

АДАПТИРОВАННЫЙ МЕТОД СПЛАТТИНГА ДЛЯ МЕДИЦИНСКИХ ПРИЛОЖЕНИЙ

Вяткин С. И., Романюк А. Н.

Аннотация

В данной работе предложен метод сплаттинга, адаптированный к быстрым вычислениям за счёт отбраковки вокселей, которые невидимы наблюдателю. По сравнению со стандартным методом хранения воксельных данных, количество информации было сокращено за счёт группового описания свободных вокселей. Разработан специальный формат хранения данных, позволяющий пользователю вводить дополнительные атрибуты вокселя, необходимые для решения конкретной задачи.

Ключевые слова: объемная визуализация, воксели, формат данных, воксель, сплаттинг.

1. ВВЕДЕНИЕ

Альтернативой представления объектов поверхностями в сцене является объем [1, 2], в котором вся сцена представляет собой набор вокселей – элементарных объёмов. Они являются своего рода аналогами пикселей в двумерной графике. Каждый воксель обычно реализуется кубом или шаром. Главным недостатком объёмов является их размер. Объем со средним разрешением 256^3 требует хранения около 16 миллионов вокселей. Чтобы сгенерировать изображение трехмерного объекта на экране, они все должны быть обработаны. Однако объём имеет и ряд важных достоинств: он может представлять внутренность объекта, а не только внешний слой. Рендеринг и обработка зависят не от сложности или типа объекта, а только от разрешения объёма.

Различают два основных метода рендеринга объёма: рендеринг в объектном пространстве или рендеринг прямого хода (object-space rendering/forward rendering) и метод пространства изображения (image-space method/backward viewing method).

В первом методе при обходе всего объёма каждый воксель рассматривается как 3D точка, которая преобразуется по матрице видового преобразования, затем проецируется в Z-буфер и рисуется на экране. Существуют два алгоритма обхода объёма: алгоритм BTF (back to front) по существу совпадает с алгоритмом Z-buffer и отличается только предварительной сортировкой массива вокселей, которая позволяет сканировать его компоненты в порядке уменьшения или увеличения расстояния до наблюдателя, используя предварительно отсортированные массивы вокселей. В алгоритме BTF обход объёма производится в порядке уменьшения расстояния до наблюдателя, а Z-buffer нужен для удаления невидимых частей объекта [3]. Алгоритм FTB (front to back) в основном совпадает с BTF, однако в

нем воксели обрабатываются в порядке возрастания расстояния [4, 5]. Это надо отметить, имея в виду то, что алгоритм Z-buffer не может обеспечивать правильный рендеринг полупрозрачных материалов, так как воксели отображаются на экран в произвольном порядке.

Во втором методе через каждый пиксель плоскости изображения пускается луч от взгляда наблюдателя сквозь объём [6, 7]. В каждом полученном пересечении луча с объемом вычисляется коэффициент непрозрачности, учитывая влияние ближайших вокселей. На протяжении траектории луча вычисляется сумма непрозрачности пересечений, пока она не достигнет определённого значения. Рассчитывается тень и освещение для полученного изображения. То есть при таком методе на один пиксель плоскости изображения влияет несколько вокселей (рис. 1 а).

Существует алгоритм (рендеринг в объектном пространстве) – сплаттинг [8], при котором один воксель влияет на несколько пикселей экрана (рис. 1 б).

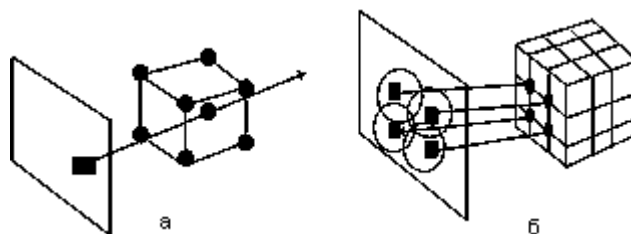


Рис. 1. а) Ray Casting и б) Splatting – противоположные методы

В данной работе рассматривается оптимизированный алгоритм сплаттинга, к которому были предъявлены следующие требования: возможность хранения дополнительных атрибутов вокселя (для каждой области применения алгоритма иметь возможность хранения специфичных только для неё атрибутов модели); хранение в описании модели минимально возможного количества информации о свободных вокселях (каждая модель имеет свободные воксели, очень часто их количество достигает большого числа, сокращение до минимума количества информации, относящейся к свободным вокселям); возможность сохранения модели в файл (разработка формата хранения описания модели в файле); возможность просмотра внутренней структуры модели; минимизация времени рендеринга (оптимизация алгоритма рендеринга с целью уменьшения времени получения изображения).

