УДК 004.272.2, 004.415.2 Научная статья

https://doi.org/10.32603/2071-8985-2023-16-1-77-84

Эффективные массово-параллельные алгоритмы восстановления данных в многомерном случае

А. Р. Лисс[1], Е. И. Сергеева[1]✉, Г. Ю. Пуеров[2]

[1] Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина), Санкт-Петербург, Россия

[2] АО «Концерн «Океанприбор», Санкт-Петербург, Россия

✉ srgv.lena@gmail.com

Аннотация. В статье рассматривается задача восстановления данных по известным значениям в узлах равномерной сетки в многомерном случае. Предложен эффективный вычислительный алгоритм восстановления данных для модели массового параллелизма. Вычислительный алгоритм основан на преобразовании сумматорного оператора, построенного на базе ядер В. А. Стеклова. Показана зависимость количества операций от параметров задачи. Программная реализация алгоритма выполнена в стандарте Open CL, широко применяемом для вычислений общего назначения на графических процессорах. Приведены результаты работы программы на примере восстановления поверхностей.

Ключевые слова: восстановление данных, параллельные вычисления, массовый параллелизм, Open CL, графические процессоры

Для цитирования: Лисс А. Р., Сергеева Е. И., Пуеров Г. Ю. Эффективные массово-параллельные алгоритмы восстановления данных в многомерном случае // Изв. СПбГЭТУ «ЛЭТИ». 2023. Т. 16, № 1. С. 77–84. doi: 10.32603/2071-8985-2023-16-1-77-84.

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Original article

Efficient Bulk-Parallel Algorithms for Data Recovery in Multidimensional Case

A. R. Liss[1], E. I. Sergeeva[1]✉, G. Yu. Puerov[2]

[1] Saint Petersburg Electrotechnical University, Saint Petersburg, Russia

[2] JSC «Concern «Oceanpribor», Saint Petersburg, Russia

✉ srgv.lena@gmail.com

Abstract. A data recovery problem in multidimensional case by given values in the uniform nodes is considered. An efficient computational data recovery algorithm for a bulk parallelism model is proposed. The algorithm is based on transformations applied to summatory on V. A. Steklov's kernels operator. The dependency between computational complexity and the problem parameters is shown. A software implementation of the algorithm is carried out in Open CL standard widely used for general-purpose computing on graphic processor units. Examples of surfaces reconstruction by the software are given.

Keywords: data recovery, parallel computing, bulk-parallelism, Open CL, graphic processor unit (GPU)

For citation: Liss A. R., Sergeeva E. I., Puerov G. Yu. Efficient Bulk-Parallel Algorithms for Data Recovery in Multidimentional Case // LETI Transactions on Electrical Engineering & Computer Science. 2023. Vol. 16, no. 1. P. 77–84. doi: 10.32603/2071-8985-2023-16-1-77-84.

Conflict of interest. The authors declare no conflicts of interest.

Восстановление данных является составной частью решения различных задач в таких областях, как компьютерная графика, промышленное проектирование, обработка сигналов и многие другие (см., например, [1]–[3]). Представляется актуальным поиск и реализация эффективных параллельных алгоритмов для решения задачи восстановления данных.

Статья посвящена вычислительным аспектам решения задачи восстановления данных в многомерном случае. Рассматривается, как и в предыдущей работе авторов [4], параллельная реализация алгоритма В. В. Жука [5]. Отметим также [6], где исследуются вопросы параллельной реализации похожих алгоритмов восстановления данных.

При проектировании алгоритмов важно выявить те части, которые могут выполняться параллельно, и то, как будут использоваться имеющиеся в распоряжении вычислительные ресурсы. Широко изучаемым в литературе (см., например, [7]–[9]) инструментом для решения поставленных вопросов являются модели параллельных вычислений, которые отражают ключевые особенности некоторого класса целевых архитектур. С другой стороны, возможность эффективной реализации на том или ином классе целевых архитектур зависит от характеристик алгоритма. Поскольку для рассматриваемого алгоритма характерен параллелизм по данным, выбрана модель массового параллелизма, описываемая стандартом Open CL (Open Computational Language) [10].

