II. Mathematische Grundlagen: Vektorgeometrie

Was dieses Kapitel nicht ist

An dieser Stelle soll kein umfassender Einstieg in die Vektoralgebra bewerkstelligt werden. Vielmehr wird in knapper Form das Wissen aus diesem Gebiet vorgestellt, das notwendig ist, um den mathematischen Prozess beim Raytracingalgorithmus verstehen zu können. Aus ähnlichen Gründen wird darauf verzichtet, auf die konkrete programmiertechnische Umsetzung des Problems einzugehen. Das Hauptaugenmerk dieses Kurses bleibt auf die Anwendung gerichtet, so dass ein Exkurs in Algorithmik und die notwendige Numerik hier keinen großen Sinn macht.

... und was es ist

Das Verständnis der Mathematik ist natürlich für die Erschaffung virtueller Welten eher unerheblich, aber es ermöglicht an den verschiedensten Stellen einen tieferen Einblick in die Verfahrensweisen beim Raytracing.

So ist es möglich leichter zu verstehen, warum manche Anforderungen an ein solches Programm mit einem eventuell überraschend stark vermehrten Rechenaufwand einhergehen und warum der eine oder andere beeindruckende Effekt bei der Rechenzeit kaum zu Buche schlägt.

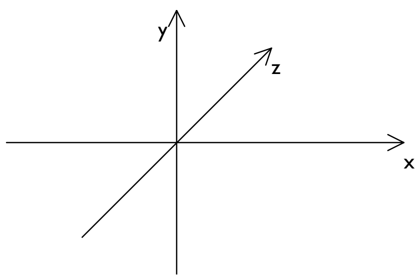
Vor allem aber dann, wenn man ein programmierbares Raytracingsystem sinnvoll umstrukturieren oder erweitern möchte, sollte man zumindest grundlegende Kenntnisse der dahinter stehenden Mathematik besitzen.

Im Klartext: Wenn Ihnen schon bei dem Wort „Vektoralgebra" ein kalter Schauer des Unbehagens über den Rücken läuft, können Sie dieses Kapitel auch überspringen. Wenn Sie aber Freude an ein wenig tieferen Verständnis haben, dann ist dies vielleicht (so hofft zumindest der Autor) eine einigermaßen erhellende Lektüre.

Es wird versucht werden, zumindest zu Beginn dieses Kapitels die Zusammenhänge möglichst einfach darzustellen, um möglichst vielen Lesern einen Zugang zu den grundlegenden Begriffen zu ermöglichen. Im Verlauf wird aber eine deutliche Steigerung im mathematischen Anspruch nicht ganz vermeidbar sein und ist auch durchaus beabsichtigt, um auch dem mathematisch begeisterbaren Leser etwas zu bieten.

Ein passendes Koordinatensystem

Um sich eine Vorstellung von geometrischen Vektoren[1] zu machen benötigt man zunächst ein geeignetes Koordinatensystem im Raum. In diesem Text wird das folgende System verwendet werden:



Dabei verläuft die z-Achse von vorne nach hinten in die Blattebene hinein.

[1] In der Mathematik ist 'Vektor' ein sehr allgemeiner Begriff, der weit mehr beschreiben kann, als eine gerichtete Strecke im Raum oder in der Ebene.. Deshalb wird hier zuweilen der Begriff 'geometrischer Vektor' verwendet.

Es handelt sich hierbei um ein so genanntes linkshändiges Koordinatensystem, da man die drei Achsen durch gespreizten Daumen, Zeige- und Mittelfinger der linken Hand symbolisieren kann. Obwohl ein linkshändiges System eher unüblich ist wird hier mit einem solchen begonnen, da auch PovRay, das in diesem Kurs verwendete Raytracingprogramm ein solches System benutzt.

Dahinter dürfte die simple Tatsache stehen, dass beim Skizzieren virtueller Welten sehr oft ein kleines Koordinatensystem als Referenz nützlich ist. Und da die Mehrzahl der Menschen Rechtshänder sind, ist die linke Hand meist noch frei.

Was ist das, ein Vektor

Ein geometrischer Vektor wird in der Mathematik wie folgt beschrieben:

Ein Vektor ist die Menge aller gleich orientierten und gleich langen, gerichteten Strecken.

Diese Beschreibung ist zwar mathematisch korrekt und sinnvoll, aber für die praktische Anwendung im hier angestrebten Sinn eher wenig hilfreich. Man kann sich einen Vektor vereinfacht vorstellen als einen Pfeil im Raum oder in der Ebene. Dabei kann ein Vektor zwei unterschiedliche Rollen übernehmen:

Er kann vom Nullpunkt des Koordinatensystems aus auf eine bestimmte Position im Raum zeigen und damit einen Punkt, wie z.B. die Ecke eines Quaders festlegen. Ein Vektor kann aber auch eine Richtung im Raum wie zum Beispiel eine Verschiebung oder die Richtung einer Geraden beschreiben. Im Folgenden wird, etwas unmathematisch[2], aber zur besseren Unterscheidung von Ortsvektoren und von Richtungsvektoren die Rede sein.

