

DIGITAL PRODUCTION

MAGAZIN FÜR DIGITALE MEDIENPRODUKTION

JULI | AUGUST 04:2018



Renderer

Alle relevanten Renderer –
und ein paar mehr!

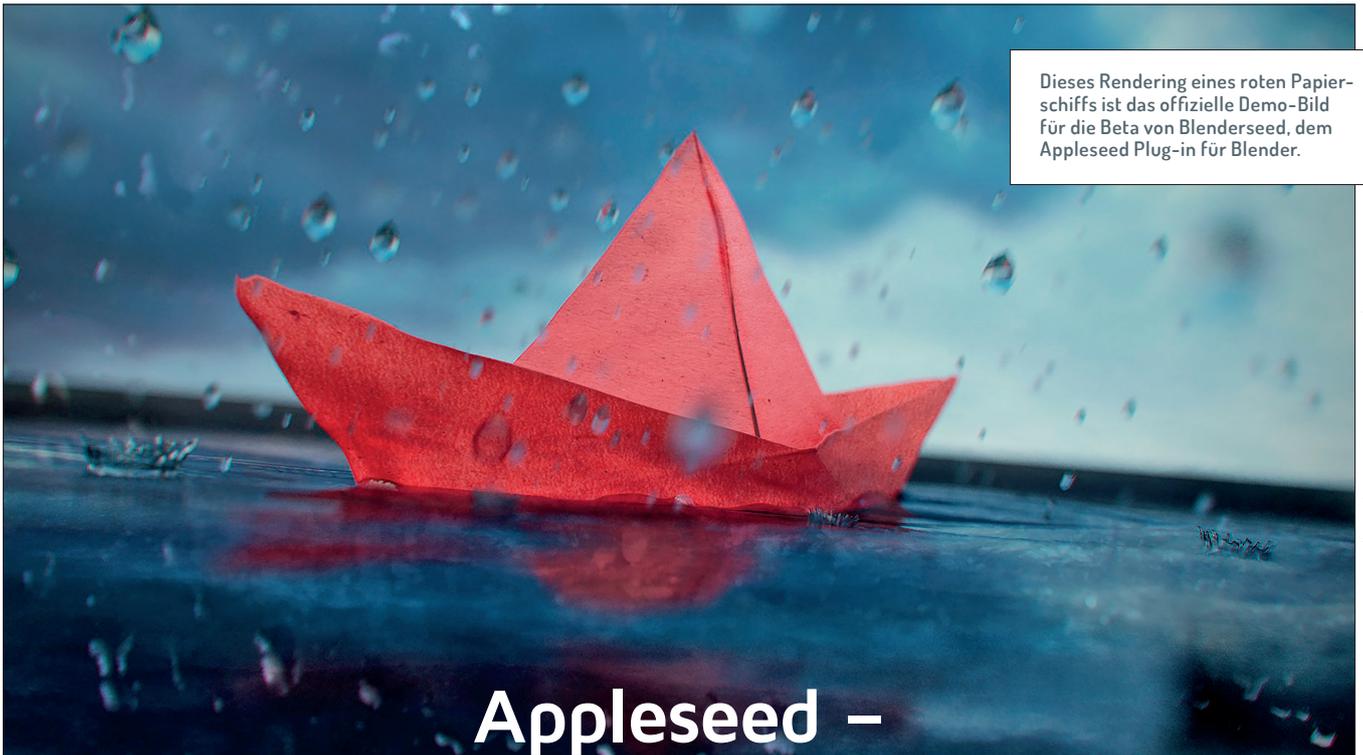
Blender

Cycles, Bforartists, Blender
One und Ton Roosendaal!

und vieles mehr!

Von Athera über Jim Knopf
und Resolve bis ZBrush





Dieses Rendering eines roten Papierschiffs ist das offizielle Demo-Bild für die Beta von Blenderseed, dem Appleseed Plug-in für Blender.

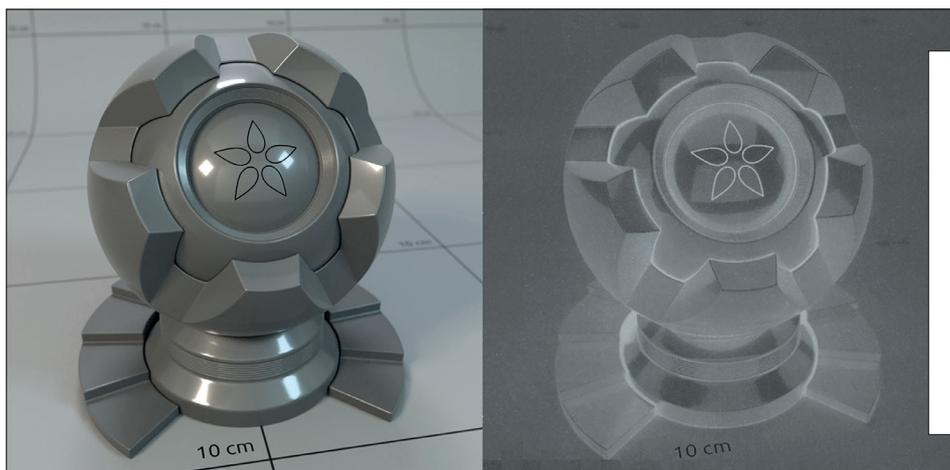
Bild: AppleseedHQ, Giuseppe Lucido, zaldor.artstation.com

