

# Моделирование теней в виртуальных сценах с направленными источниками освещения<sup>1</sup>

А.В. Мальцев, М.В. Михайлюк

**Аннотация.** Предлагаются эффективные методы и алгоритмы расчета освещенности трехмерных виртуальных сцен в реальном режиме времени с учетом теней от направленных источников света. Методы основаны на использовании теневых карт (shadow maps). Вводится расширенная модель Блинна для расчета освещенности на поверхности визуализируемого виртуального объекта, учитывающая затенение и плотность тени. Детально рассматривается процесс создания теневых карт для направленных источников света и применение этих карт в реальном режиме времени.

**Ключевые слова:** визуализация, карты теней, моделирование теней.

## Введение

Из множества графических эффектов, которые приближают трехмерную виртуальную сцену к реальному миру, тени являются одним из наиболее важных. Объекты, освещаемые лучами солнечного света, естественными или искусственными источниками, отбрасывают тени. Это неотъемлемая часть окружающего нас мира. Поэтому для правильного восприятия визуализируемой виртуальной среды очень важно наличие в такой среде теней. В настоящее время наиболее популярными и широко используемыми считаются два метода моделирования теней в трехмерных сценах: «теневые объемы» (shadow volumes) и метод, основанный на использовании теневых карт (shadow maps). Каждый из них имеет свои достоинства и недостатки, исходя из которых программист делает выбор, что лучше использовать для стоящей перед ним задачи. Метод «теневых объемов» основывается на построении для каждой пары,

состоящей из источника света и объекта виртуальной сцены, пространственной фигуры, внутри которой освещение от источника, входящего в пару, отсутствует благодаря тому, что данная область сцены закрыта рассматриваемым объектом от попадания в нее прямых лучей. Одним из основных и серьезных минусов метода «теневых объемов» является его прямая зависимость от сложности геометрии сцены. Этот недостаток не замечен в простейших сценах, но в сложных и объемных виртуальных пространствах с множеством объектов количество требуемых дополнительных расчетов для моделирования теней становится критическим при выполнении алгоритма в реальном режиме времени. В данной статье мы не будем рассматривать реализацию «теневых объемов», а подробно остановимся на втором методе, использующем теневые карты, который не имеет прямой зависимости от сложности геометрии сцены и успешно применяется в режиме реального времени визуализации.

<sup>1</sup> Работа выполняется при поддержке РФФИ грант № 09-07-00149а

## 1. Карты теней и их построение для направленных источников света

Алгоритм моделирования теней с помощью теневых карт состоит из двух шагов. На первом шаге производится формирование буфера глубины виртуальной сцены из позиции источника света. Этот буфер сохраняется в виде текстуры специального формата в градациях серого, которую принято называть *теневой картой*. На втором шаге происходит визуализация объектов сцены из позиции наблюдателя. При этом для каждой видимой точки каждого объекта выполняется специальный тест глубины. Именно, если глубина рассматриваемой точки относительно источника света больше соответствующего значения глубины, взятого из теневой карты, то данная точка находится в тени; если же эти значения приблизительно равны, то точка считается освещенной источником света.

В рамках данной статьи мы будем рассматривать только направленные источники света, такие как spotlight («фары») и target direct (прожекторы, освещающие параллельными лучами), при этом для сокращения называя их просто источниками света. Скажем пару слов об интересующих нас параметрах этих источников. Действие spotlight задается с помощью углов  $\alpha$  и  $\beta$ , где  $\alpha$  - это угол раствора конуса, в пределах которого распространяется освещение, а  $\beta$  - так называемый угол «горячего пятна», определяющий конус, внутри которого освещение от источника максимально (Рис. 1а). При этом должно соблюдаться условие  $0 < \beta \leq \alpha < 180^\circ$ . Вне конуса с углом  $\alpha$  освещение отсутствует, а между конусами оно равномерно изменяется от 0 до 1. Аналогично для target direct определены радиусы  $R$  и  $r$  цилиндров ( $0 < r \leq R$ ). Внутри цилиндра радиуса  $r$  освещение максимально (т.е. равно 1), вне цилиндра радиуса  $R$  освещение отсутствует (равно 0), а между цилиндрами - равномерно изменяется от 0 до 1 (Рис. 1б).

