

Навигация в реальном времени над моделью местности неограниченного размера

Е.В. Тетенов

Югорский Научно-Исследовательский Институт Информационных Технологий,
Ханты-Мансийск, Россия

evtet@mail.ru

Аннотация

В статье описывается технология визуализации рельефа местности в реальном времени без ограничений на объём информации. В основе лежит иерархическое представление модели в виде дерева блоков данных и алгоритм симплификации, осуществляющий корректную зависимость от положения наблюдателя непрерывную детализацию на лету. Независимая обработка блоков позволяет хранить в памяти только активные данные с необходимым разрешением, а подкачку данных производить параллельно с рендерингом на основе клиент-сервер технологии, многопоточности, оптимального кэширования и асинхронного обмена. Разработанная технология позволяет осуществлять имитацию полёта над территорией и визуализацию текстурированного рельефа земной поверхности с высоким разрешением в реальном времени на данных практически неограниченного объема.

Ключевые слова: Интерактивная визуализация, неограниченный рельеф, симплификация.

1. ВВЕДЕНИЕ

Основная проблема интерактивной визуализации земной поверхности связана с необходимостью генерировать видеоизображение с достаточно высокой частотой кадров при огромных объёмах исходных данных (модель рельефа, текстуры, техногенные объекты), представляющих большие территории с высоким пространственным разрешением.

Методы непосредственной визуализации исходной информации здесь неприменимы, так как ни графическая аппаратура, ни память компьютера не в состоянии переварить такие объёмы данных с требуемой скоростью. С другой стороны, в каждый момент времени наблюдатель может обозревать лишь малую часть всей базы данных при довольно низком разрешении, определяемым разрешением дисплея. Поэтому различные методы, связанные с пространственным отбором данных, динамической подкачкой и симплификацией (упрощением детализации) обычно используются для понижения сложности визуализации и уменьшения объема одновременно обрабатываемых данных.

Особенность здесь, в отличие от двумерных ГИС, состоит в пространственной неоднородности симплификации - степень детализации элемента поверхности напрямую зависит от расстояния до наблюдателя вследствие перспективы: изображения дальних частей поверхности имеют меньший размер и требуют меньшей точности, чем ближних. В этой ситуации применяются адаптивные методы зависимой от положения

непрерывной детализации на основе выбранного критерия – допустимой меры погрешности.

Существующие алгоритмы, которые могут интерактивно осуществлять зависимую от наблюдателя локально-адаптивную генерацию сетки, строятся на базе регулярных [1,2,3] или нерегулярных [4] данных. Первые генерируют сетку путём рекурсивного разбиения исходной матрицы, а вторые применяют триангуляцию Делоне. Для рассматриваемых здесь проблем первые методы предпочтительнее, так как исходные данные обычно представляют собой регулярные сетки – матрицы высот и алгоритмы для них значительно проще и быстрее.

2. МОДЕЛЬ ПОВЕРХНОСТИ И СТРУКТУРЫ ДАННЫХ

Полагаем, что исходные высотные данные представлены в виде одного или нескольких смежных кластеров размером $(2^n+1) \times (2^n+1)$ узлов. Тогда глобальная модель земной поверхности представляется в виде одного или нескольких квадродеревьев, в которых корень дерева представляет поверхность для всего кластера с самым низким пространственным разрешением, а каждый следующий потомок представляет одну четвертую часть поверхности своего родителя с вдвое большим разрешением (Рис. 1).

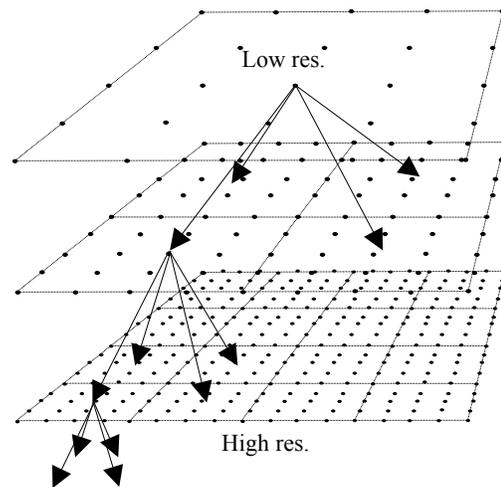


Рис. 1.