Appleseed – Open Source Path Tracing Engine für 3ds Max, Maya und Blender

Cycles ist nicht die einzige Open Source Render Engine, die auf Path Tracing setzt und für den Produktiveinsatz gedacht ist. Es gibt noch mindestens eine andere – Appleseed, die wir in DP 2015:05 bereits vorgestellt haben. Anders als Cycles setzt Appleseed ausschließlich auf CPUs zum Rendern, dafür gibt es aber offizielle Plug-ins für 3ds Max und Maya. Auch bei Appleseed hat sich in den letzten Jahren einiges getan. Mit der Unterstützung der Open Shading Language (OSL) von Sony Pictures Imageworks im neuen Release 1.9 wurde ein großer Meilenstein erreicht.

von Gottfried Hofmann

Bild: Appleseed Release Notes



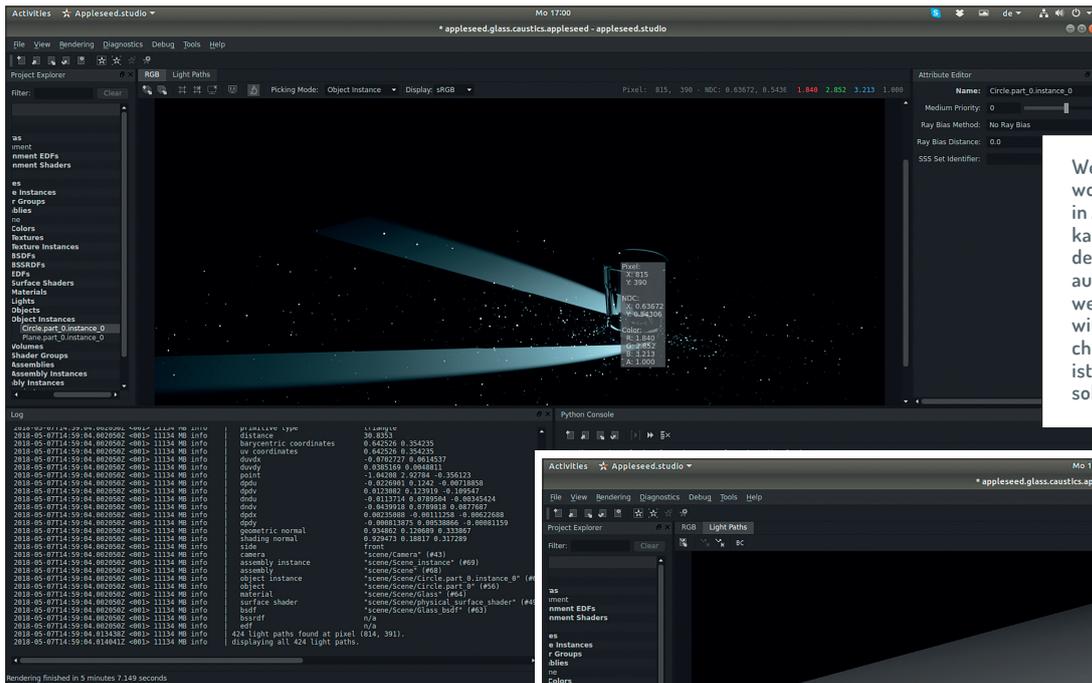
Die Renderzeit pro Pixel im Median kann als AOV bzw. Render Layer ausgegeben werden. Heller bedeutet längere relative Renderzeit. An diesem Bild erkennt man sehr gut, dass konkave Bereiche mehr Renderzeit benötigen als konvexe, da die Strahlen mehr herumspringen müssen. Besonders bei den Spalten im Modell ist der Effekt deutlich zu sehen.

Maya, Max, Blender, Gaffer

Die Anzahl der Beitragenden für das aktuelle Release liegt mit 20 Personen deutlich über der eingeschworenen Gemeinschaft, die im Jahr 2015 an Appleseed gearbeitet hat. Für ein Open-Source-Projekt ist das ein gutes Zeichen.

