

Лабораторная работа №6

«Освещение объектов»

Оглавление

Подготовка геометрических данных	2
Рисование сферы	2
8.1 Задание для самостоятельной работы	3
Освещение	3
Uniform данные	4
Вершинный шейдер	4
Фрагментный шейдер	5
Диффузное отражение	5
Определение материала	6
8.2 Задание для самостоятельной работы (по вариантам)	6
8.3 Задание для самостоятельной работы	6
Диффузное и фоновое отражение	6
8.4 Задание для самостоятельной работы	6
Зеркальное отражение	7
8.5 Задание для самостоятельной работы	7
Определение параметров источника света	7
8.6 Задание для самостоятельной работы	7
Определение точечного источника света	7
8.7 Задание для самостоятельной работы	7
Определение бесконечно удаленного источника света	7
8.8 Задание для самостоятельной работы (по вариантам)	7
Радиальное затухание интенсивности	8
8.9 Задание для самостоятельной работы (необязательное)	8
Имитация узконаправленного источника света	8
8.10 Задание для самостоятельной работы (необязательное)	8
8.11 Задание для самостоятельной работы	9
Реализация двухстороннего отображения	9
8.12 Задание для самостоятельной работы (необязательное)	9
Имитация тумана	10
8.13 Задание для самостоятельной работы (необязательное)	10
8.14 Задание для самостоятельной работы (необязательное)	10

Подготовка геометрических данных

Различные эффекты, связанные с освещением, наиболее наглядно будут проявляться на гладких поверхностях, например, на поверхности цилиндра, конуса, сферы или тора, поэтому рекомендуется выбрать и построить одно из этих тел.

Рисование сферы

Рисование треугольников сферы — нетривиальная задача. Из разработанных методов самый простой заключается в разбиении сферы на секции по широте и долготе. В примере кода 6.js сфера разделена по вертикали 16 линиями широты (`NUM_LATITUDE`) и по горизонтали 32 линиями долготы (`NUM_LONGITUDE`). На рисунке 1 показано, как она выглядит.

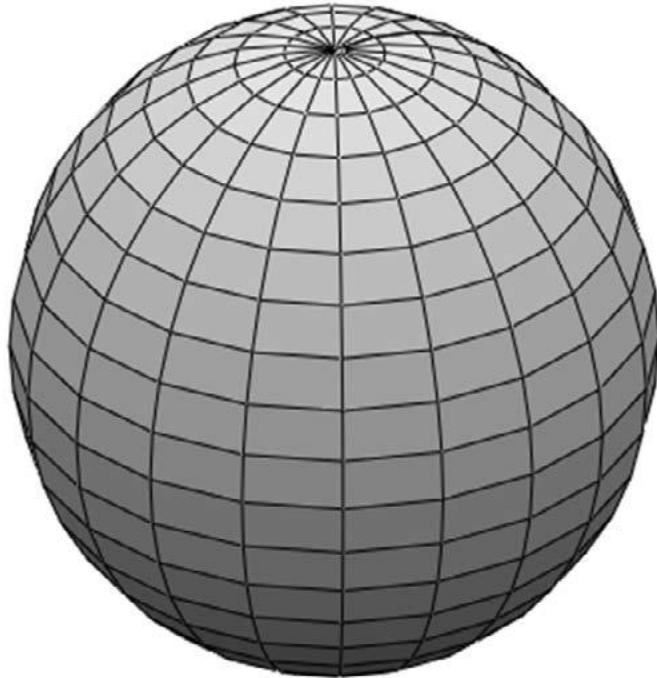


Рис. 1. Деление сферы по широте и долготе.

Количество вершин сферы легко вычислить. Есть вершина сверху, вершина снизу и сетка из вершин `NUM_LATITUDE` * `NUM_LONGITUDE`. Поэтому код устанавливает `NUM_VERTICES` равным `NUM_LATITUDE` * `NUM_LONGITUDE` + 2.

