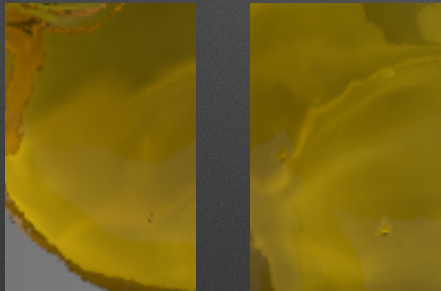
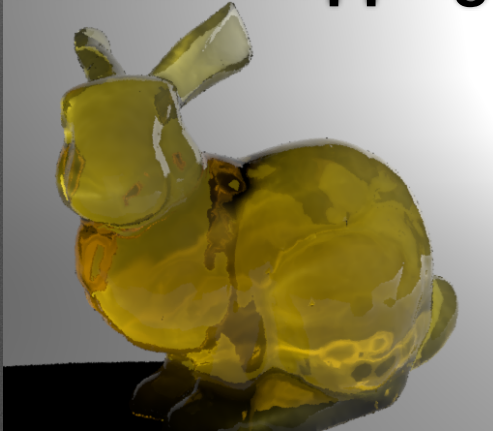
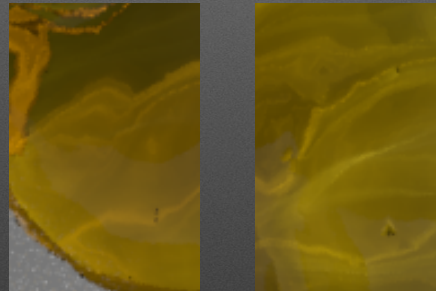
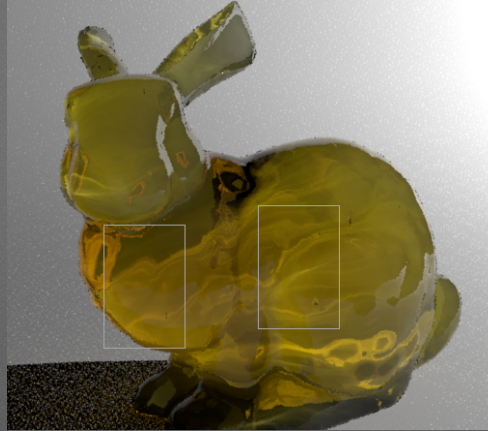


