**Слайд “Цели работы”**

Добрый день, уважаемые члены экзаменационной комиссии. Я расскажу о работе по теме “современные методы растеризации изображений”, выполненной под руководством Парфенова Дениса Васильевича. Целью моей работы было **сравнение** нескольких алгоритмов трассировки лучей, реализованных в различных программах для рендеринга изображений. Проведя эксперименты мною были сделаны выводы о качестве и скорости работы этих алгоритмов.

Чтобы провести корректное сравнение необходимо было результат их работы за ограниченный промежуток времени сравнить с фотореалистичным изображением, используя определенную метрику. Фотореалистичное изображение получалось путем рендеринга условно неограниченное время.

**Слайд “Реалистичный рендеринг”**

**Актуальность работы.**

Данная работа имеет практико-ориентированную направленность.

Сейчас существует много программ трехмерной визуализации, которые применяются в различных отраслях, например, кино- и теле-индустрии, дизайне, архитектуре и т.п. Наибольшей популярностью пользуются платные пакеты, однако для некоторых расход средств на их приобретение может быть критическим, поэтому они склонны искать бесплатные аналоги. К счастью, такие существуют, но тогда встает вопрос о качестве их работы, ведь помимо цены, важнейшими характеристиками являются время и фотореалистичность получающегося изображения. Также важный фактор, влияющий на время, – это реализация распараллеливания, которую также актуально исследовать.

*На данных* изображениях можно видеть возможный результат работы каждого из исследуемых рендеров.

Например, на первом рисунке отрисовано много различных поверхностей, таких как стекло, пластик, древесина, металл и т.д.

**Важное замечание по поводу реалистичности**: Данные изображения созданы профессионалами и приведены здесь для понимания возможностей, которых можно добиться, используя их. Я не являюсь профессионалом, поэтому реалистичность полученных мною изображений является условной.

**Слайд “Концепция PBR”**

Реалистичный рендеринг базируется на 2 вещах: алгоритмы трассировки и уравнение рендеринга. Для того, чтоб реалистично отображать поверхность, необходимо понимать, как свет взаимодействует с ней. Уравнение рендеринга

Оно просчитывает энергетическую яркость точки поверхности с учетом собственной эя и эя падающего под любым углом светового луча.

L\_е в данном уравнении это собственная энергетическая яркость поверхности, те грубо говоря, собственное излучение света.

L\_i это энергетическая яркость падающего луча из направления i. это направление определяется вектором “омега малое и”.

Омега большое, это полусфера, которая учитывает любое направление i

BSDF это двунаправленная функция распределения рассеяния. Она отвечает за то, как определенная точка поверхности реагирует на падающий свет. На данном рисунке 4 вы можете видеть примеры различных поверхностей, полученных с помощью вариаций этой функции.

**Слайд “Метрика для сравнения изображений”**

Основная метрика для сравнения изображений – (peak signal-to-noise ratio)

PSNR определяется через среднеквадратическую ошибку (англ. mean squared error):

Формула 1

где I и Ihat изображения размера *m×n*, последнее из которых считается зашумленным приближением другого.

Для цветных изображений с тремя цветовыми каналами (RGB) MSE – это сумма всех квадратов разностей значений, разделенная на размер изображения и на 3. Кроме того, для цветных изображений изображение преобразуется в другой цветовой формат – YCbCr, и PSNR сообщается по каждому каналу.

Формула 2

где – MAX максимальное значение яркости пикселя изображения.

PSNR обозначает соотношение между максимумом возможного значения сигнала и мощностью шума, искажающего значения сигнала. Поскольку многие сигналы имеют широкий динамический диапазон, PSNR обычно измеряется в логарифмической шкале в децибелах. Эта величина эквивалента физической величине отношения сигнала к шуму. Чем больше изображения схожи, тем больше значение этой функции.

В данной работе использовалась реализованная в языке python соответствующая функция.

**Слайд “Алгоритмы трассировки”**

Цифровое изображение представляется в виде набора пикселей.

Для того чтобы вычислить дискретные значения пикселей, необходимо получить значение функции изображения, которой является уравнение рендеринга. Единственный способ получить информацию об этой функции – это взять ее образец с помощью трассировки лучей. Изображение может быть создано только путем выборки значений функции в конкретных позициях пикселей. Чем больше выборок, то есть чем больше итераций алгоритма, для разных пикселей, тем лучше качество.

**Первый** алгоритм называется алгоритм **трассировки пути**.

Его идея состоит в просчете путей света начиная от камеры или условного наблюдателя - на иллюстрации изображается глазом.

