

Эффективная реализация HDR визуализации на GPU

П. Тимохин, М. Торгашев

Научно-исследовательский институт системных исследований РАН, Москва, Россия
webpismo@yahoo.de, mtorg@mail.ru

Аннотация

В статье рассматривается задача синтеза изображений виртуальных сцен в расширенном диапазоне в системах визуализации для имитационно-тренажерных комплексов и обучающих систем. Для ее решения предложен новый эффективный метод для реализации HDR технологии, использующий параллельные вычисления на GPU и обеспечивающий визуализацию в режиме реального времени. Технология использует модифицированный глобальный тональный оператор Рейнхарда с оригинальным алгоритмом вычисления глобальных яркостей кадра и обеспечивает моделирование эффекта «заплывания» ярких областей изображения.

Ключевые слова: визуализация, расширенный диапазон, тональный оператор, видеотренажерные системы, параллельные вычисления, GPU.

1. ВВЕДЕНИЕ

Одной из основных задач систем визуализации является расчет освещенности виртуальных объектов от источников света. В системах визуализации реального времени наблюдается тенденция к использованию более сложных моделей освещения, основанных на реальных физических законах, например BRDF. Предпочтительным вариантом является использование реальных фотометрических параметров для источников света и материалов. Освещенность реальных сцен может иметь огромный динамический диапазон. Стандартный подход, при котором расчет освещенности выполняется в диапазоне с глубиной представления цвета 8 бит на каждый цветовой канал (Low Dynamic Range, LDR), не позволяет получить приемлемый результат для подобных сцен. В последние годы на рынке начали появляться HDR (High Dynamic Range) мониторы, которые имеют больший динамический диапазон и поддерживают HDR форматы передачи с большей глубиной представления цвета, однако распространение таких мониторов до настоящего момента очень ограничено. В связи с этим все более актуальной задачей становится преобразование изображения из расширенного диапазона HDR в диапазон LDR. Такое преобразование имеет название тонального оператора, задачей которого является приведение изображения из расширенного диапазона в стандартный таким образом, чтобы добиться наиболее естественного отображения, близкого к тому, которое наблюдается в реальной обстановке. При реализации тонального оператора в системе визуализации для тренажерных и обучающих систем ключевым требованием является соблюдение режима реального времени синтеза изображений (с частотой не менее 25 кадров в секунду). Это ограничивает возможность использования сложных тональных операторов и требует максимально эффективного использования возможностей современных аппаратных средств, в частности параллельных вычислений на GPU. Одним из хорошо зарекомендовавших себя тональных операторов, широко используемым в системах визуализации, является глобальный оператор

Рейнхарда [1]. В исследованиях, оценивающих качество работы различных операторов, этот алгоритм занимает достойное место и в ряде случаев имеет преимущества в субъективном качестве получаемых изображений над другими, в том числе более сложными операторами. Одной из востребованных задач визуализации в расширенном диапазоне также является синтез эффектов, возникающих в средствах наблюдения, таких, как «заплывание» изображения ярких областей, блики и ореолы. Эффект «заплывания» (Bloom) изображения моделирует реальный физический процесс, при котором области изображения с избыточной яркостью засвечивают окружающие участки изображения. Например, для фото и видео камер физическая природа эффекта состоит в том, что заряд перетекает с пересвеченной ячейки светочувствительной матрицы на соседние, в результате чего яркие области «заплывают».

В последнее время возможность HDR визуализации активно интегрируется, как в приложения для построения фотореалистичных изображений (см., например, [3]), так и в графические движки и приложения виртуальной реальности (см., например, [2]). В данной работе предлагается новая реализация HDR технологии в системе визуализации с использованием параллельных вычислений на GPU. Технология реализована на базе библиотеки OpenGL и шейдерного языка GLSL, использует глобальный тональный оператор Рейнхарда с модифицированным алгоритмом вычисления глобальных яркостей кадра и обеспечивает моделирование эффекта «заплывания» ярких областей изображения. В сравнении с существующими подходами предложена модификация тонального оператора, обеспечивающая лучшее воспроизведение изображений с большой площадью темных участков, а также эффективный алгоритм расчета глобальных яркостей кадра на GPU.

