Построение динамических теней в системах визуализации реального времени

Семен Козлов, Белаго Игорь, Елыков Николай, Кузиковский Станислав, Лаврентьев Михаил
Лаборатория программных систем машинной графики
ИАиЭ СО РАН, Новосибирск, Россия
{smk, bel, nicolas, stas}@sl.iae.nsk.su, mmlavr@nsu.ru

Аннотация

Задача отображения динамических теней в масштабах всей сцены является одной из сложных и до конца не решенных задач современной трехмерной графики реального времени. Один из наиболее универсальных подходов – использование карт глубины, который страдает от проблем неравномерного распределения качества изображения по сцене. Существующий метод борьбы с этой проблемой – перспективные карты глубины, в свою очередь, обладает рядом недостатков, затрудняющих его использование. В статье предложен ряд дополнений и исправлений метода проективных карт глубины, позволяющих его эффективное использование в системах отображения реального времени.

Keywords: *shadow rendering,, shadow mapping*

1. ВВЕДЕНИЕ.

Отображение теней всегда являлось серьезной проблемой в современной трехмерной графике. Определение того, находится точка в тени или нет – нетривиальная операция для современных графических ускорителей, преимущественно потому, что ускорители работают в терминах растеризации полигонов, а не трассировки лучей.

В сегодняшних приложениях тени должны быть полностью динамическими, почти каждый объект в сцене должен получать и принимать тени, должно присутствовать самозатенение объектов и, желательно, тени должны быть мягкими. Согласно [1], только два алгоритма могут потенциально удовлетворять этим требованиям – теневые объемы (Stencil Shadows или Shadow Volumes), и карты глубины (Shadow Mapping).

Разница между этими алгоритмами сводится к различиям работы в объектном пространстве и пространстве изображения.

- Алгоритмы, работающие в объектном пространстве (к которым относятся теневые объемы), строят некую полигональную структуру, представляющую объемы, участвующие в затенении, что означает пиксельную точность тени, но вместе с тем и четкую границу затенения. Этот метод не может корректно обрабатывать объекты, форма которых определяется не только полигональной структурой – такие как объекты с отсечением по маске прозрачности или с использованием текстуры смещений. Впрочем, теневые объемы требуют огромного количества заливки, что затрудняет их использование в заполненных сценах, особенно если нужны мягкие тени.
- Алгоритмы, работающие в пространстве изображения (карты глубины) работают с любой модификацией геометрии (если мы вообще можем отрисовать объект, мы сможем получать от него тени), но подвержены проблемам алиасинга. Наиболее сильно эти проблемы выражены в больших сценах с широкими и точечными источниками с большим радиусом действия. Проблема заключается в том, что проективное преобразование, использующееся для построения карты глубины, изменяет размер текселей карты глубины в пространстве наблюдателя. В результате, чтобы обеспечить хорошее качество, приходится использовать карты глубины огромной величины (до 4-х раз больше размера экрана). Тем не менее, карты глубины существенно быстрее алгоритма теневых объемов в сложных сценах.

Алгоритм карт глубины кратко описан в разделе 2 и более подробно рассмотрена проблема алиасинга.

Алгоритм перспективных карт глубины (Perspective Shadow Maps, PSM), впервые представленный на конференции SIGGRAPH 2002 [2], предназначен для решения проблем алиасинга с помощью перехода в пост-проективное пространство, где ближние объекты по размерам больше дальних. К сожалению, алгоритм имеет следующие недостаки:

- Если источник находится позади камеры, авторы предлагают использовать «виртуальные камеры» чтобы удержать все объекты, отбрасывающие тени внутри пирамиды видимости, что ведет к серьезному ухудшению качества.
- Качество теней очень зависит от взаимного расположения источника и камеры.
- В статье никак не решены проблемы сдвижки значений в карте глубины, необходимые для отсутствия характерных артефактов алгоритма. Для обычных карт глубины эта сдвижка может быть константой, но для перспективных карт глубины – уже нет, так как распределение по сцене очень неравномерно.

Каждой из этих проблем посвящен отдельный раздел в статье, в каждом из которых авторы предлагают изменения и дополнения к алгоритму, позволяющие решить эти проблемы. Большая часть статьи посвящена обсуждению направленных источников, но все идеи и алгоритмы могут быть легко адаптированы к другим типам источников, там где нужны дополнительные детали использования алгоритмов для других типов источников, приводятся необходимые пояснения.

