

УДК 004.946+004.925.4

## **МЕТОДЫ РАСПРЕДЕЛЕННОЙ ВИЗУАЛИЗАЦИИ ВИРТУАЛЬНЫХ ОБЪЕКТОВ С ИСПОЛЬЗОВАНИЕМ МУЛЬТИТЕКСТУРИРОВАНИЯ НА ОСНОВЕ СМЕШИВАНИЯ МАТЕРИАЛОВ**

**А. В. Мальцев, П. Ю. Тимохин, А. М. Трушин**

*Федеральный научный центр Научно-исследовательский институт системных исследований  
Российской академии наук, avmaltcev@mail.ru, webpismo@yahoo.de*

В работе предлагаются новые методы и алгоритмы мультитекстурирования объектов трехмерной виртуальной среды. Рассматривается задача рендеринга объектов с применением смешанных материалов (blend materials). Такие материалы состоят из нескольких подматериалов, структурированных в виде двоичного дерева и накладываемых на одну и ту же область поверхности объекта. В процессе визуализации используется попиксельное смешивание результатов независимого применения к поверхности объекта каждого из подматериалов, входящих в дерево. Предлагаемые решения основаны на применении распределенных вычислений с помощью современных многоядерных GPU.

*Ключевые слова:* визуализация, мультитекстурирование, мульти-материал, шейдеры.

## **METHODS FOR DISTRIBUTED VISUALIZATION OF VIRTUAL OBJECTS USING MULTITEXTURING BASED ON MATERIAL BLENDING**

**A. V. Maltsev, P. Yu. Timokhin, A. M. Trushin**

*System Research Institute, Russian Academy of Sciences,  
avmaltcev@mail.ru, webpismo@yahoo.de*

The paper proposes new multi-texturing methods and algorithms for objects of 3D virtual environment. The task of object rendering with blend materials is considered. Such materials consist of several sub-materials structured in the form of a binary tree and laid on the same area of the object surface. To blend results obtained by independent application of each material from the tree to the object's surface, the visualization process uses per-pixel mixing. Proposed solutions are based on the use of distributed computing by means of modern multi-core GPUs.

*Keywords:* visualization, multitexturing, multi material, shaders.

**Введение.** В настоящее время во многих сферах деятельности человека широко используется компьютерное моделирование и визуализация трехмерных объектов. Для достижения сходства какого-либо виртуального объекта с его реальным прототипом, кроме придания нужной формы, необходимо также задать визуальные свойства поверхности этого объекта путем наложения на нее нужного рисунка – текстуры. Данный процесс называется текстурированием объекта. Общим случаем текстурирования трехмерных объектов является использование материалов. Данный подход позволяет не только определить рисунок поверхности, но также задать некоторые ее свойства, например, способность отражать и излучать свет, степень прозрачности, шероховатость поверхности. Все эти свойства обычно описываются в материале в виде различных текстур (диффузной текстуры, карты прозрачности, карты нормалей, отражения и т. д.) и параметров отражения и излучения различных компонент освещения. Такой материал далее будем называть стандартным.

В процессе моделирования и визуализации виртуальной среды часто возникает задача мультитекстурирования объектов [1, 2]. Ее суть состоит в необходимости использования на

одной и той же поверхности сразу несколько текстур одного типа. Как правило, эти текстуры являются составной частью каких-либо материалов. Например, при моделировании частичного покрытия снегом некоторого ландшафта в зимних условиях требуется имитировать как минимум две составляющие: саму подстилающую поверхность и лежащие на ней участки снега. Эти элементы имеют различную природу и характеристики, следовательно, должны моделироваться разными материалами. Одним из эффективных способов решения задачи мультитекстурирования является применение смешиваемых материалов или, как их еще называют, blend-материалов [3].

В данной работе предлагаются новые распределенные методы и подходы для визуализации на GPU объектов трехмерной виртуальной среды с использованием смешиваемых материалов в масштабе реального времени. Далее рассмотрим эти решения подробно.