Для визуализации сферы нам нужно найти координаты (x, y, z) точек пересечения широты и долготы. Верхняя точка находится в точке $(0, 0, R)$, а нижняя — в точке $(0, 0, -R)$, где R — радиус сферы. Обозначив угол широты как φ , а угол долготы как ϑ , координаты остальных точек можно вычислить за пять шагов:

1. Используем широту для определения угла φ в радианах.
2. Используем долготу для определения угла ϑ в радианах.
3. Установим x равным $R \cos \varphi \cos \vartheta$.
4. Установим y равным $R \cos \varphi \sin \vartheta$.
5. Установим z равным $R \sin \varphi$.

Чтобы покрыть сферу треугольниками, приложению необходимо обращаться к каждой вершине несколько раз. Вместо повторения координат в буфере вершин эффективнее обращаться к вершинам по индексу. Для назначения индексов пример кода разбивает сферу на вертикальные области, называемые срезами. Сфера имеет один срез для каждой линии долготы, а на рисунке 2 последовательно показаны срезы сферы с четырьмя линиями широты.

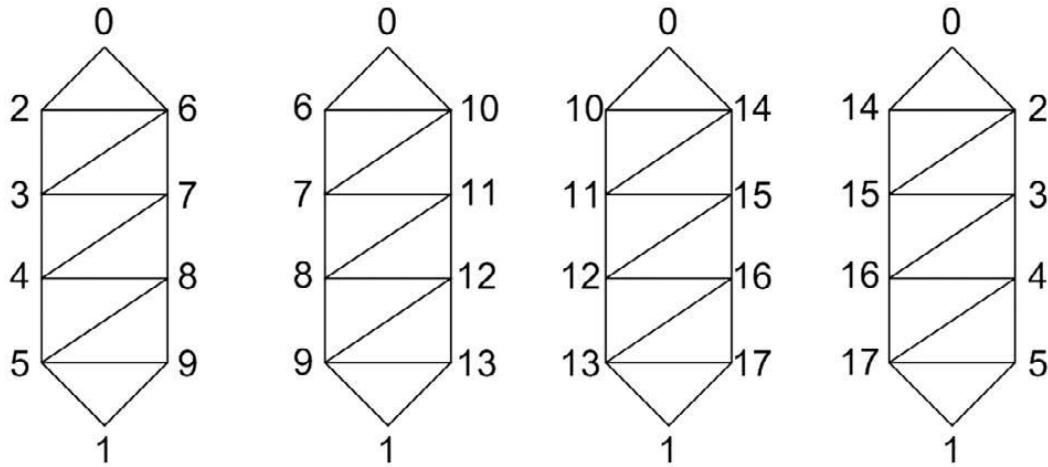


Рис. 2. Треугольники смежных срезов.

Как показано, каждый срез имеет треугольник вверху, один четырёхугольник для каждой линии широты (минус 1) и треугольник внизу. Можно нарисовать срезы связно, но для простоты мы сделаем каждый срез отдельно. То есть, как только мы достигнем нижней вершины, мы добавим примитивное значение перезапуска (0xffff) и начнём с вершины следующего среза.

Теперь мы можем определить необходимое количество индексов. Для первого треугольника требуется три, для каждого четырёхугольника — два, для последнего треугольника — один, а для примитивного индекса перезапуска — один. В сумме для каждого среза требуется $2 * \text{NUM_LATITUDE} + 3$ значения индекса. Общее количество требуемых индексов составляет $\text{NUM_LONGITUDE} * (2 * \text{NUM_LATITUDE} + 3) - 1$ значение (последний примитивный индекс перезапуска не нужен).

Мы присвоим индекс 0 верхней вершине и индекс 1 нижней вершине. Остальные значения индексов увеличиваются сверху вниз и слева направо. Например, вершины вдоль долготы 0 находятся в диапазоне от 2 до $\text{NUM_LATITUDE} + 1$.

Координаты вершин хранятся в массиве `vData`, а значения индексов — в массиве `iData`. Первые элементы `vData` — это координаты верхней точки сферы $(0, 0, R)$ и координаты нижней точки $(0, 0, -R)$. В результате индекс верхней точки равен 0, а индекс нижней точки — 1.

