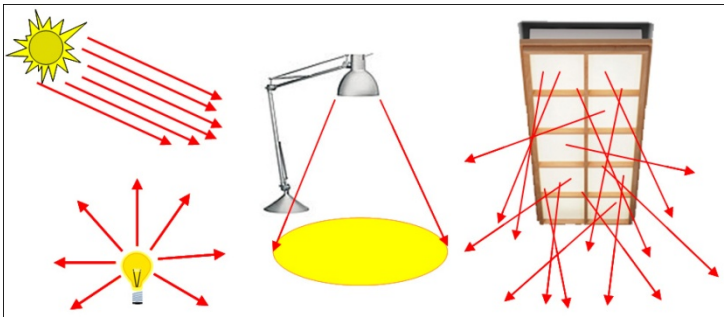
Licht und Schattierung

Ein Beleuchtungsmodell oder auch Schattierungsmodell (Illumination Model oder Lighting Model oder Shading Model) dient dazu, aus der Beschreibung der Lichtverhältnisse und den Oberflächeneigenschaften eines Objektes zu berechnen, welche Farbe/Helligkeit der Betrachter wahrnimmt, also welche Farbe das zugehörige Pixel erhalten soll. Zusammen mit der perspektivischen Projektion ist das der wichtigste Beitrag für das realistische Aussehen von Computergraphik-Bildern.

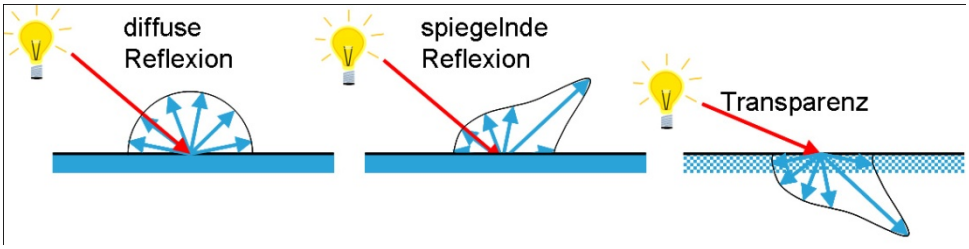
Alle folgenden Betrachtungen und Formeln beziehen sich der Einfachheit halber nur auf die Helligkeit der Beleuchtung. Um Farben zu behandeln, müssen diese Berechnungen in mehreren Wellenlängen durchgeführt werden, im einfachsten Fall in Rot, Grün und Blau.

Lichtquellen und Oberflächen



Lichtquellen
Um einen Beleuchtungseinfluss berechnen zu können, braucht man zuerst einmal Lichtquellen. Merkmale von Lichtquellen können sein:
Form: Lichtrichtung, Punktlichtquellen, gerichtete Punktlichtquellen, flächige Lichtquellen, …
Eigenschaften: Helligkeit, Farbe, Entfernung, …

Objektoberflächen
Oberflächen können das einfallende Licht entweder diffus reflektieren, d.h. es wird in jede Richtung gleich viel Licht reflektiert (z.B. Papier, Kreide), oder spiegelnd reflektieren, d.h. in die Spiegelungsrichtung wird mehr Licht reflektiert als in die anderen Richtungen (z.B. Lack, Metall), oder transparent sein, d.h. das Licht geht durch die Oberfläche durch und kommt auf der anderen Seite wieder heraus (z.B. Glas, Wasser). Reale Oberflächen besitzen meist eine Mischung aus diesen Eigenschaften. Weiters darf nicht übersehen werden, dass Licht nicht nur von den Lichtquellen auf Flächen auftrifft, sondern dass auch reflektiertes Licht von anderen Flächen mitspielt.

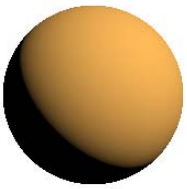


Ein einfaches Beleuchtungsmodell

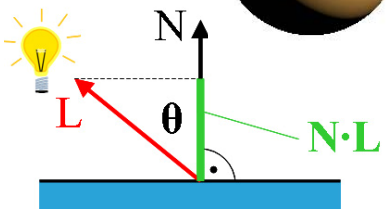
Die physikalisch exakte Simulierung von Licht und dessen Interaktion mit Objektoberflächen ist sehr komplex. Daher verwendet man in der Praxis vereinfachte, empirische Beleuchtungsmodelle. Diese sind etwa so aufgebaut.

Hintergrundlicht (ambientes Licht)
Da jedes Objekt einen Teil des auf ihn auftreffenden Lichtes auch wieder abstrahlt, ist es auch dort nicht ganz dunkel, wo keine Lichtquelle direkt hinleuchten kann. Dieses überall vorhandene Basislicht wird ambientes Licht genannt, auch Hintergrundlicht. Einfache Beleuchtungsmodelle inkludieren dazu einen konstanten Wert I_a zu jeder Beleuchtungsberechnung.

