

Оглавление

Введение	2
1 Аналитическая часть	4
1.1 Основы квантовых вычислений	4
1.1.1 Квантовый бит	4
1.1.2 Квантовый регистр	4
1.2 Синтез изображения в квантовом представлении	5
1.2.1 Квантовый пиксельный шейдер	5
1.2.2 Квантовая фаза	5
1.2.3 Усиление комплексной амплитуды	6
1.2.4 Квантовая фазовая логика	6
1.3 Квантовая избыточная выборка	7
1.4 Колоризация изображения	7
2 Конструкторская часть	8
3 Технологическая часть	9
4 Исследовательская часть	10
Заключение	11
Литература	12

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;
- выбрать структуру геометрической модели сцены;

- проанализировать и выбрать квантовые алгоритмы, которые будут использоваться в алгоритме трассировки;
- доказать построенную концепцию, реализовав квантовый алгоритм трассировки лучей.

1 Аналитическая часть

1.1 Основы квантовых вычислений

1.1.1 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 $a|0\rangle + b|1\rangle$, 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 $|\beta|^2$.

1.1.2 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.

1.2 Синтез изображения в квантовом представлении

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

1.2.2 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 ку-

бит 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.

1.2.3 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.

1.2.4 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 $qacc$.

1.3 Квантовая избыточная выборка

1.4 Колоризация изображения

Вывод

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.