

Proseminar “3D Grafikprogrammierung”

WS 2004/05

Thema 6: Beleuchtung

Matthias Nieuwenhuisen

08. November 2004

Zusammenfassung

Diese Arbeit über das Thema Beleuchtung soll einige grundlegende Einblicke in verschiedene Bereiche der lokalen Beleuchtungsmodelle geben. Anfangs soll die Bedeutung von Lichtquellen erläutert und diverse Quellen vorgestellt werden, wie globales Licht, Flächenlicht oder auch Spots. Weiter werden Materialeigenschaften beschrieben und in Zusammenhang mit der Reflexion bzw. Spiegelung erklärt was Materialien sind und welchen Einfluss diese auf ein Objekt und die Gesamtszene haben. Hierbei soll jeweils kurz auf die Nutzung dieser Eigenschaften in OpenGL eingegangen werden. Zuletzt soll noch ein Einblick in die Mathematik die hinter den Beleuchtungsmodellen steht gewährt und die wichtigsten Gleichungen erklärt werden.

1 Einleitung

Ein wichtiger Bereich bei der Darstellung 3-dimensionaler Computergrafiken ist der richtige Einsatz von Beleuchtungsmöglichkeiten. Hierbei wird unterschieden zwischen globalen Beleuchtungsmodellen wie Raytracing und Radiosity und den lokalen Beleuchtungsmodellen. Globale Beleuchtungsmodelle berechnen einen Bildpunkt unter Berücksichtigung aller in der Szene vorhandenen Lichtquellen und Objekte, lokale Modelle hingegen berücksichtigen nur das Material des zu berechnenden Bildpunktes und das dort eintreffende Licht. Daher ist in lokalen Modellen keine Interaktion zwischen Objekten möglich, wie ein Objekt dass sich in einem anderen spiegelt oder ähnliche Effekte. In diesen Modellen wird versucht mit Approximation und Interpolation der physikalischen Eigenschaften des Lichtes eine Szene plastischer zu gestalten und grundlegende Lichteffekte wie Reflexion und Spiegelung zu simulieren. Diese Beleuchtungsmodelle, wie sie auch in OpenGL verwendet werden, eignen sich jedoch nicht um photorealistische Grafiken zu erstellen, dieser Bereich bleibt den globalen Modellen vorbehalten. Jedoch eignen sich lokale Modelle auf Grund der einfacheren Komplexität besser um sie in Hardware zu implementieren und so in Echtzeit zu berechnen.

2 Beleuchtungsmodelle

Ein Beleuchtungsmodell in der Computergraphik beschreibt die gesamte Beleuchtung einer Szene in Abhängigkeit der Lichtquellen und der zu zeichnenden Objekte. Einerseits werden hier sowohl die Position und Art von Lichtquellen definiert, als auch die Eigenschaften der Objekte in Bezug auf Licht Ein- und Ausfall. Weiterhin beschreibt dieses Modell wie Farben zu Bildpunkten (Pixeln) zugeordnet werden, das sogenannte Shading.

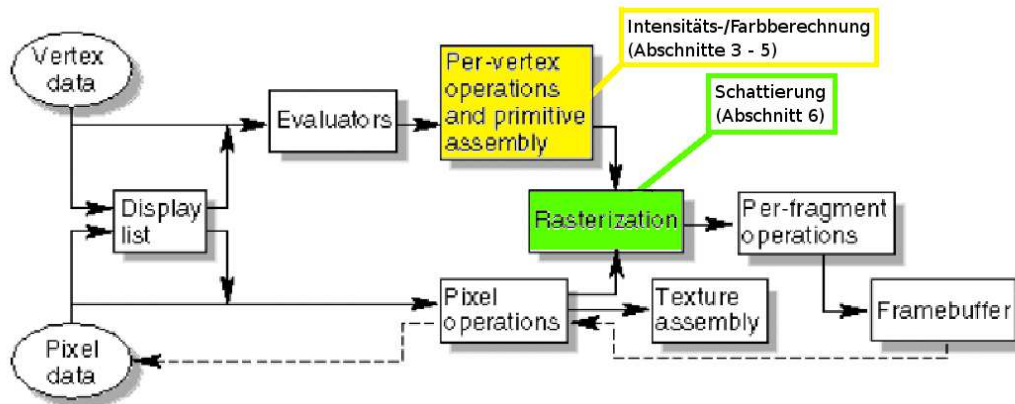


Abbildung 1: Die Beleuchtungsberechnung findet an zwei Stellen in der Rendering Pipeline statt (Bildquelle [1]).

