

спутника охватывает всю территорию Советского Союза и всю северную полярную область Земли. Получить такую же зону радиовидимости со спутников, находящихся на геостационарной орбите в любых точках над экватором, невозможно.

С вводом в эксплуатацию многоствольного спутника «Горизонт», выведенного на геостационарную орбиту в 1975 г., завершилось создание новой системы спутниковой связи в нашей стране. Система была названа по имени спутника – «Горизонт». Первые спутники системы были расположены в точках стояния над экватором  $14^\circ$  з. д.;  $35^\circ$  и  $80^\circ$  в. д. На каждом из них было по шесть стволов, работающих в диапазоне 4/6 ГГц, и один ствол в диапазоне 11/14 ГГц.

В 1976 г. был выведен на геостационарную орбиту спутник «Экран» с точкой стояния над экватором  $99^\circ$  в. д., который позволил создать зону непосредственного телевизионного вещания в районах Урала и Западной Сибири.

В 1965 г. начала действовать самая крупная международная организация спутниковой связи ИНТЕЛСАТ. С помощью этой системы в 1987 г. обеспечивалось около двух третей международных каналов спутниковой связи, а в настоящее время – около одной трети.

В 1980-е гг. началось развитие персональной спутниковой связи. При этой связи спутниковый телефон непосредственно соединяется со спутником, находящимся на околоземной орбите.

В настоящее время в общем балансе связи на спутниковые системы пока приходится примерно 3 % мирового трафика. Но потребности в спутниковых линиях продолжают расти, поскольку при дальности свыше 800 км спутниковые каналы становятся экономически более выгодными по сравнению с другими видами дальней связи.

#### **Список используемых источников**

1. Украинцев Ю.Д., Цветов М.А. История связи и перспективы развития телекоммуникаций: учебное пособие. Ульяновск: УлГТУ, 2009. 128 с.
2. Танетов В.Н. От «внеземной ретрансляции» к первым спутникам связи // Электро-связь. 1998. № 11.
3. Сворень Р. Щедрая звезда «Экрана»: Заметки с международной выставки «Связь-81» // Наука и жизнь. 1981. № 12, С. 22–24, С. 2–3 вкл.

## **РЕШЕНИЕ ЗАДАЧИ ВИЗУАЛИЗАЦИИ ПРОЦЕДУРНО ГЕНЕРИРУЕМОГО ЛАНДШАФТА ПРИ ПОМОЩИ АЛГОРИТМА НА ОСНОВЕ СТРУКТУРЫ «ДЕРЕВО КВАДРАНТОВ»**

**Ю.Ф. Болтов, Л.С. Дудаков, А.В. Тарлыков**

*В статье описан алгоритм визуализации процедурно генерируемого ландшафта, который слабо занимает процессор и практически полностью оставляет для приложения. Кратко затронута тема процедурной генерации ландшафта.*

*Ключевые слова: компьютерная графика, ландшафт, визуализация, генерация, дерево квадрантов.*

## SOLUTION OF VISUALIZATION TASKS PROCEDURALLY GENERATED LANDSCAPE USING ALGORITHM BASED ON STRUCTURE OF «QUADTREE»

Boltov Yu., Dudakov L., Tarlykov A.

*This article describes the visualization algorithm procedurally generated landscape that occupies little processor and almost completely leaves for the application. Briefly touched upon the procedural terrain generation.*

*Keywords: computer graphics, landscape, visualization, generation, tree quadrants.*

### **Введение**

Визуализация ландшафта является важной задачей в области компьютерной графики. В играх и других приложениях часто присутствует необходимость отрисовки больших открытых территорий. Создание этих территорий является трудоемкой задачей для человека, поэтому часто её выполнением частично или полностью занимается компьютер. Таким образом, для использования ландшафта в компьютерной графике необходимо решить две задачи: его генерацию и визуализацию.