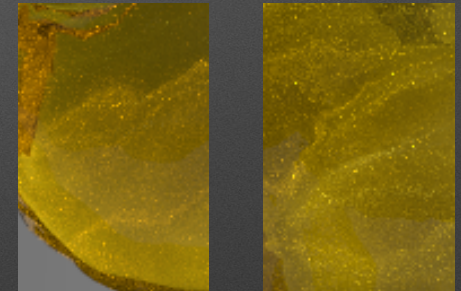
Photon mapping



Our method



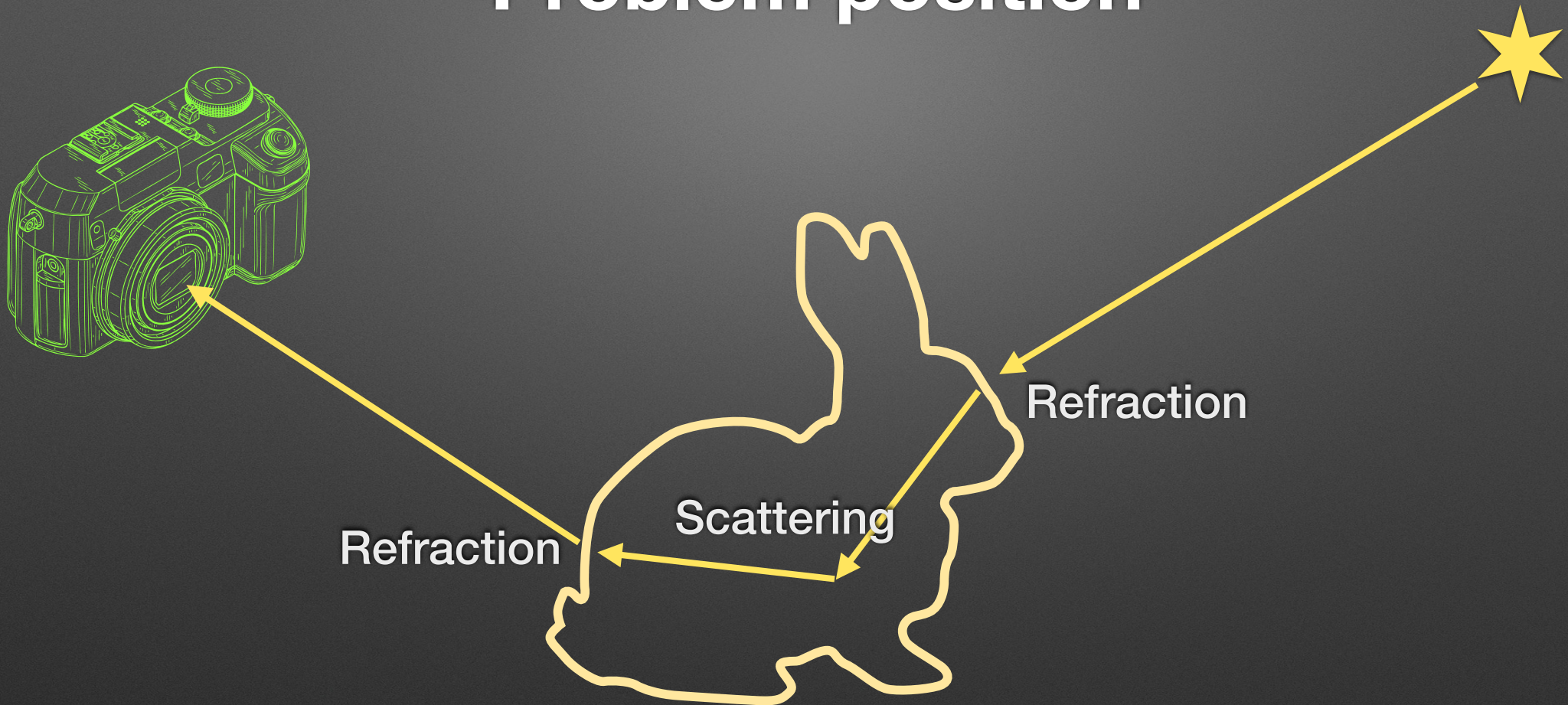
Point samples



Single scattering in participating media with refractive boundaries

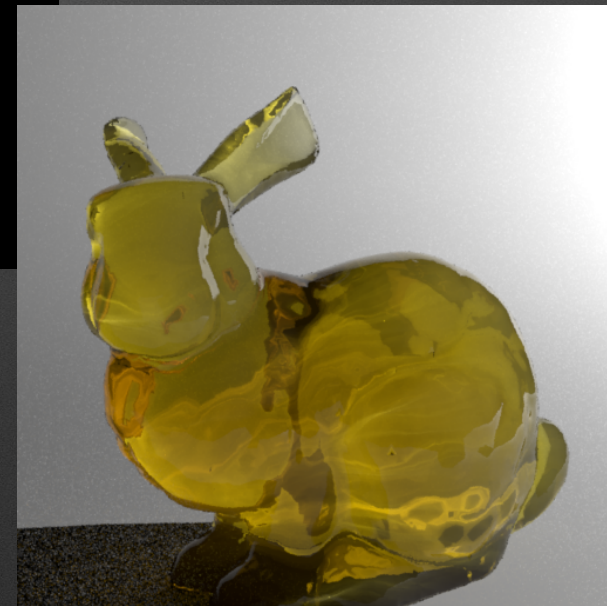
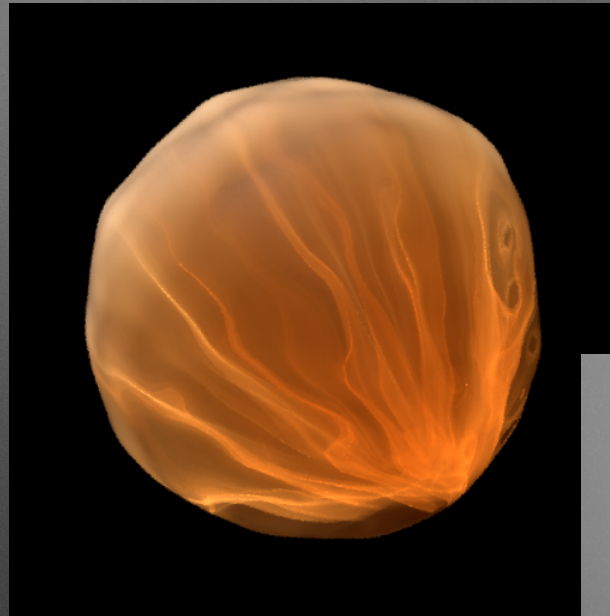
Nicolas Holzschuch
INRIA ; Univ. Grenoble-Alpes, LJK ; CNRS, LJK