Wichtig für die Beleuchtung eines Polygons sind die Normalen. Dabei unterscheidet man zwischen den Flächennormalen, dies sind Vektoren die senkrecht auf der Fläche eines Polygons stehen und den Eckennormalen, die durch einen Vertex gehen. Sollten diese nicht bereits bei der Modellierung festgelegt worden sein, kann man die Eckennormalen ermitteln, indem der Durchschnittsvektor der Flächennormalen aller anliegenden Polygone berechnet wird. Soll zwischen zwei Polygonen eine scharfe Kante entstehen, muss pro Seite der Kante jeweils eine Normale ermittelt werden unter Zuhilfenahme der dort liegenden Polygone.

3 Lichtquellen

Lichtquellen werden eingesetzt um einer Szene mehr Realismus zu verleihen. So wirkt eine Szene wesentlich plastischer, wenn durch einfallendes Licht Spiegelungen oder Reflektionen entstehen und Farben auf der einer Lichtquelle zugewandten Seite eines Objekts heller wirken als welche auf einer lichtabgewandten. Auch wird erst durch geeignete Beleuchtung ein Gefühl von Tiefe in der Szene erzeugt.

Lichtquellen haben jeweils eine ambiente, eine diffuse und eine spekulare Komponente. Unter ambientem Licht versteht man Licht, dessen Richtung, zum Beispiel durch vielfache Reflektion, nicht mehr zu orten ist. Der ambiente Anteil an einer Lichtquelle beleuchtet alle Seiten aller Objekte gleich stark, egal ob sie der Lichtquelle zu- oder abgewandt sind. Die diffuse Komponente trifft zwar gerichtet auf das Objekt auf und hat einen feststellbaren Ursprung, wird jedoch ebenfalls ungerichtet reflektiert. Schliesslich hat spekulares Licht die Eigenschaft gerichtet von einer Lichtquelle auf ein Objekt zu treffen. Um eine Lichtquelle in OpenGL zu erzeugen stehen folgende Funktionen zur Verfügung:

```
// Lichtquellenparameter setzen
void glLightf(GLenum light, GLenum pname, TYPE param);
void glLightfv(GLenum light, GLenum pname, TYPE *param);

// Lichtquelle aktivieren
void glEnable(GLenum light);
```

GLenum Light entspricht einer Lichtquelle GL_LIGHT0, ... , GL_LIGHT7 und GLenum pname einem Parameternamen mit zugehörigen Parametern param oder *param. Die wichtigsten allgemeinen Parameter:

Parameter Name	Standardwert	Beschreibung
GL_AMBIENT	(0.0, 0.0, 0.0, 1.0)	ambienter RGBA Anteil
GL_DIFFUSE	(1.0, 1.0, 1.0, 1.0)	diffuser RGBA Anteil
GL_SPECULAR	(1.0, 1.0, 1.0, 1.0)	spekularer RGBA Anteil
GL_POSITION	(0.0, 0.0, 1.0, 0.0)	(x,y,z,w) Position

Hierbei beschreibt der Vektor RGBA den Rot-, Grün-, Blau- und Alphaanteil des Lichtes. Der Alphaanteil hat Bedeutung für die Berechnung der Transparenz von Objekten und wird im Folgenden als 1.0 angenommen (\rightarrow *Thema 9: Blending und Transparenz*). Bei (x,y,z,w) gibt es zwei Interpretationen: Bei $w = 0$ beschreiben (x,y,z) die Richtung aus der Licht einer weit entfernten Lichtquelle auf die Szene trifft, bei $w \neq 0$ gibt (x,y,z) eine Position der Lichtquelle innerhalb des Koordinatensystems der Szene an. Im zweiten Fall wird das Licht mit zunehmender Entfernung zur Lichtquelle schwächer. Hierzu kann einer Lichtquelle ein Abschwächungsfaktor zugewiesen werden, der sich in einen konstanten (k_c), einen linearen (k_l) und einen Quadratischen (k_q) Faktor aufspaltet. Die Gesamtabschwächung des Lichtes berechnet sich nun in Abhängigkeit der Distanz zwischen Objekt und Lichtquelle (d) wie folgt:

$$\text{Abschwächung} = k_c + k_l \cdot d + k_q \cdot d^2 \quad (1)$$

Die resultierende Intensität des Lichtes wird nun durch Division der vollen Intensität durch den Abschwächungsfaktor an dem entsprechenden Punkt bestimmt. In OpenGL kann die Abschwächung durch folgende Parameter geändert werden:

Parameter Name	Standardwert	Beschreibung
GL_CONSTANT_-ATTENUATION	1.0	konstante Abschwächung
GL_LINEAR_-ATTENUATION	0.0	lineare Abschwächung
GL_QUADRATIC_-ATTENUATION	0.0	quadratische Abschwächung

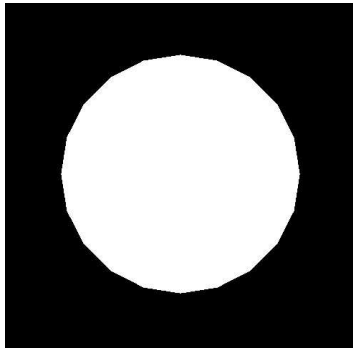


Abbildung 2: *OpenGL Szene ohne Beleuchtung*

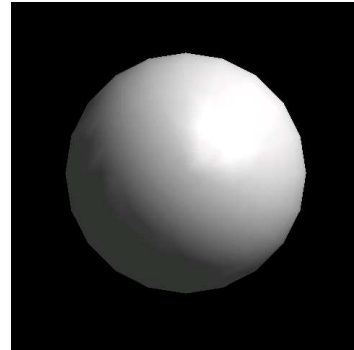


Abbildung 3: *mit globaler Beleuchtung und einer Lichtquelle*

3.1 globales ambientes Licht

Unter globalem ambienten Licht versteht man die Hintergrundbeleuchtung einer Szene. Diese Art von Licht kommt in der Natur nicht vor, sondern entsteht durch häufige Reflektion und Brechung von einfallendem Licht aus diversen Lichtquellen. Dadurch entsteht eine Art Hintergrundleuchten, dass in der Computergraphik durch ambientes Licht simuliert wird. Dieses Licht wird global für die gesamte Szene festgelegt und ist daher an sich keine Lichtquelle sondern eine Eigenschaft des Beleuchtungsmodells. Parameter sind hierbei die Farbe und die Intensität. In Abbildung 3 ist globales ambientes Licht dafür verantwortlich, dass auch die lichtabgewandte Seite der Kugel beleuchtet ist, im Gegensatz zu Abbildung 10 in der kein globales ambientes Licht aktiviert wurde.

```
GLfloat lmodel_ambient[] = { 0.2, 0.2, 0.2, 1.0 };
glLightModelfv(GL_LIGHT_MODEL_AMBIENT, lmodel_ambient);
```

3.2 Punktlichtquelle

Eine Punktlichtquelle sendet Licht in alle Richtungen (gleichmäßig) aus. Punktlichtquellen werden durch ihre Leuchtstärke und ihre Position beschrieben. Glühbirnen sind ein Beispiel für die Anwendung von Punktlichtquellen. Ist eine Punktlichtquelle weit ausserhalb einer Szene positioniert, verhält sich einfallendes Licht wie das einer gerichteten Flächenlichtquelle.

3.3 Gerichtete Flächenlichtquelle

Diese Lichtquelle besteht aus einer Fläche, die Lichtstrahlen gerade in eine bestimmte Richtung aussendet. Dadurch können zum Beispiel große, entfernte Punktlichtquellen simuliert werden, wie zum Beispiel die Sonne, die im Normalfall weit

ausserhalb der Szene liegt und daher nicht als Punktlichtquelle realisiert werden kann.

3.4 Spot

Ein Spot ist eine Punktlichtquelle, die jedoch nur in einem bestimmten Winkel Licht ausstrahlt. Als Beispiel können Schreibtischlampen oder Autoscheinwerfer angeführt werden. Der Spot wird charakterisiert durch Position, Beleuchtungsstärke, Abstrahlwinkel und Reichweite.

