

УДК 004.925  
DOI: 10.15827/0236-235X.128.665-672

Дата подачи статьи: 31.07.19  
2019. Т. 32. № 4. С. 665–672

## **Визуализация земной поверхности в имитационных системах**

*А.М. Гуацинтов*<sup>1</sup>, к.т.н., старший научный сотрудник, *algts@inbox.ru*

*К.А. Мамросенко*<sup>1</sup>, к.т.н., руководитель, *matrosenko\_k@niisi.ras.ru*

*П.С. Баженов*<sup>1</sup>, программист, *bps@niisi.ras.ru*

<sup>1</sup> *Центр визуализации и спутниковых информационных технологий ФНЦ НИИСИ РАН, г. Москва, 117218, Россия*

Визуализация планет применяется во многих отраслях – при разработке геоинформационных систем, мультимедийных и имитационных систем, тренажеров. В данной статье описаны подходы к отображению поверхности Земли, обеспечивающие визуализацию в режиме реального времени. Названы проблемы, которые возникают при визуализации протяженных ландшафтов, связанные с производительностью, преобразованием систем координат, точностью визуализации.

В работе представлен подход к визуализации поверхности Земли, основанный на применении текстурных выборов (slrmap). Он позволяет упростить подготовку данных для различных частей земной поверхности, а также сократить количество наборов подготавливаемых данных как подстилающей поверхности, так и текстур с данными высот.

Для реализации предлагаемого подхода была разработана архитектура компонента генерации и визуализации земной поверхности, встраиваемого в существующие системы визуализации. Преимуществами такой архитектуры являются отсутствие необходимости в разработке собственной специализированной системы визуализации с нуля и возможность использования существующих как открытых, так и проприетарных систем визуализации. Вычислительная нагрузка распределяется по доступным потокам в пуле потоков.

Одним из примеров многопоточной обработки является создание вызовов прорисовки, которые будут отправлены в систему визуализации. Алгоритм обработки поступающих вызовов прорисовки зависит от реализации интеграционного слоя с конкретной системой визуализации. Архитектурой компонента предусмотрена возможность задания времени, в течение которого будет производиться обработка данных внутри компонента. После выполнения каждой операции подсчитывается время, затраченное на выполнение операции, и определяется возможность продолжения обработки других задач. При превышении временного бюджета обработка задач будет прекращена до следующего вызова обновления компонента.

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

Увеличение числа и сложности технических систем приводит к необходимости создания новых видов технических средств обучения. Одним из них являются *тренажерно-обучающие системы* (ТОС). Подобные системы незаменимы в тех отраслях, где ошибки при обучении на реальных объектах могут привести к негативным последствиям, а их устранение – к значительным финансовым затратам. Данные системы применяются в военном деле, медицине, атомной энергетике, авиации, космосе. Этим и объясняется актуальность решения задач построения современных ТОС.

Под ТОС оператора сложной технической системы 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].

Одной из задач, возникающих при разработке ТОС, является создание подсистемы отображения земной поверхности. Подсистемы такого класса используются для визуализации реалистичной поверхности Земли, включающей в себя моделирование 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 [2] и развлекательных проектов.

Существуют несколько основных подходов к визуализации земной поверхности. Первый

подход предполагает использование плоскости. Он достаточно прост для реализации, однако требует применения различного вида проекций, таких как Меркатора, цилиндрических или конических. Использование проекций приводит к появлению определенного вида искажений, таких как искажения по площади или по углам.

