

УДК 004.627

ЭФФЕКТИВНОЕ КОДИРОВАНИЕ АДАПТИВНОЙ ТРИАНГУЛЯЦИИ РЕЛЬЕФА В КОНТЕКСТЕ ИЕРАРХИЧЕСКОГО ВЕЙВЛЕТ-СЖАТИЯ СЕТКИ ВЫСОТ

© 2010 г.

Е.А. Юсов, В.Е. Турлапов

Нижегородский госуниверситет им. Н.И. Лобачевского

yusov_egor@mail.ru

Поступила в редакцию 16.07.2010

Представлена новая многомасштабная модель рельефа местности, которая реализует единый метод кодирования сетки высот и адаптивной триангуляции. Структура нуль-деревьев квантованных вейвлет-коэффициентов, полученных при сжатии сетки высот, используется для построения триангуляции, которая почти везде удовлетворяет заданному допуску. Чтобы гарантировать допуск, кодируется остаточная информация, которая, где необходимо, уточняет триангуляцию. Приведены результаты экспериментов и сравнение с другими методами, подтверждающие эффективность представленных алгоритмов.

Ключевые слова: рельеф, вейвлет-преобразование, сжатие, нуль-дерево, адаптивная триангуляция.

Введение

Интенсивное развитие цифровых технологий в последние десятилетия привело к чрезвычайно быстрому росту объема данных, которые требуется обрабатывать и хранить. Как следствие, разработка эффективных методов сжатия, позволяющих значительно снизить расход памяти при контролируемой точности восстановления, является одной из важнейших задач современной информатики. Такие методы, в частности, необходимы для решения проблемы хранения информации о рельефе местности (карт высот и фотоснимков поверхности), которая особенно актуальна для таких приложений, как геоинформационные системы, авиакосмические симуляторы, среды виртуальной реальности, компьютерные игры и т.п. Карта высот, полученная путем спутникового сканирования местности, может содержать миллиарды отсчетов и занимать в исходном виде терабайты информации. Ее размер может превысить объем не только доступной оперативной памяти, но и дискового накопителя. Фотоснимки местности могут требовать для хранения еще большего пространства. Обработка таких больших массивов данных является весьма непростой задачей, особенно в контексте визуализации больших открытых территорий в реальном времени, т.к. связана с необходимостью использовать сложные алгоритмы динамической подкачки данных. Возникает и другая проблема: вычислительная сложность исходной равномерно триангулированной модели рельефа оказывается

слишком высокой для вывода даже на самых мощных современных графических ускорителях. Поэтому задача построения и хранения адаптивной триангуляции, отвечающей особенностям поверхности, также является актуальной.

1. Обзор работ

Построению адаптивной триангуляции ландшафта посвящено множество работ, начиная с алгоритмов, представленных П. Линдстромом (P. Lindstrom) и др. [1] и М. Дюшейн (M. Duchaineau) и др. [2]. Подробный обзор различных многомасштабных моделей для визуализации рельефа в реальном времени представлен в работах [3, 4]. В последние годы среди исследований в области моделирования рельефа стало появляться все больше работ, посвященных методам сжатия, подходящим для систем реального времени. Одной из первых подобных публикаций является [5]. В ней представлен метод геометрических карт отсечений (geometry clipmaps), в котором для кодирования сетки высот применяется техника сжатия изображений с потерями, описанная в [6]. Триангуляция, порождаяемая методом, является регулярной и зависит только от положения камеры в пространстве. Благодаря этому, с одной стороны, удастся достичь высокой эффективности использования GPU. С другой стороны, из-за того что триангуляция не адаптирована к локальным особенностям поверхности, она оказыва-

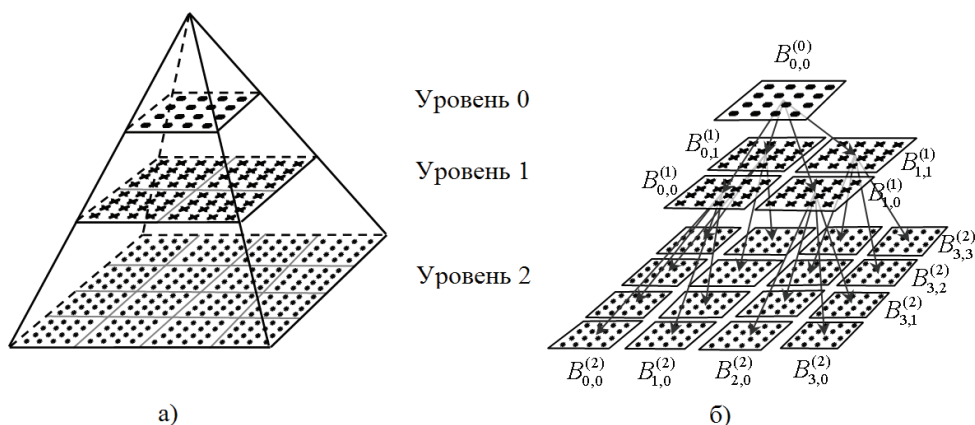


Рис. 1. а) Фильтрованная пирамида; б) квадродерево блоков рельефа

ется чрезмерно избыточной. Более существенным недостатком метода является то, что он не может гарантировать заданную точность аппроксимации, что особенно заметно на сильно гористом рельефе. Другой метод, в котором применяется сжатие сетки высот, называется C-BDAM и представлен в работе [7]. Метод использует двухэтапный алгоритм сжатия изображений с помощью вейвлет-преобразования [8], позволяющий гарантировать заданную точность восстановления. В методе C-BDAM для каждого фрагмента упрощенной модели используются регулярные триангуляции, не адаптированные к особенностям поверхности, что увеличивает нагрузку на GPU. Другой алгоритм, представленный в работе [9], предлагает метод кодирования адаптивной триангуляции и позволяет добиться умеренной степени сжатия в 8–9 раз. Вопрос сжатия сетки высот в [9] не рассматривается.