### **1. Обзор задач**

Задача генерации ландшафта не является основной для этой статьи, поэтому будут рассмотрены лишь некоторые аспекты в области взаимодействия с задачей визуализации. Для генерации будут использованы комбинации многомерных функций, а именно: симплекс шум и диаграмма Вороного. Правильно подобранная сумма результатов работы таких функций неплохо справляется с задачей генерации, при этом вычисление такой суммы вполне возможно в реальном времени.

Сложность задачи визуализации состоит в том, что наблюдатель должен видеть все детали ландшафта на значительном расстоянии: как крошечные детали вблизи, так и крупные элементы вдалеке. В этом же состоит и основной путь оптимизации: вблизи ландшафт должен отображаться детализированно (высокий уровень детализации), а вдалеке можно обойтись гораздо меньшей детализацией (низкий уровень детализации), снижая тем самым количество данных для отрисовки.

Ландшафт может быть задан трехмерной моделью, или функцией от координат точки. В предлагаемой реализации ландшафт не моделируется заранее, а генерируется, поэтому представляться он будет функцией от координат. Можно выделить два вида таких функций: функция плотности и функция высоты. Функция плотности – это функция от координат точки  $(x, y, z)$ , которая воз-

вращает значение: если оно положительно, точка находится внутри ландшафта, если отрицательно – снаружи (в воздухе), если равно нулю, точка находится на поверхности ландшафта. Функция высоты – это функция от координат точки, которая возвращает расстояние, на которое эту точку необходимо сместить вверх (или вниз, если значение отрицательное). Использование функции высоты имеет ограничение в строгой горизонтальной направленности ландшафта. С ее помощью нельзя задать ландшафт с пещерами или навесами. Но это ограничение, будучи минусом, значительно упрощает и ускоряет работу с таким ландшафтом за счет устранения одного измерения. Поэтому алгоритм, описанный в этой статье, работает именно с ландшафтом, заданным такой функцией.

Далее будет описан алгоритм визуализации, который практически не задействует процессор при работе и хорошо подходит для отображения ландшафта, генерируемого в реальном времени. Он основывается на простейшем алгоритме визуализации ландшафта с помощью дерева квадрантов, но имеет и свои особенности. Освобождение процессора является важным моментом, поскольку именно процессор обычно занят логикой самого приложения или компьютерной игры.

## **2. Описание базового алгоритма на основе дерева квадрантов**

Дерево квадрантов находится в основе многих алгоритмов визуализации с уровнями детализации. Описать базовый алгоритм его использования можно следующим образом:

- а) В основании дерева находится квадрат (четыре трехмерные точки);
- б) При необходимости повышения детализации (приближение камеры) он делится на четыре квадрата. Подобное деление применимо и к созданным дочерним квадратам;
- в) При необходимости понижения детализации (отдаление камеры) четыре дочерних квадрата удаляются, вместо них остается один родительский. Удаления возможны пока не останется один квадрат – корень дерева.

При рисовании дерева происходит отбрасывание ветвей, которые находятся вне области видимости. Эта операция называется отсечением по пирамиде вида и подробнее про нее можно прочесть в статье [1, 2, 3, 4].

## **3. Недостатки базовой версии**

Базовая версия алгоритма предполагает, что каждый квадрат представляет из себя четыре трехмерные точки. И если вызывать процедуру рисования для каждого квадрата, то происходит обработка всего лишь этих четырех точек. Загружать все точки в буфер не представляется возможным, поскольку структура динамична и все время перестраивается при движении камеры. При приближении сильно возрастает количество квадратов-листьев в дереве, вместе с тем возрастает количество вызовов процедуры рисования. Это неэффективно: много раз вызывать рисование малого числа элементов. К тому же приходится хранить все точки, не смотря на то, что в соседних квадратах они дублируются.

