

# Инженерная и компьютерная графика

## 6 семестр (диф.зачет)

Лектор:

Таранцев Игорь Геннадьевич  
*Доцент ФИТ НГУ, ИАиЭ, «СофтЛаб-НСК»*

Создатели курса:

Дебелов Виктор Алексеевич  
Валеев Тагир Фаридович  
Козлов Дмитрий Сергеевич

# Лекция №12

Тени и текстуры

# Тени и их значение для восприятия трёхмерного мира

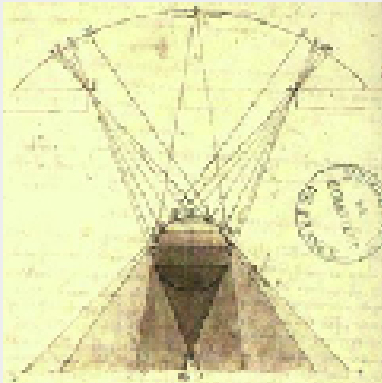


Fig.1 *Study Of Shadows*  
by Leonardo Da Vinci

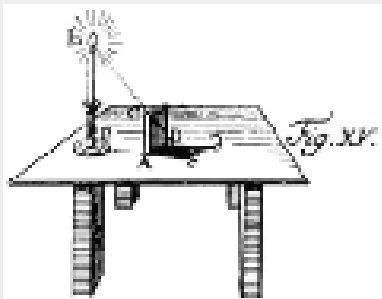


Fig.2 *Shadow construction*  
by Lambert

- Позиция затенителя и затеняемого
- Геометрия затенителя
- Геометрия затеняемого
- Динамика
- Восприятие глубины

# Важность теней

Высота и ее изменение по времени



Fig. 1



Fig. 2a



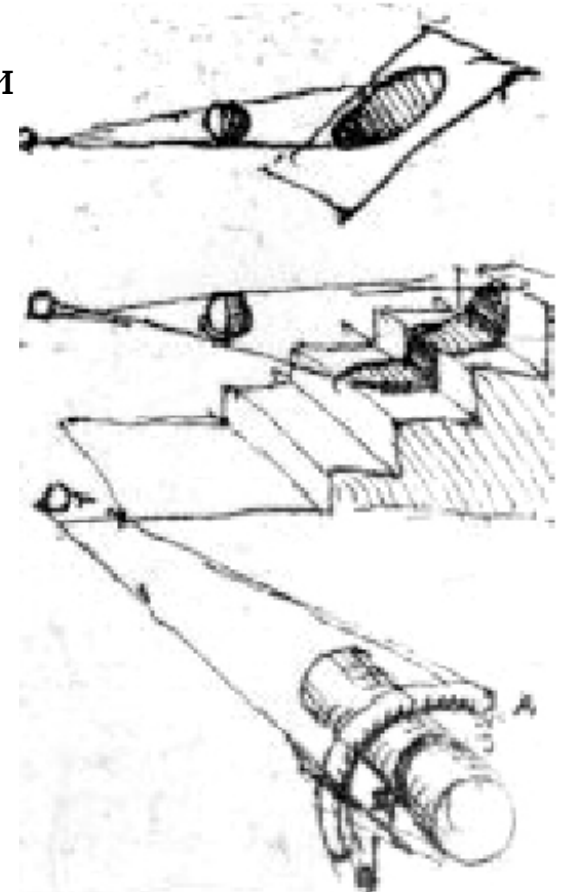
Fig. 2b



Fig. 3

- Позиция затенителя и затеняемого
- Геометрия затенителя
- Геометрия затеняемого

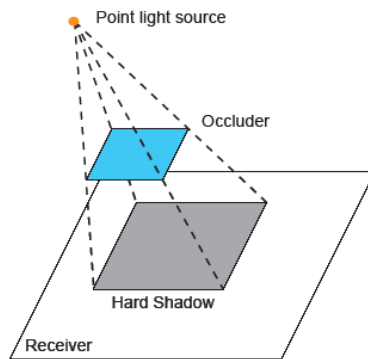
Геометрия с другого ракурса  
(чайник в руке у робота)



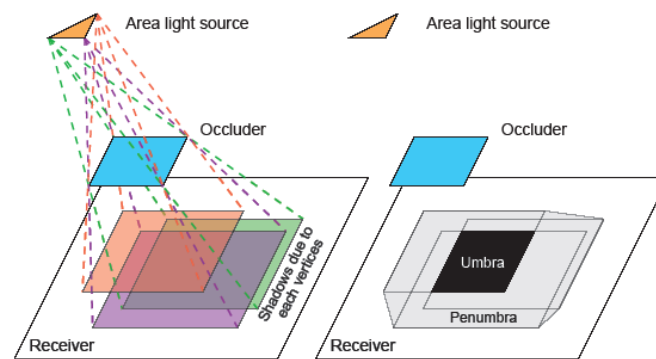
# Четкие и мягкие Hard vs. Soft

Четкие: возможна иллюзия  
дополнительных объектов  
(шарахаться от тени)

Мягкие: оценивание конфигурации  
источника по теням и препятствию



(a) Geometry of hard shadows



(b) Geometry of soft shadows



# Множественные тени

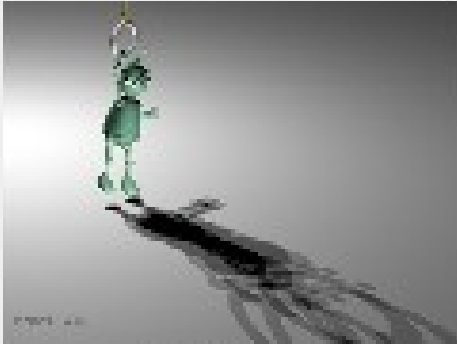


Fig. 1a

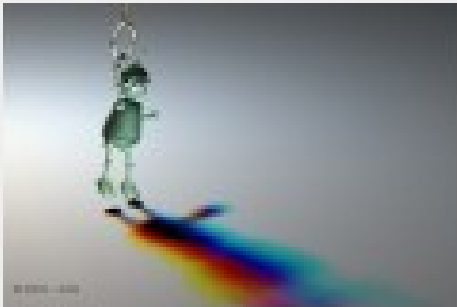


Fig. 1b

- От многих источников
- От многих объектов
- Тень от совокупности объектов, как правило, меньше объединения теней

Такие графики удобны на плоскости, в пространстве приходится иметь дело с многоугольниками (см.далее)

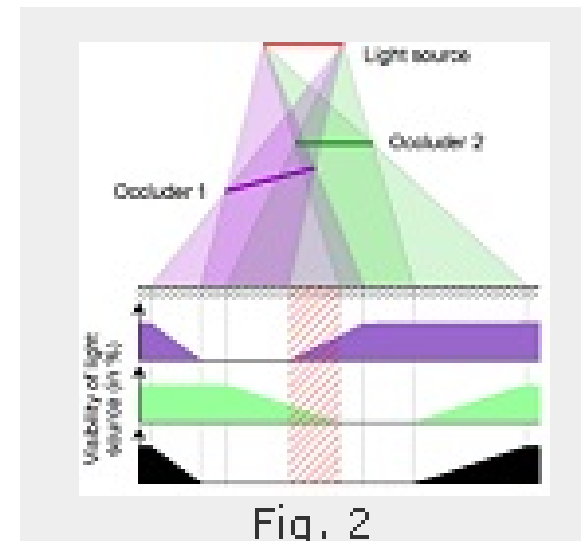
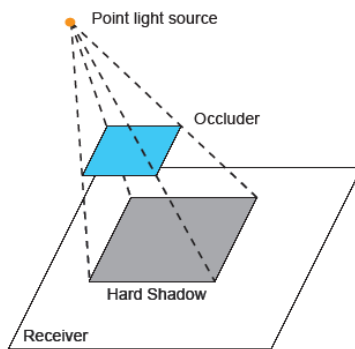


Fig. 2

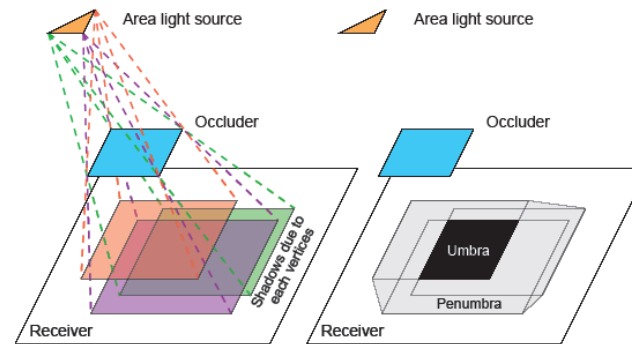
# Чёткие тени

В обычной лучевой трассировке источники света только точечные: тени чёткие, получаются автоматически

В полигональных рендерах тени нужно обрабатывать дополнительно



(a) Geometry of hard shadows



(b) Geometry of soft shadows



# Модель освещенности

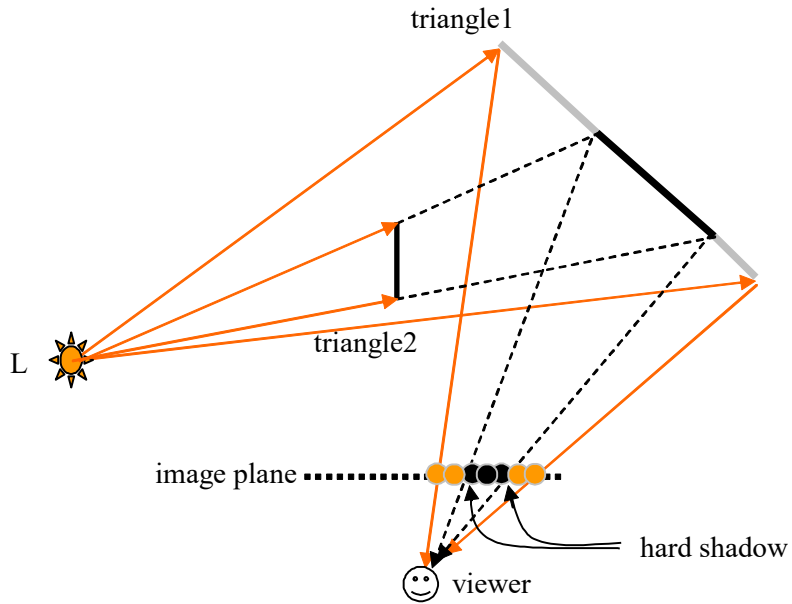
$$I = k_a I_A + \sum_{i=1}^n V_i f_{att,i} I_i [k_d (N \cdot L_i) + k_s (R_i \cdot V)^{pow}] + k_s f_{att,s} I_r$$

$$U_{RT}(P) = \sum_{i=1}^{nL} V(i, P) \Omega(i, P)$$

где  $U_{RT}(P)$  – это искомая закраска,  $V(i, P)$  – видимость (0 или 1)  $i$ -го источника из точки  $P$ , а  $\Omega(i, P)$  учитывает все отражательные свойства поверхности в точке  $P$  и параметры  $i$ -го источника.

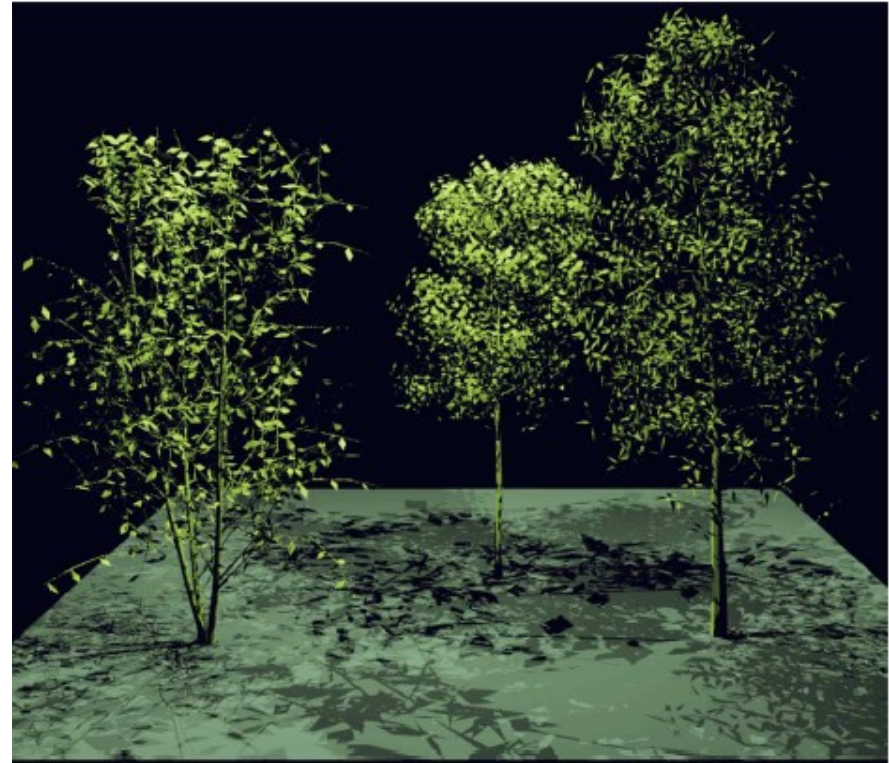


# Ray Tracing



$$U_{RT}(P) = \sum_{i=1}^{nL} V(L_i, P) \Omega(i, P)$$

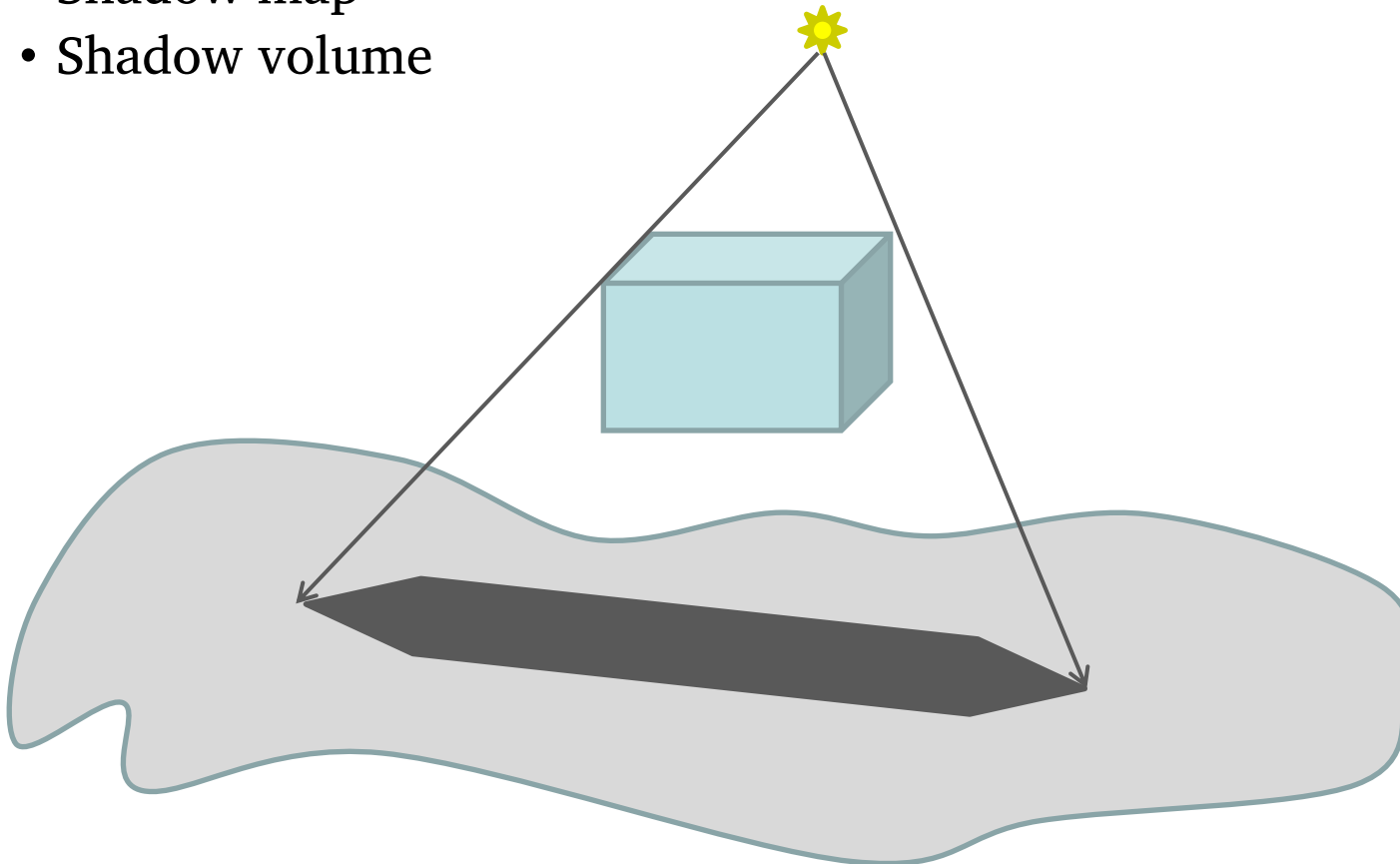
$$V(L_i, P) = \begin{cases} 0 \\ 1 \end{cases}$$



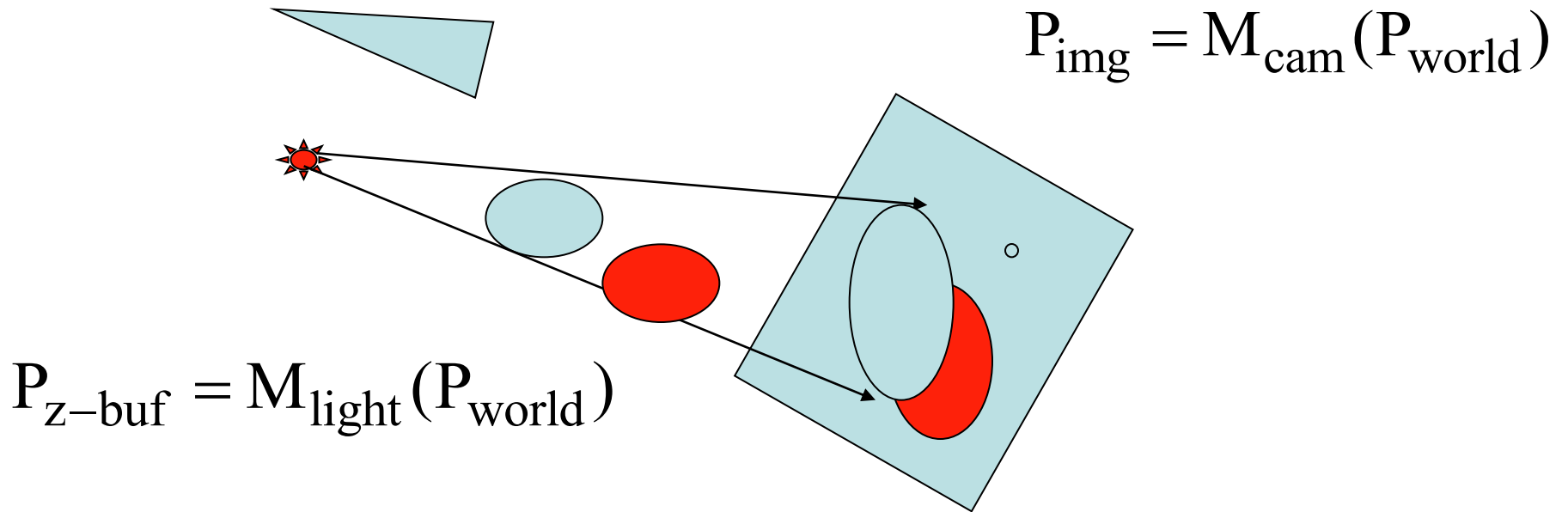
Hard shadows

# Тени в полигональных рендерерах

- Дополнительные объекты
- Shadow map
- Shadow volume



# Shadow map



Несколько источников?

