

## МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ, ЧИСЛЕННЫЕ МЕТОДЫ И КОМПЛЕКСЫ

УДК 51-37; 004.021; 004.4'6

### ПОСТРОЕНИЕ 3D-МОДЕЛИ ВНУТРЕННИХ ОРГАНОВ ЧЕЛОВЕКА С ПРИМЕНЕНИЕМ СУПЕРКОМПЬЮТЕРНЫХ ВЫЧИСЛЕНИЙ<sup>1</sup>

Статья поступила в редакцию 28.12.2018, в окончательном варианте – 11.02.2019.

**Курочкин Михаил Александрович**, Санкт-Петербургский политехнический университет Петра Великого, 195251, Российская Федерация, г. Санкт-Петербург, ул. Политехническая, 29, кандидат технических наук, доцент, e-mail: kurochkin.m@gmail.com

**Касимов Эрнест Махмадкулович**, Санкт-Петербургский политехнический университет Петра Великого, 195251, Российская Федерация, г. Санкт-Петербург, ул. Политехническая, 29, студент, e-mail: kasimov.ernest@gmail.com

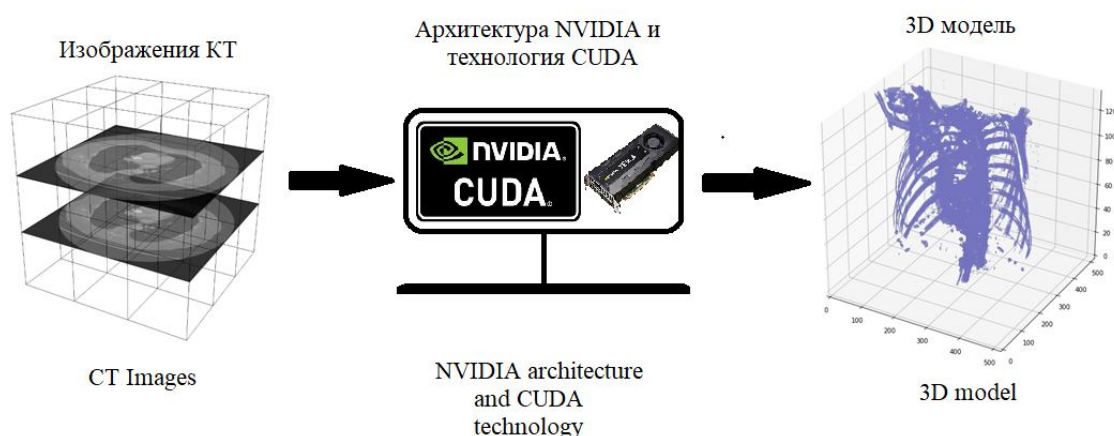
**Пурий Александр Александрович**, Санкт-Петербургский политехнический университет Петра Великого, 195251, Российская Федерация, г. Санкт-Петербург, ул. Политехническая, 29, студент, e-mail: zloypur@gmail.com

**Востров Алексей Владимирович**, Санкт-Петербургский политехнический университет Петра Великого, 195251, Российская Федерация, г. Санкт-Петербург, ул. Политехническая, 29, старший преподаватель, e-mail: alex.sinkriver@gmail.com

В современной медицине компьютерная томография является универсальным средством изучения внутреннего строения организма. На основании трехмерной модели формируются срезы, которые впоследствии необходимо визуализировать. В статье рассмотрены вопросы распараллеливания задачи визуализации данных рентгеновской компьютерной томографии на примере 3D-модели грудной клетки. Для решения поставленной задачи выбран алгоритм маршевых кубов, который позволяет использовать как последовательную, так и параллельную реализацию обработки данных. Последовательная реализация выполняется с применением CPU, в то время как параллельная – использует вычисления на GPU архитектуры Nvidia с применением технологии CUDA. Проведен анализ зависимости времени выполнения приложения от различных параметров: способа реализации (CPU-GPU), количества обрабатываемых данных на GPU, настройки параметров ядра для GPU. В результате проведена визуализация сечений для 3D-моделей грудной клетки, построенных по разному количеству исходных точек; оценены длительности работы различных алгоритмов решения этой задачи.

**Ключевые слова:** компьютерная томография, CUDA, параллельные вычисления, 3D-визуализация данных, формат DICOM, GPU, CPU, шкала Хаунсфилда

#### Графическая аннотация (Graphical annotation)



<sup>1</sup> Работа выполнена при поддержке гранта РФФИ 16-29-04319. Контекстно-ориентированные методы принятия решений и планирования операций смешанной группой мобильных роботов.

