

Назаренко Д.И.

Ростовский государственный экономический университет, г. Ростов, Россия

«PHYSICALLY-BASED RENDERING» РЕНДЕРИНГ НА ФИЗИЧЕСКОЙ ОСНОВЕ

АННОТАЦИЯ

В статье обсуждается точное воспроизведение поведения реальных материалов с помощью физически точных шейдеров, основанных на использовании реалистичных моделей затенения и освещения.

КЛЮЧЕВЫЕ СЛОВА

Шейдер; рендеринг; текстура; освещение; материал; поверхность.

Nazarenko D.I.

Rostov State University of Economics, Rostov, Russia

ЗАРОЖДЕНИЕ И ЭВОЛЮЦИЯ ИГРОВОЙ ВИДЕО ИНДУСТРИИ

ABSTRACT

The article discusses the accurate reproduction of the behavior of real materials using a physically accurate shaders, on the basis of the use of realistic shading and lighting models.

KEYWORDS

Shader; rendering; texture; lighting; material.

Компьютерные игры берут свое начало в 50-ые года XX века.

До сегодняшних дней не определили, кто является их первоначальным создателем. Но определяют трех основных людей:

1. Ральф Базр – будучи инженером в 1951 году он предложил общественности идею интерактивного телевиденья.
2. А.С. Дуглас – в 1952 году им была написана игра, получившая название «ОХО», игра представляла собой программную реализацию всем известных «крестиков-ноликов».
3. Уильям Хигинботам – в 1958 году создал игру «Tennis for Two», в которой могли играть два человека.

С этого момента и примерно до 80-х годов XX века видеоигры представляли собой статичные 2D спрайты низкого разрешения. Спрос на видеоигры все рос и рос. Игровая индустрия росла и стала тянуть за собой технический прогресс компьютеров.

В 1987 году появился видеоадаптер VGA, а следом SVGA. Эти события отправили в прошлое 16 цветное царство. Теперь на мониторах 256 цветов, что естественно сделало игры более красочными и относительно похожими на современные.

Компьютерные игры в 90-е годы XX века

Начинания с 90-х бум компьютерных игр стал только набирать обороты, а индустрия расти и расти. В 1993 году 10 декабря компанией Id Software, был выпущен великий Doom. Игра, которая заложила основы жанра шутер (Shooter).

В 1994 году появилась первая игра с мультиплеером – Rise of the Triad. А на следующий год появляется The Terminator: Future Shock, первый шутер с элементами трехмерности мира и врагов, а так же свободным обзором при помощи мышки.

В 1996 году на свет появилась Voodoo I – первая видео карта с поддержкой полноценного 3D. Это позволило выпустить Duke Nukem 3D и Quake – первые полностью трехмерные игры. Так же в этом году появились такие игры, как Super Mario, Command & Conquer: Red Alert, Tomb Raider, Resident Evil, Diablo и многие другие.

В половине 90х годов началось действительно активное прогрессирующее развитие игровой индустрии. Компьютеры стали более мощнее, а самое главное – доступнее. С приходом 3D технологий, компьютерные игры стали более походить на современные и спрос на них увеличился.

Стали появляться первые прототипы шейдеров, рендеринга и 3D – моделирования.

К началу 2000-х годов сфера применения данных технологий перестала ограничиваться компьютерными играми. Их начали применять в мультфильмах, анимационных фильмах и 3D

моделирования.

Основы

Шейдер (англ. *shader* — затеняющая программа) — компьютерная программа, предназначенная для исполнения процессорами видеокарты (GPU). Шейдеры состоят из одного из специализированных языков программирования (см. ниже) и компилируются в инструкции для GPU.

Шейдеры применяются в программах, работающих с трёхмерной графикой и видео (игры, GIS, CAD, CAM и др.), используют шейдеры для определения параметров геометрических объектов или изображения, для изменения изображения (для создания эффектов сдвига, отражения, преломления, затемнения с учётом заданных параметров поглощения и рассеяния света, для наложения текстур на геометрические объекты и др.).

