Фотореалистичный рендеринг *(aka raytracing)*

Лекция 4: Variance reduction, importance sampling, cosine-weighted sampling, light surface sampling, multiple importance sampling

2024

Об оптимизации рендеринга

- Мы видели, что Монте-Карло интегрирование занимает очень много времени, если мы хотим получить качественную картинку
- Хочется научиться оптимизировать процесс
- Оптимизировать = получить картинку более высокого качества за то же время
- Два варианта:
    - Ускорять вычисление каждого луча (следующая лекция)
    - Увеличивать качество каждого луча (эта лекция)

Variance reduction

- Как увеличить качество луча?
- Мы используем рандомизированный алгоритм, цвет луча – случайная величина
- Близость цвета луча к искомому цвету определяется дисперсией этой величины
- $\Longrightarrow$ Надо уменьшать дисперсию!
- Этот подход называется *variance reduction*

Variance reduction

- Вспомним формулу Монте-Карло интегрирования:

$$I = \int\limits_{\Omega} f(x)dx \approx \frac{1}{N} \sum \frac{f(X_i)}{p(X_i)}$$

- Вычислим дисперсию одного семпла:

$$\operatorname{Var} \frac{f(X)}{p(X)} = \mathbb{E}\left[\frac{f(X)^2}{p(X)^2}\right] - \mathbb{E}\left[\frac{f(X)}{p(X)}\right]^2 =$$
$$= \int\limits_{\Omega} \frac{f(x)^2}{p(x)^2} p(x)dx - I^2 = \int\limits_{\Omega} \frac{f(x)^2}{p(x)} dx - I^2$$

Variance reduction

- Вычислим дисперсию одного семпла:

$$\operatorname{Var}\frac{f(X)}{p(X)} = \int\limits_{\Omega} \frac{f(x)^2}{p(x)} dx - I^2$$

- Можно показать (напр. вариационным методом), что дисперсия минимизируется если

$$p^*(x) \sim |f(x)| \Longrightarrow p^*(x) = \frac{|f(x)|}{\int |f(x)| dx}$$

- $\Longrightarrow$ Поиск идеального распределения семплов сводится к интегрированию $\pm$ той же функции

Variance reduction

- В частности, если $f \geq 0$, то

$$p^*(x) = \frac{f(x)}{\int f(x)dx} = \frac{f(x)}{I}$$

- Тогда любой семпл даст точное значение интеграла:

$$\frac{f(x)}{p^*(x)} = f(x) \cdot \left[\frac{f(x)}{I}\right]^{-1} = I$$

- При этом дисперсия равна нулю:

$$\operatorname{Var} \frac{f(X)}{p^*(X)} = \int\limits_{\Omega} \frac{f(x)^2}{p^*(x)} dx - I^2 = \int\limits_{\Omega} I \cdot f(x)dx - I^2 = I^2 - I^2 = 0$$

Variance reduction

- N.B.: Если f бывает как положительна, так и отрицательна, то даже с идеальным распределением p^* дисперсия будет отличаться от нуля:

$$\mathrm{Var}\,\frac{f(X)}{p^*(X)} = \int\limits_{\Omega} \frac{f(x)^2}{|f(x)|} \cdot I_{abs} dx - I^2 = I_{abs}^2 - I^2$$

$$I_{abs} = \int\limits_{\Omega} |f(x)| dx$$

Importance sampling

- Итак, идеальное распределение для Монте-Карло интегрирования пропорционально искомой функции
- Вычисление такого распределения сводится к искомому интегралу, и само по себе нам не помогает
- Мысль: давайте брать распределение, в какой-то мере *повторяющее форму* нашей искомой функции
- Этот подход называется *importance sampling*
- Интуитивно: полезнее брать больше семплов там, где и функция больше, иначе мы делаем бесполезную работу