CONSTRUCTING 3D MODELS OF HUMAN INTERNAL ORGANS  
USING SUPERCOMPUTER CALCULATIONS

*The article was received by editorial board on 28.12.2018, in the final version – 11.02.2019.*

**Kurochkin Mihail A.**, Peter the Great St. Petersburg Polytechnic University, 29 Politekhnikeskaya St., Saint Petersburg, 195251, Russian Federation

Cand. Sci. (Engineering), e-mail: kurochkin.m@gmail.com

**Kasimov Ernest M.**, Peter the Great St. Petersburg Polytechnic University, 9 Politekhnikeskaya St., Saint Petersburg, 195251, Russian Federation

student, e-mail: kasimov.ernest@gmail.com

**Purii Alexander A.**, Peter the Great St. Petersburg Polytechnic University, 9 Politekhnikeskaya St., Saint Petersburg, 195251, Russian Federation

student, e-mail: zloypur@gmail.com

**Vostrov Alexey V.**, Peter the Great St. Petersburg Polytechnic University, 9 Politekhnikeskaya St., Saint Petersburg, 195251, Russian Federation

Senior Lecturer, e-mail: zloypur@gmail.com

In today's modern medicine, computer tomography is a multipurpose device for diagnosing the condition of human organism. The principles of its work help to produce three-dimensional final images, that consist of separate slices. When we receive the images, they need to be visualized for further application in patient treatment. The article touches upon the parallelization questions of visualization tasks CT data with the help of chest 3D model example. In order to solve the task the marching cubes algorithm is used. It allows to use not only sequential realization, but also the parallel one. The former is performed using CPU, however the latter uses computing on GPU with Nvidia architecture and CUDA technology. A comparative analysis of execution time and different parameters (such as realization method (CPU-GPU), the quantity of processed data on GPU, kernel parameters settings) is carried out. The visualization of chest 3D models, that are constructed in different quantity of initial nodes, is done as a result.

**Key words:** computer tomography, CUDA, parallel computing, 3D data visualization, DICOM, GPU, CPU, Hounsfield scale

**Введение.** Компьютерная томография (КТ) [5] является практически универсальным методом диагностики, позволяющим врачу получить максимально подробную картину состояния внутренних органов. Для правильной диагностики заболевания специалисту необходимо проанализировать трехмерную форму отсканированного органа. При этом чем выше точность построенной модели, тем больше вероятность своевременной постановки правильного диагноза [2]. В настоящее время наиболее используемыми являются спиральные и мультиспиральные КТ, для которых программное обеспечение визуализации данных сканирования поставляется в комплекте с этим медицинским оборудованием или может быть приобретено впоследствии в качестве дополнительных программных модулей. Также разработаны специальные приложения для отображения результатов КТ, которые реализованы в рамках технологии «Клиент-сервер» [9]. В то же время актуальной остается задача доступа к этим данным для врачей-диагностов, не обладающих непосредственным доступом к этому дорогостоящему оборудованию. Категория этих специалистов работает на типовых персональных компьютерах и обычно не имеет доступа к специализированным медицинским серверам, которые бы могли обеспечить работу необходимых средств динамической визуализации объектов в различных ракурсах. В первую очередь это связано с необходимостью обработки в реальном времени более чем миллиарда точек для трехмерных представлений визуализируемых объектов. Разработанный алгоритм позволяет запускать программное средство визуализации данных на любой системе как с использованием CPU, так и при помощи GPU с технологией CUDA [1, 4, 14]. При этом актуальны вопросы обеспечения вычислительной эффективности алгоритмов, которые в существующей литературе рассмотрены недостаточно комплексно. Поэтому цель работы – исследовать варианты распараллеливания алгоритма построения 3D-модели отсканированного органа в реальном масштабе времени.

**Общая характеристика проблематики работы и ее содержания.** Результаты сканирования для распространенного медицинского томографического оборудования фиксируются в формате DICOM 3.0, базовая структура которого приведена ниже. Файл DICOM 3.0 – это объектно-ориентированный файл с теговой организацией [10, 13], который описывает следующие параметры:

- атрибуты пациента;
- модель аппарата, на котором было проведено обследование;
- атрибуты изображения в серии;
- атрибуты, которые определяют передаваемый массив пикселей двумерной плоскости изображения;
- модуль пикселей изображений;
- определение объектной информации компьютерной томографии.

Непосредственно синтезированные изображения хранит модуль пикселей изображений. Это набор двумерных проекций (рис. 1), которые можно рассматривать как параллельные сечения.

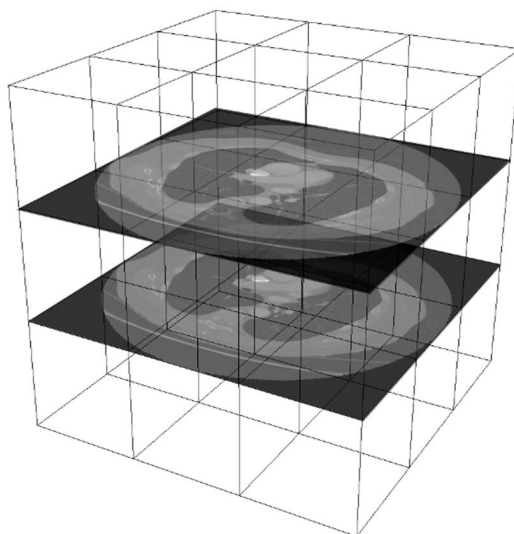


Рисунок 1 – Два смежных скана легких

Результатом обработки этих проекций будет полигональное представление формы отсканированного органа. Сравним два подхода к решению задачи: построение модели ресурсами CPU и построение модели в архитектуре Nvidia технологии CUDA [12].

