

Оглавление

Введение	2
1 Аналитическая часть	4
1.1 Трассировка лучей	4
1.1.1 Классическая трассировка лучей	4
1.1.2 Трассировка лучей методом Монте-Карло	4
1.2 Квантовая обработка изображений	5
1.3 Основные положения квантовых вычислений	5
1.3.1 Квантовый бит	5
1.3.2 Квантовый регистр	6
1.4 Синтез изображения в квантовом представлении	6
1.4.1 Квантовый пиксельный шейдер	6
1.4.2 Квантовая фаза	7
1.4.3 Усиление комплексной амплитуды	7
1.4.4 Квантовая фазовая логика	8
1.5 Квантовая избыточная выборка	9
1.5.1 Основополагающая идея	10
1.5.2 Поисковая таблица	11
1.5.3 Карта достоверности	12
1.6 Колоризация изображения	12
2 Конструкторская часть	14
3 Технологическая часть	15
4 Исследовательская часть	16
Заключение	17
Литература	18

Введение

Трассировка лучей – метод геометрической оптики – исследование оптических систем путём отслеживания взаимодействия отдельных лучей с поверхностями. Специальный алгоритм отслеживает путь луча, начиная от объекта освещения до объектов, расположенных на сцене. Далее, алгоритм создает симуляцию взаимодействия с объектами: отражение, преломление и так далее. Полученная информация используется для определения цвета каждого пикселя в итоговом изображении.

Качество результирующего изображения, а также стоимость вычислений возрастают с увеличением количества испускаемых лучей. Трассировка лучей коммерчески используется в киноиндустрии [1], что мотивировано снижением затрат. Для высокобюджетного фильма с 24 кадрами в секунду, на рендеринг одного кадра уходит до двух часов [1]. Сокращение этого времени хотя бы на 50% приведет к деловой мотивации [1].

Квантовые вычисления – это альтернатива классическим алгоритмам, основанная на процессах квантовой физики, которая гласит, что без взаимодействия с другими частицами (то есть до момента измерения), электрон не размещен в однозначных координатах, а одновременно расположен в каждой точке орбиты. Область, в которой расположен электрон, называется электронным облаком. В ходе опыта Юнга, эксперимента с двумя щелями один электрон проходит одновременно через обе щели, интерферируя при этом с самим собой. Только при измерении эта неопределенность схлопывается и координаты электрона становятся однозначными.

Случаи квантового ускорения, на фоне массы классических алгоритмов, редки [2]. Однако, это не умаляет значения квантовых вычислений, потому что они способны ускорить выполнение задач переборного типа.

Цель работы – реализовать ПО, которое использует квантовый алгоритм трассировки лучей.

Чтобы достигнуть поставленной цели, требуется решить следующие задачи:

- проанализировать стандартный алгоритм трассировки лучей, чтобы понять какую часть вычислений стоит на квантовые;
- проанализировать и выбрать квантовые алгоритмы и структуры дан-

ных, которые будут использоваться в алгоритме трассировки;

- провести сравнение алгоритма трассировки с использованием квантовых и традиционных вычислений.
- доказать построенную концепцию, реализовав квантовый алгоритм трассировки лучей;

1 Аналитическая часть

1.1 Трассировка лучей

1.1.1 Классическая трассировка лучей

Трассировка лучей – метод синтеза компьютерных изображений, который позволяет добиться повышения качества за счёт повышения затрат вычислительных ресурсов. Для каждого пикселя в итоговом изображении математический луч проецируется в трехмерном пространстве из камеры через пиксел по направлению к сцене. В классическом варианте луч сталкивается с объектом сцены, и, не учитывая отражение и прозрачность, цвет объекта определяет цвет соответствующего пикселя в изображении.

Хотя при отслеживании всего одного луча на пиксель будет построено правильное изображение, удаленные подробности (например, края и мелкие детали сцены) будут потеряны. Кроме того, при перемещении камеры объекта появляются шумовые эффекты на таких объектах как листья и ограды.

1.1.2 Трассировка лучей методом Монте-Карло

Для того чтобы решить задачу без увеличения размеров изображения, программы трассировки лучей проецируют несколько лучей на пиксель, с небольшим изменением направлением для каждого луча. В таком случае, сохраняется только средний цвет, а остальные детали не используются. Процесс получения одного среднего значения называют избыточной выборкой, или выборкой по методу Монте-Карло. Трассировку лучей с использованием такого алгоритма называется трассировкой лучей методом Монте-Карло. Чем больше образцов (сэмплов) будет получено, тем ниже будет уровень шума в итоговом изображении.

В избыточной выборке возможна параллельная обработка – подсчет результатов многих лучей, взаимодействующих со сценой, но в конечном итоге используются не отдельные результаты, а только их сумма.

