МЕТОД ТРЕХМЕРНОГО СЖАТИЯ ОБЪЕМНЫХ ДАННЫХ

Вяткин С. И.,
к.т.н,, с.н.с
Институт автоматики и электрометрии

Романюк А. Н.,
д.т.н, професор
Винницкий национальный технический университет

Круподерова Л. М.
ст. преподаватель
Винницкий национальный технический университет

Объемная визуализация одна из самых активно развивающихся областей научной визуализации [1]. В ней применяются скалярные и векторный данные заданные в трехмерном и более высоком многомерном пространстве, которые называются объемными данными. В различных областях науки и медицины объемные данные занимают от нескольких сотен мегабайт до нескольких десятков гигабайт информации, что требует специальных средств для манипулирования ими [2, 3]. Во многих методах объемной визуализации данные всего объема загружаются в главную память во время рендеринга. Однако работать с объемом в несколько гигабайт информации проблематично. Решением данной проблемы является сжатие информации. Одним из важнейших требований является быстрая произвольная выборка вокселей из сжатых данных. В известных методах существуют ограничения произвольной выборки данных при реконструкции изображений, сжатых данных с минимальными искажениями [4, 5].

В работе [6] предложен метод векторного квантования. Векторы состоят из значений плотности и заранее вычисляются поля нормалей вокселей в субблоках деления объема. Они квантуются в коды и каждый субблок представлен индексом. Рейкастинг с параллельной проекцией обрабатывает векторы согласно кодам и собирает изображение в правильном порядке.

Техника преобразования Лапласа двухмерных изображений была расширена для объемный данных [7]. В ней используется фильтр Гаусса. Воксельные данные реконструируются в результате обхода пирамиды.

Вейвлет-базируемый метод описан в [8], в котором двумерные слои преобразуются с тремя базовыми компонентами: вейвлет преобразованием, векторным квантованием и кодированием Хаффмана. Использование трехмерной вейвлет аппроксимации объемных данных показано в [9, 10].

В данной работе объемные данные делятся в единичные блоки размером 16x16x16. Для каждого единичного блока применяется трехмерное вейвлет преобразование. Используется простое преобразование Хаара с восьмиполосным фильтром. Процесс декомпозиции конвертирует каждый единичный блок в массив с одинаковым числом элементов (разложенные единичные блоки). Информация в единичном блоке может быть выражена через весовую сумму вейвлет базовых функций. Хранятся лишь коэффициенты больше некоторого порогового значения, остальные приравниваются к нулевому значению. Затем исходная информация может быть аппроксимирована небольшим числом ненулевых вейвлет коэффициентов.

В работе выбран компромисс между коэффициентом сжатия, скоростью и качеством изображения. Желательно иметь хороший коэффициент сжатия информации и быстрой произвольной выборкой. Когда сжатые объемные данные загружены в главную память, важно получить быструю реконструкцию значений индивидуальных элементов. Рассмотрим разложенный единичный блок размером 16x16x16, который представляет двухуровневую структуру исходного единичного блока. Следует отметить, что более 90% коэффициентов менее порогового значения и приравниваются к нулю. Выполняется деление единичного декомпозиционного блока на 64 субблоков, называемые клетками, где каждая клетка представляет 4x4x4 субрегион. Клетки в декомпозиционном единичном блоке перечислены друг за другом в порядке спереди-назад, сверху-вниз и слева-направо, тегированные нулями, чьи коэффициенты - все нули, и с положительными целыми числами, в порядке увеличения, клетки, содержащие по крайней мере один ненулевой коэффициент. Когда каждый тег представлен одним байтом, для 64 байт требуется клеточная теговая таблица.

При декомпозиции используется 4 байта на воксель. Поскольку исходное воксельное значение записывается 12 битами, в диапазоне от 0 до $2^{12}-1$, среднее значение будет между 0 и $2^{12}-1$, а детальное между $-2^{12}-1/2$ и $2^{12}-1/2$. Клетки с ненулевыми тегами, имеющими по крайней мере один ненулевой коэффициент, классифицируются в две группы. Первая группа состоит из клеток, чьи коэффициенты в интервале от $-(\upsilon+128),-\upsilon$ до $\upsilon,\upsilon+127$. Вторую группу составляют оставшиеся. Мы разбиваем сдвиг на r, каждому ненулевому коэффициенту клеток первой группы назначается признак (один байт), округление дробной части. Максимальная ошибка при округлении равна 0.5. Каждый ненулевой коэффициент клеток второй группы представлен целым числом (двумя байтами). Поскольку целая часть коэффициентов хранится в 12 битовом формате, используются 4 бита для дробной части.

Эти две группы ненулевых коэффициентов формируются в два массива, называемые однобайтовым и двухбайтовым потоками соответственно. Чтобы хранить коэффициенты, 64 коэффициента в клетке с ненулевыми тегами упорядочиваются в соответствующем потоке. Для поиска, коэффициенты с ненулевым значением в соответствующем потоке должны быть закодированы. Отводится дополнительная часть памяти, называемая клеточной информацией для каждой клетки 4x4x4 однобитового поля флагов и сдвига. Этот блок однобитовых флагов требует 8 байт и состоит из значимой карты, или бинарной информации о коэффициентах клетки с ненулевым значением. Информация о сдвиге хранится с помощью двух байт, состоящая из адресов соответствующих потоков коэффициентов клетки с ненулевыми значениями.

