Компьютерные инструменты в образовании, 2019
№ 2: 31–42
УДК: 004.925
http://cte.eltech.ru
doi:10.32603/2071-2340-2019-2-31-42

алгоритмическая математика и математическое моделирование

МЕТОДЫ ВИЗУАЛИЗАЦИИ ПРОТЯЖЕННЫХ ЛАНДШАФТОВ В ТРЕНАЖЕРНО-ОБУЧАЮЩИХ СИСТЕМАХ*

Гиацинтов А. М.[1], кандидат технических наук, старший научный сотрудник, algts@inbox.ru
Решетников В. Н.[1], доктор физико-математических наук, профессор, главный научный сотрудник, rvn@niisi.ras.ru
Родителев А. В.[1], ведущий программист, avrod_94@mail.ru

[1]Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук, Нахимовский просп., 36, корп. 1, 117218, Москва, Россия

Аннотация

Одной из задач, возникающих при разработке тренажерных систем, является создание подсистемы отображения земной поверхности. Подсистемы такого класса обеспечивают моделирование и визуализацию ландшафта, подстилающей поверхности (рек, дорог, лесных массивов и т. д.), геометрических объектов, например, имитирующих аэропорты или города и др. Существует два основных подхода к визуализации протяженных ландшафтов, используемых при прорисовке поверхности Земли — кластерная триангуляция и геометрические текстурные выборки (clipmap). Представленный в работе метод генерации и визуализации поверхности Земли основан на втором подходе. Обеспечена возможность производить вычисления на GPU с использованием одинарной точности, что позволяет ускорить вычисления, по сравнению с использованием типов данных с двойной точностью. Кроме того, это позволяет использовать данный подход на мобильных графических процессорах, не поддерживающих двойную точность. Предлагаемый новый метод динамического управления ресурсами обеспечивает уменьшение занимаемой видеопамяти, что позволяет загружать более детализированные текстурные данные и для большего числа объектов одновременно.

Ключевые слова: *WGS84, земная поверхность, визуализация, имитационные системы, система координат.*

Цитирование: Гиацинтов А. М., Решетников В. Н., Родителев А. В. Методы визуализации протяженных ландшафтов в тренажерно-обучающих системах // Компьютерные инструменты в образовании. 2019. № 2. С. 31–42. doi:10.32603/2071-2340-2019-2-31-42

1. ВВЕДЕНИЕ

Одной из задач, возникающих при разработке тренажерных систем, является создание подсистемы отображения земной поверхности. Подсистемы такого класса используются для визуализации реалистичной поверхности Земли, включающей в себя моде-

*Работа выполнена при поддержке РФФИ, грант № 17-07-00169 А.

лирование ландшафта, подстилающей поверхности (рек, дорог, лесных массивов и т. д.), геометрических объектов, например, имитирующих аэропорты или города. Подобные задачи также возникают при разработке геоинформационных систем и развлекательных проектов.

Под тренажерно-обучающей системой (ТОС) оператора сложной технической системы (СТС) будем понимать техническое средство для подготовки операторов в едином информационном окружении, отвечающее требованиям методик подготовки и создающее условия для получения знаний, навыков и умений, реализующее модель таких систем и обеспечивающее контроль над действиями обучаемого, а также для исследований [1].

Значительное развитие алгоритмов визуализации протяженных открытых пространств и ландшафтов, необходимых при визуализации Земли, произошло во второй половине 90х годов 20 века, с разработкой системы Satellite Tool Kit (STK) [3d virtual globes]. Одними из первых были разработаны алгоритмы, которые осуществляли разбиение ландшафта на блоки с различным уровнем детализации [2]. Блоки загружались из системы хранения данных (СХД) в оперативную память компьютера при перемещении наблюдателя и прорисовывались только, когда находились в поле зрения наблюдателя [3]. Детализация блоков определялась на основании дальности до наблюдателя. Последующие алгоритмы повысили качество визуализации за счет использования квадродеревьев и двоичных деревьев триангуляции (BSP-дерево). Они предложили концепцию так называемой очереди с приоритетом для объединения/разделения треугольников и методы оценки появления ошибок прорисовки на основании позиции наблюдателя, что привело к созданию непрерывных ландшафтов с различным уровнем детализации.

Несмотря на преимущества, значительным недостатком разработанных алгоритмов стала значительно возросшая нагрузка на центральный процессор, а также необходимость постоянной передачи новых данных на видеокарту. В дальнейшем эти алгоритмы были вытеснены алгоритмами кластерной триангуляции [4]. Основной идеей данных алгоритмов является разделение ландшафта на сектора (в другой терминологии — куски) с различной детализацией, а различия в детализации между секторами нивелируются использованием вертикальной отсекающей границы (skirt). До сих пор подобные алгоритмы часто применяются при визуализации протяженных ландшафтов, особенно если данные ландшафтов в реальном времени передаются с удаленных серверов [5].

