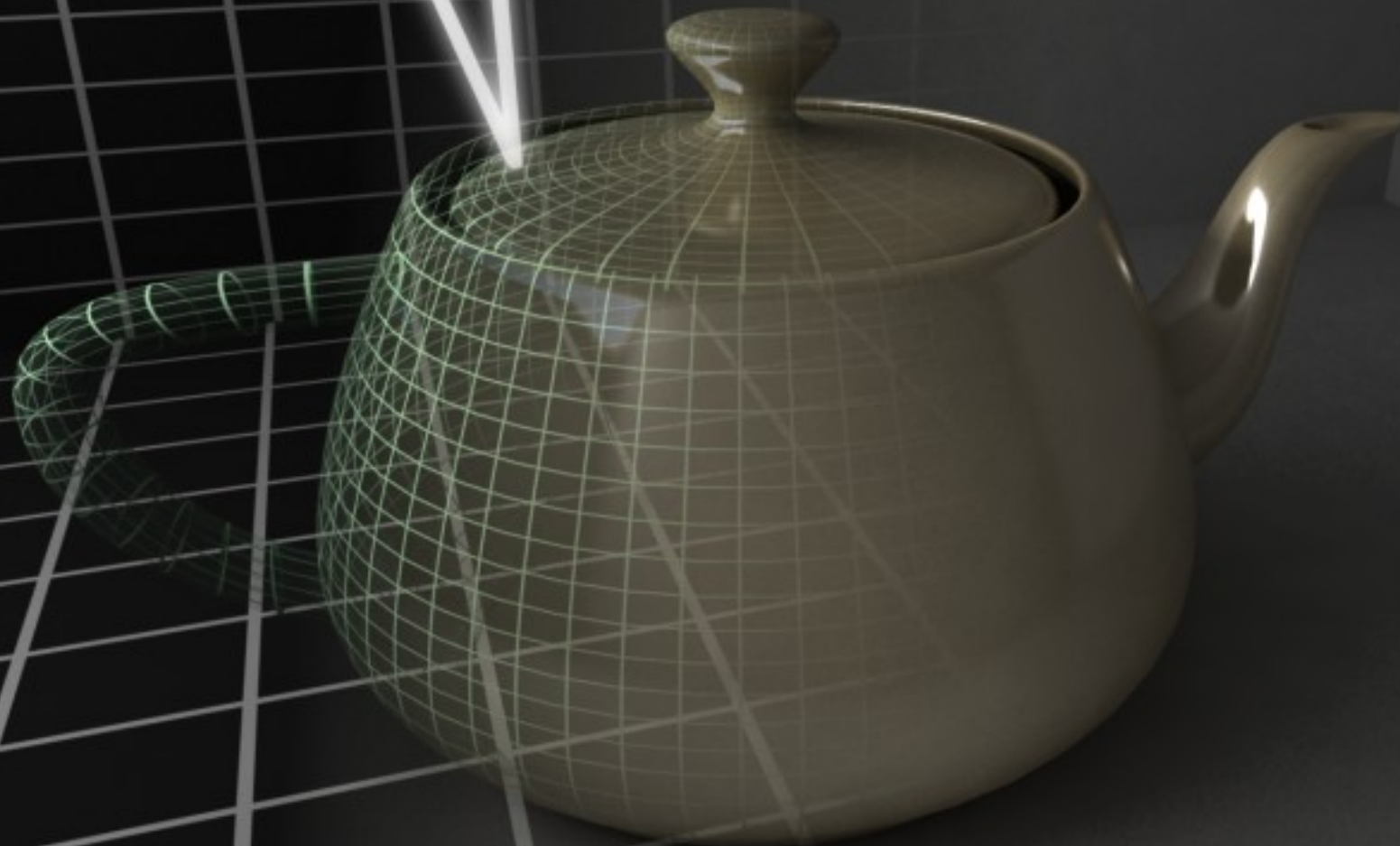


3D графика и трасиране на лъчи



<http://raytracing-bg.net/>

BofH





Тема 11

Стереоскопия

Глобално осветление

Фундаментална формула на Каджия

(транспортно уравнение на светлината)

Основни алгоритми за глобално осветление

Съдържание

- Анонси
- Стереоскопия
 - Видове
 - Реализация на анаглифни изображения
- Глобално осветление
 - BRDF-и
 - Фундаментална формула на Каджия (транспортно уравнение на светлината)

Съдържание (2)

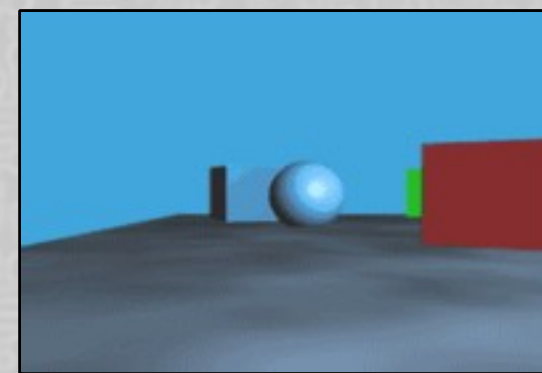
- Алгоритми за глобално осветление
 - Path tracing
 - Light tracing (накратко)
 - Bidirectional path tracing (накратко)

Анонси

- Следващата лекция е на 8-ми Януари
 - (: Весели празници :)
- Ще публикуваме курсови проекти от утре
 - Ще трябва да си изберете проект и да изпратите e-mail за да ви направим акаунти в SVN базата
 - Подробности – на сайта (от утре)

Стереоскопия

- Стереоскопията е техника, с която можем да придадем усещане за обем в някакво изображение, като показваме различни картинки на двете очи
 - По подобие на това, което се случва в реалния свят
 - Паралакс ефект – при две различни гледни точки, по-близките обекти изглеждат сякаш са се изместили повече от по-далечните



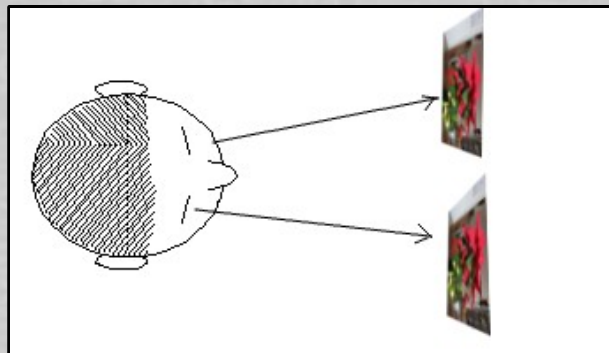
Стереоскопия

- Технически, стереоскопичните изображения са просто двойка кадри, единият от които е предназначен за лявото око, другият за дясното
- Разнообразие от начини за създаване на стереоскопични двойки:
 - Комбиниране на двойки снимки
 - Специални фотоапарати
 - Компютърно-генерирани изображения (тук се намесваме ние)



Стереоскопия

- Разглеждане на стереоскопични изображения
 - Още по-голямо разнообразие
 - Ще се спрем на част от методите тук
 - Основната идея на всички методи е: единият кадър да се вижда само от лявото око, другият – само от дясното



Двойка екрани (комплект)

- Представяват малки екрани, по един за всяко око
 - Недостатъци:
 - Изисква специализиран софтуер и хардуер
 - Относително скъпи
 - Ниска разделителна способност



LCD затъмнителни очила (shutter glasses)

- Пред всяко око има един LCD елемент, който може да закрие плътно гледката на окото
- CRT монитор показва редуващи се кадри (ляв-десен-ляв-десен-...). Затъмняването на очилата се редува по подобен начин и е синхронно с монитора.
- Така всяко око вижда само „полагащите“ му се кадри

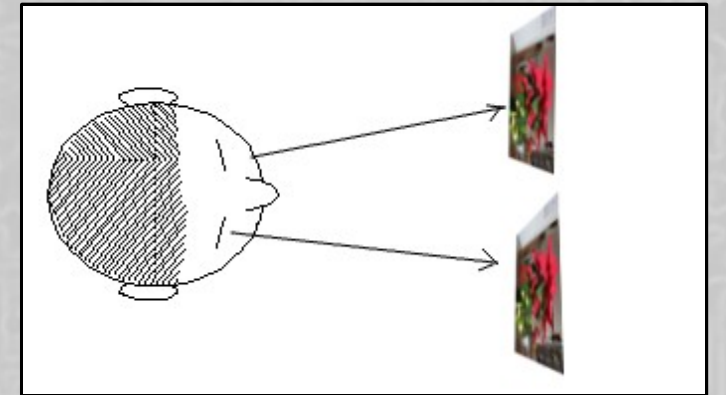


LCD затъмнителни очила (shutter glasses)

- Необходим е драйвер, който да реализира бързото сменяне на кадрите на екрана (на всяко опресняване) и да синхронизира затъмняванията на очилата
- Предимства
 - Относително евтино
 - Добро цвето предаване, добра разделителна способност
- Недостатъци
 - Трептене, монитора трябва да опреснява поне на 120 херца
 - Не работи с TFT монитори
 - Остатъчно изображение върху фосфора (ghosting)

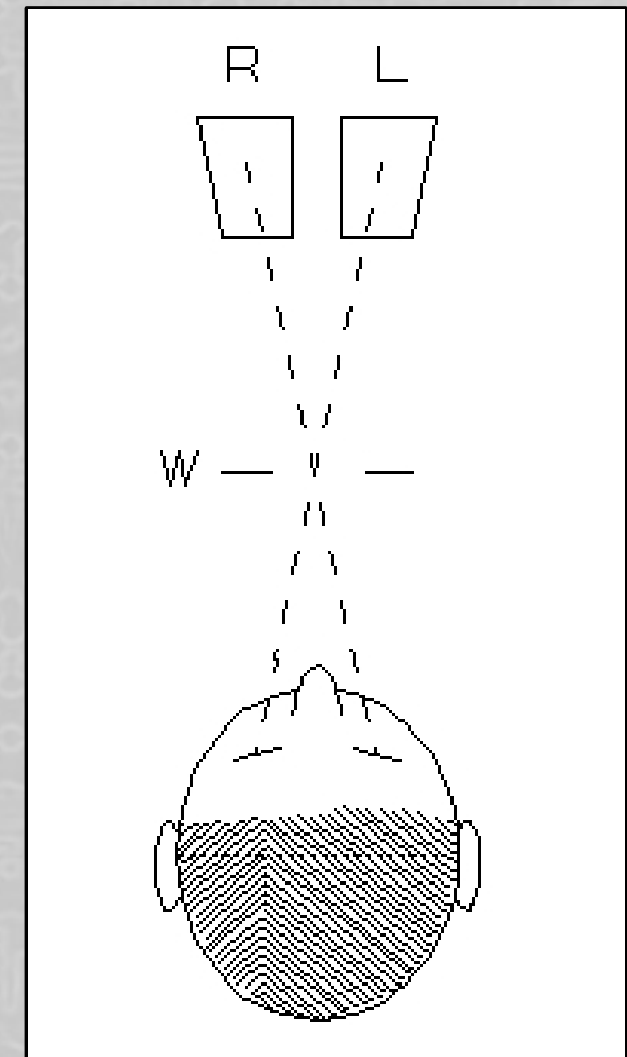
„Успореден“ метод

- Пред очите на човека се слагат две малки картинки на разстояние не повече от 65 мм една от друга
- Гледат се отблизо, очите трябва да гледат успоредно (в безкрайност), а фокусът трябва да е върху картинките
 - Това е трудната част; изисква малко тренировка



„Кръстосан“ метод

- Картинката за дясното око се поставя отляво, лявата – отдясно
- Погледът трябва да се кръстоса (сякаш гледаме нещо наблизо), а фокуса да е върху картинките (трудната част)
 - Човек трябва да „отдели“ функциите си за „следене“ и „фокусиране“ на очите
- По-добър метод от успоредния – позволява картинките да са много по-големи (и по-детайлни)



Примерна „кръстосана“ снимка



(картинките се различават много малко, но човешкото око усеща разликите много добре)

„Кръстосан“ метод

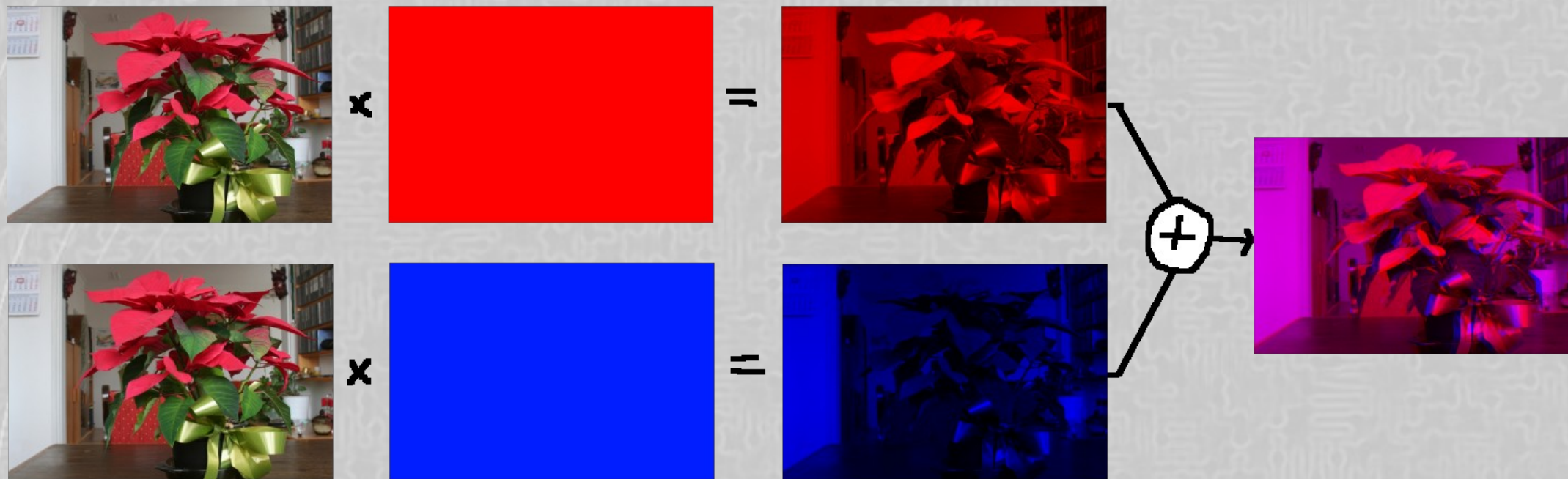
- Предимства
 - Не изисква никаква техника
 - Пълно качество: и откъм разделителна способност, и откъм цветове
- Недостатъци
 - Изисква научаване, което за някои хора е доста трудно
 - Кара мускулите на очите да се напрягат по необичайни за тях начини

Анаглифни очила

- Идеята: всеки от двата кадъра да мине предварително през цветен филтър (например, червен или син). Резултатите се смесват в един кадър. Разглеждането става чрез очила със същия цвят филтри. Всеки филтър пропуска само „съответстващия“ си кадър, и поглъща другия (светлината, минала веднъж през син филтър, не минава през червен).