При рассмотрении источников такого вида, в отличие от полусферических (hemispherical) и всенаправленных (omnidirectional), для моделирования теней достаточно построения одного стандартного буфера глубины (теневого карты), плоскость которого перпендикулярна направлению источника освещения.

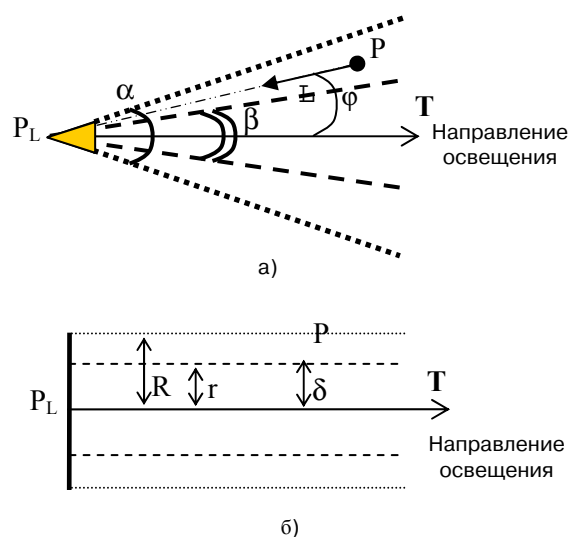


Рис. 1.

Для генерации теневой карты необходимо создать фиктивную камеру в точке расположения источника света и направить ее по вектору свечения. После этого с помощью стандартных функций OpenGL задается вид проецирования. Для источника типа spotlight задается перспективное проецирование (функция gluPerspective) с углом  $\alpha$  раствора камеры и величиной aspect=1. Ортогографическое проецирование (функция gluOrtho) для источника типа target direct определяется положениями отсекающих плоскостей left=bottom =  $-R$ , а right = top= $R$ . Далее отключается запись в буфер цвета, включается буфер глубины и осуществляется визуализация сцены. Необходимо отметить, что в дальнейшем, при использовании теневой карты, нам потребуются матрицы видового ( $M_{fv}$ ) и проекционного ( $M_{pr}$ ) преобразований, использованные для фиктивной камеры, поэтому их нужно сохранить. В результате описанной последовательности действий текущий буфер глубины будет содержать нужную нам информацию, которую надо одним из доступных способов (копирование, присоединение) перенести в текстуру, имеющую формат GL\_DEPTH\_COMPONENT (данный формат доступен через расширение OpenGL GL\_ARB\_depth\_texture) и являющуюся собственно теневой картой.

На Рис. 2 представлен пример простой виртуальной сцены с графином, стоящим на деревянной поверхности. Сцена освещается направленным источником света, теневая карта для которого изображена на Рис. 3.

Для генерации теневой карты и визуализации сцены в реальном режиме времени целесообразно применение прямого вывода изображения в текстуру с использованием современных программно-аппаратных средств, а именно технологии FBO (framebuffer objects). FBO – это расширение OpenGL, которое обеспечивает простой интерфейс для визуализации в контексты, отличные от буферов GL, предоставленных оконной системой. Framebuffer objects является одним из современных средств вывода изображения в текстуру и имеет преимущества по сравнению с более старой технологией использования пиксельных буферов (P-buffers) - более высокую скорость визуализации, независимость от используемой программной платформы (ОС), простоту настройки параметров буфера.

Процесс работы с framebuffer object коротко можно описать так.

1) Создаётся FBO – объект буфера кадра (framebuffer object). Он не зависит от оконной системы и является скорее неким «контейнером» установок OpenGL.