Die Beschreibung von Vektoren

Ein Vektor wird durch seine Komponenten entlang der Koordinatenachsen im Raum festgelegt. So beschreibt zum Beispiel der folgende Vektor eine Verschiebung um 3 Einheiten in x-Richtung, 4 in y-Richtung und 2 Einheiten in z-Richtung:

$$\vec{a} = \begin{pmatrix} 3 \\ 4 \\ 2 \end{pmatrix}$$

Wie oben beschrieben (als Verschiebung) wäre $\vec{a}$ also ein Richtungsvektor. Man könnte ihn auch verwenden, um die Position des Punktes $A(3|4|2)$ im Raum zu beschreiben, womit es sich um einen Ortsvektor handeln würde. Letzteres ist auch der häufigere Fall in der Computergrafik, denn hauptsächlich dienen hier Vektoren zur Positionsbeschreibung von verschiedenen Objekten.

Grundlegende Rechenoperationen mit Vektoren

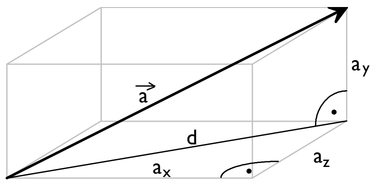
- *Addition von Vektoren*
  Zwei Vektoren werden addiert, indem man ihre einzelnen Komponenten addiert:
  $$\vec{a} \pm \vec{b} = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} \pm \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} = \begin{pmatrix} a_x \pm b_x \\ a_y \pm b_y \\ a_z \pm b_z \end{pmatrix}$$
  Ist zum Beispiel $\vec{a}$ ein Vektor, der ein Eck eines Tisches beschreibt und $\vec{b}$ ein Verschiebungsvektor, so wäre die Summe der beiden die neue Position der Tischecke, nachdem der Tisch um $\vec{b}$ verschoben wurde.
- *S-Multiplikation eines Vektors*
  Die Multiplikation mit einer Zahl, die zur Unterscheidung von Vektoren auch als Skalar bezeichnet wird (deshalb *S*-Multiplikation), geschieht komponentenweise.

[2] Von mathematischer Seite her ist diese Unterscheidung in zwei 'Vektorarten' völlig unnötig, da sie rechnerisch vollkommen identisch behandelt werden.

$$\lambda \cdot \vec{a} = \lambda \cdot \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} = \begin{pmatrix} \lambda \cdot a_x \\ \lambda \cdot a_y \\ \lambda \cdot a_z \end{pmatrix}$$

Dies entspricht einer Streckung mit dem Streckungsfaktor λ vom Koordinatenursprung aus.

- *Verbindungsvektor zweier Punkte*
  Sind die Punkte $A(a_x|a_y|a_z)$ und $B(b_x|b_y|b_z)$ mit ihren Koordinaten gegeben, so berechnet sich der Verbindungsvektor $\overrightarrow{AB}$ von A nach B wie folgt:
  $$\overrightarrow{AB} = \vec{b} - \vec{a} = \begin{pmatrix} b_x - a_x \\ b_y - a_y \\ b_z - a_z \end{pmatrix}$$
- *Vektorlänge*
  Diese wird auch als Betrag des Vektors bezeichnet. Symbolisiert wird sie dementsprechend entweder durch Betragsstriche, oder durch Weglassen des Vektorpfeils:
  $$|\vec{a}| = a = \sqrt{a_x^2 + a_y^2 + a_z^2}$$
  Diese Formel leitet sich aus dem Satz von Pythagoras ab. An Hand der folgenden Skizze soll dies kurz verdeutlicht werden:



Betrachtet man $\vec{a}$ als Diagonale eines Quaders, dessen Kantenlängen seinen Komponenten entsprechen, so gilt für die Bodendiagonale d nach Pythagoras:
$$d^2 = a_x^2 + a_z^2$$
Eingesetzt in die entsprechende Beziehung im anderen rechtwinkligen Dreieck ergibt sich insgesamt:
$$a^2 = d^2 + a_y^2 = a_x^2 + a_z^2 + a_y^2$$
und daraus durch Ziehen der Wurzel die obige Gleichung.

Übung macht den Meister

Im Anschluss an diese ersten Grundlagen ist es sinnvoll, die entsprechenden Begriffe zumindest im Geiste zu trainieren. So man Schwierigkeiten mit der räumlichen Vorstellung hat ist es hilfreich, sich in einem gedachten Koordinatensystem, dessen Achsen zum Beispiel von drei Kanten eines Raumes gebildet werden, verschiedene Positionen mittels ihrer Koordinaten zu beschreiben.

Überlegen Sie sich, welche Komponenten der Ortsvektor, der auf den Lichtschalter zeigt, besitzt, wo die Steckdose liegt oder welcher Verschiebungsvektor den Bleistift an den Rand des Tisches bringen würde. Um das Denken in Koordinaten etwas zu schulen können solche einfachen Überlegungen in kurzen, ruhigen Momenten durchaus hilfreich sein. Ein konzentriertes, mehrstündiges Intensivtraining ist meist weniger effektiv.

Auch für die noch folgenden Inhalte kann es sinnvoll sein, sich immer wieder entsprechende Objekte geistig oder in der momentanen Umgebung vorzustellen.

Das räumliche Vorstellungsvermögen ist individuell sehr unterschiedlich ausgeprägt, so dass es gut sein kann, dass ein Training in der oben genannten Weise für manchen völlig überflüssig ist. Aber obwohl viele, an dreidimensionaler Computergrafik interessierte Personen eine gute Raumvorstellung besitzen kann auch hier ein Training im Umgang mit Koordinaten hilfreich sein. Ein gutes Gefühl für Abstände und Lagebeziehungen erspart manch zeitaufwendiges Herumprobieren oder Verzweifeln, weil ein Objekt schon wieder nicht an der korrekten Position platziert wurde.

