

Hydra Renderer FAQ's

Список наиболее часто-задаваемых вопросов

Необходим ли 3ds Max для запуска рендера?	2
Насколько быстрый у вас трассировщик лучей?	2
На чем написана GPU часть рендера?	2
Что конкретно в рендере реализовано на GPU?	2
Какие алгоритмы реализованы в рендере?	2
Как алгоритмы связаны с режимами рендера во вкладке 'Methods'?	2
Какие типы максовских объектов поддерживаются сейчас?	3
Что пока не поддерживается но скоро будет?	3
Куда сохраняются картинки?	3
Где сохраняется время рендера?	3
Что с progress bar-ом?	3
Как работать с внешним gui?	3
Как сделать tone mapping?	4
Как сделать Final Gathering?	4
Что представляет из себя механизм профилей?	4

Необходим ли 3ds Max для запуска рендера?

Для демонстрации работы рендера макс не обязателен; stand-alone версия управляется через hydra_gui.exe и открывает сцены в формате .dae или .vsgf (внутренний).

Насколько быстрый у вас трассировщик лучей?

Мы периодически проводим сравнение с самой быстрой из известных GPU реализаций [1] и стабильно достигаем 70% их скорости. Вы можете провести сравнения самостоятельно, запустив hydra_gui.exe и включив статистику (Ray Tracing -> Statistics). Список Measure показывает, какие типы лучей замеряются в данный момент.

На чем написана GPU часть рендера?

В основном CUDA, немного OpenGL3+. Последнее используется для отображения различной отладочной информации вроде положений фотонов.

Что конкретно в рендере реализовано на GPU?

Все алгоритмы реализованы полностью на GPU. Центральный процессор (Ц.П.) занимается только построением BVH дерева перед началом рендеринга. В процессе рендера Hydra не грузит Ц.П. Вы можете параллельно работать с другими программами, не использующими интенсивно GPU.

Какие алгоритмы реализованы в рендере?

В настоящий момент Hydra использует 4 основных расчетных алгоритма:

1. Адаптивную трассировку путей - Path Tracing [2].
2. Кэш освещенности - Irradiance Cache (IC) [3].
3. Стохастические прогрессивные фотонные карты - SPPM [4].
4. Хитрый многомерный фильтр на основе [5] (режим Multi-Layered).

Как алгоритмы связаны с режимами рендера во вкладке 'Methods'?

Как правило, различные режимы используют комбинацию методов.

1. Path Tracing - обыкновенная трассировка путей.
2. Irradiance Cache + Path Tracing - сначала строится кэш освещенности, после чего адаптивной трассировкой путей просчитывается финальное изображение. Подробно можно прочитать в [2].
3. Path Tracing + SPPM (caustics) - режим использует SPPM для расчета каустиков и трассировку путей для остальных компонент освещения.

4. Path Tracing + SPPM (caustics) + IC - то же что и предыдущее, но вторичную диффузную освещенность вычисляет при помощи кэша освещенности.
5. SPPM (secondary) - использует SPPM для вычисления вторичной диффузной освещенности.
6. SPPM (all) - использует SPPM для вычисления и первичной и вторичной освещенности.
7. Multi-Layered - отдельно вычисляет изображения падающей вторичной освещенности, фильтрует при помощи многомерного фильтра из [5] и комбинирует результат. В настоящий момент над методом идет активная работа. Он неплохо показывает себя на архитектурных сценах.

Какие типы максовских объектов поддерживаются сейчас?

1. Материалы - Standart и свои hydraMaterial.
2. Текстуры - Bitmap и NormalBump.
3. Источники - все стандартные, квадратный mr area и сферический mr area.

Что пока не поддерживается но скоро будет?

Анимация, Multi-GPU, Очень Большие Текстуры, Сеть (возможность рендерить на удаленной машине), разные хитрые модели материалов и источников (пишите нам - какие именно Вам нужны больше всего!), импорт VRay-евского контента.

Куда сохраняются картинки?

В рабочий каталог hydra.exe (как правило это 'C:/[Hydra]/bin'). Для каждого рендера сохраняется обычное изображение, hdr изображение и изображение с примененным tone-mapping ом (с суффиксом 'tm').

Где сохраняется время рендера?

К имени сохраняемого изображения дописывается время рендера в секундах.

Что с progress bar-ом?

Он показывает что что-то происходит, но пока не сильно правильно оценивает время. Поэтому рендер может завершиться на 60-80% или прыгнуть сразу с 0 до 50%. Не обращайте на это внимания, это нормально. Рендер завершиться только когда закончит расчет в соответствии с установленными настройками (в текущей версии поменять эти настройки вы можете только через внешний gui).

Как работать с внешним gui?