Importance sampling

- Рассмотрим частный случай, когда функция $A \to \mathbb{R}$ равна нулю вне некоторого подмножества $B \subset A$
- Возьмём равномерные распределения $p_A(x) = \frac{1}{|A|}$ и $p_B(x) = \frac{1}{|B|}$ на множествах A и B соответственно
- Тогда разность дисперсий равна

$$\operatorname{Var} \frac{f(X)}{p_A(X)} - \operatorname{Var} \frac{f(X)}{p_B(X)} = \int\limits_B f(x)^2 \left(\frac{1}{p_A(x)} - \frac{1}{p_B(x)} \right) dx =$$

$$= (|A| - |B|) \int\limits_B f(x)^2 dx$$

Importance sampling

- Тогда разность дисперсий равна

$$\operatorname{Var}\frac{f(X)}{p_A(X)} - \operatorname{Var}\frac{f(X)}{p_B(X)} = (|A| - |B|)\int\limits_B f(x)^2 dx$$

- $\implies$ Чем ближе носитель распределения к носителю функции, тем меньше дисперсия

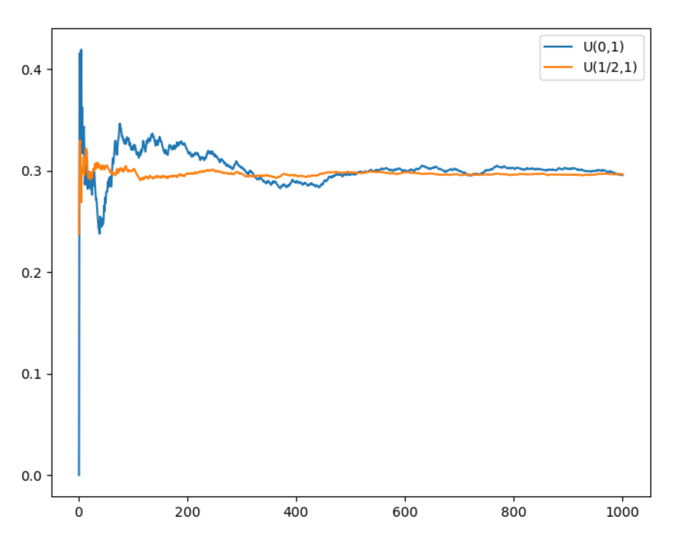
Importance sampling: пример №1

- Пусть мы интегрируем функцию $f(x) = x^2 \cdot \mathbf{1}_{[1/2,1]}(x)$ на отрезке $[0, 1]$:

$$\int\limits_0^1 f(x)dx = \int\limits_{\frac{1}{2}}^1 x^2 dx = \frac{7}{24} \approx 0.2916$$

- Возьмём два распределения: равномерное на $[0, 1]$ и на $[1/2.1]$

Importance sampling: пример №1



Importance sampling: пример №2

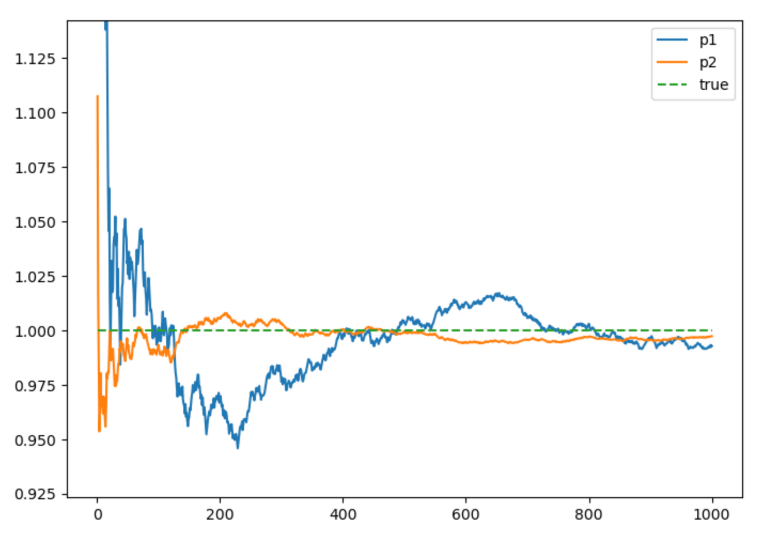
- Пусть мы интегрируем функцию $f(x) = \sin x$ на отрезке $[0, \frac{\pi}{2}]$:

$$\int\limits_0^{\frac{\pi}{2}} \sin x dx = 1$$

- Возьмём два распределения: равномерное на $[0, \frac{\pi}{2}]$ и линейно возрастающее на том же отрезке:

$$p_1(x) = \frac{2}{\pi}$$
$$p_2(x) = x \cdot \frac{8}{\pi^2}$$

Importance sampling: пример №2



Importance sampling в вычислении освещённости

- Наша интегрируемая функция выглядит как

$$L_{in}(\omega_i) \cdot f(\omega_i, \omega_o) \cdot (\omega_i \cdot n)$$

- $L_{in}(\omega_i)$ зависит от геометрии сцены и освещённости других объектов
- $f(\omega_i, \omega_o) \cdot (\omega_i \cdot n)$ зависит от материала объекта
- В общем случае поиск подходящего распределения для произведения двух функций – очень сложная задача
- $\implies$ Имеет смысл подобрать распределения для двух множителей по отдельности, и потом скомбинировать

Cosine-weighted distribution

- Для ламбертовой поверхности $f(\omega_i, \omega_o) = \text{const}$, и нам достаточно подобрать распределение на полусфере, пропорциональное $\omega_i \cdot n = \cos\theta$
- Такое распределение называется *cosine-weighted distribution* и идеально подходит для диффузных поверхностей
- Есть разные способы его получить:
    - CDF inversion
    - Проекция с диска
    - Сдвиг единичной сферы

Cosine-weighted: CDF inversion

- Берём два равномерных $U_1, U_2 \sim U(0,1)$
- Положим

$$\Theta = \cos^{-1}(\sqrt{U_1})$$
$$\Phi = 2\pi U_2$$

- Тогда (Θ, Φ) – сферические координаты точки на полусфере с cosine-weighted распределением (в направлении Z)

Cosine-weighted: проекция с диска

- Генерируем точку (X, Y) в единичном диске с равномерным распределением
- Положим $Z = \sqrt{1 - X^2 - Y^2}$
- Тогда (X, Y, Z) – евклидовы координаты точки на полусфере с cosine-weighted распределением (в направлении Z)

Cosine-weighted: сдвиг единичной сферы

- Генерируем точку $V = (X, Y, Z)$ на единичной сфере с равномерным распределением
- Положим $V' = \frac{V+N}{||V+N||}$
- Тогда V' – евклидовы координаты точки на полусфере с cosine-weighted распределением в направлении N

Cosine-weighted distribution

- Первые два способа требуют конвертации из сферических в евклидовы координаты в локальной системе координат поверхности
- Третий способ использует только вектор нормали, без необходимости восстанавливать локальный базис
- Как обычно, можете использовать любой способ

Cosine-weighted importance sampling

- Так как $\int\limits_{\Omega(n)} (\omega \cdot n) d\omega = \pi$, плотность вероятности cosine-weighted распределения равна

$$p(\omega) = \max\left(0, \frac{\omega \cdot n}{\pi}\right)$$

- Подставим в формулу Монте-Карло интегрирования диффузной поверхности:

$$E + \frac{C}{\pi} \cdot \frac{L_{in}(\omega) \cdot (\omega \cdot n)}{p(\omega)} = E + \frac{C}{\pi} \cdot L_{in}(\omega) \cdot (\omega \cdot n) \cdot \frac{\pi}{\omega \cdot n} =$$
$$= E + C \cdot L_{in}(\omega)$$

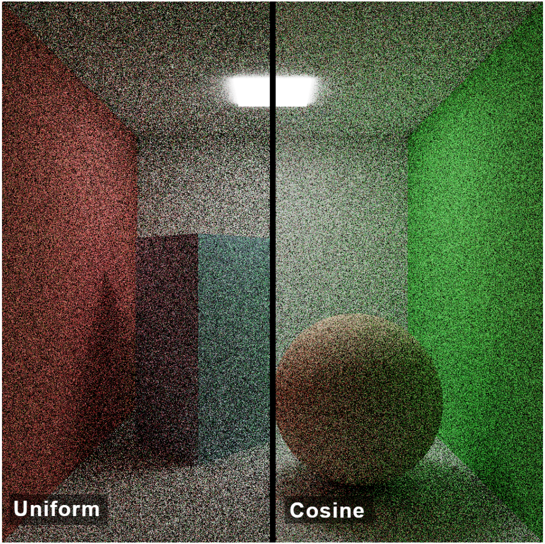
Cosine-weighted importance sampling

- Формула Монте-Карло интегрирования диффузной поверхности с cosine-weighted распределением семплов:

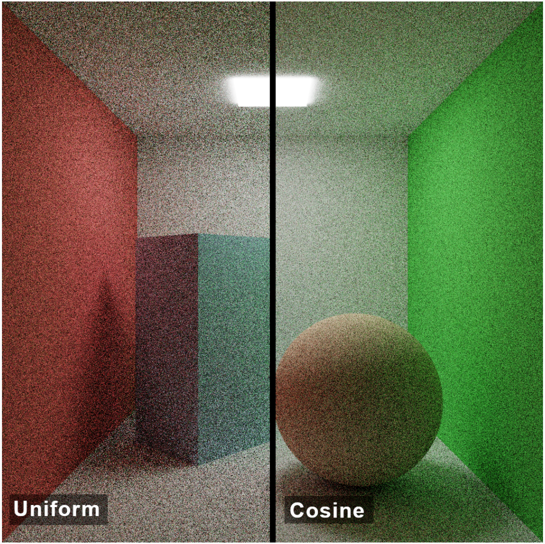
$$E + C \cdot L_{in}(\omega)$$

- N.B.: π в нормализации цвета сократилась с π в плотности вероятности
- N.B.: Множителя 2 нет, так как плотность теперь пропорциональна $\frac{1}{\pi}$, а не $\frac{1}{2\pi}$
- N.B.: Косинус угла $\omega \cdot n$ в интегрируемой функции сократился с косинусом из плотности вероятности (это ожидаемо – мы специально подбирали плотность пропорциональную этому множителю)

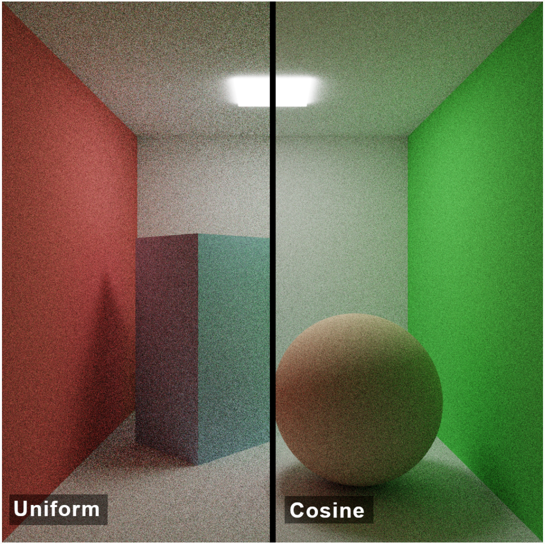
Cosine-weighted importance sampling: 64 семпла



Cosine-weighted importance sampling: 256 семплов



Cosine-weighted importance sampling: 1024 семпла



Incoming radiance sampling

- Cosine-weighted распределение уменьшает дисперсию, но картинка всё ещё достаточно шумная
- В искомой интегрируемой функции есть ещё один множитель: входящее количество света $L_{in}(\omega_i)$
- К сожалению, явный вид этой функции безумно сложен и зависит от всей сцены
- …но мы может попробовать какие-нибудь эвристики!

Direct light sampling

- Свет, отраженный объектами, обычно гораздо слабее света, напрямую излучённого источниками света
- $\Longrightarrow$ Давайте семплить направления на источники света!
- Монте-Карло аналог *теневого луча* из Whitted-style raytracing'а

Direct light sampling

- Нам нужно придумать какое-то распределение направлений на источник света
- Пытаться равномерно семплить сами направления – плохая идея: проекция объекта на сферу направлений может быть очень сложно устроена
- Другой вариант – семплить точки на поверхности источника света (например, равномерно) и превращать их в семплы направлений

Равномерное распределение на плоскости

- Как сгенерировать равномерное распределение на поверхности бесконечной плоскости? Никак
- Можно взять 2D нормальное распределение с центром в точке на плоскости, ближайшей к освещаемой точке
- Можно придумать какое-нибудь распределение с учётом множителя $\frac{1}{r^2}$, который мы увидим чуть позже
- Мы не будем учитывать плоскости, излучающие свет, в importance sampling'е (но свет они всё равно излучают!)

