

# Сравнительный анализ современных рендер-систем для 3ds Max

В. Фролов, В. Галактионов, М. Трофимов

Институт прикладной математики имени М.В. Келдыша, Москва, Россия  
vova@frolov.pp.ru, vlgal@gin.keldysh.ru, trofimovmax@mail.ru

## Аннотация

В данной статье представлен сравнительный анализ производительности современных рендер-систем. Нами было проведено сравнение среди 7 наиболее популярными продуктов для 3D Studio Max (Mental Ray, IRay, VRay3, VRayRT3, Corona, Octane, Hydra) на скорость интегрирования освещения на 7 различных сценариях. В нашей работе нас, прежде всего, интересовало сравнение законченных реализаций в целом в промышленном решении в противовес сравнению отдельных алгоритмов в одной и той же системе. Мы постарались ответить на некоторые важные вопросы: что на практике дают последние достижения в области алгоритмов, такие, как SPPM [1] и VCM [2]? Есть ли выигрыш от GPU? Смещенные и несмещенные рендер-системы – что и когда лучше? Какие алгоритмы несут наибольшую практическую пользу?

На основе проведенного исследования нами были сделаны заключения о современном положении дел в области рендер-систем, преимуществах и недостатках CPU и GPU реализаций, а также о достоинствах и недостатках различных алгоритмов, применяемых в исследуемых системах и возможных векторах развития. Рендер-системы были ранжированы по очкам относительной производительности для каждого из сценариев и в сумме. Наши тестовые сцены и данные находятся в открытом доступе.

**Ключевые слова:** сравнение, рендер-системы, CPU, GPU.

## 1. ВВЕДЕНИЕ

Сравнение современных рендер-систем – нетривиальная задача, встречающая определенный ряд сложностей.

Во-первых, недостаточно сравнить системы на одной или двух сценах, поскольку существует достаточно большое число причин, по которым расчет освещения может стать медленным. Необходимо покрытие достаточно широкого спектра сценариев. Во-вторых, во время проведения сравнения необходимо добиваться максимально-возможного совпадения условий освещения, поскольку даже незначительное различие в условиях освещения может серьезно влиять на производительность. Например, еле заметное для глаза (из-за гамма-коррекции) увеличение яркости источника освещения в 1.5 раза может вызывать увеличение времени синтеза изображения до 2 раз, поскольку скорость сходимости метода Монте-Карло пропорциональна квадрату времени расчета, а яркость источника напрямую влияет на абсолютное значение ошибки метода Монте-Карло. В-третьих, системы, использующие смещенное решение (такие, как Mental Ray, VRay и Hydra), не всегда позволяют напрямую регулировать время синтеза и точность получаемого изображения, что чрезвычайно усложняет процесс получения изображения с заданным качеством или временем. Огромное число настроек различных алгоритмов в этих системах создает простор для экспериментов, и опытный пользователь может добиться более быстрого и точного синтеза, чем новичок. Наконец, рендер-системы не

идентичны по набору функциональности, моделям материалов и источников света. Например, источник типа 'Небесный Портал' (Sky Portal), позволяющий значительно ускорить расчет дневного освещения в помещениях корректно, реализован не во всех системах.

## 2. РЕНДЕР-СИСТЕМЫ

### 2.1. VRay и Mental Ray

Mental Ray – один из пионеров среди систем рендеринга со смещенным решением [3]. Со временем система не потеряла популярность и продолжает активно развиваться. Используется преимущественно в мультипликации. Ключевой особенностью системы является высокая степень гибкости, позволяющая дизайнеру настроить расчет в точности, так как нужно. Ключевыми алгоритмами расчета являются кэш освещенности в сочетании с финальным сбором, использующим фотонные карты [3]. Алгоритм интерполяции освещенности интересен тем, что использует не строго ближайшие точки, как в классической реализации кэша освещенности, а фильтрует освещение, учитывая многие точки. Mental также может использовать фотонные карты для расчета каустиков.

VRay [4] – одна из наиболее популярных рендер-систем в архитектуре. VRay – CPU рендер со смещенным решением. Ключевыми алгоритмами являются кэш освещенности (называемый в VRay термином 'Irradiance Maps') и карты светимости (называемые в VRay термином 'Light Cache'). Каустики в VRay могут рассчитываться при помощи фотонных карт. Для сравнения была использована новейшая версия VRay 3.0.

