УДК 004.92

С.А. Зори, д-р техн. наук, проф.
Донецкий национальный технический университет, г. Донецк, Украина
sa.zori1968@gmail.com

Синтез трехмерных стерео изображений в параллельной вычислительной системе 3D- стерео визуализации

Предложена организация параллельной вычислительной системы на базе графических процессоров для синтеза 3D стерео изображений методом трассировки лучей с применением разработанной автором модификации алгоритма поиска пересечений трассирующих лучей с объектами сцены. Система позволяет существенно увеличить производительность 3D стерео синтеза фотореалистичного качества.

Ключевые слова: 3D визуализация, трассировка лучей, 3D стерео изображение, синтез, графический процессор, вычислительная система

Введение

Организация компьютерных систем реалистичной 3D пространственной визуализации предусматривает принципиально новую организацию вычислительного процесса, отличную от стандартного графического 3D конвейера, и применения сложных методов синтеза и визуализации (таких, как трассировка лучей и пр.) для повышения реалистичности пространственного 3D-синтеза. В связи с этим, современные компьютерные системы синтеза изображений и визуализации окружающей обстановки требуют качественного и эффективного применения и совмещения методов реалистичной трехмерной графики как с традиционным механизмом визуализации, так и пока еще нетрадиционными методами объемной 3D- визуализации.

Необходимо отметить, что применяемые в таких системах методы и алгоритмы, обеспечивая высокую реалистичность синтезируемых изображений, имеют большую временную сложность, требуют обязательную современную аппаратную поддержку синтеза на многоядерных центральных процессорах и графических мультипроцессорах, подтверждая таким образом, что повышение скорости и реалистичности синтеза в системах 3D-визуализации по-прежнему является важной актуальной и перспективной задачей [1, 2 - 4].

Трехмерная пространственная визуализация предполагает такое компьютерное преобразование модели сцены (пространственный 3D- синтез, рендеринг), которое позволит создать полностью трехмерный визуальный объемный образ (3D-изображение) в специфическом устройстве пространственного отображения (устройстве 3D- визуализации, 3D- мониторе). 3D- изображение (пространственный визуальный образ) затем некоторым способом отображается в экранном пространстве 3D- дисплея, и человек воспринимает его (возможно, с использованием дополнительных средств сопряжения) в объемном виде.

В настоящее время подавляющее большинство систем 3D- синтеза и визуализации изображений (отображения информации) - 3D- СОИ - используют способ стереоскопической 3D- визуализации, как дающий «разумный компромисс» между качеством, скоростью и ценой 3D- визуализации [1 -5]. При этом самым применимым с точки зрения стоимости процессорного ядра подходом к реализации параллельных систем аппаратной поддержки является их реализация на базе графических процессоров (GPU).

В работе рассмотрена организация вычислительных процессов для параллельной вычислительной системы 3D- стерео визуализации на основе [6 - 8] и разработанных автором подходов и алгоритмов [1, 2, 5, 9, 10].

Организация системы 3D- стерео визуализации на базе параллельного графического процессора

С точки зрения практической реализации системы 3D- стерео визуализации на GPU, напомним, что синтез 3D- стереоизображений повышенного качества предложено выполнять как двойной рейтрейсинг независимых изображений стереопары [2, 9 -12]:

- параллельная независимая реализация синтеза кадров «левый» - «правый» на GPU;
- параллельная «внутрикадровая» реализация рендеринга методом трассировки лучей на ресурсах GPU, выделенных под каждый канал.

Далее (по необходимости, с учетом конкретики устройства 3D-отображения) на GPU также может быть проведен процесс постобработки полученных изображений стереопары (преобразование кадров в один из форматов стандартного вывода для устройств 3D- визуализации).

В связи с этим, общая организация вычислительного процесса процедуры 3D- стерео синтеза изображения методом трассировки лучей на архи-

тектуре GPU (или графического кластера GPC) может быть проиллюстрирована рисунком 1.

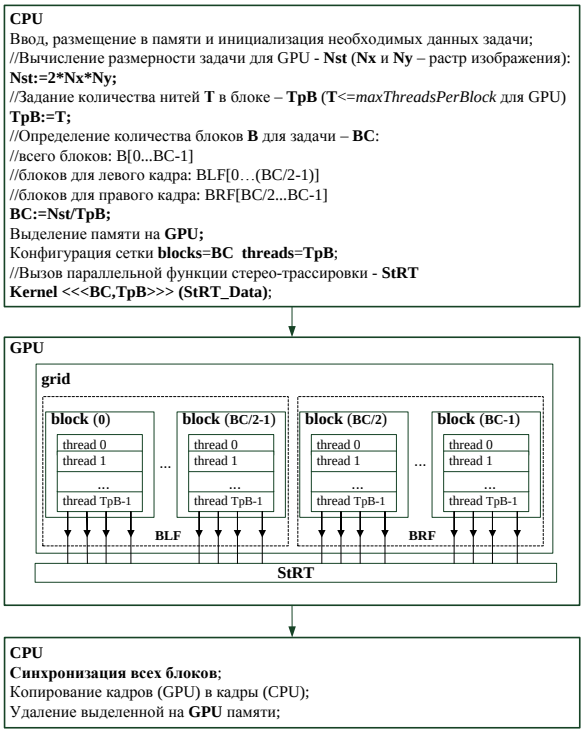


Рисунок 1 – Отображение процедуры 3D- стерео синтеза на GPU/GPC

Блоки CUDA параллельно выполняют процедуру трассировки для левого и правого кадров изображения, которые разбиты на одинаковое количество растровых кластерных блоков (подмассивов растра), внутри которых также параллельно выполняется SIMT трассировка лучей.

