

Kapitel 5

Lokale Beleuchtung und Schattierung

Bei lokalen Beleuchtungsmodellen werden Lichtstrahlen nur bis zum Auftreffen auf ein Objekt betrachtet und nicht darüber hinaus (siehe Abb. 5.1 links). Reflektierende und transmittierende Lichtstrahlen werden also nicht weiterverfolgt. Transmittierende Lichtstrahlen dringen durch transparente Materialien hindurch. Damit lassen sich weder Schatten, Spiegelungen noch Transparenz direkt bestimmen.

Schattierungsmodelle stellen erweiterte Verfahren dar, die Beleuchtungsmodelle verwenden. So können Schattierungsverfahren beispielsweise für jedes Pixel ein Beleuchtungsmodell aufstellen oder dieses für eine ganze Fläche interpolieren.

5.1 Kurzeinführung in Notation und Funktionen

Aus Gründen der Lesbarkeit und dem besseren Verständnis werden wir Vektoren unterscheiden, die Positionen, Richtungen oder Farben repräsentieren (siehe Abb. 5.2).

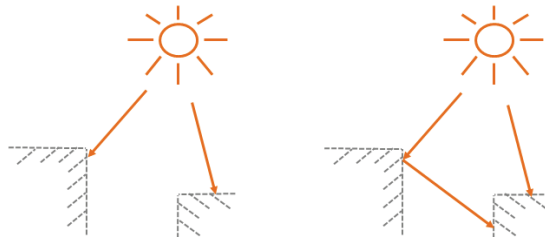


Abbildung 5.1: Links: Lichtstrahlen werden bei lokalen Beleuchtungsmodellen von den Objekten kompensiert und nicht reflektiert oder transmittiert. Rechts: Bei globalen Beleuchtungsmodellen, die oft mit großen zeitlichen Aufwand zu berechnen sind, können reflektierende und transmittierende Lichtstrahlen mit mehreren Objekten interagieren.

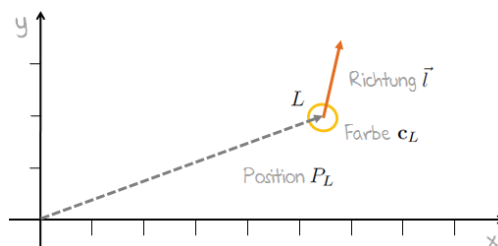


Abbildung 5.2: Für die übersichtlichere Darstellung werden wir Vektoren die Positionen, Richtungen und Farben repräsentieren unterscheiden. In dieser Darstellung beschreiben wir beispielsweise eine Lichtquelle L mit Position P_L , Vorzugsrichtung \vec{l} und Farbe c_L .

Alle drei Varianten liegen in 3-dimensionaler Form vor. Wir werden, wann immer es machbar ist, alle Formeln motivieren, herleiten und Einblicke in Originalarbeiten vornehmen, anstatt diese wie in einer Formelsammlung nur aufzulisten. Auf diese Weise können wir ganz nebenbei ein wenig Lineare Algebra in der praktischen Anwendung erleben.

Am Ende jedes Kapitels gibt es einen Abschnitt, der die wichtigsten Erkenntnisse und Formeln kompakt und übersichtlich darstellt.

5.1.1 Vektoren als Richtungen

Die bekannte Vektorschreibweise \vec{v} , \vec{w} , ... werden wir für Vektoren verwenden, die Richtungen repräsentieren. Es handelt sich dabei wie üblich immer um Spaltenvektoren. Sollten wir aus Platzgründen Vektoren im Text beschreiben, verwenden wir die Zeilenschreibweise von Spaltenvektoren $\vec{v} = (v_1, v_2, \dots, v_n)$. Aber Achtung: Der Vektor \vec{v} ist immernoch ein Spaltenvektor.

Ob eine Richtung beispielsweise zu einer Lichtquelle hin- oder von dieser wegzeigt, kann für die Berechnungen variieren ist dem jeweiligen Kontext zu entnehmen. Um die Richtung zu ändern müssen wir lediglich die Vorzeichen der Komponenten umdrehen.

Sei als Beispiel \vec{v} gegeben mit $\vec{v} = (-1, 2, 0)$, dann zeigt der Vektor $-\vec{v}$ in die entgegengesetzte Richtung:

$$-\vec{v} = (-1) \cdot \vec{v} = (-1) \cdot \begin{bmatrix} -1 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} (-1) \cdot (-1) & \cdot & (-1) \cdot 2 \\ (-1) \cdot 2 & \cdot & (-1) \cdot 0 \\ (-1) \cdot 0 & \cdot & (-1) \cdot 0 \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \\ 0 \end{bmatrix}$$

Skalare und Vektoren

Skalare s, k, \dots haben, wenn nicht anders angegeben, immer den Wertebereich \mathbb{R} und können Richtungen $\vec{w} = s\vec{v}$ mit $s > 1$ strecken, mit $s < 1$ stauchen oder mit $s < 0$ umdrehen, wobei der Skalarwert s komponentenweise mit \vec{v} multipliziert wird (siehe Abb. 5.3).

Besondere Skalare die Winkel darstellen werden wir mit griechischen Buchstaben α, β, \dots bezeichnen.

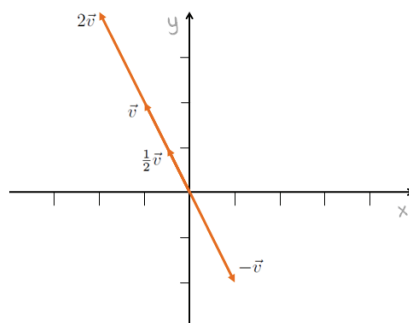


Abbildung 5.3: Wir sehen hier die Veränderungen des Vektors \vec{v} , wenn dieser mit verschiedenen Skalarwerten multipliziert wird.

Hilfreiche Funktionen

Die Länge eines Vektors \vec{v} , symbolisiert durch $\|\cdot\|$, ergibt sich aus $length(\vec{v}) = \|\vec{v}\| = \sqrt{v_1^2 + v_2^2 + \dots + v_n^2}$ und liefert ein Skalar. Durch Anwendung der Funktion $norm(\vec{v})$ erhalten wir einen Einheitsvektor in Richtung von \vec{v} der Länge 1. Damit können wir beispielsweise $\vec{w} = norm(\vec{v}) = \frac{\vec{v}}{\|\vec{v}\|}$ schreiben und speichern in \vec{w} die normierte Version des Vektors \vec{v} , so dass $\|\vec{w}\| = 1$ gilt. Wir können für $\vec{v} = norm(\vec{v})$ auch als eine Anweisung kurz $norm(\vec{v})$ schreiben. Das Skalarprodukt zweier Vektoren $\vec{v} \bullet \vec{w}$ ergibt sich aus der Summe der komponentenweisen Multiplikation:

$$\vec{v} \bullet \vec{w} = \sum_{i=1}^n v_i \cdot w_i$$

Über die wichtige Kosinusformel können wir den Kosinus des Winkels α zwischen zwei Vektoren \vec{v} und \vec{w} ermitteln:

$$\cos \alpha = \frac{\vec{v} \bullet \vec{w}}{\|\vec{v}\| \cdot \|\vec{w}\|}$$

Zwei Vektoren \vec{v} und \vec{w} liegen im selben Halbraum, wenn der Winkel α zwischen diesen positiv ist. Da $\cos \alpha$ in vielen Beleuchtungsmodellen einen wichtigen multiplikativen Gewichtungsfaktor darstellt, sind oft nur positive Winkel interessant. Wir gehen also davon aus, dass ein Lichtstrahl zu der Oberflächennormalen mit einem Winkel größer als 90° keinen Einfluss auf die Helligkeit an dieser Position hat. Nun können wir über die Kosinusformel auch negative Werte erzeugen. Wir verlangen für diese Fälle also statt einem negativen Wert eine 0:

$$\max\left(\frac{\vec{v} \bullet \vec{w}}{\|\vec{v}\| \cdot \|\vec{w}\|}, 0\right)$$

Oder falls \vec{v} und \vec{w} bereits normiert sind:

$$\max(\vec{v} \bullet \vec{w}, 0)$$

Damit wir unnötig lange Formeln vermeiden, wollen wir zukünftig dafür folgende Notation verwenden:

$$\vec{v} \bullet \vec{w} = \max\left(\frac{\vec{v} \cdot \vec{w}}{\|\vec{v}\| \cdot \|\vec{w}\|}, 0\right)$$

Alternativ werden wir mit dem Strich über einem Ausdruck eine Kurzform für Formeln der folgenden Form verwenden:

$$\bar{a} = \min(a, 1)$$

5.1.2 Vektoren als Positionen

Die Notation von Punkten L, P, Q, \dots wird für Vektoren verwendet, die Positionen repräsentieren. Dabei können wir jederzeit zwischen Punkt- und Vektordarstellung wechseln. Z. B. können wir den Vektor zwischen zwei gegebenen Punkten P und Q angeben über $\vec{p} = P - Q$ ¹ oder den Punkt L mit $L = \vec{l}$ angeben, der sich am Ende des Vektors \vec{l} befindet, wenn der Vektor \vec{l} im Ursprung startet.

Hilfreiche Funktionen

In einigen Fällen werden wir den Abstand zwischen zwei Punkten benötigen. Es gibt verschiedene Entfernungsmetriken, in den meisten Fällen werden wir den euklidischen Abstand $dist_{EUK}$ verwenden. Dann schreiben wir einfach:

$$dist_{EUK}(A, B) = \|A - B\| = length(\vec{a} - \vec{b})$$

Hier hilft uns die Darstellung der Punkte als Vektoren, denn wir berechnen einfach die Länge des Vektor $A - B$, der zwischen A und B besteht.

