



МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ М. В. ЛОМОНОСОВА
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ
КАФЕДРА ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Хабибулин Марат Ильдарович
**Реконструкция поверхности методом движущихся наименьших квадратов на
суперкомпьютере**

КУРСОВАЯ РАБОТА

Научный руководитель:

доцент, к.ф.-м.н.

Никольский Илья Михайлович

С текстом работы ознакомлен

И. М. Никольский

Оглавление

Введение	3
1 Описание последовательного алгоритма	4
2 Существующие алгоритмы	4
2.1 Оператор локально-оптимальной проекции(LOP)	4
2.2 Радиальные базисные функции(RBFs)	6
3 Параллелизм	7
4 Результаты экспериментов	9
5 Результаты работы	13
Список литературы	14

Введение

Проблема определения поверхности по набору точек активно изучается многие годы. Несмотря на распространение методов реконструкции поверхности, многие аспекты проблемы остаются открытыми. Одними из основных трудностей в процессе реконструкции являются сложность формы и шум. Проекционный оператор MLS [Levin 2003] зарекомендовал себя как мощный метод реконструкции поверхности. Метод MLS позволяет достичь простого и эффективного представления. В настоящее время поверхности MLS широко применяются при обработке и рендеринге моделей с точечной выборкой и все чаще используются в качестве стандартного определения поверхностей набора точек. Вычисление точек на поверхности является локальным, что приводит к нестандартной методике, которая может обрабатывать любой набор точек. Помимо реконструкции поверхности функцией сглаживания, у метода движущихся наименьших квадратов есть много других преимуществ, таких как внутренняя способность обрабатывать входные данные с шумом, простота вычисления дифференциальных геометрических свойств поверхности (например, нормали, кривизны).

1. Описание последовательного алгоритма

Пусть точки $p_i \in R^3, i \in \{1, \dots, N\}$, взяты с поверхности S (возможно, с шумом измерения). Цель состоит в том, чтобы спроецировать точку $r \in R^3$ вблизи S на двумерную поверхность SP , аппроксимирующую p_i . Процедура MLS мотивирована дифференциальной геометрией, а именно тем, что поверхность может быть локально аппроксимирована функцией.

Эталонная плоскость: Локальная плоскость $H = \{x \mid \langle n, x \rangle - D = 0, x \in R^3\}, n \in R^3, \|n\| = 1$ вычисляется так, чтобы минимизировать локальную взвешенную сумму квадратов расстояний точек p_i до плоскости (см. рис. 1). Веса, прикрепленные к p_i , определяются как функция расстояния от p_i до проекции q на плоскость H , а не от расстояния до q . Предположим, что q является проекцией r на H , тогда H находится путем локальной минимизации

$$\sum_{i=1}^N (\langle n, p_i \rangle - D)^2 \theta(\|p_i - q\|) \quad (1.1)$$

где θ — гладкая монотонно убывающая функция, положительная на всем пространстве. Полагая $q = r + tn$ для некоторого $t \in R$, уравнение 1.1 можно переписать как:

$$\sum_{i=1}^N (\langle n, p_i - r - tn \rangle)^2 \theta(\|p_i - r - tn\|)$$

Оператор $Q(r) = q = r + tn$ определяется как локальный минимум уравнения с наименьшим t и локальной касательной плоскостью H вблизи q соответственно. Затем локальная эталонная область задается ортонормированной системой координат на H , так что q является началом этой системы.

2. Существующие алгоритмы

2.1. Оператор локально-оптимальной проекции (LOP)

Происхождением метода является алгоритм Вайсфельда для решения задачи Ферма-Вебера о расположении точек, также известный как многомерная медиана L1. Это статистический инструмент, который традиционно применяется во всем мире для многомерных непараметрических точечных выборок, чтобы получить хорошего представителя для большого количества выборок при наличии шума и выбросов. Проблема была впервые известна как проблема оптимального местоположения Вебера [1]. Задача состояла в том, чтобы найти оптимальное место для промплощадки, минимизирующее стоимость доступа. В статистике проблема известна как медиана L1 [2, 3].

Задача Ферма-Вебера (глобальная) о расположении точек рассматривается как пространственная медиана, поскольку, будучи ограничена одномерным случаем, она совпадает с одномерной медианой и наследует некоторые ее свойства в многомерной постановке.