Необходимо отметить, что топологическая организация сети и блоков не единственна и может быть предметом оптимизации для конкретной архитектуры GPU. Для предварительной настройки конфигурации можно воспользоваться, например, CUDA GPU Occupancy Calculator [13]. Отображение процедуры 3D- синтеза на структуру GPC можно представить аналогично, например разместив блоки BLF и BRF в различные кластеры.

Основной базовой процедурой 3D- стерео визуализации при этом является трассировка лучей.

При отображении архитектуры рейтрейсинга на архитектуру GPU основной идеей является разделение алгоритма трассировки лучей на несколько kernel- ядер, каждое из которых выполняет только одну конкретную задачу. Также следует учитывать особенности конкретной микроархитектуры GPU, а также особенности самого алгоритма трассировки лучей - параллельная реализация алгоритмов рейтрейсинга осложнена большим количеством факторов, как, например, возможное завершение потоков выполнения в разные моменты времени (в случае, если часть лучей поглощена или не пересекла ни одной поверхности сцены), некогерентность при расчете вторичных лучей, приводящая к разветвлению потоков (если разные лучи пересеклись с поверхностями, которые имеют разные свойства материалов и модели рассеяния света). Следует также учитывать, что для осуществления поиска пересечения луча с поверхностями сцены и расчета освещения в найденной точке потоки выполнения требуют регулярного обращения к описанию сцены, поэтому эффективная организация ускоряющей структуры, размещение данных в памяти и доступ к ней также является немаловажным фактором.

Заметим, что ввиду актуальности проблемы параллельной организации рейтрейсинга, и в первую очередь на GPU, в литературе предложено множество практических решений. Наиболее удачным в смысле универсальности и показанных качественных показателей, является глобальная концепция, предложенная в [8]. При этом авторы в [8] указывают, что одним из наиболее трудоемких этапов визуализации является поиск ближайшего пересечения луча с объектами сцены, для повышения производительности которого применяют ускоряющие структуры.

В связи с вышеизложенным, представляется нецелесообразным рассматривать и разрабатывать подходы к реализации на архитектуре GPU общей методики трассировки лучей (достаточно воспользоваться универсальным подходом, либо более узкоспециализированными, с принятыми упрощениями), а усилия были сосредоточены на ускорении «узких мест» этапов методов путем их отображения на архитектуру GPU.

В работах [1, 5, 10] для ускорения нахождения пересечений при трассировке лучей была предложена модификация алгоритма нахождения пересечения луча с использованием ускоряющей техники на основе двухуровневой иерархии ограничивающих объемов и AABB для ускорения многопоточных вычислений. Модификация алгоритма была разработана в соответствии с рекомендациями по оптимизации, одной из важнейших которой является уменьшение ветвлений в пределах каждого отдельного блока и warp'а.

Рассмотрим экспериментальные результаты оценки быстродействия предложенной модификации алгоритма поиска пересечений при его реализации на GPU. Конфигурация тестового стенда, на котором проводились исследования, приведена в табл. 1, общий механизм отображения предложенного в [1, 11, 12] алгоритма поиска пересечений на архитектуру GPU показан на рис. 2.

Следует отметить, что конфигурирование сетки GPU в общем случае неоднозначная задача, решение которой непосредственно влияет на производительность программы, выполняемой GPU. Для определения теоретической оценки степени загрузки мультипроцессора и конфигурирования

вычислительной сети GPU использовалась утилита CUDA SDK Occupancy Calculator [13]. Согласно теоретической оценке, максимальная загрузка мультипроцессора (100%) возможна при использовании менее 10 регистров на нитку, и размерах блоков в 192, 256 или 384 нити. Загрузка мультипроцессора GPU в 67% возможна при использовании 16 регистров на нитку, и размерах блоков в 192, 256 или 512 нитей. Большее количество выделенных регистров на нитку приводит к понижению загрузки мультипроцессора GPU до 50% (от 17 до 20), до 33% (от 21 до 32) и ниже (табл. 2, рис. 3, табл. 3, рис. 4).

Таблица 1 - Конфигурация тестового стенда

OS	CPU	RAM	GPU
Windows 8 Ultimate, x64	Intel Core 2 Quad Q9100 @ 2,26 ГГц	3 ГБ DDR3	NVIDIA GeForce G110M (16 CUDA Cores)

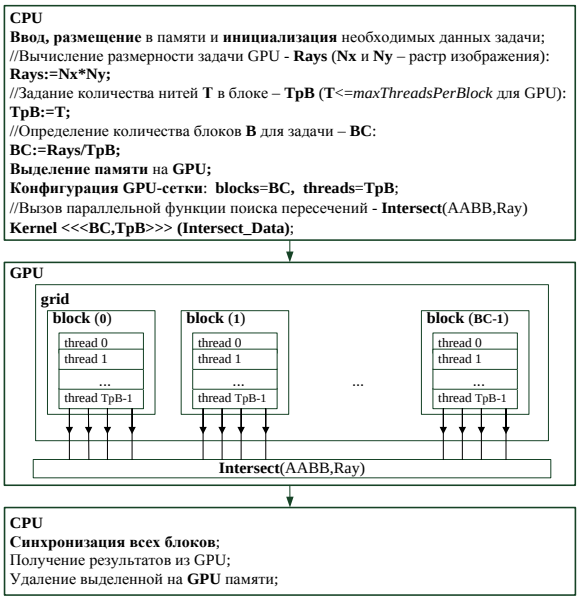


Рисунок 2 – Общий механизм вычисления пересечений на GPU

Таблица 2 - Характеристики конфигурации сети GPU (100% загруженность)

