

## **ПОСТРОЕНИЕ ПОВЕРХНОСТИ ОБЪЕКТОВ СЛОЖНОЙ ФОРМЫ ПО НАБОРУ ТОЧЕК ДЛЯ ВИЗУАЛИЗАЦИИ И МОДЕЛИРОВАНИЯ**

Башков Е.А., Пауков Д.П.

Донецкий национальный технический университет

*Йдеться про побудову полігонального подання складних поверхонь, що будуються за набором тривимірних точок. Представлена модифікація метода, що використовує розділення простору на осередки та допускає паралельне обчислення.*

Построение поверхности объектов сложной формы – задача актуальная для многих отраслей науки и техники, медицины, истории и культуры. Моделирование и визуализация сложных поверхностей лежит в основе трехмерной компьютерной томографии [1], визуализации результатов научных экспериментов [2]. Построение сложных поверхностей часто встречается при проектировании различных механизмов в машиностроении, в строительстве и архитектуре [3]. Многие музеи используют построение сложных поверхностей, как для создания цифровых копий скульптур, так и для повышения качества реставрационных работ [4].

Первые публикации, посвященные проблеме построения поверхностей по набору точек, появились сравнительно недавно: в 1990-х годах [5-6]. За это время было предложено несколько подходов в решении этой задачи: методы, использующие триангуляцию Делоне [7] в качестве первоначальной модели поверхности с последующей обработкой [8-9], использующие пространственное разбиение [5], использующие функцию расстояния до искомой поверхности [10-11], а также использующие техники деформации [12].

Задача построения поверхности по набору точек состоит в следующем. Имеется набор трехмерных точек, описывающих поверхность (замкнутую или разомкнутую). Используя только эти сведения, а также некоторую величину среднего расстояния между соседними точками, необходимо получить полигональное представление поверхности, проходящей вблизи исходных точек.

В этой работе рассматривается модификация метода построения полигонального представления поверхности по набору точек, основанного на пространственном разбиении и функции знакового

расстояния до искомой поверхности [5-6, 13]. В качестве элемента полигональной сетки используется треугольник – как наиболее простая геометрическая фигура, позволяющая эффективно визуализировать поверхность. Вершины треугольников (узлы сетки) вычисляются с помощью метода marching cubes [6]. Размер ребра куба задается дополнительно и определяет размеры треугольников. Первоначальные плоскости вычисляются путем отбора для каждой исходной точки множества ближайших точек, при этом используется заданная величина среднего расстояния между ними. В случае, когда эта величина не задана, её можно подобрать экспериментально.

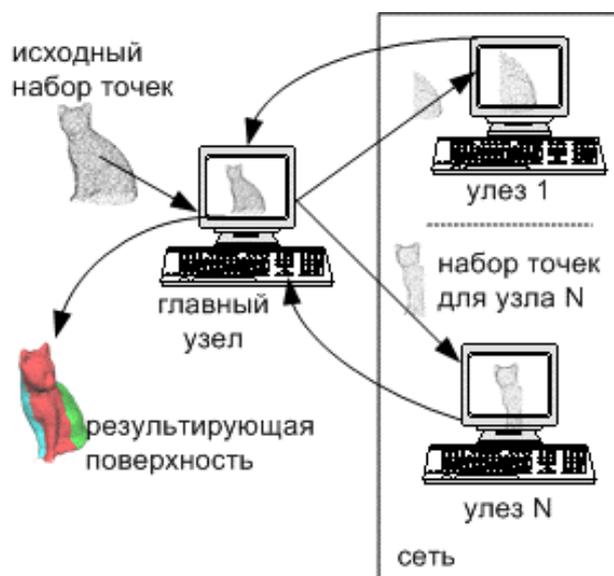


Рис. 1 – Схема построения поверхности по набору точек с помощью N вычислительных узлов.

При выполнении построения сложной поверхности по набору точек на одном вычислительном устройстве возникает проблема недостатка ресурсов памяти, так как сложные поверхности описываются большим количеством точек, и как следствием высокими временными затратами. Решить эту проблему можно путем разделения исходных точек на части, которые могут быть обработаны независимо.

