

## • Wie lässt sich das Problem der photorealistischen Bildsynthese beschreiben? Welche Eigenschaften hat diese Gl.??

Durch die Renderinggl.:  $L = L^e + \underbrace{L_r}_T \rightarrow$  unendlich. dim. Integral

- integriert über die gesamte Hemisphäre
- Term, der für d. direkten Einfluss verantwortlich ist, ist außerhalb d. Integrals
- beschreibt das Problem d. Lichttransports vom Blickpunkt d. Betrachters
- individuelle Glanzwerte sind berechnet für jeden Strahl
- rekursiv, nicht analytisch lösbar
- $L^e$  gibt an, wie viel Licht von  $x'$  nach  $x$  abgestrahlt wird, Emissionsterm
- $L_r$  ist d. Streuterm: jener Anteil d. Lichts, das  $x'$  von  $x'$  aus erreicht, in Richtung  $x$  reflektiert wird

## • Welche grundsätzlichen Lösungsstrategien gibt es bei der RG?

⇒ Inversion: → nicht häufig genutzt in Praxis

- gruppiert unbekannte Terme auf derselben Seite d. Gleichung
- numerisch instabil
- $T$  ist unendlich und ist nicht invertierbar in geschlossener Form
- Lösung durch Approximierung

⇒ Expansion: → rekursive Auswertung v. Teilintegralen entlang eines Pfades d. vom Auge od. v. der Lichtquelle ausgehen kann

→ rekursive Substitution  $n$ -Mal wiederholt, entspricht Rekursionsleveln während d. ray-casting Prozesses

$$\rightarrow x = 0,1x + 1,8 \quad x_1 = 0,1 \cdot (0,1x + 1,8) + 1,8 \quad \dots \text{ usw.}$$

→ da  $T$  eine Kontraktion ist:  $\lim_{n \rightarrow \infty} T^{n+1} L = 0$

→ daher ist  $L = \sum_{i=0}^{\infty} T^i L^e$  eine Lösung

→ für jedes Rekursionslevel muss über die gesamte Hemisphäre integriert werden

→ hoch-dimensionales Integral muss gelöst werden

→ Pfade müssen unabhängig sein, keine temporäre Darstellung nötig, Berechnung paralleler Wege

⇒ Iteration: → rechnen mit einem State, der die Beleuchtung der Szene kodiert

→ inhärente Diskretisierung bei allen Radiosity-Methoden

$$\rightarrow \text{z.B. } x_n = 0,1x_{n-1} + 1,8$$

→ erfordert überlappungsfreie Überdeckung und endliche Elementrepräsentation

→ Lösung auch berechnet für nicht sichtbare Teile d. Szene

→ nicht exakt bei hochfrequenten Schatten, Spiegelungen, etc.

→ 4/10 Härte kann gut ausgenutzt werden

→ tolle errechneten Funktionen sind Betrachterunabhängig

→ bessere Ergebnisse als bei Expansion

## • Gibt es Verfahren bei d. Bildsynthese, die nicht als Lösungsstrategie d. Gleichung verstanden werden können?

Nein, auch OpenGL gehört dazu. ^^

## • Was ist Iteration? Welches Verfahren verwendet diese Methode in d. Praxis?

→ rechnen mit einem State, der die Beleuchtung der Szene kodiert

→ inhärente Diskretisierung bei allen Radiosity-Methoden

$$\rightarrow \text{z.B. } x_n = 0,1x_{n-1} + 1,8 \quad L_{n+1} = L^e + T L_n$$

→ erfordert überlappungsfreie Überdeckung und endliche Elementrepräsentation

→ Lösung auch berechnet für nicht sichtbare Teile d. Szene

→ nicht exakt bei hochfrequenten Schatten, Spiegelungen, etc.

→ 4/10 Härte kann gut ausgenutzt werden

→ tolle errechneten Funktionen sind Betrachterunabhängig

→ bessere Ergebnisse als bei Expansion

## • Wenn das Markenzeichen d. Iteration d. Diskretisierung ist, warum erzeugt dann Photon Tracing (ein Expansionsverfahren) Lightmaps, die auf d. 1. Blick genauso aussehen wie Radiosity?