Threads Per Block	192	256	384
Registers Per Thread	10	10	10
Active Threads per Multiprocessor	768	768	768
Active Warps per Multiprocessor	24	24	24
Active Thread Blocks per Multiprocessor	4	3	2
Occupancy of each Multiprocessor	100%	100%	100%

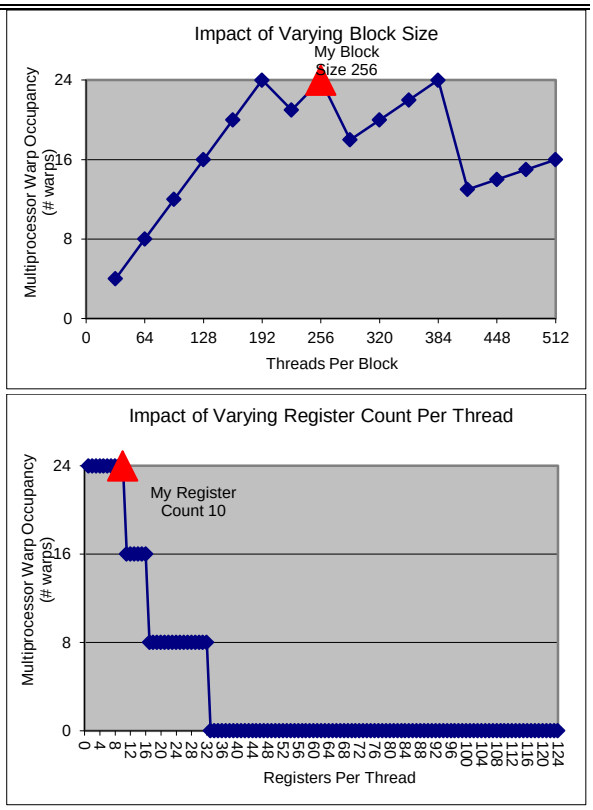


Рисунок 3 – Загрузка (100%) мультипроцессора (варпов, из 24) в зависимости от количества ниток и выделенного количества регистров на нить в блоке GPU

Таблица 3 - Характеристики конфигурации сети GPU (67% загруженность)

Threads Per Block	128	256	512
Registers Per Thread	16	16	16
Active Threads per Multiprocessor	512	512	512
Active Warps per Multiprocessor	16	16	6
Active Thread Blocks per Multiprocessor	4	2	1
Occupancy of each Multiprocessor	67%	67%	67%

Однако результаты проведенных экспериментов на этих двух конфигурациях показали, что, при теоретической загрузке мультипроцессора в 100% скорость работы выполнения всего алгоритма оказалось меньше, чем при конфигурациях сети GPU с 67% загрузкой (была выбрана для дальнейших экспериментов). Одна из причин – более интенсивное использование глобальной памяти при малом количестве выделенных регистров.

Заметим, что это не означает ее оптимальности по быстродействию, экспериментальные исследования по оптимизации вычислительного процесса в CUDA- сети (как использованного для экспери-

мента GPU, так и на других, более мощных архитектурах GPU) не проводились. Поэтому полученные и описанные ниже экспериментальные оценки могут быть существенно улучшены при реализации на новых GPU с большим количеством вычислительных ядер и мультипроцессоров, а также проведении оптимизации конфигурации вычислительной CUDA- сети.

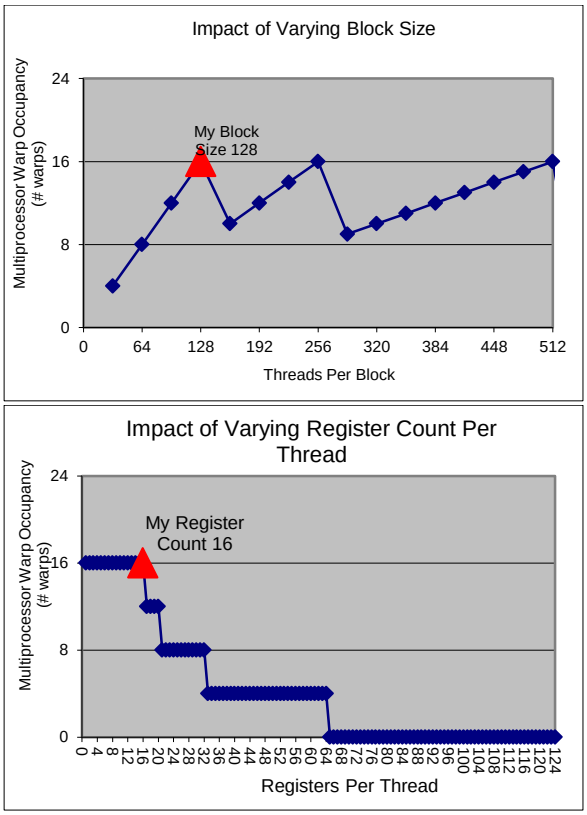


Рисунок 4 – Загрузка (67%) мультипроцессора (варпов, из 24) в зависимости от количества ниток и выделенного количества регистров на нить в блоке GPU

Экспериментальное исследование разработанной системы 3D- стерео визуализации на базе GPU и предложенного алгоритмического обеспечения

Результаты экспериментальных оценок повышения быстродействия разработанного модифицированного алгоритма по сравнению с базовой версией при реализации на архитектуре GPU [10 - 12] тестовой системы показаны на рис. 5 и 6.

Видно, что модифицированный алгоритм дает прирост производительности по отношению к базовому варианту от 27% до 32% при его реализации на GPU тестового стенда (однопоточная CPU реализация показала прирост производительности 3-11%) [1], при этом оптимизация реализации на конкретной архитектуре используемого GPU не проводилась.