Реконструкция с помощью оператора проекции имеет важное достоинство: она определяет непротиворечивую геометрию на основе точек данных и предоставляет конструктивные

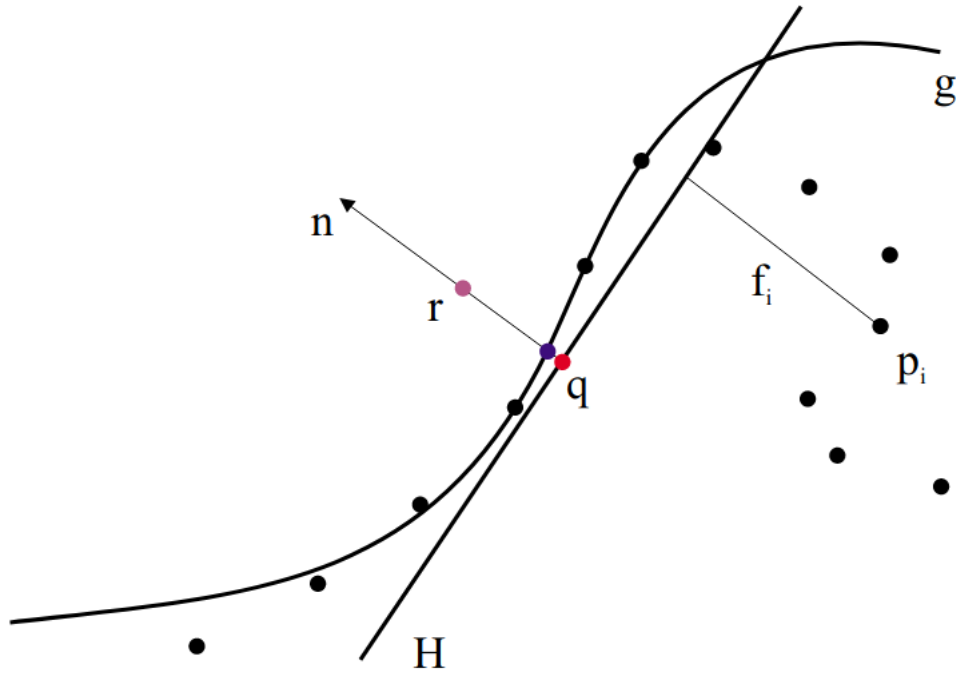


Рис. 1. Процедура проекции MLS: сначала создается локальная эталонная плоскость H для точки p_j . Проекция p_j на H определяет начало координат q . Затем вычисляется локальная полиномиальная аппроксимация g высот f_j точек p_j над H . В обоих случаях вес для каждого из p_j является функцией расстояния до q . Проекция p_j на g является результатом работы алгоритма.

средства для повышения ее дискретизации. Оператор локально-оптимальной проекции без параметризации использует более примитивный механизм проецирования, но поскольку он не основан на локальной 2D-параметризации, он более надежен и хорошо работает в сложных сценариях. Кроме того, если точки данных взяты локально с гладкой поверхности, оператор обеспечивает аппроксимацию второго порядка, что приводит к правдоподобной аппроксимации выбранной поверхности.

Оператор LOP имеет две непосредственные функции: во-первых, его можно использовать в качестве этапа предварительной обработки для любого другого метода реконструкции более высокого порядка (например, RBF). LOP можно применять к необработанным отсканированным данным для создания чистого набора данных, в качестве средства эффективного уменьшения шума и выбросов, а также для упрощения определения ориентации и топологии локальной поверхности. Во-вторых, его можно использовать для уточнения данного набора данных.

Для множества точек данных $P = \{p_j\}_{j \in J} \subset \mathbf{R}^3$, LOP проецирует произвольное множество точек $X^{(0)} = \{x_i^{(0)}\}_{i \in I} \subset \mathbf{R}^3$ на множество P , где I, J обозначают наборы индексов. Множество спроецированных точек $Q = \{q_i\}_{i \in I}$ определяется так, чтобы оно минимизировало сумму взвешенных расстояний до точек P относительно радиальных весов с центром в том же множестве точек Q . Кроме того, точки Q не должны быть слишком близко друг к