2) К FBO подсоединяются логические буферы - созданные пользователем текстуры и буферы вывода (renderbuffers). Для этого используются так называемые точки присоединения (attachment points).

3) Проверяется поддержка аппаратурой желаемой конфигурации FBO.

4) FBO устанавливается в качестве текущей цели для вывода изображения.

5) Осуществляется рисование изображения в заданные логические буферы.

6) Происходит возврат к буферу кадра основного окна программы.

Итак, визуализируя в FBO сцену из позиции источника света, мы получаем готовый буфер глубины, находящийся в видеопамяти и пригодный для использования в качестве теневой карты. Данный метод позволяет избежать лишнего копирования данных из видеопамяти в оперативную память и обратно, которое требо-

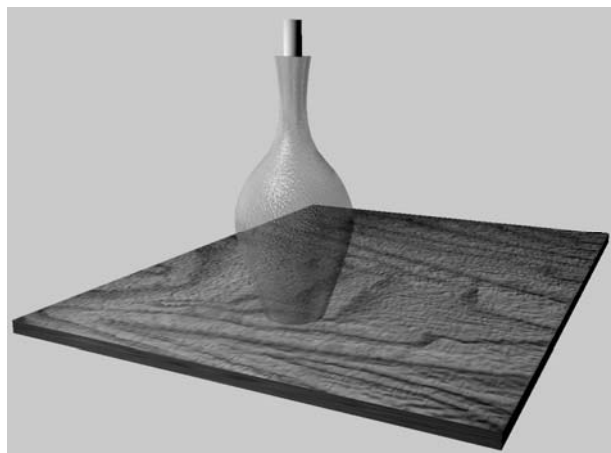


Рис. 2.

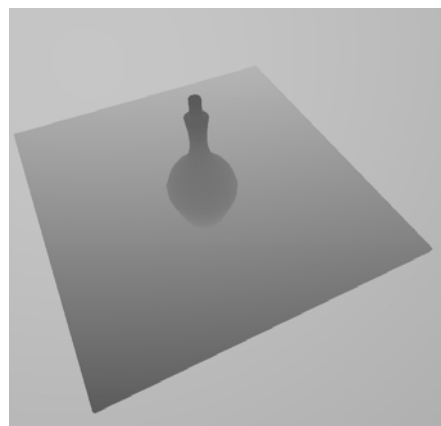


Рис. 3.

валось бы в других реализациях, что заметно повышает скорость вывода изображения в реальном режиме времени и дает возможность генерировать карты теней с большим разрешением. Для подключения FBO необходимо инициализировать расширение OpenGL `GL_EXT_framebuffer_object`. С более подробной информацией по использованию FBO можно ознакомиться в [5].

## 2. Расширенная модель расчета освещенности в точке объекта

При реализации попиксельного освещения трехмерных виртуальных сцен обычно используется модель расчета освещенности, учитывающая рассеянную, диффузную и зеркальную составляющие света. Рассмотрим одну из наиболее часто применяемых моделей – модель Блинна.

Пусть имеется один источник света и некоторая точка  $P$  на поверхности объекта. Обозначим через  $\mathbf{N}$  вектор нормали в  $P$ , через  $\mathbf{L}$  – вектор, направленный из  $P$  к источнику освещения, а через  $\mathbf{V}$  – вектор, направленный из  $P$  к наблюдателю (Рис. 4). Мы предполагаем, что все эти векторы нормализованы, т.е. имеют единичную длину.

