

# State of the Art in Photon Density Estimation

## (S)PPM in LuxRender

Guillaume Bouchard

Université Claude Bernard Lyon 1  
Laboratoire d'InfoRmatique en Image et Systèmes d'information (LIRIS)  
Centre National de la Recherche Scientifique (CNRS)  
LuxRender Developer



**LuxRender**  
GPL PHYSICALLY BASED RENDERER

**Version 1.0RC3**

Released on July 30th, 2012 - Licensed under GPL v3 - [www.luxrender.net](http://www.luxrender.net) - Artwork by Mourelas Konstantinos

# LuxRender Details

- ▶ Fork of PBRT I
- ▶ GPL v2 Software
- ▶ C++ on Linux, Windows, Mac OsX, (Android?)

# LuxRender Features

- ▶ Concurrency (GPU, Threading, Networking)
- ▶ Bidirectional MLT path tracing
- ▶ Integration with many modeling tools
- ▶ Complex materials
- ▶ Volumetric rendering
- ▶ Resuming
- ▶ Light Groups
- ▶ Spectral Rendering
- ▶ ...



Threads: 2



Display Interval: 15 sec



700 x 700



100%



## Tone Mapping

Kernel:

Linear

Sensitivity



800

800.0

Exposure



1/4

0.250

FStop



22

22.00

Gamma



Presets:

2.20



Color Space



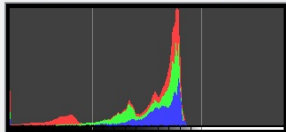
Gamma + Film Response



HDR Histogram

Channel:

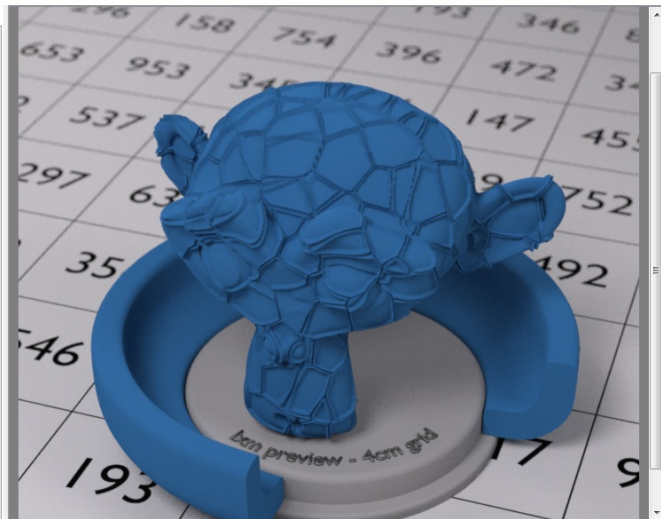
RGB

Output: ☐ Log

Reset

☒ Auto

Apply



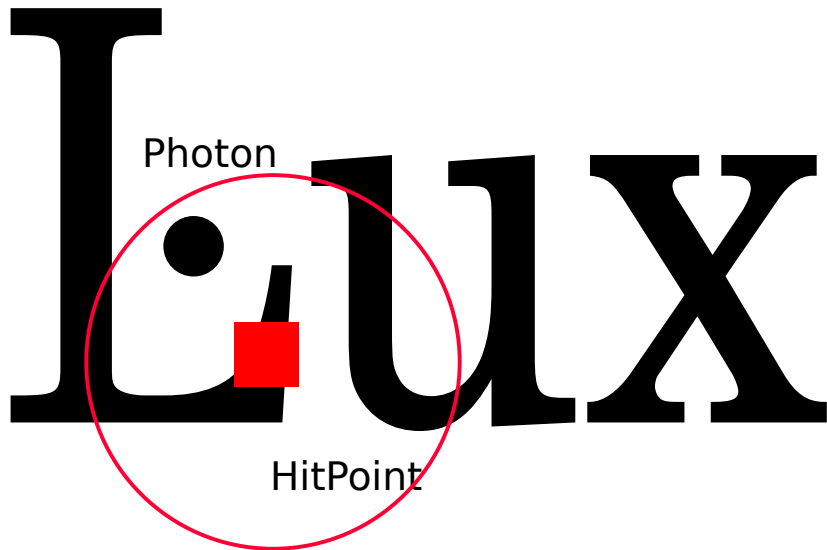
# Design Choices

- ▶ Statistics-less radius reduction heuristic
- ▶ Eye pass first, photon pass next
- ▶ Global radius
- ▶ Hash Grid for photon density estimation