Обозначения. В дальнейшем $\mathbb{R}, \mathbb{Z}, \mathbb{Z}_+, \mathbb{N}$ – множества вещественных, целых, целых неотрицательных, натуральных чисел соответственно. Если A – некоторое множество, $n \in \mathbb{N}$, то $A^n = A \times \ldots \times A$ (n раз). Если $a \in \mathbb{R}$, то $\mathbf{a}^n = (a, \ldots, a)$. Пусть рассматривается некоторая величина $x \in \mathbb{R}^n$, тогда x_k ($k = 1, \ldots, n$) обозначает k-ю координату. Пусть $a, b \in \mathbb{R}^n$, тогда запись $a \le b$ ($a < b$) означает, что $a_k \le b_k$ ($a_k < b_k$) при каждом $k = 1, \ldots, n$, $ab = (a_1 b_1, \ldots, a_n b_n)$, если $f : \mathbb{R} \to \mathbb{R}$, то $f(a) = (f(a_1), \ldots, f(a_n))$, $\overline{a} = \prod_{i=1}^{n} a_i$, если дополнительно $b_k \ne 0$ ($k = 1, \ldots, n$), то $a/b = (a_1/b_1, \ldots, a_n/b_n)$. Пусть $a \in \mathbb{Z}^n$, $b \in \mathbb{N}^n$, тогда $a \bmod b = (a_1 \bmod b_1, \ldots, a_n \bmod b_n)$. Если $a, b \in \mathbb{Z}^n$, $a \le b$, то $[a : b]$ – множество точек $x \in \mathbb{Z}^n$, удовлетворяющих неравенствам $a \le x \le b$.

Ядро В. А. Стеклова определяется следующим образом (см., например, [5]). Положим при $t \in \mathbb{R}$, $h > 0$, $r \in \mathbb{N}$

$$\psi_r(t) = \begin{cases} \dfrac{1}{(r-1)!} \sum\limits_{0 \le k < \left(|t| + \frac{r}{2}\right)} (-1)^k C_r^k \left(|t| + \dfrac{r}{2} - k \right)^{r-1}, \\ \quad \text{если } |t| \le r/2; \\ 0, \ \text{если } |t| > r/2; \end{cases}$$

$$\psi_{h,r}(t) = \psi_r\left(\frac{t}{h}\right).$$

Если $t \in \mathbb{N}^n$, $h > \mathbf{0}^n$, $r \in \mathbb{N}^n$, то

$$\psi_{h,r}(t) = \prod_{k=1}^{n} \psi_{h_k, r_k}(t_k).$$

Пусть известны значения функции ϕ, принадлежащей некоторому классу гладкости, в узлах $\{u^{(k)} = kh\}$, $k \in [\mathbf{0}^n : K - \mathbf{1}^n]$, $K \in \mathbb{N}^n$, $h > \mathbf{0}^n$, прямоугольной равномерной сетки. Сетку с известными значениями в узлах будем называть *крупной сеткой*. Необходимо приближенно найти неизвестные значения функции ϕ между заданными узлами. Считаем, что узлы $\{v^{(m)} = mt\}$, $m \in [\mathbf{0}^n : M - \mathbf{1}^n]$, $M \in \mathbb{N}^n$, $t > \mathbf{0}^n$, в которых требуется найти значения, также расположены равномерно. Такую систему узлов будем называть *мелкой сеткой*. Будем считать, что крупная сетка является подмножеством мелкой сетки и $N \in \mathbb{N}^n$, таким, что $N - \mathbf{1}^n$ узлов мелкой сетки содержится строго между смежными узлами крупной сетки. Множество из $\overline{N}$ точек мелкой сетки, содержащееся между узлами крупной, будем называть *ячейкой*.

На рис. 1 приведен пример крупной и мелкой сеток для случая $n = 2$.

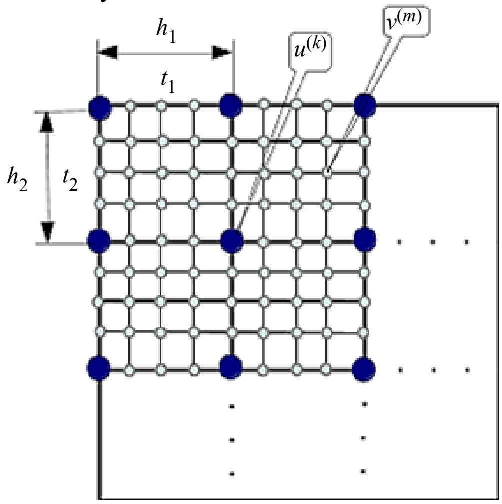

Рис. 1. Крупная и мелкая сетки
Fig. 1. Coarse and fine nets

Алгоритм восстановления данных. Восстановление значения в произвольном узле x мелкой сетки функции ϕ, удовлетворяющей определенным условиям гладкости, выполняется следующим образом [5]:

$$g_{h,r+\mathbf{2}^n}(\phi,x)= \\ =\sum_{k\in\Delta_{K,r}}\phi\left(u^{(k)}\right)\psi_{h,r+\mathbf{2}^n}\left(x-u^{(k)}\right), \quad (1)$$