Unterschied: Iteration muss mit State iterativ rechnen → Datenstruktur wird eingeschränkt, Photon Tracing zeichnet nur einmal Informationen über den Endzustand auf

## • Welche Expansionsmethoden gibt es?

## • Welche Expansionsmethoden gibt es?

- ⇒ Raytracing: → wertet d. rekursive Integral nur teilweise aus
  - unvollständige Lösung
  - sendet Strahlen aus zur Überprüfung auf Sichtbarkeit
  - Schnittpunkt d. Strahls mit Objekten berechnet
  - Pixel d. Objekts mit niedrigster Distanz zum Ausgangspunkt wird hypernom
  - Shading durch Berechnen d. Normalen am Schnittpunkt
- ⇒ Path Tracing: → sendet Strahl, d. solange Reflektiert wird, bis er die Lichtquellen trifft od. ein Grenzwert überschritten wird, mit. od. ohne Streuung
  - schnell, einfach zu coden
  - funktioniert nicht für kleine Lichtquellen → Variance → Bildrauschen
  - wird d. multiple Importance Sampling reduziert
- ⇒ Bidirektionales Path Tracing: → sowohl vom Ausgangspunkt als auch v. d. Lichtquelle werden Strahlen gesendet und anschließend kombiniert
  - bessere Leistung als PT, auch besser bei Kaustiken
- ⇒ Metropolis: → Erweiterung d. BPT, schwierig zu implementieren
  - BPT bis ein mittlerer Pfad gefunden wurde
  - Andern d. Wege um bessere zu erhalten (Vertizen hinzugefügt od. gelöscht)
  - meist komplexe Situationen gut
  - Bias: für normale Szenen suboptimal
- ⇒ Photon Tracing: → arbeitet mit Photon Maps / Light Maps
  - sehr effizient für Kaustiken
  - Photonen gespeichert in KD-Bäumen
  - langsam, frisst Speicher, Beleuchtungsrekonstruktion abhängig v. Distanz / Bias
  - Photonen können gestreut, absorbiert, reflektiert od. gebrochen werden
  - Photonen werden v. Lichtquelle gesendet
  - Photonen haben Größe & Leistung → Beleuchtungsberechnung

## • Was für Datenstrukturen kann man zur Aufzeichnung der Photonen beim Photon Tracing verwenden?

- ⇒ Light Maps: → Lichttexturen mit fixer Auflösung
  - Vorteile: fixer, vergleichsweise niedriger Speicherbedarf; einfache Rekonstruktion, schnelle Auswertung; einfache Implementierung
  - Nachteile: begrenzte Auflösung; hohe Setup-zeiten (Flächeninhalt aller Pixel muss bekannt sein); Interpolation nötig
  - jedes Texturelement beinhaltet d. Durchschnitt d. Energie von allen Photonenelementen
- ⇒ Photon Maps: → speichern einzelne Photonen
  - Vorteile: hohe Genauigkeit, hohe Auflösung v. A. bei Kaustiken; kaum Setupzeiten
  - Nachteil: extrem hoher Speicheraufwand; schwierig, fehleranfällige und langsame Rekonstruktion, komplexe Implementierung
  - immer genutzt in 2- od. multipass-Renderern
  - 3 Arten: Caustic, Global, Shadow Photon Map

## • Was für Szenen kann klassisches Whitted Raytracing gut darstellen, welche nicht?

- gut: Spiegel, Transparenz, diffuse Oberflächen, Schatten, die d. direkte Beleuchtung entstehen
- schlecht: beliebige BRDFs (seidenmatte Oberflächen, indirekte Beleuchtung, Flächenlichtquellen (letzteres wird häufig d. einfache Erweiterungen angeboten))

## • Wie ist die Struktur d. Pixar Renderman in etwa aufgebaut?

Was sind die Stärken & Schwächen?