Lambert'sches Gesetz

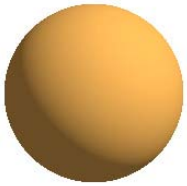


Dieses Gesetz besagt, dass je flacher Licht auf eine Oberfläche auffällt, desto dunkler erscheint diese Oberfläche. Erst durch diesen Effekt erhalten wir den Eindruck einer räumlichen Form.



Sei I_l die Helligkeit der relevanten Lichtquelle, und sei k_d der *diffuse Relexionskoeffizient* der beleuchteten Oberfläche, der also angibt, wieviel Prozent des einfallenden Lichtes in alle Richtungen gleichmäßig wieder abgestrahlt wird. Natürlich gilt $0 \le k_d \le 1$. Weiters sei θ der Winkel zwischen der Oberflächennormale und der Richtung zur Lichtquelle, also der Lichteinfallsrichtung. Dann gilt für die resultierende Intensität I an der Oberflächenstelle:

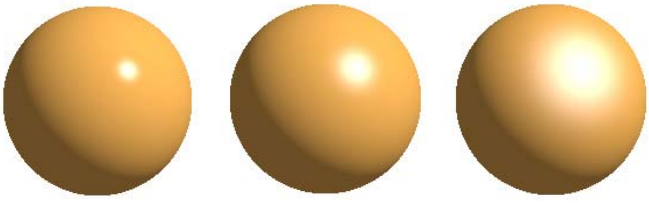
$$\mathbf{I = k_d \cdot I_l \cdot \cos\theta = k_d \cdot I_l \cdot N \cdot L}$$ [N·L ist skalares Produkt]

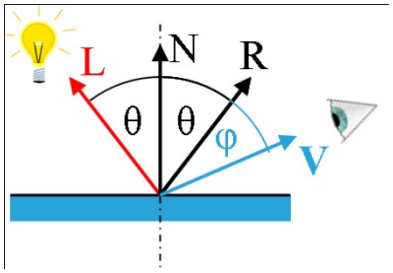


Wenn man nun auch noch das ambiente Licht hinzufügt, dann erhält man schon eine recht schöne Kugel (obere Kugel = nur diffuse Beleuchtung, untere Kugel = diffus + ambient).

Glanzpunkte (Specular Highlights)

Fast jede Oberfläche ist auch etwas spiegelnd. Wenn man diesen Aspekt nicht mitmodelliert, dann wirken alle Materialien gleich stumpf. Da die exakte spiegelnde Reflexion äußerst kompliziert zu berechnen ist, behilft man sich mit einer einfachen Funktion, die einen ähnlichen Verlauf hat wie das Highlight: $\mathbf{\cos^n}$. Mit dem freien Parameter n lässt sich dabei die „Poliertheit" der Oberfläche steuern: je größer n ist, desto kleiner wird der Glanzpunkt und desto glatter wirkt die Oberfläche (linke Kugel), je kleiner das n ist, desto matter wirkt die Oberfläche (rechte Kugel). Um diesen Effekt im richtigen Ausmaß zur Beleuchtung hinzufügen zu können, wird noch ein weiterer Faktor eingeführt, der *spiegelnde Reflexionskoeffizient* k_s. Der Glanz berechnet sich dann nach diesem sogenannten *Phong-Beleuchtungsmodell* so:

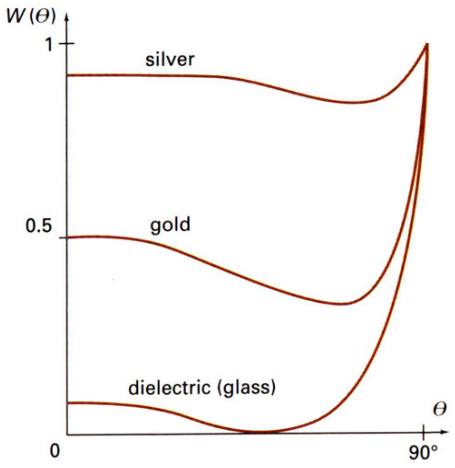




$$\mathbf{I_{l,spec} = k_s \cdot I_l \cdot \cos^n\varphi = k_s \cdot I_l \cdot (R \cdot V)^n}.$$

Der Winkel φ ist dabei der Unterschied zwischen dem exakten Reflexionsstrahl und der Richtung zum Auge.