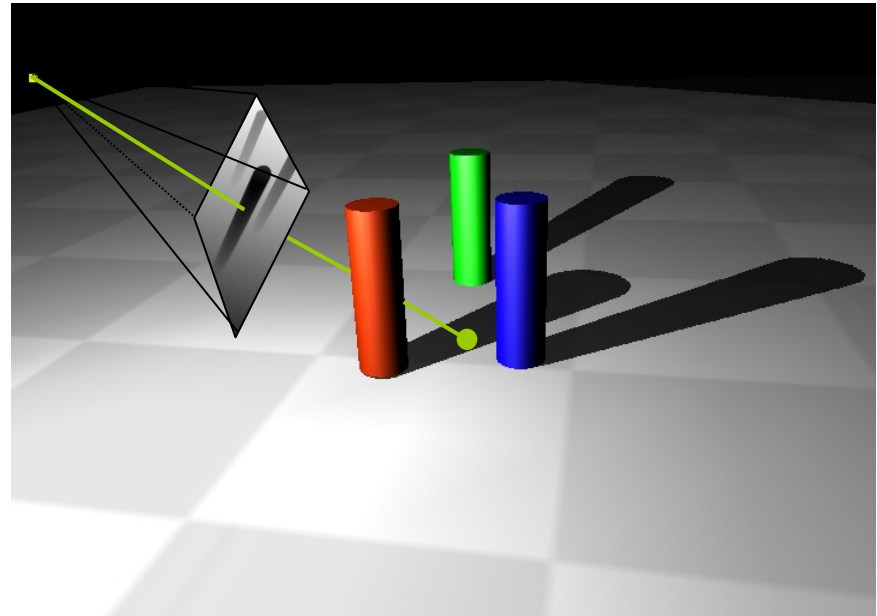
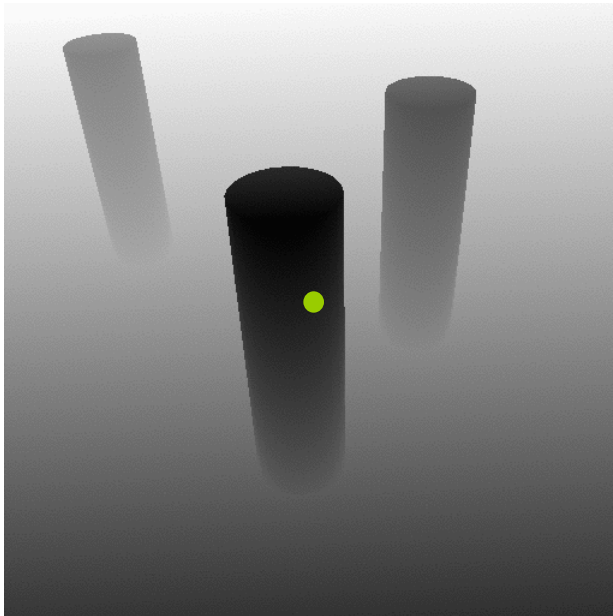
# Shadow Mapping [Williams78]

Image-based method

- Compute shadow map from light view (depth buffer)
- Compute final image
  - compare actual vs. stored depth

$$P_{\text{img}} = M_{\text{cam}}(P_{\text{world}})$$

$$P_{\text{z-buf}} = M_{\text{light}}(P_{\text{world}})$$



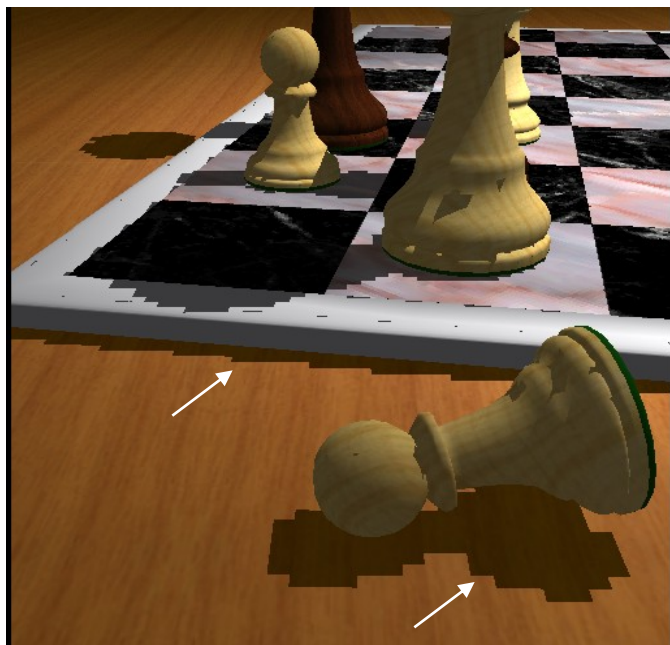
# Shadow mapping



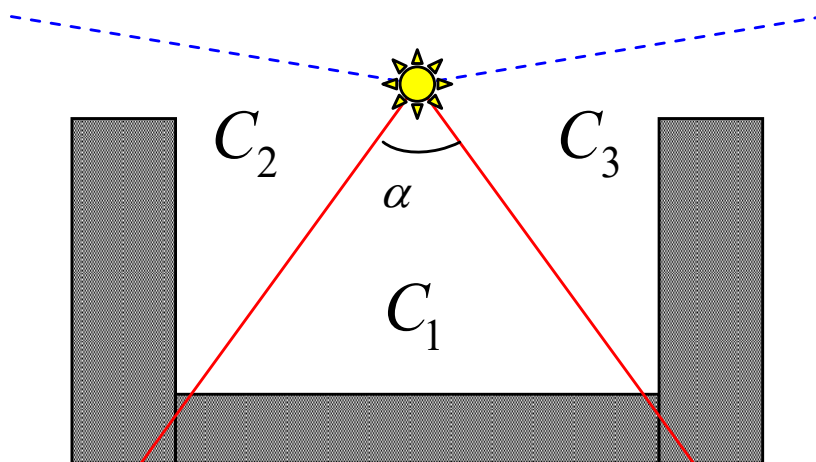
Артефакты

Неровный край

# Shadow maps (проблемы)



Aliasing



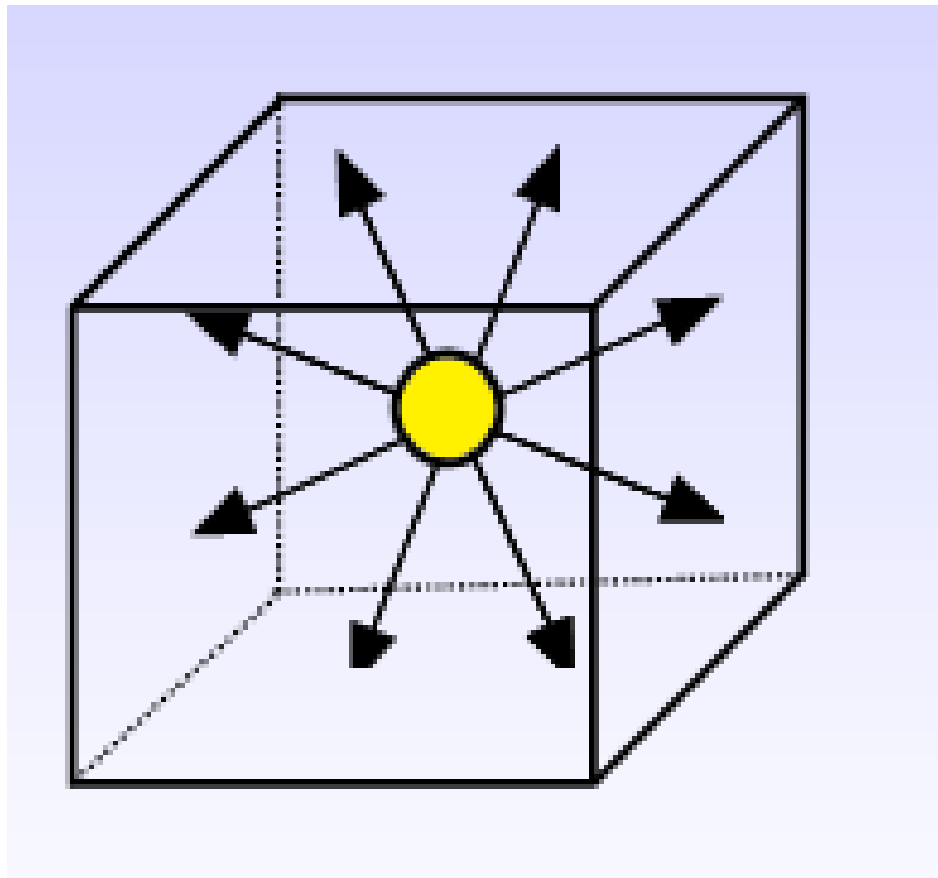
Limited FOV

Shadow map  $n \times m$  = подготовка изображения  $n \times m$

$L$  источников света =  $L$  shadow maps = рендеринг  $L \times n \times m$

Multiple POVs

Сколько z-буферов? 6?





# Shadow volumes

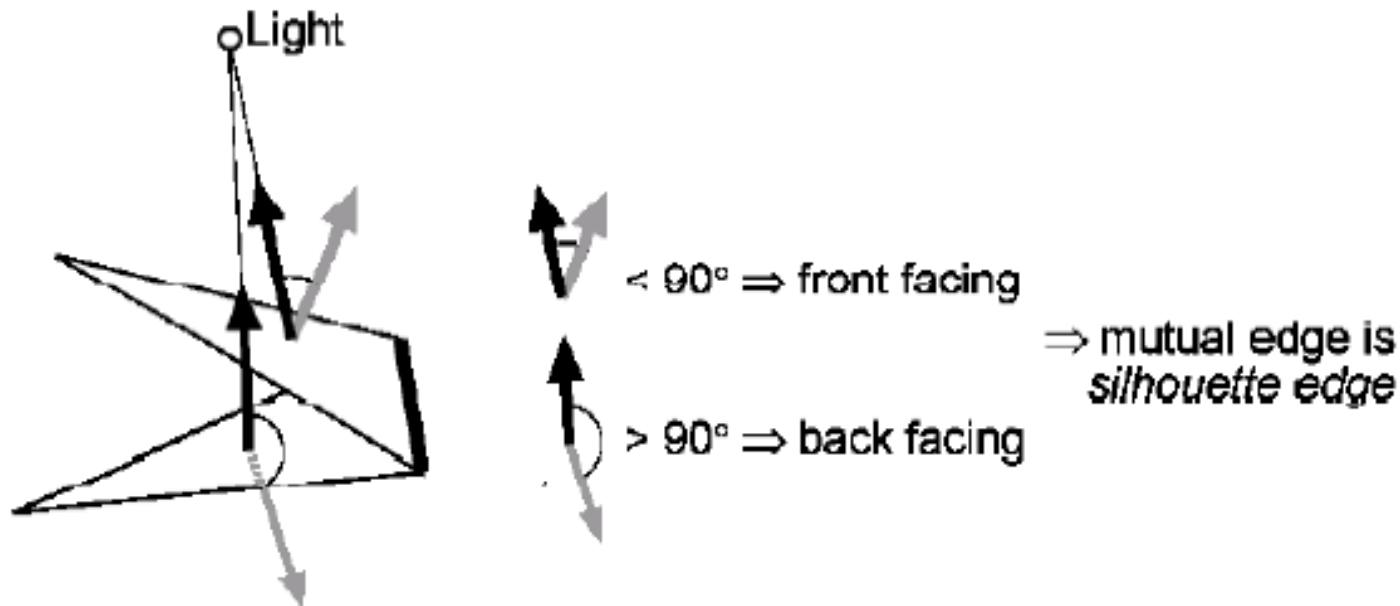




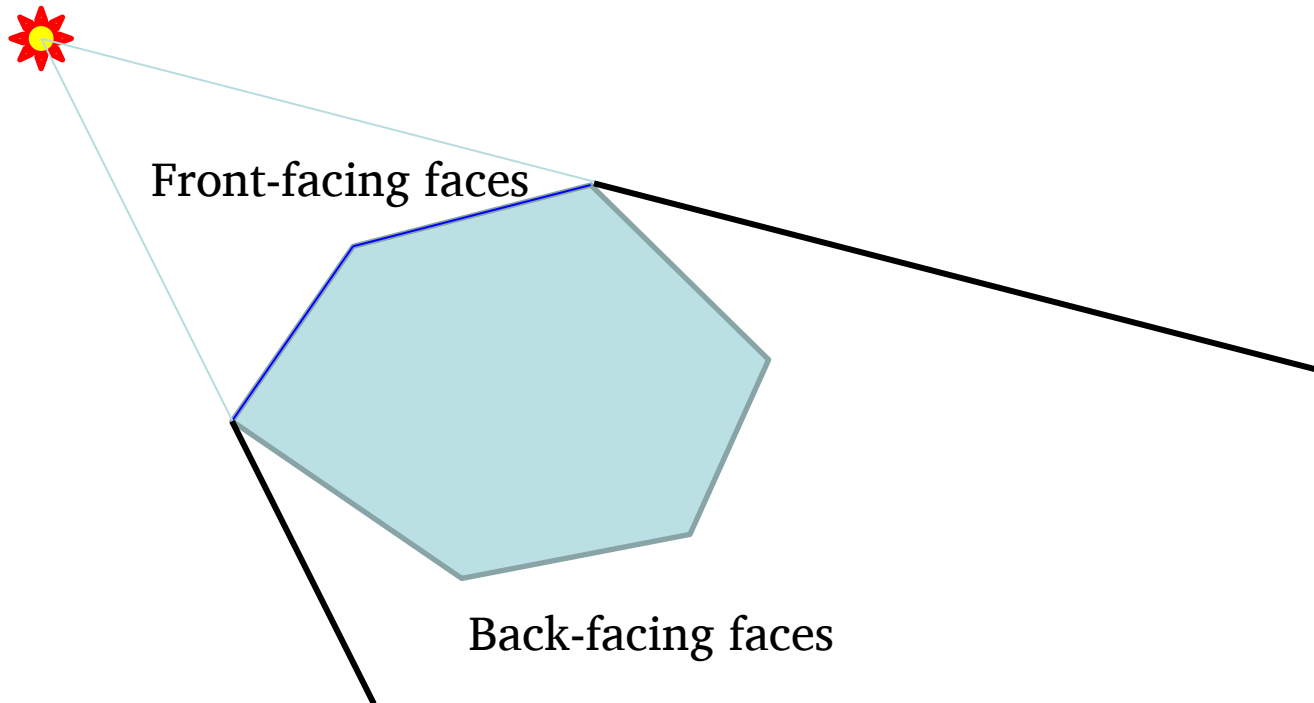
# Shadow volumes

Пусть сцена – это сетка из треугольников

Первый шаг – определение силуэтных ребер  
(ребра на границе видимой и невидимой грани)



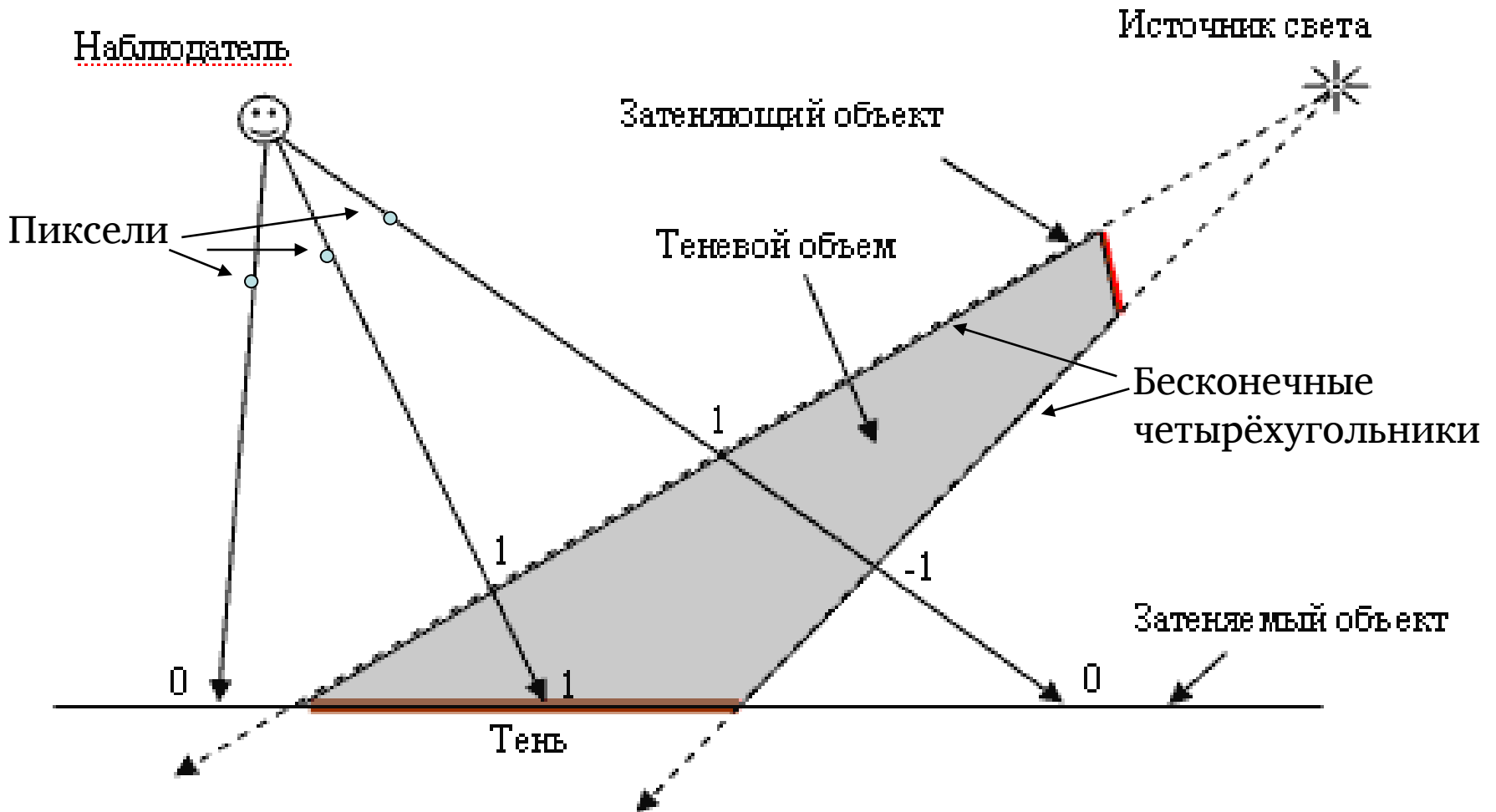
# Shadow volumes



От источника света от всех силуэтных ребер создаются бесконечные прямоугольники из следующих четырех сторон:

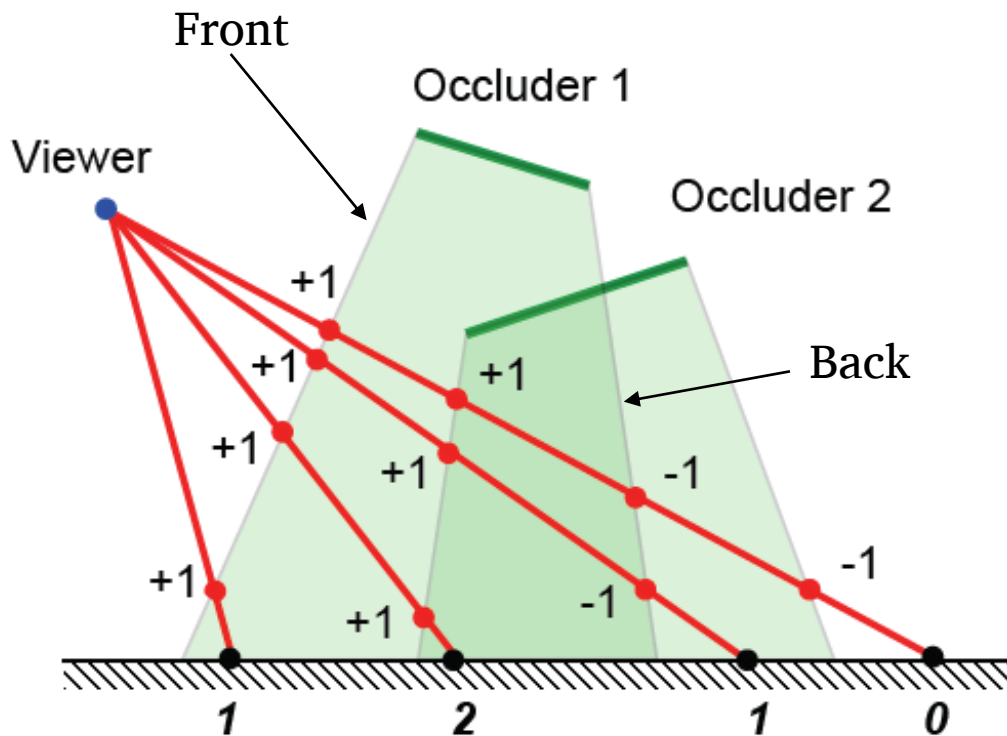
- 1) силуэтное ребро
- 2) луч от одной вершины ребра вдоль источника света
- 3) луч от другой вершины ребра вдоль источника света
- 4) последняя сторона где-то в бесконечности