1.2 Квантовая обработка изображений

Технология квантовой обработки изображений использует квантовые вычисления для расширения возможностей обработки графики. Одно из применений квантовых вычислений, которое подходит под условие поставленной задачи, является квантовая избыточная выборка. Метод квантовой избыточной выборки в итоге получает результат примерно такой же, как и у существующего не квантового аналога [4]. Но, изображение полученное квантовой супер выборкой обладает другими преимуществами: средний уровень шума в изображениях, полученных в результате традиционной выборки примерно одинаков, но характер шума сильно различается [4]. В изображениях с квантовой выборкой некоторые пиксели чрезвычайно зашумлены, тогда как остальные идеальны.

Таким образом, квантовая выборка выигрывает у традиционной при необходимости пост-обработки синтезируемого изображения, например, удаления всего видимого шума. В случае квантовой выборки этот процесс будет очень прост, так как на полученном изображении будут только идеальные и чрезвычайно зашумленные пиксели.

К сожалению, для реализации полноценного квантового трассировщика лучей, потребуется намного больше квантовых мощностей, чем доступно в данный момент [4]. В данной работе будет использован метод квантовой избыточной выборки в качестве той части алгоритма, которая будет заменена на квантовые вычисления.

1.3 Основные положения квантовых вычислений

1.3.1 Квантовый бит

В квантовых вычислениях физические свойства квантовых объектов реализованы в кубитах. Классический бит принимает только два значения – 0 или 1. Кубит до измерения принимает одновременно оба значения. Из-

за этого, кубит принято обозначать выражением $\alpha|0\rangle + \beta|1\rangle$, где α и β — комплексные числа, удовлетворяющие условию (1.1)

$$|\alpha|^2 + |\beta|^2 = 1. \quad (1.1)$$

Измерение кубита мгновенно «схлопывает» его состояние в базисное — 0 или 1. Вероятности перехода в эти состояния равны соответственно $|\alpha|^2$ и $|\beta|^2$.

1.3.2 Квантовый регистр

На кубиты может быть наложена ненаблюдаемая связь — при всяком изменении над одним из нескольких кубитов остальные меняются согласованно с ним. Таким образом, можно интерпретировать такую совокупность кубитов как заполненный квантовый регистр. Такой регистр может находиться во всех комбинациях составляющих его битов, и, кроме этого, реализовывать зависимости между ними.

1.4 Синтез изображения в квантовом представлении

1.4.1 Квантовый пиксельный шейдер

Пиксельный шейдер — это программа (которая чаще всего выполняется на графическом процессоре), которая на вход принимает координаты x и y и на выходе выдает цвет пикселя, находящегося в заданных координатах. Для реализации квантового пиксельного шейдера, воспользуемся двумя квантовыми регистрами, каждый из которых состоит из N кубитов. Назовём их qx и qy для осей x и y соответственно. Таким образом, размер синтезируемого изображения зависит от величины N . Например, при $N = 4$, регистры qx и qy будут содержать $2^N = 2^4 = 16$ кубитов, что будет соответствовать размеру изображения 16x16 пикселей. Такой пик-

сель может быть только черным (1) или белым (0), потому что он будет соответствовать значениям которые может принимать один кубит.

1.4.2 Квантовая фаза

Произвольное квантовое состояние, обозначаемое $|\psi\rangle$, может быть любая суперпозиция, записанная в виде (1.2):

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (1.2)$$

базисных векторов. Согласно условию (1.1) и основываясь на том факте, что глобальная фаза не наблюдаема (то есть $|\psi\rangle$ тоже самое что и $e^{j\gamma}|\psi\rangle$) [3], выражение (1.2) можно переписать в виде (1.3):

$$|\psi\rangle = \sqrt{1-p}|0\rangle + e^{j\gamma}\sqrt{p}|1\rangle \quad (1.3)$$

где $0 \leq p \leq 1$ – вероятность того, что бит находится в состоянии 1 и $0 \leq \psi < 2\pi$ – квантовая фаза.

Применив операцию смены фазы на противоположную (повернуть кубит на 180 градусов), можно изменить состояние кубита аналогично на противоположное. Операцию смены фазы можно применять сразу на несколько кубитов. Таким образом, всего за одну операцию, можно, например, изменить цвет половины (или всех) пикселей синтезируемого изображения. Выигрыш по сравнению с обычными вычислениями очевиден. Кроме того, стало понятно, как можно изобразить прямые.

