Interaktive Computergrafik

Vorlesung im Sommersemester 2017
Kapitel 4: Precomputed Radiance Transfer

Prof. Dr.-Ing. Carsten Dachsbacher
Lehrstuhl für Computergrafik
Karlsruher Institut für Technologie

Karlsruhe Institute of Technology

Motivation

- Ziel: realistische, globale Beleuchtung in Echtzeitanwendungen
  - herkömmliche Methoden (Vorlesung: Photorealistische Bildsynthese) sind nicht schnell genug sind
  - Bildraum-Approximationen meist nur für räumlich begrenzte Effekte
- in diesem Kapitel lernen wir ein Verfahren kennen, das den **Lichttransport vorberechnet** und geschickt speichert
  - Licht**transport** ≠ Lösung („Farbe“) der Transportgleichung

www.coretechniques.info

Motivation

- bei der Vorberechnung (von Lichttransport) gibt es Trade-Offs zwischen
  - Aufwand bei der Vorberechnung,
  - Aufwand zur Laufzeit und
  - Speicherverbrauch

- Vorberechnung bedeutet: irgendetwas ist statisch
  - z.B. Lightmaps für diffuse Flächen: alles statisch, nur die Kamera nicht
  - in diesem Kapitel: nur die Geometrie ist statisch (vorerst)

Motivation

Image-based Lighting

- aufgenommene reale Beleuchtung für synthetische Objekte
- Beleuchtung durch (dynamische) Umgebung

synthetische Objekte und Beleuchtung

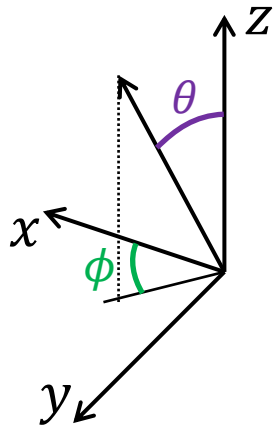
synthetische Objekte und „echtes“ Licht

Bilder: Paul Debevec

Image-based Lighting

Was kennen wir schon? Environment Mapping

- Textur speichert reflektiertes Licht für alle Richtungen: $P(\theta, \phi, x_c, y_c, z_c)$



- berechne Reflexionsvektor und schlage in Textur nach

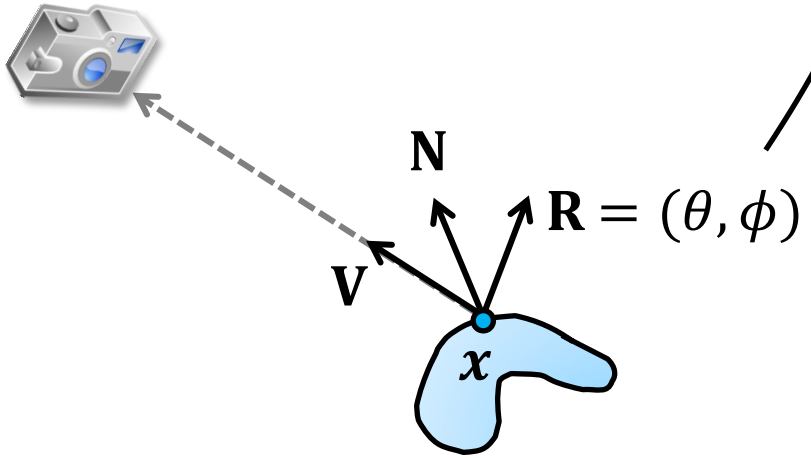
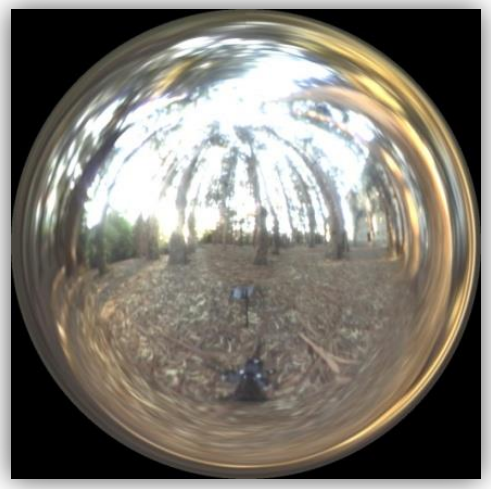


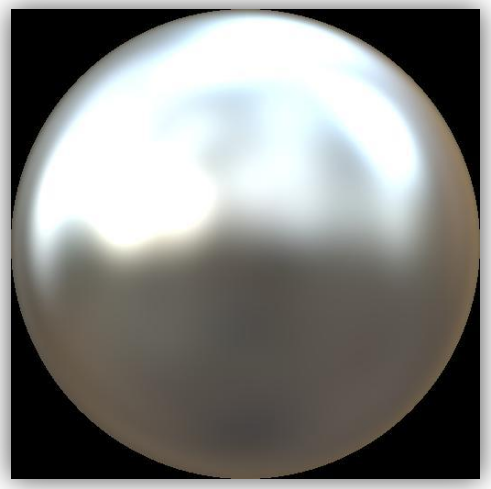
Image-based Lighting

Vorgefilterte Environment Maps

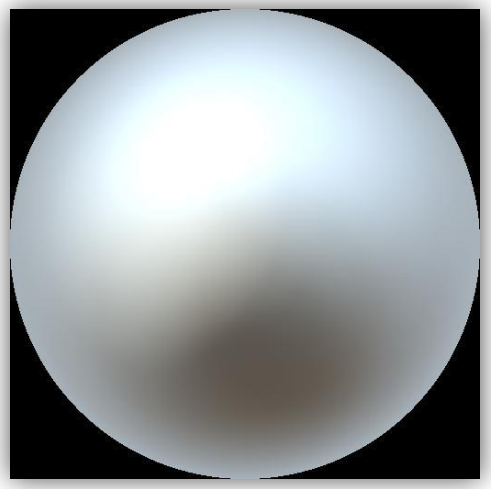
- einfach und schnell zu verwenden
- Approximation verschiedener BRDFs
- aber: keine lokale Verschattung, keine Interreflexion



nicht vorgefiltert:
spiegelnde Objekte
(Zugriff mit $\mathbf{r}$)



vorgefiltert für
imperfekte Spiegelung
(Zugriff mit $\mathbf{r}$)



vorgefiltert für
diffuse Reflexion
(Zugriff mit $\mathbf{n}$)

Ziel/Inhalt dieses Kapitels

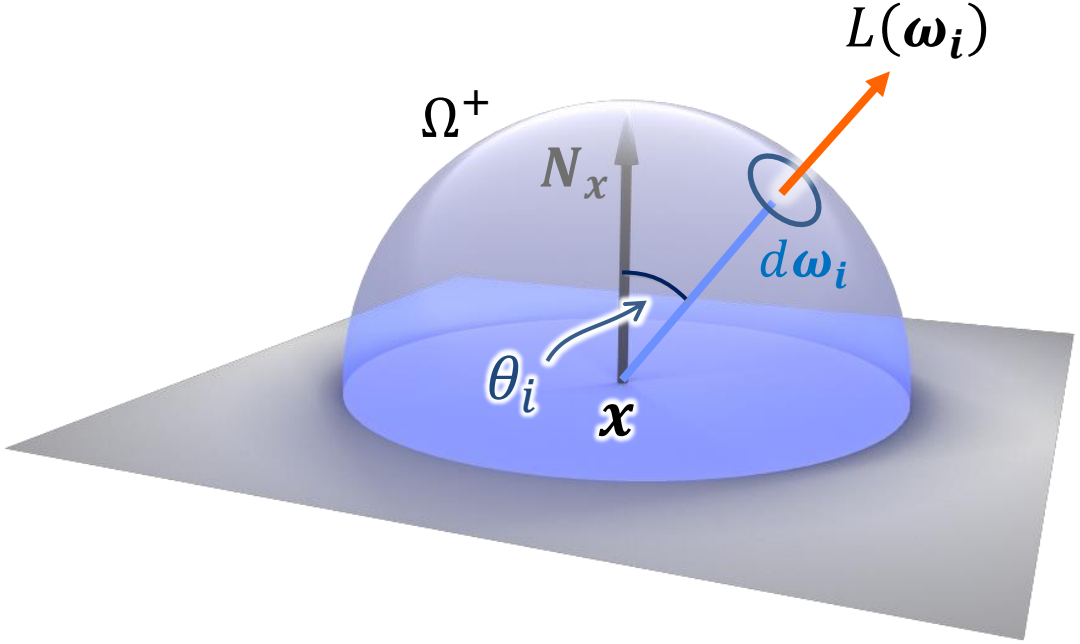
- zunächst: **effiziente** Beleuchtung von **diffusen** (und moderat spiegelnden) Flächen durch **Environment Maps**
  - mit Verschattung, indirekter Beleuchtung, Transluzenz, …
  - weit verbreitet in vielen interaktiven Anwendungen und Spielen (z.B. Halo 3, Crysis 2, von Activision), Basis für div. Rendering-Techniken

Ground-Truth, Reflexionsintegral

- Licht erreicht Oberflächen aus allen Richtungen (= Environment Map)
- reflektiertes **direktes** Licht an Oberflächenpkt. $\boldsymbol{x}$ aufgrund EnvMap $L(\boldsymbol{\omega_i})$

$$L_r(\boldsymbol{x}, \boldsymbol{\omega_r}) = \int_{\Omega} f_r(\boldsymbol{\omega_i}, \boldsymbol{x}, \boldsymbol{\omega_r}) V(\boldsymbol{x}, \boldsymbol{\omega_i}) L(\boldsymbol{\omega_i}) |\cos\theta_i| d\boldsymbol{\omega_i}$$

- Ω ist die Menge aller Richtungen: wg. Transparenz auch Richtungen unterhalb der Fläche (dann ist f_r eine BSDF statt einer BRDF), aber wir beschränken uns zunächst auf Reflexion

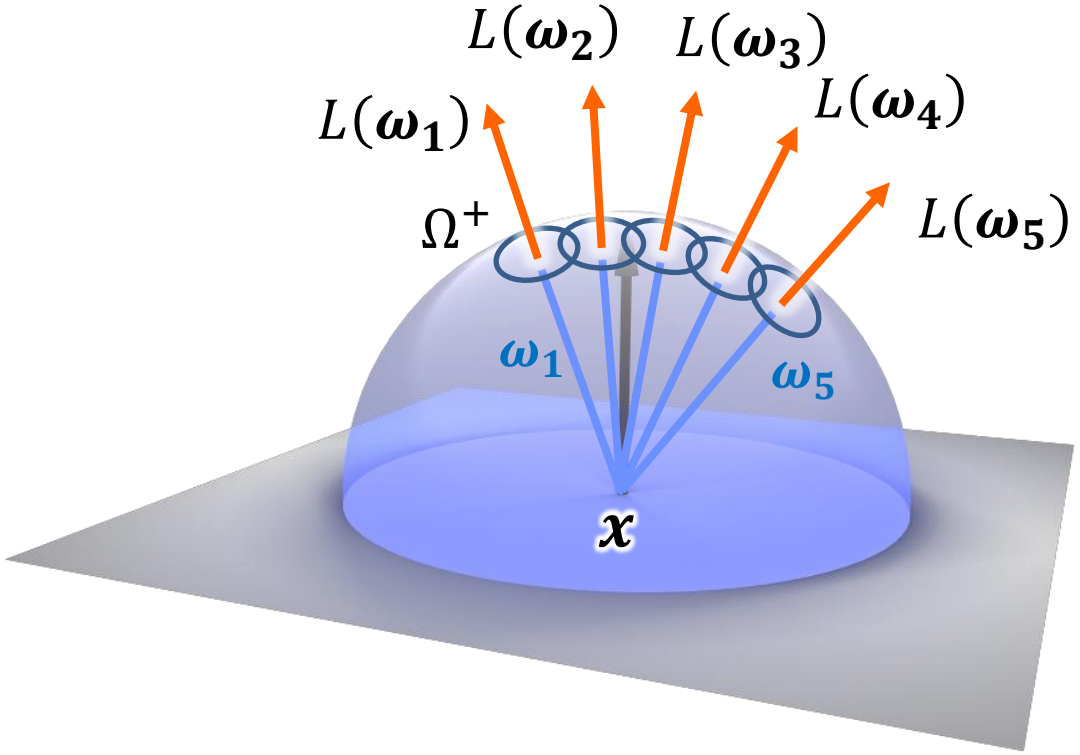


Ground-Truth, Reflexionsintegral

- Licht erreicht Oberflächen aus allen Richtungen (= Environment Map)
- Monte-Carlo Integration mit N uniform-verteilten Richtungen $\boldsymbol{\omega_i} \in \Omega$

$$L_r(\boldsymbol{x}, \boldsymbol{\omega_r}) \approx \frac{4\pi}{N} \sum_{i=1}^{N} f_r(\boldsymbol{\omega_i}, \boldsymbol{x}, \boldsymbol{\omega_r}) V(\boldsymbol{x}, \boldsymbol{\omega_i}) L(\boldsymbol{\omega_i}) |\boldsymbol{N_x} \cdot \boldsymbol{\omega_i}|$$

  - hier im Bild für die positive Hemisphäre Ω^+

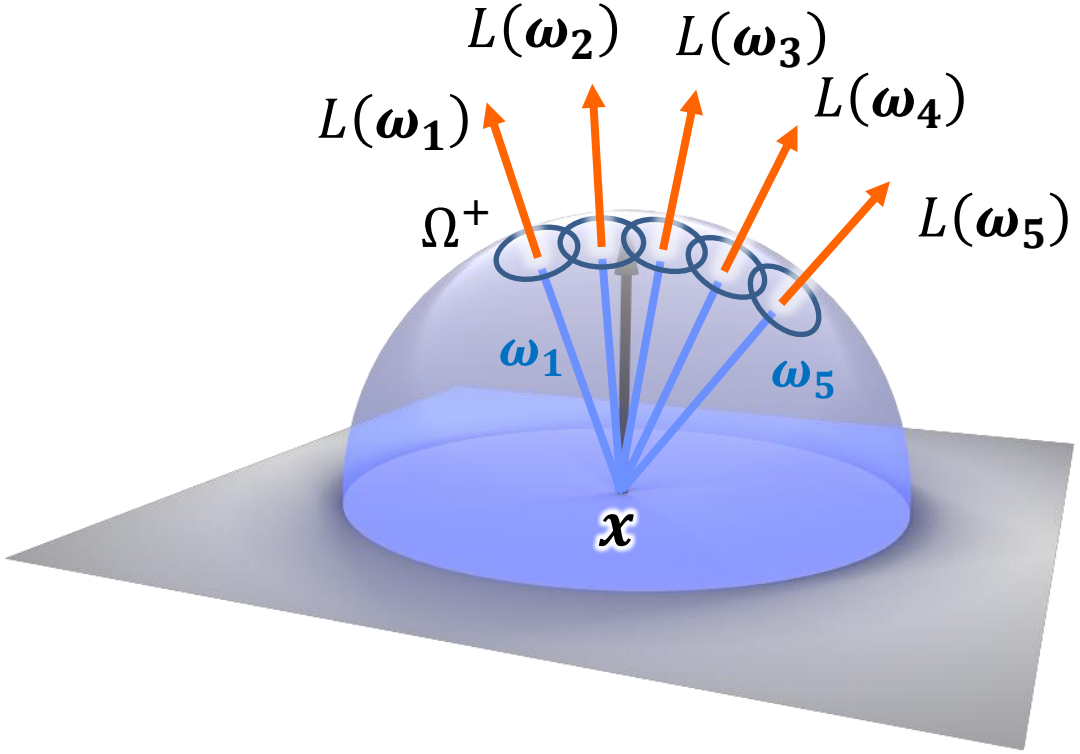


Ground-Truth, Reflexionsintegral

- Licht erreicht Oberflächen aus allen Richtungen (= Environment Map)
- Monte-Carlo Integration mit N uniform-verteilten Richtungen $\boldsymbol{\omega_i} \in \Omega^+$

$$L_r(\boldsymbol{x}, \boldsymbol{\omega_r}) \approx \frac{2\pi}{N} \sum_{i=1}^{N} f_r(\boldsymbol{\omega_i}, \boldsymbol{x}, \boldsymbol{\omega_r}) V(\boldsymbol{x}, \boldsymbol{\omega_i}) L(\boldsymbol{\omega_i}) (\boldsymbol{N_x} \cdot \boldsymbol{\omega_i})$$

  - hier im Bild für die positive Hemisphäre Ω^+

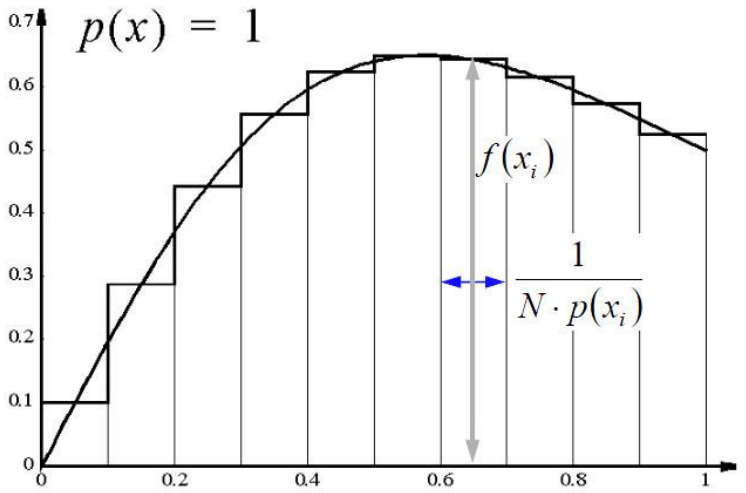


Monte Carlo Integration

Anschauliche Interpretation

- Monte Carlo Integration: numerische Integration mit Zufallszahlen
- das bestimmte Integral entspricht der Fläche unter der Kurve im Intervall
- man unterteilt die Fläche in Rechtecke der Höhe $f(x)$ und der Breite $\frac{1}{Np(x)}$
  - sind die Samples an einer Stelle dichter, dann ist die Breite des Rechtecks entsprechend schmäler

$$\int_a^b f(x)dx \approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)}$$



Monte Carlo Integration

Beispiel

- berechne $I = \int_0^1 5x^4 dx = 1$
- naïve Monte Carlo Integration mit $N = 1 \dots 1000$