- ist ein batch-orientiertes Scanline Renderingssystem
- hoch flexibel, zuverlässig und robust
- flexibel, da an beliebiger Stelle im Arbeitsablauf eingegriffen werden kann
- ⇒ Struktur:

- einheitliche Benennung d. API Aufrufe
- alle Funktionen sind eingeklammert zw.  $R_{begin}$  &  $R_{end}$
- ein globaler Grafik-state ist enthalten in d. Klammer
- Jeder API Aufruf modifiziert diesen state
- alle API Aufrufe sind ständig vorangeht, die mit  $R_{NULL}$  terminieren
- die meisten Aufrufe arbeiten mit Oberflächeneigenschaften
- RM Interface unterstützt folgende Shader: light Source, Surface, Volume, Displacement, Image

→ robust, da PRM Algorithmus extrem primitiv ist: weder Schatten noch Reflexionen können berechnet werden, werden durch shadow & reflection maps ersetzt aber dafür kann man Schatten dort hinplotieren wo man will und ist nicht von der physikalischen Korrektheit abhängig

### • Wie arbeitet d. Radiance-Renderer in etwa? Was hat er für Stärken u. Schwächen?

- teilweise stochastischer Hybrid-Renderer, auf typische Architekturszenen optimiert
- Kaustiken nicht korrekt dargestellt
- nutzt große Anzahl an aufeinander abgestimmten Heuristiken
  - ↳ automatische u. benutzerdefinierte Erzeugung v. Hilfslichtquellen
  - ↳ Irradiance Caching
  - ↳ frühzeitige Eliminierung unwichtiger Lichtquellen
- Sammlungen v. mehreren simplen Tools, die d. Front-Ends gemanaged werden

### • Was ist der Grund für die Vorliebe d. Renderers für Monte Carlo Integration?

- hauptsächlich Expansionsverfahren erfordern rekursive Berechnung v. hochdim. Integralen → lässt sich mit MC effizient lösen
- unterlegen DNI (deterministischen numerischen Integrationsmethoden) bei niedrigen Dimensionen, aber je höher Dimensionen, desto ineffizienter DNI, während MC konstante Leistung bringt

### • Was ist Quasi Monte Carlo? "Löcher"

Effizienz v. Dimension abhängig

- weder reine Zufallszahlen noch eine gleichmäßige Abtastung sind sonderlich effizient
- daher will man Zufallszahlen, die die Ebene regelmäßig bedecken ohne regelmäßige Struktur → Pseudozufallszahlen
- Fehler hängt v. Anzahl d. Punkte und deren "Gleichmäßigkeit" (Discrepanz) ab
- Discrepanz = Abweichung aller Punkte v. idealer Gleichförmigkeit auf d. Intervall  $B: [0,1]^d$

### • Was ist importance sampling in Zusammenhang mit stochastischen Renderingprogrammen?

- Technik zum Beschleunigen d. Konvergenz der MC: Integral wird dort ausgewertet, wo  $\sigma$  numerisch groß ist
- Bsp: Beim Verfolgen einer rekursiven, stochastischen Pfader an einem gegebenen Punkt wird die neue Strahlrichtung bevorzugt in die Richtung gewählt, in die die PDF groß ist (am meisten Licht wird reflektiert)
- gleichmäßige Abtastung → viel langsamer

### • Was ist das Problem bei Multiple Importance Sampling?

Bei welchen Renderingmethoden tritt es auf?

- mehr als ein Schätzwert wird für ein Integral ausgerechnet, indem man mehrere Samples annimmt
- Problem: nur einer dieser Schätzwerte qualitativ gut
  - ↳ Auswahl dieses Wertes nur durch kleine Heuristiken möglich
- Bsp: Path Tracing, Bidirektionaler Path Tracing

### • Was ist Spectralrendering?

- Bildsynthese, bei der nicht mit Farbwerten sondern mit spektralen Darstellungen v. Licht und Reflektivität gerechnet wird
- Schritte: Spektrum vorbereiten, Aufbereitung d. spektralen Abtastungen durch Renderingberechnung, Berechnung d. Endbildes mittels CIE-Farbgleich-Funktionen und Standardtransformationen

### • Was sind die Vor- & Nachteile von Spectralrendering gegenüber der Bildsynthese im Farbraum?

- Vorteile: Vorhersage d. Aussahs möglich, Einbindungen v. aufwendigen

- Vorteile: Vorhersage d. Aussehens möglich, Einbindungen v. aufwendigen physikalischen Effekten (Dispersion, Polarisation, Fluoreszenz) möglich
- Nachteile: hoher Rechenaufwand, Vorhersagefähigkeit selten benötigt, Rohdaten (Spektral Daten für Oberflächen) schwer zu bekommen, Implementierung aufwendiger

### • Wie können Spektren intern dargestellt werden?

- regelmäßiger sampling (schnell, ungenau od. hoher Speicheraufwand)
- Basisfunktionen (gerau, geringer Speicheraufwand, langsam)
- Hybridansätze (komplex, besten Eigenschaften aus beiden Ansätzen)

### • Welche Schwierigkeit bringt es bei d. Implementierung wenn man Polarisation und/od. Fluoreszenz in einem Rendering-System einbaut?