Wer nach einer Rendering-Lösung sucht, die in Blender, 3ds Max, Maya und Gaffer (gafferhq.org) die gleichen Ergebnisse zeigt, sollte zumindest mal ein Auge auf Appleseed werfen, zumal die Integration in bestehende Pipelines dank der neu hinzugekommenen

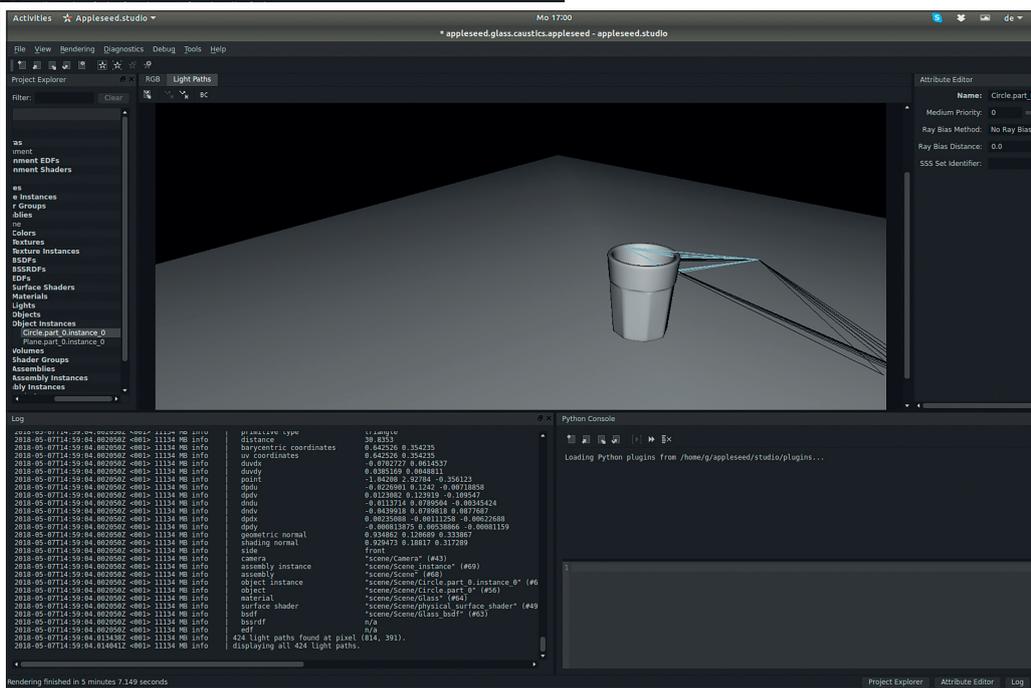
OSL-Unterstützung nochmals einfacher geworden ist. Für Substance Painter wird zudem ein Export-Skript mitgeliefert. Maps, die mit diesem Preset exportiert worden sind, lassen sich sehr einfach mit dem Standardmaterial von Appleseed nutzen.



Wenn man in einer Szene wissen möchte, woher die Fireflies stammen, kann man sie in Applesseed Studio, der Standalone-Applikation von Applesseed, laden und die Pfade der Strahlen aufzeichnen lassen. Ein Klick auf einen Pixel, und in einem extra Tab werden sie visualisiert. In diesem Fall sehen wir, dass das Glas Grund für die Glühwürmchen ist. Für herkömmliches Path Tracing ist eine solche Szene eine harte Nuss. Hier sollten wir auf SPPM umsteigen.

Nodes mit Open Shading Language

OSL hat auch die Bedienung mächtiger gemacht, da man dadurch endlich seine Shader mittels Node-Netzwerken aufbauen kann. Und natürlich kann man seine eigenen Skripte verwenden. OSL bringt auch eine Reihe nützlicher prozeduraler Texturen mit, darunter das berühmte Perlin Noise, Gabor, Voronoi und Simplex. Für die Zukunft sind weitere prozedurale Texturen geplant. Zentraler Shader ist sowohl beim Einsatz von



Kaustiken gefällig? Dank Stochastic Progressive Photon Mapping (SPPM) kann Applesseed Kaustiken relativ effizient rendern.





Der in Appleseed integrierte Denoiser ist darauf ausgelegt, bei hoher Sample-Dichte den letzten Rest Rauschen zu entfernen. Sprich er ist auf Qualität und nicht auf Geschwindigkeit ausgelegt, womit er sich von den Denoisern anderer Engines unterscheidet, die darauf abzielen, bereits bei geringer Sample-Anzahl ein rauschfreies Bild zu erzeugen, was auf Kosten feiner Details geht.

Bildquelle: AppleseedHQ, Giuseppe Lucido, zaldor.artstation.com

OSL als auch beim alten Interface der Disney Shader, welcher inzwischen in so vielen Render Engines Einzug gehalten hat, dass man ihn getrost als neuen Standard ansehen kann.