### Мульти-материал blend и общая схема его применения при рендеринге объекта

Blend-материал представляет собой совокупность некоторого числа подматериалов, структурированных в виде двоичного дерева  $T$  и смешиваемых по определенным правилам. В качестве подматериалов могут выступать как стандартные, так и blend-материалы, при этом первые являются листьями упомянутого дерева, а вторые – внутренними узлами и корнем.

Чтобы визуализировать виртуальный объект с наложенным на него blend-материалом, произведем рекурсивный обход дерева  $T$ , начиная с корня. Для каждого внутреннего узла уровня  $n$  вычислим промежуточное (для корневого узла – результирующее) значение интенсивности  $I^n$  отраженного освещения в каждом пикселе поверхности виртуального объекта путем смешивания интенсивностей  $I_1^{n+1}$  и  $I_2^{n+1}$  в том же пикселе, полученных в процессе обработки двух дочерних узлов  $n + 1$  уровня. Алгоритм смешивания устанавливается отдельно для каждого родительского узла на этапе создания дизайнером трехмерной сцены. Для этого можно использовать коэффициент смешивания  $k_b \in [0,1]$  или текстурную маску  $M_b$ .

Коэффициент  $k_b$  задает процентное соотношение вкладов каждой из интенсивностей освещения  $I_1^{n+1}$  и  $I_2^{n+1}$ , вычисленных в дочерних узлах, в значение интенсивности  $I^n$  родительского узла:

$$I^n = (1 - k_b) I_1^{n+1} + k_b I_2^{n+1}. \quad (1)$$

Параметр  $k_b$  распространяется на все пиксели поверхности визуализируемого виртуального объекта. Маска же дает возможность задать подобный коэффициент для каждого пиксела объекта в отдельности. Она представляет собой текстуру в оттенках серого цвета, где более темные области означают преобладание первого подматериала в смеси, а более светлые – второго. Формула расчета интенсивности отраженного освещения в пикселе для родительского узла  $n$ -ого уровня в таком случае аналогична (1):

$$I^n = (1 - M_b(s, t)) I_1^{n+1} + M_b(s, t) I_2^{n+1}, \quad (2)$$

где  $s, t$  – текстурные координаты для рассматриваемого пиксела объекта;

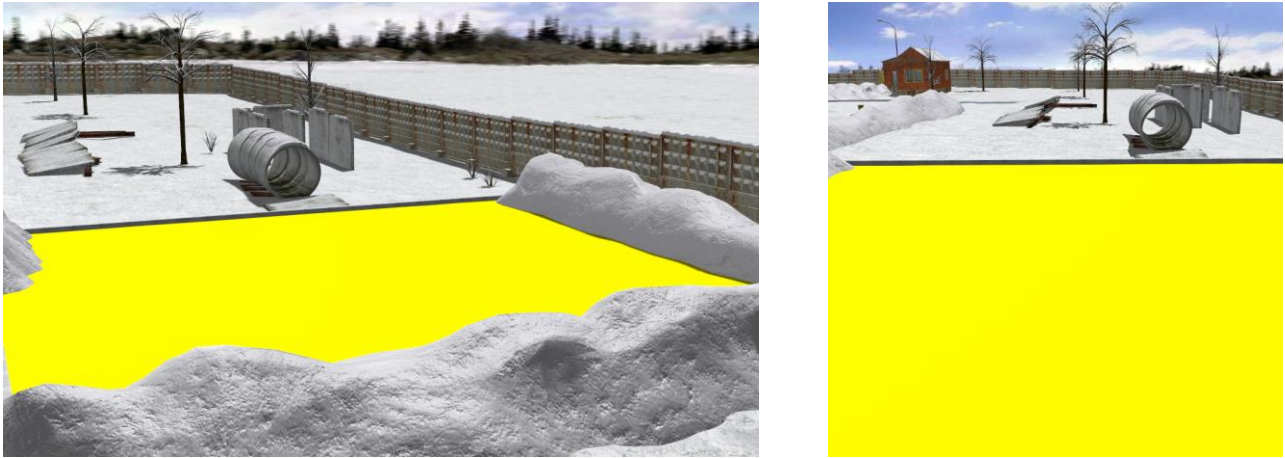
$M_b(s, t) \in [0,1]$  – значение коэффициента смешивания для этого пиксела, взятое из маски  $M_b$  по координатам  $s, t$ .