5.1.3 Vektoren als Farben

Da sich Farben besonders bei der Multiplikation (Farbmodulation) im Vergleich zu Vektoren etwas anders verhalten sollen, führen wir Farben eine eigene Notation ein.

Mit $\mathbf{c}, \mathbf{d}, \dots$ werden Vektoren repräsentiert, die Farben des RGB-Farbraumes darstellen (siehe Abb. 5.4).

Dabei stellt eine Farbe \mathbf{c} eine Mischung $\mathbf{c} = (c_R, c_G, c_B)$ aus den drei Farbkämen c_R für Rot (*red*), c_G für Grün (*green*) und c_B für Blau (*blue*) dar. Für sichtbare Farben liegen die Komponenten jeweils im Wertebereich $[0, 1]$. Damit entspricht $(0, 0, 0)$ der Farbe Schwarz und $(1, 1, 1)$ der Farbe Weiss. Wir wollen für die Komponenten bei den Berechnungen allerdings die Wertemenge \mathbb{R} vorübergehend erlauben. Sollte ein Farbwert später dargestellt werden, müssen

¹Kleiner Tipp: Die Eselsbrücke „Köpfchen minus Schwänzchen“ verrät, in welche Richtung der resultierende Vektor zeigt.

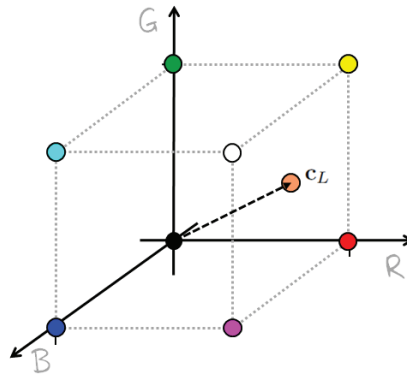


Abbildung 5.4: Der RGB-Farbraum ist ein additiver Farbraum mit den drei Grundfarben Rot, Grün und Blau. In OpenGL liegen alle Komponenten im Wertebereich $[0, 1]$.

kurz vor der Visualisierung mit $c_k = \min(\max(c_k, 0), 1)$ für $k = R, G, B$ die Intervallgrenzen zu $[0, 1]$ transformiert werden.

Jetzt können wir zwei Farben \mathbf{c} und \mathbf{b} komponentenweise addieren $\mathbf{c} + \mathbf{b} = (c_R + d_R, c_G + d_G, c_B + d_B)$ oder multiplizieren $\mathbf{c} \cdot \mathbf{b} = (c_R \cdot d_R, c_G \cdot d_G, c_B \cdot d_B)$. Die Intensität lässt sich auch hier durch die Multiplikation mit einem Skalar s variieren $s\mathbf{c} = (sc_R, sc_G, sc_B)$.

5.2 Materialeigenschaften

Die unterschiedlichen Materialeigenschaften entscheiden maßgeblich über die Interaktionen zwischen einfallenden Lichtstrahlen und Oberfläche und sind so für die Vielfalt der Visualisierungsmöglichkeiten verantwortlich. Nehmen wir als Vergleich die lackierte Oberfläche eines roten Autos und eine grüne Kreidetafel. Lackierte Oberflächen glänzen und haben einen hohen spiegelnden Reflexionsanteil. Kreidetafeln reflektieren im Vergleich dazu eher stark diffus.

Es gibt auch transparente Materialien, die Anteile der Lichtstrahlen hindurchdringen lassen. Diese transmittierende Lichtstrahlen werden in aller Regel von ihrer Bahn abgelenkt, was als Refraktion bezeichnet wird. Eine kleine Übersicht der wichtigsten Begriffe in diesem Kontext sind der Abb. 5.5 zu entnehmen.

5.3 Lichtquellenmodelle

Alle Lichtquellen L haben gemein, dass sie eine Farbe bzw. Intensität \mathbf{c}_L haben. Sollte keine angegeben sein, so gehen wir von der Farbe Weiss mit der maximalen Intensität $\mathbf{c}_L = (1, 1, 1)$ aus. Die grundlegenden Lichtquellenarten unterscheiden sich zunächst im Vorhandensein einer Position P_L und einer Richtung \vec{l} (siehe dazu Abb. 5.6).

Ambientes Licht und Richtungslicht werden als elementare Beleuchtungsmodelle in Abschnitt 5.5 besprochen. Die Besonderheiten von Positionslicht und Strahler

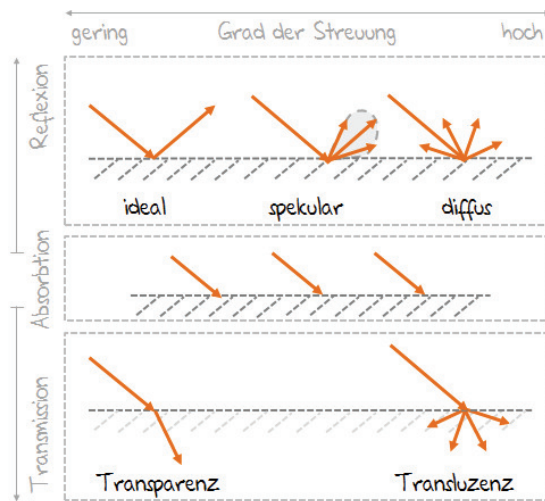


Abbildung 5.5: Oben: Von links nach rechts erhöht sich in dieser Darstellung der Grad der Streuung. So sehen wir bei der Reflexion, dass im idealen Fall ein einfallender Lichtstrahl direkt reflektiert wird. Erhöht sich der Grad der Streuung so haben wir einen Streukegel und nennen diese Art der Reflexion spekulare. Bei der idealen und der spekularen Reflexion ist die Position des Betrachters für die Wahrnehmung entscheidend. Je näher dieser zum idealen Reflexionsbereich steht, desto intensiver ist die Wahrnehmung. Bei einem sehr hohen Streuungsgrad wird die Reflexion sehr diffus. Das bedeutet, dass die Position des Betrachters jetzt keine Rolle mehr spielt. Mitte: Materialien können Lichtstrahlen absorbieren, es finden also keine Reflexionen statt. Unten: Je transparenter die Eigenschaft eines Materials, desto mehr Lichtstrahlen dringen hindurch. Auch hier spielt der Grad der Streuung für die Wahrnehmung des Betrachters eine Rolle. Die Eigenschaften treten bei realen Materialien nie einzeln auf, sind also zunächst eher von theoretischem Interesse. Allerdings können wir Kombinationen dieser Eigenschaften wahrnehmen.

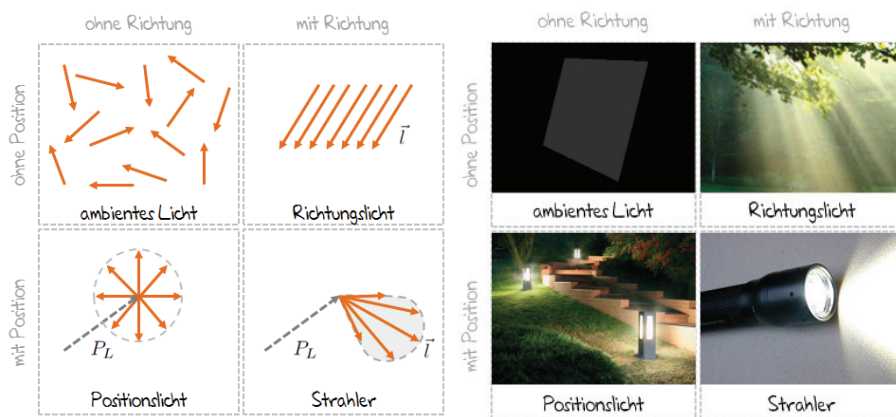


Abbildung 5.6: Links: Ambientes Licht hat weder eine Position noch eine Vorzugsrichtung. Richtungslicht besteht nur aus einer Richtung. Positionslight nur aus einer Position. Ein Strahler vereint sowohl Position als auch Vorzugsrichtung. Rechts: Beispiele zu den entsprechenden Lichtarten. Da ambientes Licht real nicht existiert, wurde ein selbstleuchtendes, rotierendes Viereck als Beispiel gewählt.

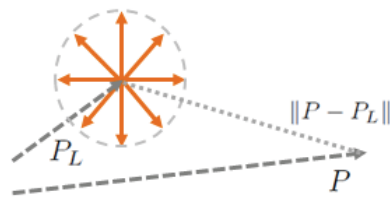


Abbildung 5.7: Das Positionslight wird durch die Position P_L bestimmt. Die Helligkeit nimmt bei größer werdender Entfernung zum Objektpixel an der Position P ab.

betrachten wir in den zwei folgenden Abschnitten.

5.3.1 Positionslight

Neben der Farbe \mathbf{c}_L wird ein Positionslight (engl. *point light*) L , wie es der Name vermuten lässt, maßgeblich durch seine Position P_L bestimmt. Dabei nimmt die Intensität bei größerer Entfernung ab (siehe Abb. 5.7).