Geht es später dann um die Erstellung eigener dreidimensionaler Welten sollte man sich auch nicht scheuen, vorher zumindest eine Skizze der zu erschaffenden Szene anzufertigen, da durch diese ebenfalls viele unnötige Fehler vermieden werden können. Ein wenig zeichnerisches Geschick ist also ebenfalls sinnvoll.

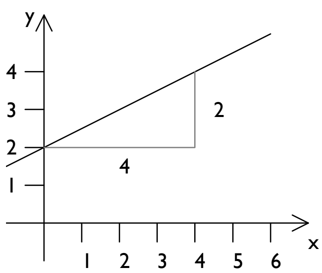
Aufgaben

Ein wenig konkretere Aufgaben sollen bei Bedarf helfen, die obigen Begriffe ein wenig zu vertiefen:

1. Ein Raum hat die Abmessungen lxbxh: 6x4x3. Zeichnen Sie eine Skizze und beschriften Sie die Eckpunkte, so wie die Kanten und Seitenmitten mit ihren Koordinaten. Welche Koordinaten hat die Raummitte des Quaders?
2. Die Ecken eines Dreiecks liegen bei A(2; 3; 5), B(-1; 4; -2) und C(5; 7; 3). Bestimmen Sie die Vektoren, die von einer zur nächsten Ecke zeigen und bilden Sie dann deren Summe! Warum ist diese Summe eine mögliche Überprüfung der vorher ermittelten Vektoren?
3. Bestimmen Sie eine Formel, mit der man aus den Endpunktkoordinaten einer Strecke die Koordinaten des Mittelpunktes bestimmen kann und überprüfen Sie diese in einer Zeichnung.
4. Bestimmen Sie zur Strecke [AB] aus Aufgabe 2 die beiden Teilpunkte T1 und T2 die diese im Verhältnis 1:3 teilen!
5. Berechnen Sie den Umfang des Dreiecks aus Aufgabe 2!

Geraden im Raum

In der Algebra werden Geraden schon früh behandelt. Sie tauchen als Graphen von linearen Funktionen auf. Will man eine bestimmte vorgegebene Gerade durch eine geeignete Funktionsvorschrift mathematisch beschreiben, so benötigt man dazu zwei Informationen. Dies sind im klassischen Fall der Schnittpunkt der Geraden mit der y-Achse und die Steigung der Geraden, die sich mit einem Steigungsdreieck ablesen lässt:



Die beiden Informationen führen dann zu der zugehörigen Funktionsgleichung:

$$g : y = 2 + \tfrac{2}{4} \cdot x = 2 + \tfrac{1}{2}x$$

Die Gerade ergibt sich hieraus, wenn x jeden beliebigen Wert einmal annimmt.

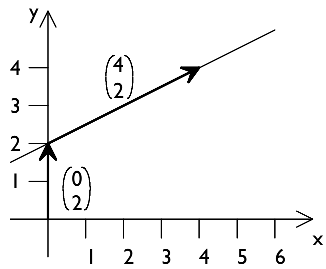
Die gleichen Informationen lassen sich auch vektoriell beschreiben. Ein Ortsvektor beschreibt $\vec{p}$ die Position eines Punktes der Geraden, wobei hier vereinfachend wieder der Schnittpunkt mit der y-Achse verwendet wird. Ein Richtungsvektor $\vec{r}$ legt die Richtung der Geraden fest:

Addiert man zum Ortsvektor nacheinander alle möglichen Vielfachen des Richtungsvektors, so bilden die entsprechenden Summenvektoren die gewünschte Gerade. Die formale Beschreibung der obigen Geraden in der Vektorgeometrie sieht dann zum Beispiel wie folgt aus[3]:

[3] Zur besseren Vorstellung werden die ersten Beispiele zunächst zweidimensional behandelt. Die Form einer Geraden im Raum ist bis auf die zusätzliche dritte Koordinate identisch.

$$g : \vec{x} = \vec{p} + \lambda \cdot \vec{r} = \begin{pmatrix} 0 \\ 2 \end{pmatrix} + \lambda \cdot \begin{pmatrix} 4 \\ 2 \end{pmatrix}$$

Der Parameter λ darf dabei beliebige Werte annehmen, was den oben beschriebenen Vielfachen des Richtungsvektors entspricht.



Sowohl der Ortsvektor als 'Startpunkt' der Geraden, wie auch ihr Richtungsvektor können unterschiedlich gewählt werden. So kann ein und die selbe Gerade auf unendlich viele verschiedene Arten dargestellt werden. Zum Beispiel beschreiben die folgenden beiden Gleichungen ebenfalls die oben gezeichnete Gerade:

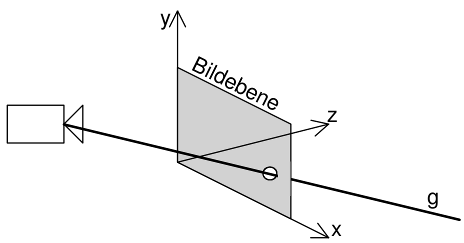
$$g : \vec{x} = \begin{pmatrix} 2 \\ 3 \end{pmatrix} + \lambda \cdot \begin{pmatrix} 1 \\ 0,5 \end{pmatrix} \qquad g : \vec{x} = \begin{pmatrix} 0 \\ 2 \end{pmatrix} + \lambda \cdot \frac{\sqrt{5}}{5} \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

Im zweiten Fall ist der Richtungsvektor normiert, was bedeutet, das er die Länge 1 besitzt. Dies ist sinnvoll, da später an verschiedensten Stellen der Abstand zwischen Objekten und damit die Länge von Strecken eine entscheidende Bedeutung haben. Liegt eine Gerade mit normierten Richtungsvektor vor, so kann an Hand des berechneten Parameters λ direkt der Abstand abgelesen werden. Schneidet zum Beispiel die oben angegebene Gerade eine Kugeloberfläche für $\lambda = 3,7$ so bedeutet dies, dass der 'Startpunkt' der Geraden (zum Beispiel die Kamera) von diesem Auftreffpunkt 3,7 Längeneinheiten entfernt ist.