2. КРАТКОЕ ОПИСАНИЕ АЛГОРИТМА КАРТ ГЛУБИНЫ.

Основная идея этого алгоритма – отрисовать «слепок глубины» сцены текстуру с точки зрения источника света (shadow map), и потом использовать эту информацию для отображения теней. Это подразумевает проецирование этой текстуры на сцену, что дает в каждой точке информацию о расстоянии ближайшего к источнику объекта по лучу, доходящему до точки. Очевидно, что если это расстояние меньше, чем расстояние от самой точки до источника, то точка находится в тени, если нет – то точка не затенена.

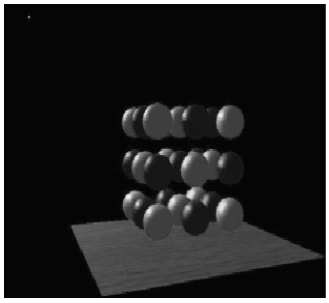

Рис 1. Сцена без теней.

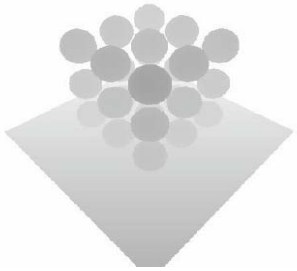

Рис 2. Слепок глубины с точки зрения источника света

В качестве примера на рисунке 1 изображена некая сцена, для которой необходимо построить тени. Для этого мы отрисовываем слепок глубины сцены с позиции источника (рис. 2), после этого проецируем этот слепок на сцену и сравниваем со значением в глубины в каждой точке.
На рисунке 3 равномерным серым цветом показаны те точки, где значения глубины в карте глубины и в точке совпадают, эти точки не затенены, все остальное – в тени. После учета отсутствия вклада источника света в тех точках, которые оказались затенены, мы получим изображение сцены с тенями (рис. 4).

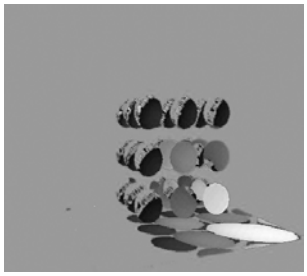

Рис. 3. Разница между расстоянием до источника и значением в слепке.

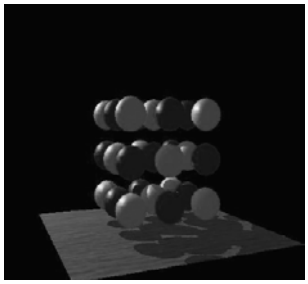

Рис 4. Результат – сцена с тенями.

2.1. Проблемы разрешения карты глубины.

Фактически, разрешение карты глубины определяет качество тени на экране (рис. 5).

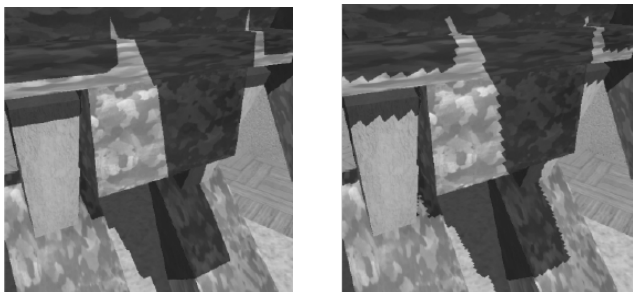

Рис. 5. Различие качества при недостатке разрешения.

Проблема заключается в том, что проекция устроена так, что если текстурная площадь распределена в пространстве равномерно, то тексели, находящиеся вблизи от камеры, займут достаточно много места на экране, и наоборот, тексели вдали от камеры займут мало места. В результате, тексельная площадь распределяется по экрану очень неравномерно и чаще всего ее очень не хватает вблизи, и она слишком избыточна вдалеке.

В алгоритме перспективных карт глубины [2] предлагается отрисовывать и накладывать карту глубины не в обычном мировом пространстве, а в пост-проективном. Пост-проективное пространство – то пространство, которое получается после проецирования сцены камерой. Фактически, именно из-за этого преобразования близкие объекты становятся больше, а далекие – меньше. Если использовать метод карт глубины в этом пространстве, близкие объекты займут больше текстурной площади, а далекие – меньше.

На рисунке 6 видно, как переход в пост-проективное пространство изменяет размеры объектов.