Um die Farbe \mathbf{c}_P für eine Position P zu bestimmen, wollen wir in unsere Gleichung der Einfachheit halber eine noch unbekannte Abschwächungsfunktion (*attenuation*) att einführen:

$$\mathbf{c}_P = att(d(P, P_L)) \cdot \mathbf{c}_L$$

Die Funktion att liefert immer Werte kleiner 1 und ist abhängig von der Entfernung (*distance*) d der beiden Punkte L und P . Nehmen wir beispielsweise den euklidischen Abstand über die Länge des Vektors zwischen den zwei Punkten

$$d(P, P_L) = \|P - P_L\|$$

als sehr intuitives und gebräuchliches Entfernungsmaß. Bei größer werdendem d soll $att(d)$ entsprechend kleiner werden (*energy density falloff*). Das erfüllt beispielsweise schon folgende Version:

$$att(d) = \frac{1}{d}$$

Ein linearer Abfall sieht in praktischen Beispielen allerdings nicht schön aus. Daher könnten wir es mit einem quadratischen Abfall versuchen:

$$att(d) = \frac{1}{d^2}$$

Auch hier haben wir Darstellungsprobleme. Bei weit entferntem Licht variiert $att(d)$ kaum und bei dichtem Licht zu viel. Das kann zu unterschiedlichen Beleuchtungen von Objekten bei gleicher Ausrichtung zur Lichtquelle führen (siehe z. B. S. 726 in [6]).

Betrachtung atmosphärischer Einflüsse

Nachdem wir die Objektfarben bestimmt haben, können wir atmosphärische Einflüsse wie beispielsweise Nebel betrachten. Dann werden weiter entfernt liegende Objekte mit weniger Intensität dargestellt. Eine einfache Möglichkeit ist die Festlegung einer Nebelfarbe \mathbf{n} , die in Abhängigkeit zur Entfernung von Betrachter und Objekt eine Abstufung vornimmt mit:

$$\mathbf{c}_L = f \cdot \mathbf{c}_L + (1 - f) \cdot \mathbf{n}$$

Dabei gibt es verschiedene Variationen für f mit linearem, quadratischem oder exponentiellem Verhalten (siehe z. B. [5]).

Variabler Abschwächungskompromiss

Motiviert aus dem Phänomen der atmosphärischen Abschwächung (*atmospheric attenuation*) resultiert folgende Idee: Sagen wir weiterhin, dass Konstanten k_c (für konstanten Entfernungsanteil), k_l (für linearen Entfernungsanteil) und k_q (für quadratischen Entfernungsanteil) als justierbare Parameter existieren, um das Beleuchtungsverhalten über die Entfernung d weiter zu variieren, dann haben wir mit

$$att(d) = \frac{1}{k_c + k_l d + k_q d^2}$$

eine gute, flexible Beschreibung dieses Verhaltens.

Jetzt können bei sehr kleiner Entfernung und kleinen Konstanten, Werte größer als 1 entstehen. Um das zu vermeiden, wollen wir die Abschwächung nach oben mit 1 begrenzen:

$$att(d(P, P_L)) = \min\left(\frac{1}{k_c + k_l d + k_q d^2}, 1\right) \quad (5.1)$$

Eingefügt in unsere ursprüngliche Formel bedeutet das schließlich:

$$\mathbf{c}_P = \min\left(\frac{1}{k_c + k_l d + k_q d^2}, 1\right) \cdot \mathbf{c}_L$$

In Kurzform:

$$\mathbf{c}_P = \overline{\left(\frac{1}{k_c + k_l d + k_q d^2}\right)} \cdot \mathbf{c}_L$$

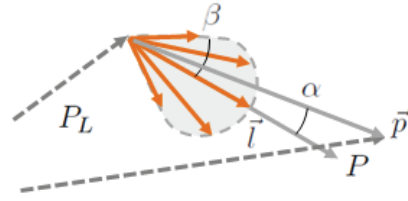


Abbildung 5.8: Ein Strahler L besitzt eine eindeutige Position P_L und eine Vorzugsstrahlrichtung \vec{l} . Der Lichtkegel erstreckt sich in einem Winkel β um \vec{l} . In Abhängigkeit vom Winkel α zwischen der Vorzugsrichtung \vec{l} und der Richtung \vec{p} von P_L zu P wird die Lichtintensität bestimmt.

5.3.2 Strahler

Wie bei den anderen Lichtquellen gibt es auch hier zunächst eine Farbe \mathbf{c}_L . Bei einem Strahler (*spotlight*) kommen weiterhin der Wunsch nach einer Position \mathbf{P}_L und einer Vorzugsrichtung \vec{l} zusammen. Dabei wäre es (außer bei der Simulation einer laserähnlichen Lichtquelle) sicherlich sinnvoll, analog zu einer Taschenlampe einen variablen Lichtkegel anzugeben. Den Öffnungswinkel werden wir mit β angeben (siehe Abb. 5.8).

Hier nimmt wie beim Positionslicht, die Intensität bei größerer Entfernung ab. Allerdings soll in Abhängigkeit zum Winkel α zwischen der Haupttrichtung des Lichts \vec{l} und der direkten Richtung \vec{p} zwischen dem Punkt P und der Lichtposition L eine Abschwächung bei größerem Winkel stattfinden. Liegen beide übereinander, ist also der Winkel $\alpha = 0$, liegt die maximale Intensität vor.

Gesucht ist also eine Strahlerfunktion *spot*, die den Winkel α aus der Vorzugsrichtung \vec{l} und der Richtung \vec{p} bestimmt und daraus in Abhängigkeit zur Abschwächungsfunktion $att(d(P, P_L))$ (siehe dazu vorhergehenden Abschnitt Formel 5.1) ein Maß für die Lichtwirkung erzeugt:

$$\begin{aligned} \mathbf{c}_P &= spot(\vec{l}, \vec{p}) \cdot att(d(P, P_L)) \cdot \mathbf{c}_L \\ &= spot(\vec{l}, \vec{p}) \cdot \min\left(\frac{1}{k_c + k_l d + k_q d^2}, 1\right) \cdot \mathbf{c}_L \\ &= \min\left(\frac{spot(\vec{l}, \vec{p})}{k_c + k_l d + k_q d^2}, 1\right) \cdot \mathbf{c}_L \end{aligned}$$

In der Regel notieren wir Lichtstrahlen immer in Richtung der Lichtquelle, denn bei lokalen Beleuchtungsmodellen entscheiden wir unabhängig für die Pixel, welche Lichtverhältnisse dort existieren und ermitteln dort die Richtungen zu den vorhandenen Lichtquellen. Für die bessere Anschauung hatten wir \vec{l} für den Strahler von der Lichtquelle wegstrahlen lassen. Deshalb müssen wir \vec{p} ebenfalls so konstruieren, dass der Vektor von der Lichtquelle wegzeigt. Das Skalarprodukt von den Einheitsvektoren \vec{l} und \vec{p} liefert uns den Kosinus des Winkels:

$$\cos \alpha = \vec{p} \bullet \vec{l}$$

Damit wir nur positive Werte erlauben, wollen wir 0 als untere Wertegrenze festlegen und haben die erste Version für s :

$$\text{spot}(\vec{l}, \vec{p}) = \max(\vec{p} \bullet \vec{l}, 0)$$

Um das lineare Verhalten des Lichtkegels zu vermeiden und etwas mehr Realismus zu erzeugen, fügen wir einen Strahlungsexponenten sp hinzu, der die Lichtkonzentration um die Hauptrichtung herum erhöht:

$$\text{spot}(\vec{l}, \vec{p}) = \max(\vec{p} \bullet \vec{l}, 0)^{sp}$$

Um den Winkel β jetzt wieder ins Spiel zu bringen und die Wirkung des Strahlers darauf zu beschränken, müssten wir spot eigentlich mit einer Fallunterscheidung versehen:

$$\text{spot}(\vec{p}, \vec{l}, \beta) = \begin{cases} 1 & , \text{ falls kein Strahler (z.B. Positionslicht)} \\ \max(\vec{l} \bullet \vec{p}, 0)^{sp} & , \text{ falls } \vec{p} \text{ innerhalb des Lichtkegels } \beta \\ 0 & , \text{ falls } \vec{p} \text{ ausserhalb des Lichtkegels } \beta \end{cases}$$

Aber da der künstliche Kegelrand bereits durch den Exponenten sp gesteuert wird, können wir unsere Gesamtformel vereinfachen zu:

$$\mathbf{c}_P = \min\left(\frac{\max(\vec{p} \bullet \vec{l}, 0)^{sp}}{k_c + k_l d + k_q d^2}, 1\right) \cdot \mathbf{c}_L \quad (5.2)$$

In Abschnitt 5.1.1 haben wir eine verkürzte Notation eingeführt und schreiben $\vec{p} \bullet \vec{l}$ statt $\max(\vec{p} \bullet \vec{l}, 0)$ und \bar{a} für $\min(a, 1)$:

$$\mathbf{c}_P = \left(\frac{(\vec{p} \bullet \vec{l})^{sp}}{k_c + k_l d + k_q d^2}\right) \cdot \mathbf{c}_L \quad (5.3)$$

5.4 Textur-Mapping

Um die Objekte sichtbar zu machen, werden wir mit einem einfachen Textur-Mapping (*texture mapping*) beginnen. Dazu rendern wir eine Szene ganz ohne Lichtquellen. Es stehen uns lediglich die entsprechenden Texturen zur Verfügung:

$$\mathbf{c}_P = \mathbf{t} \quad (5.4)$$

Im Prinzip entscheidet beim Texture-Mapping ohne Beleuchtung die korrespondierende Farbe \mathbf{t} in einer Textur über den Farbwert an der Position P .

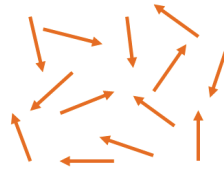


Abbildung 5.9: Ambientes Licht hat weder eine Position noch eine Richtung.

