

# **J. W. Goethe-Universität Frankfurt**

## **Seminar Technik virtueller Realitäten WS95**

### **Thema : Raytracing**

**Stefan Steinhaus**

<b>Inhalt</b>	<b>Seite</b>
<b>1. Einleitung</b>	<b>2</b>
<b>2. Grundlagen</b>	<b>2 - 10</b>
<b>2.1 Einfachste Modelle</b>	<b>2 - 7</b>
2.1.1. Mathematische Grundlagen	2 - 5
2.1.2. Lichtbrechung an Kugeln	5 - 6
2.1.3. Lichtbrechung an Parallelogrammen	6 - 7
<b>2.2 Reflexionseffekte</b>	<b>7 - 9</b>
2.2.1. Diffuse Reflexion	8
2.2.2. Spiegelnde Reflexion	9
<b>2.3 Schattenbildung</b>	<b>9</b>
<b>2.4 Modelle mit speziellen Lichtquellen</b>	<b>10</b>
<b>3. Spezielle Funktionen</b>	<b>10 - 13</b>
<b>3.1 Transparente Objekte</b>	<b>10 - 11</b>
3.1.1. Objekte ohne Lichtbrechung	10
3.1.2. Objekte mit Lichtbrechung	10 - 11
<b>3.2 Texturen</b>	<b>11 - 13</b>
<b>3.3 Ausblick auf nicht genannte Funktionen</b>	<b>12 - 13</b>
<b>4. Anwendungen in der Praxis</b>	<b>13</b>
<b>5. Probleme und Nachteile von Raytracing</b>	<b>13 - 14</b>
<b>5.1 Reflexion und Streuung</b>	<b>13 - 14</b>
<b>5.2 Rechenaufwand</b>	<b>14</b>
<b>5.3 Endlos Spiegelung</b>	<b>14</b>
<b>Anhang A : Raytracing Programme</b>	<b>15</b>
<b>Anhang B : Literatur</b>	<b>15</b>

## **1. Einleitung**

Zur Darstellung von Grafiken auf Computern gibt es heute eine Anzahl an verschiedenen mehr oder weniger guten Verfahren. Das älteste und wohl auch noch am weitesten verbreitete Verfahren ist die direkte Projektion von Körpern und Objekten jeglicher Art auf dem Bildschirm. Dieses Verfahren läßt sich sehr leicht durchführen und benötigt nur eine geringe Anzahl an Berechnungen, birgt aber auf der anderen Seite eine Menge von Problemen. Bei der einfachen Projektion von Objekten werden z.B. spezielle Routinen zur Lösung des "Hidden-Line"-Problems benötigt, Überlappungen von verschiedenen Objekten im Raum müssen speziell behandelt werden, Schattenbildung bzw. Lichtquellen, Spiegelbilder und verzerrte Bilder durch transparente Objekte werden nicht berücksichtigt und vieles mehr.

An diesem Punkt kommt die Methode des "Raytracing" zum Tragen. "Raytracing" heißt soviel wie "Strahlenverfolgung" was die Methode schon ziemlich passend beschreibt. In der Natur wird ein Objekt dadurch für einen Betrachter sichtbar, daß darauf auftreffende Lichtstrahlen gebrochen und in einem bestimmten Winkel reflektiert werden, von denen einige Lichtstrahlen durch das menschliche Auge erfaßt werden und somit unser Bild von der Umwelt formen. Objekte, die keine Lichtstrahlen reflektieren sind für das menschliche Auge nicht sichtbar. Eine Mauer z.B. reflektiert die Lichtstrahlen sehr gut und ist somit eindeutig sichtbar, eine Glasscheibe (selbstverständlich gut geputzt) allerdings ist "fast" unsichtbar, da Sie das Licht so gut wie überhaupt nicht oder nur in einem bestimmte Einfallswinkel reflektiert.

Diese physikalische Grundlage macht sich die Raytracingmethode in umgekehrter Reihenfolge zu Nutze. Der Raytracing-Algorithmus geht vom Auge des Betrachters aus und berechnet die Herkunft der eintreffenden Lichtstrahlen. Kommt ein eintreffender Lichtstrahl von einem Objekt, so wird der Reflexionspunkt des Objektes durch dessen Farbe markiert; kommt ein Lichtstrahl direkt aus dem Hintergrund, so nimmt dieser die Farbe des Hintergrundes an.

## **2. Grundlagen**

In diesem Kapitel möchte ich kurz die Grundlagen zur Rückverfolgung von Lichtstrahlen, die Kollisionsberechnung und einfache Effekte von Lichtstrahlen ansprechen. Mit Hilfe dieser ersten Grundlagen lassen sich einfache Bildberechnungen bereits sehr effektiv erstellen.

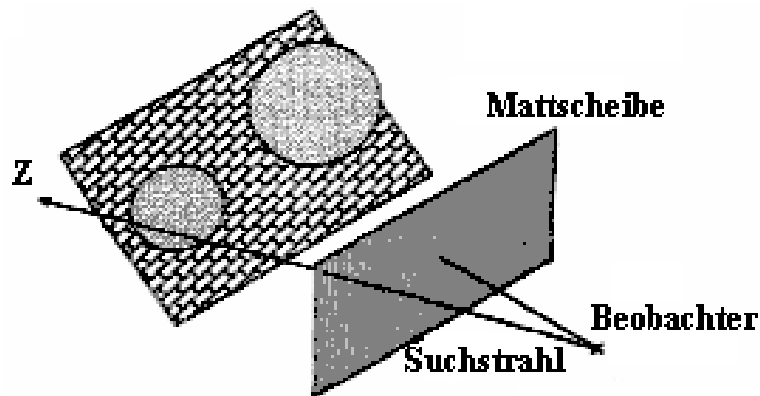
### **2.1. Einfachste Modelle**

Unter einfachsten Modellen versteht man Bilder die nur durch die einfache Zurückverfolgung und der Berechnung von Kollisionen erstellt werden und keinerlei naturverbundene Effekte wie Spiegelungen oder Transparenz berücksichtigen. Mit Hilfe dieser sehr grundlegenden Eigenschaften werden im Allgemeinen nur sehr monotone Bilder bzw. nicht sehr reale Bilder erstellt, die aber in der praktischen Anwendung durch seine schnelle Berechenbarkeit oft als eine Art Preview zur danach folgenden ausführlichen Modellberechnung dienen.