Parameter Name	Standardwert	Beschreibung
GL_SPOT_DIRECTION	(0.0, 0.0, -1.0)	(x,y,z) Abstrahlrichtung
GL_SPOT_CUTOFF	180.0	Abstrahlwinkel

4 Materialien

Das Material einer Oberfläche legt fest, wie dieses Objekt und seine Farbe bei Beleuchtung reagieren. So wirkt ein roter Teppich anders als ein rot lackiertes Auto, obwohl beide die selbe Farbe haben. Es gibt in OpenGL Eigenschaften für verschiedene Arten der Reflexion, für Emission eines Objektes und die Farbe. Im einzelnen sind diese in der untenstehenden Tabelle aufgeführt.

Die ambiente Farbe gibt an wie ambientes Licht reflektiert wird, so würde ein Objekt, dass als ambiente Farbe den Farbvektor (1,0,0,1), welcher Rot entspricht, nur ambientes Licht mit der Farbe rot reflektieren. Für die diffuse bzw. spekulare Farbe des Objekts gilt das selbe in Zusammenhang mit diffusem bzw. spekularem Lichteinfall. Eine weitere Materialeigenschaft ist die Eigenemission des Objektes. Diese beschreibt mit welcher Farbe die Oberfläche von sich aus leuchtet. Hiermit können Lichtquellen in der Szene simuliert werden, zum Beispiel eine Schirmlampe. Dies ist nötig, da eine Lichtquelle an sich nicht sichtbar ist, sondern nur ihre Auswirkungen beim Auftreffen auf ein Material. Jedoch wirken diese Eigenemissionen nicht als Lichtquelle für andere Objekte.

Parameter Name	Standardwert	Beschreibung
GL_AMBIENT	(0.2, 0.2, 0.2, 1.0)	Ambiente Farbe
GL_DIFFUSE	(0.8, 0.8, 0.8, 1.0)	Diffuse Farbe
GL_SPECULAR	(0.0, 0.0, 0.0, 1.0)	Spekulare Farbe
GL_AMBIENT_AND_DIFFUSE		ambientes und diffusen Anteil der Farbe gemeinsam setzen
GL_SHININESS	0.0	Exponent des Glanzes
GL_EMISSION	(0.0, 0.0, 0.0, 1.0)	Farbe der Eigenemission

5 Reflexion

Reflexionen entstehen beim Auftreffen eines Lichtstrahls auf ein Material. Dabei gilt im idealisierten Fall, dass der Einfallswinkel des Lichtes gleich dem Ausfallswinkel des Lichtes ist. Jedoch tritt dieser Idealfall in der Realität nicht auf, da selbst ein Spiegel eine raue Oberfläche hat an der das Licht gebrochen wird, wenn auch weniger stark als bei matten Objekten. Um dieses reale Verhalten nachzuempfinden gibt es in der Computergraphik mehrere Ansätze.

5.1 Diffuse Reflexion

Wenn Licht auf eine sehr raue Oberfläche trifft wird es mehrfach gebrochen und nicht in eine bestimmte Richtung reflektiert, sondern ungerichtet in alle Richtungen wieder abgestrahlt. Bei der ideal diffusen Reflexion hat reflektiertes Licht keine Vorzugsrichtung und strahlt in alle Richtungen mit der selben Intensität wieder ab. Hiermit entsteht der Effekt eintreffenden Lichtes auf nicht glänzendes Material.

5.2 Spiegelnde Reflexion

Bei der ideal spiegelnden Reflexion wird einfallendes Licht nach den Strahlgesetzen aus der Optik reflektiert. Hierbei gilt Einfallswinkel des Lichtes ist gleich dem Ausfallswinkel. Da jedoch, wie oben beschrieben, in der Realität keine perfekten Spiegel vorkommen, wurden verschiedene Modelle erstellt um mehr Realitätsnähe zu bekommen.

5.3 Spekulare Reflexion

Die spekulare Reflexion ist eine Mischung aus der diffusen und der spiegelnden Reflexion. Hierbei wird ein Teil des Lichtes ideal diffus abgestrahlt, jedoch der Rest mit einer Vorzugsrichtung. Hierbei ist die Abstrahlintensität am höchsten auf der Achse der ideal spiegelnden Reflexion und nimmt ab, je weiter es sich von dieser Achse entfernt. In der Computergraphik wird meistens von einer spekularen, statt einer idealen, Reflexion ausgegangen. Durch Variation des Verhältnisses des diffusen und des spiegelnden Anteils können so ideale Reflexionen angenähert werden. Zur Berechnung der spekularen Reflexion gibt es mehrere Verfahren, von denen hier zwei aufgeführt sind.