5.5 Elementare Beleuchtungsmodelle

Wir werden Schritt für Schritt die elementaren Beleuchtungsmodelle unabhängig voneinander kennenlernen und sie in den darauf folgenden Abschnitten in komplexeren Beleuchtungsmodellen zusammenführen.

5.5.1 Ambientes Licht

Obwohl das ambiente Licht (*ambient light*) in den Bereich der Lichtmodelle gehört, wollen wir es hier als ein elementares Beleuchtungsmodell betrachten. Ambientes Licht, auch Streulicht genannt, hat keine Position und keine Richtung (siehe Abb. 5.9).

Wie andere Lichtquellen auch verfügt das ambiente Licht über eine Farbe \mathbf{c}_L . Die resultierende Farbe \mathbf{c}_P an einer Position P ist abhängig von den Materialeigenschaften $\mathbf{c}_P^{(A)}$ (*ambient-reflection coefficient*) in Bezug auf ambientes Licht an dieser Stelle:

$$\mathbf{c}_P = \mathbf{c}_P^{(a)} \cdot \mathbf{c}_L$$

So können zwei Objekte mit unterschiedlichen Materialeigenschaften auf das gleiche ambiente Licht unterschiedlich reagieren. Damit wir später die Beleuchtungsmodelle verständlicher kombinieren können, schreiben wir für die Farbe des ambientes Lichts \mathbf{a} statt \mathbf{c}_L und für die Materialreaktion \mathbf{a}_P statt $\mathbf{c}_P^{(a)}$:

$$\mathbf{c}_P = \mathbf{a}_P \cdot \mathbf{a} \quad (5.5)$$

Künstler mögen ambientes Licht nicht, einen Grund dafür sehen wir in Abb. 5.10.

5.5.2 Diffuses Licht

Matte Oberflächen reflektieren einen großen Anteil diffusen Lichts. Dieser Effekt ist auch als Lambert-Reflexion (*lambertian reflectance*) bekannt. Dabei ist der Winkel zwischen der Normalen der Objektoberfläche \vec{n} und der Lichtrichtung \vec{l} entscheidend. Die Position des Betrachters spielt dabei keine Rolle (siehe Abb. 5.11).

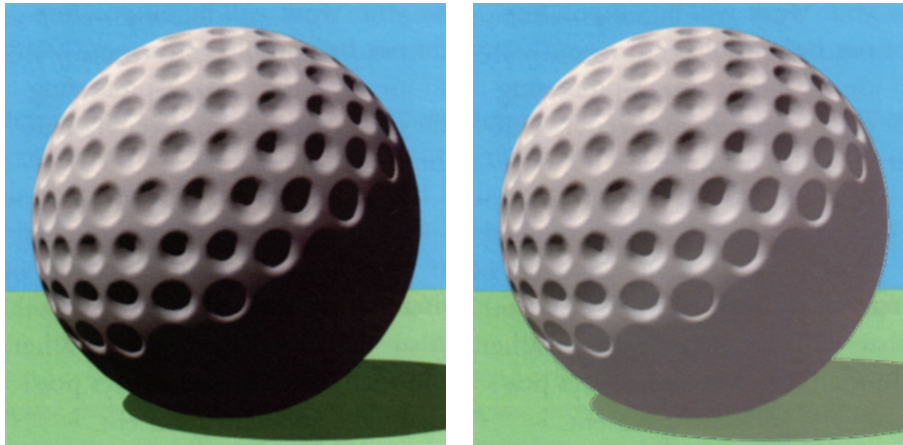


Abbildung 5.10: Wenn wir eine Szene aufhellen wollen, dann führt der Einsatz von ambientem Licht dazu, Schattenbereiche heller erscheinen zu lassen, ohne mehr Struktur zu zeigen (Abbildungen aus [3]).

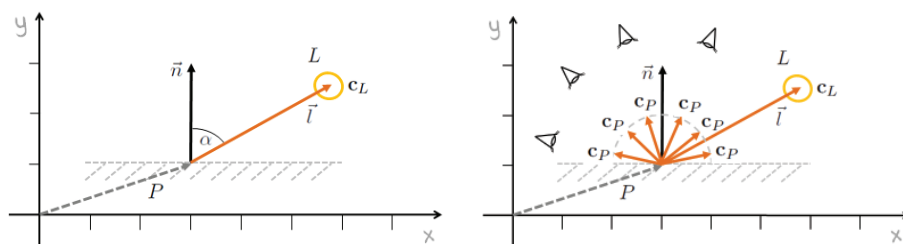


Abbildung 5.11: Der Winkel α zwischen der Oberflächennormalen \vec{n} und der Lichtrichtung \vec{l} (links) entscheidet maßgeblich über die Intensität c_P der diffusen Reflexion an der Position P (rechts). Die verschiedenen Versionen von c_P zeigen, dass die Position des Betrachters für das diffuse Licht keine Rolle spielt. Natürlich wird nur ein c_P pro Position P berechnet.

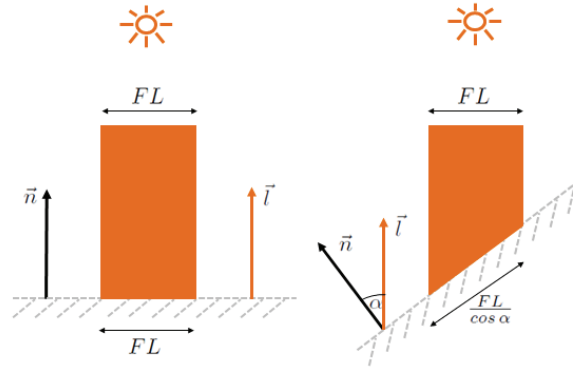


Abbildung 5.12: Links: Lassen wir eine Menge von Lichtstrahlen, die über eine Fläche FL verteilt sind, senkrecht auf eine Oberfläche treffen, so entspricht das der Fläche der auftreffenden Lichtstrahlen. Rechts: Wenn wir die gleiche Lichtmenge über FL auf eine mit dem Winkel α geneigte Oberfläche treffen lassen, so vergrößert sich die Oberfläche zu $\frac{FL}{\cos \alpha}$. Das führt dazu, dass sich die Lichtstrahlen auf eine größere Fläche verteilen müssen, was für den Beobachter die Wahrnehmung einer Abschwächung der Gesamtintensität zur Folge hat.

Im Abschnitt 5.3.2 haben wir uns bereits für die Lichtquelle Strahler eine Version der Funktion *spot* erarbeitet, die für unseren Fall ein Maß in Abhängigkeit zum Winkel zwischen zwei Vektoren \vec{n} und \vec{l} liefert:

$$\text{spot}(\vec{n}, \vec{l}) = \max(\vec{n} \bullet \vec{l}, 0)$$

Dabei müssen \vec{n} und \vec{l} bereits normiert sein. Warum hier der Kosinus verwendet wird, zeigt Abb. 5.12.

Bei kleiner werdendem Winkel α konvergiert $\cos \alpha$ zu 1, was zur Folge hat, dass das einfallende Licht komplett reflektiert wird. Bei größer werdendem α soll $\cos \alpha$ gegen 0 konvergieren. Wir müssen dazu wieder die Funktionswerte am unteren Intervallende auf 0 begrenzen. Das führt uns zu:

$$\mathbf{c}_P = \max(\vec{n} \bullet \vec{l}, 0) \cdot \mathbf{c}_L \quad (5.6)$$

Analog zum ambienten Licht, können wir auch hier wieder unterschiedliche Materialeigenschaften $\mathbf{c}_P^{(d)}$ in Abhängigkeit zu diffusem Licht definieren. Das führt uns zu:

$$\mathbf{c}_P = \mathbf{c}_P^{(d)} \cdot \max(\vec{n} \bullet \vec{l}, 0) \cdot \mathbf{c}_L$$

Damit wir später die Beleuchtungsmodelle verständlicher kombinieren können, schreiben wir für die Farbe des diffusen Lichts \mathbf{d} statt \mathbf{c}_L und für die Materialreaktion darauf \mathbf{d}_P statt $\mathbf{c}_P^{(d)}$:

$$\mathbf{c}_P = \max(\vec{n} \bullet \vec{l}, 0) \cdot \mathbf{d}_P \cdot \mathbf{d} \quad (5.7)$$

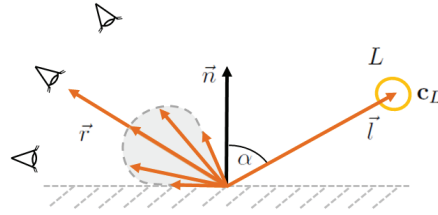


Abbildung 5.13: Analog zum Lichtmodell Strahler haben wir hier eine Vorzugsleuchtrichtung, die sich kegelförmig um den Reflexionsvektor \vec{r} herum aufbaut. Dabei spielt die Position des Betrachters eine wichtige Rolle, denn je größer der Winkel β zwischen Betrachter \vec{v} und \vec{r} , desto abnehmender die Wahrnehmung der Reflexionsintensität.

Auch hier können wir die kürzere Notation (siehe Abschnitt 5.1.1) verwenden:

$$\mathbf{c}_P = \underline{\vec{n}} \bullet \vec{l} \cdot \mathbf{d}_P \cdot \mathbf{d} \quad (5.8)$$

5.5.3 Spiegelndes Licht

Bei spiegelnden oder auch spekularen Oberflächen spielt die Position des Betrachters eine entscheidende Rolle. Trifft ein Lichtstrahl \vec{l} auf eine perfekt spiegelnde Oberfläche, dann wird dieser in Richtung des Reflexionsvektor \vec{r} bei gleichem Winkel α zurückgeworfen. Wenn sich ein Beobachter in Richtung \vec{v} befindet und gilt $norm(\vec{v}) = norm(\vec{r})$, dann erscheint die spiegelnde Reflexion maximal.