В конце цикла итераций по срезам к `iData` добавляется индекс 0xffff. Это значение перезапуска примитива, сообщающее рендереру, что следующий индекс начнёт новую фигуру (срез).

При построении поверхностей, вместе с расчетом координат вершин, нужно рассчитывать значения координат векторов нормалей.

8.1 Задание для самостоятельной работы

Рассчитайте значения нормалей в вершинах изображаемой фигуры и передайте их в вершинный шейдер.

Освещение

Чтобы двухмерная сцена выглядела трёхмерной, недостаточно просто разместить вершины в трехмерном пространстве. Для создания иллюзии глубины объекты должны по-разному отражать свет в зависимости от своей ориентации относительно источников света.

Освещение — это процесс изменения цвета объекта в зависимости от отражения падающего от источников света. Объекты, удалённые от источника света, должны быть темнее объектов,

расположенных рядом с источником света, а блестящие объекты должны отражать больше света, чем тусклые.

Разработано множество моделей для определения того, как свет отражается от поверхности объекта. Рассмотрим две популярные модели: модель Фонга и модель Блинна–Фонга. В обоих случаях методология состоит из трёх этапов:

1. Задание одного или нескольких источников света. Передача этой информации шейдерам с помощью uniform буфера.
2. Передача данных о векторах-нормалей шейдерам как часть буфера вершин.
3. Используя координаты источников света и векторов нормалей, вершинный или фрагментный шейдер рассчитывает цвет каждого фрагмента с помощью модели освещения.

Uniform данные

Создадим один uniform буфер, содержащий массив `uniformData`, который передадим в шейдеры.

Первые два элемента `uniformData` — это матрицы вида и вида-проекции, определяющие преобразование сферы.

Далее `uniformData` содержит ряд векторов для определения уровня освещённости:

- `centerPos` определяет центр сферы в системе координат вида;
- `viewerPos` определяет местоположение наблюдателя в системе координат вида;
- `lightPos` устанавливает положение источника света в системе координат вида;
- `ambient` — интенсивность фонового освещения;
- `diffuse` — интенсивность диффузного освещения;
- `specular` — интенсивность зеркального освещения;
- `shininess` — устанавливает показатель степени для зеркального отражения равным 1.5.

Вершинный шейдер

Вершинный шейдер выполняет три операции:

1. Преобразует каждую вершину с использованием матрицы вида-проекции.
2. Вычисляет вектор нормали.
3. Передаёт данные во фрагментный шейдер.

Чтобы это стало возможным, код шейдера определяет две структуры. Структура `InputData` содержит данные из uniform буфера. Структура `OutputData` содержит данные, которые вершинный шейдер передаёт фрагментному шейдеру.

Может показаться странным, что большинство векторов в структуре `InputData` имеют тип `vec4f` вместо `vec3f`. Причина этого в том, что WGSL выравнивает векторы `vec3f` по границам `vec4f`. Это может привести к тому, что шейдеры будут обращаться к несовпадающим данным, и простой (хотя и неэффективный) способ решения этой проблемы — использовать тип `vec4f` вместо `vec3f`.

После определения структуры `InputData` этот код создаёт переменную с именем `input`. Благодаря атрибуту `@group(0)` `@binding(0)` эта переменная обращается к uniform данным как к структуре `InputData`.

Первое поле переменной `OutputData` содержит местоположение вершины, которое получается путём умножения входных координат на матрицу преобразования.

Затем шейдер получает три важных вектора. Он вычисляет вектор нормали, вычитая центр сферы из нового местоположения вершины. Он вычисляет вектор от вершины до наблюдателя, вычитая положение вершины из положения наблюдателя. Вектор освещения получается вычитанием положения вершины из положения источника света.

В последней части вершинного шейдера задаются значения освещённости, которые будут использоваться фрагментным шейдером: фоновое освещение, диффузное освещение, зеркальное освещение и блеск.