В данной статье представлена новая многомасштабная модель рельефа местности, которая позволяет эффективно кодировать сетку высот и адаптивную триангуляцию поверхности, используя единый подход. В основе представленного метода лежит идея, что вейвлет-коэффициенты, полученные при сжатии сетки высот, могут быть также использованы и для построения адаптивной триангуляции. Близкие подходы были реализованы в методах М. Гросса (M. Gross) и др. [10], а также А. Переберина [11]. В этих двух методах вейвлет-коэффициенты используются только для построения адаптивной триангуляции, а сжатие не применяется. Что еще более важно, описанные методы не могут обеспечить заданную точность восстановления. В качестве решения данной проблемы предлагается двухэтапный метод кодирования адаптивной триангуляции, при котором вейвлет-коэффициенты образуют базовую ин-

формацию, которая дополняется остаточной информацией, гарантирующей, что адаптивная триангуляция аппроксимирует поверхность с заданной погрешностью.

2. Описание предлагаемого многомасштабного представления сетки высот рельефа и адаптивной триангуляции

2.1. Иерархическое сжатие сетки высот.

На первом этапе построения сжатого многомасштабного представления исходная сетка высот последовательно фильтруется и прореживается, в результате чего получается многоуровневая пирамида (рис. 1а), каждый следующий уровень которой представляет собой сетку с возрастающим в два раза шагом дискретизации по каждому направлению, аппроксимирующую исходную с уменьшающейся точностью. Такая же структура данных используется в методе геометрических карт отсечений [5] и наших предыдущих подходах [12, 13]. Для построения пирамиды мы использовали нормализованный низкочастотный вейвлет-фильтр анализа Коэна–Добеши–Фово 9/7, который, как показали наши эксперименты, является более выгодным, чем фильтр Невилля, примененный нами в [12, 13]. Важно отметить, что поскольку используемый вейвлет-фильтр имеет нечетную длину, положения отсчетов в более грубых уровнях пирамиды не смещаются.

Каждый уровень пирамиды разбивается на квадратные блоки, имеющие одинаковое количество элементов (64×64 , 128×128 , 256×256 и т.д.), которые объединяются в иерархическую структуру квадродерева (рис. 1б). В дальнейшем будем через $B_{m,n}^{(k)}$ обозначать блок, расположенный в уровне k дерева в узле (m, n) .

Построенная иерархия сжимается нисходящим рекурсивным алгоритмом, начиная от

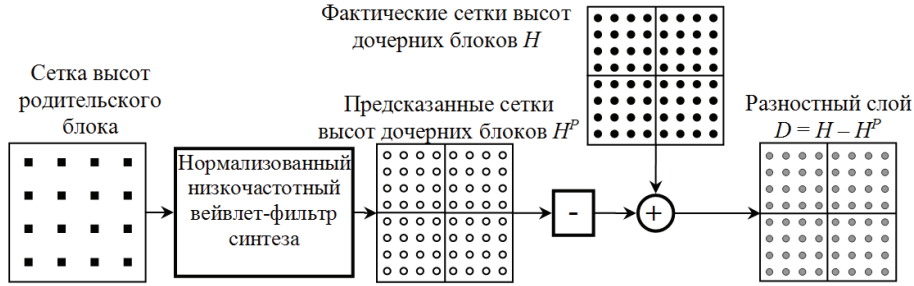


Рис. 2. Вычисление разностного слоя блока



Рис. 3. Отклонение линейно интерполированной сетки высот родителя от сеток высот его потомков

блоков на самом грубом уровне приближения (уровень 0). При этом для каждого блока кодируется информация, которая позволяет уточнить его сетку высот и получить сетки высот 4 дочерних блоков. С этой целью алгоритм для каждого блока вычисляет разность между предсказанными сетками высот его четырех потомков ($H^P = \{h_{i,j}^P\}^{2n \times 2n}$) и их фактическими сетками высот ($H = \{h_{i,j}\}^{2n \times 2n}$) (см. рис. 2), которая образует разностный слой D : $D = H - H^P$. Предсказанные сетки высот H^P вычисляются путем применения нормализованного низкочастотного вейвлет-фильтра синтеза Коэна–Добеши–Фово 9/7 к сетке высот родительского блока. Для блока на самом грубом разрешении ($B_{0,0}^{(0)}$) предсказанная сетка высот по определению нулевая.

Чтобы избежать накопления ошибки от грубых к более детальным уровням, в качестве сетки высот родительского блока используется сетка, восстановленная из сжатого представления, а не фактическая сетка высот.

Разностный слой D каждого блока сжимается при помощи двухэтапного метода кодирования, описанного в [14]. На первом этапе разностный слой подвергается вейвлет-декомпозиции, полученные вейвлет-коэффициенты квантуются и кодируются с помощью техники нулевых деревьев. На втором этапе выполняется рекон-

струкция слоя и кодируется остаточная информация, гарантирующая заданную точность восстановления сетки высот в метрике L^∞ (см. [14]). Вначале мы пробовали сжимать иерархию с экспоненциально возрастающей максимально допустимой ошибкой: блоки на самом детальном уровне сжимались с погрешностью δ , блоки на следующем, более грубом уровне – с погрешностью 2δ и т.д. Однако позднее было обнаружено, что более эффективным методом является сжатие каждого уровня пирамиды с одинаковым допуском δ . В последнем случае предсказание сетки высот дочерних блоков оказывается более точным и в результате разностный слой сжимается более компактно. Для сжатия всех уровней пирамиды, кроме самого детального, мы использовали следующий метод квантования остатков $r_{i,j}$, отличный от примененного в [14]:

$$\tilde{r}_{i,j} = \lfloor r_{i,j} / \delta \rfloor,$$

где через $\lfloor \cdot \rfloor$ обозначена операция округления аргумента до ближайшего целого.

Восстановление остатков выполняется согласно следующей формуле:

$$\hat{r}_{i,j} = \begin{cases} \text{sign}(r_{i,j}) \cdot (\tilde{r}_{i,j} + 0.5) \cdot \delta, & \tilde{r}_{i,j} \neq 0 \\ 0, & \tilde{r}_{i,j} = 0 \end{cases}.$$

Использование такого квантования также позволяет выполнять более точное предсказание и,

как результат, получать более компактное представление. Поскольку отсчеты сетки высот являются не целыми, а вещественными числами, остатки самого детального уровня иерархии квантуются в нашем методе по следующей формуле:

$$\tilde{r}_{i,j} = \lfloor (|r_{i,j}| + \delta) / (2\delta + p) \rfloor, \quad (1)$$

где p – точность представления отсчетов сетки высот.