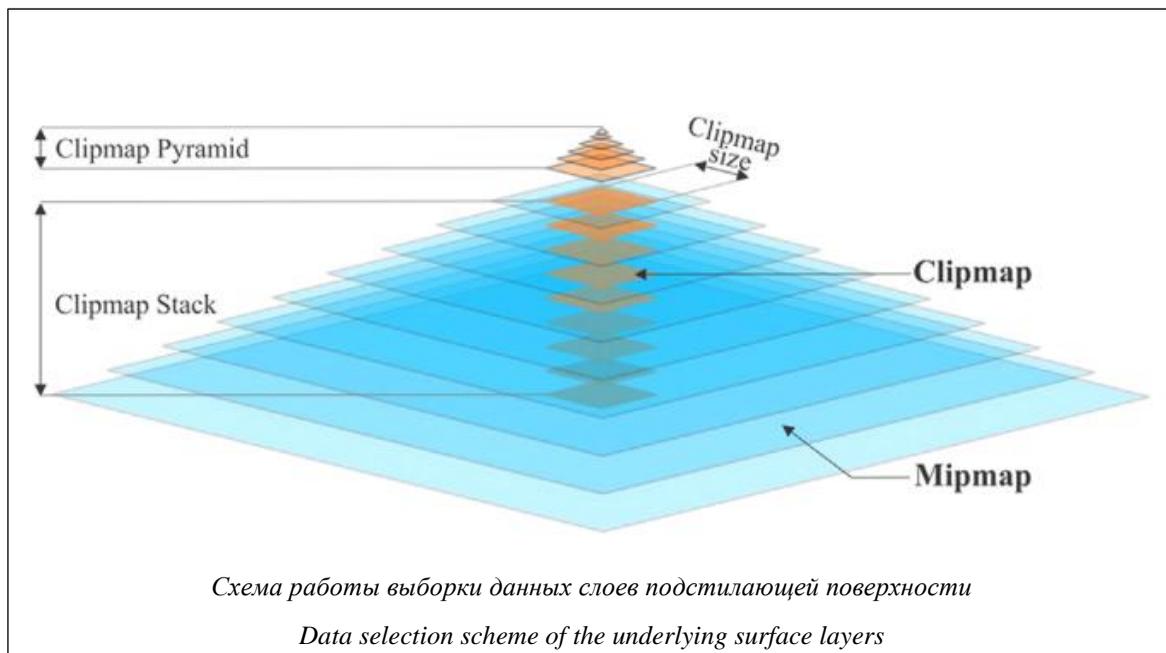
Второй подход предполагает использование сферической аппроксимации поверхности Земли. Он позволяет достичь более реалистичной визуализации, однако приводит к неточному позиционированию объектов на поверхности, в частности, по высоте.

Третий подход предполагает использование эллипсоида определенной формы. Стандартизированные системы координат, такие как WGS84 или ПЗ-90, определяют форму эллипсоида и другие его характеристики. В большинстве случаев различие в позиционировании одной и той же точки в различных системах координат не превышает 10 метров [3]. В настоящее время все большее количество информационных систем применяют эллипсоиды для аппроксимации поверхности Земли [4].

Другая группа алгоритмов предлагает использовать выборку текстурных данных (clipmap) для визуализации ландшафта [5]. 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, clipmap представляет собой ограниченную выборку данных изображения из конкретного слоя подстилающей поверхности (см. рисунок). 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 – только часть высокодетализированного изображения, обновляемого при перемещении наблюдателя.

Clipmap был представлен 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 он был адаптирован Френком Лосассо и Хью Хоппом [6]. 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 системы хранения данных. 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 [7], позволивший практически полностью перенести обработку геометрических clipmap на видеокарты, использующие Shader Model 3. Центральный процессор осуществлял только 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 его нельзя было использовать для визуализации протяженных ландшафтов из-за недостаточной точности при обработке детализированных сеток и сужения треугольников в окрестностях полюсов, что приводило к излишней детализации сетки без повышения качества визуализации.

С появлением видеокарт, поддерживающих Shader Model 4 [8], в конце 2006 года стало возможным использовать clipmap при помощи текстурных массивов [9]. 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, однако даже бюджетные карты предлагали до 512 слоев, позволяя дополнительно разгрузить работу центрального процессора за счет передачи нагрузки при обработке текстурных данных на видеокарту [10].