Фрагментный шейдер

Для определения цвета фрагмента шейдер получает несколько переменных из вершинного шейдера, включая векторы и параметры освещения. Затем он использует модель освещения Блинна–Фонга для определения того, как фрагмент должен быть освещён. Этот метод включает пять шагов:

1. Вычисление значения $n \cdot l$, где n — вектор нормали, а l — направление света.
2. Вычисление вектора полупути h , который расположенный посередине между l и v .
3. Вычисление значения $(n \cdot h)^{m'}$, где m' — блеск поверхности.
4. Используя результаты шагов 1–3, складывает компоненты рассеянного, диффузного и зеркального света.
5. Усреднение вычисленных компонентов света и исходного цвета.

В большинстве вычислений возможные результаты могут превышать 1.0, что является максимальным значением компонента цвета с плавающей точкой. По этой причине для удержания значения в диапазоне от 0.0 до 1.0 часто используется функция `clamp`.

Каждый из шагов пронумерован в комментариях к коду. На втором шаге шейдер складывает векторы, представляющие направление к источнику света (`fragData.lightVec`) и направление к наблюдателю (`fragData.viewerVec`). Затем он вызывает функцию `normalize`, чтобы гарантировать, что длина вектора будет равна 1.0. Если вектор предназначен только для определения направления, то его длина должна быть нормализована до 1.0.

После добавления компонентов окружающего, диффузного и зеркального освещения фрагментный шейдер смешивает полученную сумму с цветом, полученным от вершинного шейдера. Таким образом, шейдер освещает или затемняет цвет исходной поверхности.

Диффузное отражение

Уточним формулы расчета освещения в соответствии с лекцией 11.

Определение материала

В табл. 1 приведены значения коэффициентов отражения и показателя зеркального отражения (степени шероховатости, блеска) для некоторых материалов.

Табл. 1. Значения коэффициентов отражения и степени шероховатости для некоторых материалов

№ вар.	Материал	k_{aR}	k_{aG}	k_{aB}	k_{dR}	k_{dG}	k_{dB}	k_{sR}	k_{sG}	k_{sB}	m
1	Латунь	0.3294	0.2235	0.0275	0.7804	0.5687	0.1137	0.9922	0.9412	0.8078	28
2	Бронза	0.2125	0.1275	0.0540	0.7140	0.4284	0.1814	0.3935	0.2719	0.1667	26
3	Хром	0.25	0.25	0.25	0.4	0.4	0.4	0.7746	0.7746	0.7746	77
4	Медь	0.1913	0.0735	0.0225	0.7038	0.2705	0.0828	0.2568	0.1376	0.0860	13
5	Золото	0.2473	0.1995	0.0745	0.7516	0.6065	0.2265	0.6283	0.5558	0.3661	51
6	Олово	0.1059	0.0588	0.1137	0.4275	0.4706	0.5412	0.3333	0.3333	0.5216	10
7	Серебро	0.1923	0.1923	0.1923	0.5075	0.5075	0.5075	0.5083	0.5083	0.5083	51
8	Полированное серебро	0.2313	0.2313	0.2313	0.2775	0.2775	0.2775	0.7739	0.7739	0.7739	90

8.2 Задание для самостоятельной работы (по вариантам)

Выберите из таблицы материал поверхности, соответствующий вашему варианту, и передайте его отражательные характеристики в вершинный шейдер.

8.3 Задание для самостоятельной работы

В вершинном шейдере для каждой вершины рассчитайте значение ее цвета с помощью формул диффузного отражения (7) из лекции 11.

Подсказка: для реализации формулы (7) из лекции 11 можно воспользоваться функцией \max .

Подсказка: если параметры источника света задаются в системе координат наблюдения, то матрица преобразования нормали считается как транспонированная обратная матрица, полученная на основе подматрицы 3×3 от произведения матриц вида и модели (см. лекцию 8, слайд 64).

Диффузное и фоновое отражение

8.4 Задание для самостоятельной работы

Определите рассеянный свет интенсивностью $I_a = [0.2, 0.2, 0.2]$. Добавьте его учет в модель диффузного отражения в соответствие с формулой (10) из лекции 11.