Etwas näher an der Wahrheit ist die Verwendung der Fresnel'schen Reflexionsgesetze, die beschreiben, dass der Spiegelungsgrad auch vom Lichteinfallswinkel abhängt, dass also der Koeffizient k_s eigentlich eine Funktion W(θ) der Lichteinfallsrichtung ist. Für die meisten Materialien ist dieser Wert aber fast konstant. Daher wird auf diesen Aufwand verzichtet, wenn man nicht gerade ein Material darstellen will, bei dem der Effekt auffällt. Das Bild rechts zeigt die Abhängigkeit dieser Funktion W(θ) vom Winkel zwischen Lichteinfall und Normale auf die Oberfläche für drei verschiedene Materialien.



Bei der Berechnung von R muss man noch bedenken, dass es sich hier um Vektoren im 3D-Raum handelt, wo L, N und R in einer Ebene liegen müssen und alle Länge 1 haben sollen. R ergibt sich zu $R = (2N \cdot L)N - L$

Weil die Glanzfunktion sowieso nur eine grobe Näherung ist, verwendet man auch häufig eine einfache Formel, in der R·V durch N·H ersetzt wird. Der Winkel zwischen N und der Halbierenden H zwischen L und V ist φ sehr ähnlich.

Wenn wir alle bisherigen Komponenten zusammensetzen, erhalten wir

ein einfaches komplettes Beleuchtungsmodell: $I = k_a \cdot I_a + \Sigma_{l=1-n} (k_d \cdot I_l \cdot N \cdot L + k_s \cdot I_l \cdot (N \cdot H_l)^n)$

Es gibt noch viele weitere Aspekte, die man berücksichtigen muss, um der Realität näher zu kommen, aber diese werden hier nicht näher beschrieben: Farbverschiebungen in Abhängigkeit der Blickrichtung, Einfluss der Entfernung der Lichtquelle, anisotrope Oberflächen und Lichtquellen, Transparenz, atmosphärische Effekte, Schatten, und so weiter.

Schattierung von Polygonen

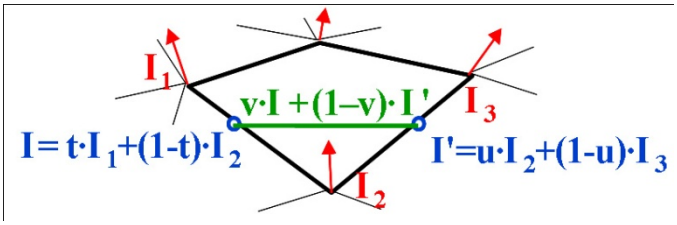
Flat-Shading

Beim Schattieren eines Polygons hat klarerweise jeder Punkt die gleichen Oberflächeneigenschaften, vor allem auch den gleichen Normalvektor. Beim einfachen Ausfüllen jedes Polygons mit einer Farbe werden die Grenzen zwischen den Polygonen deutlich störend erkennbar. Der sogenannte Mach-Band-Effekt, das ist ein kantenverstärkender Mechanismus des Auges, macht das Problem dabei noch ärger als es ist. Dieser Effekt lässt uns an Kanten die dunklere Seite dunkler wahrnehmen als sie ist, und die hellere Seite heller als sie ist. Die einfachste Lösung dieses Problems ist das Interpolieren der Schattierung zwischen den Polygonen. Dazu sind zwei Verfahren üblich: Gouraud-Schattierung und Phong-Schattierung.

Gouraud-Schattierung

Die Gouraud-Schattierung interpoliert die berechneten Helligkeitswerte über die Polygonflächen. Dazu werden an den Eckpunkten der Polygone Helligkeitswerte berechnet und von diesen aus wird durch lineare Interpolation jedes Polygon gefüllt. Konkret geht das so:

(1) An jedem Eckpunkt wird eine Normale als Mittelwert der Normalen aller angrenzenden Polygone berechnet. Das ist natürlich nur ein Näherungswert der Normale der echten zugrundeliegenden Fläche.



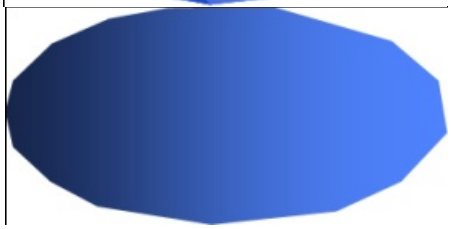
(2) Aus den Eigenschaften der Oberfläche, der Normale und der Lichteinfallsrichtung wird ein Helligkeitswert („Schattierung“) für jeden Eckpunkt berechnet. Man beachte, dass dadurch angrenzende Polygone an diesen Eckpunkten alle die gleichen Werte erhalten.