Рендеринг (англ. *rendering* — «визуализация») — термин в компьютерной графике, обозначающий процесс получения изображения по модели с помощью компьютерной программы. Здесь модель — это описание любых объектов или явлений на строго определённом языке или в виде структуры данных.

Текстура — это картинка, графическое изображение, которое наносится на стороны объектов. Благодаря текстурам, мы отличаем песчаную дорожку от асфальтированной, деревянный ящик от каменной глыбы, кирпичную стенку от скалы и т.д.

Physically-Based Rendering

Физически обоснованный рендеринг или рендеринг на физической основе (PBR) берет за основу идею использования реалистичных моделей затенения / освещения вместе с заданными значениями свойств поверхности с целью точного воспроизведения реальных материалов.

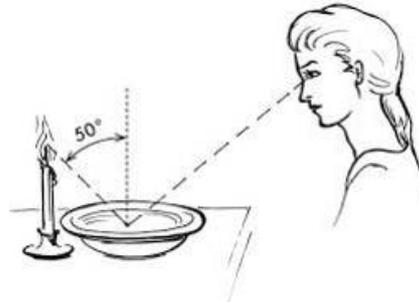
PBR и PBS это технология шейдинга и визуализации, которая более полноценно и точно описывает-диктует освещение и материалы. Физически корректную визуализацию сокращенно называют PBR (Physically-Based Rendering), а физически корректный шейдинг PBS (Physically-Based Shading). Соответственно, в определенном контексте, PBS может упоминаться только в отношении шейдинга, а PBR – в отношении визуализации и освещения. Однако, по сути, оба понятия относятся к одному и тому же, то есть, к технологии корректной визуализации, основанной на научном подходе.

Преимущества

1. Намного легче создавать реалистичные вещи, так как отпадает необходимость в бесконечном переборе множества вторичных параметров материала (таких как Specular). Пользователь задает лишь несколько фундаментальных характеристик (как обычно, через текстурные карты) – все остальное, что нужно, вычисляется на основе точных физических формул и алгоритмов.
2. Модели выглядят одинаково адекватно и круто во всех мыслимых условиях освещения и затенения
3. Сам процесс разработки более логичный, интуитивный и линейный, не смотря на широчайший спектр возможных применений и реализаций.

Структура

Шейдеры и рендеринг в общем тесно взаимодействуют с физикой и ее законами. Даже самые первые и примитивные шейдеры были основаны на основных законах отражения света.



Объект отражает лучи света от его источника, которые в свою очередь попадают в человеческий глаз. Таким образом мы и воспринимаем объекты вокруг нас. Следовательно, если бы не было этого источника света, то человеческий глаз его не увидел.

С помощью данных законов художники научились создавать текстурные карты (Texture

Map), описывающие свойства поверхности.

Самые первые и примитивные шейдеры, которые использовались в игровых объектах, использовали только диффузную текстуру (Diffuse Texture). Которая в свою очередь представляет из себя просто картинку, которая используется для отличия одного объекта от другого.

Постоянно развивающаяся мощь компьютерной индустрии теперь стала позволять более точно симулировать физику света. Например, отражение, поглощение, преломление и сохранение. На основе которых были разработаны карты нормалей (Normal Map) и карты отражения (Specular Map), которые лишь приблизительно имитировали поведение материалов про освещении.

Сохранение энергии

Понятие сохранения энергии гласит, что объект не может отражать больше света, чем принимает.



(Наглядная демонстрация принципа работы сохранения энергии)

В практическом смысле, это означает, что более грубые материалы, склонные к поглощению света будут иметь тусклые и широкие блики, а гладкие и более отражающие материалы – яркие и резкие.