Попадая в любую точку поверхности, как можно видеть на правом изображении, каждый первичный луч, который обозначен большой стрелкой, формирует вторичные лучи, отскакивающие под разным углом и уходящие в пространство сцены.

Отражения вторичных лучей происходят до полного поглощения луча, которое в данном алгоритме определяется по степени вклада в конечный результат.

**Первый алгоритм** - самый простой, но и самый ресурсоемкий. Он был описан в той же статье 86 года, где впервые было описано уравнение рендеринга. Этот алгоритм заключается в симулировании пути света начиная от наблюдателя. На изображении это глаз. Попадая в любую точку поверхности, как можно видеть на правом изображении, каждый первичный луч, который обозначен большой стрелкой, формирует вторичные лучи, отскакивающие под разным углом и уходящие в пространство сцены.

Метод Монте-Карло (МС) отличается от метода трассировки лучей количеством первичных лучей: там, где трассировка лучей подразумевает группу из нескольких первичных лучей, Монте-Карло использует только один первичный луч. При отскоке от поверхности первичный луч разбивается на группу из вторичных лучей.

Вторичные лучи, в свою очередь дойдя до поверхностей объектов поглощаются. И если точка, в которой поглотился вторичный луч не была освещена источником света в сцене (например лампой), то энергетическая яркость света в этой точке будет равна 0. В каждой точке падения вычисляется, так называемая **оценка монте-карло** - числовой ряд, который приближенно вычисляет значение уравнения рендеринга.

Отражения вторичных лучей происходят до полного поглощения луча, которое в данном алгоритме определяется по степени вклада в конечный результат. Они реализуются с помощью рекурсии, которая останавливается, если энергетическая яркость близка или равна нулю.

**Второй алгоритм - Метод фотонных карт – он** состоит из двух этапов.

**На первом** этапе световые лучи, которые содержат некоторую дискретную информацию, выпускаются из источников света, как это видно на левой картинке, и для каждой точки на поверхности сохраняется информация об ее освещенности в кешэ.

**На втором этапе** лучи выпускаются из камеры и при попадании на поверхность информация об освещении собирается в некоторой окрестности точки падения, которая изображена в виде пунктирной окружности справа.

**Этот алгоритм содержит проблему кэша**, когда кол-во информации об освещении может быть ограничено количеством памяти.

В рендерах реализован улучшенный алгоритм, который называется **SPPM**, в котором, во-первых, пер**вый** и вто**рой** этапы меняются местами, а во вторых используется более ресурсо экономичная модель.

**Третий алгоритм** - двунаправленная трассировка пути, в котором строится сначала путь из камеры, на картинке х1, х2, потом строиться путь из источника света у1, у2 и потом они соединяются.

**Слайд “Сцены для сравнения рендеров”**

Сегодня я расскажу только о двух, наиболее интересных на мой взгляд, экспериментах. На данном изображении представлена эталонная сцена, с которой сравнивались полученные наборы экспериментальных изображений.

**Первый эксперимент** - модель стакана со стеклянной, преломляющей свет, поверхностью. В основном исследуется способность рендеров и алгоритмов отображать каустики.

Каустика – это огибающая световых лучей, преломленных или отраженных от некоторой кривой поверхности или объекта. Каустика возникает из-за концентрации световых лучей в определенной точке. Так как они возникают в результате преломления лучей, то в сцене присутствует непрямое освещение, которое алгоритмам рендеринга сложнее вычислять.

**В appleseed** используется второй алгоритм - SPPM

**В Cycles** используется алгоритм Path Tracing.

**В LuxCore** используется алгоритм Path Tracing + Caustic light cache;

Это технология похожа на первый этап метода фотонных карт.

**Следующий слайд - результаты первого эксперимента**

На рисунке 2 видно, что appleseed и luxcore практически с одинаковым качеством отображают первую модель. Это связано с тем, что в них используются похожие алгоритмы. В cycles же используется трассировка пути, которой сложно даются вычисления каустик. Также следует обратить внимание на скорость роста графиков, бОльшая скорость роста рендера cycles говорит о том, что он только приближается к эталону, в то время как другими рендерами был достигнут некоторый пик качества, когда последующие итерации не вносят значительный вклад в результат.

Алгоритмы трассировки хорошо распараллеливаются, тк процесс построения одного пути никак не связан с процессом построения другого пути.  
В рисунке 3 отражено процентное соотношение увеличения скорости рендеринга при увеличении кол-ва потоков вдвое: с 6ти до 12ти.