Для начала необходимо определить, попадает ли рассматриваемая точка  $P$  в конус (цилиндр) освещения от источника. Пусть  $\mathbf{T}$  – единичный вектор направления свечения,  $\varphi = \angle(-\mathbf{L}, \mathbf{T})$  (Рис. 1), а  $I$  – искомая освещенность в точке  $P$ . Здесь мы считаем, что все точки и векторы заданы в локальной системе координат объекта, которому принадлежит  $P$ . Так как векторы  $\mathbf{L}$  и  $\mathbf{T}$  – единичные, то

$$(-\mathbf{L} \cdot \mathbf{T}) = \|-\mathbf{L}\| \|\mathbf{T}\| \cos \varphi = \cos \varphi. \quad (1)$$

Допустим, что источник освещения имеет тип spotlight. Если угол  $\varphi$  больше угла  $\alpha/2$ , то точка не попадет в конус освещенности. В графическом конвейере проще посчитать значение  $\cos \varphi$ , поэтому условие непопадания точки в конус освещенности запишется в виде  $\cos \varphi < \cos \frac{\alpha}{2}$ . В этом случае  $I = 0$ .

Для попадания точки  $P$  в полубесконечный цилиндр освещения источника target direct должны быть выполнены два условия. Во-первых, необходимо проверить, принадлежит ли точка  $P$  тому же полупространству, что и цилиндр освещения, другими словами, не находится ли она за источником света. Для этого нужно найти  $\cos \varphi$  по формуле (1) и сравнить полученное значение с нулем. Если  $\cos \varphi < 0$ , то рассматриваемая точка находится за источником света и  $I = 0$ . Во-вторых, надо проверить, лежит ли точка внутри цилиндра освещенности. Для этого вычислим  $\delta = \rho(P, \mathbf{T})$  – расстояние между  $P$  и осью  $Q$  цилиндра (Рис. 5). Имеем

$$\delta = \|\overline{P_L P}\| \sin \varphi = \|\mathbf{T}\| \|\overline{P_L P}\| \sin \varphi = \|\mathbf{T}, \overline{P_L P}\|,$$

(здесь вычисление  $\delta$  сводится к нахождению модуля векторного произведения, т.к. эти величины легко вычисляются в графическом конвейере). Если  $\delta > R$ , то рассматриваемая точка  $P$  лежит вне цилиндра освещения и, следовательно,  $I = 0$ .

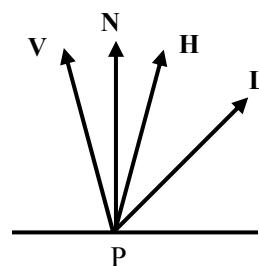


Рис. 4.

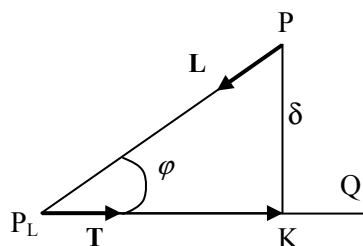


Рис. 5.

Если точка  $P$  попадает в конус (цилиндр) освещенности, то необходимо определить для нее значение  $I$ . Пусть вектор  $\mathbf{H} = (\mathbf{L} + \mathbf{V}) / \|\mathbf{L} + \mathbf{V}\|$  – нормализованный средний вектор между  $\mathbf{L}$  и  $\mathbf{V}$ . Тогда для расчета освещенности  $I$  в точке  $P$  будем использовать следующую формулу:

$$I = \lambda(k_A I_A + k_D I_D \max(0, \mathbf{L} \cdot \mathbf{N}) + k_S I_S \max(0, \mathbf{H} \cdot \mathbf{N})^s), \quad (2)$$

где  $k$  – коэффициенты отражения материалом соответственно рассеянной, диффузной и зеркальной компонент освещения, а  $I$  – интенсивности этих компонент, попадающие в точку  $P$ . Таким образом, первое слагаемое в формуле отвечает за рассеянную составляющую света, второе – за диффузную, а третье – за зеркальную. Это последнее слагаемое реализует модель Блинна для зеркальной составляющей освещенности, степень  $s$  (действительное число от 0 до 128) характеризует резкость бликов (чем выше  $s$ , тем блик меньше и резче). Функция  $\max$  введена, чтобы исключить «отрицательную» освещенность, когда угол между векторами (например,  $\mathbf{L}$  и  $\mathbf{N}$ ) больше  $90^\circ$  (в этом случае свет не может падать в точку  $P$ , а скалярное произведение окажется отрицательным).