## 4. Расширение алгоритма

### Сетка вместо четырех точек в элементе дерева

Загрузка всех точек в единый буфер не представляется возможным, поэтому необходимо вызывать процедуру рисования для каждого листа дерева отдельно. Очевидно, что для повышения эффективности вызовов квадрат придется заменить на структуру, содержащую больше данных о ландшафте, чем четыре точки. Самым простым вариантом является квадратная регулярная сетка точек (рис. 1а). Регулярной называется сетка, все точки которой равномерно расположены. Вместо одного квадрата элемент дерева теперь представляет из себя сетку  $N \times N$  квадратов. Вызовы процедуры рисования становятся намного более эффективными, и кроме того снижается число этих вызовов. Действительно, раз один элемент становится эквивалентен  $N \times N$  элементам старого дерева, то и для поддержания требуемого уровня детализации при приближении камеры придется производить гораздо меньше делений. Меньше делений – меньше листьев у дерева, а значит и меньше вызовов процедуры рисования.

На рисунке можно наглядно наблюдать использование дерева квадратов с регулярной сеткой в качестве элемента дерева: а) дерево состоит из одного элемента-основания; б) появляются четыре дочерних элемента (выделены красным), с аналогичными сетками  $4 \times 4$ ; в) один из красных элементов делится, образуя четыре дочерних (выделены зеленым), с теми же сетками. Размер  $4 \times 4$  используется только для демонстрации и не слишком повысит эффективность рисования по сравнению с алгоритмом без сетки. При решении реальных задач нужно использовать больший размер сетки, например,  $64 \times 64$ .

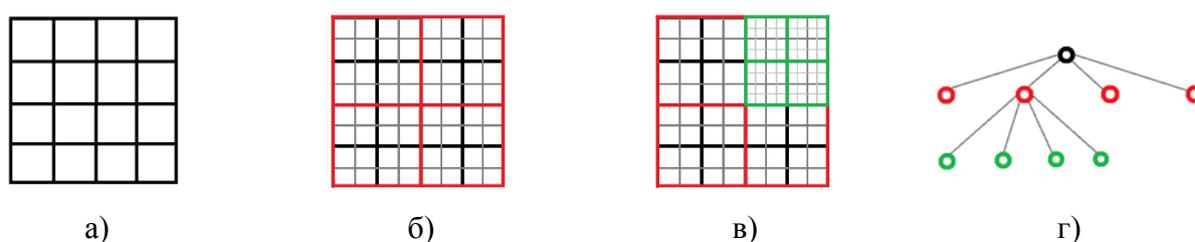


Рисунок. Дерево квадратов с использованием сетки  $4 \times 4$

### Одна единственная сетка на все элементы дерева

Решение одной проблемы найдено. И общее число точек снизилось за счет того, что в сетке дублирующиеся точки квадратов представляются в единичном экземпляре. Но возможно ли еще снизить количество точек?

Из рисунка видно, что сетки разных элементов различаются лишь масштабом и высотой точек. Каждая точка в сетке обладает уникальным значением высоты. В данной реализации высота точки – это результат функции высоты от координат точки. Иными словами, высоту не обязательно хранить. Ее можно вычислить в любой момент, зная лишь координаты точки. А если не хранить высоты, то все сетки становятся абсолютно одинаковыми за исключением координат центра и масштаба.

Главная особенность, предлагаемого в данной статье алгоритма, заключается в использовании для отрисовки всех элементов одной единственной квадратной регулярной сетки двумерных (без значений высоты) точек. Левый верхний ее угол имеет координаты  $(-1; -1)$ , а правый нижний  $(1; 1)$ . В элементе дерева (квадратной области) находится информация о центре и о масштабе.

При отрисовке ландшафта на графический конвейер один раз передается сетка точек. Далее выполняется обход дерева и для конечных элементов дерева (листьев), которые находятся внутри пирамиды вида, выполняется следующее:

- В видеокарту посылаются данные о трансформации квадратной области (центр, масштаб).

- На этапе вершинного шейдера к каждой точке сетки применяются трансформации (сдвиг, масштабирование). Рассчитывается и применяется к точке значение функции высоты от полученных координат. Так сетка принимает облик той части ландшафта, которая относится к текущей квадратной области.

- На этапе фрагментного шейдера через градиент функции высоты вычисляется нормаль. По значениям высоты и нормали генерируется цвет фрагмента. Применяется освещение.

Таким образом, значительно сокращается количество данных, передаваемых видеокarte для отрисовки.