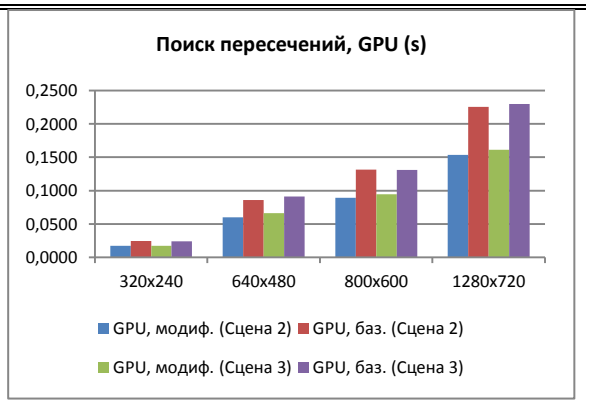


Рисунок 5 – Время поиска пересечений (модифицированная – базовая реализации), GPU

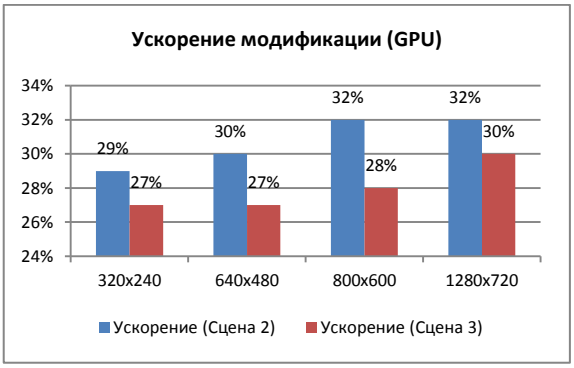


Рисунок 6 – Эффективность модификации, GPU

Результаты экспериментальных исследований времени реализации на CPU/GPU тестовой системы (достигнутое ускорение) предложенной модификации алгоритма поиска пересечений показаны на рис. 7 и 8. Диаграммы 8 и 9 обобщают результаты оценок производительности программной реализации модифицированного алгоритма поиска пересечения луча с AABB и позволяет сравнить между собой временные показатели, которые были получены при его выполнении на четырех потоках на CPU (4 ядра) и на тестовом GPU (16 CUDA ядер). Реализация на тестовом GPU позволила получить среднее ускорение в 750%.

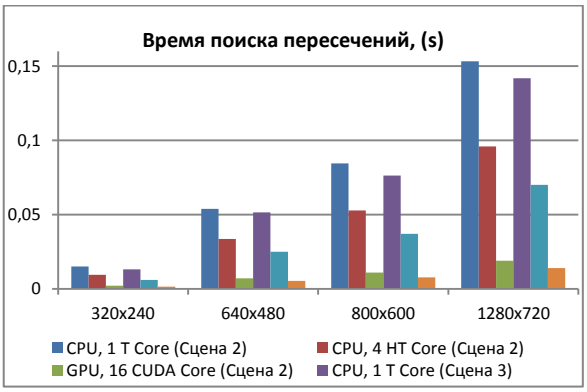


Рисунок 7 – Время поиска пересечений

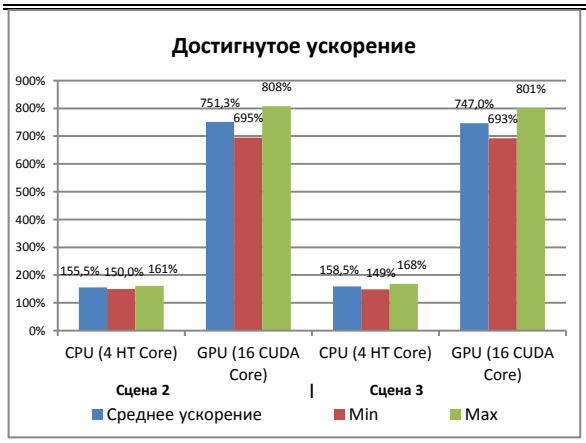


Рисунок 8 – Ускорение при многопоточной реализации на тестовых CPU и GPU

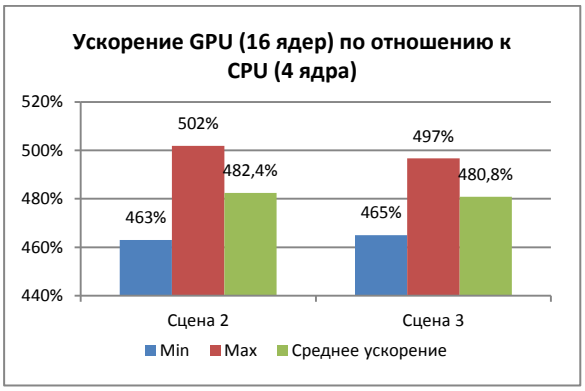


Рисунок 9 – Достигнутое ускорение на GPU в сравнении с CPU

Еще раз следует отметить, что полученные и описанные выше экспериментальные оценки могут быть существенно улучшены при реализации на новых архитектурах GPU с большим количеством вычислительных ядер и мультипроцессоров, а также проведении оптимизации конфигурации вычислительной CUDA- сети (при экспериментах не использовалась).

Разработан экспериментальный прототип программной системы 3D- стерео визуализации методом трассировки лучей с использованием разработанных алгоритмических и архитектурных средств, который успешно решает задачу синтеза 3D- стерео изображений.