Da es keine perfekt spiegelnden Oberflächen in der Realität gibt, werden wir hier analog zur Idee des Strahlers, eine Hauptrichtung (in diesem Fall \vec{r}) angeben und die Intensität in Abhängigkeit zum Winkel β zwischen Beobachter \vec{v} und Reflexionsvektor \vec{r} vornehmen. Erhöht sich der Winkel, dann verringert sich die Intensität (siehe Abb. 5.13).

Jetzt können wir diesen in vergleichbarer Weise zum diffusen Licht (siehe Gl. 5.6) verwenden, aber nehmen den Winkel zwischen \vec{r} und \vec{v} :

$$\mathbf{c}_P = \underline{\vec{r}} \bullet \vec{v} \cdot \mathbf{c}_L$$

Um die Dichte des Lichtkegels zu beeinflussen wird auch hier analog zum Strahler (siehe Abschnitt 5.3.2) ein Exponent (*specular exponent*) verwendet:

$$\mathbf{c}_P = (\underline{\vec{r}} \bullet \vec{v})^s \cdot \mathbf{c}_L$$

Da Materialien den spekularen Anteil unterschiedlich reflektieren, können wir zusätzlich wieder eine Materialeigenschaft $\mathbf{c}_P^{(s)}$ angeben:

$$\mathbf{c}_P = (\underline{\vec{r}} \bullet \vec{v})^s \cdot \mathbf{c}_P^{(s)} \cdot \mathbf{c}_L$$

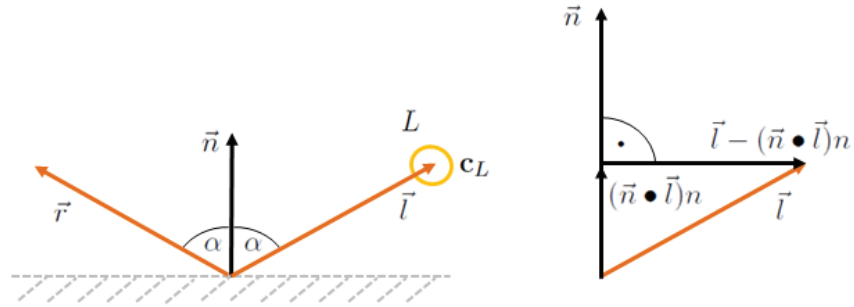


Abbildung 5.14: Wir gehen davon aus, dass \vec{l} und \vec{n} normiert vorliegen. Die Projektion von \vec{l} auf \vec{n} hat die Länge $\cos \alpha$ also $\vec{n} \cdot \vec{l}$ und ergibt somit projiziert auf \vec{n} den Vektor $(\vec{n} \cdot \vec{l})\vec{n}$. Der Vektor von der Spitze des neuen Vektors zu \vec{l} lautet dann $\vec{l} - (\vec{n} \cdot \vec{l})\vec{n}$. Wenn wir diesen Vektor zweimal von \vec{l} abziehen, erhalten wir schließlich den Reflexionsvektor \vec{r} .

Damit wir später die Beleuchtungsmodelle verständlicher kombinieren können, schreiben wir für die Farbe des spekularen Lichts \mathbf{s} statt \mathbf{c}_L und für die Materialreaktion darauf \mathbf{s}_P statt $\mathbf{c}_P^{(s)}$:

$$\mathbf{c}_P = (\vec{r} \cdot \vec{v})^s \cdot \mathbf{s}_P \cdot \mathbf{s} \quad (5.9)$$

Jetzt müssen wir aber erstmal den Reflexionsvektor \vec{r} bestimmen.

Reflexionsvektor bestimmen

Zunächst klingt das Vorhaben einen reflektierenden Vektor für \vec{l} zu bestimmen sehr einfach, denn wir wissen ja, dass der Einfallswinkel gleich dem Ausfallswinkel ist. Allerdings könnten wir unendlich viele Vektoren angeben, die in einem Winkel von α zu einem zweiten Vektor stehen. Was wir konkret suchen ist der Vektor \vec{r} auf der „anderen Seite“ zu \vec{l} . Es müssen also \vec{n} , \vec{l} und \vec{r} in derselben Ebene liegen. Dann gibt es genau zwei Vektoren, die das Kriterium erfüllen und einen kennen wir bereits.

Diese Idee nutzen wir und leiten uns den Reflexionsvektor \vec{r} her (siehe dazu Abb. 5.14).

Daraus ergibt sich der Reflexionsvektor \vec{r} mit den normierten Vektoren \vec{l} und \vec{n} :

$$\begin{aligned} \vec{r} &= \vec{l} - 2 \cdot [\vec{l} - (\vec{n} \cdot \vec{l})\vec{n}] \\ &= \vec{l} - 2\vec{l} + 2(\vec{n} \cdot \vec{l})\vec{n} \\ &= -\vec{l} + 2(\vec{n} \cdot \vec{l})\vec{n} \\ &= 2(\vec{n} \cdot \vec{l})\vec{n} - \vec{l} \end{aligned}$$

Der Reflexionsvektor sollte normiert vorliegen, denn nur seine Richtung ist entscheidend:

$$\vec{r} = \text{norm} \left(2(\vec{n} \cdot \vec{l})\vec{n} - \vec{l} \right)$$

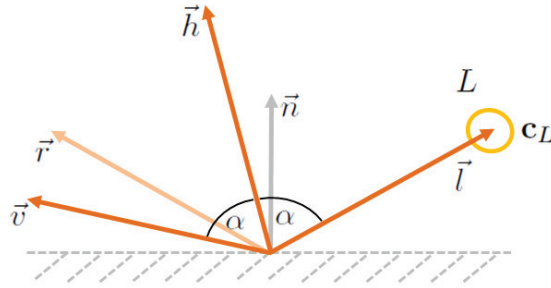


Abbildung 5.15: Statt des Reflexionsvektors in Abhängigkeit von \vec{n} und \vec{l} , wird der Halfway-Vektor als Summe von \vec{v} und \vec{l} bestimmt.

Wir sehen schon, dass es etwas aufwendig ist, den Reflexionsvektor zu bestimmen.

Beschleunigung durch Alternative \vec{h}

Statt des Reflexionsvektors \vec{r} können wir auch einen weniger aufwändigen Vektor \vec{h} berechnen, den sogenannten Halfway-Vektor [10] (siehe Abb. 5.15).

Dabei beziehen wir die Hauptrichtung nicht durch die Vektoren \vec{n} und \vec{l} , sondern verwenden lediglich den mittleren Vektor zwischen \vec{v} und \vec{l} mit:

$$\vec{h} = \text{norm}(\vec{v} + \vec{l})$$

Der Berechnungsaufwand ist hier im Vergleich zum Reflexionsvektor geringer. Die Intensität machen wir jetzt abhängig vom Winkel zwischen \vec{h} und \vec{n} . Damit können wir eine beschleunigte Version der spekularen Reflexion angeben mit:

$$\mathbf{c}_P = \left(\vec{h} \bullet \vec{n} \right)^s \cdot \mathbf{s}_P \cdot \mathbf{s}$$

Beschleunigung durch Fallunterscheidung

Wenn z. B. der Winkel zwischen den beiden Bezugsvektoren größer als 90° ist, haben wir unnötigen Aufwand in die Berechnung gesteckt. Daher sollten wir erstmal das Skalarprodukt befragen, ob beide Vektoren im selben Halbraum liegen. Das bringt uns eine Erweiterung der Gleichung 5.9 auf (siehe [11]):

$$\mathbf{c}_P = \begin{cases} (\vec{r} \bullet \vec{v})^s \cdot \mathbf{s}_P \cdot \mathbf{s} & , \text{wenn } \vec{n} \bullet \vec{l} > 0 \\ 0 & , \text{sonst} \end{cases} \quad (5.10)$$

Die Variante mit dem Halfway-Vektor erweitern wir zu (siehe [10]):

$$\mathbf{c}_P = \begin{cases} \left(\vec{h} \bullet \vec{n} \right)^s \cdot \mathbf{s}_P \cdot \mathbf{s} & , \text{wenn } \vec{n} \bullet \vec{l} > 0 \\ 0 & , \text{sonst} \end{cases} \quad (5.11)$$

Beschleunigung durch Exponentenalternative

Die Hauptrechenlast für die spekulare Reflexion liegt bei den vorgestellten Berechnungen immernoch beim Exponenten. Die Formeln selbst sind gute Annäherungen der Realität, unterliegen aber keinem physikalischen Modell. Daher können wir hier alternative Berechnungsvorschriften einsetzen, die ähnliche Eigenschaften aufweisen.

Es gibt eine Vielzahl von Möglichkeiten, hier sei eine von Christophe Schlick vorgestellt [8]:

$$\begin{aligned} t &= \vec{h} \bullet \vec{n} \\ \mathbf{c}_P &= \frac{t}{s - st + t} \end{aligned}$$

Zunächst wird $\max(\vec{h} \bullet \vec{n}, 0)$ einmal ermittelt und in t zwischengespeichert. Anschließend kommt wieder der Spekularexponent s zum Einsatz.

5.5.4 Emission

Obwohl wir mit dem ambienten Licht den Objekten bereits einen konstanten Lichtanteil (ohne Lichtquelle) spendiert haben, auf die jedes individuell reagieren kann, gibt es Objekte, die eigenständig leuchten. Sie emittieren Licht, geben also aktiv Licht ab. Allerdings hat diese Lichtabgabe keine Auswirkung auf umliegende Objekte.

