Inhaltsverzeichnis - Themen

Weitere Themen: Netze, Fraktale, Animation, ...

6.1 Farbmodelle

- Farbe ist ein wesentlicher Bestandteil realistischer Computergraphik
- **Quantitative Aspekte** von Farbe sind wichtig zur Verarbeitung von Farbinformationen während des Renderings
- Fragestellungen:
  - Wie kann man eine bestimmte Farbe **exakt spezifizieren**?
    ⇒ Farbräume, Farbmodelle
  - **Wie viele verschiedene** Farben können durch eine Graphikhardware spezifiziert werden?
    ⇒ technische Realisierbarkeit
  - **Wie exakt** kann eine exakt spezifizierte Farbe auf einem Ausgabegerät angezeigt werden?
    ⇒ geräteabhängige Farbräume

6.1 Farbmodelle

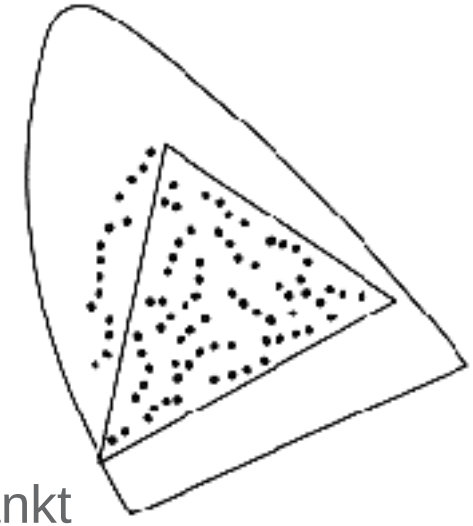
Hierarchien bekannter Farbmengen

A: Menge aller vom Menschen **wahrnehmbaren Farben**

B: Menge aller von einem Ausgabegerät (z. B. Monitor) **darstellbaren Farben** – dies ist eine Untermenge von A

C: Menge aller von einem Programm **spezifizierbaren Farben** – beschränkt durch die Graphikhardware (Bildspeicher)
24 Bit/Pixel $\Rightarrow$ 16777216 Farben
– i. A. Untermenge von A und Obermenge von B

Ebener Schnitt durch den dreidimensionalen Farbraum. Jeder Punkt entspricht einer Farbe.



Dreidimensionaler Farbraum

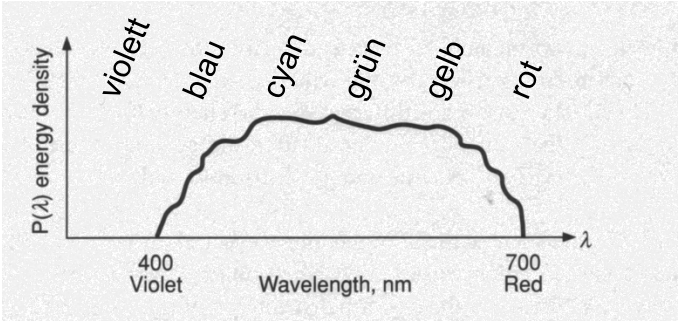
- Traditionelle Beschreibung von Farben durch Namen **ist ungenau** – Aschgrau, Steingrau, Mausgrau, ...
- Präzise Festlegung erst durch **objektive, quantitative** Spezifikation.
- Physikalisch ist Farbe **Energieverteilung im elektromagnetischen** Spektrum zwischen 400 und 700 Nanometern Wellenlänge
- **Physikalische Experimente** und **physiologische Untersuchungen** der Farbwahrnehmung durch das menschliche Auge:
  - Nahezu alle Farben, die das Auge unterscheiden kann, können auf eine **additive Mischung dreier Grundfarben** zurückgeführt werden

6.1 Farbmodelle

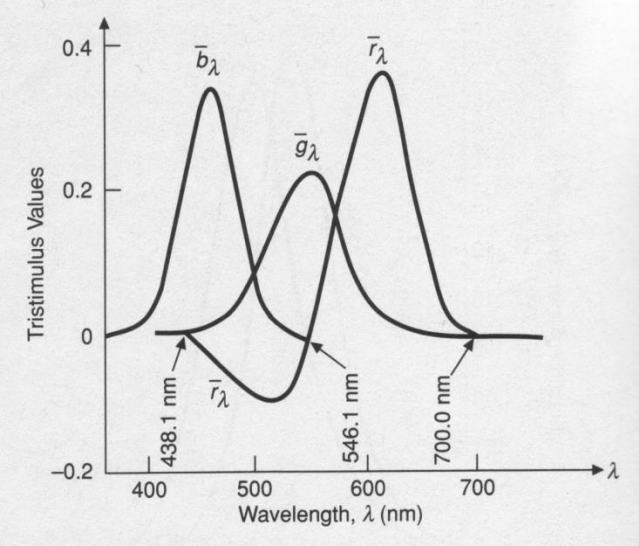
Dreidimensionaler Farbraum (Fortsetzung)

- Definition der Farbe C durch mischenden Anteile zB. der **Grundfarben** Rot, Grün und Blau (R,G,B) und **Gewichtsfaktoren** (r, g, b) von :
  $C = r\,R + g\,G + b\,B$
- Dies ist jedoch bei weitem **nicht die einzige Möglichkeit** zur Definition eines dreidimensionalen Farbraums
- Je nach Anforderung können **verschiedene standardisierte Farbräume** (Farbmodelle) verwendet werden, zB.
  - RGB: **traditioneller** Farbraum für Computergraphik, Monitore, ..
  - HSV: erleichtert die **intuitive** Farbauswahl
  - CIE: **internationaler** Standard zur Farbspezifikation
  - CMY: **subtraktives** Farbmodell für die Drucktechnik

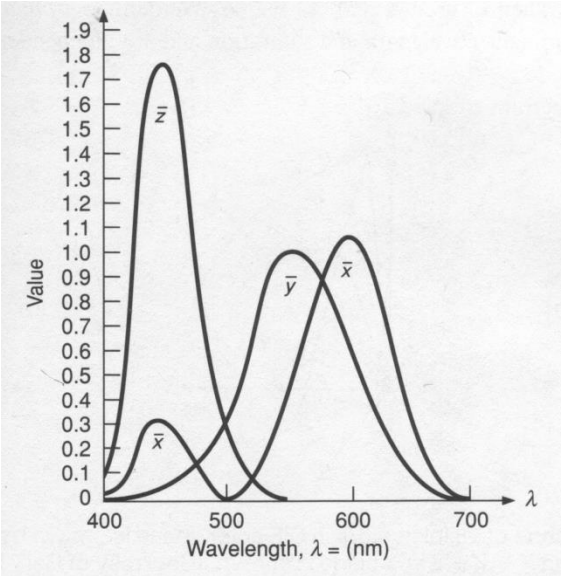
6.1 Farbmodelle



Sichtbarer Frequenzbereich



RGB Anteile zum „Simulieren“ einzelner Wellenlängen

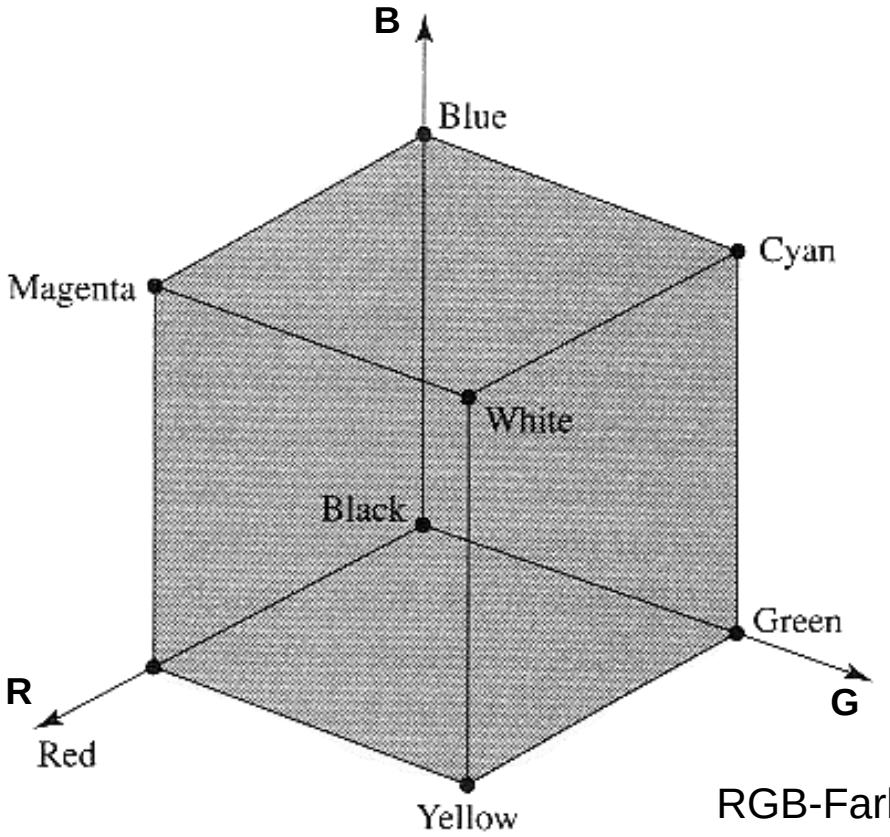


XYZ Anteile (CIE Basis)

RGB-Farbmodell

- Verwendet die Grundfarben **Rot, Grün und Blau** zur additiven Farbmischung
- Beschreibung einer Farbe durch Gewichtsfaktoren (r, g, b) mit $0 \leq r,g,b \leq 1$. Es gilt:
  - (0, 0, 0) = Schwarz (1, 1, 1) = Weiß
  - (1, 0, 0) = Rot (0, 1, 0) = Grün (0, 0, 1) = Blau
  - (0, 1, 1) = Cyan (1, 0, 1) = Magenta (1, 1, 0) = Gelb
- Im Rechner: zB. **8 Bit pro Grundfarbe**, dh. $0 \leq r,g,b \leq 255$.
- Die Menge aller spezifizierbaren Farben wird im 3D-Raum durch **einen Würfel** repräsentiert („Farbkörper“).
- Würfel **deckt jedoch nicht** den gesamten wahrnehmbaren Farbraum ab.

RGB-Farbmodell (Fortsetzung)



RGB-Farbwürfel

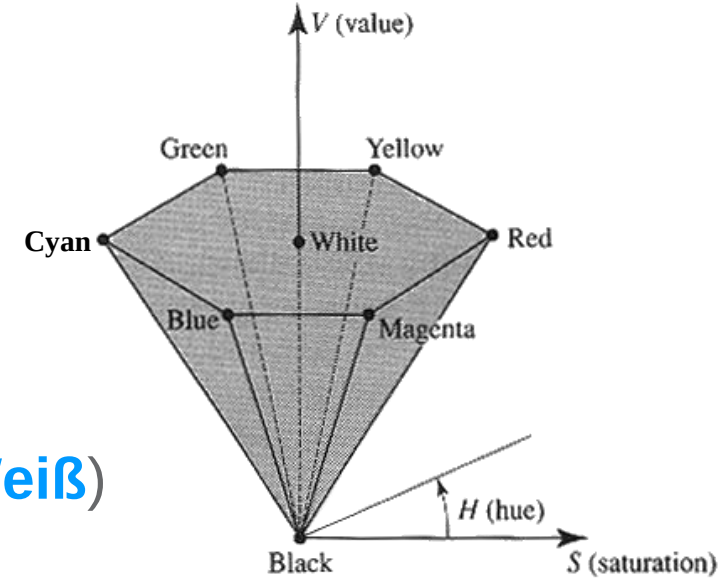
RGB-Farbmodell (Fortsetzung)

- Das Modell ist bezüglich der Farbwahrnehmung **nicht linear**
- Bei typischer Farbauflösung von 8 Bit pro Grundfarbe (so genanntes TrueColor)
  - Es existieren im Farbwürfel Regionen, in denen benachbarte Punkte für das Auge **denselben Farbeindruck** hervorrufen
  - In anderen Regionen sind die Farben benachbarter Punkte für das Auge **sehr unterschiedlich**
- Für den Anwender kann es schwierig sein
  - zu einer gewünschten Farbe (zB. Kastanienbraun) ein entsprechendes (r, g, b)**-Tupel** zu ermitteln
  - Eine Farbe etwas **abzuschwächen** (erfordert ungleichmäßige Änderungen von r, g und b)

$\Rightarrow$ **HSV-Farbmodell**

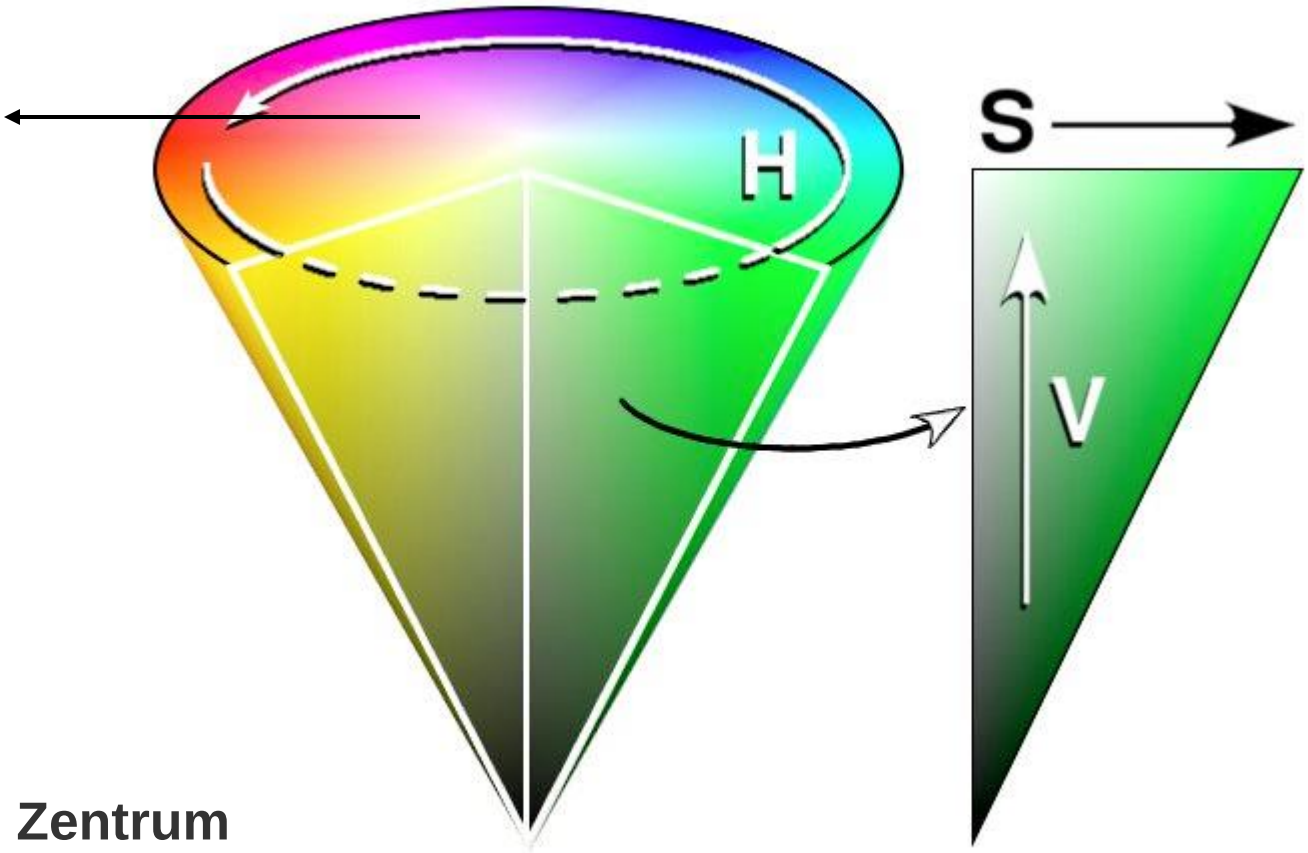
HSV-Farbmodell

- entwickelt zur Unterstützung einer **intuitiven** Farbauswahl (wahrnehmungsorientiertes Farbmodell)
- Farbkörper im 3D-Farbraum ist **Pyramide** mit sechseckiger Grundfläche
- verwendet Polarkoordinaten:
  - Hue: Farbe („Farbfamilie“) als „**Farbwinkel**“ in Grad: $0^\circ \leq H < 360^\circ$
  - Saturation: Sättigung: $0 \leq S \leq 1$ (Verkleinerung **addiert Weiß**)
  - Value: Helligkeit: $0 \leq V \leq 1$ (Verkleinerung **addiert Schwarz**)