- Die Datenstrukturen für Licht & Reflexionseigenschaften sind nicht mehr gleich, man muss zwischen Filter und Licht unterscheiden

### • Was sind HDR-Bilder? Warum sind sie für physikalisches Rendering interessant?

- ⇒ Bildformate, die mehr als 8bit pro Kanal speichern können
- ⇒ weil reale Szenen einen weit höheren Kontrastumfang haben als d. in 24 bit verfügbare

### • Welche HDR-Graphikformate gibt es?

- OpenEXR (16-bit Gleitkommazahlen pro Farbkanal, 32 bit Daten für Speicherung v. Tiefeninformation & Objektkennungen, verlustlose Komprimierung v. Bildern durch mitgelieferte Coders)
- logluv Tiff (nutzt CIE, 32 bit)

### • Warum gibt es praktisch keine spektralen Bildformate?

- hoher Speicheraufwand
- kaum Bedarf
- einzige Verwendung: interne Speicherformate v. spektralem Rendering

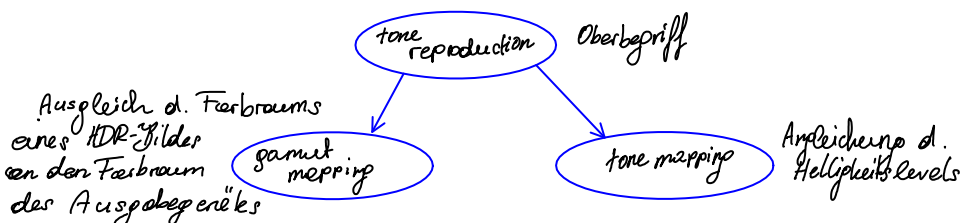
### • Was ist tone mapping?

- ⇒ Angleichung d. Helligkeitslevels in einem HDR-Bild auf die Möglichkeiten eines LDR-Ausgabegeräts

### • Was für Kategorien v. TM/TR-Verfahren gibt es?

- ⇒ 3 verschiedene Ansätze
  - Globale Methoden (skalieren d. Beleuchtungsstärke durch einen gep. Faktor, einfach und schnell)
    - ↳ räumlich uniform
    - ↳ linearer Skalierungsfaktor
    - ↳ nicht-linearer Skalierungsfaktor
  - lokale Methoden (Unterschiede zw. Bildbereichen mit einbeziehen)
    - ↳ räumlich nicht uniform
  - wahrnehmende Ansätze (Eigenschaften d. menschl. visuellen Systems werden mit einbezogen → realistische Darstellung)

### • Kurze Begriffs-erläuterung gamut mapping, tone mapping & tone reproduction?



### • Mit welchen Funktionen werden Oberflächen beschrieben?

- ⇒ BRDF: Funktionen für d. Reflexionsverhalten v. Oberflächen
- 2 Arten der Representation von BRDF-Werten
  - explizite Speicherung d. Messwerte
    - ↳ hoher Speicheraufwand
    - ↳ hoher Zeit- & Hardwareaufwand



- ↳ hoher Zeit- & Hardwareaufwand
- ↳ schlecht für Importance Sampling
- Approximierung durch analytische Funktionen

⇒ BTDF: ähnlich BRDF, aber für Strahlen, die zum Medium zeigen

⇒ BSDF: Kombination aus BRDF & BTDF

- ↳ selten genutzt
- ↳ über den ganzen Raum definiert

### • Was gibt es für Ansätze eine Oberfläche mit Werten zu beschreiben?

→ explizite Speicherung d. Messwerte

Nachteile:

- ↳ hoher Speicheraufwand
- ↳ hoher Zeit- & Hardwareaufwand
- ↳ schlecht für Importance Sampling (rejection sampling nötig)
- ↳ keine gezielte Kontrolle für Anwender
- ↳ schwere Erfassung