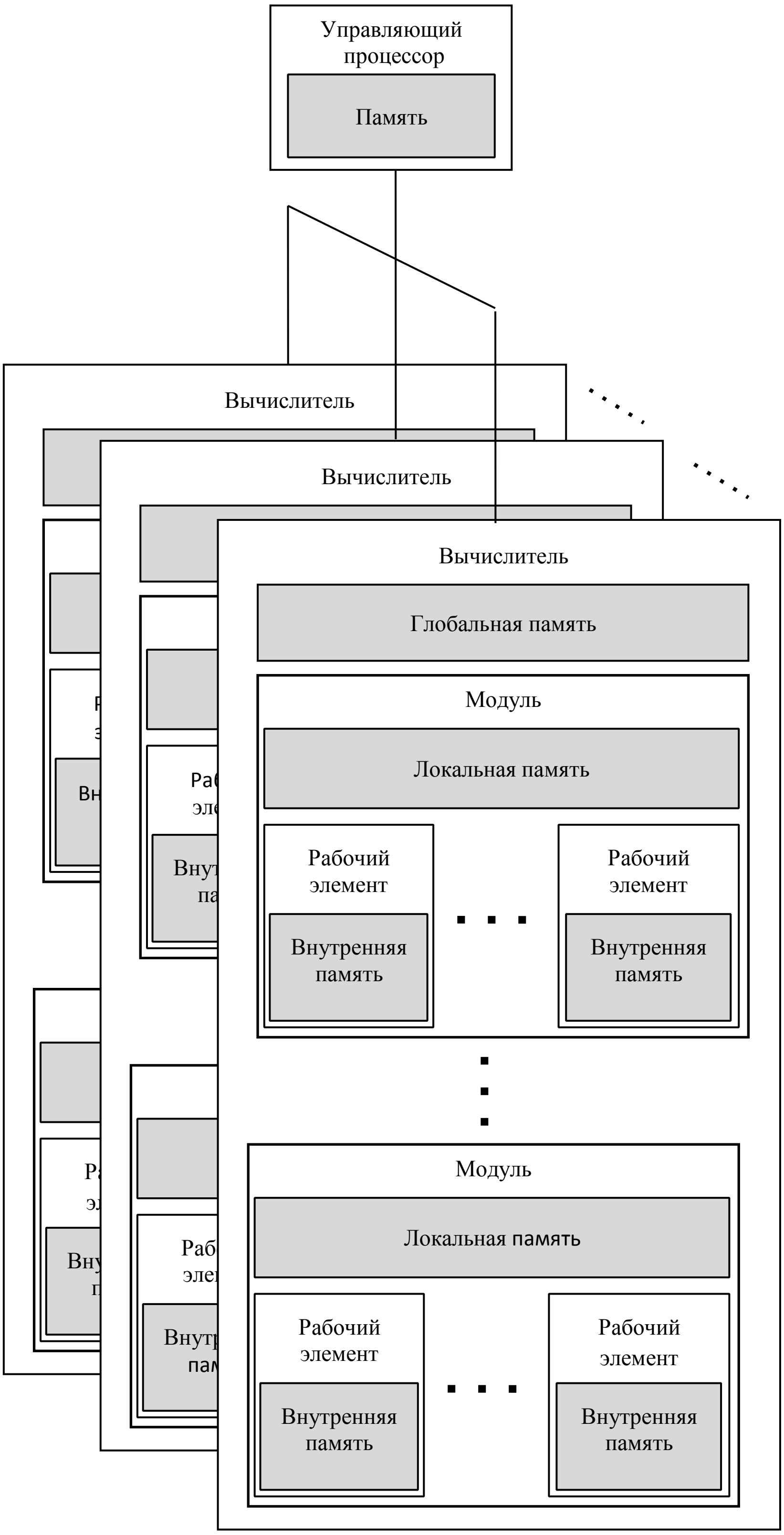

Рис. 2. Модель гетерогенной вычислительной системы
Fig. 2. Model of heterogeneous computing system

где $\Delta_{K,r} = \left[-\lfloor (r+\mathbf{1}^n)/2 \rfloor : K - \mathbf{1}^n + \lfloor (r+\mathbf{1}^n)/2 \rfloor\right]$, $r \in \mathbb{Z}_+^n$.

Из (1) следует, что для вычислений могут быть нужны дополнительные узлы по краям исходной сетки. Способы выбора значений в этих узлах зависят от специфики задачи, для которой применяется (1), и в данной работе не рассматриваются.

