

Структуры данных в построении сложных трехмерных поверхностей по набору точек

Пауков Д. П.

Донецкий национальный технический университет

paukov@pmi.dgtu.donetsk.ua

Abstract

Paukov D. P., Data structures in complex 3D surface reconstruction from points set. The data structure is shown at handling a set of points and complex three-dimensional surfaces construction. The main data structures that describe geometric objects, such as points, planes, vectors, triangles, cells, storage of points, mappings and hash-tables are presented. It is specified what tools of the modern programming languages (C++, Java and C#) allow to implement these structures.

Введение

Моделирование и визуализация сложных объектов становится актуальной задачей в современной науке и технике, особенно сейчас, когда производительность вычислительных машин позволяет решать задачи моделирования сложных объектов и визуализации результатов сложных научных экспериментов.

Использование набора трехмерных точек, полученных, например, путем оптического или ультразвукового сканирования реальных объектов в системах геометрического моделирования и визуализации без дополнительной обработки не эффективно. Использование вместо набора точек поверхности позволит существенно улучшить качество моделирования сложного объекта. Например, полигональная модель поверхности более компактная, чем набор точек, позволяет получить более реалистичное изображение объекта и проще выполнить различные операции с поверхностью объекта.

Методы построения полигональной модели поверхности разрабатываются в компьютерной графике последние несколько лет. Однако уже можно выделить некоторые отличающиеся подходы в решении этой проблемы. К ним относятся:

- пространственное разбиение (spatial subdivision), когда исследуемая область пространства разбивается на ячейки, определяющие части искомой поверхности [1];
- использование функции расстояния до искомой поверхности [2-3];
- техники деформации, основанные на грубом начальном приближении искомой поверхности [4];
- использование триангуляции Делоне [5] в качестве первоначальной модели поверхности с последующей обработкой [6-8].

Все представленные методы подразумевают определенную организацию обрабатываемых геометрических данных. Как правило, исходные данные помещаются в специальное хранилище, а результат представляется в виде триангуляционной сети или набора треугольников.

Цель статьи состоит в описании структур данных, используемых в компьютерной графике для построения полигональных моделей по набору точек. Задача статьи – адаптировать общие структуры данных, такие как списки, хеш-таблицы и отображения, для использования в методе построения сложных трехмерных поверхностей [14].

Списки и простые структуры

К простым структурам данных, не требующих особого пояснения, можно отнести структуры, описывающие трехмерные точки и плоскости. Такие структуры представляются, как правило, соответствующими классами. На рис. 1 в виде диаграммы UML [9] показаны классы точки (Point), вектора (Vector), плоскости (Plane), треугольника (Triangle) и ячейки (Cell).

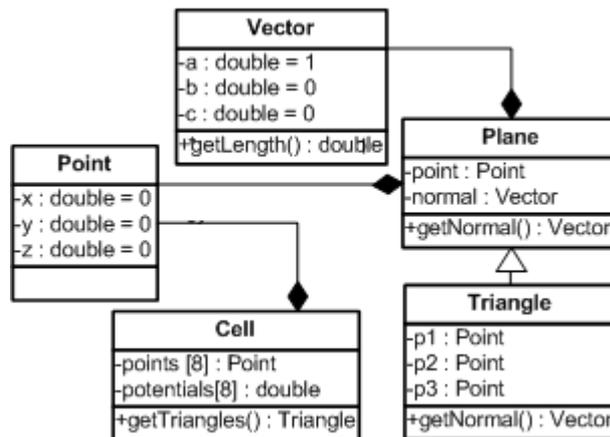


Рисунок 1 – Взаимосвязь простых структур

Между простыми структурами данных (рис. 1) существует взаимосвязь: треугольник может представлять плоскость, а точка является базовой сущностью для плоскости и ячейки, вектор является составной частью описания плоскости. Для представления набора данных можно использовать коллекции, такие как множества или списки. Таким образом, возникают списочные структуры данных: точки (Points), плоскости (Planes), треугольники (Triangles) и решетка (Grid) – эти структуры в качестве своих элементов списка точку, плоскость, треугольник и ячейку соответственно.

Простые структуры данных, как будет показано ниже, являются основой для более сложных структур, таких как хеш-таблицы и отображения. Списочные структуры лежат в основе хранилищ данных, среди которых особое место занимает хранилище точек потому, что именно к точкам чаще всего в упомянутых выше методах [1-4,6-8] осуществляется доступ.