2.2. Вычисление верхней оценки погрешности аппроксимации блока. Для построения адаптивной модели рельефа (раздел 2.8) необходимо для каждого блока $B_{m,n}^{(k)}$ иметь оценку погрешности $Err_{Appr}(B_{m,n}^{(k)})$, с которой он аппроксимирует участок местности. Такая оценка может быть вычислена рекурсивно по следующей формуле:

$$\begin{aligned} & Err_{Appr}(B_{m,n}^{(k)}) = \\ & = \max\{Err_{Appr}(B_{2m,2n}^{(k+1)}), Err_{Appr}(B_{2m+1,2n}^{(k+1)}), \\ & Err_{Appr}(B_{2m,2n+1}^{(k+1)}), Err_{Appr}(B_{2m+1,2n+1}^{(k+1)})\} + \\ & + Err_{Int}(B_{m,n}^{(k)}) + \delta, \end{aligned} \quad (2)$$

где $Err_{Int}(B_{m,n}^{(k)})$ – максимальное отклонение линейно интерполированной восстановленной из сжатого представления сетки высот родителя от восстановленных сеток высот его дочерних блоков (рис. 3). Линейная интерполяция применяется потому, что выводимая триангулированная поверхность является кусочно-билинейной.

Для блоков на максимальном разрешении k_{\max} ошибка аппроксимации принимается равной погрешности восстановления: $Err_{Appr}(B_{m,n}^{(k_{\max})}) = \delta$. Поскольку восстановленная сетка высот может отличаться от точной на величину, не превосходящую δ , формула (2) дает гарантированную верхнюю оценку погрешности аппроксимации блока $B_{m,n}^{(k)}$.

2.3. Обеспечение бесшовного соединения соседних блоков. В предыдущих многомасштабных моделях рельефа проблеме бесшовного соединения соседних блоков, как правило, уделялось недостаточное внимание. В нашем подходе соседние блоки имеют общие отсчеты высоты и каждый блок «отвечает» за бесшовное соединение со своим правым и верхним соседями. Метод сжатия, описанный в [14], гарантирует, что восстановленное значение каждого отсчета высоты будет находиться в пределах заданного отклонения от точного, однако нет гарантии, что общие граничные отсчеты соседних блоков будут восстановлены одинаково.

Чтобы обойти эту проблему, в нашем методе мы ввели специальную область вокруг внешней границы сетки высот четырех блоков, имеющих общего родителя (рис. 4). Мы называем эту область *согласующей*.

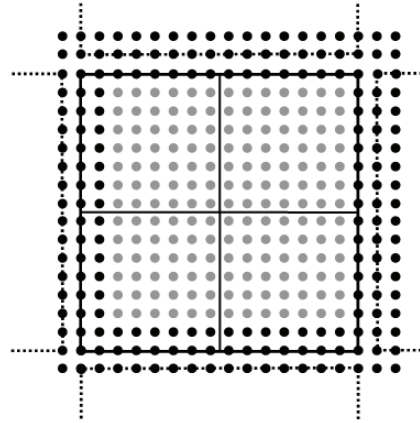


Рис. 4. Согласующая область

Согласующая область состоит из нескольких контуров (трех на рис. 4). Для обеспечения согласованности сеток высот достаточно одного контура, но для того чтобы элементы карты нормалей на границах были вычислены одинаково, требуется большее количество. Каждый контур сжимается без использования вейвлет-преобразования. Элементы контура рассматриваются как одномерная последовательность отсчетов высоты h_i , которые квантуются при помощи (1) и кодируются. При этом для каждого квантованного отсчета \tilde{h}_i (исключая \tilde{h}_0) вычисляется разность $d_i = \tilde{h}_i - \tilde{h}_{i-1}$, кодируемая при помощи арифметического кодера. Мы применяем простую контекстную модель, учитывающую значение предыдущей квантованной разницы d_{i-1} для увеличения степени сжатия. Знаки также арифметически кодируются с использованием зависимости от предыдущего знака.

2.4. Построение «негарантирующей» триангуляции на основе структуры тройных нуль-деревьев сжатого разностного слоя. Как было отмечено в разделе 2.1, каждый следующий уровень пирамиды (рис. 1а) является сеткой с увеличивающимся интервалом дискретизации, которая аппроксимирует исходную сетку высот с уменьшающейся точностью. Каждому блоку рельефа соответствует фрагмент этой пирамиды, который сам является многомасштабным представлением участка местности (будем далее называть его *локальным*), аппроксимируемого блоком (рис. 5а). Структура нуль-деревьев сжатого разностного слоя, полученная в про-

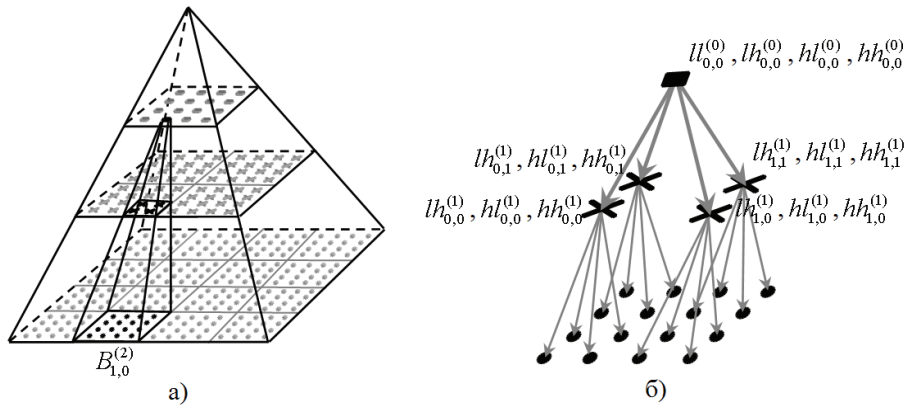


Рис. 5. а) Фрагмент пирамиды, задающий локальное многомасштабное представление участка местности, аппроксимируемого блоком $B_{1,0}^{(2)}$; б) тройное дерево вейвлет-коэффициентов сжатого разностного слоя