### 2.2. IRay, VRayRT, Octane

IRay является гибридной CPU/GPU рендер-системой с несмещенным решением, частично совместимой с Mental Ray. IRay использует GPU для ускорения трассировки лучей [5]. На основе документации в IRay реализован один из вариантов алгоритма переноса света Метрополиса. VRayRT – частично совместимая с VRay GPU рендер-система с несмещенным решением. В системе реализована только простая Монте-Карло трассировка путей. Эффективный расчет каустик невозможен. Система Octane [6] является одной из первых рендер-систем с несмещенным решением, реализованная полностью на GPU. В Octane реализован метод PMC [7].

### 2.3. Corona и Hydra

Corona [8] – дебютированная в 2012 г. CPU рендер-система со смещенным и несмещенным решениями. В отличие от VRay и Mental Ray, Corona использует кэш освещенности только для третьего переотражения, выполняя, таким образом, попиксельно финальный сбор. Отличительной чертой системы Corona является алгоритм Vertex Connection Merging (VCM) [2], значительно ускоряющий расчет каустик. Кроме этого, в системе Corona реализована двунаправленная трассировка лучей (BDPT) и двунаправленные прогрессивные фотонные карты (BDPM). Hydra [9] – дебютированная также в 2012 г. GPU рендер-система Hydra является системой со смещенным и

несмещенным решениями. Ключевыми алгоритмами являются кэш освещенности и стохастические прогрессивные фотонные карты (SPPM) [1] для расчета каустик.

### 3. МЕТОДИКА СРАВНЕНИЯ

Для того чтобы решить указанные во введении проблемы, применялась следующая методика:

1. Для сравнения была выбрана одна среда (3D Studio Max), что позволило добиться геометрического совпадения элементов сцены, включая положения источников света и камеры.
2. Для каждой системы и каждого сценария рассчитывался свой эталон, с которым впоследствии производилось сравнение. Это позволяет исключить различия реализации отдельных незначительных моментов и сосредоточиться на сравнении скорости интегрирования.
3. Для всех систем сравнивались два типа. Сначала фиксировалась небольшая ошибка (насколько это возможно) и измерялось время синтеза изображения. Так, сравнивалось время синтеза качественного изображения. Затем фиксировалось время синтеза изображения (обычно небольшое, в пределах 1 минуты) и измерялась ошибка. Сравнивалось качество полученных изображений в условиях фиксированного времени. Из двух сравнений для систем со смещенным решением в результирующий график входил средний индекс производительности. Такой способ позволяет частично решить проблему оптимизации настроек для систем со смещенным решением, поскольку на итоговый результат влияют оба типа оптимизаций (ориентированные на качество и скорость). Для систем с несмещенным решением соотношение скорость/качество для обоих типов сравнений совпадает.

#### 3.1. Индекс производительности

Для того чтобы оценить производительность рендер-систем на различных сценах и сопоставить их друг с другом, введем абсолютный индекс производительности (1).

$$P_t = \frac{1}{MSE * MSE * t} \quad (1)$$

Здесь  $MSE$  - квадратичная ошибка (Mean Square Error), вычисляемая при помощи программы 'The Compressorator', а  $t$  - время в секундах. Если сцена, условия освещения и оборудование фиксированы, отношение индексов производительности для двух систем будет адекватно отражать отношение производительности этих систем. Однако зависимость сравнения от оборудования и абсолютные значения индекса снижают наглядность сравнения. По этой причине введем относительный индекс производительности (2).

$$P_{I(Rel)} = 100 * \frac{P_t}{MAX(P_{I(VRay)}, P_{I(MentalRay)}, P_{I(IRay)}, \dots)} \quad (2)$$

Относительный индекс производительности будет равен 100 баллам для системы, которая является на данной сцене самой быстрой. Остальные системы получают баллы пропорционально тому, насколько они медленнее. Таким образом, относительный индекс производительности не зависит от сложности сцены, и его можно усреднить по всем тестовым сценам, оценив, насколько система проявила себя в комплексе.