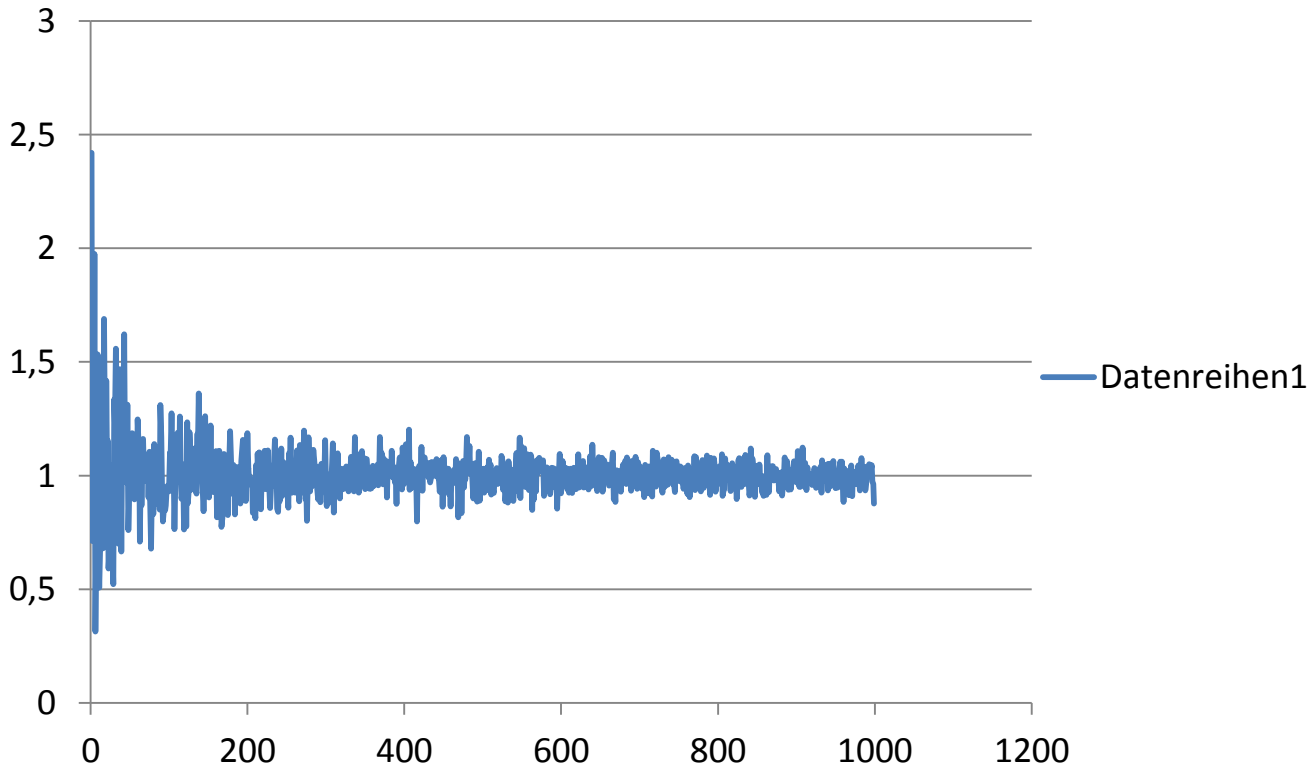
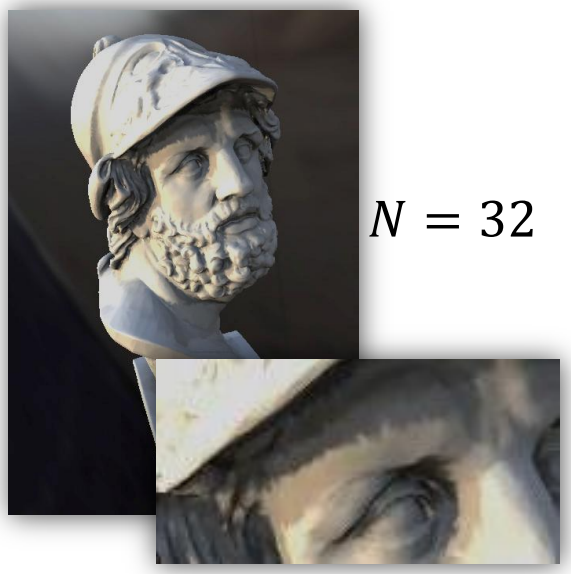
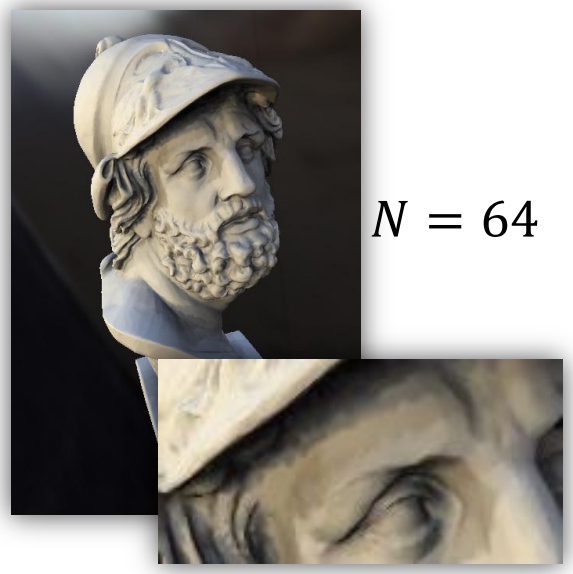


Image-based Lighting

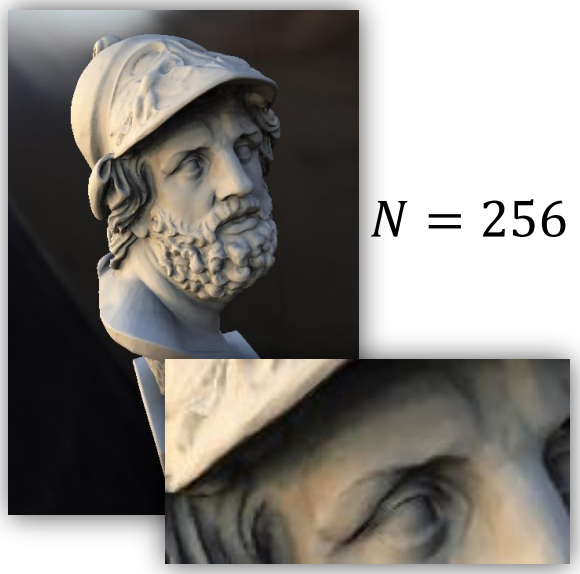
- Berechnung teuer, da N entsprechend groß sein muss…



$N = 32$



$N = 64$



$N = 256$

- Idee des **Precomputed Radiance Transfer**
  - berechne und speichere wie die **Helligkeit** eines Oberflächenpunkts von der **Beleuchtung** und der gesamten **Geometrie** abhängt
  - oder: „wie einfallendes Licht in reflektiertes Licht umgewandelt wird“
- benötigt besser geeignete Darstellung von Beleuchtung und Reflexion

Precomputed Radiance Transfer

Grundidee einfacher Fall: diffuse BRDF, direkte verschattete Beleuchtung

- diffuse BRDF bedeutet $f_r(\boldsymbol{\omega_i}, \boldsymbol{x}, \boldsymbol{\omega_r}) = \text{const}$
- Beleuchtung durch eine EnvMap $L(\boldsymbol{\omega_i})$
- Sichtbarkeitsfunktion $V(\boldsymbol{x}, \boldsymbol{\omega_i})$
- reflektiertes Licht am Oberflächenpunkt $\boldsymbol{x}$ ist:

$$L_r(\boldsymbol{x}, \cancel{\boldsymbol{\omega_r}}) = \underbrace{k_d(\boldsymbol{x})}_{\text{Albedo/BRDF}} \cdot \int_\Omega \underbrace{V(\boldsymbol{x}, \boldsymbol{\omega_i}) \cdot \max(0, \cos\theta_i)}_{\text{Transferfunktion}} \cdot \underbrace{L(\boldsymbol{\omega_i}) d\boldsymbol{\omega_i}}_{\text{direkte Beleuchtung}}$$

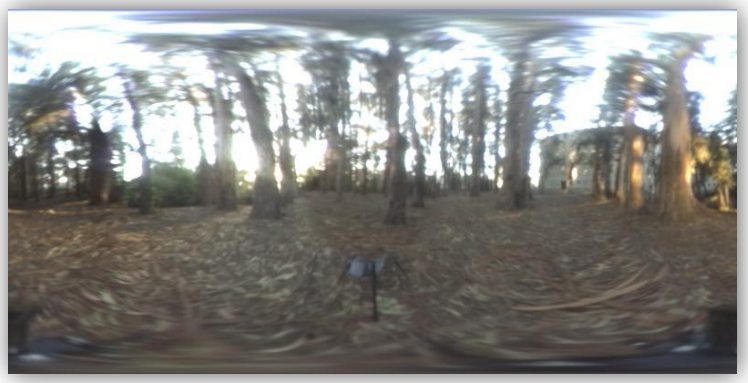
⇒ Ziel: Darstellung der Transferfunktion und Beleuchtung so, dass das Integral schnell berechnet werden kann!
(Transferfunktion ist abhängig vom Ort $\boldsymbol{x}$ auf der Oberfläche)

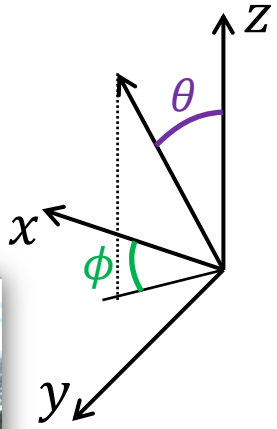
Precomputed Radiance Transfer

Idee in „Flatland“ (2D) und Definitionen

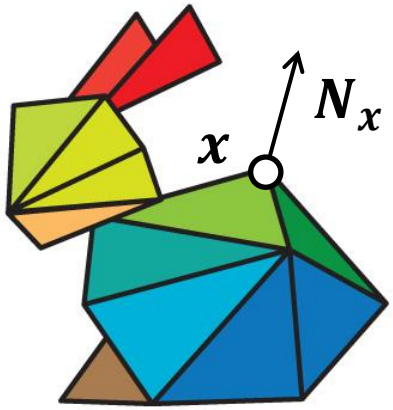
- Oberflächenpunkte für die wir etwas vorberechnen sind typischerweise Vertices eines Dreiecksnetzes
- Vorberechnung der Reflexion (abh. von Orientierung) und Verschattung

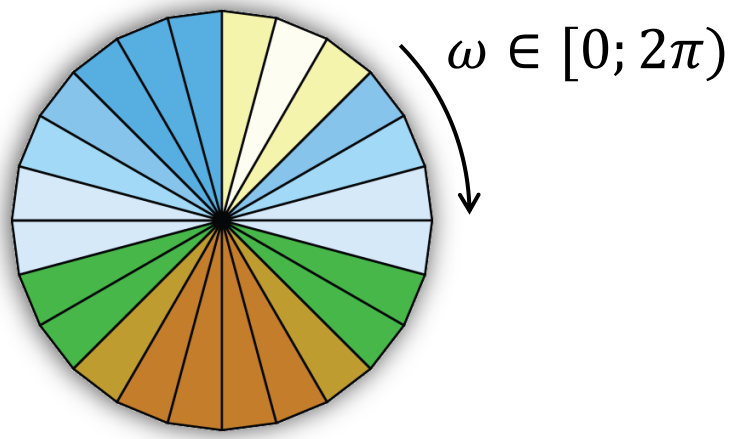
3D





2D

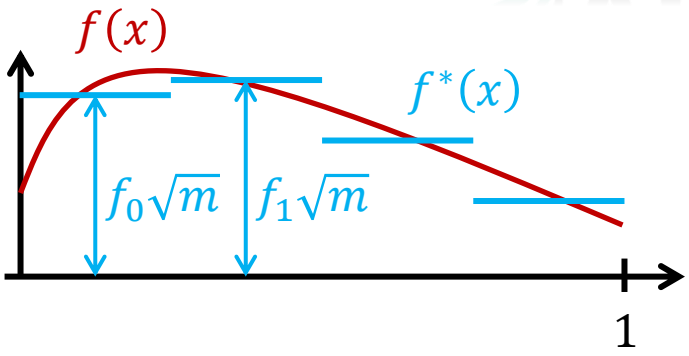




Hintergrund: Projektion von Funktionen

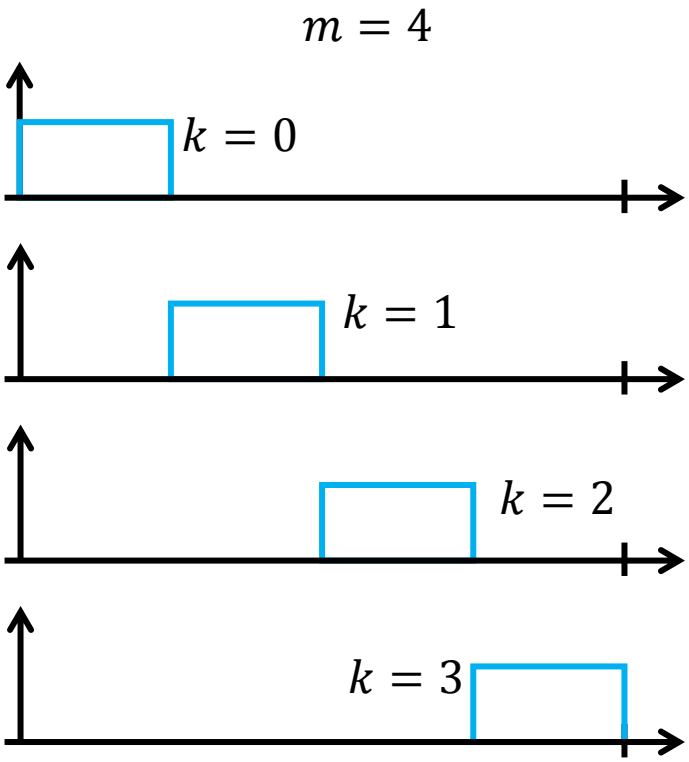
- jede Funktion $f(x)$ kann durch einen Satz von geeigneten Basisfunktionen $B_k(x)$ und Koeffizienten $\{f_k\}$ dargestellt/approx. werden

$$f(x) \approx f^*(x) = \sum_{k=1}^{m} f_k B_k(x)$$



- Beispiel:
  - stückweise-konstante Basis mit $x \in [0; 1)$

$$B_k(x) = \begin{cases} \sqrt{m} & {}^{k}/_{m} \leq x < {}^{k+1}/_{m} \\ 0 & else \end{cases}$$



- wie berechnet man $\{f_k\}$?

$$f_k = f \cdot B_k = \int_0^1 f(x) B_k(x) dx$$

(wenn $B_k(x)$ orthonormale Basis ist)

Hintergrund: Projektion von Funktionen

Berechnung der Koeffizienten

- geg. B_k und f, was sind die Koeffizienten f_k?
- Basisfunktionen spannen einen Vektorraum auf, ein **Skalarprodukt** ist ebenfalls **für Funktionen** definiert:

$$f, g: [a; b] \to \mathbb{R}$$

$$f \cdot g = \int_a^b f(x) g(x) dx$$

- (in einem geometrischen Vektorraum würden wir die Koeffizienten bzgl. einer Basis durch das Skalarprodukt berechnen – genauso auch hier!)
- wir interessieren uns für **orthonormale** Basen, in diesem Fall gilt
  - $B_i \cdot B_j = 0 \quad \Leftrightarrow \quad i \neq j$ und
  - $B_i \cdot B_j = 1 \quad \Leftrightarrow \quad i = j$
  - nur dann können wir Koeffizienten direkt berechnen mit $f_k = f \cdot B_k$

Precomputed Radiance Transfer

Environment Map und Basis in Flatland

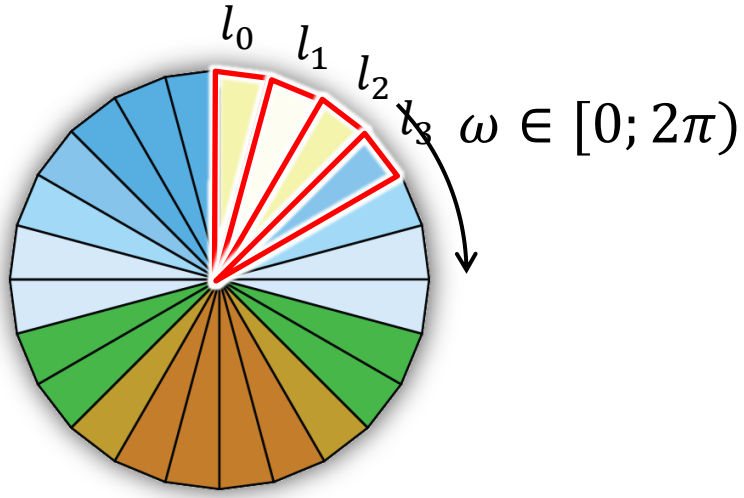
- Illustration mit stückweise-konstanter Basis mit m Funktionen $y_k(\omega)$

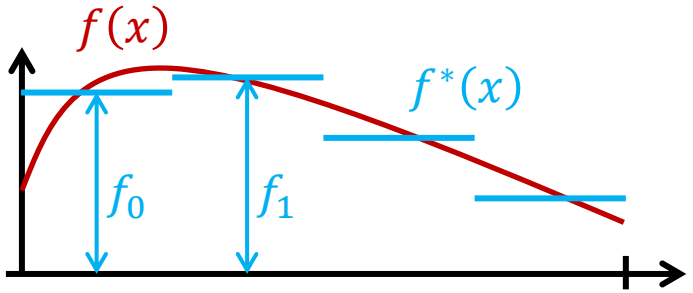
$$y_k(\omega) = \begin{cases} \sqrt{m/2\pi} & 2\pi\, {}^{k}/_{m} \le \omega < 2\pi\, {}^{k+1}/_{m} \\ 0 & sonst \end{cases} \quad \text{mit } \omega \in [0; 2\pi)$$

- Projektion der Flatland-EnvMap in diese Basis (hier: ein Farbkanal):

$$l_k = L \cdot y_k = \int_{2\pi} L(\omega_i) y_k(\omega_i) d\omega_i \approx \frac{2\pi}{N} \sum_{i=1}^{N} L(\omega_i) y_k(\omega_i)$$

- **in dieser Basis** beschreibt l_k wie viel Licht aus einem Richtungskegel ankommt





Precomputed Radiance Transfer

Diffuse Transferfunktion ohne Verdeckung

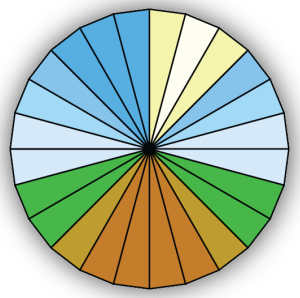
- anschließend definieren wir eine **Transferfunktion** für jeden Oberflächenpunkt/Vertex $\boldsymbol{x}$ mit Normale $\boldsymbol{N_x}$
- die Transferfunktion $T_{\boldsymbol{x}}(\omega)$ beschreibt, wie einfallendes Licht aus einer bestimmten Richtung zur Helligkeit der Oberfläche beiträgt

$$T_{\boldsymbol{x}}(\omega) = \max(0, \boldsymbol{N_x} \cdot \omega)$$

$$L_r(\boldsymbol{x}) = k_d(\boldsymbol{x}) \int_{2\pi} T_{\boldsymbol{x}}(\omega_i) L(\omega_i) d\omega_i$$

Albedo

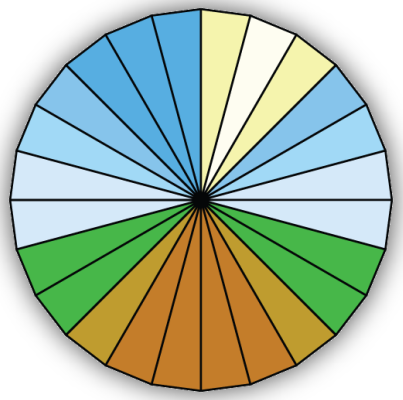
Transfer-funktion



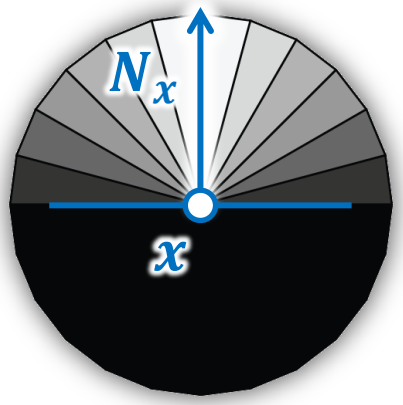
Precomputed Radiance Transfer

Idee: Beleuchtungsberechnung

- EnvMap projiziert in die Basis $y_k(\omega) \rightarrow$ Koeffizienten $\{l_k\}$
  - l_k beschreibt wie viel Licht aus einem bestimmten Richtungsintervall ankommt



- Transferfunktion $T_{\boldsymbol{x}}(\omega)$ bei $\boldsymbol{x}$ in Basis $y_k(\omega) \rightarrow$ Koeffizienten $t_{\boldsymbol{x},k}$
  - $t_{\boldsymbol{x},k} = T_{\boldsymbol{x}} \cdot y_k \approx \frac{2\pi}{N} \sum_{i=1}^{N} T_{\boldsymbol{x}}(\omega_i) y_k(\omega_i)$
  - $t_{\boldsymbol{x},k}$ beschreibt, wie Licht aus einem Richtungsintervall zum reflektierten Licht bei $\boldsymbol{x}$ beiträgt
  - Koeffizientenvektor $\{t_{\boldsymbol{x},k}\}$ wird pro Vertex gespeichert



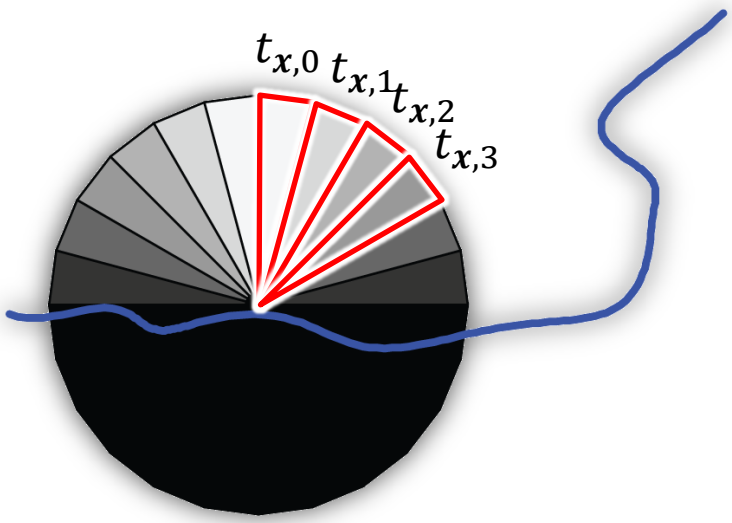
Precomputed Radiance Transfer

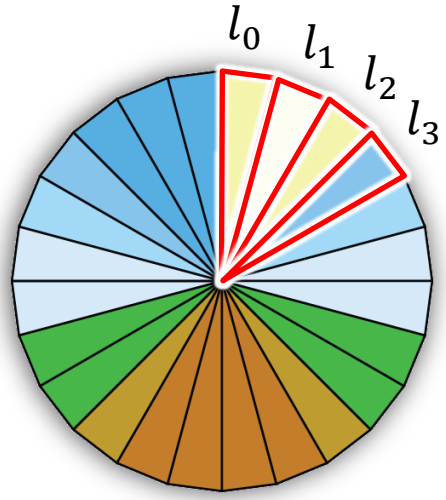
Idee: Beleuchtungsberechnung

- approximiere reflektiertes Licht mit

$$L_r(\boldsymbol{x}) = k_d(\boldsymbol{x}) \int_{2\pi} T_{\boldsymbol{x}}(\omega_i) L(\omega_i) d\omega_i \approx k_d(\boldsymbol{x}) \sum_{i=1}^{m} t_{\boldsymbol{x},i} \cdot l_i$$

  - Berechnung des Integrals wird zu Skalarprodukt zw. $\{l_k\}$ und $\{t_{\boldsymbol{x},k}\}$ **(wichtig: das gilt für alle orthonormalen Basen!)**
  - hier keine Überraschung: Basis ist eine Diskretisierung der Richtungen
- **was haben wir dadurch gewonnen? noch nichts… aber gleich!**





Precomputed Radiance Transfer

Integral über Produkt zweier Funktionen

- das Integral über das Produkt zweier Funktionen dargestellt in einer orthonormalen Basis lässt sich effizient berechnen: als Skalarprodukt der Koeffizientenvektoren

$$\int_{2\pi} a^*(\omega) b^*(\omega) d\omega =$$

$$\int_{2\pi} \sum_{i=1}^{m} a_i y_i(\omega) \sum_{j=1}^{m} b_j y_j(\omega)\, d\omega =$$

$$\sum_{i=1}^{m} \sum_{j=1}^{m} a_i b_j \int_{2\pi} y_i(\omega) y_j(\omega)\, d\omega =$$

$$\sum_{i=1}^{m} \sum_{j=1}^{m} a_i b_j\, \delta_{ij} = \sum_{i=1}^{m} a_i b_i$$

Orthonormalität der Basisfunktionen!