В качестве метода решения задачи был выбран метод маршевых кубов [7], позволяющий рассматривать два смежных скана независимо от других. Этот метод обеспечивает возможность максимального распараллеливания процесса построения элементарных полигонов (рис. 2).

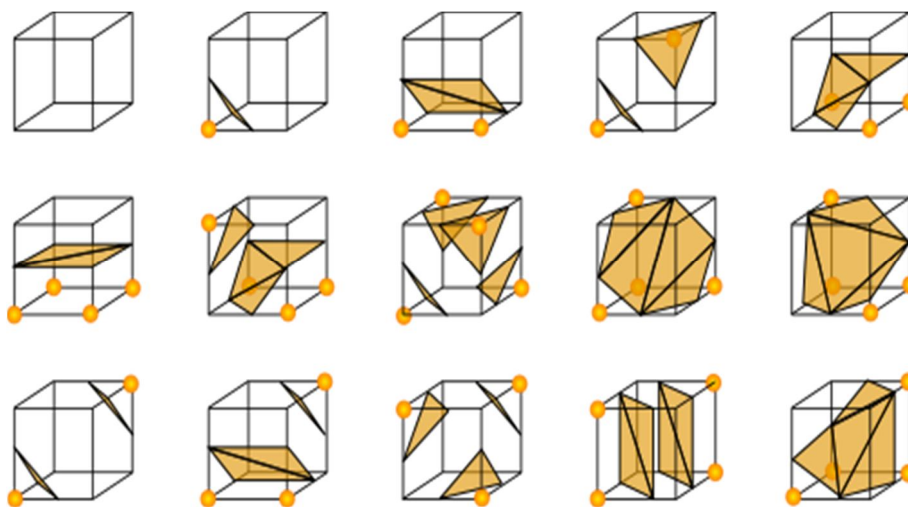


Рисунок 2 – Пример построения элементарных полигонов методом маршевых кубов

#### Работа алгоритма на CPU.

1. На этапе инициализации происходит считывание изображений в формате DICOM в память и сортирование срезов по параметру  $z$ , указывающему положение слайса (сечения) в объемном представлении.
2. Приведение каждого значения тега к значениям по шкале Хаунсфилда [6] осуществляется по формуле:

где  $H$  – значение по шкале Хаунсфилда;  $I$  – исходное значение;  $b$  – значение тега (0x0028, 0x1052) Rescale Intercept;  $m$  – значение тега (0x0028, 0x1053) Rescale Slope [11]. Результат перевода (приведения) сохраняется в трехмерный массив.

3. Полученные на втором этапе значения были отфильтрованы по критерию соответствия изображению грудной клетки [8]. При этом по шкале Хаунсфилда кости имеют значения не меньше четырех-

сот единиц. Следовательно, все элементы изображения с параметрами, большими либо равными этому значению, соответствуют изображениям кости.

4. С применением алгоритма маршевых кубов осуществляется поиск координат вершин, через которые будут проходить все элементарные полигоны.

5. Применяется алгоритм маршевых кубов и на основе него выполняется вычисление координат всех точек элементарных полигонов.

Блок-схема преобразования исходных данных к критерию соответствия грудной клетке приведена на рисунке 3.