Для выполнения сравнительной оценки визуальной реалистичности полученных экспериментальных изображений необходимо провести генерацию эталонных изображений тестовых сцен методом трассировки лучей в одном из программных пакетов, использующих физически корректные рейтрейсинг-рендеры.

Разработанный прототип программной системы синтеза 3D- стереоизображений использует метод обратной трассировки лучей, основанный на алгоритме Уиттеда [6], и не производит генерации некоторых высокореалистичных эффектов глобального освещения, реализуемых мощными физически корректными рендерами, такими как FryRender (http://fryrender.com/), Indigo Renderer (http://www.indigorenderer.com/), Kerkythea Rendering System (http://www.softlab.ntua.gr/~jpanta/Graphics/Kerkythea/), Maxwell Render (http://maxwellrender.com/) и пр., так как основной задачей разработки была проверка работоспособности предложенных алгоритмических средств – предложенной архитектурной реализации на GPU синтеза 3D- стереоизображений методом трассировки лучей и разработанной модификации ускоряющей техники поиска пересечений лучей с AABB.

В связи с этим, для определения сравнительной оценки визуальной реалистичности полученных изображений была использована генерация «эталонных» изображений тестовой сцены (отдельно для левого и правого кадров) методом трассировки лучей в пакете 3ds Max [14] с использованием рендера mental ray (настройки рендера в основном соответствуют настройкам программной реализации разработанного прототипа программной системы, использованы Standard-точечный и Area Lights источники света, для реализации каустики использована прямолинейная трассировка с учетом коэффициента преломления, метод окончательного сбора реализован при помощи модуля Final Gathering) и методика определения нормированной среднеквадратичной ошибки NMSE визуальной отличности эталонного и экспериментального изображений.

Согласно этой методики, визуальное отличие между двумя изображениями можно оценить нормированной среднеквадратичной погрешностью (NMSE) [15], которая рассчитывается следующим образом:

$$NMSE = \frac{\sum_i (R(i)_1 - R(i)_2)^2 + (G(i)_1 - G(i)_2)^2 + (B(i)_1 - B(i)_2)^2}{\sum_i (R(i)_1)^2 + (G(i)_1)^2 + (B(i)_1)^2} \quad (1)$$

где:

i- количество пикселей изображения;

$(R_1(i), G_1(i), B_1(i))$, $(R_2(i), G_2(i), B_2(i))$ - интенсивности цвета красной, зеленой и синей составляющих цвета i-го пикселя эталонного (1) и сформированного (2) изображений.

В компьютерной графике при тестировании изображений используют такие оценки [15]: если значение NMSE не больше 0,0001, то визуально изображения не отличаются одно от другого; если NMSE находится в диапазоне [0,0001-0,00025], то два изображения имеют незначительные отличия; если NMSE находится в диапазоне [0,00025-0,001], то изображения имеют визуально заметные отличия; если NMSE больше 0,001, то два изоб-

ражения существенно отличаются один от другого.

Сравнение по этому критерию эталонного и экспериментального, сгенерированного системой, изображений (рис. 10) показал значение NMSE равное 0,00021. Это означает, что изображения имеют незначительные визуальные отличия и объясняется невозможностью полной идентичности настроек рендеров в эталонной и экспериментальной системах при рейтрейсинг- синтезе (закрытая, не отключаемая оптимизация рендеринга и реализация применяемых шейдеров в 3ds Max, встроенный механизм генерации "мягких" теней Area Shadow и антиэлайсинга в mental ray и пр.).

Примеры некоторых сформированных системой изображений приведены на рис. 11.