Высшая степень детализации, соответствующая разрешению исходной матрицы, принадлежит листьям дерева. Такая структура применяется для квадратных блоков высотных данных и текстур и создаётся на этапе подготовки данных.

Использование блочно-иерархического представления диктуется прежде всего необходимостью визуализации неограниченно больших участков земной поверхности и

оперативной навигации по базе данных. Квадро-блоки используются для выбора нужного уровня детализации в зависимости от расстояния до наблюдателя и для отсека невидимых данных конусом обзора (Рис.2), а также для оперативной подкачки, когда данные расположены вне ЦПУ.

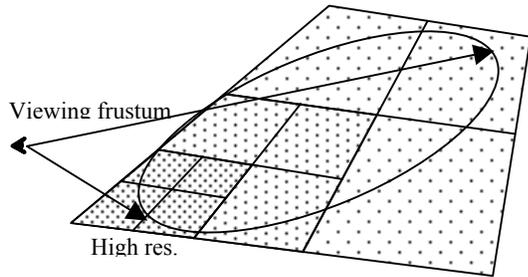


Рис. 2.

Для зданий, сооружений и других техногенных объектов применяется согласованная с описанными деревьями структура в виде R-дерева [5].

Триангуляция высотных данных внутри каждого блока осуществляется непосредственно в процессе подготовки каждого кадра при помощи рекурсивного разбиения матрицы. Последовательность разбиений блока размером $(2^k+1) \times (2^k+1)$ узлов прямоугольными треугольниками образует бинарное дерево из $2k+1$ уровней детализации по 2^{l+1} треугольников на каждом уровне l ($0 \leq l \leq 2k$), поскольку каждый треугольник уровня l порождает два потомка уровня $l+1$ (Рис.3).

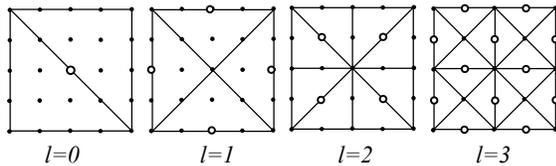


Рис. 3.

Для построения оптимальной конфигурации треугольников с нужным уровнем детализации применяется нисходящий алгоритм на основе явного бинарного дерева [6]. Алгоритм представляет собой рекурсивную процедуру деления треугольника на два и применения процедуры к двум полученным (Рис.4). Такой процесс начинается с корневого уровня и продолжается до прекращения выполнения заданного условия (критерия разбиения) или достижения последнего уровня.

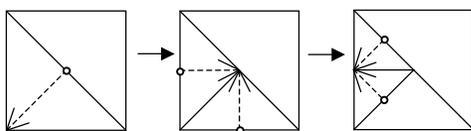


Рис. 4.

Узлы, лежащие на серединах гипотенуз треугольников разных уровней детализации через которые производится разбиение и показанные кружками на Рис.3-4, называются опорными для соответствующих треугольников. Опорные узлы взаимно-однозначно соответствуют общим гипотенузам пар смежных треугольников уровня l и образуют перебегающее квадро-дерево - каждый опорный узел уровня l - v_l имеет 4-х потомков $\{v_{l+1}^{--}, v_{l+1}^{+-}, v_{l+1}^{+}, v_{l+1}^{++}\}$ и двух предков (Рис.5).

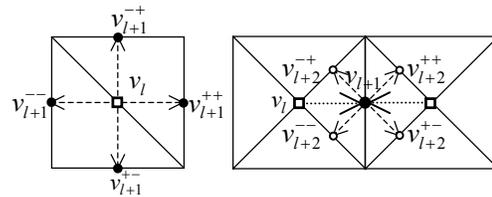


Рис. 5. Связи опорных узлов уровней l , $l+1$ и $l+2$.