Дальнейшее развитие графических процессоров позволило задействовать аппаратную тесселляцию для задания уровня детализации ландшафта [6]. Конечный облик ландшафта полностью определяется шейдерами тесселляции на основании позиции наблюдателя, позволяя создавать непрерывный ландшафт без разрывов. Уровень тесселляции определяется отдельно для каждой группы вершин (насчитывающей от 3х до 32 вершин [7]), и может быть изменен в зависимости от желаемого соотношения производительности и качества визуализации.

Другая группа алгоритмов предлагает использовать выборку текстурных данных (clipmap) для визуализации ландшафта [8]. Термин clipmap изначально был придуман для обозначения динамического представления неопределенно большого объема текстурных данных в конечном объеме физической памяти. Так как при визуализации земной поверхности обычно применяется значительное количество слоев подстилающей поверхности, то clipmap представляет собой ограниченную выборку данных изображения из конкретного слоя подстилающей поверхности. Выборки делятся на два типа — пирамида и стек. В пирамидальном представлении clipmap содержит в себе все данные подстилающей поверхности для конкретного слоя, в то время как в стеке

clipmap содержит только часть высоко детализированного изображения, обновляемого при перемещении наблюдателя (рис. 1).

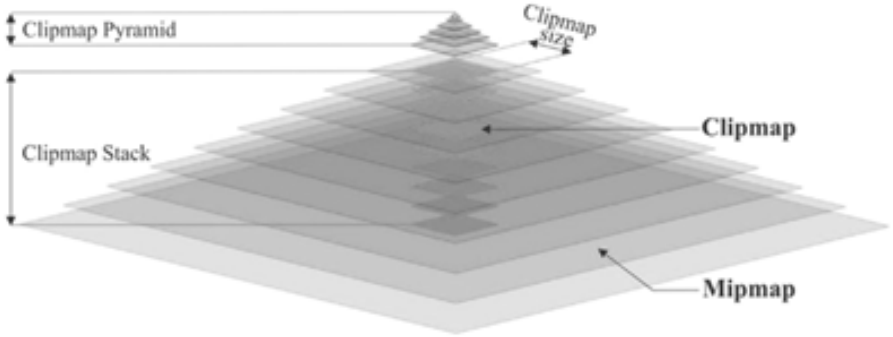

Рис. 1

Первоначально clipmaps были представлены в 1998 году. Методика требовала наличия специализированного оборудования и была предназначена для повышения скорости и качества визуализации в пакете моделирования IRIS Performer. Представленный метод не был предназначен для использования при прорисовке ландшафтов — только в 2004 году метод был адаптирован Френком Лосассо и Хью Хоппом [9]. В отличие от первоначального метода, новый не требовал использования специализированного оборудования и позволял работать с геометрией на аппаратуре потребительского класса, поддерживающей Shader Model 2. Помимо расширения функциональности clipmap для поддержки работы с геометрией, новый метод позволил хранить больше данных в системной памяти и снизить количество медленных загрузок с СХД. Видеокарты с поддержкой Shader Model 2 не позволяли работать с текстурами в вершинных шейдерах, что требовало использования вершинных буферов для передачи данных вместо текстур и обновления всех индексных буферов при перемещении наблюдателя.

В 2005 году был предложен улучшенный метод [10, 14], позволивший практически полностью перенести обработку геометрических clipmap на видеокарты, использующие Shader Model 3. Центральный процессор только осуществлял разжатие изображений, загрузку новых данных и задание вызовов прорисовки. Вершинные буферы, используемые предыдущим методом для хранения данных о высотах, были заменены на двумерные текстуры, а небольшие вершинные и индексные буферы стали использоваться для задания двумерной сетки, к которой применялись данные высот для генерации итогового ландшафта. Метод оказался эффективным, но был не предназначен для визуализации протяженных ландшафтов. Основными проблемами при визуализации земной поверхности с использованием данного метода оказалась недостаточная точность при обработке детализированных сеток и сужение треугольников в окрестностях полюсов, что приводило к излишней детализации сетки без повышения качества визуализации.

С появлением видеокарт, поддерживающих Shader Model 4, в конце 2006 года стало возможным использовать clipmap при помощи текстурных массивов [11, 15]. Текстурный массив — это набор одномерных или двумерных текстур идентичного размера и формата, к которым можно получить доступ в шейдерах. По спецификации OpenGL количество слоев в текстурном массиве должно быть не менее 64-х, однако даже бюджет-

ные карты предлагали до 512 слоев, позволяя дополнительно разгрузить работу центрального процессора за счет передачи нагрузки по обработке текстурных данных на видеокарту.

