Einführung in die Computergrafik

Matthias B. Hullin

Institut für Informatik II, Universität Bonn

UNIVERSITÄT BONN

Organisatorisches

Prüfungstermin Klausur 1:

Mittwoch 7. August 2024, 10:00-12:00, HS1

Prüfungstermin Klausur 2:

Montag 9. September 2024, 10:00-12:00, MA II

Organisatorisches zum Projekt

Uploadlink:
https://uni-bonn.sciebo.de/s/yAsHycaf1WYNJld

Deliverable 0 (bitte ZIP bis **Di 16. Juli** 23:59 hochladen):

- **Video-Vorschau**, beliebige Qualität/Auflösung
- Verpflichtend!

Deliverable 1 (bitte ZIP bis **Do 18. Juli** 23:59 hochladen):

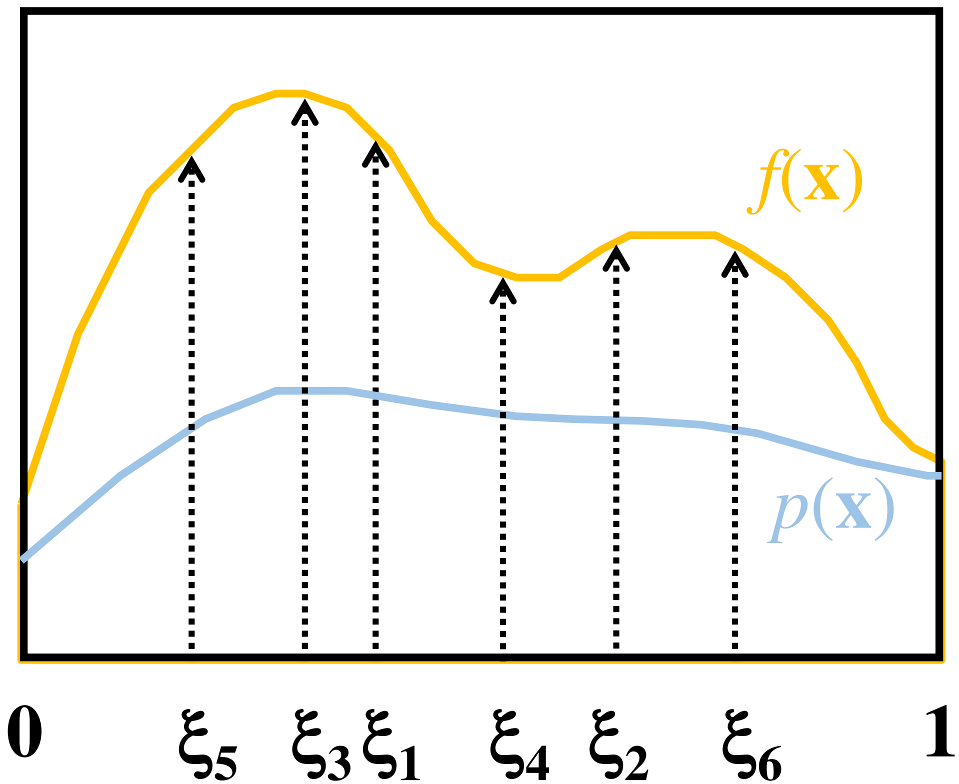
- **Kurzfilm-Video** (in hoher Qualität als H.264 codiert)
- **Kurzvortrag** (2 Minuten) als PDF, PPT(x) oder Video
  - Dieser Vortrag ist optional, wenn ein Film eingereicht wurde. Ohne Film ist er **umso wichtiger!**

Monte Carlo-Integration II

Enthält Material von Jaroslav Křivánek, MFF UK

Monte-Carlo-Integration

- Allgemeines Werkzeug zur Schätzung bestimmter Integrale



Integral:

$$I = \int f(\mathbf{x})\mathrm{d}\mathbf{x}$$

Monte-Carlo-Schätzwert für *I*:

$$\langle I \rangle = \frac{1}{N}\sum_{i=1}^{N}\frac{f(\xi_i)}{p(\xi_i)}; \quad \xi_i \propto p(\mathbf{x})$$

"Im Mittel" funktioniert es:

$$E[\langle I \rangle] = I$$

Zufallswerte nach gegebenen Verteilungen

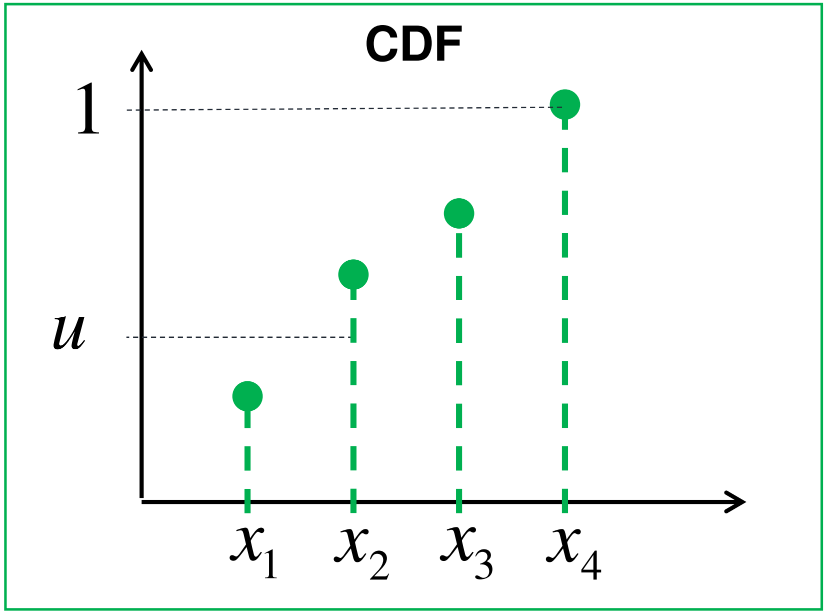
Samples nach diskreter 1D-Verteilung erzeugen

- Gegeben Wahrscheinlichkeitsmassefunktion *p*(*i*), und die dazugehörende CDF *P*(*i*)

- Schema
  1. Erzeuge *u* aus Uniform(0,1)
  2. Wähle x_i für welches

$$\boxed{P(i-1) < u \le P(i)}$$

(wir definieren $P(0) = 0$)



- Die Suche wird üblicherweise durch Intervallschachtelung durchgeführt

Samples nach diskreter 2D-Verteilung erzeugen

- Gegeben Wahrscheinlichkeitsmassefunktion $p_{I,J}(i, j)$

- Option 1:
  - Interpretiere 2D PMF als 1D-Vektor von Wahrscheinlichkeiten
  - Erzeuge Zufallswerte wie im 1D-Fall

Samples nach diskreter 2D-Verteilung erzeugen

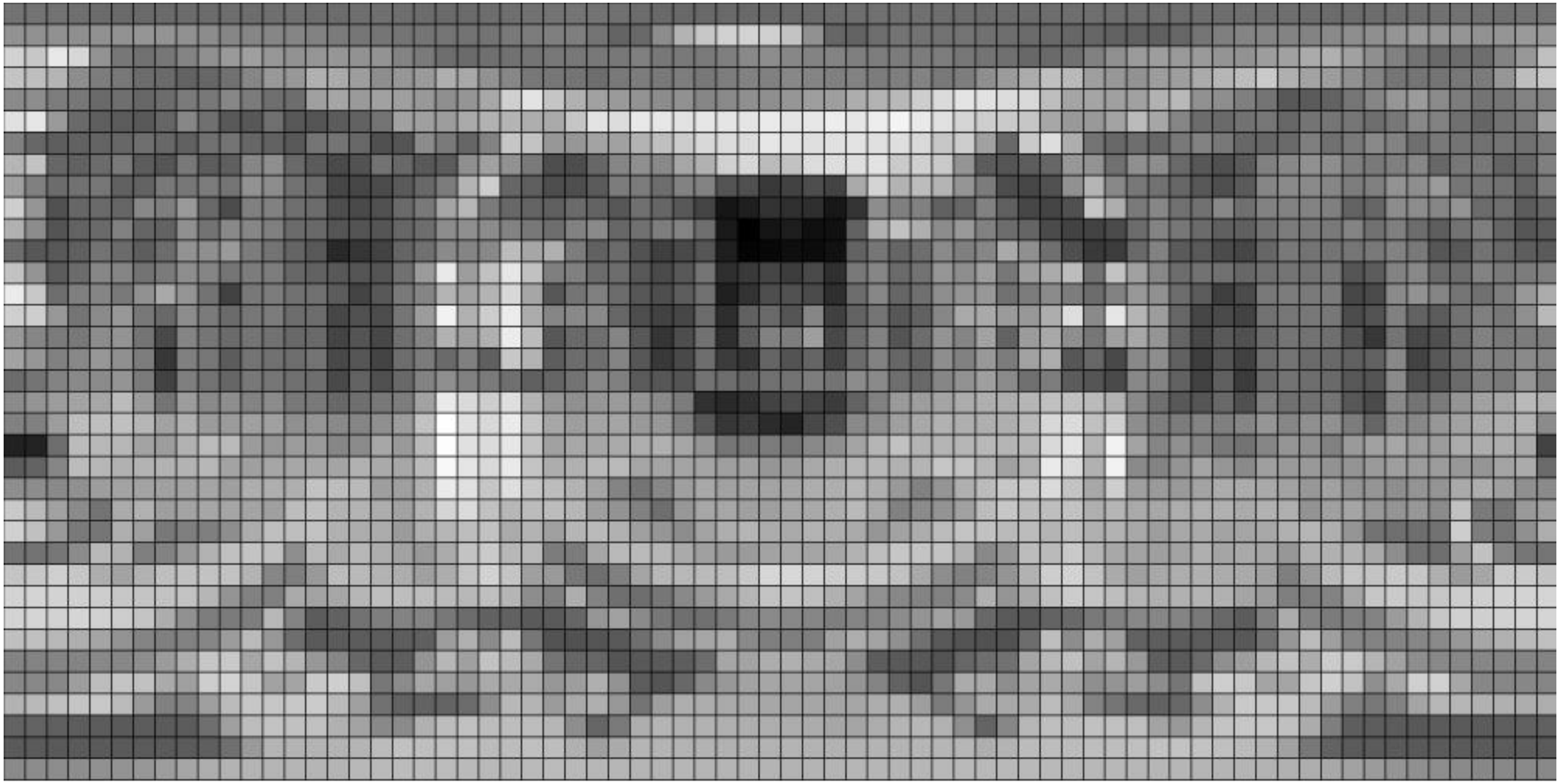
- Option 2 (besser)
  1. “Spalte” i_{sel} wird gemäß Randverteilung (marginal distribution) gewählt, gegeben in Form der 1D marginalen PMF