Схема вычислений в модели Open CL. Open CL – открытый и широко применяемый стандарт для вычислений общего назначения в гетерогенных (неоднородных) системах (см., напр., [11]–[13]). Одним из известных примеров гетерогенных систем служат системы, содержащие в своем составе графические процессоры (GPU – Graphic Processor Unit). Такие системы, в силу архитектурной специфики, хорошо подходят для эффективной реализации задач, обладающих массовым параллелизмом. Вычислительные системы с GPU в настоящее время находят применение не только для графического рендеринга, но и для эффективного выполнения различных вычислений общего назначения [12]–[14].

Модель гетерогенной вычислительной системы в стандарте Open CL состоит из управляющего процессора (Host) и вычислителей (Compute Device), содержащих, в свою очередь, модули (Compute Unit), которые включают в себя рабочие элементы (Work Item). Представление вычислительной платформы в Open CL показано на рис. 2.

Иерархия памяти вычислителей показана на рис. 2. Всем рабочим элементам доступна глобальная память. Через глобальную память также происходит обмен данными с памятью управляющего процессора. Каждый модуль обладает локальной памятью, к которой имеют доступ его рабочие элементы. Каждый рабочий элемент имеет внутреннюю память (регистры), доступную только ему.

Исполняемая программа, разработанная в стандарте Open CL, содержит вычислительные ядра (kernels), выполняющиеся параллельно в режиме SIMD на рабочих элементах, и управляющую программу, которая определяет контекст и управляет запуском вычислительных ядер.

Все рабочие элементы организованы в Open CL в рабочее пространство – n-мерную решетку, размер которой обозначим $\mathrm{NDRange} \in \mathbb{N}^n$. Помимо этого, решетка разбивается на рабочие группы, их размер обозначим $\mathrm{NDWorkGroup} \in \mathbb{N}^n$.

Для каждого рабочего элемента определены:

• $\mathrm{GlobalID} \in [0 : \mathrm{NDRange} - \mathbf{1}^n]$ – индекс вычислительного элемента;

• $\mathrm{LocalID} \in [0 : \mathrm{NDWorkGroup} - \mathbf{1}^n]$ – индекс вычислительного элемента в рабочей группе.

На рис. 3 приведен пример двумерного рабочего пространства.

Описание параллельного вычислительного алгоритма восстановления данных. Введем следующие обозначения:

• F – массив известных значений функции ϕ в узлах крупной сетки;

• $G_{r+\mathbf{2}^n}$ – массив значений аппроксимирующей функции в узлах мелкой сетки, вычисленных по формуле (1);

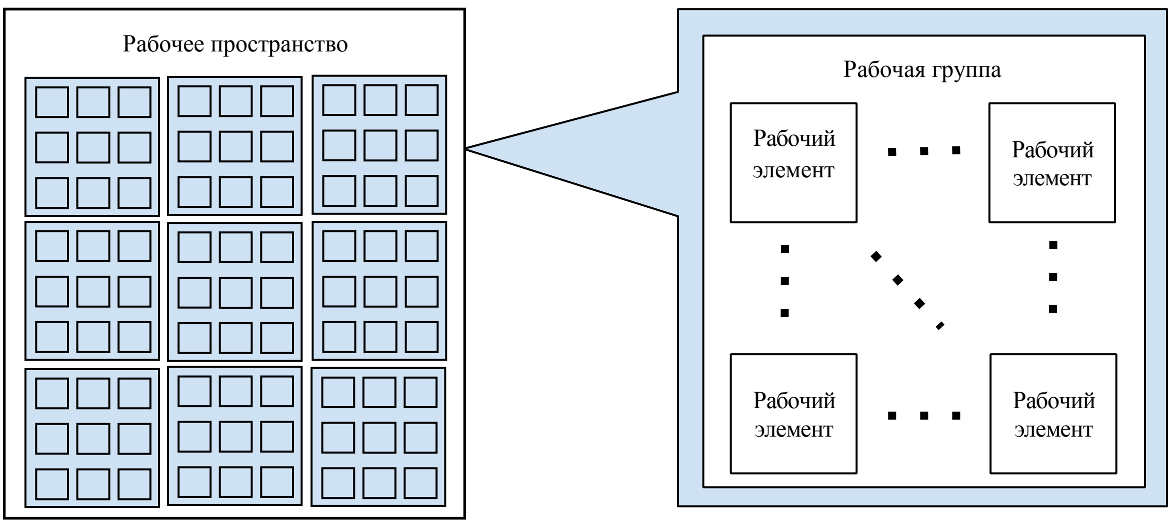

Рис. 3. Двумерное рабочее пространство в модели Open CL
Fig. 3. Two-dimensional work range in Open CL

