor und regt sie dazu an, über das übliche Maß hinaus zu Hause mit der Software zu arbeiten. Es ist also möglich, dass sich in vielen Lerngruppen „Experten“ im Umgang damit herausbilden. Durch geschickte Unterrichtsgestaltung (Arbeit in heterogenen Gruppen) kann deren Spezialwissen genutzt werden. Eine weitere Möglichkeit, Unterschiede sowohl in den mathematischen Fähigkeiten als auch in den Fähigkeiten im Umgang mit dem Computer zu berücksichtigen, besteht darin, in differenzierter Weise vorgefertigte Dateien für die Arbeit bereitzustellen, bei denen Teile der zu erstellenden 3D-Szenen bereits vorbereitet sind. Einige Beispiele hierfür wurden in diesem Artikel erwähnt; entsprechende Materialien sind auf der Internetseite [26] zu finden.

## Literatur

- [1] Andraschko, H.: DreiDGeo – ein Programm zur Veranschaulichung der analytischen Geometrie im  $E^3$ . In: Der Mathematikunterricht 47 (2001) 5, S. 54–68.

---

<sup>54</sup> Allerdings kann der durch die Visualisierung von Gegenständen mögliche anschauliche Zugang auch das Verständnis erleichtern und somit Zeit sparen helfen. Straffungsmöglichkeiten bestehen m. E. bei den teilweise sehr ausgedehnten Routineaufgaben z. B. zu Lagebeziehungen und Schnittberechnungen.

<sup>55</sup> Andererseits ist die Schulung des Raumvorstellungsvermögens (soweit dies in dem Alter der Schüler noch möglich ist) ein wichtiges Unterrichtsziel. Durch die Erstellung von 3D-Szenen kann dazu ein beachtlicher Beitrag geleistet werden, da die Schüler hierbei ständig Verbindungen zwischen betrachteten Bildern, ihren eigenen Vorstellungen von der Anordnung der Objekte im Raum sowie den Koordinaten herstellen müssen.

- [2] Aumann, G.; Spitzmüller, K.: Computerorientierte Geometrie. B.I. Wissenschaftsverlag, Mannheim, 1993.
- [3] Borneleit, P.; Danckwerts, R.; Henn, H.-W.; Weigand, H.-G.: Expertise zum Mathematikunterricht in der gymnasialen Oberstufe. in: Journal für Mathematik-Didaktik 22 (2001) 1, S. 73–90.
- [4] Bürger, H.; Fischer, R.; Malle, G.; Reichel, H.-C.: Zur Einführung des Vektorbegriffes: Arithmetische Vektoren mit geometrischer Deutung. In: Journal für Mathematik-Didaktik 1 (1980) 3, S. 171–187.
- [5] Bungartz, H.-J.; Griebel, M.; Zenger, C.: Einführung in die Computergrafik. 2. Aufl., Vieweg, Braunschweig, Wiesbaden, 2002.
- [6] Einheitliche Prüfungsanforderungen (EPA) im Fach Mathematik. Beschluss der 298. Kultusministerkonferenz am 23./24.05.2002 in Eisenach, Vorabdruck, Berlin, 2002.
- [7] Fellner, W. D.: Computergrafik. B.I. Wissenschaftsverlag, Mannheim, 1992.
- [8] Foley, J. D.; Van Dam, A.; Feiner, S. K.; Hughes, J. F.; Phillips, R. L.: Grundlagen der Computergraphik. Addison Wesley, Bonn, 1994.
- [9] Gieding, M.: Möglichkeiten und Aspekte des Arbeitens mit Elementen der Computergrafik im Geometrieunterricht der Abiturstufe in der DDR, Humboldt-Universität zu Berlin, Sektion Mathematik, Dissertation, 1988.
- [10] Gouraud, H.: Continuous Shading of Curved Surfaces, IEEE Transactions on Computers, Vol. C-20, 1971.
- [11] Maeder, R. E.: Ray Tracing and Graphics Extensions. In: The Mathematica Journal 4 (1994) 3 (<http://www.mathconsult.ch/showroom/ray/ray.html>).
- [12] Meyer, J.: Bezierkurven. In: Istron – Materialien für einen realitätsbezogenen Mathematikunterricht, Band 6: Computeranwendungen, Franzbecker, Hildesheim, 2000, S. 44–60.
- [13] Lehmann, E.: Mathematik-Unterricht mit Computer-Einsatz (Bd. 2). Dümmler, Bonn, 1988.
- [14] Lehmann, E.: Abbildungsgeometrie mit Matrizen als Grundlage der Computergrafik. In: Proceedings – Bericht über die 14. Arbeitstagung des Arbeitskreises Mathematikunterricht und Informatik in der Gesellschaft für Didaktik der Mathematik e.V., Franzbecker, Hildesheim, 1997.
- [15] Schumann, H.: Computergestützte Behandlung analytisch-geometrischer Aufgaben im  $E^3$ . In: Beiträge zum Computereinsatz in der Schule, PH Weingarten, 15 (2001) 1, S. 65–76.
- [16] Schulz, W.; Stoye, W. (Hrsg.): Analytische Geometrie, Leistungskurs. Volk und Wissen, Berlin, 1998.