Precomputed Radiance Transfer

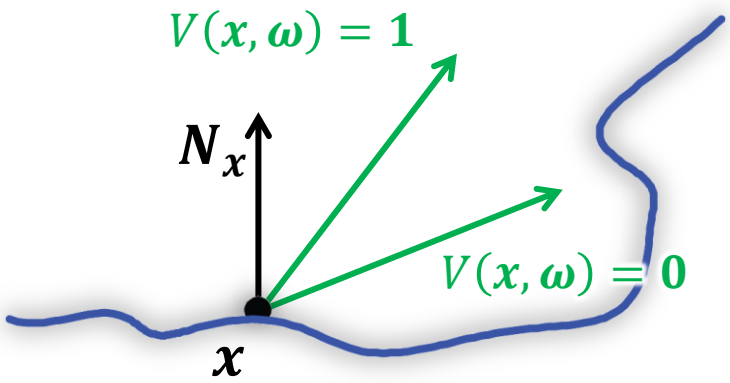
Transferfunktion: direkte Beleuchtung mit Verschattung

- wir können Sichtbarkeit bei $\boldsymbol{x}$ einfach in die Transferfunktion aufnehmen

$$T_{\boldsymbol{x}}(\omega) = \max(0, \boldsymbol{N_x} \cdot \omega) \cdot V(\boldsymbol{x}, \omega)$$

$$V(\boldsymbol{x}, \omega) = \begin{cases} 1 & \text{wenn Licht } \boldsymbol{x} \text{ aus Richtung } \omega \text{ erreicht} \\ 0 & sonst \end{cases}$$

- Projektion in Basis: $t_{\boldsymbol{x},k} \approx \frac{2\pi}{N} \sum_{i=1}^{N} \max(0, \boldsymbol{N_x} \cdot \omega_i) \cdot V(\boldsymbol{x}, \omega_i) \cdot y_k(\omega_i)$



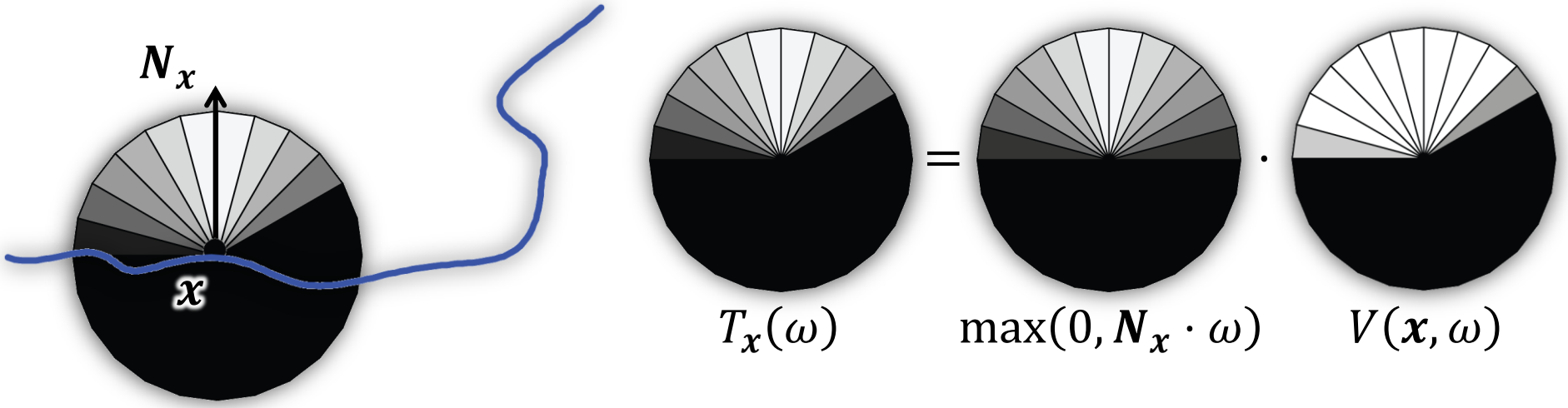
Precomputed Radiance Transfer

Transferfunktion: direkte Beleuchtung mit Verschattung

- wir können Sichtbarkeit bei $\boldsymbol{x}$ einfach in die Transferfunktion aufnehmen

$$T_{\boldsymbol{x}}(\omega) = \max(0, \boldsymbol{N_x} \cdot \omega) \cdot V(\boldsymbol{x}, \omega)$$

$$V(\boldsymbol{x}, \omega) = \begin{cases} 1 & \text{wenn Licht } \boldsymbol{x} \text{ aus Richtung } \omega \text{ erreicht} \\ 0 & sonst \end{cases}$$

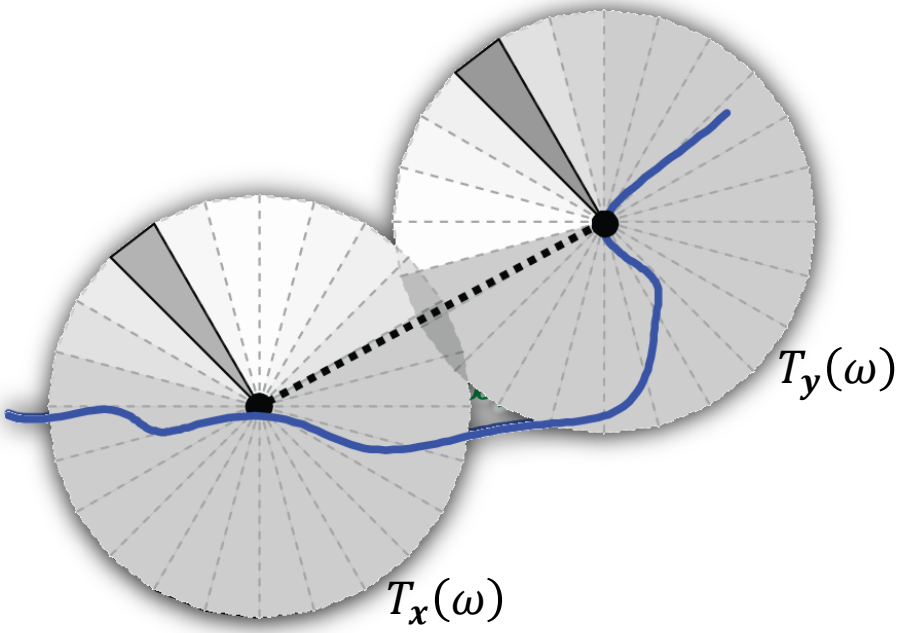


Precomputed Radiance Transfer

Transferfunktion mit (einfach-)indirekter Beleuchtung

- Prinzip: zur Reflexion kommt das Licht aus einer Richtung ω hinzu, das von allen anderen Oberflächenpunkten $\boldsymbol{y}$ zu $\boldsymbol{x}$ hinreflektiert wird

$$T'_{\boldsymbol{x}}(\omega) = T_{\boldsymbol{x}}(\omega) + T_{\boldsymbol{y}}(\omega) \cdot \underbrace{k_d(\boldsymbol{y})}_{\text{Albedo bei } \boldsymbol{y}} \cdot \underbrace{\max(0, \boldsymbol{N_x} \cdot \omega)}_{\substack{\text{verkürzter} \\ \text{Raumwinkel bei } \boldsymbol{x}}} \cdot \underbrace{V(\boldsymbol{x}, \boldsymbol{y})}_{\substack{\text{wenn } \boldsymbol{x} \text{ und } \boldsymbol{y} \\ \text{ggs. sichtbar}}}$$

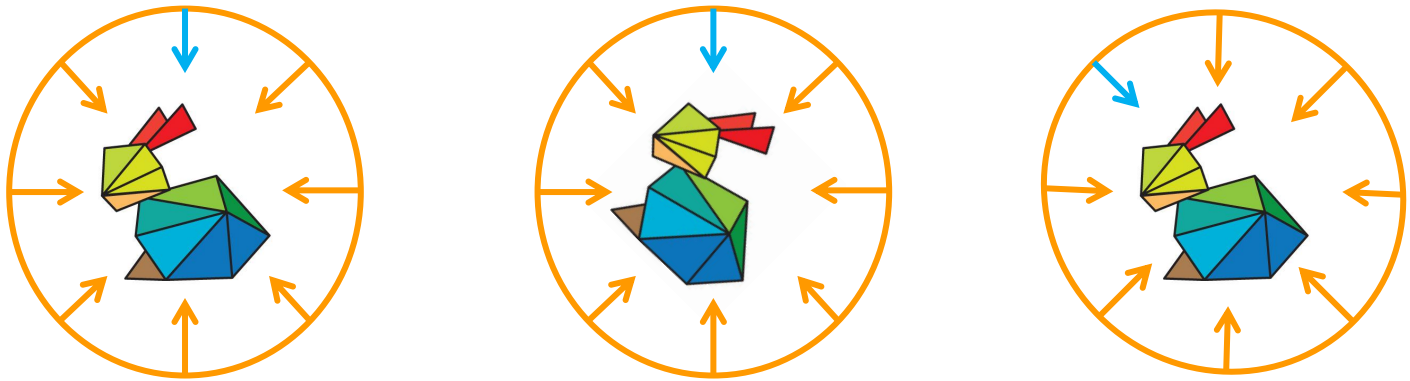


Precomputed Radiance Transfer

Rendering-Algorithmus

- Vorberechnung von $\{t_{x,k}\}$, d.h. die **Geometrie ist** danach **statisch**
- zur Laufzeit
  - berechne $\{l_k\}$
    (erlaubt Änderung der Beleuchtung und starre Rotation des Objekts)
  - für jeden Vertex: berechne Beleuchtung mit $k_d(\boldsymbol{x}) \sum_{i=1}^{m} t_{x,k} \cdot l_k$
  - auch möglich: interpoliere $\{t_{x,k}\}$ zwischen den Vertices

- Beispiel: soll das Objekt gedreht werden (Mitte), dann können die Transferkoeffizienten $\{t_{x,k}\}$ beibehalten werden und $\{l_k\}$ mit der inversen Rotation neu berechnet werden (rechts)



Precomputed Radiance Transfer

Rendering-Algorithmus

- Vorberechnung von $\{t_{x,k}\}$, d.h. die **Geometrie ist** danach **statisch**
- zur Laufzeit
  - berechne $\{l_k\}$
    (erlaubt Änderung der Beleuchtung und starre Rotation des Objekts)
  - für jeden Vertex: berechne Beleuchtung mit $k_d(\boldsymbol{x}) \sum_{i=1}^{m} t_{x,k} \cdot l_k$
  - auch möglich: interpoliere $\{t_{x,k}\}$ zwischen den Vertices

- die verbleibende Frage: welche Basis verwenden wir (in 3D)?
  - muss über alle Richtungen definiert sein
  - sollte mit wenigen Basisfunktionen auskommen
  - sollte glatt sein (nicht stückweise-konstant!)

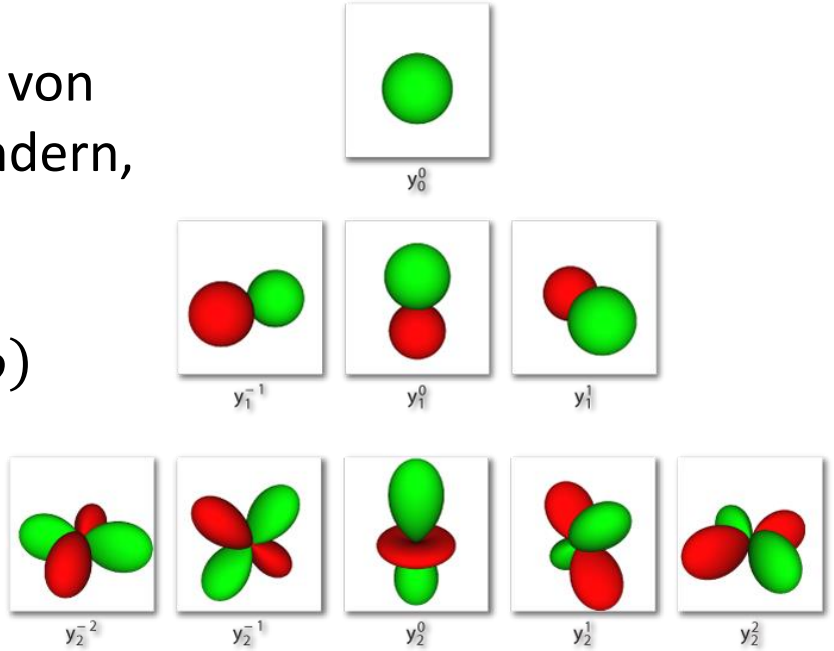
Kugelflächenfunktionen, Spherical Harmonics

- wir verwenden sog. Kugelflächenfunktionen (Spherical Harmonics, SH)
  - Basisfunktionen werden typischerweise indiziert mit Band l und innerhalb der Bänder mit „Quantenzahl“ $-l \leq m \leq l$
  - Basisfunktionen in niedrigen Bändern sind glatt/niederfrequent und werden mit zunehmendem l hochfrequenter, es gilt:

$$f(\theta,\phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} f_l^m y_l^m(\theta,\phi)$$

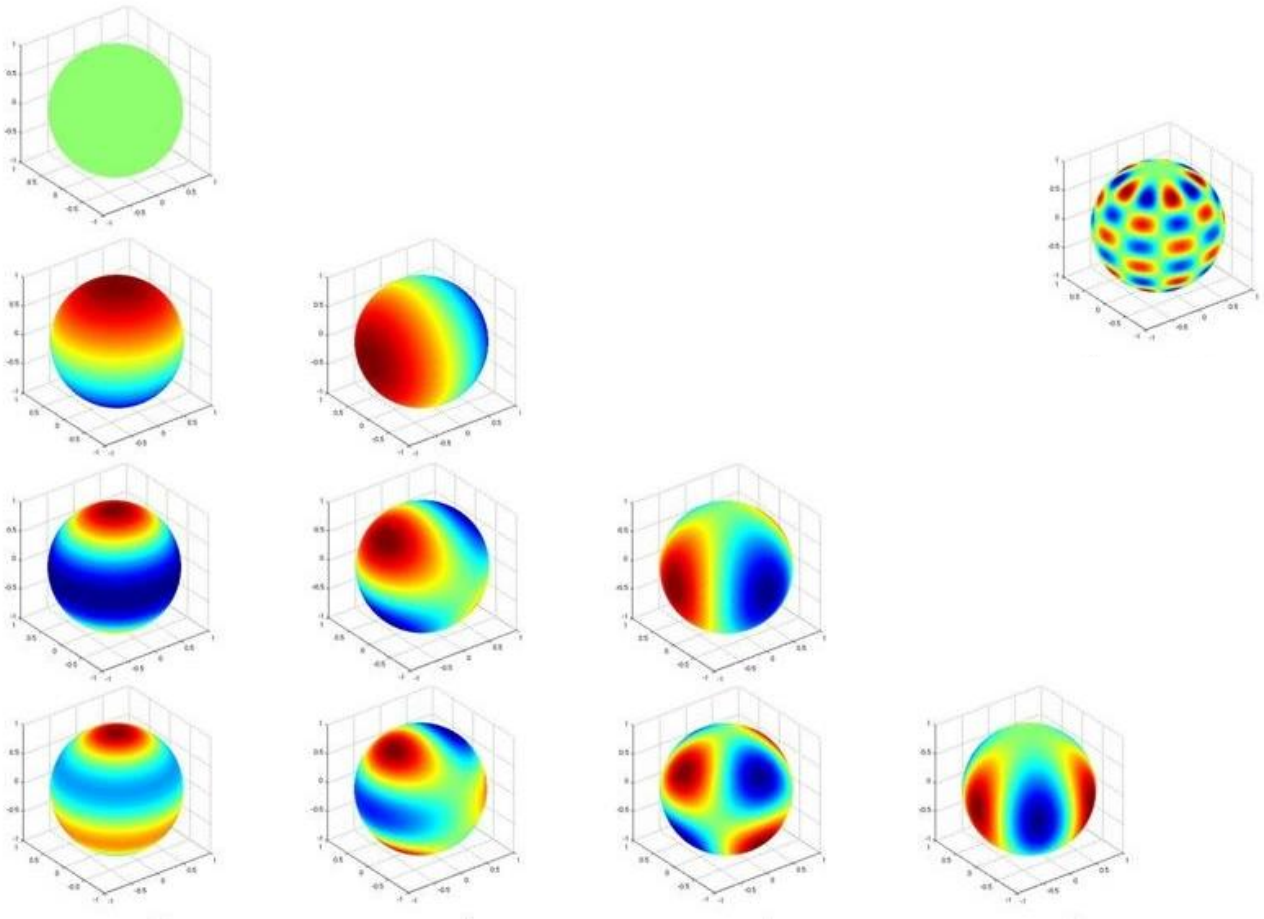
  - eine niederfrequente Approximation von $f(\theta,\phi)$ erhalten wir mit weniger Bändern, z.B. $n = 3$ (ergibt n^2 Koeffizienten)

$$f(\theta,\phi) \approx \sum_{l=0}^{n-1} \sum_{m=-l}^{l} f_l^m y_l^m(\theta,\phi)$$



Kugelflächenfunktionen, Spherical Harmonics

- Darstellung als Farbkodierung auf Kugeloberflächen



Reelle Spherical Harmonics

- woher kommen die Kugelflächenfunktionen?
  - Eigenfunktionen zum Winkelanteil des Laplace-Operators (Teil der Laplace-Gleichung, Beschreibung elektrischer Felder)
  - eine Funktion, die die Laplacesche DGL erfüllt nennt man harmonisch
  - Anwendungen: theoretische Physik (Elektrodynamik, Beschreibung von Bahndrehimpulsen, Quantenmechanik), Geodäsie, Atomorbitale…

$$y_0^0(\theta,\varphi) = \frac{1}{2}\sqrt{\frac{1}{\pi}} \qquad y_0^0(x,y,z) = \frac{1}{2}\sqrt{\frac{1}{\pi}}$$

$$y_1^{-1}(\theta,\varphi) = -\frac{1}{2}\sqrt{\frac{3}{\pi}}\sin\theta\sin\varphi \qquad y_1^{-1}(x,y,z) = -\frac{1}{2}\sqrt{\frac{3}{\pi}}y$$

$$y_1^0(\theta,\varphi) = \frac{1}{2}\sqrt{\frac{3}{\pi}}\cos\theta \qquad y_1^0(x,y,z) = \frac{1}{2}\sqrt{\frac{3}{\pi}}z$$

$$y_1^1(\theta,\varphi) = -\frac{1}{2}\sqrt{\frac{3}{\pi}}\sin\theta\cos\varphi \qquad y_1^1(x,y,z) = -\frac{1}{2}\sqrt{\frac{3}{\pi}}x$$