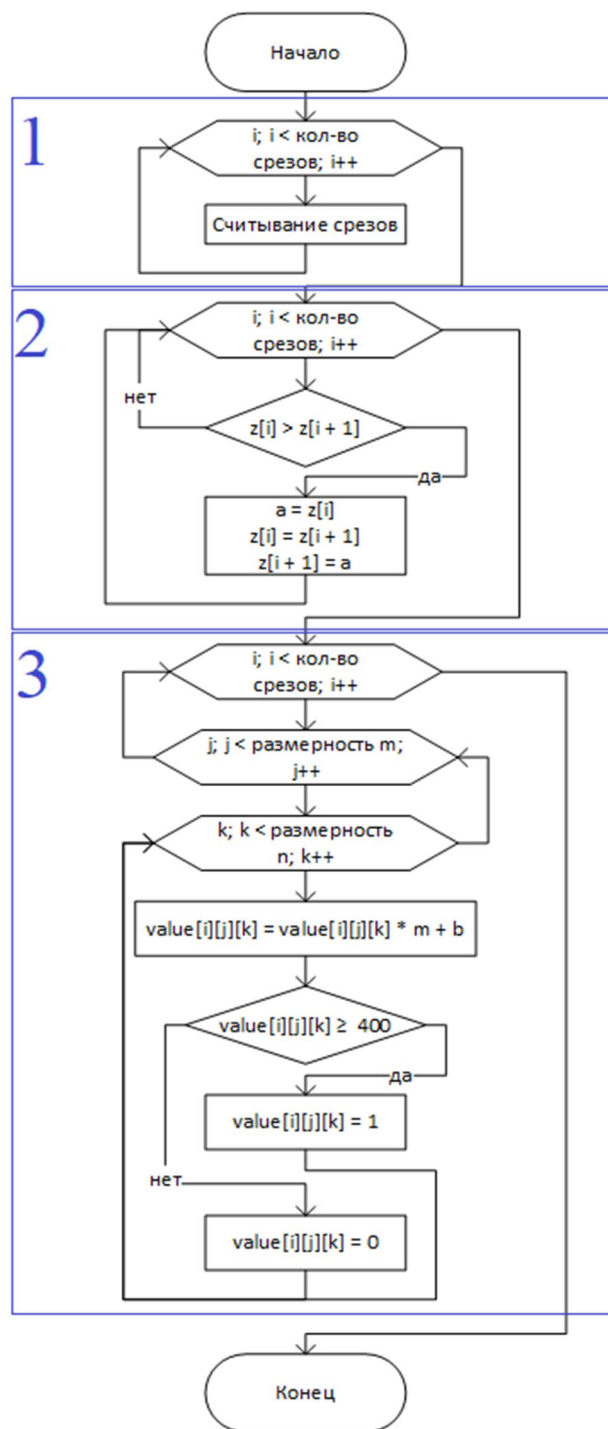


Рисунок 3 – Блок-схема определения принадлежности элемента изображения к грудной клетке

Под пунктом 1 на блок-схеме происходит считывание всех срезов. В пункте 2 происходит их сортировка по параметру  $z$ . Под пунктом 3 для каждого отдельно взятого среза размерностью  $(m \times n)$  происходит преобразование его пикселей в шкалу Хаунсфилда с последующим определением на принадлежность содержимого пикселя области грудной клетки.

**Реализация алгоритма на GPU.** Считывание DICOM-файлов и преобразование значений в шкалу Хаунсфилда отличается от версии на CPU лишь тем, что сохранение происходит не в трехмерную матрицу, а в одномерный массив. Далее данные записываются в память GPU. Для поиска координат, через которые будет проходить элементарная плоскость, выделяется память для копирования массива, полученного с помощью функции `kernel0` (размер этого массива равен  $133 \times 512 \times 512$  элементов), а также для массива `cubeIndexMas`, в котором будет храниться значение `cubeIndex` для каждого элемента исходного массива. Результат вычислений хранится в памяти GPU размером  $133 \times 512 \times 512 \times 15 \times 3$  элементов.

Для исследования производительности вычислений построения 3D-модели легких были проведены две серии экспериментов. В первой серии измерялось время построения 3D-модели грудной клетки в зависимости от объема первичных данных, полученных из DICOM-файла. Объем данных изменялся – это были соответственно 34 865 152, 17 563 648, 11 796 480, 7 077 888, 4 980 736, 3 670 016 точек.

Во второй серии измерялось время построения модели в зависимости от выделяемых суперкомпьютером числа нитей (непосредственный исполнитель вычислений) и числа блоков. Число блоков выбиралось из следующего набора: [64, 1024, 16 384, 65 535], а число нитей из набора [64, 256, 512]. Результаты построения модели визуализировались на экране средствами графической библиотеки `matplotlib` языка Python 3 [3].

**Результаты исследований и их анализ.** Последовательная реализация алгоритма построения 3D-модели грудной клетки на CPU для 34 965 152 точек потребовала 1 час 22 минуты 50 секунд (без визуализации – 1 час 15 минут 15 секунд). В то же время параллельная GPU в технологии CUDA потребовала 8 минут 21 секунду (без визуализации – 1 минуту 11 секунд).

На рисунке 4 приведена зависимость времени решения задачи от объема первичных данных как с учетом времени на визуализацию данных, так и без неё. Построение модели на GPU было реализовано с использованием 256 блоков по 256 нитей в каждом.

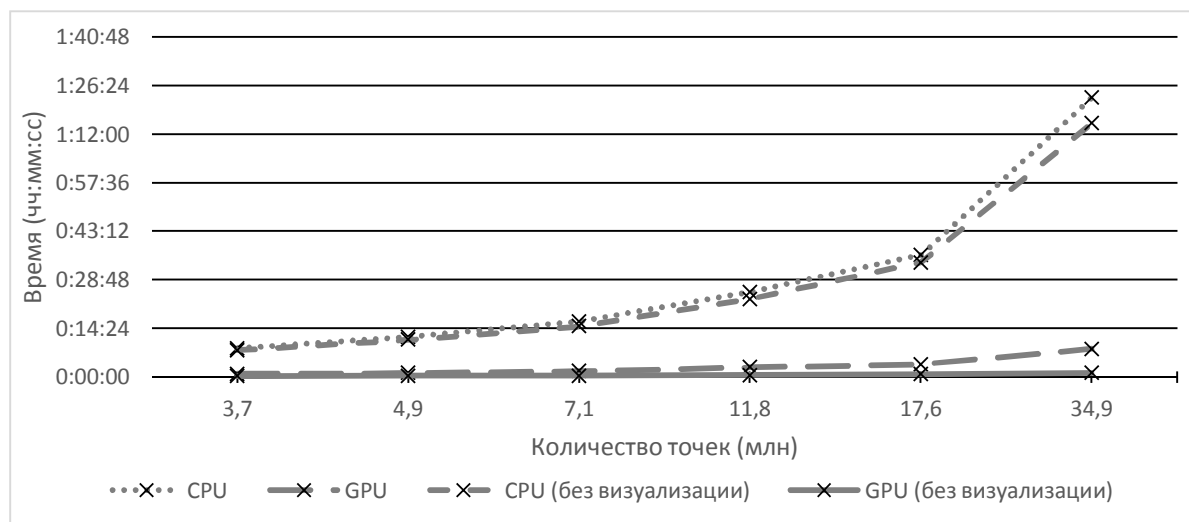


Рисунок 4 – Зависимость времени решения задачи от объема первичных данных

На рисунках 5 и 6 представлена зависимость времени выполнения задачи для GPU с учетом времени визуализации данных, а также без этого учета.