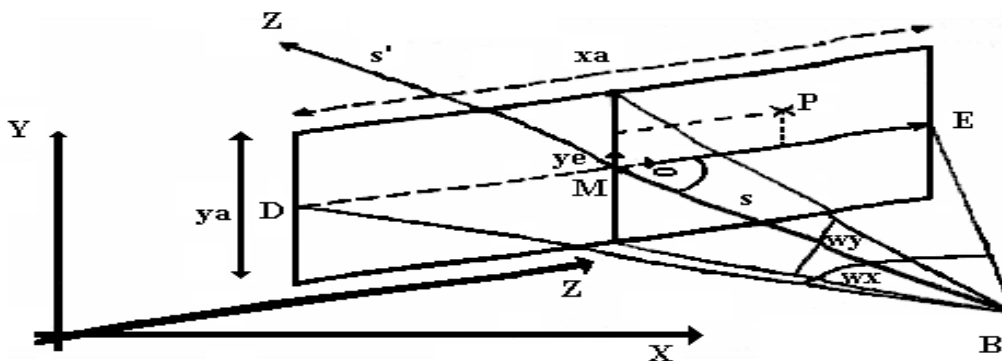
#### **2.1.1. Mathematische Grundlagen**

Gehen wir zuerst einmal von einem sehr einfachen Modell aus, bei dem alle Effekte weggelassen werden und wir nur die Strahlverfolgung berechnen. Unser Modell besteht (bildlich gesehen) aus zwei

Welten, die durch eine Scheibe getrennt sind (in der Praxis stellt dies der Com



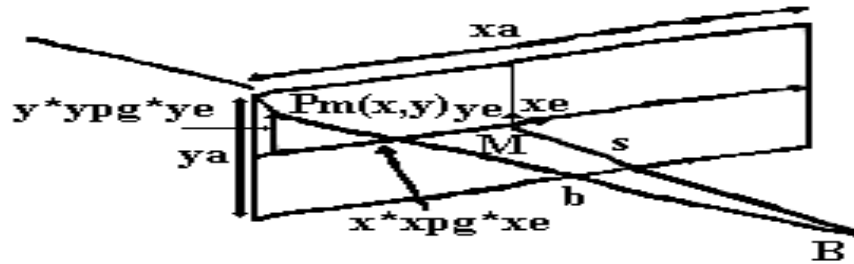
putermonitor dar) und zwar die Welt der Objekte und die Welt des Beobachters. Wir stellen uns die Position des Beobachters als fixen Punkt B vor und ziehen von dort aus nun eine orthogonale Linie durch den Punkt M auf der Scheibe zu einem "imaginären" Punkt Z in der Welt der Objekte.



Die Strecke  $\vec{s}$  von B nach M, der sogenannte Sichtvektor oder auch Brennweite des Objektes, nehmen wir für den Anfang mit Wert 1 an. Er kann allerdings beliebige Werte  $> 0$  annehmen. Die Strecke  $\vec{s}'$  von B nach Z wird verlängerter Sichtvektor genannt. Da uns zu  $\vec{s}$  bis jetzt nur die Länge bekannt ist und nicht die Richtung, werden wir diese mit Hilfe von  $\vec{s}'$  jetzt wie folgt herleiten :

$$\vec{s} = n * \vec{s}' \Leftrightarrow \vec{s} = n * (Z - B) \text{ wegen } |\vec{s}| = n * |Z - B| \Leftrightarrow \vec{s} = \frac{|\vec{s}|}{|Z - B|} * (Z - B)$$

Da wir für jeden Punkt auf der Scheibe (bzw. Pixel auf dem Monitor) jeweils einen eintreffenden Lichtstrahl berechnen wollen, benötigen wir jetzt noch eine Umrechnung der Punktkoordinaten auf der Scheibe in die entsprechenden Vektorkoordinaten. Diese erlangen wir vom Punkt B aus durch Verschiebung um  $\vec{s}$  und einer Verschiebung in X- und Y-Richtung auf der Scheibe. Die Verschiebungen in X- und Y-Richtung sind die für uns noch unbekannt Faktoren. Sie setzen sich aus den Punktkoordinaten, den Einheitsvektoren  $\vec{x}_e$  und  $\vec{y}_e$  und dem Faktor der Punktgröße xpg und ypg zusammen. Die Einheitsvektoren beschreiben das Verhältnis zwischen den Punktkoordinaten und den Vektorkoordinaten.



Die Punktgrößen xpg und ypg kompensieren die unterschiedliche Bildgrößen in X- und Y-Richtung, die eventuelle Bildverzerrungen ergeben können. Ein beliebiger Punkt P auf der Scheibe errechnet sich dann aus folgender Formel :

$$P = B + \vec{s} + (x * xpg) * \vec{xē} + (y * ypg) * \vec{yē}$$

Noch unbekannt sind bis jetzt die Vektoren  $\vec{xē}$  und  $\vec{yē}$  sowie die Konstanten xpg und ypg. Die Konstanten ypg bzw. xpg sind leicht zu berechnen, da Sie nur einen Verhältnisfaktor für die Verzerrung des Bildes in X- oder Y-Richtung darstellen. Ist die maximale Anzahl der zu berechnenden Punkte auf der X-Ebene nicht gleich der maximalen Anzahl der zu berechnenden Punkte auf der Y-Ebene dann wird ypg bzw. xpg als Verhältnis zwischen max. Punktzahl auf der X-Ebene und der Y-Ebene angenommen (Typische Auflösungen sind 1280\*1024, 1024\*768, 800\*600...). Die maximale Punktauflösung werde ich von nun ab  $x_{max}$  bzw.  $y_{max}$  nennen.

Zu Berechnen sind nun noch die Einheitsvektoren  $\vec{xē}$  und  $\vec{yē}$ . Bekannte Eigenschaften für  $\vec{xē}$  sind daß der Sichtvektor  $\vec{s}$  senkrecht zu diesem steht und somit gilt

$$\vec{s} * \vec{xē} = 0 \Leftrightarrow s_x * x_{e_x} + s_y * x_{e_y} + s_z * x_{e_z} = 0$$

und wir wissen die Strecke zwischen dem Mittelpunkt M und dem Bildschirmrand :

$$ME = \frac{x_{max}}{2} * xpg * |\vec{xē}| \Leftrightarrow |\vec{xē}| = \frac{2 * ME}{x_{max} * xpg}$$

In dieser Gleichung befindet sich noch ein unbekannter Faktor, nämlich die Strecke ME. Wir können diese leicht berechnen, indem wir die Eigenschaften des Tangens zu Nutze machen. Die Strecken SM, ME und SE bilden ein rechtwinkliges Dreieck deren Blickwinkel  $\frac{w_x}{2}$  zwischen SM und SE gegeben ist. Daher ergibt sich für ME folgende Formel :

$$ME = \tan\left(\frac{w_x}{2}\right) * |\vec{s}|$$

Eingesetzt ergibt sich somit für  $|\vec{xē}|$  die Formel :

$$|\vec{xē}| = \frac{2 * \tan\left(\frac{w_x}{2}\right) * |\vec{s}|}{x_{max} * xpg}$$

Die Länge  $|\vec{yē}|$  des Einheitsvektors  $\vec{yē}$  wird auf dieselbe Art und Weise berechnet so daß gilt:

$$|\vec{xē}| = |\vec{yē}|$$

Da wir unseren Bildschirm auf der xz-Ebene gerade stehen haben, können wir für die y-Dimension  $x_{e_y} = 0$  setzen. Es gilt somit  $s_x * x_{e_x} + s_z * x_{e_z} = 0$ .

