УДК 004.92

РЕШЕНИЕ ЗАДАЧ КОМПЬЮТЕРНОЙ ОПТИКИ НА ГРАФИЧЕСКИХ ВЫЧИСЛИТЕЛЬНЫХ УСТРОЙСТВАХ

© 2010 Д.Л. Головашкин[2], Н.Л. Казанский[1]

[1]Самарский государственный аэрокосмический университет
имени академика С.П. Королёва
(национальный исследовательский университет)
[2]Институт систем обработки изображений РАН

Анализируются появившиеся в последние годы методики использования графического процессора видеокарты для неграфических расчётов общего назначения (GPGPU: General-purpose graphics processing units), представленные компаниями ATI (2006 г., AMD FireStream) и NVIDIA (2007 г., CUDA: Compute Unified Device Architecture), давшие импульс к разработке алгоритмов и программных комплексов решения задач компьютерной оптики на GPU.

Особое внимание в работе уделяется обзору реализаций FDTD метода (Finite-difference time-domain method) на GPU, позволяющего решать задачи дифракционной микро- и нанооптики в рамках строгой электромагнитной теории света. Представлен обзор работ по этому направлению: от первых исследований (с использованием текстур) до законченных программных решений (FDTD Accelerware, FastFDTD).

Графический процессор, Brook, CUDA, AMD FireStream, Ray-Tracyng, FDTD.

Введение

На протяжении достаточно длительного времени магистральное направление развития вычислительной техники было связано с увеличением тактовой частоты центрального процессора. Так, с 1978 года тактовая частота процессоров архитектуры x86 возросла в 700 раз (с 4,77 МГц у Intel 8086 до 3,33 ГГц у Intel Core i7) [1]. Однако последние годы отмеченный рост существенно замедлился, что связывают [2] с “кремниевым тупиком” – физическими ограничениями, присущими технологии изготовления процессоров. Выход из положения ведущие производители вычислительной техники нашли в размещении некоторого количества ядер на одном кристалле и нескольких процессоров на одной плате (например, 8 для персонального компьютера).

Основным недостатком этого экстенсивного пути, по мнению авторов настоящей работы, является возросшая сложность архитектуры таких систем. В одном вычислительном устройстве предусмотрены обычные скалярные вычисления, суперскалярные (начиная с Intel Pentium), векторные (технология SSE, начиная с Intel Pentium III) и параллельные (в силу многоядерности и многопроцессорности). Универсализация обычно отрицательно сказывается на производительности. Для производства векторных вычислений (SIMD по классификации Флинна [3]), предпочтительнее оказываются специализированные графические процессоры (Graphics Processing Unit, GPU), отличающиеся лучшим быстродействием и распространённостью [1, 2, 4].

Следуя принятой в [5] классификации видеопроцессоров, будем различать пять поколений. Процессоры первого поколения (например, 3Dfx Voodoo I, 1996 г.) аппаратно переводили трёхмерную сцену в двумерное её изображение; второго поколения (NVidia GeForce 256, 1999г.) – дополнительно рассчитывали освещённость проецируемых объектов. Возможность программирования графических процессоров появилась в третьем поколении (NVidia GeForce 3, 2001 г.). В четвёртом (ATI Radeon X850 XT Platinum, 2004 г.) программы для GPU (шейдеры) уже могли содержать операции ветвления и циклов, записываясь на языках высокого уровня (GLSL, Cg). С этих пор начинаются вычисления общего назначения на GPU (GPGPU – General-Purpose computation on

GPU). Графические процессоры пятого поколения (например NVidia GeForce GTX 295, 2009 г.) отличаются универсальной шейдерной архитектурой: конкретные вычислительные блоки более не специализированы на исполнение шейдеров определённого типа (пиксельных, вершинных либо геометрических).

Первые публикации по применению GPU в оптических вычислениях относятся ко времени появления процессоров четвёртого поколения [6]. В то время доступ к ресурсам GPU осуществлялся через специализированный интерфейс (API – Application Programming Interface), ориентированный на работу с графикой (OpenGL, Direct3D) [7], что накладывало определённые неудобства. Программа писалась на двух языках: традиционном (для CPU) и шейдерном (для GPU). Входные данные представлялись в виде текстур, подлежащих визуализации. Взаимодействие между параллельно обрабатываемыми пикселами не предусматривалось.