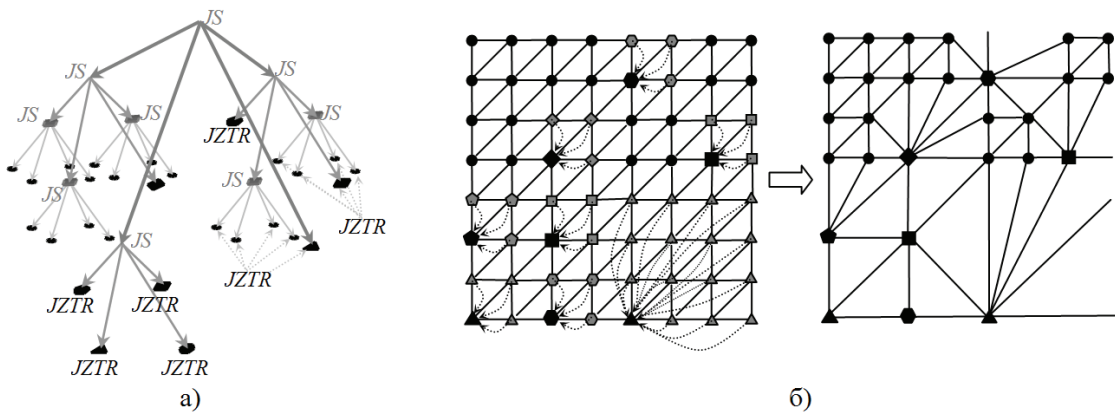


Рис. 6. а) Усеченное квадродерево, определяемое структурой тройных нуль-деревьев (помеченных JZTR) сжатого разностного слоя; б) построение адаптивной триангуляции по данному дереву

цессе иерархического сжатия сетки высот, несет информацию об особенностях поверхности рельефа и может быть использована для построения на базе локального многомасштабного представления адаптивной выборки (а по ней – триангуляции), отражающей особенности поверхности.

Чтобы использовать свойство пространственной локализации вейвлет-преобразования с целью построения адаптивной триангуляции в контексте сжатия данных, мы расширили идею нуль-дерева и ввели понятие *тройного нуль-дерева*. Будем называть тройным деревом вейвлет-коэффициентов дерево, каждому узлу которого приписаны три коэффициента из субполос LH, HL и HH, соответствующие одному и тому же пространственному положению, а корню приписаны четыре коэффициента (рис. 5б). Аналогичная интерпретация вейвлет-коэффициентов использовалась А. Перебериним [11].

Будем называть узел (i, j) тройного дерева из уровня l *значимым* по отношению к порогу τ , если хотя бы один из приписанных данному узлу вейвлет-коэффициентов является значи-

мым, т.е. если его абсолютное значение превосходит τ : $|lh_{i,j}^{(l)}| \geq \tau \vee |hl_{i,j}^{(l)}| \geq \tau \vee |hh_{i,j}^{(l)}| \geq \tau$. Будем называть узел незначимым, если все приписанные ему коэффициенты незначимы: $|lh_{i,j}^{(l)}| < \tau \wedge |hl_{i,j}^{(l)}| < \tau \wedge |hh_{i,j}^{(l)}| < \tau$. По аналогии с нуль-деревом, введенным Дж. Шапиро в работе [15] для эффективного кодирования изображений, определим понятие *тройного нуль-дерева* как тройного дерева вейвлет-коэффициентов, содержащего только незначимые узлы. Будем называть узел (i, j) тройного дерева из уровня l *корнем тройного нуль-дерева* по отношению к порогу τ , если из него растет нуль-дерево и он не принадлежит к тройному нуль-дереву, растущему из узла на более грубом разрешении. В то время как обычное нуль-дерево из [15] содержит незначимые коэффициенты, принадлежащие одной детализирующей субполосе, тройное нуль-дерево содержит незначимые коэффициенты, соответствующие данному пространственному положению, во всех трех субполосах (LH, HL и HH).

Корни тройных нуль-деревьев сжатого разностного слоя задают усеченное квадродерево

(рис. 6а) с высокой плотностью узлов в резких участках рельефа и низкой плотностью в гладких областях. На рис. 6а корни тройных нуль-деревьев изображены черным цветом и помечены меткой *JZTR* (от joint zero tree root). Нелистовые узлы усеченного дерева изображены серым цветом и помечены меткой *JS* (от significant). Узлы, которые принадлежат тройным нуль-деревьям, будут далее отмечаться меткой *JZTC*. Если собрать элементы сеток локального многомасштабного представления блока, соответствующие каждому корню тройного нуль-дерева, получится нерегулярная выборка, адаптированная к особенностям поверхности. Идея анализа вейвлет-коэффициентов для построения адаптивной выборки была выдвинута в работе А. Переберина [11], но представленный метод обладает недостатком: при принятии решения об исключении вершины учитывались только коэффициенты, приписанные данному узлу дерева, а его потомки из более детальных уровней иерархии игнорировались. Метод тройных нуль-деревьев позволяет устранить этот недостаток, т.к. вершина удаляется из выборки только в том случае, если все коэффициенты растущего из нее поддерева являются незначимыми.

Триангуляция полученной выборки может быть построена по правилу, предложенному А. Перебериним [11]: триангуляция строится так, как если бы в исходной полной триангуляции все узлы были заменены узлами, соответствующими корням тройных нуль-деревьев, которым они принадлежат, а все вырожденные треугольники после этого были отброшены. Этот процесс проиллюстрирован на рис. 6б.

Хотя триангуляция, определяемая структурой тройных нуль-деревьев, и адаптирована к локальным особенностям поверхности, она не может гарантировать заданной пространственной погрешности. Поэтому мы называем ее «негарантирующей». Чтобы решить эту проблему, мы применили ту же концепцию, что и при сжатии сетки высот: сначала строится триангуляция, удовлетворяющая заданной пространственной погрешности, а затем «разность» между полученной триангуляцией и «негарантирующей» кодируется.

2.5. Построение «гарантирующей» триангуляции. Для удобства дальнейшего изложения введем некоторые обозначения. Обозначим через $\text{proj}_{xy}(t)$ проекцию некоторого треугольника t на плоскость XY . Далее, обозначим через $\text{cov}(t)$ покрытие треугольника t , содержащее все вершины из восстановленной сетки высот максимального разрешения P локального многомас-

штабного представления данного рельефного блока, которые попадают в $\text{proj}_{xy}(t)$:

$$\text{cov}(t) = \{p \in P : \text{proj}_{xy}(p) \in \text{proj}_{xy}(t)\}.$$

Будем обозначать через $Err(t)$ погрешность треугольника t , вычисляемую как максимальное отклонение вершин из его покрытия от плоскости треугольника:

$$Err(t) = \max_{p \in \text{cov}(t)} d(t, p),$$

где $d(t, p)$ – расстояние от вершины p до плоскости треугольника t .