На рис. 1 показана схема метода построения поверхности путем разделения вычислений между несколькими вычислительными узлами работающими параллельно. В качестве параллельных вычислительных узлов может быть использован кластер или многопроцессорная вычислительная машина. Исследуемое пространство разделяется на параллелепипеды в зависимости от количества имеющихся вычислительных узлов. Каждому

вычислительному узлу ставится в соответствие полученный параллелепипед. Исходные точки разделяются на группы, принадлежащие пространству, заключенному в параллелепипеде, и являются исходными данными построения части поверхности для соответствующего вычислительного узла. После окончания работы всех вычислительных узлов, полученные части поверхности объединяются в единую полигональную модель [10].

Таблица 1 – Результаты построения сеточной модели

№	Средн. расст., $\rho$	Ребро решетки, $e$	Кол-во точек	Кол-во треугольников	Первон. построение плоскостей, мс	Выбор направления нормалей, мс	Построение сетки, мс	Общее время, мс
стопа человека								
1	5.5	3.0	1 362	7 568	290	1 512	32 446	34 248
2	5.6	3.5	1 257	4 665	210	1 082	16 134	17 426
3	5.5	3.0	1 432	7 552	300	1 552	36 012	37 864
4	6.2	5.0	1 041	1 753	30	982	4 436	5 448
$\Sigma$	-	-	5 092	21 538	300	1 552	36 012	37 864
$\Omega$	5.5	3.0	5 092	24 720	480	9 734	256 459	266 673
кошка (4 части)								
1	0.018	0.018	1 763	1 626	60	1 783	5 508	7 351
2	0.018	0.018	3 137	2 744	120	6 749	13 880	20 749
3	0.018	0.018	1 989	1 861	50	2 233	6 529	8 812
4	0.018	0.018	3 111	2 705	80	5 358	12 117	17 555
$\Sigma$	-	-	10000	8 936	120	6 749	13 880	20 749
$\Omega$	0.018	0.018	10000	8 719	290	35 451	99 705	135 446
кошка (8 частей)								
1	0.019	0.018	1 599	1 427	66	1 748	4 824	6 638
2	0.019	0.018	164	181	3	89	116	208
3	0.019	0.018	1 323	1 091	31	1 749	2 892	4 671
4	0.019	0.018	1 814	1 553	33	2 376	4 594	7 003
5	0.019	0.018	1 575	1 396	31	1 748	4 243	6 023
6	0.019	0.018	414	436	33	270	423	726
7	0.019	0.018	1 598	1 367	33	1 794	3 705	5 532
8	0.019	0.018	1 513	1 320	33	1 571	3 552	5 156
$\Sigma$	-	-	10000	8 771	66	2 376	4 824	7 003

Рассматриваемый метод построения сетки на практике продемонстрировал хорошие результаты. В табл. 1 представлены временные характеристики построения сетки для модели стопы человека и модели кошки, выполненные как на одном вычислительном узле, так и на нескольких узлах. Символом  $\Sigma$  в таблице отмечены результирующие показатели характеристик построения сетки на нескольких узлах. Временные характеристики в этом случае определяются как наибольшее значение времени работы

любого из узлов. Символ  $\Omega$  означает построение поверхности без разделения исходных точек на части и соответственно без привлечения дополнительных вычислительных узлов. Экспериментальные данные получены на ЭВМ с процессором Intel Pentium III 733 МГц, ОЗУ 512 МВ.

Представленная на рис. 2 сравнительная диаграмма временных затрат демонстрирует, что параллельная обработка частей исходных точек может существенно улучшить временные характеристики построения сложных поверхностей и соответственно повысить сложность обрабатываемых поверхностей.