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

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

В статье предлагается новая архитектура компонента генерации и визуализации земной поверхности, встраиваемого в существующие системы визуализации. Преимуществами разработанной архитектуры являются:

- распределение вычислительной нагрузки по доступным потокам в пуле потоков;

- возможность задания времени, в течение которого будет производиться обработка данных внутри компонента;

- отсутствие необходимости разработки собственной специализированной системы визуализации с нуля и возможность использования существующих систем визуализации, как открытых, так и проприетарных.

### Метод генерации земной поверхности

Предлагаемый метод генерации земной поверхности предполагает разделение поверхности планеты на три сегмента – экваториальный и два полярных [11]. 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  $-45^\circ$  до  $+45^\circ$  и использует цилиндрическую проекцию. Полярные сегменты используют цилиндрические проекции, повернутые на  $90^\circ$  относительно оси X планеты, и ограничены широтами  $\pm 45^\circ$ . Основные преимущества разделения поверхности Земли на три сегмента:

- 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 низкодетализированной сферы;

- значительное количество исходных данных может быть использовано без применения дополнительных проекций (около 70 %);

- простота применения проекций для полярных сегментов;

- текстурные искажения, вносимые при использовании данного метода, незначительны.

Количество сегментов играет важную роль, если учитывать объем используемой как оперативной, так и видеопамати. Для каждого сегмента требуется свой набор данных, состоящий как минимум из двух выборок текстурных данных (clipmap) – данных высот и подстилающей поверхности. Каждая выборка текстурных данных представлена в виде текстурного массива.

В предлагаемом подходе применяются несколько систем координат при обработке данных и последующей прорисовке – глобальная система координат, система координат сегмента, система координат блока. Каждая си-

стема координат оперирует понятиями 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  $\pm 90^\circ$ .

Преобразование между системами координат сегментов производится следующим образом:

$$\begin{aligned}\theta_{es} &= \arcsin(\cos\theta_{ps} \cos\varphi_{ps}); \\ \varphi_{es} &= \arcsin(-\sin\theta_{ps}/\cos\theta_{ps}); \\ \theta_{ps} &= \arcsin(-\cos\theta_{es} \cos\varphi_{es}); \\ \varphi_{ps} &= \arcsin(\sin\theta_{es}/\cos\theta_{es}),\end{aligned}$$

где  $\theta_{es}$  – широта в системе координат экваториального сегмента;  $\varphi_{es}$  – долгота в системе координат экваториального сегмента;  $\theta_{ps}$  – широта в системе координат полярного сегмента;  $\varphi_{ps}$  – долгота в системе координат полярного сегмента.

Сегменты состоят из процедурно генерируемых блоков, обновляемых на каждом кадре подсистемы визуализации, центр которых находится в позиции наблюдателя. Дистанции внутри блока определяются в системе координат блока, начало которой находится в центре блока. Оси системы координат блока всегда совпадают с осями системы координат сегмента, в то время как начальные точки системы координат отличаются: координаты внутри блока являются дистанцией от центра блока в системе координат сегмента.

Описанные выше системы координат применяются при обработке данных, однако при прорисовке требуется прямоугольная система координат. В подходе используются два вида подобных систем координат: геоцентрические и топоцентрические. Глобальная геоцентрическая система координат применяется для большинства расчетов, в то время как топоцентрическая система координат – при визуализации блоков.

Каждый сегмент отображаемой земной поверхности использует как минимум две выборки текстурных данных: одну для данных высот, а другую для подстилающей поверхности. Все выборки являются независимыми и состоят из слоев, определяемых исходными данными. В каждом слое один или несколько блоков. Использование большого числа блоков в слое может повысить производительность визуализации при эффективном отбрасывании

блоков, не попадающих в область видимости наблюдателя.

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 вершинном шейдере, передаются при помощи шейдерных констант.

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 превысит пороговое значение; на каждой итерации размер увеличивается вдвое.