Wir unterscheiden 3 Fälle für s :

- ◆ Ist  $s_x \neq 0$  so löst sich unsere Formel nach  $x_{e_x}$  zu  $x_{e_x} = -\frac{s_z * x_{e_z}}{s_x}$  auf.
- ◆ Ist  $s_x = 0$  und  $s_z \neq 0$  so können wir nur nach  $x_{e_z}$  zu  $x_{e_z} = -\frac{s_x * x_{e_x}}{s_z}$  auflösen.
- ◆ Der Fall daß  $s_x = 0$  und  $s_z = 0$  müssen wir nicht weiter betrachten, da der Blickvektor in diesem Fall parallel zum Bildschirm verläuft.

Durch Einsetzen und Umformen in die Gleichung  $|\vec{x}\vec{e}|^2 = xe_x^2 + xe_z^2$  erhalten wir

$$xe_z = \sqrt{\frac{|\vec{x}\vec{e}|^2}{1 + \frac{sx}{sz}^2}} \quad \text{und} \quad xe_x = \sqrt{\frac{|\vec{x}\vec{e}|^2}{1 + \frac{sx}{sz}^2}}$$

bzw. entsprechendes für  $|\vec{y}\vec{e}|$ .

Aus dem bereits Bekannten erhalten wir nun eine Geradengleichung die einen Lichtstrahl vom Beobachter über einen beliebigen Punkt auf dem Bildschirm bis hin zu einem Objekt oder in den Hintergrund zurückverfolgt. Der beliebige Punkt auf dem Bildschirm errechnet sich aus bereits Bekannten wie folgt:

$$P_m = B + \vec{s} + x * ypg * \vec{x}\vec{e} + y * ypg * \vec{y}\vec{e}$$

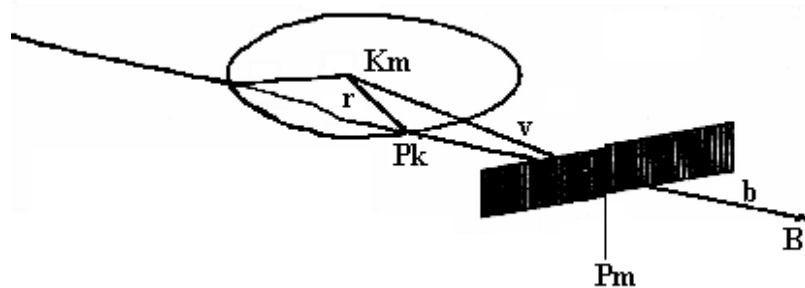
Die Geradengleichung die vom Punkt B ausgeht und den Punkt P<sub>k</sub> kreuzt ergibt sich dann wie folgt :

$$P = P_m + k * (P_m - B)$$

Den Vektor P<sub>m</sub>-B werde ich zukünftig nur noch als Blickvektor  $\vec{b}$  bezeichnen.

### 2.1.2. Lichtbrechung an Kugeln

Kugeln sind die wohl am häufigsten verwendeten Objekte für Raytracing, da die Lichtbrechung für diese Objekte noch am einfachsten berechnet werden kann. Gegebene Daten für dieses Objekt sind ein Mittelpunkt K<sub>m</sub>, der Radius R und selbstverständlich die Farbe des Objektes. Wir vermuten, daß die Gerade des Lichtstrahles die Kugel an einem Punkt P<sub>k</sub> schneidet und können damit die Formel des Radiusvektors aufstellen :



$$\vec{r} = P_k - K_m$$

Zur Erinnerung : Unsere Geradengleichung für den Lichtstrahl lautet :

$$P_k = P_m + k * \vec{b}$$

Daraus ergibt sich eingesetzt folgende Formel :

$$\vec{r} = P_k - K_m \Leftrightarrow \vec{r} = P_m + k * \vec{b} - K_m$$

Wir müssen jetzt den Verlängerungsfaktor k berechnen so daß  $|\vec{r}|^2 = R^2$  gilt.

$$\left| (P_m - K_m) + k * \vec{b} \right|^2 = R^2$$

Setzen wir den Vektor P<sub>m</sub>-K<sub>m</sub> gleich  $\vec{v}$  so können wir unsere Gleichung gut auf eine Gleichung vereinfachen die mit einer PQ-Formel lösbar ist.

$$\left| \vec{v} + k * \vec{b} \right|^2 = R^2 \Leftrightarrow \vec{v}^2 + 2 * k * \vec{v} * \vec{b} + k^2 * \vec{b}^2 = R^2$$

$$\Leftrightarrow \left( \vec{b}^2 \right) * k^2 + \left( 2 * \vec{v} * \vec{b} \right) * k + \left( \vec{v}^2 - R^2 \right) = 0$$

Wir setzen

$$a = \vec{b}^2 \quad b = 2 * \vec{v} * \vec{b} \quad c = \vec{v}^2 - R^2$$

und erhalten eine einfache quadratische Gleichung

$$a * k^2 + b * k + c = 0$$

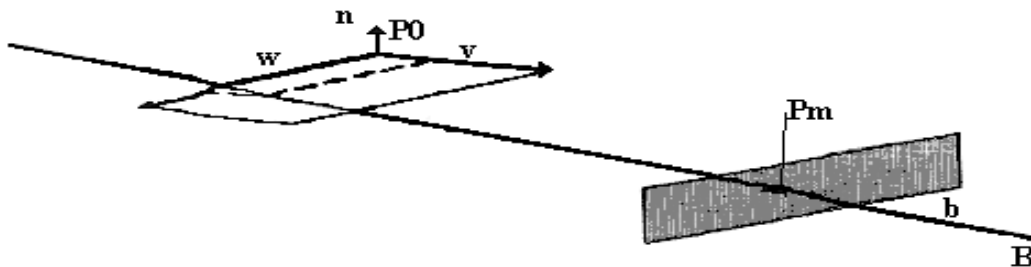
die wir mit einer PQ-Formel auflösen.

$$k_{1,2} = -\frac{b}{2*a} \pm \sqrt{b^2 - 4 * a * c} \quad D = b^2 - 4 * a * c$$

Ist die "Diskriminante"  $D < 0$  so schneidet die Lichtstrahlgerade die Kugel nicht, ist die Diskriminante  $D \geq 0$  so schneidet die Lichtstrahlgerade die Kugel in mindestens einen Punkt. Schneidet die Lichtstrahlgerade die Kugel, so wird der Ausgangspunkt  $P_m$  auf dem Bildschirm in der Farbe der Kugel gesetzt. Gibt es mehrere Objekte, die dieser Lichtstrahl trifft, so wird der Punkt  $P_m$  in der Farbe des Objektes gesetzt, welches sich am nächsten zum Bildschirm befindet. Das nächste Objekt ist immer das Objekt, welches das kleinste  $k$  aus obiger Formel ergibt.