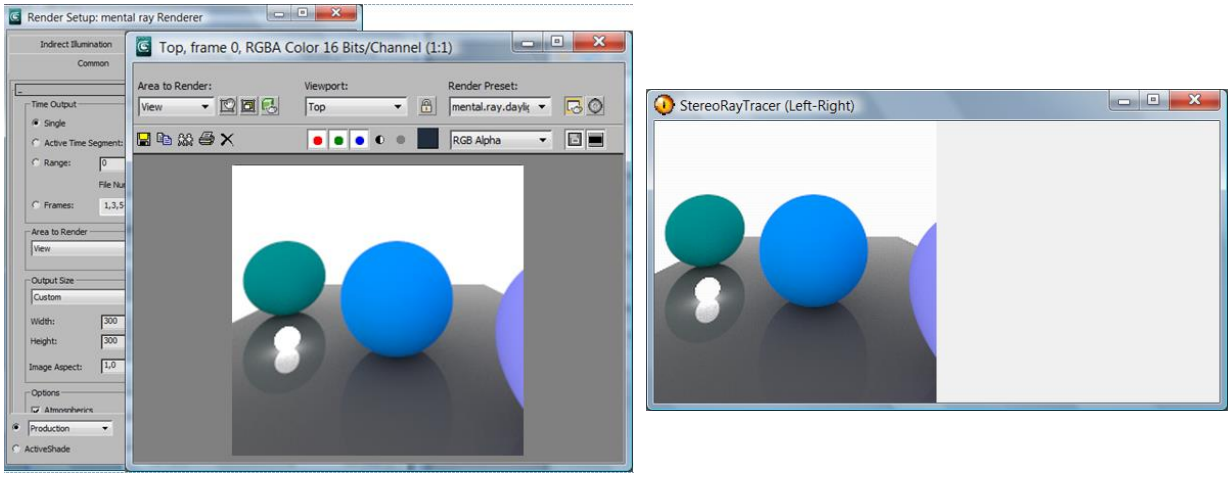
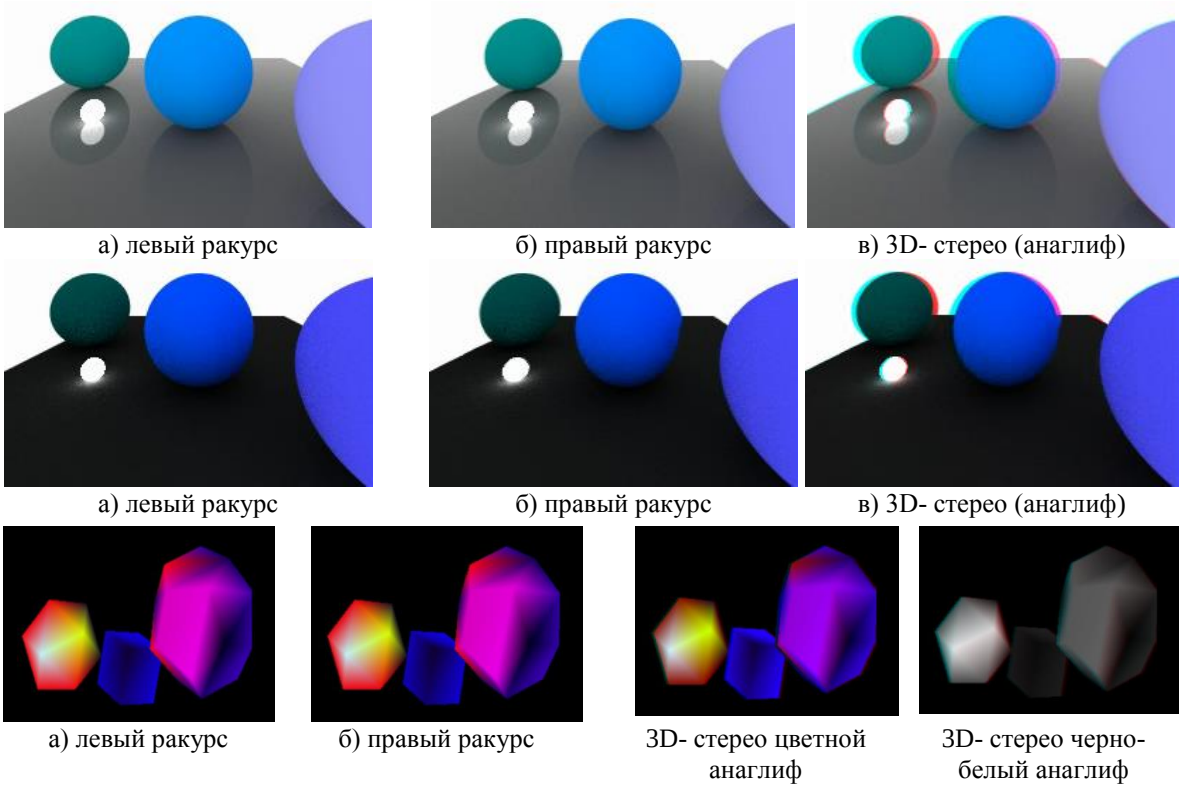


Рисунок 10 – Эталонное (слева) и сгенерированное программной системой (справа, левый ракурс стереопары) изображения тестовой сцены



а) левый ракурс б) правый ракурс в) 3D- стерео (анаглиф)

а) левый ракурс б) правый ракурс в) 3D- стерео (анаглиф)

а) левый ракурс б) правый ракурс 3D- стерео цветной анаглиф 3D- стерео черно-белый анаглиф

Рисунок 11 – Синтезированные 3D- стереоизображения тестовых сцен

Исследование работы системы показало, что:

1. Разработанный прототип программной системы 3D- стерео визуализации с использованием предложенных алгоритмических и архитектурных средств успешно решает задачу синтеза 3D- стерео изображений.

2. Синтезированные изображения ракурсов имеют незначительные визуальные отличия по сравнению с эталонными, рассчитанными рейтрейсинг- рендером 3ds Max, что объясняются невозможностью полной идентичности настроек рендеров эталонной и экспериментальной систем.

3. Около 70-ти процентов времени в среднем тратится на решение вычислительной задачи, значительная часть времени (до 30%) тратится на пересылку данных между центральным процессором и графической видеокартой для вычислений и выполнения процесса визуализации.

4. При увеличении вычислительной сложности сцены время для ее обработки тоже увеличивается, зависимость имеет почти линейный характер.

5. При использовании одной видеокарты и вычислениях GPGPU, процессы расчета сцены, постобработки и визуализации фактически происходят последовательно, что в итоге замедляет весь процесс.

6. Исследование влияния выбора размера и конфигурации вычислительной CUDA- сети на скорость вычислений показали важность их правильного задания.

Заключение

В работе выполнено отображение предложенной архитектуры параллельной вычислительной системы реалистичной 3D- стерео визуализации трехмерных сцен ускоренным методом трассировки лучей на GPU и отображение обобщенной процедуры 3D- стерео синтеза на GPU/GPC.

Выполнено исследование эффективности предложенных решений при GPU- реализации по сравнению с однопоточной и многопоточной реализациями на CPU тестовых стендов. Результаты исследований показали, что:

1. Модифицированный алгоритм [1, 5, 10, 12] дает прирост производительности по отношению к базовому варианту от 27% до 32% при его реализации на GPU тестового стенда (однопоточная CPU реализация показала прирост производительности 3-11%), при этом оптимизация реализации на конкретной архитектуре используемого GPU не проводилась.

2. Обобщенные результаты оценок производительности программных реализаций показали общее среднее достигнутое ускорение при многопоточной реализации на тестовых GPU и CPU составило около 7,5 и 1,55 раз соответственно.