# Memory Usage Discussion

Storing HitPoints + Using BSDF HitPoints  $\Rightarrow$  Memory usage  
1080p matte material scene = 500 MB of BSDF

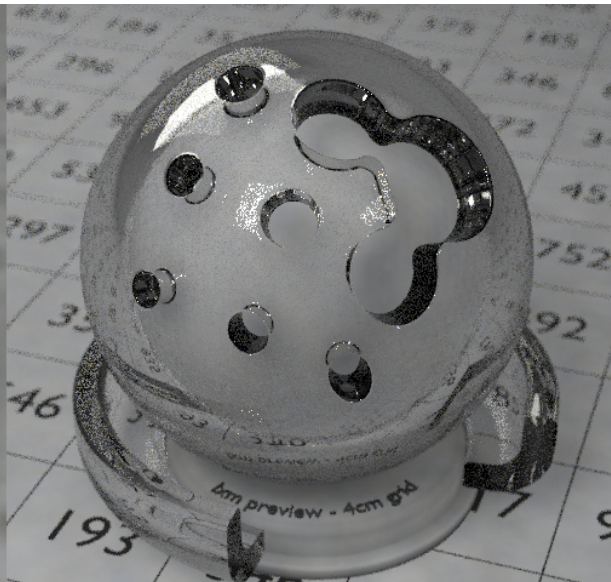
## Photon BSDF or HitPoint BSDF ?







LuxRender 0.8RC3 | Saved: 5/18/11 6:04 PM |  
Statistics: 02:31:30 - 8T: 18 pass 18.00 MP 1.98 kP/s 100% PEff



LuxRender 0.8RC3 | Saved: 5/18/11 6:05 PM |  
Statistics: 02:25:11 - 8T: 8 pass 8.00 MP 918.36 P/s 100% PEff