### 2.1.3. Lichtbrechung an Parallelogrammen

Die Lichtbrechung an einem Parallelogramm wird ähnlich der an einer Kugel berechnet und ist ebenfalls noch relativ leicht zu konstruieren.



Wir vermuten einen Schnittpunkt der Lichtstrahlgerade mit dem Parallelogramm im Punkt  $P_s$  und erhalten folgende Punktgleichung :

$$P_s = P_0 + k * \vec{v} + m * \vec{w} \text{ bzw. } P_s = P_m + s * \vec{b}$$

Nun wenden wir anstatt der Ebenengleichung die Normalen-Gleichung an, die für unser Modell  $(P_s - P_0) * \vec{n} = 0$  lautet.

Die Normale  $\vec{n}$  ist gegeben, da sich  $\vec{n}$  auch als  $\vec{n} = \vec{v} \times \vec{w}$  schreiben lässt und die Vektoren  $\vec{v}$  und  $\vec{w}$  sind gegeben.

Setzen wir in die Normalengleichung die Geradengleichung ein so erhalten wir

$$(P_m + s * \vec{b} - P_0) * \vec{n} = 0 \Leftrightarrow (P_m - P_0) * \vec{n} + s * \vec{b} * \vec{n} = 0$$

$$\Leftrightarrow s * \vec{b} * \vec{n} = (P_0 - P_m) * \vec{n}$$

Hierbei müssen wir 2 Fälle unterscheiden, nämlich wenn  $\vec{b} * \vec{n} = 0$  ist, also der Blickvektor senkrecht auf der Normale liegt, somit also entweder unendlich viele Schnittpunkte existieren oder die Lichtstrahl genau parallel zur Ebene verläuft und dem Fall  $\vec{b} * \vec{n} \neq 0$ , in diesem gibt es genau einen Schnittpunkt.

Für letzteren Fall ergibt sich für  $s$  folgende umgestellte Formel :

$$s = \frac{(P_0 - P_m) * \vec{n}}{\vec{b} * \vec{n}}$$

Wir erinnern uns an die am Anfang definierte Gleichung  $P_s = P_0 + k * \vec{v} + m * \vec{w}$ . Sollten die Vektoren  $\vec{v}$  und  $\vec{w}$  linear unabhängig sein, so ergibt sich für  $m$  und  $k$  folgende Auflösungen :

$$m = \frac{(p_{sj} - p_{0j}) * v_i - (p_{si} - p_{0i}) * v_j}{w_j * v_i - w_i * v_j}$$

$$k = \frac{(p_{si} - p_{0i} - m * w_i)}{v_i}$$

$p_{si}$ ,  $p_{sj}$  ist aus  $P_s$  und  $p_{0i}$ ,  $p_{0j}$  ist aus  $P_0$  für die gilt  $w_j * v_i - w_i * v_j \neq 0 \wedge v_i \neq 0$  mit  $i, j \in \{x, y, z\}$ .

## 2.2. Reflexionseffekte

Die bis jetzt betrachteten Rechnungen setzen für jeden Bildpunkt einen Idealzustand voraus der keinerlei Einflüssen unterliegt und nur eine diffuse Hintergrundbeleuchtung besitzt. Dies ist in der Praxis aber nicht der Fall. Eine besonders wichtige Rolle für die Farbsetzung und die Lichtstrahlverfolgung spielt vor allem die Reflexion am Objekt und die Entfernung. Ist ein Objekt weit entfernt, so muß man davon ausgehen daß sich die Lichtintensität verringert. Man muß auch zwischen den verschiedenen Arten der Reflexion des Lichtstrahles auf dem Objekt unterscheiden. Wir unterscheiden vor allen zwischen der diffusen Reflexion und der spiegelnden Reflexion. Nach den Gesetzen der Physik sollte ein Lichtstrahl in dem selben Winkel reflektiert werden, wie er auf dem Objekt eintrifft. In der Praxis müssen wir hier aber vor allem das Material des Objektes berücksichtigen, welches eine unterschiedliche Reflexion hervorrufen kann (z.B. reflektiert eine Stoffkugel einen Lichtstrahl völlig anders als eine polierte Stahlkugel).

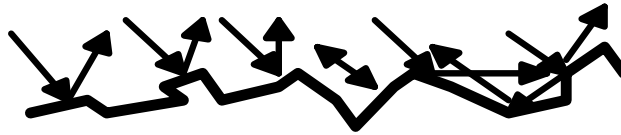


*(Abbildung mit spiegelnder und diffuser Reflexion)*

### 2.2.1. Diffuse Reflexion

Wir sprechen von einer sogenannten diffusen Reflexion (oder auch Lambertreflexion) wenn ein Lichtstrahl auf Objekte trifft, die eine matte Oberfläche haben. In diesem Fall wird ein Lichtstrahl nicht nur in seinem Einfallswinkel reflektiert sondern oftmals auch gesplittet und mehrere Lichtstrahlen verlassen dann das Objekt.

Beispiel für eine mögliche Reflexion an rauhen Oberflächen



Je matter die Oberfläche des Objektes ist um so breiter wird der reflektierte Lichtstrahl gefächert und um so mehr nimmt die Intensität des Lichtstrahles ab. Die Intensität des reflektierten Lichtstrahles wird hierbei maßgeblich von der Reflexionskonstanten  $K_{od}$  des Objektes ( $0 \leq K_{od} \leq 1$ ) und der Intensität  $I_q$  der Lichtquelle bestimmt. Haben wir außerdem noch eine diffuse Hintergrundbeleuchtung, dann fällt in unsere Berechnung zudem der Faktor der Hintergrundreflexion  $K_{oh}$  und die Intensität  $I_h$  der Hintergrundbeleuchtung an. Der Einfallswinkel des Lichtstrahles (Einfallswinkel wird immer zum Lot des Objektes angegeben) auf dem Objekt erwirkt zudem den Effekt der Fächerung und somit der Lichtdämmung. Fällt der ankommende Lichtstrahl senkrecht auf das Objekt, so kann theoretisch die optimalste Reflexion, die Totalreflexion eintreten (0 Grad). Fällt der Lichtstrahl senkrecht zum Lot, so wird kein Licht reflektiert. Man kann sich diesen Fall praktisch anschauen, wenn man einen gebündelten Lichtstrahl auf eine transparente Glaskugel lenkt. Im optimalsten Fall bricht sich der Lichtstrahl nur ins Kugellinnere bzw. innerhalb der Oberfläche weg, es gibt keinen reflektierenden Lichtstrahl.