Наконец, будем называть погрешностью $Err(T)$ триангуляции T максимальной погрешностью всех треугольников, в нее входящих:

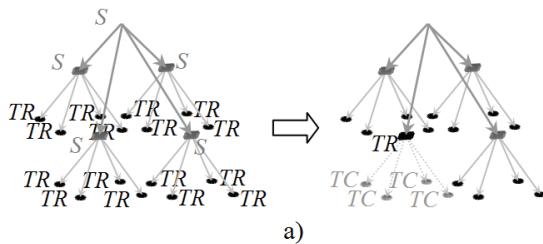
$$Err(T) = \max_{t \in T} Err(t).$$

Построение «гарантирующей» триангуляции каждого блока, которая удовлетворяет заданной пространственной погрешности, выполняется восходящим алгоритмом. Процедура принимает на вход структуру тройных нуль-деревьев и помечает каждый узел дерева локального многомасштабного представления специальной меткой. Будем использовать метки *TR* и *S* (являющиеся аналогами меток *JZTR* и *JS*), чтобы отметить соответственно корень дерева, являющийся элементом адаптивной выборки, на которой будет строиться гарантирующая триангуляция, и значимый узел соответственно. Также будем использовать дополнительную метку *TC* (являющуюся аналогом метки *JZTC*) для элементов дерева, растущего из узла, помеченного *TR*. Построение гарантирующей триангуляции начинается с сетки полного разрешения, которая последовательно огрубляется (см. рис. 7). Если после некоторого огрубления порог пространственной погрешности оказывается превышен, то изменения отменяются. Данный процесс гарантирует, что полученная в результате триангуляция будет удовлетворять заданному порогу пространственной погрешности. Порождаемая алгоритмом триангуляция ни в одном участке не может быть менее детальной, чем задает структура тройных нуль-деревьев. Обозначим через $L_l(i, j, l)$ «негарантирующую» (lossy) метку (*JS*, *JZTR* или *JZTC*) узла (i, j) дерева, лежащего в уровне l , и через $L_s(i, j, l)$ – метку дерева, задающего гарантирующую триангуляцию (*S*, *TR* или *TC*). Приведенный ниже алгоритм I помечает узлы дерева метками, задающими гарантирующую адаптивную триангуляцию. Через $T^{<m>}$ обозначены промежуточные триангуляции, получаемые после m шагов упрощения.

Алгоритм I. Построение гарантирующей триангуляции

1. Пометить все узлы на самом высоком уровне разрешения дерева меткой TR , а узлы всех остальных уровней – меткой S : $L_s(i, j, l_{\max}) \leftarrow TR$ (это задает начальную триангуляцию $T^{<0>}$ максимального разрешения, см. рис. 7а, б).
2. для каждого уровня l от самого детального (l_{\max}) до самого грубого (0) **выполнить**
 - 2.1. для каждого узла (i, j) в уровне l **выполнить**
 - 2.1.1. если $L_l(i, j, l) = JS$ то перейти к следующему узлу (триангуляция не может быть менее детальной, чем задает структура тройных нуль-деревьев)
 - 2.1.2. если $L_s(2i, 2j, l+1) = TR$ и $L_s(2i+1, 2j, l+1) = TR$ и $L_s(2i, 2j+1, l+1) = TR$ и $L_s(2i+1, 2j+1, l+1) = TR$ (т.е. все потомки данного узла помечены меткой TR) то попытаться огрубить триангуляцию (см. рис. 7):
 - * пометить узел меткой TR , а его потомков – меткой TC : $L_s(i, j, l) \leftarrow TR$; $L_s(2i, 2j, l+1) \leftarrow TC$; $L_s(2i+1, 2j, l+1) \leftarrow TC$; $L_s(2i, 2j+1, l+1) \leftarrow TC$; $L_s(2i+1, 2j+1, l+1) \leftarrow TC$
 - * Вычислить новую пространственную погрешность $Err(T^{<m+1>})$ для новой триангуляции
 - * если $Err(T^{<m+1>}) \geq \delta$ то отменить изменения: $L_s(i, j, l) \leftarrow S$; $L_s(2i, 2j, l+1) \leftarrow TR$; $L_s(2i+1, 2j, l+1) \leftarrow TR$; $L_s(2i, 2j+1, l+1) \leftarrow TR$; $L_s(2i+1, 2j+1, l+1) \leftarrow TR$

Следует отметить, что для того чтобы проверить выполнение условия $Err(T^{<m+1>}) \geq \delta$, нет необходимости проверять все треугольники из триангуляции $T^{<m+1>}$. Поскольку на любом шаге процесса построения гарантируется, что погрешность текущей триангуляции $T^{<m+1>}$ не превосходит порога, т.е. $\forall m \rightarrow Err(T^{<m+1>}) < \delta$, то недопустимая ошибка может возникнуть



только в связи с появлением новых треугольников в триангуляции. Обозначим эти треугольники через $\Delta T^{<m+1>}$: $\Delta T^{<m+1>} = \{t \in T^{<m+1>} : t \notin T^{<m>}\}$. Тогда при построении триангуляции достаточно будет проверять только следующее условие: $Err(\Delta T^{<m+1>}) < \delta$, что позволяет существенно сэкономить вычислительные затраты.

2.6. Кодирование гарантирующей триангуляции при помощи уточняющих меток. Сейчас мы имеем два набора меток узлов тройного дерева вейвлет-коэффициентов (и дерева локального многомасштабного представления). Первый набор определяется структурой тройных нуль-деревьев (метки $L_l(i, j, l)$); второй набор (метки $L_s(i, j, l)$) задает гарантирующую триангуляцию, которая несколько более детальная, чем негарантирующая. Следующий нисходящий алгоритм кодирует уточняющую информацию, которая представляет собой «разность» между данными двумя наборами меток.