В случае листового узла, значение интенсивности  $I^n$  в каждом пикселе поверхности объекта вычислим по одной из известных моделей расчета освещенности (например, Фонга [4], Блинна [5] или расширенной модели, описанной в [6]) с учетом параметров материала и входящих в него текстур прозрачности, рельефа, отражения и т. д.

### Визуализация изображения объекта с blend-материалом

Рассмотрим задачу отображения виртуального объекта трехмерной сцены с использованием blend-материала на примере моделирования покрытия поверхности площадки неко-

торого виртуального полигона в зимних условиях. Данный полигон и упомянутая площадка (выделена цветом) представлены в двух ракурсах на рис. 1.



**Рис. 1. Площадка виртуального полигона в зимних условиях**

Используемый для покрытия площадки мульти-материал  $B$  включает два стандартных материала:  $S_1$  «асфальт» и  $S_2$  «снег». Каждый из них содержит свои коэффициенты отражения зеркальной составляющей освещения и степени блика, диффузную текстуру (рис. 2–3), определяющую цвет, а  $S_2$  еще и карту нормалей (рис. 4), которая позволяет имитировать микрорельеф. Смешивание материалов будем производить с помощью текстурной маски  $M_b$ , изображенной на рис. 5. Темные ее части соответствуют преобладанию  $S_1$ , светлые –  $S_2$ . Заметим, что текстуры, содержащиеся в материалах, представляют собой небольшие тайлы, которые при наложении на поверхность моделируемой площадки циклически повторяются, а маска  $M_b$  – покрывает всю площадку целиком. В поставленной задаче двоичное дерево  $T$  состоит из корневого узла, связанного с blend-материалом  $B$ , и двух его дочерних узлов – листьев, представленных стандартными материалами  $S_1$  и  $S_2$ . Чтобы определить по формуле (2) результирующее значение интенсивности  $I^0$  отраженного освещения в каждом пикселе поверхности нашей виртуальной площадки, необходимо вычислить интенсивности  $I_1^1$  и  $I_2^1$  освещения для тех же пикселей в случае наложения на площадку каждого из материалов  $S_1$  и  $S_2$  по отдельности. Для этого из всех объектов, попадающих в пирамиду видимости текущей виртуальной камеры, дважды визуализируем только рассматриваемую площадку: сначала с материалом  $S_1$ , затем с материалом  $S_2$ . Расчет освещенности поверхности виртуального объекта будем производить попиксельно на графическом процессоре, используя шейдеры, реализующие расширенную вычислительную модель из [6]. Требуемые для расчета данные о коэффициентах отражения диффузной и зеркальной компоненты освещения, а также о нормалях к поверхности в пикселях объекта возьмем из диффузной текстуры, текстуры нормалей и параметров материала. Отметим, что при рендеринге площадки с материалом  $S_1$ , в нашем случае, нормаль перпендикулярна плоскости площадки и одинакова для всех ее пикселей. Результат каждой визуализации сохраним в виде текстуры. При этом целесообразно применение так называемого прямого рендеринга в текстуру с использованием технологии FBO (framebuffer objects). FBO – это расширение OpenGL, которое обеспечивает эффективный интерфейс для отрисовки в контексты, отличные от буферов GL, предоставленных оконной системой. Более подробно про использование технологии FBO можно изучить в [7]. Заметим, что ширина  $W$  и высота  $H$  буферов (как и ассоциированных с ними текстур) в пикселях должны быть не меньше соответствующих размеров буфера кадра. При этом предполагается, что все буферы, используемые для отрисовки одного и того же объекта с мульти-материалом, имеют одинаковые размеры.