Равномерное распределение на параллелепипеде

- Разные грани параллелепипеда имеют разную площадь, вероятность выбрать точку на грани должна быть пропорциональна её площади
- В наших терминах, у параллелепипеда с параметрами (S_X, S_Y, S_Z) рёбра имеют длины $(2S_X, 2S_Y, 2S_Z)$, соответственно
- Площади граней – $(4S_YS_Z, 4S_XS_Z, 4S_XS_Y)$ – это их 'веса' (W_X, W_Y, W_Z)
- Сумма весов – $W = W_X + W_Y + W_Z = 4(S_YS_Z + S_XS_Z + S_XS_Y)$
- Можно сгенерировать число в $U(0, W)$ и сравнивать с весами граней, а можно генерировать число в $U(0, 1)$ и отнормировать веса на W – это примерно одно и то же

Равномерное распределение на параллелепипеде

- Итак, сгенерируем равномерно случайное число $U \sim U(0, W)$
- Если $U < W_X$, выбираем грань, перпендикулярную оси X (т.е. грань YZ)
- Если $W_X \leq U < W_X + W_Y$, выбираем грань, перпендикулярную оси Y (т.е. грань XZ)
- Иначе выбираем грань, перпендикулярную оси Z (т.е. грань XY)
- У каждой грани есть противоположная грань с такой же площадью: подбросим монетку, и с вероятностью 50% выберем одну из двух граней
- Например, если мы выбрали грань -Y, то точку на грани можно сгенерировать как $(U_X S_X, -S_Y, U_Z S_Z)$, где $U_X, U_Z \sim U(-1, 1)$

Равномерное распределение на эллипсоиде

- Равномерное распределение на эллипсоиде построить очень тяжело: например, его плотность, зависящая от площади поверхности эллипсоида, не выражается в элементарных функциях (нужны т.н. *эллиптические интегралы*)
- Мы можем взять неравномерное распределение, которое легко получить: сгенерировать точку (X, Y, Z) равномерно на единичной сфере, и отмасштабируем до размеров эллипсоида $(R_X X, R_Y Y, R_Z Z)$
- Нам всё ещё нужно посчитать плотность такого распределения!

Преобразование распределения

- Рассмотрим общий случай: есть некоторое распределение p_A на некой поверхности A
- Мы применяем к нему трёхмерное преобразование F, переводящее поверхность A в поверхность B и распределение p_A в распределение p_B
- Рассмотрим маленький кусочек $d\sigma_A$ поверхности A: на него приходится $\approx p_A(a) \cdot d\sigma_A$ вероятности
- Под действием преобразования он переводится в кусочек $d\sigma_B$ поверхности B, и на него приходится $\approx p_B(F(a)) \cdot d\sigma_B$ вероятности

Преобразование распределения

- Отсюда

$$p_B(F(a))d\sigma_B = p_A(a)d\sigma_A$$
$$p_B(F(a)) = p_A(a)\frac{d\sigma_A}{d\sigma_B}$$

- То есть, изменение плотности вероятности *обратно пропорционально* локальному изменению площади поверхности

Преобразование распределения

- Локальное изменение площади зависит от *якобиана* J_F нашего преобразования, и зависит от локальной ориентации поверхности
- Например, растягивание по оси X в 10 раз никак не меняет кусок поверхности, параллельный плоскости YZ
- Формально, преобразование плоскости сводится к *внешнему квадрату* якобиана $\Lambda^2 J_F$, т.е. к главным минорам 2×2 матрицы якобиана
- В 3D есть более удобная формула

Преобразование распределения

- Рассмотрим маленький кубик, ‘лежащий‘ на нашей поверхности: одна его грань параллельна поверхности, и одно его ребро параллельно нормали к поверхности
- Мы знаем, что под действием J_F кубик превратится в некий параллелепипед (в общем случае не прямоугольный!), его объём увеличится в $\det J_F$ раз
- Мы знаем, что параллельная поверхности грань перейдёт в грань, параллельную преобразованной поверхности; нас интересует, как изменилась площадь этой грани
- Объём параллелепипеда это площадь грани, умноженная на высоту
- Высота это проекция преобразованной нормали на новую нормаль