Im Falle der ersten beiden Geradengleichungen wären hier immer zusätzliche Berechnungen für die Abstände notwendig.

Eine Abtastgerade beim Raytracingverfahren

Beim Raytracing ist es immer wieder notwendig eine Gerade zwischen zwei gegebenen Punkten zu bestimmen. So zum Beispiel die im ersten Kapitel beschriebene Gerade von der Kamera durch einen der Bildpunkte der zu berechnenden Grafik. Eine solche Situation ist im Folgenden vereinfacht dargestellt:

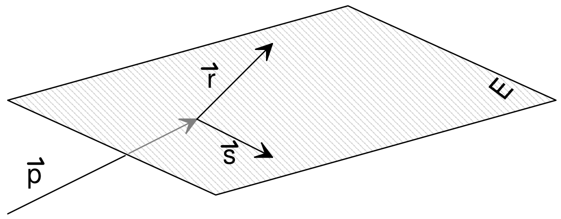


Als Ortsvektor benutzt man zum Beispiel die Kameraposition K. Der Richtungsvektor ist dann der Vektor von der Kamera zum Bildpunkt B:

$$g : \vec{x} = \vec{k} + \lambda \cdot (\vec{b} - \vec{k})$$

Ebenen im Raum

Eine Ebene im Raum wird in ganz ähnlicher Weise beschrieben wie eine Gerade. Auch hier wird zunächst ein Ortsvektor $\vec{p}$ benötigt, der wie bei der Geraden als Startpunkt dient. Der entscheidende Unterschied ist der, dass in diesem Fall zwei Richtungsvektoren $\vec{r}$ und $\vec{s}$ notwendig sind um die Lage der Ebene im Raum zu beschreiben. Die folgende Zeichnung soll dies verdeutlichen.



Zur Vorstellung mag es behilflich sein, wenn man sich die Ebene E als Platte vorstellt, die auf zwei Stangen (den Richtungsvektoren $\vec{r}$ und $\vec{s}$) gelagert ist. Die Ebene selbst ist in der Zeichnung als endliches Rechteck dargestellt, obwohl sie natürlich unendlich groß ist. Die Darstellung obiger Ebene in Parameterform sieht wie folgt aus:

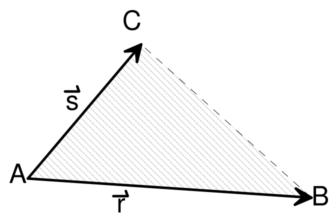
$$E: \vec{x} = \vec{p} + \lambda \cdot \vec{r} + \mu \cdot \vec{s}$$

Dabei durchlaufen die Parameter λ und μ alle möglichen Zahlenkombinationen, wodurch die dargestellten Vektoren $\vec{x}$ sämtliche Punkte der gewünschten Ebene gewissermaßen abtasten.

Dreidimensionale Dreiecksnetze

Um später beliebige räumliche Formen beschreiben zu können benötigt man zunächst einige wenige Grundelemente, aus denen diese Formen dann zusammengesetzt werden können. In den meisten Raytracingprogrammen werden die Oberflächen von Gegenständen[4] durch ein Netz aus aneinander gesetzten Dreiecken beschrieben. Indem man die Aufteilung in Dreiecke nur fein genug macht, kann so jede beliebige Form ausreichend exakt beschrieben werden.

Wie stellt man aber nun ein Dreieck mathematisch dar? Indem man für λ und μ nicht mehr beliebige Werte erlaubt, sondern an diese die passenden Bedingungen stellt, kann die unendliche Ebene von oben auf ein endliches Dreieck begrenzt werden, das zwischen den beiden Richtungsvektoren liegt:



In den meisten Fällen wird ein solches Dreieck über die Koordinaten seiner Eckpunkte festgelegt. Die benötigten Richtungsvektoren ergeben sich, wie oben bereits beschrieben als Vektoren zwischen zwei Punkten:

$$E_{\Delta}: \vec{x} = \vec{a} + \lambda \cdot \overrightarrow{AB} + \mu \cdot \overrightarrow{AC} = \vec{a} + \lambda \cdot (\vec{b} - \vec{a}) + \mu \cdot (\vec{c} - \vec{a})$$

[4] Im Allgemeinen werden Objekte beim Raytracing durch ihre Oberfläche festgelegt, da selbst bei transparenten Objekte die, für den Lichtweg entscheidenden, Prozesse an der Oberfläche stattfinden. Inhomogene innerer Strukturen wie z.B. Rauch oder Schlieren in Glas werden meist auf anderem Weg realisiert.