Попыткой решить указанные проблемы являлся язык BrookGPU (2004 г.) [8], представляющий собой расширение языка программирования C, позволяющее абстрагироваться от графического интерфейса при написании программ (однако фактически вычисления производились через него). Настоящим прорывом в области GPGPU оказалось появление в 2007 году технологии CUDA [9] (Compute Unified Device Architecture) для графических процессоров фирмы NVidia пятого поколения, представленной этой же фирмой. Теперь связь с процессором осуществляется через драйвер видеокарты, минуя графический API. При этом в распоряжении исследователя оказывается C-подобный язык высокого уровня с развитыми библиотеками. Аналогичный проект «AMD FireStream» [10] фирмы AMD, развиваемый с 2006 года, не добился аналогичной популярности в силу большей закрытости на начальной стадии разработки. В другом современном проекте – OpenCL (Open Computing Language) [11] – создаётся аппаратно-независимый стандарт, включающий программы для GPGPU и пока ещё несколько отстающий по производительности от CUDA и AMD FireStream [12].

Совершенствование аппаратно-программной базы дало импульс к разработке алгоритмов и программных комплексов решения задач компьютерной оптики [13, 14] на GPU, которым и посвящено дальнейшее изложение.

1. Трассировка лучей

Метод трассировки лучей (Ray Tracing) традиционно используется [14,15] для расчёта световых полей в рамках геометрической оптики. Он основан на формировании изображения лучами, исходящими от источников света (прямая трассировка) либо наблюдателя (обратная трассировка), претерпевающими отражения и преломления от изучаемых объектов. Основным недостатком метода является высокая вычислительная сложность, связанная с необходимостью рассчитывать ход большого числа лучей для построения изображения объектов с хорошим качеством.

Первый программный инструмент, осуществляющий расчёт трассировки лучей на графическом вычислительном устройстве, появился в 2007 году и был разработан группой исследователей из Стэндфордского университета [16] с использованием языка BrookGPU [8]. Авторы проекта сосредоточились на графическом приложении метода трассировки, рассматривая задачу построения проекции освещённых трёхмерных сцен на двумерный экран. Современные программные продукты (OptiX фирмы NVidia [17] и RTE Лаборатории компьютерной графики и мультимедиа факультета ВМК МГУ [18]), реализующие трассировку лучей на GPU по технологии CUDA, также специализированы для решения задачи графической трассировки.

Авторы обнаружили лишь одну некоммерческую разработку, посвящённую приложению метода в оптике. Программа LINZIK [19] предназначена для трассировки лучей через сферические и асферические оптические поверхности. Она может быть использована для расчёта опти-

ческих схем (в качестве тестового примера приводится схема телескопа). При организации вычислений данные оптических поверхностей, список длин волн и данные решётки лучей передаются с центрального процессора в раздел памяти constant [20] GPU, каждому лучу соответствует своя нить (node [20]), которая оставляет в глобальной памяти координату прохождения луча через фокальную плоскость. В зависимости от типа графического процессора допускается применение технологий CUDA и AMD FireStream. Проведённые автором программы тесты на графической карте NVIDIA GeForce 8800GT (112 потоковых процессоров) продемонстрировали ускорение расчётов от 2 до 10 раз по сравнению с вычислениями на центральном процессоре Intel Pentium E2180 (два ядра, 2 ГГц). Выигрыш от использования GPU возрастал с увеличением количества оптических поверхностей в схеме. Представленное в [21] сравнение для NVIDIA GeForce 9800 GTX (G92, 1,69 ГГц, 128 ALU) и Intel Core 2 Duo E6600 (2,4 ГГц) характеризовалось шестикратным ускорением вычислений.