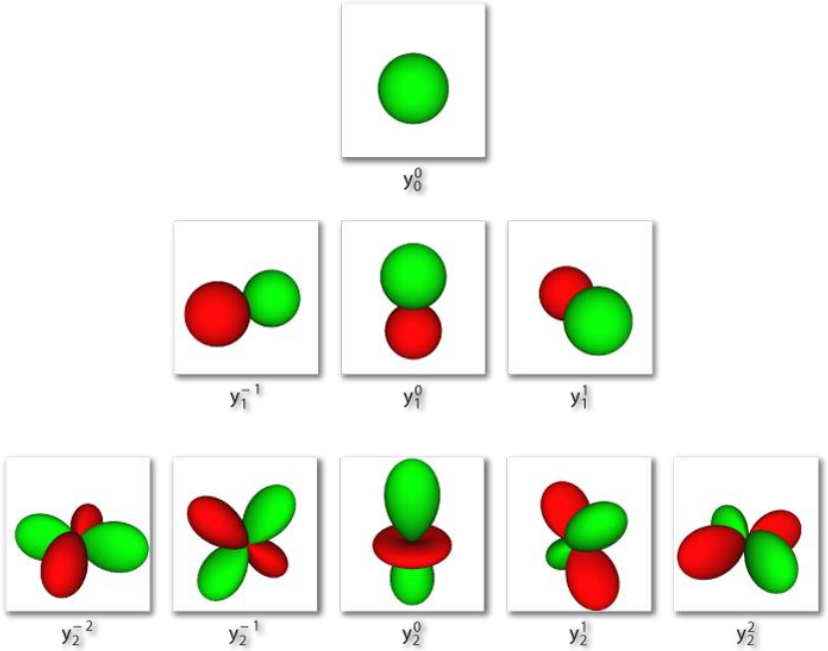
$$y_2^{-2}(\theta,\varphi) = \frac{1}{2}\sqrt{\frac{15}{\pi}}\sin^2\theta\sin\varphi\cos\varphi \qquad y_2^{-2}(x,y,z) = \frac{1}{2}\sqrt{\frac{15}{\pi}}xy$$

$$y_2^{-1}(\theta,\varphi) = -\frac{1}{2}\sqrt{\frac{15}{\pi}}\sin\theta\cos\theta\sin\varphi \qquad y_2^{-1}(x,y,z) = -\frac{1}{2}\sqrt{\frac{15}{\pi}}yz$$

$$y_2^0(\theta,\varphi) = \frac{1}{4}\sqrt{\frac{5}{\pi}}\left(3\cos^2\theta - 1\right) \qquad y_2^0(x,y,z) = \frac{1}{4}\sqrt{\frac{5}{\pi}}\left(3z^2 - 1\right)$$

$$y_2^1(\theta,\varphi) = -\frac{1}{2}\sqrt{\frac{15}{\pi}}\sin\theta\cos\theta\cos\varphi \qquad y_2^1(x,y,z) = -\frac{1}{2}\sqrt{\frac{15}{\pi}}xz$$

$$y_2^2(\theta,\varphi) = \frac{1}{2}\sqrt{\frac{15}{\pi}}\sin^2\theta\left(2\cos^2\varphi - 1\right) \qquad y_2^2(x,y,z) = \frac{1}{2}\sqrt{\frac{15}{\pi}}\left(x^2 - y^2\right)$$



Spherical Harmonics (allgemein)

Nur damit Sie es nachschlagen können…

- komplexe Basisfunktionen $Y_l^m(\theta,\phi) = K_l^m P_l^{|m|}(\cos\theta) e^{im\phi}$
- die reelle Basis bildet eine Untermenge

$$y_l^m = \begin{cases} \sqrt{2}\,\mathrm{Re}(Y_l^m) = \sqrt{2} K_l^m P_l^m(\cos\theta)\cos(m\phi) & m > 0 \\ \sqrt{2}\,\mathrm{Im}(Y_l^m) = \sqrt{2} K_l^m P_l^{-m}(\cos\theta)\sin(-m\phi) & m < 0 \\ Y_l^0 = K_l^0 P_l^0(\cos\theta) & m = 0 \end{cases}$$

- Normalisierungsterm

$$K_l^m = \sqrt{\frac{(2l+1)}{4\pi}\frac{(l-|m|)!}{(l+|m|)!}} \qquad \int_{\theta=0}^{\pi}\int_{\phi=0}^{2\pi} Y_l^m(\theta,\phi) Y_{l'}^{m'}(\theta,\phi)\sin\theta\, d\phi\, d\theta = \delta_{ll'}\delta_{mm'}$$

- Legendreschen Polynome mit Doppelfakultät

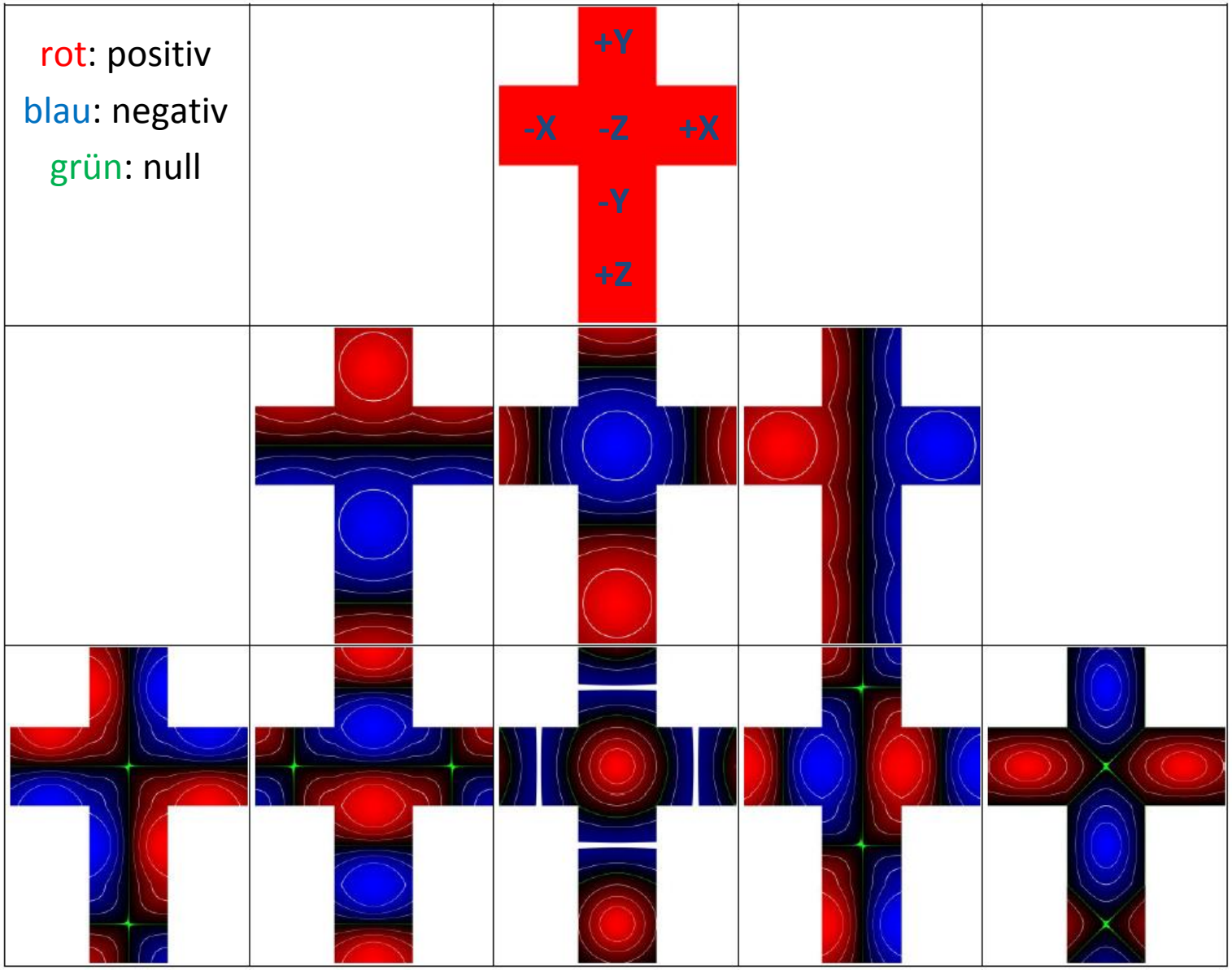
$$P_l^m(x) = \frac{x(2l-1)P_{l-1}^m - (l+m-1)P_{l-2}^m}{l-m}$$

$$P_m^m(x) = (-1)^m (2m-1)!!\left(\sqrt{1-x^2}\right)^m$$

$$P_{m+1}^m(x) = x(2m+1)P_m^m,\ P_0^0 = 1$$

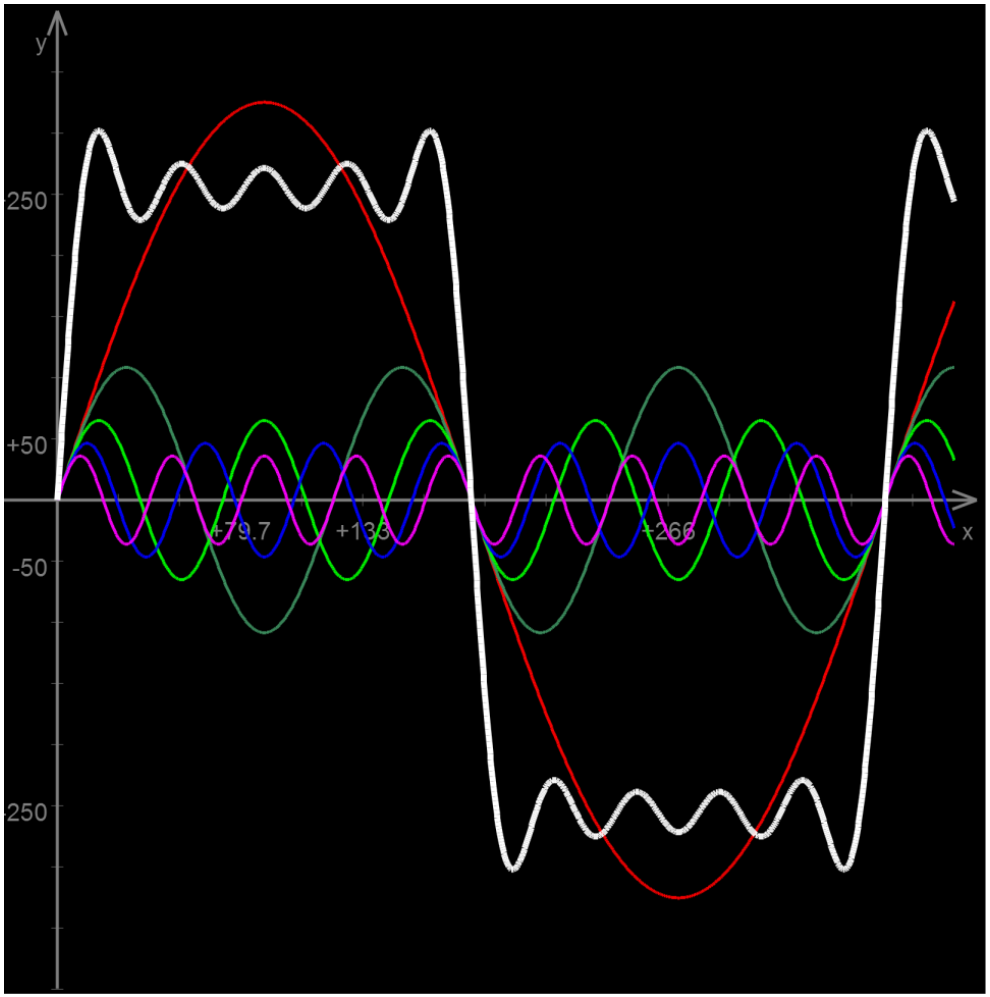
$$n!! = \begin{cases} n\cdot(n-2)\cdot\ldots\cdot 3\cdot 1 & n>0 \text{ ungerade} \\ n\cdot(n-2)\cdot\ldots\cdot 4\cdot 2 & n>0 \text{ gerade} \\ 1 & n\in\{-1,0\} \end{cases}$$

Spherical Harmonics auf den Seiten einer Cube-Map



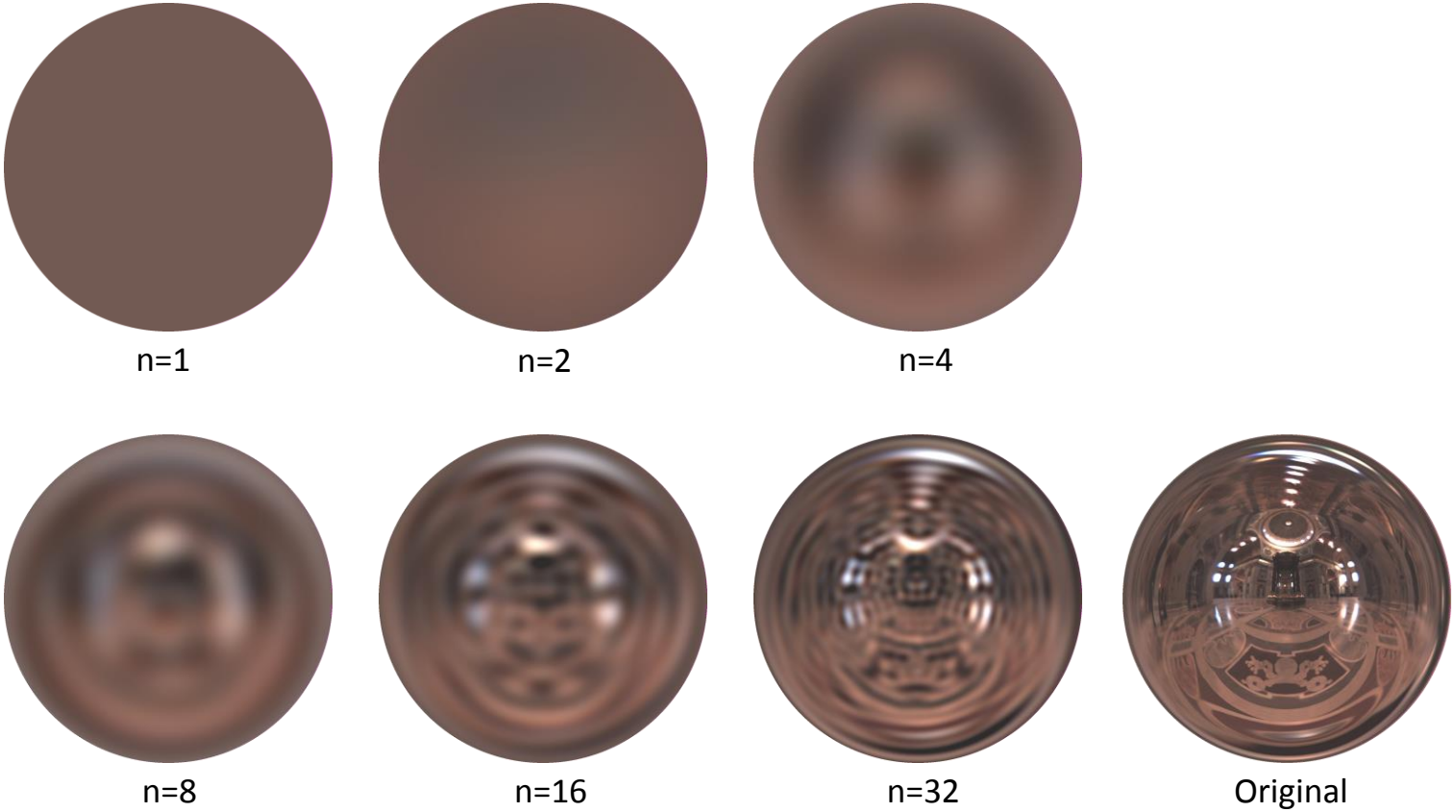
Spherical Harmonics und Frequenzen in Bildern

- eine Environment Map (hier als LatLong-Bild dargestellt) kann in der SH-Basis dargestellt werden
- Analogon: Überlagerung von Schwingungen unterschiedlicher Frequenz



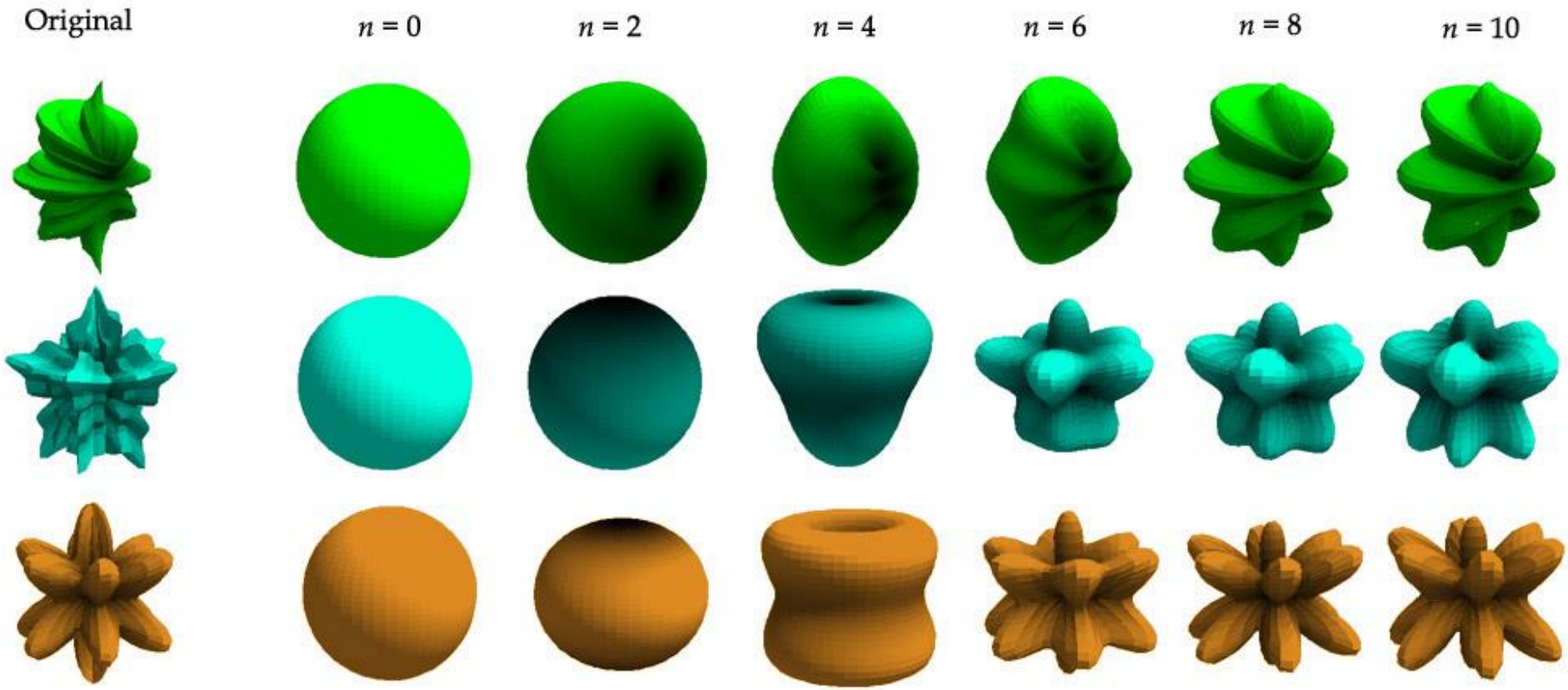
Approximation mit SH

- wir wollen u.a. eine Environment Map mit SH darstellen
  (für RGB: ein Koeffizientenvektor pro Farbkanal)
- wir wollen Integrale über Beleuchtung und BRDF berechnen, d.h. wir haben es mit (meist) glatten Funktionen zu tun, die mit wenigen Koeffizienten ausreichend gut dargestellt werden können!