2. ТЕХНОЛОГИЯ СИНТЕЗА ИЗОБРАЖЕНИЙ ВИРТУАЛЬНЫХ СЦЕН В HDR РЕЖИМЕ

В данной работе HDR режим визуализации реализуется путем поэтапной попиксельной обработки изображения сцены, синтезированного в расширенном диапазоне (высокодиапазонное изображение). Предлагаемая реализация основана на синтезе и обработке текстур на многоядерном графическом процессоре (GPU) с помощью разработанных шейдерных программ (см. схему). Рассмотрим основные этапы предлагаемой технологии.

Подготовка к визуализации. Для обеспечения расширенного диапазона яркостей в виртуальной сцене у источников освещения устанавливаются интенсивности, соответствующие их реальным прототипам, а перед визуализацией сцены в графическом конвейере отключается автоматическое отсечение компонент цвета по границам диапазона [0,1].

Синтез высокодиапазонного изображения сцены. Синтез высокодиапазонного изображения сцены выполняется путем визуализации виртуальной сцены в текстуру rgb-формата, каждый цветовой канал которой задается 32-битным

вещественным числом. Для этого используются две такие текстуры размера, совпадающего с размером кадра, в одной из которых для каждого тексела хранятся сэмплы. Виртуальная сцена визуализируется в первую текстуру (с сэмплами), из которой изображение копируется во вторую текстуру так, что цвет C_{HDR} каждого тексела второй текстуры вычисляется путем интерполяции соответствующих ему сэмплов первой текстуры.



Схема визуализации сцены в HDR режиме

Вычисление глобальных яркостей изображения. На основе синтезированного высокодиапазонного изображения сцены вычисляются средняя L_{cp} , минимальная L_{min} и максимальная L_{max} яркость изображения. В данной работе в качестве яркости пиксела берется Y -компонента цвета C_{HDR} , вычисленного в эталонном цветовом пространстве $CIE\ XYZ$, которая вычисляется как

$$L_{HDR} = 0.2126 \cdot r + 0.7152 \cdot g + 0.0722 \cdot b, \quad (1)$$

где r, g, b - компоненты цвета C_{HDR} . В данной работе L_{cp} вычисляется как среднее геометрическое взвешенное

$$L_{cp} = \exp\left(\frac{\sum w \cdot \ln(L_{HDR} + \delta)}{\sum w}\right), \quad (2)$$

где $w = 1 - \exp(-L_{HDR})$ - весовой коэффициент; а δ - малая константа для обработки случая $L_{HDR} = 0$. В отличие от существующих реализаций, где используется среднее геометрическое [1, 2], применение предложенных весовых коэффициентов позволяет уменьшить чувствительность средней яркости к областям изображения, яркость пикселей которых близка к нулю. Это позволяет добиться сбалансированного изменения яркости изображения от кадра к кадру и устранить пересвечивание объектов на кадрах с преобладающими темными областями (рис. 1). В разделе 3 описана предложенная реализация, позволяющая вычислить L_{cp} , L_{min} и L_{max} с помощью геометрического шейдера за один проход, учитывая яркость каждого пиксела кадра.

Моделирование эффекта «заплывания». Эффект «заплывания» моделируется путем коррекции цвета для каждого пиксела изображения сцены. Для этого синтезируется текстура эффекта, которая смешивается с исходным изображением сцены. В данной работе текстура эффекта синтезируется на основе изображения в расширенном диапазоне, а смешивание выполняется после его тонального преобразования (в стандартном диапазоне), что позволяет уменьшить вычислительную сложность эффекта, сохраняя при этом визуальную привлекательность результирующего изображения.

Моделирование текстуры эффекта выполняется с помощью двух этапов. На первом этапе выполняется выделение областей исходного кадра с избыточной яркостью, а на втором этапе - размытие изображения с помощью гауссовского фильтра. Поскольку эффект сам по себе является размытием, то его пространственное разрешение может быть уменьшено, что позволяет значительно сократить вычислительную сложность и расход памяти без большого ущерба качеству изображения. В данной работе размытие выполняется в половинном разрешении по горизонтали и вертикали.