Независимо от области применения метода трассировки (графической или оптической) следует признать его структуру весьма удачной для реализации на GPU: ставя в соответствие одному лучу одну нить, можно одновременно задействовать все потоковые процессоры GPU в силу многочисленности лучей. Вместе с тем в ряде источников [1, 22] указывается на недостаточность числа регистров современных графических процессоров для эффективной реализации трассировки.

Отметим, что векторные алгоритмы трассировки помимо графических процессоров разрабатывают и для CPU, поддерживающих инструкции SSE [23]. Следует ожидать появления программных пакетов трассировки, написанных на OpenCL, что позволило бы объединить векторные ресурсы центрального и нескольких графических процессоров на одной плате.

2. Метод FDTD

Метод конечных разностей для решения уравнений Максвелла [13, 24] (FDTD, Finite-Difference Time-Domain) получил исключительную популярность благодаря широкой области применения. Расчёты по нему производятся в рамках строгой теории дифракции при изучении распространения импульсов и гармонических электромагнитных полей в линейных и нелинейных следах, характеризующихся произвольными диэлектрической и магнитной проницаемостями.

Следствием учёта всех компонент электромагнитного поля и их связей является высокая вычислительная сложность метода, обусловившая его первую реализацию на графическом процессоре сразу вслед за появлением такой возможности [6]. В силу необходимости использовать в то время графический интерфейс электромагнитные поля хранились в виде текстур, при этом узлы сеточной области соответствовали пикселям.

В работе [25] произведён расчёт по FDTD методу дифракционной картины на идеально проводящем цилиндре (двумерный случай, 1300 узлов по осям x и y, 20000 по времени) и кольцевом резонаторе (трёхмерный случай, 300, 480, 40 – число узлов по x, y, z, 8000 по времени) с использованием интерфейса OpenGL и графических процессоров NVidia 6800GT и 7800GTX. Сравнивая два приведённых процессора, авторы отмечают пятикратный скачок производительности при переходе к более современной карте в двумерном случае и трёхкратный – в трёхмерном. Показано десятикратное превосходство в быстродействии NVidia 7800GTX над специализированными (программируемая пользователем вентильная матрица) под FDTD микросхемами из [26].

Сравнение реализаций метода конечных разностей для решения уравнения Максвелла на GPU и неспециализированных CPU проведено в работе [27]. Если взять за основу производительность процессора Opteron 270 (2 GHz), то при реа-

лизации FDTD метода на центральных процессорах Opteron 890 (2,8 GHz), Core 2 Duo T7600 и графических процессорах GForce 7400GO, GForce 7400GS, GForce 8800GTX ускорение вычислений составит 2,2; 2,16 и 12,4; 67,82; 429,2 раза. Обсуждается зависимость быстродействия от количества узлов сеточной области: для небольших сеток использование CPU оказывается выгоднее в силу простоя большого числа потоковых процессоров GPU. При постановке вычислительных экспериментов для связи с видеокартой использовался интерфейс OpenGL, программы записывались на языках C и Cg.

С появлением технологии CUDA реализация FDTD на GPU значительно упростилась. В настоящее время оптику-исследователю не только нет нужды знакомиться с API OpenGL и Direct3D (что требует специализированных знаний в области разработки графических приложений), но даже не обязательно изучать расширенный C-подобный язык CUD'ы. Компания NVidia разработала плагины для языков MATLAB, Mathematica и LabView [28], позволяющие задействовать возможности графических процессоров. К сожалению, такая поддержка появилась совсем недавно – Atef Elsherbeni и Veysel Demir, авторы известной монографии 2008 года издания [29], посвящённой реализации FDTD на MatLab'е, в главе об использовании графических процессоров (GPU Acceleration of FDTD) ещё обращаются к инструментарию языка Brook [8].

В 2007 году NVidia начала сотрудничество с компанией Acceleware с целью разработки библиотеки CUDA FDTD Library. В настоящее время библиотека позволяет [30] моделировать распространение электромагнитного излучения через различные среды (диэлектрики, проводники и анизотропные среды), ограничивая область вычислительного эксперимента PML слоями, электрической/магнитной стенкой, условиями Mur'а или Higdon'а [13, 31, 32]. При исследовании резонатора в виде кубической каверны в идеально проводящем материале была достигнута следующая производительность (определяется в количестве значений сеточных функций, рассчитанных за одну секунду [миллион в секунду]): 800 для GPU Tesla C1060 (при объеме сеточной области в 100 миллионов узлов), 1600 для вычислительной системы из 2 процессоров Quadro Plex D2 (200 миллионов узлов) и 2800 для 4-процессорной системы, объединяющей GPU Tesla S1070 (500 миллионов узлов).

