

Сверхбольшие текстуры для высоко реалистичной визуализации виртуальных ландшафтов¹

П.Ю. Тимохин, М.В. Михайлюк

Аннотация. В статье предлагается новый метод быстрого определения видимых текстурных тайлов, позволяющий выполнять высоко реалистичную визуализацию виртуальных ландшафтов со сверхбольшими текстурами в режиме реального времени. Описываются разработанные параллельные алгоритмы, реализующие предлагаемый метод и использующие современные возможности по организации параллельных вычислений на графическом процессоре. Приведена технология динамической подкачки тайлов сверхбольшой текстуры в реальном времени и алгоритм текстурирования ландшафта.

Ключевые слова: высоко реалистичная визуализация, текстура, виртуальный ландшафт, реальное время, GPU.

Введение

В настоящее время одним из активно развивающихся направлений компьютерной графики является высокореалистичная визуализация виртуальных текстурированных ландшафтов в масштабе реального времени. Такая визуализация особенно востребована в видеотренажерных комплексах космического, авиационного и наземного назначения [1, 2]. В таких комплексах в качестве текстур часто используют данные дистанционной фотосъемки реального ландшафта.

Одним из эффективных методов визуализации текстурированных ландшафтов является мипмэппинг, при котором наряду с исходными текстурами используются текстуры с меньшими уровнями детализации [3]. В этом случае при синтезе изображения для каждого пиксела подбираются текстуры тех уровней, детализация которых наиболее близка к разрешению экрана. Реализация такой технологии требует использования мип-пирамид, размеры которых для сверхбольших текстур превышают аппаратно допус-

тимые значения современных графических ускорителей, что не позволяет загружать их в видеопамять и аппаратно обрабатывать.

Распространенным подходом к решению данной проблемы является разбиение уровней мип-пирамиды с помощью регулярной сетки на небольшие участки одинакового размера (тайлы) и подкачка в видеопамять только тех тайлов, которые нужны для визуализации текущего кадра. В связи с этим возникает задача разработки эффективных алгоритмов отбора, подкачки и визуализации видимых тайлов. В данной работе предлагается новый метод решения этой задачи, позволяющий визуализировать в реальном времени текстуры сверхбольших размеров (порядка $10^6 \times 10^6$ текселов). Предлагаемые алгоритмы используют быстрые параллельные вычисления на современных многоядерных графических процессорах (GPU) и адаптированы для эффективного применения геометрических шейдеров и технологии возврата примитивов (transform feedback).

1. Предшествующие работы

Можно выделить два ключевых подхода к решению поставленной задачи.

При первом подходе отбор, подкачка и визуализация тайлов выполняются исходя из проверки видимости участков геометрии ландшафта, которые они покрывают. Достаточно распространенными являются методы, использующие проверку по полигонам [4], по ограничивающим боксам [5], проверку иерархического дерева геометрии ландшафта [6]. При относительной простоте наложения тайлов (для ландшафтов с регулярной геометрией) в данном подходе в геометрическую модель ландшафта нужно вносить дополнительные вершины (для ландшафтов с динамической оптимизацией сетки вершин или нерегулярной структурой) и текстурные координаты, обеспечивающие привязку тайлов к геометрии ландшафта, что неэффективно расходует видеопамять и увеличивает время визуализации такой модели, ограничивая применение данного подхода в режиме реального времени [7].

При втором подходе работа с мип-пирамидой ведется так, как будто она полностью присутствует в видеопамети, при этом физически в видеопаметь подкачивается лишь небольшая ее часть (по аналогии с виртуальной памятью) [8]. В этом случае для каждой вершины модели ландшафта задаются только глобальные текстурные координаты в полной текстуре, которые при визуализации пересчитываются в локальные текстурные координаты тайлов, загруженных в видеопаметь. В отличие от первого подхода, где тайлы являются самодостаточными текстурами, здесь они выступают в роли страниц, которыми мип-пирамида подкачивается в единый буфер в видеопамети, что позволяет сделать текстурирование ландшафта независимым от моделирования его геометрии. Для обеспечения высокой реалистичности визуализации ландшафтов, имеющих произвольную геометрию, с помощью данного подхода необходимо точное определение видимых на экране тайлов-страниц. Такую точность позволяет обеспечить попиксельный анализ экрана [9, 10], однако при этом возникает задача быстрого и эффективного выполнения большого количества вычислений в режиме ре-

ального времени. Многие существующие методы используют подход [11], при котором такие вычисления распределяются между графическим (GPU) и центральным процессором (CPU), однако временные задержки, связанные с пересылкой промежуточных результатов от GPU к CPU и их CPU-обработкой, ограничивают использование данных методов для визуализации в реальном времени.

Подход, предлагаемый в данной работе, позволяет быстро выделить список видимых тайлов-страниц полностью на стороне GPU, используя его мощный потенциал параллельных вычислений и сводя к минимуму объем данных, передаваемых в оперативную память (CPU отводится лишь функция управления подкачкой полученного списка тайлов), что в результате позволяет обойти описанные выше ограничения.

2. Предлагаемый подход

Пусть задана исходная текстура T_0 ландшафта размером $w = 2^m$ на $h = 2^n$ текселов. Будем называть ее текстурой нулевого уровня. По этой текстуре создаются текстуры 1-го, 2-го, ..., L -го уровней детализации. Текстура T_s s -го уровня имеет размеры $2^{m-s} \times 2^{n-s}$ текселов и получается из текстуры T_{s-1} , например, вычислением среднего значения цветов четырех соседних текселов. Разобьем каждую текстуру s -го уровня на $2^{m-s-k} \times 2^{n-s-k}$ тайлов размером $2^k \times 2^k$ текселов (Рис.1) и обозначим массив этих тайлов через M_s . Ввиду того, что видеопаметь имеет ограниченные размеры, выделим в ней кусок, в котором постоянно будут находиться L' текстур $T_{L-L'+1}, \dots, T_L$ (группа G_1 текстур), и кусок памяти, в который будут динамически (по мере необходимости) подгружаться тайлы текстур остальных уровней (группа G_2 тайлов текстур). Нашей задачей является определение множества тайлов каждого уровня, которые необходимо подкачать для визуализации текущего кадра. Заметим, что оставшаяся часть видеопамети будет использоваться для текущих работ графического процессора.

Решение поставленной задачи предлагается выполнить в 3 этапа. На первом этапе для каж-