Zur Berechnung der Intensität des austretenden Lichtstrahls ergibt sich somit folgende Formel:

$$I = I_h * K_{oh} + I_q * K_{od} * \cos(\alpha)$$

Der Faktor des Hintergrundleuchtens ersetzt in unserem Fall effizient die Nachrechnung jedes einzelnen gefächerten Lichtstrahles von allen Objekten des Modell. Wir gehen statt dessen von einem allgemeinen diffusen Hintergrundleuchtens aus, welches uns nur einen einfachen Faktor in die Rechnung bringt und nicht einen riesigen Wust von neuen Berechnungen.

In der jetzigen Formel vernachlässigen wir allerdings immer noch einen kleinen Effekt den wir schon erwähnt haben, nämlich die Entfernung. Gemäß den Gesetzen der Physik nimmt die Intensität eines Lichtstrahles proportional zum Quadrat der Entfernung ab. Es gilt also

$$I' = \frac{I}{d^2}$$

und es folgt

$$I = I_h * K_{oh} + \frac{I_q * K_{od} * \cos(\alpha)}{d^2}$$

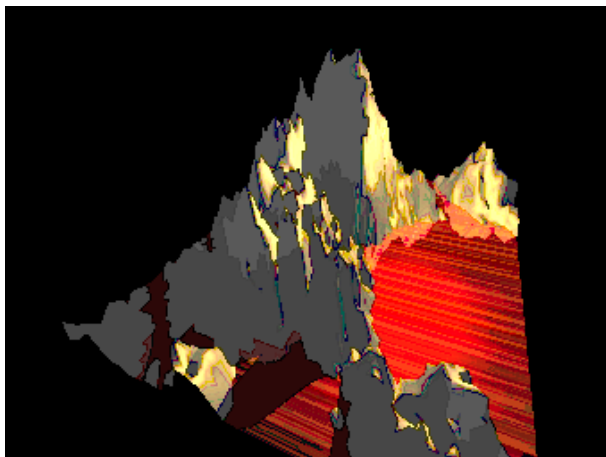


Abbildung einer Bergkette deren Abbild vom Wasser diffus reflektiert wird.



### 2.2.2. Spiegelnde Reflexion

Die spiegelnde Reflexion tritt ein, wenn ein Objekt eine spiegelnde Oberfläche besitzt wie z.B. Metall (poliert). Zu beachten ist allerdings das eine "exakte" Spiegelung nur in dem Falle eintritt wenn der Lichtstrahl genau senkrecht auf das Objekt trifft, also genau im Lot steht. Trifft der Lichtstrahl nicht genau im Lot auf das Objekt, so ist der Einfallswinkel gleich dem Ausfallswinkel. Allerdings kann der Ausfallswinkel um einen gewissen Winkel  $\vec{b}$  variieren. Die Materialart des Objektes charakterisiert maßgeblich die Varianz des Ausfallswinkels und teilweise sogar die Eigenschaften des Lichtes. So kann z.B. der Rotanteil eines Lichtstrahles in einem anderen Winkel reflektiert werden als z.B. der Grünanteil. Sind keine Angaben über Materialarten vorhanden so kann man von einem idealisierten Modell ausgehen und mit Einfallswinkel = Ausfallswinkel rechnen. Für weitere Erklärungen gehe ich jetzt erst einmal davon aus daß die Ablenkkorrekturfunktion  $r(a,l)$ , die abhängig vom Einfallswinkel und der Wellenlänge des Lichtstrahles ist, gegeben sei.

Einen weiteren Faktor den wir ebenfalls beachten müssen ist die sogenannte "Fokussierung". Die Fokussierung  $f$  stellt eine Materialkonstante dar, die ausdrückt in welcher Intensität ein Lichtstrahl reflektiert wird.

Daraus ergibt sich eine ähnliche Formel zur Berechnung der Intensität des Lichtstrahles wie bei der diffusen Reflexion nämlich :

$$I_s = I_q * r(a, l) * \cos(b)^f$$

Bzw. unter Berücksichtigung der Entfernung und der Hintergrundstrahlung ergibt sich:

$$I = I_h * K_{oh} + \frac{I_q}{d^2} * (K_{od} * \cos(a) + r(a, l) * \cos(b)^f)$$

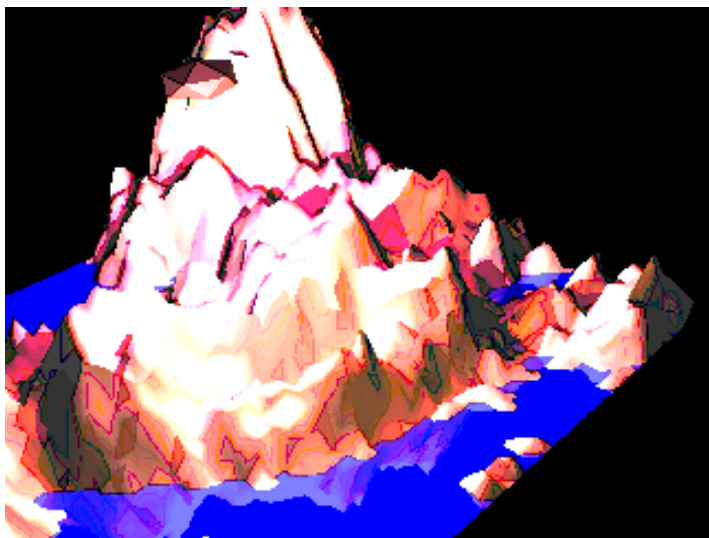


Abbildung einer Bergkette deren Abbild vom Wasser spiegelnd reflektiert wird.

### 2.3. Schattenbildung

Das Problem des Schattenwurfes ist dem Problem der verdeckten Linien (Hidden-Line Problem) ziemlich ähnlich, da es auch hierbei um die Verdeckung eines Objektes, in diesem Fall die Lichtquelle, geht. Dadurch, das ein Objekt vor der Lichtquelle steht, kann ein berechneter Punkt eventuell nicht oder nur in geringer Lichtintensität gesehen werden. Dies behandeln wir ganz einfach, indem wir von dem berechneten Punkt des Objektes eine direkte Linie zu allen definierten Lichtquellen ziehen und sehen, ob wir dabei ein weiteres Objekt scheiden. Ist dies der Fall so müssen wir die Lichtintensität unseres Punktes entsprechend herunterrechnen (die diffuse Hintergrundbeleuchtung beleuchtet den Punkt noch). Sind mehrere Lichtquellen vorhanden, so wenden wir dieses Verfahren für jede einzelne Lichtquelle separat an. Die Lichtintensität unseres Punktes ergibt sich dann aus der Summe aller Lichtintensitäten der einfallenden Strahlen.