Рис. 2



Рис. 3

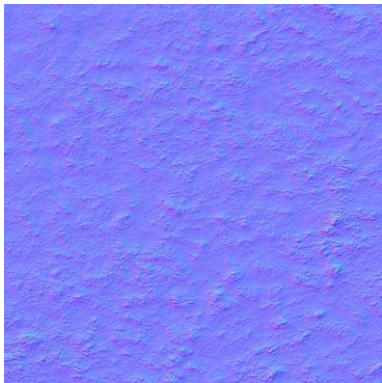


Рис. 4

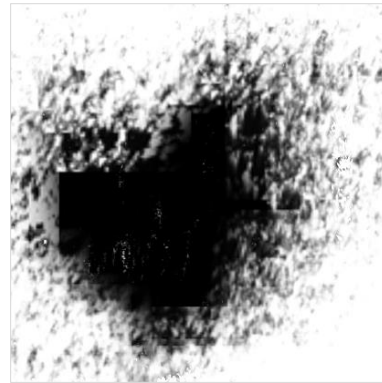


Рис. 5

Полученные в результате текстуры  $M_1$  и  $M_2$  с изображениями площадки покрытой асфальтом (материал  $S_1$ ) и покрытой снегом (материал  $S_2$ ) представлены соответственно на рис. 6 и 7. В них цвет каждого пиксела, принадлежащего объекту, соответствует интенсивности света, отраженного в точке поверхности этого объекта. Таким образом,  $M_1$  и  $M_2$  хранят иско-  
мые нами значения  $I_1^1$  и  $I_2^1$  для каждого пиксела площадки.



Рис. 6

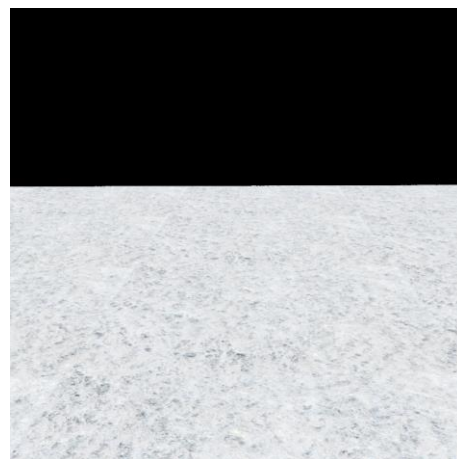


Рис. 7

Далее проведем смешивание материалов на поверхности виртуального объекта. Для этого еще раз выполним рендеринг площадки, но записывать полученный результат будем уже не в текстуру, а в буфер кадра. В ходе обработки каждого пиксела  $P$  объекта (площадки) во фрагментном шейдере необходимо прочитать значения интенсивностей  $I_1^1$  и  $I_2^1$  из соот-



ветствующих  $P$  текселов текстур  $M_1$  и  $M_2$ , а также значение коэффициента смешивания из текстуры маски  $M_b$ . Чтение данных из  $M_b$  осуществим по текстурным координатам  $s_P, t_P$  пиксела  $P$ . Они доступны во фрагментном шейдере в результате выполняемой графическим конвейером интерполяции текстурных координат, заданных для вершин полигональной модели площадки. На рис. 8 визуально продемонстрировано распределение считываемых из  $M_b$  коэффициентов смешивания  $M_b(s_P, t_P) \in [0,1]$  по поверхности моделируемой площадки. Более светлые области соответствуют большему значению коэффициента и, следовательно, преобладанию материала  $S_2$ .