(Half life 1999г)

Рост вычислительной мощности и повсеместная потребность в стандартизации художественного контента приводит к тому, что физически обоснованный рендеринг быстро становится стандартом в индустрии, нацеленным на переосмысление того, как мы производим и рендерим арт.

Что использует Physically-Based Rendering:

- Альбедо [Albedo]

Albedo, это входной параметр базового цвета, широко известный как Diffuse Map.



(Пример принципа работы Albedo)

Карта Albedo задает цвет диффузного света. Одним из крупнейших различий между Albedo в PBR и традиционной Diffuse Map является отсутствие засвеченного направленного света или Ambient Occlusion. Иными словами, Albedo не повторяет ошибок Diffuse Map, когда тот выглядел либо тусклым, либо засвеченным. запеченный свет в некоторых условиях освещения будет смотреться неправильно, а Ambient Occlusion следует назначать в специальный слот под АО.

Albedo иногда задает больше, чем просто диффузный цвет. Например, при использовании Metalness Map, Albedo задаёт диффузный цвет для диэлектриков и Reflectivity для металлических поверхностей.

- **Карта нормалей [Normal map]**

Это такая карта, которая задает векторы нормалей для поверхности. Она используется для генерации поверхностей, BumpMarring-a и прочих алгоритмов. Иначе говоря, эта текстура, с помощью которой определяются места выпуклостей и вогнутостей на поверхности, которые в свою очередь преломляют световые волны, что придает большую реалистичность графики.

- **Микроповерхность [Microsurface] или [Gloss Map]**

Microsurface задает насколько гладкой или шероховатой является поверхность материала.



(Пример принципа работы Gloss Map)

На картинке выше видим, как принципы сохранения энергии подвергаются влиянию микроповерхности материала. На грубых поверхностях блик шире, но тусклее, в то время, как на гладких – ярче, но резче.

В зависимости от движка, под который вы готовите контент, карта может называться Roughness Map вместо Gloss Map. На практике разницы особой нет, хотя Roughness Map может иметь инвертированные значения, то есть темные места – глянцевая/гладкая поверхность, а светлые – грубая/матовая.

- **Отражательная способность [Reflectivity]**

Reflectivity, это процент света, который поверхность отражает.



(Пример принципа работы Reflectivity)

Важно отметить, насколько узкий диапазон Reflectivity отводится диэлектрикам. Прибавив к этому закон сохранения энергии, легко сделать вывод, что вариации поверхности должны передаваться через Microsurface Map, а не через Reflectivity Map. Для заданного типа материала, Reflectivity остается практически постоянной. Цвет отражений у непроводников обычно нейтральный/белый, и цветной у металлов. Таким образом, карта, специально направленная на определение цвета/силы отражений (известная как Specular Map) может быть отброшена в пользу Metalness Map.

- **Френель [Fresnel]**

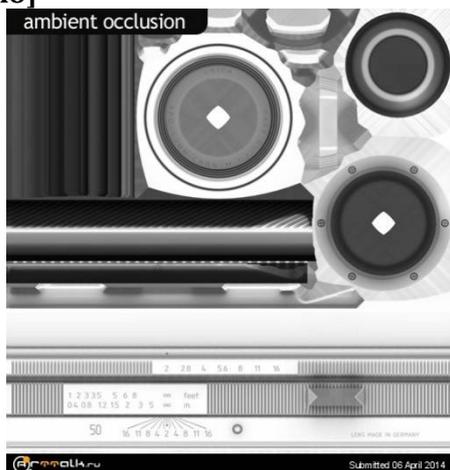
Fresnel, это процент света, отражаемого поверхностью под крутыми углами.



(Пример принципа работы Fresnel)

Fresnel обычно стоит ставить на 1 (если в модуле Reflectivity выбран *mat Metalness*, то Fresnel блокируется в положении 1), так как все виды материалов становятся на 100% отражающими под критическими углами. Вариации микроповерхности, из-за которых эффект Френеля становится более или менее выраженным, автоматически учитываются по содержимому Gloss Map.

