

## СЖАТИЕ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ С ИСПОЛЬЗОВАНИЕМ ТЕМАТИЧЕСКИХ ТЕКСТУР

Вяткин С.И.

Институт автоматики и электрометрии СО РАН, Новосибирск, Россия

Костюкова Н.С.

Донецкий национальный технический университет, Донецк,

Украина

Романюк О.В.

Винницкий национальный технический университет, Винница,

Украина

В системах визуализации реального времени существует задача отображения больших районов местности. Возникает проблема, как отобразить район местности с минимальным расходом ресурсов системы визуализации (в основном памяти) на хранение и обработку текстуры. В данной работе предлагается метод сжатия графической информации с применением тематических текстур.

### **Введение**

Компьютерная графика реального времени, ориентированная на визуализацию трехмерных сцен, достигла на сегодняшний день существенных успехов. Она находит широкое применение от сложных систем визуализации для тренажерных комплексов (авиационных, космических, морских, автомобильных и т.п.) до графических акселераторов, используемых в компьютерных играх. И хотя достигнут достаточно высокий реализм отображаемых сцен в реальном времени, существует ряд задач, например, отображение больших районов местности, где требуется хранить и отображать сцены, содержащие большое количество графической информации.

Повышение реализма достигается уже не только за счет количества примитивов, которыми представлена сцена, а за счет моделирования специальных эффектов окружающей среды и мелких деталей поверхности объектов сцены. Алгоритмы построения изображения, применяемые в современных графических акселераторах, не решают эту задачу в полном объеме.

Для реалистичности изображаемой местности требуется текстура большого размера. Обычно для этого используется

фотография той реальной местности, которая предполагается использоваться в системе визуализации.

Поэтому необходимо сжатие не только геометрических данных, т.е. примитивов, но и текстуры [1]. В настоящее время в компьютерной графике используется несколько способов представления геометрических объектов, каждый из которых, в силу своих свойств, используется в различных областях: от систем 3D - моделирования и CAD -систем до систем визуализации реального времени. Полигональное задание является кусочно-линейной аппроксимацией некой поверхности. Достоинства такого задания заключаются в простоте представления и универсальности (кусочно-линейное представление существует для любой поверхности). Недостатки - это большие размеры баз данных, необходимые для хранения информации о геометрии поверхности. К настоящему времени разработано довольно много алгоритмов, предназначенных для построения адаптивной триангуляции рельефа путем использования иерархических структур данных (бинарных деревьев, квадродеревьев) [2-4]. Основное назначение таких алгоритмов – сократить число выводимых треугольников без существенной потери качества изображения. Однако для сложных поверхностей, каким является, к примеру, горный рельеф местности, требуется огромное число треугольников. Функциональное представление [5] наиболее точно из всех существующих описывает геометрию объекта и имеет наименьший размер данных, необходимых для описания геометрии объекта. В работе [6] приведены результаты исследований по моделированию и визуализации сложных поверхностей с применением скалярных функций возмущения.

В данной работе предлагается метод сжатия текстурных данных больших районов местности.

### **Постановка задачи**

Перечислим несколько условий, необходимых для того, чтобы получить сжатие корректным образом.

Во-первых, предположим, что изображение местности состоит из однородных областей, каждая из которых может быть представлена своей собственной текстурой. Например, лес может быть представлен текстурой “лес”, поле – текстурой “поле”, и т.д. Тогда результирующее изображение можно представить в виде тематической текстуры, состоящей из массива индексов, являющихся определённым номером темы [1]. Каждая тема – это небольшая однородная текстура (текстура леса, поля, водной поверхности и т.д.),

удовлетворяющая условию сшивки – левая граница совпадает с правой, верхняя – с нижней.

Во-вторых, в задаче отображения местности нам не важно точное совпадение исходной фототекстуры и сжатой тематической текстуры. Сжатие исходного изображения происходит за счёт его преобразования в тематическую текстуру с потерей информации (аналогия - JPEG).

Так как текстура должна находиться в постоянно сжатом состоянии на этапе непосредственного применения в системе визуализации, стандартные методы сжатия неприменимы, потому что требуют дополнительных вычислений при распаковке и дополнительную память системы.

### **Описание метода**

В предлагаемом методе цвет изображения в определённой точке вычисляется в процессе визуализации. Результирующее сжатие осуществляется следующим способом. Как было выше сказано, тематическая текстура представляет набор индексов. Обычная текстура состоит из текселов, хранящих значение цвета в каком-либо формате (индекс в палитре цветов или полное RGB-значение). Индексы тематической текстуры представляют собой сетку. Каждая ячейка этой сетки, образуемая четырьмя соседними индексами, кодирует  $N \times M$  текселов, где  $N$  и  $M$  параметры, задающие размер ячейки. Поэтому, результирующее сжатие пропорционально произведению  $M$  на  $N$ . Чем больше эти параметры, тем больше степень сжатия, что в свою очередь, приводит к снижению детализации изображения тематической текстуры. Поэтому эти параметры подбираются вручную так, чтобы получить правильный визуальный эффект.