Чтобы получить интенсивности  $I_1^1$  и  $I_2^1$ , определим текстурные координаты  $s_m, t_m$  тексела, соответствующего точке  $P$ , в текстурах  $M_1$  и  $M_2$ . Для этого во фрагментном шейдере считаем экранные координаты  $P_x, P_y$  пиксела  $P$  из стандартной переменной  $gl\_FragCoord$  и поделим их на соответствующие размеры  $W, H$  используемых буферов FBO и ассоциированных с ними текстур  $M_1$  и  $M_2$ :  $s_m = P_x/W, t_m = P_y/H$ . Подставляя найденные данные в (2), получим следующую формулу для вычисления интенсивности света, отраженного в точке  $P$  поверхности моделируемой площадки, с учетом смешивания двух материалов:

$$I^0 = (1 - M_b(s_P, t_P)) M_1(P_x/W, P_y/H) + M_b(s_P, t_P) M_2(P_x/W, P_y/H).$$

Результат визуализации покрытия поверхности виртуальной площадки с использованием изложенного метода рендеринга объектов с мульти-материалом blend, представлен на рис. 9. Описанный подход применим к дереву  $T$  произвольной глубины, которая может быть ограничена лишь объемом видеопамяти, требуемой под буферы FBO, и максимальным временем обработки этого дерева, обеспечивающим выполнение визуализации виртуальной сцены в реальном времени.

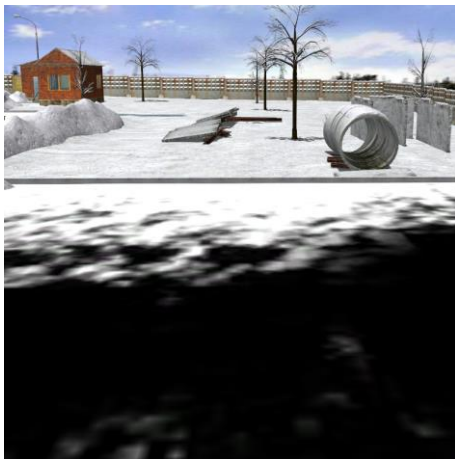


Рис. 8



Рис. 9

**Закключение.** В результате проведенных исследований были созданы новые методы и подходы для визуализации трехмерных виртуальных объектов с использованием мультитекстурирования. Особенностью предлагаемых решений является смешивание на объекте не отдельных текстур, а полноценных материалов, что позволяет более реалистично моделировать визуальные характеристики объектов виртуальной среды. Кроме того, отдельный рендеринг подматериалов из бинарного дерева в буферы FBO снимает ограничение на глубину такого дерева, вызываемое максимальным числом доступных в шейдерах текстурных блоков. Созданные методы и алгоритмы широко используют технологии шейдерной обработки на современных GPU и поддерживают рендеринг сложных сцен в реальном времени. Предложенные решения реализованы в виде программных модулей и прошли успешную апробацию в составе системы визуализации, разработанной в ФГУ ФНЦ НИИСИ РАН.

### **Литература**

1. Benstead L. Beginning OpenGL Game Programming (2nd Ed.). Boston : Course Technology PTR, 2009. 290 p.
2. Федорищев Л. А. Мультитекстурирование с помощью шейдеров // Программные продукты и системы. 2013. № 1. С. 58–61.
3. Blend Material [Электронный ресурс] URL: <http://help.autodesk.com/view/3DSMAX/2017/ENU/?guid=GUID-D2B59023-7D53-4E86-804F-7A037E787055> (дата обращения: 16.10.2017).
4. Phong B. T. Illumination for computer generated pictures // Communications of ACM 18. 1975. № 6. P. 311–317.
5. Blinn J. F. Models of light reflection for computer synthesized pictures // Proceedings of 4th annual conference on computer graphics and interactive techniques. 1977. P. 192–198.
6. Мальцев А. В., Михайлюк М. В. Моделирование теней в виртуальных сценах с направленными источниками освещения // Информационные технологии и вычислительные системы. 2010. № 2. С. 68–74.
7. Framebuffer Object [Электронный ресурс] URL: [http://www.opengl.org/wiki/Framebuffer\\_Object](http://www.opengl.org/wiki/Framebuffer_Object) (дата обращения: 16.10.2017).