5.3.1 Torrance-Sparrow

Das Verfahren von Torrance-Sparrow geht davon aus, dass die Oberfläche eines Objektes durch viele Facetten modelliert wurde. Diese Facetten haben entweder

eine gemeinsame grobe Ausrichtung, wodurch das Material spiegelnder wird oder liegen sehr zufällig ausgerichtet, was zu mattem Material führt. Dieses Verfahren berücksichtigt viele physikalische Faktoren, wie die Dämpfung des Lichtes wenn ein Teil der Strahlen von einer anders ausgerichteten Facette abgeschottet wird. Durch die hohe Anzahl an zu berücksichtigenden Parametern ist dieses Verfahren mit umfangreichen Berechnungen verbunden und wird in der Echtzeitgraphik eher nicht eingesetzt.

5.3.2 Phong

Phong Bui-Tong entwickelte ein Verfahren dass im Gegensatz zu Torrance-Sparrow nicht auf physikalischen Grundsätzen basiert, sondern rein heuristisch ist. Dieses Verfahren geht davon aus, dass eine Spiegelung die größte Intensität hat, wenn der Betrachter genau aus der Richtung auf die Szene schaut in der die ideale Abstrahlrichtung des Lichtes wäre. Phong sieht eine einzelne Materialabhängige Konstante vor, mit der beschrieben wird wie spiegelnd das Material ist. So berechnet sich der Anteil des spiegelnden Lichtes für eine Abstrahlrichtung als

$$R_s := R_{s0} \cdot \cos^m(\lambda) \quad (2)$$

wobei R_{s0} die Materialkonstante ist, die den spiegelnden Anteil des Materials angibt. λ ist der Winkel zwischen der idealen Abstrahlrichtung des Lichtes und der Richtung aus der der Betrachter die Szene sieht und m der Exponent des Glanzes (shininess exponent). Dieser legt fest, wie schnell der spiegelnde Anteil des Lichtes abfällt, je weiter man sich von der idealen Abstrahlrichtung entfernt. Bei höherem Exponenten entstehen kleinere Glanzpunkte (*Bild 5*), bei kleinerem Exponenten große ausgeleuchtete Flächen (*Bild 4*). In der Praxis wird häufig statt der idealen Abstrahlrichtung des Lichtes der Halbvektor H benutzt, da dieser schneller berechnet werden kann und vergleichbare Ergebnisse erzielt werden. Dies ist ein Vektor, der mit Lichteinfallvektor und Betrachtungsrichtungsvektor in einer Ebene liegt und in dieser Ebene eine Winkelhalbierende darstellt (*siehe Bild 12*).

6 Schattierung (Shading)

Beim Shading, nicht zu verwechseln mit Schattenwurf, werden den einzelnen Pixeln einer Szene Farben zugeordnet. Im Allgemeinen wird hierbei aus Gründen der Performance das erstellte Beleuchtungsmodell nicht auf jeden Pixel einzeln angewendet, sondern eine geeignete Interpolation ausgewählt nach der das Shading vorgeht.

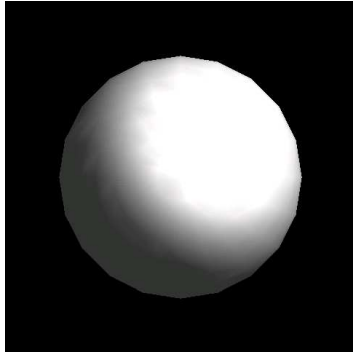


Abbildung 4: *niedriger Glanz-Exponent*

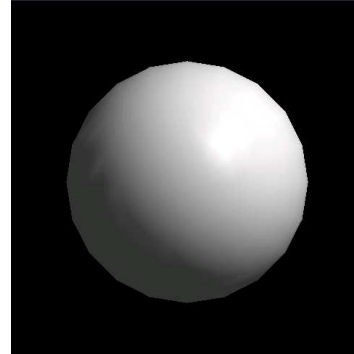


Abbildung 5: *hoher Glanz-Exponent*

6.1 Flat-Shading

Dies ist die einfachste angewandte Methode eine Szene zu schattieren. Beim Flat-Shading wird im ersten Schritt von jedem Polygon ein einzelner Punkt ausgewählt und diesem nach den Regeln des Beleuchtungsmodells eine Farbe zugeordnet. Danach wird diese Farbe auf alle weiteren Punkte des Polygons übertragen. Flat Shading lässt sich auf Grund des relativ geringen Aufwandes schnell durchführen, jedoch entstehen hierbei, insbesondere bei runden Objekten, starke Farbabstufungen und damit unrealistische Bilder (*Bild 6*).

```
glShadeModel(GL_FLAT);
```