### Архитектура компонента генерации и визуализации земной поверхности

В большинстве открытых источников предполагается создание специализированной системы визуализации, основной задачей которой является эффективная визуализация земной поверхности. К сожалению, во многих случаях это означает отсутствие многих функций, реализованных в других системах визуализации, или недостаточное качество их реализации. Например, подобные специализированные системы визуализации могут иметь проблемы с визуализацией большого числа анимированных трехмерных объектов с высокой детализацией, находящихся на отображаемой земной поверхности.

Авторами разработана архитектура компонента генерации и визуализации земной поверхности, встраиваемого в существующие системы визуализации. Преимуществами такого подхода являются отсутствие необходимости в разработке собственной специализированной системы визуализации с нуля и возможность использования существующих систем визуализации, как открытых, так и проприетарных. Рассмотрим архитектуру предлагаемого компонента подробнее.

Компонент разделен на три основные части: ядро, внешние зависимости и интеграционная прослойка с системой визуализации. В ядре представлена основная функциональность компонента – создание эллипсоида, управление детализацией, обработка поступающих команд. Внешние зависимости отвечают за выполнение таких операций, как обработка входящих изображений, представленных в различных форматах. Интеграционный слой отвечает за взаимодействие с системой визуализации, обеспечивая передачу моделей, текстур и шейдеров в форматах, требуемых системой визуализации.

Функциональность ядра разделена на несколько категорий: ресурсы, узлы сцены, менеджеры, генераторы, обработчики входных форматов, сервисные функции.

При помощи ресурсов обеспечивается управление входящими данными различных типов. В настоящее время архитектурой предусмотрены два типа ресурсов – геометрия и изображение. Ресурс «геометрия» содержит данные о вершинах, нормалях и текстурных координатах геометрической модели. Ресурс «изображение» содержит текстурные данные, применяемые как для хранения данных о высотах ландшафта, так и для имитации поверхности трехмерных объектов.

Узлы сцены используются для хранения информации о конкретном типе трехмерного объекта, например, планеты. В узле сцены типа «планета» могут храниться информация о применяемом эллипсоиде, ссылки на используемые буферы вершин и текстуры.

Менеджеры отвечают за общее управление частями компонента визуализации. В частности, в архитектуре предусмотрены менеджеры ресурсов, узлов сцены, рендеринга и утилит. Менеджер ресурсов предоставляет функциональность по работе с ресурсами, используемыми в компоненте, такую как загрузка и выгрузка ресурсов. Менеджер узлов сцены обеспечивает общее управление узлами, присутствующими в виртуальной сцене: добавление и удаление узлов, а также изменение свойств для групп узлов. Менеджер рендеринга осуществляет связь внутренних частей ядра с интеграционным слоем системы визуализации. Менеджер утилит используется для внутренних операций компонента, не подпадающих под другие категории.

Основная работа, связанная с визуализацией, производится в генераторах. Генератор отвечает за прорисовку определенного типа

или типов узлов сцены, определяя параметры обработки вершин, отправляемых на прорисовку (как системе визуализации и нижележащему графическому интерфейсу следует интерпретировать поступающие данные о вершинах, нормалях, текстурных координатах и т.д.). В генераторах также определяется уровень детализации отображаемых объектов на основании позиции наблюдателя.

Обработчики входных форматов предназначены для интерпретации и обработки данных, поступающих в различных форматах. Примерами подобных обработчиков может служить обработчик формата kml, определяющий позиции объектов на виртуальной земной поверхности, или обработчик изображений, определяющий формат и параметры входного изображения.

В сервисные функции входят функции ядра, необходимые для работы, но не попадающие в другие категории. В частности, в сервисные функции входят управление памятью, математическая библиотека, журналирование, управление файлами.

Интеграционный слой состоит из интерфейса взаимодействия с системой визуализации, предоставляемого компонентом, и реализации этого интерфейса для конкретной системы визуализации. Интерфейс взаимодействия определяет необходимую для работы компонента функциональность. В частности, он позволяет запросить у системы визуализации наличие поддержки различных форматов (текстурных, шейдерных) или возможность многопоточной обработки поступающих от компонента данных.