Vorteile:

- ↳ hoch realistisch

→ Approximierung durch analytische Funktionen

- ↳ ad hoc: ohne tiefen physikalischen Sinn, zB Phong, Blinn
- ↳ physikalisch basiert, zB Cook-Torrance

Nachteile:

- ↳ selten physikalisch plausibel
- ↳ realistische Modelle komplex und langsam → nicht leicht implementierbar

Vorteile:

- ↳ meistens leicht implementierbar
- ↳ gute Kontrolle für d. Anwender durch die Parameter
- ↳ oft effizient stochastisch auswertbar

### • Was ist eine BRDF?

- diese Funktion kodiert das Reflexionsverhalten einer perfekten Textur auf einmal
- Erfassung & Auswertung sehr Daten- & zeitintensiv
- Einbindung in Renderer aufwendig
- addiert Positionsinformation um individuelle BRDF Datensätze zu kombinieren

### • Was für eine Eigenschaft wünscht man sich vom Standpunkt eines stochastischen Renderingprogrammes aus bei einer BRDF-Darstellung?

- Invertierbarkeit: effiziente Ableitung aus gleichverteilten Zufallszahlen der Samples, die dieser Verteilung gehören
- nur dann ist effizientes importance sampling gewährleistet

### • Was für eine Art von Kamera ist die klassische CG-Kamera?

Was ist charakteristisch für sie?

- Hochkamera ohne Tiefenunschärfe

### • Warum haben Kameras trotz kleiner "Blendenöffnungen"?

- weil die Tiefenunschärfe mit größerer Blendenöffnung zunimmt
- Trade-off ist bei realen Kameras zw. Belichtungszeit & Tiefenunschärfe

### • Wie kann man die Tiefenunschärfe mit einem stochastischen Renderer simulieren?

- indem man den Anfangspunkt d. Strahls durch einen zufällig gewählten Ort in einer simulierten Blende gehen lässt

### • Was ist participating Media?

- im Vakuum ist Grenz konstant entlang d. Strahls
- in realen Situationen ist Licht gestreut und vermindert
- zwei Schwierigkeiten
  - ↳ Schnittphänomen gibt es bei jedem Punkt d. Mediums
  - ↳ Spektralabhängigkeit ist ein charakteristischer Parameter
- zB.: Wolken, Rauch, Nebel
- Strahlung unterliegt 3 Phänomenen
  - ↳ Absorption (Beers law, Absorptionskoeffizient  $\kappa_a$ )

- ↳ Emission (Abgeben d. Energie)
- ↳ Scattering: in & out (Streuung, Phasenflut, gibt räumliche Ausbreitung an)

### • Was ist der Unterschied zw. pathing & shooting Algorithmen?

- shooting: → Speichertechniken vergleichsweise flexibel
- startet bei der Lichtressource
  - streut die Strahlen bei den getroffenen Punkten, welche dann im Bildraum vorhergesagt werden
- pathing: → startet beim Auge
- sammelt die Strahlen von den getroffenen Punkten
  - z.B.: ray casting, ray tracing, path tracing

- ⇒ welche Algorithmen besser performen, hängt ab von:
- Oberflächentypen
  - Lichtressourcen
  - Bildgröße vs. Szenengröße

### • Was ist radiosity?

- Annahme: alle Oberflächen sind ideal diffus
- Schritte:
- Unterteilung in Oberflächen
  - Festlegen d. Beschrift.
  - Berechnung d. Formfaktoren
  - Berechnung d. radiosity Werte
  - Rendern
- modelliert Lichtausbreitung unter Beachtung d. Energiegleichgewichts
- Probleme:
- ↳ hoher Speicher & Zeitbedarf
  - ↳ globale Beleuchtung schwer realisierbar
  - ↳ keine analytischen Primitive

### • Was ist hierarchisches radiosity?

- erhöhte Anzahl v. patches sowohl auf Empfänger als auch Senderebene
  - ↳ erhöhte Berechnungszeit - Signifikanz
  - ↳ erhöht Exaktheit d. Lösung nicht sehr viel
- Berechnung zeitintensiv v. Formfaktoren
- Entscheidungsprozeß für jeden Strahl ob er verfolgt werden soll
  - ↳ rekursiv!