Разработка Acceleware используется в коммерческих пакетах W2405 Agilent FDTD Simulation Element (компания Agilent) [33], Concerto (Cobham Technical Sevices) [34], SEMCAD X OPTICS (Speag) [35] и Xfdtd (RemCom) [36].

Свободно распространяемым является пакет FastFDTD [37]. По свидетельству разработчиков он позволяет получить на видеокартах NVIDIA серии GeForce 6000 и выше 30-50-кратные ускорения вычислений. Независимый источник [38] подтверждает 5-кратное ускорение на GPU GeForce 6800XT и 25-кратное на GeForce 7800 GTX по сравнению с процессором Pentium 4 3,2 ГГц.

Комментируя прогресс в представленной области, необходимо отметить очевидность векторизации метода FDTD – вектор сеточных функций посредством операций сложения и вычитания выражается через векторы на предыдущих слоях. Вместе с тем, отображению FDTD на архитектуру GPU свойственна весьма существенная проблема. При моделировании распространения излучения через трёхмерные оптические элементы на наиболее распространённых графических процессорах серии GeForce ограниченный объем их оперативной памяти вынуждает организовывать многочисленные обмены с глобальной памятью, что сразу сводит на нет преимущества GPU. Преодоление указанного недостатка авторы настоящей работы видят в использовании декомпозиции вычислительной области [39,40], позволяющей за счёт принципа суперпозиции электромагнитных волн многократно снизить количество обменов между глобальной памятью и памятью видеокарты.

Заключение

Появление и бурное развитие вычислений на графических процессорах позволило за несколько последних лет на порядок снизить длительность расчётов на персональном компьютере по актуальным методам компьютерной оптики, что открывает широкие перспективы использования данных методов не только в оптике (трассировка лучей), но и в дифракционной нанофотонике (решение уравнений Максвелла) [41,42].

Благодарности

Работа выполнена при поддержке грантов РФФИ № 10-07-00553, 09-07-12147, 10-01-00453, 09-07-92421 и гранта Президента РФ №НШ-7414.2010.9.

Библиографический список

1. **Боресков, А.В.** Основы работы с технологией CUDA [Текст] / А.В. Боресков, А.А. Харламов – М.: ДМК Пресс, 2010. – 232 с.

2. **Фролов, В.В.** Введение в технологию CUDA [Электронный ресурс] // URL: http://cgm.computergraphics.ru/issues/issue16/cuda.

3. **Flynn, M.J.** Computer Architecture: Pipelined and Parallel Processor Design [Text] / M.J. Flynn – Boston: Jones and Bartlett, 1995. – 782 p.

4. **Берилло, А.** NVIDIA CUDA – неграфические вычисления на графических процессорах. Часть 1 [Электронный ресурс] // URL: http://www.ixbt.com/video3/cuda-1.shtml.

5. **Скрябин, В.** История появления графических процессоров [Электронный ресурс] // URL: http://cgm.computergraphics.ru/issues/issue18/gpuhistory .

6. **Krakiwsky, S.E.** Graphics Processor Unit (GPU) Acceleration of Finite-Difference Time-Domain (FDTD) Algorithm [Text] / S.E. Krakiwsky, L.E. Turner and M.M. Okoniewski – Microwave Symposium Digest, June 2004. – P. 1033-1036

7. **Чеканов, Д.** NVidia CUDA: вычисления на видеокарте или смерть CPU? [Электронный ресурс] // URL: http://www.thg.ru/graphic/nvidia_cuda/print.html .

8. Brook and BrookGPU research projects at the Stanford University Graphics Lab [Electronics Resource] // URL: http://graphics.stanford.edu/projects/brookgpu/ .