При отображении тематической текстуры, каждая ячейка обрабатывается отдельно. Метод вычисления состоит из нескольких шагов:

1. Проверяем, если все четыре соседних индекса рассматриваемой ячейки тематической текстуры совпадают, то изображение данного участка образуется одной темой.
2. Иначе, вычисляем цвет каждой точки данного участка изображения путём замешивания цветов тем, определяемых индексами текущей ячейки тематической текстуры.

Замешивание в п.2 необходимо для того, чтобы обеспечить плавность перехода на границе между различными темами. Результирующий цвет получается равным взвешенной сумме цветов трёх соседних тем, где вес определяется расстоянием между текущей точкой ячейки до вершины ячейки. Для этого прямоугольная ячейка разбивается на два треугольника (рис 1.).

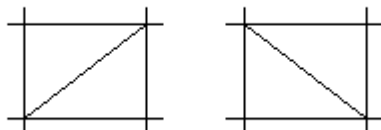


Рисунок 1 - Способы разбиения ячейки на треугольники

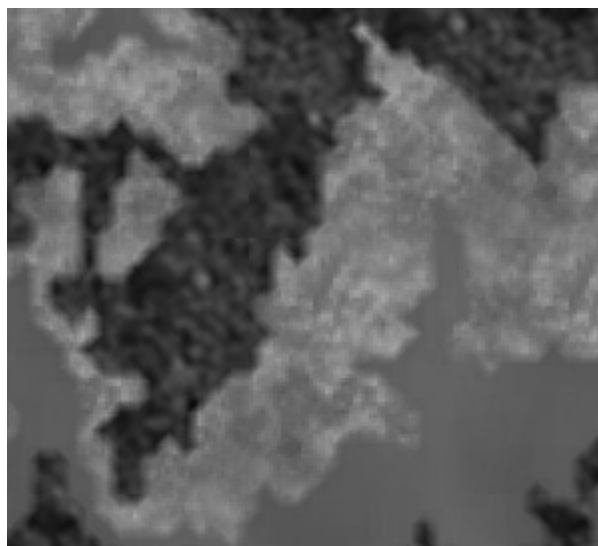


Рисунок 2 - Пример тематической текстуры

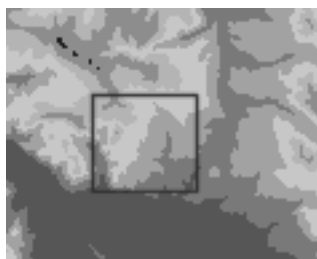


Рисунок 3 - Участок изображения карты индексов

Выбор способа разбиения ячейки необходим для того, чтобы не возникали дефекты при отображении. В данном методе предлагается производить разбиение по диагонали ячейки, концы которой

соответствуют одинаковым индексам. Если все индексы различны, то разбиваем произвольным способом. На рис. 2 изображён участок тематической текстуры. При её построении в качестве массива индексов использовалось обычное изображение (рис. 3), предварительно обработанное в редакторе. Выделенная прямоугольником область, изображенная на рис. 3, соответствует изображению тематической текстуры на рис. 2. Каждому индексу соответствует свой цвет, определённый пользователем.

### Заключение

Разработан метод создания и отображения тематической текстуры, позволяющий значительно снизить затраты на память системы визуализации. Что касается дальнейших планов, то предполагается разработать способы интерактивного взаимодействия с построенной сценой. В частности, предполагается разработка алгоритмов изменения текстуры пользователем во время растрирования сцены. То есть необходимо создать некий удобный аппарат визуального воздействия на сцену.

### Литература

1. Sergei I. Vyatkin, Boris S. Dolgovesov, Valerie V. Ovechkin, Sergei E. Chizhik, Nail R. Kaipov, Photorealistic imaging of digital terrains, freeforms and thematic textures in realtime visualization system Voxel Volumes, GraphiCon '97, Moscow. - p.121–126.
2. Markus H. Gross, Roger Gatti, and Oliver G. Staadt. *Fast multiresolution surface meshing*. In Proc. of 14th International Conf. on Data Engineering, ICDE'98, IEEE, 1998. - p.550–557.
3. Gross M.H., Staadt O.G., Gatti R. *Efficient Triangular Surface Approximations Using Wavelets and Quadtree Data Structures*. IEEE Trans. on Visualization and Computer Graphics. Vol. 2, N 2, June 1996. - p.130-143.
4. Renato Pajarola. *QuadTIN: Quadtree based Triangulated Irregular Networks*. In Proceedings IEEE Visualization 2002, IEEE Computer Society Press, 2002. - p.395–402.
5. Вяткин С.И., Долговесов Б.С., Есин А.В. и др. Геометрическое моделирование и визуализация функционально-заданных объектов // Автометрия. 1999. №6. С.65-68.
6. Вяткин С.И. Моделирование и визуализация сложных поверхностей на основе скалярных функций возмущения // Сборник трудов ДонНТУ серии "Информатика, кибернетика и вычислительная техника". - 2009. - Вып. 10 (153). С. 105-111.

Получено 26.05.09