Typischerweise wird dazu eine emittierende Farbe \mathbf{e} des Objekts angegeben, die in Verbindung mit einer Emissionstextur \mathbf{e}_P einen Farbwert an der Position P erzeugt:

$$\mathbf{c}_P = \mathbf{e} \cdot \mathbf{e}_P$$

Emissionslicht wird beispielsweise in Verbindung mit sichtbaren Lichtquellen verwendet, um das Durchscheinen zu simulieren.

5.6 Phong-Beleuchtungsmodell

Ein oft verwendetes oder erweitertes Beleuchtungsmodell ist das von Bui Tuong Phong [11]. Er schlägt dabei folgende Berechnungsvorschrift für eine Lichtquelle vor:

$$S_P = C_P[\cos(i)(1 - d) + d] + W(i)[\cos(s)]^n$$

Wir haben also:

$$\text{farbe} = \text{diff} + \text{spec}$$

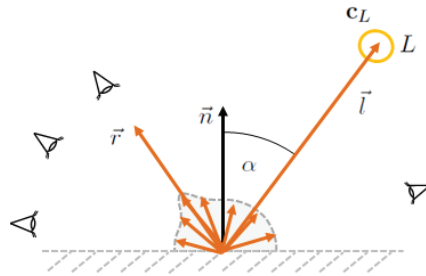


Abbildung 5.16: Beim Beleuchtungsmodell von Phong werden der diffuse und spekulare Anteil kombiniert.

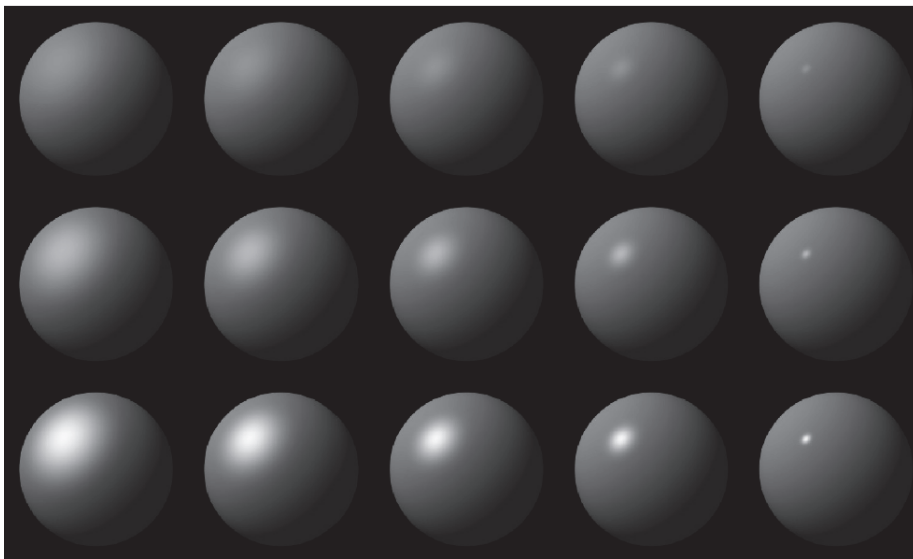


Abbildung 5.17: Beispiele mit unterschiedlichen Werten für s das Phong-Beleuchtungsmodell (Abb. aus [6])

Wir kombinieren also einen diffusen und einen spekularen Lichtanteil (siehe Abb. 5.16), wobei die bereits in den vorhergehenden Abschnitten vorgestellten diffusen und spekularen Reflexionsmodelle noch etwas allgemeiner sind.

Wenn wir das in unsere Notation überführen erhalten wir:

$$\mathbf{c}_P = \mathbf{c}_L [(\vec{n} \cdot \vec{l}) \cdot (1 - \mathbf{d}_P) + \mathbf{d}_P] + s_P \cdot (\vec{r} \cdot \vec{v})^s \quad (5.12)$$

Phongs Beleuchtungsmodell eignet sich gut als Einstieg und Grundlage für komplexere Beleuchtungsmodelle (Beispiele siehe Abb. 5.17).

Wir können das Beleuchtungsschema aufgrund der Linearitätseigenschaften sehr einfach auf n Lichtquellen erweitern:

$$farbe = \sum_{i=1}^n [amb_{L(i)} + diff_{L(i)} + spec_{L(i)}]$$

Das bringt uns schließlich zu folgendem Gesamtausdruck:

$$\mathbf{c}_P = \sum_{i=1}^n \left[\mathbf{a}_P \cdot \mathbf{a}_{(i)} + \vec{n} \bullet \vec{l} \cdot \mathbf{d}_p \cdot \mathbf{d} + (\vec{r} \bullet \vec{v})^s \cdot \mathbf{s}_p \cdot \mathbf{d} \right] \quad (5.13)$$

Wobei hier \mathbf{d} und \mathbf{s} meist identisch sind und über $\mathbf{c}_{L(i)} = \mathbf{s} = \mathbf{d}$ angegeben werden.

5.7 OpenGL-Beleuchtungsmodell

Das in OpenGL verwendete Beleuchtungsmodell basiert auf dem von Phong, verwendet dabei aber den Halfway-Vektor für den spekularen Anteil. So wurden weiterhin folgende Erweiterungen vorgenommen: Emissionsfarbe des Objekts $\mathbf{e} \cdot \mathbf{e}_p$, globaler Einfluss eines ambienten Lichts $\mathbf{a}_P \cdot \mathbf{a}_L$, Abschwächung der Intensitäten bei Positionslicht und Strahler *att*, Strahlerfunktion *spot* zur Manipulation des Lichtkegels.

Schauen wir uns das Beleuchtungsmodell von OpenGL für n Lichtquellen im Detail an:

$$\begin{aligned} \mathbf{c}_{P,L(i)} &= att(\vec{n}, \vec{l}_{L(i)}) \cdot spot(\vec{p}, \vec{l}_{L(i)}, \beta_{L(i)}) \cdot (amb_{L(i)} + diff_{L(i)} + spec_{L(i)}) \\ \mathbf{c}_P &= \mathbf{e} \cdot emi + \mathbf{t} \cdot amb_L + \mathbf{t} \cdot \sum_{i=1}^n \mathbf{c}_{P,L(i)} \end{aligned}$$

Wir setzen entsprechend die bekannten Beleuchtungselemente ein:

$$\begin{aligned} \mathbf{c}_{P,L(i)} &= att(\vec{n}, \vec{l}_{L(i)}) \cdot spot(\vec{p}, \vec{l}_{L(i)}, \beta_{L(i)}) \cdot \\ &\quad ((\mathbf{a}_P \cdot \mathbf{a}_{L(i)}) + [\vec{n} \bullet \vec{l} \cdot \mathbf{d}_p \cdot \mathbf{c}_{L(i)}] + [(\vec{h} \bullet \vec{n})^s \cdot \mathbf{s}_p \cdot \mathbf{c}_{L(i)}]) \\ \mathbf{c}_P &= \mathbf{e} \cdot [\mathbf{e}_p] + \mathbf{t} \cdot [\mathbf{a}_P \cdot \mathbf{a}_L] + \mathbf{t} \cdot \sum_{i=1}^n \mathbf{c}_{P,L(i)} \end{aligned}$$

Das führt uns schließlich zu:

$$\begin{aligned} \mathbf{c}_{P,L(i)} &= att(\vec{n}, \vec{l}_{L(i)}) \cdot spot(\vec{p}, \vec{l}_{L(i)}, \beta_{L(i)}) \cdot \\ &\quad (\mathbf{a}_P \cdot \mathbf{a}_{L(i)} + \vec{n} \bullet \vec{l} \cdot \mathbf{d}_p \cdot \mathbf{c}_{L(i)} + (\vec{h} \bullet \vec{n})^s \cdot \mathbf{s}_p \cdot \mathbf{c}_{L(i)}) \\ \mathbf{c}_P &= \mathbf{e} \cdot \mathbf{e}_p + \mathbf{t} \cdot \mathbf{a}_P \cdot \mathbf{a}_L + \mathbf{t} \cdot \sum_{i=1}^n \mathbf{c}_{P,L(i)} \end{aligned}$$

Leider wird das Phong-Beleuchtungsmodell in vielen Publikationen zur Computergrafik basierend auf der Implementierung in OpenGL vorgestellt, obwohl sich die Darstellung von der Originalarbeit von Bui Tuong Phong sehr unterscheidet [11].

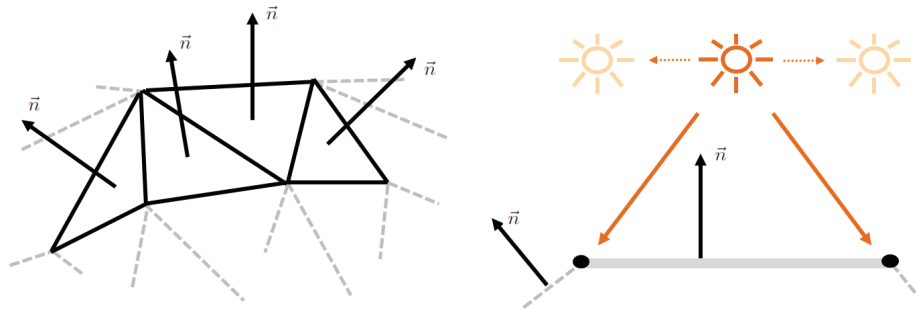


Abbildung 5.18: Links: Beim Flat-Shading verwenden wir eine Normale pro Fläche. Rechts: Wir schauen von der Seite auf das Dreieck und betrachten den Farbverlauf zwischen zwei Knoten. Da es für jeden Punkt nur eine Normale gibt erscheint die Fläche gleichfarbig. Auf die Position der Lichtquelle kann der Betrachter je nach Flächengröße nur grob schließen.

