Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего профессионального образования

«Комсомольский-на-Амуре государственный

технический университет»

На правах рукописи

Валова Виктория Сергеевна

**Исследование и программная реализация параллельных алгоритмов и технологий платформы WPF для генерации растровых изображений**

Направление 010400.68 – «Прикладная математика и информатика»

Профиль ­– «Математическое моделирование»

АВТОРЕФЕРАТ ДИССЕРТАЦИИ

на соискание академической степени магистра

2015

Работа выполнена в ФГБОУ ВПО

«Комсомольский-на-Амуре государственный технический университет»

|  |  |
| --- | --- |
| Научный руководитель: | кандидат физико-математических наук, доцент, доцент кафедры «Прикладная математика и информатика»**Лошманов Антон Юрьевич** |
| Рецензент: | кандидат технических наук, научныйсотрудник Института машиноведенияи металлургии ДВО РАН**Соснин Александр Александрович**  |

Защита состоится 17 июня 2015 г. в 8.30 часов по адресу: 681000, г. Комсомольск-на-Амуре, пр. Ленина, 27, ауд. 321.

Автореферат разослан 10 июня 2015 г.

**ОБЩАЯ ХАРАКТЕРИСТИКА ДИССЕРТАЦИОННОЙ РАБОТЫ**

*Актуальность темы.* На протяжении практически всей истории компьютерной графики наиболее актуальной задачей являлось получение изображений, неотличимых от реальной фотографии. Обычно для того чтобы получить изображение, задается информация о геометрии визуализируемой сцены, информация о материалах (степень отражения, преломления и т.п.), позиция и яркость источников света, положение камеры.

В основе алгоритмов фотореалистичного синтеза изображений лежат различные физические принципы. Построив достаточно сложную модель света, учитывающую все законы физики, можно добиться неотличимости синтезированного изображения от фотографии. Однако, как правило, чем больше модель света учитывает различных эффектов, тем большей вычислительной сложностью обладают алгоритмы синтеза изображений.

Рендеринг – процесс создания реалистичных изображений на экране, использующий математические модели и формулы для добавления цвета, тени и т.д. Одна из основных задач рендеринга - создание изображения, максимально приближенного к действительности. Понятно, что для этого компьютер должен учитывать все законы реального мира. По этой причине количество вычислений может быть достаточно большим.

*Целью* данной работы является программная реализация параллельных алгоритмов и технологий платформы Windows Presentation Foundation и исследование их влияния на скорость генерации растровых изображений.

Для достижения указанной цели поставлены следующие *задачи*:

* рассмотреть и изучить дипломную работу С.Д. Ким 2014 г. «Повышение реалистичности трехмерных сцен»;
* изучить основные понятия многопоточного программирования и его подходы;
* провести сравнительный анализ компоненты GDI/CDI+ и технологии Windows Presentation Foundation;
* исследовать влияние применения параллельных алгоритмов и технологии Windows Presentation Foundation на скорость генерации растрового изображения;
* программная реализация параллельных вычислений и вывода растрового изображения на экран средствами WPF.

*Объектом* исследования является процесс рендеринга трехмерных сцен.

*Предметом* исследования являются параметры, влияющие на скорость рендеринга трехмерных сцен.

Для решения поставленных задач использовались следующие *методы* исследования: теоретические (сравнение, анализ) и эмпирические (тестирование, изучение литературы и результатов деятельности).

*Научная новизна* исследования заключается в следующем:

* применение многопоточности к алгоритму удаления невидимых точек Z-буфера (известно, что исключительно удачно подходит для параллельных вычислений метод трассировки лучей, а не метод Z-буфера);
* исследование применения совместного использования параллельных вычислений и новых технологий Windows Presentation Foundation для увеличения скорости рендеринга.

*Достоверность и обоснованность результатов исследования.* Основные положения и выводы, полученные в диссертации, достаточно обоснованы и аргументированы. Сформулированная в диссертации научная задача, заключающаяся в применение параллельных алгоритмов и новых технологий для увеличения скорости рендеринга растровых изображений, была исследована и решена на основе корректного использования многопоточного программирования, технологии Windows Presentation Foundation.

Достоверность основных выводов и результатов диссертации подтверждается:

1. Обоснованием выбора технологии Windows Presentation Foundation вместо интерфейса Windows Forms, применения многопоточного программирования при рендеринге трехмерных сцен;
2. Использованием современного апробированного научно-методического аппарата для формализации и решения сформулированной в диссертации научной задачи;
3. Полнотой опубликования результатов исследования и их широко апробацией.