1.4.3 Усиление комплексной амплитуды

Допустим, что у нас имеется четырехкубитный квантовый регистр, который содержит одно из трех квантовых состояний, но мы не знаем какое именно. В каждом из этих состояний присутствует некоторое значение с инвертированной фазой. Назовем его помеченным значением. При чтении из квантового регистра, мы получим случайно число с равномерным распределением, и ничего не сможем узнать о том, какое из трех квантовых

состояний было исходным.

Введем зеркальную операцию. Её суть заключается в следующих действиях: выполняется инвертирование фазы, берет регистр, находящийся в состоянии $|0\rangle$ и помечает одно из значений регистра. Теперь амплитуды в каждом состоянии очень сильно различаются, и выполнение операции чтения из регистра с большой вероятностью покажет, у какого значения инвертирована фаза – а следовательно, в каком из трех состояний находился регистр.

1.4.4 Квантовая фазовая логика

Квантовая фазовая логика инвертирует фазу каждого входного значения, которое дает 1 в результате. Фазовая логика принципиально отличается от любой традиционной логики – результаты логических операций скрыты в фазах и их невозможно прочесть. Но, при этом, инвертируя фазы в суперпозиции, можно пометить несколько решений в одном регистре. Кроме того, при использовании инвертирования и усиления комплексной амплитуды можно создавать результаты, доступные для чтения.

С помощью комбинации усиления амплитуды и операций фазовой логики, можно сохранить значение логической операции в фазе состояния [4]. Таким образом, мы можем описывать более сложные фигуры.

Окружность задается уравнением вида (1.4):

$$x^2 + y^2 = r^2 \quad (1.4)$$

Предположим, что мы хотим заполнить все пиксели, находящиеся внутри окружности, то есть пиксели, подходящие под условие (1.5):

$$x^2 + y^2 < r^2 \quad (1.5)$$

Для выполнения этого действия нам потребуются выше описанные регистры qx и qy , а так же дополнительные регистр-аккумулятор $qacc$. Дальнейший алгоритм таков:

- инициализировать регистры qx , qy и $qacc$;

- ввести регистры qx и qy в суперпозицию;
- добавить в регистр $qacc$ сумму квадратов регистров qx и qy ;
- вычесть из регистра $qacc$ квадрат радиуса описываемой окружности;
- инвертировать регистр $qacc$ для всех значащих битов;
- восстановить регистр $qacc$.

1.5 Квантовая избыточная выборка

Избыточная выборка – процесс увеличения число дискретных выборок на пиксель. Если очередная выборка оказывается внутри растеризуемого примитива, её результат сохраняется в соответствующий субпиксель. В остальных случаях результат выборки игнорируется. После того, как все нужные выборки сохранены в экранном буфере, итоговый цвет пикселя определяется как усреднённый цвет всех соответствующих ему субпикселей. Таким образом, формула принимает вид (1.6):

$$res = \frac{sample_0 + sample_1 + \dots + sample_{n-1}}{n} = \frac{\sum_{i=0}^{n-1} sample_i}{n} \quad (1.6)$$

где:

- $result$ – итоговый цвет пикселя;
- n – количество выборок на пиксель;
- $sample_i$ – цвет i -ой выборки.

В случае квантовой избыточной выборки, для каждого блока необходимо оценить количество субпикселей с инвертированной фазой. Для черных и белых субпикселей (представленных инвертированной или неинвертированной фазой) это позволит нам получить значение для каждого результирующего пикселя, характеризующее интенсивность исходных составляющих субпикселей.

Преимущества использования квантовой избыточной выборки (по сравнению с обычной) связано не с количеством операций графического вывода, а с различиями в характере наблюдаемого шума. В среднем, при сравнении двух идентичных синтезируемых изображений, погрешность на пиксель у квантовой избыточной выборки на 33% ниже, чем у метода Монте-Карло (обычная избыточная выборка) [4]. Помимо этого, количество пикселей с нулевой погрешностью в среднем в два раза больше, чем у метода Монте-Карло [4].

1.5.1 Основополагающая идея

Основополагающая идея квантовой избыточной выборки заключается в применении метода объединения итераций усиления комплексной амплитуды с квантовым преобразованием Фурье. Квантовое преобразование Фурье позволит оценить количество элементов, инвертированных квантовой логикой, используемой в подсхеме инвертирования каждой итерации усиления комплексной амплитуды. В данном случае подсхемой инвертирования является программа, которая инвертирует фазу белы субпикселей.

Определим регистр квантовый регистр, который будет выполнять роль «счётчика». Значение этого регистра будет определять, сколько итераций выполнит наша схема. Введя регистр в суперпозицию, будет выполнено суперпозиция разного количества итераций усиления комплексной амплитуды. Как уже было написано выше, вероятность чтения нескольких инвертированных значений в регистре зависит от количества выполняемых итераций. Кроме этого, колебания вводятся в зависимости от количества инвертированных значений. Таким образом, при выполнении суперпозиции разного количества итераций усиления комплексной амплитуды, вводятся периодические колебания по комплексным амплитудам квантового регистра с частотой, зависящей от количества инвертированных значений.