$$p_I(i) = \sum_{j=1}^{n_j} p_{I,J}(i, j)$$

  2. “Zeile” j_{sel} wird aus der bedingten Verteilung bei gegebener “Spalte” i_{sel} gewählt

$$p_{J|I}(j \mid I = i_{\text{sel}}) = \frac{p_{I,J}(i_{\text{sel}}, j)}{p_I(i_{\text{sel}})}$$

Samples nach diskreter 2D-Verteilung erzeugen



Transformationsmethode vs. Rejection sampling

- Inverse-CDF-Methode (Transformations-Methode) **Vorteile**
  - Fast immer effizienter als rejection sampling (es sei denn, die Transformationsformel $x = P^{-1}(u)$ stellt sich als extrem komplex heraus)
  - Konstante Zeitkomplexität; Anzahl von Zufallswerten vorab bekannt
- Transformations-Methode **Nachteile**
  - Möglicherweise nicht durchführbar (evtl. lässt sich keine geeignete Form für $x = P^{-1}(u)$ finden), aber rejection sampling ist immer anwendbar, solange wir die PDF auswerten können (d.h., rejection sampling allgemeiner)
- Schlaues rejection sampling kann sehr effizient sein (siehe etwa die "Ziggurat method" / Wikipedia)

Sampling einer kontinuierlichen 2D-Verteilung

- Ähnlich zum 2D diskreten Fall
- Vorgehensweise:
  - Gegeben: Dichte $p_{X,Y}(x, y) = p_X(x)\, p_{Y/X}(y \mid x)$

1. Wähle x_{sel} gemäß **marginaler PDF**

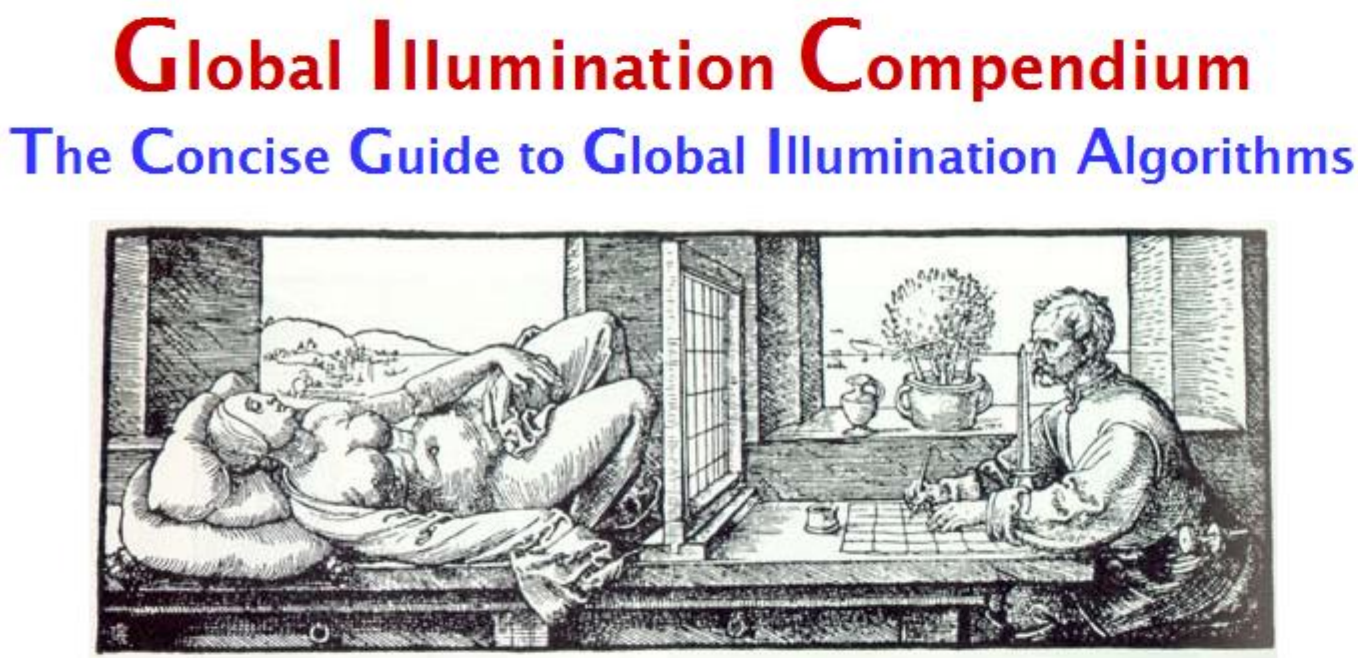
$$p_X(x) = \int p_{X,Y}(x, y)\,\mathrm{d}y$$

2. Wähle y_{sel} gemäß **bedingter PDF**

$$p_{Y|X}(y \mid X = x_{\mathrm{sel}}) = \frac{p_{X,Y}(x_{\mathrm{sel}}, y)}{p_X(x_{\mathrm{sel}})}$$

Transformationen für gängige Fälle im Rendering

- P. Dutré: **Global Illumination Compendium**, https://people.cs.kuleuven.be/~philip.dutre/GI/



Albrecht Duerer, *Underweysung der Messung mit dem Zirkel und Richtscheyt* (Nurenberg, 1525), Book 3, figure 67.

Importance Sampling der Phong-BRDF

- Strahl trifft Oberfläche mit Phong-BRDF. Wie erzeugen wir den Strahl, um den Lichtpfad fortzusetzen??

- Vorgehensweise

  - Wähle die BRDF-Komponente (diffuse Reflexion, spekulare Reflexion, Brechung, ...)

  - Sample die gewählte Komponente

  - Werte die gesamte PDF und BRDF aus

Phong-BRDF

$$f_r^{\text{Phong}}(\omega_\text{i} \to \omega_\text{o}) = \frac{\rho_\text{d}}{\pi} + \frac{n+2}{2\pi}\rho_\text{s} \max\{0, \cos\theta_\text{r}\}^n$$

- wobei

$$\cos\theta_\text{r} = \omega_\text{o} \cdot \omega_\text{r}$$

$$\omega_\text{r} = 2(\omega_\text{i} \cdot \mathbf{n})\mathbf{n} - \omega_\text{i}$$

- Energieerhaltung:

$$\rho_d + \rho_s \leq 1$$

Auswahl der BRDF-Komponente

```
pd = max(rhoD.r, rhoD.g, rhoD.b);
ps = max(rhoS.r, rhoS.g, rhoS.b);
pd /= (pd + ps);    // prob of choosing the diffuse component
ps /= (pd + ps);    // prob of choosing the specular comp.

if (rand(0,1) <= pd)
    genDir = sampleDiffuse();
else
    genDir = sampleSpecular(incDir);

pdf = evalPdf(incDir, genDir, pd, ps);
```

Diese Folie enthält einen Bug! Viel Spaß beim Suchen :)

Sampling der diffusen Reflexion

- Importance Sampling mit Dichte $p(\theta) = \cos(\theta) / \pi$
  - θ…Winkel zwischen Oberflächennormale und erzeugtem Strahl
  - Erzeuge die Richtung:

$$\varphi = 2\pi r_1 \qquad \theta = \operatorname{acos}(\sqrt{r_2})$$

$$x = \cos(2\pi r_1)\sqrt{1-r_2}$$
$$y = \sin(2\pi r_1)\sqrt{1-r_2}$$
$$z = \sqrt{r_2}$$

    - r_1, r_2 … uniforme Zufallsvariablen auf $[0,1)$
  - Referenz: Dutre, Global illumination Compendium (online)
  - Herleitung: Pharr/Huphreys/Jakob, PBRT

sampleDiffuse()