Um aus dieser Ebene nur die Punkte auszuwählen, die innerhalb des dargestellten Dreiecks liegen müssen die Parameter positiv sein und ihre Summe darf nicht größer als Eins werden:

$$\lambda,\mu \geq 0; \quad \lambda+\mu \leq 1$$

Mit diesen Elementen ist es nun möglich das Dreieck als Menge von Ortsvektoren zu beschreiben:

$$D_{ABC} = \{\vec{x} = \vec{a} + \lambda \cdot (\vec{b} - \vec{a}) + \mu \cdot (\vec{c} - \vec{a}) \mid \lambda,\mu \geq 0;\ \lambda+\mu \leq 1\}$$

Um verständlicher zu machen, warum genau diese Einschränkung der Parameter das gewünschte Ergebnis liefert kann es helfen, wenn man sich klar macht, wie die Linien beschrieben werden, die das Dreieck begrenzen:

Ist $\lambda = 0$, so erhält man Punkte auf der Kante [AC] des Dreiecks und für $\mu = 0$ ergeben sich die Punkte der Seite [AB]. Alle Punkte der Strecke [BC] erreicht man, wenn man alle Kombinationen der Parameter durchläuft, für die $\lambda + \mu = 1$ gilt, durchläuft. An dieser Stelle, wie auch an vielen folgenden ist es sinnvoll diese Gedankengänge auf einem Stück karierten Papier selbst an einem konkreten Beispiel einmal durchzuspielen.

Der Preis vieler Details

Wie man sich leicht vorstellen kann steigt die Anzahl der notwendigen Dreiecke mit der Komplexität der Objekte sehr schnell an. Reichen für ein einfacheres Gebäude oder eckige Maschinenteile noch einige Tausend Dreiecke vollkommen aus, so sind für organische Formen sehr schnell mehrere Zehntausend oder gar Hunderttausend Dreiecke notwendig. Dies wiederum steigert den benötigten Speicherplatz als auch die Zeit für die Bildberechnung enorm, trotz aller, heute gebräuchlicher Ideen, um den Rechenprozess zu optimieren.

Auf der anderen Seite sind Polygonnetze beliebig deformierbar, indem die Koordinaten der Eckpunkte verschoben werden. Und in den meisten Fällen ist dies der übliche Weg, um die Bewegungen von animierten Charakteren zu ermöglichen.

Beim Entwerfen einer virtuellen Szenerie wird es aus den obigen Gründen immer wieder notwendig, das richtige Maß zwischen dem Anspruch an möglichst realistischer Darstellung und Detailtreue auf der einen Seite und Begrenzung der Rechenzeit auf der andern zu treffen. Eine Problematik, die, in unterschiedlicher Größenordnung den Hobbykünstler am heimischen PC genauso betrifft wie eine Firma für professionelle Spezialeffekte, der eine so genannte Renderwall[5] zur Verfügung steht.

Andere endliche Objekte

Ein Dreieck ist bei weitem nicht die einzige Grundform, die mit einer relativ einfachen Gleichung als Punktmenge beschrieben werden kann. So ist zum Beispiel die Oberfläche einer Kugel definiert als die Menge aller Punkte mit gleichem Abstand zu einem gegebenen Mittelpunkt M. Im vereinfachten Fall, dass der Mittelpunkt im Ursprung des Koordinatensystems liegt ist eine Kugel mit Radius r beschreibbar durch:

$$K = \{\vec{x} \mid |\vec{x}| = r\}$$

Die Koordinaten x, y und z der Oberflächenpunkte müssen also die folgende Gleichung erfüllen:

$$\sqrt{x^2 + y^2 + z^2} = r \quad \text{oder} \quad \frac{x^2}{r^2} + \frac{y^2}{r^2} + \frac{z^2}{r^2} = 1$$

Wandelt man diese nur leicht ab, so ergeben sich ein Ellipsoid: $\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1$ mit den Halbachsen a, b und c oder unendliche Formen wie ein zweischaliges Hyperboloid, auf die hier aber noch nicht detailliert eingegangen werden soll.

---

[5] Eine Phalanx aus mehreren Hochleistungsrechnern, die für Bildberechnung optimiert wurden. Im Fall der Firma Weta Digital (verantwortlich für die Spezialeffekte im Herrn der Ringe) sind dies zum Beispiel 600 Dualprozessorrechner mit bis zu 2,2GHz Prozessortakt.

Die durch Gleichungen beschreibbaren Objekte sind dabei unbegrenzt. Der interessierte Leser könnte sich an dieser Stelle überlegen, wie man in ähnlicher Form eine Kreisscheibe, einen Zylinder oder einen Kegel beschreiben könnte.

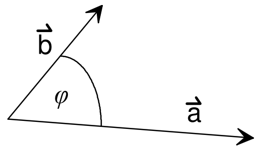
An dieser Stelle soll dieser kurze Ausblick in die Möglichkeiten mit komplexeren Grundformen zunächst genügen. Bei der Behandlung von Objekttypen in PovRay wird auf die verschiedenen, dort gebotenen Optionen noch genauer eingegangen. PovRay kann zwar auch mit Polygonnetzen (sog. Meshes) arbeiten, bietet aber im Gegensatz zu vielen anderen Tracern eine Vielzahl an teilweise hochkomplexen Grundobjekten zur Verfügung.

Diese haben dann den Nachteil, dass sie nicht so wahlfrei deformierbar sind wie Netzobjekte, aber sie kommen oft trotz relativ komplexer Formen mit einer deutlich kleineren Anzahl an Objekten aus.