В настоящее время разработка новых методов визуализации земной поверхности продолжается, хотя в текущем десятилетии она несколько замедлилась.

В связи с наличием ряда нерешенных проблем при визуализации земной поверхности актуальной задачей является разработка новых методов, позволяющих отображать поверхность с большей точностью и меньшими временными затратами.

В статье предлагается метод генерации и визуализации земной поверхности, основными преимуществами которого являются:

— сокращение количества наборов подготавливаемых данных;
— возможность использования одинарной точности вещественных типов данных на графическом процессоре для ускорения вычислений;
— динамическое управление состоянием загруженных ресурсов, обеспечивающее подгрузку и выгрузку данных при перемещении наблюдателя. За счет использования кэширования снижается вероятность скачкообразного изменения производительности системы визуализации.

2. МЕТОД ГЕНЕРАЦИИ И ВИЗУАЛИЗАЦИИ ЗЕМНОЙ ПОВЕРХНОСТИ

Рассматриваемый метод генерации и визуализации земной поверхности предполагает разделение Рассматриваемый метод генерации и визуализации земной поверхности предполагает разделение поверхности планеты на три сегмента — экваториальный и два полярных [12]. По сравнению с другими подходами, в которых земная поверхность делится на шесть частей, данный подход позволяет сократить количество используемых наборов данных и линий соприкосновения сегментов. Экваториальная часть расположена в широтах от -45° до $+45^\circ$, и использует цилиндрическую проекцию. Полярные сегменты используют цилиндрические проекции, повернутые на 90° вокруг оси X глобальной системы координат, и ограничены широтами $\pm 45^\circ$. Основными преимуществами разделения поверхности Земли на три сегмента являются:

— количество наборов подготавливаемых данных сокращено до трех;
— максимум два сегмента земной поверхности видны из любой точки до достижения предельной высоты, после которой Землю можно представить в виде низко детализированной сферы;
— значительное количество исходных данных может быть использовано без применения дополнительных проекций (около 70 процентов);
— простота применения проекций для полярных сегментов;
— текстурные искажения, вносимые при использовании данного метода, незначительны.

Количество сегментов играет значительную роль, если учитывать количество используемой памяти, как оперативной, так и видеопамяти. Для каждого сегмента требуется свой набор данных. Набор данных состоит как минимум из двух выборок текстурных данных (clipmap) — данных высот и подстилающей поверхности. Каждая выборка текстурных данных представлена в виде текстурного массива.

Одной из причин, по которой было введено разделение земной поверхности на три части, является необходимость уменьшения искажений, получаемых при проекции сферической поверхности на плоскость и обратно. Коэффициент масштабирования и угловые искажения значительно увеличиваются при приближении к полюсам планеты.

Начиная от экватора, где искажения отсутствуют, коэффициент масштабирования стремится к бесконечности, в то время как максимальные угловые искажения (разница в значении угла пересечения двух линий до и после проекции) достигают 180°. Разделяя поверхность Земли на три сегмента, удается достичь значений коэффициента масштабирования равным 1,42 и угловых искажений равных 19.76°. Кроме того, объем хранимых данных сокращен приблизительно на 30°, по сравнению с использованием одного сегмента для всей поверхности Земли и применения цилиндрической проекции. По сравнению с другими подходами визуализации земной поверхности, предлагаемый подход вносит незначительные искажения при отображении подстилающей поверхности, которые также могут быть частично нивелированы использованием анизотропной фильтрации.

В предлагаемом подходе применяется несколько систем координат (СК) при обработке данных и последующей прорисовке — глобальная система координат, система координат сегмента, система координат блока. Каждая система координат использует понятие широты и долготы. Глобальная система координат используется для установки начальных позиций других систем координат. Каждый из трех сегментов имеет собственную систему координат — система координат экваториального сегмента совпадает с глобальной, в то время как системы координат полярных сегментов повернуты на ±90°.

Преобразование между системами координат сегментов производится следующим образом:

$$\Theta_{es} = arcsin(cos\Theta_{ps} \cdot cos\phi_{ps}), \tag{1}$$

$$\phi_{es} = arcsin(\frac{-sin\Theta_{ps}}{cos\Theta_{ps}}), \tag{2}$$

$$\Theta_{ps} = arcsin(-cos\Theta_{es} \cdot cos\phi_{es}), \tag{3}$$

$$\phi_{ps} = arcsin(\frac{sin\Theta_{es}}{cos\Theta_{es}}), \tag{4}$$

где Θ_{es} — широта в экваториальной системе координат, ϕ_{es} — долгота в экваториальной системе координат, Θ_{ps} — широта в системе координат полярного сегмента, ϕ_{ps} — долгота в системе координат полярного сегмента.