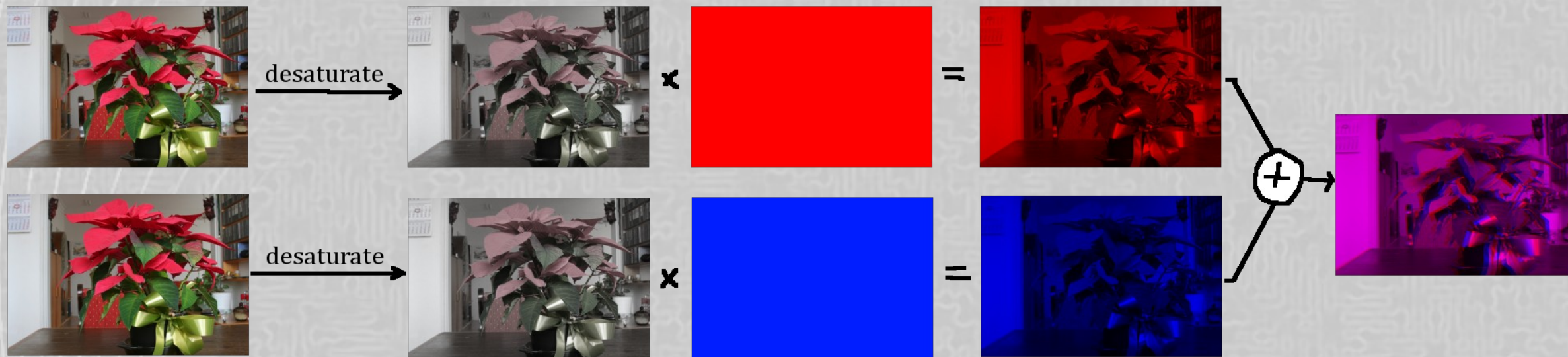
Анаглифни очила



При картинки с наситени цветове (като примерната), обаче, филтрирането отрязва доста информация (например, червените листа не се виждат в десния кадър)

Анаглифни очила

- Оказва се благоприятно да „убием“ малко цветовете на входните изображения (desaturation):

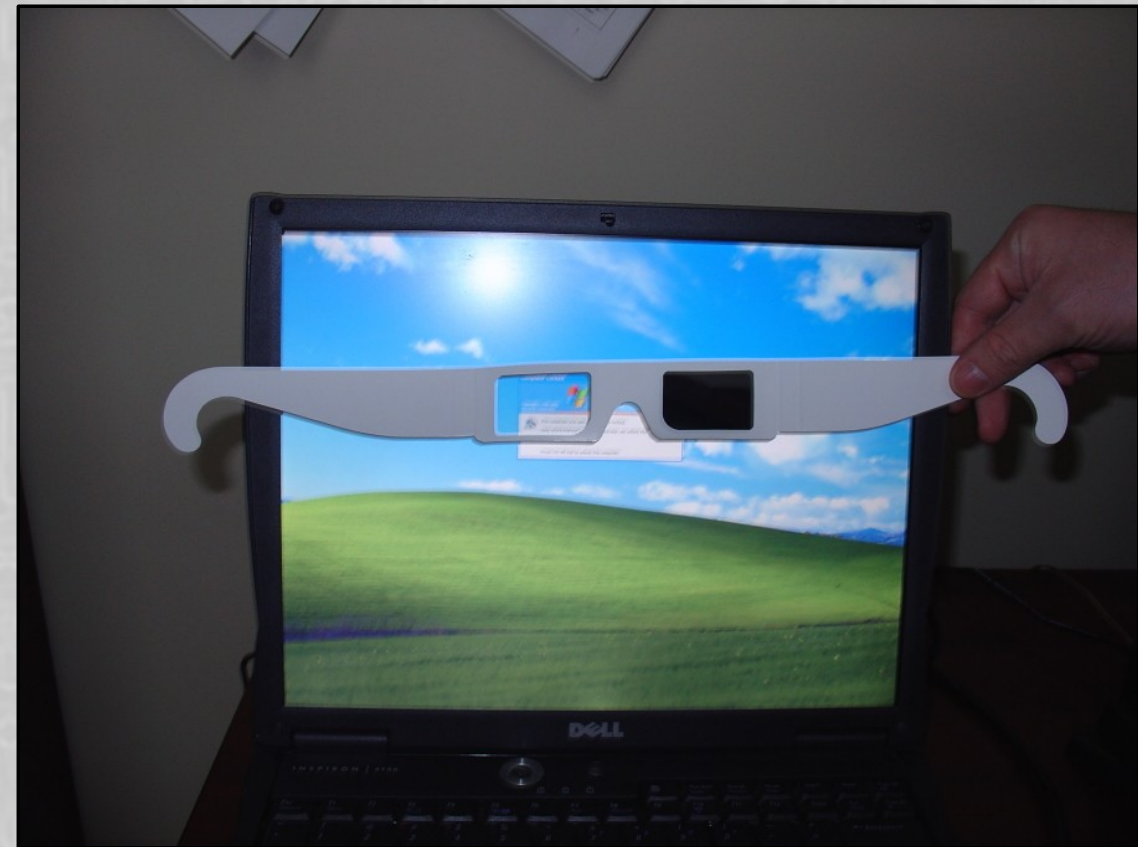


Анаглифни очила

- Освен червено-сини, често ползвани са червено-циан, жълто-циан, червено-зелени ...
- Недостатъци
 - Губи се много от усещането за цветовете. В повечето случаи, картинките изглеждат все едно са били черно-бели
 - Ако филтрите не са перфектни, може част от неправилния кадър да прозира (ghosting)

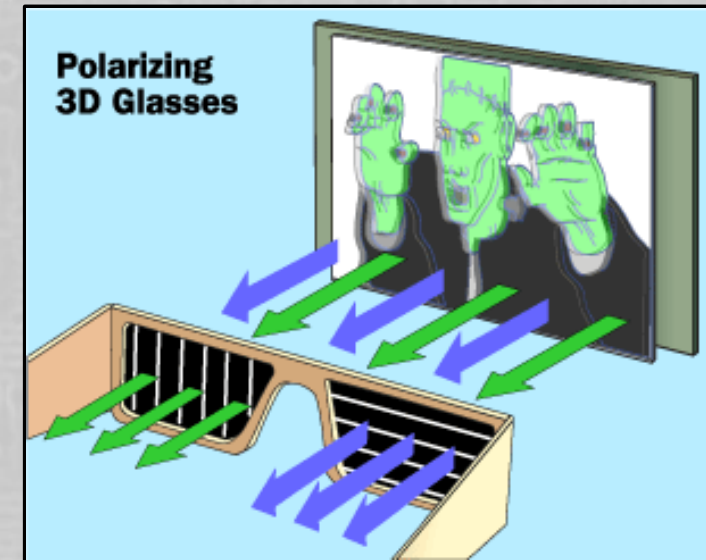
Очила с поляризационни филтри

- Използват се свойството „поляризация“ на светлината
 - Съществуват филтри, които пропускат светлина само с определена поляризация (например хоризонтална или вертикална)
 - Очилата са с два различни типа филтри пред всяко око



Очила с поляризационни филтри

- Кадърът се смесва от два прожектора, пред всеки от които също са поставени разнородни поляризирани филтри
- Тъй като всеки филтър блокира светлината с несъвпадаща поляризация, то лявото око вижда само светлината от „левия“ прожектор, а дясното – само от „десния“



Очила с поляризационни филтри

- Предимства
 - Не се губят цветовете или разделителна способност
- Недостатъци
 - Ghosting
 - Заради недобри филтри
 - Ако главата на човек не е точно изправена. Дори при леко завъртане на главата, филтрите почват да пропускат и от другия кадър
 - Изисква два проектора (не особено евтино)

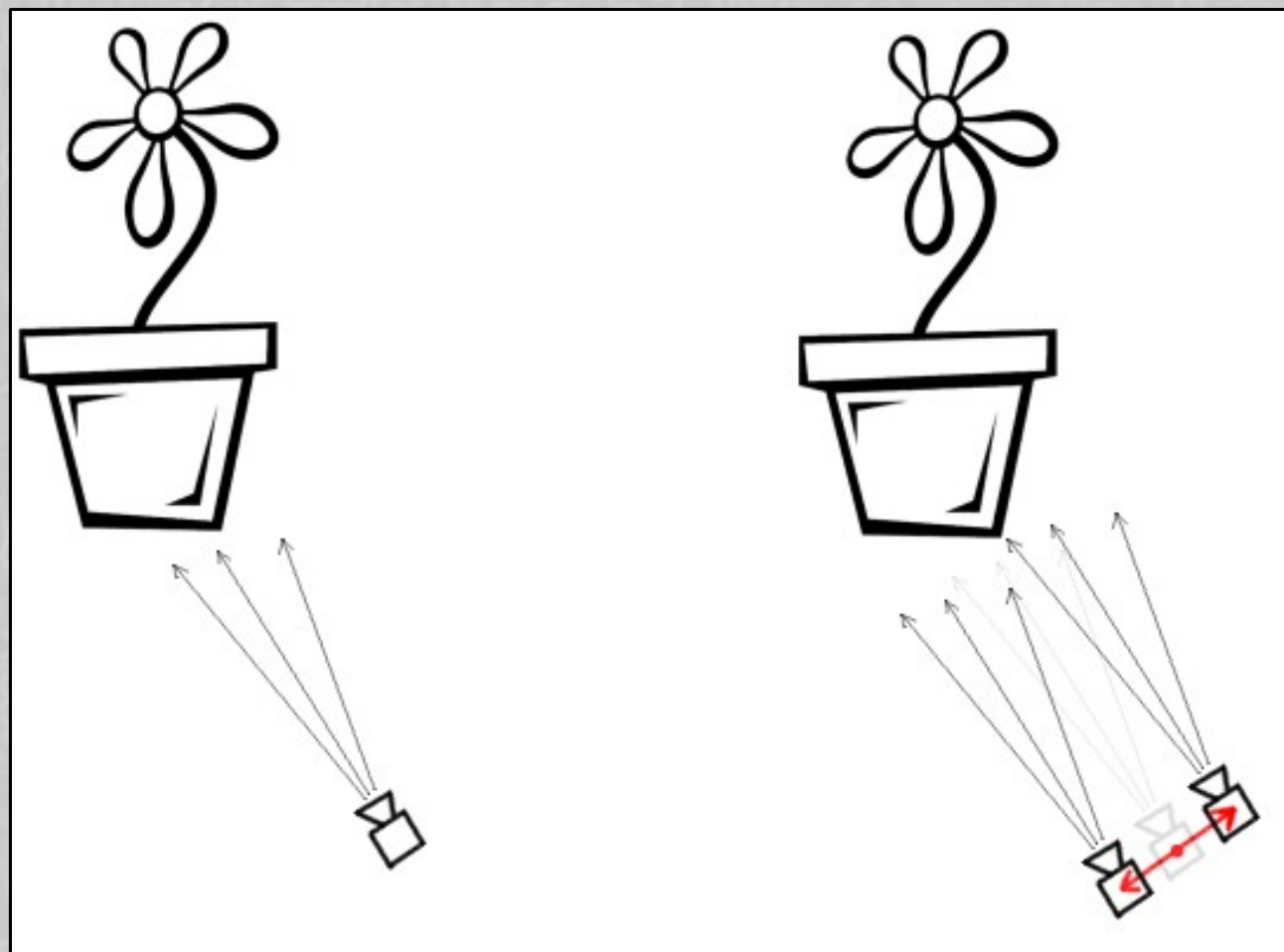
Спектрално-разделени очила

- Същата идея като при поляризиращите филтри, само че филтрите са спектрални – пропускат само различни (непресичащи се) части от спектъра
 - За разлика от поляризиращите филтри, няма проблеми при накланянето на главата
- Човек не усеща „нарязаността“ на спектъра; ако дразненето на R, G и B рецепторите му правилно, ще приеме изображението за пълноцветно