3. КРИТЕРИЙ СИМПЛИФИКАЦИИ И ФУНКЦИЯ РАЗБИЕНИЯ

В качестве условия деления выбирается геометрический критерий, сопоставляющий принятую метрику (характерный размер), например, диагональ треугольника d или величину отклонения δ по высоте (z - координате) при замене искривленного сегмента DEBC плоским DBC, с определенным пороговым значением τ (Рис.6). Значение δ вычисляется в центральном узле E по формуле

$$\delta = \max \{ |E_z - (A_z + C_z)/2|, |E_z - (B_z + D_z)/2| \} \quad (1)$$

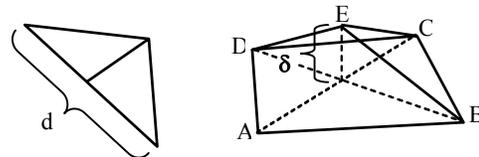


Рис. 6.

Последний критерий является частным случаем общего критерия симплификации поверхности по величине кривизны: разбиение плоского сегмента производится, если отклонение от плоскости превышает порог. Однако для полной оценки кривизны сегмента нужно проанализировать величины δ для всех потомков до последнего уровня, что сводит на нет преимущества нисходящего алгоритма. Выход заключается во введении некоторой заранее вычисляемой функции разбиения e , дающей интегральную оценку кривизны сегмента. Другая причина для введения такой функции это проблема геометрического согласования триангулированных блоков с различной степенью детализации. Задача заключается в построении связанного разбиения, когда соседние треугольники контактируют сторона к стороне и вершина к вершине и никакая вершина треугольника не лежит на стороне другого (Рис.7а).

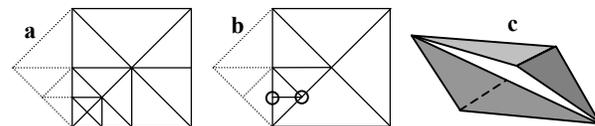


Рис. 7.

Иначе возникает ситуация, показанная на Рис.7b (кружками отмечены несвязные вершины), и между соседними треугольниками возникают разрывы (Рис.7c). Для устранения этих проблем используются достаточно сложные алгоритмы коррекции сетки (слияния - деления) и дополнительные

структуры данных [1,2], которые существенным образом усложняют и, как следствие, замедляют или делают невозможным процесс визуализации в реальном времени. Для получения связного разбиения без дополнительной перестройки, необходимо корректное определение функции разбиения.

Из анализа свойств разбиения видно, что критерий деления должен быть общим для пары сопряженных треугольников, значит, функция e должна быть определена в опорных узлах. Далее ясно, что если критерий выполняется для данного треугольника, то он должен выполняться для всех его предков. Отсюда следует утверждение:

Пусть функция разбиения e определена для всех опорных узлов исходной сетки и условие деления для нисходящего алгоритма задано в виде:

$$e \geq \tau, \quad (2)$$

где пороговое значение τ - константа, тогда для непрерывности (связности) разбиения функция e_{ij} должна быть монотонной относительно уровней квадродерева:

$$e_l \geq \max \{e_{l+1}^-, e_{l+1}^+, e_{l+1}^{++}\}. \quad (3)$$

Последнее соотношение получено без учета перспективной коррекции, в то время как реальный интерес представляет видимый размер отклонения в единицах дисплея и пороговое значение τ определяется в экранных пикселях.

