УДК 519.688

А.В. СКВОРЦОВ, Ю.Л. КОСТЮК

СЖАТИЕ КООРДИНАТ УЗЛОВ ТРИАНГУЛЯЦИИ

Рассматривается задача упаковки триангуляции. Для сжатия координат узлов триангуляции предлагается специальный алгоритм с предсказанием, работающий совместно с алгоритмом сжатия топологии и позволяющий достичь плотности упаковки в среднем порядка 9 бит на узел.

Введение

Задача построения триангуляции по заданному набору точек широко используется для аппроксимации различного рода физических поверхностей и непрерывных полей, разбиения плоскости на элементарные части в методах конечных элементов, учёта взаимного влияния смежных тел и в других физических задачах. Кроме того, триангуляция широко используется в машинной графике и геоинформатике [1].

В связи с продолжающимся быстрым ростом производительности вычислительной техники стремительно растут и объемы обрабатываемых данных, в том числе и размеры триангуляционных моделей. В настоящее время реально используемые триангуляционные модели содержат миллионы и миллиарды треугольников. Это порождает серьезные проблемы по хранению, передаче, визуализации и обработке столь огромных моделей данных.

Стандартные триангуляционные модели данных требуют примерно 8–16 байт на хранение координат узлов триангуляции и 28–72 байта на представление топологических отношений треугольников (отношений смежности узлов, ребер и треугольников) [2]. Видно, что наибольшую долю памяти (до 90%) занимает топология триангуляции. При представлении триангуляции, содержащей миллионы узлов и треугольников, такие затраты памяти являются слишком расточительными. Одной из проблем, возникающей в связи с этим, является задача упаковки (сжатия) триангуляции для хранения во внешней памяти. При этом требуется получить два алгоритма, один из которых должен выдавать сжатое представление триангуляции для передачи во внешнюю память, а другой должен уметь генерировать исходную триангуляцию по её сжатому представлению.

Проблема сжатия триангуляции распадается на две основные подзадачи:

1. *Сжатие узлов* – задача сжатия координат узлов триангуляции и других их числовых характеристик.

2. *Сжатие топологии (топологических связей)* – задача сжатия информации, описывающей треугольники как тройки узлов триангуляции, и информации о смежности пар соседних треугольников.

Эти две задачи имеют совершенно разную природу и поэтому решаются разными методами. Задача сжатия топологии триангуляции в настоящее время достаточно исследована, и имеющиеся методы, такие, как алгоритм шелушения, позволяют достичь сжатия до 50–140 раз [3, 6].

Настоящая работа посвящена задаче сжатия координат узлов триангуляции. Обычно она решается с помощью комбинации методов с потерей точности (методы квантизации, спектральные методы) и точных методов (кодирование энтропии) [3–5]. В этих методах уровень сжатия является значительно меньшим, чем при сжатии топологии.

Основной недостаток существующих методов кодирования с потерей точности заключается в том, что в результате независимого огрубления координат узлов возможно нарушение структуры триангуляции из-за наложения треугольников друг на друга. Большинство методов с потерей точности были разработаны для использования в системах машинной графики, где нарушение структуры триангуляции не является существенным, так как триангуляция используется только для визуализации. В тоже время во многих других областях (методы конечных элементов, моделирова-

ние непрерывных полей и поверхностей) это очень важно.

В настоящей работе предлагается алгоритм сжатия координат узлов триангуляции, позволяющих достичь плотности упаковки порядка 9 бит на узел триангуляции.

Важной особенностью предлагаемого алгоритма является возможность его совместного использования с алгоритмами сжатия топологии триангуляции на основе метода шелушения.

Метод шелушения

Во время работы алгоритмов шелушения [6] для сжатия и распаковки топологии триангуляции поддерживается некоторый *граничный многоугольник*, охватывающий область обработанных треугольников. Кроме того, имеется очередь *активных ребер* триангуляции, т.е. ребер, входящих в состав граничного многоугольника, но еще не обработанных. При сохранении триангуляции в выходной поток записывается с помощью 2 бит один из управляющих кодов: **`VERTEX`**, **`SKIP`**, **`LEFT`** или **`RIGHT`**. После кода **`VERTEX`** всегда идут две координаты некоторой вершины. Рассмотрим соответствующие алгоритмы этого метода.