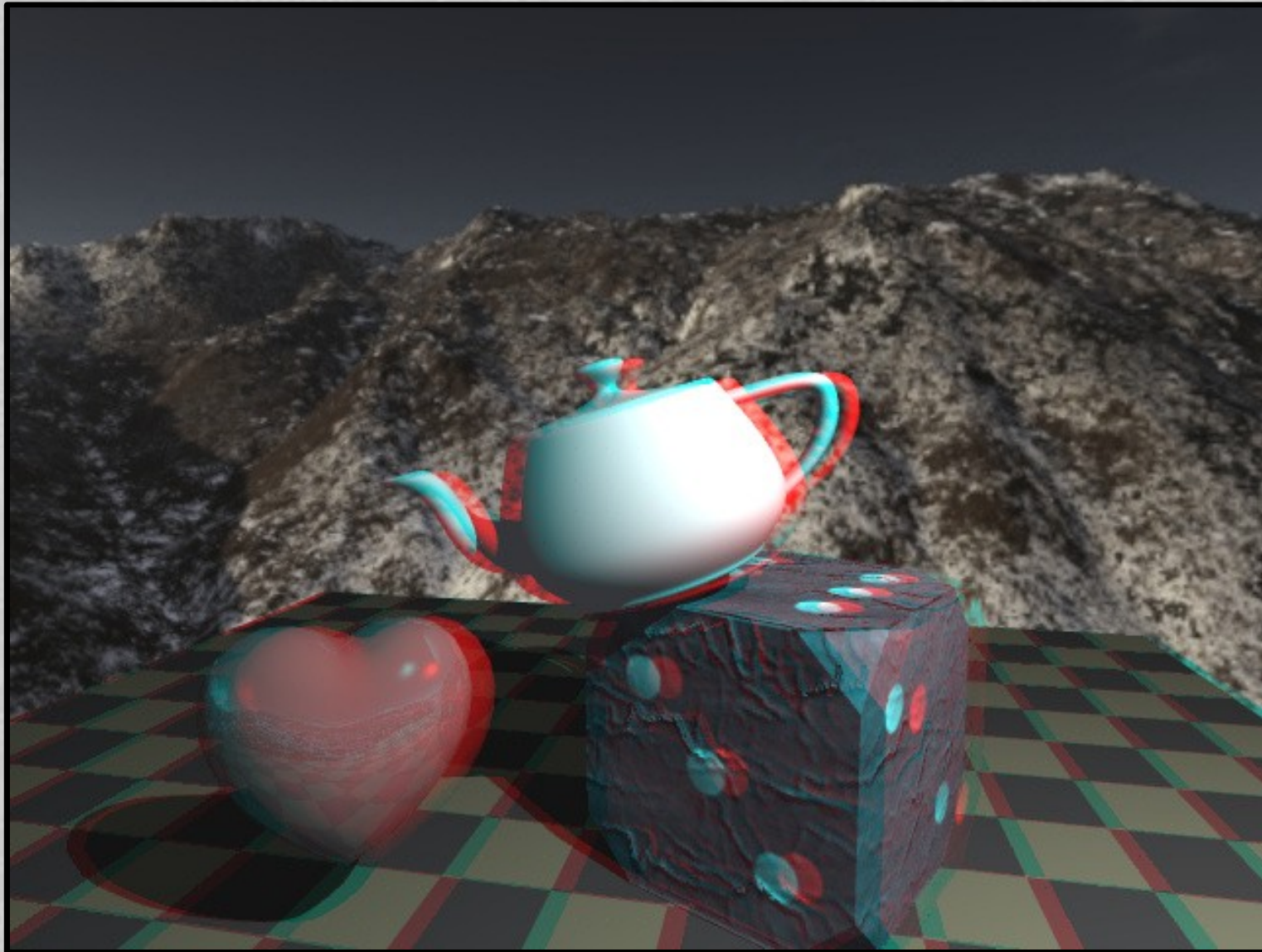


Стереоскопично рендериране

- Рендерираме два кадъра от две гледни точки, раздалечени по посоката `rightDir` на камерата
- За анаглифен образ, смесваме двата кадъра с подходящо маскиране

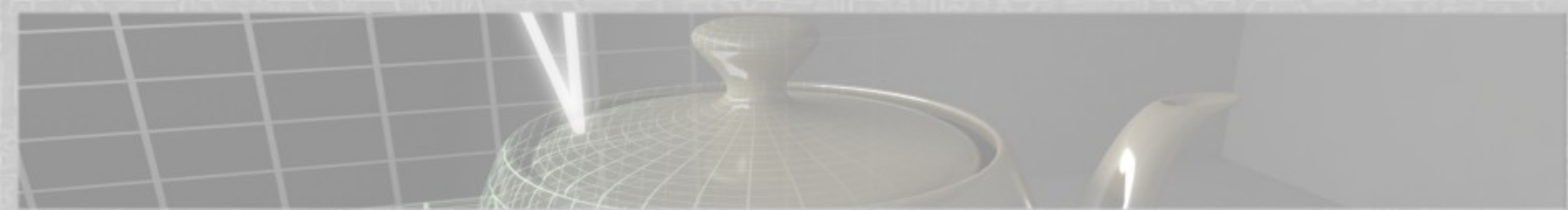


Результат



Стереоскопично рендериране

- Разделението между двете виртуални камери – `stereoSeparation` – е съществен параметър; то определя колко „голям“ е наблюдателя в сравнение със сцената
 - По-голям `stereoSeparation` допринася за по-силно усещане за обем. Възможно е да преувеличим с цел артистичен ефект
 - При прекалено голям `stereoSeparation`, картината изглежда неестествена и „сбъркана“
 - При прекалено малък `stereoSeparation`, стерео-ефекта се губи
 - Точното му калибриране е индивидуално за всяка сцена



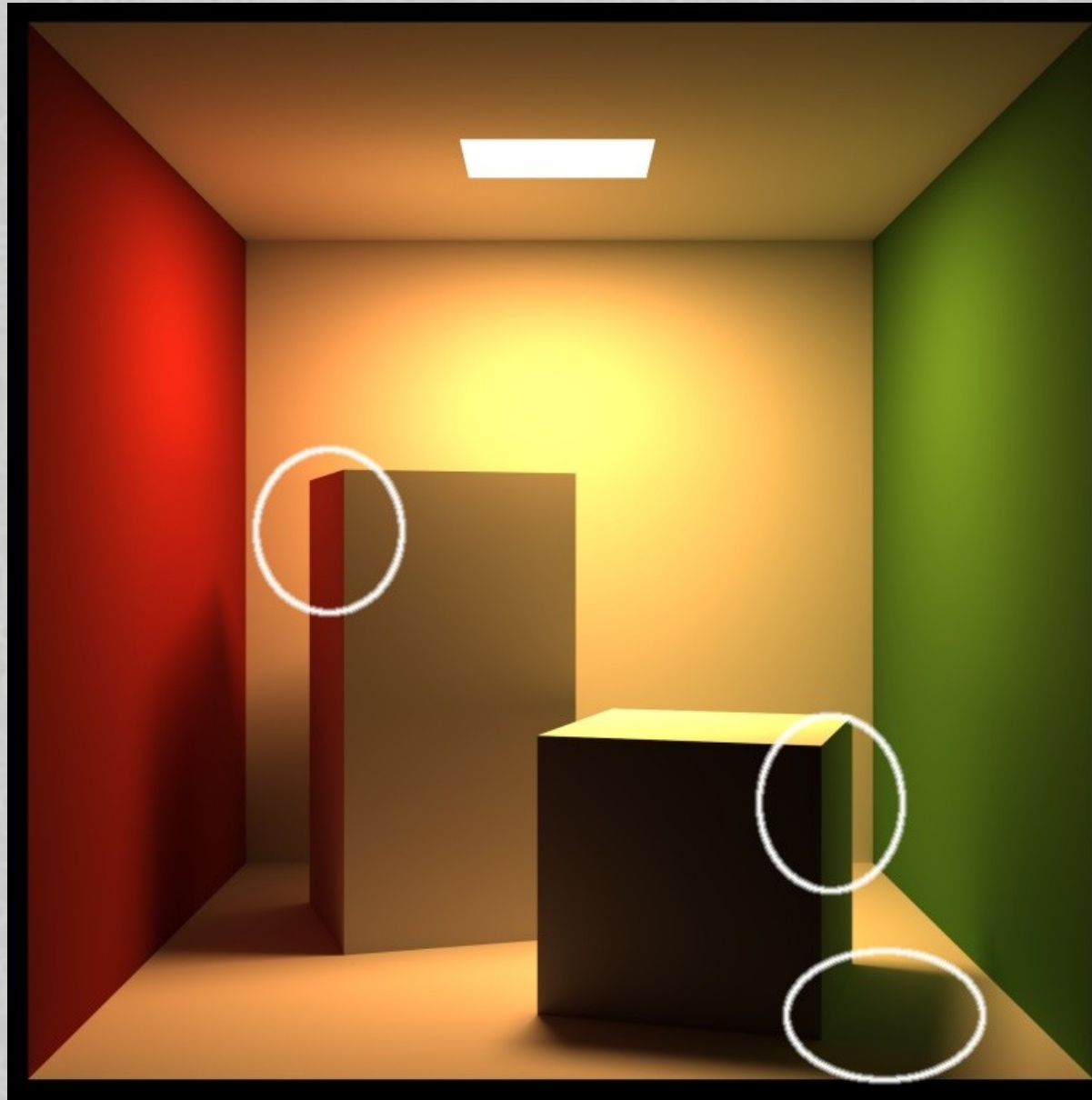
Глобально освещение

Глобално осветление

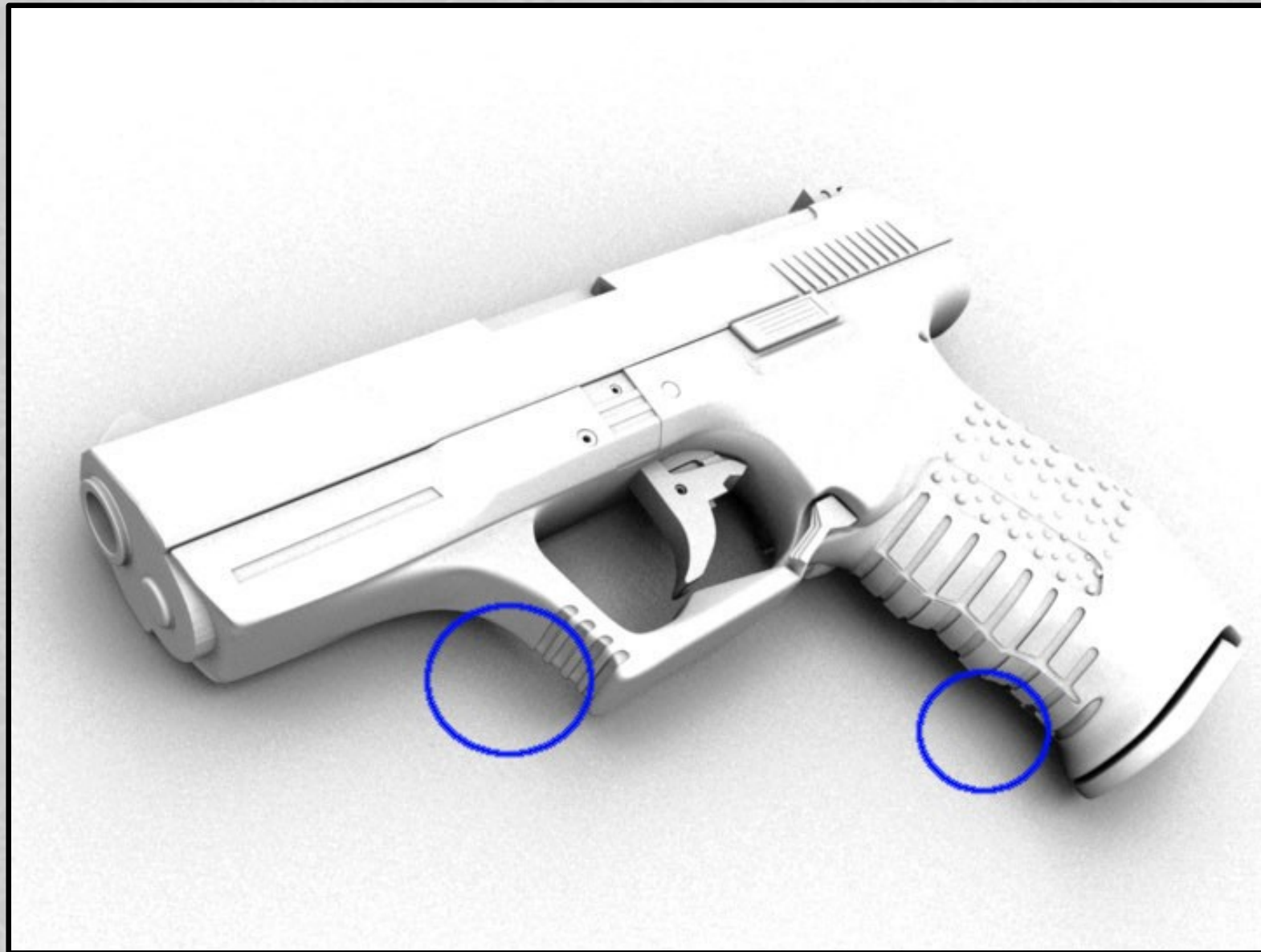
Глобално осветление = директно осветление + индиректно осветление

- Досега апроксимирахме индиректното осветление чрез ambient light – но това е много неточен подход
- Има редица ефекти, които не можем да пресъздадем само чрез директно осветление