TD-Fest

Neben OSL dürften Technical Directors (TDs) die vielen Analysetools ansprechen, die Appleseed mitbringt. So kann man einen Render Pass bzw. eine AOV ausgeben, die die Renderzeit für jeden Pixel in Form von Graustufen speichert. Damit lassen sich die Bereiche einer Szene erkennen und isolieren, die deutlich mehr Rechenzeit benötigen als der Rest. Ein guter Anhaltspunkt für Optimierungen.

Strahlen visualisieren

Das mitgelieferte UI „Appleseed Studio“ bringt neben extensiven Logs noch weitere Werkzeuge mit. So können während des Rendervorgangs mit nur geringem Overhead die Pfade sämtlicher Strahlen aufgezeichnet werden. Man kann dann die Szene als Appleseed-Projekt exportieren und in Appleseed

Studio kurz den Rendervorgang starten und wieder abbrechen. Dann kann man einzelne Pixel auswählen und die Strahlen anzeigen lassen, die zu besagtem Pixel beigetragen haben. Natürlich kann man auch nur in Appleseed Studio rendern und dann die Debug-Tools nutzen – ein sehr nützliches Feature, das als Teil eines größeren Projekts aus dem Bereich der industriellen Visualisierung entwickelt wird. Ziel sind Werkzeuge zur Analyse der Streuung von Licht in und auf Maschinen und Werkzeugen.

Für diejenigen, die Plug-ins entwickeln oder Appleseed in bestehende Pipelines integrieren wollen, findet sich sowohl eine Python- als auch eine C++-API; unter Open Source Render Engines für den Produktiv-einsatz ein Alleinstellungsmerkmal.

Der eigene Anspruch

Die Entwickler von Appleseed beschreiben die Ziele des Projekts in vier Punkten. Erstens soll Appleseed vollständig Open Source sein, damit man jedes noch so kleine Detail nachsehen und verändern kann. Zweitens soll aktuelle Technologie eingesetzt werden, spricht wenn ein nützliches Paper im Bereich

Path Tracing erscheint, sollte es kurze Zeit später auch in Appleseed implementiert sein.

Der dritte Punkt ist eigentlich der wichtigste, nämlich dass die Software „Production Grade“ ist. Viele andere Open Source Render Engines kranken daran, dass sie eigentlich aus einem akademischen Kontext heraus entwickelt wurden und selbst kaum bis gar nicht in Projekten außerhalb des Elfenbeinturms eingesetzt werden. Denn nicht jeder Algorithmus ist robust genug für den Alltagsgebrauch. Andere wiederum erfordern Parameter und Feintuning, die erst nach intensivem Gebrauch in realen Projekten gefunden werden. Production Grade heißt auch, dass Features nicht halbherzig implementiert werden.

Der vierte Punkt heißt: Realismus ohne Kompromisse. Andere Engines geben zu einem gewissen Grad die physikalische Korrektheit auf, um dafür schneller rendern zu können. Appleseed will diese Kompromisse nicht eingehen, was besonders erfreulich ist für diejenigen, die im Visualisierungsbereich arbeiten, wo ein hoher Grad an Realismus erforderlich ist.

Pläne für die Zukunft

Appleseed beherrscht zwar bereits spektrales Rendering, allerdings noch keine Dispersion. Diese soll im nächsten Release, das ab September erwartet werden kann, nachgeliefert werden. Das Plug-in für Blender soll interaktives Rendering ähnlich wie Cycles erhalten, und es sind Demo-Szenen für Max, Maya und Blender geplant, um den Einstieg zu erleichtern. Insgesamt ist Politur angesagt. Die Dokumentation soll verbessert, die Auswahl an OSL-Shadern um nützliche Helferlein wie z.B. Ambient Occlusion oder Pointiness erweitert und die Benutzeroberflächen sowohl von Appleseed Studio als auch der Plug-ins von technischem Kaunderwelsch befreit werden. Eventuell wird auch bidirektionales Path Tracing rechtzeitig fertig. Unterstützung für IES-Lichtprofil-Dateien, BRDFs aus Messdaten und diverse Falschfarbendarstellungen aus dem Bereich Lichtsimulation sind ebenfalls auf der Roadmap, wenn auch in etwas fernerer Zukunft.

Fazit

Appleseed steht momentan quasi in den Startlöchern. Der nächste Release im September sollte die letzten Hürden für den Produktiveinsatz nehmen. Gegenüber dem Stand von 2015, als wir das erste Mal über Appleseed berichteten, hat sich enorm viel getan. Appleseed wird tatsächlich schon kommerziell eingesetzt, wenn auch noch eher im R&D-Bereich. >ei