3. Отмечено, что полученные экспериментальные оценки могут быть существенно улучшены при реализации на новых архитектурах GPU с большим количеством вычислительных ядер и мультипроцессоров, а также проведении оптимизации конфигурации вычислительной CUDA- сети.

4. Разработанный прототип программной системы 3D- стерео визуализации предложенными алгоритмическими и архитектурными средствами успешно решает задачу синтеза 3D- стерео изображений. При этом синтезированные изображения имеют незначительные визуальные отличия по сравнению с эталонными, рассчитанными рейтрейсинг- рендером 3ds Max (NMSE = 0,00021), что объясняются невозможностью полной идентичности настроек рендеров эталонной и экспериментальной систем.

5. При использовании одной видеокарты и вычислениях GPGPU, процессы расчета сцены, постобработки и визуализации фактически происходят последовательно, что в итоге замедляет весь процесс.

6. Исследование влияния выбора размера и конфигурации вычислительной CUDA- сети на скорость вычислений показали важность их правильного задания.

Список литературы

1. Зори С.А., Башков Е.А. Системы пространственной визуализации окружающей обстановки // Науковий вісник Донецького національного технічного університету – Красноармійськ, ДонНТУ, Міністерство освіти та науки України , 2016. - №1 (1). - С. 20-45.
2. Реалистическая пространственная визуализация с использованием технологий объемного отображения: Монография / Е.А.Башков, С.А.Зори. - Донецк, ГВУЗ "ДонНТУ", 2014. - 150 с.
3. Barry G. Blundell, Adam J. The Classification of Volumetric Display Systems // Schwarz Characteristics and Predictability of the Image Space," IEEE Transactions on Visualization and Computer Graphics. – Vol. 8 – No. 1– January–March, 2002.– P. 66–75.
4. Самарин А. Современые технологии дисплеев объемного изображения // Современная электроника. — 2005, №2. — С. 2–7 .
5. S. A. Zori, P. A. Porfirov Productivity Increasing of Realistic Ray Tracing Stereo- Image Synthesis //Journal of Qafqaz University, Mathematics and Computer Science ISSN 1302-6763, Vol. 3, No 1, 2015.- p. 30- 38
6. Realistic Ray Tracing, Second Edition / P. Shirley, R. K. Morley. — Wellesley: A K Peters/CRC Press, 2003. — 235 с.
7. Distributed ray tracing / R. L. Cook, T. Porter, L. Carpenter // ACM SIGGRAPH Computer Graphics: ACM, 1984. — V. 18 — с. 137–145.

8. Боголепов Д., Ульянов Д., Сопин Д., Турлапов В. Оптимизация метода двунаправленной трассировки путей для моделирования оптического эксперимента на графическом процессоре // Научная визуализация. – 2013. – Кв.2., Т.5. – №2. – С. 1 – 15.
9. С.А. Зори, И.А. Запорожченко, М.А. Григорьев Анализ алгоритмов трассировки лучей для реалистичной визуализации трёхмерных сцен и способов уменьшения их вычислительной сложности // Цифровая обработка сигналов и изображений. — 2013. — с. 352–357.
10. Зори С.А. Объемная визуализация алгоритмом трассировки лучей с использованием двухуровневой иерархии ограничивающих объемов и AABB // Наукові праці Донецького національного технічного університету. Серія: "Інформатика, кібернетика та обчислювальна техніка" ISSN 1996-1588. №2 (21)' 2015.- с. 5-10
11. Зори С.А. GPU- реализация параллельной вычислительной системы 3D- стерео визуализации с использованием метода трассировки лучей // Системи обробки інформації: Збірник наукових праць. – Х.: Харківський університет Повітряних Сил ім. Івана Кожедуба Міністерства Оборони України, 2016. – Вип. 6 (143). – С. 201-204.
12. Zori S.A., Al-Oraiqat Anas M. 3D- Visualization by Raytracing Image Synthesis on GPU // International Journal of Engineering Science and Technology (IJEST). - Vol. 8, № 06, Jun. 2016. – P. 97 – 104.
13. CUDA Occupancy Calculator Helps pick optimal thread block size [Электронный ресурс]. – Электрон. текстовые данные. [Цитировано: 11.01.2017 г.] – Режим доступа: https://devtalk.nvidia.com/default/topic/368105/cuda – occupancy – calculator – helps – pick – optimal – thread – block – size/
14. Kelly L. Murdock 3ds Max 2010 Bible // Wiley Publ. Inc. - 2009. - 1312 p.
15. Meyer L. A comparison of two image quality models / L. Meyer, R. Klassen // In Human Vision and Electronic Imaging III. SPIE. – Vol. 3299. – 1998. – P. 98 – 109.

Надійшла до редакції 30.03.2017

С.А. ЗОРІ
Донецький національний технічний університет (Україна)

СИНТЕЗ ТРИВИМІРНИХ СТЕРЕОЗОБРАЖЕНЬ В ПАРАЛЕЛЬНІЙ ОБЧИСЛЮВАЛЬНІЙ СИСТЕМІ 3D- СТЕРЕО ВІЗУАЛІЗАЦІЇ

В роботі запропоновано організацію паралельної обчислювальної системи на базі графічних процесорів для синтезу 3D стереозображень методом трасування променів з застосуванням розробленої автором модифікації алгоритму пошуку перетинань трасуючих променів з об'єктами сцени. Система дозволяє істотно збільшити продуктивність 3D стерео синтезу зображень фотореалістичної якості.

Ключові слова: 3D візуалізація, трасування променів, 3D стерео зображення, синтез, графічний процесор, обчислювальна система.