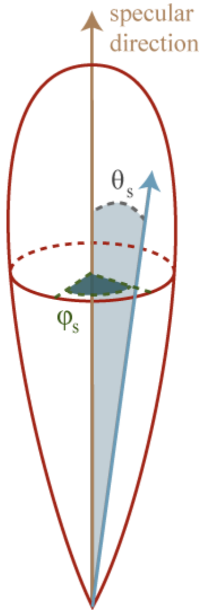
```
// generate spherical coordinates of the direction
float r1 = rand(0,1), r2 = rand(0,1);
float sinTheta = sqrt(1 – r2);
float cosTheta = sqrt(r2);
float phi      = 2.0*PI*r1;
float pdf      = cosTheta/PI;

// convert [theta, phi] to Cartesian coordinates
Vec3 dir (cos(phi)*sinTheta, sin(phi)*sinTheta, cosTheta);

return dir;
```

Sampling der glänzenden (spekularen) Reflexion

- Importance Sampling mit PDF $p(\theta) = \frac{n+1}{2\pi}\cos^n(\theta)$
  - θ …Winkel zwischen idealer Spiegelrichtung von ω_0 und dem erzeugten Strahl
  - Formeln zum Erzeugen der Richtung:



$$\varphi = 2\pi r_1 \qquad x = \cos(2\pi r_1)\sqrt{1 - r_2^{\frac{2}{n+1}}}$$

$$\theta = \operatorname{acos}\left(r_2^{\frac{1}{n+1}}\right) \qquad y = \sin(2\pi r_1)\sqrt{1 - r_2^{\frac{2}{n+1}}}$$

$$z = r_2^{\frac{1}{n+1}}$$

- r_1, r_2 … uniforme Zufallsvariablen auf $[0,1)$

sampleSpecular()

```
// build a local coordinate frame with R = z-axis
Vec3 R = 2*dot(N,incDir)*N – incDir;  // ideal reflected direction
Vec3 U = arbitraryNormal(R);          // U is perpendicular to R
Vec3 V = crossProd(R, U);             // orthonormal basis with R and U

// generate direction in local coordinate frame
Vec3 locDir = rndHemiCosN(n);  // formulas form prev. slide, n=phong exp.

// transform locDir to global coordinate frame
Vec3 dir = locDir.x * U + locDir.y * V + locDir.z * R;

return dir;
```

evalPdf(incDir, genDir, pd, ps)

```
return
        pd * getDiffusePdf(genDir) +
        ps * getSpecularPdf(incdir, genDir);
```

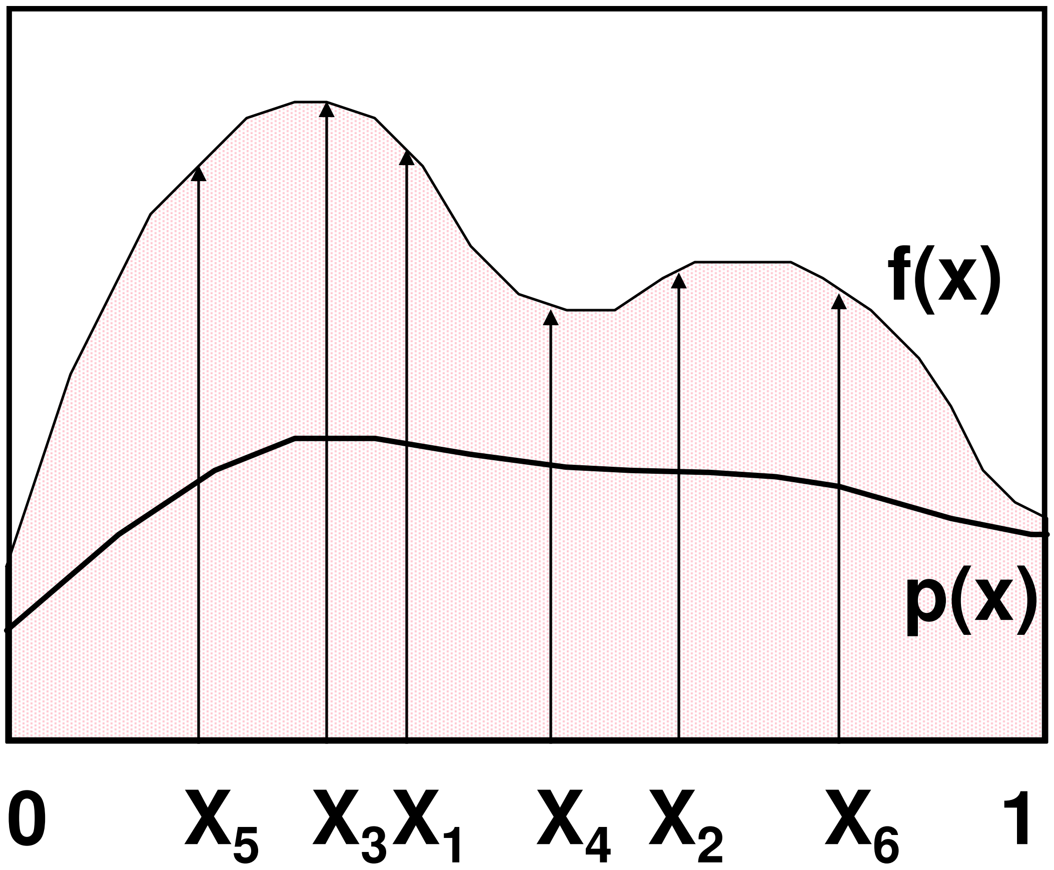
Formeln von vorigen Folien

Varianzreduzierung für MC-Schätzer

Methoden zur Varianzreduzierung

- **Importance sampling**
  - Häufigster Ansatz im physikalisch basierten Rendering (meist verwenden wir BRDF-proportionales Importance Sampling)
- **Kontrollvariate**
- **Verbesserte Sampleverteilung**
  - Stratifizierung
  - quasi-Monte Carlo (QMC)

Importance Sampling



Importance Sampling

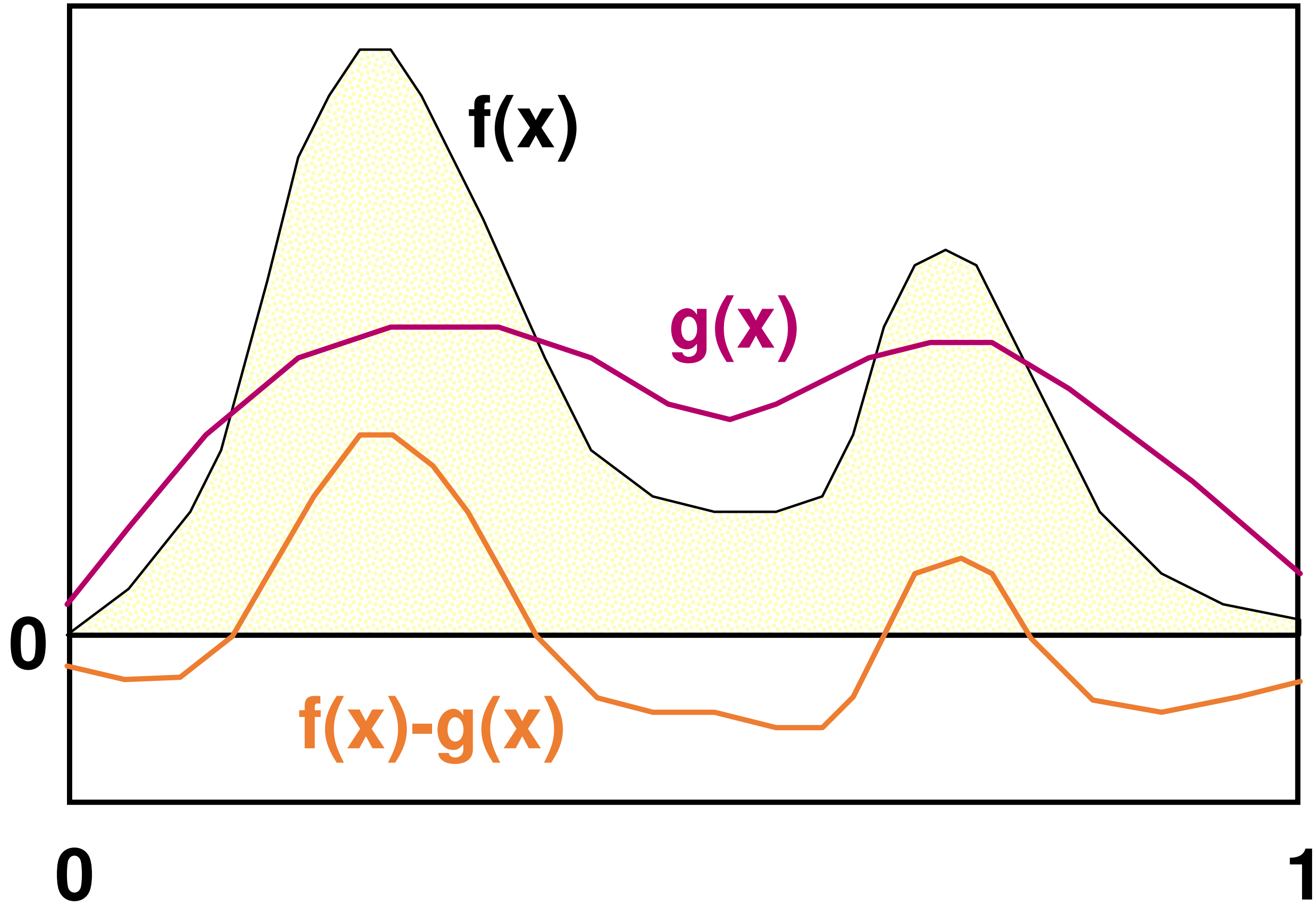
- Teile des Integrationsgebiets mit hohem Wert des Integranden f sind wichtiger
  - Samples aus diesen Bereichen haben eine höhere Auswirkung aufs Ergebnis

- **Importance Sampling** bevorzugt Auswertung dieser Bereiche

  - d.h., die **PDF** p ist “ähnlich” zum Integranden f

- **Reduziert Varianz** unter Beibehaltung der Erwartungstreue

Kontrollvariate



Kontrollvariate

Betrachte eine Funktion $g(x)$, die den **Integranden annähert** und analytisch integrierbar ist:

$$I = \int f(\mathbf{x})\,\mathrm{d}\mathbf{x} = \underbrace{\int [f(\mathbf{x}) - g(\mathbf{x})]\,\mathrm{d}\mathbf{x}} + \underbrace{\int g(\mathbf{x})\,\mathrm{d}\mathbf{x}}$$

Numerische Integration (MC) hoffentlich mit geringerer Varianz als $f(x)$ direkt zu integrieren.