В настоящее время в связи с массовой распространенностью многоядерных процессоров целесообразно использовать многопоточный подход к обработке данных – использование только одного потока позволит задействовать лишь малую часть вычислительной мощности процессора. Для эффективного использования многоядерных процессоров определенные функции должны быть заложены на этапе проектирования архитектуры, в частности, разделение объема вычислительной нагрузки на небольшие подзадачи, которые могут быть выполнены независимо друг от друга.

В представленной архитектуре вычислительная нагрузка распределяется по доступным потокам в пуле потоков. Количество потоков в пуле потоков зависит от количества ядер используемого процессора или может быть задано пользователем. Одним из примеров мно-

гопотоchnой обработки является создание вызовов прорисовки, которые будут отправлены в систему визуализации. При обновлении внутреннего состояния системы генераторы, отвечающие за прорисовку отдельных типов объектов, независимо друг от друга создают собственные очереди вызовов прорисовки, которые затем передаются в систему визуализации при помощи интеграционного слоя. Алгоритм обработки поступающих вызовов прорисовки зависит от реализации интеграционного слоя с конкретной системой визуализации.

При визуализации земной поверхности может возникнуть ситуация, когда время обработки данных в компоненте превысит определенные границы, что приведет к снижению общей производительности системы визуализации. Система визуализации должна обеспечивать визуализацию с частотой смены кадров не менее 25 в секунду. Предпочтительным значением является частота смены кадров, составляющая не менее 60 в секунду, или 16.67 миллисекунд на один кадр. Рассмотрим случай, когда на время формирования кадра влияют только компонент генерации и визуализации земной поверхности и система визуализации без сторонних компонентов (например, обработки физической модели). В этом случае в зависимости от детализации отображаемой виртуальной сцены время выполнения задач компонентом в среднем должно составлять от 2 до 8 миллисекунд.

Компонентом предусмотрена возможность задания времени, в течение которого будет производиться обработка данных внутри компонента. После выполнения каждой операции подсчитывается время, затраченное на выполнение операции, и определяется, есть ли возможность продолжить обработку других задач. Если временной бюджет превышен, обработка задач будет прекращена до следующего вызова обновления компонента. В зависимости от ресурсоемкости выполняемых компонентом задач заданное граничное время может быть превышено в случае, когда время выполнения одной задачи превышает заданный лимит.

Предлагаемая архитектура компонента является расширяемой. Интерфейс взаимодействия с компонентом предоставляет возможность регистрации новых типов ресурсов, узлов сцены, генераторов, обработчиков входных форматов. Также при необходимости могут быть изменены функции работы с памятью и файлами, что позволяет эффективнее интегрировать компонент с существующими системами визуализации, использующими собственные алгоритмы выделения памяти и обработки файлов.

### Заключение

Для корректной прорисовки поверхности планеты требуется ее разделение на несколько частей. В противном случае появляются искажения при отображении подстилающей поверхности. Предлагаемый метод предусматривает разделение планеты на три сегмента, снижая число отдельных наборов данных. Каждый сегмент использует собственную текстурную выборку для ускорения подгрузки данных и повышения эффективности кэширования.

Для генерации и визуализации земной поверхности разработана новая расширяемая архитектура встраиваемого компонента. Преимуществом предлагаемой архитектуры является независимость от конкретной системы визуализации, что позволяет использовать существующие как открытые, так и проприетарные системы без необходимости разработки собственной специализированной системы с нуля. Также архитектурой предусмотрена возможность задания времени, в течение которого будут обрабатываться данные внутри компонента. В случае превышения временного бюджета обработка задач прекращается до следующего вызова обновления компонента. Данная возможность в большинстве случаев позволяет избежать скачкообразного снижения производительности применяемой системы визуализации и сохранить частоту смены кадров на уровне не менее 25 в секунду.