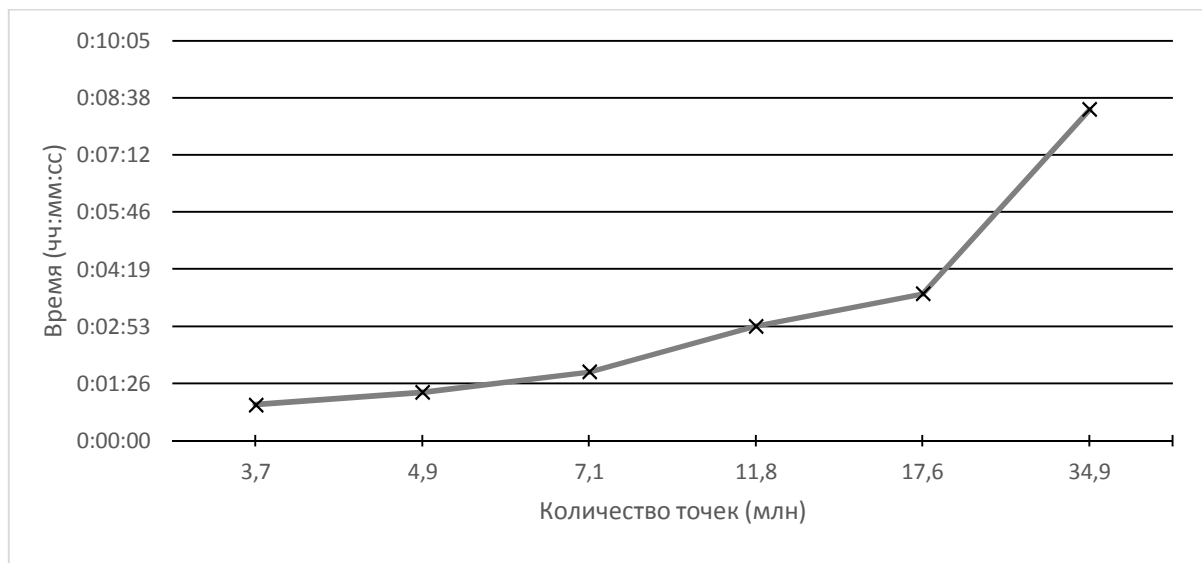


Рисунок 5 – Зависимость времени выполнения задачи на GPU при помощи технологии CUDA с учетом времени визуализации результатов

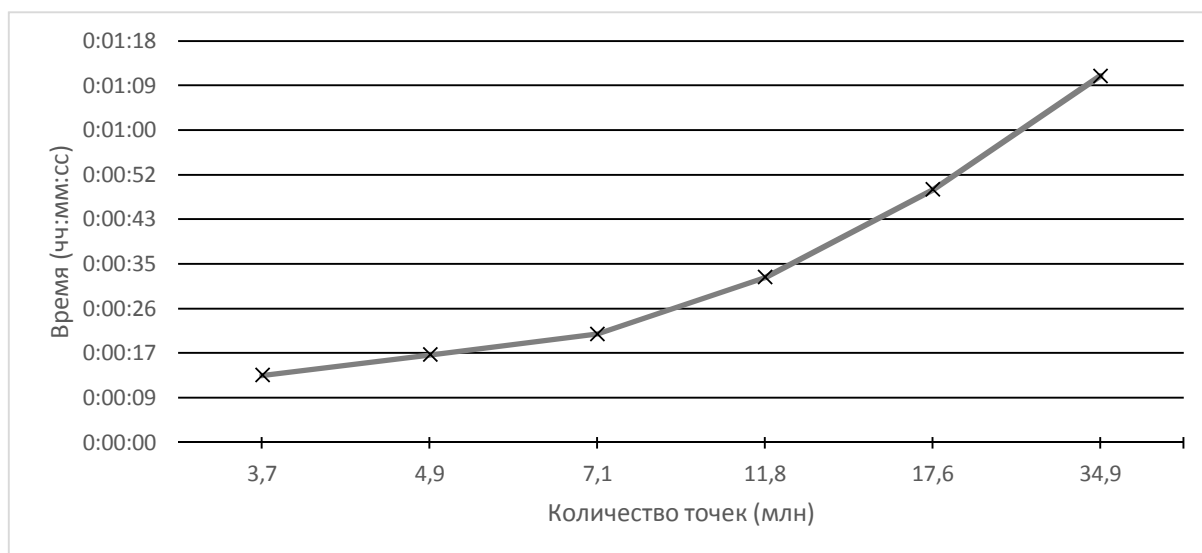


Рисунок 6 – Зависимость времени выполнения задачи на GPU при помощи технологии CUDA без учета времени, необходимого для визуализации

В обоих случаях время выполнения приложения линейно зависит от объема первичных данных. Использование технологии CUDA позволяет повысить производительность (скорость получения результатов) более чем в 10 раз, причем без учета части визуализации, производительность повышается более чем в 63 раза.