Хранилище точек

Для уменьшения времени доступа к набору точек и упрощения выполнения различных запросов на отбор точек в заданной области лучше организовать специальную структуру данных, основанную на нескольких отображениях и списках (рис. 2). Основная идея организации хранилища точек состоит в том, что точка не хранится как единое целое, а разбивается на координатные компоненты, которые по отдельности помещаются в специальные таблицы отображений в качестве ключа. Каждая такая таблица состоит из поля отсортированных в порядке возрастания уникальных ключей и списков точек. Каждому ключу ставится в соответствие список точек, у которых соответствующая координатная компонента равняется значению ключа. Таким образом, для хранилища трехмерных точек потребуется три таблицы: x-Map, y-Map и z-Map.

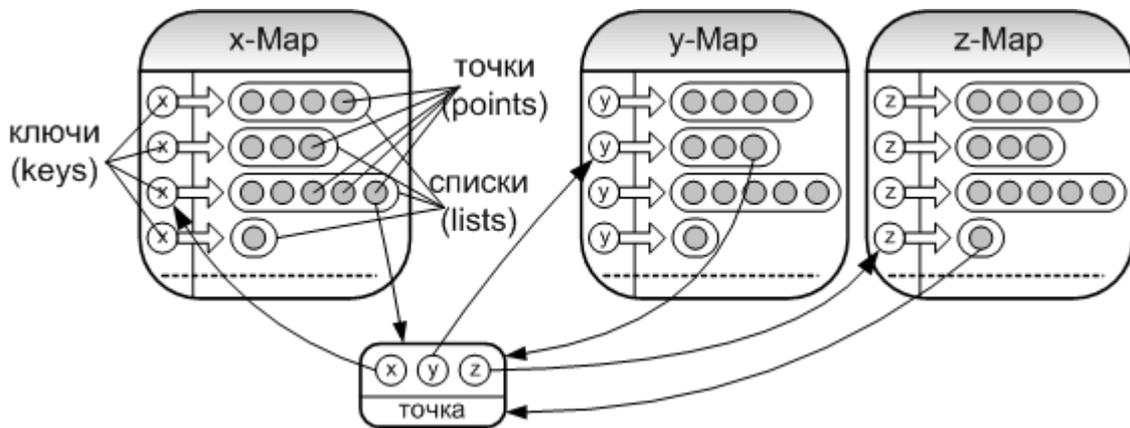


Рисунок 2 – Организация хранилища точек

Точка помещается в хранилище таким образом, что координатные компоненты не нарушают порядок сортировки ключей, и сама точка в хранилище не находится. Вместо неё используются ссылки на точки. Порядок точек в списке определяется порядком их поступления.

Реализовать хранилище точек можно с помощью коллекции map из стандартной библиотеки STL C++ [10], TreeMap в Java [11] или TreeDictionary из библиотеки C5 [13].

Хеш-таблицы и отображения

Для выполнения манипуляций с точками и поиска соседних с точкой плоскостей используют отображения или хеш-таблицы. Каждой точке ставится в соответствие плоскость, которая наименее удалена от всех соседних точек [14]. Это соответствие представляется соответствующей хеш-таблицей (рис. 3), в качестве хеш-функции выбирается значение, зависящее от координат точки.

Кроме отображения точки на плоскость, используется отображение плоскости на множество соседних плоскостей. Такое отображение позво-

ляет быстро определить значение функции расстояния [1, 14] до всех соседних плоскостей. Отображение плоскости на множество соседних плоскостей строится на основе хеш-таблицы отображений точек на плоскость. Множество соседних плоскостей состоит из плоскостей, на которые отображается множество соседних точек.

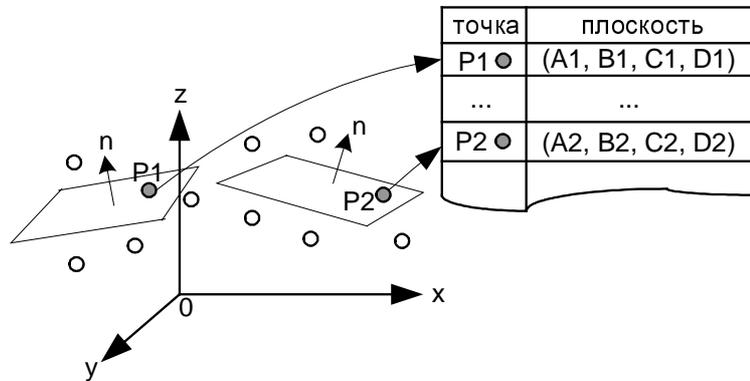


Рисунок 3 – Хеш-таблица отображения точек на плоскость

Реализовать хеш-таблицы и отображения можно с помощью шаблона `map` стандартной библиотеки STL C++ [10] или коллекции `HashMap` в Java [11], `Dictionary` в C# [12] и `HashSet` в C5 [13].