Зеркальное отражение

8.5 Задание для самостоятельной работы

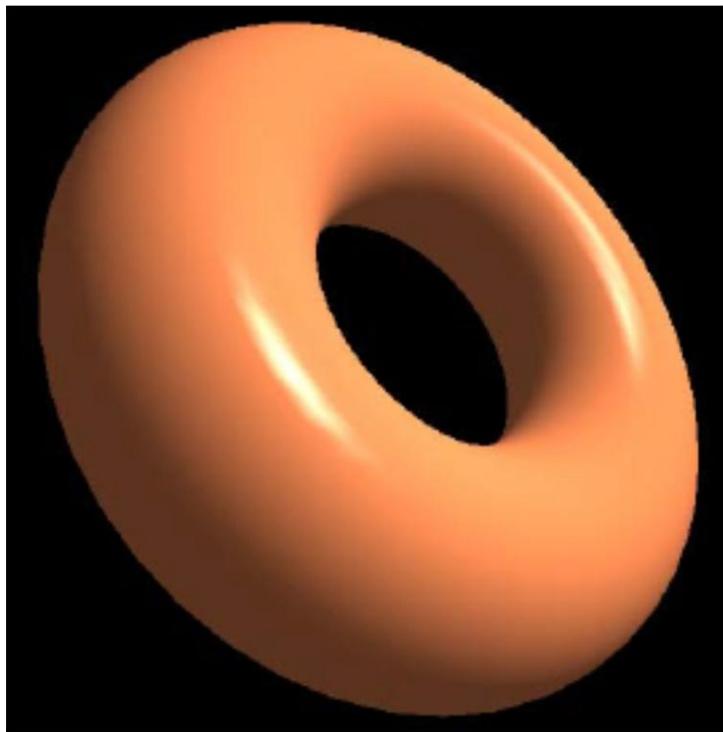


Рис. 3. Изображение тора, дающего зеркальное, диффузное и фоновое отражение

В формулу расчета освещения в вершине добавьте зеркальное отражение (рис. 3). Для нахождения координат вектора r в языке шейдеров уже существует функция `reflect(-l, n)` в первый аргумент которой следует передать значение вектора падающего света (т.е. вектор: $-l$), а во второй – вектор нормали n .

Определение параметров источника света

Геометрические параметры, определяющие источник света, т.е. его координаты или направление лучей, могут задаваться как в мировой (внешней) системе координат, так и в системе координат, связанной с камерой (система координат наблюдения).

8.6 Задание для самостоятельной работы

Ответьте на вопрос – какой тип имеет заданный источник света и в какой системе координат он задан?

Определение точечного источника света

8.7 Задание для самостоятельной работы

Задайте в программе параметры, описывающие точечный источник белого света в мировой системе координат. В вершинном шейдере вычислите координаты вершин и нормалей в них в той же самой системе координат, в которой задается положение источника света. Дополните формулы расчета отражений с использованием новых данных.

Определение бесконечно удаленного источника света

8.8 Задание для самостоятельной работы (по вариантам)

Нужно выполнить только **один** вариант, соответствующий вашему номеру в журнале.

Вариант 1. Вместо точечного источника света, задайте в программе параметры, описывающие бесконечно удаленный источник **белого** света в мировой системе координат. Дополните формулы расчета отражений с использованием новых данных.

Вариант 2. Вместо точечного источника света, задайте в программе параметры, описывающие бесконечно удаленный источник **белого** света в системе координат *наблюдения*. Дополните формулы расчета отражений с использованием новых данных.

Радиальное затухание интенсивности

8.9 Задание для самостоятельной работы (необязательное)

Задайте в программе параметры, определяющие радиальное затухание интенсивности точечного источника света.

Нарисуйте еще одну такую же фигуру, переместив ее с помощью матрицы модели подальше от источника света по отношению к первой фигуре. Покажите различие в освещении этих фигур с использованием радиального затухания интенсивности и без него.

После демонстрации, радиальное затухание интенсивности и рисование второй фигуры можно отключить.

Имитация узконаправленного источника света

8.10 Задание для самостоятельной работы (необязательное)

Задайте в программе параметры, определяющие угловое затухание интенсивности точечного источника света и передайте их в вершинный шейдер.