другу. Эта структура индуцирует определение искомым точек Q как решение уравнения с фиксированной точкой

$$Q = G(Q),$$

где

$$G(C) = \operatorname{argmin}_{X=\{x_i\}_{i \in I}} \{E_1(X, P, C) + E_2(X, C)\},$$

$$E_1(X, P, C) = \sum_{i \in I} \sum_{j \in J} \|x_i - p_j\| \theta(\|c_i - p_j\|),$$

$$E_2(X, C) = \sum_{i' \in I} \lambda_{i'} \sum_{i \in I \setminus \{i'\}} \eta(\|x_{i'} - c_i\|) \theta(\|c_{i'} - c_i\|)$$

Здесь $\theta(r)$ — быстро убывающая гладкая весовая функция с компактным опорным радиусом h , определяющая размер радиуса влияния, $\eta(r)$ — другая убывающая функция, штрафующая $x_{i'}$ за то, что они подходят слишком близко к другим точкам, и $\{\lambda_i\}_{i \in I}$ являются уравновешивающими членами, которые обозначены через Λ . В двух словах, термин E_1 заставляет спроецированные точки Q аппроксимировать геометрию P , а член E_2 стремится сохранить справедливое распределение точек Q . Правильные значения Λ могут гарантировать степень аппроксимации второго порядка оператора LOP при условии, что данные отбираются с поверхности C^2 .

2.2. Радиальные базисные функции(RBFs)

Радиальные базисные функции - это хорошо известный метод интерполяции разбросанных данных. Учитывая набор точек с заданными значениями функций, RBFs воспроизводят функции, содержащие высокую степень гладкости, посредством линейной комбинации радиально-симметричных базисных функций. Для реконструкции поверхности метод [4] строит поверхность, находя скалярное поле со знаком, определенное через RBFs, набор нулевых уровней которого представляет поверхность. В частности, они используют глобально поддерживаемые базисные функции $\varphi : R^+ \rightarrow R$. Затем неявная функция Φ может быть выражена как:

$$\Phi(\mathbf{x}) = g(\mathbf{x}) + \sum_j \lambda_j \varphi(\|\mathbf{x} - \mathbf{q}_j\|),$$

где $g(x)$ обозначает (глобально поддерживаемый) полином низкой степени, а базисные функции сосредоточены в узлах $\mathbf{q}_j \in R^3$. Неизвестные коэффициенты j находятся путем задания интерполяционных ограничений значения функции θ при $\mathbf{p}_i \in P$; см. рис. 2. Ограничения вне поверхности необходимы, чтобы избежать тривиального решения $f(\mathbf{x}) = 0$ для $\mathbf{x} \in R^3$. Положительные (соответственно отрицательные) ограничения устанавливаются для точек, смещенных в точке \mathbf{p}_i по \mathbf{n}_i в положительном (соответственно отрицательном) направлении. Интерполяция выполняется путем объединения точек ограничения на поверхности и вне поверхности как множество центров узлов \mathbf{q}_j . Коэффициенты i находятся с помощью плотной линейной системы с n неизвестными, эффективно вычисляемой с помощью быстрых мультипольных методов [4]. Преимущество использования глобально поддерживаемых базисных функций для реконструкции поверхности заключается в том, что результирующая

неявная функция является глобально гладкой. Следовательно, RBFs могут быть эффективными для создания водонепроницаемой поверхности при наличии неравномерной выборки и недостающих данных. Однако, когда входные данные содержат умеренный шум, определение правильного размещения точек вне поверхности может стать сложной задачей (см. рис. 2 справа).