В силу свойств перспективного преобразования, соотношение между линейным размером в мировых координатах d_w и верхней границей соответствующих проекций на экране d_s не зависимо от ориентации имеет следующий простой вид:

$$d_s = d_w c / z, \quad (4)$$

для $d_w \ll z$, где $z > 0$ - глубина в координатах камеры, c - коэффициент. Соотношения, полученные в пространстве модели, должны быть преобразованы в экранные координаты с учетом (4). Тогда критерий (2) примет вид:

$$e/z \geq \tau_s, \quad (5)$$

где τ_s - пороговая величина в экранных координатах, а в условии (3) e заменится на e/z . Однако z - переменная и зависит от положения наблюдателя. Чтобы избавиться от этой зависимости, рассмотрим пару опорных узлов v_1 и v_2 (уровни l и $l+1$) в пространстве наблюдателя (Рис.8).

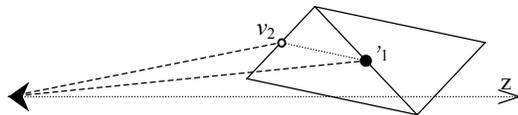


Рис. 8.

Условие монотонности (3) для этой пары запишется в виде: $e_1/z_1 \geq e_2/z_2$ или $e_1 \geq e_2(z_1/z_2)$. Так как $|z_1 - z_2| < d_2/2$, где d_2 - размер диагонали соответствующей узлу v_2 , то последнее соотношение мажорируется неравенством $e_1 \geq e_2(1 + d_2/2z_2)$. Это соотношение можно сделать независимым от z , если ввести ограничение на размер диагонали треугольника:

$$d/z < \varepsilon, \quad (6)$$

где ε - константа, откуда следует:

$$e_l \geq e_{l+1}(1 + \varepsilon/2). \quad (7)$$

Так как величина d/z приблизительно равна отношению видимого размера треугольника к размеру окна на экране, то ограничение (6) не является жестким. Приемлемое значение ε порядка $1/10$ дает величину $(1 + \varepsilon/2) = 1.05$ в (7) мало отличающуюся от 1, однако играющую существенную роль при триангуляции. Суммируя (5) и (6) получим комбинированный критерий разбиения в виде:

$$\max \{d/\varepsilon, e/\tau_s\} \geq z, \quad (8)$$

а условие монотонности для функции e :

$$e_l \geq \max \{qe_{l+1}^-, qe_{l+1}^+, qe_{l+1}^{++}\}, \quad q = 1 + \varepsilon/2 \quad (9)$$

Корректное значение функции разбиения в опорном узле v_l , основанной на критерии (1) и удовлетворяющей условию монотонности (9), можно определить рекуррентной формулой:

$$e_l = \max \{\delta_l, qe_{l+1}^-, qe_{l+1}^+, qe_{l+1}^{++}\}, \quad q = 1 + \varepsilon/2, \quad (10)$$

дающей интегральную оценку кривизны сегмента.

Значения e вычисляются для всех опорных узлов базовой сетки каждого уровня l снизу-вверх, рекурсивно по формуле (10) на этапе подготовки данных и используются совместно с исходными высотными значениями. Адаптивная триангуляция каждого блока производится по нисходящему алгоритму с обобщенным критерием (8).

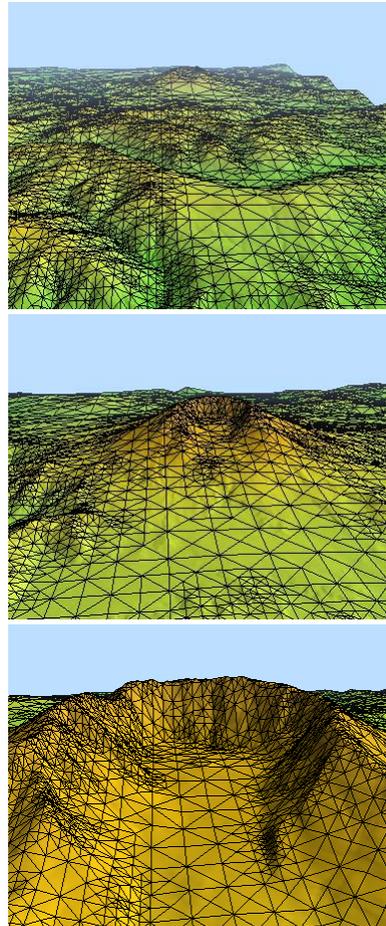


Рис. 9.