# Shadow volumes



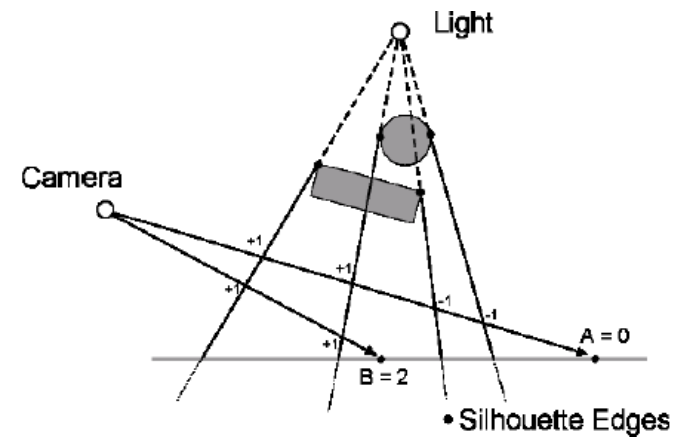
# Shadow volumes

• Light source

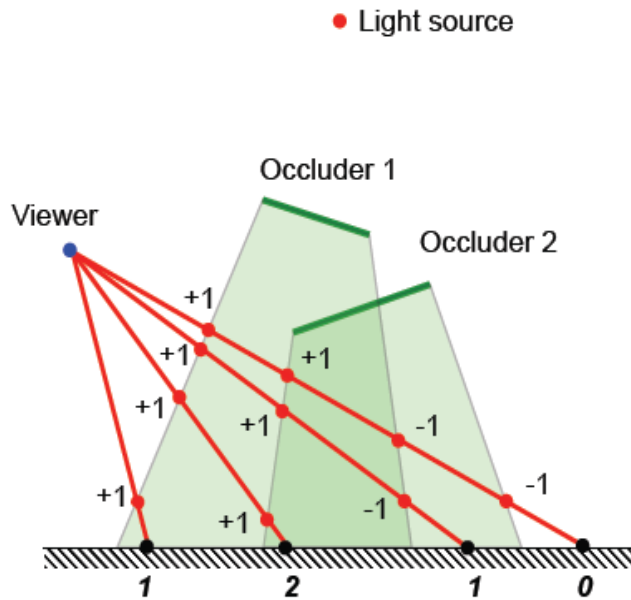


Полигональный рендеринг:

1. Рисуем все только в рассеянном
2. Формируем стенил (z-test)
3. Рисуем только те точки, где в стениле нулевое значение



# Shadow volumes

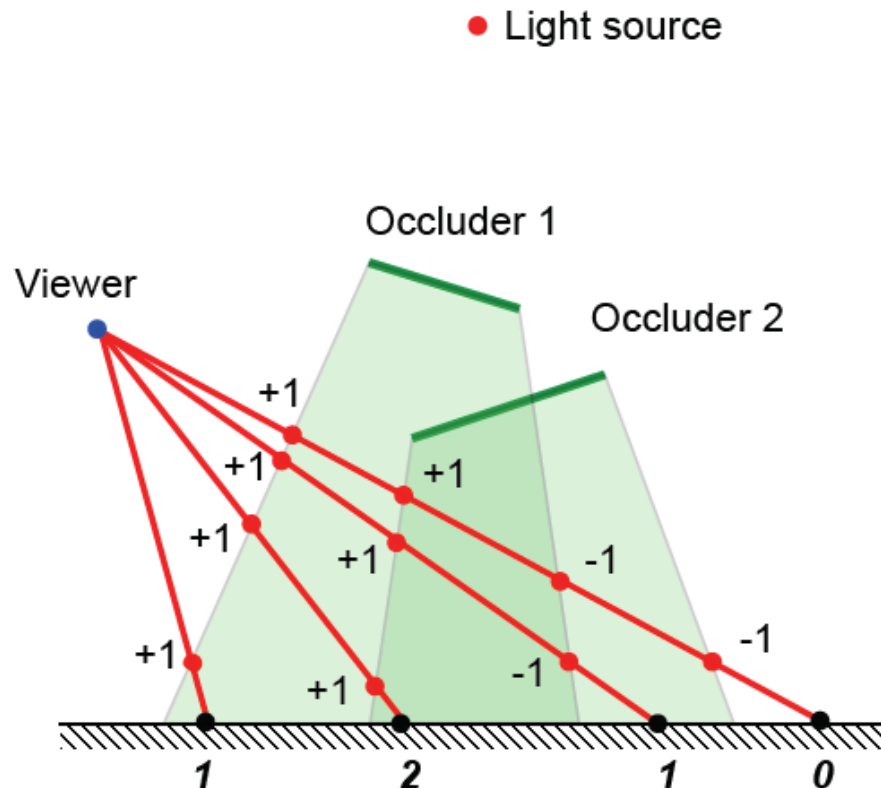


Shadow volumes: области пространства, не видимые от источника света

Hard shadow

# Shadow volumes

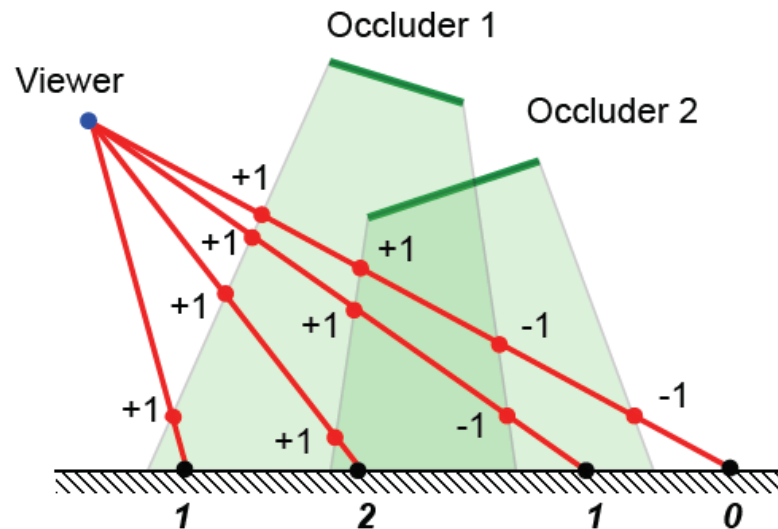
- Сначала стенил-буфер  $\equiv 0$ : все пиксели освещены
- Вся сцена рисуется с точки зрения камеры для получения информации о глубине – Буфер глубины – z-буфер. Вычисления по интенсивности запрещены – только рассеянный.



# Shadow volumes

- Теперь запрет писать в z-буфер
- Рендеринг всех 4-угольников передних (front-facing):  
+1 в стенил-буфер, для тех, что прошли тест глубины
- Рендеринг всех 4-угольников задних (backfacing):  
-1 в стенил-буфер, для тех, что прошли тест глубины
- На современных ГПУ это можно сделать одновременно

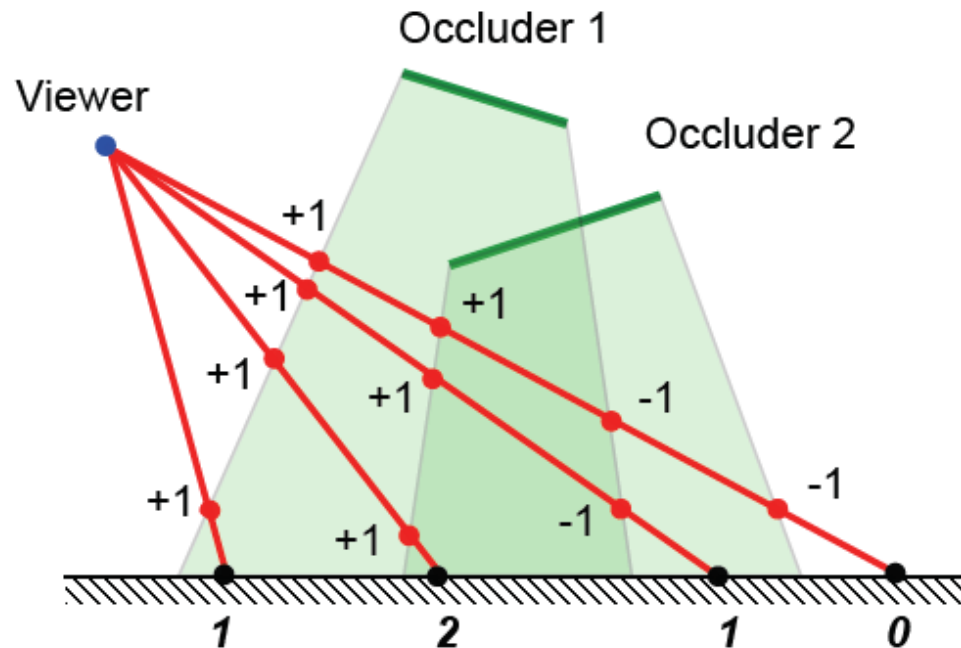
● Light source



# Shadow volumes

- Итоговая сцена рендерится снова с разрешенным расчетом освещенности но только для пикселей, чье значение в стенси́л-буфере = 0.
- Можно сказать, что «теневая маска» идеально совпадает со сценой
- *А много источников?*

● Light source

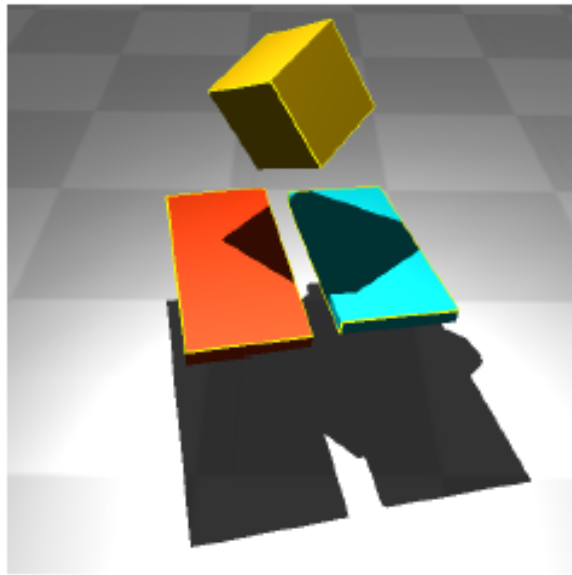




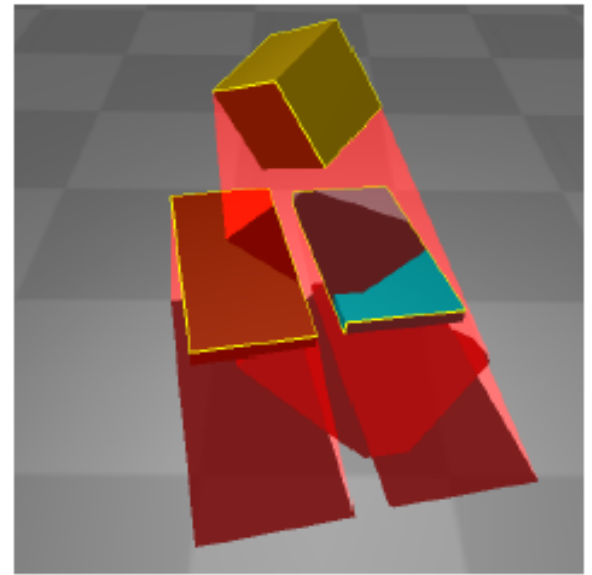
# Shadow volumes



(a) final image

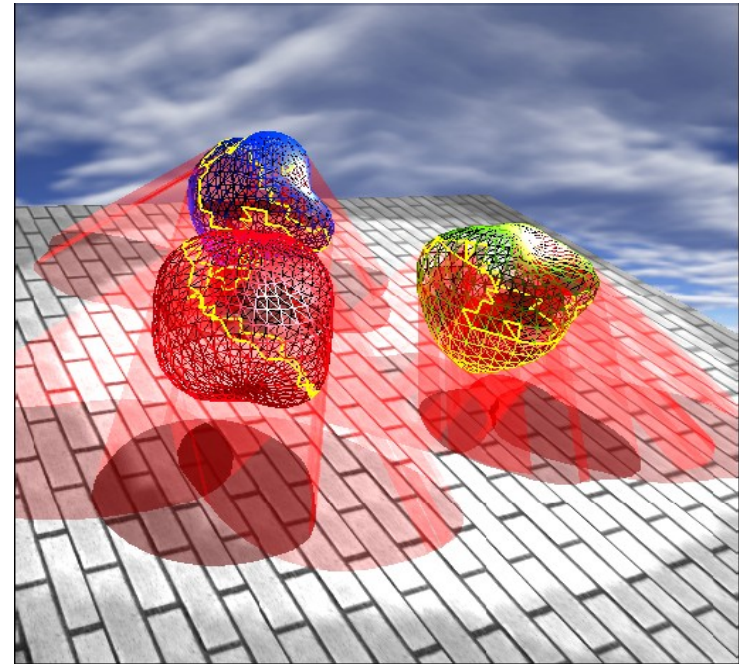
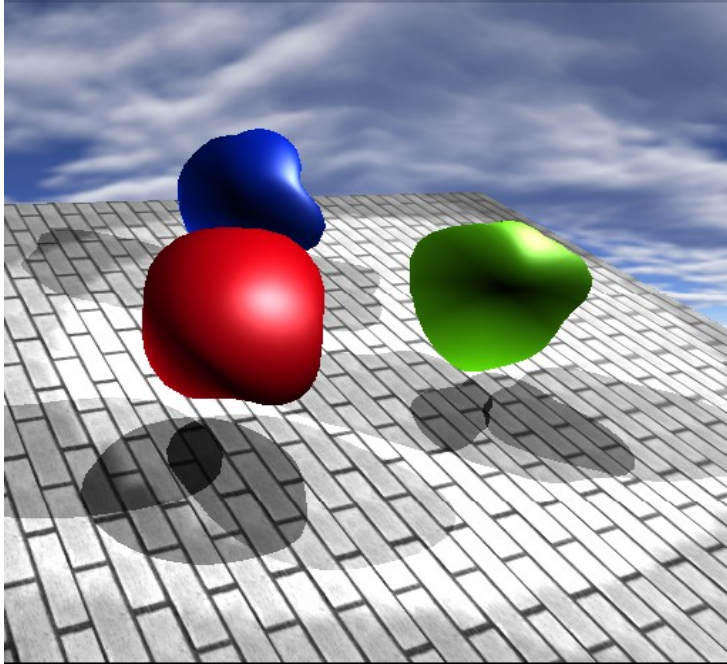


(b) silhouette edges



(c) shadow volumes

# Shadow volumes (problems)



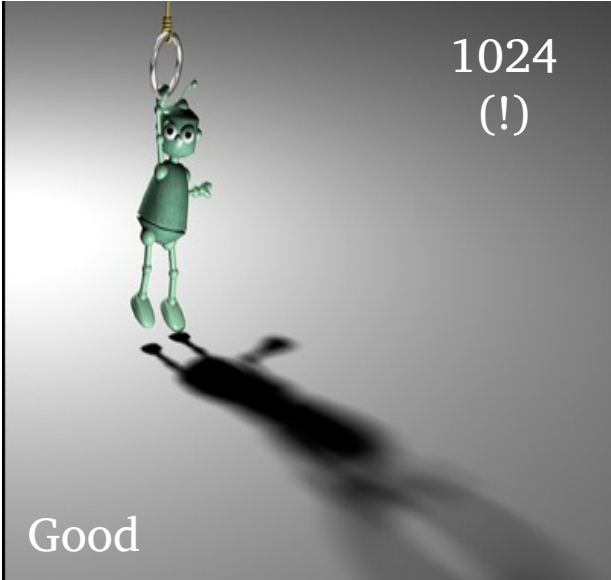
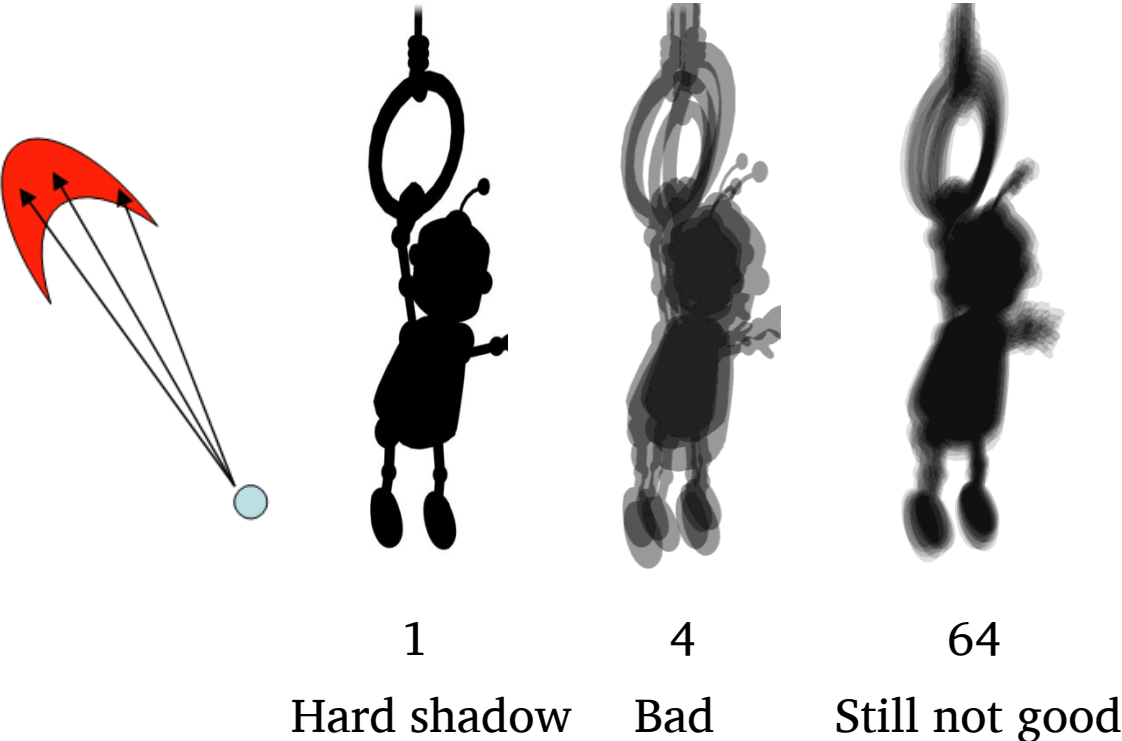
Shadow volumes: дополнительная геометрия, охватывающая большие части видимой сцены = дополнительное время генерации этой геометрии и рендеринга теневой маски

N объектов = N shadow volumes

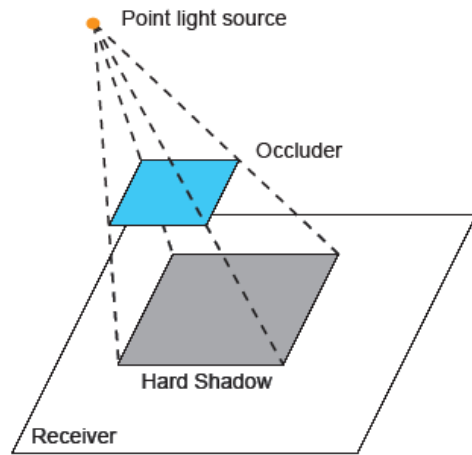
L источников =  $N \times L$  shadow volumes