Сегменты состоят из процедурно генерируемых блоков с центром в позиции наблюдателя, обновляющих каждый кадр подсистемы визуализации. Дистанции внутри блока определяются в координатной системе блока, начало которой находится в центре блока. Оси СК блока всегда совпадают с осями СК сегмента, в то время как начальные точки СК отличаются, соответственно координаты точки внутри блока в СК сегмента определяются с использованием дистанции от центра блока.

Описанные выше системы координат применяются при обработке данных, однако при прорисовке требуется использование прямоугольной системы координат. В подходе используется два вида подобных систем координат: геоцентрические СК и топоцентрические СК. Глобальная геоцентрическая СК используется для большинства расчетов, в то время как топоцентрическая СК используется при визуализации блоков.

При прорисовке масштабных виртуальных миров зачастую началом координат при прорисовке объектов является позиция наблюдателя. Это необходимо для сохранения точности вещественных типов данных. В предлагаемом подходе применяется топоцентрическая СК, началом которой является задаваемая точка на поверхности планеты, вместо использования позиции наблюдателя в качестве начала координат и пересчета

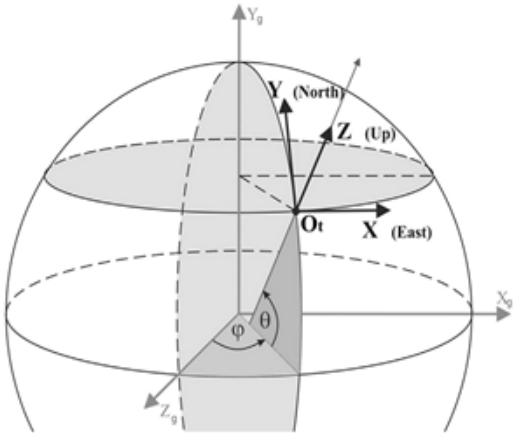

Рис. 2

координат отображаемых объектов при перемещении наблюдателя. В топоцентрической СК ось X направлена на восток, ось Y направлена на север, а ось Z направлена вверх, перпендикулярно плоскости XY (рис. 2).

Каждый сегмент отображаемой земной поверхности использует как минимум две выборки текстурных данных: одну для данных высот, а вторую для подстилающей поверхности. Все выборки являются независимыми и состоят из слоев, определяемых исходными данными. Каждый слой состоит из одного или нескольких блоков. Использование большого числа блоков в слое может повысить производительность визуализации при эффективном отбрасывании блоков, не попадающих в область видимости наблюдателя.

Одним из преимуществ предлагаемого подхода является возможность процедурного создания блоков, определяющих поверхность планеты, с уровнем детализации, зависящим от позиции наблюдателя. Необходимый уровень детализации блока достигается без рекурсивного деления сетки. Центр блоков находится в позиции наблюдателя, и оси СК блока совпадают с осями СК сегмента. Из-за того что СК сегментов в смежных областях различаются, блоки необходимо повернуть вокруг оси Z топоцентрической СК для исключения провалов в геометрии.

Отделение процесса наложения текстур подстилающей поверхности от процесса создания блока позволяет выровнять различные слои подстилающей поверхности попиксельно в фрагментном шейдере, в соответствии с дистанцией до наблюдателя в географических координатах сегмента. Аппроксимация, используемая для компенсации углового смещения и вращения, достаточна для отображения текстур, один пиксель которых по размеру не превышает 7,5 угловых секунд. Для текстур, превышающих этот параметр, будут присутствовать разрывы в уровнях детализации на линии соприкосновения двух сегментов, однако в большинстве случаев они будут не заметны из-за перекрытия ландшафтом и смешивания текстур различных слоев.

3. ПРОРИСОВКА БЛОКОВ

Блоки являются базовой единицей при создании геометрии ландшафта и создаются процедурно на каждом кадре системы визуализации. Блок или группа блоков задают уровень детализации ландшафта. Все данные, необходимые для обработки блока в вер-

шинном шейдере, передаются при помощи шейдерных констант. Для расчета позиции вершин требуются следующие параметры: позиция вершины в СК блоке; нормаль к поверхности (опционально); коэффициент смешивания уровней детализации по периметру блока; дистанции ограничения прорисовки (ОП) на основании уровня детализации и ближайшей границы сегмента; позиция вершины в локальной топоцентрической СК.