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

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

### Литература

1. Родителив А.В., Гиацинтов А.М. Высокоуровневая архитектура тренажерно-обучающих систем сложных технических комплексов // Программные продукты и системы. 2018. Т. 31. № 3. С. 439–443. DOI: 10.15827/0236-235X.123.439-443.

2. Сухов А.В., Решетников В.Н., Савилкин С.Б. Мониторинг частотного ресурса геостационарных спутников-ретрансляторов с использованием энтропии покрытия // Программные продукты и системы. 2017. № 1. С. 119–123. DOI: 10.15827/0236-235X.117.119-123.
3. Капралов Е.Г., Кошкарёв А.В., Тикунов В.С. Геоинформатика. М.: Академия, 2005. 480 с.
4. Cozzi P., Ring K. 3D Engine Design for Virtual Globes. CRC Press Taylor & Francis Group, 2011, 514 p.
5. Tanner C.C., Migdal C.J., Jones M. The clipmap: a virtual mipmap. Proc. 25th Conf. on Comp. Graph. and Inter. Tech., 1998, pp. 151–158.
6. Losasso F., Hoppe H. Geometry clipmaps: Terrain rendering using nested regular grids. ACM Transactions on Graphics, 2004, no. 23, pp. 769–776. DOI: 10.1145/1015706.1015799.
7. Asirvatham A., Hoppe H. Terrain rendering using GPU-based geometry clipmaps. GPU Gems 2. 2005, pp. 27–45.
8. Shader Model 4. Windows applications. URL: <https://docs.microsoft.com/en-us/windows/win32/direct3dhls/dx-graphics-hlsl-sm4> (дата обращения: 07.07.2019).
9. Brown P., Leech J., Kilgard M. EXT texture array. URL: [http://www.opengl.org/registry/specs/EXT/texture\\_array.txt](http://www.opengl.org/registry/specs/EXT/texture_array.txt) (дата обращения: 07.07.2019).
10. Khronos Group. OpenGL Tessellation support. URL: [https://www.khronos.org/registry/OpenGL/extensions/ARB/ARB\\_tessellation\\_shader.txt](https://www.khronos.org/registry/OpenGL/extensions/ARB/ARB_tessellation_shader.txt) (дата обращения: 07.07.2019).
11. Dimitrijević A.M., Rančić D.D. Ellipsoidal Clipmaps – A planet-sized terrain rendering algorithm. Comput. Graph. 2015, vol. 52, iss. C, pp. 43–61. DOI: 10.1016/j.cag.2015.06.006.

Software & Systems  
DOI: 10.15827/0236-235X.128.665-672

Received 31.07.19  
2019, vol. 32, no. 4, pp. 665–672

### Earth's surface visualization in simulation systems

*A.M. Giachintov*<sup>1</sup>, Ph.D. (Engineering), Senior Researcher, [algts@inbox.ru](mailto:algts@inbox.ru)

*K.A. Mamrosenko*<sup>1</sup>, Ph.D. (Engineering), Head of the Center, [mamrosenko\\_k@niisi.ras.ru](mailto:mamrosenko_k@niisi.ras.ru)

*P.S. Bazhenov*<sup>1</sup>, Programmer, [bps@niisi.ras.ru](mailto:bps@niisi.ras.ru)

<sup>1</sup> Center of Visualization and Satellite Information Technologies SRISA,  
Moscow, 117218, Russian Federation

**Abstract.** Planet visualization is used in many fields: development of geographic information systems, multimedia systems, simulation systems and simulators. The present paper describes approaches to displaying the Earth's surface that provide real-time visualization. It also lists the problems that arise during extended landscapes visualization and are related to productivity, the coordinates system transformation and visualization accuracy.

The paper presents an approach to the Earth's surface visualization that is based upon the use of clipmaps. It allows simplifying data preparation for various parts of the Earth's surface, as well as reducing the number of prepared data sets for both the underlying surface and textures with height data.