Color bleeding



Ambient occlusion

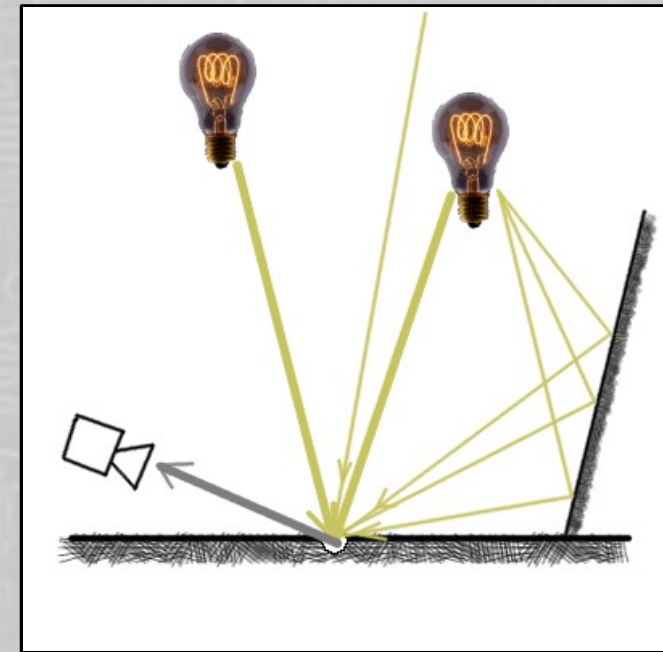


Indirect illumination



Глобално осветление

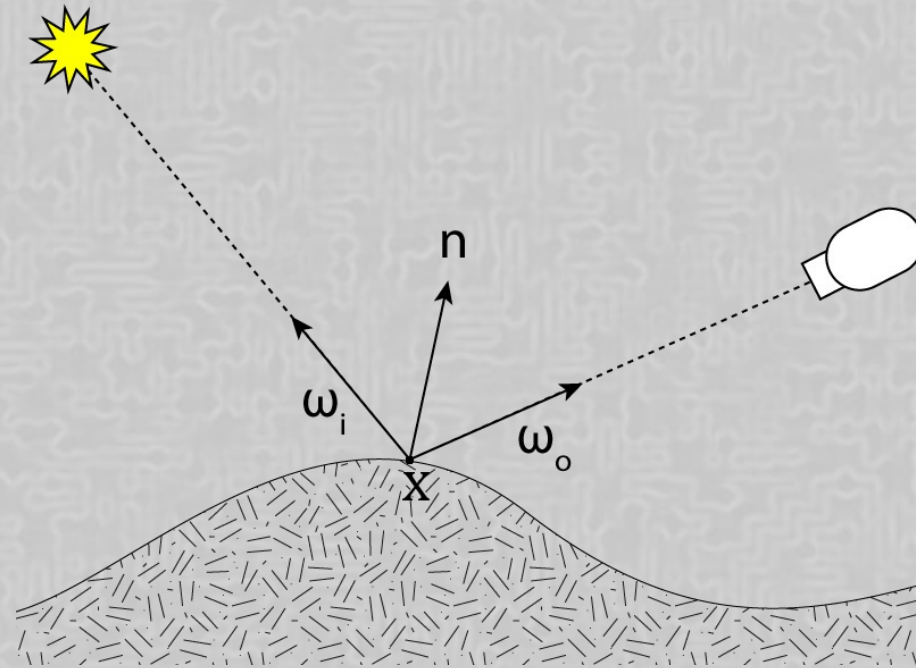
- Най-общо казано, за да пресмятаме глобалното осветление в дадена точка, трябва да:
 - Пресметнем светлината, идваща от цялата сцена, в тази точка (irradiance)
 - В зависимост от отражателните свойства на повърхността, да сметнем каква част от входната светлина се отразява по посока камерата
 - Тук ни трябва някакъв начин да моделираме отражателните свойства на повърхността (шейдърите не са подходящи)



Шейдъри vs BRDF-и

- Шейдърите не са подходящи за моделиране на отражателните свойства на повърхността
 - Те по-скоро задават алгоритъм, който описва какви други лъчи трябва да пуснем от точката, за да пресметнем осветлението; отражателните свойства са „вградени“ вътре в този алгоритъм, и са „недостъпни“ отвън
 - Иначе казано, пресмятането чрез шейдъри е ефективно, но не достатъчно общо, за да се ползва в GI алгоритмите
 - Необходимо е да използваме отделна функция, която моделира именно отражателните свойства.

Функция на отраженията (BRDF)



Bidirectional Reflectance Distribution Function или f_r :

$f_r(x, \omega_i, \omega_o) :=$ „каква част от светлината, идваща от ω_i в точката x , се отразява в посоката ω_o “

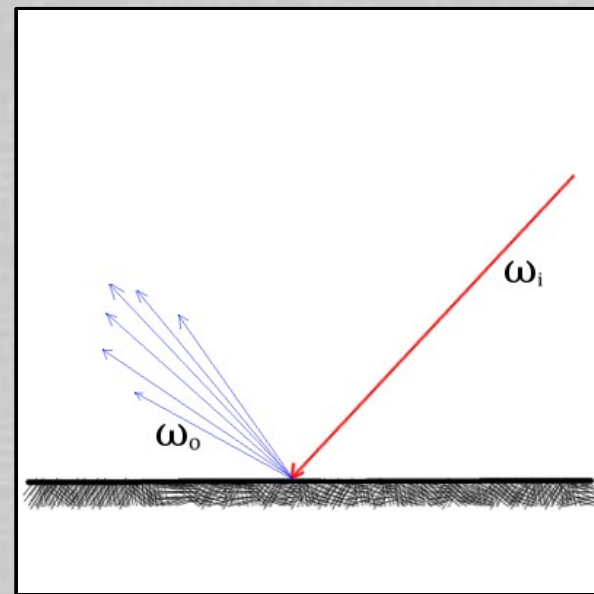
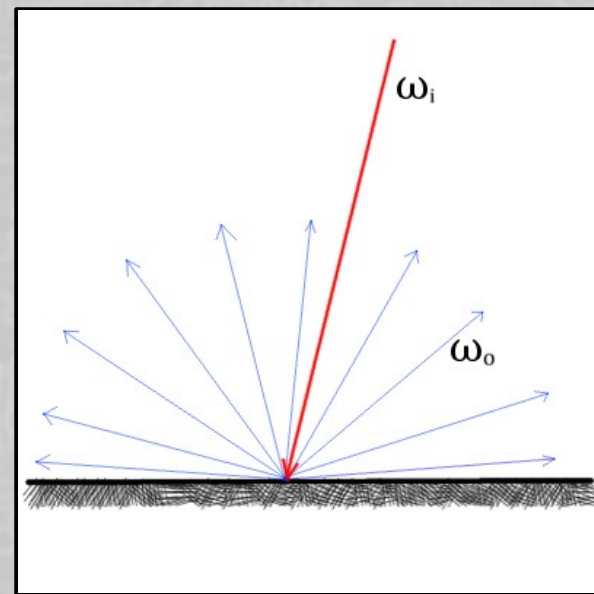
Примери за BRDF

- Дифузен BRDF (Lambertian – това, което стои зад Flat шейдъра), за бяла повърхност:

$$f_r(x, \omega_i, \omega_o) := 1$$

- Glossy BRDF (за грапави отражения):

$$f_r(x, \omega_i, \omega_o) := \max(0, \cos(\angle(\text{reflect}(\omega_i), \omega_o)))^n$$

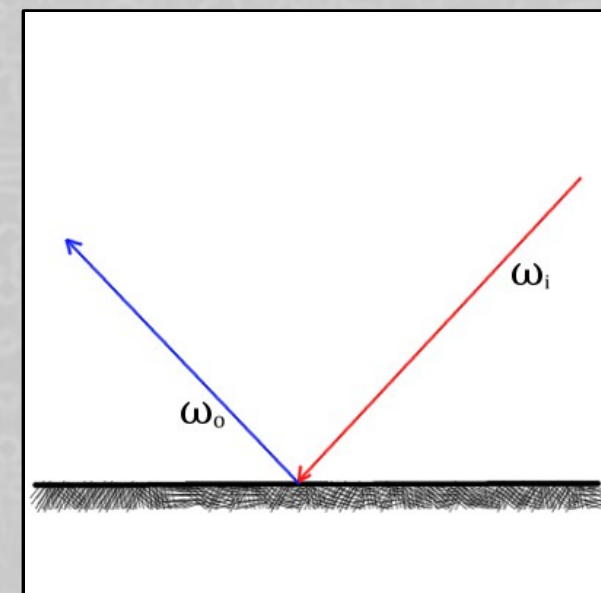


Примери за BRDF

- Чисто отражение:

$$f_r(x, \omega_i, \omega_o) := \begin{cases} 1, & \text{ако } \omega_o = \text{reflect}(\omega_i) \\ 0, & \text{иначе} \end{cases}$$

- Т.е., BRDF-ът може и да е прекъснатата функция



Свойства на BRDF-ите

- Запазване на енергията: $f_r(x, \omega_i, \omega_o) \leq 1$
 - Това гарантира, че не се „създава“ светлина
- Симетричност: $f_r(x, \omega_i, \omega_o) = f_r(x, \omega_o, \omega_i)$

BRDFs: малко по-сложно

- Казахме, че дифузния BRDF е 1 навсякъде, но това е само при бял обект без текстура. Ако има (дифузна) текстура, то връщаният резултат зависи от точката x :

$$f_r(x, \omega_i, \omega_o) := \text{texture.getColor}(x)$$

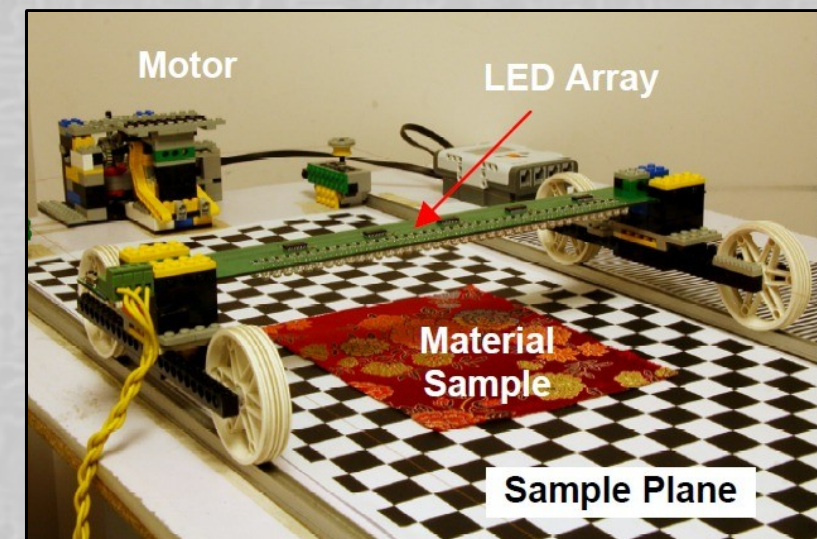
- Подобно и за останалите BRDF-и, които споменахме. Например, `reflect()` функцията изисква нормал, около който да отразява, а той зависи от точката x .

BRDFs: още по-сложно

- В някои статии, BRDF-ите имат и четвърти параметър – дължина на вълната. Това е полезно, ако се опитваме да реализираме BRDF на хроматично пречупване
 - Ние няма да го реализираме, но е хубаво да се знае

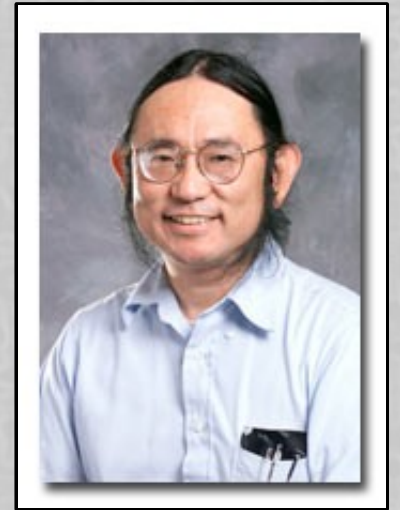
Непроцедурни BRDF-и

- В някои случаи, свойствата на BRDF-а може да не се изчисляват чрез формула, а да се взимат директно от текстура (или някакъв друг начин за справка с предварително известни данни)
- Съществуват устройства, които „сканират“ даден материал и извличат BRDF-то му
[041-wang-video]



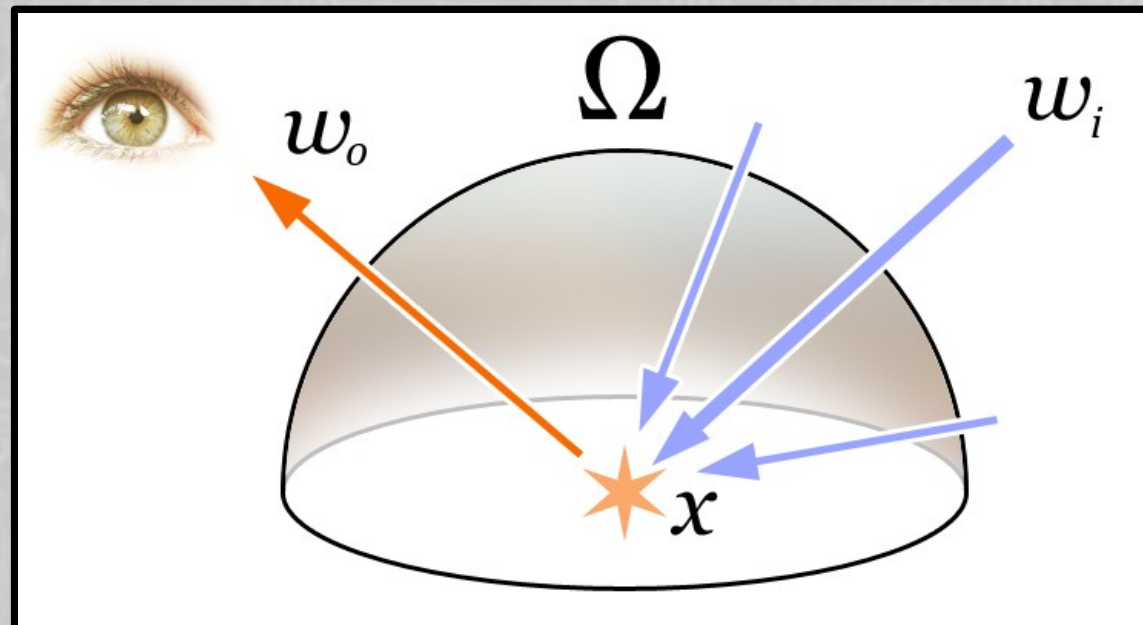
„Фундаменталната формула“ на Каджия

- Jim Kajiya през 1986 представя своя основополагащ труд в областта на глобалното осветление - „The Rendering Equation“ [SIGGRAPH-1986]
- Самата „фундаментална формула“ представлява транспортно уравнение на светлината, което ни дава израз за пресмятане на глобалното осветление във всяка точка от сцената