Коэффициент  $\lambda$  отвечает за реализацию направленных источников света. Для spotlight:

$$\lambda = \begin{cases} 1, & \text{если } \varphi \leq \frac{\beta}{2} \\ \frac{\cos \varphi - \cos \frac{\alpha}{2}}{\cos \frac{\beta}{2} - \cos \frac{\alpha}{2}}, & \text{если } \varphi \in \left( \frac{\beta}{2}, \frac{\alpha}{2} \right] \end{cases},$$

а для target direct:

$$\lambda = \begin{cases} 1, & \text{если } \delta \leq r \\ \frac{R - \delta}{R - r}, & \text{если } \delta \in (r, R] \end{cases}.$$

Значения  $\lambda$  для случаев  $\varphi > \frac{\alpha}{2}$  и  $\delta > R$  не рассматриваются, так как такие ситуации были изложены выше.

Рассмотренная модель Блинна хорошо описывает освещенность в точке виртуального объекта, однако она не подходит при реализации теней, поскольку никак их не учитывает.

Для учета теней будем использовать параметры, применяемые в системе моделирования 3D Studio Max - коэффициент плотности тени *density* (вещественное число от 0 до 1) и цвет тени *sc* (*shadeColor* в формате RGB).

Также нам потребуется переменная *shade*, принимающая значение 1, если рассматриваемая точка поверхности объекта затенена, или 0, если тень в точке отсутствует.

Если обозначить через  $sh = density \cdot shade$ , через  $D$  - диффузную, а через  $S$  - зеркальную составляющие освещенности в модели Блинна, то в расширенной модели эти составляющие рассчитываются по следующим формулам:

$$D' = (1 - sh) \cdot D + sh \cdot sc, \quad S' = shade \cdot S.$$

Формула расширенной модели расчета освещенности в точке примет следующий вид:

$$I = \lambda(k_A I_A + (1 - sh) \cdot k_D I_D \max(0, \mathbf{L} \cdot \mathbf{N}) + sh \cdot sc + shade \cdot k_S I_S \max(0, \mathbf{H} \cdot \mathbf{N})^s) \quad (3)$$

### 3. Применение теневых карт для реализации эффекта теней

Для применения сгенерированной теневой карты необходимо во фрагментном шейдере реализовать специальный тест глубины пикселей относительно источника света. Рассмотрим его суть на простом примере.

Пусть имеется точка  $P$ , принадлежащая поверхности некоторого объекта виртуальной сцены, и пусть известны ее координаты в объектной системе координат (OCS). Для проведения теста глубины с использованием теневой карты необходимо найти координаты этой точки  $P$  в преобразованной системе координат нормализованного объема видимости (LNDCS') для фиктивной камеры. На Рис. 6 показана последовательность перехода от системы OCS к системе LNDCS'.

Формулу для преобразования координат точки  $P$  запишем в виде

$$P_{LNDCS'} = M_b \cdot M_{pr} \cdot M_{lv} \cdot M_m \cdot P_{OCS}, \quad (4)$$

где  $M_m$  - матрица модельного преобразования, переводящая координаты точки из объектной системы координат (OCS) в мировую (WCS),  $M_{lv}$  - матрица видового преобразования, переводящая координаты точки из мировой системы координат в видовую систему (LVCS) фиктивной камеры, и  $M_{pr}$  - проекционная матрица фиктивной камеры, осуществляющая преобразование из LVCS в систему координат нормализованного объема видимости (LNDCS). Матрица

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

осуществляет такое преобразование системы координат LNDCS, при котором ее начало перемещается в вершину (-1, -1, -1) куба видимости, а оси растягиваются в 2 раза (Рис. 7).