## 2.4. Modelle mit speziellen Lichtquellen

Unter speziellen Lichtquellen können wir uns Lichtquellen vorstellen, die einen besonderen Effekt auf unser Berechnungsmodell haben. Hierzu gehören vor allen Lichtquellen, die kein weißes Licht ausstrahlen, sondern "farbiges" Licht. Tritt ein solcher Fall ein, sind wir gezwungen unser Modell der Intensitätsberechnung leicht zu erweitern. Wir verwenden hierzu das RGB-Modell und verwenden 3 Intensitätsvariablen. Eine Variable für den Rotanteil, eine für den Grünanteil und eine für den Blauanteil. Strahlt jetzt eine Lichtquelle nur rotes Licht aus, dann sind die Intensitätsvariablen der anderen Farbanteile entsprechend auf Null gesetzt. Bei Farbmischungen werden die Intensitätsvariablen gemäß des Anteils im Licht gesetzt. Weißes Licht enthält gemäß diesem Modell 100% Farbanteil von allen 3 Farben.

## 3. Spezielle Funktionen

### 3.1. Transparente Objekte

Objekte die aus einem transparenten Material bestehen, wie z.B. Glas, müssen besonders behandelt werden, da hierbei keine Verdeckung von dahinterliegenden Objekten stattfindet, allerdings die Lichtstrahlen geschwächt oder aber verzerrt werden können.

#### 3.1.1. Objekte ohne Lichtbrechung

Transparente Objekte, die einen einfallenden Lichtstrahl nicht brechen, sind ein sehr idealisierter Fall und kommen in der Natur so sicherlich nur selten vor. Mit Hilfe des Transparenzfaktors  $k$  des gegebenen Materials ergibt sich die Formel für die neue Lichtintensität :

$$I = k * I_v$$

Ist  $k = 0$  so spricht man von einem undurchsichtigen Objekt, ist  $k = 1$  dann spricht man von einem perfekten transparenten Objekte (blickt man senkrecht auf dieses Objekt so ist es nicht sichtbar).

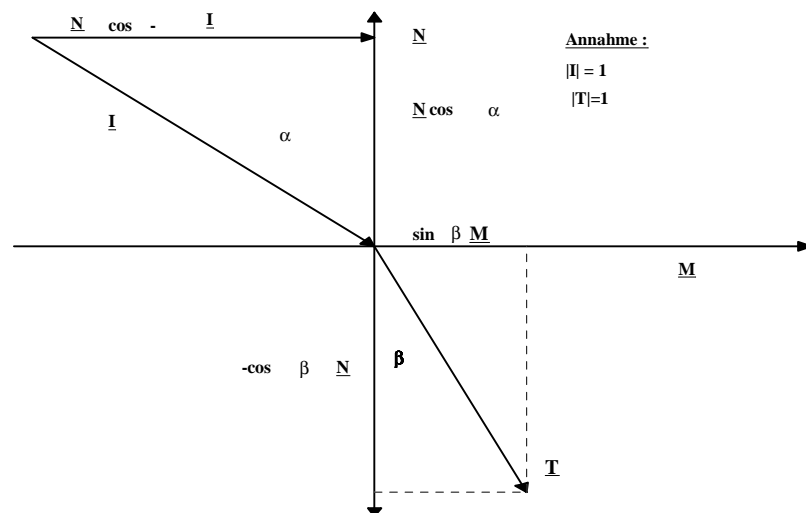
#### 3.1.2. Objekte mit Lichtbrechung

Viele transparente Objekte lassen einen einfallenden Lichtstrahl nicht einfach gerade durchgehen sondern lenken diesen in einen bestimmten Winkel um.

Der Brechungswinkel errechnet sich, wie man in jeder physikalischen Formelsammlung nachlesen kann, wie folgt :

$$\frac{n_b}{n_a} = \frac{\sin \alpha}{\sin \beta}$$

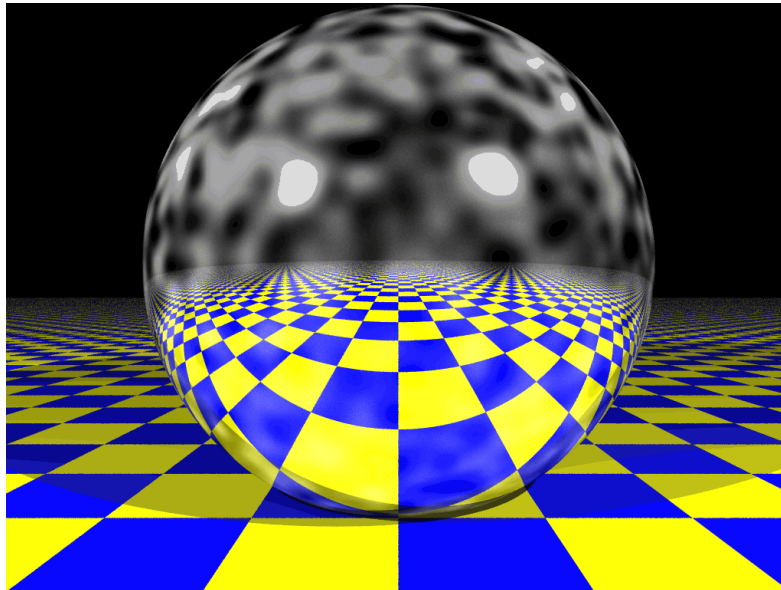
( $n_a, n_b$  - Brechungszahl von Medium a und b,  $\alpha$  - Einfallswinkel,  $\beta$  - Ausfallswinkel)



Der ausfallende Lichtstrahl berechnet sich dann wie folgt :

$$\begin{aligned} \vec{T} &= \sin(\alpha) * \vec{M} - \cos(\beta) * \vec{N} \Leftrightarrow \vec{T} = \frac{\sin\beta}{\sin\alpha} * (\vec{N} \cos\alpha - \vec{I}) - \cos(\beta) * \vec{N} \\ \Leftrightarrow \vec{T} &= \left(\frac{n_a}{n_b} * \cos\alpha - \cos\beta\right) * \vec{N} - \frac{n_a}{n_b} * \vec{I} \\ \Leftrightarrow \vec{T} &= \left(\frac{n_a}{n_b} * (\vec{N} * \vec{I}) - \sqrt{1 - \left(\frac{n_a}{n_b}\right)^2 * \left(1 - (\vec{N} * \vec{I})^2\right)}\right) * \vec{N} - \frac{n_a}{n_b} * \vec{I} \end{aligned}$$

Die Berechnung der Lichtintensität ist dieselbe wie die für transparente Objekte ohne Lichtbrechung. Zu beachten ist allerdings noch, daß wenn diese transparenten Objekte eingefärbt sind, Sie die Lichtintensität in bestimmten Farbbereichen maßgeblich beeinflussen bzw. gewisse Lichtspektren überhaupt nicht durchlassen.