Первым этапом обработки вершины является расчет позиции вершины в блоке. Координаты блока используются в качестве текстурных координат для доступа к данным о высотах и текстурам подстилающей поверхности, а также применяются для расчета вращения нормали к поверхности.

В имитационных системах часто применяются изображения, полученные при помощи аэрофотосъемки, в качестве подстилающей поверхности. В случае, когда необходимо дополнительное освещение отображаемой земной поверхности (например имитация цикла день/ночь), необходим расчет нормалей к поверхности. Нормали могут быть рассчитаны заранее и сохранены в текстурных выборках или же могут процедурно генерировать каждый кадр в вершинном шейдере. Влияние расчета нормали в вершинном шейдере на производительность системы визуализации на современных видеокартах незначительно. Направление вектора нормали определяется склоном местности и дистанцией конкретной вершины до центра блока.

Для вершин в блоке также производится расчет границ ОП. При прорисовке блока некоторые его вершины могут быть перекрыты поверхностью с более высоким уровнем детализации. Дистанции и необходимость ОП вычисляется в вершинном шейдере, основываясь на позиции блока с более высоким уровнем детализации. Вершины блока, перекрытые вершинами блока с более высоким уровнем детализации, опускаются ниже для создания «юбки». «Юбка» используется для скрытия артефактов, образуемых при соприкосновении объектов с разным уровнем детализации.

Также вычисляется необходимость ограничения прорисовки по краям сегмента, если блок пересекает 45-ю параллель. В полярных сегментах для расчета необходимости ОП требуется преобразование широты и долготы вершины из системы координат блока в систему координат экваториального сегмента. Это необходимо для поддержания высокой производительности расчетов. Для блоков низкого уровня детализации дистанции ОП рассчитываются в вершинном шейдере на основании широты вершины в глобальной СК. Для блоков более высокого уровня детализации дистанции ограничения прорисовки рассчитываются на основании дистанции между центром блока и 45-ой параллелью, а также смещения конкретной вершины блока, полученного из-за поворота полярного сегмента относительно глобальной СК. Расчет смещения осуществляется центральным процессором на основании позиции центра блока.

Преобразование из СК блока в топоцентрическую СК выполняется при помощи проекции вектора расстояния между текущей позицией и началом топоцентрической СК на оси топоцентрической СК. Все вершины блока задаются в одной топоцентрической СК, поэтому начало координат и угол поворота осей СК для сокращения времени вычислений рассчитываются на центральном процессоре с использованием вещественных типов двойной точности (double). Однако остальные операции, включающие расчет позиции вершины в геоцентрической СК и проекцию на оси топоцентрической СК, выполняются в вершинном шейдере. Большинство современных GPU, за исключением мобильных, поддерживают вычисления с двойной точностью, но производительность таких вычислений значительно меньше по сравнению с вычислениями с одинарной точностью — в среднем вычисления с двойной точностью выполняются в восемь–

шестнадцать раз медленнее, чем вычисления с одинарной точностью [13]. Кроме того, тригонометрические и многие другие функции, аппаратно ускоряемые графическим адаптером, могут работать только с вещественными типами с одинарной точностью. Возможен также вариант с эмуляцией двойной точности в шейдерах. Однако данный подход ведет к повышению вычислительной нагрузки на графический адаптер, а также к общему усложнению реализации метода. В частности, все вершины, рассчитанные с использованием вещественных типов данных двойной точности, требуется преобразовать в два отдельных числа вещественного типа с одинарной точностью. Если данная операция выполняется каждый кадр системы визуализации на большем объеме данных, то возможно существенное снижение производительности системы визуализации. С учетом этого все расчеты, выполняемые данным методом на GPU, выполняются с одинарной точностью.

В зависимости от дистанции вершины до центра блока или группы блоков, позиция вершины в топоцентрической СК рассчитывается описанным выше способом или же с использованием интерполяции. Для небольших дистанций используется линейная интерполяция кривизны поверхности эллипсоида. Коэффициенты интерполяции рассчитываются центральным процессором для центра блока или группы блоков в СК сегмента. Полученные коэффициенты пересылаются в вершинный шейдер при помощи шейдерных констант.

Последним этапом, выполняемым вершинным шейдером, является получение дистанции ограничения прорисовки вершины путем перемножения топоцентрических координат вершины на проекционную и модельно-видовую матрицу.

Применение текстур подстилающей поверхности к ландшафту происходит в фрагментном (пиксельном) шейдере. В фрагментном шейдере происходит выбор необходимого слоя текстурной выборки для каждого фрагмента на основании дистанции в СК блоке, начиная от центра блока или группы блоков. Это необходимо для корректного отображения данных из различных источников с различным масштабом и цветовой схемой.