Алгоритм упаковки триангуляции методом шелушения

Шаг 1. Выбирается любой треугольник в триангуляции. В выходной поток посылаются координаты трех образующих узлов этого треугольника. Три его ребра образуют начальный *граничный многоугольник* и входят в состав очереди активных ребер (рис. 1,*а*).

Шаг 2. Пока очередь активных ребер не пуста, извлекаем из ее начала ребро *r* и пытаемся увеличить *граничный многоугольник* за счет треугольника *t*, смежного с *r* с внешней стороны от текущей границы:

Шаг 2.1. Если узел *n* в *t* напротив ребра *r* не лежит на *граничном многоугольнике*, то посылаем в поток код **`VERTEX`** и координаты узла *n*, увеличиваем границу и очередь активных ребер (рис. 1,*г*).

Шаг 2.2. Если узел *n* в *t* является следующим вдоль границы слева от ребра *r*, то посылаем код **`LEFT`** и увеличиваем границу и очередь активных ребер (рис. 1,*б*).

Шаг 2.3. Если узел *n* в *t* является следующим вдоль границы справа от ребра *r*, то посылаем код **`RIGHT`** и увеличиваем границу и очередь активных ребер (рис. 1,*в*).

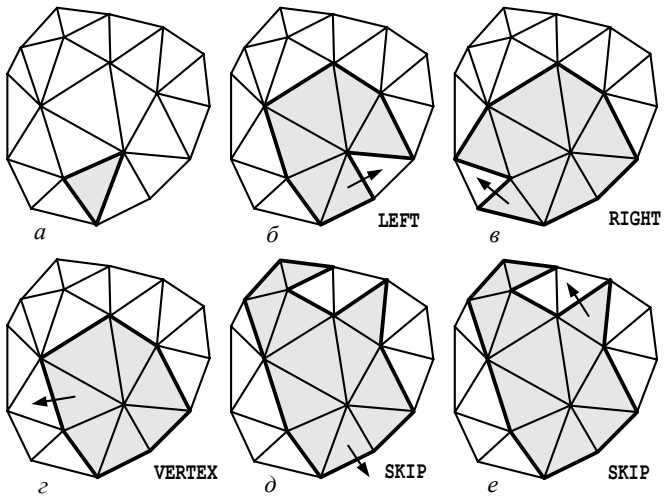

Рис. 1. Упаковка триангуляции методом шелушения: *а* – выбор начального треугольника; *б* – треугольник, замыкаемый налево; *в* – треугольник, замыкаемый направо; *г* – треугольник с новым узлом; *д* – треугольник не существует; *е* – треугольник замыкается некорректно

Шаг 2.4. Если треугольника *t* не существует (рис. 1,*д*) или узел *n* в *t* не является смежным

вдоль границы к ребру *r* (рис. 1,*e*), то посылаем в поток код **SKIP**. *Конец алгоритма.*

Аналогично построен и алгоритм распаковки. В нем также заводится текущая граница и очередь ребер, выполняется последовательное чтение управляющих кодов из входного потока и в соответствии с очередным кодом наращивается граница. Подробно приводить алгоритм распаковки здесь не будем.

Сжатие координат узлов триангуляции с предсказанием

Описанный метод шелушения является одним из наиболее эффективных алгоритмов сжатия топологии триангуляции. Но применять его совместно с эффективными методами сжатия с потерями невозможно. Если сжать координаты до применения метода шелушения, то возможно нарушение структуры триангуляции, и поэтому метод шелушения работать не будет. Если сжимать координаты после применения шелушения, то это приведет к неверному декодированию топологии, и в результате либо будет образована совершенно другая триангуляция, либо ее структура будет нарушена.

Для устранения этих недостатков в данной работе предлагается производить сжатие координат по мере сжатия топологии и генерации кода **VERTEX**.