Пример адаптивной триангуляции рельефа показан на Рис.9 - вид вулкана Helens (Washington, USA) с расстояния 36, 12 и 4 км, сгенерированный разработанной программой. Данные (10-метровый DEM ~ 1 500 000 000 узлов) взяты с сайта <http://duff.geology.washington.edu/data/raster/tenmeter/bil10/>.

4. ОРГАНИЗАЦИЯ ДАННЫХ И КОНВЕЙЕРА

Исходные данные – матрицы высот и текстуры преобразуются к виду, приспособленному для построения согласованных структур, описанных в п.1, то есть к размерам $(m1(2^n+1)) \times (m2(2^n+1))$ для высотных и $(m1(2^k+1)) \times (m2(2^k+1))$ для текстурных данных. В качестве текстур используются как фото-текстуры, полученные при помощи аэро- и спутниковой съёмки, так и синтетические - сгенерированные по высотным данным, включая информацию ГИС. Затем создаются базы данных, где располагаются структурированная информация.

Возможны различные варианты размещения построенных баз в зависимости от размера и характера доступа.

- Данные небольшого размера (например, до 10ГБ) могут располагаться на диске персонального компьютера и обеспечивают локальный доступ.
- Данные большого объема имеет смысл разместить на файл-сервере для корпоративного использования в режиме сетевого мультидоступа.
- Более продвинутый вариант предполагает использование СУБД (типа ORACLE) для размещения данных.

Эффективная подкачка (обмен) данных в реальном времени реализована с использованием многопоточности, клиент-сервер технологии, оптимального кэширования и асинхронного обмена с диском в случае локального набора данных и через IP Socket для удаленного сервера. Основная идея состоит в том, чтобы хранить на клиенте только актуальные (видимые) данные, объем которых невелик, а обновление данных при навигации производить параллельно с рендерингом.

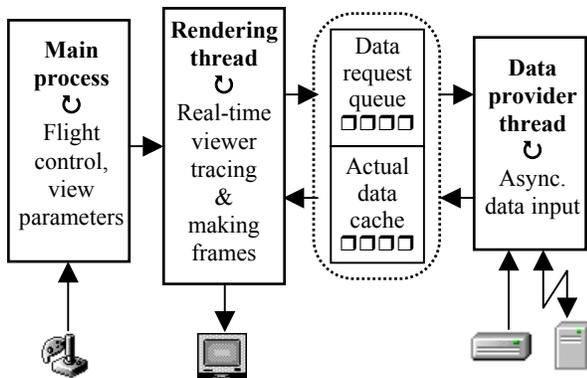


Рис. 10.

Многопоточная модель алгоритма показана на Рис.10. В основном (Main) потоке осуществляется управление полетом (движением камеры) и параметрами отображения. Второй поток предназначен исключительно для рендеринга, его задача отслеживать перемещение камеры во времени, посылать запросы провайдеру на обновление данных и генерировать кадры с постоянной частотой. Провайдер

данных в третьем потоке обслуживает очередь запросов и возвращает блоки данных в асинхронном режиме. Для локальных данных это асинхронный обмен с диском, а в сетевом варианте – обмен по сети через асинхронный Win Socket.

В последнем случае на выделенном компьютере размещается серверный модуль – Application Server, реализованный как Win Service и обслуживающий запросы клиентов (Рис.11). Для каждого подключенного клиента выделяется свой провайдер, точно такой - же, как в локальном варианте только возврат данных осуществляется через собственный Win Socket.