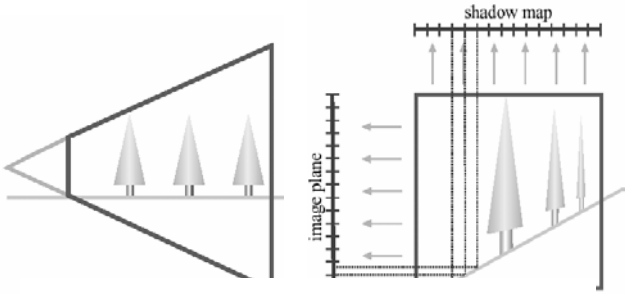

Рис.6. Переход в пост-проективное пространство.

Как видно из рисунков, если источник светит сверху, то действительно алгоритм полностью решает проблему – размер текселя на экране точно соответствует его размеру в сцене. К сожалению, это так только для вертикального направления источника и близких к нему направлений, если же источник светит при другом угле, все становится намного хуже.

3. ВИРТУАЛЬНЫЕ КАМЕРЫ.

В случаях, когда источник находится позади камеры, существенный вклад в затенение видимой части сцены вносят объекты, находящиеся позади камеры. Но так как

проективное преобразование переносит объекты, находящиеся за камерой, в участок пространства за плоскостью, соответствующей бесконечности в обычном пространстве, возникает вопрос как заносить эти объекты в карту глубины.

Проблема заключается в том, что результате проекционного преобразования меняется последовательность точек на луче из источника (рис.7). В этом случае авторы предлагают «виртуально» сдвинуть камеру назад, чтобы обеспечить наличие всех объектов, отбрасывающих тени, в пирамиде видимости (рис. 8).

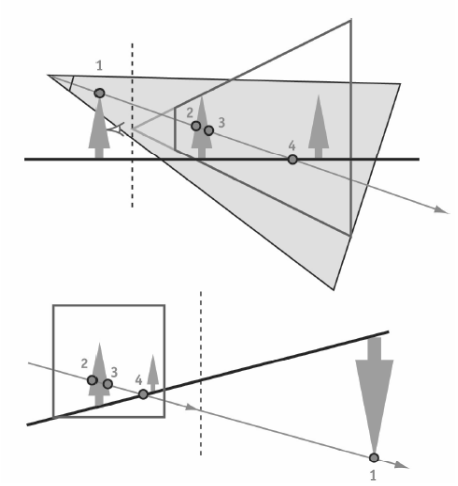

Рис.7. Объекты за камерой после проекции.

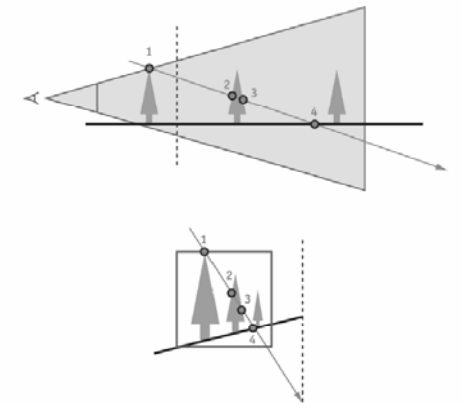

Рис.8. Использование «виртуальных камер».

На практике применение такого метода означает серьезную потерю качества, так как такой «виртуальный сдвиг» сильно уменьшает эффективное разрешение карты глубины, и объекты близкие к настоящей камере становятся меньше. Все это приводит к тому, что распределение текстурной площади очень неэффективно. Положение становится тем хуже, чем больше нужно сдвигать камеру назад, что определяется размером объектов, отбрасывающих тени. На рис. 9 показано, насколько меняется качество при сдвиге на расстояние порядка ближней плоскости отсечения (что типично много меньше размера объектов).

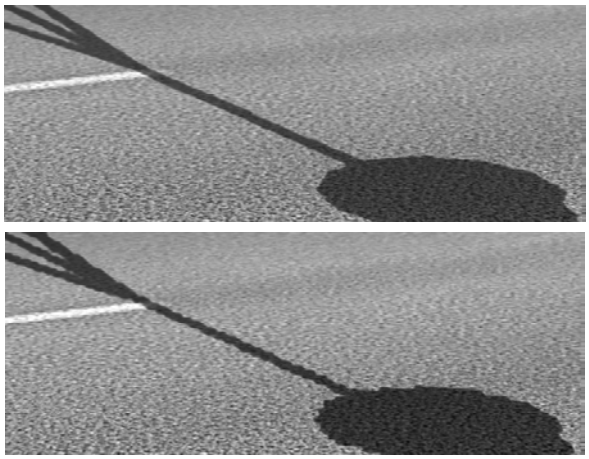

Рис 9. Разница в качестве с небольшим сдвигом.