To implement the proposed approach, the architecture of the component used to generate and visualize the Earth's surface was developed. The component is built into existing visualization systems. This architecture has the following advantages. It is not necessary to develop own specialized visualization system from scratch. It is possible to use existing visualization systems both open and proprietary. Computing load is distributed across available threads in the thread pool.

Creation of drawing calls that will be sent to the visualization system is one of multi-threaded processing examples. The algorithm for processing incoming drawing calls depends on implementation of integration with the particular visualization system. Component's architecture provides the ability to set the time during which the data will be processed inside the component. After each operation is finished the time taken to complete

the operation is calculated and the ability to continue processing other tasks is determined. If the time limit is exceeded, task processing will be terminated until the next component update call.

**Keywords:** WGS84, terrain, visualization, simulation systems, coordinates system.

**Acknowledgements.** The paper was financially supported by RFBR, grant no. 17-07-00169 A.

### References

1. Roditelev A.V., Giatsintov A.M. High-level architecture of training simulation systems of complex technical systems. *Programmnye Produkty i Sistemy [Software and Systems]*. 2018, vol. 31, no. 3, pp. 439–443 (in Russ.). DOI: 10.15827/0236-235X.123.439-443.
2. Sukhov A.V., Reshetnikov V.N., Savilkin S.B. Monitoring of frequency resource of geostationary repeater satellites using cover entropy. *Programmnye Produkty i Sistemy [Software and Systems]*. 2017, no. 1, pp. 119–123 (in Russ.). DOI: 10.15827/0236-235X.117.119-123.
3. Kapralov E.G., Koshkarev A.V., Tikunov V.S. *Geoinformatics*. Moscow, Akademiya Publ., 2005. 477 p. (in Russ.).
4. Cozzi P., Ring K. *3D Engine Design for Virtual Globes*. CRC Press, 2011, 514 p.
5. Tanner C.C., Migdal C.J., Jones M. The clipmap: a virtual mipmap. *Proc. 25th Conf. on Comp. Graph. and Inter. Tech.*, 1998, pp. 151–158.
6. Losasso F., Hoppe H. Geometry clipmaps: terrain rendering using nested regular grids. *Proc. ACM TOG*. 2004, no. 23, pp. 769–776. DOI: 10.1145/1015706.1015799.
7. Asirvatham A., Hoppe H. Terrain rendering using GPU-based geometry clipmaps. *GPU Gems*. 2005, vol. 2, no. 2, pp. 27–45.
8. *Shader Model 4 Windows Applications+*. Available at: <https://docs.microsoft.com/en-us/windows/win32/direct3dhlsldx-graphics-hlsl-sm4> (accessed August 07, 2019).
9. Brown P., Leech J., Kilgard M. *EXT Texture Array*. Available at: [http://www.opengl.org/registry/specs/EXT/texture\\_array.txt](http://www.opengl.org/registry/specs/EXT/texture_array.txt) (accessed August 07, 2019).
10. *Khronos Group. OpenGL Tessellation Support*. Available at: [https://www.khronos.org/registry/OpenGL/extensions/ARB/ARB\\_tessellation\\_shader.txt](https://www.khronos.org/registry/OpenGL/extensions/ARB/ARB_tessellation_shader.txt) (accessed August 07, 2019).
11. Dimitrijević A.M., Rančić D.D. Ellipsoidal Clipmaps – A planet-sized terrain rendering algorithm. *Comput. Graph.* 2015, no. 52, pp. 43–61. DOI: 10.1016/j.cag.2015.06.006.

### Для цитирования

Гиацинтов А.М., Мамросенко К.А., Баженов П.С. Визуализация земной поверхности в имитационных системах // Программные продукты и системы. 2019. Т. 32. № 4. С. 665–672. DOI: 10.15827/0236-235X.128.665-672.

### For citation

Giachintov A.M., Mamrosenko K.A., Bazhenov P.S. Earth's surface visualization in simulation systems. *Software & Systems*. 2019, vol. 32, no. 4, pp. 665–672 (in Russ.). DOI: 10.15827/0236-235X.128.665-672.