# Soft shadows simulation in RT

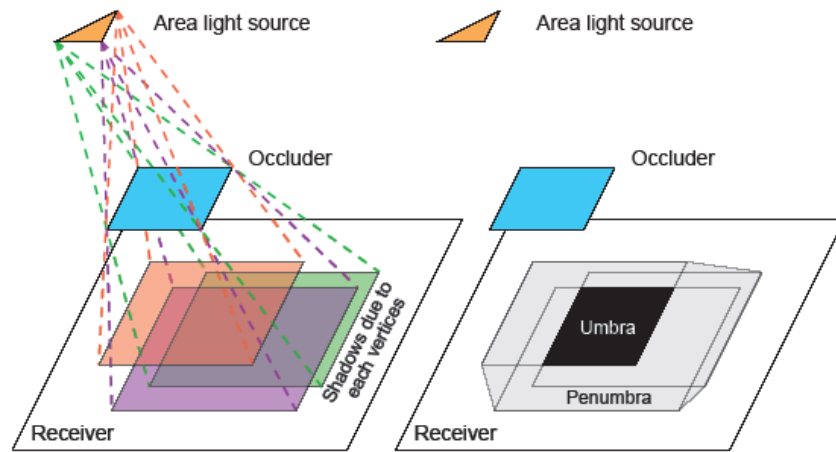
Наивный подход  
(много точечных источников)



Для мягких теней надо определять  
видимую долю источника



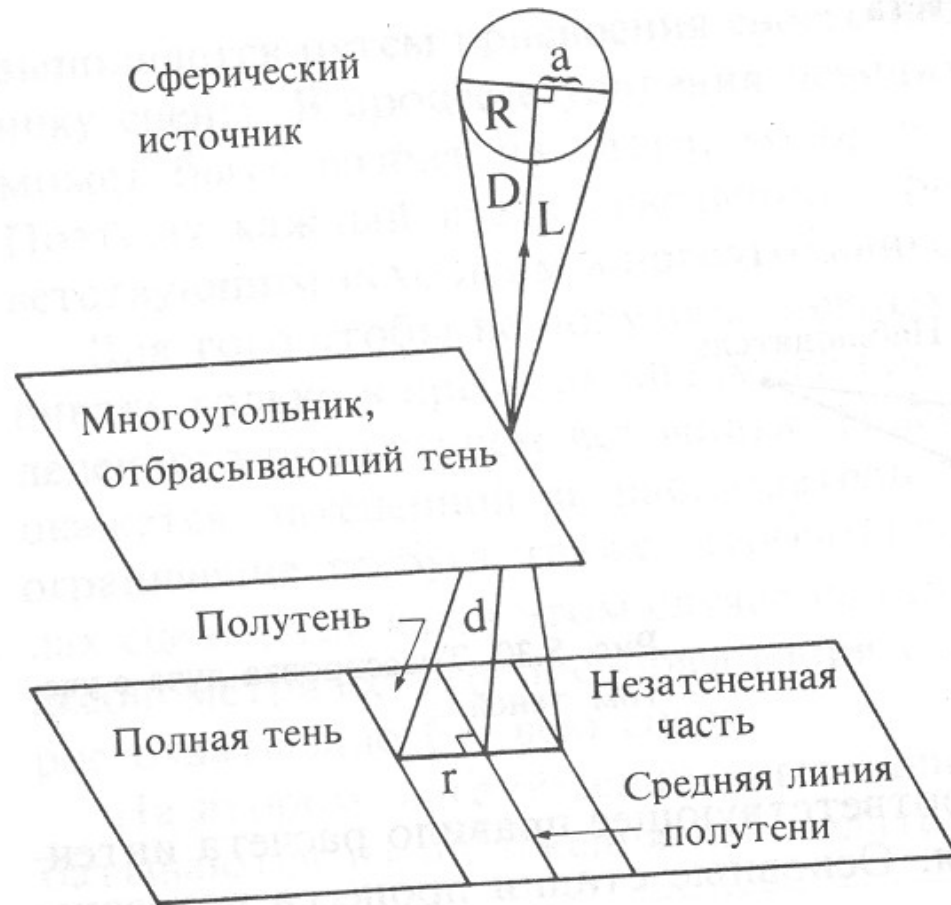
(a) Geometry of hard shadows



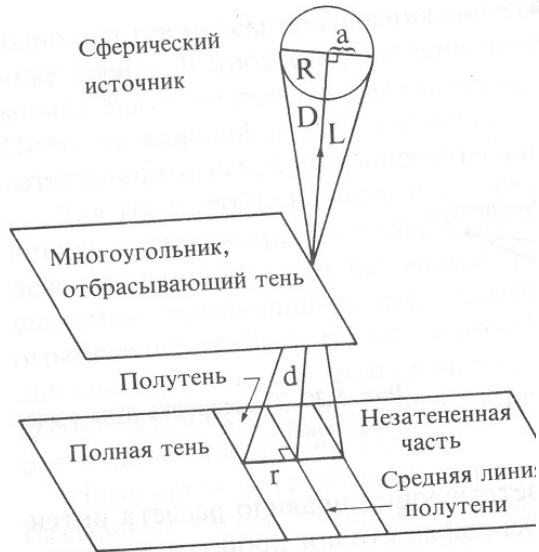
(b) Geometry of soft shadows



# Тени от сферического источника (см. Роджерса)



# Тени от сферического источника



предположим, что точечный источник находится в центре сферы. С помощью подобных треугольников найдем проекцию половины ширины полутени  $r$  на направление  $\mathbf{L}$  (рис. 5.31):

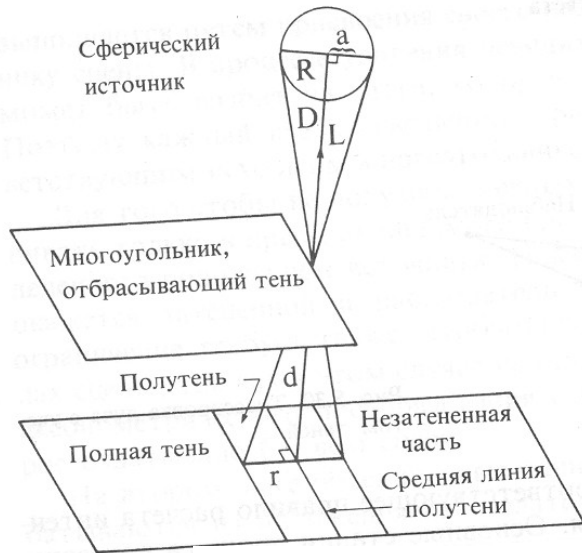
$$\frac{r(\mathbf{n} \cdot \mathbf{L})}{d} = \frac{R}{D}$$

где  $d$  — расстояние от точки, отбрасывающей тень, до соответствующей точки на средней линии полутени;  $D$  — расстояние от точки, отбрасывающей тень, до центра сферического источника;  $R$  — радиус сферического источника.

Если смотреть от многоугольника, отбрасывающего тень, то телесный угол источника  $d\omega$  есть

$$d\omega = \pi \left( \frac{R}{D} \right)^2$$

# Тени от сферического источника

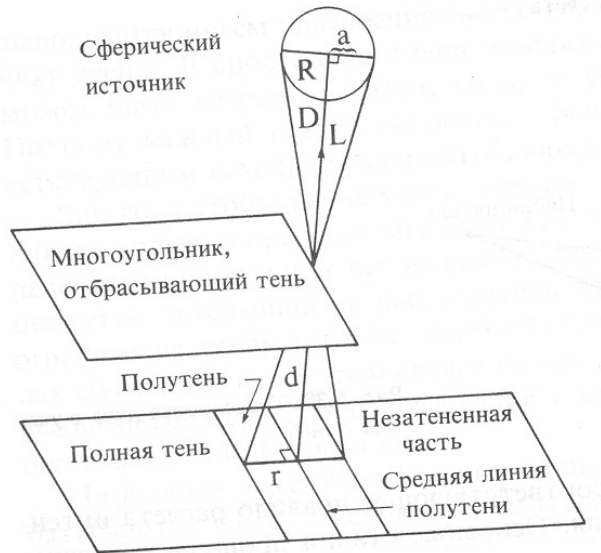


ПОЭТОМУ ПОЛОВИНА ШИРИНЫ ПОЛУТЕНИ РАВНА

$$r = \frac{d}{\hat{n} \cdot \hat{L}} \frac{R}{D} = \frac{d}{\hat{n} \cdot \hat{L}} \sqrt{\frac{d\omega}{\pi}}$$

Это означает, что если телесный угол источника меньше, то он создает более резкую тень (т. е. с меньшим  $r$ ). У точечного источника  $d\omega = 0$  и  $r = 0$ , поэтому полутени нет вообще. При сближении затеняемой и затеняющей поверхности  $d$  и  $r$  уменьшаются, и тень становится резче.

# Тени от сферического источника

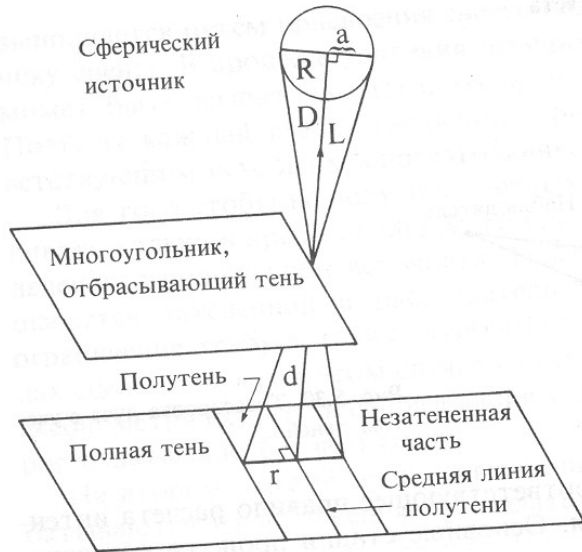


Интенсивность точек полутени определяется видимой частью источника. Для сферического источника, частично видимого от  $-R$  до  $a$ , эта доля составляет

$$A_{frac} = \frac{1}{\pi R^2} \int_{-R}^a 2\sqrt{R^2 - x^2} dx = \frac{1}{2} + \frac{1}{\pi} \left[ \frac{a}{R} \sqrt{1 - \left(\frac{a}{R}\right)^2} + \sin^{-1} \left(\frac{a}{R}\right) \right]$$



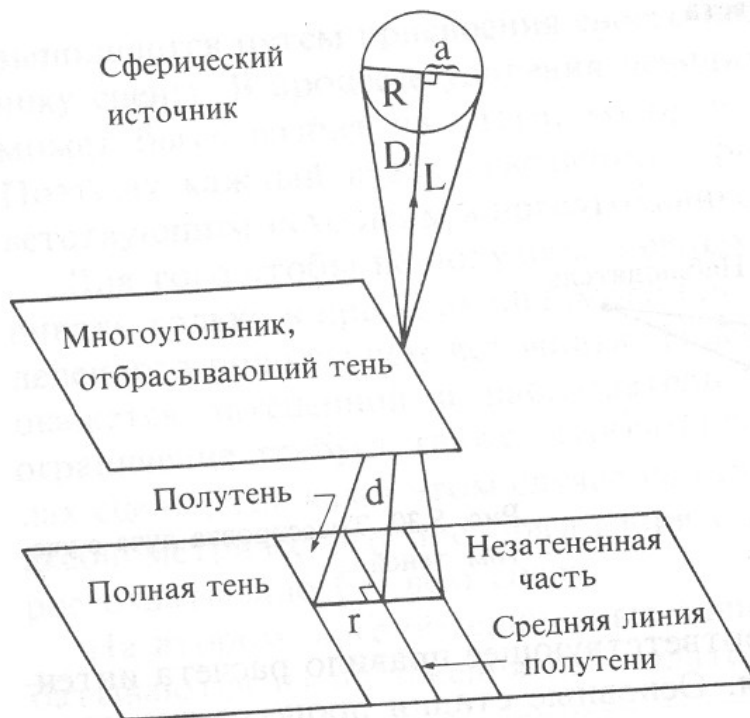
# Тени от сферического источника



$$A_{frac} = \frac{1}{\pi R^2} \int_{-R}^a 2\sqrt{R^2 - x^2} dx = \frac{1}{2} + \frac{1}{\pi} \left[ \frac{a}{R} \sqrt{1 - \left(\frac{a}{R}\right)^2} + \sin^{-1} \left(\frac{a}{R}\right) \right]$$

Результаты показывают, что на одном краю полутень получается более четкой. Кук рекомендует хранить эти данные в таблице цветов. Однако проще рассчитывать линейную аппроксимацию

# Тени от сферического источника



$$A_{frac} = \frac{1}{2} \left( 1 + \frac{a}{R} \right)$$

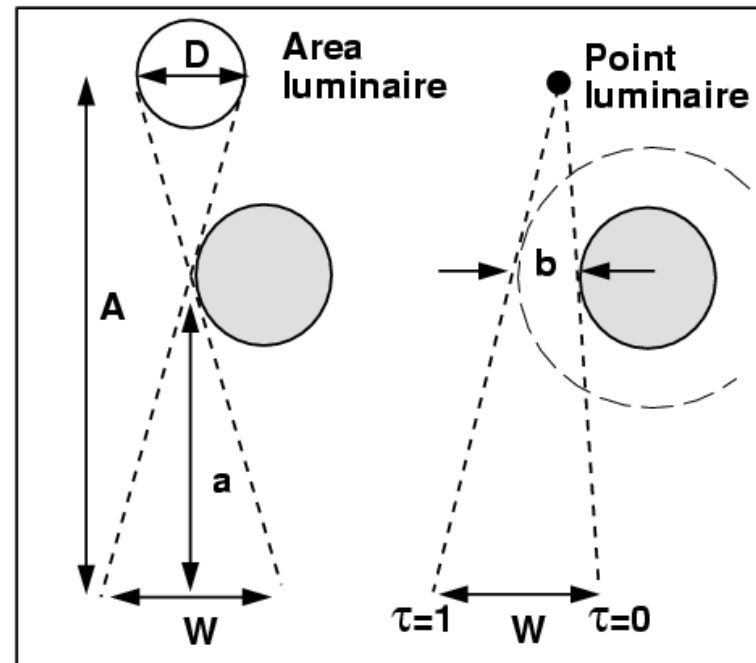
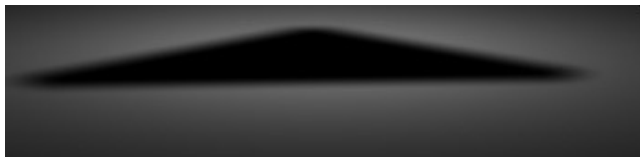
которая дает погрешность меньше 7%.

Монотонна по а

# Мягкие тени

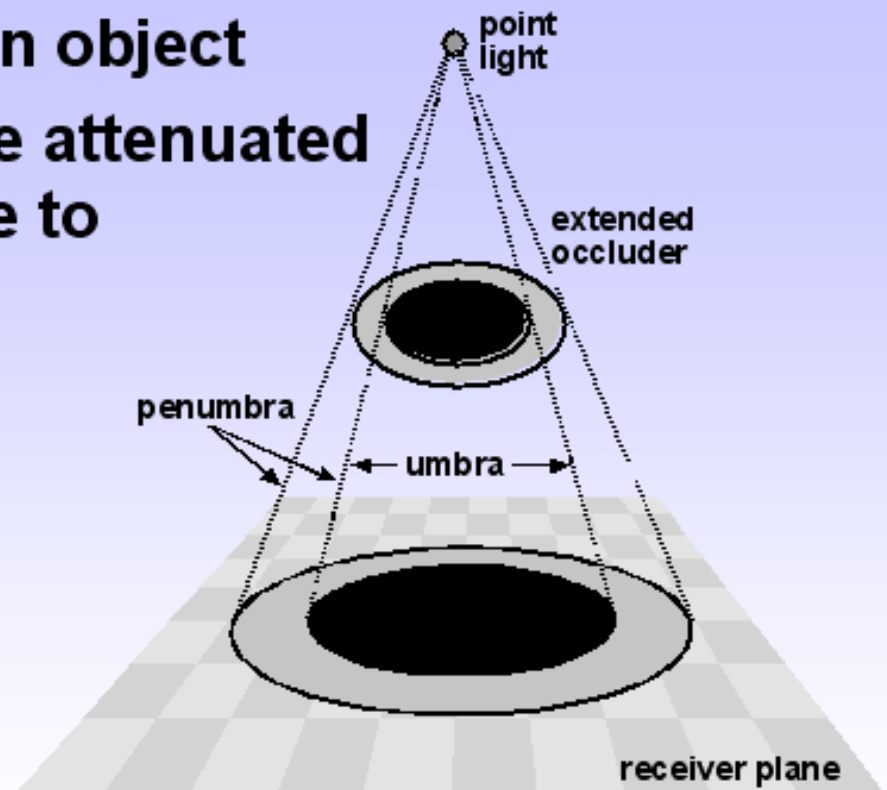
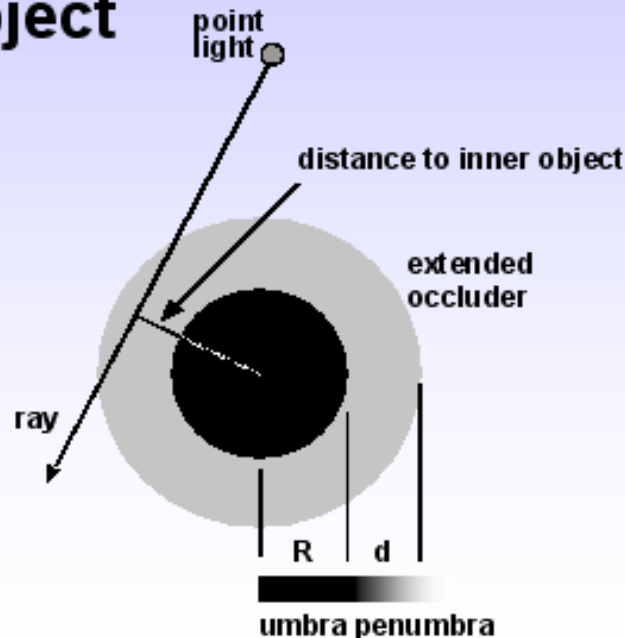
P. Shirley (1999) : extra 20%

## Тени от мелких объектов

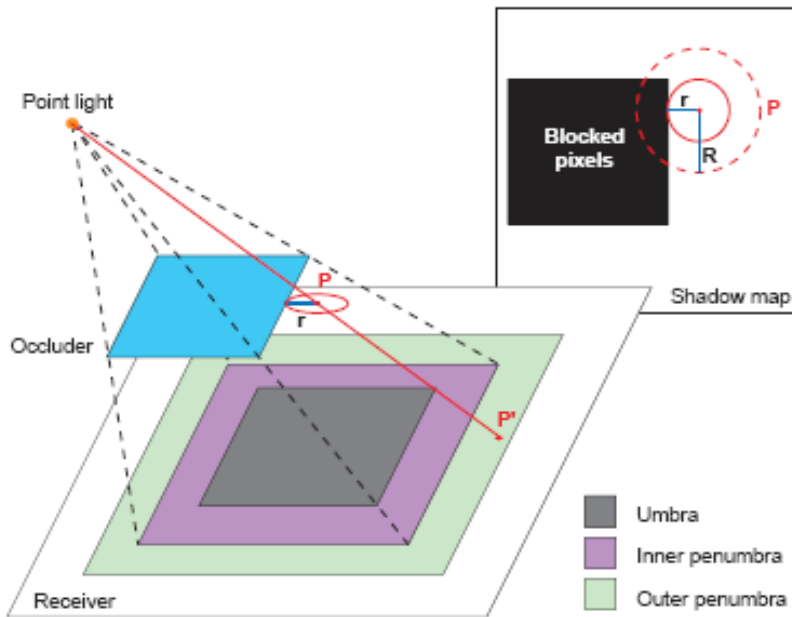


# Single Sample Soft Shadows

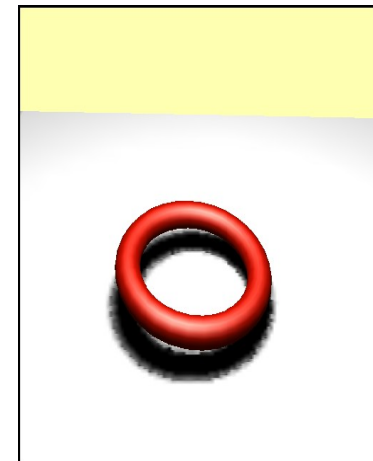
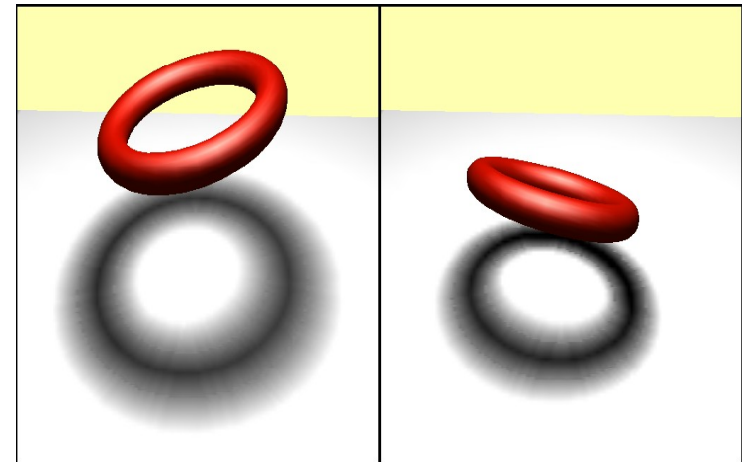
- Based on [Parker98] method for ray tracing
  - Adding an outer hull to an object
  - Rays inside outer hull are attenuated according to the distance to object