The Rendering Equation

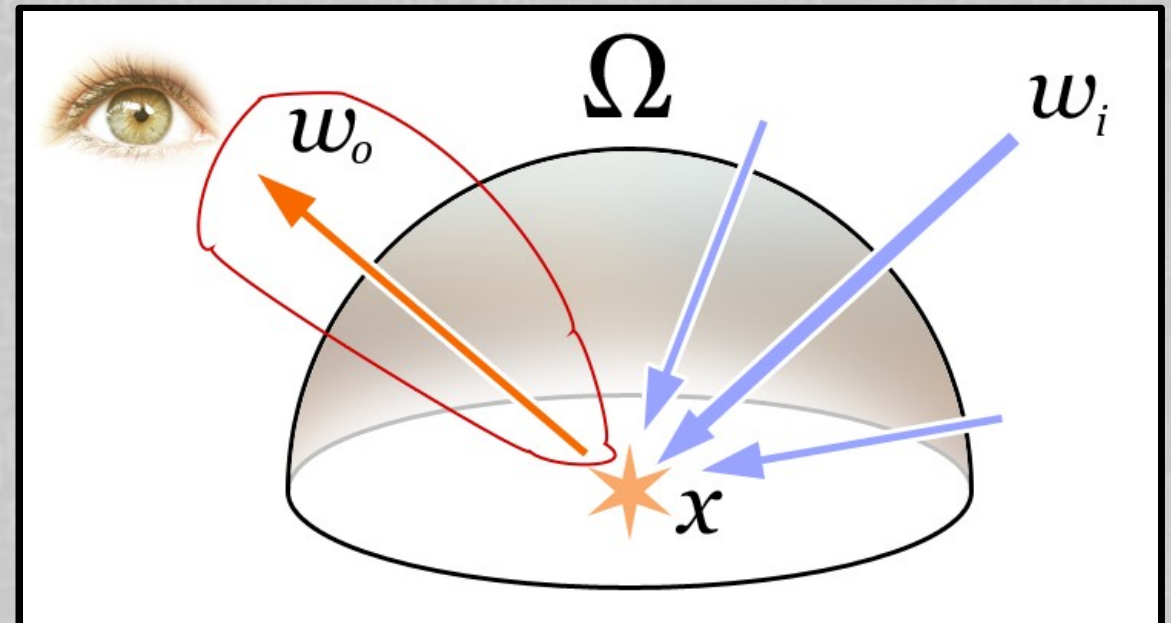
$$\mathbf{L}_o(\mathbf{x}, \omega_o) = \mathbf{L}_e(\mathbf{x}, \omega_o) + \int_{\Omega} \mathbf{f}_r(\mathbf{x}, \omega_i, \omega_o) \mathbf{L}_i(\mathbf{x}, \omega_i) (-\omega_i \cdot \mathbf{n}) d\omega_i$$



The Rendering Equation

$$\mathbf{L}_o(\mathbf{x}, \omega_o) = \mathbf{L}_e(\mathbf{x}, \omega_o) + \int_{\Omega} \mathbf{f}_r(\mathbf{x}, \omega_i, \omega_o) \mathbf{L}_i(\mathbf{x}, \omega_i) (-\omega_i \cdot \mathbf{n}) d\omega_i$$

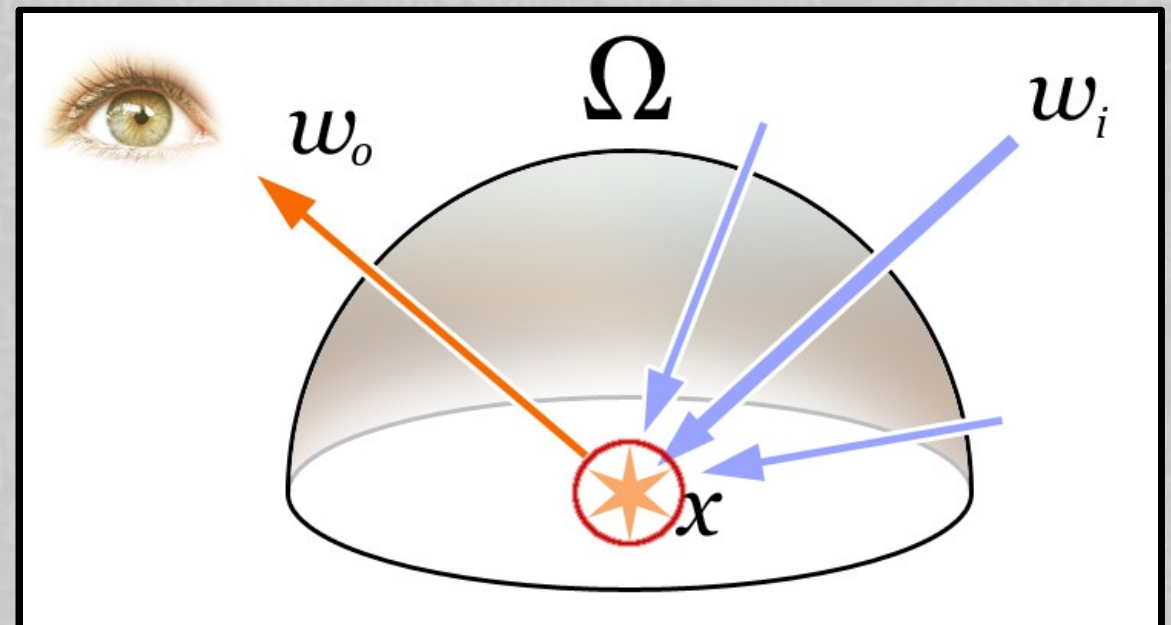
- $L_o(x, \omega_o)$ показва отразената светлина от точката x в посока ω_o
Например, към камерата:
- Ако от камерата сме пуснали лъч $-\omega_o$, който е пресякъл геометрията в точка x .



The Rendering Equation

$$L_o(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \int_{\Omega} f_r(\mathbf{x}, \omega_i, \omega_o) L_i(\mathbf{x}, \omega_i) (-\omega_i \cdot \mathbf{n}) d\omega_i$$

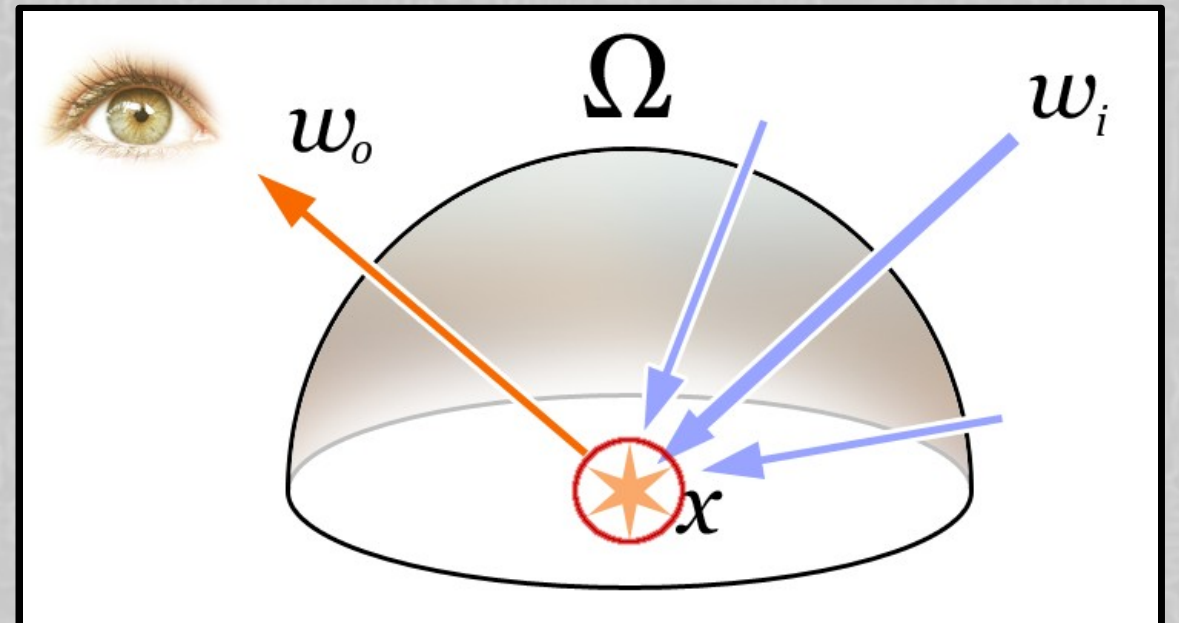
- Ако x е част от лампа, $L_e(x, \omega_o)$ показва излъчваната светлина от точката x в посока ω_o
- 0 за всички неизлъчващи обекти.



The Rendering Equation

$$\mathbf{L}_o(\mathbf{x}, \omega_o) = \mathbf{L}_e(\mathbf{x}, \omega_o) + \int_{\Omega} \mathbf{f}_r(\mathbf{x}, \omega_i, \omega_o) \mathbf{L}_i(\mathbf{x}, \omega_i) (-\omega_i \cdot \mathbf{n}) d\omega_i$$

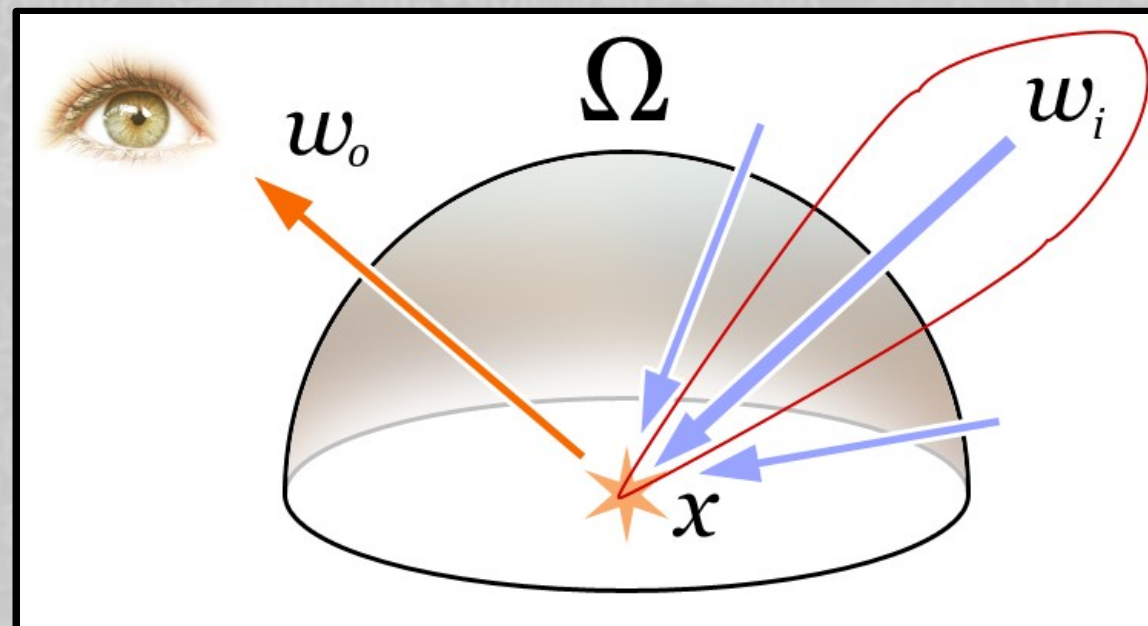
- $f_r(x, \omega_i, \omega_o)$ е BRDF-ът на материала в точката x



The Rendering Equation

$$L_o(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \int_{\Omega} f_r(\mathbf{x}, \omega_i, \omega_o) L_i(\mathbf{x}, \omega_i) (-\omega_i \cdot \mathbf{n}) d\omega_i$$

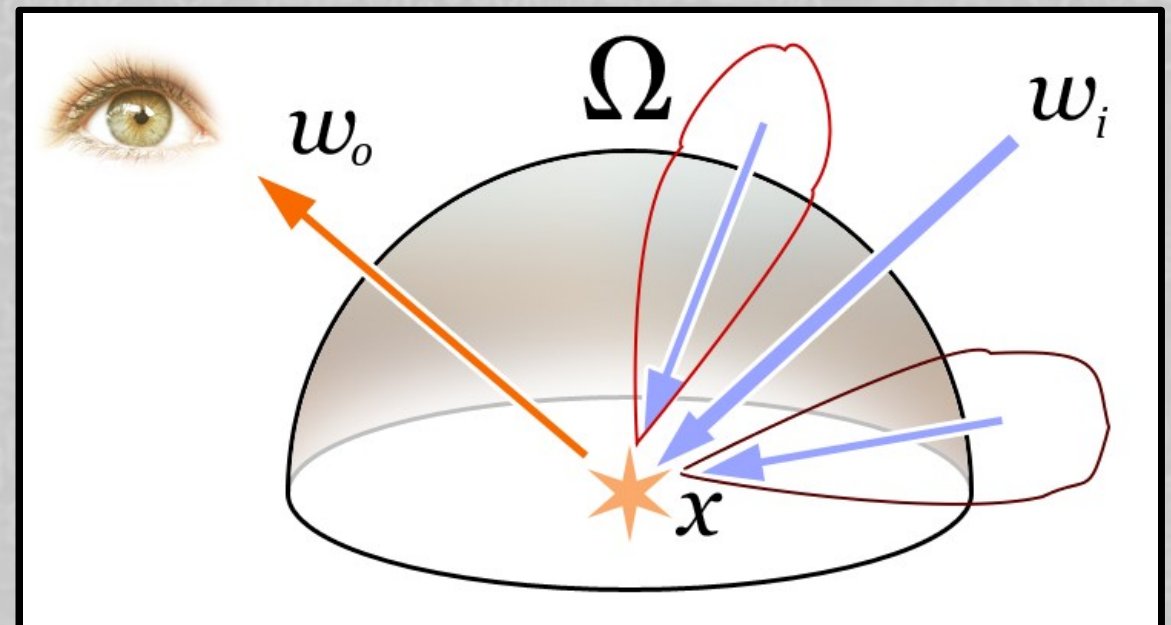
- $L_i(\mathbf{x}, \omega_i)$ показва входящата светлина в точката \mathbf{x} по някое направление ω_i . Тя може да идва както от лампа, така и от неизлъчващ обект (чрез отражение на светлината, т.е. от друг екземпляр на транспортното уравнение)