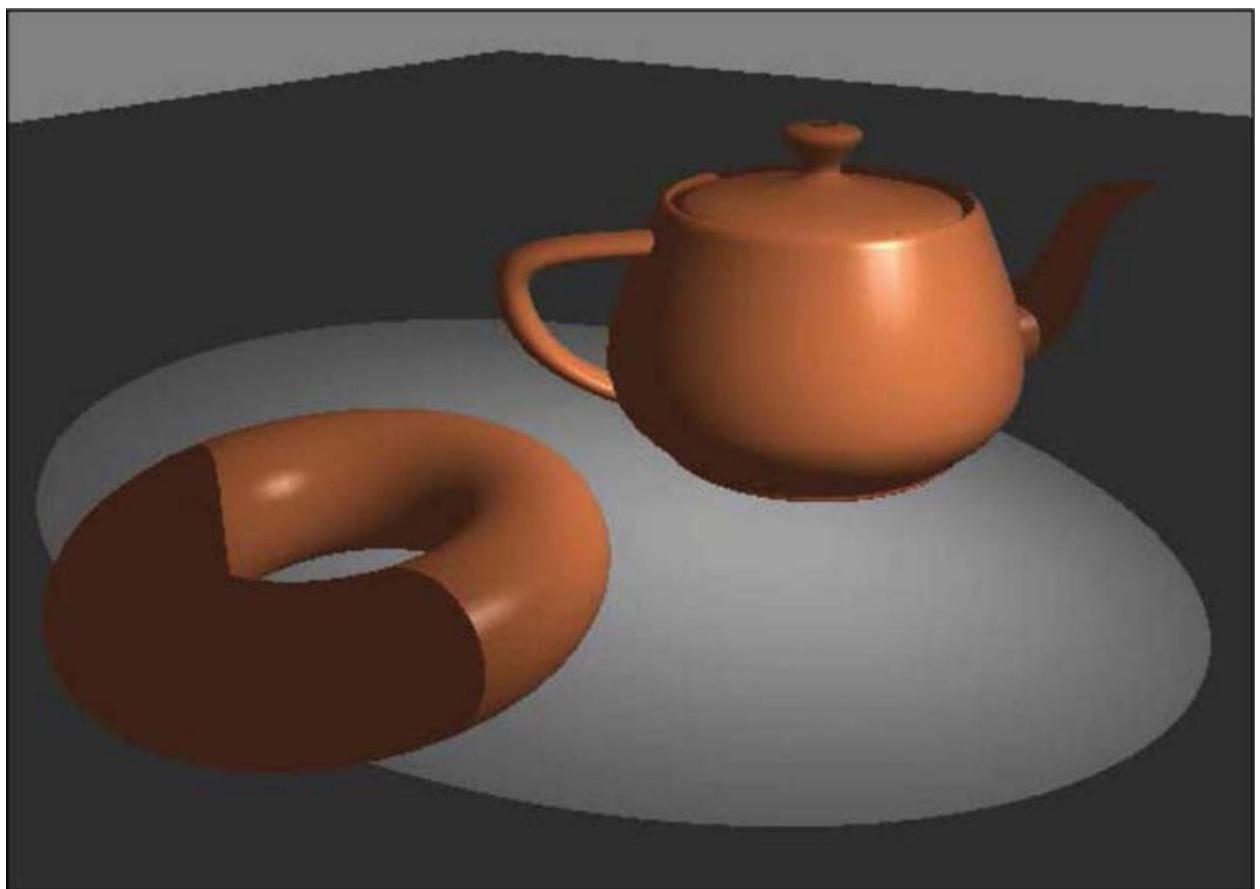


Рис. 4. Сцена, освещаемая единственным источником узконаправленного света

Реализуйте в вершинном шейдере угловое затухание интенсивности точечного источника света (имитация прожекторного эффекта как на рис. 4).

После демонстрации, угловое затухание интенсивности точечного источника света можно отключить.

8.11 Задание для самостоятельной работы

Оформите код вычисления фонового, диффузного и зеркального отражений в виде отдельной функции `fn phongModel(position: vec3, normal: vec3)`, принимающей на вход координаты точки и нормаль в ней. Остальные параметры можно брать из глобальной области видимости. Функция должна возвращать найденный цвет поверхности.