(Abbildung einer transparenten Kugel mit Lichtbrechung und Spiegeleffekten)

### **3.2. Texturen**

Für viele Grafikmodelle, im Speziellen in Anwendung für Filme oder für Architekten, ist es nicht ausreichend, einfarbige Objekte für die Modelle einzusetzen. Sollen also Objektmodelle berechnet werden in denen mehrfarbige Objekte oder Objekte mit Bildoberflächen vorkommen, dann werden sogenannten Texturen (engl. texture mapping) verwendet. Die Methode funktioniert wie folgt. Das Bild, das auf die Oberfläche eines Objektes gebracht werden soll, wird logisch auf einem "Texture map" mit den Ausmaßen (u,v) gebracht. Durch Koordinatentransformation wird jeder Punkt (x,y) aus dem "Texture map" auf einem Punkt auf dem Objekt transformiert. Da die Ausmaße des Objektes nicht unbedingt mit denen des "Texture map" übereinstimmen müssen, kann es vorkommen das ein Punkt des "Texture map" mehreren Punkten bzw. einem Bereich auf dem Objekt zugeordnet wird oder auch umgekehrt. Dieses Verfahren ist zwar noch eines der Einfachsten, für die meisten Objekte ist die Anwendung aber aus Ermangelung passender Transformationsfunktionen bereits sehr schwierig bis nahezu unmöglich. Einfachste Objekte sind dabei noch Körper, die auf planaren Vierecken basieren (Würfel, Ebenen, Zylinder über Mantelfläche,...). Bei der Berechnung der Punktsetzung auf dem Bildschirm muß nun für ein solches Objekt bei jedem Oberflächenpunkt überprüft werden, welche Farbe dieser trägt. Genügt es normalerweise, eine Farbdefinition für ein Objekt anzugeben, so müssen diesmal eine endliche Anzahl an Farbdefinitionen gespeichert werden.



Abbildung mit aufgebracht  
Holzmasse und  
Etikettierung der Dose

### **3.3. Ausblick auf nicht genannte Funktionen**

Zum Raytracing gehören eine Menge an verschiedenen Methoden und Modellen die hier sicherlich nicht alle erwähnt werden können. Ich möchte zur Vervollständigung aber dennoch einige Methoden stichwortartig erwähnen.

#### Das Phong Helligkeitsmodell :

Das Helligkeitsmodell nach Phong Bui-Tong ist ein Berechnungsmodell für die Intensität von Reflexionen an sogenannten "Nichtperfekten Reflektoren" und geht von einem Verringerungsfaktor der Lichtintensität von  $\cos^2 \alpha$  aus.  $\alpha$  ist der Abweichungswinkel vom vermuteten Ausfallwinkel und  $n$  ist ein Materialfaktor. Dieses Modell wird auf Grund seiner realitätsnähe besonders gern verwendet.

#### Beleuchtungsmodell nach Warn :

Lichtquellen leuchten in der Realität nicht in jede Richtung mit der gleichen Intensität. Nach dem Modell von Warn spielt bei einer Lichtquelle nicht nur die Position und die Lichtintensität eine Rolle sondern auch die Verteilung des Lichtaustrittes. Ein Laserstrahl z.B. ist eine stark gebündelte Lichtquelle, eine normale Deckenlampe hat schon eine sehr große Streuung und die Sonneneinstrahlung kann man als sehr diffus bezeichnen. Warn berechnet diesen Streuungseinfluß einer Lichtquelle durch einen gegebenen Austrittswinkel des Lichtes und eines perfekten Pseudoreflektors der diesen Effekt berechenbar macht.

#### Beleuchtungsmodell nach Hall :

Hall berücksichtigt bei seinem Beleuchtungsmodell nicht nur das Medium durch dem die Lichtstrahlen gehen, sondern er berücksichtigt auch die Mischung von Lichtstrahlen aus verschiedenen Lichtquellen und eine dadurch bedingte Veränderung der Lichtverhältnisse.

#### Raytracing nach Cone :

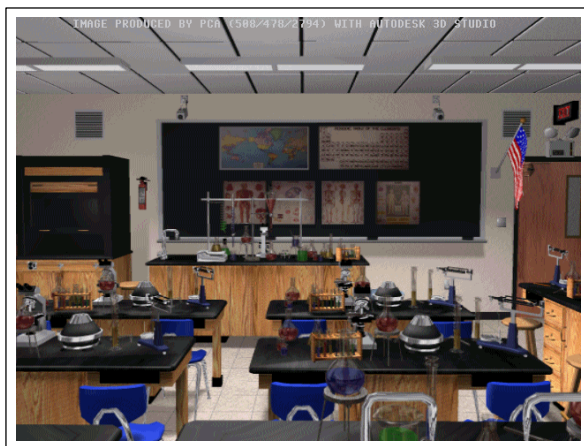
Cone verwendet für seine Raytracingmodelle zwar alle grundlegenden altbekannten Raytracingverfahren auch, aber er geht bei seinem Modell der Strahlenverfolgung nicht davon aus, daß ein Lichtstrahl gebündelt ist. Sein Lichtstrahl bildet einen Kegel der mit zunehmender Entfernung immer größere Bereiche ablichtet. Da Lichtstrahlen in der Realität normalerweise nicht scharf gebündelt vorkommen ist die Ausbreitung als Kegel, nach Cone, sehr realitätsnah aber auch wesentlich rechenintensiver da der Kegel durch Dutzende von einzelnen Lichtstrahlen repräsentiert werden muß.

### Raytracing nach Heckbert und Hanrahan :

Die Raytracingmethode nach Heckbert und Hanrahan (auch Beam tracing) geht wie die von Cone ebenfalls nicht davon aus, daß ein Lichtstrahl gebündelt ist. Sie geht davon aus, das sich ein Lichtstrahl mit zunehmender Entfernung pyramidenförmig ausbreitet. Dieses Verfahren bietet nahezu die Selben Vor- bzw. Nachteile wie das Modell nach Cone.

## **4. Anwendungen in der Praxis**

In der Praxis wird Raytracing meistens als Teil von "Renderprogrammen" verwendet. Unter Renderprogrammen versteht man Programme die mit Hilfe von Raytracing, Rayshading, Radiosity und vieler anderer Methoden einfache Körpermodelle zu reellen Gegenständen umwandeln, also naturgebundene Effekte einfließen lassen (z.B. können architektonische Modelle unter einem CAD-CAM Programm erstellt werden und dann mit bestimmten Lichtverhältnissen und Oberflächen zu reellen Räumen verwandelt werden die nahezu Photoqualität erreichen).



Renderprogramme werden zur Zeit vor allen in der Filmindustrie verwendet, um virtuelle Gegenstände und Personen an bestehende Lichtverhältnisse in Filmszenen einzupassen oder sogar ganze Filmszenen aus nichtrealen Gegenständen und Personen zu berechnen. Beispiele für solche Verwendungen sind die Filme Star Wars, Die unendliche Geschichte, Terminator 2, Deep Space 9, Abyss, Species,...

Architekten verwenden Renderprogramme um Konstruktionen zu visualisieren, um diese auf verschiedene Charakteristika zu überprüfen.