Winkel und Vektoren

Wir kehren zunächst zur reinen Vektorrechnung zurück:

- *Skalarprodukt*
  An vielen Stellen ist es notwendig den Winkel zwischen zwei Vektoren berechnen zu können. Zu diesem Zweck kann das Skalarprodukt benutzt werden:
  $$\vec{a}\cdot\vec{b}=\begin{pmatrix}a_x\\a_y\\a_z\end{pmatrix}\cdot\begin{pmatrix}b_x\\b_y\\b_z\end{pmatrix}=a_x\cdot b_x+a_y\cdot b_y+a_z\cdot b_z=|a|\cdot|b|\cdot\cos\varphi$$
  Dabei ist φ der Winkel zwischen den beiden Vektoren.



- *Vektorprodukt* oder *Kreuzprodukt*
  $$\vec{a}\times\vec{b}=\begin{pmatrix}a_x\\a_y\\a_z\end{pmatrix}\times\begin{pmatrix}b_x\\b_y\\b_z\end{pmatrix}=\begin{pmatrix}a_yb_z-a_zb_y\\a_zb_x-a_xb_z\\a_xb_y-a_yb_x\end{pmatrix}$$
  Der so berechnete Vektor steht sowohl auf $\vec{a}$ als auch auf $\vec{b}$ senkrecht. Sind $\vec{a}$ und $\vec{b}$ die Richtungsvektoren einer Ebene, so ist $\vec{a}\times\vec{b}$ der so genannte *Normalenvektor* dieser Ebene. Um sich vorstellen zu können wie dieser Normalenvektor im Bezug zu den beiden anderen Vektoren steht richtet man den Daumen der rechten Hand in Richtung von $\vec{a}$ und den Zeigefinger in Richtung von $\vec{b}$ aus. Der senkrecht zu beiden gestellte Mittelfinger ergibt die Richtung des Normalenvektors.

Aufgaben

Zur Übung im Umgang mit Geraden, Ebenen und anderen Objekten können die folgenden Aufgaben hilfreich sein:

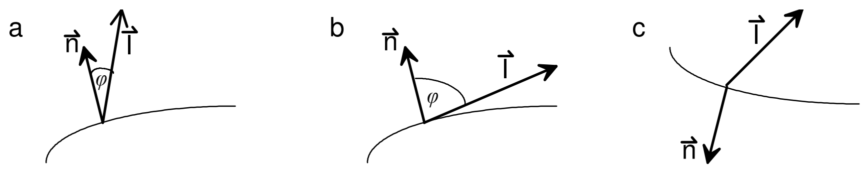
6. Gegeben sind die Punkte A(2; 3; 0), B(5; 0; 4) und C(0; 2; 2), sowie T(-2; -1;-1) und S(5; 5; 1). Stellen Sie die Ebene durch A, B und C, sowie die Gerade durch die anderen beiden Punkte in Parameterschreibweise dar!
7. Ermitteln Sie den Schnittpunkt von Ebene und Gerade aus Aufgabe 6! Hierzu müssen Sie zuerst die Parameter bestimmen, indem Sie die beiden Ausdrücke gleich setzen.
8. Beschreiben Sie mathematisch einen Kreis mit Radius 2 um den Mittelpunkt (3; 3)!
9. Ermitteln Sie durch Rechnung die Schnittpunkte des Kreises aus Aufgabe 8 mit der Ursprungsgeraden durch P(5; 3)! Überprüfen Sie eventuell in einer Zeichnung ihr Ergebnis!
10. Bestimmen Sie einen Vektor, der senkrecht auf der Ebene von Aufgabe 6 steht (der sogenannte Normalenvektor)!
11. In welchem Winkel trifft die Gerade aus Aufgabe 6 auf die Ebene?
12. Mit welchem Einfallswinkel trifft die Gerade aus Aufgabe 9 den Kreis aus Aufgabe 8?

Nutzung von Winkelinformationen

Wann immer bei der Strahlverfolgung ein Sehstrahl auf eine Oberfläche trifft kommen der Normalenvektor der entsprechenden Oberfläche, als auch das Skalarprodukt zum Einsatz, meist sogar mehrfach:

Wie In Kapitel I beschrieben ist der Winkel, in dem das Licht auf eine Oberfläche trifft entscheidend für die Helligkeit der selben. Rechnerisch wird das Skalarprodukt des Strahls zur Lichtquelle $\vec{l}$ und des Normalenvektors $\vec{n}$ berechnet. Ist der Winkel zwischen den beiden Vektoren klein (siehe unten bei a), so bedeutet dies nahezu senkrechten Lichteinfall und damit einen hell beleuchteten Punkt, während mit größer werdendem Winkel auch die Helligkeit der Oberfläche abnimmt (bei b).

Ein nützlicher Nebeneffekt der obigen Berechnung ist, dass problemlos festgestellt werden kann, ob der Lichtstrahl von außen oder von innen auf die Oberfläche trifft, da im letzteren Fall das Skalarprodukt ein negatives Vorzeichen besitzt (siehe c). Dabei muss natürlich der übliche Standard eingehalten werden, bei dem der Normalenvektor immer von einer Oberfläche nach außen zeigt.



Der gleiche Winkel ist natürlich auch von entscheidender Bedeutung bei der Berechnung von reflektierten oder gebrochenen Lichtstrahlen.