5.8 Schattierungsmodelle

Im vorhergehenden Abschnitt haben wir die Interaktion von Lichtquellen mit einem Oberflächenpunkt P besprochen, von dem über die Normale \vec{n}_P bekannt ist, in welche Richtung die Oberfläche an der Stelle zeigt.

Jetzt vergrößern wir unseren Blick auf die gesamte Fläche (wir nehmen für alle Beispiele das Dreieck) und müssen uns fragen, woher die Punkte mit den Normalen versorgt werden. Dabei gibt es typischerweise drei Ansätze:

- Es gibt eine Normale pro Fläche (Flat-Shading)
- Es gibt eine Normale pro Vertex (Gouraud-Shading)
- Es gibt eine Normale pro Punkt (Phong-Shading)

Fangen wir mit der einfachsten Version an.

5.8.1 Flat-Shading

Beim Flat-Shading (*flat shading*) verwenden wir eine Normale \vec{n} für alle Punkte des Dreieck. Ein Dreieck wird durch drei Knoten (*vertices*) definiert. Wir können beispielsweise die Normale des ersten Knotens verwenden oder die mittlere Normale der drei Knoten.

In Abb. 5.18 sehen wir links, wie sich die Orientierungen der Normalen von Dreieck zu Dreieck unterscheiden können, dabei verfügt jedes Dreieck über eine.

Die Einfärbung einer Fläche in Bezug auf eine Lichtquelle ist überall gleich (siehe Abb. 5.18 rechts).

Wir berechnen also zunächst für die eine Normale in Abhängigkeit vom vorhandenen Beleuchtungsmodell die Lichtintensität an dieser Stelle und reichen dieses Ergebnis dann an alle Punkte weiter.

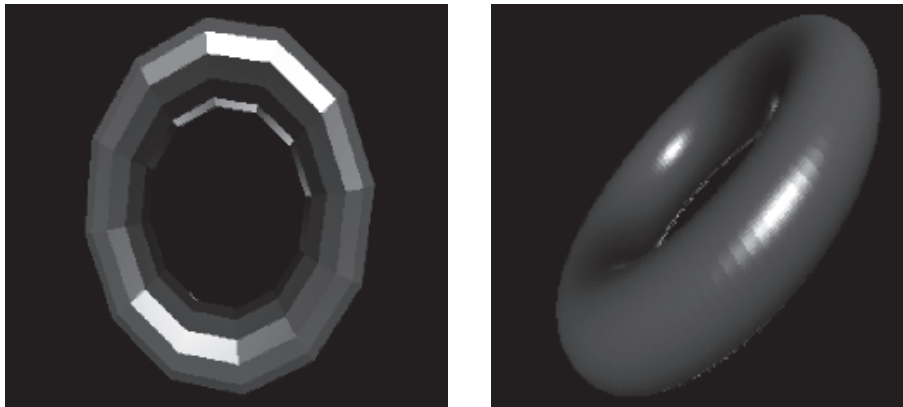


Abbildung 5.19: Wenn wir beispielsweise die Normalen über unseren Torus mit Flat-Shading verteilen, erhalten wir je nach Anzahl der Dreiecke sehr große einfarbige Flächen (wenige links) oder eine bessere Darstellung der Reflexion (viele rechts).

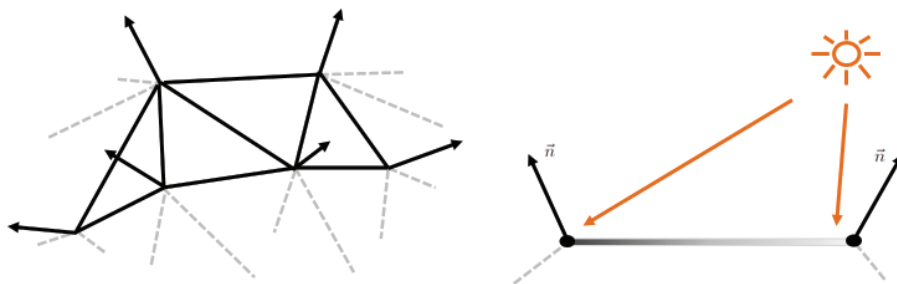


Abbildung 5.20: Links: Beim Gouraud-Shading verwenden wir pro Knoten eine Normale. Rechts: Das Beleuchtungsmodell ermittelt die Lichtintensität für jeden Knoten einmal. Das Ergebnis wird über die Fläche interpoliert und variiert entsprechend die vorliegenden Farbwerte an den Punkten. Auf diese Weise können wir, wie in diesem konkreten Beispiel gezeigt, die Position der Lichtquelle gut unterscheiden.

5.8.2 Gouraud-Shading

Beim Gouraud-Shading (*gouraud shading*) findet die Berechnung der Lichtintensitäten zunächst an allen Knoten statt und wird im Rasterisierungsschritt für jeden Punkt im Dreieck linear interpoliert (siehe Abb. 5.20).

Wenn wir uns die Abbildung auf der rechten Seite nochmal genauer anschauen, erkennen wir allerdings ein kleines Problem. Angenommen, die Lichtquelle läge genau zwischen den Knoten. Wie verhielte sich der Farbverlauf jetzt? Haben wir einen hellen Reflexionsbereich in der Mitte der Fläche?

Nein. Wenn genau der Fall wie im Beispiel vorliegt, dass beide Normalen voneinander weg zeigen, würde auch hier das Dreieck einfarbig vorliegen, denn wenn die Winkel der Normalen zu der Lichtquelle gleich groß sind, werden auch die reflektierenden Intensitäten gleich sein.

Nun entstehen bei der Visualisierung zusammenhängender Dreiecke an gewölb-



Abbildung 5.21: Wenn wir uns die Kanten der Flächen zueinander genau anschauen, wirken die jeweils linken Pixel deutlich dunkler und die rechten deutlich heller. Allerdings wurden die grauen Flächen vollkommen gleichmäßig eingefärbt. Wenn Sie nicht glauben, dass die Flächen jeweils eine gleiche Färbung aufweisen, dann testen Sie das, indem Sie die Seiten mit einem weißen Blatt verdecken.

ten Oberflächen Kanten. Obwohl sich die Intensitäten der Dreieck nur marginal unterscheiden, verstärkt die Art und Weise, wie unsere visuelle Wahrnehmung funktioniert, die vorliegenden Kanten. Das wollen wir uns in dem folgenden Abschnitt kurz anschauen.

5.8.3 Machsche Streifen

Bei den Übergängen von einfarbigen, grauen Flächen lassen sich entlang der Grenzen sogenannte Machsche Streifen als optische Illusion erkennen (siehe Abb. 5.21), dieses Phänomen wurde bereits 1865 von Ernst Mach beschrieben [12].

Es handelt sich dabei um die besondere Eigenschaft unserer visuellen Wahrnehmung aufgrund der Verschaltung unserer lichtempfindlichen Neuronen in der Retina, Kanten besonders hervorzuheben. Das lässt sich hervorragend an einem kleinen Modell demonstrieren.

Schauen wir uns dazu zunächst kurz an, wie wir als Informatiker neuronale Zellen sogenannte Neuronen im Rechner darstellen.

Geschichte der Neuronenmodelle

Um diesen Abschnitt sehr kurz zu halten, werden die Modelle nur kompakt dargestellt. Eine Vertiefung in die Materie bietet beispielsweise der Klassiker „Neuronale Netze“ von Raúl Rojas [7].

Eines der ersten Modelle ist das bereits 1943 von McCulloch Pitts vorgeschlagene, das sich stark am biologischen Modell orientiert (siehe Abb. 5.22).

Wir haben einen Zellkern (*nucleus*) und eingehende excitatorische Verbindungen x_1, x_2, \dots, x_n , die jeweils mit den Werten 0 oder 1 vorliegen können. Die Werte der anliegenden Kanten werden aufsummiert. Wenn genügend Erregung in der Summe vorhanden ist, der Wert also mindestens so groß wie ein Schwellenwert Θ ist, wird über eine Ausgabeleitung (*axon*) ebenfalls eine 1 geliefert. Ansonsten zeigt die Zelle keine Erregung, der Wert der Ausgabeleitung bleibt bei 0.

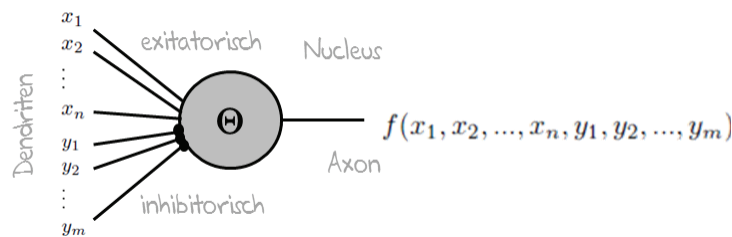


Abbildung 5.22: Es gibt excitatorische und inhibitorische Eingangsverbindungen (Dendriten) und eine Ausgangsverbindung (Axon). Der Zellkern (Nucleus) entscheidet in Abhängigkeit der Belegung der Eingangssignale, ob gefeuert wird oder nicht.

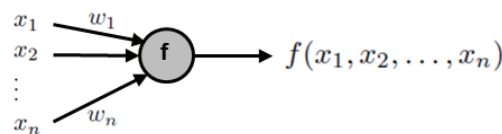


Abbildung 5.23: Das Perzeptron stellt eine Verallgemeinerung der McCulloch-Pitts-Zelle dar. Es kommen Gewichte an den Kanten hinzu.