Выделение ярких областей выполняется отсечением пикселей с яркостью L_{HDR} (см. (1)), меньшей порогового значения L_T , с одновременным линейным приведением цвета ярких областей в стандартный диапазон на основе максимальной яркости L_{max} (см. раздел 3):

$$C_{Bright} = \begin{cases} C_{HDR} \cdot \frac{L_{HDR}}{L_{max}}, & L_{HDR} > L_T \\ 0, & L_{HDR} \leq L_T \end{cases}$$

Пороговая яркость L_T является регулируемым параметром, значение которого можно выбирать, исходя из светочувствительности матрицы или других соображений.

Наиболее качественное и близкое к реальности размытие обеспечивает гауссовский фильтр. Известно, что этот фильтр обладает свойством сепарабельности, означаящим, что свертка с гауссианом в двумерном пространстве может быть заменена на две последовательных одномерных свертки по осям X и Y :

$$C_{Bloom} = C_{Bright} \otimes G_N(x, y) = (C_{Bright} \otimes G_N(x)) \otimes G_N(y).$$

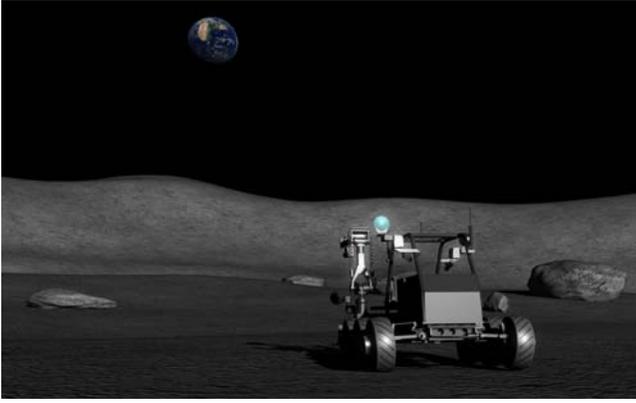


Рис. 1. Сравнение результатов работы модифицированной (слева) и исходной версии (справа) глобального оператора Рейнхарда на изображении со значительной площадью темных участков

Благодаря этому, вычислительная сложность преобразования существенно уменьшается (с $O(N^2)$ до $O(N)$, где N - размер фильтра). Размер фильтра задается пользователем и регулирует степень размытия.

Тональное отображение. На данном этапе высокодиапазонное изображение сцены приводится к стандартному диапазону визуализируемого кадра. С помощью фрагментного шейдера для каждого пиксела кадра вычисляется цвет C_{tone} . Для этого вначале шейдер вычисляет отмасштабированную яркость L_{scaled} вида

$$L_{scaled} = a \cdot \frac{L_{HDR}}{L_{cp}} \quad (3)$$

где L_{HDR} - яркость пиксела, вычисленная согласно (1); a - ключевой параметр (аналог экспозиции у фотокамер). Параметр a вычисляется для текущего кадра с помощью следующей формулы [1]:

$$a = 0.18 \cdot 2^{2(B-A)/(A+B)},$$

где $A = L_{max} - L_{cp}$; $B = L_{cp} - L_{min}$. Полученная в (3) яркость L_{scaled} затем отображается в стандартный диапазон с помощью глобального оператора Рейнхарда [1]:

$$L_{LDR} = \frac{L_{scaled} \cdot \left(1 + \frac{L_{scaled}}{L_{white}^2}\right)}{1 + L_{scaled}},$$

где L_{white} - яркость, которая соответствует белому цвету; в данной работе в качестве L_{white} используется максимальная яркость L_{max} , отмасштабированная аналогично (3). Результирующий цвет C_{tone} пиксела после тонального отображения вычисляется путем обработки цвета C_{HDR} вида

$$C_{tone} = \frac{L_{LDR}}{L_{HDR}} \cdot C_{HDR}.$$

Данное выражение получено на основе комбинации прямого и обратного преобразования из цветового пространства RGB в $CIE\ xYz$ (производное от $CIE\ XYZ$).