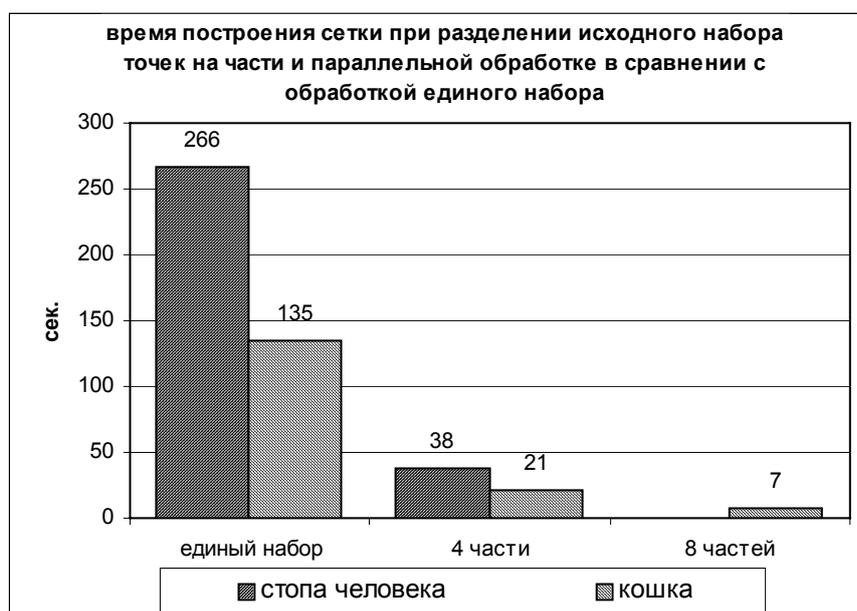


Рис. 2 – Сравнительная диаграмма времени построения поверхностей в зависимости от количества вычислительных узлов

На рис. 3 показаны три модели поверхности кошки, полученные построением на одной вычислительной машине и разделенные на 4 и 8 частей закрашенные различным цветом. На рис. 4 показаны две модели поверхности стопы человека, полученные построением на одной вычислительной машине и разделенные на 4 части закрашенные различным цветом.

Анализ экспериментальных данных из табл. 1 и рис. 2-4 показывает, что параллельная обработка исходных точек при построении поверхности сложной формы существенно снижает временные затраты: использование 4-х вычислителей параллельно вместо одного уменьшает время построения более чем в 6 раз, 8 вычислительных узлов позволяют снизить время построения поверхности более чем в 15 раз. В таблице не учитывалось время

разделения набора исходных точек на группы и время объединения частей поверхностей в единую модель, т.к. для представленных случаев эти показатели незначительны.

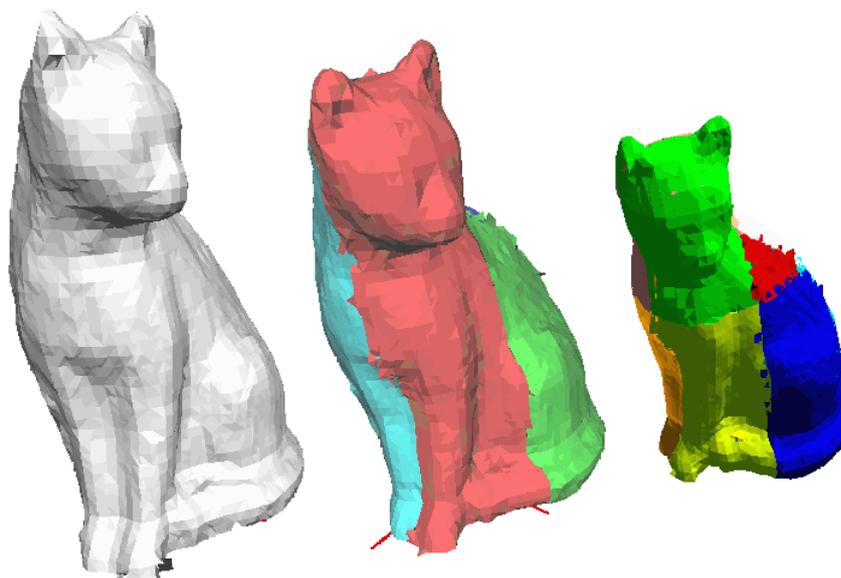


Рис. 3 – Поверхность кошки, построенная по единому набору точек (слева), разделенная на 4 части (в центре) и на 8 частей (справа)