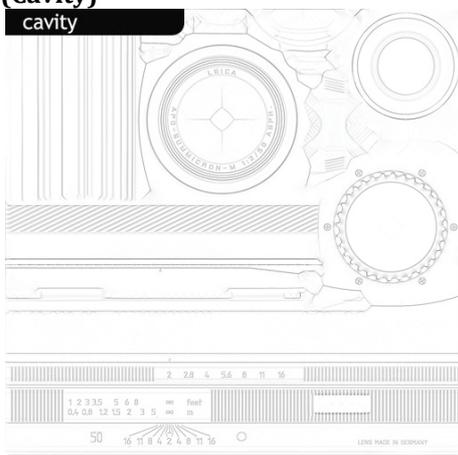
- **Ambient Occlusion [AO]**



(Примерный вид карты AO)

Добавление AO в качестве отдельной карты, вместо того, чтобы запекать в Albedo и Specular map дает шейдеру возможность использовать его более разумно. Например, Функция AO затемняет окружающий рассеянный свет (diffuse компонент image based lightning-системы в Toolabag 2), а не направленный свет от динамических источников света или specularных отражений любого вида.

- **Трещины\полости (Cavity)**



(Примерный вид карты Cavity)

Cavity Map представляет собой затененный свет в мелком масштабе и обычно запекается с 3D модели или из Normal Map.

Cavity map должна содержать только углубления (ямы) поверхности, а не выпуклые места, так как Cavity Map накладывается через Multiply. Содержимое должно в основном быть белым с темными местами, передающими впалые участки поверхности, где будет теряться свет. Cavity Map влияет и на Diffuse и на Specular компоненты окружающего и динамического освещения.

Создание

Для создания материалов для PBR существуют два метода:

1. Ручной – когда дизайнеры используют фотографию и на ее основе создают текстурную развертку для 3D модели.
2. Автоматизированный – этот метод подразумевает использование дизайнерами программ процедурного генерирования. Самые популярные из них:
 - 2.1. Quixel Suite – плагин для Photoshop;
 - 2.2. Substance Painter;

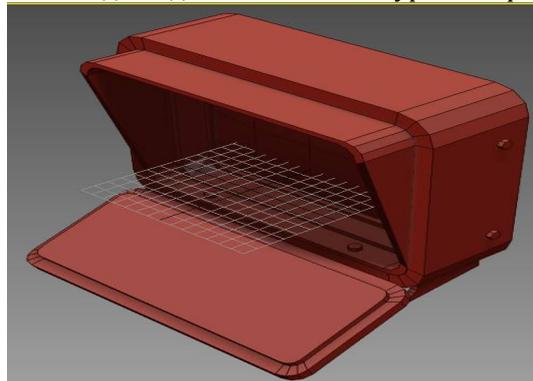
Пример

Наглядный пример с помощью программы процедурного генерирования – Quixel Suite 2.0

В качестве 3D модели я использую данную модель Терминала, созданная лично мной.

3D модель состоит из вершин и полигонов, находящиеся в 3D пространстве и обладающие своими координатами.

Перенос этих полигонов из трехмерного пространства на 2D сетку дает нам текстурную развертку. На основе, которой и создают дальнейшие текстурные карты.



(Модель в трехмерном пространстве)



(Та же самая модель, но уже в виде текстуры на 2D сетке)



(Albedo Map)



(Specular Map)