Смешивание тонированного изображения и эффекта. На заключительном этапе выполняется смешивание изображения эффекта с изображением после тонального преобразования:

$$C_{LDR} = C_{tone} + k_{EFF} \cdot C_{Bloom},$$

где k_{EFF} - пользовательский параметр, регулирующий «мощность» эффекта.

3. РЕАЛИЗАЦИЯ ВЫЧИСЛЕНИЯ MAX, MIN И СРЕДНЕЙ ЯРКОСТИ НА GPU

В описанной технологии визуализации вычисление L_{cp} , L_{min} , L_{max} обладает высокой вычислительной сложностью, так как, по определению, вычисление каждой такой яркости включает в себя обработку всех пикселов кадра. Для эффективного вычисления таких глобальных яркостей в режиме реального времени предлагается следующая схема их распределенного вычисления на множестве ядер графического процессора. Текстура с изображением виртуальной сцены разбивается логически на блоки текселов одинакового размера. Для каждого такого блока ядро GPU вычисляет минимальную и максимальную яркость блока ($L_{min,block}$ и $L_{max,block}$), а также сумму S взвешенных логарифмов яркостей пикселов блока и сумму W весов пикселов блока; см. выражение (2). Минимальная яркость L_{min} находится как минимальная из всех вычисленных $L_{min,block}$, а L_{max} - как максимальная из всех $L_{max,block}$. Для получения средней яркости L_{cp} вычисляется сумма всех S и сумма всех W , которые подставляются, соответственно, в числитель и знаменатель дроби в выражении (2).

Предлагаемая реализация описанной схемы вычислений основана на возможности графического ускорителя записывать в один тексел результаты параллельной обработки дублирующихся вершин, сгенерированных геометрическим шейдером. Для выполнения такой обработки в видеопамати создается массив вершин, имеющий размер, равный количеству блоков. Для каждой вершины в массив записываются координаты x , y крайнего левого нижнего тексела соответствующего блока. Также в видеопамати создается текстура из двух текселов (такого же формата, что и текстура с изображением сцены), $rgba$ -каналы которой проинициализированы значениями 0 , $-L_s$, 0 , 0 , соответственно, где L_s - некоторая большая величина, заведомо большая максимально возможной яркости освещения в сцене. Предварительно установив в графическом конвейере режим смешивания, при котором в rg,b -каналы

записываются максимальные значения, а в альфа-канал – сумма значений, мы визуализируем массив вершин в текстуру с помощью разработанной шейдерной программы, выполняющей следующий параллельный

Алгоритм вычисления L_{\min} , L_{\max} , L_{cp}

1. Передаем из вершинного шейдера в геометрический шейдер вершину с координатами (x, y) .

2. В геометрическом шейдере выполняем

$$L_{\min, block} = L_S, L_{\max, block} = 0, S = 0, W = 0;$$

Цикл по всем текстелам блока, заданного (x, y) .

Вычисляем L_{HDR} , согласно (1);

$$w = 1 - \exp(-L_{HDR});$$

$$S = S + w \cdot \ln(L_{HDR} + \delta);$$

$$W = W + w;$$

$$L_{\min, block} = \min(L_{\min, block}, L_{HDR});$$

$$L_{\max, block} = \max(L_{\max, block}, L_{HDR});$$

Конец цикла.

Генерируем вершину V_0 с координатами $x = -0.5$ и $y = 0.0$ и цветом C_0 с компонентами $r = L_{\max, block}$, $g = -L_{\max, block}$, $a = S$.

Генерируем вершину V_1 с координатами $x = 0.5$ и $y = 0.0$ и цветом C_1 с компонентой $a = W$.

3. С помощью фрагментного шейдера добавляем каждый цвет C_0 и C_1 в 0-й и 1-й текстел текстуры соответственно.