Алгоритм II. Кодирование уточняющих меток

- ```
RecursiveEncodeRefinementData(0,0,0)

RecursiveEncodeRefinementData(i, j, l):
1. если $L_l(i, j, l) = JS$ то
 1.1. для каждого потомка (i_c, j_c, l_c) узла (i, j, l) выполнить RecursiveEncodeRefinementData(i_c, j_c, l_c)
2. иначе если $L_s(i, j, l) = S$ то
 2.1. Арифметически_закодировать (1) (гарантирующая триангуляция в данном участке более детальная, чем негарантирующая)
 2.2. для каждого потомка (i_c, j_c, l_c) узла (i, j, l) выполнить RecursiveEncodeRefinementData(i_c, j_c, l_c)
3. иначе Арифметически_закодировать (0) (дальнейшее уточнение не требуется)
```

Однобитовые уточняющие метки арифметически кодируются с использованием отдельной модели для каждого уровня уточнения. Это по-

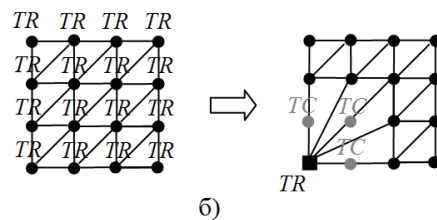


Рис. 7. а) Усечение дерева при условии, что пространственная погрешность не превосходит заданного порога; б) соответствующее изменение триангуляции

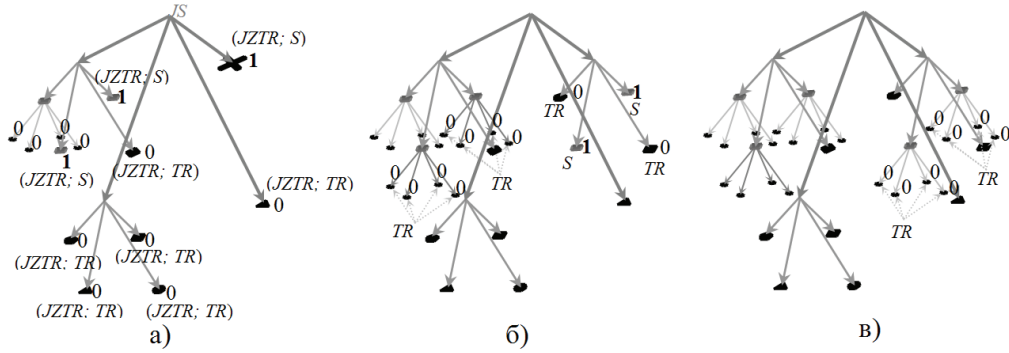


Рис. 8. а) Начальная структура тройных нуль-деревьев и уточняющие метки 0 уровня; б) уточняющие метки 1 уровня для новых узлов; в) уточняющие метки 2 уровня; больше уточнений не требуется

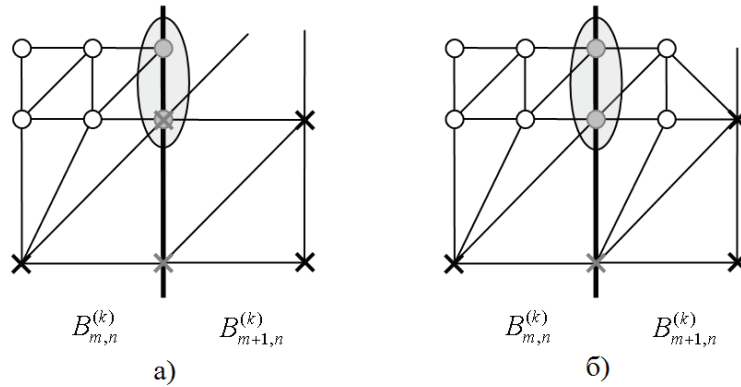


Рис. 9. Согласование адаптивных триангуляций соседних блоков

звояет учесть тот факт, что чем глубже спуск по дереву, тем менее вероятно появление единичной метки. Следует заметить, что представленный алгоритм кодирования может быть применен и без использования тройных нуль-деревьев. В этом случае пункт 1 алгоритма II должен быть исключен.

Процесс кодирования уточняющих меток проиллюстрирован на рис. 8. На рис. 8а изображено усеченное квадродерево, заданное структурой тройных нуль-деревьев. Для каждого узла дерева изображена пара меток: метки негарантирующей ( $L_l(i, j, l)$ ) и гарантирующей ( $L_s(i, j, l)$ ) триангуляций, а также соответствующая однобитовая уточняющая метка. Следующий шаг рекурсивного процесса уточнения и новые узлы, которые присутствуют в гарантирующей триангуляции, но отсутствуют в негарантирующей, показаны на рис. 8б. Финальная гарантирующая триангуляция задается деревом, изображенным на рис. 8в.

Процесс построения сжатого многомасштабного представления, в котором сетка высот и триангуляция каждого блока кодируются по описанным алгоритмам, выполняется от самого грубого до самого детального приближения.

Адаптивная триангуляция блока  $B_{m,n}^{(k)}$  строится так, чтобы удовлетворять следующей погрешности:

$$Err_{Tri}(B_{m,n}^{(k)}) = \max(\delta \cdot 2^{k_{max}-k}, Err_{Appr}(B_{m,n}^{(k)})/4).$$

Гарантированная верхняя оценка погрешности, с которой адаптивно триангулированный блок  $B_{m,n}^{(k)}$  аппроксимирует участок местности под ним, дается следующей формулой:

$$Err(B_{m,n}^{(k)}) = Err_{Appr}(B_{m,n}^{(k)}) + Err_{Tri}(B_{m,n}^{(k)}). \quad (3)$$

2.7. Обеспечение согласованности триангуляции вдоль границ блока. Чтобы обеспечить согласованность адаптивных триангуляций соседних блоков вдоль их общей границы, метки смежных граничных узлов квадродеревьев, задающих гарантирующие триангуляции блоков, должны совпадать. Иначе говоря, когда в адаптивной выборке блока присутствует граничная вершина, в адаптивной выборке соседнего блока должна присутствовать смежная с ней вершина. Чтобы обеспечить выполнение этого требования, перед кодированием меток гарантирующих триангуляций выполняется дополнительная процедура согласования границ блоков. При этом последовательно обходятся все уров-



ни дерева блоков, начиная с самого детального до самого грубого, и в каждом уровне для каждого блока проверяется согласованность вершин его адаптивной выборки с вершинами выборок соседей. При обнаружении несогласованности (рис. 9а) выполняется увеличение уровня детализации так, чтобы триангуляции блоков по обе стороны границы совпадали (рис. 9б).