Реализация двухстороннего отображения

8.12 Задание для самостоятельной работы (необязательное)

Создайте незамкнутую поверхность. Для этого отрисуйте только часть поверхности сферы, например, половину по одной из угловых координат. Как отображается внутренняя часть поверхности?

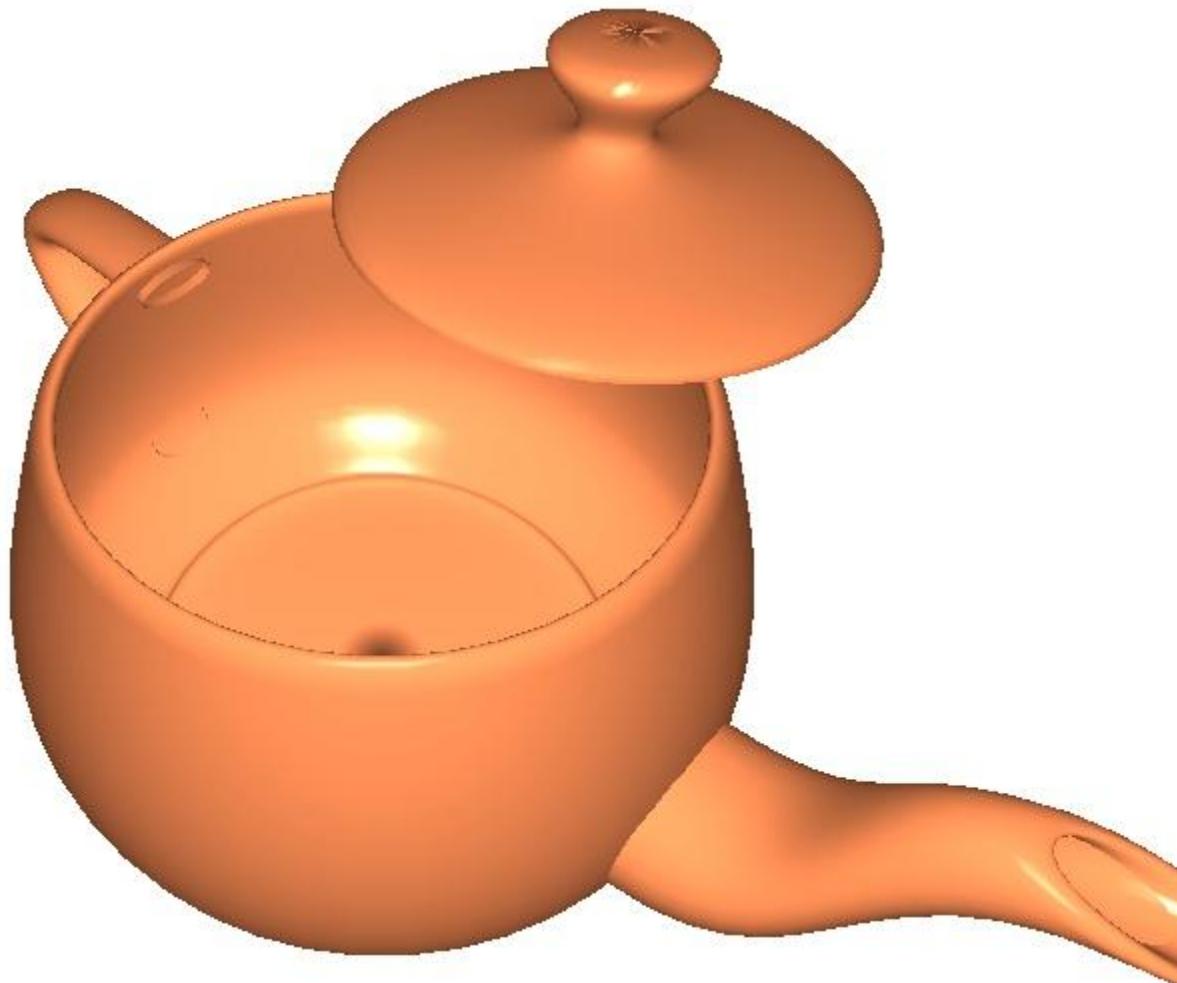


Рис. 5. Двухстороннее освещение поверхности чайника из Юты

С помощью метода, приведенного на слайдах 31-32 лекции 11, реализуйте правильную закраску внутренней части поверхности.