Approximation mit SH

- weiteres Beispiel: hier wird eine Abstandsfunktion $f: \Omega \to \mathbb{R}$ (dargestellt als Fläche) mit SH approximiert



Spherical Harmonics

Indizierung, Schreibweise

- oft verwendet man folgende Indizierung der Basisfunktionen

$$f(\theta,\phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} f_l^m y_l^m(\theta,\phi) = \sum_{i=1}^{\infty} f_i y_i(\theta,\phi)$$

mit $i = l(l+1) + m + 1$, d.h. die Reihenfolge ist $(0,0)$, $(1,-1)$, $(1,0)$, $(1,1)$, $(2,-2)$, $(2,-1)$, …

- man verwendet üblicherweise alle Funktionen der jeweils ersten n Bänder, d.h. also n^2 Koeffizienten

$$f(\theta,\phi) \approx f^*(\theta,\phi) = \sum_{l=0}^{n-1} \sum_{m=-l}^{l} f_l^m y_l^m(\theta,\phi) = \sum_{i=1}^{n^2} f_i y_i(\theta,\phi)$$

Spherical Harmonics

Berechnung der Koeffizienten (im Prinzip wie bisher)

- … über das Skalarprodukt für Funktionen

$$f(\theta,\phi) \approx f^*(\theta,\phi) = \sum_{i=1}^{n^2} f_i y_i(\theta,\phi)$$

$$f_i = \int_\Omega f(\boldsymbol{\omega}) y_i(\boldsymbol{\omega}) d\boldsymbol{\omega} = \int_0^{2\pi} \int_0^{\pi} f(\theta,\phi) y_i(\theta,\phi) \sin\theta \, d\theta d\phi$$

- in der Praxis verwenden wir Monte Carlo Integration
  - z.B. mit N uniform verteilten Richtungen $\boldsymbol{\omega_j}$

$$f_i \approx \frac{4\pi}{N} \sum_{j=1}^{N} f(\boldsymbol{\omega_j}) y_i(\boldsymbol{\omega_j})$$

Spherical Harmonics

Integral über Produkt zweier Funktionen

- mit SH lässt sich das Integral über das Produkt zweier mit SH approximierter Funktionen effizient berechnen (vgl. stückweise-konstante Basis), z.B. Integral über Produkt aus Transferfkt. und Licht

$$\int_\Omega a^*(\boldsymbol{\omega}) b^*(\boldsymbol{\omega}) d\boldsymbol{\omega} =$$

$$\int_\Omega \sum_{i=1}^{n^2} a_i y_i(\boldsymbol{\omega}) \sum_{j=1}^{n^2} b_j y_j(\boldsymbol{\omega})\, d\boldsymbol{\omega} =$$

$$\sum_{i=1}^{n^2} \sum_{j=1}^{n^2} a_i b_j \int_\Omega y_i(\boldsymbol{\omega}) y_j(\boldsymbol{\omega})\, d\boldsymbol{\omega} =$$

$$\sum_{i=1}^{n^2} \sum_{j=1}^{n^2} a_i b_j\, \delta_{ij} = \sum_{i=1}^{n^2} a_i b_i$$

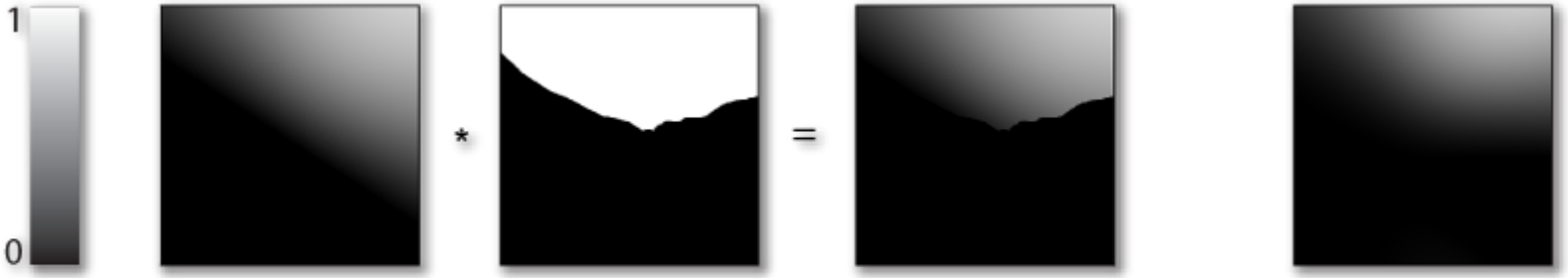
Orthonormalität der Basisfunktionen!

PRT mit Verschattung und SH

Beispiel: diffuser verschatteter Transfer, Transferfunktion

$$T_x(\boldsymbol{\omega}) = \begin{cases} \frac{1}{\pi}\langle \boldsymbol{n_x}, \boldsymbol{\omega}\rangle V(\boldsymbol{x}, \boldsymbol{\omega}) & \langle \boldsymbol{n_x}, \boldsymbol{\omega}\rangle \geq 0 \\ 0 & sonst \end{cases}$$

$$V(\boldsymbol{x}, \boldsymbol{\omega}) = \begin{cases} 1 & \text{wenn bei } \boldsymbol{x} \text{ Umgebungslicht aus Richtung } \boldsymbol{\omega} \text{ ankommt} \\ 0 & sonst \end{cases}$$



$\langle \boldsymbol{n_x}, \boldsymbol{\omega}\rangle^+$ $V(\boldsymbol{x}, \boldsymbol{\omega})$ $\langle \boldsymbol{n_x}, \boldsymbol{\omega}\rangle^+ V(\boldsymbol{x}, \boldsymbol{\omega})$

- $t_{x,i}$ beschreibt wir das Licht – projiziert in die Basisfunktion $y_i(\boldsymbol{\omega})$ – zum reflektierten Licht bei $\boldsymbol{x}$ beiträgt

PRT mit Verschattung und SH

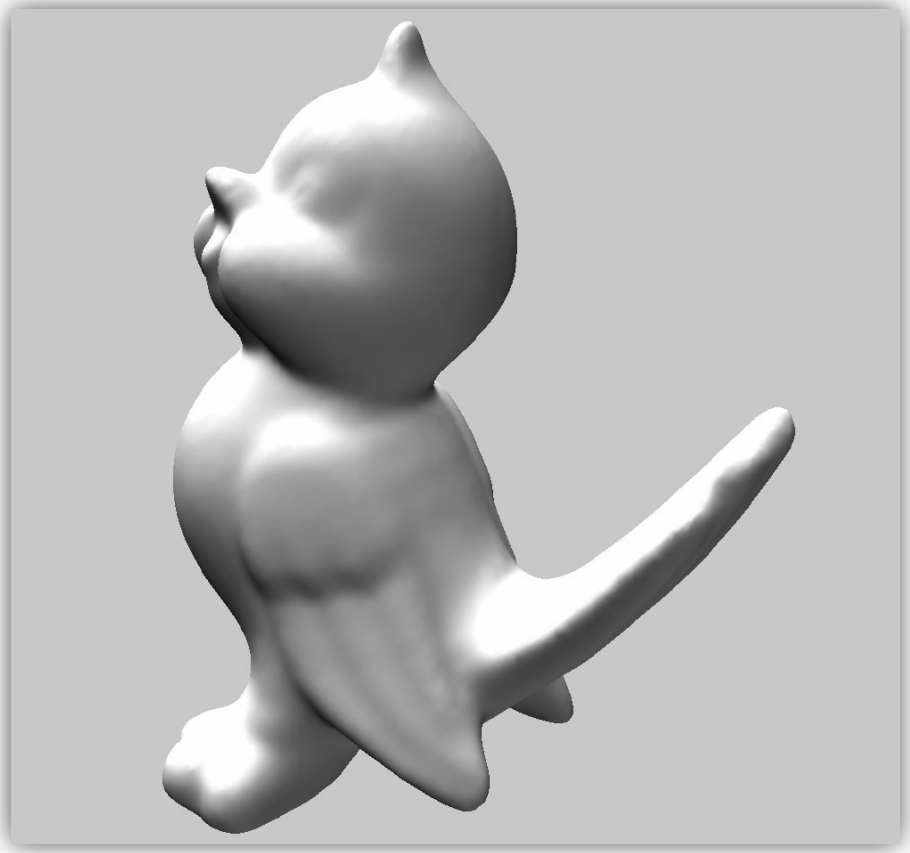
Beispiel: diffuser verschatteter Transfer, Beleuchtung

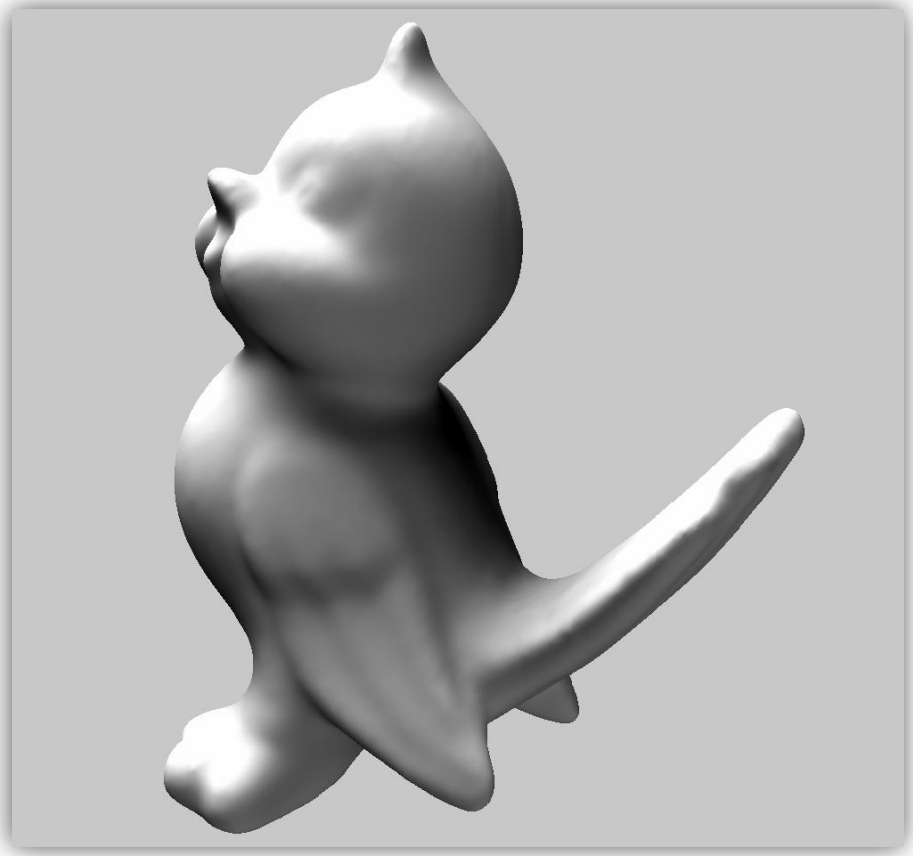
$$L_r(\boldsymbol{x}) = \underbrace{k_d(\boldsymbol{x})}_{\text{Albedo}} \int_{2\pi} \underbrace{\frac{1}{\pi} T_{\boldsymbol{x}}(\omega_i)}_{\text{Transfer-funktion}} \underbrace{L(\omega_i)}_{} d\omega_i$$

- $L_i(\boldsymbol{\omega_i})$ projiziert in 5-SH-Bänder und rekonstruiert

PRT Beispiel

- Transport ohne (links) und mit Verschattung (rechts)
- Resultate dank SH (nur 3 Bänder, also 9 Koeffizienten pro Vertex!) glatt



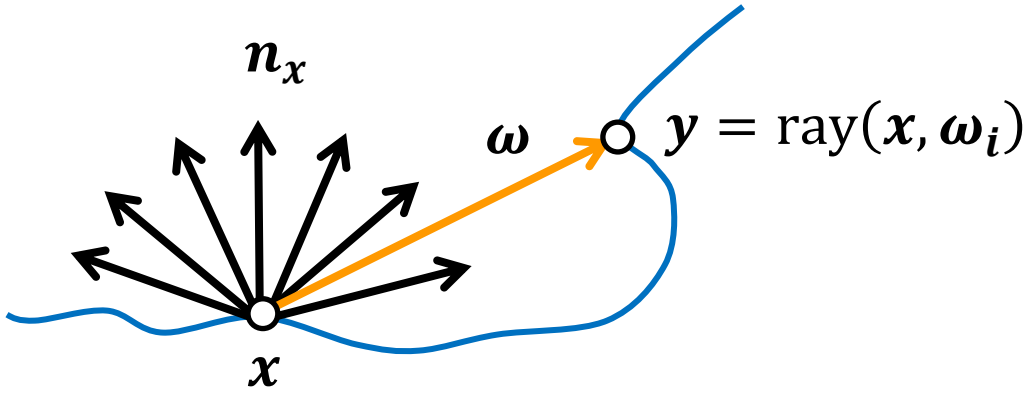


PRT: Transfer mit Mehrfachreflexion

- Mehrfachreflexion: wenn der Strahl von $\boldsymbol{x}$ in Richtung $\boldsymbol{\omega}$ eine Oberfläche bei $\boldsymbol{y} = \mathrm{ray}(\boldsymbol{x}, \boldsymbol{\omega})$ schneidet, dann muss die dort reflektierte Radiance zum Transport bei $\boldsymbol{x}$ beitragen

$$L_r(\boldsymbol{x}) = k_d(\boldsymbol{x}) \left(\int_{\Omega^+} \frac{1}{\pi} \langle \boldsymbol{n_x}, \boldsymbol{\omega_i} \rangle V(\boldsymbol{x}, \boldsymbol{\omega_i}) L_i(\boldsymbol{\omega_i}) d\boldsymbol{\omega_i} \right.$$

$$\left. + \int_{\Omega^+} \frac{1}{\pi} \langle \boldsymbol{n_x}, \boldsymbol{\omega_i} \rangle \big(1 - V(\boldsymbol{x}, \boldsymbol{\omega_i})\big) L_r\big(\mathrm{ray}(\boldsymbol{x}, \boldsymbol{\omega_i})\big) d\boldsymbol{\omega_i} \right)$$



PRT: Transfer mit Interreflexion

- von der Darstellung als Neumann Reihe wissen wir, dass wir n-fach reflektiertes Licht aus den $(n-1)$-fach reflektierten berechnen können
- direkte Beleuchtung

$$L_r^0(\boldsymbol{x}) = k_d(\boldsymbol{x}) \int_{\Omega^+} \frac{1}{\pi} \langle \boldsymbol{n_x}, \boldsymbol{\omega_i} \rangle V(\boldsymbol{x}, \boldsymbol{\omega_i}) L_i(\boldsymbol{\omega_i}) d\boldsymbol{\omega_i}$$

- indirekte Beleuchtung unter Verwendung der vorherigen Iteration(en) (b=„bounce“, $b > 0$)

$$L_r^b(\boldsymbol{x}) = k_d(\boldsymbol{x}) \int_{\Omega^+} \frac{1}{\pi} \langle \boldsymbol{n_x}, \boldsymbol{\omega_i} \rangle \big(1 - V(\boldsymbol{x}, \boldsymbol{\omega_i})\big) L_r^{b-1}(\boldsymbol{y}) d\boldsymbol{\omega_i}$$

- direkte und indirekte Beleuchtung mit bis zu B Indirektionen

$$L_r(\boldsymbol{x}) = \sum_{b=0}^{B} L_r^b(\boldsymbol{x})$$

- wie berechnet man die Transferkoeffizienten? Muss man jeden „Bounce“ separat projizieren?

PRT: Transfer mit Interreflexion

BONUS

Nebenrechnung (Bonusmaterial)

$$L_r^1(\boldsymbol{x}) = k_d(\boldsymbol{x}) \int_{\Omega^+} \frac{1}{\pi} \langle \boldsymbol{n_x}, \boldsymbol{\omega_i} \rangle \big(1 - V(\boldsymbol{x}, \boldsymbol{\omega_i})\big) L_r^0(\boldsymbol{y}) d\boldsymbol{\omega_i}$$

- der Integrand ist $\neq 0$ für eine Menge von Punkten $\boldsymbol{Y}$ (Oberflächen, die für Richtungen ω_i getroffen werden)

$$L_r^1(\boldsymbol{x}) = k_d(\boldsymbol{x}) \int_{\boldsymbol{y} \in \boldsymbol{Y}} T_{\boldsymbol{x}} \left(\frac{\boldsymbol{y} - \boldsymbol{x}}{\|\boldsymbol{y} - \boldsymbol{x}\|} \right) L_r^0(\boldsymbol{y}) d\boldsymbol{y}$$

  - das Integral über $\boldsymbol{Y}$ ist nur eine Schreibweise und macht die Herleitung etwas übersichtlicher…

- Einsetzen des Reflexionsintegrals bei $\boldsymbol{y}$

$$L_r^1(\boldsymbol{x}) = k_d(\boldsymbol{x}) \int_{\boldsymbol{y} \in \boldsymbol{Y}} T_{\boldsymbol{x}}(.) k_d(\boldsymbol{y}) \int_{\Omega^+} T_{\boldsymbol{y}}(\boldsymbol{\omega_j}) L_i(\boldsymbol{\omega_j}) d\boldsymbol{\omega_j} \, d\boldsymbol{y}$$

PRT: Transfer mit Interreflexion

BONUS

Nebenrechnung (Bonusmaterial)

- Terme umordnen

$$L_r^1(\boldsymbol{x}) = k_d(\boldsymbol{x}) \int_{\Omega^+} \underbrace{\int_{\boldsymbol{y}\in\boldsymbol{Y}} T_{\boldsymbol{x}}(.)\, k_d(\boldsymbol{y})\, T_{\boldsymbol{y}}(\boldsymbol{\omega_j})\, d\boldsymbol{y}}_{\text{Transferfunktion für einfach indirektes Licht}} \underbrace{L_i(\boldsymbol{\omega_j})}_{\text{Licht}}\, d\boldsymbol{\omega_j}$$

- Projektion in Basis (und nochmal umsortieren)

$$t_{\boldsymbol{x},i}^1 = \int_{\Omega^+} \int_{\boldsymbol{y}\in\boldsymbol{Y}} T_{\boldsymbol{x}}(.)\, k_d(\boldsymbol{y})\, T_{\boldsymbol{y}}(\boldsymbol{\omega_j})\, d\boldsymbol{y} \cdot y_i(\boldsymbol{\omega_j})\, d\boldsymbol{\omega_j}$$

$$t_{\boldsymbol{x},i}^1 = \int_{\boldsymbol{y}\in\boldsymbol{Y}} T_{\boldsymbol{x}}(.)\, k_d(\boldsymbol{y}) \underbrace{\int_{\Omega^+} T_{\boldsymbol{y}}(\boldsymbol{\omega_j}) \cdot y_i(\boldsymbol{\omega_j})\, d\boldsymbol{\omega_j}}_{t_{\boldsymbol{y},i}^0}\, d\boldsymbol{y}$$

- wir erkennen: $t_{\boldsymbol{x},i}^1$ hängt ab von
  - Transferkoeffizient $t_{\boldsymbol{y},i}^0$
  - Transfer $T_{\boldsymbol{x}}(.)$ zwischen $\boldsymbol{x}$ und $\boldsymbol{y}$
  - Albedo bei $\boldsymbol{y}$

PRT: Transfer mit Interreflexion

BONUS

Herleitung: einfach indirektes Licht

- direkte Beleuchtung

$$L_r^0(\boldsymbol{x}) = k_d(\boldsymbol{x}) \int_{\Omega^+} \frac{1}{\pi} \langle \boldsymbol{n_x}, \boldsymbol{\omega_i} \rangle V(\boldsymbol{x}, \boldsymbol{\omega_i}) L_i(\boldsymbol{\omega_i}) d\boldsymbol{\omega_i} \approx \rho(\boldsymbol{x}) \sum_{i=1}^{n^2} t_{x,i}^0 l_i$$

- einfach indirekte Beleuchtung aus $L_r^0(\boldsymbol{x})$

$$L_r^1(\boldsymbol{x}) = k_d(\boldsymbol{x}) \int_{\Omega^+} \frac{1}{\pi} \langle \boldsymbol{n_x}, \boldsymbol{\omega_i} \rangle \big(1 - V(\boldsymbol{x}, \boldsymbol{\omega_i})\big) L_r^0(\boldsymbol{y}) d\boldsymbol{\omega_i}$$

$$L_r^1(\boldsymbol{x}) \approx k_d(\boldsymbol{x}) \int_{\Omega^+} \frac{1}{\pi} \langle \boldsymbol{n_x}, \boldsymbol{\omega_i} \rangle \big(1 - V(\boldsymbol{x}, \boldsymbol{\omega_i})\big) k_d(\boldsymbol{y}) \sum_{i=1}^{n^2} t_{y,i}^0 l_i \, d\boldsymbol{\omega_i}$$

- wie trägt das bei $\boldsymbol{y}$ reflektierte Licht, zur Gesamtreflexion bei $\boldsymbol{x}$ bei (bzw. zu den Transferkoeffizienten $t_{x,i}$)?