Заметим, что на практике обычно строят триангуляции как можно «более равносторонними», т.е. можно ожидать, что каждый треугольник в среднем близок к равностороннему. Пусть AB – анализируемое ребро текущей границы. Найдем такую точку $\overline{C}$, что треугольник $AB\overline{C}$ будет равносторонним (рис. 2): $\overline{C}_x = (A_x + B_x)/2 + (A_y - B_y)\cdot\sqrt{3}/2$, $\overline{C}_y = (A_y + B_y)/2 + (B_x - A_x)\cdot\sqrt{3}/2$. Вычислим разницу между координатами C и $\overline{C}$: $\Delta x = C_x - \overline{C}_x$, $\Delta y = C_y - \overline{C}_y$. Естественно ожидать, что истинная точка C будет близка к $\overline{C}$ (рис. 2), а поэтому значения Δx и Δy должны быть малы.

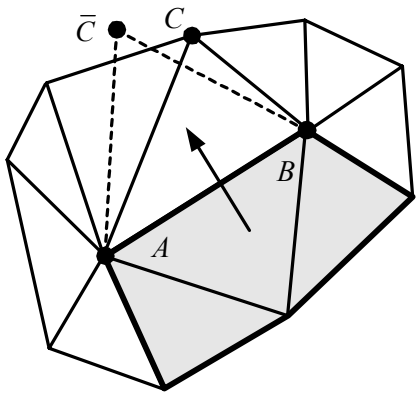

Рис. 2. Предсказание положения очередной вершины

Авторами проведено имитационное моделирование работы алгоритма шелушения и найдены эмпирические плотности распределения Δx и Δy (для большинства триангуляций они совпадают). Математическое ожидание величин Δx и Δy, как и ожидалось, оказалось равно нулю, а дисперсия $\sigma \geq 0,27/N$, где N – количество узлов в триангуляции. Коэффициент в формуле дисперсии может существенно различаться на различных видах триангуляций. Например, оценка $\overline{\sigma} = 0,27/N$ верна с вероятностью 0,95 для триангуляции Делоне, построенной на множестве из N точек, распределенных равномерно и независимо в круге единичного диаметра (рис. 3).

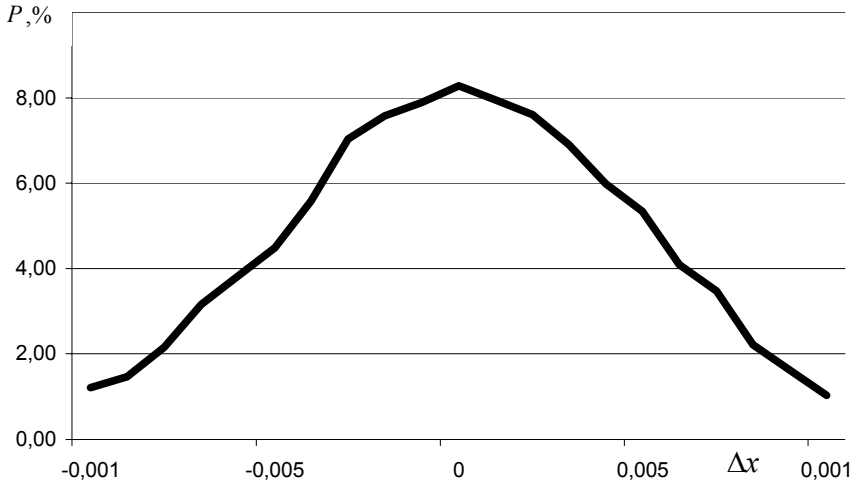

Рис. 3. Эмпирическая плотность распределения отклонений Δx – координаты очередного узла при шелушении от спрогнозированного положения в триангуляции Делоне с 10 000 узлами, распределенными равномерно и независимо в круге единичного диаметра (среднее = 0, дисперсия $\sigma = 0{,}000027$)

Из полученной эмпирической плотности распределения можно сделать вывод, что более 90% всех величин Δx и Δy распределены в диапазоне $[-0{,}1/\sqrt{N};\ 0{,}1/\sqrt{N}]$, а поэтому для их кодирования требуется гораздо меньше битов, чем для полных координат узла C. Например, при $N = 10000$ величины Δx и Δy в основном распределены в диапазоне $[-0{,}001;\ 0{,}001]$, а поэтому для их представления требуется примерно на $\log_2(N \cdot (0{,}001 - (-0{,}001))) = \log_2 500 \approx 9$ бит меньше, чем для представления исходных координат. При $N = 1000000$ величины Δx и Δy в основном распределены в диапазоне $[-0{,}0001;\ 0{,}0001]$, а поэтому для их представления требуется примерно на $\log_2(N \cdot (0{,}0001 - (-0{,}0001))) = \log_2 5000 \approx 12$ бит меньше.