## 4. ЭКСПЕРИМЕНТЫ

Все эксперименты производились на машине с CPU Core i7 3770 3.4 Ghz, и GPU Nvidia GTX680. Память типа DDR3, 16 Gb. В силу различия архитектуры CPU и GPU мы опирались на компромиссное решение, основанное на выборе оборудования одного ценового диапазона. Все ошибки были посчитаны при помощи программы 'The Compressorator' [10].

### 4.1. Тестовые сценарии (рис. 1)

Сценарий номер 1 – Cornell Box с зеркальным чайником. Несмотря на свою простоту в данном сценарии присутствуют практически все основные эффекты трехмерной компьютерной графики: шумное первичное освещение и мягкие тени, зеркальные блики от источника освещения, отраженные каустики. Являясь геометрически простой, сцена в некоторой степени амортизирует стандартные потери производительности GPU трассировщиков лучей на ветвлениях, а CPU трассировщиков лучей – на кэш промахах.

Сценарий номер 2 – уличная (outdoor) сцена. Данная сцена, являясь геометрически сложной (из-за травы), тем не менее, с точки зрения освещения достаточно проста – однородная панорама и один относительно неярко источник освещения. Время синтеза изображения на такой сцене должно в основном быть обусловлено скоростью трассировки лучей.

Сценарий номер 3 – верхний коридор Crytek Sponza. В данном сценарии практически все видимое освещение вторичное. Силен вклад от второго и третьего диффузных переотражений, поэтому финальный сбор должен давать значительное ускорение. Указанный сценарий наиболее точно отражает скорость расчета вторичного освещения. Сценарий нельзя назвать 'сложным для расчета' (hard sampling), поскольку яркость солнца относительно невелика и солнцем освещена значительная часть поверхности сцены.

Сценарий номер 4 – зал конференций. Сложность расчета освещения в данной сцене заключается в большом числе источников освещения. Под потолком находится 20 источников прожекторного типа. Тем не менее, каждый из источников светит на относительно небольшой участок сцены, поэтому, если в рендер-системе реализована эффективная схема сэмплирования источников, в каждой конкретной точке сцены большинство источников должны быть эффективно отброшены или учитываться относительно нечасто.

Сценарий номер 5 – освещение Небесными Порталами. Данная сцена демонстрирует достаточно типичный сценарий расчета дневного освещения в помещении. При таком сценарии напротив окон ставят источники света, имитирующие свет от внешнего окружения, проникающий через окно. Такой источник называется Небесным Порталом (Sky Portal). При правильной реализации он является полностью корректным решением. Такой источник света служит средством расчета освещения от окружения при помощи явной стратегии (стратегии сэмплирования источников света). Иными словами, Небесный Портал является не самостоятельным источником света, а всего лишь подсказкой для Монте-Карло трассировки лучей, позволяющей вычислять освещение от окружения более эффективно в тех случаях, когда внутри помещения видима относительно небольшая часть окружения. Тем не менее, при использовании небесных порталов расчет первичного освещения в

определенной степени осложняется по двум причинам. Во-первых, небесные порталы могут иметь значительные размеры, в результате чего замедляется расчет мягких теней. Во-вторых, число источников этого типа может быть достаточно большим (5 – 10), что еще более осложняет вычисление первичного освещения. На данной сцене были использованы различные комбинации методов для различных систем, наилучшим образом показавшие себя.

**Сценарий номер 6** – тест на MLT (Metropolis Light Transport). В данном сценарии небольшой участок сцены освещается исключительно ярким направленным источником света, имитирующим солнце. Вторичное освещение, вызванное таким освещением, является трудным для расчета (hard sampling). Фотонные карты в сочетании с финальным сбором (как и карты светимости) амортизируют увеличение времени расчета только за счет ускорения вычисления компоненты от третьего и более переотражений. Однако вычисление компоненты от второго переотражения света данным методом не ускоряется. С другой стороны, алгоритм переноса света Метрополиса (MLT) и аналогичные алгоритмы при условии корректной реализации их в той или иной системе должны давать на данной сцене значительное ускорение по сравнению с традиционным методом Монте-Карло и финальным сбором.