9. Технология CUDA компании Nvidia [Электронный ресурс] // URL: http://www.nvidia.ru/object/cuda_home_new_ru.html .

10. Технология ATIStream компании ATI [Электронный ресурс] // URL: http://developer.amd.com/gpu/ATIStreamSDK/Pages/default.aspx.

11. Проект OpenCL [Электронный ресурс] // URL: http://www.khronos.org/opencl/ .

12. Сравнение OpenCL с CUDA, GLSL и OpenMP [Электронный ресурс] // URL: http://habrahabr.ru/blogs/hi/96122/.

13. Дифракционная компьютерная оптика [Текст] / под редакцией В.А. Сойфера. – М.: Физматлит, 2007. – 736 с.

14. **Казанский, Н.Л.** Исследовательский комплекс для решения задач компьютерной оптики [Текст] / Н.Л. Казанский // Компьютерная оптика. – 2006. – № 29. – С. 58-77.

15. **Казанский, Н.Л.** Математическое моделирование оптических систем [Текст] / Н.Л. Казанский – Самара: СГАУ, 2005. – 240 с.

16. **Horn, D.** Interactive k-D Tree GPU Raytracing [Electronics Resource] / Daniel Horn, Jeremy Sugerman, Mike Houston, Pat Hanrahan // URL: http://graphics.stanford.edu/papers/i3dkdtree/ .

17. Пакет OptiX [Электронный ресурс] // URL: http://developer.nvidia.com/object/optix-examples.html .

18. Пакет RTE [Электронный ресурс] // URL: http://www.ray-tracing.ru/ .

19. Пакет LINZIK [Электронный ресурс] // URL: linzik.com .

20. CUDA, Published by NVIDIA Corporation – 2701 San Tomas Expressway Santa Clara, CA 95050 [Electronics Resource] // URL: http:/www.nvidia.com/object/cuda_home_new.html .

21. **Берилло, А.** NVIDIA CUDA – неграфические вычисления на графических процессорах. Часть 2 [Электронный ресурс] //URL: http://www.ixbt.com/video3/cuda-2.shtml.

22. **Luebke, D.** Interactive Ray Tracing with CUDA [Electronics Resource] / David Luebke and Steven Parker // URL: http://developer.nvidia.com/object/nvision08-IRT.html .

23. Интерактивная трассировка лучей с использованием SIMD инструкций [Электронный ресурс] // URL: http://software.intel.com/ru-ru/articles/interactive-ray-tracing/.

24. **Taflove, A.** Computational Electrodynamics: The Finite-Difference Time-Domain Method: 2nd. ed. [Text] / A. Taflove, S. Hagness – Boston: Arthech House Publishers, 2000. – 852 p.

25. **Price, D.K.** GPU-based accelerated 2D and 3d FDTD solvers [Text] / D.K. Price, J.R. Humphrey and E.J. Kelmelis – Phisics and Simulation of Optoelectronic Devices XV, vol. 6468 of Proceedings of SPIE, San Jose, Calif, USA, January 2007.

26. **Durbano, J.P.** FPGA-based Acceleration of the Three-Dimensional Finite-Difference Time-Domain Method for Electromagnetic Calculations [Text] / J.P. Durbano – Global Signal Processing Expo & Conference (GSPx), 2004.

27. **Adams, S.** Boppana Finite Difference Time Domain (FDTD Simulations Using Graphics Processors [Text] / S. Adams, J. Payne, R. Boppana – Proceedings of the 2007 DoD High Performance Computing Modernization Program Users Group Conference, 2007. – P. 334-338.

28. Программное обеспечение для MATLAB, использующее графические процессоры с архитектурой CUDA [Электронный ресурс] // URL: http://www.nvidia.ru/object/matlab_acceleration_ru.html.

29. **Elsherbeni, A.** The Finite Difference Time Domain Method for Electromagnetics: With MATLAB Simulations [Text] / Atef Elsherbeni and Veysel Demir – SciTech Publishing, 2009. – 450 p.