Können wir analytisch integrieren

Kontrollvariate vs. Importance Sampling

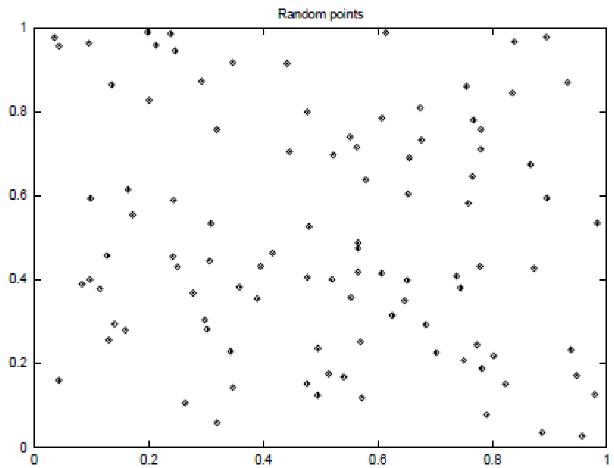
- **Importance Sampling**
  - Vorteilhaft, wenn die Funktion, die wir sampeln wollen, im Integranden als **multiplikativer Faktor** auftaucht (z.B. BRDF in der Reflexionsgleichung)

- **Kontrollvariate**
  - Besser, wenn die Funktion, die wir analytisch integrieren können, im Integranden als **additiver Term** auftritt.

- Daher verwenden wir im Rendering fast immer Importance Sampling und fast nie Kontrollvariate.

Bessere Sampleverteilung

- Erzeugung unabhängiger Samples führt oft zu Klumpenbildung
  - Ergebnis: hohe Varianz des Schätzers

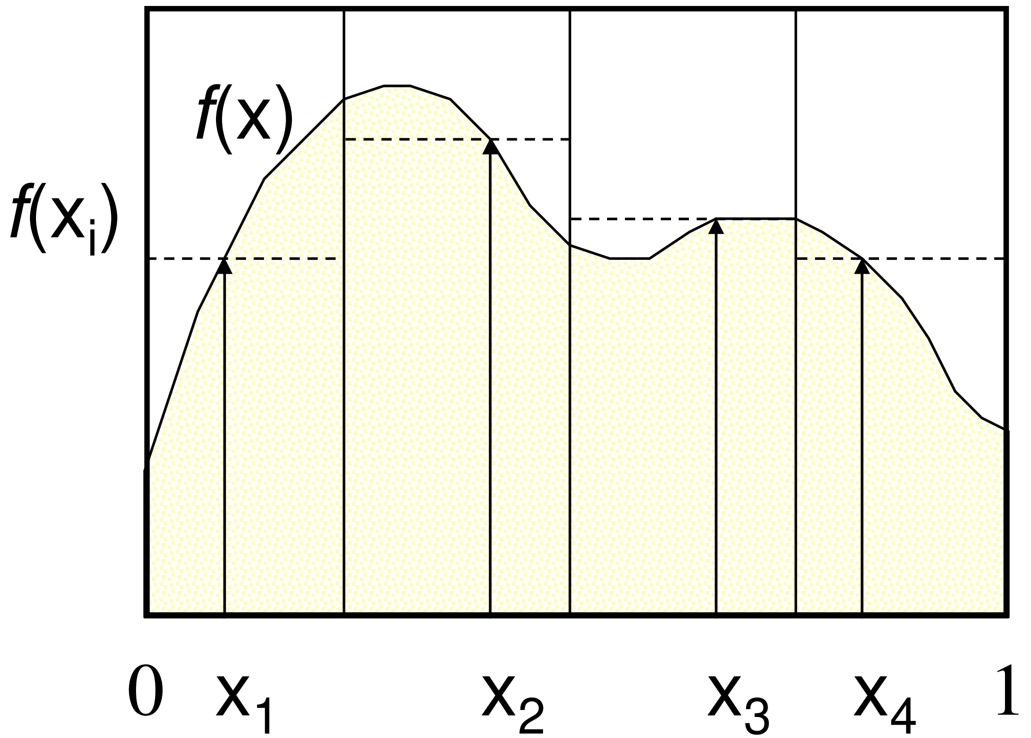


- Bessere Sampleverteilung: => bessere Abdeckung des Integrations-bereichs mit Samples => geringere Varianz

- Ansätze:
  - **Stratifiziertes Sampling** ("geschichtete Zufallsstichprobe")
  - **Quasi-Monte Carlo (QMC)**

Stratifizierte Abtastung

- Abtastgebiet unterteilt in disjunkte Teilgebiete, die unabhängig voneinander abgetastet werden



Stratifizierte Abtastung

Unterteilung des Gebietes Ω in N Teile Ω_i:

$$I = \int_{\Omega} f(x)\,\mathrm{d}x = \underline{\sum_{i=1}^{N} \int_{\Omega_i} f(x)\,\mathrm{d}x} = \sum_{i=1}^{N} I_i$$

Resultierender Schätzer:

$$\hat{I}_{\text{strat}} = \underline{\frac{1}{N} \sum_{i=1}^{N} f(X_i),} \quad X_i \in \Omega_i$$

Stratifizierte Abtastung

- Unterdrückt Klumpenbildung

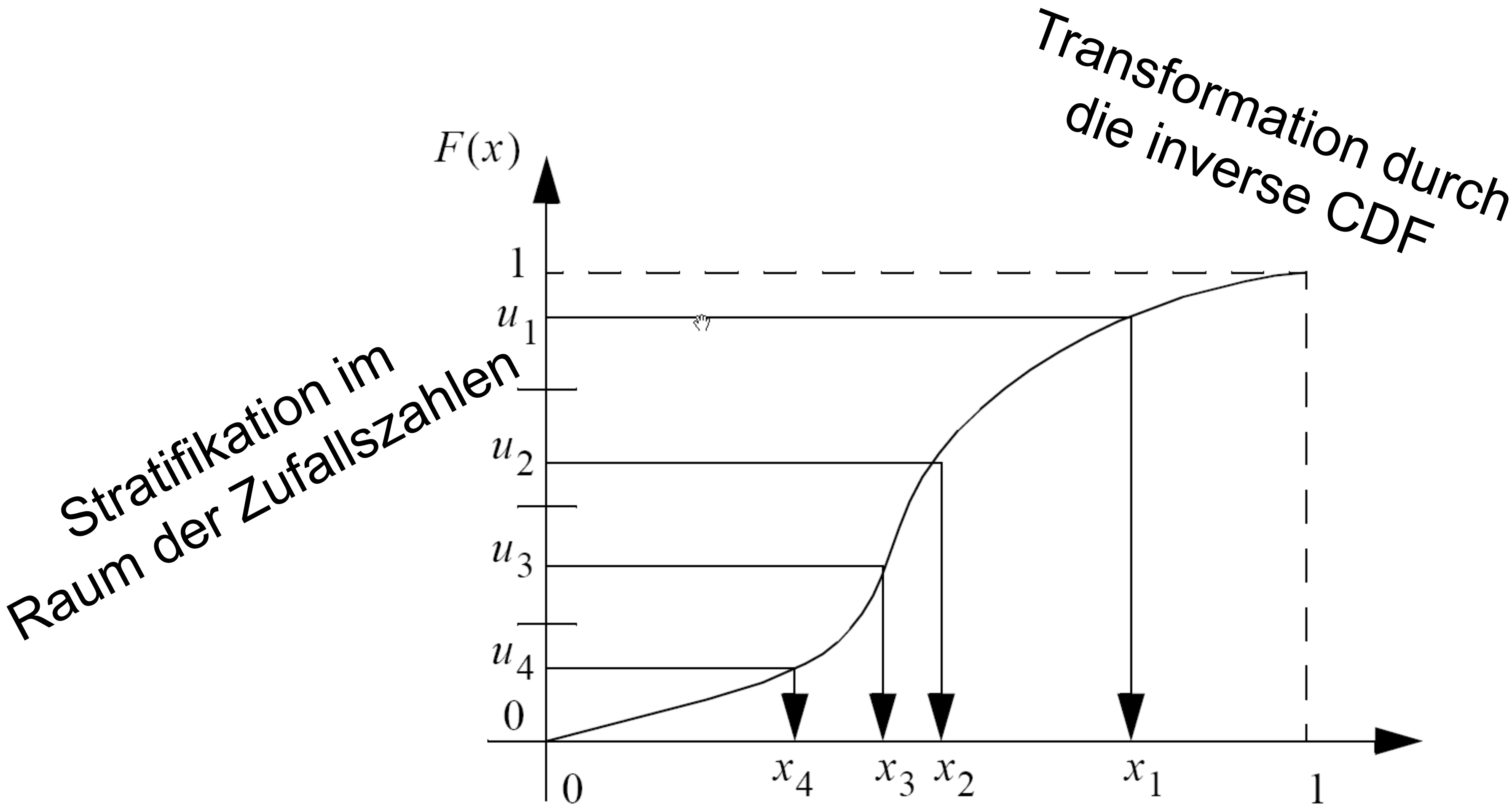
- Reduziert Varianz des Schätzers
  - Varianz beweisbar kleiner oder gleich der eines regulären Sekundärschätzers

- Sehr effektiv in niedrigen Dimensionen
  - Effektivität nimmt für hochdimensionale Integranden ab

Wie unterteilen wir das Intervall?