6.2 Gouraud-Shading

Das Gouraud-Shading bietet unter OpenGL die Möglichkeit einer realistischeren Interpolation als Flat-Shading. Bei dieser Art der Schattierung wird das Beleuchtungsmodell nicht auf einen Punkt im Polygon, sondern auf die jeweiligen Eckpunkte des Polygons angewandt (*Bild 7*). Im nächsten Schritt werden die die Punkte verbindenden Linien durch lineare Interpolation der Eckfarben schattiert (*Bild 8*). Nun tastet eine sogenannte Scan-Line das Polygon ab und schattiert jeweils die Line zwischen den Kreuzungspunkten der Scan-Line mit den bereits schattierten Aussenlinien des Polygons nach dem selben Verfahren wie bereits die Aussenlinien selbst (*Bild 9*). Durch dieses Verfahren erscheinen Rundungen wesentlich weicher und auch Reflektionen realistischer. Jedoch wirken auch beim Gouraud-Shading Reflektionen sehr weich, harte Glanzpunkte wie sie bei Metall auftreten sind nicht möglich (*Bild 10*).

```
glShadeModel(GL_SMOOTH);
```

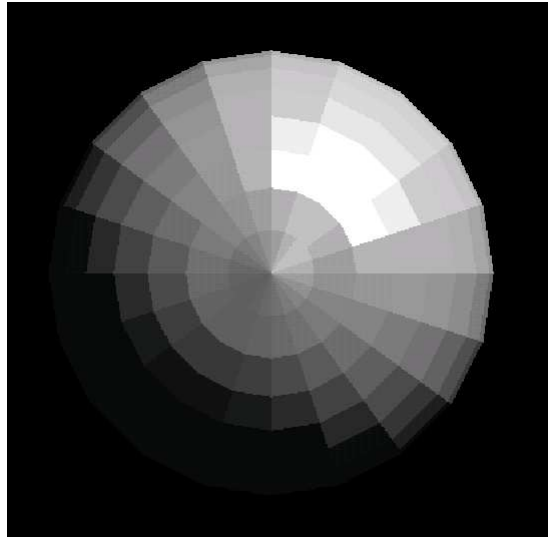


Abbildung 6: Beim Flat-Shading treten deutliche Farbunterschiede zwischen den Polygonen auf.

6.3 Phong Shading

Das Phong-Shading arbeitet ebenso wie das Gouraud-Shading nach dem Scan-Line Verfahren zur Interpolation, jedoch werden hier nicht die Farbwerte interpoliert, sondern die Normalenvektoren. Dafür werden wiederrum erst die Normalen auf den Seiten der Polygone berechnet anhand der Ecknormalen und danach die Normalen auf den Scan-Lines. Zu letzt wird das Beleuchtungsmodell auf jedem Pixel einzeln angewandt. Bei diesem Verfahren ist ein wesentlich höherer Aufwand pro Pixel durchzuführen. Mit diesem Verfahren sind auch Glanzpunkte wie sie bei metallischen Objekten auftreten möglich, jedoch wirken scharfe Kanten

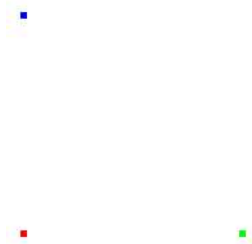


Abbildung 7: Bei Gouraud werden zuerst die Eckpunkte eingefärbt.

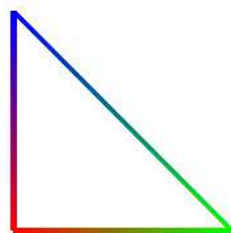


Abbildung 8: Danach wird die Farbe linear zwischen den Eckpunkten interpoliert.

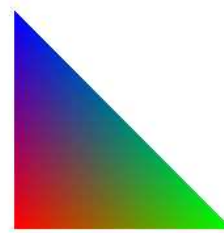


Abbildung 9: Im letzten Schritt werden die Pixel zwischen den Seiten eingefärbt.