# Atomic HashGrid - Init

Hash Cells



Jump List



# Atomic HashGrid - Insert

HitPoint 0  
Hash = 4

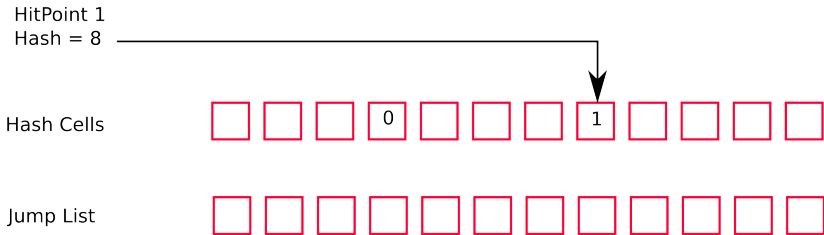
Hash Cells



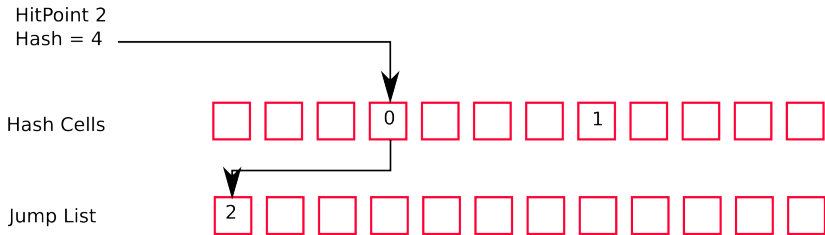
Jump List



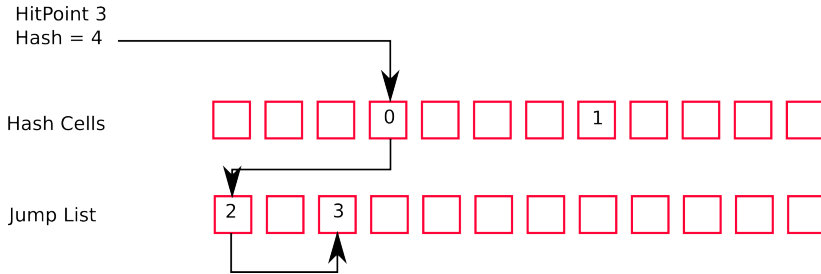
# Atomic HashGrid - Insert



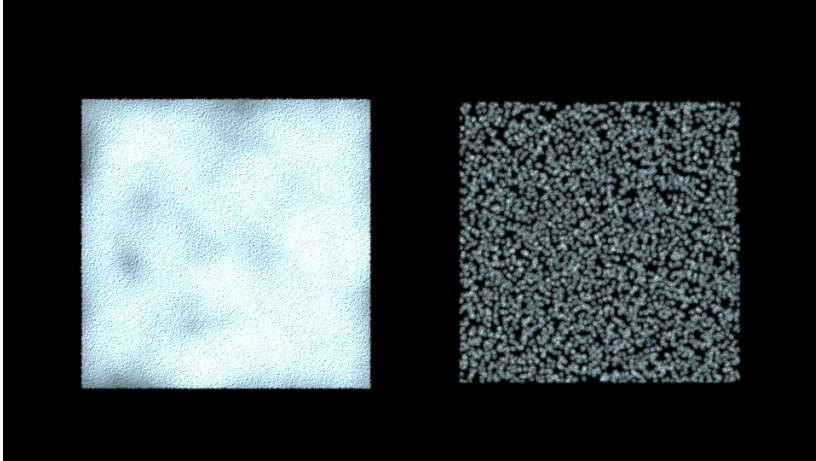
