

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

Durch die Renderinggl.: $L = L^e + L \cdot T \rightarrow$ unendlich. dim. Integral

- ⇒ integriert über die ganze Hemisphäre
- ⇒ Term, der für d. direkten Einfluss verantwortlich ist, ist außerhalb d. Integrals
- ⇒ beschreibt das Problem d. Lichttransports vom Blickpunkt d. Betrachters
- ⇒ individuelle Gegenwart sind bezogen für jeden Strahl
- ⇒ rekursiv, nicht analytisch lösbar
- ⇒ L^e gibt an, wie viel Licht von x' nach x abstrahlt wird, Emissionsterm
- ⇒ $L \cdot T$ 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 Approximation
- ⇒ 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 Prozess
 - $x = 0,1x + 1,8$ $x_1 = 0,1 \cdot (0,1x + 1,8) + 1,8 \dots$ usw.
 - die Terme Konsolidation ist: $\lim_{n \rightarrow \infty} T^n \cdot L = 0$
 - daher ist $L = \sum_{n=0}^{\infty} T^n \cdot L^e$ eine Lösung
 - für jedes Rekursionslevel muss über die gesamte Hemisphäre integriert werden
 - hoch-dimensionale Integrale müssen gelöst werden
 - Pfade müssen unabhängig sein, keine temporäre Darstellung nötig, Bezeichnung paralleler Wege
- ⇒ Iteration: → rechnet mit einem Stahl, der die Beleuchtung der Szene kodiert
 - ⇒ inhaltliche Diskretisierung bei allen Radiosity-Methoden
 - 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.
 - Volumen kann gut ausgenutzt werden
 - toll 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 Photon? Welches Verfahren verwendet diese Methode in d. Praxis?

- ⇒ rechnet mit einem Stahl, der die Beleuchtung der Szene kodiert
- ⇒ inhaltliche Diskretisierung bei allen Radiosity-Methoden
- z.B. $x_n = 0,1x_{n-1} + 1,8$ $L_{n+1} = L^e + T L_n$
- erfordert überlappungsfreie Überdeckung und endliche Elementrepräsentation
- Lösung auch berechnet für nicht sichtbare Teile d. Szene
- nicht exakt bei hochfrequenten Schatten, Spiegelungen, etc.
- Volumen kann gut ausgenutzt werden
- toll errechneten Funktionen sind Betrachterunabhängig
- bessere Ergebnisse als bei Expansion

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

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

• Welche Expansionsmethoden gibt es?

• Welche Expansionsmethoden gibt es?

- ⇒ Raytracing: → werkt 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 Augenpunkt wird hervorgehoben
 - 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 und cool
 - funktioniert nicht für kleine Lichtquellen → Verlierer → Bildrauschen
 - wird d. multiple Importance Sampling reduziert
- ⇒ Bidirektionales Path Tracing: → sowohl vom Augenpunkt als auch v. d. Lichtquelle werden Strahlen gesendet und anschließend kombiniert
 - bessere Leistung als PT, auch besser bei Kästchen
- ⇒ Metropolis: → Erweiterung d. BPT, schwierig zu implementieren
 - BPT bis ein nützlicher Pfad gefunden wurde
 - Abhängig d. Wege um bessere zu erhalten (Varianten hierausgefiltert und gelöscht)
 - meistert komplexe Situationen gut
 - Bias: für normale Szenen suboptimal
- ⇒ Photon Tracing: → arbeitet mit Photon Maps / Light Maps
 - sehr effizient für Kästchen
 - Photonen gespeichert in 1D-Bäumen
 - langsamer, tritt Speicher, Beleuchtungsrekonstruktion abhängt v. Diskretisierung, biasfrei
 - Photonen können gestraut, absorbiert, reflektiert od. gebrochen werden
 - Photonen werden v. Lichtquelle gewendet
 - Photonen haben Tiefe & Leistung → Beleuchtungsrekonstruktion

• 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
 - Nachteil: begrenzte Auflösung, hohe Setup-Kosten (Flächeninhalt aller Texel muss bekannt sein); Interpolation notig
 - jedes Texturelement beinhaltet d. Durchschnitt d. Energie von allen Photontreffern
- ⇒ Photon Maps: → speichern einzelne Photonen
 - Vorteile: hohe Genauigkeit, hohe Auflösung v. A. bei Kästchen, kaum Setup-Kosten
 - Nachteil: extrem hoher Speicheraufwand; schwierige, fehleranfällige und langsame Rekonstruktion, komplexe Implementierung
 - immer genutzt in 2- od. multipass-Rendern
 - 3 Arten: Caustic, Global, Shadow Photon Map

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

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

• Wie ist die Struktur d. Pixar Renderkern in etwa aufgebaut? Was sind die Stärken & Schwächen?