Целью данной работы является разработка алгоритма, чтобы визуализировать воксельные модели на персональных, карманных компьютерах, портативных PDA и т.д. в реальном времени.

Работа обладает большой актуальностью: реализация найдёт применение там, где важна физически точная визуализация модели (геофизика), и должна присутствовать возможность просмотра внутренней структуры объекта (медицина).

2. СПЛАТТИНГ

Алгоритм базируется на методе, называемом сплаттингом (splatting). Это вычисление формы "отпечатка" точки на плоскости экрана. Объёмный объект условно разделяется на воксели — трёхмерные точки, и плоский отпечаток

каждой из них на плоскости экрана именуется "сплатом" [8-12]. В работах [13] и [14] описаны новые примитивы - элементы поверхности (Surfels). 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, поэтому оно может быть рассчитано на стадии препроцессинга (рис. 2).

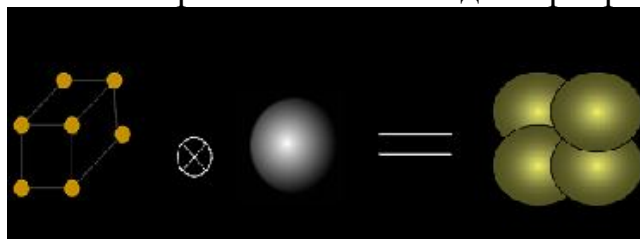


Рис. 2. Ядро

Подсчёт значения функции отпечатка во время рендеринга отнимает много времени, поэтому строим таблицу значений перед отрисовкой. Наиболее распространённой функцией отпечатка является гауссиан (gaussian) в силу того, что таблица значений для него считается относительно несложно. Для обеспечения более высокого качества и предотвращения артефактов можно увеличивать разбиение таблицы.

3. ОПТИМИЗАЦИЯ СПЛАТТИНГА

Традиционно объёмная графика требует большого количества памяти, так как хранится каждый воксель с некоторыми данными, ассоциированными с ним. Кроме этого, для качественного изображения в алгоритме сплаттинга необходима фильтрация, требующая больших вычислений [15, 16]. В работе [17] описан метод объёмной визуализации, в котором обрабатываются только те воксели, которые принадлежат поверхности объекта. Благодаря этому значительно уменьшается количество вычислений. Другой вариант оптимизации сплаттинга – иерархический сплаттинг описан в работе [18], в котором используется пирамидальная структура данных для представления объема с множественными разрешениями.

3.1 Описание модели

В данной работе предложен способ описания модели, который требует гораздо меньшего количества памяти для хранения данных, чем стандартный подход. Данный способ позволит хранить меньше информации о свободных вокселях. Модель представляется в виде отрезков вдоль оси Z (рис. 3).

3.2 Хранение модели



Рис. 3. Описание модели

В каждом вокселе кроме основных атрибутов (координат), хранятся дополнительные данные, специфичные для определённого приложения.

Хранение модели в файле происходит следующим образом:

1. Для каждого дополнительного атрибута вокселя задаётся количество байт необходимое для его хранения.
2. Каждый отрезок выглядит, как показано на рис. 4.

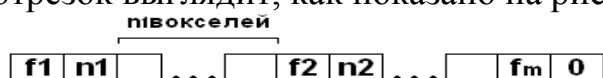


Рис. 4. Описание отрезка в файле

где f_i – количество свободных подряд идущих вокселей, а n_i – занятых; после n_i следует n_i вокселей, включая дополнительные атрибуты. Описание отрезка заканчивается, когда n_i равняется нулю.