Дистанция текселя l и текстурные координаты s, t рассчитываются по формуле:

$$l = \frac{\sqrt{s^2 \cdot cos^2\Theta + t^2}}{cos\Theta_w}, \tag{5}$$

$$\Theta_w = \left\{ \begin{array}{c} \Theta \\ \frac{\pi}{2} - |\Theta| \end{array} \right\} \begin{array}{l} \text{для экваториального сегмента} \\ \text{для полярных сегментов,} \end{array} \tag{6}$$

где s, t — текстурные координаты, Θ — широта в глобальной СК, Θ_w — широтное расстояние.

Выбор необходимого слоя λ в текстурной выборке происходит на основе ранее рассчитанной дистанции l и расстояния видимого региона, для которого есть текстуры с максимальной детализацией (D_0). Видимый регион — это часть текстурной выборки, используемая для прорисовки в конкретный момент времени. Целая часть λ используется для выбора базового слоя в текстурной выборке, в то время как дробная часть используется в качестве смешивания с менее детализированным уровнем:

$$\lambda = max(0, 2 + \log_2(l \cdot D_0)). \tag{7}$$

Отделение процесса обработки ландшафта от применения текстур подстилающей поверхности позволяет использовать различные размеры для текстур, отвечающих за

данные о высотах, и текстур, отвечающих за подстилающую поверхность. За счет этого возможно применение высоко детализированных текстур подстилающей поверхности без значительного влияния на производительность системы визуализации.

4. ПРОРИСОВКА СЕГМЕНТОВ

При прорисовке сегментов выполняются следующие шаги:

— преобразование позиции наблюдателя из глобальной системы координат в локальную систему координат сегмента;
— расчет угла вращения координатной сетки (для полярных сегментов);
— расчет минимального размера блока или группы блоков;
— определение полноты текстурных выборок;
— прорисовка блока или группы блоков до тех пор, пока размер прорисовываемого блока не превышает пороговое значение; на каждой итерации размер увеличивается вдвое.

Размер наиболее детализированного блока (минимальный размер блока ρ) определяется на основании высоты наблюдателя над поверхностью h, горизонтального угла обзора камеры f и радиуса Земли R:

$$\rho = \frac{h}{R \cdot \tan(0,5 \cdot f)} \cdot \frac{180}{\pi}. \tag{8}$$

Количество слоев текстурной выборки, используемых при прорисовке сегмента, не определено заранее. Прорисовка слоев, включающих в себя блоки или группы блоков, осуществляется до тех пор, пока не превышен задаваемый максимальный размер блока. Перед прорисовкой текстурные выборки проверяются на полноту. Для каждого слоя происходит проверка текущей позиции наблюдателя и при необходимости происходит обновление содержимого слоя текстурной выборки. Также на этом этапе происходит выбор наибольшей детализации подстилающей поверхности в зависимости от высоты позиции наблюдателя.

При визуализации сегментов возникает ситуация, когда все данные текстурных выборок могут не поместиться в видео или оперативную память. Соответственно, требуется эффективный метод подгрузки и выгрузки ресурсов из памяти. Процесс обновления состоит из четырех этапов: загрузка (стриминг) ресурсов, кэширование, перекодирование, прорисовка. Рассмотрим процесс обновления текстурных выборок в целом и каждый этап в частности подробнее.

При перемещении наблюдателя происходит создание события о необходимости проверки полноты данных в текстурных выборках. В случае, когда данные уже присутствуют в памяти и не требуют обновления, происходит изменение сектора используемых данных о высотах и подстилающей поверхности на величину изменения позиции наблюдателя. Если же необходима подгрузка новых данных в одну из текстурных выборок, то определяется, есть ли данные в оперативной памяти; если данные присутствуют, то производится запрос на загрузку этих данных в видеопамять в соответствующий массив текстур. В случае, когда данных нет в оперативной памяти, производится запрос на чтение данных из внешних источников — в большинстве случаев это будет чтение из файла с жесткого диска или SSD. Одновременно с необходимой порцией данных могут быть загружены и смежные данные для сокращения количества обращений к низкоскоростным внешним источникам.

Запрос на чтение данных помещается в общую очередь обработки. Свободный поток из пула потоков начинает чтение данных из файла. По окончании чтения данных в оперативную память создается событие о необходимости обработки загруженных данных.

В зависимости от типа данных подбирается необходимый обработчик входящих данных. Обработка также осуществляется свободным потоком в пуле потоков. В случае загрузки данных, которые не могут напрямую быть помещены в видеопамять, происходит их декодирование и/или преобразование. Если происходит загрузка изображений, то этап преобразования может также включать сжатие данных в формат, поддерживаемый графическим чипом нативно — такими форматами являются DXT1-5, BC7, ETC2, ASTC. При использовании сжатых форматов достигаемая экономия занимаемой памяти составляет от 30 до 50 %.