- ist ein batch-orientiertes Scanline Rendernsystem
- hoch flexibel, zuverlässig und robust
- flexibel, da an beliebiger Stelle im Arbeitsablauf eingriffen 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. Klientmer
- Jeder API Aufruf modifiziert diesen state
- alle API Aufrufe sind ständig veragg, die mit R1_NULL terminieren
- die meisten Aufrufe arbeiten mit Oberflächeneigenschaften
- PRM Interface unterstützt folgende Shader: light Source, Surface, Volume, Displacement, Images
- robust, da PRM Algorithmus extrem primitiv ist: weder Schatten noch Reflexionen können berechnet werden, werden durch shadow & reflection maps vorgetäuscht aber dafür kann man Schatten dort hinplatzieren wo man will und ist nicht von der physikalischen Korrektheit abhängig

- Wie arbeitet d. Radiance-Renderer in etwa? Was hat er für Stufen u. Schichten?
- teilweise stochastischer Hybrid-Renderer, auf typische Architekturen optimiert
- Kausatien nicht korrekt dargestellt
- nutzt großes Maß an aufeinander abgestimmten Heuristiken
 - ↳ automatische u. benutzerdefinierte Erzeugung v. Hilflichtquellen
 - ↳ Radiance 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 \int_X präzisionsverfahren erfordern rekursive Berechnung v. hochdim. Integranden → lässt sich mit MC effizient lösen
- unterliegen DNI (deterministischen numerischen Integrationsmethoden) bei niedrigen Dimensionen, aber je höher Dimensionen, desto ineffizienter DNI, während MC konstante Leistung bringt

- Was ist Quasi Monte Carlo? Effiziente v. Dimension abhängig
- weder reine Zufallszahlen noch eine gleichmäßige Abstufung sind sonderlich effizient
- daher will man Zufallszahlen, die die Ebene regelmäßig bedecken ohne regelmäßige Struktur jedoch → Pseudozufallszahlen
- Fehler hängt v. Anzahl d. Punkte und deren "Gleichmäßigkeit" (Diskrepanz)
- Diskrepanz = Abweichung aller Punkte v. idealer Gleichverteilung auf d. Intervall $B: [0,1]^d$

- Was ist importance sampling im Zusammenhang mit stochastischen Renderingprogrammen?

- Technik zum Beschleunigen d. Konvergenz der MC: Integrand wird dort ausgewertet, wo er numerisch groß ist
- Bsp: Beim Verfolgen einer rekursiven, stochastischen Pfade an einem gegebenen Punkt wird die neue Strahlrichtung basierend in die Richtung gewählt, in die die Ψ RDF groß ist (am meisten Licht wird reflektiert)
- gleichmäßige Abstufung → 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 Schätzwerk qualitativ gut
 - ↳ Auswahl dieses Werks nur durch Heuristiken möglich
 - Bsp: Path Tracing, Bidirektionales Path Tracing

- Was ist Spektralrendering?

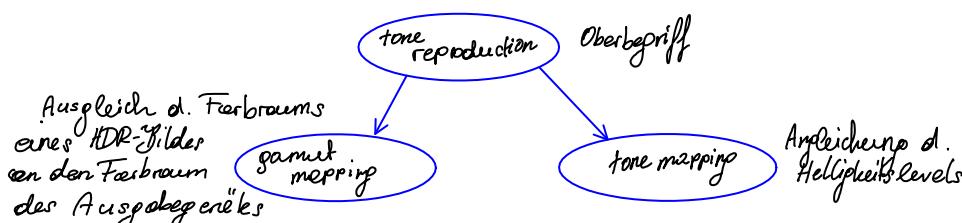
- Bildsynthese, bei der nicht nur Farbwerten, sondern mit spektralen Darstellungen v. Licht und Reflektivität gerechnet wird
- Schritte: Spektrum vorbereiten, Aufbereitung d. spektralen Abstufungen durch Rendereingberechnung, Berechnung d. Endbildes mittels CIE-Farbabgleich-Funktionen und Standardtransformationen

- Was sind die Vor- & Nachteile von Spektralrendering gegenüber der Bildsynthese im Farbraum?

