



LOBACHEVSKY STATE UNIVERSITY
of NIZHNI NOVGOROD
National Research University

Computing Mathematics and Cybernetics faculty
Software department

CS255. Computer Graphics Introduction Course

Методы построения теней

Турлапов, Вадим Евгеньевич
проф. каф. МО ЭВМ

Введение

В компьютерной графике (особенно в режиме реального времени) часто объекты изображаются без теней, что приводит к тому, объект выглядит незакрепленным в окружающей обстановке. Тень несёт очень много информации – фактически, она представляет объект с другой точки зрения и закрепляет его в сцене.

В отличие от локального освещения, тени являются одним из эффектов глобального освещения. Для их построения необходимо учитывать расположение всех объектов в моделируемой сцене. Современные графические укорители работают на достаточно низком уровне и ничего не знают о наличие сцены, поэтому тени не поддерживаются ими напрямую и вся подготовительная работа ложится на плечи программиста.

Введение

- ▶ Геометрическая оптика Френеля
- ▶ Чёткие (*hard shadows*) тени
 - ▶ Точечный источник или источник направленного параллельного света

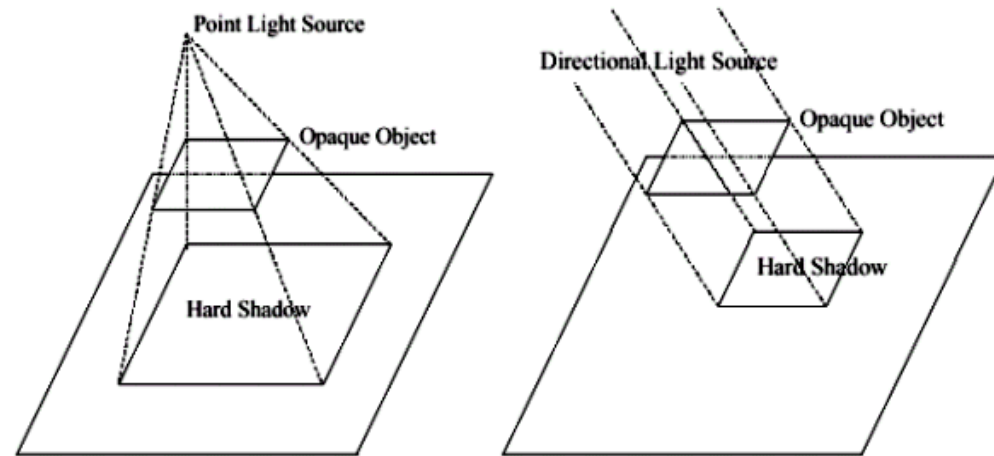


рис. 1.1. чёткая тень.

- ▶ Мягкие (*soft shadows*) тени
 - ▶ Протяжённый источник света

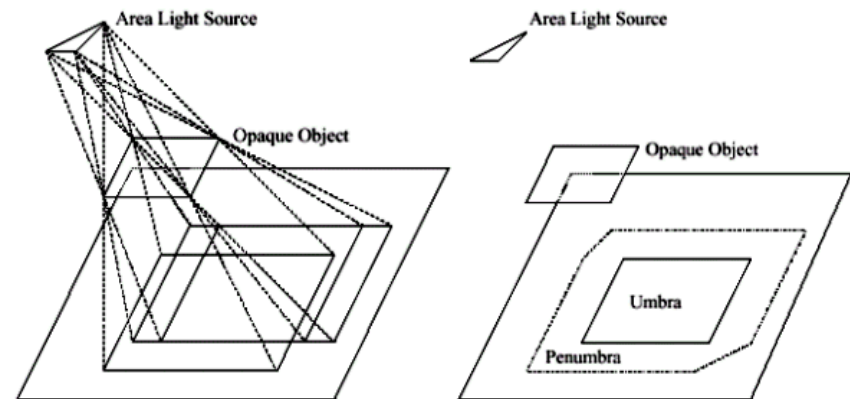


рис. 1.2. мягкая тень.