HSV-Farbmodell

- H Hue
- S Sättigung
- V Helligkeit



Achtung: Hue im Zentrum gespiegelt

Zusammenhang zwischen HSV- und RGB-Modell

- Die **Grundfläche** der HSV-Pyramide entsteht aus dem RGB-Würfel durch **Projektion entlang der Raumdiagonale** von Weiß nach Schwarz auf eine dazu senkrecht stehende Ebene.
- Korrespondierende Punkte:

RGB	Farbe	HSV
(1, 0, 0)	Rot	(0, 1, 1)
(1, 1, 0)	Gelb	(60, 1, 1)
(0, 1, 0)	Grün	(120, 1, 1)
(0, 1, 1)	Cyan	(180, 1, 1)
(0, 0, 1)	Blau	(240, 1, 1)
(1, 0, 1)	Magenta	(300, 1, 1)

- Im HSV-Modell besitzen **Komplementärfarben** eine Winkeldifferenz von 180° im H-Wert

Zusammenhang HSV- und RGB-Modell (Fortsetzung)

- **Für jeden Punkt P der Hauptdiagonale** im RGB-Würfel **von Schwarz nach Weiß**, ergibt sich ein Teilwürfel wie folgt
  - Hauptdiagonalen der Würfel fallen zusammen
  - **Eine Ecke** des Teilwürfels liegt bei Schwarz = (0, 0, 0) , die **gegenüberliegende Ecke** liegt bei P
  - der Teilwürfel liegt **im Innern** des RGB-Würfels
- Jeder Teilwürfel ergibt bei Anwendung der vorher **beschriebenen Projektion** entlang der Hauptdiagonale ein Sechseck
- Entspricht einem Schnitt durch die HSV-Pyramide für V = const.
- Die **Hauptdiagonale des RGB-Würfels** entspricht also der V-Achse der HSV-Pyramide

Zusammenhang HSV- und RGB-Modell (Fortsetzung)

HSV/RGB-Transformation

$h_i = \lfloor H/60 \rfloor \bmod 6$ // Farbe

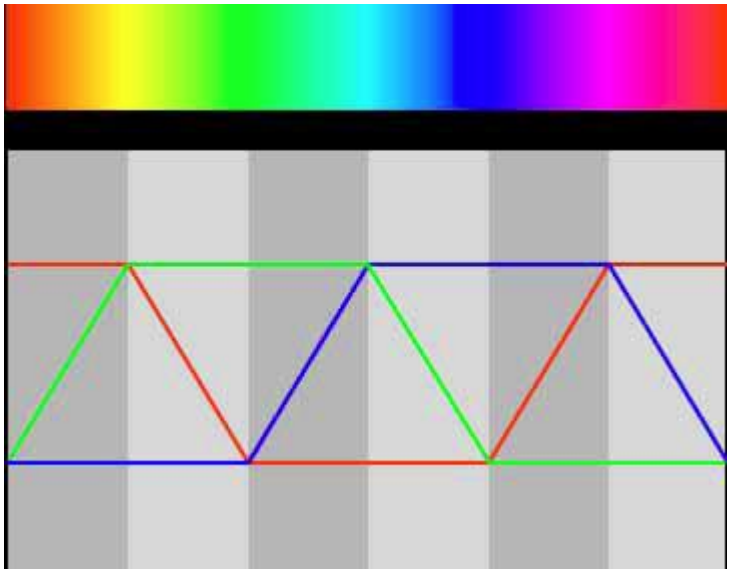
$f = H/60 - \lfloor H/60 \rfloor$ // Feinabstimmung

$p = V \cdot (1 - S)$

$q = V \cdot (1 - f \cdot S)$

$t = V \cdot (1 - (1 - f) \cdot S)$

$$(r,g,b) = \begin{cases} (v,t,p), & \text{wenn } h_i = 0 \\ (q,v,p), & \text{wenn } h_i = 1 \\ (p,v,t), & \text{wenn } h_i = 2 \\ (p,q,v), & \text{wenn } h_i = 3 \\ (t,p,v), & \text{wenn } h_i = 4 \\ (v,p,q), & \text{wenn } h_i = 6 \end{cases}$$

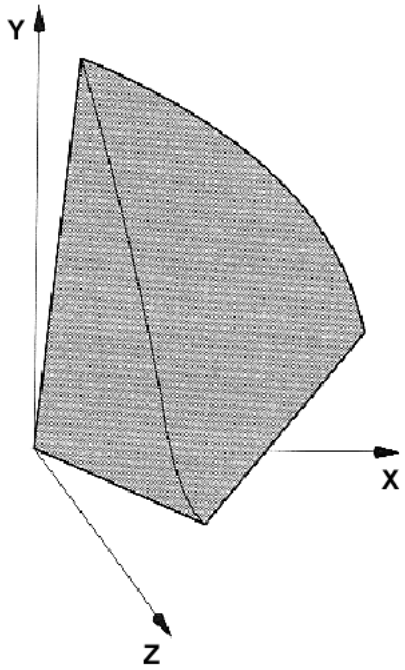


6.1 Farbmodelle

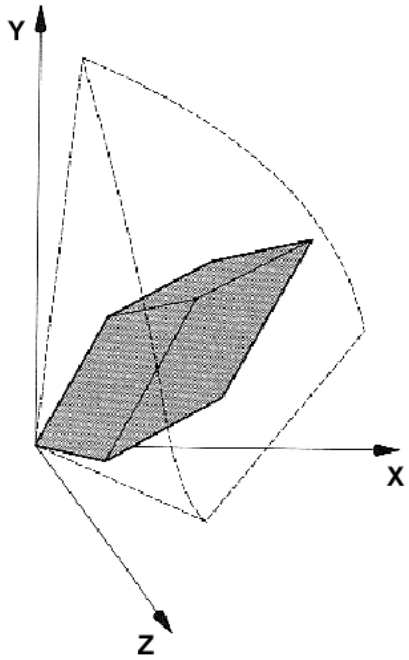
CIE-Farbraum (Commission Internationale de l´Éclairage, 1931)

- Internationaler, **geräteunabhängiger Standard** zur Farbspezifikation
- Geeignet zur Beschreibung **aller vom Menschen wahrnehmbaren** Farben (RGB-Farbkörper ist nicht geeignet)
- Universeller Farbraum, verwendet die **künstlichen Grundfarben** X, Y und Z zur additiven Farbmischung (CIE XYZ-Farbraum)
  - Da **keine Auswahl** dreier Grundfarben aus dem sichtbaren Farbbereich durch additive Mischung mit **nicht negativen Gewichten** alle wahrnehmbaren Farben abdecken kann.
  - Mischung schon zweier Grundfarben ergibt immer eine **weniger gesättigte** Farbe
- Repräsentation einer Farbe C durch $C = X\,\mathbf{X} + Y\,\mathbf{Y} + Z\,\mathbf{Z}$

CIE-Farbraum (Fortsetzung)



CIE XYZ Farbkörper: enthält alle wahrnehmbaren Farben



von einem Monitor darstellbare Farben

CIE-Farbraum (Fortsetzung)

- Eine **alternative Spezifikation** des CIE XYZ-Farbtripels (X, Y, Z) ergibt sich durch eine Abbildung (X, Y, Z) $\Rightarrow$ (x, y, Y) mit

$$x = \frac{X}{X+Y+Z} \quad und \quad y = \frac{Y}{X+Y+Z}.$$

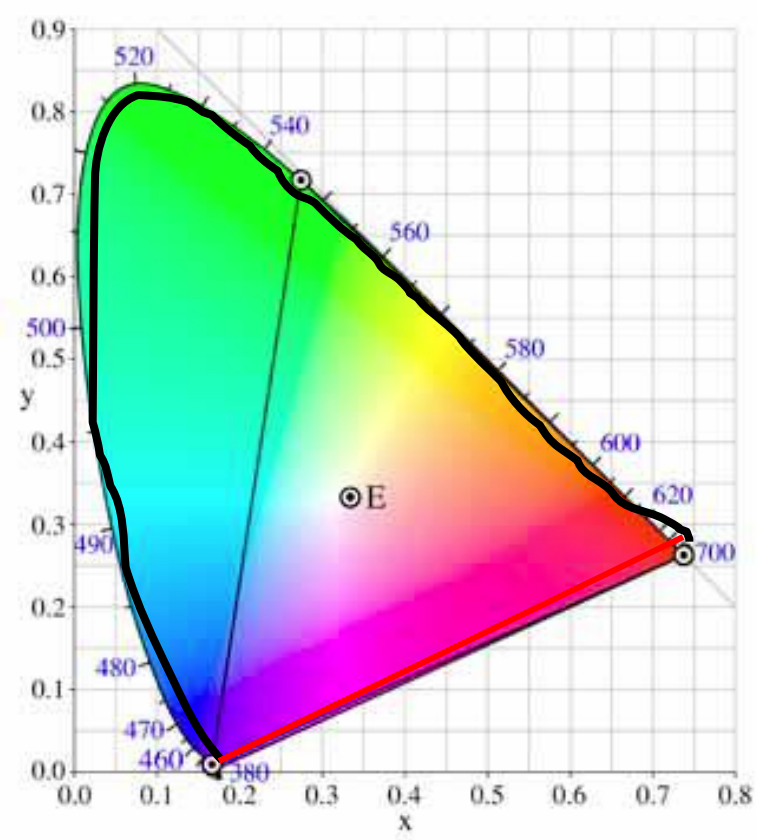
(CIE xyY-Farbraum)

- Wertet man die **Gleichungen für alle Farben des XYZ-Farbkörpers** im (x, y)-Diagramm aus, so erhält man das hufeisenförmige **CIE-Diagramm der Chromatizität**.

6.1 Farbmodelle

CIE-Farbraum (Fortsetzung)

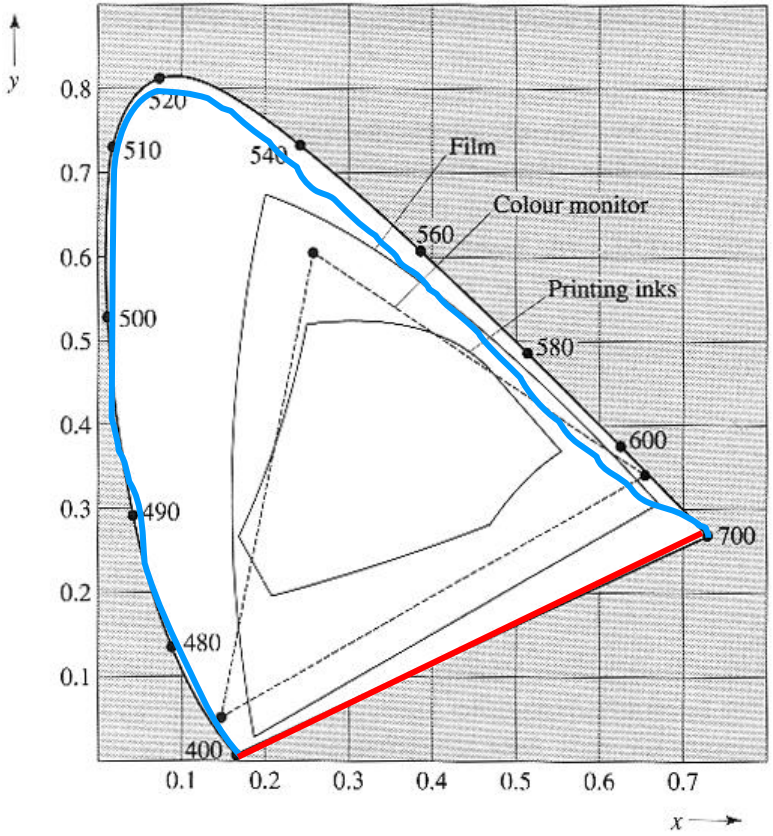
- Das (x, y)-Diagramm enthält in der **zweidimensionalen Projektion** alle sichtbaren Farben
- Luminanz-Anteil wird **ignoriert**
- Auf dem **äußeren Rand** der Hufeisenform liegen die reinen Spektralfarben von Blau (400 nm) bis Rot (700 nm).
- Auf der **Geraden zwischen Blau und Rot** befinden sich die Lila- und Magenta-Farben.
- E ist **Weißpunkt**



6.1 Farbmodelle

CIE-Farbraum (Fortsetzung)

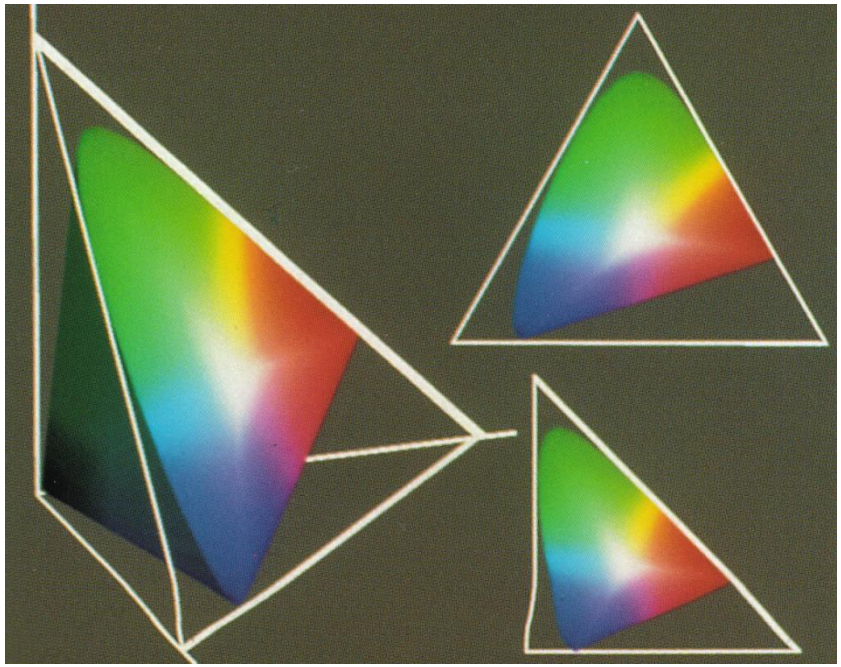
- Das (x, y)-Diagramm enthält in der **zweidimensionalen Projektion** alle sichtbaren Farben
- Luminanz-Anteil wird **ignoriert**
- Auf dem **äußeren Rand** der Hufeisenform liegen die reinen Spektralfarben von Blau (400 nm) bis Rot (700 nm).
- Auf der **Geraden zwischen Blau und Rot** befinden sich die Lila- und Magenta-Farben.