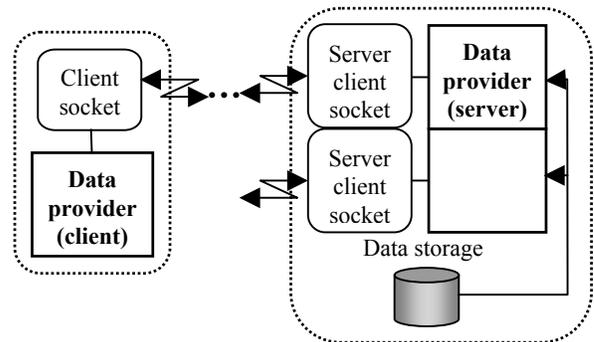


Рис. 11.

При использовании СУБД применяется "three-tiered" клиент-сервер модель. Также как в предыдущем случае присутствует Application Server, но провайдер вместо чтения с диска получает блоки данных из СУБД посредством асинхронных SQL-запросов.

При тестировании сервера не наблюдалось каких-либо задержек при одновременной интенсивной работе четырёх клиентов.

5. ЗАКЛЮЧЕНИЕ

Описанная технология была реализована в виде законченных программ с развитым пользовательским интерфейсом в локальном и сетевом вариантах. Кроме того, реализован стерео - режим и визуализация строений. Создан также инструментарий для построения необходимых структур данных из различных источников, включая высотные данные, фотографии местности и информацию ГИС. За пять лет эксплуатации системы реализовано около 20 проектов с объёмами баз до 10ГБ. Система устойчиво функционирует на персональных компьютерах без каких-либо требований к ресурсам, при наличии у них графической карты с аппаратной поддержкой OpenGL.

6. БИБЛИОГРАФИЯ

- [1] P. Lindstrom, D. Koller, W. Ribarsky, L.F. Hodges, N. Faust, and G.A. Turner, Real-time, continuous level of detail rendering of height fields. In *SIGGRAPH '96 Proc.*, Aug. 1996, pp. 109-118.
- [2] M. Duchaineau, M. Wolinsky, D.E. Sigeti, M.C. Miller, C. Aldrich, M.B. Mineev-Weinstein, ROAMing terrain: Real-time optimally adapting meshes, In *Visualization '97 Proc.*, 1997, pp. 81-88.

[3] S. Röttger, W. Heidrich, P. Slusallec and H. Seidel, Real-time generation of continuous levels of detail for height fields, *6th Intl. Conf. in Central Europe on Computer Graphics and Visualization '98*, Feb. 1998.

[4] H. Hope, Smooth view-dependent level-of-detail control and its application to terrain rendering, *IEEE Visualization '98*, Feb. 1998.

[5] A. Guttman, R-trees: A Dynamic Index Structure for Spatial Searching. Proceedings of the 1984 ACM SIGMOD Int'l Conf on Mgmt of Data, 45-57

[6] S. McNally, Binary Triangle Trees and Terrain Tessellation, *Longbow Digital Arts web page*, <http://www.longbowdigitalarts.com/seumas/progbintri.html>

Об авторе

Тетенев Евгений Викторович – к. ф.-м. н., в. н. с. Югорского Научно-Исследовательского Института Информационных Технологий.

Адрес: 628011, г. Ханты-Мансийск, ул. Мира, 151.

Телефон: 8 (34671) 59054

E-mail: evtet@mail.ru

Real-time navigation over unlimited terrain

Abstract

The paper describes technology for real-time terrain visualization without any limitation on data size. In the basis lies hierarchical representation of terrain model as tree of data blocks and algorithm for view-dependent continuous detail simplification on the fly. Independent block processing allow keep in memory only active data of required resolution and perform parallel data swapping using client-server technology, multithreading, optimal caching and asynchronous data exchange. Developed technology allows real-time fly trough simulation over practically unlimited high resolution textured terrain data.

Keywords: *Real-time visualization, unlimited terrain, simplification.*

About the author(s)

Evgueny Tetenov is a leading scientist at Ugra Research Institute of Information Technologies. His contact email is evtet@mail.ru.