Описание отрезков, не имеющих занятых вокселей, в памяти не хранится. Отрезки идут последовательно, начиная с первого, образуя единую последовательность байтов.

3. В отдельном массиве для каждого отрезка хранится номер байта в описанной в предыдущем пункте последовательности, с которого он начинается. Если отрезок не содержит занятых вокселей, то в массиве ему соответствует -1.

Во время работы с моделью в оперативной памяти хранятся только последовательность, описанная в пункте 2 и массив, описанный в пункте 3.

Данный способ описания и хранения модели позволяет существенно уменьшить количество информации о свободных вокселях, а, следовательно, и размер файла, при этом, сохраняя высокую скорость работы с данными в приложении, а также хранить и использовать дополнительные атрибуты вокселей.

3.3 Оптимизированный рендеринг

Недостатком оригинального алгоритма сплаттинга является то, что для получения изображения он проходит все точки модели. Однако для получения изображения по качеству не хуже, чем в изначальном алгоритме, нам достаточно использовать небольшое количество вокселей. Это те воксели, которые находятся ближе к экрану и перекрывают радиусом ядра воксели, которые находятся позади. Для отбора таких вокселей и применения алгоритма сплаттинга только к ним была выполнена оптимизация. Алгоритм оптимизации состоит из следующих шагов:

1. Создаём буферную плоскость таких же размеров, как и экранная плоскость.

2. Проходя по всей модели, проецируем на буферную плоскость воксель с учётом действия радиуса ядра. Заполняем пиксели плоскости в этом радиусе значением вокселя по Z , если это значение меньше текущего значения пикселя, и оставляем прежним, если больше.

3. Для применения алгоритма сплаттинга отбираем только те воксели, проекция которых на плоскости хотя бы частично существует.

На рисунке 5 показано: точка в центре находится по Z дальше всех – после проекции на плоскость-буфер её радиус перекрывается полностью радиусами других точек, следовательно, она отбрасывается.

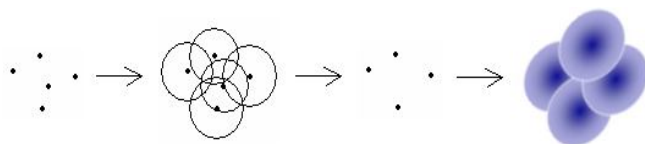


Рис. 5. Оптимизированный сплаттинг

3.4 Тестирование приложения

3.4.1 Размер файла

Было проведено сравнение размера файлов разработанного метода с размером файлов при стандартном методе хранения воксельно-базируемых моделей. Результаты представлены в таблице 1.

Таблица 1. Размеры файлов стандартного способа хранения модели и разработанного.

	Стандартный способ	VMD
Грудная клетка	8411000	382318
Берцовая кость	72457	36220
Череп	122600	56487

Из таблицы 1 видно, что размер файла разработанного метода существенно меньше размера файла стандартного подхода. Это достигнуто за счет того, что палитра хранится отдельно и в вокселе содержится лишь индекс цвета. А также за счет сокращения информации о свободных вокселях путём их группировки. Так же из приведённой таблицы можно заметить, что чем больше модель, тем существеннее сокращение размера файла: на небольшой модели берцовой кости – это в 2 раза, на большой же модели самолёта это число достигло 22.

3.4.2 Время рендеринга

Также было проведено тестирование времени рендеринга и сравнение полученных результатов со временем рендеринга оригинального алгоритма сплаттинга. Ниже приведена таблица 2 результатов тестирования. Как видно из таблицы, время рендеринга, адаптированного сплаттинга значительно меньше времени рендеринга оригинального. Это объясняется тем, что алгоритм сплаттинга, при котором к каждому вокселю применяется функция ядра с расчётом влияния на соседние воксели, применяется только к видимым в данный момент вокселям.

Таблица 2. Время рендеринга оригинального сплаттинга и адаптированного (среднее), в миллисекундах.