- **Uniforme** Unterteilung des Intervalls
  - Natürlicher Ansatz für komplett unbekannten Integranden f
- Wenn die Form des Integranden f wenigstens grob bekannt ist, streben wir eine Unterteilung an, so dass die Teilgebiete eine kleinstmögliche Varianz haben
- Unterteilung eines d-dimensionalen Intervals führt zu N^d Samples
  - Besserer Ansatz in hohen Dimensionen ist "***N*-rooks** sampling".

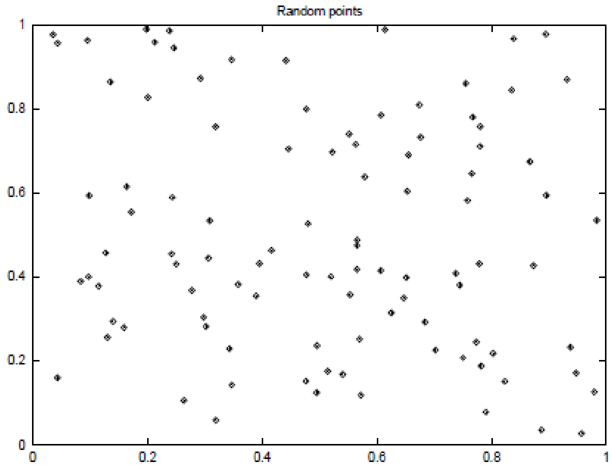
Stratified Sampling + Transformationsmethode



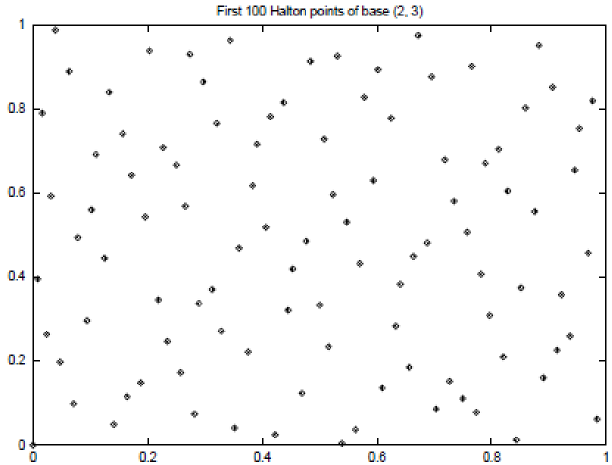
Quasi-Monte-Carlo-Verfahren (QMC)

- Verwende strikt deterministische Folgen statt (Pseudo-) Zufallszahlen
- Pseudo-Zufallszahlen ersetzt durch **low-discrepancy sequences** (Folgen niedriger Diskrepanz)
- Alles funktioniert wie in "normalem" MC, aber die zugrundeliegende Mathematik ist anders (nichts ist zufällig, daher kann die Mathematik nicht auf Wahrscheinlichkeitstheorie aufgebaut werden)

Diskrepanz

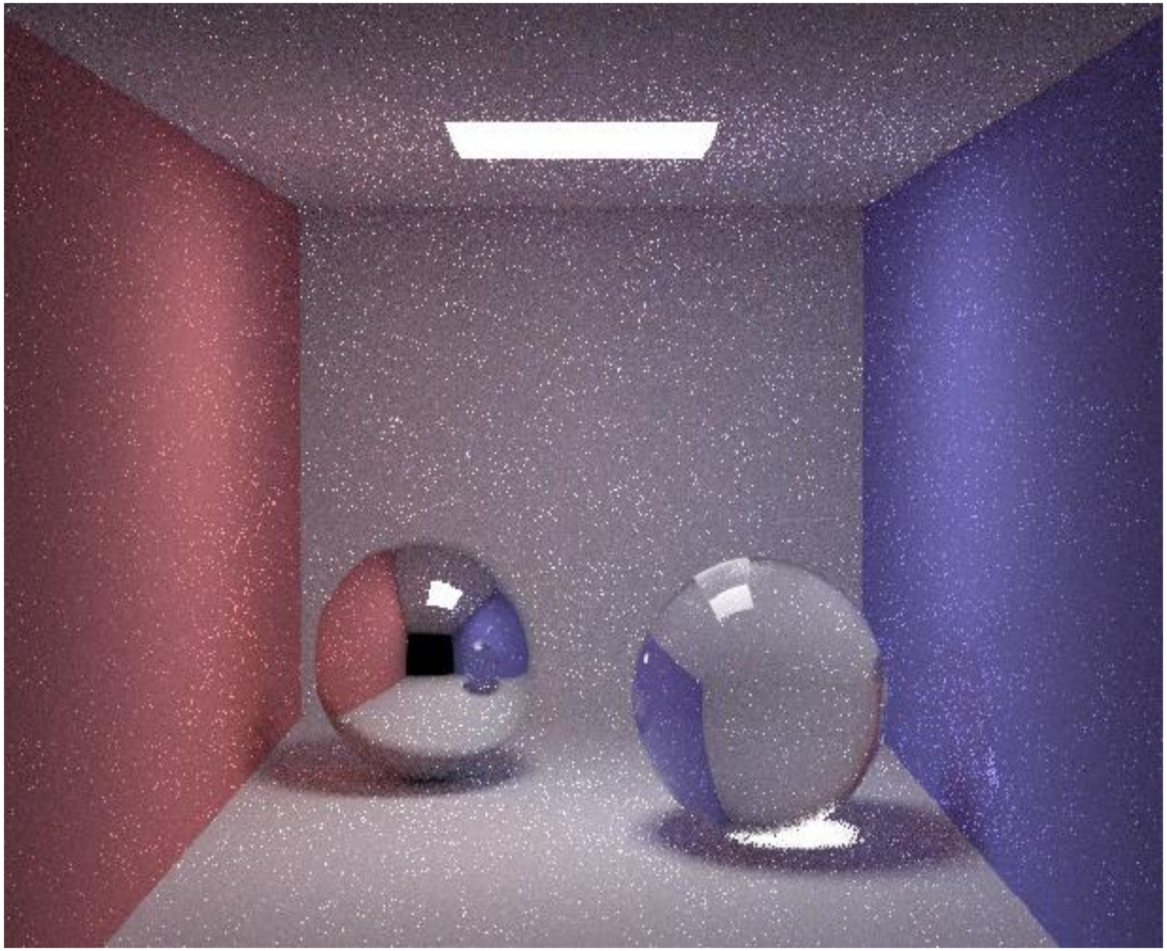


Hohe Diskrepanz
(Cluster von Punkten)



Niedrige Diskrepanz
(gleichmäßigere Verteilung)