Eine kleine Beispielrechnung

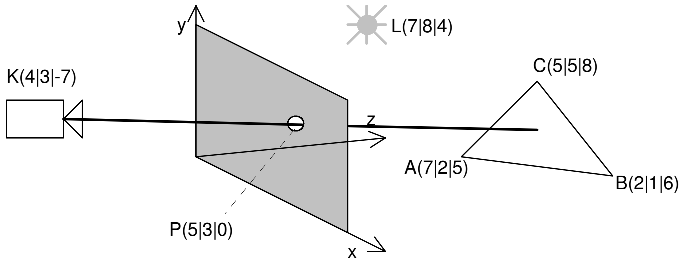
Nun stehen prinzipiell die notwendigen Kenntnisse aus der Mathematik zur Verfügung, um in einem Beispiel die einzelnen Rechenschritte nachvollziehen zu können, die notwendig sind, um die Helligkeit eines einzelnen Bildpunktes zu berechnen. Um dies nicht unnötig kompliziert zu gestalten wird auf die Einbeziehung von speziellen Oberflächeneigenschaften, sowie von Brechung und Reflexion bewusst verzichtet.

An dieser Stelle könnte ein neuerliches Durchlesen von Kapitel I sinnvoll sein, um nochmals im Überblick zu verstehen, was nun im Detail rechnerisch vollzogen werden soll. Beim Durcharbeiten der Beispielrechnung ist es durchaus sinnvoll Papier und Stift bereitliegen zu haben, um so viele Rechenschritte, wie man sich zutraut selbst zu erarbeiten und die hier folgenden Rechnungen nur als Kontrolle zu nutzen.

Aber es sei hier auch noch einmal die Bemerkung vom Beginn dieses Kapitels wiederholt: Es ist nicht notwendig die folgenden Rechnungen vollständig zu verstehen, aber wenn der Respekt vor dem eigenen Computer schwindet, so könnte dies ein wirksames Gegenmittel sein.

Die virtuelle Szenerie

Im Folgenden ist eine sehr stark vereinfachte Szene dargestellt, mit deren Daten die Beispielrechnung durchgeführt werden soll. Nur ein einzelnes Dreieck, sowie eine Lichtquelle befinden sich neben der Kamera im Raum.



Vorbereitungen

Zunächst werden die Gleichungen aufgestellt, die die einzelnen beteiligten Objekte mathematisch darstellen. Dies sind der zu verfolgende Abtaststrahl, das Dreieck und auch dessen Normalenvektor. Ähnliche Berechnungen stellt auch ein Rytracingprogramm im Voraus an und legt diejenigen Daten, die noch mehrfach gebraucht werden, wie zum Beispiel die Normalenvektoren aller Dreiecke, im Speicher ab um Rechenzeit durch Wiederholungen zu sparen.

- *Sehstrahl*

$$s: \vec{x} = \vec{k} + \lambda \cdot \overrightarrow{KP} = \begin{pmatrix} 4 \\ 3 \\ -7 \end{pmatrix} + \lambda \cdot \begin{pmatrix} 1 \\ 0 \\ 7 \end{pmatrix} \quad \text{mit } \lambda \geq 0$$

Die Bedingung an λ wird gestellt, damit nicht die ganze Gerade in die Berechnung einbezogen wird, wodurch auch Objekte hinter der Kamera auf dem Bild erscheinen würden.

- *Dreieck*

$$D: \vec{x} = \vec{a} + \mu \cdot \overrightarrow{AB} + \nu \cdot \overrightarrow{AC} = \begin{pmatrix} 7 \\ 2 \\ 5 \end{pmatrix} + \mu \cdot \begin{pmatrix} -5 \\ -1 \\ 1 \end{pmatrix} + \nu \cdot \begin{pmatrix} -2 \\ 3 \\ 3 \end{pmatrix}; \quad \mu, \nu \geq 0; \ \mu + \nu \leq 1$$

- *Normalenvektor*

$$\vec{n} = \overrightarrow{AB} \times \overrightarrow{AC} = \begin{pmatrix} -5 \\ -1 \\ 1 \end{pmatrix} \times \begin{pmatrix} -2 \\ 3 \\ 3 \end{pmatrix} = \begin{pmatrix} -3-3 \\ -2+15 \\ -15-2 \end{pmatrix} = \begin{pmatrix} -6 \\ 13 \\ -17 \end{pmatrix}$$

Schnittpunktberechnung

Um zu ermitteln, ob der Sehstrahl das Dreieck trifft und wenn ja, wo sich der Schnittpunkt befindet, werden die rechten Seiten der Gleichungen von Sehstrahl und Dreieck gleichgesetzt.

$$\begin{pmatrix} 4 \\ 3 \\ -7 \end{pmatrix} + \lambda \cdot \begin{pmatrix} 1 \\ 0 \\ 7 \end{pmatrix} = \begin{pmatrix} 7 \\ 2 \\ 5 \end{pmatrix} + \mu \cdot \begin{pmatrix} -5 \\ -1 \\ 1 \end{pmatrix} + \nu \cdot \begin{pmatrix} -2 \\ 3 \\ 3 \end{pmatrix}$$

Daraus ergibt sich ein Gleichungssystem mit den drei Unbekannten λ, μ und ν.

$$\begin{array}{llllll} 4 & +\lambda & =7 & -5\mu & -2\nu \\ 3 & & =2 & -\mu & +3\nu \\ -7 & +7\lambda & =5 & +\mu & +3\nu \end{array}$$

Die detaillierte Auflösung dieses Systems wird hier übersprungen. Es ergeben sich die Lösungen $\lambda = \frac{47}{25}; \mu = \frac{2}{25}$ und $\nu = \frac{9}{25}$.