PRT: Transfer mit Interreflexion

- es lässt sich also zeigen: es trägt nur der Koeffizient $t^0_{\boldsymbol{y},i}$ zu $t^1_{x,i}$ bei (Herleitung siehe Bonusmaterial eben + weiteres)
- Berechnung der Koeffizientenvektoren (für alle i zusammen)

$$t^0_{\boldsymbol{x},i} = \frac{4\pi}{N}\sum_{j=1}^{N} T'_{\boldsymbol{x}}(\boldsymbol{\omega}) y_i(\boldsymbol{\omega}) \quad \text{mit} \quad T'_{\boldsymbol{x}}(\boldsymbol{\omega}) = \frac{1}{\pi}\langle \boldsymbol{n_x}, \boldsymbol{\omega}\rangle^+ V(\boldsymbol{x}, \boldsymbol{\omega})$$

$$t^b_{\boldsymbol{x},i} = \frac{4\pi}{N}\sum_{j=1}^{N} \frac{1}{\pi}\langle \boldsymbol{n_x}, \boldsymbol{\omega}\rangle^+ \big(1 - V(\boldsymbol{x}, \boldsymbol{\omega})\big) k_d(\boldsymbol{y}) t^{b-1}_{\boldsymbol{y},i} \quad \text{für } b > 0$$

$$t_{\boldsymbol{x},i} = \sum_{b=0}^{B} t^b_{\boldsymbol{x},i}$$

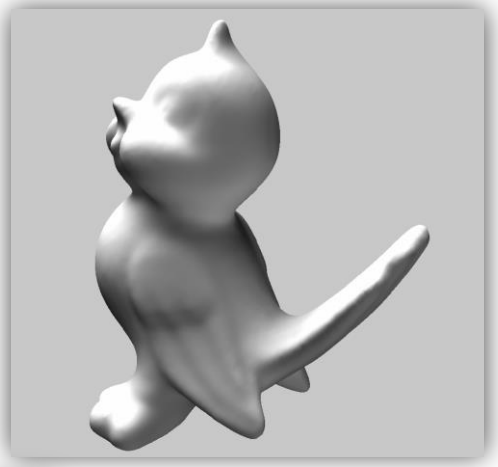
- im Gegensatz zu Transfer ohne Mehrfachreflexion geht jetzt die **Reflektivität** des Objekts über $k_d(\boldsymbol{y})$**in die Transferkoeffizienten** ein, d.h. auch die **Oberflächenreflexion/-farbe ist jetzt statisch**!
- gilt nur unter der Annahme, das die Beleuchtung für alle Oberflächenpunkte konstant ist (Umgebungslicht, also wie bisher)

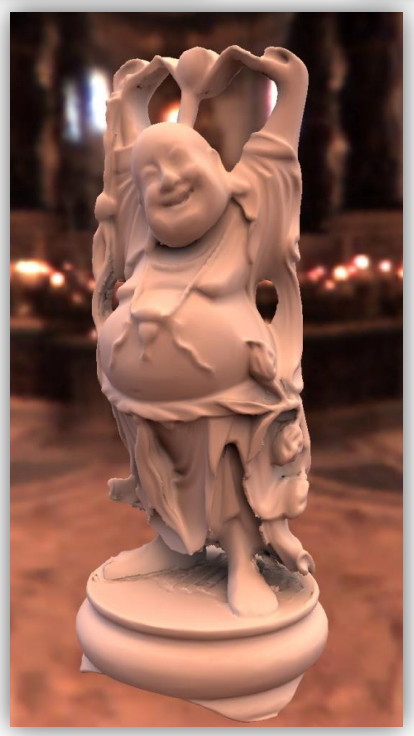
PRT Beispiel

diffuse unverschattet

diffus verschattet

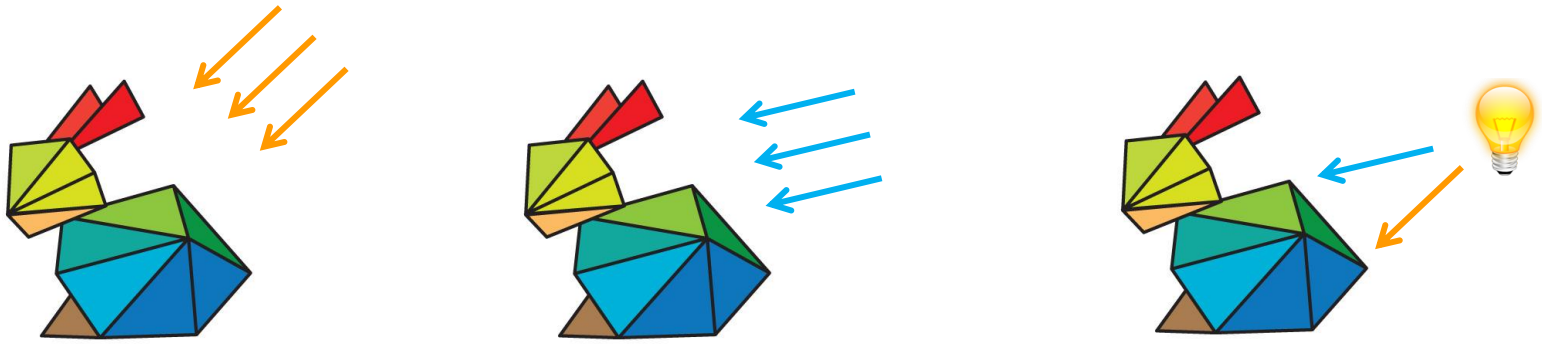
diffuse Interreflexion





PRT: Erweiterung auf lokale Lichtquellen

- bisher haben wir räumlich unveränderliche Beleuchtung betrachtet
  - bei lokalen Lichtquellen gilt diese Annahme natürlich nicht
- eine Möglichkeit: mehrere Koeffizientenvektoren für verschiedene Lichtrichtungen berechnen (also für direktionale Lichtquellen)
  - dann für eine geg. Lichtposition die Koeffizienten (entsprechend der Richtung zur Oberfläche) auswählen oder komponentenweise interpolieren



  - hier macht man sich eine wichtige „Eigenschaft“ der SH zunutze: man kann **Koeffizientenvektoren interpolieren** und es kommt i.d.R. etwas Plausibles/Sinnvolles heraus

Transluzente Materialen

- Beleuchtungsberechnung mit Subsurface Scattering durch BSSRDF S

$$L_r(\boldsymbol{x}, \boldsymbol{\omega_r}) = \int_A \int_\Omega S(\boldsymbol{x}, \boldsymbol{\omega_r}, \boldsymbol{x_i}, \boldsymbol{\omega_i}) L_i(\boldsymbol{x_i}, \boldsymbol{\omega_i}) \cos\theta_i \, d\boldsymbol{\omega_i} \, dA(\boldsymbol{x_i})$$

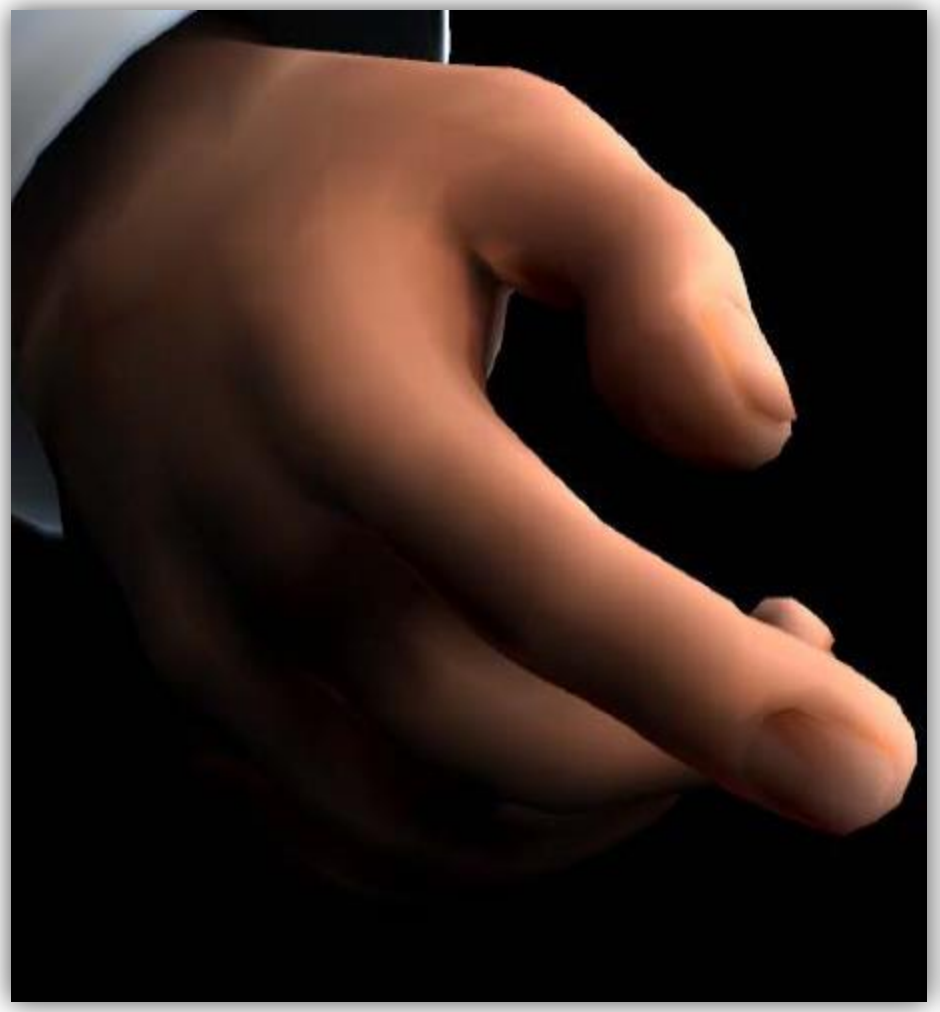
- mit den bekannten Vereinfachungen…
  - Beleuchtung durch entferntes Licht (keine Abhängigkeit von $\boldsymbol{x_i}$ bei L_i)
  - ausgehendes Licht ist diffus (keine Abhängigkeit von $\boldsymbol{\omega_r}$)
  - erhalten wir

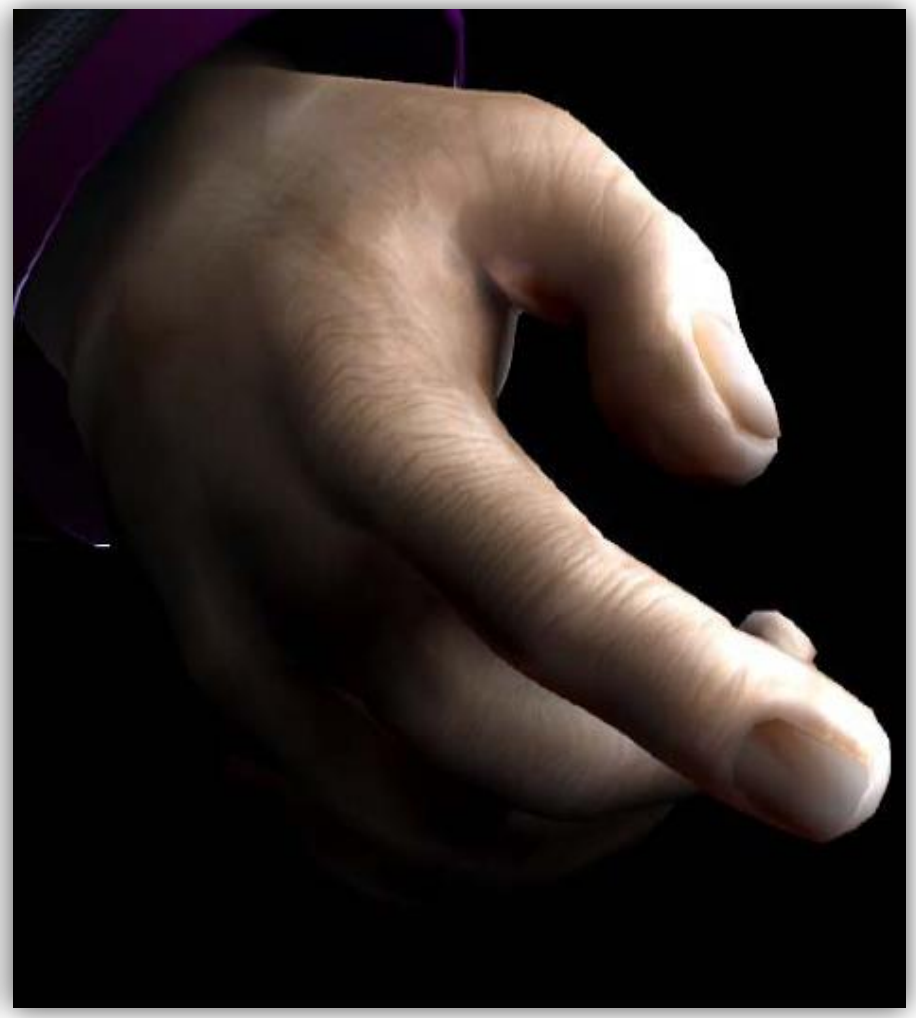
$$L_r(\boldsymbol{x}) = \int_A \int_\Omega S(\boldsymbol{x}, \boldsymbol{x_i}, \boldsymbol{\omega_i}) L_i(\boldsymbol{\omega_i}) \cos\theta_i \, d\boldsymbol{\omega_i} \, dA(\boldsymbol{x_i})$$

- …und können wieder eine Transferfunktion $T_{\boldsymbol{x}}(\boldsymbol{\omega_i}) = \int_A S(\boldsymbol{x}, \boldsymbol{x_i}, \boldsymbol{\omega_i}) \cos\theta_i \, dA(\boldsymbol{x_i})$ angeben
- berechnen würde man die Transferfunktion z.B. mit der (vereinfachten) Dipolapproximation (siehe Vorlesung Fotorealistische Bildsynthese)

BSSRDF = Bidirectional Surface Scattering Reflectance Distribution Function

Beispiel – SH Translucency (ATI)

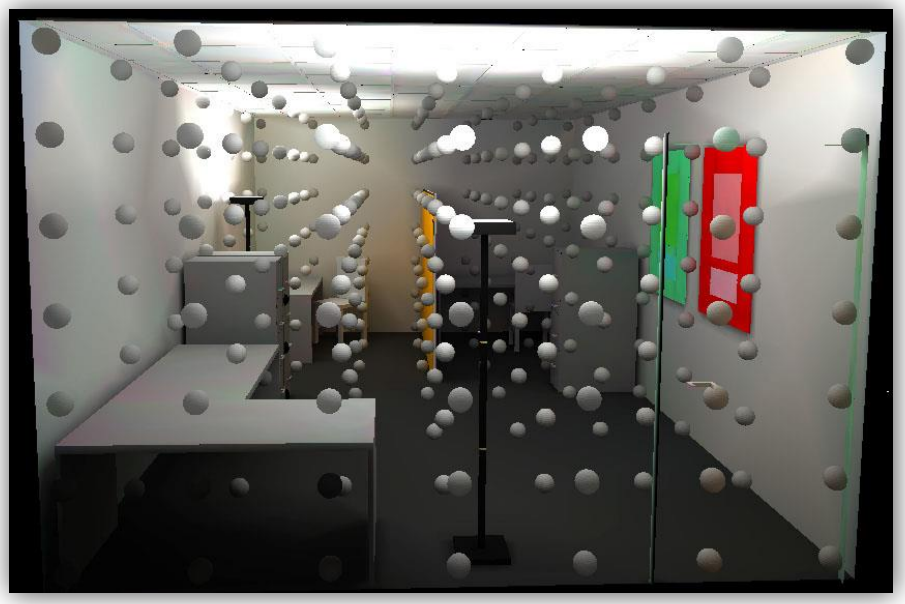


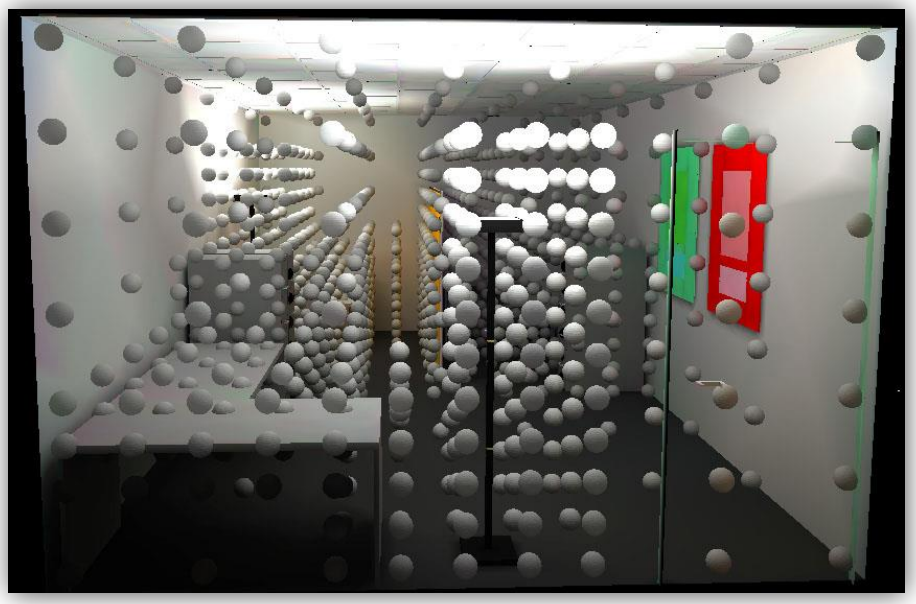


http://www.game-tech.com/Talks/SkinRendering.pdf

SH und Irradiance Volumes

- eine weitere wichtige Anwendung von SH sind Irradiance Volumes
  - für viele Punkte im Raum (meist auf einem regelmäßigen Gitter oder in einem Octree) wird das einfallende Licht (Irradiance) gespeichert
  - richtungsabh. Irradiance-Funktion $E(\boldsymbol{\omega}) = \int_{\Omega} L(\boldsymbol{\omega_i}) \langle \boldsymbol{\omega}, \boldsymbol{\omega_i} \rangle^{+} \, d\boldsymbol{\omega_i}$ an diesen Punkten wird durch Spherical Harmonics dargestellt
  - Vorteil: wenige Koeffizienten bei „glatten“ Funktionen, die Möglichkeit das Licht zu interpolieren und schnelles Nachschlagen

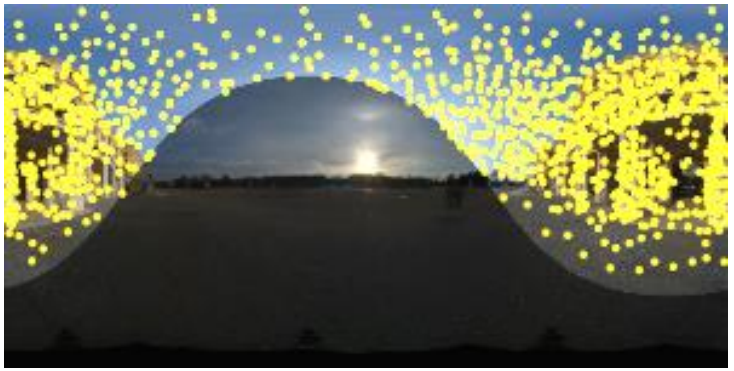




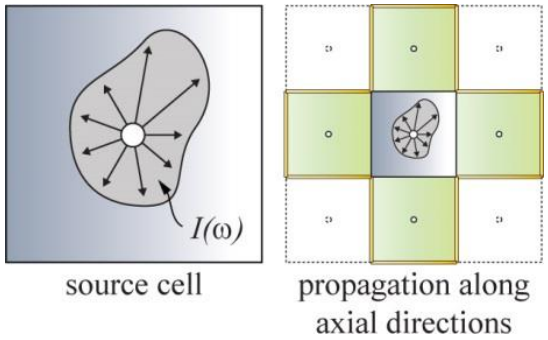
Bilder: The Irradiance Volume, Gene S. Greger, MSc Thesis

Vielfältige Anwendungen von SH

- Importance Sampling von Environment Maps



- Repräsentation von Radiance-Verteilungen (Light Propagation Volumes)



- Frequenzanalyse (siehe u.a. https://cseweb.ucsd.edu/~ravir/)
- u.v.m.

