**Реализация метода.**

**Простейший подход.**

**Для реализации метода фотонных карт может быть использован следующий двухпроходный алгоритм. На первом проходе в соответствии с расположением и ориентацией источника света генерируются фотоны, число которых заранее задано. В простейшем случае используются точечные источники (испускающие свет равномерно по всем направлениям). При реализации разумно использовать специфику сцены и не генерировать лучи, заведомо не пересекающиеся ни с одним объектом сцены. Для каждого фотона прослеживается траектория его движения и взаимодействия с объектами сцены. При соударении с диффузным объектом возвращаются координаты найденные координаты точки соударения. При взаимодействии с объектом, обладающим отражающими или диффузными свойствами, отслеживаются дальнейшие взаимодействия в направлении отражения или преломления соответственно. Для фотонов, покинувших сцену без взаимодействия с объектами, возвращаются недопустимые координаты. Интенсивность всех фотонов при этом считается одинаковой и равной параметру . Подход может быть обобщен на случай учёта падения интенсивности фотона за счёт взаимодействия с неидеальными отражающими (или преломляющими) объектами (для этого необходимо сохранять не только координаты взаимодействия фотона с объектом сцены, но и его интенсивность на момент этого взаимодействия); или для моделирования хроматических аберраций (необходимо генерировать фотоны для каждого цвета в отдельности).**

**При втором проходе используется традиционный алгоритм трассировки лучей. При этом к вычисленной в точке соударения освещённости прибавляется интенсивность фотонов из некоторой окрестности этой точки, которые доступны с помощью обращения к созданной на первом проходе фотонной карте. Для вычисления итоговой освещённости в точке использовалась следующая формула: , где – координаты обрабатываемой точки соударения, – координаты i-ого элемента фотонной карты. Параметр ε – радиус окрестности, с которой происходит сбор фотонов. Подбирается эмпирически для каждой конкретной сцены.**

**Развитие подхода.**

**Описанный выше подход малоэффективен, поскольку на втором проходе требует для расчета освещённости в каждой точке соударения перебора всех элементов фотонной карты. В предыдущей работе было предложено отсортировать фононную карту по координатам фотонов в лексикографическом порядке. При этом вычисления на втором проходе сокращаются до перебора всех фотонов лежащих в карте между фотонами с минимальной и максимальной координатой, попадающими в ε - окрестность обрабатываемой точки соударения. Этот подход позволил значительно повысить производительность второго прохода алгоритма, но также оказался не совершенен, т.к. перебор осуществлялся не по точкам, попавшим в ε - окрестность исследуемой точки соударения (как хотелось бы), а по точкам, попавшим в «ε - полосу» (множество точек, первые координаты которых отличаются от координат точки соударения не более чем на ε). Далее будет представлен алгоритм, дающий возможность сократить перебор до некоторой окрестности точки соударения.**

**«Воксельная» сортировка.**

**Главный недостаток лексикографической сортировки в том, что она не сохраняет близость точек (близкие в пространстве точки могут оказаться далеко в отсортированной карте, далёкие – близко). Следующий метод позволяет в каком-то смысле обойти этот недостаток.**

**Разделим всё пространство на воксели со стороной ε(радиус окрестности для сборки фотонов). Тогда каждому фотону кроме его координат в пространстве можно сопоставить координаты вокселя, в котором он находится. Для этого можно использовать следующую формулу:**

**,**

**или в случае сцены конечного размера ( как это обычно и бывает в практических задачах )**

**, , где – минимальная точка сцены. В этом случае индексы принимают ограниченное положительное значение. Это обстоятельство будет использовано в дальнейшем.**

**Далее все фотоны в фотонной карте сортируются в лексикографическом порядке по координатам вокселя, в котором находятся (фотоны, попавшие в один воксель при сортировке, считаются равными).**

**Для поиска всех фотонов, лежащих в окрестности заданной точки (как того требует второй проход алгоритма) будем использовать следующее соображение: все фотоны, лежащие в ε - окрестности данной точки имеют координаты вокселя, отличающиеся не более чем на 1 от координат вокселя для текущей точки (в противном случае разность координат составляет не менее длины вокселя, которая равна ε). То есть для вычисления освещённости в конкретной точке достаточно произвести суммирование по всем фотонам из 27 вокселей. Конечная формула будет выглядеть следующим образом:**

**, где – координаты вокселя, соответствующего рассматриваемой точки,и - индексы соответственно первого и последнего фотона, лежащего в вокселе с координатами i,j,k(могут быть найдены линейным поиском по фотонной карте или предложенным далее способом).**

**Ускорение поиска.**

**Множество вокселей, на которые разбита сцена, очевидно, является дискретным и, в случае конечных размеров сцены, конечным. Поэтому возможно вычислить индексы первого и последнего фотонов в каждом вокселе один раз вместо многократного вычисления для каждой точки. В нашей работе величина ε бралась порядка 1% от размеров сцены, что порождает множество вокселей размером 100\*100\*100(всего 10 млн.). Таким образом создание ускоряющей структуры, содержащей индекс первого и последнего фотона в каждом вокселе( или невозможные значения в случае их отсутствия ), реально. Для её заполнения можно использовать следующий параллельный алгоритм.**

**Для каждого элемента отсортированной карты фотонов проверяется, является ли он первым элементом в своём вокселе (т.е. не лежит ли предыдущий в другом). В случае положительного результата номер элемента записывается в ускоряющую структуру в качестве номера первого элемента. Аналогично поступаем с последним элементом. Т.к. в каждом вокселе существуют единственные первый и последний элемент, ни каких коллизий при параллельном выполнении не возникнет.**

**Результат.**

**В результате получаем следующий четырёхпроходный алгоритм:**

1. **Генерация фотонной карты (имеет сложность O(n), где n – число генерируемых фотонов).**
2. **«Воксельная» сортировка фотонной карты. Сложность определяется используемым алгоритмом сортировки( при использовании QSort, при использовании Radix Sort).**
3. **Заполнение ускоряющей структуры( сложность O(n), где n – число фотонов в фотонной карте ).**
4. **Трассировка лучей с учётом фотонной карты. Имеет сложность , где m – число трассируемых лучей, – максимальное количество фотонов в ε-окрестности точки( в общем случае может быть принято за константу ).**

**Таким образом мы получили алгоритм реализации метода фотонных карт, имеющий сложность O(n+m), где n – число генерируемых фотонов, m – число трассируемых лучей.**