### **Использование карт высот, нормалей, цветов**

Если функция высоты является простой, то на данном этапе можно остановиться. Но получение сложного и интересного ландшафта возможно только с использованием сложной функции. А ее вычисление при каждом рисовании является слишком затратной операцией. Хорошо было бы по-прежнему не хранить значения высот и использовать одну единственную сетку для отрисовки, но при этом производить генерацию только по мере необходимости (при разделении квадрата на четыре дочерних). Решением этой задачи будет использование карт высот, нормалей и цветов. Карты представляют из себя текстуры с данными о куске ландшафта, который соответствует данной квадратной области. Для заполнения карты высот необходимо рассчитать значения функции высоты в квадратной области точек. Такие массовые и независимые расчеты хорошо подходят для вычисления на видеокarte, тем более можно сразу производить запись результатов в текстуру, и не произойдет дополнительной нагрузки процессора. В используемом для этого фрагментном шейдере описаны функции генерации (шумы, ячеечные шумы и др.) и итоговая функция высоты. Затем по карте высот создается карта нормалей. Создание карты нормалей в отдельной операции экономит на расчетах градиентов (значения функции высоты берутся из карты высот). Во время этой операции можно также вычислять значения цветов и записывать их в карту цветов.

## Заключение

В данной статье был описан эффективный алгоритм визуализации процедурно генерируемого в реальном времени ландшафта. Он прост в реализации и практически не занимает времени процессора, что является немаловажным фактором при разработке требовательных к процессору приложений и компьютерных игр. Также этот алгоритм обеспечивает высокую производительность на видеокартах средней мощности.

## Список используемых источников

1. Бессмельцев М. Отсечение по пирамиде видимости в OpenGL [Электронный ресурс] // Работа с OpenGL. [2004]. URL: <http://pmg.org.ru/nehe/nehex2.htm> (дата обращения 11.06.2016).
2. Noise [Электронный ресурс] // Shadertoy Wikia: [сайт]. URL: <http://shadertoy.wikia.com/wiki/Noise> (дата обращения 27.05.2016).
3. Framebuffer Object [Электронный ресурс] // OpenGL Wiki: [сайт]. [2015]. URL: [https://www.opengl.org/wiki/Framebuffer\\_Object](https://www.opengl.org/wiki/Framebuffer_Object) (дата обращения 03.06.2016).
4. Framebuffer Object Examples [Электронный ресурс] // OpenGL Wiki: [сайт]. Электрон. текстовые дан. [2013]. URL: [https://www.opengl.org/wiki/Framebuffer\\_Object](https://www.opengl.org/wiki/Framebuffer_Object) (дата обращения 28.05.2016).

## МЕТОДИКА РАСЧЕТА ДЛИНЫ РЕГЕНЕРАЦИОННОГО УЧАСТКА ВОЛС С КОМПЕНСАТОРАМИ ДИСПЕРСИИ

**М.С. Былина, С.С. Попова**

*Хроматическая дисперсия является одним из основных факторов, ограничивающих длину регенерационного участка. В докладе рассмотрены методы компенсации хроматической дисперсии и предложены методики расчета длины регенерационного участка ВОЛС с компенсаторами дисперсии на основе специального волокна DCF и волоконной брэгговской решетки с переменным периодом.*

*Ключевые слова: длина регенерационного участка, хроматическая дисперсия, компенсация дисперсии, волокно для компенсации дисперсии, волоконная брэгговская решетка.*

## METHODS OF CALCULATING THE LENGTH OF REGENERATION AREA OFTL WITH DISPERSION COMPENSATORS

**Bylina M., Popova S.**

*Chromatic dispersion is one of the main factors limiting the length of the regeneration area. The report examines the methods of compensation of chromatic dispersion and the proposed method of calculating the length of the regeneration area OFTL a dispersion compensator based on a special DCF fiber and fiber Bragg grating with a variable period.*

*Keywords: regenerator section length, chromatic dispersion, dispersion compensation fiber for dispersion compensation fiber Bragg grating.*