Этап прорисовки включает в себя передачу данных на GPU и формирование вызова прорисовки. В случае, когда конечная система визуализации поддерживает многопоточный прием команд (например система визуализации поддерживает графические интерфейсы DirectX 12, Vulkan, Metal 2), передача данных на GPU может вестись одним из свободных потоков из пула потоков. В противном случае передача данных может происходить только в главном потоке приложения или потоке, в котором создан экземпляр контекста прорисовки.

5. ЗАКЛЮЧЕНИЕ

Для корректной прорисовки поверхности планеты требуется разделение поверхности на несколько частей. В противном случае, появляются искажения при отображении подстилающей поверхности. Предлагаемый метод разделяет планету на три сегмента, снижая число отдельных наборов данных. Каждый сегмент использует собственную текстурную выборку для ускорения подгрузки данных и уменьшения количества обращений к системе хранения данных. Применение вещественных типов данных одинарной точности для расчетов позволяют значительно ускорить вычисления на GPU по сравнению с использованием типов данных двойной точности или способов эмуляции двойной точности. Предлагаемый подход к подгрузке и выгрузке ресурсов позволяет снизить вероятность скачкообразного изменения производительности системы визуализации, а также обеспечить экономию используемой видеопамяти за счет сжатия текстурных данных в форматы, поддерживаемые GPU.

В дальнейшем планируется разработка методов отображения процедурно генерируемых объектов, таких как здания (на основании данных о площади и высоте), деревья, реки.

Список литературы

1. *Родителев А. В., Гиацинтов А. М.* Высокоуровневая архитектура тренажерно-обучающих систем сложных технических комплексов // Программные продукты и системы. 2018. № 31 (3). С. 439–443.
2. *Капралов Е. Г. и др.* Геоинформатика. М.: Академия, 2005. 477 с.
3. *Falby JS, Zyda MJ, Pratt DR, Mackey RL, Mackey YL.* NPSNET: hierarchical data structures for real-time three-dimensional visual simulation // Comput Graph. 1993. P. 65–69.
4. *Ulrich T.* Rendering massive terrains using chunked level of detail control. 2002.
5. *Cozzi P., Ring K.* 3D Engine Design for Virtual Globes. CRC Press, 2011.
6. *Iain Cantlay.* DirectX 11 Terrain Tessellation. NVIDIA Corporation, 2011.

7. *Khronos Group.* OpenGL Tessellation support [Электронный ресурс]. URL: https://www.khronos.org/registry/OpenGL/extensions/ARB/ARB_tessellation_shader.txt (дата обращения: 14.03.2019).
8. *Tanner C. C., Migdal C. J., Jones M.* The clipmap: a virtual mipmap // Proceedings of the 25th annual conference on computer graphics and interactive techniques. 1998. С. 151–158.
9. *Losasso F., Hoppe H.* Geometry clipmaps: terrain rendering using nested regular grids // ACM Trans Graph. 2004. № 23 (3). С. 769–776.
10. *Asirvatham A, Hoppe H.* Terrain rendering using gpu-based geometry clip-maps // GPU Gems 2. Addison-Wesley Professional, 2005. С. 27–45.
11. *Brown P, Leech J, Kilgard M.* EXT texture array [Электронный ресурс]. 2008. URL: http://www.opengl.org/registry/specs/EXT/texture_array.txt (дата обращения: 14.03.2019).
12. *Dimitrijević A. M., Rančić D. D.* Ellipsoidal Clipmaps — A planet-sized terrain rendering algorithm // Computers & Graphics. 2015. № 52. С. 43–61. doi: 10.1016/j.cag.2015.06.006
13. Performance of Nvidia Geforce 1080 TI [Электронный ресурс]. URL: https://db.thegpu.guru/card/GTX%201080%20Ti (дата обращения: 14.03.2019).
14. Terrain Rendering with Geometry Clipmaps [Электронный ресурс]. URL: https://arm-software.github.io/opengl-es-sdk-for-android/terrain.html (дата обращения: 14.03.2019).
15. *Song G., Yang H., Ji Y.* Geometry Clipmaps Terrain Rendering Using HardwareTessellation // IEICE TRANSACTIONS on Information and Systems. 2017. № 2. С. 401–404.

Поступила в редакцию 15.01.2019, окончательный вариант — 14.03.2019.