The Rendering Equation

$$\mathbf{L}_o(\mathbf{x}, \omega_o) = \mathbf{L}_e(\mathbf{x}, \omega_o) + \int_{\Omega} \mathbf{f}_r(\mathbf{x}, \omega_i, \omega_o) \mathbf{L}_i(\mathbf{x}, \omega_i) (-\omega_i \cdot \mathbf{n}) d\omega_i$$

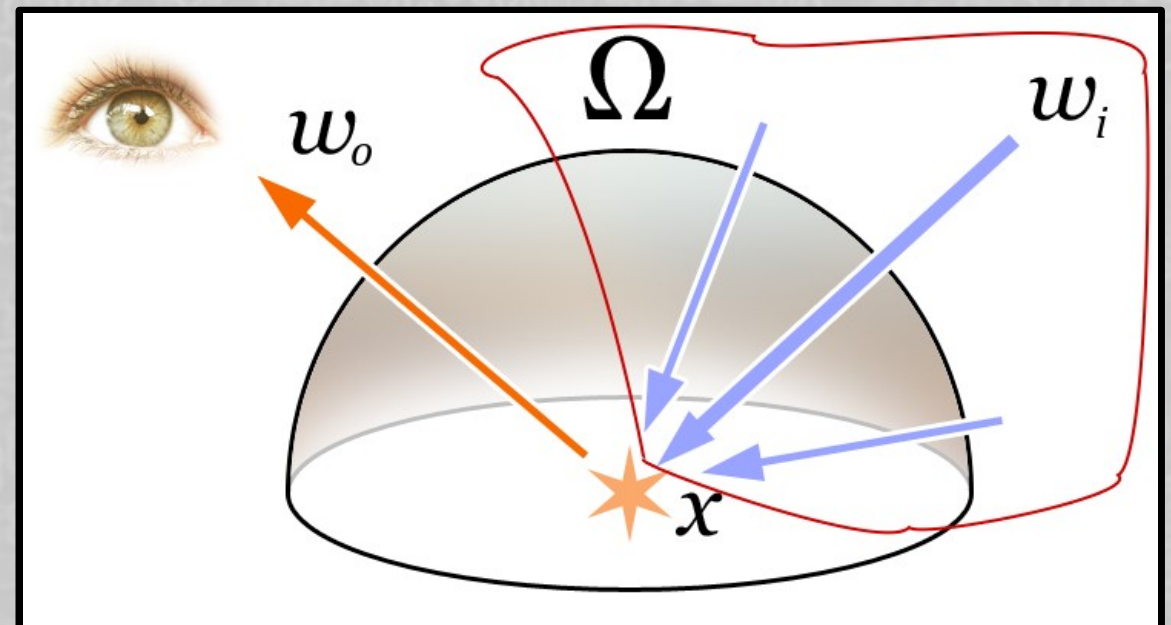
- Косинусът на ъгъла между ω_i и нормалата \mathbf{n} в точката \mathbf{x} участва, защото, под коси ъгли, входната светлина се разпределя на по-голяма площ. Така, на единица площ, осветяването е по-слабо, следователно и отражението трябва да е по-слабо.



The Rendering Equation

$$L_o(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \int_{\Omega} f_r(\mathbf{x}, \omega_i, \omega_o) L_i(\mathbf{x}, \omega_i) (-\omega_i \cdot \mathbf{n}) d\omega_i$$

- Чрез интеграла ние сумираме цялата входяща светлина, отразена от BRDF-а в конкретната посока. Т.е. вътрешната сметка извършваме за всички входящи направления на светлината – което е цялата полусфера Ω над точката x .

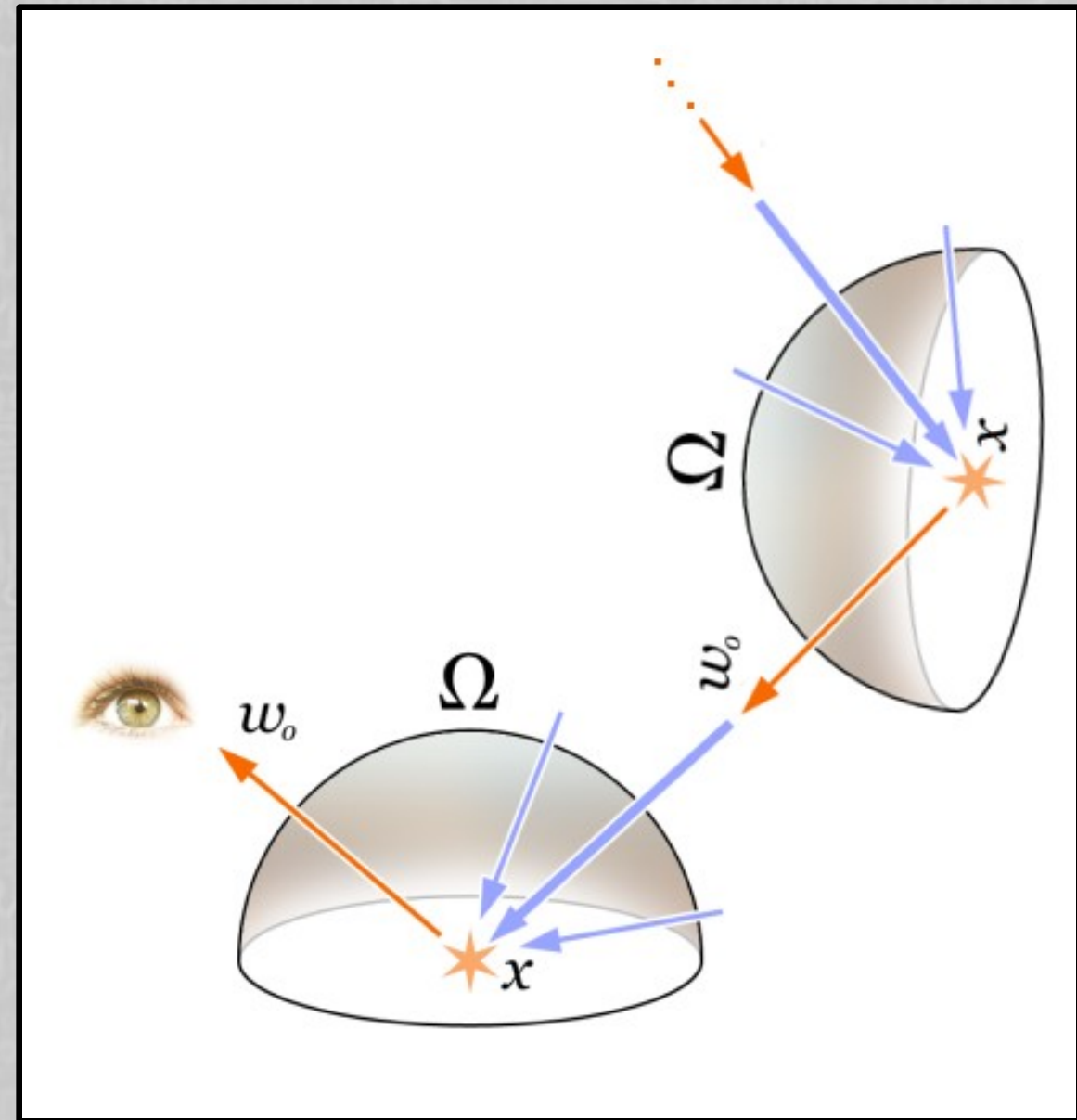


Особености на уравнението

- Уравнението е линейно – абстрахирайки се от BRDF-а, във формулата участват само събиращения и умножения. Това дава възможност на практика да се правят разнообразни опростявания и апроксимации.
- Уравнението е рекурсивно – ако тръгнем да го решаваме с Monte Carlo метод, ще стигнем до друг екземпляр от уравнението, за решението на който отново ще ни трябва да решим друг екземпляр, и т.н.
- Изглежда, че всяка точка от сцената зависи от всяка друга точка от сцената!

Рекурсивност

- За да се получим повечето от GI ефектите, то обикновено трябва да преровим сцената поне няколко нива рекурсивно.
- Т.е., като от точката x пуснем случаен лъч $-\omega_i$, проследим къде се удря той в сцената (x') и после пресметнем транспортното уравнение в точката x'

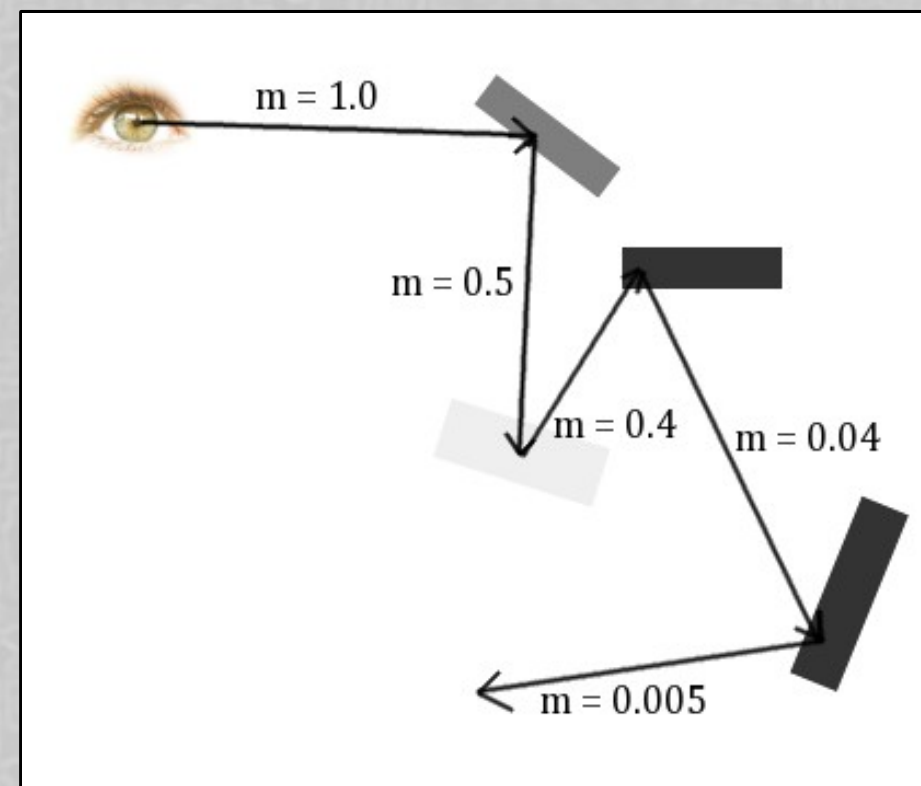


Рекурсивност

- Кога приключва рекурсията?
 - Ако ударим светлина
 - BRDF-а на светлината спокойно може да приемем за 0
 - Ако не ударим нищо (т.е. попаднем в Environment-а – тогава вземаме цвета от околната среда в тази посока)
 - Ако ударим слабо-отразяващ обект (т.е. BRDF-а е 0 или почти 0)

Рекурсивност

- Последната идея може да разширим така: може да пазим текущото произведение от всички BRDF-и по пътя на рекурсията. Ако това произведение стане почти 0, то може да прекратим рекурсията (приносът от обхождането още надолу ще е прекалено малък и може да го пренебрегнем)



Решаване чрез Monte Carlo метод

- Така зададеното уравнение може да сметнем чрез Monte Carlo метод
 - При пресмятането на интеграла, ще пуснем много случайни лъчи от точката x , ще сметнем осветлението от всеки от тях, и ще ги съберем, умножавайки по BRDF-а. Резултата ще разделим на броя на лъчите
 - Но така получаваме комбинаторна експлозия: за добро качество трябва да пускаме по много лъчи (над 20), и само няколко нива надолу вече трябва да трасираме милиони лъчи (за единствен пиксел от екрана!)

Решаване чрез Monte Carlo метод

- Можем да пускаме по много лъчи само на най-горното ниво на рекурсията, а по-долните да смятаме само с по няколко лъча; въпреки това, този метод не се справя добре и е много бавен
- По-добър алгоритъм предлага самият Каджия, в същата статия. Алгоритъмът се нарича Path tracing.

Rendering Equation (yet again =-)

$$L_o(x, \vec{w}) = L_e(x, \vec{w}) + \int_{\Omega} f_r(x, \vec{w}', \vec{w}) L_i(x, \vec{w}') (\vec{w}' \cdot \vec{n}) d\vec{w}'$$

- Как може да се сметне нещо подобно?