Преобразование распределения

- Преобразованная нормаль: $J_F \cdot N$
- Новая нормаль: $\frac{J_F^{-T} \cdot N}{\|J_F^{-T} \cdot N\|}$
- Проекция: $(J_F \cdot N) \cdot \left(\frac{J_F^{-T} \cdot N}{\|J_F^{-T} \cdot N\|} \right)$
- Площадь грани:

$$\det J_F \cdot \frac{\|J_F^{-T} \cdot N\|}{(J_F \cdot N) \cdot (J_F^{-T} \cdot N)} =$$
$$= \det J_F \cdot \frac{\|J_F^{-T} \cdot N\|}{(J_F^{-1} \cdot J_F \cdot N) \cdot N} =$$
$$= \det J_F \cdot \frac{\|J_F^{-T} \cdot N\|}{N \cdot N} = \det J_F \cdot \|J_F^{-T} \cdot N\|$$

Преобразование распределения

- Итак, зная преобразование поверхности и нормаль к ней, можно вычислить локальное изменение площади как

$$\det J_F \cdot \|J_F^{-T} \cdot N\|$$

- Тогда, изменение плотности вероятности будет равно

$$p_B(F(a)) = \frac{p_A(a)}{\det J_F(a) \cdot \|J_F^{-T}(a) \cdot N(a)\|}$$

Распределение на эллипсоиде

- Эллипсоид получен из единичной сферы линейным преобразованием

$$F = \begin{pmatrix} R_X & 0 & 0 \\ 0 & R_Y & 0 \\ 0 & 0 & R_Z \end{pmatrix}$$

- Оно линейно, поэтому якобиан совпадаем с самим преобразованием
- Определитель: $\det F = R_X R_Y R_Z$

$$\|J_F^{-T} \cdot N\| = \left\| \left(\frac{N_X}{R_X}, \frac{N_Y}{R_Y}, \frac{N_Z}{R_Z} \right) \right\| =$$
$$= \sqrt{\left(\frac{N_X}{R_X}\right)^2 + \left(\frac{N_Y}{R_Y}\right)^2 + \left(\frac{N_Z}{R_Z}\right)^2}$$

Распределение на эллипсоиде

- Преобразование площади:

$$\det J_F \cdot \|J_F^{-T} \cdot N\| = \sqrt{N_X^2 R_Y^2 R_Z^2 + R_X^2 N_Y^2 R_Z^2 + R_X^2 R_Y^2 N_Z^2}$$

- Итого плотность точки на эллипсоиде, полученной из равномерного распределения на единичной сфере:

$$p(R_X N_X, R_Y N_Y, R_Z N_Z) = \frac{1}{4\pi\sqrt{N_X^2 R_Y^2 R_Z^2 + R_X^2 N_Y^2 R_Z^2 + R_X^2 R_Y^2 N_Z^2}}$$

Распределение на эллипсоиде

- Преобразование площади:

$$\det J_F \cdot \|J_F^{-T} \cdot N\| = \sqrt{N_X^2 R_Y^2 R_Z^2 + R_X^2 N_Y^2 R_Z^2 + R_X^2 R_Y^2 N_Z^2}$$

- Итого плотность точки на эллипсоиде, полученной из равномерного распределения на единичной сфере:

$$p(R_X N_X, R_Y N_Y, R_Z N_Z) = \frac{1}{4\pi \sqrt{N_X^2 R_Y^2 R_Z^2 + R_X^2 N_Y^2 R_Z^2 + R_X^2 R_Y^2 N_Z^2}}$$

Распределение направлений

- Мы научились генерировать точки на поверхности источника света и вычислять их плотность вероятности
- Для интегрирования нам нужны направления на эти точки и их плотность вероятности
- Само направление получить просто: это нормированный вектор из освещаемой точки в случайную точку на поверхности источника света
- Вопрос в том, как получить плотность вероятности

Area formulation of light transport

- Мы писали уравнение рендеринга как интеграл по всем входным направлениям
- Вместо этого можно написать интеграл по всем точкам всех поверхностей сцены, и учесть количество света, приходящее из этих точек
- Вывод смотри в здесь и здесь