Другая проблема заключается в минимизации этой сдвижки, что необходимо для достижения лучшего качества. Это подразумевает анализ сцены, определение потенциальных затеняющих объектов и так далее, из чего следует, что всегда будут резкие скачки качества тени в тот момент, когда объект перестает быть потенциальным затенителем. В этом случае сдвижка мгновенно изменяется, и также мгновенно изменяется и качество тени.

Предлагаемое решение состоит в построении специальной проекционной матрицы, которая может восстановить правильный порядок точек на луче. Рассмотрим подробнее картину в пост-проективном пространстве после применения изначального преобразования без виртуальных камер (Рис.10).

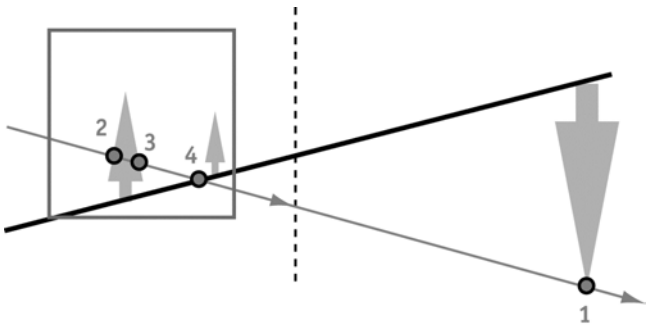

Рис 10. Пост-проективное пространство без виртуальных сдвигов.

Проблема заключается в том, что луч должен выйти из точки, соответствующей источнику, пройти через точку 1, уйти на минус бесконечность, перейти на плюс бесконечность, и снова дойти до источника, захватив точки 2, 3 и 4. Оказывается, можно построить проекционную матрицу, соответствующую такому «невозможному» ходу луча. Достаточно всего лишь поставить расстояние до ближней плоскости отсечения отрицательным числом, а до дальней – положительным (Рис. 11.)

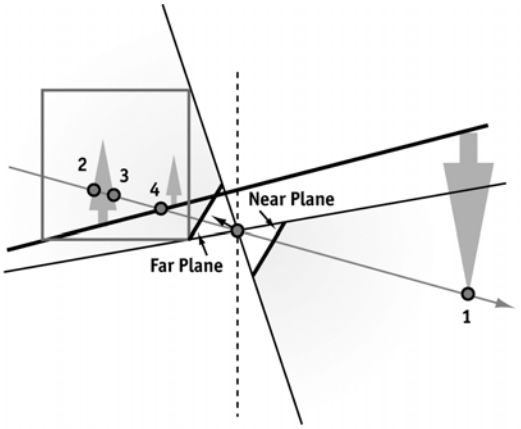

Рис 11. «Инверсная» проекционная матрица.

В простейшем случае, $|Z_n| = |Z_f| = a$, где a достаточно мало, чтобы весь куб видимости был внутри. Тогда матрица проекции будет выглядеть как (строковой порядок записи):

$$\begin{pmatrix} c & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & -QZ_n & 1 \\ 0 & 0 & Q & 0 \end{pmatrix} \rightarrow \begin{pmatrix} c & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & \frac{1}{2}a & 1 \\ 0 & 0 & \frac{1}{2} & 0 \end{pmatrix}$$

$Q = Z_f / (Z_f - Z_n) = a / (a - (-a) = ½$

И формула для финального значения Z, которое записывается в карту глубины:

$$Z_{psm} = Q(1 - \frac{Z_n}{Z}) = \frac{1}{2}(1 + \frac{a}{Z})$$

$Z_{psm}(-a) = 0$, и $\lim_{z \to -\infty} Z_{psm}(Z) = \frac{1}{2}$. В то же время значение $\frac{1}{2}$ соответствует плюс бесконечности, и при движении от плюс бесконечности к дальней плоскости отсечения Z_{psm} увеличивается, достигая значения 1 на дальней плоскости отсечения. Таким образом, луч проходит все точки в нужном порядке, и не нужно использовать никаких дополнительных сдвигов в построении камеры для перехода в пост-проективное пространство.