Връзка между L_i и L_o

Color $L_i(x, w)$:

$$x' = \text{trace}(x, w)$$

$$w' = -w$$

return $L_o(x', w')$

Rendering equation in pseudocode

$$L_o(x, \vec{w}) = L_e(x, \vec{w}) + \int_{\Omega} f_r(x, \vec{w}', \vec{w}) L_i(x, \vec{w}') (\vec{w}' \cdot \vec{n}) d\vec{w}'$$

Color $L_o(x, w)$:

Color sum = 0

for i in 1..num_samples:

$w', pdf = \text{generate_new_random_dir}(x, w)$

 sum += $F_r(x, w, w') * L_i(x, -w') * \text{dot}(w', N) * (1 / pdf)$

sum /= num_samples

return $L_e(x, w) + \text{sum}$

Algorithms for Global Illumination

- Алгоритмите за GI могат основно да се разделят на
 - Неотместени (unbiased)
 - path tracing, light tracing, bidirectional tracing, metropolis light transport, etc.
 - Отместени (biased)
 - photon mapping, irradiance caching, instant radiosity, etc.
- Неотместен алгоритъм (грубо казано) е такъв, при който грешката се изразява само в шум
 - При отместените могат да се получат резултати, в които няма шум, но грешката да е произволно голяма

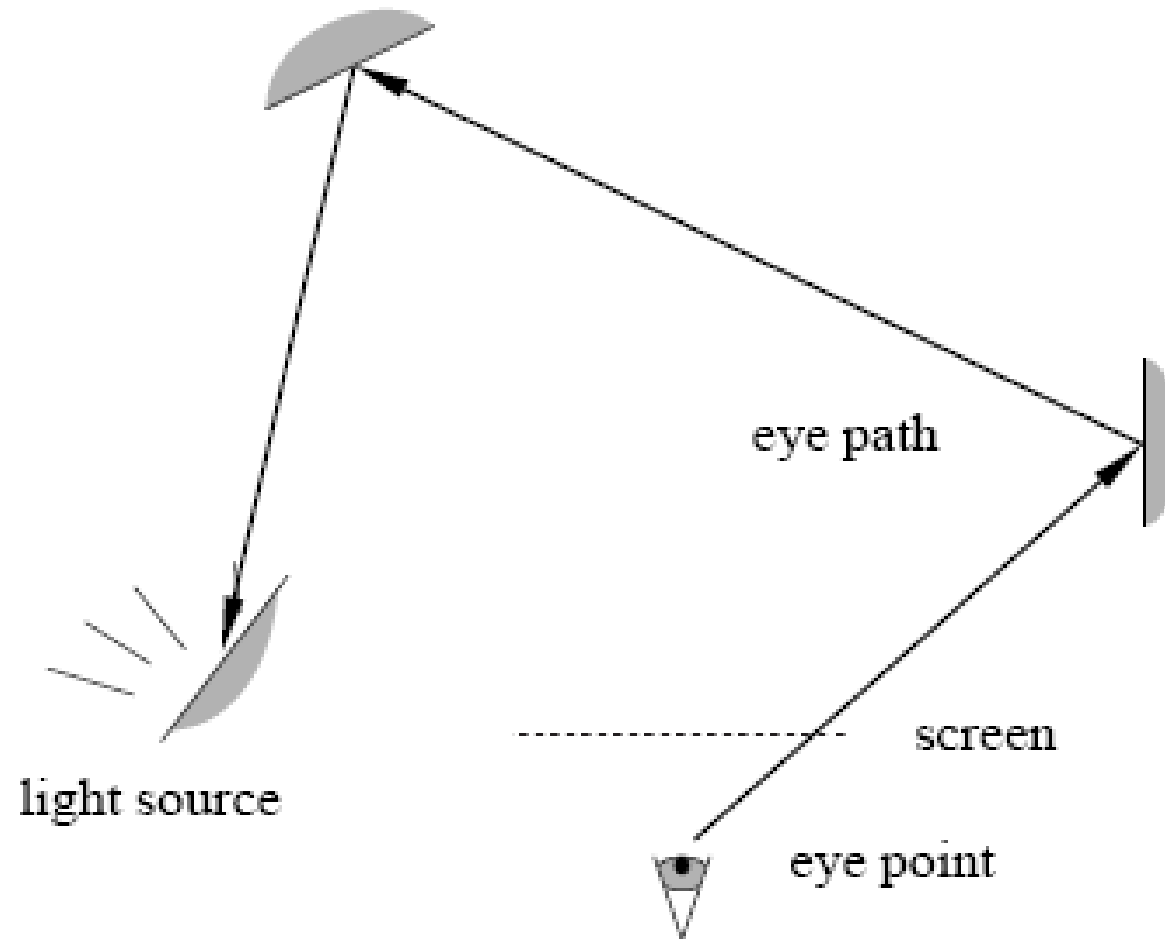
Algorithms for Global Illumination (II)

- Неотместени алгоритми са
 - + предвидими
 - и в математически смисъл на думата
 - + фотореалистични
 - но имат и ограничения, например path tracing не може да получи слънчеви зайчета от не-glossy пречупване и точкова светлина
 - - бавни
 - т.е. отместените алгоритми могат да получат гладка, но грешна картинка по-бързо, отколкото неотместените могат да сметнат гладка, но вярна

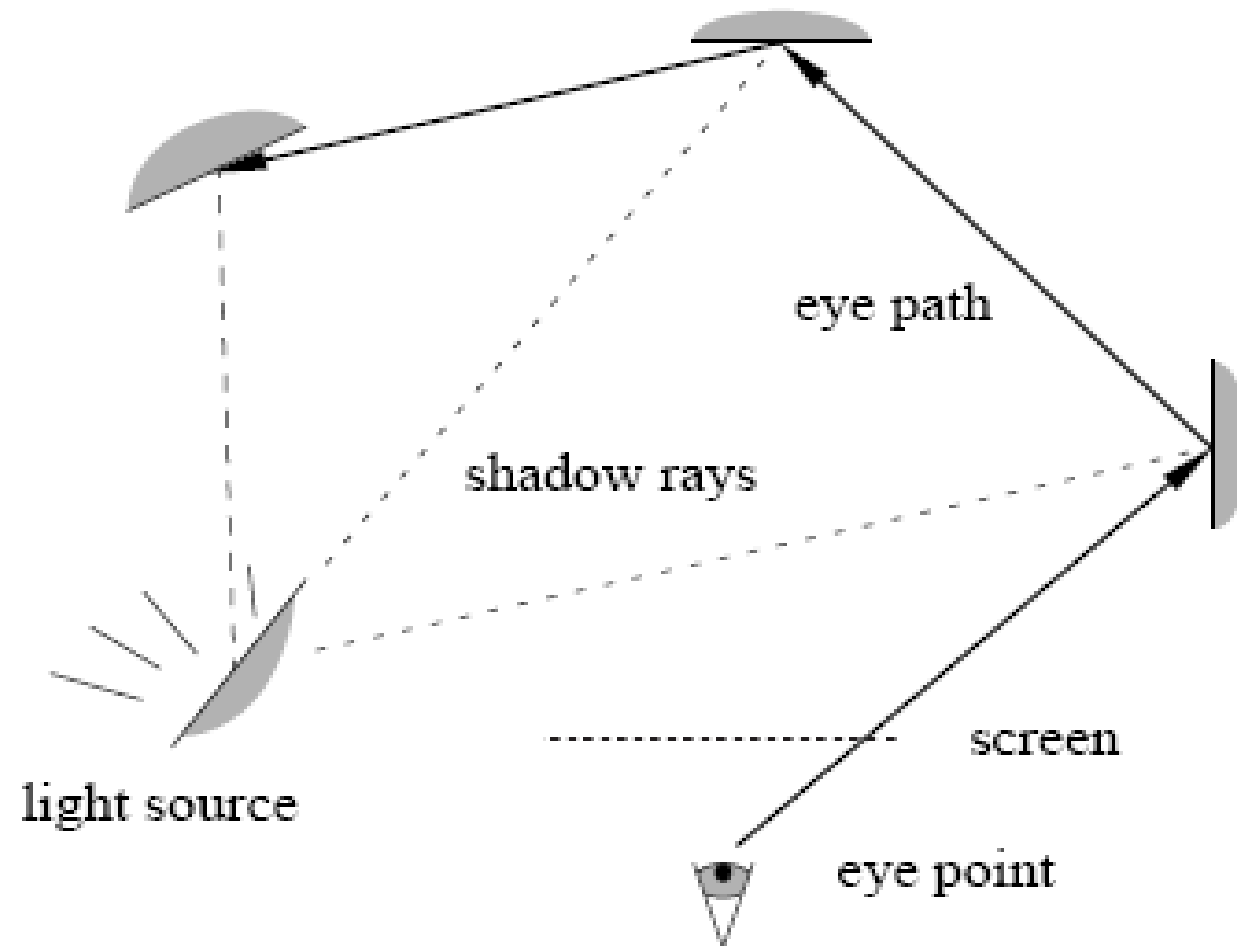
Algorithms for Global Illumination (III)

- Ще разгледаме path tracing, light tracing и bidirectional tracing
- „Път“ ще наричаме редица от точки, такива че
 - Първата точка е върху лещата на камерата
 - Последната точка е върху някоя светлина
 - Останалите точки са върху някои от обектите в сцената
- И трите алгоритъм използват, че формулата на Каджия може да се разпише като интеграл над всички пътища от сцената
- Основната разлика между трите е в начина, по-който се генерират пътищата