В результате такой визуализации в r и g каналах 0-го текстела текстуры записываются величины L_{\max} и $-L_{\min}$, а альфа-канале 0-го и 1-го текстела формируется, соответственно, числитель и знаменатель дроби из выражения (2), по которым (предварительно выгрузив текстуру в оперативную память) легко вычисляется величина L_{cp} .

4. ЗАКЛЮЧЕНИЕ

В работе предложена технология реализации режима HDR визуализации виртуальных сцен в реальном времени, адаптированная для выполнения на графическом процессоре.

В технологии реализовано сглаживание синтезируемых изображений, моделирование эффекта «заплывания» пересвеченных областей и тональное отображение с помощью модифицированной версии глобального тонального оператора Рейнхарда. В разработанной модификации для вычисления средней яркости кадра используются взвешенные значения яркости пикселей, что позволяет корректно визуализировать кадры с существенным процентом темных областей (см. рис. 1) и обеспечивает сбалансированное изменение яркости синтезируемых изображений от кадра к кадру. Представлен эффективный алгоритм для быстрого вычисления глобальных яркостей, обеспечивающий обработку всех пикселей кадра с использованием параллельных вычислений на GPU. Использование данного алгоритма позволяет выполнять HDR визуализацию

виртуальных сцен в режиме реального времени на средствах отображения с высоким разрешением (HD, Full HD). Разработанные алгоритмы, методы и подходы были интегрированы в программный комплекс визуализации [4] и прошли успешную апробацию на ряде сцен для имитационно-тренажерных комплексов. На рис. 2 представлен пример кадра визуализации сцены со сложной моделью освещения, включающей в себя сумеречное освещение, атмосферу и освещение фарой, установленной на мобильном роботе.

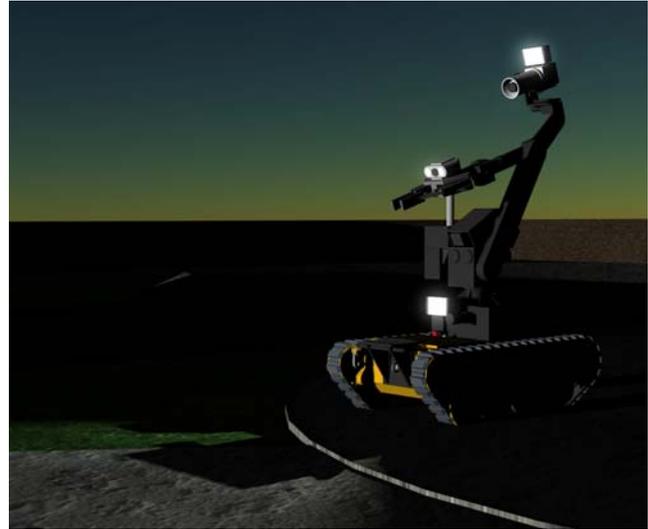


Рис. 2. Пример кадра визуализации сцены со сложным освещением

5. БЛАГОДАРНОСТИ

Работа была выполнена при финансовой поддержке РФФИ в рамках гранта №13-07-00674.

6. ССЫЛКИ

- [1] Kiser C., Reinhard E., Tocci M., Tocci N. Real time automated tone mapping system for HDR video. In IEEE International Conference on Image Processing, 2012. – P. 2749 - 2752.
- [2] Luksh, C.: Realtime HDR rendering. Tech. Rep., Institute of Computer Graphics and Algorithms, TU Vienna (2007).
- [3] Барладян Б.Х., Волобой А.Г., Галактионов В.А., Копылов Э.А.. Эффективный оператор сжатия динамического диапазона яркостей // Программирование, 2004. – № 5. – С. 35 - 42.
- [4] Михайлюк М.В., Торгашев М.А.. Система «GLVIEW» визуализации для моделирующих комплексов и систем виртуальной реальности // Вестник Российской академии естественных наук, 2011. – № 2. – С. 20 - 28.

Сведения об авторах

Тимохин Петр – научный сотрудник, Научно-исследовательский институт системных исследований РАН, webpismo@yahoo.de.

Торгашев Михаил – зав. сектором, Научно-исследовательский институт системных исследований РАН, mtorg@mail.ru.