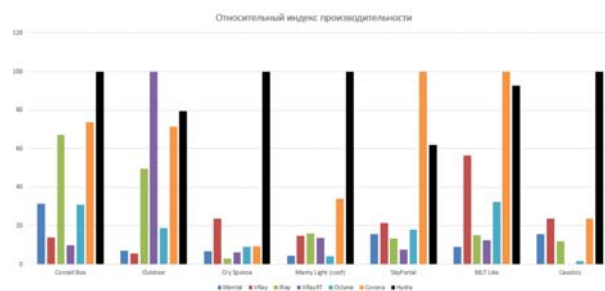
**Сценарий номер 7** – водные каустики. В данной сцене присутствуют отраженные каустики и подводные каустики SDS типа (Specular Diffuse Specular), являющиеся сложными для расчета. IRay и VRayRT не были способны корректно рассчитать данный тип каустиков за приемлемое время, хотя система IRay способна эффективно считать каустики, видимые напрямую (ESDL).

## 5. ВЫВОДЫ

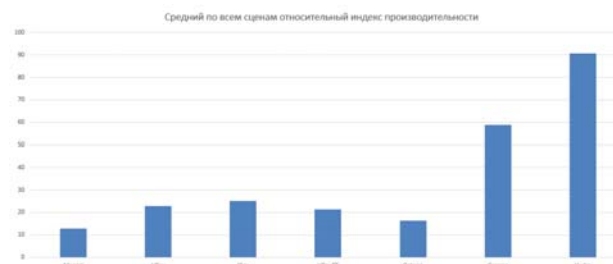
Анализируя графики на рис. 1 и 2, можно заключить, что несмещенные рендер-системы на GPU (IRay, VRayRT, Octane) обеспечивают в лучшем случае паритет с Mental Ray и VRay, а в худшем – проигрывают последним в 1.5 – 2 раза. Выигрыш этих систем (IRay, VRayRT, Octane) на открытой сцене (сценарий 2) можно объяснить простой формой освещения, при которой время рендера зависит только от скорости трассировки лучей. С другой стороны, GPU и рендер-система на основе алгоритмов со смещенным решением Hydra показала значительный отрыв по скорости (в 4 – 5 раз) при вычислении вторичного освещения (сценарий 3) и каустиков (сценарий 7). На каустиках второе место занимает Corona с алгоритмом VCM. Несмотря на такое же соотношение скорости у VRay, в области вычисления каустиков мы отдадим ей третье место, поскольку в VRay сложнее добиться каустиков высокого качества, так как в этой системе фотонные карты не прогрессивные. Однако на вторичном освещении VRay прочно обосновалась на втором месте. Рендер-система Corona лучше всего себя показала при расчете освещения от небесных порталов (сценарий 6) и в условиях сложного диффузного освещения (сценарий 7). Неплохие результаты в среднем могут говорить о том, что классический подход вычисления освещения при помощи финального сбора и карт светимости и сейчас является весьма практичным решением.

Сравнивая скорость интегрирования каустиков в CPU рендерах VRay и Corona, мы приходим к выводу о том, что внимание, прикованное к алгоритму VCM в последние годы, возможно, завышено. Реализация этого алгоритма на практике в системе Corona дала выигрыш только на два

сценариях (1 и 7) и только за счет каустик. При этом каустики могут быть столь же эффективно вычислены при помощи более простого в реализации алгоритма SPPM, а во всех остальных случаях VCM был значительно медленнее (до 10 раз), чем стандартный вариант Corona с попиксельным финальным сбором. Что касается MLT, ни Octane, ни IRay, предполагающие реализацию идей MLT, не показали ожидаемого для этого алгоритма увеличения производительности по сравнению со стандартной Монте-Карло трассировкой. Метод PMC в системе Octane позволил за приемлемое время рассчитать каустики в бассейне, однако он приблизительно на два порядка проигрывает алгоритму SPPM, реализованному в системе Hydra.