*Практическая значимость* полученных результатов исследований состоит в том, что разработанные алгоритмы позволяют увеличить скорость математических вычислений и вывод растрового изображения на экран.

В основу диссертационной работы положены результаты исследований:

1. Исследование влияния применения параллельных алгоритмов на скорость генерации растрового изображения.
2. Исследование влияния применения новых технологий (технология Windows Presentation Foundation) на вывод растрового изображения.

*Апробация результатов.* Результаты работы докладывалась на:

* Международной научно-практической конференции «Актуальные вопросы образования и науки», Тамбов, сентябрь 2014 г;
* II научной конференции студентов, аспирантов и молодых ученых «Проблемы моделирования, проектирования и разработки программных средств», декабрь 2014 г;
* 45-ой научно-технической конференции студентов и аспирантов «Научно-техническое творчество аспирантов и студентов», Комсомольск-на-Амуре, апрель 2015 г.

*Публикации.* По результатам выполненных в диссертации исследований автором опубликовано 3 работы.

*Структура и объем.* Магистерская диссертация состоит из введения, общей характеристик, трех глав, заключения и списка литературы. Объем работы – 94 страницы, в том числе 9 рисунков, 3 таблицы и 1 приложение.

**ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ**

*Введение* раскрывает актуальность темы, определяются цели и задачи исследования, объект, предмет, указываются научная новизна, практическая значимость, достоверность и обоснованность результатов исследования.

*В первой главе* рассматриваются основные алгоритмы затенения таких объектов, как сфера, тор и параллелепипед.

Тени придают изображения значительную реалистичность. Алгоритмы затенения в случае точечных источников света идентичны алгоритмам удаления скрытых поверхностей. В алгоритме удаления скрытых поверхностей определяются поверхности, которые можно увидеть из точки зрения, а в алгоритме затенения выделяются поверхности, которые можно «увидеть» из источника света.

Алгоритмы затенения в случае точечных источников света идентичны алгоритмам удаления скрытых поверхностей. В алгоритме удаления скрытых поверхностей определяются поверхности, которые можно увидеть из точки зрения, а в алгоритме затенения выделяются поверхности, которые можно «увидеть» из источника света

Также был рассмотрен один из самых простых и распространённых методов удаления невидимых граней – Z-буфер.

Z-буфер это область в оперативной памяти компьютера или графического контроллера, в которой для каждого пикселя изображения, формируемого на экране дисплея в процессе компьютерной сцены, хранится информация о значениях глубины. Под "глубиной" понимается расстояние от каждой точки поверхности объектов сцены до некоторой произвольной, но фиксированной поверхности, расположенной за сценой.

Главное преимущество метода Z-буфера – наиболее простой из всех алгоритмов удаления невидимых элементов сцены. А также сцены могут быть любой сложности. Поскольку элементы сцены или картинки можно заносить в буфер кадра или в Z-буфер в произвольном порядке, их не нужно предварительно сортировать по приоритету глубины. Поэтому экономится вычислительное время, затрачиваемое на сортировку по глубине.

*Во второй главе* дается понятие многопоточного программирования, его основные подходы, дается обзор класса Task.

Многопоточное программирование является одной из важных особенностей языка **C#**. Отличительной чертой многопоточной программы является то, что она состоит из двух или более частей, выполняемых параллельно. Каждая часть такой программы называется потоком и определяет отдельный путь выполнения команд. Таким образом, многопоточная обработка является особой формой многозадачности.

Благодаря встроенной в C# поддержке многопоточной обработки сводятся к минимуму или вообще устраняются многие трудности, связанные с организацией многопоточной обработки в других языках программирования.

Всего различают две разновидности многозадачности: на основе процессов (многозадачность организуется для параллельного выполнения программ) и на основе потоков (многозадачность организуется для параллельного выполнения отдельных частей одной программы).

С выпуском версии 4.0 в среде .NET Framework появились два важных дополнения, имеющих отношение к многопоточным приложениям:

1. TPL (библиотека распараллеливания задач) – усовершенствует многопоточное программирование двумя основными способами: она упрощает создание и применение многих потоков и позволяет автоматически использовать несколько процессоров. Благодаря этим двум особенностям, TPL рекомендуется в большинстве случаев к применению для организации многопоточной обработки.