Можно видеть, что все рендеры увеличивают свою скорость примерно на 70%, кроме cycles, у которого 60%.

**Третий эксперимент**

**В третьем эксперименте** исследуется способность симулирования пропускающей свет ткани.

1. В appleseed используется трассировка пути с двумя проходами для оценки прямого освещения. Просчет каустиков отключен.

2. В Cycles используется трассировка пути.

3. Для luxcore было получено 2 набора экспериментальных изображений алгоритмами трассировки пути и двунаправленной трассировки пути.

**Следующий слайд – результаты третьего эксперимента**

Из рис. 5 видно, что наибольшую реалистичность показал рендер appleseed. Это связано с тем, что эталонное изображение было получено с помощью этого рендера. Остальные алгоритмы показали приблизительно схожие друг с другом результаты.

На рис. 6 видно, что среди них за то же время лучшее качество продемонстрировал алгоритм двунаправленной трассировки пути. Из этого можно сделать вывод, что для отрисовки данного типа сцен оптимально выбирать appleseed или luxcore.

**Слайд «Другие эксперименты»**

Также мною были проведены другие эксперименты.

**Во втором** эксперименте достаточно простая модель, однако в некоторых местах здесь достаточно сложные пути света из-за преломлений через стеклянные поверхности бутылок и отражений от зеркала.

**В четвертом** эксперименте исследуется способность рендеров реалистично симулировать поверхность жидкости. В данной сцене существует два источника света.

Данные источники состоят из материалов с собственным излучением. То есть тут как раз участвует компонента интеграла

**Следующий слайд**

**Пятый эксперимент** является наиболее сложным для алгоритмов трассировки. это модель световода. В данном эксперименте источник света находится вне области видимости камеры. Свет направлен в стеклянную трубу с индексом преломления 1.9. 3D модель сцены можно увидеть на левом изображении.

По результатам всех экспериментов, в большинстве тестов наиболее фотореалистичным являлся рендер appleseed, рендер luxcore демонстрировал похожий результат. Рендер Cycles плохо справлялся с сложными моделями освещения, но достаточно хорошо и быстро с простыми.

**Слайд «выводы»**

Таким образом, для моделей, в которых преобладает непрямое освещение, лучше использовать двунаправленный алгоритм, такой как двунаправленная трассировка или SPPM. В них лучи трассируются как из источников освещения, так и из камеры. Для простых моделей с преобладанием прямого освещения для наиболее качественного результата следует использовать трассировку пути, реализованную в рендере appleseed, если же в приоритете является время рендеринга, то лучшим выбором будет рендер Cycles.

Данный способ сравнения качества путём бескомпромиссного рендеринга лучшим рендером и сравнения с остальными по PSNR корректно отражал результаты экспериментов, состыкуясь с визуальной оценкой автора. Однако оценивание с помощью метрики PSNR все же остается лишь приближенным к человеческому виденью. Это видно по результатам третьего эксперимента, когда метрика определила схожесть, но не учла различие в способах рассеивания света тканью.

Также на основе этой работы к публикации в Российском технологическом журнале готовится статья. Спасибо. Готов ответить на ваши вопросы.

**Вопросы**

**1. Научная новизна**

Работа носит практико-ориентированную направленность, в работе представлено подробное теоретическое описание используемых технологий: концепции PBR, алгоритмов трассировки. Так что сам текст ВКР может служить хорошим теоретическим начальным пособием для интересующихся. На сегодняшний день подобного не существует в литературе на русском языке

**2. Практическое применение**

Работа носит практико-ориентированную направленность, в работе представлено подробное теоретическое описание используемых технологий: концепции PBR, алгоритмов трассировки. Так что сам текст ВКР может служить хорошим теоретическим начальным пособием для интересующихся. На сегодняшний день подобного не существует в литературе на русском языке

**3. Личный вклад**

Мною были построены 3д сцены, настроены параметры материалов для этих сцен для каждого из рендеров, в общей сложности получилось 15 сцен. По итогу мною было отрендерено более 70 экспериментальных изображений + 5 эталонных.

Также, так как моя работа носит практико-ориентированную направленность, в работе представлено подробное теоретическое описание используемых технологий: концепции PBR, алгоритмов трассировки. Так что сам текст ВКР может служить хорошим начальным теоретическим пособием для интересующихся. На сегодняшний день подобного не существует в литературе на русском языке

так что его тоже можно, как вы сказали, альбина викторовна, вставить в учебник

**4 PSNR**