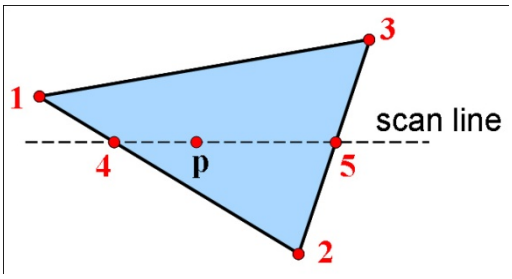
(3) Entlang der Polygonkanten werden die Helligkeitswerte linear interpoliert, d.h. es wird für jeden Schnittpunkt mit einer Scanline ein Wert ermittelt. Man beachte, dass dadurch für aneinander grenzende Polygone entlang der gemeinsamen Kante die gleichen Werte entstehen.

(4) Entlang jeder Scanline wird von der linken bis zur rechten Polygongrenze wieder linear interpoliert. Dadurch haben nebeneinander liegende Pixel immer eine sehr ähnliche Helligkeit und es kommt zu keinen sichtbaren Kanten.

Dennoch verbleiben Fehlerquellen. So wird die Silhouette natürlich nicht verändert, dadurch verbleiben störende Polygonkanten sichtbar (siehe Bild rechts).

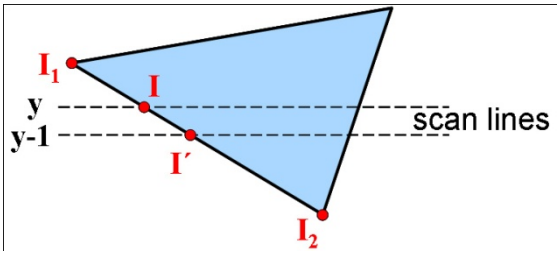
Weiters kommt es im Bereich von Glanzpunkten zu zufälligen Interpolationsergebnissen, je nachdem ob es zufällig eine Normale gibt, die genau einen Glanzpunkt erzeugt oder nicht. Dies stört besonders bei bewegten Objekten.





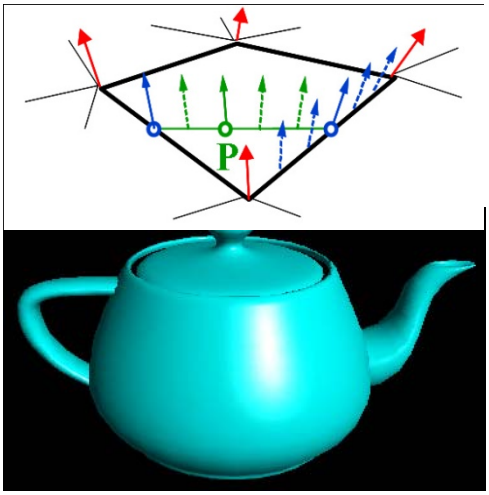
$$I_4 = \frac{y_4 - y_2}{y_1 - y_2} I_1 + \frac{y_1 - y_4}{y_1 - y_2} I_2 \qquad I_p = \frac{x_5 - x_p}{x_5 - x_4} I_4 + \frac{x_p - x_4}{x_5 - x_4} I_5$$

Die (lineare!) *Interpolation der Intensitäten* kann natürlich wieder inkrementell erfolgen:



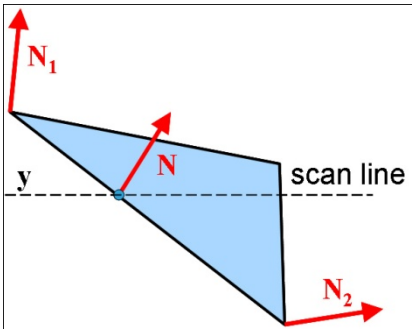
$$I = \frac{y - y_2}{y_1 - y_2} I_1 + \frac{y_1 - y}{y_1 - y_2} I_2 \qquad I' = I + \frac{I_2 - I_1}{y_1 - y_2}$$

Phong-Schattierung



Als Alternative zur Gouraud-Schattierung erzeugt die Phong-Schattierung (nicht mit dem Phong-Beleuchtungsmodell verwechseln!) viel konsistentere Glanzeffekte. Ebenso wie bei der Gouraud-Schattierung werden in den Polygoneckpunkten die Normalen berechnet, aber nun werden diese Normalen entlang der Polygonkanten interpoliert (es sind noch keine Helligkeiten berechnet worden!), und anschließend entlang der Scanlines. Dann wird für jedes Pixel extra die Helligkeit nach einen Beleuchtungsmodell berechnet. Dies verursacht zwar einen höheren Aufwand, führt aber auch zu schöneren Ergebnissen.

Normalvektorinterpolation:



$$N = N_1(y-y_2)/(y_1-y_2) + N_2\,(y_1-y)/(y_1-y_2)$$

ZUR BEACHTUNG : Das Phong-Beleuchtungsmodell (auch Phong-Schattierungsmodell) und die Phong-Schattierung (auch Phong-Interpolation) sind zwei vollkommen unabhängige Dinge!