На рисунке 7 представлен график зависимости времени работы алгоритма для 34 865 152 точек от количества запущенных блоков и нитей.

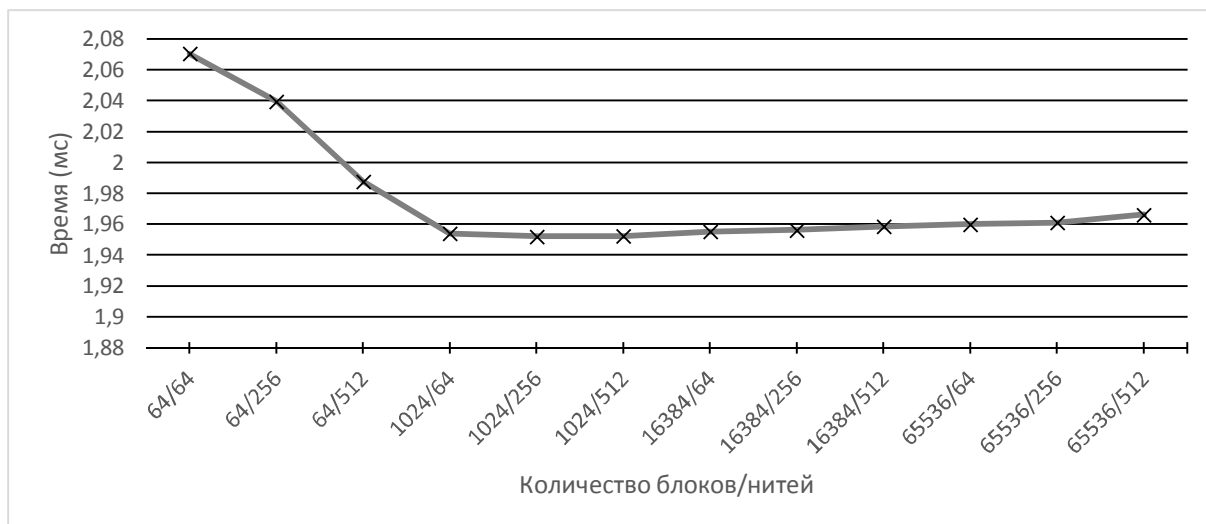


Рисунок 7 – Зависимость времени работы алгоритма для 34 865 152 точек от количества запущенных блоков и нитей

На рисунке 8 показан график зависимости времени работы алгоритма для 17 563 648 точек от количества запущенных блоков и нитей.

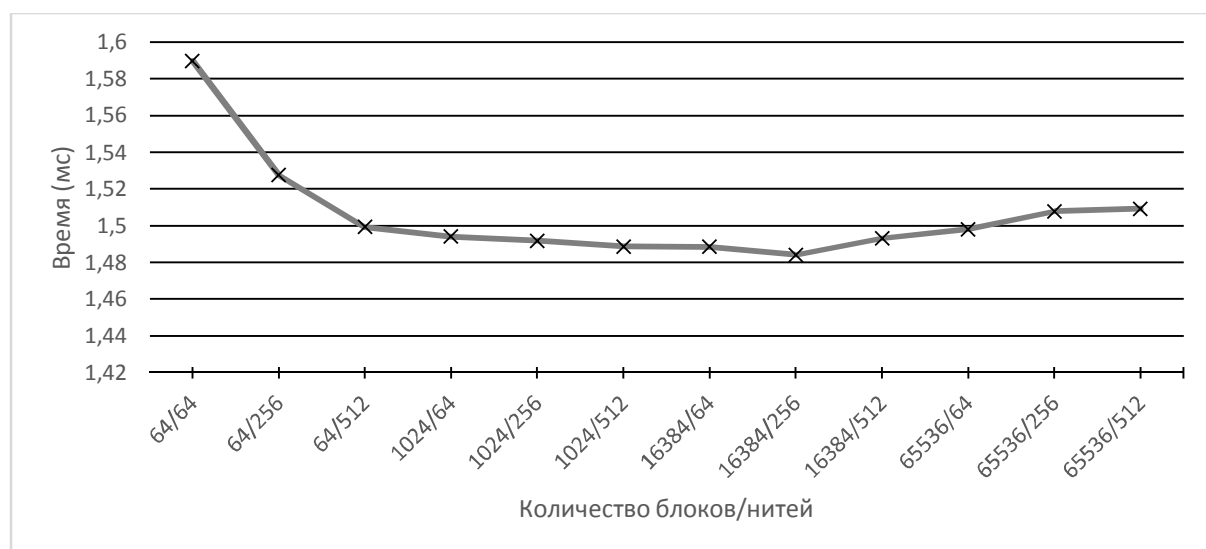


Рисунок 8 – Зависимость времени работы алгоритма для 17 563 648 точек от количества запущенных блоков и нитей

Незначительное увеличение времени работы алгоритма связано с нахождением оптимального по времени инициализации и вычислений запускаемых блоков и нитей для заданного количества точек. Также стоит отметить, что время копирования данных с GPU в память компьютера не постоянно и может отличаться на 3–5 %.