Таким образом, во время шелушения при сохранении координат очередной точки проверим смещения Δx и Δy от прогнозируемого значения $\overline{C}$. Если смещения Δx и Δy попадают в диапазон $[-0{,}1/\sqrt{N};\ 0{,}1/\sqrt{N}]$, т.е. могут быть представлены меньшим количеством бит, чем исходные координаты, то сохраняем в поток бит со значением 1, а после него величины Δx и Δy с меньшим числом бит. Иначе, сохраняем в поток бит 0, а после него – координаты точки C.

Описанный способ наиболее эффективен, если исходные координаты узлов уже имеют достаточно малую точность представления, например, только 16 бит на координату. Тогда уменьшение примерно на 10 количества используемых бит будет существенным.

Однако зачастую для представления координат используется большая разрядность, например 32 бита, хотя реальная точность исходных физических данных не превышает, например, 16 бит! Поэтому необходимо также научиться уменьшать разрядность координат, т.е. выполнять их квантизацию. Если просто отбросить младшие биты координат, то в результате возможно нарушение структуры триангуляции из-за наложений треугольников. Поэтому квантизацию можно выполнять не для всех узлов.

Модифицируем алгоритм шелушения следующим образом. При обработке очередного узла C попытаемся выполнить квантизацию, например, снизив точность представления координат узла с 32 до 24, 16 или 8 бит. Если в результате этого структура триангуляции вокруг узла C не будет нарушена, то квантизацию принимаем, иначе необходимо воспользоваться точным значением. При этом в выходной поток будем посылать некоторый код, говорящий о количестве используе-

мых бит в представлении координат, и только после него – огрубленные координаты.

После снижения разрядности новые координаты необходимо также сразу же занести в текущую структуру триангуляции, чтобы в дальнейшем использовать их в кодировании. Иначе, при раскодировании будет строиться триангуляция, имеющая другие координаты, а поэтому будет построена совершенно другая триангуляции или триангуляция с нарушенной структурой.

Заключение

Описанный в работе алгоритм был применен авторами для сжатия тестовых триангуляционных моделей поверхностей, содержащих 1 млн узлов. При изначальном использовании точности представления каждой координаты в 16 бит (это соответствует, например, горизонтальной точности 10 см на участке местности, имеющем размеры 6×6 км) при использовании сжатия без потерь достигается сжатие до 9 бит на узел триангуляции, т.е. более чем в 3 раза. Итого, в совокупности с методом шелушения для сжатия топологии общее сжатие всей триангуляции составляет порядка до 9 бит на узел триангуляции, т.е. примерно на 70% по сравнению с несжатыми данными.

При использовании исходной точности каждой координаты в 32 бита алгоритм сжатие с потерями упаковывает до 10 бит на узел триангуляции, т.е. примерно на 85%.

СПИСОК ЛИТЕРАТУРЫ

1. Препарата Ф., Шеймос М. Вычислительная геометрия. Введение: Пер. с англ. – М.: Мир, 1989. – 478 с.
2. Скворцов А.В. // Вычислительные методы и программирование. 2002. Т. 3. С. 14–39. (http://num-meth.srcc.msu.su)
3. Chow M.M. // IEEE Visualization Proceedings. 1997. P. 347–354.
4. Deering M. // Comp. Graph. Proc. (SIGGRAPH). 1995. P. 13–20.
5. Evans F., Skiena S., Varshney A. // Proc. IEEE Visualization. 1996. P. 319–326.
6. De Floriani L., Magillo P., Puppo E. // Geoinformatica. 2000. No. 4. V. 1. P. 67–88.

Томский госуниверситет

Поступила в редакцию 01.04.02.