Основные подходы

1. Преобразование модели "на землю" и её отрисовка как тени
2. Построение теневой маски объекта и её проективное наложение на другие объекты
3. Теневые объёмы
4. Использование буфера глубины (*depth buffer*)

1. Преобразование «на землю»

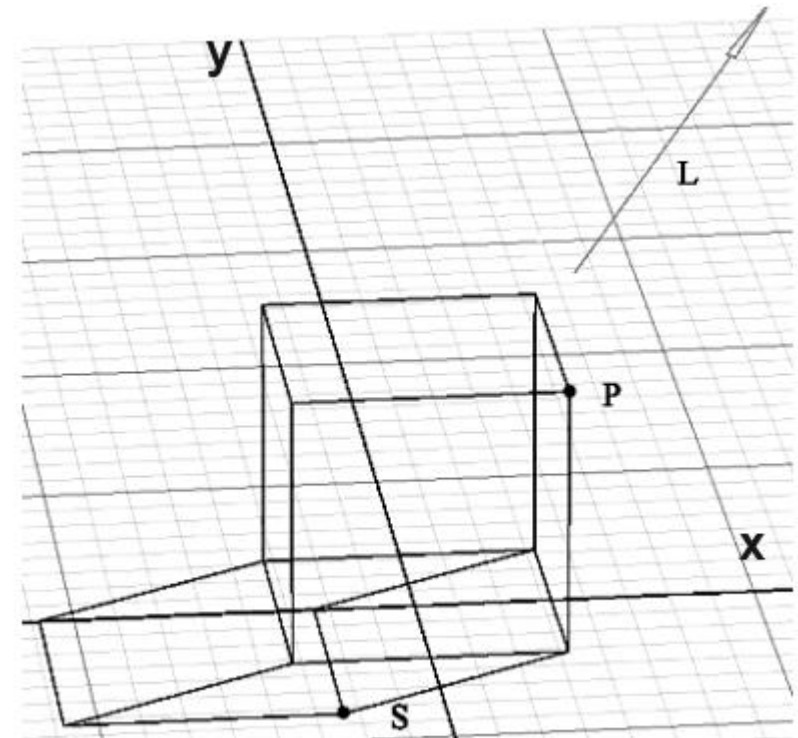
- ▶ Первый алгоритм, применённый в играх. Впервые описан Джимом Блинном
- ▶ Простота реализации и хорошее качество получаемых теней
- ▶ Рассматриваются два случая
 - ▶ Источник в бесконечности (параллельный направленный свет)
 - ▶ Локальный источник (точечный источник недалеко от объекта)
- ▶ Расчет теневых проекций полигона для каждого источника света

1. Преобразование «на землю»

Источник в бесконечности

В случае бесконечно удалённого источника света мы полагаем, что лучи света, приходящие к объекту, полностью параллельны.

Это позволит нам решить уравнение проекции только раз и применять полученное решение ко всем вершинам объекта



1. Преобразование «на землю»

Источник в бесконечности

Имея направление света (x_l, y_l, z_l) и вершину объекта (x_p, y_p, z_p) хотим получить проекцию вершины объекта на плоскость $z=0$, т.е. точку тени (x_s, y_s, z_s) . Из подобия треугольников, используя стандартные матричные преобразования, получаем матрицу перехода

$$M_s = \begin{pmatrix} 1 & 0 & -x_l/z_l & 0 \\ 0 & 1 & -y_l/z_l & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Теперь, имея координаты точки P в мировом пространстве, можно получить её проекцию S на плоскость $z=0$ просто путём умножения на матрицу:

$$S = M_s P$$

1. Преобразование «на землю»