Такая матрица работает только в пост-проективном пространстве, так как обычно все точки за плоскостью, соответствующей бесконечности, имеют четвертую однородную координату $w < 0$, и поэтому не могут быть отображены. Но для второй проекции в пост-проективном пространстве эти точки расположены позади камеры, w-координата инвертируется еще раз и становится положительной.

Таким образом, мы добиваемся лучшего качества тени без всякого анализа геометрии и связанных с этим артефактов.

Единственный недостаток такого подхода в том, что он требует большей точности значения в карте глубины, так как захватываются большие объемы Z-значений. Впрочем, точности 24 бит достаточно для большинства сцен, используемых на практике, такая точность поддерживается современными графическими процессорами.

4. ПОСТРОЕНИЕ КАМЕРЫ ИСТОЧНИКА СВЕТА.

Другая важная проблема с перспективными картами глубины в том, что качество тени зависит от взаимного положения камеры и источника света. В случае вертикального направленного источника проблемы алиасинга полностью решаются, в то время, как если источник светит по направлению камеры, проблемы алиасинга существенны.

Это неизбежно, если пытаться удержать весь куб видимости в одной текстуре карты глубины, так как в этом случае приходится расширять угол раствора камеры источника света до значений, близких к 180º, что означает плохое качество вблизи от камеры (Рис.13).

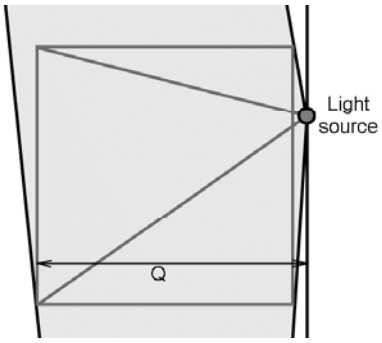

Рис.13. Камера источника света при низком источнике.

Ситуация тем хуже, чем ближе к кубу находится источник в пост-проективном пространстве. Направленные источники в пост-проективном пространстве находятся на плоскости, соответствующей бесконечности, расстояние до которой:

$$Z_\infty = Q = \frac{Z_f}{Z_f - Z_n},$$

для открытых сцен вполне нормальны значения $Z_n = 1$, $Z_f = 4000$, следовательно, $Q = 1.0002$, и источник находится на расстоянии 0.0002 от задней грани, что очень близко к кубу. Вообще, отношение Z_f / Z_n обычно больше 50, что соответствует $Q = 1.02$, что достаточно близко, чтобы вызывать проблемы.

Два предлагаемых решения подходят к проблеме с разных сторон – в первой части рассматривается алгоритм, который строит камеру источника только для необходимой части куба видимости, во второй части используются несколько карт глубины.

4.1. Усечение куба видимости.

Одно из возможных решений проблемы довольно очевидно – действительно, тени должны быть только на самих объектах – а их объем обычно меньше всей пирамиды видимости, особенно близко к дальней плоскости отсечения, и если

настроить камеру только на них, а не на весь куб, качество улучшится. Разумеется, настройка камеры производится по приближенному представлению геометрии (Рис.14).
В статье [2] упоминался этот подход, но камера настраивалась на все объекты в сцене, как получающие, так и отбрасывающие тени. Но так как уже не нужны «виртуальные» камеры, можно настраиваться только на объекты, принимающие тени, что более эффективно.

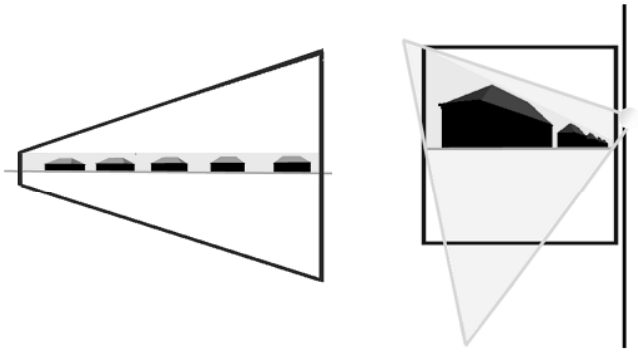

Рис. 14. Отсечение куба на основе ограничивающих объемов.

На практике достаточно использовать достаточно приближенные ограничивающие объемы, сохраняя хорошее качество – достаточно примерно показать, какие участки сцены требуют затенения. Для открытых сцен это примерная карта высот местности, для закрытых – примерная геометрия комнат и т.д.