30. FDTD solver компании Acceleware [Электронный ресурс] // URL: http://www.acceleware.com/fdtd-solvers .

31. **Головашкин, Д.Л.** Методика формирования падающей волны при разностном решении уравнений Максвелла. Одномерный случай [Текст] / Д.Л. Головашкин, Н.Л. Казанский // Автометрия. – 2006. – Том 42, № 6. – С. 78-85.

32. **Головашкин, Д.Л.** Методика формирования падающей волны при разностном решении уравнений Максвелла. Двумерный случай [Текст] / Д.Л. Головашкин, Н.Л. Казанский // Автометрия. – 2007. – Том 43, № 6. – С. 78-88.

33. Страница пакета W2405 Agilent FDTD Simulation Element компании Agilent [Электронный ресурс] // URL: http://www.home.agilent.com/agilent/product.jspx .

34. Страница пакета Concerto компании Cobham Technical Sevices [Электронный ресурс] // URL: http://www.vectorfields.com/concerto.php .

35. Страница пакета SEMCAD X OPTICS компании Speag [Электронный ресурс] // URL: http://www.speag.com/products/semcad/solutions/optics/ .

36. Страница пакета Xfdtd компании RemCom [Электронный ресурс] // URL: http://www.remcom.com/xf7.

37. Страница пакета FastFDTD [Электронный ресурс] // URL: http://www.emphotonics.com/products/fastfdtd.

38. СуперЭВМ и «Нано» [Электронный ресурс] // URL: www.nanometer.ru/2007/09/12/super_4233.html.

39. **Головашкин, Д.Л.** Декомпозиция сеточной области при разностном ре-

шении уравнений Максвелла [Текст] / Д.Л. Головашкин, Н.Л. Казанский // Математическое моделирование. – 2007. – Том 19, № 2. – С. 48-58.

40. **Golovashkin, D.L.** Mesh Domain Decomposition in the Finite-Difference Solution of Maxwell's Equations [Text] / D.L. Golovashkin, N.L. Kazanskiy // Optical Memory & Neural Networks (Information Optics). – 2009. – Vol. 18, N 3. – P. 203-211.

41. **Сойфер, В.А.** Нанофотоника и дифракционная оптика [Текст] / В.А. Сойфер // Компьютерная оптика. – 2008. – Том 32, № 2. – С. 110-118.

42. **Сойфер, В.А.** Дифракционные оптические элементы в устройствах нанофотоники [Текст] / В.А. Сойфер, В.В. Котляр, Л.Л. Досколович // Компьютерная оптика. – 2009. – Том 33, № 4. – С. 352-368.

References

1. **Boreskov, A.V.** Basics of operating CUDA technology / A.V. Boreskov, A.A. Kharlamov – Moscow: DMK Press, 2010. – 232 pp. – [in Russian].

2. Frolov, V.V. Introduction to CUDA technology // URL: http://cgm.computergraphics.ru/issues/issue16/cuda. – [in Russian].

3. Flynn, M.J. Computer Architecture: Pipelined and Parallel Processor Design / M.J. Flynn – Boston: Jones and Bartlett, 1995. – 782 p.

4. Berillo, A. NVIDIA CUDA – nongraphics computation ongraphics processing units. Part 1 // URL: http://www.ixbt.com/video3/cuda-1.shtml. – [in Russian].

5. Skryabin, V. Graphics processing units history // URL: http://cgm.computergraphics.ru/issues/issue18/gpuhistory . – [in Russian].

6. Krakiwsky, S.E. Graphics Processor Unit (GPU) Acceleration of Finite-Difference Time-Domain (FDTD) Algorithm / S.E. Krakiwsky, L.E. Turner and M.M. Okoniewski – Microwave Symposium Digest, June 2004. – P. 1033-1036,

7. Chekanov, D. NVidia CUDA: GPU-aided computation or death of the CPU? // URL: http://www.thg.ru/graphic/nvidia_cuda/print.html . – [in Russian].

8. Brook and BrookGPU research projects at the Stanford University Graphics Lab // URL: http://graphics.stanford.edu/projects/brookgpu/ .