Offensichtlich gilt $\mu, \nu \geq 0$ und $\mu + \nu = \frac{2}{25} + \frac{9}{25} = \frac{11}{25} \leq 1$ womit feststeht, dass es nicht nur einen Schnittpunkt von Strahl und Ebene gibt, sondern, dass dieser auch innerhalb des Dreiecks liegt.

Die Koordinaten des Schnittpunktes lassen sich bestimmen, indem $\lambda = \frac{47}{25}$ in die Gleichung des Sehstrahls eingesetzt wird.

$$\vec{q} = \begin{pmatrix} 4 \\ 3 \\ -7 \end{pmatrix} + \frac{47}{25} \cdot \begin{pmatrix} 1 \\ 0 \\ 7 \end{pmatrix} = \begin{pmatrix} \frac{147}{25} \\ 3 \\ \frac{154}{25} \end{pmatrix}$$

Helligkeitsbestimmung

Von dem gefundenen Schnittpunkt wird nun der Lichtweg weiter verfolgt in Richtung Lichtquelle. An dieser Stelle würde sich die obige Prozedur wiederholen, da ja auch dieser Strahl weitere Objekte treffen könnte. Zur Vereinfachung wird hier die Information genutzt, dass es hierzu in Ermangelung weiterer Objekte nicht kommen kann.

Um nun die Helligkeit ermitteln zu können genügt es, den Richtungsvektor vom Schnittpunkt Q zur Lichtquelle zu bestimmen.

$$\overrightarrow{QL} = \begin{pmatrix} 7 \\ 8 \\ 4 \end{pmatrix} - \begin{pmatrix} \frac{147}{25} \\ 3 \\ \frac{154}{25} \end{pmatrix} = \begin{pmatrix} \frac{28}{25} \\ 5 \\ -\frac{54}{25} \end{pmatrix}$$

Aus dem Skalarprodukt und der Länge der beiden Vektoren kann der Kosinus des Zwischenwinkels bestimmt werden.

$$\vec{n} \cdot \overrightarrow{QL} = \begin{pmatrix} -6 \\ 13 \\ -17 \end{pmatrix} \cdot \begin{pmatrix} \frac{28}{25} \\ 5 \\ -\frac{54}{25} \end{pmatrix} = -\frac{168}{25} + 65 + \frac{918}{25} = 95 = |\vec{n}| \cdot |\overrightarrow{QL}| \cdot \cos\varphi$$

$$\Rightarrow \cos\varphi = \frac{95}{|\vec{n}| \cdot |\overrightarrow{QL}|} = \frac{95}{\sqrt{6^2+13^2+17^2} \cdot \sqrt{\frac{28^2+125^2+54^2}{25^2}}} \approx 0,769$$

$$\Rightarrow \varphi \approx 39,8^\circ$$

Dieser Winkel wird wieder gespeichert, da er nicht nur für die Helligkeit, sondern auch für Reflexion und Brechung benötigt wird.

Und in Wahrheit ist alles noch viel schlimmer

Die oben durchgeführte Rechnung zur Bestimmung eines möglichen Schnittpunktes muss prinzipiell für jedes einzelne Objekt innerhalb der Szenerie durchgeführt werden. Um hier den Rechenaufwand nicht ins unermessliche zu steigern wird durch schnellere Berechnungen im Voraus bestimmt, welche Objekte für einen Schnittpunkt in Frage kommen. Dennoch nimmt mit der Anzahl der Objekte natürlich immer auch der Rechenaufwand deutlich zu.

Verstärkt wird dieser Effekt noch dadurch, dass sich die gesamte Prozedur für den, zur Lichtquelle weiterverfolgten Strahl mindestens einmal, bei Reflexion und Brechung sogar vielfach wiederholt.

Wenn man nun nochmals die obigen Rechnungen überflogen hat und sich die Konsequenzen der eben gemachten Anmerkungen vor Augen führt, dann ist spätestens jetzt der passende Zeitpunkt gekommen, um zum Beispiel mit PovRay ein Beispielbild berechnen zu lassen. Lehnt man sich dann entspannt zurück während der Computer arbeitet, so sollte man dabei einmal im Geiste grob überschlagen, wie viele Berechnungen allein für einen einzelnen Bildpunkt notwendig sind, geschweige denn für mehrere Zeilen, die teilweise in nur einer Sekunde berechnet werden.

Dies ist zwar kein Grund vor seinem PC auf die Knie zu fallen, aber es mag doch recht gut veranschaulichen, welche Rechenleistung selbst in „langsamen" Computern heutzutage steckt.

Was am obigen Modell nicht stimmt

Die Bildebene vor der Kamera ist eine Hilfskonstruktion um zu verstehen, wie die Lage der einzelnen Abtaststrahlen berechnet werden kann. Sie liegt natürlich nicht zwingend in der x-y-Ebene, sondern bewegt sich mit der Kamera mit, was an dem obigen Rechenbeispiel allerdings nur die Koordinaten ändern würde.

Was auf den ersten Blick jedoch verwundern mag ist die Tatsache, dass auch Objekte, die sich zwischen Kamera und Bildebene befinden, später in der Grafik erscheinen. Dies liegt an der Tatsache, dass einer der Abtaststrahlen natürlich auch schon vor der Bildebene auf ein Objekt treffen kann, was in den obigen Berechnungen bis auf die negative z-Koordinate keinen Unterschied macht.

Dies ist auch durchaus so gewollt, da sonst aus dem berechneten Bild alle Teile, die näher als die Bildebene vor der Kamera liegen herausgeschnitten würden.