PRT: spiegelnder (glossy) Transfer

- bisher: nur diffuser Transfer
  - das reflektierte Licht einer diffusen Fläche ist unabhängig von der Betrachtungsrichtung
  - deshalb kann direkt eine Transferfunktion „$L_i(\boldsymbol{\omega}) \rightarrow L_r$“ angegeben werden
- bei spiegelnden Flächen muss am Ende eine Auswertung abh. von der Blickrichtung stehen

PRT: spiegelnder (glossy) Transfer

Grundidee: beliebige BRDF, direkte verschattete Beleuchtung

- Beleuchtung durch eine EnvMap $L(\boldsymbol{\omega_i})$
- Sichtbarkeitsfunktion $V(\boldsymbol{x}, \boldsymbol{\omega_i})$
- reflektiertes Licht am Oberflächenpunkt $\boldsymbol{x}$

$$L_r(\boldsymbol{x}, \boldsymbol{\omega_r}) = \int_\Omega \underbrace{f_r(\boldsymbol{\omega_i}, \boldsymbol{x}, \boldsymbol{\omega_r})}_{\text{BRDF}} \cdot \underbrace{V(\boldsymbol{x}, \boldsymbol{\omega_i}) \max(0, \cos\theta_i)\, L(\boldsymbol{\omega_i})}_{=: L_i(\boldsymbol{x}, \boldsymbol{\omega_i})} d\boldsymbol{\omega_i}$$

projiziere BRDF in Spherical Harmonics: $f_{\boldsymbol{x},\boldsymbol{\omega_r}}(\boldsymbol{\omega_i}) = f_r(\boldsymbol{\omega_i}, \boldsymbol{x}, \boldsymbol{\omega_r})$

- mit $f_i \approx \frac{4\pi}{N} \sum_{j=1}^{N} f_{\boldsymbol{x},\boldsymbol{\omega_r}}(\boldsymbol{\omega_j}) y_i(\boldsymbol{\omega_j})$
- erlaubt schnelle Berechnung von $\int_\Omega f_{\boldsymbol{x},\boldsymbol{\omega_r}}(\boldsymbol{\omega_i}) \cdot L_i(\boldsymbol{x}, \boldsymbol{\omega_i}) d\boldsymbol{\omega_i}$

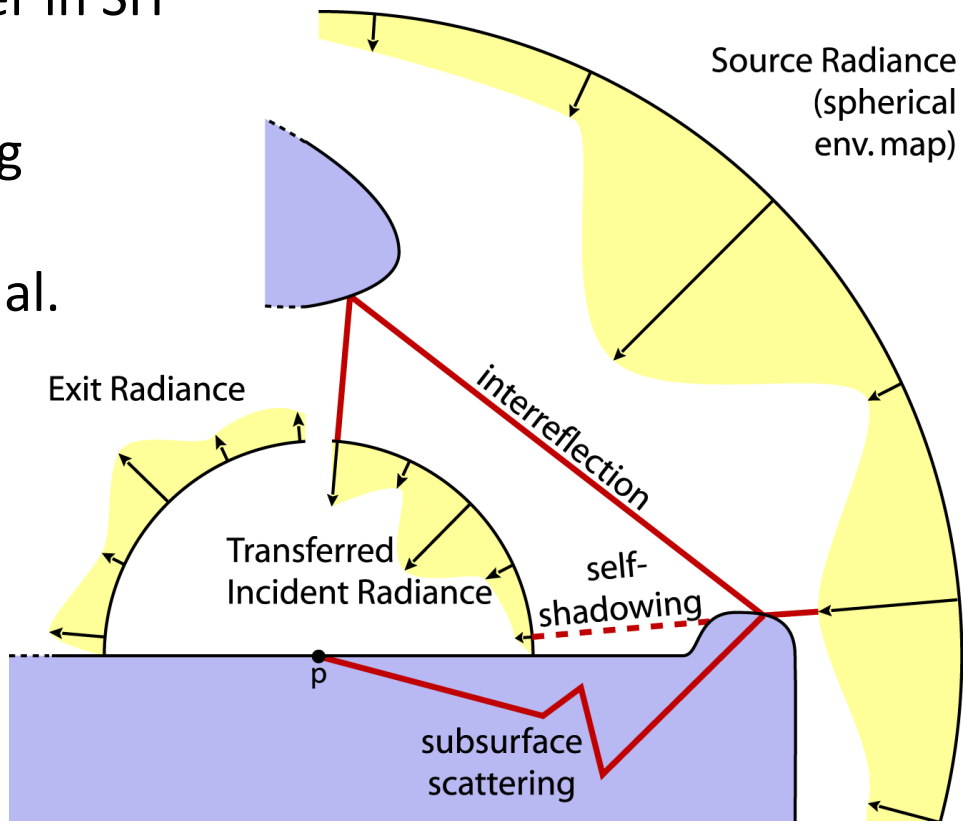
- aber woher bekommt man $L_i(\boldsymbol{x}, \boldsymbol{\omega_i})$? …mit Interreflexion?

PRT: spiegelnder (glossy) Transfer

- bisher: nur diffuser Transfer
  - das reflektierte Licht einer diffusen Fläche ist unabhängig von der Betrachtungsrichtung
  - deshalb kann direkt eine Transferfunktion „$L_i(\boldsymbol{\omega}) \rightarrow L_r$“ angegeben werden
- bei spiegelnden Flächen muss am Ende eine Auswertung abh. von der Blickrichtung stehen
- das erfordert einen aufwändigeren Ansatz:
  - wir müssen aus dem einfallenden Licht $L(\boldsymbol{\omega_i})$ das zu einem Punkt transferierte (= an einem Punkt ankommende) Licht berechnen, also $L_i(\boldsymbol{x}, \boldsymbol{\omega_i})$
  - und anschließend dort die BRDF auswerten

PRT: spiegelnder (glossy) Transfer

- transformiere einfallendes Licht in transferiertes Licht
  - einfallendes Licht wird nach wie vor in SH dargestellt, also mit $\{l_i\}$
  - eine **Koeffizientenmatrix** transformiert $\{l_i\}$ in ankommendes Licht an einem Oberflächenpunkt, „globale → lokal einfallende Radiance“
- Grundidee: finde Darstellung, die die Berechnung von $L_i(\boldsymbol{x}, \boldsymbol{\omega_i})$ aus $\{l_i\}$ erlaubt → projiziere diese wieder in SH
- Details: Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments, Sloan et al.



PRT: spiegelnder (glossy) Transfer

Beispiel (Transfer nur für direkte Beleuchtung und Verschattung)

- einfallende Radiance an einem Punkt $\boldsymbol{x}$ ist $L_{in}(\boldsymbol{x},\boldsymbol{\omega}) = L_i(\boldsymbol{\omega})V(\boldsymbol{x},\boldsymbol{\omega})$
- wir können die Beleuchtung L_i natürlich wieder mit SH darstellen

$$L_{in}(\boldsymbol{x},\boldsymbol{\omega}) \approx \left(\sum_{j=1}^{n^2} l_j y_j(\boldsymbol{\omega})\right) V(\boldsymbol{x},\boldsymbol{\omega}) = \sum_{j=1}^{n^2} l_j y_j(\boldsymbol{\omega}) V(\boldsymbol{x},\boldsymbol{\omega})$$

- die Transfermatrix $\mathrm{T} = \left[T_{j,i}^{\boldsymbol{x}}\right]$ erhalten wir durch erneutes Projizieren von $L_{in}(\boldsymbol{x},\boldsymbol{\omega})$ in die SH-Basis

$$L_{in}(\boldsymbol{x},\boldsymbol{\omega}) \approx \sum_{i=1}^{n^2} l_i^{\boldsymbol{x}} y_i(\boldsymbol{\omega}) \quad \text{mit} \quad l_i^{\boldsymbol{x}} = \int_{\Omega} L_{in}(\boldsymbol{x},\boldsymbol{\omega}) y_i(\boldsymbol{\omega}) d\boldsymbol{\omega}$$

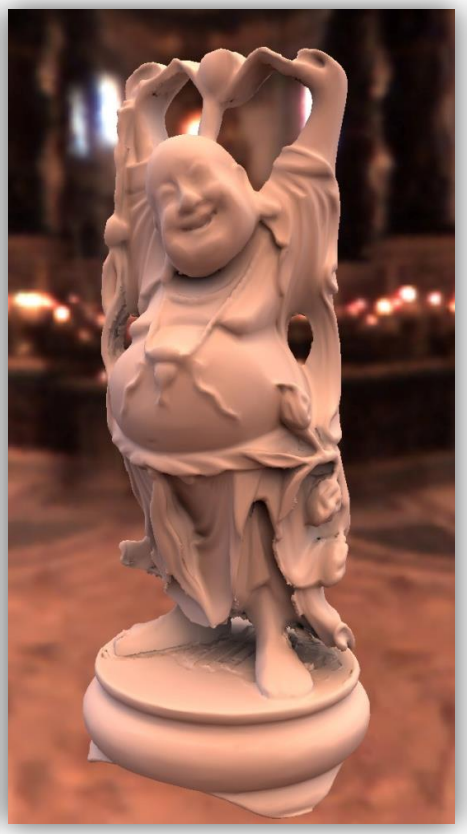
$$l_i^{\boldsymbol{x}} = \int_{\Omega} \sum_{j=1}^{n^2} l_j y_j(\boldsymbol{\omega}) V(\boldsymbol{x},\boldsymbol{\omega})\, y_i(\boldsymbol{\omega}) d\boldsymbol{\omega} = \sum_{j=1}^{n^2} l_j \underbrace{\int_{\Omega} y_j(\boldsymbol{\omega}) V(\boldsymbol{x},\boldsymbol{\omega}) y_i(\boldsymbol{\omega}) d\boldsymbol{\omega}}_{T_{j,i}^{\boldsymbol{x}}}$$

- $T_{j,i}^{\boldsymbol{x}}$: „wir trägt der j-te SH Koeffizient der Beleuchtung zum i-ten Koeffizient der lokal-einfallenden Radiance bei“

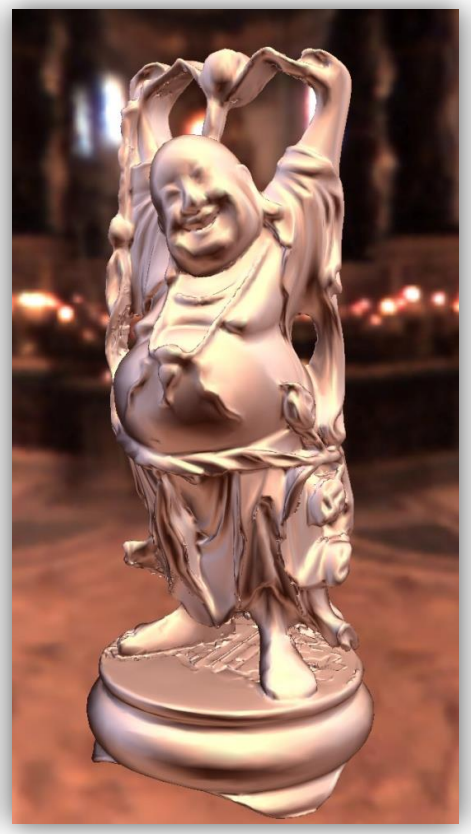
PRT: Beispiele

- dadurch sind moderat glänzende BRDFs möglich
- aufgrund der erhöhten Koeffizientenanzahl wenig(er) praktikabel (Transfermatrix pro Vertex; mehr Bänder als im diffusen Fall benötigt: n Bänder $\rightarrow n^2$ Basisfunktionen $\rightarrow n^2 \times n^2$ Transfermatrix)

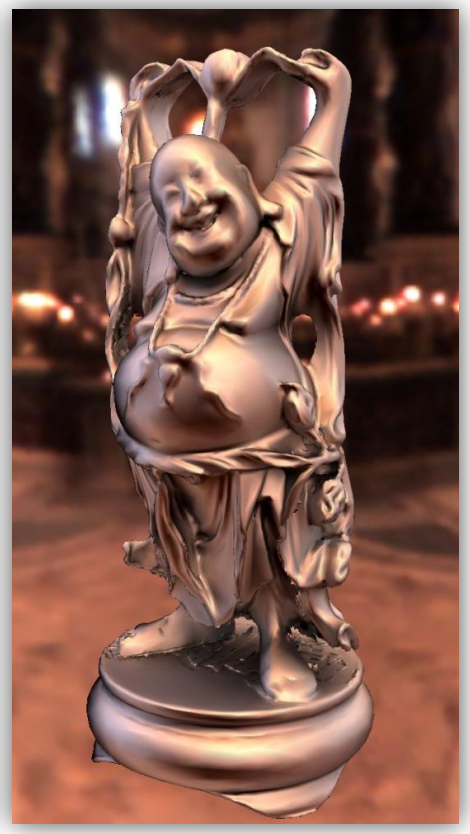
diffus



diffus, verschattet mit Interreflexion



glossy



glossy, verschattet mit Interreflexion

PRT: diffus vs. glossy, 0/1/2 bounces

- Reduktion der Koeffizienten durch Hauptkomponentenanalyse und Clustering (Clustered Principal Components for Precomputed Radiance Transfer, Sloan et al.)

Rotationen von Spherical Harmonics

Rotation der Beleuchtung – wie ändern sich die Koeffizienten l_i

- Rotation einer in SH projizierten Funktion kann direkt auf den Koeffizienten berechnet werden
  - Prinzip: erneute Projektion der rotierten y_i
  - die Rotationsmatrix $\mathrm{M} = [M_{j,i}]$ beschreibt, wir die i-te rotierte Basisfunktion die j-te Basisfunktion beeinflusst:

$$M_{j,i} = \int_\Omega y_i\big(R(\boldsymbol{\omega})\big) y_j(\boldsymbol{\omega}) d\boldsymbol{\omega}$$

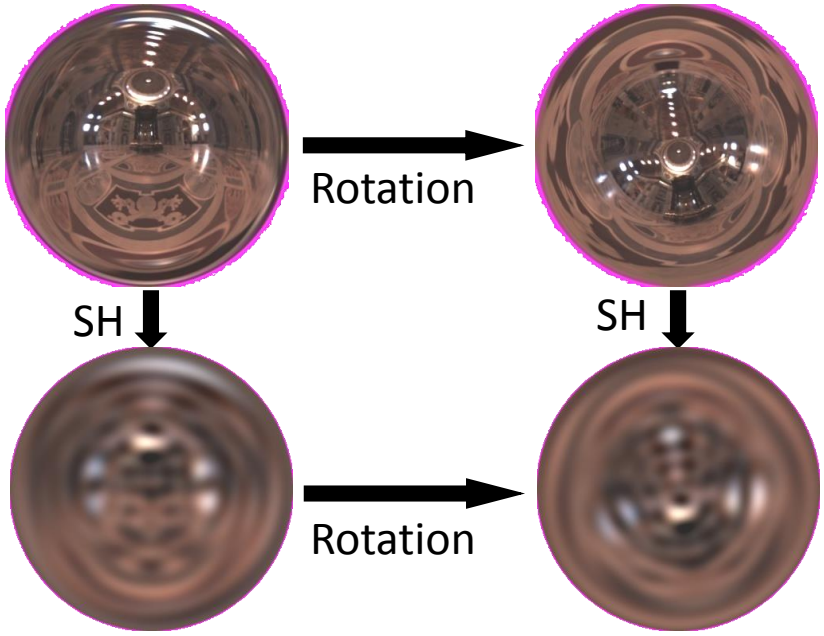
  - Rotation $l_{i,R} = \mathrm{M} \cdot l_i$
  - n Bänder $\rightarrow n^2 \times n^2$ Matrix
  - Matrix ist aber blockdiagonal: Koeffizienten verschiedener Bänder interagieren nicht

$$R_{SH} = \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & \mathbf{X} & \mathbf{X} & \mathbf{X} & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & \mathbf{X} & \mathbf{X} & \mathbf{X} & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & \mathbf{X} & \mathbf{X} & \mathbf{X} & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & \mathbf{X} & \mathbf{X} & \mathbf{X} & \mathbf{X} & \mathbf{X} & \cdots \\ 0 & 0 & 0 & 0 & \mathbf{X} & \mathbf{X} & \mathbf{X} & \mathbf{X} & \mathbf{X} & \cdots \\ 0 & 0 & 0 & 0 & \mathbf{X} & \mathbf{X} & \mathbf{X} & \mathbf{X} & \mathbf{X} & \cdots \\ 0 & 0 & 0 & 0 & \mathbf{X} & \mathbf{X} & \mathbf{X} & \mathbf{X} & \mathbf{X} & \cdots \\ 0 & 0 & 0 & 0 & \mathbf{X} & \mathbf{X} & \mathbf{X} & \mathbf{X} & \mathbf{X} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Eigenschaften von Spherical Harmonics

Rotationsinvarianz

- SH Rotationen sind für allgemeine Rotationen meist zu aufwändig
- Alternative: Euler-Rotation, z.B. um die Achsen ZYZ
  - die Rotation um Y wird noch durch $X(90°)ZX(-90°)$ ersetzt → damit ist nur eine Rotationsmatrix um Z notwendig
  - diese Rotationsmatrizen sind dünn besetzt (und teils sind die Basisfunktionen rotationssymmetrisch)
- wenn sinnvoll macht man sich zu nutze, dass SH rotationsinvariant sind
  - SH-Rotation und Rotation der Eingabefunktion bei der Projektion liefern dasselbe Resultat
  - $l_{i,R} = \mathrm{M} l_i = \int_\Omega L_{in}\big(R(\boldsymbol{\omega})\big) y_i(\boldsymbol{\omega}) d\boldsymbol{\omega}$



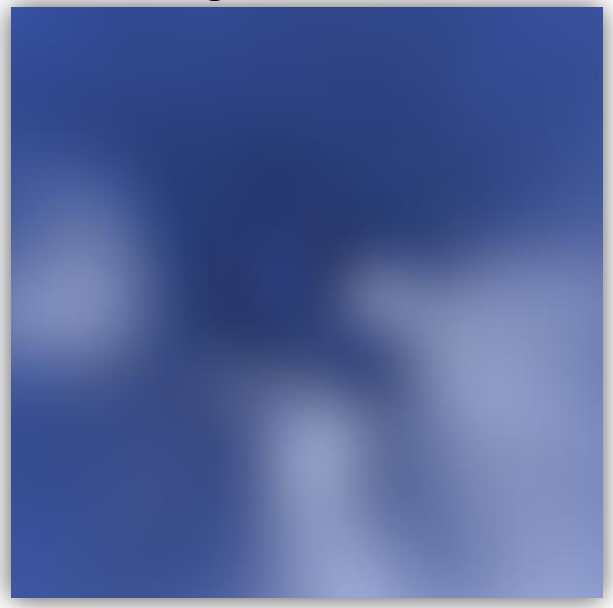
Eigenschaften von Spherical Harmonics

- PRT und SH werden i.d.R. eingesetzt für „low-frequency lighting“, d.h. niederfrequente Beleuchtung und diffuse/moderat spiegelnde BRDFs
  - meist separiert man hoch- und niederfrequente Beleuchtung, z.B. Sonnenlicht als direktionale Lichtquelle, Himmelslicht mittels SH
  - Rekonstruktion hoher Frequenzen benötigt viele SH Bänder, dementsprechend mehr Speicher
  - weiteres Problem: Überschwinger oder „Ringing“

original

ungefiltert: Ringing

gefiltert

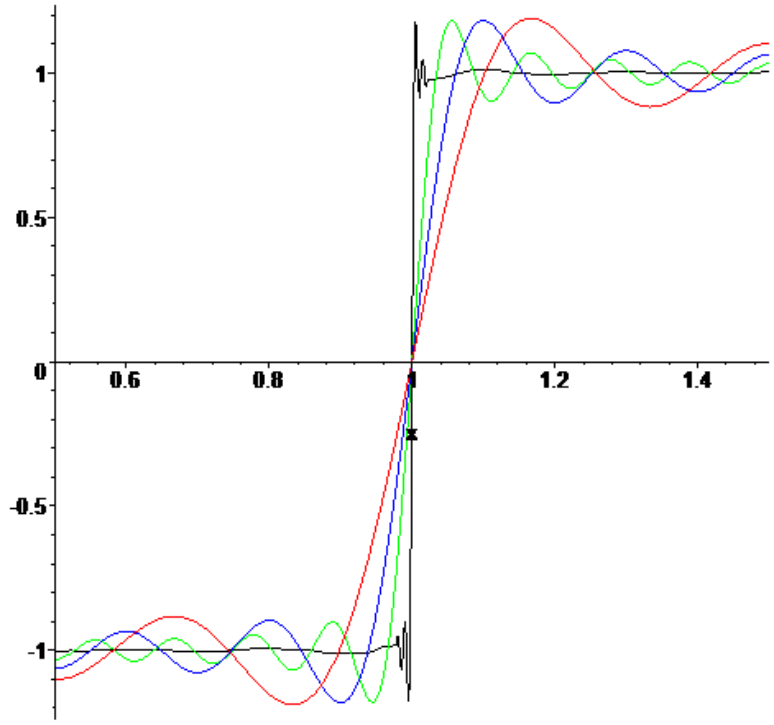


Eigenschaften von Spherical Harmonics

Überschwinger

- mit glatten Basisfunktionen können Diskontinuitäten nicht dargestellt werden (vgl. Darstellung einer Rechteckfunktion durch Sinusfunktionen)
- Diskontinuitäten in diskreten EnvMaps und durch Sichtbarkeitsfunktion
- Überschwinger treten auch bei nicht unendlichen-hohen Frequenzen auf