9. CUDA technology by NVidia // URL: http://www.nvidia.ru/object/cuda_home_new_ru.html . – [in Russian].

10. Technology ATIStream by ATI // URL: http://developer.amd.com/gpu/ATIStreamSDK/Pages/default.aspx .

11. OpenCL Project // URL: http://www.khronos.org/opencl/.

12. Comparison of OpenCL and CUDA, and of GLSL and OpenMP // URL: http://habrahabr.ru/blogs/hi/96122/. – [in Russian].

13. Diffractive Computer Optics / Ed. by V.A. Soifer. – Moscow: "Fizmatlit" Publishers, 2007. – 736 pp. – [in Russian].

14. Kazanskiy, N.L. A research complex for solving problems of computer optics / N.L. Kazanskiy // Computer Optics. – 2006. – V. 29. – P. 58-77. – [in Russian].

15. Kazanskiy, N.L. Mathematical modeling of optical systems / N.L. Kazanskiy – Samara: "SSAU" Publisher, 2005. – 240 pp. – [in Russian].

16. Horn, D. Interactive k-D Tree GPU Raytracing / Daniel Horn, Jeremy Sugerman, Mike Houston, Pat Hanrahan // URL: http://graphics.stanford.edu/papers/i3dkdtree/ .

17. Program OptiX // URL: http://developer.nvidia.com/object/optix-examples.html .

18. Program RTE // URL: http://www.raytracing.ru/. – [in Russian].

19. Program LINZIK // URL: linzik.com.

20. CUDA, Published by NVIDIA Corporation – 2701 San Tomas Expressway Santa Clara, CA 95050 // URL: http:/www.nvidia.com/object/cuda_home_new.html.

21. Berillo, A. NVIDIA CUDA – nongraphics computation ongraphics processing units. Part 2 // URL: http://www.ixbt.com/video3/cuda-2.shtml. – [in Russian].

22. Luebke, D. Interactive Ray Tracing with CUDA / David Luebke and Steven Parker
// URL: http://developer.nvidia.com/object/nvision08-IRT.html .

23. Interacting Ray-tracing with SIMD instructions // URL: http://software.intel.com/ru-ru/articles/interactive-ray-tracing/. – [in Russian].

24. Taflove, A. Computational Electrodynamics: The Finite-Difference Time-Domain Method: 2nd. ed. / A. Taflove, S. Hagness – Boston: Arthech House Publishers, 2000. – 852 p.

25. Price, D.K. GPU-based accelerated 2D and 3d FDTD solvers / D.K. Price, J.R. Humphrey and E.J. Kelmelis – Phisics and Simulation of Optoelectronic Devices XV, vol. 6468 of Proceedings of SPIE, San Jose, Calif, USA, January 2007.

26. Durbano, J.P. FPGA-based Acceleration of the Three-Dimensional Finite-Difference Time-Domain Method for Electromagnetic Calculations / J.P. Durbano – Global Signal Processing Expo & Conference (GSPx), 2004.

27. Adams, S. Boppana Finite Difference Time Domain (FDTD Simulations Using Graphics Processors / S. Adams, J. Payne, R. Boppana – Proceedings of the 2007 DoD High Performance Computing Modernization Program Users Group Conference, 2007. – P. 334-338

28. Software for MATLAB, using GPU with CUDA architecture // URL: http://www.nvidia.ru/object/matlab_acceleration_ru.html. – [in Russian].

29. Elsherbeni, A. The Finite Difference Time Domain Method for Electromagnetics: With MATLAB Simulations / Atef Elsherbeni and Veysel Demir – SciTech Publishing, 2009. – 450 p.

30. FDTD solver by Acceleware // URL: http://www.acceleware.com/fdtd-solvers .

31. Golovashkin, D.L. Metod of generation TE-wave for FDTD technique. One dimensional case / D.L. Golovashkin, N.L. Kazanskiy // Autometriya. – 2006. – Vol. 42, N 6. – P. 78-85. – [in Russian].

32. Golovashkin, D.L. Metod of generation TE-wave for FDTD technique. Two dimensional case / D.L. Golovashkin, N.L. Kazanskiy // Autometriya. – 2007. – Vol. 43, N 6. – P. 78-88. – [in Russian].