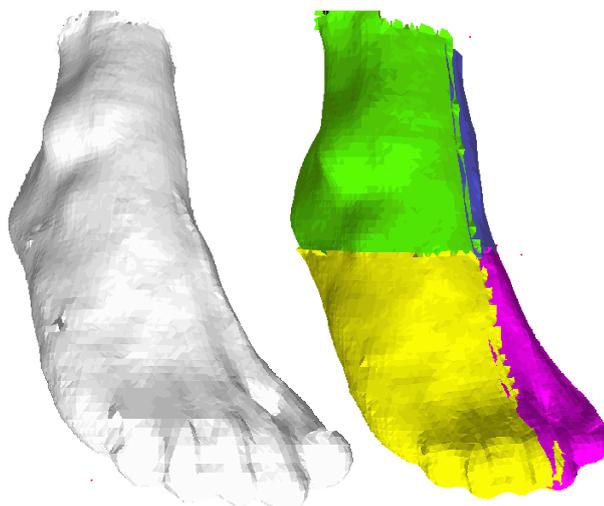


Рис. 4 – Поверхность стопы человека, построенная по единому набору точек (слева) и разделенная на 4 части (справа)

Рассматриваемый метод позволяет существенно повысить сложность моделируемой поверхности, т.к. сложные поверхности представляются большим набором точек, который практически невозможно обработать на одном вычислителе. При этом визуальное

качество результирующей поверхности, полученной рассматриваемым методом, остается примерно таким же, как и поверхности, полученной при обработке сразу всех исходных точек.

Однако следует отметить, что вместе с существенным улучшением времени построения полигональной модели были обнаружены следующие проблемы параллельной обработки данных при построении сеток для сложных поверхностей. Нагрузка различных узлов может существенно отличаться, что влечет за собой простаивание некоторых вычислительных узлов в ожидании завершения работы других узлов. В общем случае сложно выполнить разделение исходных данных на части предназначенные для обработки на узлах так, чтобы все узлы завершили работу одновременно. В общем случае сложно выполнить объединение полученных на каждом узле поверхностей в единую поверхность [10]. В случае если исходные точки располагаются неравномерно, то параметры среднего расстояния между соседними точками для каждого узла могут отличаться, что снижает качество полигональной сетки или требует дополнительного механизма корректировки этого показателя. Дальнейшие исследования необходимо направить на решение вышеуказанных проблем.

### ***Литература***

1. X-ray MicroCT examination of bone specimens [http://www.tecno.ior.it/VRLAB/researchers/repository/MicroCT/MicroCT\\_filesInfo.htm](http://www.tecno.ior.it/VRLAB/researchers/repository/MicroCT/MicroCT_filesInfo.htm)
2. Эйнджел Эдвард Интерактивная компьютерная графика. Вводный курс на базе OpenGL, 2 изд.: Пер. с англ. – М.: Издательский дом «Вильямс», 2001. – 592 с.: ил.
3. Официальный сайт ЗАО НПП "НАВГЕОКОМ" <http://navgeocom.ru>
4. M.Callieri, P.Cignoni, F.Ganovelli and so on "Visualization and 3D Data Processing in the David Restoration" IEEE Computer Graphics and Applications March/April 2004, pp. 16-21.
5. Hoppe H., DeRose T., Duchamp T., McDonald J., Stuetzle W. Surface Reconstruction from Unorganized Points. – University of Washington. – 1992.
6. Lorensen W.E., Cline H.E. Marching cubes: A high resolution 3D surface construction algorithm, Computer Graphics // SIGGRAPH '87 Proceedings. – 1987. – vol. 21, pp. 163-169.
7. Пауков Д. П. Триангуляция Делоне: итеративные алгоритмы построения триангуляции. // Збірка праць магістрантів Донецького національного технічного університету. Випуск 2. – Донецьк: ДонНТУ. – 2003. – с. 461-469.
8. Cohen-Steiner D., Da F. A Greedy Delaunay Based Surface Reconstruction Algorithm, Research Report. – INRIA. – 2002.