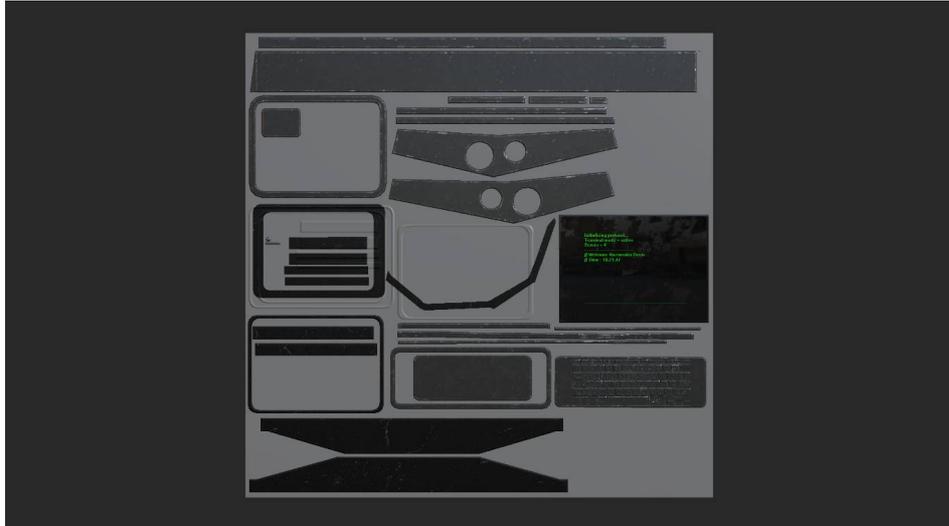


(Normal Map)



(Emission Map)

Для создания именно этой модели мной была использована дополнительная карта, отвечающая за свечение отдельных элементов – Emission Map. В данном случае – для текста на экране терминала.



(Когда текстуры накладываются друг на друга – конечный результат в 2D)

Конечный результат рендера



(Terminal – фронт)



(Terminal – бок 2)



(Terminal – бок 3)

На примере выше видно, как различные материалы этого Терминала реагируют на внешние источники света.

Видно, как стекло на экране имеет большую отражающую способность в отличие от железного корпуса и пластмассовых деталей и можно даже увидеть отраженную окружающую среду. А также через дополнительную карту, а именно Emission Map, была реализована светящаяся панель управления данного терминала, что придает ему особой реалистичности.

Литература

1. Jeff Russell PBR-Theory/. URL:<http://www.marmoset.co/toolbag/learn/pbr-theory>
2. Jeff Russell PBR-Theory-RUSSIA/. URL: <http://art-talk.ru/topic/10504/>
3. Статья Sebastien Lagarde Adopting a physically based shading model/. URL:<http://seblagarde.wordpress.com/tag/physically-based-renderi...>
4. URL: <http://art-talk.ru/topic/10547/>
5. Отчет Sébastien Lagarde Rendering Remember Me /. URL:<http://www.fxguide.com/featured/game-environments-parta-reme...>
6. Слайды Real Shading in Unreal Engine 4 /. URL:<http://blog.selfshadow.com/publications/s2013-shading-course...>
7. Статья Mike Seymour Monsters University: rendering physically based monsters/. URL:<http://www.fxguide.com/featured/monsters-university-renderin...>

References

1. Jeff Russell PBR-Theory/. URL:<http://www.marmoset.co/toolbag/learn/pbr-theory>
2. Jeff Russell PBR-Theory-RUSSIA/. URL: <http://art-talk.ru/topic/10504/>
3. Stat Sebastien Lagarde Adopting a physically based shading model/.
4. URL: <http://art-talk.ru/topic/10547/>
5. URL: <http://seblagarde.wordpress.com/tag/physically-based-renderi...>
6. Sébastien Lagarde Rendering Remember Me /. URL: <http://www.fxguide.com/featured/game-environments-parta-reme...>
7. Real Shading in Unreal Engine 4 /. URL:<http://blog.selfshadow.com/publications/s2013-shading-course...>
8. Mike Seymour Monsters University: rendering physically based monsters/. URL:<http://www.fxguide.com/featured/monsters-university-renderin...>

Поступила 25.05.2016

Об авторах:

Назаренко Денис Игоревич, студент факультета компьютерных технологий и информационной безопасности Ростовского Государственного Экономического Университета (РИНХ), Mail – den_nazarenko_00@mail.ru.