33. Program W2405 Agilent FDTD Simula-
tion Element by Agilent // URL: http://www.home.agilent.com/agilent/product.jspx.

34. Program Concerto by Cobham Technical Sevices // URL: http://www.vectorfields.com/concerto.php .

35. Program SEMCAD X OPTICS by Speag // URL: http://www.speag.com/products/semcad/solutions/optics/ .

36. Program Xfdtd by RemCom //
URL: http://www.remcom.com/xf7 .

37. Program FastFDTD // URL: http://www.emphotonics.com/products/fastfdtd .

38. Article “Super Computers and “nano” // URL: www.nanometer.ru/2007/09/12/super_4233.html. – [in Russian].

39. Golovashkin, D.L. Mesh Domain Decomposition in the Finite-Difference Solution of Maxwell’s Equations/ D.L. Golovashkin, N.L. Kazanskiy // Mathematical Modeling. – 2007. – Vol. 19, N 2. – P. 48-58. – [in Russian].

40. Golovashkin, D.L. Mesh Domain Decomposition in the Finite-Difference Solution of Maxwell’s Equations/ D.L. Golovashkin, N.L. Kazanskiy // Optical Memory & Neural Networks (Information Optics). – 2009. – Vol. 18, N 3. – P. 203-211.

41. Soifer, V.A. Nanofotonics and difractive optics / V.A. Soifer // Computer op-

tics. – 2008. – Vol. 32, N 2. – P. 110-118. – [in Russian].

42. Soifer, V.A. Difractive optical elements in nanofotonics devices / V.A. Soifer, V.V. Kotlyar, L.L. Doskolovich // Computer optics. – 2009. – Vol. 33, N 4. – P. 352-368. – [in Russian].

SOLVING DIFFRACTIVE OPTICS PROBLEMS USING GRAPHICS PROCESSING UNITS

© 2010 D.L. Golovashkin[2], N.L. Kasanskiy[1]

[1]Samara State Aerospace University named after academician S.P. Korolyov
(National Research University)
[2]Image Processing Systems Institute of the Russian Academy of Sciences

We analyze techniques for applying graphics processing units (GPU) to the general-purpose non-graphics computations proposed in recent years by the companies ATI (AMD FireStream, 2006) and NVIDIA (CUDA: Compute Unified Device Architecture, 2007) which gave an impetus to developing algorithms and software packages for solving problems of diffractive optics with the aid of the GPU.

In this work, a special attention is given to the overview of techniques for the GPU-aided implementation of the FDTD (finite-difference time-domain) method, which offers an instrument for solving problems of micro- and nano-optics using the rigorous electromagnetic theory. The review of the related papers ranges from the initial research (based on the use of textures) to the complete software solutions (like FDTD Software and FastFDTD).

Graphics processor, Brook, CUDA, AMD FireStream,Ray-Tracing FDTD.

Информация об авторах

Головашкин Димитрий Львович, доктор физико-математических наук, доцент, ведущий научный сотрудник. Институт систем обработки изображений РАН. Область научных интересов: дифракционная оптика, теория разностных схем, векторные и параллельные вычисления. E-mail: dimitriy@smr.ru.

Казанский Николай Львович, доктор физико-математических наук, профессор кафедры технической кибернетики. Самарский государственный аэрокосмический университет имени академика С.П. Королёва (национальный исследовательский университет). Область научных интересов: дифракционная оптика, математическое моделирование, обработка изображений и нанофотоники. E-mail: kazansky@smr.ru.

Golovashkin Dimitry Lvovich, doctor of, associate professor, senior researcher. Image Processing Systems Institute of the Russian Academy of Sciences. Area of research: FDTD method, subwave optics, vector and parallel algorithms for matrix computation. E-mail: dimitriy@smr.ru.

Kazanskiy, Nikolay Lvovich, doctor of physical and mathematical sciences, professor at technical cybernetics department. Samara State Aerospace University named after academician S.P. Korolyov (National Research University). Area of research: diffractive optics, mathematical modeling, image processing and nanophotonics. E-mail: kazansky@smr.ru.