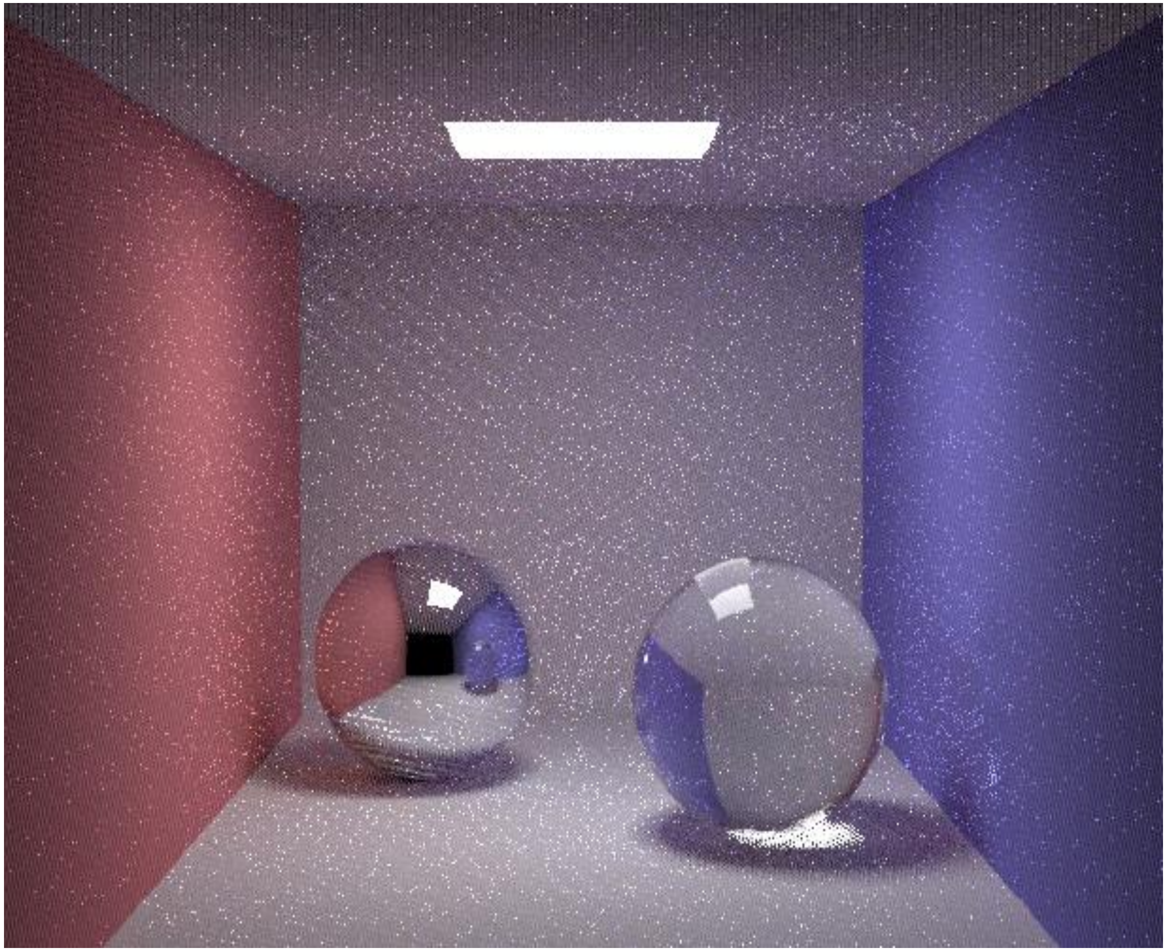
Stratified sampling



Henrik Wann Jensen

10 paths per pixel

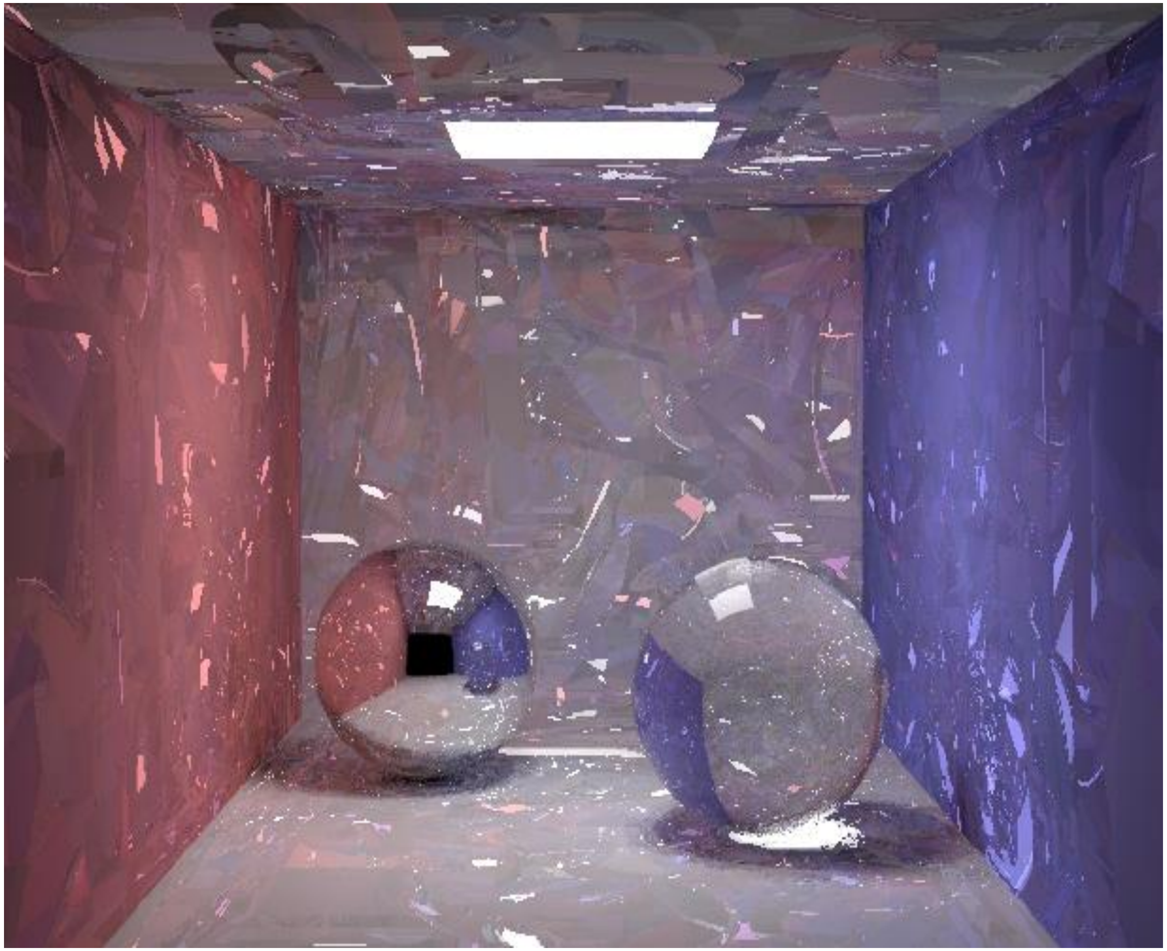
Quasi-Monte Carlo



Henrik Wann Jensen

10 paths per pixel

Same random sequence for all pixels



10 paths per pixel

Henrik Wann Jensen

Path tracing

Pfade von der Kamera aus verfolgen

```
renderImage()
{
  for all pixels
   {
          Color pixelColor = (0,0,0);
          for k = 1 to N
          {
          wk := random direction through the pixel
          pixelColor += getLi(camPos, wk)
          }
          pixelColor /= N;
          writePixel(pixelColor);
   }
}
```

Pathtracing, Version Null (rekursive Form)

```
getLi(x, ω):
    y = traceRay(x, ω)
    return
        Le(y, -ω) + // emitted radiance
        Lr(y, -ω) // reflected radiance


Lr(x, ω):
    ω’ = genUniformHemisphereRandomDir(n(x))
    return 2p * brdf(x, ω, ω’)
                * dot(n(x), ω’) * getLi(x, ω’)
```

Path Tracing – Schleifenversion

```
getLi(x, w)
{
   Color thrput = (1,1,1)
   Color accum  = (0,0,0)
   while(1)
   {
      hit = NearestIntersect(x, w)
      if no intersection
         return accum + thrput * bgRadiance(x, w)
      if isOnLightSource(hit)
         accum += thrput * Le(hit.pos, -w)
      ρ = reflectance(hit.pos, -w)
      if rand() < ρ // russian roulette - survive (reflect)
         wi := SampleDir(hit)
         thrput *= fr(hit.pos, wi, -w) * dot(hit.n, wi) / (ρ * pdf(wi))
         x  := hit.pos
         w  := wi
      else // absorb
          break;
   }
   return accum;
}
```

Pfade abbrechen – Russisch Roulette

- Führe den Pfad mit Wahrscheinlichkeit q fort

- Multipliziere Gewicht (Durchsatz) der überlebenden Pfade mit $1/q$

$$Z = \begin{cases} Y/q & \text{if } \xi < q \\ 0 & \text{otherwise} \end{cases}$$

- Russich Roulette ist unbiased!

$$E[Z] = \frac{E[Y]}{q} \cdot q + 0 \cdot \frac{1}{q-1} = E[Y]$$

Diese Folie enthält einen (harmlosen) Bug! Viel Spaß beim Suchen :)

Überlebenswahrscheinlichkeit

- Es ist sinnvoll, die Oberflächenreflektivität r als Überlebenswahrscheinlichkeit zu verwenden
  - Wenn die Oberfläche nur 30% des Lichts reflektiert, fahren wir mit 30% Wahrscheinlichkeit fort. Dies ist konsistent mit der physikalischen Wirklichkeit.
- Was, wenn wir r nicht berechnen können? Dann gibt es eine bequeme Alternative:
  - Wähle erst eine zufällige Richtung ω_i gemäß $p(\omega_i)$
  - Verwende dieses ω_i, um Überlebenswahrscheinlichkeit zu berechnen:

$$q_{\text{survival}} = \min\left\{ 1, \ \frac{f_r(\omega_\mathrm{i} \to \omega_\mathrm{o}) \cos\theta_\mathrm{i}}{p(\omega_\mathrm{i})} \right\}$$

Richtungssampling

```
getLi(x, w)
{
   Color thrput = (1,1,1)
   Color accum  = (0,0,0)
   while(1)
   {
      hit = NearestIntersect(x, w)
      if no intersection
         return accum + thrput * bgRadiance(x, w)
      if isOnLightSource(hit)
         accum += thrput * Le(hit.pos, -w)
      ρ = reflectance(hit.pos, -w)
      if rand() < ρ // russian roulette – survive (reflect)
         wi := SampleDir(hit)
         thrput *= fr(hit.pos, wi, -w) * dot(hit.n, wi) / (ρ * pdf(wi))
         x  := hit.pos
         w  := wi
      else // absorb
          break;
   }
   return accum;
}
```

Richtungssampling

- Wir sampeln die Richtung ω_i üblicherweise mit einer PDF ungefähr wie

$$f_r(\omega_i, \omega_o)\cos\theta_i$$

- Idealerweise würden wir proportional zum Integranden sampeln,

$$L_i(\omega_i) f_r(\omega_i, \omega_o)\cos\theta_i,$$

- aber das ist schwierig, weil wir L_i vorab nicht kennen. Mit einiger Vorberechnung kann man eine grobe Schätzung von L_i furs Sampling verwenden [Jensen 95, Vorba et al. 2014].