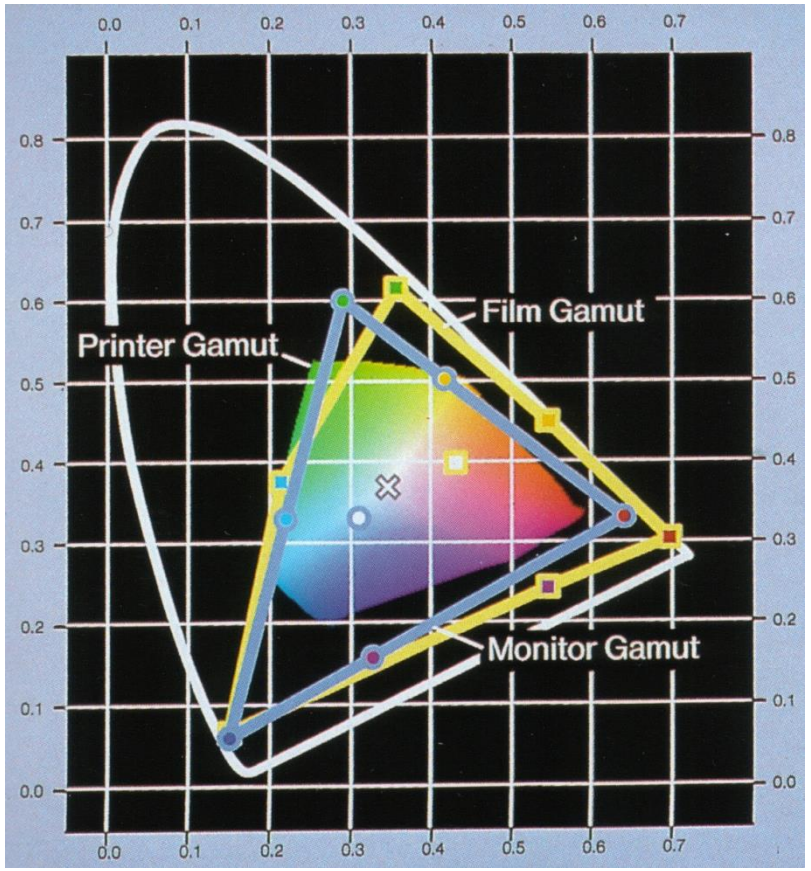


CIE-Farbraum (Fortsetzung)

- **Aus CIE XYZ-Modell** entwickelt (1976)
- L*a*b*-Farbraum mit
  - a*: **Grün/Rot**-Achse [-150,+100]
  - b*: **Blau/Gelb**-Achse [-100,+150]
  - L*: **Luminanz** [0,100]
- Im Gegensatz zu XYZ
  - **Isometrisch** (Gleichabständig)
  - Grünbereich etwas kleiner
  - Purpur-Blau-Cyan etwas größer

CIE-Farbraum





Cyan-Magenta-Yellow-(CMY)-Farbraum

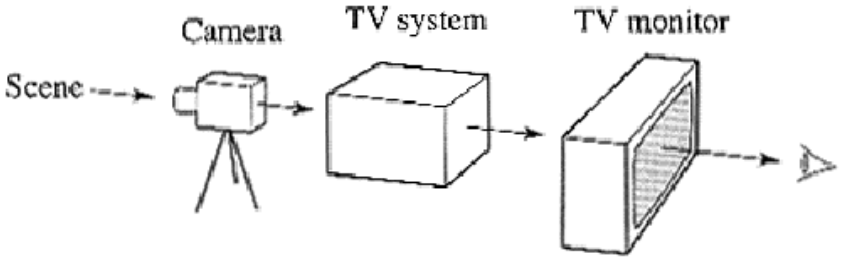
- Drucker nutzen **subtraktive** Farbmodelle
- Die zu RGB komplementären Farben **cyan, magenta, und gelb** werden am häufigsten eingesetzt:

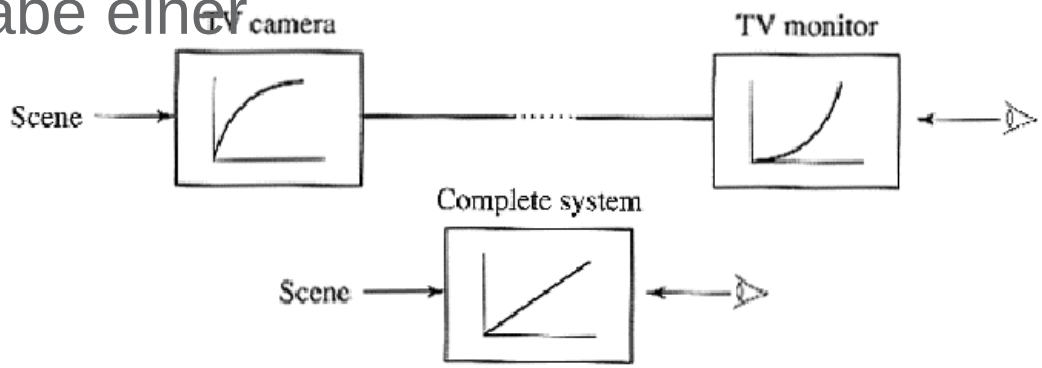
$$\begin{pmatrix} C \\ M \\ Y \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

- Oft kommt es bei der Farbwiedergabe auf Papier zu **kleineren Differenzen**
- Drucker verwenden häufig noch **zusätzliche Farben**, zB. Schwarz

Gamma-Korrektur

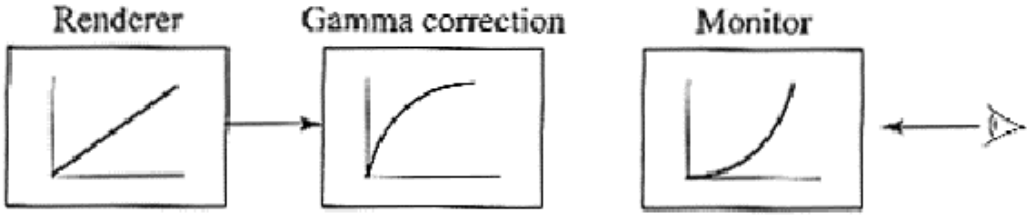
- Zuschauer erwarten **gleiche Bildqualität** auf TV wie in der Realität
- In Fernsehkamera wird bereits eine **Präkompensation** vorgenommen
- **Korrigiert die Nichtlinearität** der Intensitätswiedergabe einer üblichen Fernsehröhre





Gamma-Korrektur (Fortsetzung)

- In der Computergraphik übernimmt ein **Renderer die Rolle der Kamera**
- Er besitzt generell eine **lineare Intensitätscharakteristik**
- **Gamma-Korrektur notwendig**, da Monitor ähnliche Nichtlinearität bei der Helligkeitswiedergabe aufweist wie TV
- **Lookup-Tabelle für neue Werte** im RGB-Modell für jeden Farbkanal (bei 8 Bit/Grundfarbe: drei Arrays der Länge 256).



Inhaltsverzeichnis - Themen

Weitere Themen: Netze, Fraktale, Animation, ...

6.2 Beleuchtung und Schattierung

- Die Bestimmung der Intensität (Farbe) derjenigen Pixel - auf die ein Objekt (zB. ein Polygon) projiziert wird - durch so genannte **Beleuchtungs-, Reflexions- und Schattierungsalgorithmen** bzw. -modelle
- **Inkonsistente** Terminologie in der Literatur
  - Beleuchtungsmodell = Illumination-Model, Lighting-Model, Reflection-Model
  - NICHT: Shading-Model

Beleuchtungsmodell (Illumination-Model, Fortsetzung)

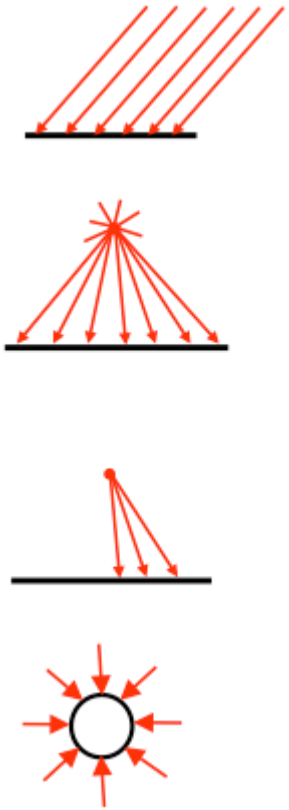
- **Lokal** ~ (nur direkte Beleuchtung)
  - Berechnung der Intensität (Farbe) eines Punktes in Abhängigkeit von **direktem Lichteinfall einer Lichtquelle**
    ⇒ z. B. Phong local reflection model,
    physikalisch **basierte** Modelle (nicht physik-basiert)

Beleuchtungsmodell (Illumination-Model, Fortsetzung)

- **Global** ~ direkte und indirekte Beleuchtung
  Berechnung der Intensität (Farbe) eines Punktes in Abhängigkeit von
  - **direktem Lichteinfall einer Lichtquelle** und
  - **indirekt einfallendem Licht**,
- Berechnung **Reflexion(en) an oder Transmission(en)** durch die eigene oder andere Oberflächen
  ⇒ zB. RayTracing-Verfahren, Radiosity-Verfahren
- Globale Beleuchtungsverfahren benutzen oft **erweiterte lokale Beleuchtungsmodelle**

Beleuchtungsmodell (Illumination-Model, Fortsetzung) - Lichtquellen

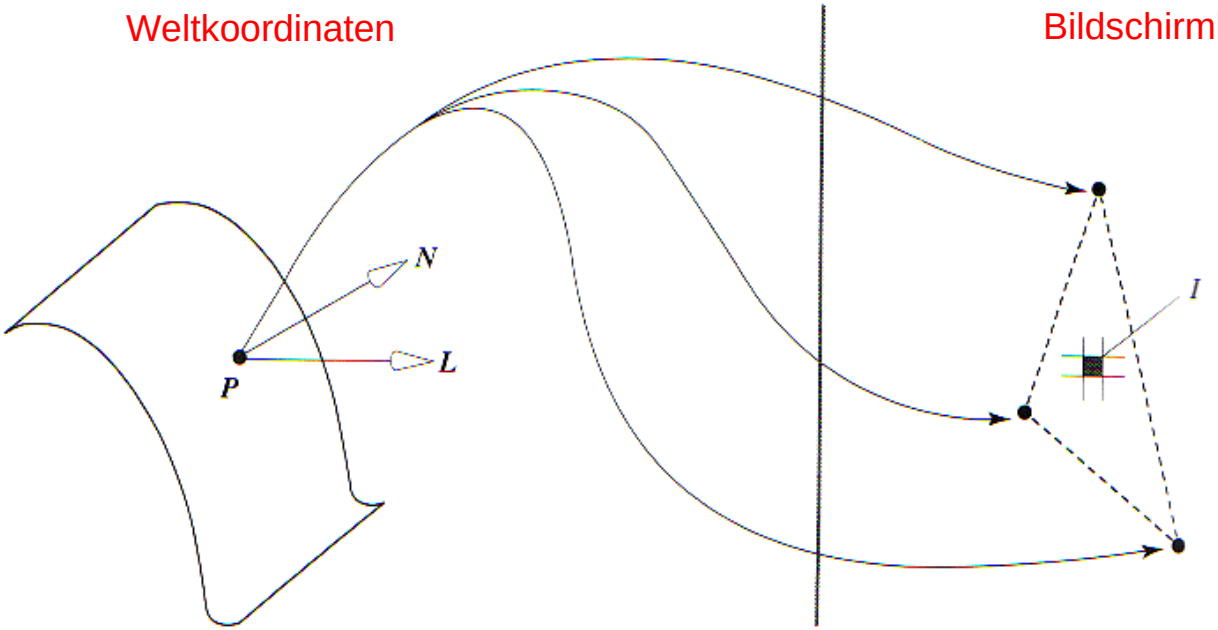
- **Gerichtete** Lichtquellen -
  Position im Unendlichen $(x,y,z,0)^T$
- **Punkt**lichtquellen -
  Endliche Position $(x,y,z,1)^T$
- **Spot**lichtquellen -
  Cutoff-Winkel, Position, Richtung
- **Ambientes** Licht



Schattierungsmodell (Shading-Model)

- Grundstruktur **bettet** Beleuchtungsmodell ein
- Ein Schattierungsmodell bestimmt, **wann ein Beleuchtungsmodell angewendet wird**, z. B.
  - Auswertung eines Beleuchtungsmodells **für jedes (Sub-)Pixel**
    ⇒ z. B. oft bei RayTracing-Verfahren angewendet
- Oder
  - Auswertung eines Beleuchtungsmodells für **ausgewählte Pixel/Punkte: Stützpunkte**
  - Farben von „Zwischenpixel“ werden **per Interpolation** bestimmt
    ⇒ interpolative Schattierung, z. B. flat/uniform Shading, Gouraud-Shading, Phong-Shading

Gängige Praxis-Kombination



Lokales Reflektionsmodell:
berechnet die Lichtintensität an **jedem Punkt P** auf der Oberfläche eines Objektes

Interpolative Schattierung:
interpoliert Pixelintensitäten I aus berechneten **Lichtintensitäten in den Polygonecken**

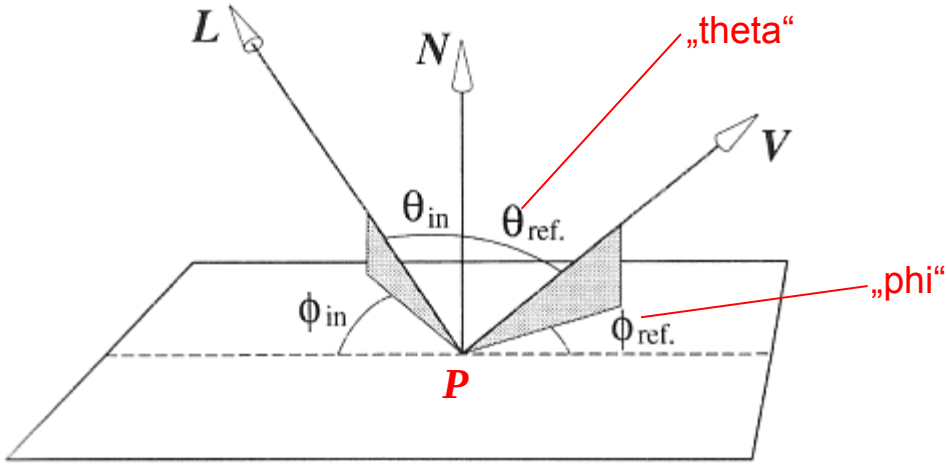
Gängige Praxis-Kombination: (Fortsetzung)

- Gibt es hier kein Problem?
  - Beleuchtung (und Betrachtung) der Szene erfolgt in **Weltkoordinaten**
  - Interpolation zwischen Intensitätswerten erfolgt in **Bildschirmkoordinaten**
  - Zentralprojektionen sind in der Regel **nicht affin.**

$\Rightarrow$ wir verwenden beim Interpolationsschema (z. B. linearer Interpolation) **automatisch „falsche“ Teilverhältnisse** in Bezug auf das Weltkoordinatensystem!