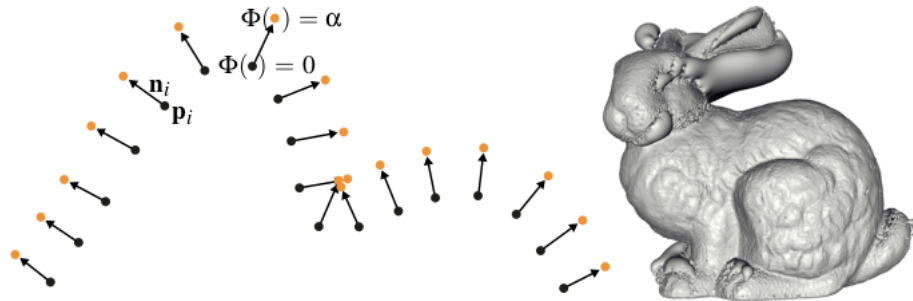


Рис. 2. (слева) Для RBFs оптимизируемое скалярное поле должно оцениваться как нуль в точках выборки $\Phi(\mathbf{p}_i) = 0$, в то время как при ограничениях вне поверхности $\Phi(\mathbf{p}_i + \alpha\mathbf{n}_i) = \alpha$; этот выбор уместен, поскольку функции расстояния со знаком почти везде имеют норму единичного градиента. Кластер выборок вне поверхности показывает, насколько тщательно нужно задавать ограничения в областях с большой кривизной. (справа) Поверхность, реконструированная с помощью RBFs, обычно имеет серьезные геометрические и топологические артефакты, когда предоставляются противоречивые внешние ограничения.

3. Параллелизм

Процедура определения поверхности методом наименьших квадратов была представлена Левином [5]. Параллельный вариант, модифицированного алгоритма MLS с использованием MPI, описан в алгоритме 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 получают от соседних процессов обмены по топологии кольцо. Дополнительных коммуникаций не требуется, а остальные вычисления выполняются локально. В цикле по локальному облаку точек $P^{(u)}$ выполняется процедура проецирования MLS: сначала создается локальная эталонная плоскость H для точки p_j . Проекция p_j на H определяет начало координат q . Затем вычисляется локальная полиномиальная аппроксимация g высот f_j точек p_j над H . В обоих случаях вес для каждого из p_j является функцией расстояния до

q. Проекция p_j на g является результатом процедуры проецирования MLS.

Algorithm 1 Parallel moving least squares with MPI

Input: a data set of points $P = \{p_i\} i = 1..n$

Output: a smooth surface SP (MLS surface) based on the input points

```

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:   for each point  $j = 1..m$  do
6:      $H = generate\_plane(p_j)$ 
7:      $g = generate\_local\_polynomial\_approximation(H)$ 
8:      $result\_point = project\_on\_polynom(p_j, polynom)$ 
9:   end for