Формализуем алгоритм просчета камеры источника света, учитывающей объекты, принимающие тень на основе приближенных ограничивающих объемов, описывающих сцену. Камера задается позицией, направлением, вектором, указывающим направление вверх и параметрами проекционной матрицы. Большинство из этих параметров заранее задано или напрямую следуют из других.

- Позиция задана изначально, это позиция источника света в пост-проективном пространстве.
- На практике, направление вектора, указывающего направление вверх, почти не сказывается на качестве.
- Параметры проективной матрицы определяются предыдущими параметрами.

Таким образом, задача сводится к нахождению направления камеры. Предлагается следующий алгоритм:

1. Просчитать вершины операции твердотельного моделирования $\bigcup_i (B_i \bigcap F)$, где B_i – i-й ограничивающий объем, F – пирамида видимости. После этого шага мы получим все вершины, которые должна «увидеть» камера источника (кстати, на основе этих точек можно найти приближенное значение ближней плоскости отсечения, обсуждавшееся в разделе 3).

2. Перевести все эти точки в пост-проективное пространство и найти наилучшее направление камеры. Критерий – минимальность угла конуса, построенного из позиции источника этому по направлению и захватывающего весь набор ключевых точек. Алгоритм нахождения такого конуса работает за время, линейное от количества точек в наборе и полностью аналогичен нахождению минимальной описывающей сферы для произвольного набора точек [3].

Так как достаточно грубого описания сцены и камера строится за линейное время от количества ограничивающих объемов, описывающих сцену, то построение оптимальной камеры является практически несущественными вычислительными ресурсами для современных приложений.

Описанный алгоритм эффективен для широких и направленных источников в открытых сценах, качество тени в плохих для оригинального алгоритма случаях становится гораздо лучше. (Рис. 15).

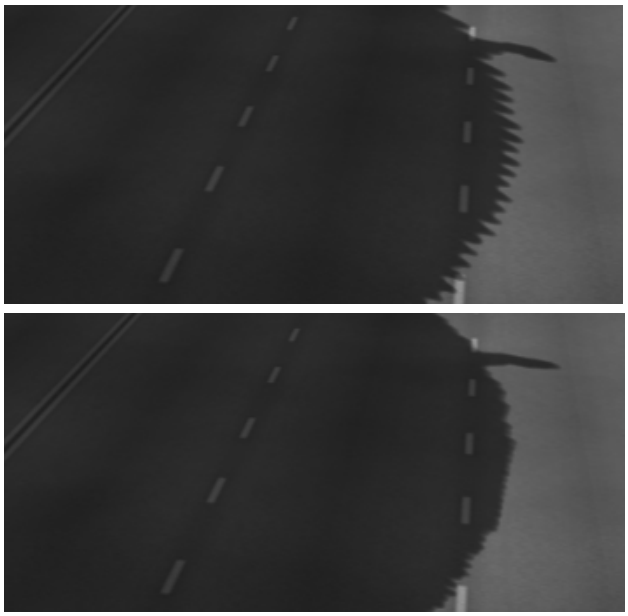

Рис.15. Улучшение качества при использовании усечения куба видимости.

4.2. Использование кубических текстур.

Хотя предыдущий подход часто эффективен, есть случаи, когда это не так. Например, в случае густо заполненных сцен, когда объекты заполняют всю пирамиду видимости, или когда нет возможности использовать описывающие объемы. Кроме того, предыдущий подход не работает для точечных источников.
Более общий подход заключается в использовании кубических текстур карт глубины. Большинство источников становятся точечными в пост-проективном пространстве, и естественно использовать для них методы построения карт глубины для точечных источников. Но в пост-проективном пространстве более эффективным становится не классическое распределение сторон кубической текстуры, так как нужна информация только для куба видимости.
Предлагаемое решение – использовать в качестве площадок для граней кубической текстуры грани куба видимости, невидимые с точки зрения источника.
Для направленного источника количество используемых сторон кубической карты может варьироваться от 3 до 5 (Рис.16). Максимальное количество сторон используется для

низких источников, когда действительно требуются дополнительные текстурные площади. Эти рисунки почти не изменяются и для источников других типов, находящихся вне куба.