На рисунке 9 представлено отображение построенной 3D-модели грудной клетки. Визуализация выполнена средствами графической библиотеки Python (см. выше). Важно отметить, что 3D-модели грудной клетки, построенные на CPU и GPU, полностью совпадают.

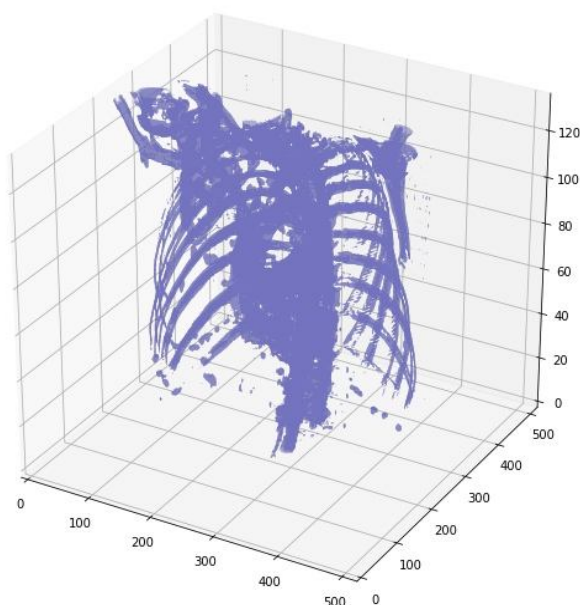


Рисунок 9 – Изображение 3D-модели грудной клетки, построенное по данным рентгеновской компьютерной томографии

На рисунке 10 представлено отображение построенной 3D-модели грудной клетки, полученное при изменении числа точек, выбранных из файла DICOM.

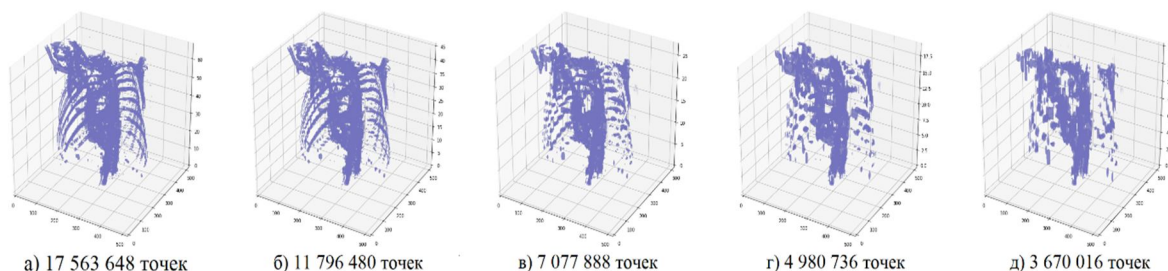


Рисунок 10 – Изображение 3D-модели грудной клетки, построенное по данным КТ

По результатам проведенной работы можно сделать вывод, что для качественной визуализации необходимо большое количество данных.

**Заключение.** Результаты проведенных вычислительных экспериментов демонстрируют возможность построения 3D-моделей внутренних органов человека в реальном времени (за 2 секунды) на персональном компьютере специалиста. Использование методов распараллеливания вычислений позволяет сократить время вычислений более чем в 60 раз. Повышение производительности обеспечивает архитектура Nvidia, GPU и технология CUDA. Кроме этого следует отметить высокую эффективность использования метода маршевых кубов для работы с форматом файла DICOM.

#### Библиографический список

1. Боресков А. В. Основы работы с технологией CUDA / А. В. Боресков, А. А. Харламов. – Москва : ДМК Пресс, 2010. – 234 с.
2. Марусина М. Я. Анализ принципов разработки системы популяционного скрининга онкологических заболеваний легких на основе использования рентгеновской компьютерной томографии / М. Я. Марусина, А. П. Волгарева, Д. В. Толкович // Прикаспийский журнал: управление и высокие технологии. – 2015. – № 2.
3. Саммерфилд М. Программирование на Python 3. Подробное руководство / М. Саммерфилд. – СПб. : Символ-Плюс, 2009. – 608 с.
4. Сандерс Дж., Кэндрот Э. Технология CUDA в примерах: введение в программирование графических процессоров / Дж. Сандерс, Э. Кэндрот ; пер. с англ. А. А. Слинкина ; науч. ред. А. В. Боресков. – М. : ДМК Пресс, 2013. – 232 с. : ил.
5. Хофер М. Компьютерная томография. Базовое руководство / М. Хофер. – М. : Медицинская литература, 2011. – 232 с.