- Trotz **mathematischer Inkorrektheit** liefert diese Kombination schnelle und akzeptable visuelle Resultate und wird so verwendet.

Geometriebetrachtung



P Punkt auf Objektoberfläche

N Flächennormalenvektor in P, normiert

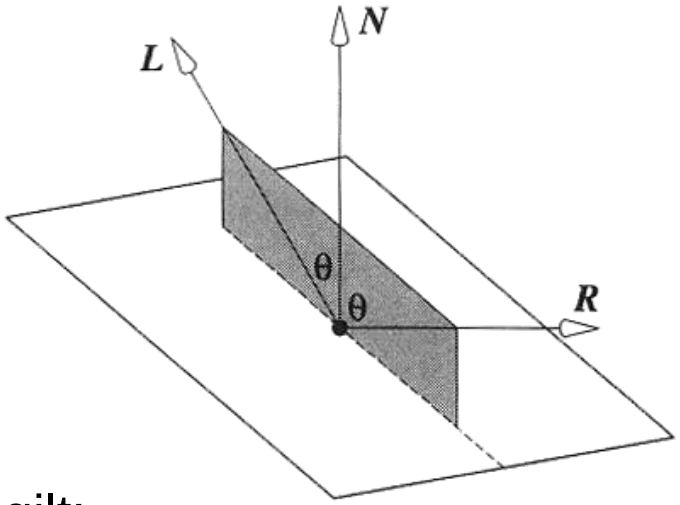
L Vektor von P zu einer Punktlichtquelle, normiert

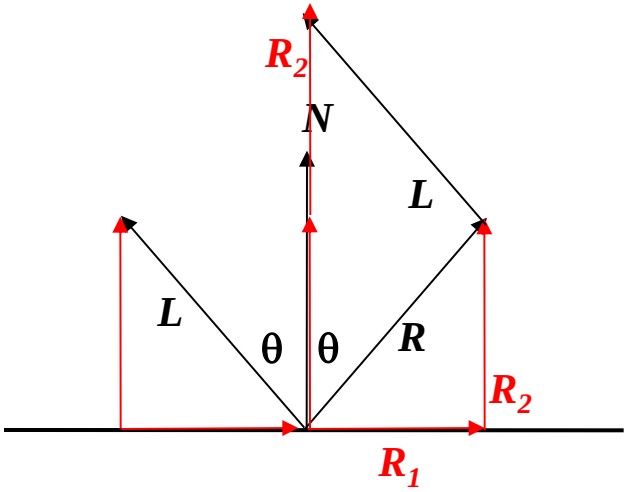
V Vektor von P zum Augpunkt (ViewingPoint), normiert

ϕ_i, θ_i (lokale) sphärische Koordinaten (von L und V)

Reflexionsgesetz, (perfekte) spiegelnde Reflexion

- R Vektor des reflektierten Strahls, normiert





Es gilt:
L und *R* liegen in einer Ebene
und $\theta = \theta_{in} = \theta_{ref}$

$$R = R_2 + R_1 \wedge R_2 = N(N \cdot L) \wedge$$

$$R = 2\mathrm{R}_2 - L \Leftrightarrow$$

$$R = 2(L \cdot N)N - L$$

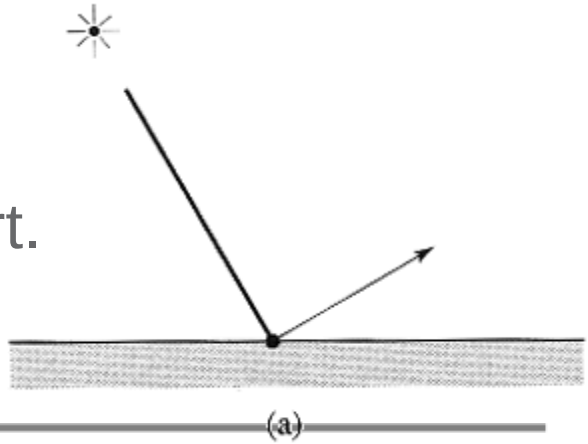
Inhaltsverzeichnis - Themen

Weitere Themen: Netze, Fraktale, Animation, ...

Phong-Beleuchtungsmodell (Bui-Tuong), 1975

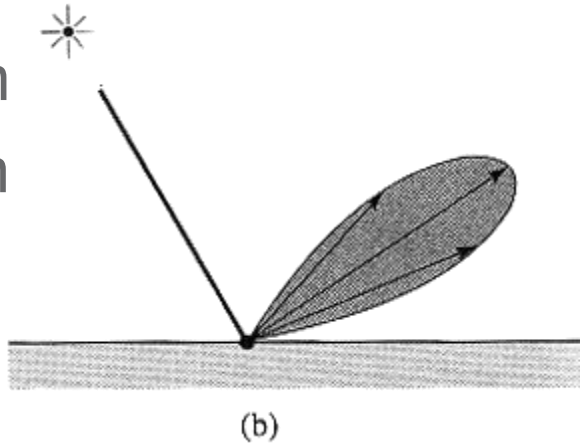
- Am häufigsten verwendetes lokales Beleuchtungsmodell
- Achtung: Es handelt sich um ein **empirisches Modell ohne wirkliche physikalische Basis**, aber mit guten praktischen Resultaten
- Das Modell **simuliert folgende** physikalische Reflexionsphänomene:

  (a) **Perfekte/vollkommene** spiegelnde Reflexion
  - Ein Lichtstrahl wird, **ohne sich aufzustreuen**, perfekt nach dem Reflexionsgesetz reflektiert.
  - Oberfläche: **idealer Spiegel** - existiert in der Realität nicht



Simulierte physikalische Reflexionsphänomene: (Fortsetzung)

(b) **Unvollkommene** spiegelnde Reflexion



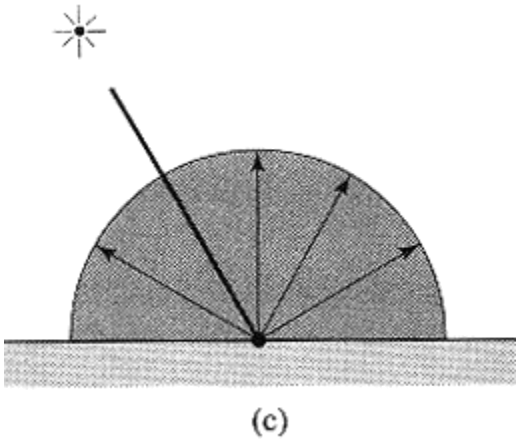
- Der Lichtstrahl wird bei der Reflexion „**aufgespalten**“- es entsteht ein **Reflexionskonus** um die ausgezeichnet Reflexionsrichtung.
- Oberfläche: **unvollkommener Spiegel**, rauhe Oberfläche.
- Oberflächenelement ist mikroskopisch aus vielen kleinen perfekten Spiegeln mit leicht unterschiedlichen Ausrichtungen zusammengesetzt.

6.3 Lokale Beleuchtungsmodelle

Simulierte physikalische Reflexionsphänomene: (Fortsetzung)

(c) **perfekte/vollkommene diffuse** Reflexion

- Der Lichtstrahl wird bei der Reflexion perfekt gestreut, dh. mit **gleichmäßiger Intensität** in alle Richtungen.
- Oberfläche: **ideale matte** Oberfläche, existiert in der Realität nicht, annäherungsweise: feine Lage Puder



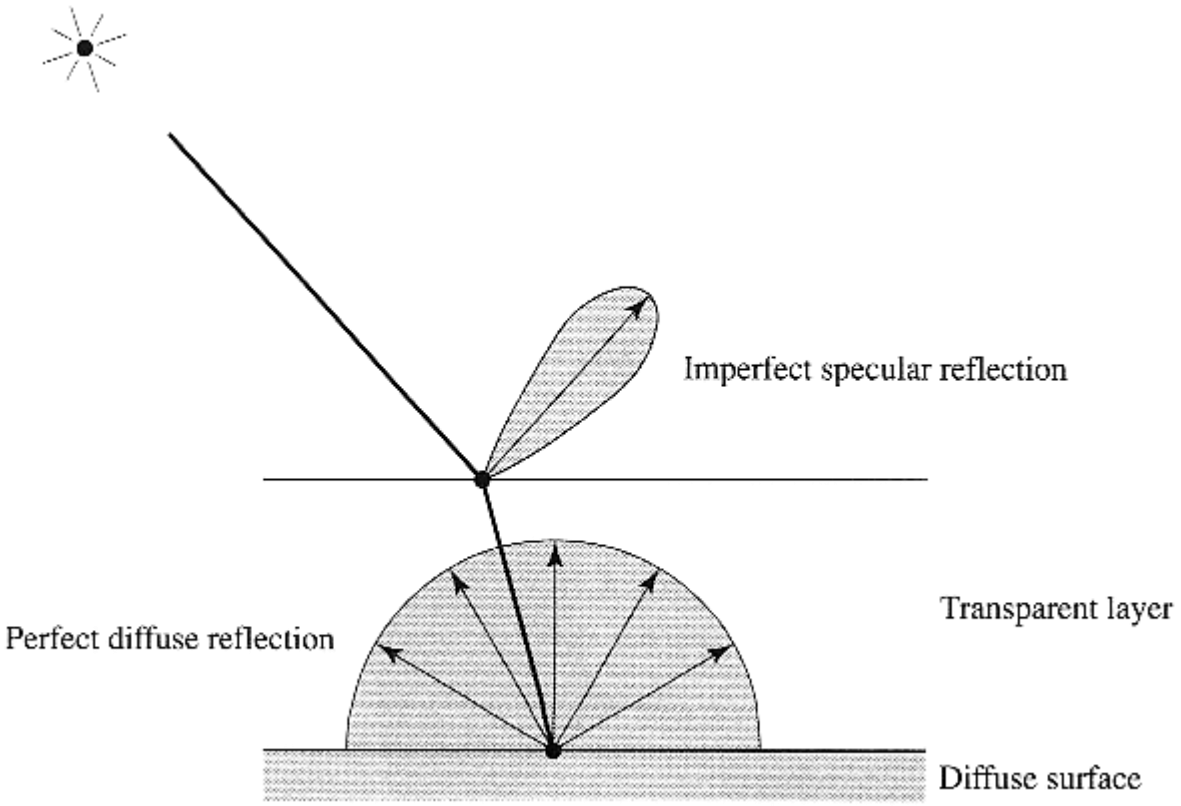
- Phong-Beleuchtungsmodell beschreibt reflektierte Licht eines Oberflächenpunktes aus **drei linear kombiniert** Anteilen:

  $\text{Licht}_{\text{reflekt}}$ = diffuse Komponente + spekulare Komponente + ambiente Komponente

Ambiente Komponente: eine Hilfskonstruktion

- IdR. **konstant** gewählt und simuliert die globale bzw. die indirekte Beleuchtung.
- **Grundbeleuchtung** notwendig, da einige Objekte die Lichtquelle(n) nicht sehen und somit in dem Modell schwarz dargestellt würden.
- In der Realität werden solche Objekte aber **indirekt beleuchtet**.
- Hochkomplexe globale Beleuchtungsberechnung wird durch **simples Addieren** einer Konstanten **ersetzt**.
- Welchen **Typ von Oberflächen** beschreibt nun das Modell?
- Die lineare Kombination von diffuser und spiegelnder Reflexion entspricht zum Beispiel der Physik **polierter Oberflächen**, z. B. poliertem Holz (transparente Schicht: spiegelnd, Oberfläche: diffus)

Polierte Oberflächen:



Das mathematische Modell: (ohne Farbe)

$I = k_a I_a + k_d I_d + k_s I_s$

- Die Physik der Oberfläche wird über die **Verhältnisse der einzelnen Komponenten** modelliert.
- Für diese Konstanten gilt: $k_a + k_d + k_s = 1$
- **Diffuse Reflexion**, Term: $k_d I_d$

  $I_d = I_i \cos(\theta) = I_i (L \cdot N)$ mit

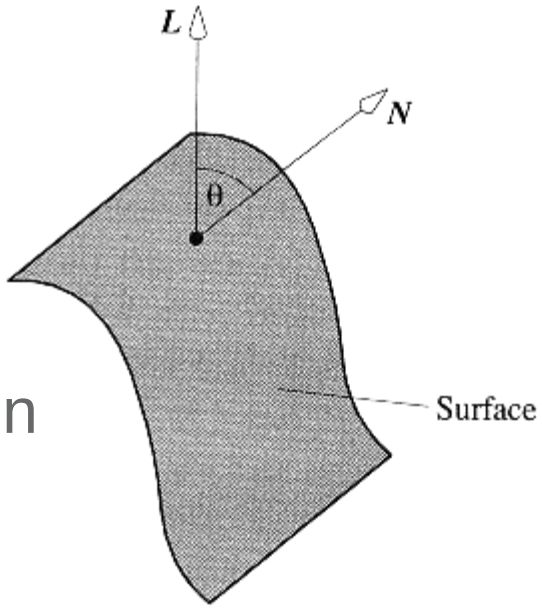
  I_i Intensität des einfallenden Lichts der Lichtquelle i

  θ Winkel zwischen Punktnormale N und Lichtvektor L

Diffuse Reflexion: (Fortsetzung)

Die diffuse Komponente des Phong-Modells modelliert das **Kosinusgesetz von Lambert**:

- Bei **ideal diffusen (matten) Oberflächen** ist die Intensität des (in alle Richtungen gleich) reflektierten Lichtes eine **Funktion des Kosinus** zwischen Oberflächennormale und Lichtvektor.