Computer tools in education, 2019
№ 2: 31–42
http://cte.eltech.ru
doi:10.32603/2071-2340-2019-2-31-42

Methods of Rendering Vast Landscapes in Training Simulation Systems

Giatsintov A. M. [1], PhD, senior researcher, algts@inbox.ru
Reshetnikov V. N.[1], professor, chief researcher, rvn@niisi.ras.ru
Roditelev A. V. [1], lead software engineer, avrod_94@mail.ru

[1]Research Institute for system studies of the Russian Academy of Sciences,
36, building 1, Nakhimovsky prosp., 117218, Moscow, Russia

Abstract

One of the tasks that arise during the development of training (simulation) systems is the creation of a subsystem for displaying the earth's surface. Subsystems of this class provide modeling and visualization of the landscape, underlying surface (rivers, roads, forests, etc.), geometric objects that simulate cities or airports and so on. There are two main approaches to the visualization of extended landscapes — continuous level of detail algorithms that use clustered triangulation and geometry clipmapbased algorithms. The method of generation and visualization of the Earth's surface presented in the work is based on the second approach. It is possible to perform calculations on the GPU using single precision, which allows faster calculations compared to using data types with double precision. In addition, this approach may be used on mobile graphics processors that do not support double precision. The proposed new method of dynamic resource management reduces the occupied video memory, which allows more detailed texture data to be loaded for a larger number of objects simultaneously.

Keywords: *WGS84, visualization, simulation systems, coordinate systems, rendering, earth surface.*

Citation: A. M. Giatsintov, V. N. Reshetnikov, and A. V. Roditelev, “Methods of Rendering Vast Landscapes in Training Simulation Systems,” *Computer tools in education*, no. 2, pp. 31–42, 2019 (in Russian); doi:10.32603/2071-2340-2019-2-31-42

References

1. A. V. Roditelev and A. M. Giatsintov, “High-level architecture of training simulation systems of complex technical systems,” *Software & Systems*, vol. 31, no. 3, pp. 439–443, 2018 (in Russian); doi: 10.15827/0236-235X.031.3.439-443
2. E.G. Kapralov, A.V. Koshkarev, V. S. Tikunov, *Geoinformatics*, Russia, Moscow: Akademy, 2005 (in Russian).
3. J. S. Falby, M. J. Zyda, D. R. Pratt, R. L. Mackey, and Y. L. Mackey, “NPSNET: hierarchical data structures for real-time three-dimensional visual simulation,” *Comput Graph.*, pp. 65–69, 1993.
4. T. Ulrich, “Rendering massive terrains using chunked level of detail control,” *SIGGRAPH Course Notes*, vol. 35, 2002.
5. P. Cozzi and K. Ring, *3D Engine Design for Virtual Globes*, AK Peters/CRC Press, 2011.
6. I. Cantlay, *DirectX 11 Terrain Tessellation*, NVIDIA Corporation, 2011.
7. Khronos Group, OpenGL Tessellation support, [Online]. Available: https://www.khronos.org/registry/OpenGL/extensions/ARB/ARB_tessellation_shader.txt
8. C. C. Tanner, C. J. Migdal, and M. Jones, “The clipmap: a virtual mipmap,” In *Proc. of the 25th annual conference on computer graphics and interactive techniques*, 1998, pp. 151–158; doi: 10.1145/1186562.1015799
9. F. Losasso and H. Hoppe, “Geometry clipmaps: terrain rendering using nested regular grids,” *ACM Trans Graph*, vol. 23, no. 3, pp. 769–776, 2004.
10. A. Asirvatham and H. Hoppe, “Terrain rendering using gpu-based geometry clip-maps,” *GPU Gems 2*, Addison-Wesley Professional, pp. 27–45, 2005.
11. P. Brown, J. Leech, and M. Kilgard, *EXT texture array*, 2008. [Online]. Available: http://www.opengl.org/registry/specs/EXT/texture_array.txt
12. A. M. Dimitrijević and D. D. Rančić. “Ellipsoidal Clipmaps — A planet-sized terrain rendering algorithm,” *Computers & Graphics*, no. 52, pp. 43–61, 2015; doi: 10.1016/j.cag.2015.06.006
13. *Performance of Nvidia Geforce 1080 TI* [Online]. Available: https://db.thegpu.guru/card/GTX%201080%20Ti
14. Terrain Rendering with Geometry Clipmaps, [Online]. Available: https://arm-software.github.io/opengl-es-sdk-for-android/terrain.html
15. G. Song, H. Yang, and Y. Ji, “Geometry Clipmaps Terrain Rendering Using HardwareTessellation,” *IEICE TRANSACTIONS on Information and Systems*, no. 2, pp. 401–404, 2017; doi: 10.1587/transinf.2016EDL8160

Received 15.01.2019, the final version — 14.03.2019.