# Single sample soft shadows



Based on shadow maps

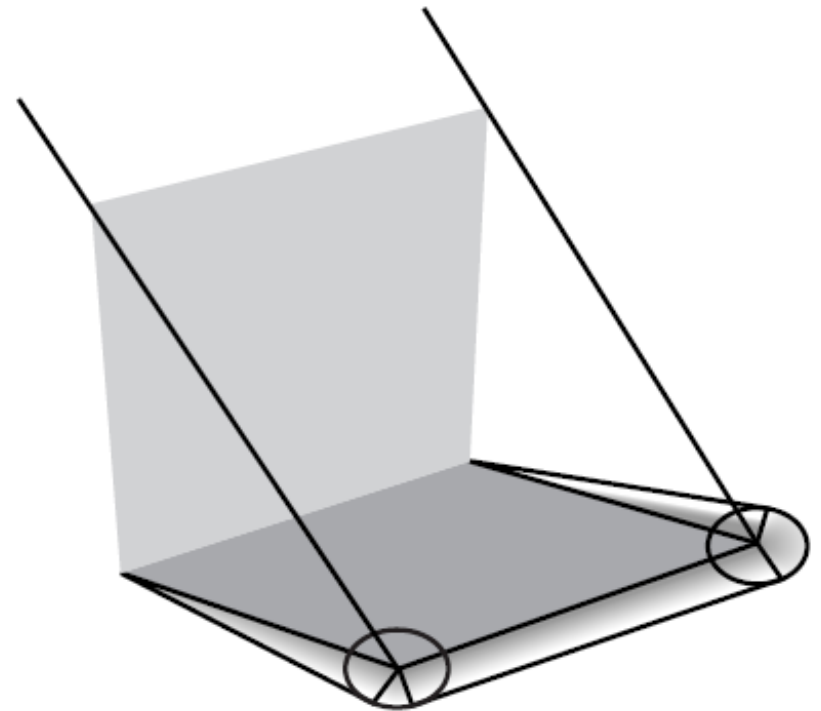
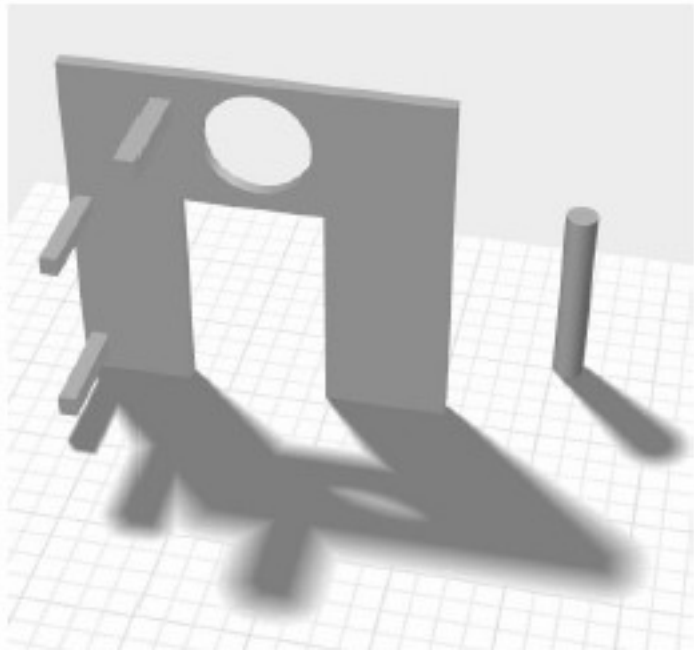
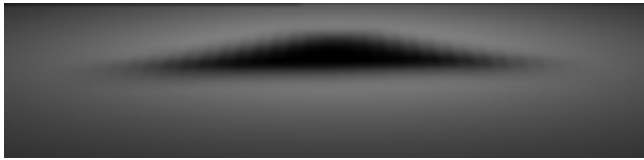


Soft shadows

# Мягкие тени

Eric Haines (2001)  $\Rightarrow$  Soft Planar Shadows

Тени от мелких объектов

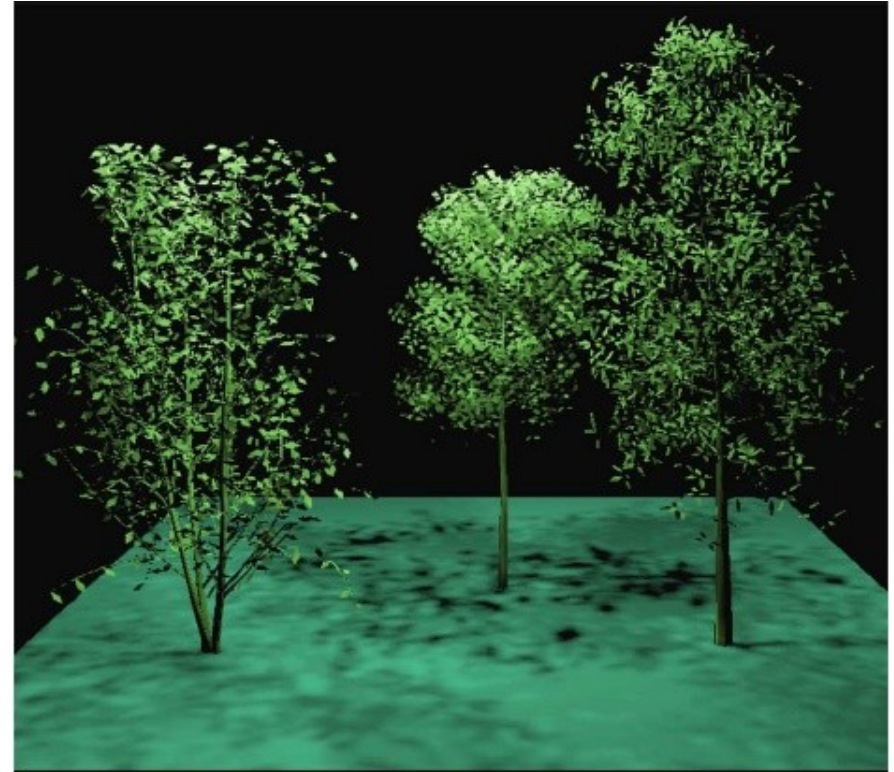
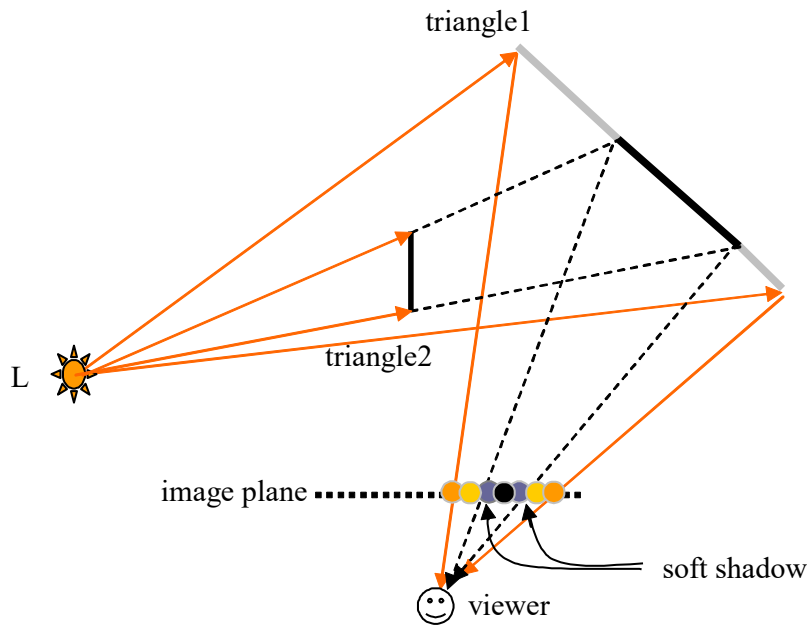


# Метод световых меток (МСС)

Вся сцена разбивается равномерной трехмерной сеткой, в узлах которой вычисляется видимость каждого из источников света. Соответственно, в каждой точке видимость источника аппроксимируется по ближайшим узлам сетки:

- Модификация ОРЛТ
- Аппроксимация решения ОРЛТ
- Непрозрачные поверхности (диффузное и зеркальное отражения)
- Нет взаимодействующей среды
- Точечные источники, рассеянный свет
- Мягкие тени
- Ускорение расчета изображений сложных сцен (число примитивов, число источников, большое разрешение)

# Light Meshes Method

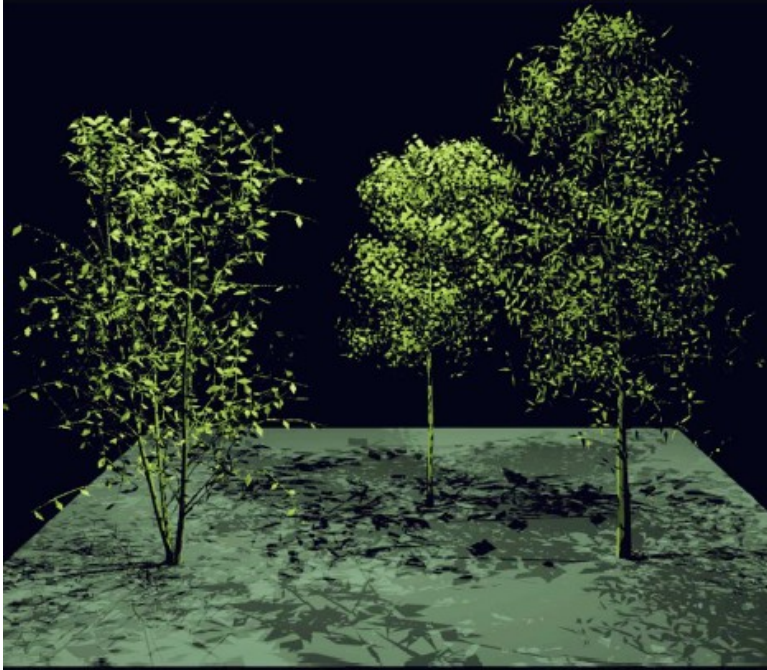


Soft shadows

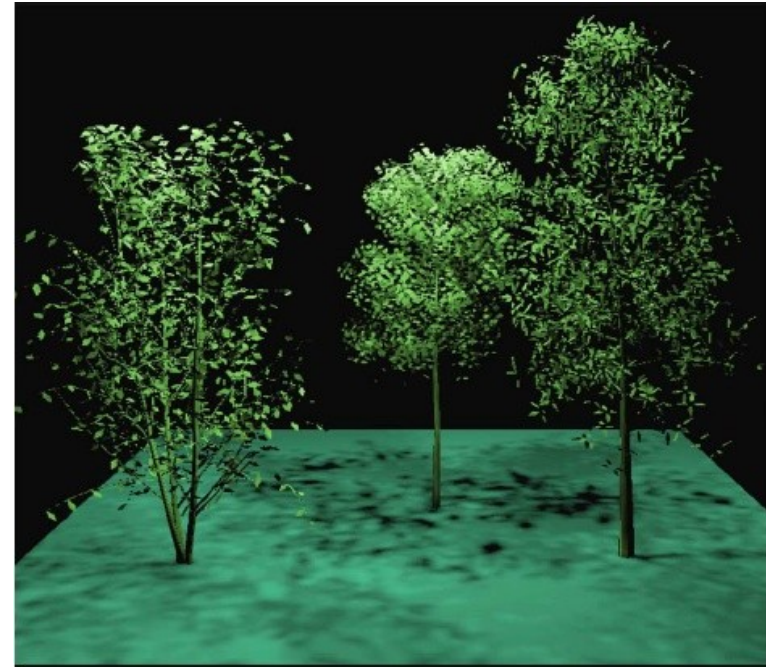
$$\begin{aligned}
 U_{LM}(P) &= \frac{1}{m} \sum_{k=1}^m \left[ \sum_{i=1}^{nL} V(L_i, x_k) \Omega(i, P) \right] = \\
 &= \sum_{i=1}^{nL} \left[ \left( \frac{1}{m} \sum_{k=1}^m V(L_i, x_k) \right) \Omega(i, P) \right] \implies 0 \leq \frac{1}{m} \sum_{k=1}^m V(L_i, x_k) \leq 1
 \end{aligned}$$



# Hard and soft shadows



Hard shadows



Soft shadows

- More realistic
- More computationally expensive

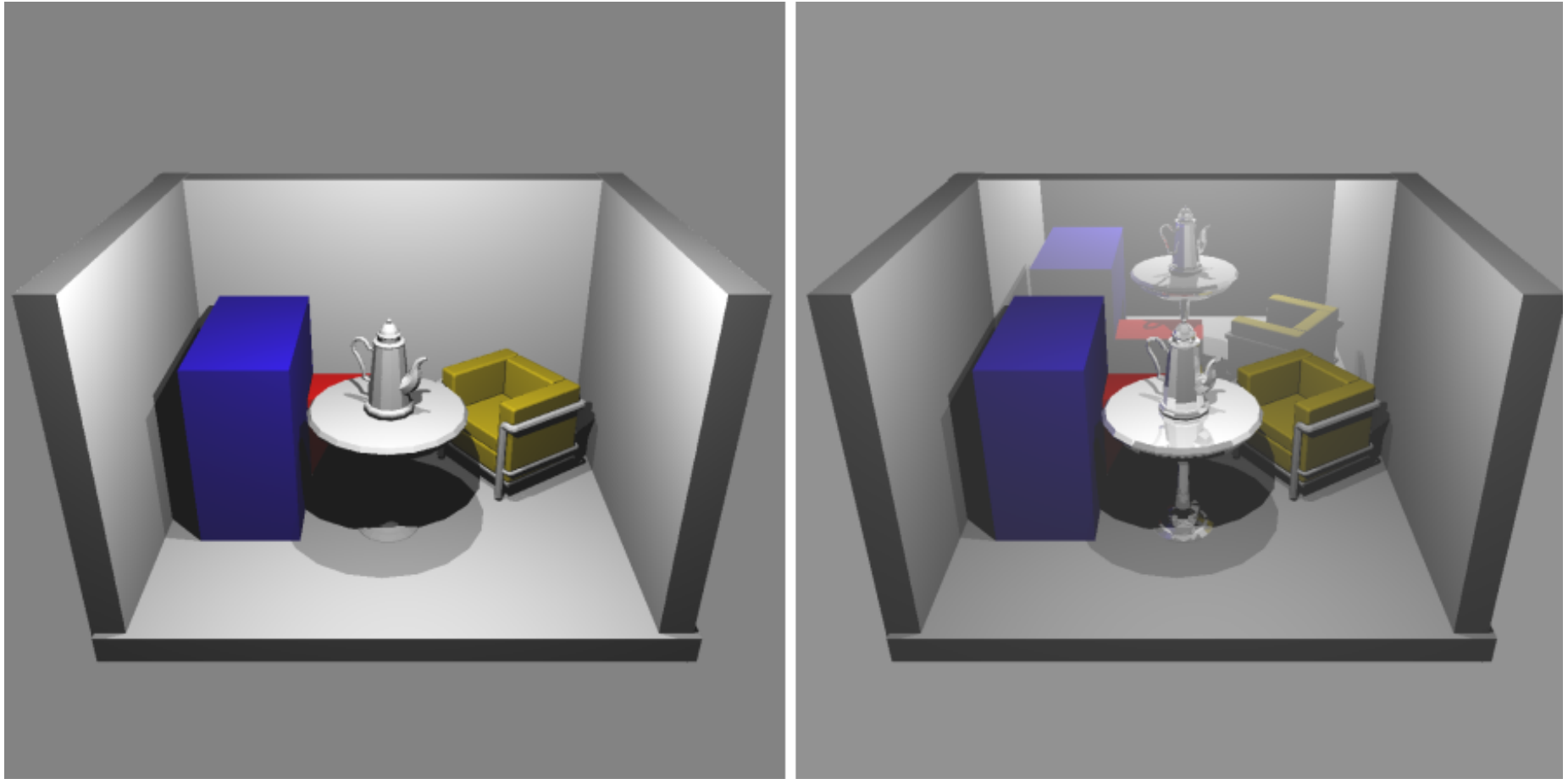


Fig. A1. RRTA without light meshes, diffuse surfaces (left) and specular surfaces (right)