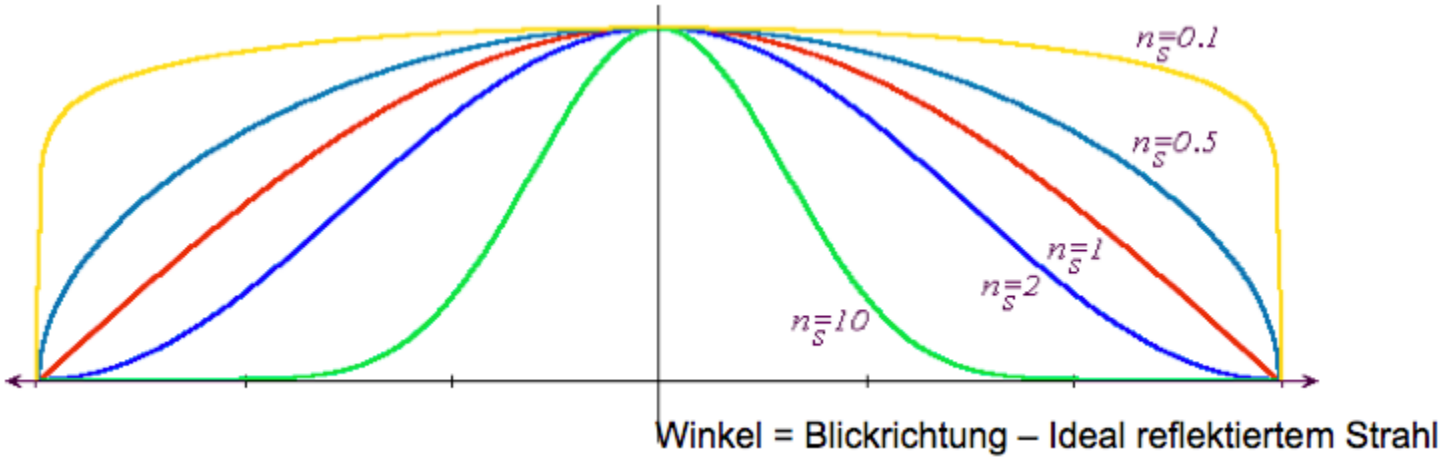


Spiegelnde Reflexion, Term $k_s I_s$:

- Physikalisch gesehen besteht die spiegelnde Reflexion aus einem **Abbild der Lichtquelle**, das über einen Teil der Oberfläche „**verschmiert**“ ist - **Highlight**
- Highlight kann vom Betrachter nur gesehen werden, wenn seine **Betrachtungsrichtung (V) nahe der Reflexionsrichtung** (R) liegt.
- Dies wird simuliert durch:
  - $I_s = I_i \cos^n(\Omega) = I_i (R \cdot V)^n$, mit
  - Ω Winkel zwischen V und R
  - n simuliert Perfektionsgrad der Oberfläche ($n \to \infty$ heißt perfekter Spiegel, d. h. reflektiertes Licht nur in Richtung R, **keine Streuung**)

Spiegelnde Reflexion: (Fortsetzung)

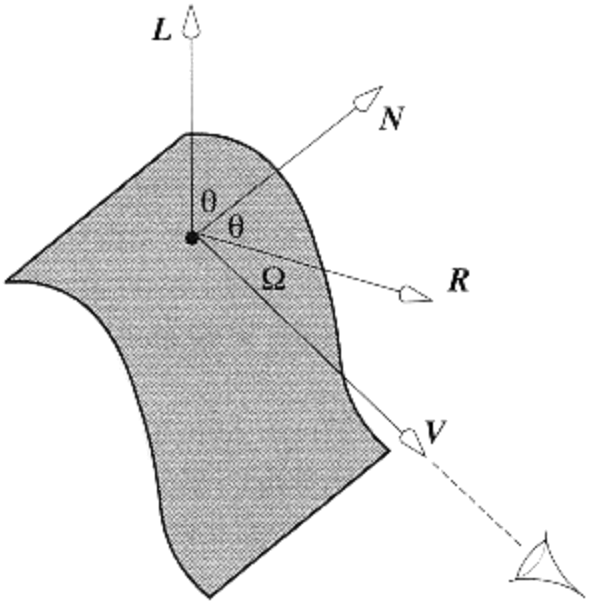
- Term wird **kleiner mit zunehmenden Winkel** zwischen Blickwinkel und (ideal) reflektierten Strahl.



Spiegelnde Reflexion: (Fortsetzung)

Bemerkung

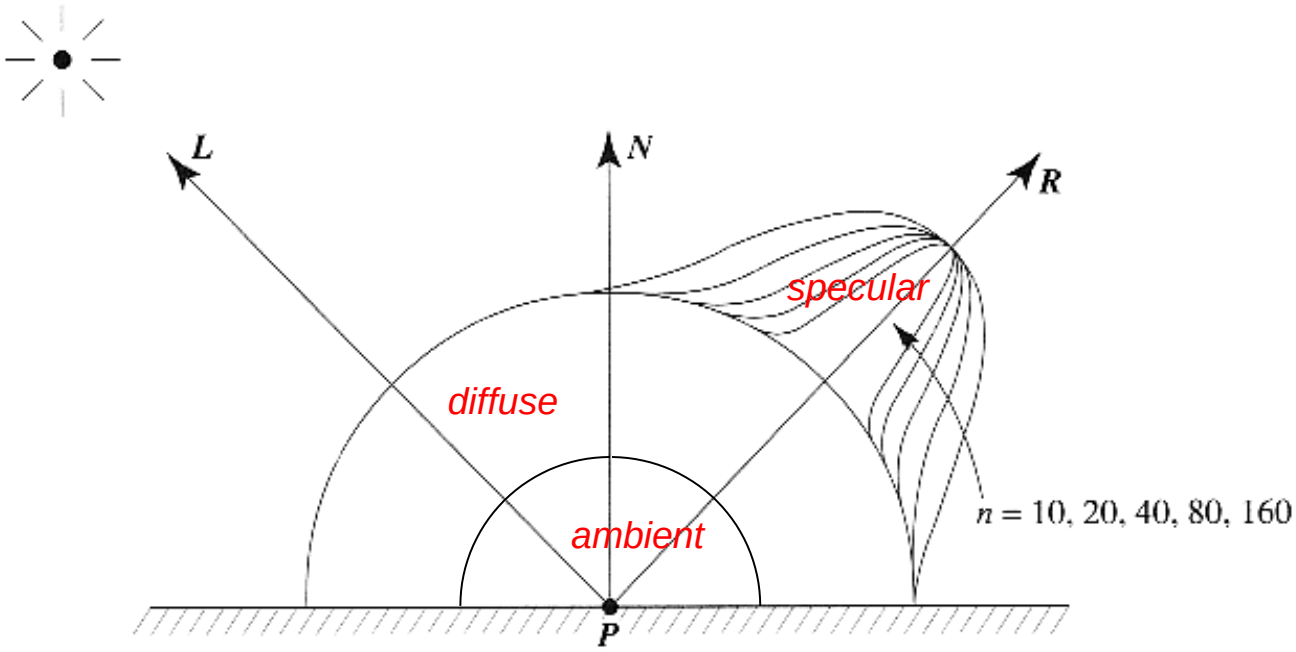
- Für verschiedene L entsteht (bis auf seine Ausrichtung um R) **immer der gleiche** Reflexions-Intensitätskonus.
- Dies entspricht **nicht der realen Abhängigkeit von Spiegelungen** von der Ausrichtung des Lichtvektors
- Gravierender **Mangel** des Modells



Das Gesamtmodell

- $I = k_a I_a + k_d I_d + k_s I_s = k_a I_a + I_i (k_d(L \cdot N) + k_s(R \cdot V)^n)$

- im 2D-Schnitt:



6.3 Lokale Beleuchtungsmodelle

Beispiel

k_a konstant

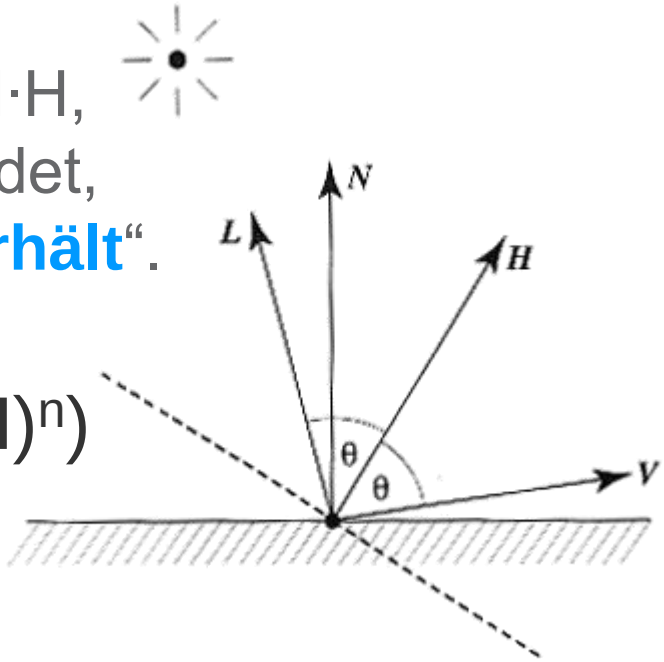
zunehmendes k_s

zunehmendes n

Bemerkung

- Sind **Betrachter und Lichtquelle unendlich** (oder hinreichend) weit entfernt, so kann statt dem Reflektionsvektor R ein **konstanter Vektor H eingesetzt** werden: $H = (L+V) / \|L+V\|$.
- Man betrachtet statt R·V jetzt N·H, das sich zwar von R·V unterscheidet, aber „**auf die gleiche Weise verhält**“.
- Damit ergibt sich:

$$I = k_a I_a + I_i \left(k_d (L \cdot N) + k_s (N \cdot H)^n\right)$$



Das mathematische Modell: (mit Farbe)

- Für farbige Objekte (Lichtquellen) wird das Modell getrennt auf die Farbkomponenten I_r, I_g, I_b angewendet:

$$I = k_{ar} I_a + I_i \, (k_{dr}(L \cdot N) + k_{sr}(N \cdot H)^n)$$

$$I = k_{ag} I_a + I_i \, (k_{dg}(L \cdot N) + k_{sg}(N \cdot H)^n)$$

$$I = k_{ab} I_a + I_i \, (k_{db}(L \cdot N) + k_{sb}(N \cdot H)^n)$$

mit

- k_{dr}, k_{dg}, k_{db} modellieren die Farbe des **Objektes**
- k_{sr}, k_{sg}, k_{sb} modellieren die Farbe der **Lichtquelle** (für weißes Licht ist $k_{sr} = k_{sg} = k_{sb}$)
- k_{ar}, k_{ag}, k_{ab} modellieren die Farbe des **Umgebungslichtes**

Bemerkungen

- Gravierende Mängel des Modells
  - **Gegenseitige Reflektionen und Spiegelungen** der Flächen werden durch den **konstanten ambienten Term** nur unzureichend beschrieben.
  - Objektoberflächen wirken „**plastikhaft**“, z. B. lässt sich **kein blankes** Metall modellieren

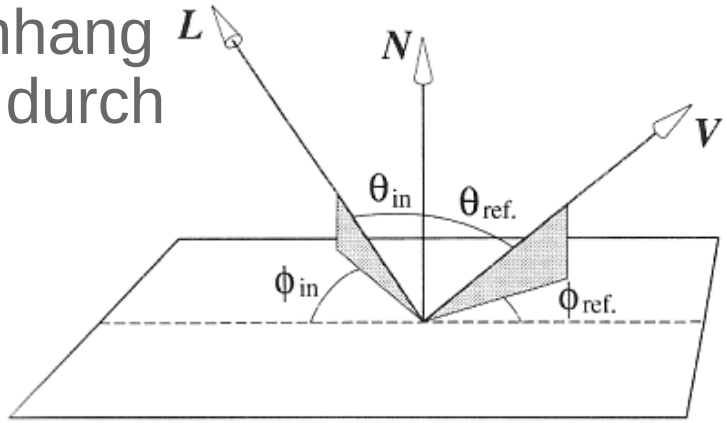
⇒ **physikalisch basierte lokale Beleuchtungsmodelle**, die versuchen die BRDF (Reflexionsfunktion, nächste Folien) korrekt zu simulieren oder gänzlich andere Techniken, wie z. B. Mapping-Verfahren

BRDF (bi-directional reflection distribution function)

- Allgemein wird das von **einem Punkt einer Oberfläche reflektierte Licht** durch eine BRDF beschrieben.
- Betont insbesondere die **Abhängigkeit des in einer beliebigen Richtung reflektierten Lichts** von der Richtung des einfallenden Lichts.
- Sind die Richtungen von **L und V gegeben**, so wird der Zusammenhang zwischen den Intensitäten also durch eine

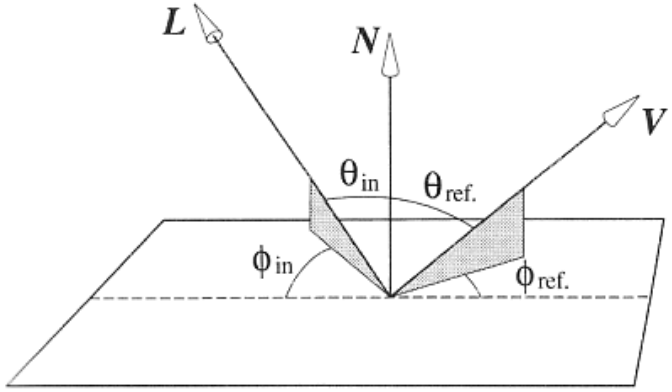
[redacted]

beschrieben.