- Vorteile: Vorfahrt d. Ausschuss möglich, Einbindungen v. nutzbar

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

- Wie können Spektren intern dargestellt werden?
 - spektrale Bänder Sampling (schnell, ungenau od. hoher Speicheranforderung)
 - Basisfunktionen (genau, geringer Speicheranforderung, langsam)
 - Hybridansätze (komplex, besten Eigenschaften aus beiden Ansätzen)
- Welche Schwierigkeit bringt es bei d. Implementierung wenn man Polarisation und/od. Fluoreszenz in einem Renderingsystem 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 physikalische Rendering interessant?
 - Bildformate, die mehr als 8bit pro Kanal speichern können
 - weil viele Szenen einen weit höheren Kontrastumfang haben als d. in 24bit verfügbare
- Welche HDR-Grafikformate gibt es?
 - OpenEXR (16-bit Gleitkommazahlen pro Farbkanal, 32bit Daten für Speicherung v. Tiefeinformation & Objektkennungen, verlustlose Komprimierung v. Bildern durch mitgelieferte Codecs)
 - Dng, Dpx, Tiff (nutzt CIE, 32bit)
- Warum gibt es praktisch keine spektralen Bildformate?
 - hoher Speicheranforderung
 - kaum Bedarf
 - einzige Verwendung: interne Speicherformate v. spektralen Renderings
- Was ist tone mapping?
 - Anpassung d. Helligkeitslevels in einem HDR-Bild auf die Möglichkeiten eines LDR-Ausgabegeräts
- Was für Kategorien v. TM/TR-V erfahren gibt es?
 - 3 verschiedene Ansätze
 - Globale Methoden (skalieren d. Belichtungsfaktur durch einen ges. Faktor, einfach und schnell)
 - ↳ räumlich uniform
 - ↳ linearer Skalierungsfaktor
 - ↳ nicht-linearer Skalierungsfaktor
 - Lokale Methoden (Unterschiede zw. Bildbereichen mit einberücksichtigen)
 - ↳ räumlich nicht uniform
 - wahrnehmende Ansätze (Eigenschaften d. menschl. visuellen Systems werden mit einberücksichtigt → realistische Darstellung)
- Kurze Begriffs erklärung: gamut mapping, tone mapping & tone reproduction?



- Mit welchen Funktionen werden Oberflächen beschrieben?
 - ⇒ BRDF: Funktionen für d. Reflexionsverhalten v. Oberflächen
 - 2 Arten der Darstellung von BRDF-Werten
 - explizit Spezifizierung d. Messwerte
 - ↳ hoher Speicheranforderung
 - ↳ hoher Zeit- & Hardwareaufwand

- ↳ hoher Zeit- & Hardwareaufwand
- ↳ schlecht für Importance Sampling
- Approximation durch analytische Funktionen
- ⇒ BTDF: ähnlich BRDF, aber für Strahlen, die zum Metrikt zeigen
- ⇒ BSDF: Kombination aus BRDF & BTDF
 - ↳ selten genutzt
 - ↳ über den ganzen Raum definiert

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

- explizit Spezierung d. Messwerte

Nachteile:

- ↳ hoher Spezierungsaufwand
- ↳ hoher Zeit- & Hardwareaufwand
- ↳ schlecht für Importance Sampling (reichen sampling nötig)
- ↳ keine serielle Kontrolle für Anwender
- ↳ schwierige Erfassung

Vorteile:

- ↳ hoch realistisch

→ Approximation durch analytische Funktionen

- ↳ ad hoc: ohne tieferen 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 BTDF?

- diese Funktion kodiert das Reflexionsverhalten einer ganzen 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 Renderingsprogrammes aus bei einer BRDF-Darstellung?

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

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

Was ist charakteristisch für sie?

- Lochkameras ohne Tiefenunschärfen

• Warum haben Kameras dann trotzdem kleine Blendenöffnungen?

- weil die Tiefenunschärfen mit größerer Blendenöffnung zunehmen
- trade-off ist bei realen Kameras zw. Belichtungszeit & Tiefenunschärfen

• Wie kann man die Tiefenunschärfen 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 Volumen ist Glanz konstant entlang d. Strahls
- in realen Situationen ist Licht gestreut und vermindert
- zwei Schwierigkeiten
 - ↳ Schattphänomene gibt es bei jedem Punkt d. Mediums
 - ↳ Spektralabhängigkeit ist ein charakteristischer Parameter
- z.B.: Wolken, Milch, Nebel
- Strahlung unterliegt 3 Phänomenen
 - ↳ Absorption (Beer's Law, Absorptionskoeffizient κ)

- ↳ Emission (Abdicken d. Energie "u")
- ↳ Scattering: in & out (Streuung, Phasenfl. gibt räumliche Ausbreitung an)

• Was ist der Unterschied zw. gathering & shooting Algorithmen?

shooting: → Speichertechniken vergleichsweise flexibel

→ spart bei der Lichtressource

→ spez. die Strahlen bei den getroffenen Punkten, welche dann im Bildraum vorhergesagt werden

gathering: → startet beim Aufruhr

→ 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 Beobachtung d. Energiedurchgangs

→ Probleme: ↳ hoher Speicher & Zeitbedarf

↳ globale Beleuchtung schwer realisierbar

↳ keine analytischen Primitiven

• Was ist hierarchischer radiosity?

→ erhöhte Anzahl v. patches sowohl auf Empfänger als auch Senderseite

↳ erhöhte Berechnungszeit - Signifikanz

↳ erhöht Exaktheit d. Lösung nicht sehr viel

→ Berechnung zeitintensiv v. Formfaktoren

→ Entscheidungsprozesse für jeden Strahl ob er verfeinert werden soll

↳ Schätzpr.!