Сначала запускаете `hydra_gui.exe` вручную. В папке со сценой (где лежит максовский файл) создаете пустой текстовый файл с именем `'dummy.dae'`. Далее `File->Open Scene`, указываете `'dummy.dae'`. Путь необходимо указывать на случай, если вы будете использовать механизм профилей. Затем в максе, в интерфейсе рендера отмечаете галочку `'Use External Hydra Gui'` и нажимаете `'Render'`. После этого рендер запустится с собственным окном, в котором Вы сможете полетать по сцене. Важный момент: если хотите перемещаться по сцене, в разделе камеры задайте тип камеры `'Euler'`. Если хотите экспортировать камеру из макса, задайте тип камеры как `'UVN'`.

Как сделать tone mapping?

В текущей версии для этого придется использовать внешний графический интерфейс. Регулируете силу тон-маппинга при помощи двух слайдеров (`White Point` и `Strenght`) и нажимаете кнопку `'Tone Map Again'` чтобы увидеть результат. Кнопка иногда работает не с первого раза. Нужно надавить, подержать и отпустить. Не забудьте внизу вместо `'Show simple image'` выбрать `'Show tone mapped image'`. Галочка `'Bloom'` пока не работает. Вы также можете использовать любую стороннюю программу для реализации более сложного тон-маппинга, скамливая ей `.hdr` или `.tiff` файл, в котором сохраняется оригинальное HDR изображение.

Как сделать Final Gathering?

К сожалению в текущей версии с этим придется немного помучиться. Запускаете рендер с использованием внешнего `gui` (сначала запускаете `gui` вручную, после чего в плагине нужно отметить галочку `external gui` и запустить нажав `'Render'`). Идете во вкладку `SPPM (diffuse)`, отмечаете галку `'Enable'`, ставите `gather bounce` в 0. Нажимаете `'Single Pass'` и приблизительно оцениваете плотность получившейся фотонной карты. Поскольку FG использует фотонную карту лишь как грубое приближение светимости, вам не нужна высокая плотность. Это только снизит скорость. Подбираете параметром `Initial Radius` и числом фотонов то что хотите получить в качестве этого грубого приближения светимости на глаз. После этого ставите `gather bounce` в 1, параметр `'Re-Trace Each Pass Of'` ставите где-нибудь в районе 20, отмечаете галочку `'Progressive'` и нажимаете кнопку `'Path Tracing'`.

Что представляет из себя механизм профилей?

Данный механизм предназначен для исправления ошибок импорта из внешних файлов. Также использовался на ранней стадии разработки плагина. Если по какой-то причине материал или источник был импортирован из внешнего файла или макса неправильно, у вас есть возможность его исправить. Для этого во вкладке `File` выбираете `Edit Hydra Profile`, после чего открывается текстовый редактор `xml` - файла `hydra_profile.xml`. Этот файл представляет из себя список всех правок для объектов сцены. Для того чтобы исправить некорректно-импортированный материал/источник, вам нужно знать его имя. В файле `hydra_profile.xml` в нужной секции (для материалов это `library_material`) вы в `xml` формате, в соответствии с документацией вы описываете параметры вашего материала или источника. Ниже показано как выглядит описание материала-стекла с именем `'MyGlass'`.

```
<material name= "MyGlass">
```

```

<hydra>

  <specular>
    <brfd_type> phong </brfd_type>
    <color>0.5 0.5 0.5 </color>
    <cos_power> 1e+006 </cos_power>
  </specular>

  <reflectivity>
    <brfd_type> fresnel_dielectric </brfd_type>
    <color> 1.0 1.0 1.0 </color>
  </reflectivity>

  <transparency>
    <color> 1.0 1.0 1.0 </color>
    <thin_surface> 0 </thin_surface>
    <cos_power> 1e+006 </cos_power>
    <IOR> 1.5 </IOR>
    <fog_color> 0.95 0.95 0.95 </fog_color>
    <fog_multiplier> 0.5 </fog_multiplier>
    <exit_color> 0.095 0.095 0.095 </exit_color>
  </transparency>

</hydra>
</material>

```

Пожалуйста, воздержитесь от использования unicode символов в именах материалов и источников.

Список литературы

- [1] *Aila, T. and Laine, S. 2009.* Understanding the efficiency of ray traversal on GPUs. // In Proceedings of the Conference on High Performance Graphics 2009 (New Orleans, Louisiana, August 01 - 03, 2009). S. N.
- [2] *Frolov V., Kharlamov A., Ignatenko A.* Biased Global Illumination via Irradiance Caching and Adaptive Path Tracing on GPUs // In Proceedings of the Graphi'Con 2010 Conference. Moscow, 2010.
- [3] *Frolov V., Vostryakov K., Kharlamov A., Galaktionov V.* Implementing Irradiance Cache in a GPU Realistic Renderer. // Trans. on Comput. Sci. XIX, LNCS 7870. 2013. Vol. 7870, no. 1. P. 17–32.
- [4] *Hachisuka T., Jensen H. W* Stochastic progressive photon mapping // ACM Trans. Graph. 2009.— Vol. 28, no. 5. P. 141:1–141:8.
- [5] *Gastal E. S. L., Oliveira M. M.* Adaptive Manifolds for Real-Time High-Dimensional Filtering. // ACM TOG. 2012. Vol. 31, no. 4. P. 33:1–33:13. Proceedings of SIGGRAPH 2012.