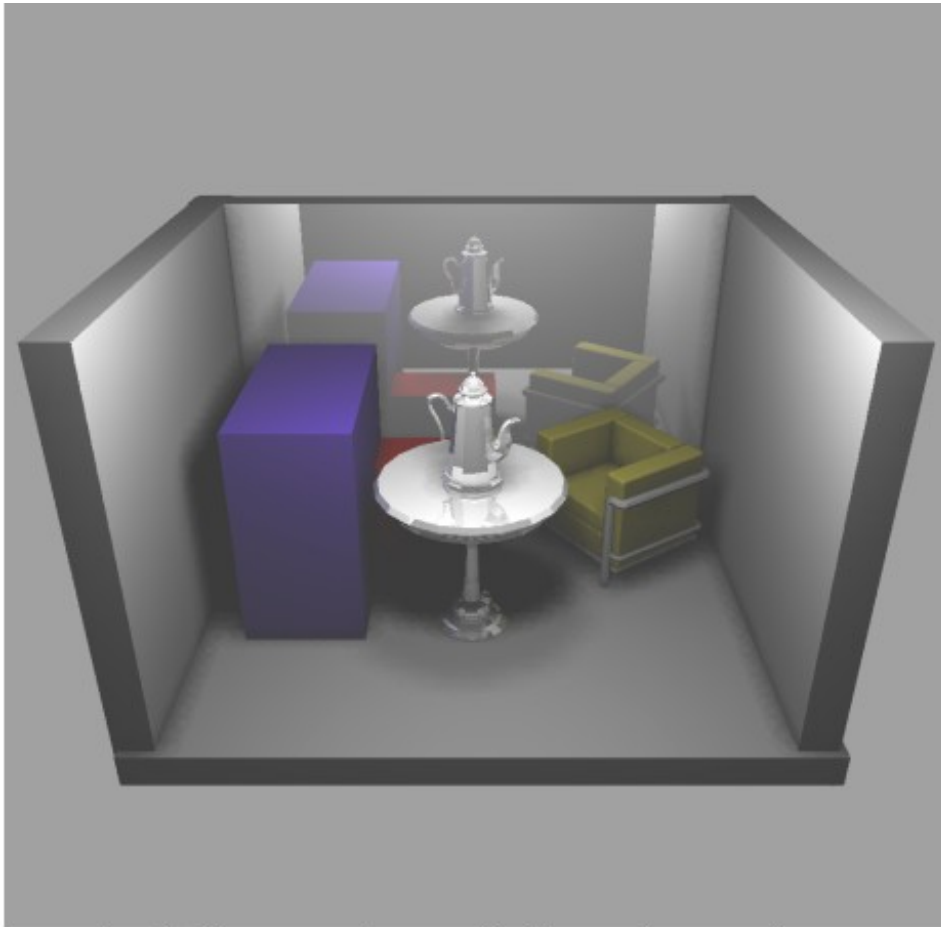
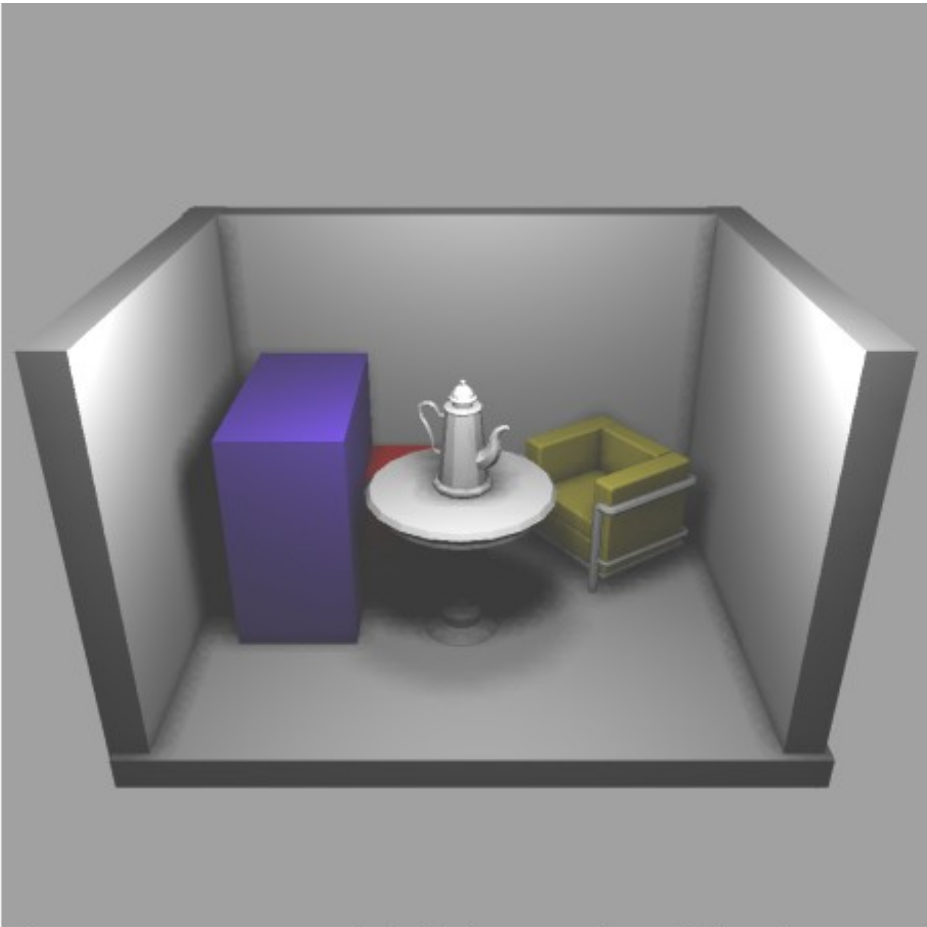


Fig. A2. RRTA with light meshes (density 50x50x50), diffuse surfaces (left) and specular surfaces (right)

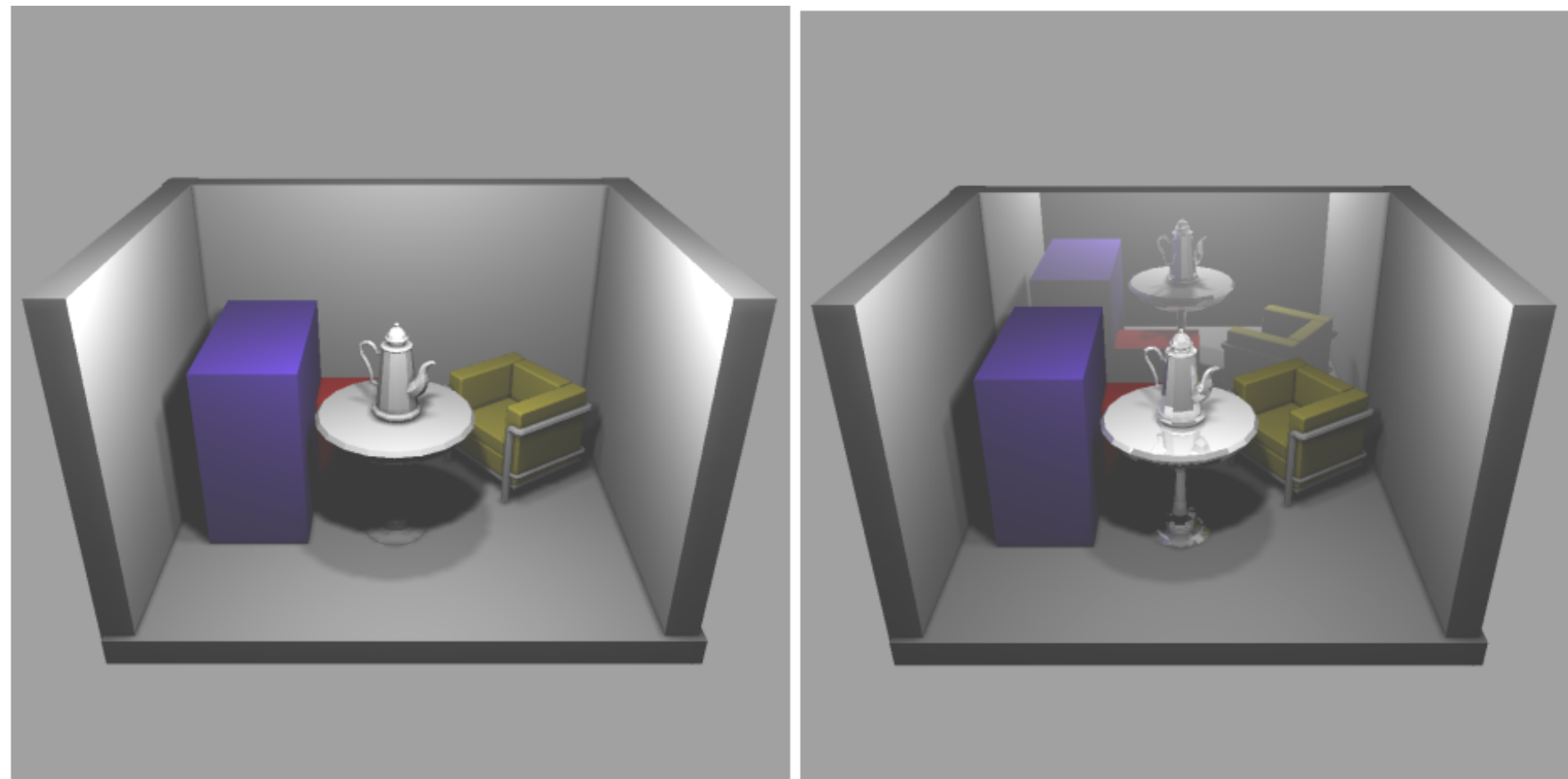
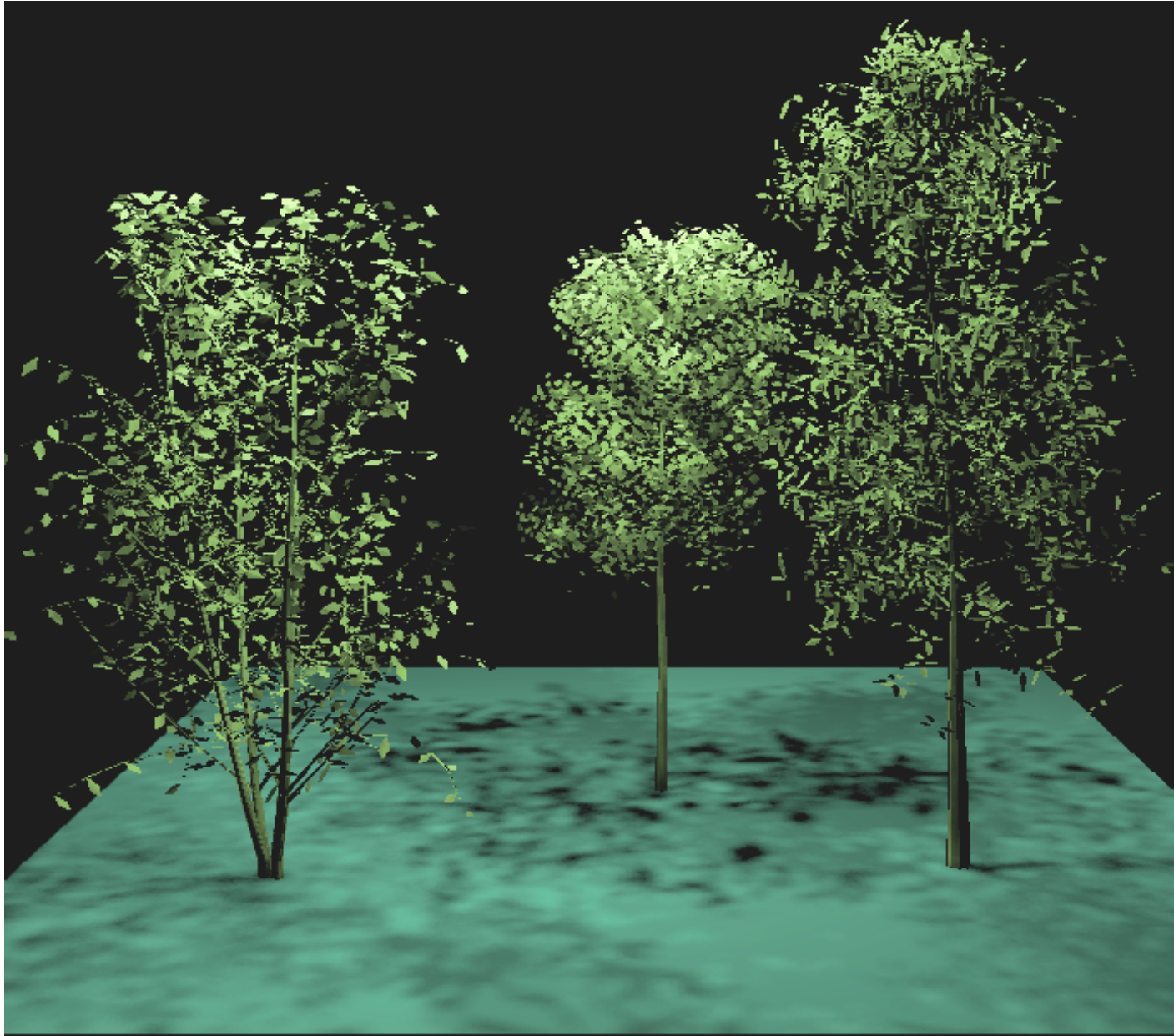


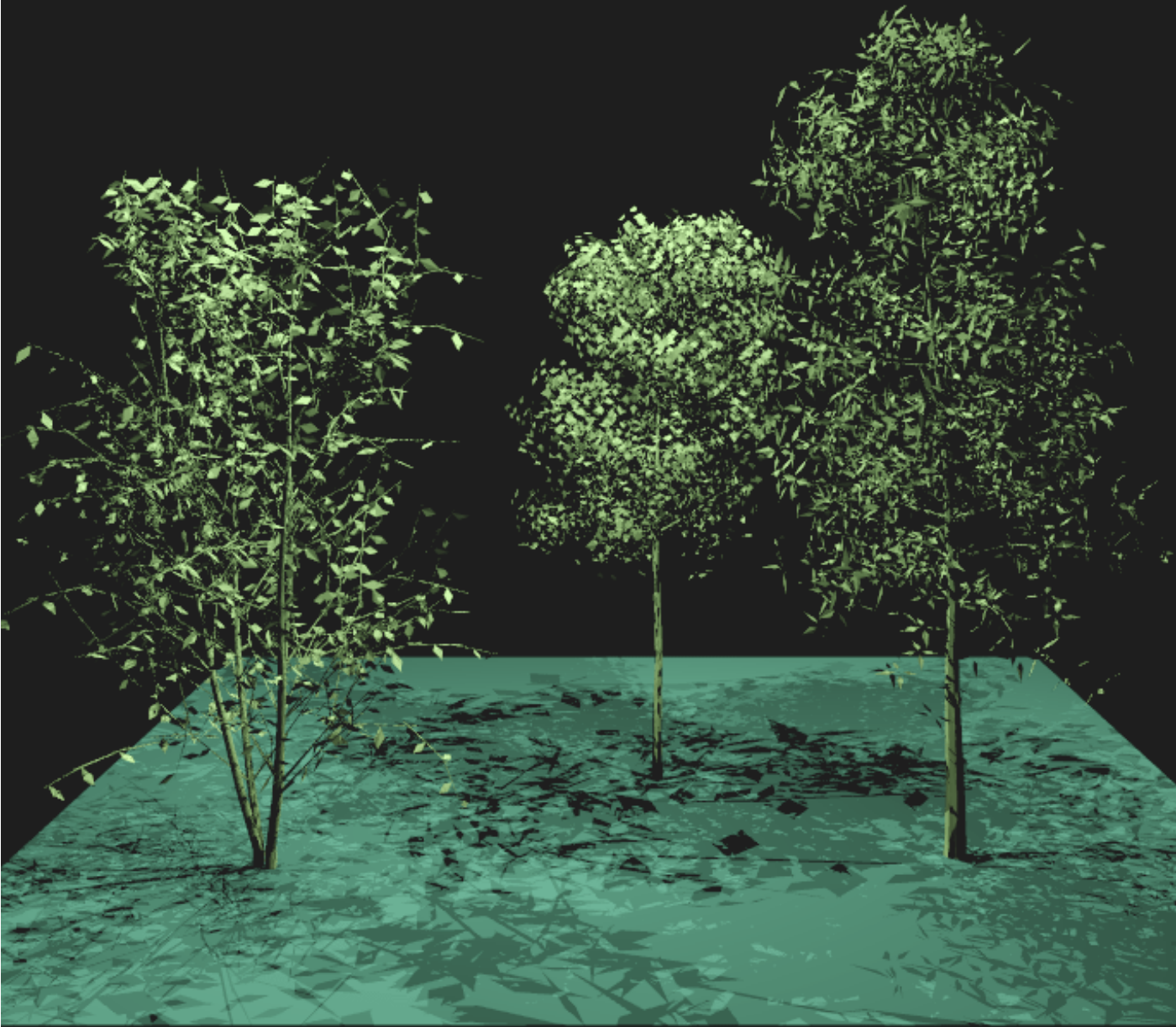
Fig. A3. RRTA with light meshes (density 150x150x150), diffuse surfaces (left) and specular surfaces (right)

# Soft → Light Meshes



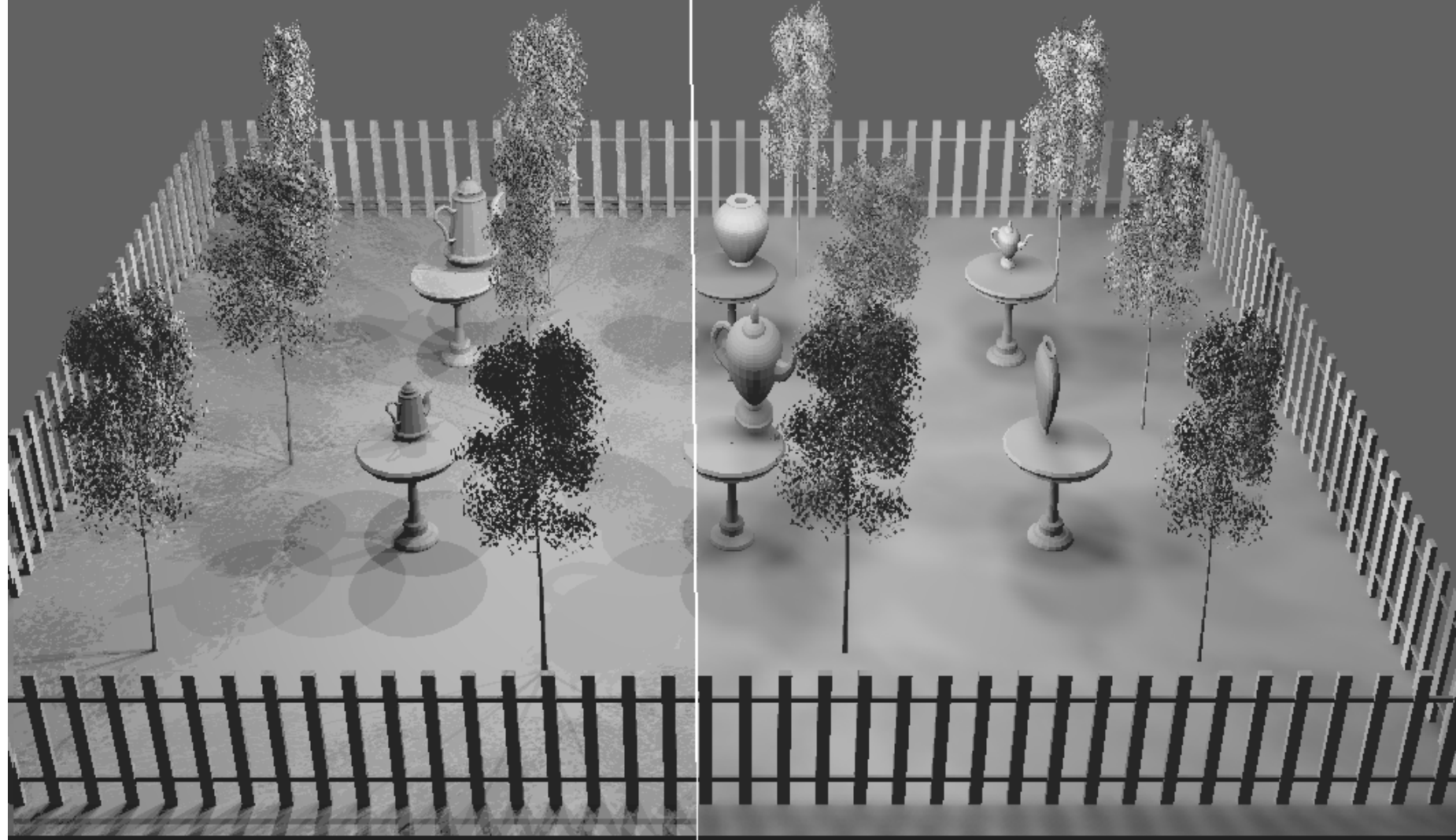


# Sharp $\rightarrow$ RRTA



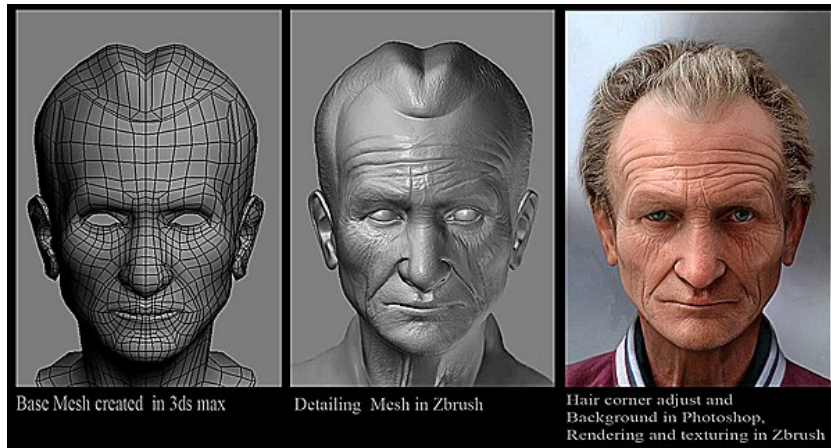
RRTA

LMM



# Текстуры

Позволяет воспроизвести малые объекты поверхности, создание которых полигонами оказалось бы чрезмерно ресурсоёмким. Например, шрамы на коже, складки на одежде, мелкие камни и прочие предметы на поверхности стен и почвы.