*2.8. Построение адаптивной блочной модели рельефа.* При визуализации модели на каждом кадре выполняется рекурсивный алгоритм, который для каждого блока в текущем представлении на основе оценки его погрешности  $Err(B_{m,n}^{(k)})$ , вычисленной согласно (3), и текущих параметров отображения вычисляет оценку экранной погрешности по следующей формуле:

$$Err_{Scr}(B_{m,n}^{(k)}) = \gamma \frac{Err(B_{m,n}^{(k)})}{d(c, V_{m,n}^{(k)})},$$

где  $d(c, V_{m,n}^{(k)})$  – расстояние от камеры  $c$  до выровненного по осям ограничивающего бокса  $V_{m,n}^{(k)}$  блока,  $\gamma = \frac{1}{2} \frac{\max(V_x, V_y)}{\text{tg}(\varphi/2)}$ ,  $V_x$  и  $V_y$  – горизонтальный и вертикальный размеры области вывода в пикселях,  $\varphi$  – угол обзора камеры.  $Err_{Scr}(B_{m,n}^{(k)})$  дает гарантированную оценку максимального отклонения в пикселях изображения аппроксимации участка поверхности блоком  $B_{m,n}^{(k)}$  от изображения того же участка, выводимого в максимальном разрешении.

На основе сравнения экранной погрешности аппроксимации с заданным пользователем максимально допустимым видимым отклонением упрощенной модели от точной алгоритм выполняет построение усеченного квадродерева представления блочной модели, которое задает упрощенную модель для текущего кадра (см. [13]).

### 3. Результаты экспериментов

Для исследования эффективности представленного сжатого многомасштабного представления рельефа мы использовали два набора данных. Первый – территория вокруг горы Puget Sound [16], используемая как стандартный тестовый набор данных и имеющая размер  $16384 \times 16384$  при шаге 10 м. Второй набор данных – рельеф гор Кавказа, имеющий размер  $8192 \times 8192$  при шаге 30 м. Точность представления данных в этих наборах – 0.1 м, поэтому при сжатии в формуле (1) принималось  $p = 0.1$ .

В таблицах 1 и 2 приведены результаты сжатия тестовых наборов данных для размеров блоков  $256 \times 256$  и  $64 \times 64$  соответственно при размере согласующей области в 3 отсчета для погрешностей 1, 3 и 10 м. Сжатие и тесты производительности выполнялись на рабочей станции, оснащенной процессором Intel Core i7 на частоте 2.67 ГГц (4 ядра с двумя гиперпоточками на каждом), 6.0 Гб ОП и графическим ускорителем NVIDIA GeForce GTX295.

Рельеф гор Кавказа существенно более сложен, чем рельеф Puget Sound, кроме того, шаг дискретизации первого рельефа в 3 раза выше. Этим объясняется существенно меньшая степень сжатия сетки высот Кавказа при одинаковой максимально допустимой погрешности.

Степень сжатия представленного алгоритма более чем в 2.5 раза превосходит наш предыдущий метод [13]. Такой же набор данных был сжат до размера 0.61 bps методом C-BDAM. Наш алгоритм для размера блока  $256 \times 256$  и размера согласующей границы 0 позволяет получить 0.588 bps. Общий размер многомасштабного представления, включающего в себя сетку высот и адаптивную триангуляцию, составляет 0.829 bps при максимальной погрешности 1 м, что намного превосходит результаты других методов. К примеру, метод BDAM [17] для хранения адаптивной триангуляции сетки высот размером  $8192 \times 8192$  требует более 1.5 Гб дискового пространства.

Для тестирования эффективности системы визуализации мы выполняли отображение рельефа Puget Sound, сжатого с точностью 1 м для блока  $64 \times 64$  вершин, в область вывода  $1920 \times 1280$  пикселя. Замеры производительности системы выполнялись в процессе движения камеры по сложной траектории, содержащей множество резких поворотов. В процессе визуализации выполнялась процедурная текстуризация поверхности по алгоритму, приведенному в [18]. Скорость визуализации не падала ниже 100 кадров в секунду при максимальной сложности сцены до 1 млн. треугольников/кадр. При использовании простого метода раскраски поверхности путем индексации по высоте скорость визуализации составляла не менее 175 кадров в секунду. Производительность вывода треугольников при этом составляла более 330 млн. треугольников/с. За счет использования адаптивных триангуляций достигается снижение сложности модели до 3–5 раз по сравнению с использованием регулярно триангулированных блоков (как, например, в методе C-BDAM [7]), благодаря чему скорость визуализации увеличивается до 3 раз.

Таблица 1

**Результаты сжатия сеток высот рельефов «Puget Sound» и «Кавказ»  
при размере блока 256×256 и ширине согласующей границы, содержащей 3 контура**

| Погрешность (м) | Puget Sound 16384×16384 |                   |                    |                   | Кавказ 8192×8192 |                   |                    |                   |
|-----------------|-------------------------|-------------------|--------------------|-------------------|------------------|-------------------|--------------------|-------------------|
|                 | СКО (м)                 | Сетка высот (bps) | Триангуляция (bps) | Общ. размер (bps) | СКО (м)          | Сетка высот (bps) | Триангуляция (bps) | Общ. размер (bps) |
| 1               | 0.449                   | 0.636             | 0.193              | 0.829             | 0.553            | 3.604             | 0.09               | 3.694             |
| 3               | 1.120                   | 0.287             | 0.114              | 0.401             | 1.592            | 2.086             | 0.189              | 2.275             |
| 10              | 3.099                   | 0.116             | 0.049              | 0.165             | 4.527            | 0.926             | 0.208              | 1.134             |

Таблица 2

**Результаты сжатия сеток высот рельефов «Puget Sound» и «Кавказ»  
при размере блока 64×64 и ширине согласующей границы, содержащей 3 контура**

| Погрешность (м) | Puget Sound 16384×16384 |                   |                    |                   | Кавказ 8192×8192 |                   |                    |                   |
|-----------------|-------------------------|-------------------|--------------------|-------------------|------------------|-------------------|--------------------|-------------------|
|                 | СКО (м)                 | Сетка высот (bps) | Триангуляция (bps) | Общ. размер (bps) | СКО (м)          | Сетка высот (bps) | Триангуляция (bps) | Общ. размер (bps) |
| 1               | 0.423                   | 0.837             | 0.036              | 0.873             | 0.566            | 3.979             | 0.088              | 4.067             |
| 3               | 1.077                   | 0.431             | 0.030              | 0.461             | 1.571            | 2.368             | 0.08               | 2.448             |
| 10              | 2.931                   | 0.210             | 0.025              | 0.235             | 4.473            | 1.106             | 0.071              | 1.177             |

Приведенные результаты показывают, что производительность алгоритма достаточна для визуализации сцены даже в разрешении HD 1920×1280, поэтому алгоритм может быть успешно использован в подсистемах визуализации реального времени.