Neben den excitatorischen gibt es noch inhibitorische Verbindungen. Wir können Sie als Veto-Verbindungen auffassen, denn sollte eine dieser Verbindungen aktiv sein, also 1 liefern, dann bleibt die gesamte Zelle ruhig, liefert also eine 0.

Es kann leicht gezeigt werden, dass damit bereits logische Bausteine realisiert werden können. Das bedeutet, dass Netze aus McCulloch-Pitts-Zellen in Bezug auf Berechenbarkeit der aktuellen Computertechnologie ebenmächtig sind. Allerdings ist das automatische Trainieren der Parameter für ein festes Netz und einer konkreten Problemstellung mit diesem Modell nicht ganz einfach.

Rosenblatt stellte 1958 mit dem Perzeptron eine verallgemeinerte Variante vor (siehe Abb. 5.23).

Die binären Eingaben werden durch reelle Gewichte an den Kanten erweitert. Dabei wird eine Eingabe x_i einfach mit w_i multipliziert, bevor sie am Zellkern eintrifft (siehe Abb. 5.24 links). Um den Effekt der inhibitorischen Kanten zu ermöglichen, können Gewichte mit hohen negativen Werten verwendet werden. Jetzt sind also die Gewichte im Zusammenspiel mit dem Schwellenwert entscheidend für das Verhalten der Zelle.

Um die unbekannt Parameter (Gewichte und Schwellenwert) an einem Ort zu haben, kann eine zusätzliche Kante mit einer konstanten Belegung von 1 und $-\Theta$ als Gewicht hinzugefügt werden (siehe Abb. 5.24 rechts).

Jetzt muss wird im Zellkern mit dem Schwellenwert 0 gerechnet, was vieles vereinfacht. Um die Treppenfunktion durch eine treppenähnliche aber differenzierbare auszutauschen (damit lassen sich später besser automatische Lernverfahren anwenden) wird oft die Sigmoidfunktion eingesetzt:

$$s(x) = \frac{1}{1 + e^{-cx}}$$

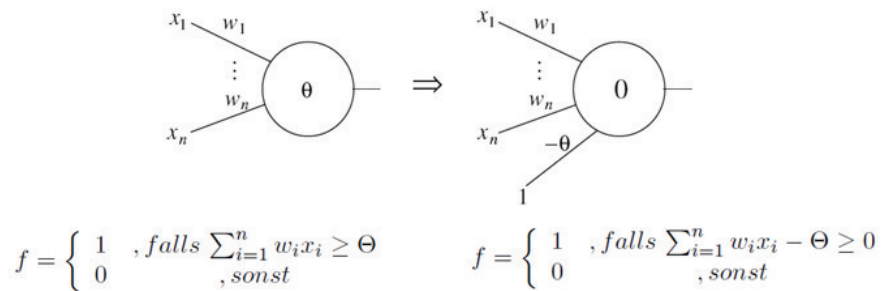


Abbildung 5.24: Links: Das Verhalten des Perzeptrons mit den gewichteten, reellen Eingaben und die entsprechende Ausgabefunktion darunter. Rechts: Das gleiche Verhalten, allerdings mit ausgelagertem Schwellenwert (Abbildungen kombiniert und neu arrangiert aus [7]).

Dabei entspricht x der gewichteten Summe der Eingaben einer Zelle. Die Ableitung dieser Funktion sieht sehr einfach aus und kann effizient ermittelt werden:

$$s'(x) = s(x)(1 - s(x))$$

Jetzt haben wir uns kurz mit dem Zusammenspiel von Ein- und Ausgaben neuronaler Modelle beschäftigt und verstehen damit das folgende Beispiel hoffentlich besser.

Retina mit sechs Rezeptoren

Das in Abb. 5.25 gewählte Beispiel verdeutlicht den Sachverhalt des Machband-Effekts sehr anschaulich.

Wir sehen im oberen Bereich die an den Zellen angelegten Werte 10, 10, 10, 30, 30 und 30. Durch die laterale Hemmung untereinander entsteht ein interessanter Effekt. Bei der Aufsummierung der Intensitäten wird genau beim Übergang von einem helleren in einen dunkleren Bereich beispielsweise, die Kante dadurch verstärkt, dass der hellere Bereich kurz vor der Kante aufgehellt und der Bereich hinter der Kante verdunkelt wird.

Durch die Verschaltung der Neuronen und die gegenseitige laterale Hemmung entsteht ein noch höherer Kontrast, wie die Werte 8, 8, 6, 26, 24 und 24 nachvollziehbar zeigen.

Dieses Phänom zeigt sich bei allen Kanten, aber in diesem kleinen Beispiel lässt sich das gut veranschaulichen.

5.8.4 Phong-Shading

Beim Phong-Shading werden die Normalen selbst durch den Rasterisierungsprozess geschickt [11]. Auf diese Weise erhält jeder zu färbende Punkt eine eigene interpolierte Variante der drei Eckpunktnormalen (siehe Abb. 5.26).

Wenn wir jetzt noch einmal in einem Beispiel die drei Schattierungsverfahren gegenüberstellen, sehen wir deutlich die Qualitätsunterschiede (siehe Abb. 5.27).

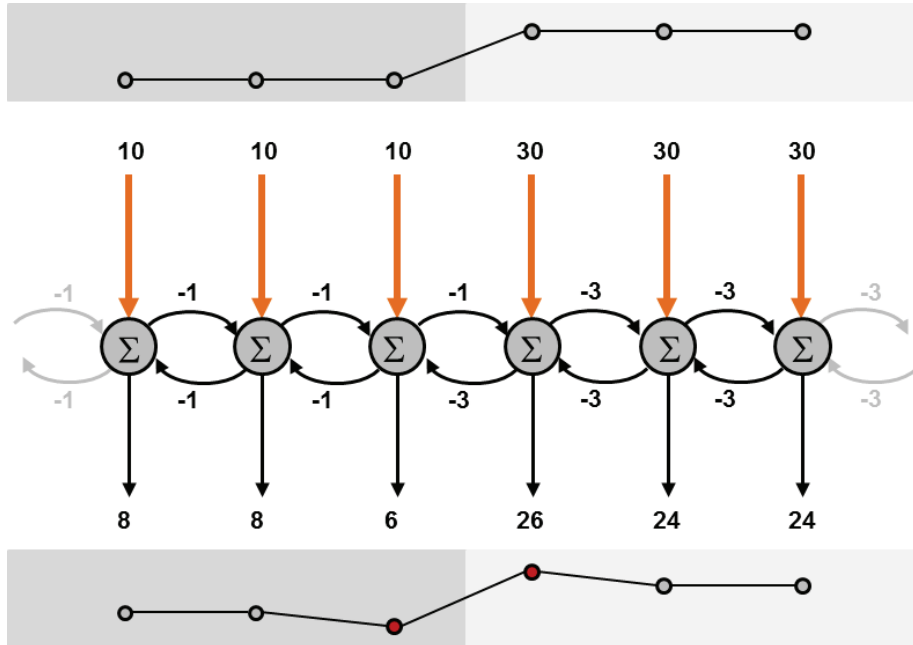


Abbildung 5.25: Oben sehen wir zwei Farbbereiche, von denen wir die Intensität mit jeweils drei Rezeptoren abgreifen. Der Einfachheit halber sind die Werte links auf 10 und rechts auf 30. Die sechs Zellen sind untereinander verschaltet. Sollte eine Zelle feuern (aber das tun ja alle, denn jede Zelle erhält einen Input), dann entscheidet die Intensität des Inputs über die weitergereichte Hemmung. In unserem Beispiel geben wir 10% der Energie hemmend also negiert weiter. Die erste Zelle links beispielsweise erhält den Wert 10 und hemmt damit die linken und rechten Nachbarn jeweils mit -1 . Jetzt müssen wir nur noch die Summe der anliegenden (zum Knoten hinzeigenden Kanten) bilden und ermitteln so den weitergereichten Intensitätswert. Der untere Werteverlauf zeigt parametrisiert unsere Wahrnehmungsverstärkung der Kante.

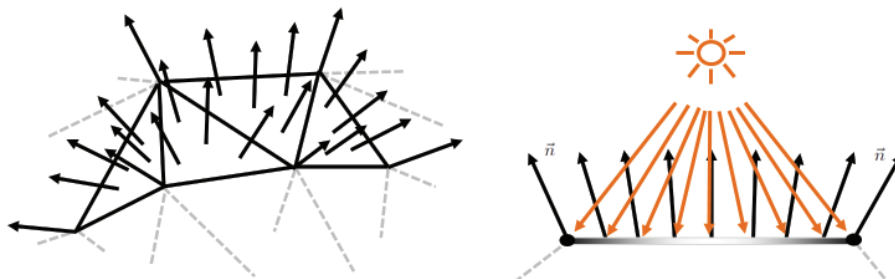


Abbildung 5.26: Links: Beim Phong-Shading werden die Normalen interpoliert. Rechts: Jetzt lassen sich kontinuierliche Reflexionseigenschaften visualisieren.



Abbildung 5.27: Werden die Ergebnisse der Verfahren gegenübergestellt, dann lassen sich die Unterschiede sehr gut erkennen. Mit dem Flat-Shading haben wir zwar das deutlich schnellste Verfahren, da die Beleuchtungsrechnung nur an einer Stelle stattfindet aber damit auch das qualitativ schlechteste. Das Phong-Shading wirkt sehr aufwändig, da wir für jedes Pixel die Beleuchtungsberechnung vornehmen, aber dafür erhalten wir einen sehr viel besseren visuellen Effekt.