Имитация тумана

8.13 Задание для самостоятельной работы (необязательное)

1. Выберите линейную функцию атмосферного поглощения. Задайте необходимые параметры и передайте их в шейдер, который рассчитывает освещение. С помощью умножения функции атмосферного поглощения на рассчитанное значение цвета фрагмента, осуществите имитацию тумана (рис. 6).

2. Выберите экспоненциальную линейную функцию атмосферного поглощения. Задайте необходимые параметры и передайте их в шейдер, который рассчитывает освещение. С помощью умножения функции атмосферного поглощения на рассчитанное значение цвета фрагмента, осуществите имитацию тумана (рис. 6).

3. Выберите экспоненциальную квадратичную функцию атмосферного поглощения. Задайте необходимые параметры и передайте их в шейдер, который рассчитывает освещение. С помощью умножения функции атмосферного поглощения на рассчитанное значение цвета фрагмента, осуществите имитацию тумана (рис. 6).

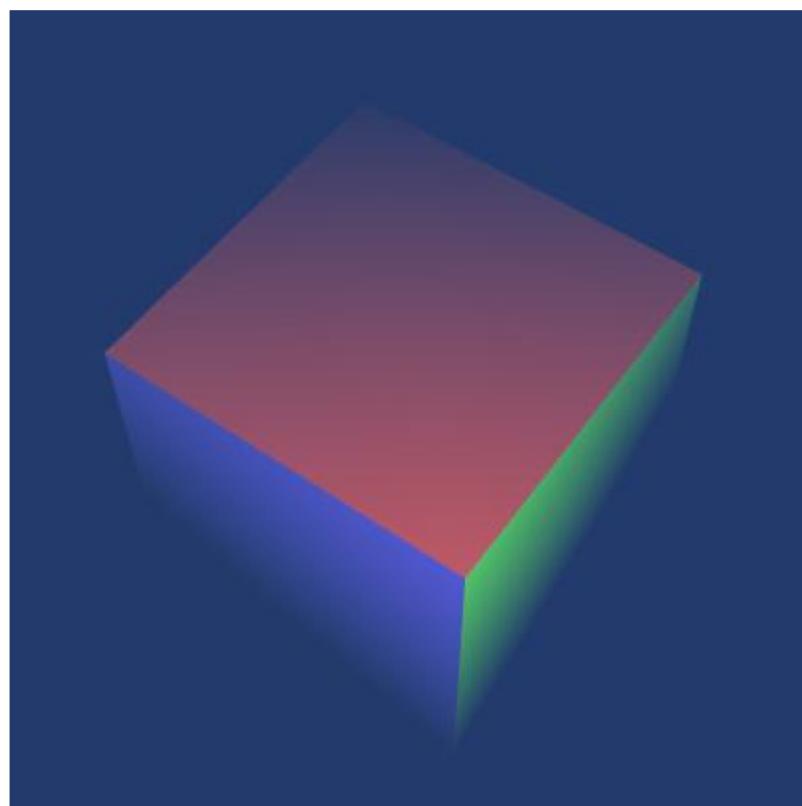


Рис. 6. Эффект тумана

8.14 Задание для самостоятельной работы (необязательное)

Задайте цвет атмосферы, передайте это значение в шейдер, который рассчитывает освещение.

По формуле (17) лекции 11 осуществите имитацию не полностью прозрачной (цветной) атмосферы.

Дополнительная литература



Вольф Д. OpenGL 4. Язык шейдеров. Книга рецептов. М.: ДМК Пресс, 2015. 368 с.

<https://e.lanbook.com/reader/book/73071/#1>

(ссылка доступна из внутренней сети МГТУ им. Н.Э. Баумана)