**ТЕХНОЛОГИЧНО УЧИЛИЩЕ ЕЛЕКТРОННИ СИСТЕМИ**

**към ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ**

**ДИПЛОМНА РАБОТА**

Тема: Реализация на RayTracer

Дипломант: Научен ръководител:

Николай Дионисов Светлин Младенов

СОФИЯ

2015

**Увод**

Рендерирането е фундаментален компонент от компютърната графика. Рендериране се нарича процеса по превръщане на описание на три дименсиална сцена в изображение. Алгоритмите за анимация, моделиране, текстуриране и други части от компютърната графика трябва да преминат през някакъв вид рендериращ процес за да могат да бъдат превърнати в видими изображения. Техниките за рендериране са разделени в две основни групи – растеризиране и тресиране на лъчи.

Растеризирането е метод използван в интерактивни приложения, който обхожда всички видими примитиви във сцената и геометрично ги прожектира върху платно. Растеризирането само по себе си не дава много добри резултати откъм фотореализъм, но е значително по-бързо от всички останали методи. Поради тази причина това е най-разпространенят метод в гейм индустрията. Видео картите имплементират работят по този начин.

Методите за синтез на изображение базирани на Тресирането на лъчи или „Ray tracing“ са вдъхновени от реалния физичен свят. Те са симулация на света около нас. Работата на рей трейсъра се изразява в проследяването на лъчи минаващи през виртуален филм. При прекъсването на лъча от обект във сцената, част от лъча рефлектира, а друга се абсорбира. Типично за разработката на рей трейсъри се използват снимки и компютърно пресъздадени сцени от снимките. След това резултатите от програмата се сравняват с оригиналните снимки. Основният проблем на този метод е дългото време на генериране на изображение. Рей трейсинга е предпочитаният метод за рендериране, там където има възможност по-голяма част от работата да се свърши предварително.

В тази дипломна работа е представена примерна имплементация на RayTracer, тип PathTracer.

**ПЪРВА ГЛАВА: МЕТОДИ И ТЕХНОЛОГИИ ЗА РЕАЛИЗИРАНЕ НА RAY TRACER**

1. **Основни принципи**

За синтезиране на изображения ни трябва някакъв модел на светлината. Предбладаващите модели на светлината са геометричните и физичните. Геометричната оптика моделира разпространението на светлина като лъчи, които се движат в прави линии и техните пътища са представени като серия от рефлекции и рефракции. Физиката гледа на разпространието на светлината като вълнов феномен, който е повлиян от ефекти на поляризация, смущения и дифракциии.

* 1. **Основни типове рей-трейсъри**

Класическият рей трейсър, използващ геометричния модел, взима проби от идващата светлина от сцената, като пуска лъчи през съответните точки на филма и ги проследява до източниците на светлина. Изображенията генерирани от имплементацията на класическият рей трейсър може да изглежда добре, но също и нереалистично. Причината за това е, че този алгоритъм пренебрегва голяма част от домейна на интегралното уравнение, което описва сложната интеракция на светлината с обектите.

Разпределеният рей трейсър, усъвършенства класическият рей трейсър като използва Монте Карло техники. Вместо да използва резултата от само един вторичен лъч, той разпределя няколко вторични лъча за да може по-точно да пресметне нужните интеграли. Този подход е много по-скъп от страна на процесорно време, но е нужен за ефекти като меки сенки, depth of field, motion blur и други. Въпреки подобренията, които разпределеният рей трейсър внася, те не решават напълно проблема за фотореалистична графика. [1]

За да имаме реалистични изображения, трябва да взимаме в предвид и светлината отразена от останалите обекти в сцената. Този ефект в компютърната графика се нарича глобална илюминация и се противопоставя на локалната илюминация, чиито представители са представените до тук типове рей трейсъри.

Пътищния рей трейсър е разработен като решение на рендериращото уравнение и е силно базирано на Монте Карло техники. Той разглежда целия домейн на уравнението, докато разпределеният рей трейсър – само част от него. Този метод пуска не един, а много първични лъчи прези всяка една точка на филма, след което пробаблистично решава в каква посока да го отрази. Рекурсията в този метод е ненужна. Основния проблем на Пътищния рей трейсър е, че разликите в крайния резултата се възприемат като шум. При ползването на повече лъчи през всеки пиксел, покриваме по голяма част от домейна, изичисляваме интеглала с по-малка грешка и шума нямалява. Това обаче струва много прецесорно време – трябва да увеличим пробите 4 пъти, за да намалим шума двойно.[2] Възможно е да се използва техника наречена „Photon mapping“ за намаляването на шума, но тя не се разглежда тук.

* 1. **The rendering equation**

Цялата теорията върху която е изграден рей трейсингът е бязирана на т.н. “rendering equation”. Рендериращото уравнение, формулирано от Джеймс Каджия през 1986, описва движението на светлината от една точка върху повърхност до друга като сума на излъчен и рефлектиран радианс.

[IMG]

Където:

х е гочката на повърхността

у е посоката бягаща от повърхността

Л е излъчвания радианс

Л е рефлектирания радианс

Радианса ни казва колко светлинна енергия пристига или напуска дадена повърхност през дадена единица време. В вакуум радианса е константа за всяка точка по даден лъч. Това прави рей трейсинга възможен.

Рефлектираната светлина може да се опише по следния начин:

[IMG]

Този интеграл взима в предвид цялата пристигаща светлина и изчислява рефлектинарата. Той взима в предвид и светлината отговаряща за глабална илюминация.

Стандартният рей трейсър генерира по един начален лъч през всеки пиксел. При прекъсване на всеки един от лъчите, алгоритъма рекурсивно генерира нови 2 лъча. Първия е перфектно огледално рефлектираната част от лъча, а другия предадената от него част в самия обект. Посоката на вторият лъч обикновенно е събръзена с закона на Снел[\*]. Закона на Снел гласи n_1\sin\theta_i = n_2\sin\theta_t \quad където n1 и n2 са коефициентите на пречупване на сътветните среди, а sin0i sin0t са косинусите на ъглите под които падат лъчите.

Path tracing е методът използван в тази дипломна. За разлика от оригиналния рей тре

1. **Технологии за разработване на рей-трейсъри**

Използвана литература

[1] <http://web.cs.wpi.edu/~emmanuel/courses/cs563/write_ups/zackw/realistic_raytracing.html>

[2] 4 пъти – 2 по-малко