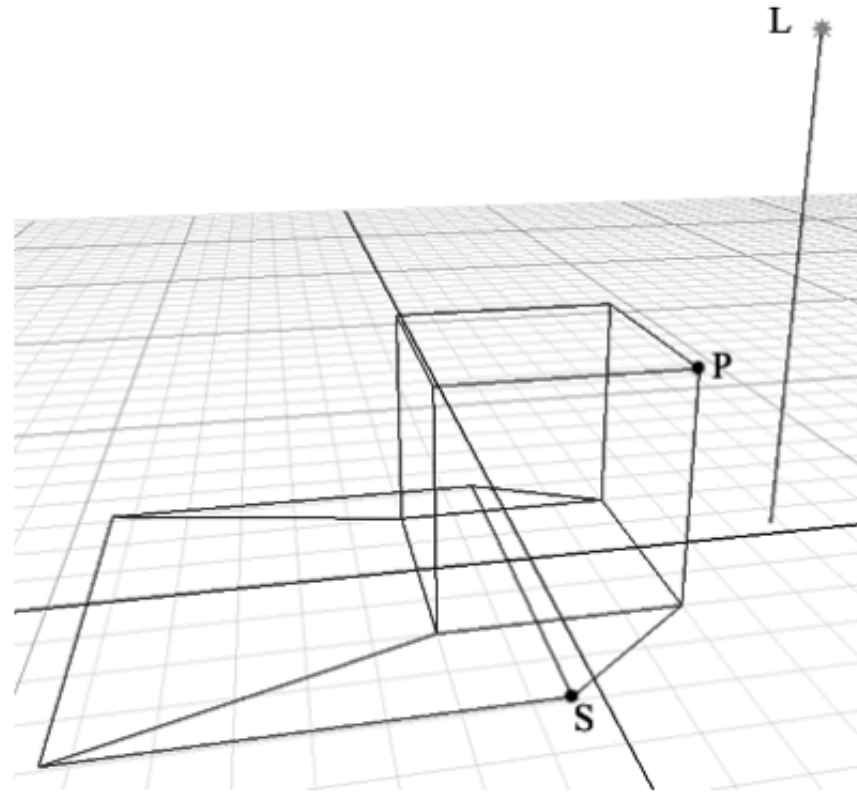
Локальный источник

В этом случае понадобятся дополнительные вычисления на каждую вершину, т.к. каждая вершина имеет, в общем случае, своё собственное направление на источник света.

$$S = P + \alpha(P - L)$$

$$\alpha = \frac{-Z_p}{Z_p - Z_l}$$

$$x_s = \frac{x_l z_p - x_p z_l}{z_p - z_l}, y_s = \frac{y_l z_p - y_p z_l}{z_p - z_l}$$



1. Преобразование «на землю»

Локальный источник

Проецируя на плоскость $z = 0$ и используя гомогенизацию после преобразования, можно записать матрицу

$$M_{sh} = \begin{pmatrix} -z_l & 0 & x_l & 0 \\ 0 & -z_l & 0 & 0 \\ 0 & 0 & y_l & 0 \\ 0 & 0 & 1 & -z_l \end{pmatrix}$$

Имея координаты точки P в мировом координатном пространстве, можно записать: $S_h = M_{sh}P$, после чего провести гомогенизацию точки для получения проекции точки P на плоскость $z=0$.

2. Построение и проективное наложение теневой маски

- ▶ Объект, который отбрасывает тень (*shadow caster*) тем или иным способом рисуется с точки зрения источника света чёрным цветом в белую текстуру. Размер текстуры зависит от того, насколько мелкие элементы объекта мы хотим видеть в тени
- ▶ Текстура, в которую производится отрисовка объекта, - плоскость, перпендикулярная направлению на источник света и расположенная сразу за объектом по лучу света
- ▶ Для получения проекции объекта на текстуру, применяем элементы стандартного конвейера трансформации. Пусть текстура, в которую нам необходимо нарисовать объект, это экран. Тогда последовательность преобразований выглядит так: *пространство объекта* => *мировое пространство* => *пространство камеры* (источник света) => *проективное пространство* => *нормализованное проективное пространство* => *экранное пространство*

$S = (M_W \cdot M_C \cdot M_{pj}) \cdot P$ + нормализация и перевод в экранное пространство

2. Построение и проективное наложение теневой маски

Пространство камеры

Построение матрицы камеры. Виртуальная камера расположена в точке нахождения источника света и направлена на *shadow caster*.

В случае бесконечно удалённого источника света виртуальная камера может располагаться в любой точке на прямой между *shadow caster* и источником света (обычно её выбирают близко к *shadow caster*).

2. Построение и проективное наложение теневой маски

Матрица существенно зависит от того, является ли источник света локальным или бесконечно удалённым.