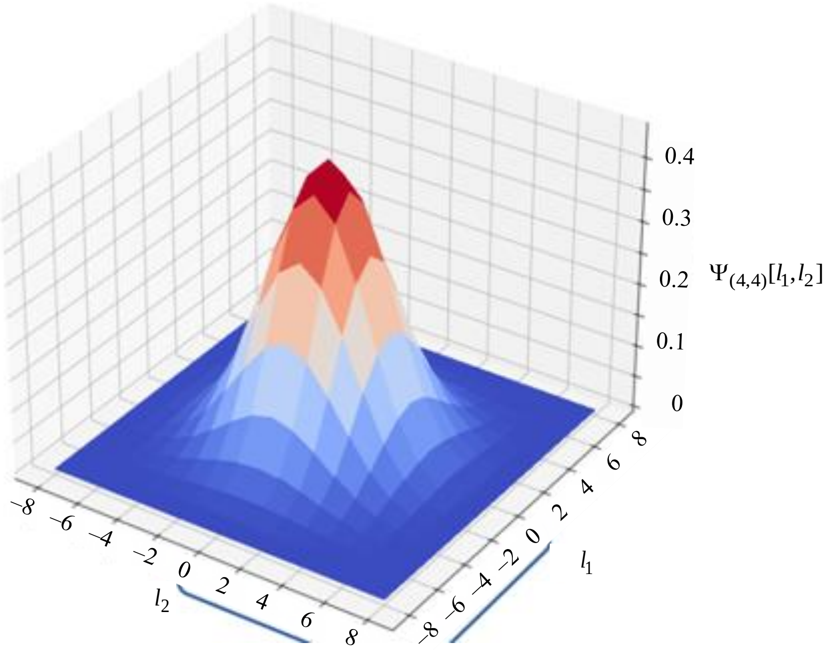

Рис. 4. Ядро при r = (2,2) и N = (4,4)
Fig. 4. Kernel for r = (2,2) and N = (4,4)

- $\Psi_{r+\mathbf{2}^n}$ – массив значений ядер В. А. Стеклова порядка $r + \mathbf{2}^n$.

Пример ядра порядка (4, 4) для N = (4, 4) представлен на рис. 4.

Далее приведены выражения для формирования искомых значений в узлах мелкой сетки. Пусть $m \in M$, $r/2 \in \mathbb{Z}_+^n$, тогда

$$G_{r+\mathbf{2}^n}[m] = \sum_{i\in[\mathbf{0}^n:r/2]} \sum_{p\in P} F\left[\lfloor m/N \rfloor + ip + s(p)\right] \times \\ \times \Psi_{r+\mathbf{2}^n}\left[| m \operatorname{mod} N - (ip + s(p))N |\right], \qquad (2)$$

где $P = \{p \in \mathbb{Z}^n : p_l \in \{-1,1\},\ l = 1,\dots,n\}$, $s(t) = (t+1)/2$ ($t \in \mathbb{R}$).

При вычислениях по формуле (2) требуется $\overline{M}\overline{(r/2+\mathbf{1}^n)}2^n$ умножений и $\overline{M}\left[\overline{(r/2+\mathbf{1}^n)}2^n - 1\right]$ сложений.

Заметим, что в (2) каждое произведение известного значения в узле крупной сетки и значений ядер в узлах мелкой сетки используется несколько раз при формировании результатов в нескольких соседних ячейках крупной сетки. Используя этот факт, можно сократить количество операций. Предложена следующая двухэтапная вычислительная схема.

1. Вычисление для k-го, $k \in [\mathbf{0}^n : K - \mathbf{1}^n]$, известного значения функции элементов массива произведений $\Pi_r[k,l] = F[k]\Psi_{r+\mathbf{2}^n}[l]$, ($l \in [\mathbf{0}^n : (r/2+\mathbf{1}^n)N]$). Выполняется $\overline{K\left[N(r/2+\mathbf{1}^n)\right]}$ умножений.

2. Выбор и суммирование полученных на предыдущем шаге произведений для формирования результирующих значений в узлах мелкой сетки. Выполняется $\overline{M}\left[\overline{(r/2+\mathbf{1}^n)}2^n - 1\right]$ сложений.

Таким образом, за счет переиспользования произведений при применении вычислительной схемы количество умножений сокращается почти в 2^n раз.

Опишем программную реализацию параллельного вычислительного алгоритма восстановления данных в стандарте Open CL в двумерном случае. Из памяти управляющего процессора в глобальную память вычислителя загружаются следующие данные:

– массив значений в узлах крупной сетки;

– предварительно инициализированный массив значений ядра порядка $r + \mathbf{2}^2$.