Применяя TPL, параллелизм в программу можно ввести двумя основными способами:

Параллелизмом данных. Основная идея заключается в том, что одна операция над совокупностью данных разбивается на два параллельно выполняемых потока или больше, в каждом из которых обрабатывается часть данных.

Параллелизмом задач – вычислительная задача разбивается на несколько относительно самостоятельных подзадач и каждый процессор загружается своей собственной подзадачей. Чем больше подзадач, тем большее число процессоров можно использовать, тем большей эффективности можно добиться.

1. PLINQ (параллельный язык интегрированных запросов) –дает возможность составлять запросы, для обработки которых автоматически используется несколько процессоров, а также принцип параллелизма, когда это уместно.

Также было проведено исследование влияния параллельных алгоритмов на скорость генерации растрового изображения. Для этого использовалась готовая трехмерная сцена из дипломной работы С.Д. Ким 2014 г. «Методы повышения реалистичности трехмерных сцен» (рисунок 1).



Рисунок 1 – Трехмерная сцена

Изначально алгоритм рендеринга трехмерной сцены был последовательным (это описание действий, которые выполняются однократно в заданном порядке), что приводило к долгому процессу генерации изображения (к низкой производительности).

Для исследования было выбрано три компьютера со следующими характеристиками:

1 – Intel® Core™2 DUO CPU E7500 @ 2.93GHz, 2 ядра, ОЗУ 2 ГБ, видеоадаптер NVIDIA GeForce GT 240 с объемом видеопамяти 512 МБ;

2 – Intel® Core™ i7-3770K CPU @ 3.50GHz, 4 ядра, ОЗУ 8 ГБ, видеоадаптер NVIDIA GeForce GT 240 с объемом видеопамяти 512 МБ;

3 – Intel® Core™ i5-3230M CPU @ 2.60GHz, 2 ядра, ОЗУ 8 ГБ, видеоадаптер Intel® HD Graphics 4000 с объемом видеопамяти 2176 МБ.

Результаты представлены в виде диаграммы (рисунок 2).



Рисунок 2 – Технологии проведения расчетов

Результаты исследований показывают, что самым медленным по времени является последовательный алгоритм, а самым быстрым – параллельный алгоритм с десятью задачами. На первом компьютере скорость увеличилась в 2 раза, на втором – в 3,16 раза, на третьем – в 2,24 раза.

*Третья глава* содержит описание интерфейса Windows Forms, его компоненты GDI/CDI+, технологии Windows Presentation Foundation (WPF).

Сразу после своего появления платформа Microsoft .NET породила большое количество новых технологий. Не менее важной среди этих новшеств была Windows Forms – библиотека классов, необходимых для создания Windows-приложений.

Главное преимущество Windows Forms в том, что она относительно проста в изучении и по-прежнему широко поддерживается в элементах управления третьих сторон. Однако если сравнивать с WPF, Windows Forms обладает рядом недостатков:

1. Позиции и размеры элементов управления задаются в пикселях, что приводит к риску некорректного отображения приложений.
2. GDI+, который, несмотря на достаточную гибкость, медленно визуализирует крупные области.
3. У элементов управления отсутствует истинная прозрачность.
4. Трудно добиться надежности динамической компоновки.

Основная часть графики в Windows обеспечивается графической подсистемой GDI. Первоначально GDI задумывался корпорацией Microsoft как библиотека для взаимодействия с графическими устройствами, которая прилагалась бы к их продуктам. Как выяснилось впоследствии, GDI идеально подходила для операционной системы нового поколения, которую разрабатывала корпорация Microsoft и которая стала известна под именем Windows. Таким образом, до сегодняшнего дня в основу графического ядра Windows входит библиотека, первая версия которой появилась около двадцати лет назад. Этим, в общем-то, и объясняется невысокая скорость работы GDI-графики.

Одно из преимуществ использования GDI вместо прямого доступа к оборудованию — это унификация работы с различными устройствами. Используя GDI, можно одними и теми же функциями рисовать на разных устройствах, таких как экран или принтер, получая на них практически одинаковые изображения. Простые игры, которые не требуют быстрой графики, могут использовать GDI. Однако GDI не обеспечивает качественной анимации, поскольку в нём нет возможности синхронизации с кадровым буфером. Также, в GDI нет растеризации для отрисовки 3D-графики.