Area formulation of light transport

$$L_{out}(x,\omega_o) = L_e(x,\omega_o) + \int\limits_y f(x,\omega_i,\omega_o) \cdot L_{out}(y,-\omega_i) \cdot V(x,y) \cdot \frac{|\cos\theta_x| \cdot |\cos\theta_y|}{\|x-y\|^2} dy$$

- x – освещаемая точка
- y – точка, из которой свет идёт в освещаемую точку
- $\omega_i = \frac{y-x}{\|y-x\|}$ – направление из x в y
- $V(x,y)$ – *visibility term*: 1, если луч из x в y ничем не заблокирован, иначе 0
- $\theta_x = \omega_i \cdot n_x$ – косинус угла между лучом и нормалью в точке x
- $\theta_y = (-\omega_i) \cdot n_y$ – косинус угла между лучом и нормалью в точке y

Распределение направлений

- От этого уравнения нам нужен только множитель $\frac{|\cos\theta_y|}{\|x-y\|^2}$: он описывает то, какой телесный угол в точке x соответствует элементу площади некой поверхности вокруг в точке y
- Плотность вероятности меняется *обратно пропорционально* изменению площади
- Если мы сгенерировали семпл y на поверхности источника света с вероятностью $p(y)$ и нормалью n_y, то плотность вероятности соответствующего угла равна

$$p(\omega) = p(y) \cdot \frac{\|x-y\|^2}{|\omega \cdot n_y|}$$

$$\omega = \frac{y-x}{\|y-x\|}$$

Распределение направлений

- Обычно наши источники света – двусторонние (параллелепипед, эллипсоид)
- Одному направлению соответствуют *две точки* на поверхности источника света
- $\Longrightarrow$ Чтобы правильно вычислить вероятность направления, нужно *сложить* вероятности двух соответствующих точек на поверхности света
- Для этого можно или модифицировать функцию пересечения, чтобы она возвращала оба пересечения, или просто послать луч второй раз 'внутри' источника от точки первого пересечения
- N.B.: Позже мы перейдём к только треугольникам, и эта проблема уйдёт

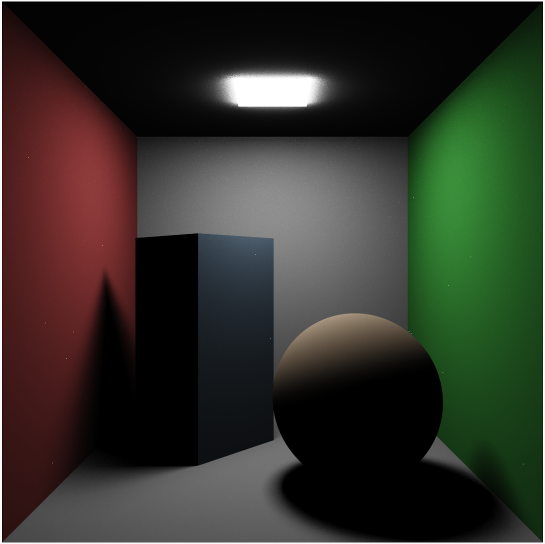
Light surface sampling: алгоритм

- Итак, если мы хотим в Монте-Карло интегрировании сэмплить некоторый источник света, нужно:
- 1. Сгенерировать точку на поверхности источника с каким-то распределением вероятности
- 2. По точке получить направление ω из освещаемой точки в точку на источнике света
- 3. Вычислить, какие две точки на поверхности источника соответствуют этому направлению (пересекать со всей сценой здесь **не нужно**, только с одним объектом – источником света)
- 4. Вычислить плотность вероятности этих точек согласно выбранному распределению
- 5. Вычислить плотность вероятности направления как сумма двух спроецированных вероятностей (формула с косинусом и расстоянием, два раза)
- 6. Наш семпл готов, вы восхитительны!