# Пикселы и текселы

- **pixel, or picture element**

минимальная единица двумерного цифрового изображения, а так же элемент матрицы дисплеев

- **texel, or *texture element* (also *texture pixel*)**

минимальная единица текстуры трёхмерного объекта.

- Для идеального отображения текстуры количество текселов должно совпадать с количеством пикселов

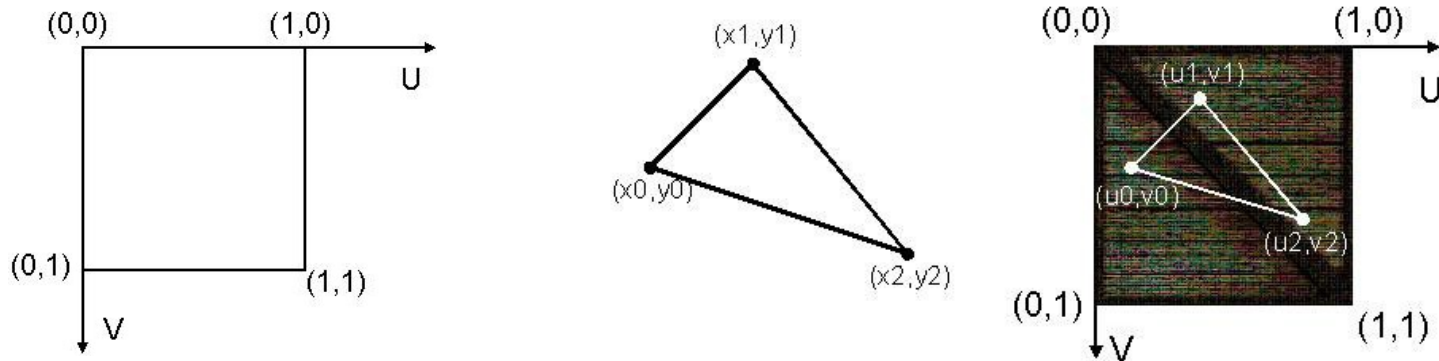
Количество  
текселей и  
пикселей  
совпадает



**Увеличенное  
изображение**  
Один тексель  
отображается во  
много пикселей  
монитора

# Текстурные координаты

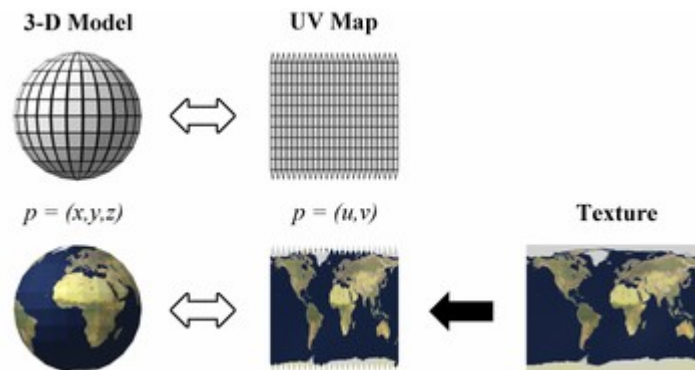
- Процесс наложения текстуры на объект называют **текстурированием** или **мапированием** (от англ. **mapping**).
- Текстуры накладываются на объект с помощью так называемых **текстурных координат (texture coords)**.



- Пара величин  $(u, v)$  однозначно указывает на элемент текстуры, называемый **текселем**.

# Текстурные координаты

- **UV-преобразование** в трёхмерной графике (англ. *UV map*) — соответствие между координатами на поверхности трёхмерного объекта ( $X, Y, Z$ ) и координатами на текстуре ( $U, V$ ).
- Развёртка может строиться как вручную, так и автоматически.



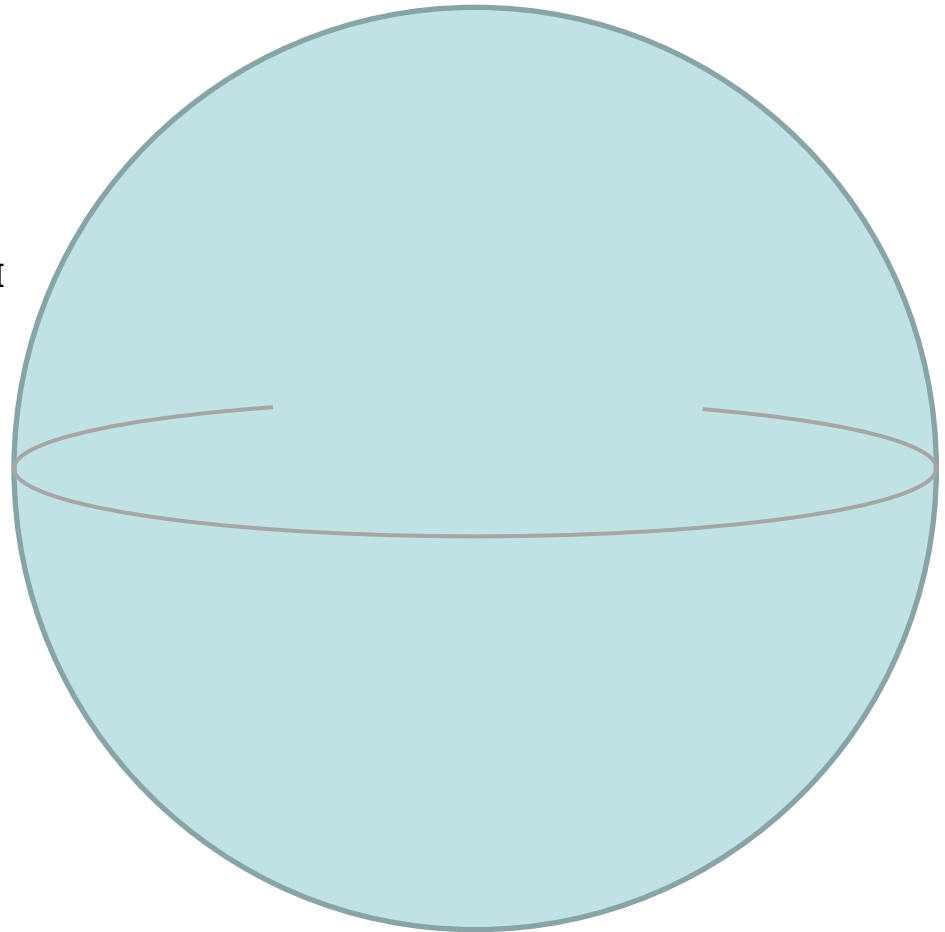
# Текстурные координаты

- Привязываются к каждой вершине сетки
- Обычно обозначаются  $s$ ,  $t$ ,  $r$ ,  $q$
- К текстурным координатам можно применять матричные преобразования
- Правила повторения или обрезания (clamp/repeat)
- Явное задание:

```
glBegin(GL_QUADS);  
glTexCoord2f(0.0,0.0); glVertex3f(-1.0,-1.0,-1.0);  
glTexCoord2f(0.0,1.0); glVertex3f(-1.0, 1.0,-1.0);  
glTexCoord2f(1.0,1.0); glVertex3f( 1.0, 1.0,-1.0);  
glTexCoord2f(1.0,0.0); glVertex3f( 1.0,-1.0,-1.0);  
...  
glEnd();
```

# Проецирование текстурных координат

- Вместо задания каждой координаты объект мысленно проецируется на сферу, цилиндр или плоскость.
- Координаты в соответствующей системе координат используются в качестве текстурной:
  - Плоскость —  $x, y$
  - Цилиндр —  $\varphi, z$
  - Сфера —  $\varphi, \theta$



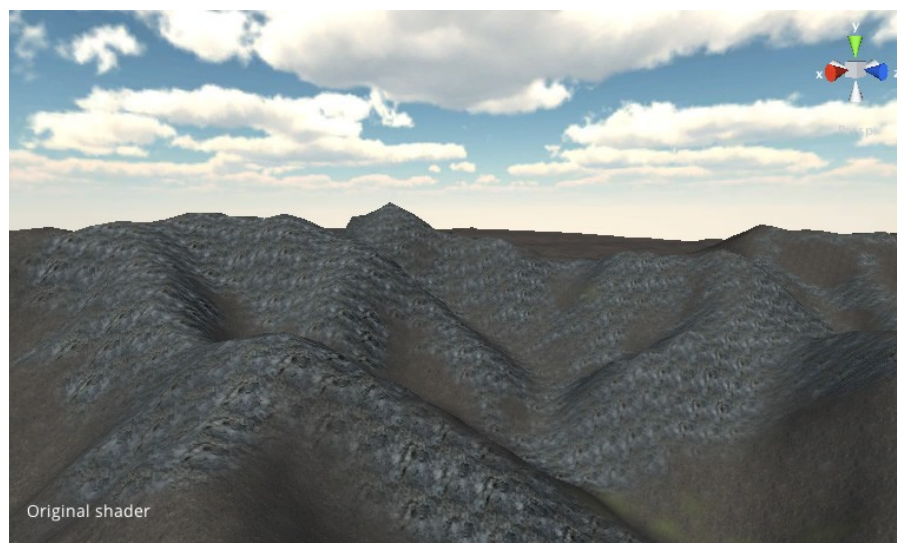
# Тайлинг (Tiling)

- вид наложения текстур на объект «по типу поклейки обоев с рисунком»



*Швов склейки не видно)*

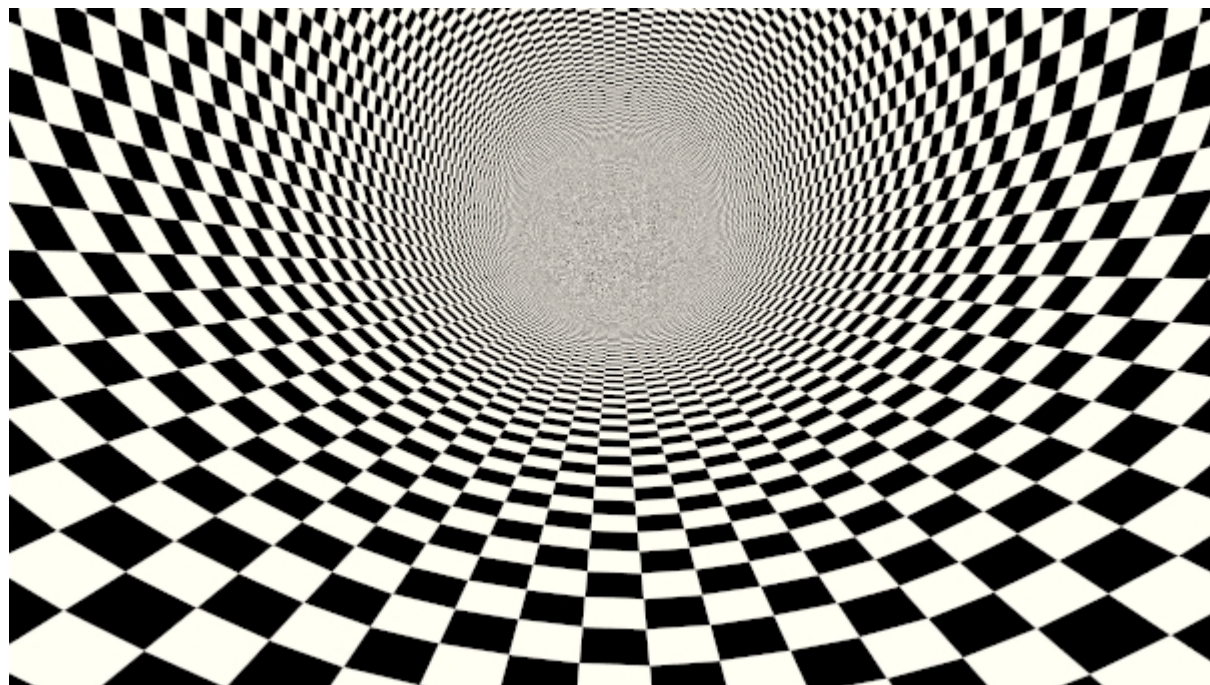
# Тайлинг (Tiling)





# Масштабирование

- Вдалеке много текселей претендуют попасть в 1 пиксель экрана





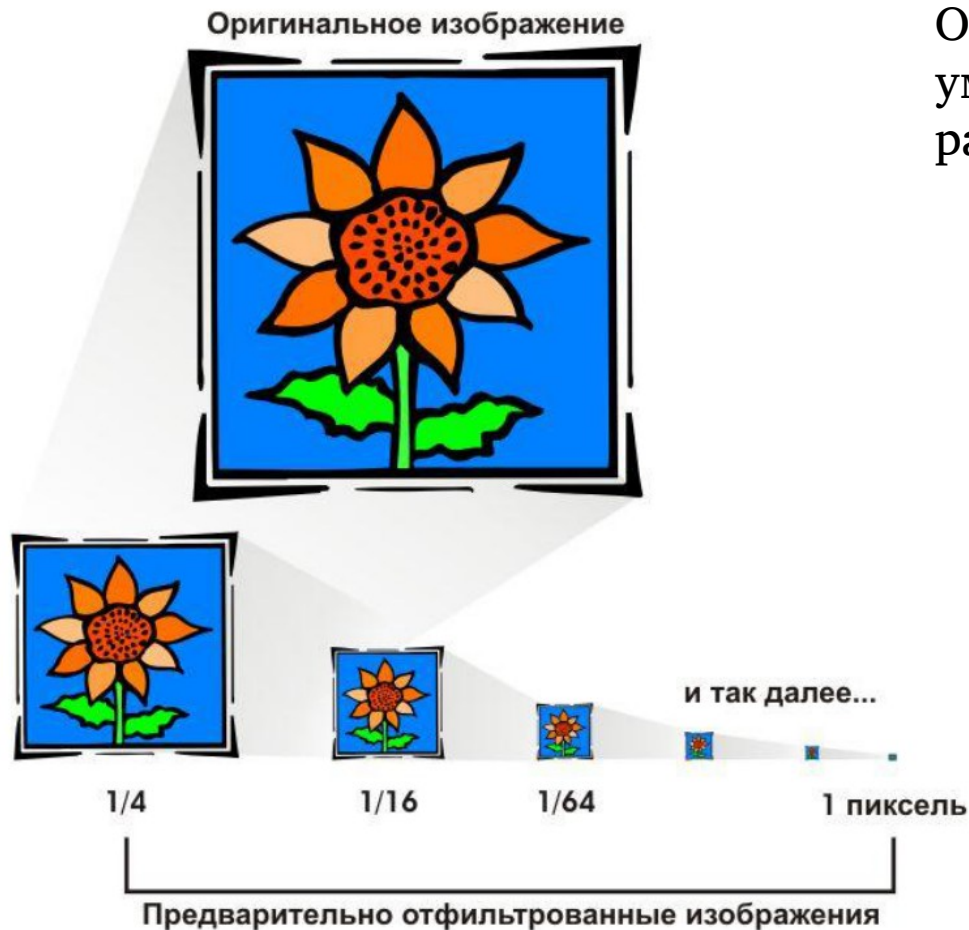
# Масштабирование

- Одному пикселю на экране может соответствовать как много меньше, так и много больше одного пикселя текстуры
- Увеличение и уменьшение
  - Взять ближайший пиксель (nearest)
  - Билинейно интерполировать четыре ближайших пикселя (linear)

# Масштабирование

- Изображение лучше всего выглядит, когда детализация текстуры близка к разрешению экрана.
- Если разрешение экрана высокое (текстура слишком маленькая/объект очень близко), получается размытое изображение.
- Если же разрешение текстуры слишком высокое (текстура слишком велика/объект очень далеко), получаем случайные пиксели — а значит, потерю мелких деталей, мерцание.
- Получается, что лучше иметь несколько текстур разной детализации и накладывать на объект ту, которая наиболее подходит в данной ситуации.

# Несколько уровней детализации текстуры (mipmap)



Обычно генерируются с уменьшением линейных размеров вдвое:

$$256 \times 64$$

$$128 \times 32$$

$$64 \times 16$$

$$32 \times 8$$

$$16 \times 4$$

$$8 \times 2$$

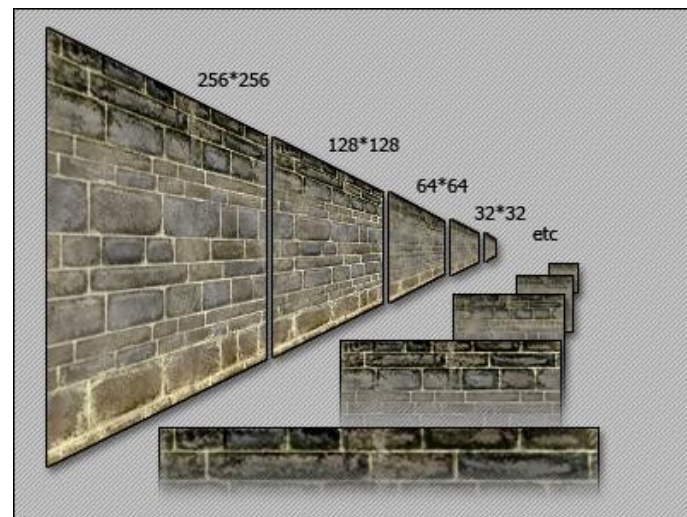
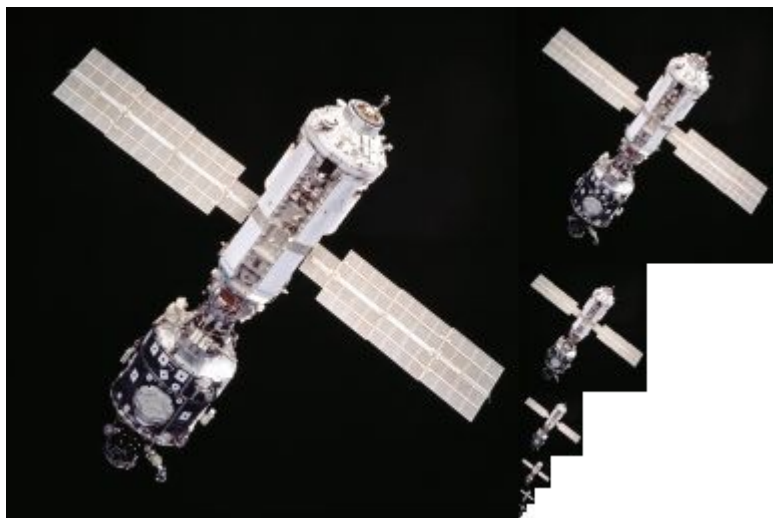
$$4 \times 1$$

$$2 \times 1$$

$$1 \times 1$$

# Несколько уровней детализации текстуры (mipmap)

Название происходит от лат. *multum in parvo* - «МНОГО В МАЛОМ».