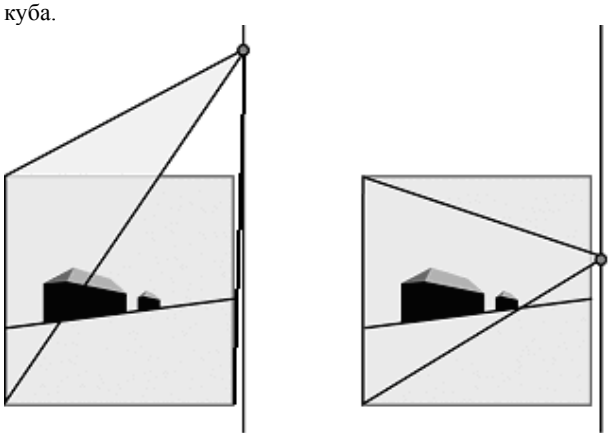

Рис.16. Использование кубических карт для направленного источника.

Для точечного источника, находящегося внутри куба нужно использовать все 6 сторон кубической текстуры, и они все также располагаются на сторонах куба (Рис.17).

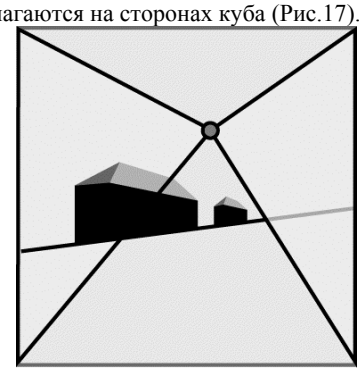

Рис.17. Использование кубических карт для точечного источника.

Можно сказать, что карты глубины формируют кубическую карту «со смещенным центром», которая отличается от обычной кубической карты только дополнительным вектором, добавленным к текстурным координатам.

Текстурная координата = позиция точки – позиция источника

Устанавливая площадки для граней кубической текстуры таким образом, распределение текстурной плоскости становится пропорционально размерам объектов экране, и качество перестает зависеть от положения источника в проективном пространстве. Размер текселя на экране определяется только углом наклона плоскости, на которую он проецируется, и так как достаточно легко регулировать параметры освещенности, ограничивая действие затенения на плоскости с разными углами, проекция текселя на экран находится в гарантированных пределах.

Так как отрисовка карты глубины – достаточно простая операция для графических ускорителей (требуется только запись в буфер глубины), определяющим фактором становится скорость заливки. Таким образом, нет разницы, использовать ли кубическую карту или одну текстуру, если общая площадь сторон кубической карты и этой текстуры совпадают. Эксперименты показывают, что при одинаковой текстурной площади кубические карты показывают лучшее среднее качество и ведут себя гораздо более предсказуемо.

Для этого подхода можно использовать как настоящую кубическую текстуру, так и несколько отдельных текстур, накладываемых на объект вместе при отрисовке теней.

5. ПРОБЛЕМЫ ТОЧНОСТИ СРАВНЕНИЯ.

При использовании перспективных карт глубины становятся важными проблемы точности сравнения значения в карте глубины с реальной глубиной объекта. Для обычных карт глубины эти проблемы решались добавкой небольшой константной «сдвижки» (англоязычный термин - *bias*)[4]. Для перспективных карт глубины проблема становится серьезнее – эта сдвижка не может быть константой, так как распределение текселей по сцене очень неравномерно и точность сравнения должна быть разной в разных местах сцены и при разных положениях источника света.

Рассмотрим два случая положения источника в пост-проективном пространстве – близко и далеко от границ куба видимости (Рис. 18).

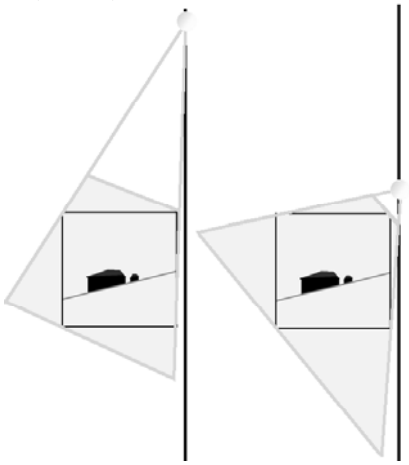

Рис. 18. Источник близко и далеко от куба видимости.

Проблема в том, что соотношение Z_f/Z_n, являющееся определяющим в распределении значений в карте глубины сильно отличается в этих двух случаях. Константная сдвижка после двойной проекции будет означать совершенно разные сдвиги в мировом пространстве, одно значение будет корректным для одного положения источника, но не для другого и наоборот. Это происходит потому, что соотношение Z_f/Z_n меняется в огромных пределах – источник может быть бесконечно удален от куба, а при другом положении камеры быть очень близко к его границе.