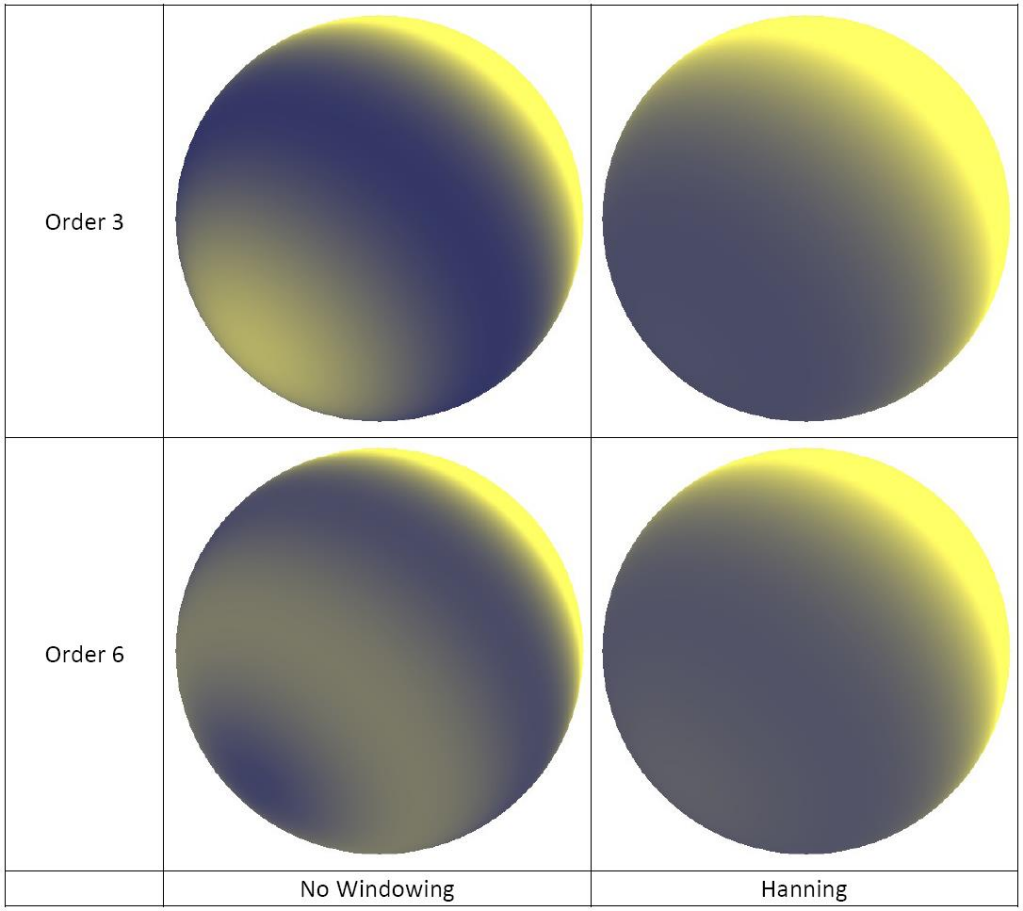
- Beispiel: Rechteck-Funktion
  - Oszillationen nahe der Diskontinuität, egal wie viele Terme man verwendet!
  - Stärke des Überschwingens ist konstant (hier ca. 18% bei der Fourierreihe)



Eigenschaften von Spherical Harmonics

Artefakte durch Überschwinger

- Beispiel: gelbes direktionales Licht und weißes Umgebungslicht
  → Überschwinger verursachen blaue Farbe
- Lösung: Unterdrückung durch Reduktion der Beiträge höherer Bänder



Eigenschaften von Spherical Harmonics

Gibbsches Phänomen

- bisher: SH-Projektion von Funktionen mit Diskontinuitäten – Überschwinger, deutlicher mit zunehmender Anzahl von Bändern
- aber selbst bei stetigen Funktionen treten **Probleme durch numerische Integration** bei der Projektion auf

- Beispiel: konstante Funktion

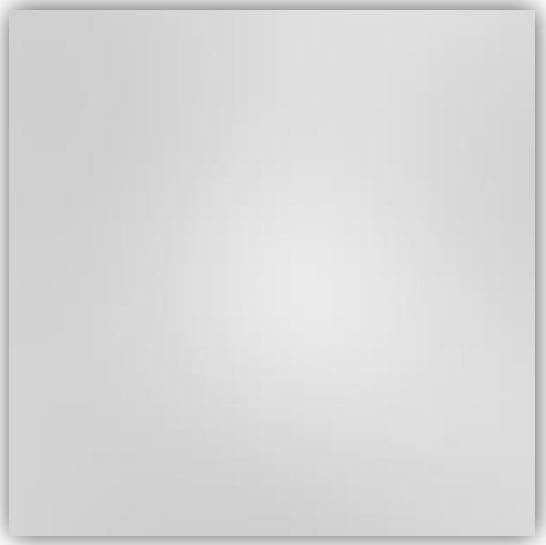
$$f(\boldsymbol{\omega}) = 1 \quad \Rightarrow \int_{\Omega} y_i(\boldsymbol{\omega}) d\boldsymbol{\omega} = \begin{cases} 2\sqrt{\pi} & \text{falls } i = 1 \\ 0 & \text{sonst} \end{cases}$$

  - bei der Projektion verschwinden alle Koeffizienten bis auf f_1, d.h. $f(\boldsymbol{\omega}) = 1$ wird schon ab dem ersten Band ($n = 0$) perfekt rekonstruiert
  - …leider nur bei symbolischer Bestimmung der Integrale…

Gibbsches Phänomen

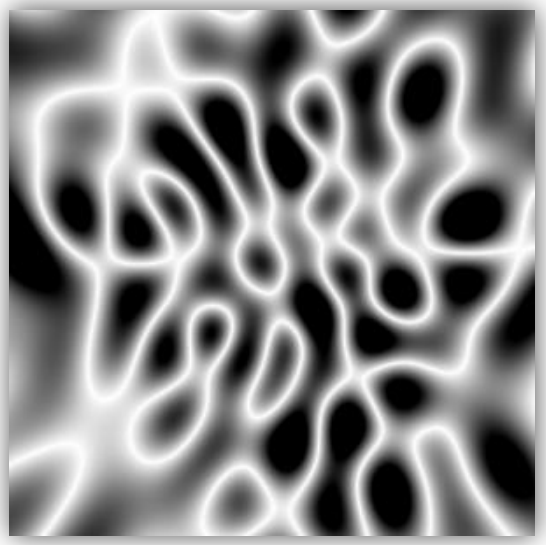
- bei der MC-Integration sind die Koeffizienten f_i $(i > 1)$ i.d.R. klein und der Fehler wird mit zunehmender Anzahl Abtastpunkte kleiner

- trotzdem: kritisch, wenn viele SH Bänder verwendet werden
  - SH-Funktionen hoher Bänder sind hochfrequent und Fehler führen schnell zu falschen (betragsmäßig zu großen) Koeffizienten



5 Bänder

10.000 Abtastpunkte



26 Bänder

10.000 Abtastpunkte

26 Bänder

1.000.000 Abtastpunkte

- **Lösung: erniedrige Koeffizienten der Basisfunktionen höherer Bänder (= Windowing)**

Ausblick Precomputed Radiance Transfer

"prime fathers"

[Ramamoorthi02]

[Ramamoorthi01]

[Sloan02]

General BRDFs

[Kautz02]

[Lehtinen03]

[Sloan03]

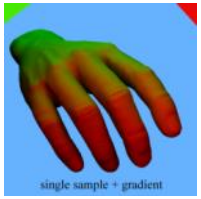
Texture Detail

[Sloan03]

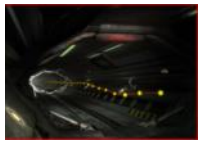
[Sloan05]

[Sloan06]

Non-distant lighting,
Shadowing, scattering,…

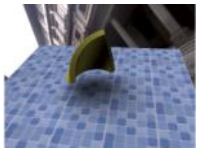


[Annen04]



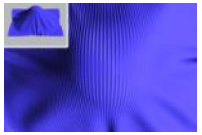
[Oat05]

[Green07]



[Wang07]

[Sun05]

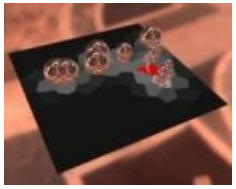


[Han07]

[Habel08]

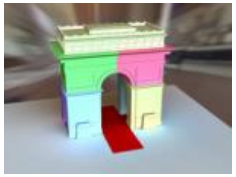
Dynamic Scenes

[Zhou05]



[Ren06]

[Sloan07]

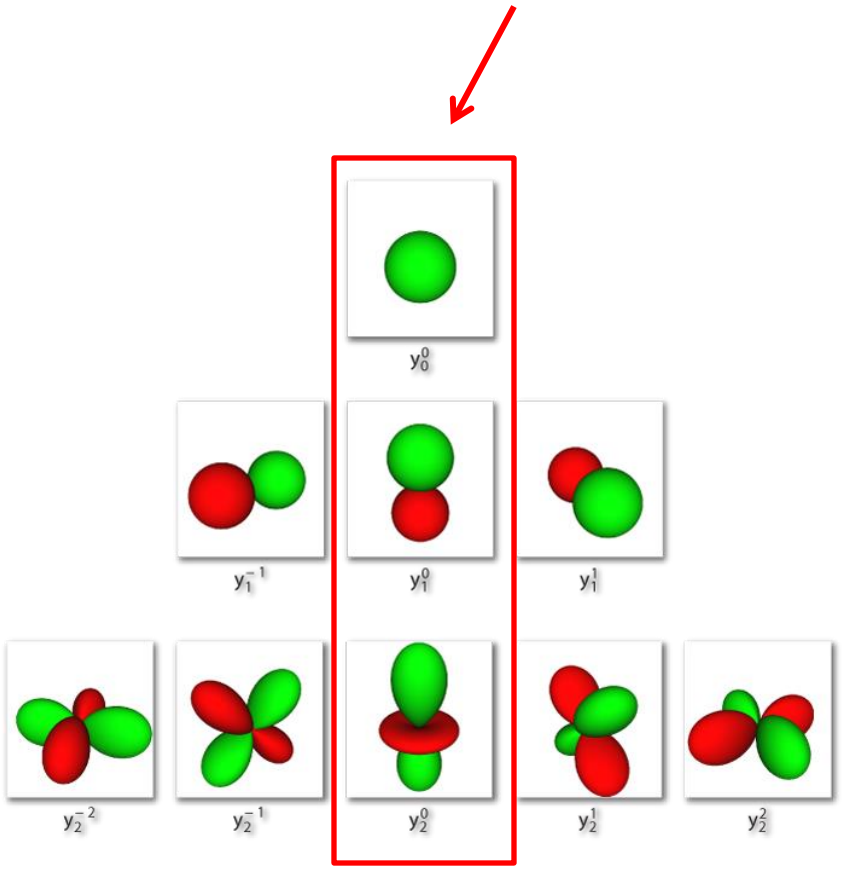


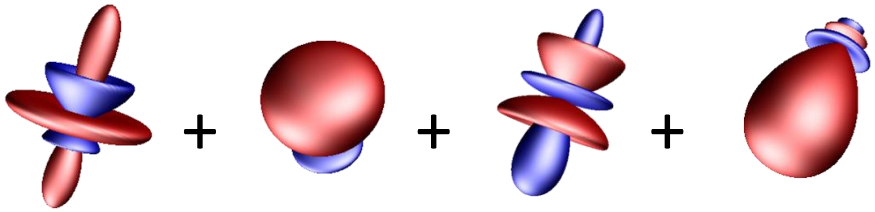
[Pan07]

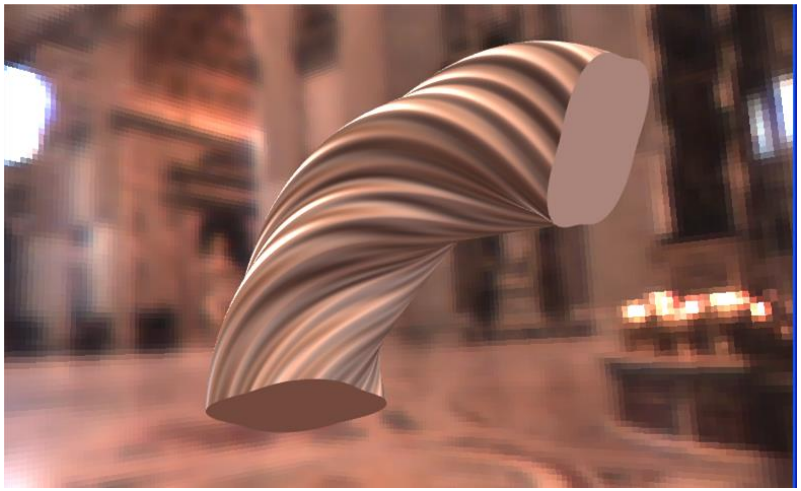
Ausblick Precomputed Radiance Transfer

- Local Deformable PRT
  - lokale, niederfrequente PRT Effekte, z.B. Verdeckung in einer Displacement Map
  - approximiere Transfervektor mit Summe von Zonal Harmonics (einfach rotierbar)

$$R_{SH} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & X & X & X & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & X & X & X & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & X & X & X & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & X & X & X & X & X & \cdots \\ 0 & 0 & 0 & 0 & X & X & X & X & X & \cdots \\ 0 & 0 & 0 & 0 & X & X & X & X & X & \cdots \\ 0 & 0 & 0 & 0 & X & X & X & X & X & \cdots \\ 0 & 0 & 0 & 0 & X & X & X & X & X & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$





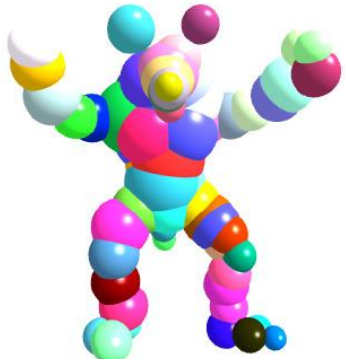


http://research.microsoft.com/~johnsny/

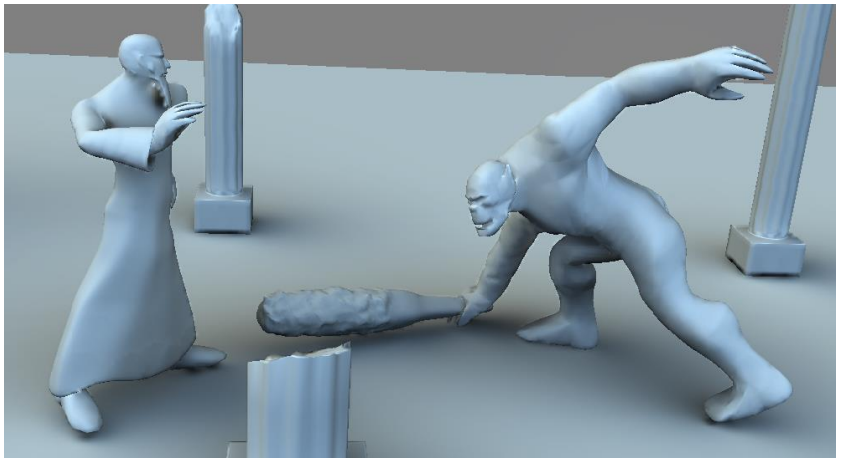
Ausblick Precomputed Radiance Transfer

SH in dynamischen Szenen

- „weiche“, globale Beleuchtung
  - niederfrequente Umgebungsbeleuchtung
  - sphärische Proxies statt tatsächlicher Geometrie



- erlaubt weiche Schatten mit sog. SH Exponentation [Ren06]
- Schatten und indirekte Beleuchtung im Bildraum [Sloan07]



Ausblick Precomputed Radiance Transfer

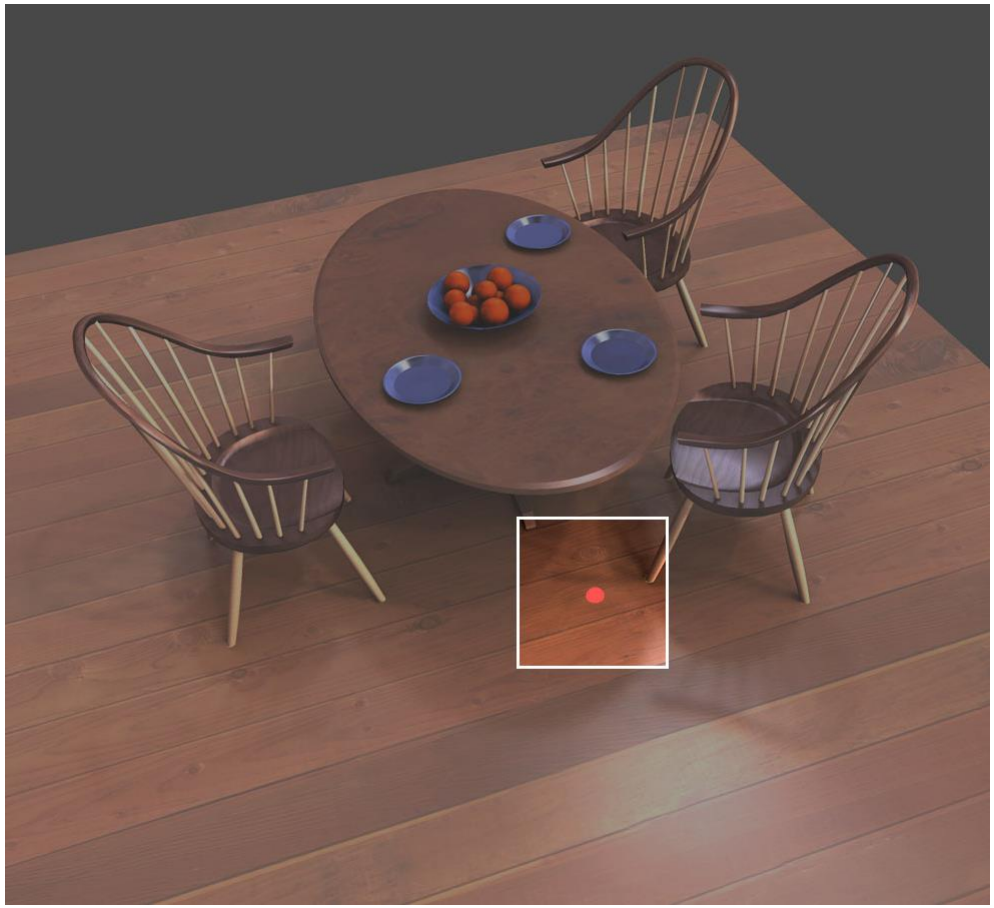
Triple Product Integrals

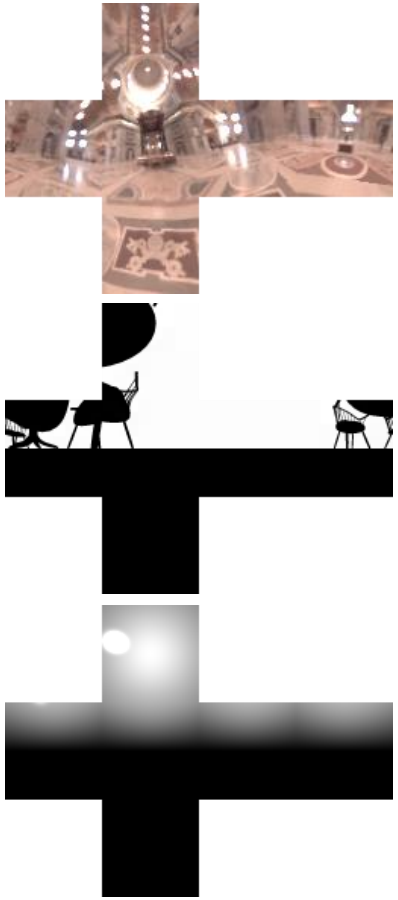
- Ziel: hochfrequente BRDFs und Beleuchtung
- 6D Problem: Einfalls- und Ausfallsrichtung, sowie Oberflächenposition

Ausblick Precomputed Radiance Transfer

Triple Product Integrals

- Vorberechnung: Sichtbarkeitsfunktion $V(\boldsymbol{x}, \boldsymbol{\omega_i})$ pro Vertex
- BRDF pro Oberflächenpunkt und ausgehende Richtung $\boldsymbol{\omega_r}$
- bis zu einigen Hundert Megabyte Daten





Ausblick Precomputed Radiance Transfer

Triple Product Integrals

- … aus BRDFs, Beleuchtung durch eine EnvMap $L(\boldsymbol{\omega_i})$ und Sichtbarkeitsfunktion $V(\boldsymbol{x}, \boldsymbol{\omega_i})$
- Berechnung des reflektierten Lichts am Oberflächenpunkt $\boldsymbol{x}$

$$L_r(\boldsymbol{x}, \boldsymbol{\omega_r}) = \int_\Omega \underbrace{f_r(\boldsymbol{\omega_i}, \boldsymbol{x}, \boldsymbol{\omega_r}) \max(0, \cos\theta_i)}_{\text{BRDF inkl. Kosinusterm}} \cdot \underbrace{V(\boldsymbol{x}, \boldsymbol{\omega_i})}_{\text{Sichtbarkeit}} \cdot \underbrace{L(\boldsymbol{\omega_i})}_{\text{Beleuchtung}} d\boldsymbol{\omega_i}$$

- in welcher Basis stellt man die Funktionen dar?
  - die Berechnung reduziert sich nicht einfach auf ein Skalarprodukt
  - Vergleich Basisfkt.: http://graphics.stanford.edu/papers/allfreqmat/
  - Beobachtung: bei geeigneter Basis sind teure Triple Products nur dort notwendig, wo sich die Sichtbarkeitsfunktion ändert (siehe http://www.cs.huji.ac.il/labs/cglab/projects/lap/)

SH Literatur

- Spherical Harmonic Lighting: The Gritty Details
  http://www.research.scea.com/gdc2003/spherical-harmonic-lighting.pdf
- Stupid Spherical Harmonics Tricks
  www.ppsloan.org/publications/
- einige „moderne“ Varianten
  http://research.microsoft.com/en-us/um/people/johnsny/