Матрицы  $M_{pr}$  и  $M_{lv}$  рассчитываются и сохраняются для каждого источника света на шаге генерации теневой карты. В целях оптимизации, если источник света статический (не изменяет положения и направления), то можно один раз рассчитать для него произведение матриц  $M_b \cdot M_{pr} \cdot M_{lv}$  и использовать это произведение в дальнейшем при вычислении формулы (4). Если положение или направление источника освещения изменяются с течением времени, то постоянным будет оставаться произведение  $M_b \cdot M_{pr}$ , которое также достаточно посчитать один раз и далее использовать в (4).

Пусть  $P_{LNDCS'} = (x_p, y_p, z_p)$ . Заметим, что  $z_p$  соответствует расстоянию (глубине) от точки  $P$  до источника освещения. В теневой карте запи-

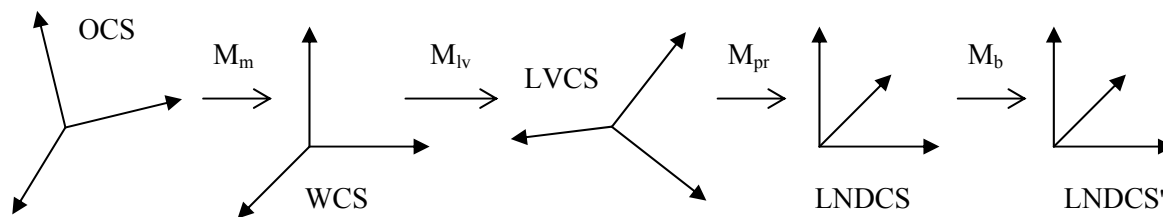


Рис. 6.

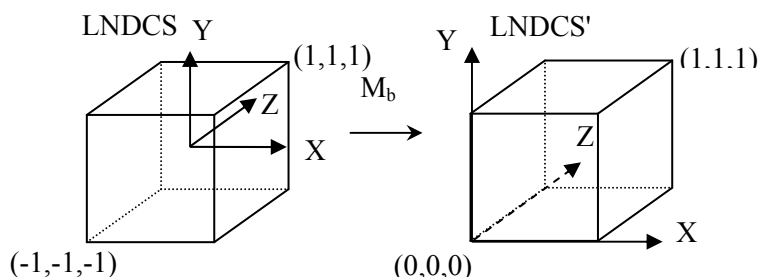


Рис. 7.

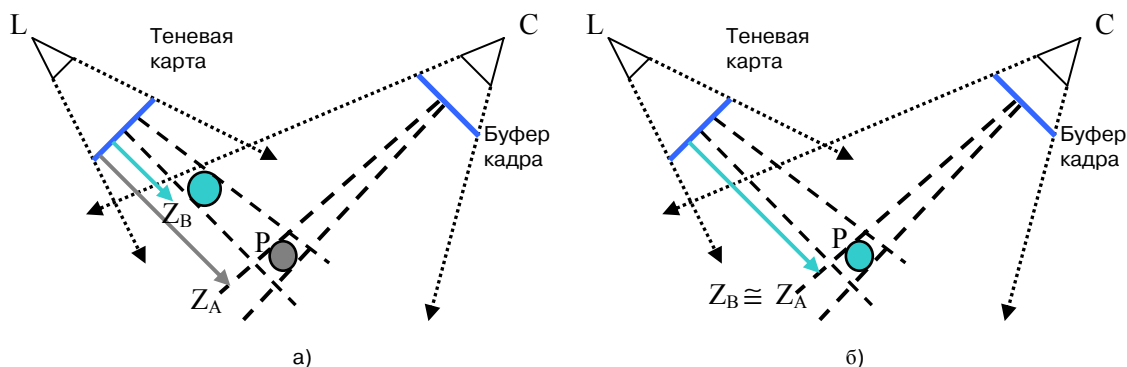


Рис. 8.