6.3 Lokale Beleuchtungsmodelle

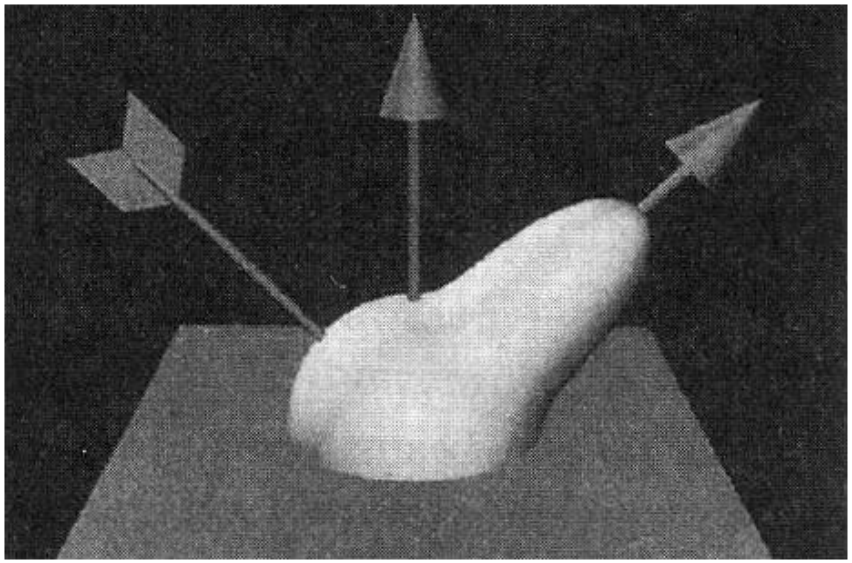
BRDF: (Fortsetzung)

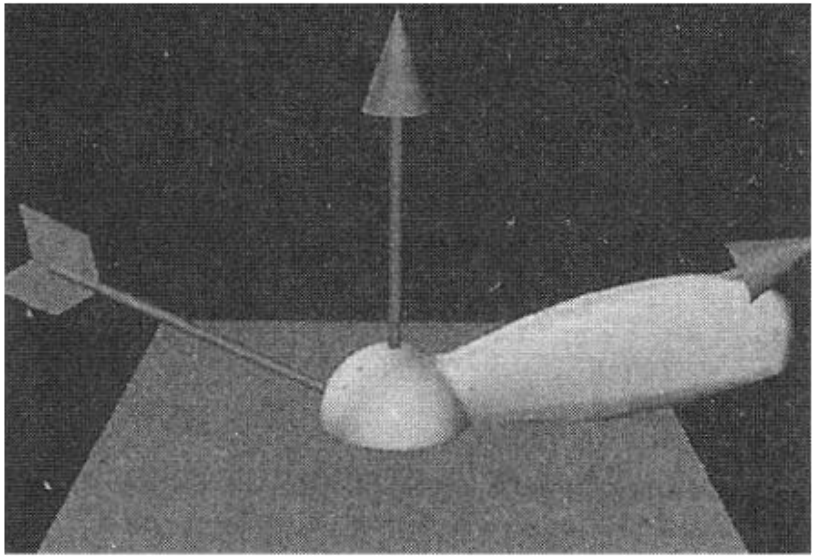


- In der Praxis fällt an einem Oberflächenpunkt **Licht von mehreren Richtungen** ein.
- Das gesamte resultierende reflektierte Licht muss dann mittels **Integration über die Hemisphäre** gewonnen werden.
- Fragestellungen:
  - Wie gewinnt man BRDFs?
    ⇒ z. B. Messung, Modelle
  - In welcher Feinheit repräsentiert man BRDFs?
    ⇒ Heuristiken, falls keine geschlossene Form
  - Wie speichert und verarbeitet man effizient BRDFs?
    ⇒ z. B. Matrizen, mittlerweile in der Regel via GPU-Shader

BRDF: (Fortsetzung)

- Darstellung von (nach Blinn (1977)) erzeugten BRDFs für **zwei verschiedene** Richtungen des einfallenden Lichtes:





Nachteile „rein“ lokaler Beleuchtungsmodelle

- Spiegeln Idealfall **eines einzelnen von einer einzigen Punktlichtquelle beleuchteten Objektes** in der Szene wider.
- Betrachten **ausschließlich direkte** Beleuchtung (bis auf Hilfskonstruktionen).
- **Interaktion** mit anderen Objekten **nicht modelliert** (d. h. keine indirekte Beleuchtung, kein Schattenwurf!)

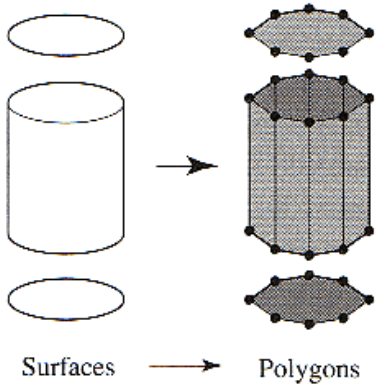
$\Rightarrow$ globale Beleuchtungsverfahren

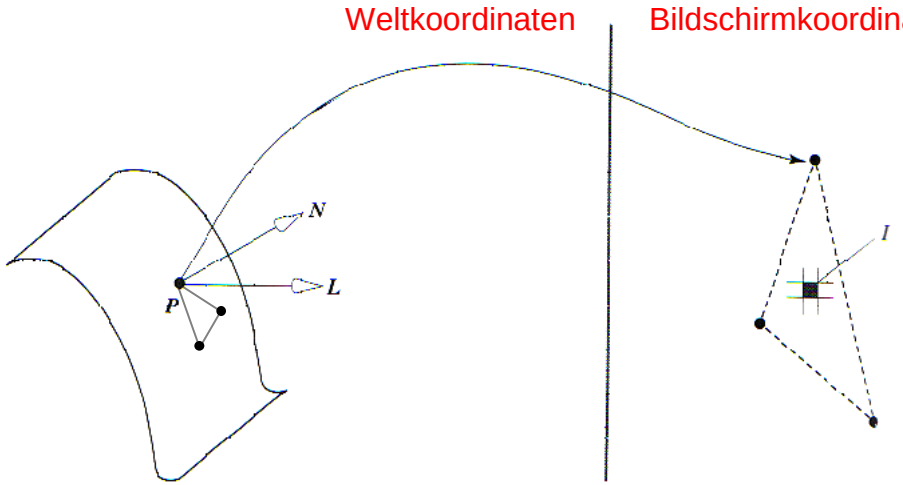
Inhaltsverzeichnis - Themen

6.4 Interpolative Schattierung

Wie wird nun die **Auswertung eines Beleuchtungsmodells** bei einem Objekt zur Bestimmung der Lichtintensität auf dessen Oberfläche angewendet?

Wir setzen im Folgenden eine polygonale Objektrepräsentation, also eine facettierte Darstellung, voraus.



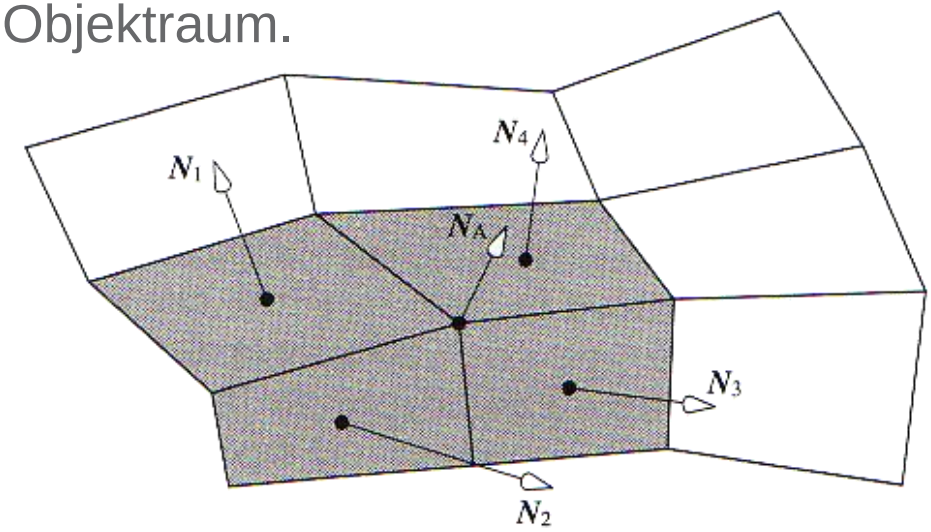


Man beachte die Unterscheidung zwischen (dreidimensionalem) Objektraum und (zweidimensionalem) Bildraum!

6.4 Interpolative Schattierung

Flat/Uniform Shading

- Pro Polygon / Facette wird das verwendete Beleuchtungsmodell **genau einmal** in einem ausgewählten Oberflächenpunkt ausgewertet.
- Die dort ermittelte Lichtintensität wird auch **allen weiteren Punkten** der Polygonoberfläche zugewiesen.
- Grundlage der Berechnung ist die **Polygon- oder Oberflächennormale** im Objektraum. (hier z. B. N_1, N_2, N_3, N_4, ...)
- Als ausgewählte Punkte werden die **Polygonschwerpunkte** oder der Einfachheit wegen **Polygoneckpunkte** gewählt.



6.4 Interpolative Schattierung

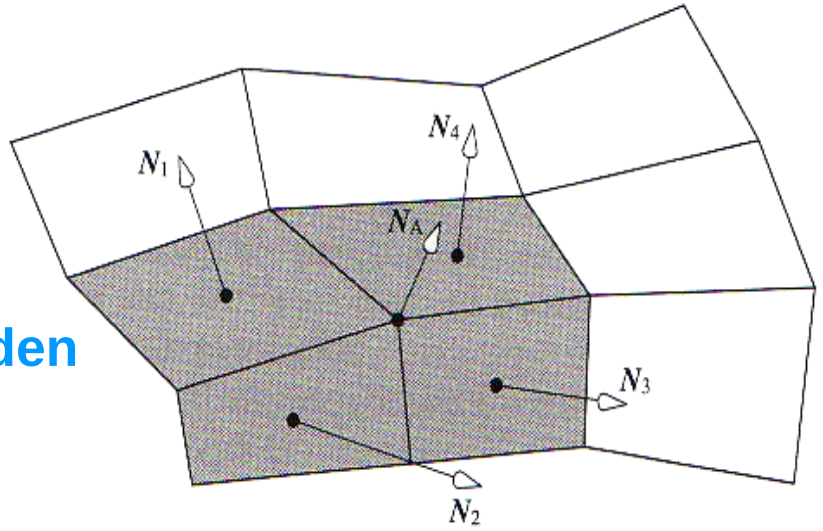
Flat/Uniform Shading (Fortsetzung)

- Einfaches, kostengünstiges Verfahren, Echte Interpolation **findet nicht statt**
- **Kanten in Polygonnetzen** bleiben bei der Darstellung sichtbar.
- Objekte werden **facettiert** dargestellt.
- **Unstetiger** Intensitätsverlauf über die Polygonkanten
- Anwendung für **Visualisierung der polygonalen Auflösung**, zB. für Qualitätsdarstellung

6.4 Interpolative Schattierung

Gouraud- und Phong-Shading

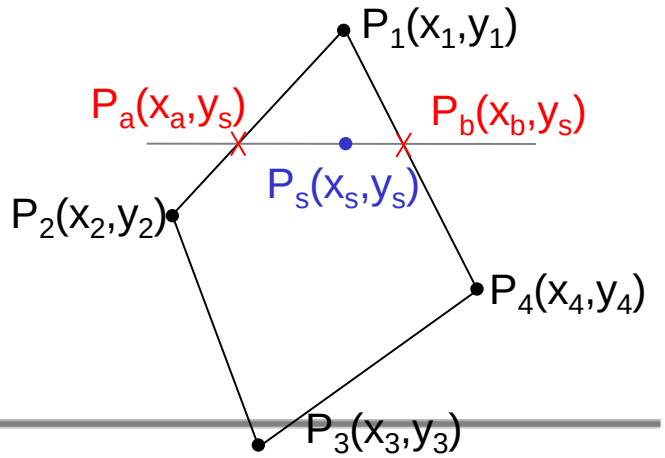
- **Interpolieren Farbe** zwischen den Stützpunkten
- **Glätten (virtueller) Kanten** zwischen einzelnen Polygonfacetten (Das Polygonnetz stellt die Approximation einer gekrümmten Oberfläche dar.)
- Grundlage der Berechnung sind die **Eckpunktnormalen** in den gemeinsamen Polygoneckpunkten. (hier zB. N_A, ...)
- Eine Eckpunktnormale entsteht aus (gewichteter) **Mittelung der Polygonnormalen aller angrenzenden Polygone** mit dem entsprechenden gemeinsamen Eckpunkt.
- **Normalisierung** notwendig



6.4 Interpolative Schattierung

Gouraud und Phong-Shading (Fortsetzung)

- Beide Verfahren bedienen sich einer **(bi)linearen Interpolation** im Bildraum
  - Farbwerte auf der **Fläche des Polygons** werden aus den Farbwerten der **Stützpunkte** (idR. Eckpunkte) des Polygons (i.a. bezüglich des Objektraums ermittelt) errechnet
  - Es wird eine **zweifacher linearer** Interpolation im Bildraum verwendet (bilinear)
- Effiziente Implementierungen arbeiten **Scanline-weise** und **inkrementell**.