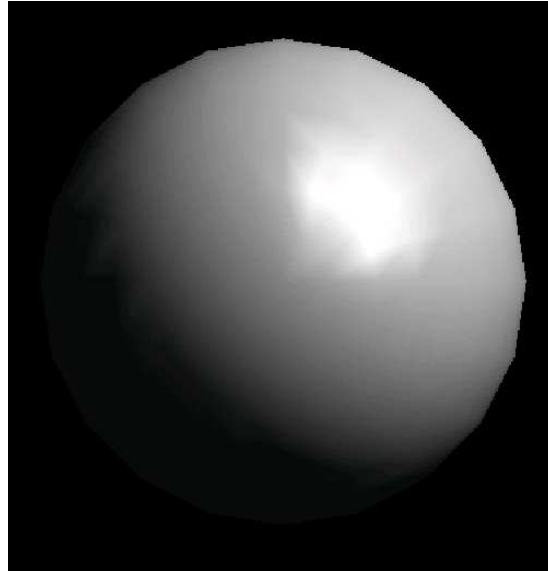


Abbildung 10: Beim Gouraud-Shading sind die Farbübergänge fließend.

oft weniger realistisch als bei der Methode von Gouraud (*Bild 11*). Da OpenGL keine Hardwarebeschleunigung für diese Art des Shadings zur Verfügung stellt, eignet sich Phong-Shading nicht ohne weiteres für Echtzeitanwendungen wie zum Beispiel Computerspiele. Es gibt jedoch inzwischen Möglichkeiten Phong-Shading durch eigene Programmierung der Grafikkipeline von OpenGL zu nutzen (\rightarrow *Thema 10: HLSL und programmierbare Grafikkipeline*).

7 Mathematische Grundlagen

7.1 Intensität des Lichtes

Unter der Intensität I des Lichtes versteht man, wieviel Licht eine Lichtquelle abstrahlt. Die normierte Intensität wird als Vektor der Rot-, Grün-, Blauanteile des Lichtes geschrieben und kann pro Farbe zwischen 0 und 1 liegen. So erhielte man zum Beispiel rotes Licht, wenn man die rote Komponente auf 1 setzte und die Grüne und Blaue jeweils auf 0.

$$I := \begin{cases} I \in [0, 1] & \text{(Schwarz-Weiß)} \\ (I_R, I_G, I_B) \in [0, 1]^3 & \text{(Farbig)} \end{cases} \quad (3)$$

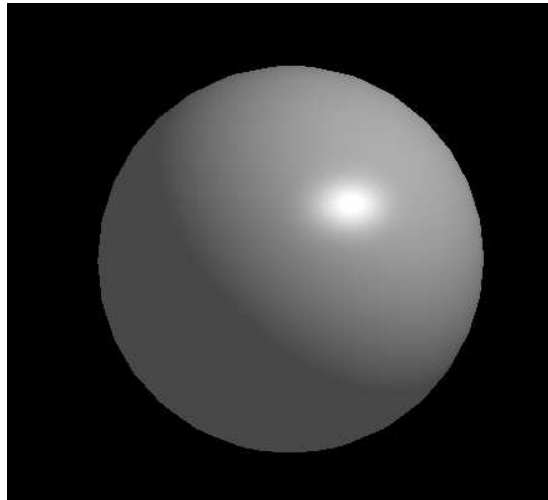


Abbildung 11: Durch Phong-Shading werden Reflexionen runder als mit Gouraud.

7.2 Reflexionskoeffizient

Der Reflexionskoeffizient R ist eine Materialkonstante und gibt an wieviel rotes, grünes und blaues Licht von dem verwendeten Material wieder reflektiert wird. Diese Konstante wird einer Fläche in ihrer Gesamtheit zugeordnet. Es wird unterschieden zwischen den Koeffizienten für ambientes (R_a), diffuses (R_d) und spekulares Licht (R_{s0}).

$$R := \begin{cases} R \in [0, 1] & \text{(Schwarz-Weiß)} \\ (R_R, R_G, R_B) \in [0, 1]^3 & \text{(Farbig)} \end{cases} \quad (4)$$

Da die spekulare Reflexion abhängig von der Betrachtungsrichtung ist, kommt noch der Spiegelreflexionskoeffizient der spekularen Reflexion nach Phong aus Formel 2 hinzu.

$$R_s := R_{s0} \cdot \cos^m(\alpha) \quad (5)$$

Dabei entspricht α dem Winkel zwischen idealer Abstrahlrichtung S des Lichtes und Betrachtungsrichtung E und m bezeichnet den Exponenten des Glanzes.