Keine Information über einfallende Radianz [Vorba2014]

Sampling durch einfallende Radianz gesteuert [Vorba2014]

Importance Sampling der BRDF

- Was passiert, wenn die PDF genau proportional zu $f_r(\omega_i, \omega_o) \cos\theta_i$ ist?

$$p(\omega_\mathrm{i}) \propto f_r(\omega_\mathrm{i} \to \omega_\mathrm{o}) \cdot \cos\theta_\mathrm{i}$$

- Normalisierung (PDFs müssen zu 1 integrieren)

$$p(\omega_\mathrm{i}) = \frac{f_r(\omega_\mathrm{i} \to \omega_\mathrm{o}) \cdot \cos\theta_\mathrm{i}}{\boxed{\int_{H(\mathbf{x})} f_r(\omega_\mathrm{i} \to \omega_\mathrm{o}) \cdot \cos\theta_\mathrm{i} \, \mathrm{d}\omega_\mathrm{i}}}$$

Dieser Faktor ist nichts anderes als die Reflektanz ρ

UNIVERSITÄT BONN

BRDF IS in einem Pathtracer

Durchsatz-Update für eine allgemeine PDF:

...

```
thrput *= f_r(.)*dot(.)/(ϱ*p(ω_i))
```

Eine PDF, die genau proportional zu BRDF*cos ist, halt den Durchsatz konstant, weil sich die verschiedenen Terme auslöschen!

$$p(\omega_\mathrm{i}) = f_r(\omega_\mathrm{i} \to \omega_\mathrm{o}) \cdot \cos\theta_i \;/\; \rho$$

...

```
thrput *= 1
```

Physiker nennen dies die “analoge” Simulation, weil sich reale Teilchen so verhalten.

Direkte Beleuchtung in einem Pathtracer

Direkte Beleuchtung: Zwei Strategien

- An jedem Pfadvertex x, berechnen wir direkte Beleuchtung
    - d.h. Radianz reflektiert von Punkt x auf einer Oberfläche, die unmittelbar von den Quellen herrührt

- Zwei Samplingstrategien
    - Sampling proportional zur BRDF
    - Sampling der Lichtquellenfläche

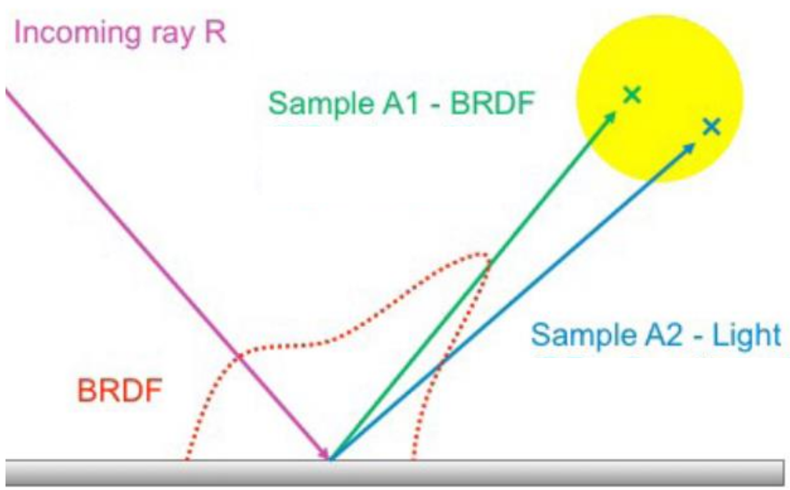
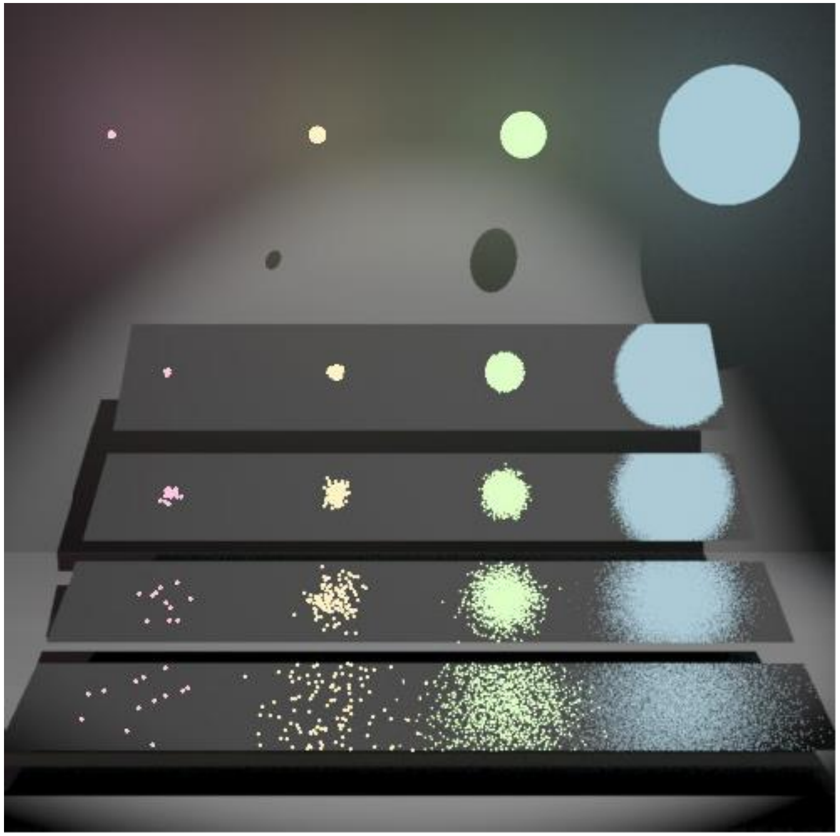
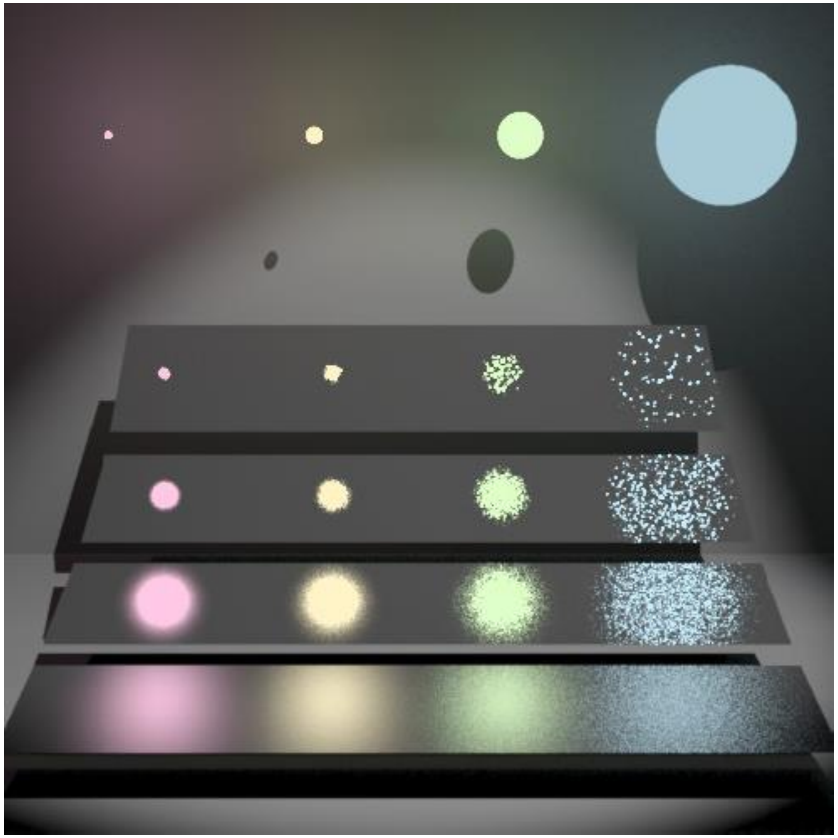