Более того, даже при фиксированном положении источника в пост-проективном пространстве часто не удается найти подходящей константы, корректно работающей для всей сцены. Так как проективное преобразование приводит к неравномерному распределению текселей и точностей сравнения для них, значение сдвижки должно зависеть еще и от положения затеняемой точки в сцене. На рисунке 19

показаны типичные артефакты попыток установить константное значение для всей сцены.

Рис. 19. Артефакты константой сдвижки.

Проблема осложняется тем, что возможности влияния на значение сдвижки в вершине крайне ограничены – в современных приложениях вершины обрабатываются графическим ускорителем, и функциональность и размер вершинных программ крайне ограничен, что не позволяет в полной мере анализировать результаты двойной проекции и вычислять необходимую точность сравнения строгими математическими методами.

Предлагается использовать сдвижку в мировом пространстве, и переводить эту сдвижку в пост-проективное пространство самими матрицами проекции. Так как это такой перевод учитывает распределение значений после применения проекций, такая сдвижка будет корректной для любых положений камеры и света. Кроме того, эту сдвижку в мировом пространстве можно легко привязать к размеру текселя, чтобы устранить артефакты, связанные с неравномерностью тексельного размера в сцене.

Все эти вычисления легко выполняются вершинными программами акселератора.

Формула для координаты точки после сдвижки и преобразования в систему координат карты глубины выглядит следующим образом:

$$P_{biased} = (P_{orig} + L(a + bL_{texel}))M$$

где P_{orig} – изначальная точка, L – направление от источника в мировом пространстве, L_{texel} – размер текселя в мировом пространстве, M – матрица перевода в пространство карты глубины, a и b – коэффициенты сдвижки.

Размер текселя в мировом пространстве может быть примерно посчитан с помощью простых матричных вычислений. Переведем точку в пространство карты глубины, сдвинем на расстояние порядка размера текселя, не изменяя глубины, и переведем обратно. Разница в координатах и есть характерный размер текселя.

$$L_{texel} = \left| P_{orig} - (P_{orig}M + c)M^{-1} \right|^2$$

$c = (\,1/S_x\,,\ 1/S_y\,, 0)$, где S_x и S_y – размеры карты глубины.

Очевидно, все эти преобразования (кроме вычисления квадрата длины) можно привести к одной результирующей матрице.

$L_{texel} = \left| P_{orig} M` \right|^2$, где $M`$ включает преобразование, сдвиг, трансформацию обратно и вычитание.

Разумеется, это скорее эмпирическое приближенное решение, так как использовать корректные вычисления слишком дорого.

Таким образом, вычисление сдвига свелось к умножению на матрицу, простым двум векторным операциям, и еще одному умножению на матрицу. Такие вычисления вполне по силам вершинным программам акселератора.

Результат таких применения этих вычислений показан на рисунке 20.

Рис.20. Сдвижка, вычисленная вершинными программами.

6. ЗАКЛЮЧЕНИЕ.

Авторы предложили ряд дополнений и исправлений алгоритма перспективных карт глубины, делающих его устойчивым, надежным и предсказуемым методом управления распределением текстурной площади карты глубины по сцене, решая главную проблему алгоритмов карт глубины – плохое качеств, связанное с недостатком разрешения, а также проблемы точности сравнения и стабильности алгоритма.

Модифицированный алгоритм перспективных карт глубины, описанный в статье, был реализован в виде системы отображения теней, и успешно используется в системах отображения, выполняя требования быстродействия, налагающиеся условиями реального времени.

Ниже приведены кадры, снятые во время работы такой системы, полигональная сложность сцен – от 100.000 до 500.000 полигонов в кадре, время генерации кадра остается в границах 30 мсек.

7. ССЫЛКИ НА ЛИТЕРАТУРУ.

1. Andrew Woo, Pierre Poulin, Alain Fournie, *"A Survey of Shadow Algorithms", IEEE Computer Graphics & Applications, 1990.*
2. Mark Stamminger, George Dettrakis, *"Perspective Shadow Maps", Proc. of SIGGRAPH 2002.*
3. Bernd Gartner, *"Fast and Robust Smallet Enclosing Balls", 1999*
4. Cass Everitt, Ashu Rege, Cem Cebenoyan, *"Hardware Shadow Mapping", NVidia technical document*