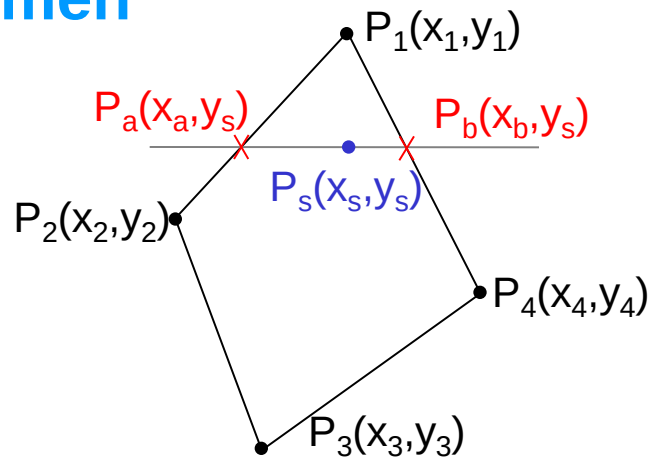


Gouraud und Phong-Shading (Fortsetzung)

1. Schritt: Stützwerte $W(P_1)$, $W(P_2)$, $W(P_3)$, $W(P_4)$ bestimmen
2. Schritt: Schnittpunkte Scanline-Polygonkanten P_a, P_b bestimmen
3. Schritt: Werte $W(P_a)$, $W(P_b)$ **bestimmen**

$$W(P_a) = \frac{1}{y_2 - y_1}\left(W(P_1)(y_2 - y_a) + W(P_2)(y_a - y_1)\right)$$

$$W(P_b) = \frac{1}{y_4 - y_1}\left(W(P_1)(y_4 - y_a) + W(P_4)(y_a - y_1)\right)$$



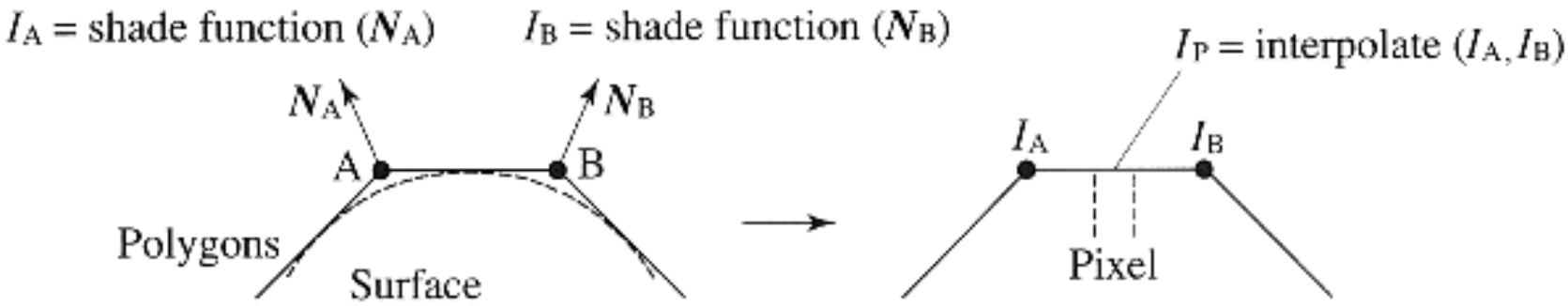
4. Schritt: Wert $W(P_s)$ **bestimmen**

$$W(P_s) = \frac{1}{x_b - x_a}\left(W(P_a)(x_b - x_s) + W(P_b)(x_s - x_a)\right)$$

6.4 Interpolative Schattierung

Gouraud-Shading

- Die Auswertung des Beleuchtungsmodells **erfolgt ausschließlich** in den Polygoneckpunkten unter Ausnutzung der Eckpunktnormalen.
- Intensitätswerte der **projizierten inneren** Polygonpunkte (Pixel) werden mit **Interpolation berechnet**

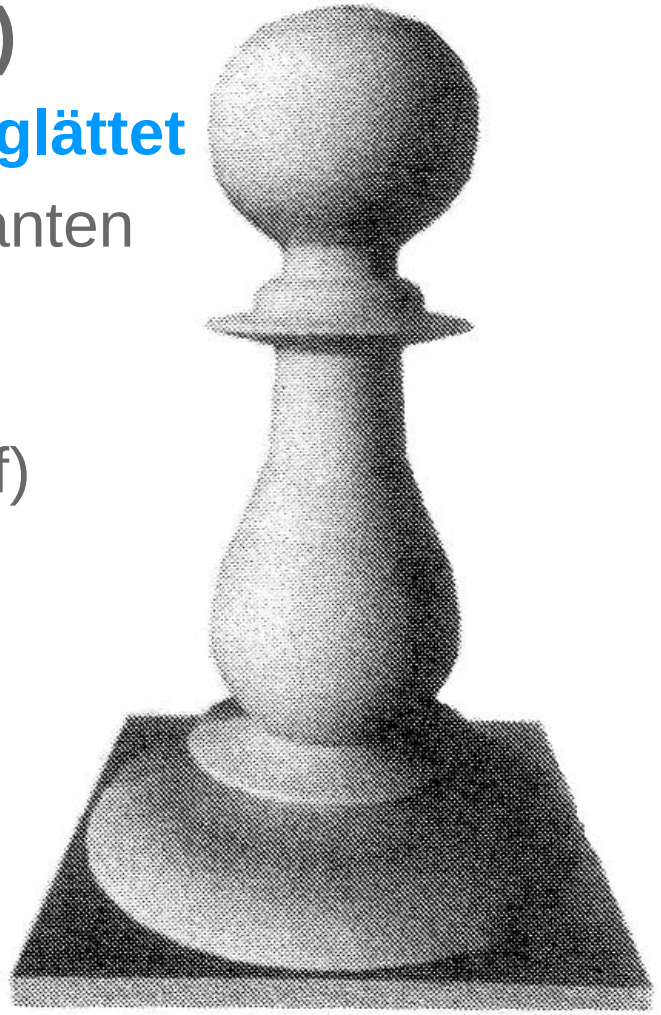


6.4 Interpolative Schattierung

Gouraud-Shading (Fortsetzung)

- Kanten in Polygonnetzen werden **geglättet**
- Intensitätsverlauf über die Polygonkanten **ist stetig**, aber **nicht wirklich glatt**

  ⇒ **Anfälligkeit** des Verfahrens für Mach-Band-Effekte (siehe §6.5ff)
- **Highlights** werden **verwischt**
- Highlights werden nur dann gut repräsentiert, wenn sie **auf Stützpunkten** (Eckpunkten) liegen



Gouraud-Shading (Fortsetzung)

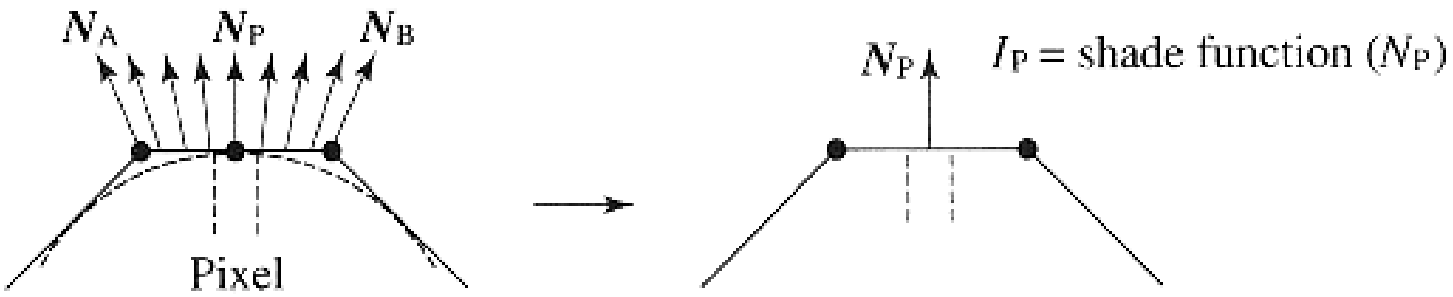
⇒ Highlights werden **durch Abtastfehler** „verschluckt“ oder erscheinen polygonförmig (statt rund)

- gängig: Kombination von Gouraud-Shading und Beleuchtungsmodell mit **ausschließlich diffuser** Reflexionskomponente
- Bemerkung:
  Gouraud-Shading wird oft als **Schattierungsverfahren** von heutiger Graphikhardware effizient umgesetzt.

6.4 Interpolative Schattierung

Phong-Shading

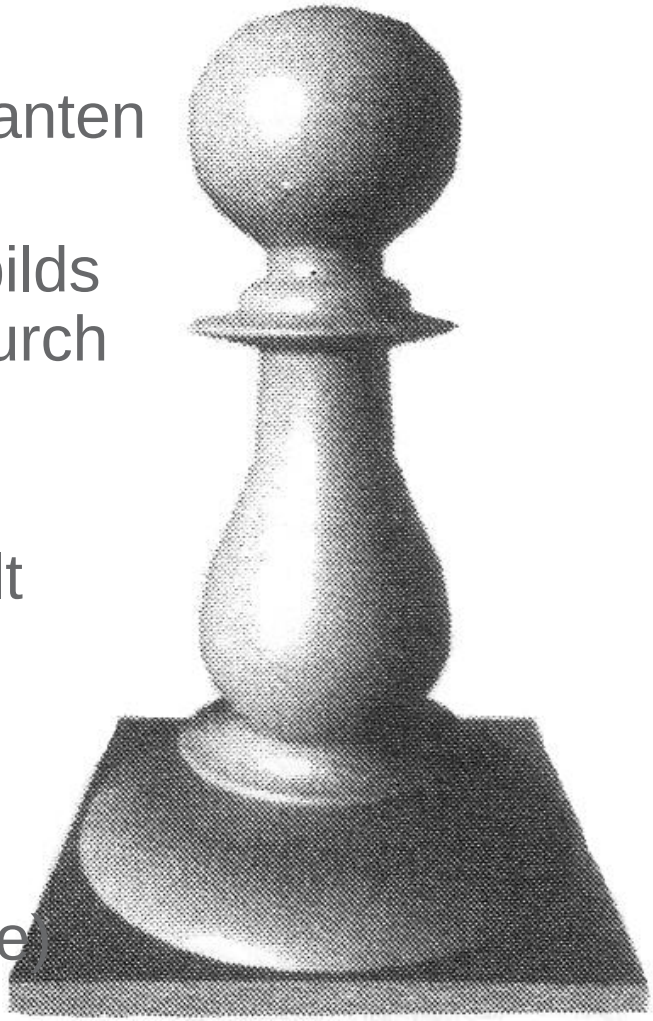
- Die Auswertung des Beleuchtungsmodells erfolgt für jeden projizierten Punkt (Pixel) der Polygonoberfläche
- Die Oberflächennormalen in den Polygonpunkten werden mittels **Interpolation der Vektorkomponenten** aus den Stützpunktnormalen ermittelt.
- Normalisierung nötig



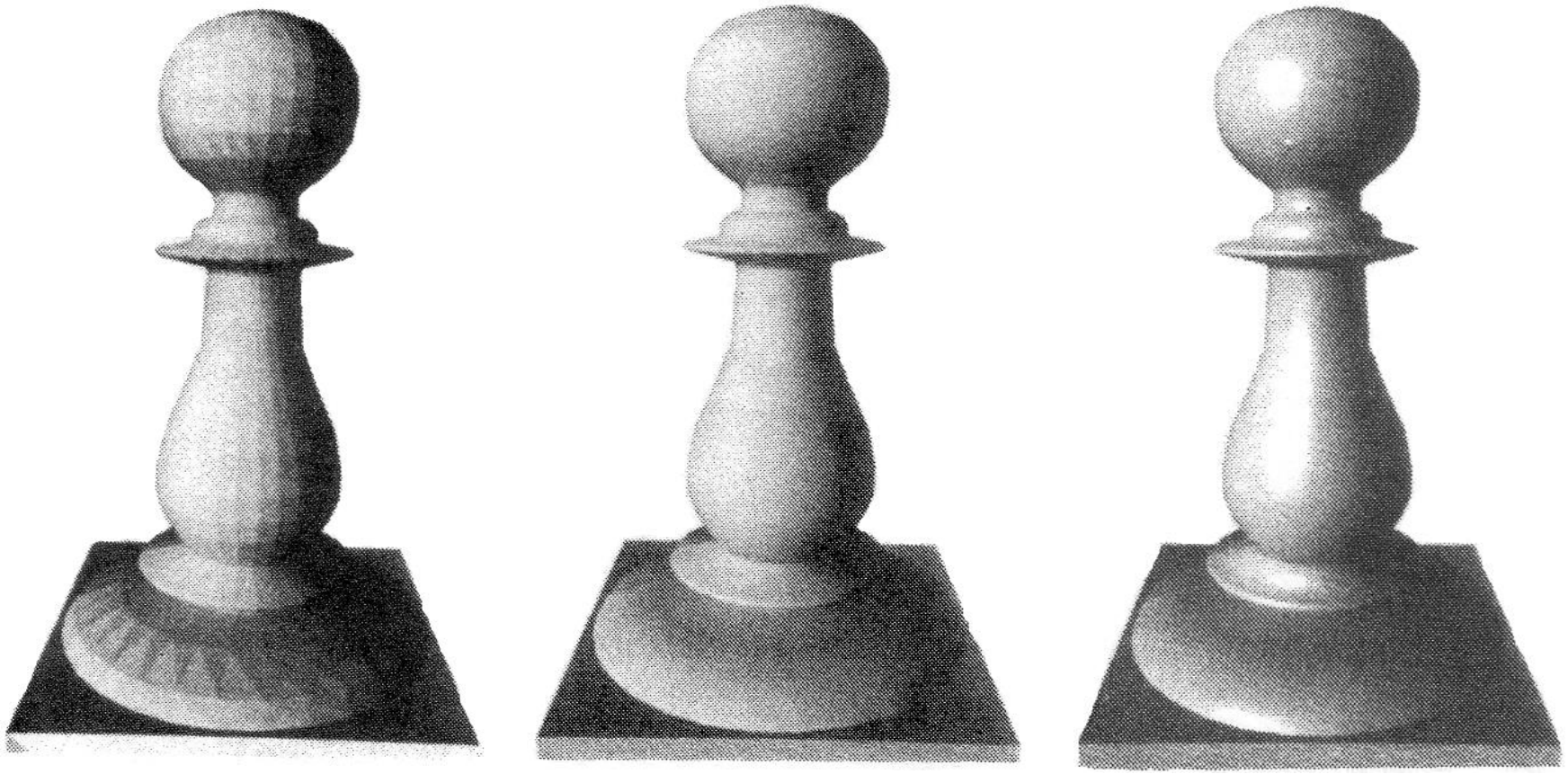
6.4 Interpolative Schattierung

Phong-Shading (Fortsetzung)

- Intensitätsverlauf über die Polygonkanten ist **stetig und glatt**
- Gute Annäherung des Erscheinungsbilds der realen gekrümmten Oberfläche durch **interpolierten Normalen**
- Rechenaufwändiges Verfahren
- Highlights werden **adäquat** dargestellt
- Bemerkung:
  Phong-Shading wird von heutiger **Graphikhardware** unterstützt.
  (d.h. Vektorinterpolation, Auswertung des Beleuchtungsmodells in Hardware)



Flat/Uniform, Gouraud und Phong-Shading



6.4 Interpolative Schattierung

Bemerkung

- Was muss beachtet werden, wenn bei der Anwendung von Gouraud oder Phong-Shading **Polygonkanten explizit als Kanten** (Feature Lines) sichtbar bleiben sollen?

⇒ Polygoneckpunkte der Feature Lines müssen für die beteiligten Polygone **separat mit unterschiedlichen** Normalenvektoren gespeichert werden

⇒ **enge Verknüpfung und Abhängigkeit** zwischen der Schattierungstechnik und der Polygonalisierung bzw. der Triangulierungsmethode für das Objekt (Feature Recognition)

(hier treten i.d.R. in der Praxis beim Datentransfer zwischen Visualisierungssystemen ungeahnte Schwierigkeiten auf)

Inhaltsverzeichnis - Themen

Weitere Themen: Netze, Fraktale, Animation, ...