**Рис. 1. Сравнение рендер-систем по относительному индексу производительности (на различных сценах)**  
[http://ray-tracing.ru/upload/free/hydra/comparison\\_final.pdf](http://ray-tracing.ru/upload/free/hydra/comparison_final.pdf)



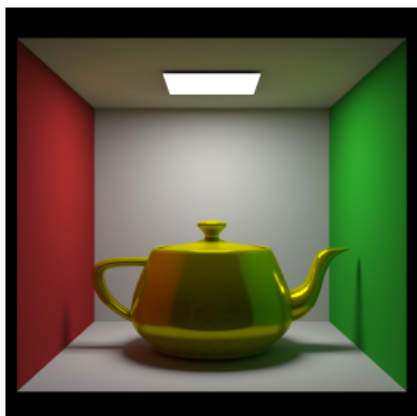
**Рис. 2. Сравнение рендер-систем по относительному индексу производительности (среднее по всем сценам)**  
[http://ray-tracing.ru/upload/free/hydra/comparison\\_final.pdf](http://ray-tracing.ru/upload/free/hydra/comparison_final.pdf)

## 6. ССЫЛКИ

- [1] Hachisuka T., Jensen H.W. Stochastic progressive photon mapping, ACM Trans. Graph., vol. 28, no. 5, dec. 2009. – P. 141:1 – 141:8
- [2] Georgiev I, Krivanek J, Slusallek Ph. Bidirectional light transport with vertex merging in SIGGRAPH Asia, 2011 Sketches, New York, NY, USA, 2011, SA '11. – P. 27:1 – 27:2, ACM.
- [3] NVIDIA, “Mental ray”, 2014.
- [4] Chaos Group, “Vray,” 2014.
- [5] Nvidia, “Nvidia iray faq,” 2014.
- [6] OTOY, “Octane renderer,” 2014.
- [7] Yu-Chi Lai, Shaohua Fan, Stephen Chenney, and Charcle Dyer, “Photorealistic image rendering with population monte carlo energy redistribution,” Eurographics Symposium on Rendering, 2007.
- [8] Ondra Karlik, Adam Hotovy, Jaroslav Krivanek, Jakub Konecny, and Ludvik Koutny, “Corona render,” 2014.
- [9] Ray Tracing Systems, “Hydra renderer,” 2014.
- [10] AMD, “The compressor,” 2014.

## Благодарности

Работа поддержана грантами РФФИ № 12-01-00560, № 13-01-00454 и стипендией президента РФ СП-4053.2013.5.



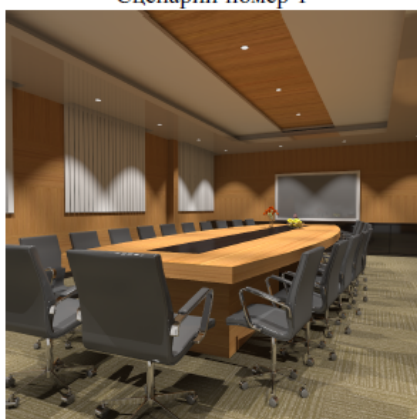
Сценарий номер 1



Сценарий номер 2



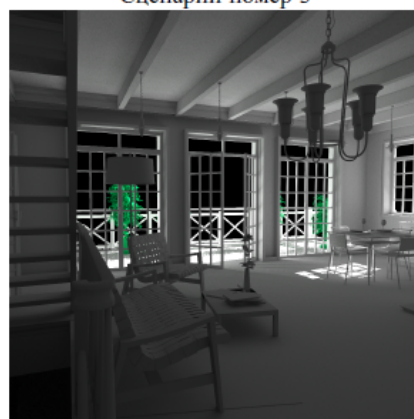
Сценарий номер 3



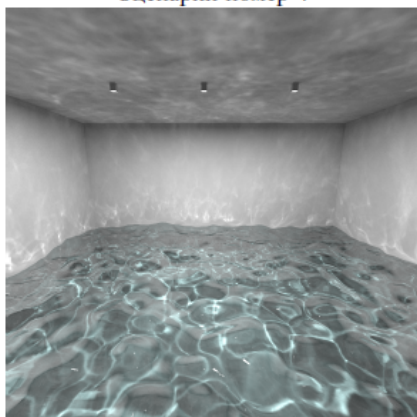
Сценарий номер 4



Сценарий номер 5



Сценарий номер 6



Сценарий номер 7

**Рис. 3. Изображения сцен, на которых проводились сравнения. Достигнута высокая степень совпадения для всех рендер-систем. Полные данные доступны по адресу: [http://ray-tracing.ru/upload/free/hydra/comparison\\_final.pdf](http://ray-tracing.ru/upload/free/hydra/comparison_final.pdf)**