	Оригинальный сплаттинг	Оптимизированный сплаттинг
Грудная клетка	8.4	1.9
Берцовая кость	3.4	1.1
Череп	4.1	1.3

Как видно из таблицы 2, время рендеринга, адаптированного сплаттинга значительно меньше времени рендеринга оригинального. Это объясняется тем, что алгоритм сплаттинга, при котором к каждому вокселю применяется функция ядра с расчётом влияния на соседние воксели, применяется только к видимым в данный момент вокселям. Тестирование производилось на компьютере с процессором Intel Core2 CPU E8400 3.0 GHz, и графическим акселератором GTX 470.

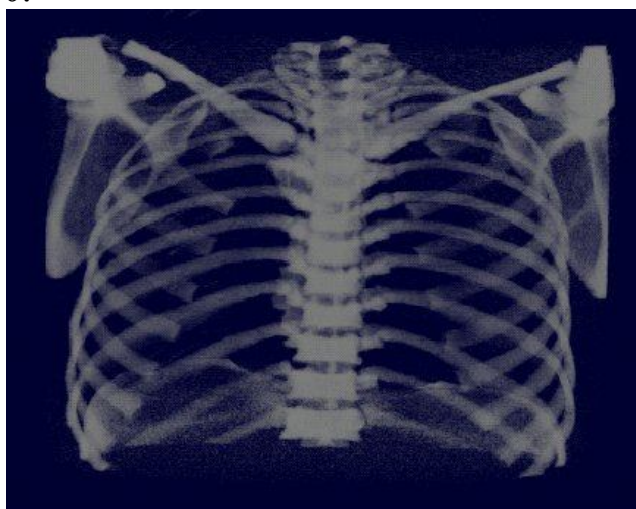


Рис. 6. Модель грудной клетки

4. ЗАКЛЮЧЕНИЕ

В данной работе предложен метод визуализации объёмных данных с использованием разработанного формата хранения моделей воксельно-базируемого подхода. В качестве алгоритма рендеринга был выбран сплаттинг, оптимизированный за счёт отбраковки вокселей, невидимых в данный момент перед отрисовкой. Разработанный формат описания модели предусматривает её хранение в файле. По сравнению со стандартным методом хранения воксельно-базируемых моделей, количество информации о свободных вокселях было сокращено за счёт группового описания свободных вокселей. Формат описания вокселя позволяет пользователю вводить дополнительные атрибуты вокселя, необходимые для решения конкретной задачи. Было реализовано приложение с использованием предложенного метода, а также проведено его тестирование, результаты которого доказывают выполнение поставленных требований: размер файлов сократился в 2-22 раза, время рендеринга также сократилось в 5-6 раз. Перед началом данной работы мы исходили из следующих предпосылок. Видимых вокселей в процентном соотношении не более 5% из всего объема

при условии, что объекты непрозрачные и сегментированные. Сегментированный объект, это когда из всего куба отсканированных данных остаются, к примеру, только кости, кровеносные сосуды и т.д. (сегментирование по плотностям). Вокселей, принадлежащих костям или другим органам не так много, в сравнении со всем объемом. Поэтому хранить весь куб данных не надо, а только - список вокселей после сегментации. Это же относится и к визуализации геофизических данных. В связи с ростом вычислительных возможностей карманных компьютеров возникает новая актуальная задача создания специализированных программных средств обработки геофизических данных полевых измерений в реальном времени на этих платформах. Сплаттинг подходит не для всех задач компьютерной графики. Но для ограниченного круга задач он оптимален как по быстродействию, так и по реалистичности изображений.