# Atomic HashGrid - Collision

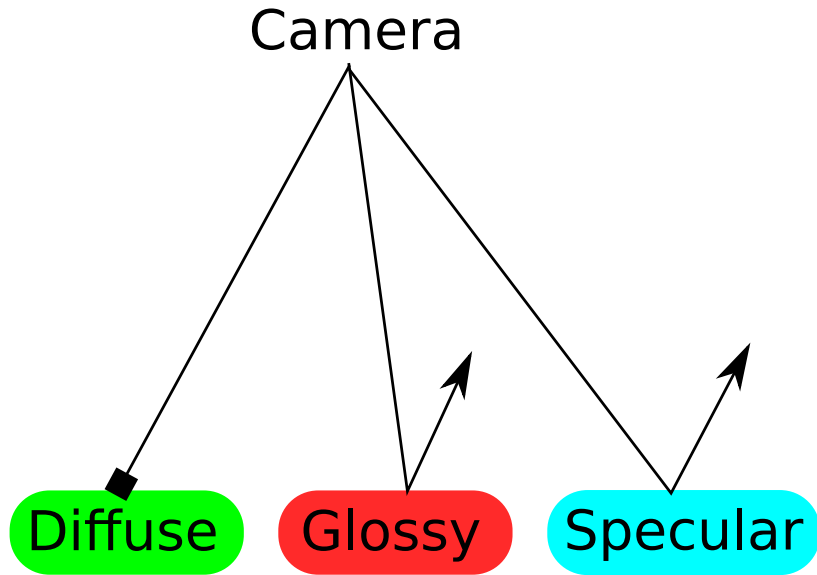


# Atomic HashGrid - Jump



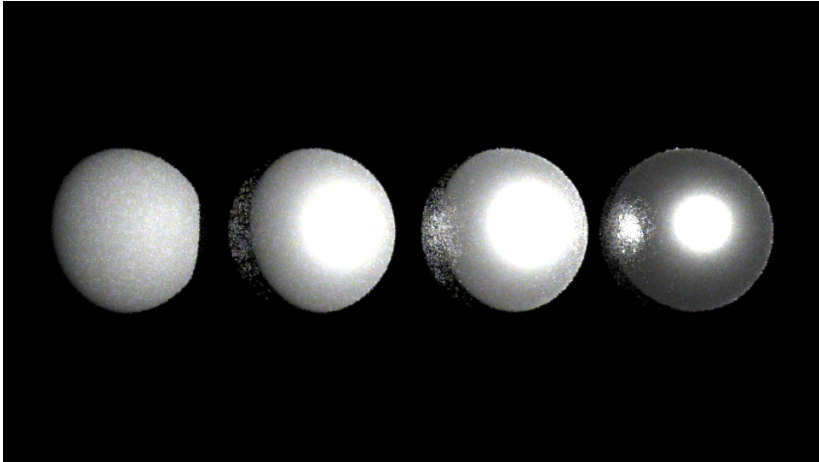
## Global Radius Worse Case



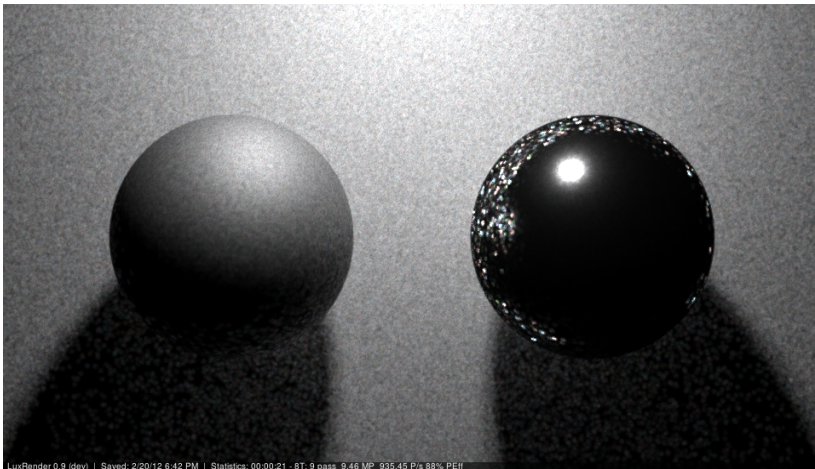




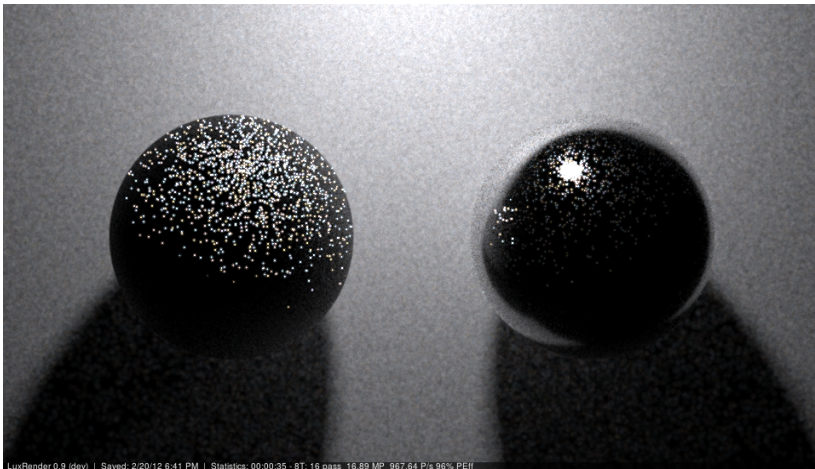
## Glossy Surface Handling



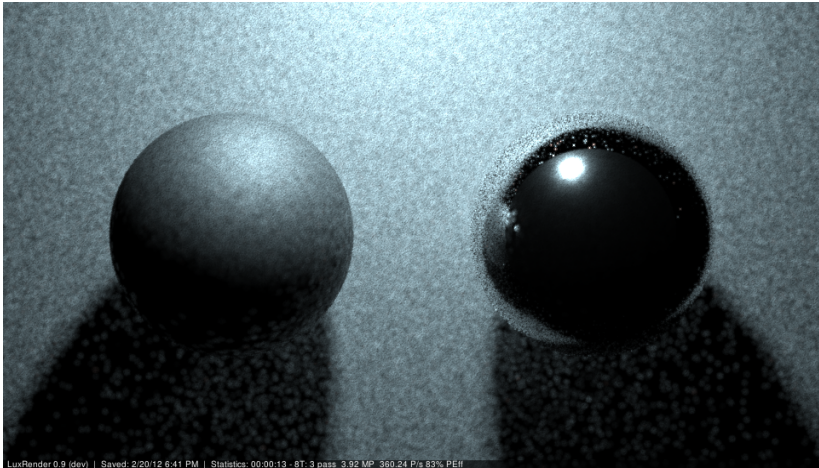
# Glossy Surface Handling: Always Stores on Glossy