сано расстояние от источника освещения до ближайшей точки какого-либо объекта в том же направлении. Поэтому для определения освещенности точки по паре координат  $(x_p, y_p)$  из теневой карты выбирается записанное в ней значение глубины  $Z_B$  и сравнивается со значением  $Z_A = z_p$ . При этом возможны два варианта:

- $Z_B < Z_A$ , что соответствует ситуации, когда точка  $P$  закрыта от попадания в нее прямых лучей из источника участком поверхности какого-то объекта в сцене (Рис. 8а). Другими словами, точка  $P$  находится в тени;
- $Z_B \cong Z_A$  ( $Z_A$  и  $Z_B$  приблизительно эквивалентны), то есть в теневой карте записана с некоторой погрешностью глубина самой точки  $P$

(Рис. 8б), и, следовательно,  $P$  освещается источником.

При попиксельном вычислении освещенности в каждой точке каждого объекта сцены, попадающей в область освещения, используется ранее описанная расширенная модель Блинна, представленная формулой (3), в которой параметр  $shade$  будет определяться следующим образом:

$$shade = \begin{cases} 1, & \text{если } Z_B < Z_A \\ 0, & \text{если } Z_B \cong Z_A \end{cases}$$

На Рис. 9 представлено применение вышеописанных алгоритмов на примере сцены с полупрозрачным графином, который отбрасывает тень на деревянную поверхность.

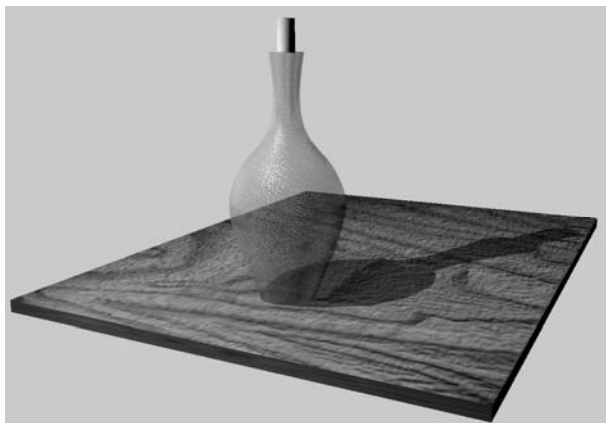


Рис.9.

## Литература

1. Mark J. Kilgard. Shadow mapping with today's opengl hardware. Technical report, CEDEC, 2001.
2. Mark Segal, et al. Fast shadows and lighting effects using texture mapping. In Proceedings of SIGGRAPH '92, pages 249-252, 1992.
3. Боресков А.В. Построение теней при помощи теневых карт (shadow maps). <http://www.steps3d.narod.ru/tutorials/shadow-maps-tutorial.html>
4. OpenGL Extension Registry. <http://www.opengl.org/registry/>
5. Framebuffer Object или скажем "нет" оконным контекстам. [www.gamedev.ru/community/opengl/articles/framebuffer\\_object](http://www.gamedev.ru/community/opengl/articles/framebuffer_object)

**Мальцев Андрей Валерьевич.** Ведущий программист Центра визуализации и спутниковых информационных технологий НИИСИ РАН. Окончил Московский государственный институт радиотехники, электроники и автоматики в 2008 году. Автор 9 научных публикаций. Область научных интересов: компьютерная графика, системы виртуальной реальности, информационные технологии. Специалист в области компьютерной графики. E-mail: [avmaltcev@mail.ru](mailto:avmaltcev@mail.ru)

**Михайлюк Михаил Васильевич.** Заведующий отделом программных средств визуализации Центра визуализации и спутниковых информационных технологий НИИСИ РАН. Московский государственный университет в 1975 году. Доктор физико-математических наук, профессор. Автор более 70-и научных работ. Область научных интересов: компьютерная графика, системы виртуальной реальности, информационные технологии. E-mail: [mix@niisi.ras.ru](mailto:mix@niisi.ras.ru)