Результирующие значения в узлах мелкой сетки выгружаются управляющим процессором из глобальной памяти вычислителя по завершении работы вычислительных ядер на рабочих элементах.

Программная реализация рассмотренного алгоритма в двумерном случае включает в себя 2 вычислительных ядра, соответствующих этапам схемы вычислений. Первое вычислительное ядро (`kernel_prod`) выполняется в рабочем пространстве размером K, для второго ядра

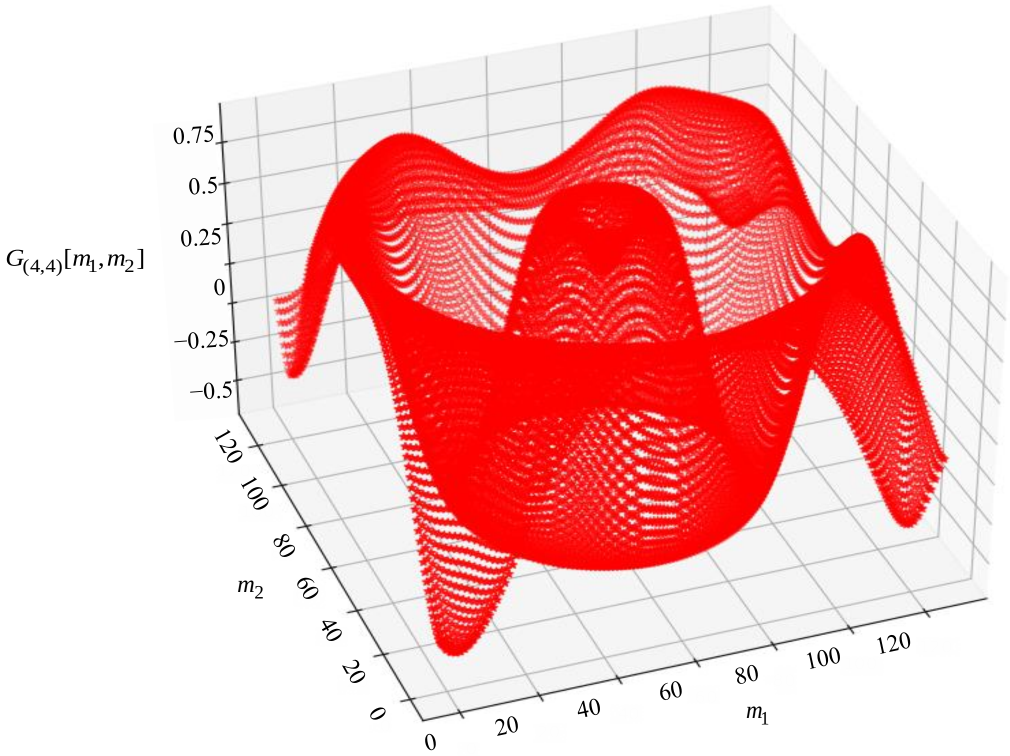

Рис. 5. Пример восстановленной поверхности
Fig. 5. Example of recovered surface

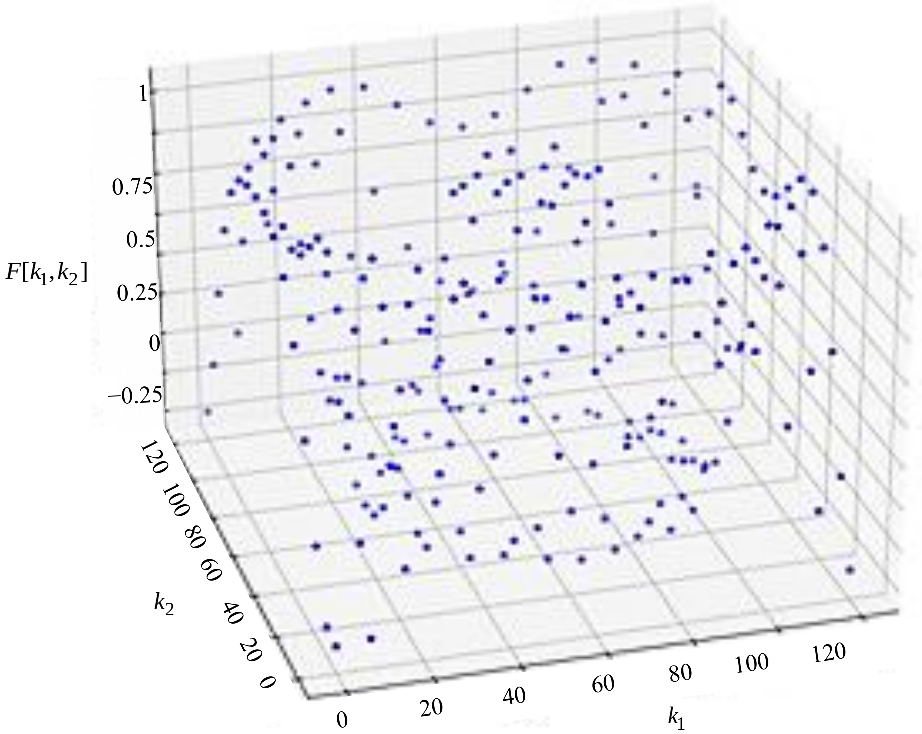

Рис. 6. Исходные данные для восстановления
Fig. 6. Raw data for recovering

(`kernel_sum`) удобно выбрать рабочее пространство размера *M*. Далее приведено описание разработанных вычислительных ядер.

`kernel_prod`

Выполняется в каждой точке крупной сетки. Каждый рабочий элемент получает значение в точке крупной сетки из глобальной памяти по индексу *globalID*.

На выходе в глобальной памяти формируются массивы произведений в ячейках мелкой сетки, соответствующие каждой точке крупной сетки.

`kernel_sum`

Выполняется в каждой точке мелкой сетки. Получает из глобальной памяти массив произведений.

Каждый рабочий элемент записывает полученное в соответствии с (2) значение в своей точке мелкой сетки в глобальную память по индексу $globalID$.

Пример восстановления поверхности при $r = (2, 2)$, $N = (4, 4)$ представлен на рис. 5. Исходные данные при $K = (16, 16)$ – на рис 6.

В статье предложена схема вычислений для алгоритмов восстановления данных в узлах равномерной сетки в многомерном случае, которая позволяет сократить количество арифметических операций. Оптимизация достигнута за счет преобразования сумматорного оператора, построенного на базе ядер В. А. Стеклова, к виду, удобному для выполнения вычислений в модели массового параллелизма. Показан выигрыш в количестве операций в зависимости от параметров задачи. Корректная работа описанной программной реализации двумерного случая подтверждена результатами тестирования на восстановлении поверхностей.

Участие в исследованиях канд. физ.-мат. наук Г. Ю. Пуерова поддержано грантом РНФ № 18-11-000555.

Список литературы