- [17] Schupp, H.: Geometrie in der Sekundarstufe II. in: Journal für Mathematik-Didaktik 21 (2000) 1, S. 50–60.
- [18] Tietze, U.-P.: Fundamentale Ideen der linearen Algebra und analytischen Geometrie — Aspekte der Curriculumsentwicklung im MU der SII. In: mathematica didactica 2 (1979) 3, S. 137–164.
- [19] Tietze, U.-P.: Analytische Geometrie und Lineare Algebra im MU — Unterschiedliche Ansätze und deren didaktische Rechtfertigung. In: Mathematica Didactica (1981), Sonderheft 1, S. 57 – 99.
- [20] Tietze, U.-P. (unter Mitarbeit von P. Schroth und G. Wittmann): Didaktik der Analytischen Geometrie und Linearen Algebra. In: Tietze, U.-P.; Klika, M.; Wolpers, H. (Hrsg.): Mathematikunterricht in der Sekundarstufe II, Band 2, Vieweg, Braunschweig, Wiesbaden, 2000.
- [21] Tönnies, K. D.; Lemke, H. U.: 3D-Computergrafische Darstellungen. Oldenbourg, München, Wien, 1994.
- [22] Watt, A.: 3D-Computergrafik. Pearson Education, München, 2002.
- [23] Weber, F.: Schulung des räumlichen Vorstellungsvermögens in der Analytischen Geometrie durch Geovek. In: mathematik lehren 34 (1989), S. 40–44.
- [24] Winter, H.: Mathematikunterricht und Allgemeinbildung. In: Mitteilungen der Gesellschaft für Didaktik der Mathematik 61 (1995), S. 37–46.
- [25] Wunderling, H.: Analytische Geometrie und Lineare Algebra mit Computergraphik und Matrizenrechner. In: mathematik lehren 34 (1989), S. 45–49.

### Materialien im Internet

- [26] Internetseite des Autors mit allen Beispieldateien, zugehörigen farbigen Abbildungen und Ergänzungen zu diesem Artikel:  
[alt: <http://www-didaktik.mathematik.hu-berlin.de/org/filler/3D>]  
Aktualisierung: <http://www.afiller.de/3dcg>
- [27] POV-Ray-Homepage mit Dokumentation, Beispielen und der Möglichkeit, das Programm herunterzuladen: <http://www.povray.org/>

### Anschrift des Verfassers

Dr. Andreas Filler  
Humboldt-Universität zu Berlin  
Mathematisch-Naturwissenschaftliche Fakultät II  
Institut für Mathematik  
Mathematik und ihre Didaktik  
Unter den Linden 6  
10099 Berlin

E-Mail: [filler@mathematik.hu-berlin.de – alt]; aktuell: filler@ph-heidelberg.de