Для бесконечно удалённого источника света это матрица **ортографического преобразования**, где *width*, *height* и *depth* это ширина, высота и глубина наблюдаемого объёма. В этом случае не требуется гомогенизация.

$$M_{\text{орт}} = \begin{pmatrix} \frac{2}{width} & 0 & 0 & 0 \\ 0 & \frac{2}{height} & 0 & 0 \\ 0 & 0 & -\frac{1}{depth} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

2. Построение и проективное наложение теневой маски

- ▶ **Для локального источника света** необходима матрица перспективного преобразования, где φ - поле зрения (FOV), $aspect$ - отношение ширины к высоте экрана (при отрисовке в квадратную текстуру $aspect=1$), far и $near$ - дальняя и ближняя стенки пирамиды видимости. После перспективного преобразования необходима гомогенизация, как и в случае локального источника при преобразовании "на землю".
- ▶ Для качественной картинке рекомендуется выбирать проективный объём таким образом, чтобы он максимально плотно охватывал *shadow caster* и не оставлял зазоров.

$$M_{pj} = \begin{pmatrix} \cos(\varphi/2) & 0 & 0 & 0 \\ 0 & aspect * \cos(\varphi/2) & 0 & 0 \\ 0 & 0 & \frac{\sin(\varphi/2) far}{far - near} & \sin(\varphi/2) \\ 0 & 0 & -\frac{\sin(\varphi/2) * far * near}{far - near} & 0 \end{pmatrix}$$

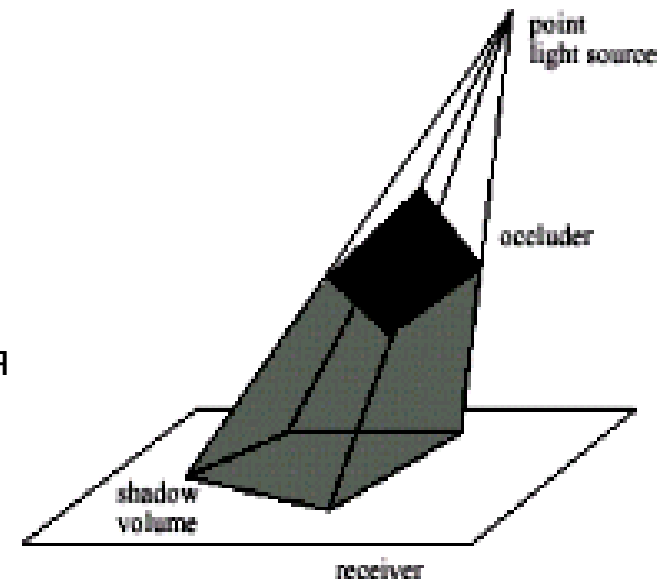
2. Построение и проективное наложение теневой маски

Экранное пространство

- ▶ Модель из нормализованного проективного пространства должна лечь в квадратную текстуру с минимальными зазорами.
- ▶ После того, как *shadow caster* нарисован в текстуру, необходимо эту текстуру наложить на затенённые объекты (*shadow receiver*). Для этого можно воспользоваться технологией проективного наложения текстур.
- ▶ В результате мы получаем тень от одного объекта (*shadow caster*) на другом (*shadow receiver*).

3. Теневые объёмы

- ▶ Теневой объём - это представление пространства за объектом, из которого не наблюдается источник света, в виде полигонального объекта. При освещении сцены в тени оказываются те объекты, которые попадают внутрь теневого объёма.
- ▶ Для того, чтобы точка была затенена, луч света должен войти в теневые объёмы большее число раз, нежели выйти из них, т.е. остаться внутри хотя бы одного теневого объёма. Для упрощения задачи можно принять, что не луч света от источника должен проделать этот путь, а луч от наблюдателя.
- ▶ Алгоритм можно разделить на два явных действия: построение "*теневой маски*" (т.е. маски освещённых и затенённых областей 2-х мерной картинки) и отрисовка сцены с использованием *теневой маски*.



3. Теневые объёмы

-первая стадия

На этой стадии мы рисуем всю сцену таким образом, как будто она вся затенена. Все последующие стадии служат цели добавления освещения в картину. Участки картинки, которые так и не будут освещены в последующих проходах, останутся затенёнными.