## Glossy Surface Handling: Never Stores on Glossy



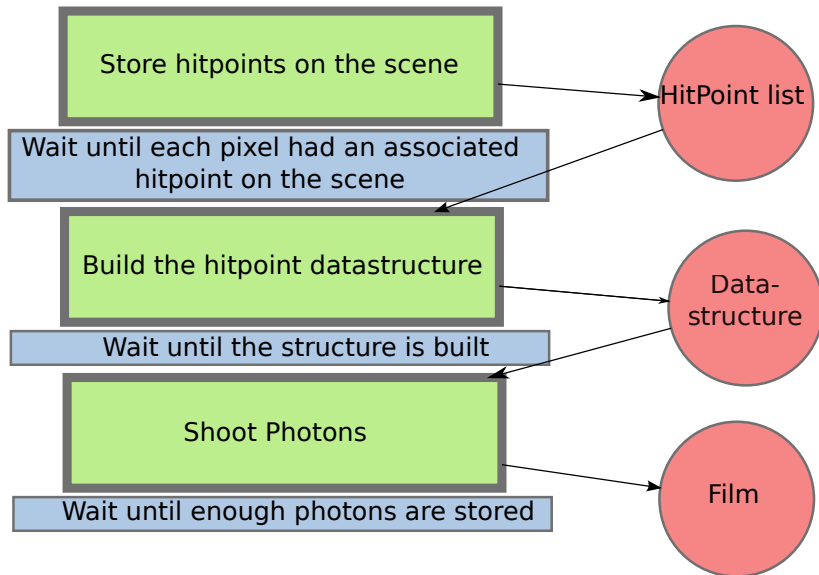
## Glossy Surface Handling: Depends on User Threshold



# Concurrency Issues

- ▶ Threading - Lot of synchronisations
- ▶ GPU - Hybrid Renderer
- ▶ Network - SPPM is stateful

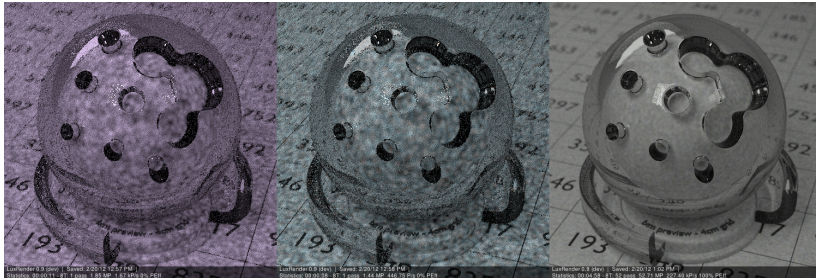
# Concurrency - Threading



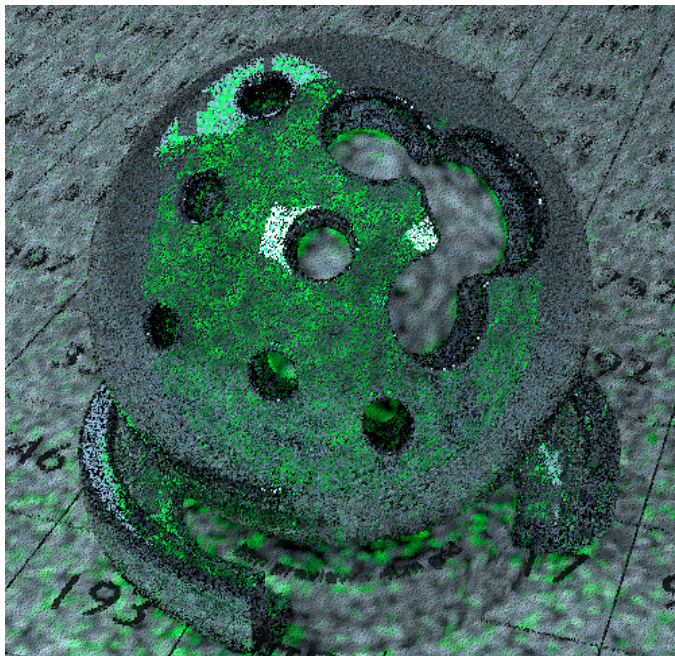
SPPM process depends on pass number

- ▶ Nodes are independents and may do the same pass many times
- ▶ Nodes are governed by a master

# Wavelength Sampling Issues







# Conclusion

SPPM still have open research issues

- ▶ Handling of glossy materials (MIS?)
- ▶ Stateless rendering
- ▶ Initial radius heuristic

# Acknowledgments

- ▶ Toshiya Hachisuka, Wojciech Jarosz
- ▶ Victor Ostromoukhov, Jean-Claude Iehl
- ▶ LuxRender Community, David \*Dade\* Bucciarelli

# Questions ?

- ▶ <http://luxrender.net/>
- ▶ [guillaume.bouchard@liris.cnrs.fr](mailto:guillaume.bouchard@liris.cnrs.fr)