Image: Alexander Wilkie

Direkte Beleuchtung: Zwei Strategien



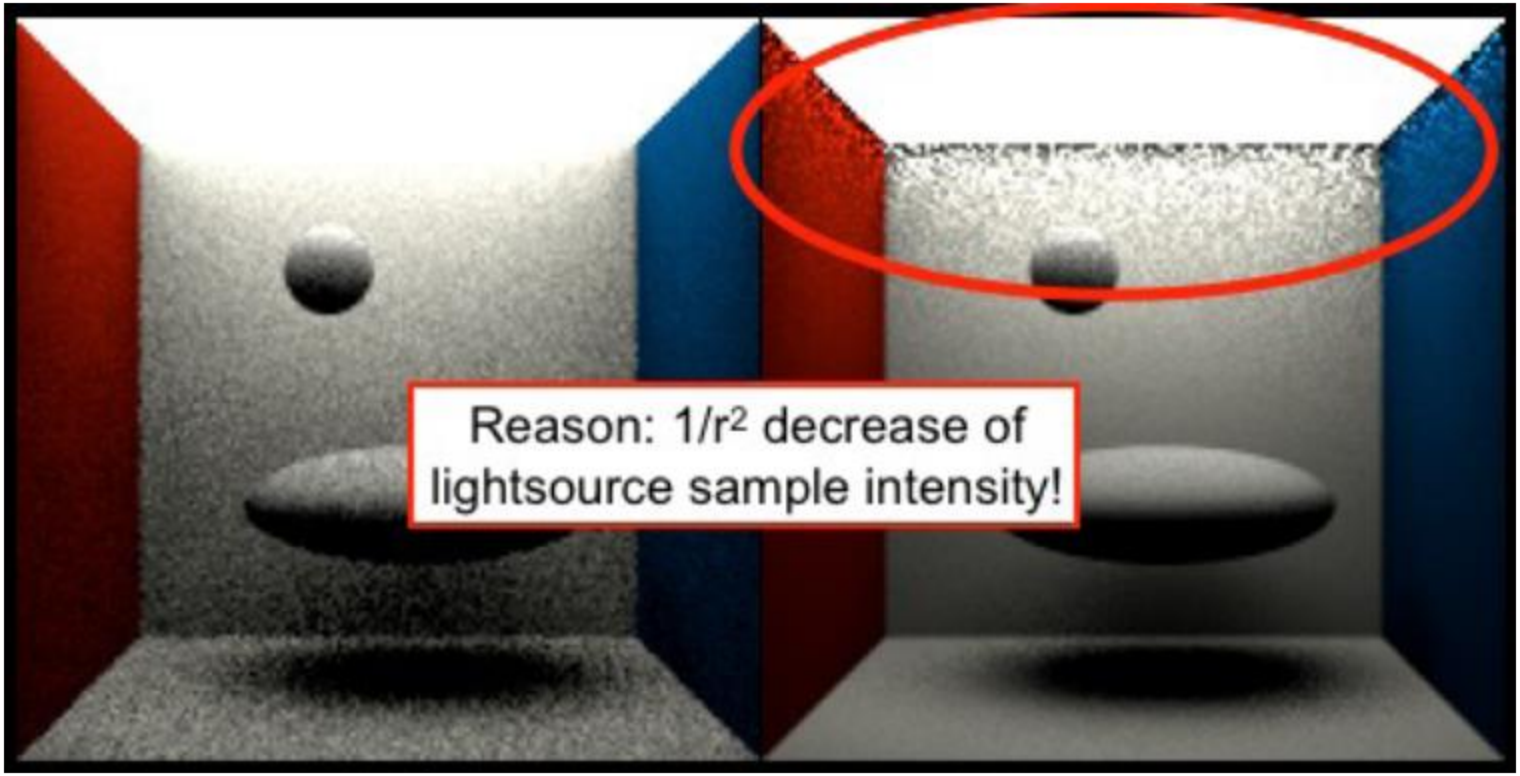
BRDF-proportionales Sampling



Lichtquellen-Sampling

Bilder von Eric Veach

Direktes Licht mit Multiple Importance Sampling

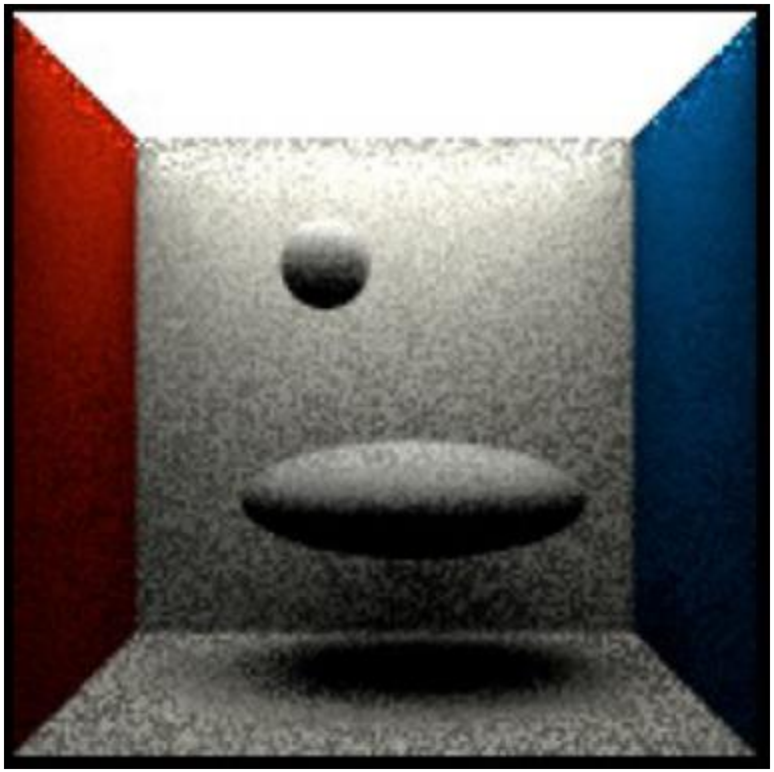


Bilder von Alexander Wilkie

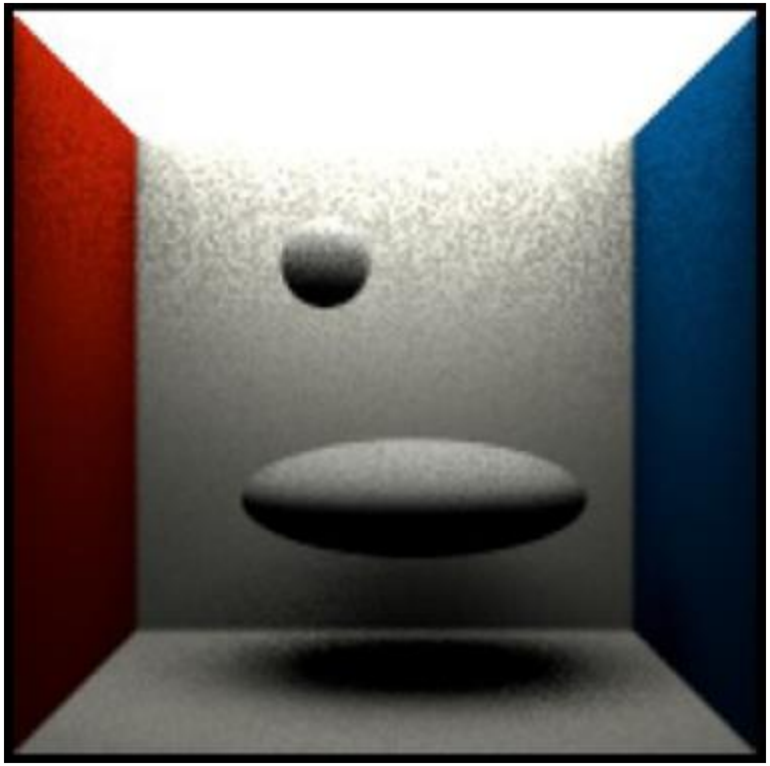
PDF p_1:
BRDF-Sampling

PDF p_2:
Lichtquellen-Sampling

Kombination



Arithmetisches Mittel
Erhält **schlechte** Eigenschaften beider Techniken



Balance heuristic
Bingo!!!

Bilder von Alexander Wilkie

MIS – Berechnung der Gewichte

Samplegewicht für BRDF-Sampling

$$w_1(\omega_j) = \frac{p_1(\omega_j)}{p_1(\omega_j) + p_2(\omega_j)}$$

PDF für BRDF-Sampling

PDF mit der die Richtung ω_j erzeugt worden wäre, wenn wir Lichtquellensampling verwendet hätten

- Im Grunde haben wir genau das schon am Anfang der heutigen Vorlesung gemacht – beim Sampeln der verschiedenen Terme der Phong-BRDF

PDFs

- BRDF-Sampling: $p_1(w)$
  - Hängt von der BRDF ab, also etwa für Lambertsche BRDF:

$$p_1(\omega) = \frac{\cos\theta_{\mathbf{x}}}{\pi}$$

- Lichtquellen-Flächensampling: $p_2(w)$

$$p_2(\omega) = \frac{1}{|A|} \underline{\frac{\|\mathbf{x}-\mathbf{y}\|^2}{\cos\theta_{\mathbf{y}}}}$$

Wandle die uniforme PDF $1/|A|$ von Flächenmaß (dA) zu Raumwinkelmaß ($d\omega$)

Warum der Umwandlungsfaktor?

- PDFs (anders als gewöhnliche Funktionen) ändern sich bei einem Koordinatenwechsel. Im Allgemeinen muss immer erfüllt sein:

$$p(\omega)d\omega = p(\mathbf{y})dA$$

- Und so

$$p(\omega) = p(\mathbf{y})\underbrace{\frac{dA}{d\omega}}_{\text{Umwandlungsfaktor}}$$

The full picture

- Dissertation Eric Veach, Stanford 1997
  (2 Academy Awards)

- Kapitel 9 über Multiple Importance Sampling