S.A. ZORI
Donetsk National Technical University (Ukraine)

THREE-DIMENSIONAL STEREO IMAGE SYNTHESIS IN PARALLEL COMPUTER STEREO 3D-VISUALIZATION SYSTEM

The paper proposes the organization of a parallel computer system based on graphics processors (GPU) for the 3D stereo image synthesis by ray tracing method with the use of the developed by the author modification of algorithm for the fast search of tracing rays intersections with scene objects.

The generalized procedure of 3D stereo image synthesis on the GPU / GPC is proposed. The efficiency of the proposed solutions by GPU implementation is compared with single-threaded and multithreaded implementations on the CPU. The average achieved acceleration in multi-thread implementation on the test GPU and CPU was about 7.5 and 1.55 times.

A hardware-software prototype of the proposed system was developed. The system allows to significantly increasing the performance of 3D stereo image synthesis in photorealistic quality. The synthesized images have insignificant visual differences in comparison with the etalon, which calculated by 3ds Max ray-tracing render (NMSE = 0.00021).

Research of influence of size selection and configuration of computer CUDA- network on the speed of calculations has shown the importance of their correct choice. When using a single video card and GPGPU calculations, the processes of scene calculation, post-processing and visualization actually occur sequentially and ultimately slows the whole process. The obtained experimental estimations can be significantly improved by new GPUs with a large number of processing cores and multiprocessors, as well as optimizing the configuration of the computing CUDA network.

Key words: 3D visualization, ray tracing, 3D stereo image, synthesis, graphics processor, computer system.

REFERENCES

1. Bashkov, Ye.A., Zori, S.A. (2016), *Systems of spatial visualization of environment*[Sistemi prostranstvennoy vizualizaciyi okrujayushey obstanovki], Naukoviy vestnik DonNTU, Krasnoarmiysk, No. 1 (1), pp. 20-45.

2. Bashkov, Ye.A., Zory, S.A. (2014), *Realistic spatial visualization using volumetric display technologies* [Realisticheskaya prostranstvennaya vizualizatsiya s ispolzovaniem tehnologiy ob'emnogo otobrazheniya], SHEI "DonNTU", Donetsk, 150 p.

3. Barry, G. Blundell, Adam, J. (2002), *The Classification of Volumetric Display Systems*, Schwarz Characteristics and Predictability of the Image Space, IEEE Transactions on Visualization and Computer Graphics, Vol. 8, No. 1, pp. 66-75.

4. Samarin, A. (2005), *Modern three-dimensional image display technologies*, [Sovremenyye tekhnologii displeyev obyemnogo izobrazheniya], Sovremennaya elektronika, No. 2, pp. 2-7.

5. Zori, S.A., Porfirov, P.A. (2015), *Productivity Increasing of Realistic Ray Tracing Stereo- Image Synthesis*, Journal of Qafqaz University, Mathematics and Computer Science, V. 3, No. 1, pp. 30-38.

6. Shirley, P., Morley, R. K. (2003), *Realistic Ray Tracing*, CRC Press, 235 p.

7. Cook, R.L., Porter, T., Carpenter, L. (1984), *Distributed ray tracing*, ACM SIGGRAPH Computer Graphics, V. 18, pp. 137-145.

8. Bogolepov, D., Ulyanov, D., Sopin, D., Turlapov, V. (2013), *Optimization of the bidirectional path trace method for modeling the optical experiment on a graphics processor* [Optimizaciya metoda dvunapravlennoy trassirovki putey dlya modelirovaniya opticheskogo eksperimenta na graficheskom processore], Nauchnaya vizualizaciya, No. 2, Issue 5, pp. 1-15.

9. Zori, S.A., Zaporozhchenko, I.A., Grigorev, M.A (2013), *Analysis of ways to reduce the computational complexity of ray tracing algorithm and methods of its parallel implementation* [Analiz metodov umensheniya vyichislitelnoy slozhnosti algoritma trassirovki luchey i sposobov ih realizatsii], Tsifrovaya obrabotka signalov i izobrazheniy, pp. 352–357.

10. Zori, S.A. (2015), *Volumetric visualization by ray tracing algorithm with two-level hierarchy of limiting volumes and AABB* [Ob'emnaya vizualizatsiya algoritmom trassirovki luchey s ispolzovaniem dvuhurovnevoy ierarhii ogranichivayuschih ob'emov i AABB], Naukovi pratsi DonNTU: Informatyka, Kybernetyka ta obchysliuvalna teknika, No. 2 (21), pp. 5-10.

11. Zori, S.A. (2016), *GPU- implementation of parallel computing system 3D stereo imaging using the ray tracing method* [GPU- realizaciya parallelnoy vyichislitelnoy sistemi 3D- stereo vizualizaciyi s ispolzovaniem metoda trassirovki luchey], Sistemi obrobki iformaciyi: Sbirnik naukovih prac', Kharkiv, No. 6 (143), pp. 201-204.

12. Zori, S.A., Al-Oraiqat, Anas M. (2016), *3D- Visualization by Raytracing Image Synthesis on GPU*, International Journal of Engineering Science and Technology (IJEST), Vol. 8, Issue 6, pp. 97 – 104.

13. "CUDA Occupancy Calculator Helps pick optimal thread block size", available at: https://devtalk.nvidia.com/default/topic/368105/cuda – occupancy – calculator – helps – pick – optimal – thread – block – size/

14. Murdock, Kelly L. (2010), *3ds Max 2010 Bible*, SDC Publications, 1312 p.

15. Meyer, L., Klassen, R. (1998), *A comparison of two image quality models*, Human Vision and Electronic Imaging III. SPIE, Vol. 3299, pp. 98 – 109.