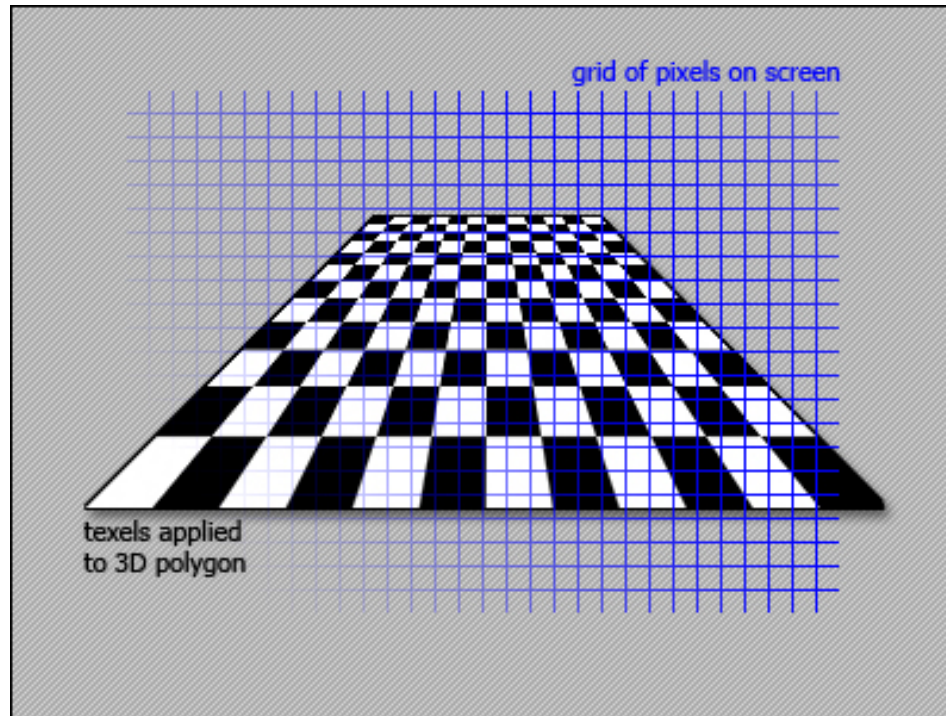
Занимаемый объем памяти в 1.5 раза больше исходной текстуры

# Уменьшение с использованием mipmaps

- Вычисляем ширину (высоту, площадь) пикселя в текстурных координатах и определяем коэффициент масштабирования текстуры
- Выбираем ближайший пиксель из ближайшего по масштабу мипмэпа (nearest-mipmap-nearest)
- Интерполируем четыре пикселя из ближайшего по масштабу мипмэпа (nearest-mipmap-linear)
- Выбираем ближайший пиксель из двух ближайших по масштабу мипмэпов и интерполируем их (linear-mipmap-nearest)
- Интерполируем по четыре пикселя из двух ближайших по масштабу мипмэпов и интерполируем результат (linear-mipmap-linear)

# Фильтрация

- Фильтрация текстур необходима, т.к. позиции текселей обычно не совпадают с пикселями экрана.
- Какой из текселей выбрать для отрисовки пиксела ?



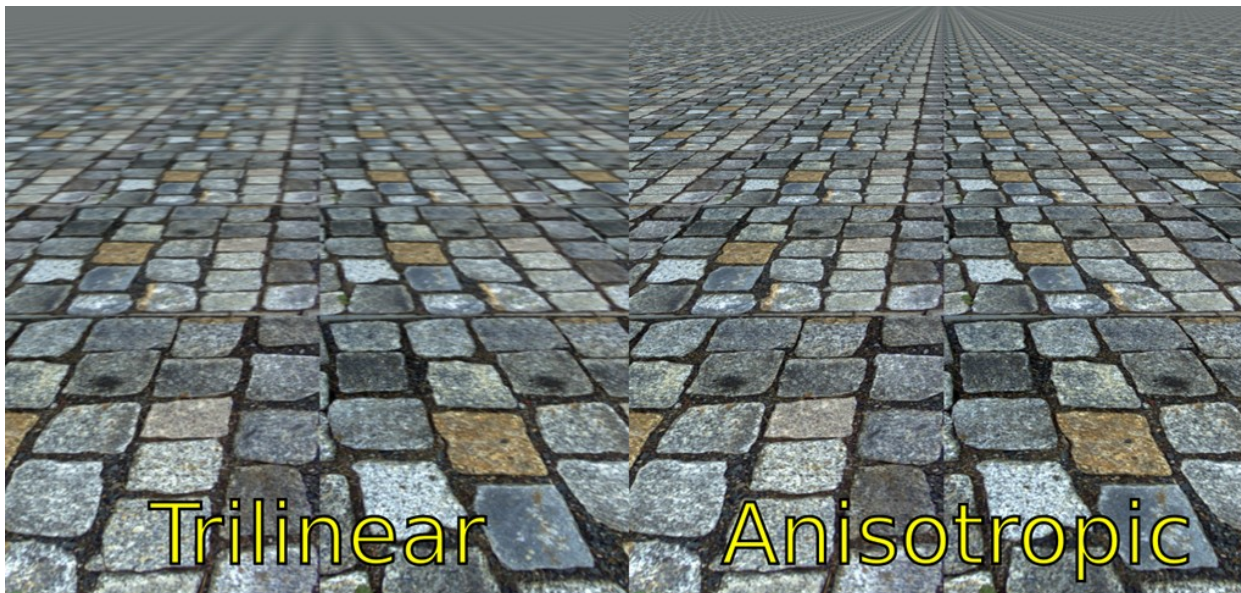
# Фильтрация. Трилинейная





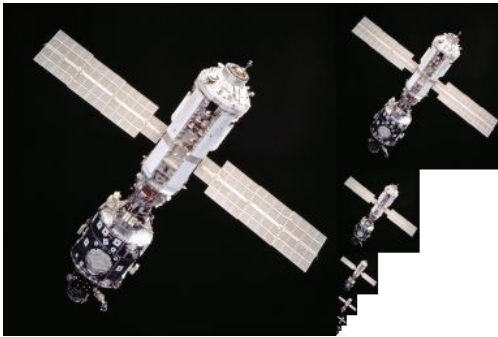
# Фильтрация. Анизотропная

- метод улучшения качества изображения текстур на поверхностях, сильно наклонённых относительно камеры
- вносит меньше размытия и поэтому позволяет сохранить больше деталей изображения





# Фильтрация. Анизотропная



MIP mapping  
256x256,  
128x128,  
...

x2    x4    x8



x2  
x4  
x8

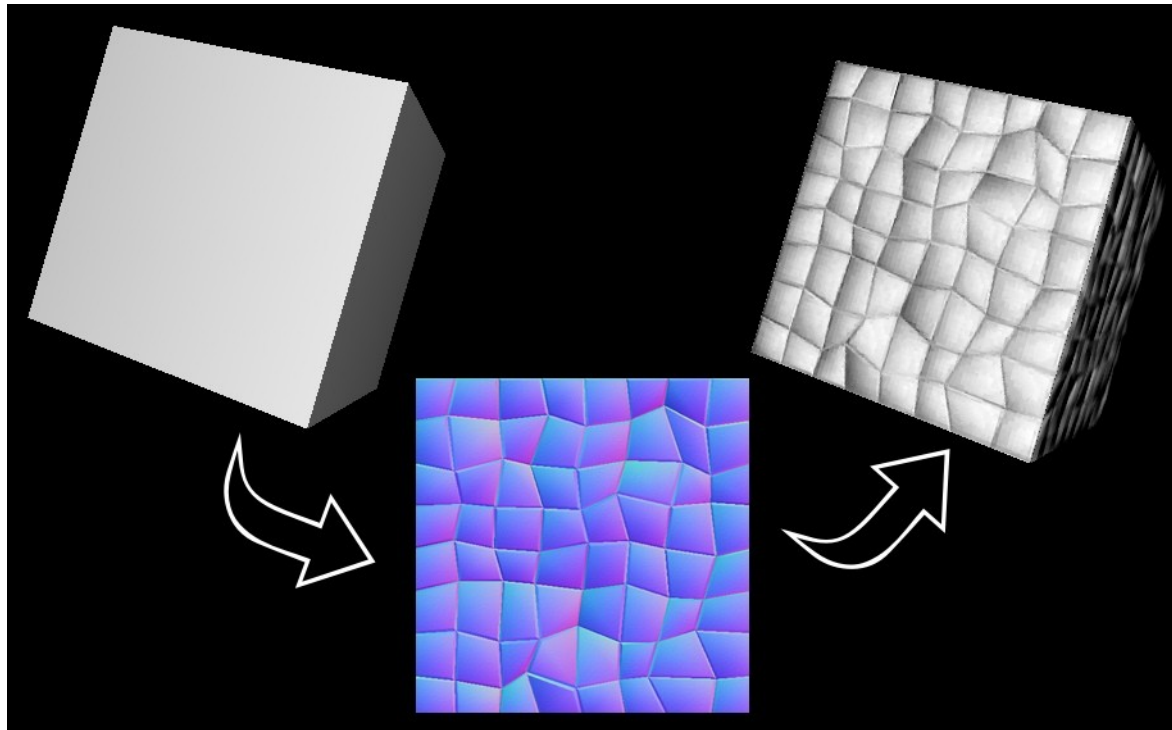
RIP mapping  
256x256, 256x128, 128x256, ...  
128x128, 128x64, 64x128, ...  
...

# Взаимодействие с материалом

- Игнорировать материал; использовать только цвет текстуры (расчёт освещённости выключен)
- Смешивать рассчитанный цвет цветом текстуры (blend)
- Модулировать (умножать) цвет цветом текстуры
- Ограничиться обработкой части компонент:
  - Только яркость
  - Только RGB
  - Только альфа
  - Альфа и яркость

# Модуляция нормалей

Простой способ создания эффекта рельефной поверхности с детализацией большей, чем позволяет полигональная поверхность.



# Модуляция нормалей

Позволяет низкополигональной модели выглядеть высокополигонально (с хорошей детализацией)



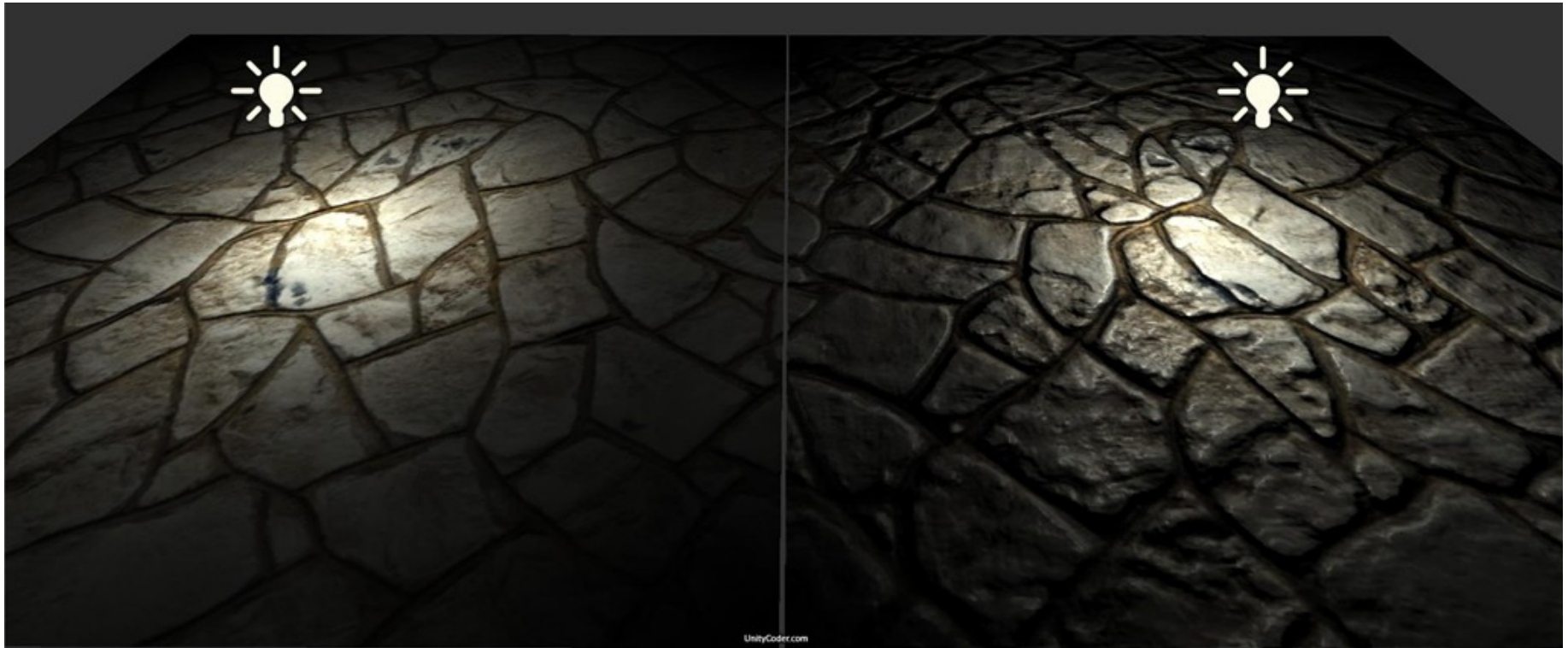
**No Normal Map**



**With Normal Map**

# Модуляция нормалей

Обязательное условие: эффект достигается за счет освещения поверхности источником света

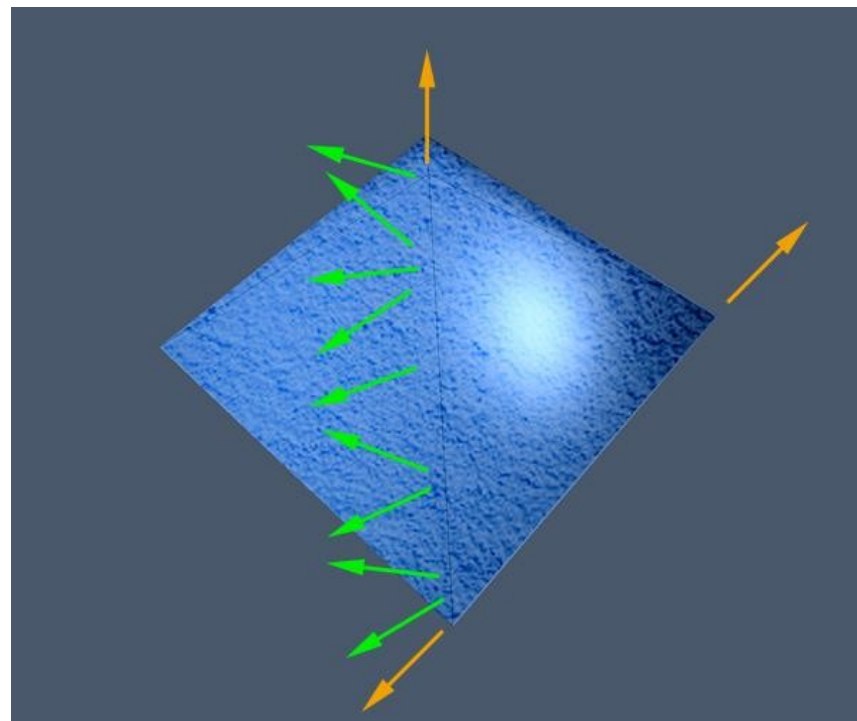
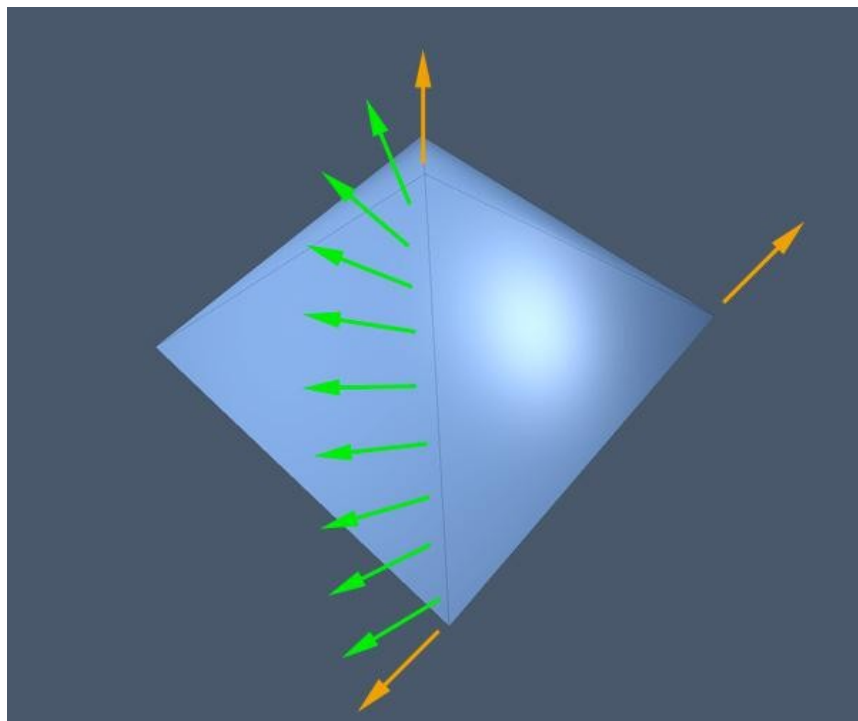


без карты нормалей

с картой нормалей

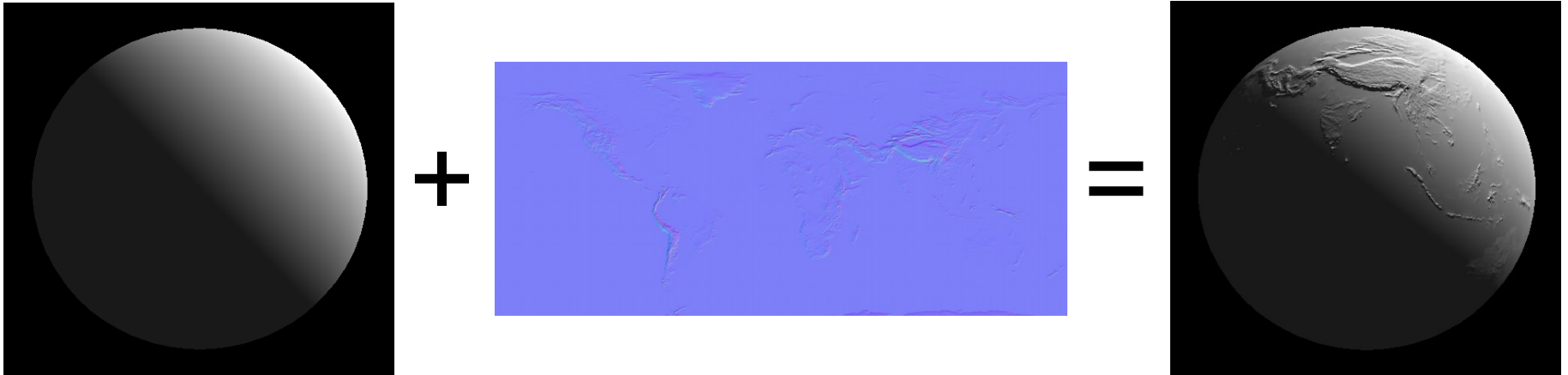
# Модуляция нормалей

К полученной обычным способом нормали в точке добавляется отклонение нормали из текстуры

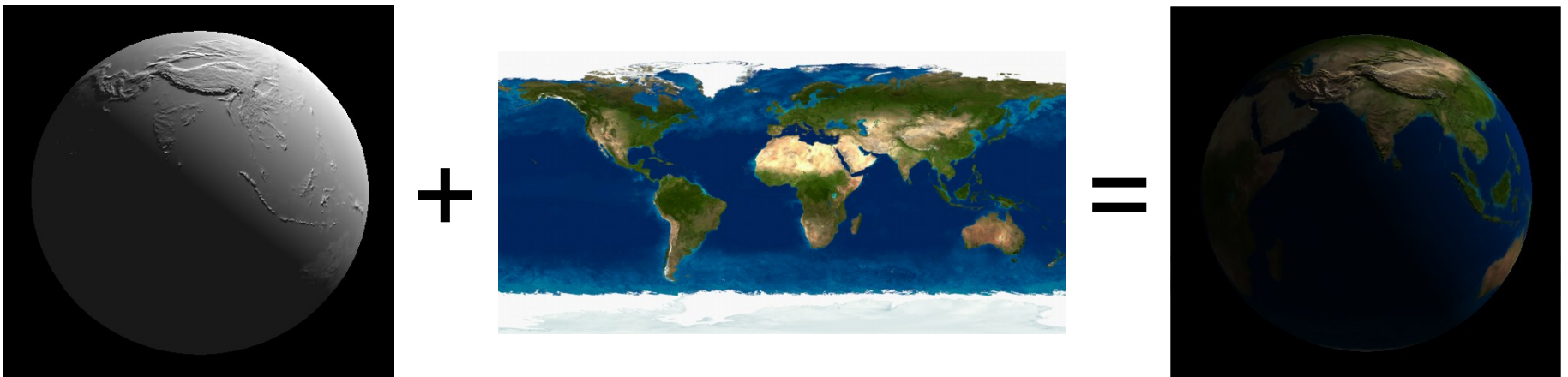


# Модуляция нормалей (пример)

1. Сначала наложить карту нормалей на сферу



2. Потом добавить текстуру Земли





# Модуляция нормалей (пример)





# Размерность текстур

- Одномерные
  - Как двумерные, но в один пиксель высотой
- Двумерные
- Трёхмерные
  - Объёмные плотности, томограмма и т. д.
  - $256 \times 256 \times 256 \times 4 = 64$  Мб