6.5 Das Auge isst mit...

- **Vorverarbeitung (low level)** im visuellen System des Menschen vor Verarbeitung in den höheren Regionen (high level)
- Wie reagieren Lichtrezeptoren im Auge auf Unterschiede einfallender Lichtintensität?

Lechners Gesetz

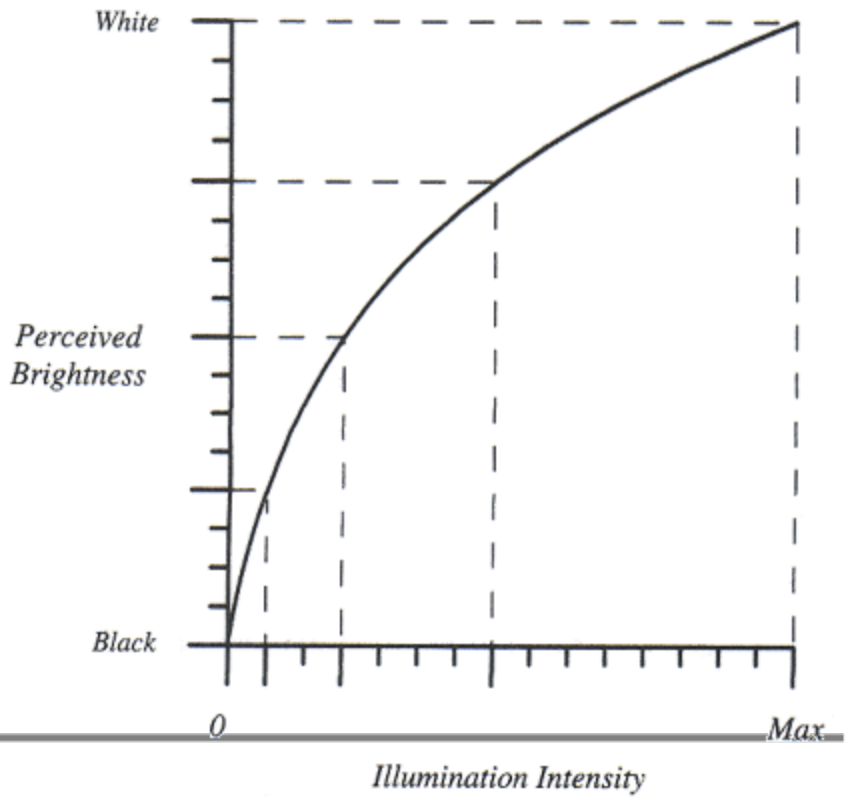
- Die Beziehung zwischen der ins Auge **einfallenden Lichtintensität** und der vom Auge **wahrgenommenen Lichtintensität** ist nicht linear, sondern **annähernd logarithmisch.**

6.5 Das Auge isst mit...

Lechners Gesetz (Fortsetzung)

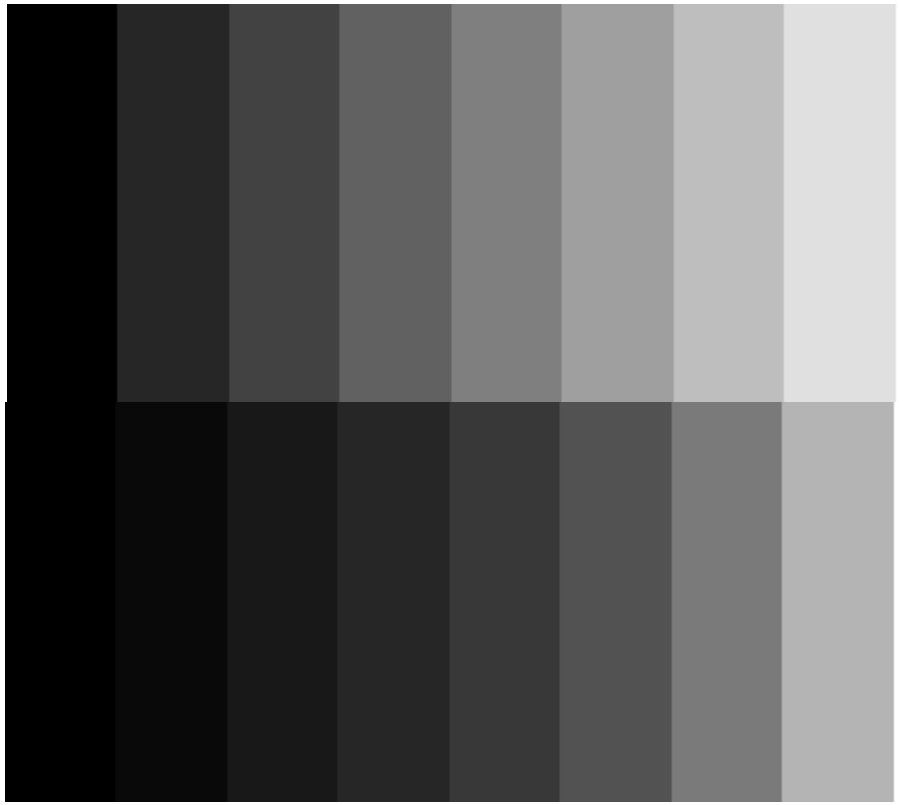
Folgerung

- **Kleine Helligkeitsunterschiede in dunklen Regionen** sind besser wahrnehmbar als vom Betrag her **identische Helligkeitsunterschiede in hellen Regionen**.



Lechners Gesetz (Fortsetzung)

- Anwendung: Helligkeitsverläufe / Farbverläufe



Intensitätssteigerung in äquidistanten Schritten von 12,5% bezogen auf die einfallende Intensität (von 0% bis 100%)
→ **Helligkeitssprung** in dunkler Region ist deutlicher als gleiche Sprünge in heller Region
→ **große Unterschiede** zwischen wahr genommenen Intensitätssprüngen

Intensitätssteigerung in äquidistanten Schritten bezogen auf die wahrgenommene Intensität, nach vorgehendem Diagramm **angepasst**
→ Wahrnehmung **nahezu äquidistanter** Intensitätssprünge

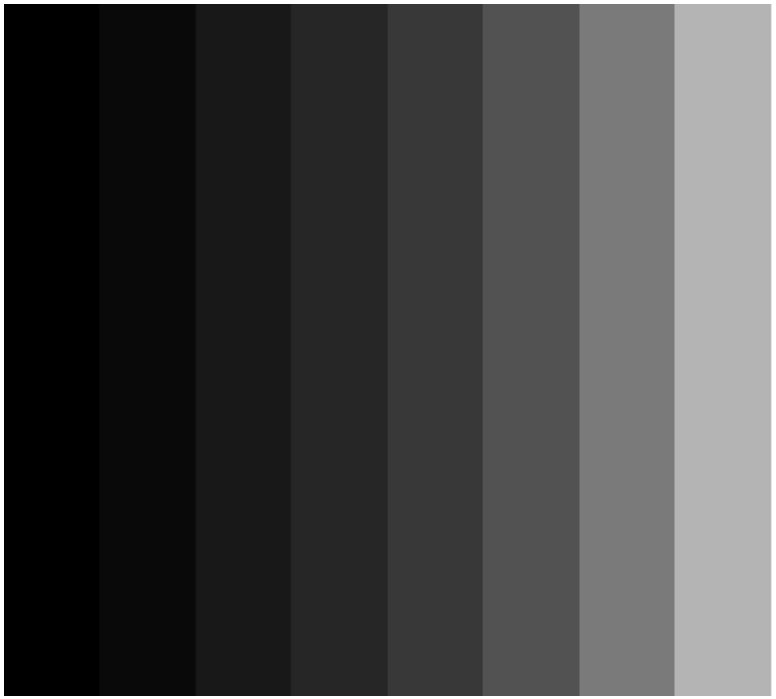
6.5 Das Auge isst mit...

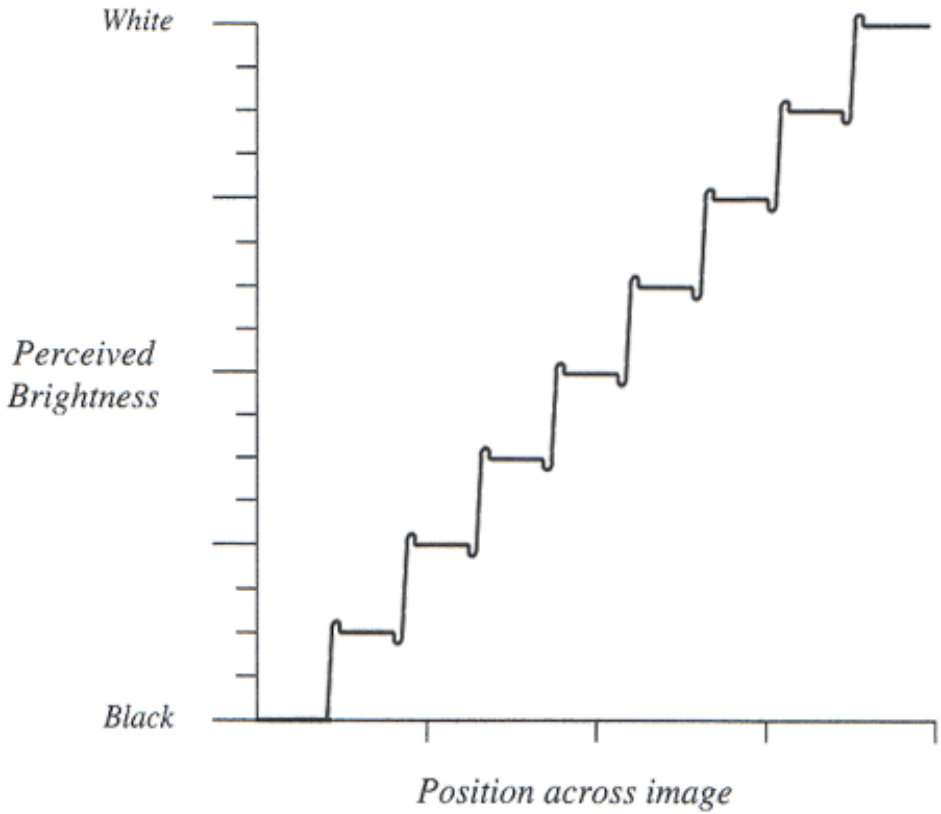
Mach Band-Effekt

- Interaktion der Lichtrezeptoren im Auge **betont „scharfe“** Intensitätsänderungen
- Sobald das Auge bei der einfallenden Intensität solche Änderungen „feststellt“, **addiert es zusätzlich** Unterschwinger und Überschwinger zur wahr-genommenen Intensität, die den **Übergang zusätzlich betonen**.
- Dieser **low-level Mechanismus der Kantenbetonung** bei Intensitätsübergängen verhilft unserer visuellen Wahrnehmung zu einer **automatischen Konturenschärfe**

Mach Band-Effekt (Fortsetzung)

- Beispiel



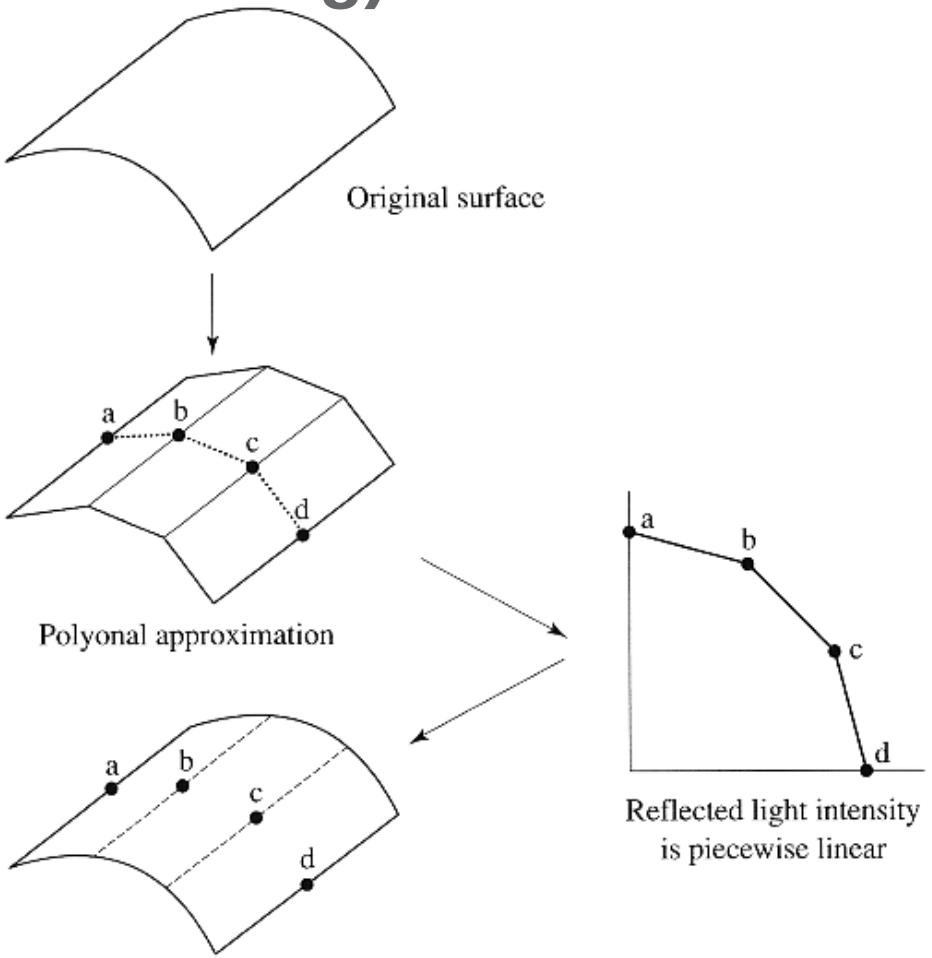


Mach Band-Effekt (Fortsetzung)

- Beim Rendering ist die **automatische Kantenbetonung** bei Intensitätsänderungen **störend** und kann lediglich durch möglichst glatte Intensitätsübergänge **reduziert** werden.
- Flat/Uniform Shading: **unstetige Intensitätswechsel**, sehr starke Mach Band-Effekte
- Gouraud-Shading: stetige Intensitätswechsel, trotzdem **abhängig von Polygonalisierung** starke Mach Band-Effekte
- Phong-Shading: glatte Intensitätswechsel **reduzieren Mach Band-Effekte** erheblich

Mach Band-Effekt (Fortsetzung)

- Entstehung von Mach Band-Effekten beim Gouraud-Shading



This produces Mach bands in the image

Quellen

- Computergraphik, Universität Leipzig
  (Prof. D. Bartz)
- Graphische Datenverarbeitung I, Universität Tübingen
  (Prof. W. Straßer)
- Graphische Datenverarbeitung I, TU Darmstadt
  (Prof. M. Alexa)
- Wikipedia: http://en.wikipedia.org