GDI+ — библиотека, призванная заменить существующий уже 20 лет интерфейс GDI. Хотя GDI+ — это в основном оболочка вокруг GDI, тем не менее, Microsoft посредством GDI+ предлагает ряд новых возможностей и увеличенную производительность по сравнению с некоторыми старыми средствами GDI.

GDI/GDI+ могут быть более эффективными, менее сложными, могут включать некоторые заранее подготовленные средства, чтобы не приходилось создавать их самостоятельно, однако они не могут преодолеть фундаментальные ограничения системных компонентов, разработанных более 20 лет назад.

Windows Presentation Foundation (WPF) - самая современная из предлагаемых корпорацией Microsoft технологий создания графических интерфейсов пользователя в ОС Windows, будь то простые формы, анимированные изображения, ЗD-среды с эффектом погружения или все вышеперечисленное. C WPF можно создавать широкий спектр, как автономных приложений, так и приложений, размещенных в веб-обозревателе.

В основе WPF лежит векторная система визуализации, не зависящая от разрешения устройства вывода и созданная с учётом возможностей современного графического оборудования.

Технология WPF чрезвычайно мощная и гибкая, она позволяет разрабатывать самые разнообразные приложения проще, чем когда бы, то ни было ранее. Она была разработана для преодоления недостатков и ограничений, присущих технологии Windows Forms. Производительность WPF выше, чем у GDI+ за счёт использования аппаратного ускорения графики через DirectX.

В этой главе были представлены результаты исследования применения технологии WPF на скорость вывода растрового изображения. Для исследования были выбраны те же компьютеры. Результаты представлены в виде диаграммы (рисунок 3).



Рисунок 3 – Технологии вывода изображения

Из диаграммы видно, что скорость вывода изображения на экран на первом компьютере увеличилась в 2512,4 раза, на втором – в 1550,75 раза, на третьем – в 1814,7 раза.

Также в рассматриваемой главе было проведено исследование влияния на скорость рендеринга трехмерной сцены при совместном использовании параллельных вычислений и технологий вывода графики (рисунок 4).



Рисунок 4 – Комбинирование технологий расчета и вывода

Результаты исследований показывают, что самым медленным по времени является вариант проведения последовательных вычислений и вывод с помощью технологий GDI/GDI+, а самым быстрым – параллельное выполнение десяти задач с технологией вывода WPF. Поэтому на компьютере 1 можно увеличить скорость генерации и вывода растрового изображения в 2,14 раза, на компьютере 2 – в 3,36 раза, на компьютере 3 – 2,41 раза.

*В заключении* подводятся итоги исследования, формируются окончательные выводы по рассматриваемой теме.

Было проведено сравнительное исследование влияния применения параллельных алгоритмов на скорость генерации растрового изображения; новой технологии Windows Presentation Foundation на скорость вывода растрового изображения на экран.

Результаты исследования показывают, что совместное использование параллельных алгоритмов для увеличения скорости математических расчетов и новой технологии WPF для увеличения скорости вывода растрового изображения на экран является наиболее эффективным. Это позволяет увеличить скорость генерации растрового изображения в 3,35 раза.

**СПИСОК ПУБЛИКАЦИЙ ПО ТЕМЕ ИССЛЕДОВАНИЯ**

1. Валова В.С., Лошманов А.Ю. Применение графических технологий WPF для увеличения скорости вывода растровых изображений // Сборник научных трудов по материалам Международной научно-практической конференции «Актуальные вопросы образования и науки» 30 сентября 2014 г.: в 11 частях. Часть 10. Тамбов, 2014. С. 35–37.
2. Валова В.С., Лошманов А.Ю. Исследование параллельных алгоритмов и технологий платформы WPF для генерации растровых изображений // II Научная конференция студентов, аспирантов и молодых ученых «Проблемы моделирования, проектирования и разработки программных средств». URL: <http://econf.rae.ru/article/8918> (дата обращения: 29.12.2014).
3. Валова В.С., Лошманов А.Ю. Исследование параллельных алгоритмов и технологий платформы WPF для генерации растровых изображений // Международный журнал экспериментального образования. № 3 (Ч. 3), 2015. С. 452-455. URL: http://rae.ru/meo/pdf/2015/3-3/7198.pdf (дата обращения: 11.04.2015).