Light surface sampling

- N.B.: Использовать *только* семплинг источника света не получится – картинка будет неправильной, так как распределение направлений не покрывает носитель интегрируемой функции

Light surface sampling



Multiple importance sampling (MIS)

- Итак, мы придумали два распределения, которые могут уменьшить variance изображения:
  - Cosine-weighted распределение идеально подходит для диффузной поверхности
  - Light surface sampling направляет лучи в сторону источников света и полезно, когда источники маленькие или далеко
- Хотелось бы их как-то скомбинировать!
- $\Longrightarrow$ *Multiple importance sampling (MIS)*

Multiple importance sampling (MIS)

- Есть много способов комбинировать распределения, подробнее смотри в статье `Veach, Guibas - Optimally Combining Sampling Techniques for Monte Carlo Rendering (1995)`
- Мы будем использовать простую эвристику: аффинную комбинацию распределений
- С вероятностью 50% возьмём cosine-weighted распределение
- С вероятностью 50% возьмём случайный источник света и случайный семпл на нём

Multiple importance sampling (MIS)

- Таким образом, пусть $p_{\cos}(\omega)$ – плотность вероятности cosine-weighted распределения, и $p_i(\omega)$ – вероятность семплинга i-ого из N источников света
- Тогда наше итоговое распределение:

$$p(\omega) = \frac{1}{2}p_{\cos}(\omega) + \frac{1}{2N}\sum p_i(\omega)$$

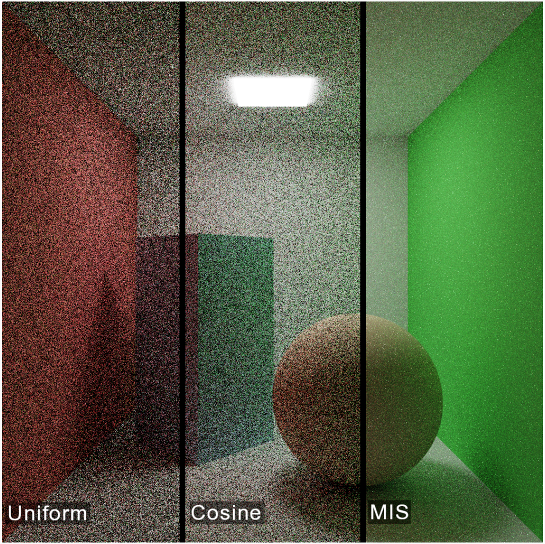
- В итоге мы сгенерируем семпл направления одним из выбранных распределений, но для Монте-Карло интегрирования нам нужна его вероятность, поэтому нам всё равно нужно вычислить вероятность этого семпла для *всех распределений*, а не только того, которым он сгенерирован!

MIS: алгоритм для диффузной поверхности

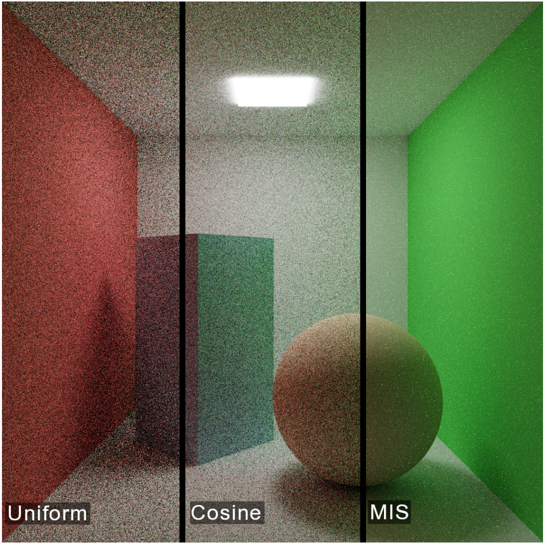
- 1. Бросаем монетку
- 2.1. С вероятностью 50% генерируем семпл направления ω cosine-weighted распределением
- 2.2. Либо с вероятностью 50% генерируем семпл направления ω в сторону случайного семпла на поверхности одного из источников света
- 3. Вычисляем итоговую вероятность этого семпла $p(\omega) = \frac{1}{2} p_{\mathrm{cos}}(\omega) + \frac{1}{2N} \sum p_i(\omega)$
- 3.1. Не забываем, что cosine-weighted вероятность равна нулю с другой стороны нормали
- 3.2. Не забываем, что источники света двухсторонние
- 4. Используем семпл в Монте-Карло интегрировании:

$$E + \frac{C}{\pi} \cdot L_{in}(\omega) \cdot (\omega \cdot n) \cdot \frac{1}{p(\omega)}$$

Uniform vs cosine vs MIS: 64 семпла



Uniform vs cosine vs MIS: 256 семпла



Uniform vs cosine vs MIS: 1024 семпла