### Заключение и развитие подхода

В работе представлена новая многомасштабная модель рельефа местности, в которой подход двухэтапного кодирования применен для сжатия сетки высот и триангуляции в едином представлении. Структура нуль-деревьев вейвлет-коэффициентов, полученная при сжатии сетки высот, используется как базовая информация. Она определяет сетку высот и триангуляцию, которые почти везде удовлетворяют заданной погрешности. Чтобы гарантировать заданную точность восстановления, базовый слой дополняется уточняющей остаточной информацией. Степень сжатия сетки высот приведенным методом не уступает лучшим из известных алгоритмов и превосходит их, вместе с тем, в отличие от предшествующих методов, представление эффективно кодирует и адаптивную триангуляцию. При этом система способна обеспечить скорость визуализации адаптивной модели не менее 100 кадров в секунду даже в разрешении HD 1920×1280.

Наши будущие исследования будут ориентированы на использование графических акселераторов для ускорения алгоритмов. Наиболее трудоемкие этапы (вейвлет-преобразование) могут быть полностью выполнены на GPU при

помощи вычислительного шейдера, новой стадии, появившейся в DirectX11.

### Список литературы

1. Lindstrom P., Koller D., Ribarsky W. et al. Real-time, Continuous Level of Detail Rendering of Height Fields // Proc. SIGGRAPH 96. ACM SIGGRAPH, 1996. P. 109–118.
2. Duchaineau M., Wolinsky M., Sigeti D.E. et al. Roaming Terrain: Real-time Optimally Adapting Meshes // In Proc. Visualization 97. IEEE, Computer Society Press, Los Alamitos, California, 1997. P. 81–88.
3. Pajarola R. Overview of Quadtree-Based Terrain Triangulation and Visualization. Technical Report UCI-ICS-02-01, I&C Science, University of California Irvine, 2002.
4. Pajarola R., Gobbetti E. Survey on Semi-Regular Multiresolution Models for Interactive Terrain Rendering // Visual Computer. 2007. V. 23. No. 8. Pp. 583–605.
5. Losasso F., Hoppe H. Geometry Clipmaps: Terrain Rendering Using Nested Regular Grids // ACM Transactions on Graphics (Proceedings of SIGGRAPH 2004). **23(3)**. P. 769–776.
6. Malvar H. Fast Progressive Image Coding without Wavelets // Data Compression Conf. (DCC'00). P. 243–252.
7. Gobbetti E., Marton F., Cignoni P. et al. C-BDAM – Compressed Batched Dynamic Adaptive Meshes for Terrain Rendering // Computer Graphics Forum. 2006. V. 25. No 3.
8. Yea S. and Pearlman W. A Wavelet-Based Two-stage Near-Lossless Coder // Proc. ICIP. 2004. Pp. 2503–2506.
9. Dick C., Schneider J., and Westermann R. Efficient Geometry Compression for GPU-based Decoding in Realtime Terrain Rendering // Computer Graphics Forum. 2009. V. 28, No 1. P. 67–83.

10. 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. V. 2, No. 2. June 1996. Pp. 130–143.
11. Переберин А.В. Многомасштабные методы синтеза и анализа изображений: Дисс... канд. физ.-мат. наук. М.: ИПМ им. М.В. Келдыша, 2002. 138 с.
12. Yusov E., Turlapov V. GPU-Optimized Efficient Quad-Tree Based Progressive Multiresolution Model for Interactive Large Scale Terrain Rendering // Proc. of the 17th International Conference on Computer Graphics and Vision «GraphiCon'2007» Moscow, June 23–27, 2007. P. 53–60.
13. Yusov E., Turlapov V. JPEG2000-based Compressed Multiresolution Model for Real-Time Large Scale Terrain Visualization // Proc. of the 18th International Conference on Computer Graphics and Vision «GraphiCon'2008», Moscow, June 23–27, 2008. P. 164–171.
14. Yusov E. Adaptive Context Modeling for Efficient Image and Elevation Data Compression // Proc. of the 20<sup>th</sup> International Conference on Computer Graphics and Vision «GraphiCon'2010». St. Petersburg, Sept. 20–24, 2010. P. 22–29.
15. Shapiro J.M. Embedded Image Coding Using Zerrotrees of Wavelet Coefficients // IEEE Transactions on Signal Processing. Dec. 1993. V. 41, no 12. P. 3445–3462.
16. [http://www.cc.gatech.edu/projects/large\\_models/ps.html](http://www.cc.gatech.edu/projects/large_models/ps.html) (дата обращения: 15.07.2010).
17. Cignoni P., Ganovelli F., Gobbetti E. et al. BDAM – Batched Dynamic Adaptive Meshes for High Performance Terrain Visualization // Computer Graphics Forum. 2003. V. 22. No 3. P. 505–514.
18. Юсов Е.А. Алгоритм фотореалистичного отображения рельефа путем комбинирования текстур, управляемого локальными особенностями поверхности // Вестник ННГУ. 2008. № 2. С. 158–165.

#### ENCODING ADAPTIVE TERRAIN TRIANGULATION IN THE CONTEXT OF HIERARCHICAL WAVELET-BASED ELEVATION MAP COMPRESSION

*E.A. Yusov, V.E. Turlapov*

A new compressed multiresolution model for real-time terrain rendering is proposed. The model is built on the basis of a “lossy plus residual coding” concept that is exploited to compress both the elevation data and the adaptive triangulation using a single framework. Quantized wavelet coefficients resulting from wavelet-based compression of the initial height map define adaptive triangulation that almost but not everywhere satisfies the specified tolerance. Special residual information is encoded that refines the triangulation where it is necessary to assure it satisfies the given error bound. Since the major part of the information describing terrain surface features (and hence the triangulation topology) can be derived from the wavelet transform coefficients, the size of the residual information is very small. Special matching boundary region is introduced that guarantees the borders of neighboring patches perfectly match. Experimental results and comparison with other methods are presented that prove the effectiveness of the algorithms proposed.

*Keywords:* terrain, wavelet transform, compression, zero tree, adaptive triangulation.