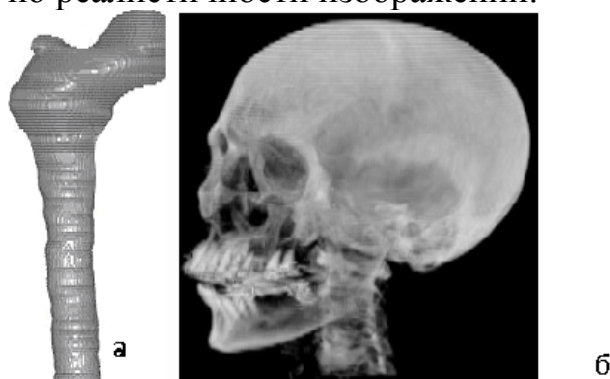


Рис. 7. Модели: а) берцовой кости и б) черепа

5. БИБЛИОГРАФИЯ

- [1] A. Kaufman, D. Cohen, and R. Yagel, "Volumetric Graphics" // IEEE Computer, 26(7):51-64, July 1993.
- [2] Р. Ягель Рендеринг объемов в реальном времени, Открытые системы, № 5, 1996, стр. 29-33.
- [3] Frieder, G., Gordon, D., and Reynolds, R. A., "Back-to-Front Display of Voxel-Based Objects", IEEE Computer Graphics and Applications, 5(1):52-60, January 1985.
- [4] Drebin, R. A., Carpenter, L., and Hanrahan, P., "Volume Rendering", Computer Graphics, 22(4):65-74, August 1988.
- [5] Lacroute P., and M. Levoy, "Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation", Proceedings SIGGRAPH' 94, Orlando, Florida, July 24-29, 1994. In Computer Graphics Proceedings, Annual Conference Series, 1994, ACM SIGGRAPH, pp.451-458.
- [6] Tuy, H. K., and Tuy, L. T., "Direct 2-D Display of 3-D Objects", IEEE Computer Graphics and Applications, 4(10):29-33, November 1984.
- [7] Upson, C., and Keeler, M., "V-BUFFER: Visible Volume Rendering", Computer Graphics, 22(4):59-64, August 1988.
- [8] L. Westover, "Footprint Evaluation for Volume Rendering" Computer Graphics, 22(4):367-376, August 1990.
- [9] B. Curless and M. Levoy. A Volumetric Method for Building Complex Models from Range Images. In Computer Graphics, SIGGRAPH 96 Proceedings, pages 303-312. New Orleans, LA, August 1996.

- [10]N. Greene and P. Heckbert. Creating Raster Omnimax Images from Multiple Perspective Views Using the Elliptical Weighted Average Filter. *IEEE Computer Graphics & Applications*, 6(6):21–27, June 1986.
- [11]M. Levoy and T. Whitted. The Use of Points as Display Primitives. Technical Report TR 85-022. The University of North Carolina at Chapel Hill, Department of Computer Science, 1985.
- [12]J. Shade, S. J. Gortler, L. He, and R. Szeliski. Layered Depth Images. In *Computer Graphics, SIGGRAPH 98 Proceedings*, pages 231–242. Orlando, FL, July 1998.
- [13]H. Pfister, M. Zwicker, J. van Baar, and M. Gross. Surfels: Surface Elements as Rendering Primitives. In *Computer Graphics, SIGGRAPH 2000 Proceedings*, pages 335–342. Los Angeles, CA, July 2000.
- [14]S. Rusinkiewicz and M. Levoy. QSplat: A Multiresolution Point Rendering System for Large Meshes. In *Computer Graphics, SIGGRAPH 2000 Proceedings*, pages 343–352. Los Angeles, CA, July 2000.
- [15]K. Mueller, T. Moeller, J.E. Swan, R. Crawfis, N. Shareef, and R. Yagel. Splatting Errors and Antialiasing. *IEEE Transactions on Visualization and Computer Graphics*, 4(2):178–191, April-June 1998.
- [16]J. E. Swan, K. Mueller, T. Moeller, N. Shareef, R. Crawfis, and R. Yagel. An Anti-Aliasing Technique for Splatting. In *Proceedings of the 1997 IEEE Visualization Conference*, pages 197–204. Phoenix, AZ, October 1997.
- [17]Sobierajski L., D. Cohen, A. Kaufman, R. Yagel, and D. Acker, “A Fast Display Method for Volumetric Data”, *The Visual Computer*, 10(2):116-124, 1993.
- [18]Laur, D. and Hanrahan, P., “Hierarchical Splatting: A Progressive Refinement Algorithm for Volume Rendering”, *Computer Graphics*, 25(4):285-288, July 1991.