Softwareschmieden wie Origin verwenden Renderprogramme, um Szenen in Spieleprogrammen möglichst naturgetreu zu erstellen. So wurden z.B. bei dem Spiel "Wing Commander 3" jede einzelne Szene von Schauspielern vor einer nicht vorhandenen Kulisse gedreht um danach die Personen auf gezeichneten Hintergründen mit Hilfe von Renderprogrammen zu realen Szenen verschmelzen zu lassen.

## **5. Probleme und Nachteile von Raytracing**

### **5.1. Reflexion und Streuung**

In der Theorie des Raytracing gibt es vor allen immer das Problem, ein gerendertes Modell möglichst naturgerecht darstellen zu können. Hierbei spielen vor allen die Beleuchtungsmodelle eine sehr große Rolle, da diese entscheiden, wie die Objekte letztendlich aussehen werden. Problempunkte sind hierbei die Errechnung von Reflexionen an rauen Oberflächen, die das Licht in alle Richtungen streuen, die Abnahme der Lichtintensität in der Entfernung und eventuelle Effekte von Farbfilterung. Nicht zu vernachlässigen sind auch die Effekte von Lichtstreuung durch atmosphärische Effekte, da wir bis jetzt unsere Lichtstrahlverfolgung nur im Vakuum, also mit keinerlei Streuung betrachtet haben. Für die Lichtbrechung ist es zudem ziemlich schwierig die Reflexionen und Brechungen an komplexen bzw. mathematisch nicht exakt definierten Objekten zu berechnen. Deshalb werden die meisten Modellbe-

rechnungen zur auf Kugeln, planaren Flächen, Kegeln, Zylindern, Quadern und symmetrischen Mehre-  
cken aufgebaut.

## **5.2. Rechenaufwand**

In der Praxis wird das Verfahren des Raytracing in den vielfältigsten Bereichen zwar immer beliebter aber doch mit einigen Problemen behaftet. Das wohl eindeutig schwierigste Problem ist der immense Rechenaufwand der schon bei Modellen mit nur wenigen Objekten zu Buche schlägt. Gehen wir von einer Standardbildschirmauflösung von  $1024 \times 768$  aus und nehmen an das im Mittel jeder Lichtstrahl 3 mal reflektiert wird, ohne dabei spezielle Effekte wie transparente Körper zu berücksichtigen, so müssen wir bereits 2.359.296 Vektoren berechnen, hinter denen natürlich jeweils auch noch einmal eine Menge von Berechnungen stecken. Ein solches Modell wird bei den heutigen PCs in wenigen Minuten berechnet. Da aber Anwendungen in der Architektur oder der Filmindustrie weit bessere Auflösungen, wesentlich mehr Objekte und wesentlich mehr Effekte einsetzen vervielfacht sich die benötigten Rechenzeit so stark, daß einfache PCs Wochen bis Monate für solche Berechnungen benötigen würden. Es werden deshalb in diesem Bereich hauptsächlich Workstations und Großrechner verwendet (z.B. Wing Commander IV wird auf einem SGI Rechner entwickelt). Es besteht also das Problem die benötigte Rechenkapazität zur Verfügung zu stellen.

## **5.3. Endlos Spiegelung**

Ein weiteres Problem stellt sich in der Positionierung von Objekten. Bei einer ungünstigen Positionierung der Objekte kann es vorkommen, daß die einfallenden Lichtstrahlen endlos zwischen den Objekten gespiegelt werden. In einem solchen Falle würden unsere Rechenroutinen nicht-deterministisch den Lichtstrahl verfolgen. Wir müssen also ein Limit für die Rekursionstiefe des Lichtstrahles festlegen, an der der Algorithmus abbricht. Daraus ergibt sich dann zwangsläufig die Frage, wie tief man die Rekursion laufen lassen muß, um ein vernünftiges Ergebnis zu erhalten.

## **Anhang A : Raytracing Programme** (In Klammern [] für welches Betriebssystem)

### **Lowcost-Bereich :**

**Markus Rahlff, Hilmar Koch** :HIGHLIGHT PC, Vogel Verlag und Druck KG, Würzburg, 1988, [MS DOS, MS Windows]

**Andreas Stabinger** :Raytrace für Windows, Data Becker, Düsseldorf, 1993, [MS Windows]

**N.N.:** Lightning 3D, DMV Daten- und Medienverlag GmbH & Co. KG, Feldkirchen [MS DOS]

**N.N.:** 3D F/x, Softline GmbH, Oberkirch-Baden [MS Windows]

**N.N.:** Reflections, PC Oberland, Kronberg [MS Windows]

### **Highend-Bereich :**

- Imagine, PC Oberland, Kronberg, 1994 [MS DOS]
- Arena Design, Softline GmbH, Oberkirch-Baden, 1995 [MS Windows]
- Real 3D, Activa International, Hamburg, 1995 [MS Windows, MS Windows NT]
- 3D Studio, Autodesk GmbH, München, 1995 [MS DOS]
- MathLive, True-D Ltd., Oxford, 1993 [MS Windows, Apple Macintosh System 7, SUN Solaris, SGI IRIX]

## **Anhang B : Literatur**

**Axel Plencke** : "3D-Grafik und Animation", Markt & Technik Verlag AG, Haar, 1989, Seite 263-389

**Dietrich Weineck** : "Spiegel, Strahlen, Reflexionen - Realistische Computerbilder mit Raytracing", DOS 9'90, DMV Daten- und Medienverlag GmbH & Co. KG, Feldkirchen, Seite 44-46, (1990)

**Foley, van Dam, Feiner, Hughes** : "Computer Graphics: Principles and Practice", Addison-Wesley, Bonn, 1990, Seite 701-853

**Hans Jörg Vasold** : "Fast wie im richtigen Leben - Drei Ray-Tracer für PCs im Vergleich", c't 6'91, Heinz Heise Verlags KG, Hannover, Seite 76-80, (1991)

**Dieter Hochacker** : "Werkstatt : Raytracing - Wunderweltenbummler", DOS 5'92, DMV Daten- und Medienverlag GmbH & Co. KG, Feldkirchen, Seite 278-286, (1992)

**Matthias Oberdorfer** : "Raytracing in C - Schöne neue Welt", DOS EXTRA 19'92, DMV Daten- und Medienverlag GmbH & Co. KG, Feldkirchen, Seite 126-157, (1992)

"Objekte im Raum", DOS 1'93, DMV Daten- und Medienverlag GmbH & Co. KG, Feldkirchen, Seite 184-192, (1993)

**Andreas Stabinger** : "Raytrace 2.0", DATA BECKER GmbH, Düsseldorf, 1993

**James D. Foley** : "Grundlagen der Computergraphik", Addison-Wesley, Bonn, 1994

**Ralf Conway** : "Zaubern mit Licht und Schatten - 3D-Bilder durch Raytracing", DOS 12'95, DMV Daten- und Medienverlag GmbH & Co. KG, Feldkirchen, Seite 122-126, (1995)