3. Теневые объёмы

-вторая стадия

- ▶ Строим теневую маску, используя информацию о 3-мерных теневых объёмах. Для создания и хранения теневой маски используется *буфер шаблонов (трафарета, stencil buffer)*. Воспользуемся простым способом создания теневого объёма - от каждого треугольника в модели.
- ▶ Необходимо подсчитать для каждой точки разность того, сколько раз луч от наблюдателя пересёк положительно ориентированные полигоны *теневого объёма* и отрицательно ориентированные. Это производится за два прохода.
 - ▶ На первом проходе рисуются все теневые объёмы (без записи точки в цветовой буфер и в буфер глубины) с нормальным (для нашего приложения) отбрасыванием треугольников (*back face culling*). Каждая поверхность, которая проходит тест по глубине в некотором пикселе, увеличивает значение в буфере шаблонов на 1.
 - ▶ На втором проходе меняем *back face culling* на противоположный - рисоваться будут задние стенки *теневых объёмов*. Теперь каждая поверхность, которая проходит тест по глубине в некотором пикселе, будет уменьшать значение в *буфере шаблонов* на 1. В итоге после этих двух проходов имеем в *буфере шаблонов* нулевые значения для освещённых областей и значения больше нуля для затенённых.

3. Теневые объёмы

-третья стадия

- ▶ Отрисовываем ещё раз всю сцену освещённой для нулевых значений буфера шаблонов (трафарета, stencil buffer).

Вторую и третью стадии повторяем для каждого источника света, но при этом очищать следует только *буфер шаблонов* перед второй стадией. Ни цветовой буфер, ни буфер глубины на этих стадиях мы не очищаем.

Чтобы этот алгоритм можно было использовать, необходимо решить ещё несколько вопросов:

- ▶ Если наблюдатель уже находится внутри *теневого объёма*, то необходимо инвертировать результат *теневой маски*. Если *теневого объёма* пересекает ближнюю плоскость отсечения проективного объёма, то часть теневой маски необходимо инвертировать, а часть - нет.
- ▶ Есть неприятный эффект, который возникает из-за совпадения передней стенки теневого объёма с реальным объектом, его отбрасывающим. В этом случае, из-за конечной точности представления глубины, часть пикселей на передней стенке теневого объёма будет затенена, а часть – нет.

4. Использование буфера теней

- Затенённые точки - это "*hidden surfaces*" с точки зрения источника света.

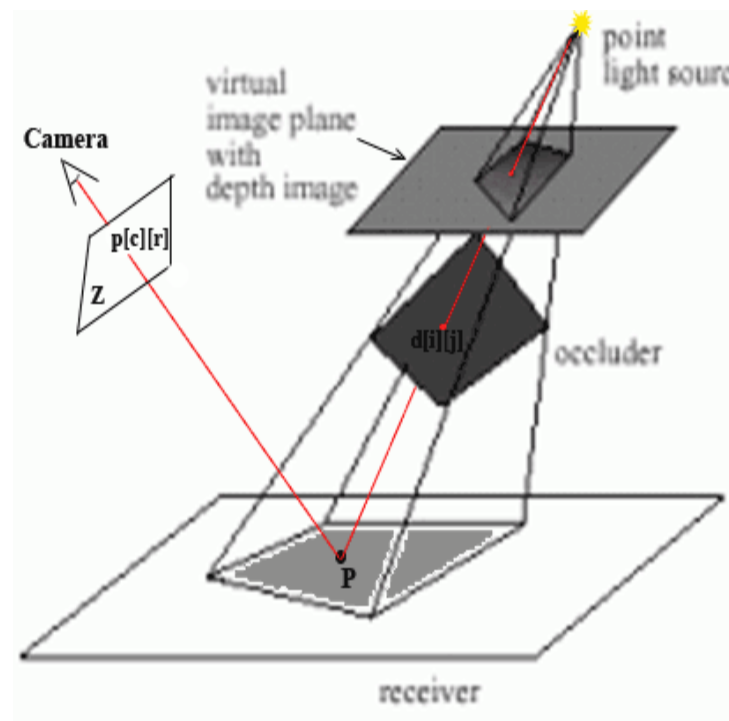
Два этапа:

- ▶ Загрузка буфера теней (аналогична Z-буферу, камера в точке источника).
- ▶ Визуализация

Визуализация

При визуализации точки P через пиксел $p[c][r]$ Z-буфера нужно найти:

- ▶ псевдоглубину D для точки P для буфера теней (нормированную в CVV);
- ▶ индексы $[i][j]$ элемента буфера теней, подлежащего проверке для точки P ;
- ▶ значение $d[i][j]$.
- ▶ Если значение $d < D$, то точка P рисуется цветом ambient.