7.3 Intensität eines Bildpunktes

Die Gesamtintensität eines Pixels berechnet sich wie folgt:

$$\begin{aligned} \text{Intensität} &= \text{Emission des Materials} \\ &+ \text{globales Ambientes Licht} * \text{ambiente Materialreflexion} \\ &+ \text{ambienter Anteil der Lichtquellen} * \text{ambiente Materialreflexion} \end{aligned}$$

- + diffuser Anteil der Lichtquellen * diffuse Materialreflexion
- + spekularer Anteil der Lichtquellen * spekulare Materialreflexion

Die Intensität V des Lichtes in einem Bildpunkt hängt von der Richtung des Lichteinfalls, beschrieben durch den Einfallsvektor L , in Relation zur Oberfläche bzw. dem Normalenvektor N ab. Hierbei muss unterschieden werden, ob die ambiente oder die diffuse bzw. spekulare Intensität berechnet werden soll. Für die Abschwächung gelten die k 's wie in Formel 1 (Seite 3).

$$\begin{aligned}
 a &:= \frac{1}{k_c + k_l d + k_q d^2} && \text{(Abschwächung)} \\
 V_a &:= I_a * R_a && \text{(ambiente Intensität)} \\
 V_d &:= a \cdot (I_d * R_d) \cdot \langle L, N \rangle = a \cdot (I_d * R_d) \cdot \cos \phi && \text{(diffus)} \\
 V_s &:= a \cdot (I_s * R_{s0}) \cdot \langle A, E \rangle = a \cdot (I_s * R_{s0}) \cdot \cos \alpha && \text{(spekular)} \\
 V_s &:= a \cdot (I_s * R_{s0}) \cdot \langle H, N \rangle = a \cdot (I_s * R_{s0}) \cdot \cos \beta && \text{(mit Halbvektor)}
 \end{aligned} \tag{6}$$

Um nun die Gesamtintensität V eines Pixels zu berechnen muss die globale ambiente Intensität mit den Intensitäten die durch die einzelnen Lichtquellen beigegeben werden aufsummiert werden. Hierbei gilt, die Intensitäten aller Lichtquellen können einzeln berechnet und danach aufsummiert werden. Für n Lichtquellen gilt also:

$$V := V_{a,global} + \sum_{j=1}^n (V_{j,a} + V_{j,d} + V_{j,s}) \tag{7}$$

Hierbei bezeichnet $V_{j,x}$ die ambiente ($x = a$), die diffuse ($x = d$) bzw. spekulare ($x = s$) Komponente der j -ten Lichtquelle.

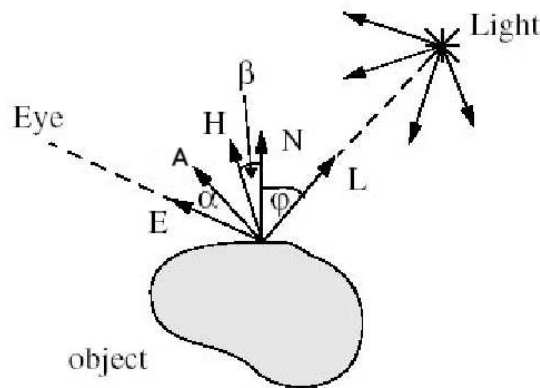


Abbildung 12: Berechnung der Intensität eines Punktes (Bildquelle [4])

Literatur

- [1] Jackie Neider Dave Shreiner, Mason Woo. *OpenGL Programming Guide (Red Book)*. Addison-Wesley Professional.
- [2] C. Zenger H.-J. Bungartz, M. Griebel. *Einführung in die Computergraphik*. vieweg, 2002.
- [3] R. Klein J. Encarnação, W. Straßer. *Graphische Datenverarbeitung 1*. Oldenbourg Verlag, 1996.
- [4] Alexander Speed Sonja Gutzeit, Hendrik Schneider. <http://www.uni-weimar.de/~schneid2/illusionhole2/>. Technical report, Uni Weimar.