Выводы

В статье рассмотрена организация структур данных, используемая в построении поверхностей сложной формы по набору точек [14]. Простые структуры данных, такие как точка, плоскость, вектор и треугольник являются базовыми, на их основе строятся более сложные структуры данных. Списочные структуры данных содержат в качестве своих элементов простые структуры. Хранилище точек является комбинацией простых структур, списочных и отображений. Хеш-таблицы и отображения используют простые структуры в качестве ключей и элементов. Поэтому к реализации простых структур данных необходимо подходить с особой тщательностью.

Таблица 1. Адаптированные структуры данных

Название	C++, STL	Java	C#
Простые структуры (точка, плоскость и т.д.)	struct, class	class	struct, class
Списочные структуры (наборы точек, плоскостей и т.д.)	list	ArrayList	List
Хранилище точек	map, list	TreeMap	TreeDictionary (C5)
Хеш-таблицы, отображения	set, map	HashMap	Dictionary, HashSet(C5)

Современные языки программирования представляют богатые библиотеки коллекций, среди прочих можно выделить библиотеку STL [10], коллекции Java [11] и C# [12], а также библиотеку C5 [13]. В табл. 1 показаны сводные данные, на основе каких стандартных структур данных были

предложены адаптированные описанные выше структуры для методов построения поверхностей по набору точек. Наиболее простыми в реализации оказались коллекции Java. Стандартная библиотека коллекций языка C# не позволяет создавать такие структуры как описанное выше хранилище точек, однако благодаря библиотеке C5 [13] такую поддержку удалось получить. Коллекции STL оказались более сложными в реализации, однако за счет быстродействия кода, полученного с помощью C++, адаптированные структуры STL имеют лучшую производительность. В целом структуры данных для построения сложных трехмерных поверхностей хорошо представляются во всех упомянутых выше языках.

Дальнейшее усовершенствование структур данных для методов построения поверхностей сложной формы может состоять в разработке структур данных и методов параллельного построения поверхностей на больших объёмах исходных данных и построения криволинейных моделей поверхностей [15].

Литература

1. Hoppe H., DeRose T., Duchamp T., McDonald J., Stuetzle W. Surface Reconstruction from Unorganized Points. – University of Washington. – 1992.
2. Позин А.Г. Использование объёмного метода для восстановления 3D поверхностей // Graphicon proceedings. – 2005.
3. H. Zhao Analysis and visualization of large set of unorganized data points using the distance function, 2003
4. J.Esteve, P.Brunet, A. Vinacua Approximation of a Variable Density Cloud of Points by Shrinking a Discrete Membrane. Computer Graphics. – 2005. - vol. 24, pp. 791-807.
5. Пауков Д. П. Триангуляция Делоне: итеративные алгоритмы построения триангуляции. // Збірка праць магістрантів Донецького національного технічного університету. Випуск 2. – Донецьк: ДонНТУ. – 2003. – с. 461-469.
6. Cohen-Steiner D., Da F. A Greedy Delaunay Based Surface Reconstruction Algorithm, Research Report. – INRIA. – 2002.
7. Dey T. K., Goswami S. Provable surface reconstruction from noisy samples // Computational Geometry Theory & Applications. – 2006. – vol. 35, pp. 124-141.
8. Chuan-Chu Kuo, Hong-Tzong Yau A Delaunay-Based Region-Growing Approach to Surface Reconstruction from Unorganized Points. pp. 73-82 2003.
9. Wendy Boggs, Michael Boggs Mastering UML with Rational Rose 2002 SYBEX, 2002. – 702 p.
10. Bjarne Stroustrup The C++ Programming Language Third Edition, Addison-Wesley – 1999. – 912 p.

11. Эккель Б. Философия Java. Библиотека программиста. – СПб: Питер, 2001. – 880 с.: ил.
12. Нейгель, Кристиан, Ивьен, Билл, Глинн, Джей, и др. C# 2005 для профессионалов.: Пер. с англ. – М.: Издательский дом «Вильямс», 2006. – 1376 с.
13. Niels Kokholm, Peter Sestoft The C5 Generic Collection Library for C# and CLI, IT University of Copenhagen Technical Report Series TR-2006-76, – 2006. – 255 p.
14. Хромова Е.Н., Пауков Д.П., Башков Е.А. Воссоздание поверхности по произвольному набору точек. Подзадача построения плоскости, наименее удаленной от совокупности точек. // Збірка доповідей II республіканської наукової конференції студентів, аспірантів та молодих вчених «Комп'ютерний моніторинг та інформаційні технології». – Донецьк: ДонНТУ. – 2006 р.
15. Е.А. Башков, Д.П. Пауков. Математические модели поверхностей сложной формы в системах моделирования и визуализации. Моделирование и компьютерная графика: Материалы 1-й международной научно-технической конференции, Донецк, 4-7 октября 2005 г. – ДонНТУ, Министерство образования и науки Украины, 2005. с. 59-65.