4. Использование информации о глубине через альфа-канал

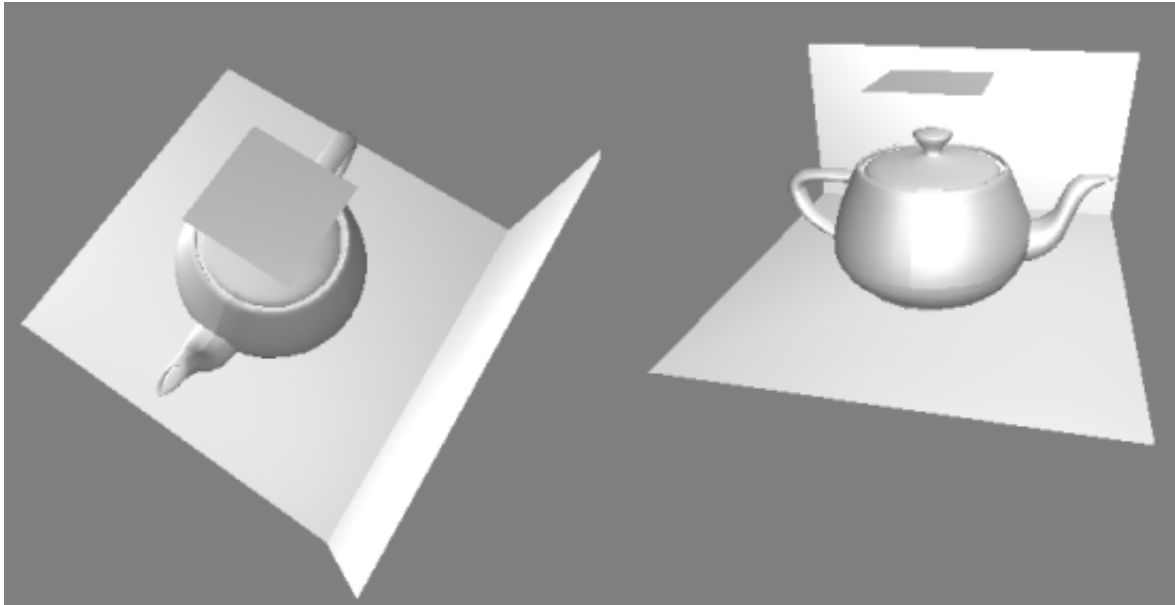
► Необходимо отрисовать всю сцену с точки зрения наблюдателя, проделывая следующую проверку: если глубина точки, переведённая в систему координат источника света, больше значения, которое мы сохранили в буфере глубины на предыдущей стадии, значит, точка затенена. В противном случае точка освещена.

Поскольку это искусственный прием, то мы не можем использовать аппаратный буфер глубины, но можем сравнивать значения альфа-каналов.

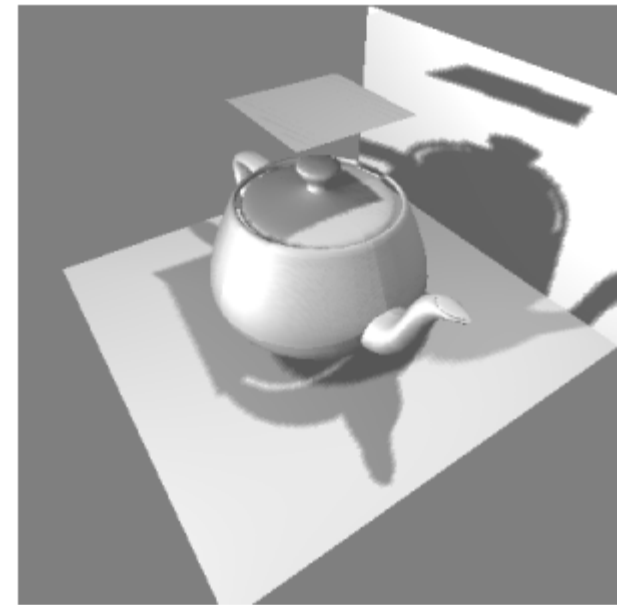
В этом случае требуется:

- 1) возможность смены цветового буфера для текущей отрисовки (render target);
- 2) 2-х текстурный растеризатор.

