Исследование эффективности метода движущихся наименьших квадратов при реконструкции трёхмерной поверхности на суперкомпьютере

Хабибулин Марат Ильдарович

Студент

Факультет ВМК МГУ имени М. В. Ломоносова, Москва, Россия

E-mail: khabibulinmi@my.msu.ru

Научный руководитель *— Никольский Илья Михайлович*

В настоящее время активно развивается область геометрического моделирования и построение 3D моделей по облакам точек с лазерных сенсоров. Одной из базовых задач геометрического моделирования является реконструкция поверхности по облаку точек. Целью настоящей работы является разработка параллельного метода восстановления поверхности, основанного на методе наименьших квадратов, на суперкомпьютере с распределенной памятью, позволяющего добиться оптимальных результатов как в масштабируемости, так и в качестве восстановления поверхности. В центре внимания работы алгоритм восстановления поверхности в основе которого метод наименьших квадратов, в связи с чем алгоритм получил название метод движущихся наименьших квадратов(Moving Least Squares).

Описание алгоритма: На вход алгоритма поступает набор точек $p_i \in R^3, i \in \{1, ..., N\}$ с поверхности. Алгоритм называется движущимся, поскольку итеративно двигается по набору точек. Точка на которой находится итерация называется точкой запроса. Для точки запроса r (см. Рис. 1) вычисляется локальная плоскость $H = \{x \mid \langle n, x\rangle - D = 0, x \in R^3\}, n \in R^3, \parallel n \parallel = 1$ с использованием метода наименьших квадратов по точкам попавшим в окрестность радиуса **R** (параметр алгоритма) так, чтобы минимизировать локальную взвешенную сумму квадратов расстояний точек p_i до плоскости. Веса, прикрепленные к p_i, определяются как функция расстояния от p_i до проекции r на плоскость H, а не от расстояния до r. Предположим, что q является проекцией r на H, тогда H находится путем локальной минимизации 1.

$$\sum_{i=1}^{N} (\langle n, p_i\rangle - D)^2 \theta(\parallel p_i - q \parallel) \tag{1}$$

где θ — гладкая монотонно убывающая функция, положительная на всем пространстве. Затем вычисляется локальная полиномиальная аппроксимация g высот f_i точек p_i над H. В обоих случаях вес для каждого из p_i является функцией расстояния до q. Проекция p_i на g является результатом работы алгоритма MLS.

Параллелизм: Параллельный вариант, модифицированного алгоритма MLS с использованием MPI, описан в **Algorithm 1**. Алгоритм предполагает что облако точек равномерно распределено по всем процессам. Поэтому часть P, доступная локально в процессе u, обозначается через $P^{(u)}$. Через P_l^u, P_r^u обозначены левая и правая граница частей облака точек. Они последовательно получаются от соседних процессов обменами по топологии кольцо. Дополнительных коммуникаций не требуется, а остальные вычисления выполняются локально.

Algorithm 1 Parallel moving least squares with MPI and OpenMP
Input: a data set of points $P = \{p_i\}\ i = 1..n$
Output: a point set surface
1: **for each** process u **do**
2: $P^{(u)} = read(P)$ // each process reads its part of point cloud $P^{(u)} = \{p_j\}\ j = 1..m$
3: $P\_l^{(u)} = send\_recv(P\_r^{(u-1)})$ // getting the left border
4: $P\_r^{(u)} = send\_recv(P\_l^{(u+1)})$ // getting the right border
5: **pragma omp parallel for**
6: **for each** point $j = 1..m$ **do**
7: $H = generate\_plane(p_j)$
8: $g = generate\_local\_polynomial\_approximation(H)$
9: $result\_point = project\_on\_polynom(p_j, polynom)$
10: **end for**

При исследование информационной структуры алгоритма был построен граф информационной зависимости (см. Рис. 2). Из графа видно отсутствие информационной зависимости между точками запроса, поэтому итерации по циклу из локального набора точек $P^{(u)}$ были распараллелены средствами OpenMP(**Algorithm 1:5**). Реализованная гибридная программа показала хорошую масштабируемость и эффективность на суперкомпьютере. Алгоритм обладает большим ресурсом параллелизма из-за своей локальности вычислений, а также отсутсвием проблемы объединения кусков поверхности ввиду точечного представления поверхности(в отличие от сеточного).

Иллюстрации

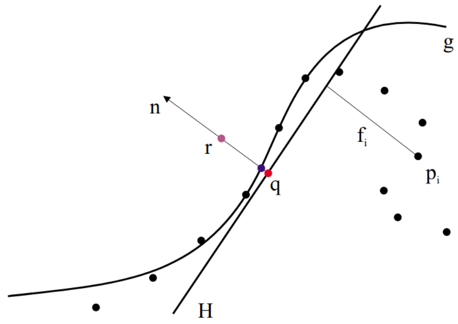

Рис. 1. Шаг алгоритма движущихся наименьших квадратов

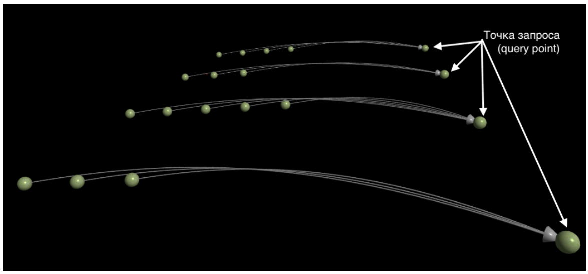

Рис. 2. Граф информационной зависимости MLS

По результатам тестов на реальных поверхностях установлено, что сильными сторонами алгоритма является реконструкция сложных форм и работа с зашумленными входными данными.

В заключение автор выражает благодарность преподавателю кафедры суперкомпьютеров и квантовой информатики факультета ВМК МГУ к.ф.-м.н. И. М. Никольскому за множество ценных замечаний по работе.

Литература

1. ALEXA M., BEHR J., LEVIN D. Computing and rendering point set surfaces //Trans. on Visualization and Computer Graphics (2003), P. 3, 4, 5, 7, 9