Для чтения частот, закодированных в квантовом регистре, можно использовать квантовое преобразование Фурье [4]. Зная количество субпикселей, использованных в квантовой супер выборке, можно определить яркость анализируемого пикселя.

Качество выборки напрямую зависит от количества кубитов для «счёт-

чиков». С увеличением количества образцов (кубитов) вероятность получения точного ответа растёт [4].

1.5.2 Поисковая таблица

При запуске квантового алгоритма супер выборки и в конце его выполнения читая значение квантового регистра, мы получаем число – оно связано с количеством белых субпикселей в заданном блоке, но не будет точно равно ему.

Поисковая таблица квантовой супер выборки – инструмент для определения количества субпикселей в блоке, подразумеваемого считанным значением из квантового регистра. Например, поисковая таблица, для квантовой супер выборки с размером квантового шейдера 4×4 и регистром счетчиком, состоящим из четырех кубитов, будет выглядеть как таблица с $2^4 = 16$ столбцами и $2^4 = 16$ строками. В строках поисковой таблицы перечисляются возможные результаты чтения значения из квантового регистра. В столбцах перечисляются возможные количества субпикселей в квантовом шейдере, которые могут привести к такому значению, который был получен путём чтения квантового регистра.

При получении значения из квантового регистра, в поисковой таблице выбирается строка соответствующая считанному значению. Далее, оценивая количество «белых» субпикселей, расположенные в этой строке (а точнее лишь вероятность нахождения этих субпикселей в анализируемом пикселе), выбирается конечная яркость пикселя. Из-за того что в строчках расположены лишь вероятности, появляется некоторая погрешность и не всегда можно однозначно определить яркость пикселя.

Поисковая таблица – характерный признак алгоритма квантовой супер выборки. С увеличением размера квантового шейдера (следовательно увеличения количества субпикселей), растёт вероятность точного определения яркости выбранного пикселя.

1.5.3 Карта достоверности

Поисковая таблица также может использоваться для оценки уверенности итоговой яркости пиксела. По расположению считанного значения из квантового регистра, в строке таблицы можно оценить вероятность того, что полученное значение было правильным. Для каждого значения из выбранной строки имеется вероятность что данный субпиксель является белым. По этим результатам можно построить «карту достоверности», обозначающую вероятное расположение ошибок в синтезируемом изображении.

1.6 Колоризация изображения

Фазы и комплексные амплитуды квантового регистра можно использовать для кодирования более широкого диапазона цветовых значений (помимо белого и черного), но тогда метод квантовой супер выборки работать не будет [4].

Для колоризации изображения можно воспользоваться технологией битовых слоёв. Квантовый пиксельный шейдер будет использоваться для построения отдельных монохромных изображений, каждое из которых будет представлять один бит изображения. Таким образом, пиксельный шейдер фактически будет генерировать N монохромных изображений, где N – количество цветов, которые нужно «запутать» в изображении. Все эти N изображений пройдут квантовую избыточную выборку по отдельности и будут объединены в итоговое цветное изображение.

Вывод

В данном разделе был проведен анализ квантовых алгоритмов и структур данных, которые возможно использовать в поставленной задаче. В качестве ключевого алгоритма, который может ускорить процесс и качество синтезируемого изображения, выбран алгоритм квантовой супер выборки,

в сочетании с такими структурами данных как квантовая поисковая таблица и квантовая карта достоверности.

2 Конструкторская часть

Вывод

3 Технологическая часть

Вывод

4 Исследовательская часть

Вывод

Заключение

Литература

- [1] The Path to Path-Traced Movies [Электронный ресурс]. Режим доступа: <https://graphics.pixar.com/library/PathTracedMovies/paper.pdf> (дата обращения: 15.11.2020).
- [2] Yuri Ozhigov, Quantum Computers Speed Up Classical with Probability Zero, Chaos Solitons Fractals 10 (1999) 1707—1714. Режим доступа: xxx.lanl.gov/abs/quant-ph/9803064 (дата обращения: 15.11.2020).
- [3] IBM Quantum Experience - Quantum phase) [Электронный ресурс]. Режим доступа: <https://quantum-computing.ibm.com/docs/iqx/guide/introducing-qubit-phase> (дата обращения: 21.09.2020).
- [4] Programming Quantum Computers. Eric R. Johnston, Nic Harrigan, and Mercedes Gimeno-Segovia. 2019.