4. Использование информации о глубине. Пример



Сцена с точки зрения
двух источников света
(не образ глубины)



Сцена с точки зрения
наблюдателя
(с построенными тенями)

4. Использование информации о глубине

Если аппаратура не поддерживает доступ к буферу глубины с т.з. источника

Стадия 1:

- Получаем образ глубины сцены с точки зрения источника света. Для этого отрисуем сцену в текстуру (как в другой *render target*). Для получения глубины воспользуемся текстурой, в которой значения размещены от 0 до 255 в альфа-канале (цветовые значения нас не интересуют). Чтобы текстура легла необходимым для нас образом, сгенерируем текстурные координаты в пространстве камеры с последующей трансформацией:

$$S = (M_{lpj} M_{sel}) P_{cam}$$

+ нормализация альфа-канала.

$$M_{sel} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

M_{lpj} - матрица проективного преобразования от источника света

M_{sel} - матрица выборки в альфа-канал

- После отрисовки в текстуре имеем образ глубины сцены с шагом = 1/255.

4. Использование информации о глубине. 2 стадия:

Создаём *теневую маску*, базирясь на сравнении глубины

- ▶ Для этого наложим первую текстуру на объект
- ▶ Для наложения первой текстуры необходимо сгенерировать текстурные координаты как позиции в пространстве камеры с последующей трансформацией и нормализацией.

4. Использование информации о глубине. 2 стадия:

- ▶ Наложим вторую текстуру, полученную на первой стадии, проективным методом. Проективное наложение второй текстуры осуществляется следующим образом

$$S = (M_{cam}^{-1} M_{lpj} M_{center} M_{sel}) P_{cam} + \text{нормализация}$$

$$M_{center} = \begin{pmatrix} 0.5 & 0 & 0 & 0.5 \\ 0 & 0.5 & 0 & 0.5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- матрица центрирования текстуры
в диапазон [0,1]

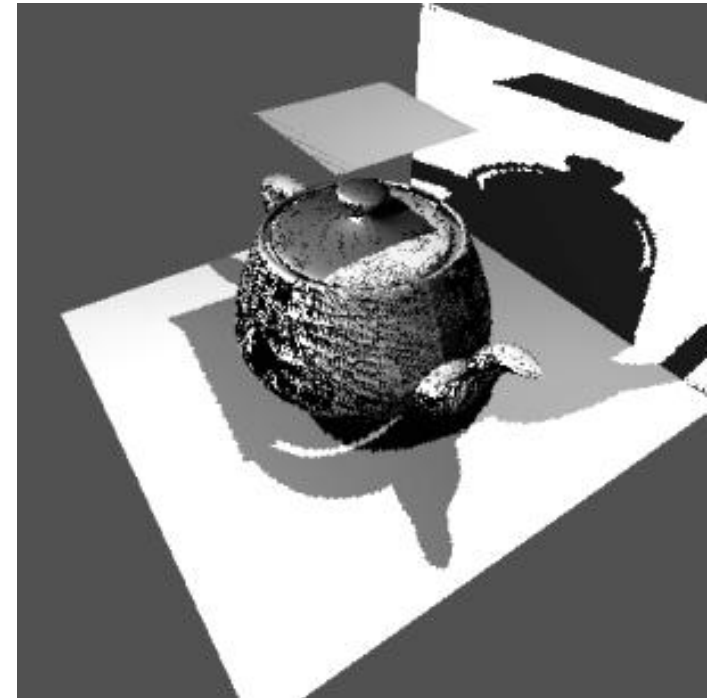
После чего, в каждой точке вычтем значения альфа-канала первой текстуры из второй и отрисуем теневую маску с проверкой по альфа-каналу. Точки, которые пройдут эту проверку, запишем в теневую маску как 1, а которые не пройдут оставим в 0.

4. Использование информации о глубине

-3 стадия, заключение

- ▶ Имея теневую маску, отрисовываем сцену →

Алгоритм таит в себе несколько неприятностей. Самая главная из них - это ошибочное самозатенение, которое возникает из-за недостаточной точности представления глубины через альфа-канал.



Источники:

- ▶ Franklin C. Crow. Shadow algorithms for computer graphics // Computer Graphics
- ▶ Blinn, James. Me and my (fake) shadow // Graphics and Applications