1. Квасов Б. И. Методы изогеометрической аппроксимации сплайнами. М.: Ижевск, 2006.
2. Uhlir K., Skala V. Radial basis function use for the restoration of damaged images // Computer Vision and Graphics. Springer, 2006. P. 839–844.
3. Sloan P.-P., Rose C., Cohen M. Shape by example // SI3D'01: Proc. of the 2001 symp. on Interactive 3D graphics. New York, 2001. P. 135–143.
4. Лисс А. Р., Пуеров Г. Ю., Сергеева Е. И. Эффективные параллельные алгоритмы восстановления данных по известным значениям в узлах // Изв. СПбГЭТУ «ЛЭТИ». 2021, № 6. С. 17–23.
5. Жук В. В. Методические указания к курсу «Теория аппроксимации функций и ее приложения». Ч. 2. СПб.: Изд-во Петербургского ун-та, 1993.
6. Масальских А. В. Параллельный алгоритм одного метода восстановления табличных данных // Изв. Тульского гос. ун-та. Естественные науки. 2014. № 3. С. 167–177.
7. Valiant L. G. A bridging model for parallel computation // Communications of the ASM. 1990. № 33 (8). P. 103–111.
8. Buurlage J.-W., Bannink T., Wits A. Bulk-synchronous pseudo-streaming algorithms for many-core accelerators. 2016. URL: arXiv.org/abs/1608.07200 (дата обращения 20.09.2022).
9. Valiant L. G. A bridging model for multi-core computing // J. of Comp. and Syst. Sci. 2008. Т. 77. P. 154–166.
10. Open CL. Open Standard for Parallel Programming of Heterogeneous Systems. URL: www.khronos.org/api/opencl (дата обращения 18.10.2022).
11. Богданов П. Б., Сударева О. Ю. Гетерогенное программирование в рамках стандарта OpenCL // Тр. XV Междунар. конф. «Супервычисления и математическое моделирование». Саров: ФГУП «РФЯЦ-ВНИИЭФ», 2015. С. 123–137.
12. Бурцев А. А. Оптимизация операции перемножения матриц на основе технологии Open CL // Тр. НИИСИ РАН. 2020. Т. 10, № 5. С. 100–112.
13. Богданов А. В., Дегтярев А. Б., Храмушин В. Н. Высокопроизводительные вычисления на гибридных системах: будут ли решены «задачи большого вызова»? // Компьютерные исследования и моделирование. 2015. Т. 7, № 3. С. 429–437.
14. Navarro C., Hitschfeld N., Mateu L. A Survey on parallel computing and its applications in data-parallel problems using GPU architectures // Communications in Computational Physics. 2013. Т. 15. P. 285–329.