```

Параллельный алгоритм

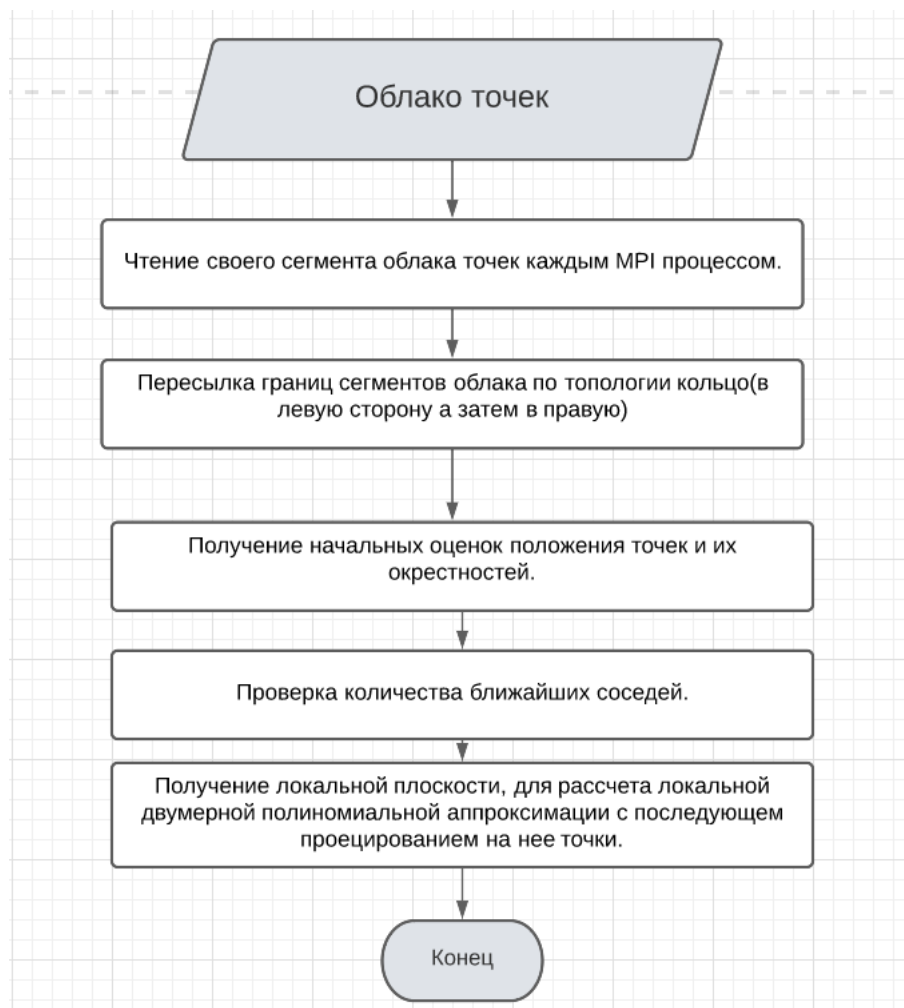


Рис. 3. Блок схема параллельной реализации алгоритма

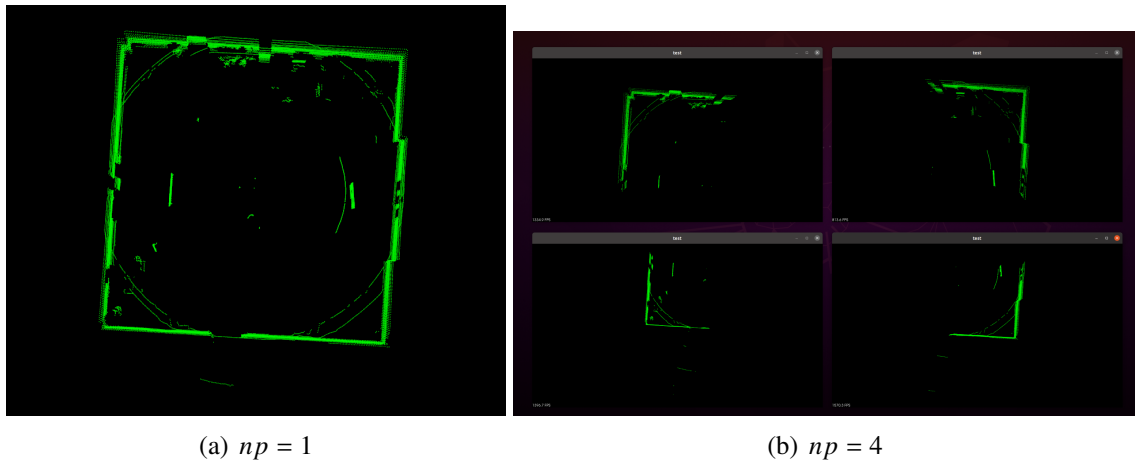


Рис. 4. Распределение сегментов облака точек по процессам

4. Результаты экспериментов

Рассмотрим теперь вопрос об эффективности алгоритма. Для этого напомним, что каждый параллельный алгоритм оценивается по двум параметрам – ускорению S_p и эффективности E_p , которые определяются по формулам:

$$S_p = \frac{t_1}{t_p},$$

$$E_p = \frac{S_p}{p} * 100\%$$

где t_1 - время решения исходной задачи на одном процессоре, t_p - время решения исходной задачи по параллельному алгоритму на p процессорах.