6. Шкала Хаунсфилда – Режим доступа: [http://polyguanidines.ru/a\\_guanidini&kompyuternaya-tomografiya&2.htm](http://polyguanidines.ru/a_guanidini&kompyuternaya-tomografiya&2.htm), свободный. – Заглавие с экрана. – Яз. рус. (дата обращения 13.11.2018).
7. Bourke P. Polygonising a scalar field also known as: "3D Contouring", "Marching Cubes", "Surface Reconstruction". – 1994. – Режим доступа: <http://paulbourke.net/geometry/polygonise/>, свободный. – Заглавие с экрана. – Яз. англ.
8. Cancer imaging archive. – Режим доступа: <https://wiki.cancerimagingarchive.net/display/Public/LIDC-IDRI/>, свободный. – Заглавие с экрана. – Яз. англ. (дата обращения 13.11.2018).
9. DICOM. – Режим доступа: <https://www.dicomstandard.org>, свободный. – Заглавие с экрана. – Яз.англ. (дата обращения 13.11.2018)
10. Dicom Library. — Режим доступа: <https://www.dicomlibrary.com/terms-of-service/>, свободный. – Заглавие с экрана. – Яз. англ. (дата обращения 13.11.2018)
11. DICOM Standard Browser. — Режим доступа: <https://dicom.innolitics.com/ciods/ct-image/image-plane/00201041/>, свободный. – Заглавие с экрана. – Яз. англ. (дата обращения 13.11.2018)
12. Nvidia C. Toolkit Documentation v10.0.130 / C. Nvidia // Nvidia Corporation (Oct. 2018). – 2018.
13. Pianykh O. S. Digital Imaging and Communications in Medicine (DICOM): A Practical Introduction and Survival Guide / O. S. Pianykh. – Berlin : Springer, 2012. – 417 p. DOI: 10.1007/978-3-642-10850-1
14. Yang C. T. Hybrid CUDA, OpenMP, and MPI parallel programming on multicore GPU clusters / C. T. Yang, C. L. Huang, C. F. Lin // Computer Physics Communications. – 2011. – Vol. 182, № 1. – P. 266–269.

### References

1. Boreskov A. V., Kharlamov A. A. *Osnovy raboty s tekhnologiyey CUDA* [Basics of work with CUDA technology]. Moscow, DMK Press, 2010. 234 p.
2. Marusina M. Ya., Volgareva A. P., Tolkovich D. V. Analiz printsipov razrabotki sistemy populyatsionnogo skringinga onkologicheskikh zabolevaniy legkikh na osnove ispolzovaniya rentgenovskoy kompyuternoy tomografii [Analysis of principles for the development of a population screening system for oncological diseases of the lung based on the use of X-ray computed tomography]. *Prikaspiyskiy zhurnal: upravlenie i vysokie tekhnologii* [Caspian Journal: Control and High Technology], 2015, no. 2.
3. Sammerfeld M. *Programmirovaniye na Python 3. Podrobnoye rukovodstvo* [Programming in Python 3. Detailed manual]. SPb., Simvol-Plyus, 2009. 608 p.
4. Sanders Dzh., Kendrot E. *Tekhnologiya CUDA v primerakh: vvedeniye v programmirovaniye graficheskikh protsessorov* [CUDA technology in the examples: an introduction to programming graphics processors]. M., DMK Press, 2013. 232 p.
5. Hofer M. *Kompyuternaya tomografiya. Bazovoye rukovodstvo* [CT scan. Basic guide]. M., Meditsinskaya literatura, 2011. 232 p.
6. *Shkala Haunsfilda* [Hounsfield scale]. Available at: [http://polyguanidines.ru/a\\_guanidini&kompyuternaya-tomografiya&2.htm](http://polyguanidines.ru/a_guanidini&kompyuternaya-tomografiya&2.htm) (accessed 13.11.2018).
7. Bourke P. *Polygonising a scalar field Also known as: "3D Contouring", "Marching Cubes", "Surface Reconstruction"*, 1994. Available at: <http://paulbourke.net/geometry/polygonise/>.
8. *Cancer imaging archive*. Available at: <https://wiki.cancerimagingarchive.net/display/Public/LIDC-IDRI/> (accessed 13.11.2018).
9. *DICOM*. Available at: <https://www.dicomstandard.org>, svobodnyj (accessed 13.11.2018).
10. *Dicom Library*. Available at: <https://www.dicomlibrary.com/terms-of-service/> (accessed 13.11.2018).
11. *DICOM Standard Browser*. Available at: <https://dicom.innolitics.com/ciods/ct-image/image-plane/00201041/> (accessed 13.11.2018).
12. Nvidia C. Toolkit Documentation v10.0.130. *Nvidia Corporation* (Oct. 2018), 2018.
13. Pianykh O. S. *Digital Imaging and Communications in Medicine (DICOM): A Practical Introduction and Survival Guide*. Berlin, Springer, 2012. 417 p. DOI: 10.1007/978-3-642-10850-1.
14. Yang C. T., Huang C. L., Lin C. F. Hybrid CUDA, OpenMP, and MPI parallel programming on multicore GPU clusters. *Computer Physics Communications*, 2011, vol. 182, no. 1, pp. 266–269.