Информация об авторах

Лисс Александр Рудольфович – д-р техн. наук, профессор СПбГЭТУ.
E-mail: alexander.liss@mail.ru

Сергеева Елена Игоревна – канд. техн. наук, ассистент каф. МОЭВМ, СПбГЭТУ «ЛЭТИ».
E-mail: srgv.lena@gmail.com

Пуеров Георгий Юрьевич – канд. физ.-мат. наук, ведущий научный сотрудник, АО «Концерн «Океанприбор».
E-mail: puerov@gmail.com

References

1. Kvasov B. I. Metody izogeometricheskoj approksimacii splajnami. M.: Izhevsk, 2006. (In Russ.).

2. Uhlir K., Skala V. Radial Basis function use for the restoration of damaged images // Computer Vision and Graphics. Springer, 2006. P. 839–844.

3. Sloan P.-P., Rose C., Cohen M. Shape by example // SI3D'01: Proc. of the 2001 symp. on Interactive 3D graphics. New York, 2001. P. 135–143.

4. Liss A. R., Puerov G. Ju., Sergeeva E. I. Jeffektivnye parallel'nye algoritmy vosstanovlenija dannyh po izvestnym znachenijam v uzlah // Izv. SPbGJeTU «LJeTI». 2021. № 6. S. 17–23. (In Russ.).

5. Zhuk V. V. Metodicheskie ukazanija k kursu «Teorija approksimacii funkcij i ee prilozhenija». Ch. 2. SPb.: Izd-vo Peterburgskogo un-ta, 1993. (In Russ.).

6. Masal'skih A. V. Parallel'nyj algoritm odnogo metoda vosstanovlenija tablichnyh dannyh // Izv. Tul'skogo gos. un-ta. Estestvennye nauki. 2014. № 3. S. 167–177. (In Russ.).

7. Valiant L. G. A bridging model for parallel computation // Communications of the ASM. 1990. № 33 (8). P. 103–111.

8. Buurlage J.-W., Bannink T., Wits A. Bulk-synchronous pseudo-streaming algorithms for many-core accelerators. 2016. URL: arXiv.org/abs/1608.07200 (data obrashchenia 20.09.2022).

9. Valiant L. G. A bridging model for multi-core computing // J. of Comp. and Syst. Sci. 2008. T. 77. P. 154–166.

10. Open CL. Open Standard for Parallel Programming of Heterogeneous Systems. URL: www.khronos.org/api/opencl (data obrashchenia 18.10.2022).

11. Bogdanov P. B., Sudareva O. Ju. Geterogennoe programmirovanie v ramkah standarta OpenCL // Tr. XV Mezhdunar. konf. «Supervychislenija i matematicheskoe modelirovanie». Sarov: FGUP «RFJaC-VNIIJeF», 2015. S. 123–137. (In Russ.).

12. Burcev A. A. Optimizacija operacii peremnozhenija matric na osnove tehnologii Open CL // Tr. NIISI RAN. 2020. T. 10, № 5. S. 100–112. (In Russ.).

13. Bogdanov A. V., Degtjarev A. B., Hramushin V. N. Vysokoproizvoditel'nye vychislenija na gibridnyh sistemah: budut li resheny «zadachi bol'shogo vyzova»? // Komp'juternye issledovanija i modelirovanie. 2015. T. 7, № 3. S. 429–437. (In Russ.).

14. Navarro C., Hitschfeld N., Mateu L. A Survey on parallel computing and its applications in data-parallel problems using GPU architectures // Communications in Computational Physics. 2013. T. 15. P. 285–329.

Information about the authors

Alexander R. Liss – Dr Sci. (Eng.), professor of the Saint Petersburg Electrotechnical University.
E-mail: alexander.liss@mail.ru

Elena I. Sergeeva – Cand. Sci. (Eng.), Assistant of the Department of Engineering and Computer Application of Saint-Petersburg Electrotechnical University.
E-mail: srgv.lena@gmail.com

Georgiy Yu. Puerov – Cand. Sci. (Phis.-Math.), Senior Scientist of JSC «Concern «Oceanpribor».
E-mail: puerov@gmail.com

Статья поступила в редакцию 22.10.2022; принята к публикации после рецензирования 29.11.2022; опубликована онлайн 30.01.2023.

Submitted 22.10.2022; accepted 29.11.2022; published online 30.01.2023.