Спецификация системы

11th Gen Intel® Core™ i7-11800H @ 2.30GHz × 16

Число вычислительных ядер: 8

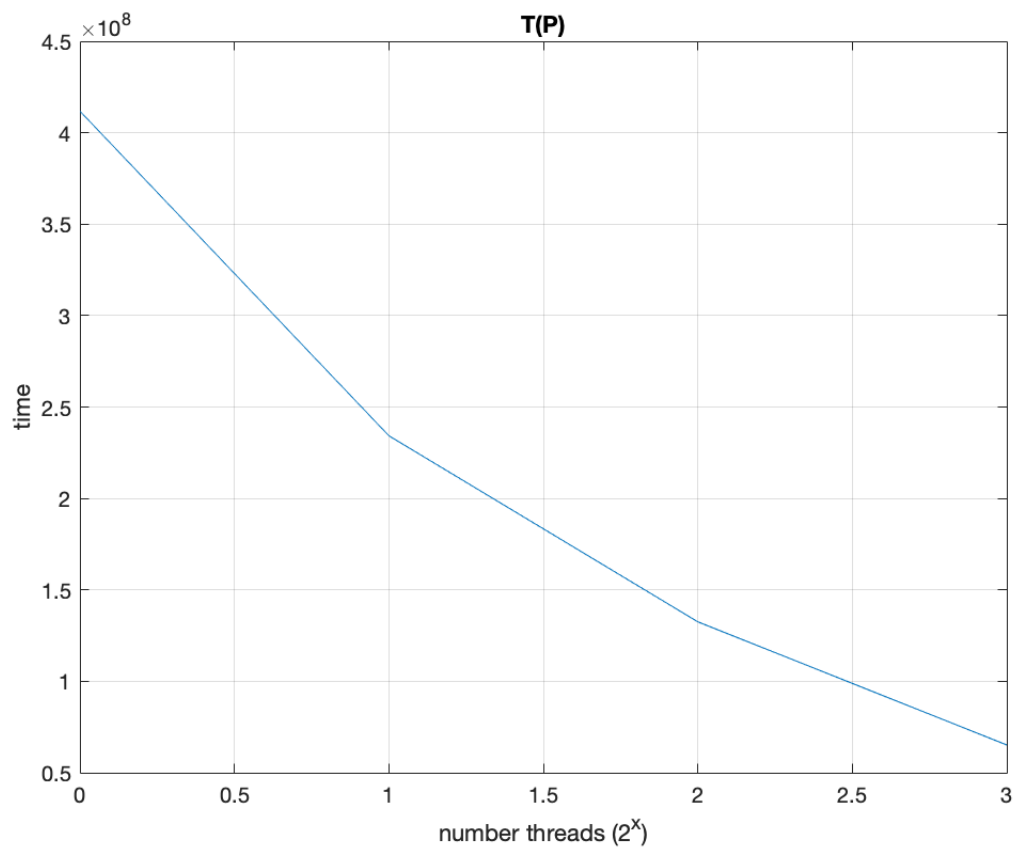


Рис. 5. Время работы (**ns**) на n процессах $n = 1 \dots 8$

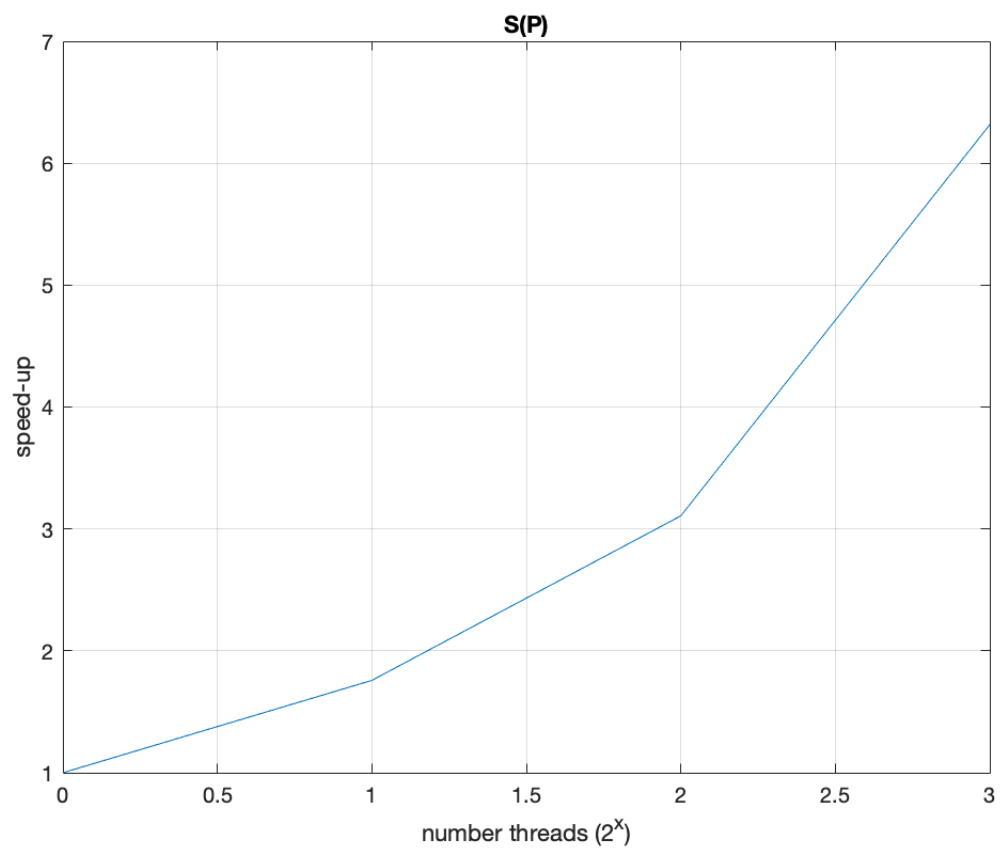


Рис. 6. Ускорение

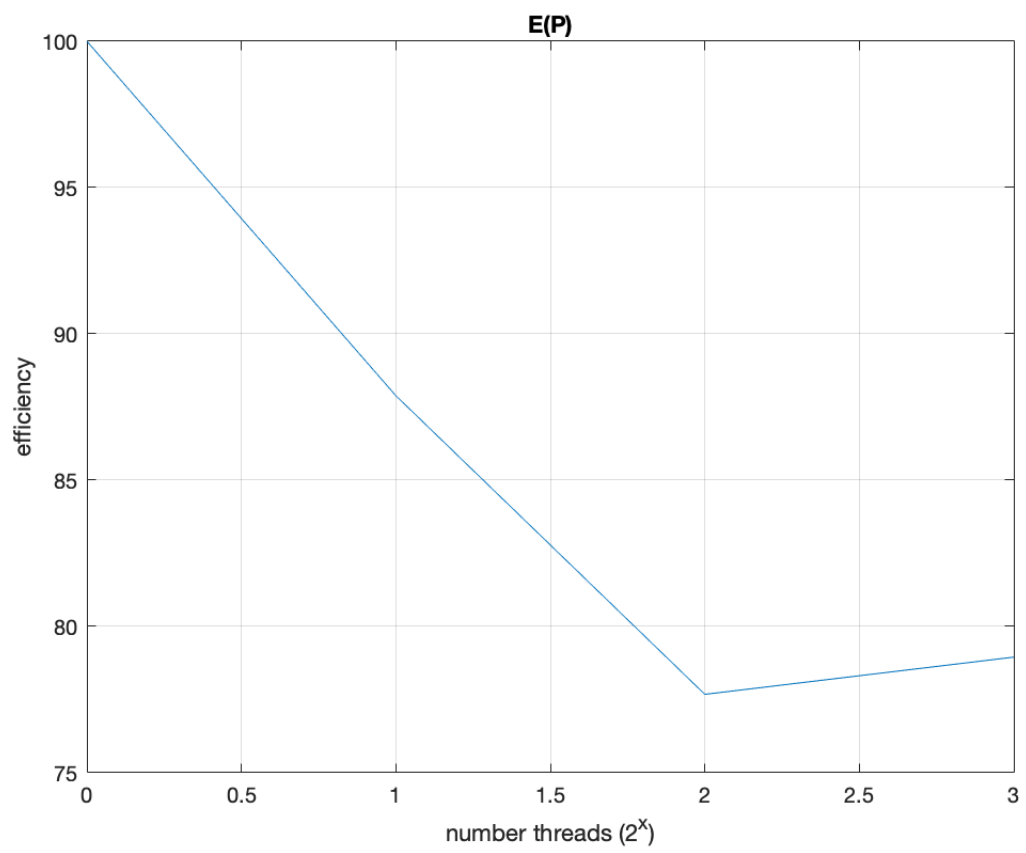


Рис. 7. Эффективность

5. Результаты работы

В работе сформулирован и реализован параллельный алгоритм реконструкции поверхности с использованием программного интерфейса Message Passing Interface для передачи сообщений, завязанный на алгоритме движущихся наименьших квадратов. Алгоритм предусматривает равномерное распределение сегментов облака точек по процессам с последующей пересылкой границ разбиения по топологии кольцо. Все последующие вычисления выполняются локально. Будучи избавленным от последующих пересылок данных, он обеспечивает максимальное ускорение.

Список литературы

1. *WEBER A.* Theory of the Location of Industries // The University of Chicago Press. — 1929.
2. *BROWN B. M.* Statistical uses of the spatial median // J. R. Stat. Soc. — 1983. — С. 25—30.
3. *SMALL C. G.* A survey of multidimensional medians // Int. Stat. rev. 58. — 1983. — С. 263—277.
4. Reconstruction and representation of 3d objects with radial basis functions / J. C. Carr [и др.] // In Proc. of ACM SIGGRAPH. — 2001.
5. *Levin. D.* The approximation power of moving least-squares // Mathematics of Computation. — 1998. — С. 224.