Процесс извлечения воксельных значений из вейвлет сжатых данных состоит из шагов. Все вейвлет коэффициенты необходимо отыскать из кодированных единичных блоков для реконструкции. Затем применяется формула реконструкции для коэффициентов. Поскольку вейвлет преобразование применяется дважды для декомпозиции, каждый 4x4x4 субрегион единичного блока можно рассматривать как восьмеричное дерево. Коэффициент среднего значения c^0 корня дерева является средним значением всех вокселей субрегиона, и семь коэффициентов детализации $d_i^0 (i = 1, 2..., 7)$ обеспечивают необходимую информацию, с помощью которой можно реконструировать восемь 2x2x2 субрегиона узлов уровня 1. Каждый массив семи коэффициентов детализации $d_{ji}^1 (j = 1, 2..., 8), (i = 1, 2..., 7)$ уровня 1 используются для реконструкции восьми воксельных значений соответствующего субрегиона. Для извлечения значения вокселя необходим обход восьмеричного дерева от корня к соответствующим листьям, применяя преобразование реконструкции дважды.

Рассмотрим алгоиртм.На входе: 16x16x16 кодированных единичных блоков и индексы (i,j,k).На выходе: декодированные значения коэффициентов с индексами (i,j,k). 1) Находится клетка C с индексами (i,j,k). 2) Если тег для C равен 0, возвращаем 0 (случай 1). 3) Вычисляется соответствующие индексы (i’,j’,k’) в C. 4) Если бит-флаг для (i’,j’,k’) равен 0, возвращаем 0 (случай 2). 5) Считаем предшествующее количество ненулевых коэффициентов с помощью таблицы выборки. 6) Добавляем смещение к сдвигу для вычисления корректного адреса. 7) Выбираем соответствующий поток данных и возвращаем значение (случай 3).

Данный алгоритм описывает как вейвлет коэффициенты находятся из 16x16x16 кодированного единичного блока. Для декодирования коэффициентов с индексами (i,j,k) выбирается таблица клеток тегов с коэффициентами. Если ноль, коэффициенты равны нулю (случай 1). Иначе, находится их клеточная информация для дальнейшей работы. Пусть (i’,j’,k’) соответствующие индексы коэффициентов (i,j,k) в клетке. Если бит-флаг для индексов (i’,j’,k’) равен 0, тогда коэффициенты равны нулю (случай 2). Если нет, то коэффициенты ненулевые (случай 3). Для выборки значений коэффициентов, необходимо вычислить адрес данных потока. Он вычисляется добавлением смещения к значению сдвига. Смещение — это количество ненулевых коэффициентов с флагом 1. Предварительно вычисляются табличные индексы. Данное слово состоит из двух байтов, соответствующим 16 битам флагов.

Метод протестирован на GPU 470 GTX. Использованы 16x16x16 субблоков в качестве единичных блоков. При разрешении объема 512x512x512, данные состоят из 32768 (32*32*32) единичных блоков. Применяя вейвлет преобразование для каждого единичного блока вычислили коэффициент сжатия (от 10 до 30) ненулевых вейвлет коэффициентов для 4096 коэффициентов.

Для визуализации использовался метод сплаттинга. Модель содержала 8000 единичных блоков ($16^3 \bullet 8000$ вокселей). Объёмный объект условно разделяется на воксели — трёхмерные точки, и плоский отпечаток каждой из них на плоскости экрана именуется "сплатом". Базовый алгоритм включает в себя следующие шаги. В результате свёртки каждого вокселя с определённой функцией – ядром (kernel) получаем 3D-образ этого вокселя. Полученный 3D-образ вокселя проецируется на экранную плоскость. Энергия 3D-образа распределяется на пиксели, оказавшиеся в области его проекции. Такая область в совокупности с энергией 3D-образа называется отпечатком (footprint) вокселя. Конечное изображение формируется интегрированием энергии по всем отпечаткам. Как было сказано, в алгоритме для заполнения пространства между спроецированными вокселями используется свёртка вокселя с определённой функцией, так называемым ядром. Ядро для каждого вокселя одно и то же, и не зависит от направления взгляда.

Результаты показали, что с помощью метода обеспечиваются высокий коэффициент сжатия информации и быстрая произвольная выборка значений вокселей. Это может быть полезным для многих применений, где требуется загрузка больших массивов объемных данных в главную память компьютера и их интерактивная визуализация.

Литература

4. Kaufman. A. E. Volume Vizualization. IEEE Computer society press 1991.

5. Nielson G. M., Shriver. B. E. Visualization in scientific computing. Computer society press 1999.

6. Rosenblum L. Scientific Visualization Advances and Challenges. IEEE Computer society press 1994.

7. Gonzalez R.C., Woods R.E. Digital Image Processing. Addison-Wesley Pub. Comp 1993.

8. Sayood K. Inroduction to Data Compression. Morgan Kaufmann Publishers 1996

9. Ning P., Hesselink L. Fast Volume Rendering of Compressed Data. In Proceedings of Visualization, San Jose October 1993, P. 11-18.

10. Ghavamnin M.H., Yang X.D. Direct Rendering of Laplacian pyramid compressed volume data. In Proceedings of Visualization, Atlanta, October 1995, P. 192-199.

11. Thoma G.R., Long L.R. Compressing and transmitting visible human images. IEEE multimedia, 4(2), 1997 P. 36-45.

12. Muraki S.. Approximation and rendering volume data using wavelet transforms. In Proceedings of Visualization, Boston, October 1992, P. 21-28.

13. Muraki S. Volume Data and Wavelet Transforms. IEEE Computer Graphics and Applications. 13(4), 1993, P. 50-56.

14. Chui C.K..An Introduction to wavelets. Academic press Inc. 1992.

15. E. Stollnitz, T. DeRose, D. Salesin. Wavelets for Computer Graphics: Theory and Applications. Morgan Kaufmann Publishers 1996.