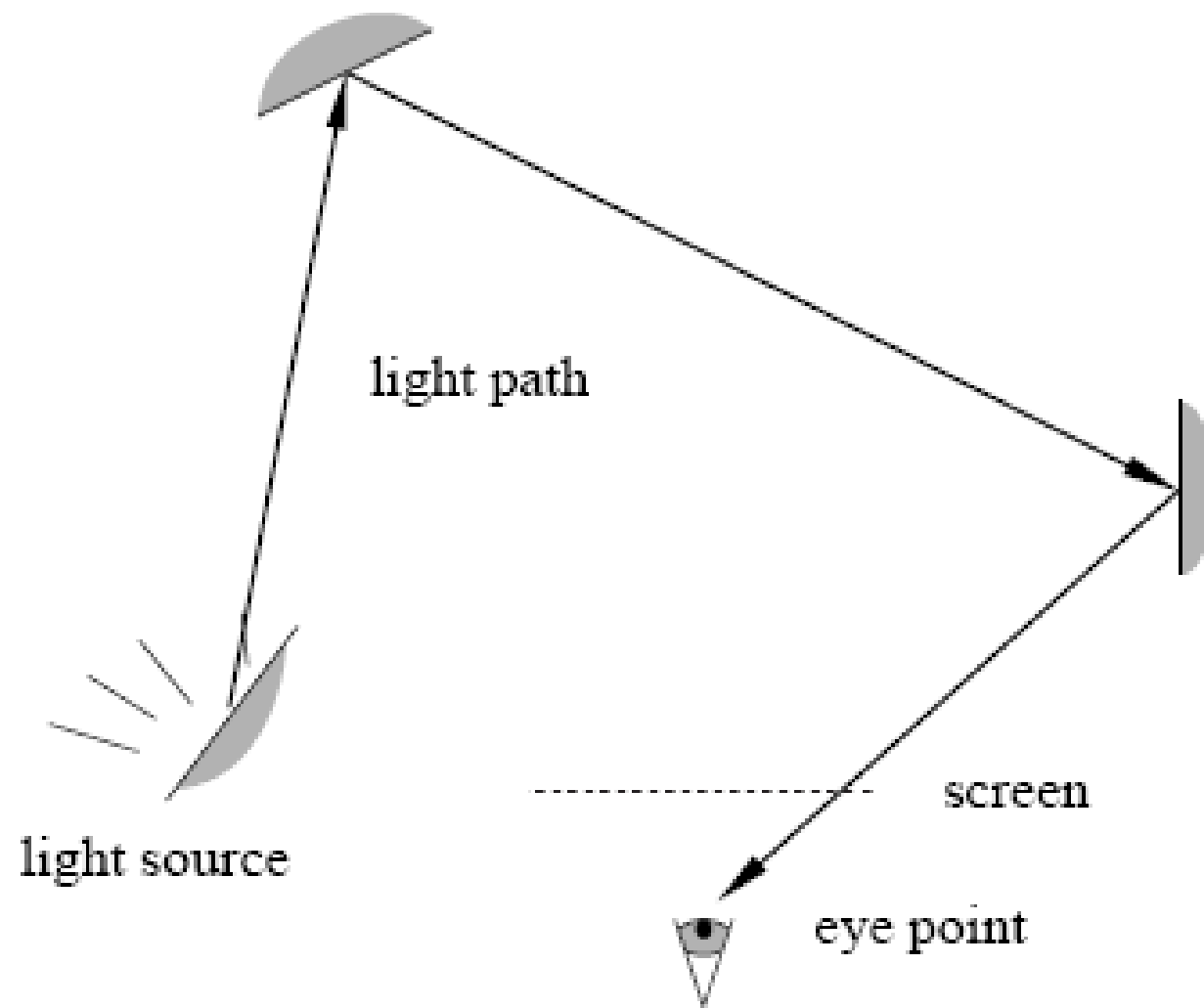
Path tracing



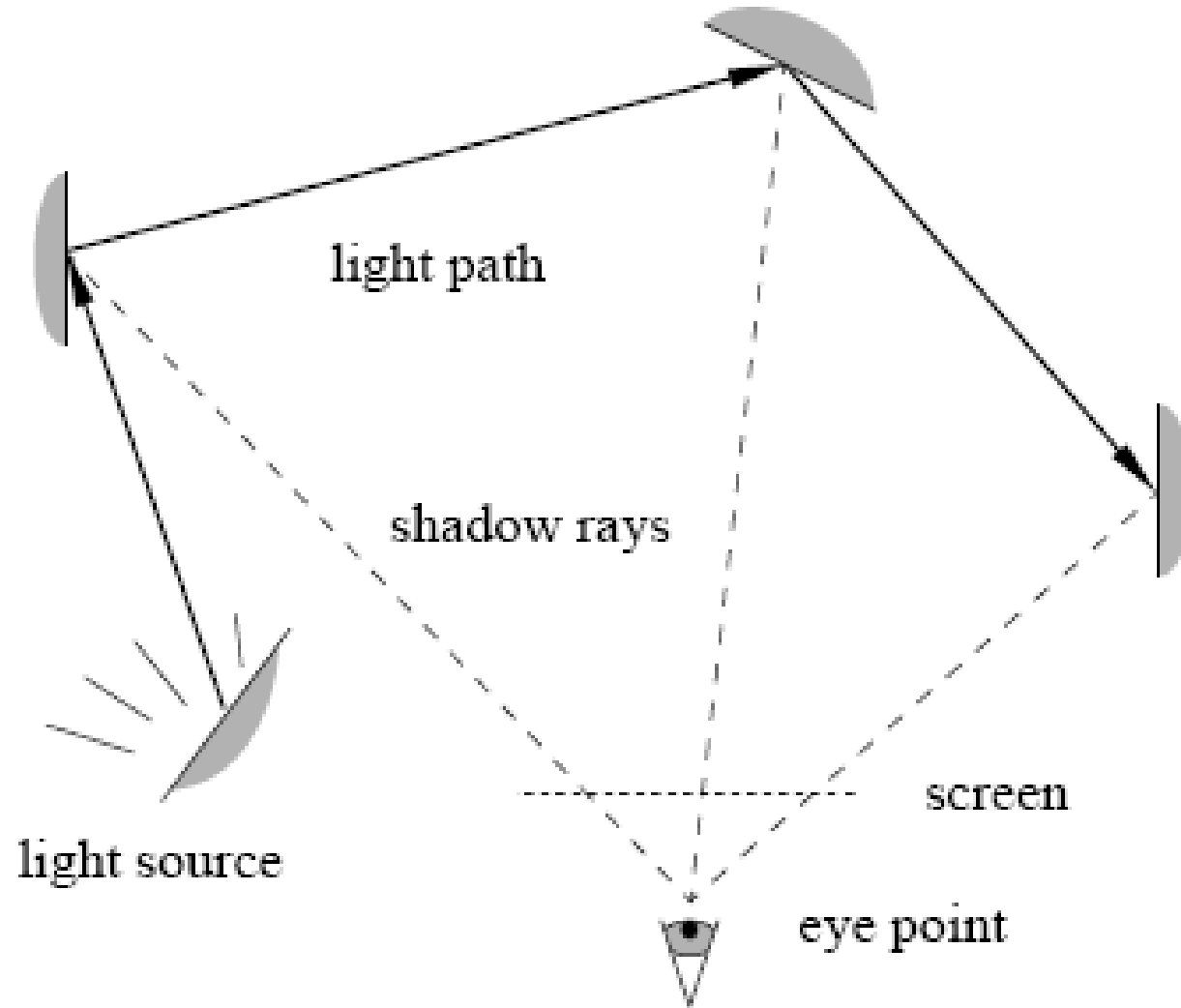
Path tracing (II)



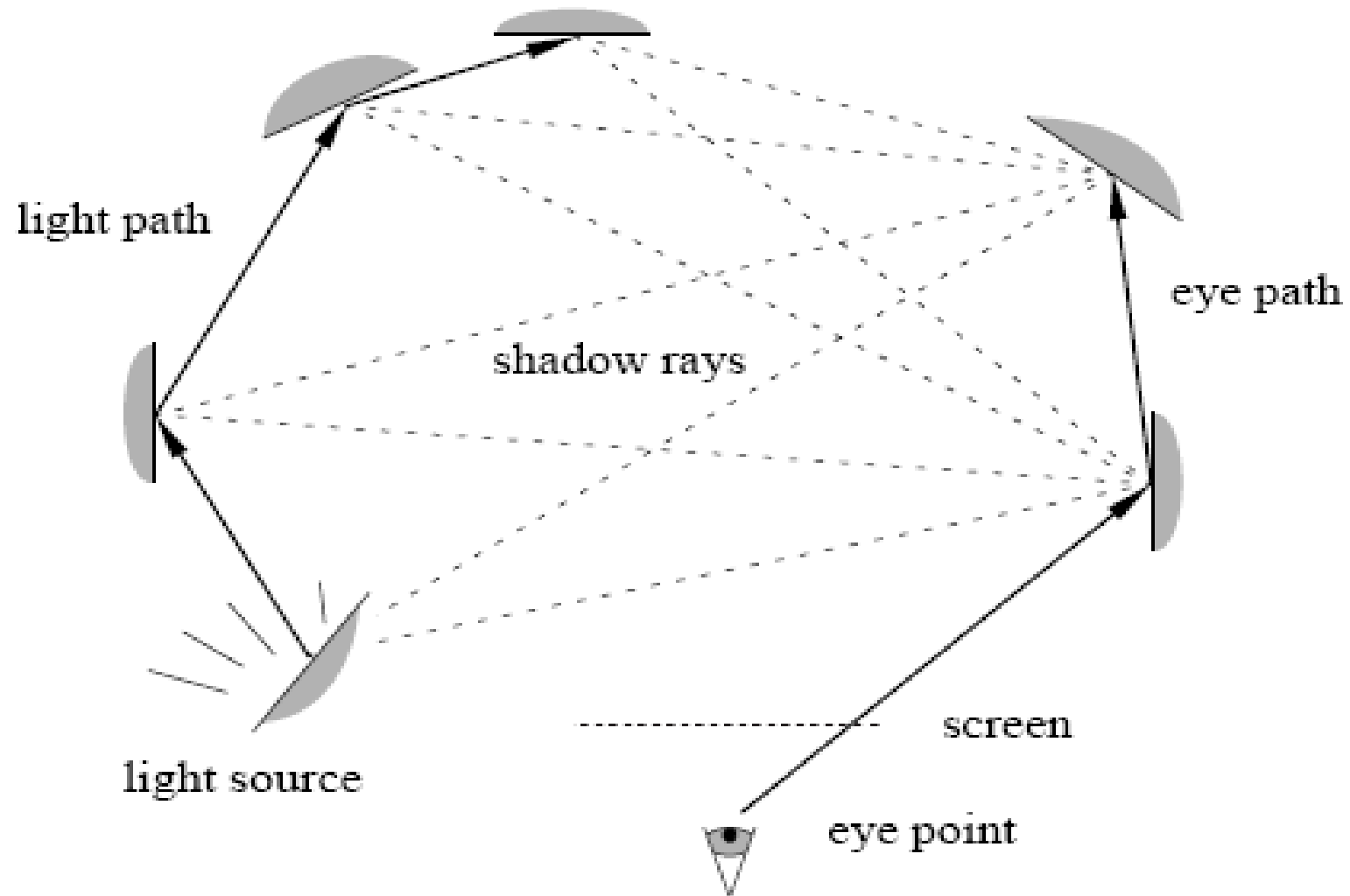
Light tracing



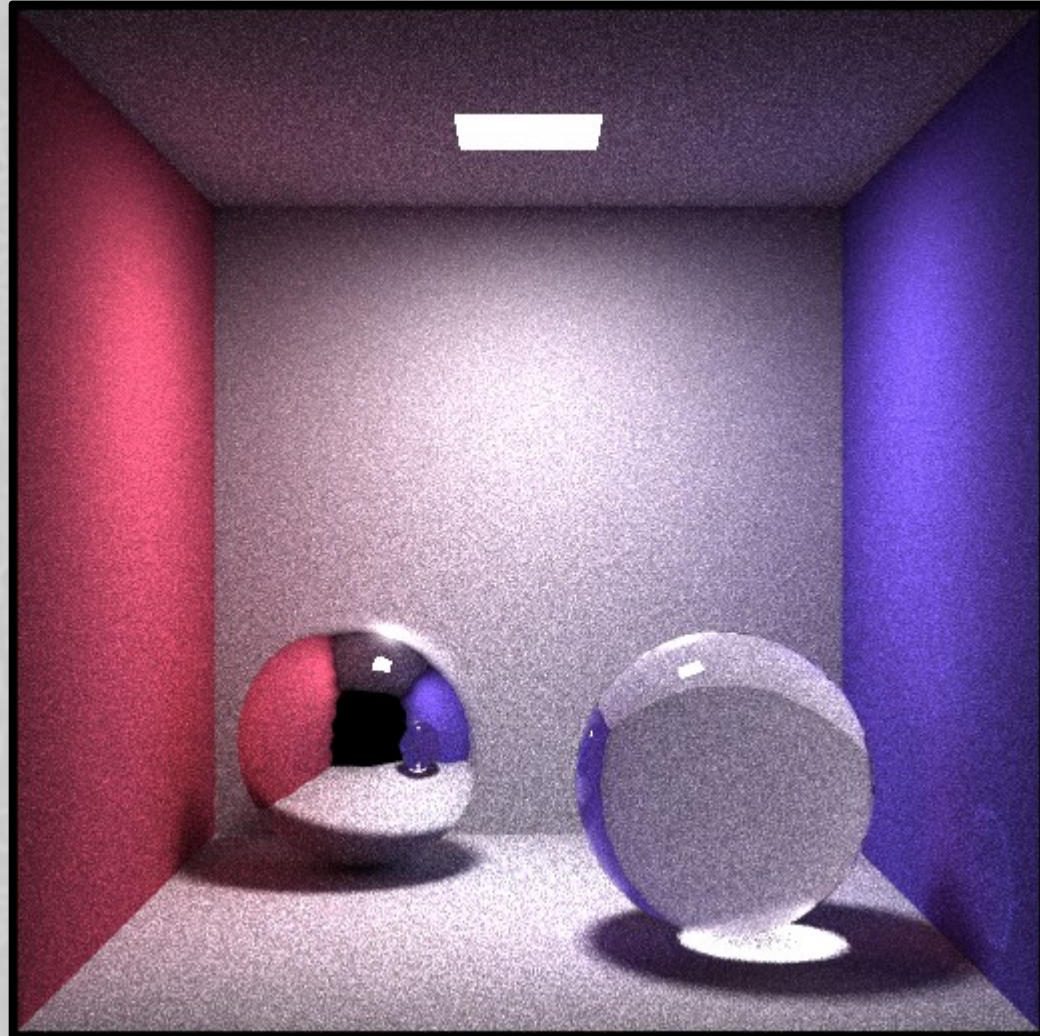
Light tracing (II)



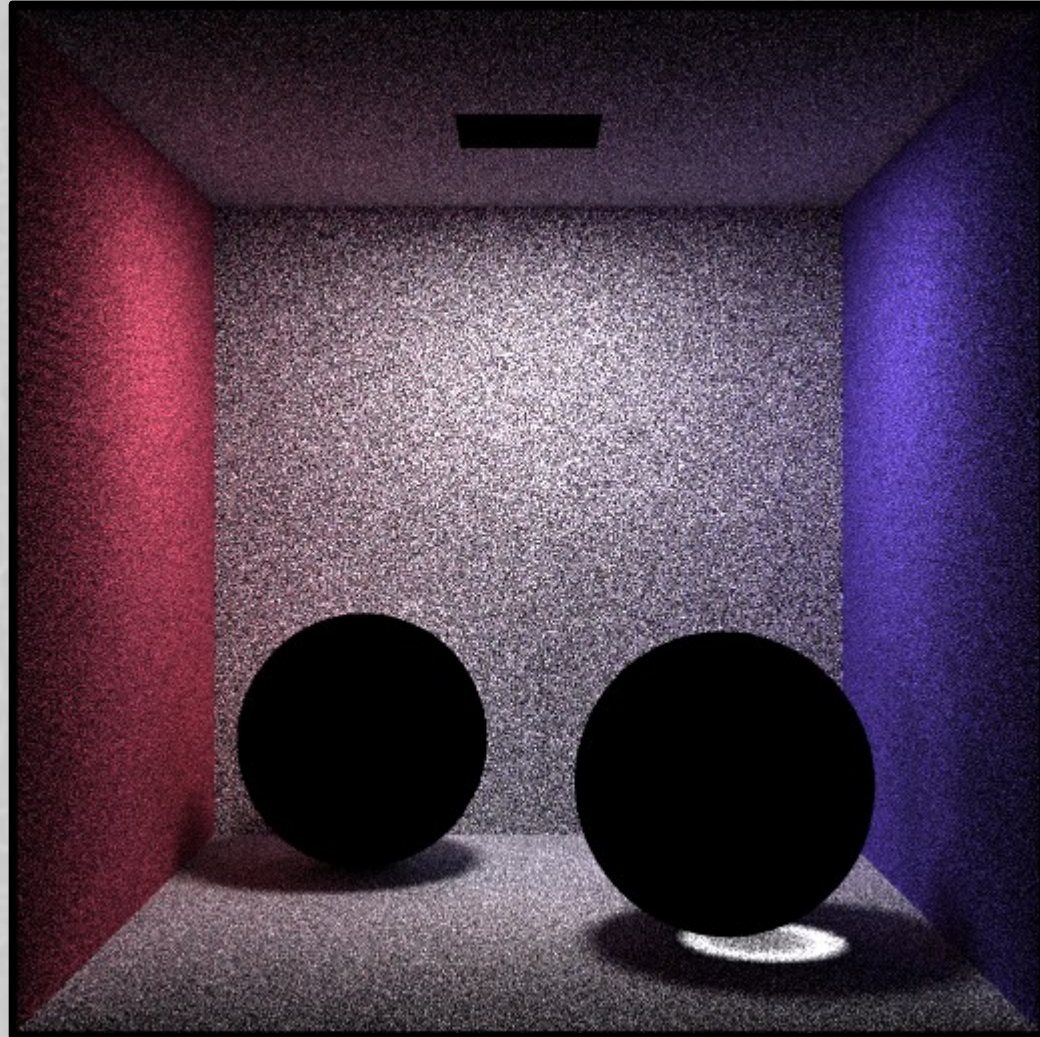
Bidirectional path tracing



Path tracing result



Light tracing result



Bidirectional tracing result