Problem position



Problem position

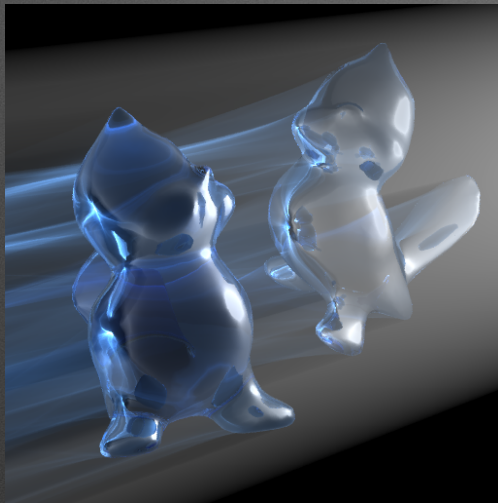
- Finding paths is hard
- Creates beautiful effects



Previous work

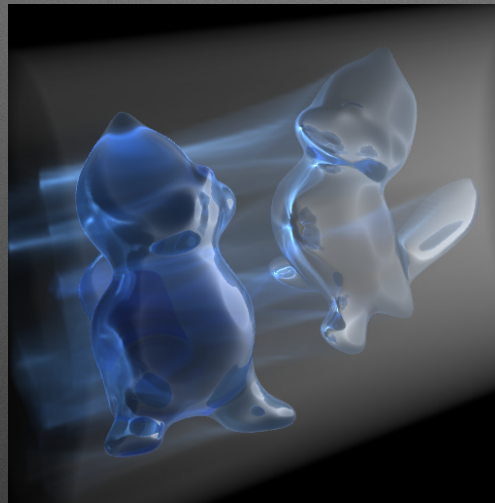
- Discretize inside the object:
 - [Sun et al. 2008] [Sun et al. 2010][Ihrke et al. 2010]
- Find paths connecting to camera [Walter et al. 2009]
- Beam radiance estimate, Photon beams
 - [Jarosz et al. 2008][Jarosz et al. 2011]
[Křivánek et al. 2014]

Object interior discretization

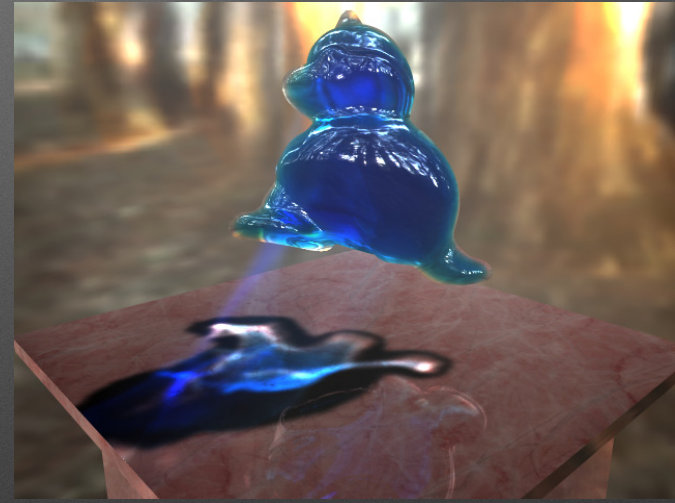


**Line-Space
Gathering
(voxelisation)**

[Sun et al. 2010]



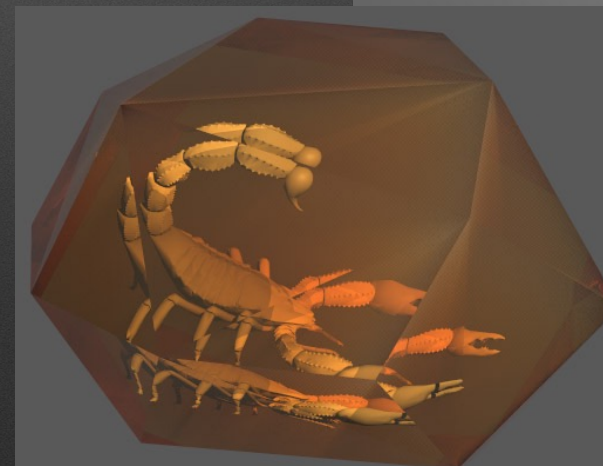
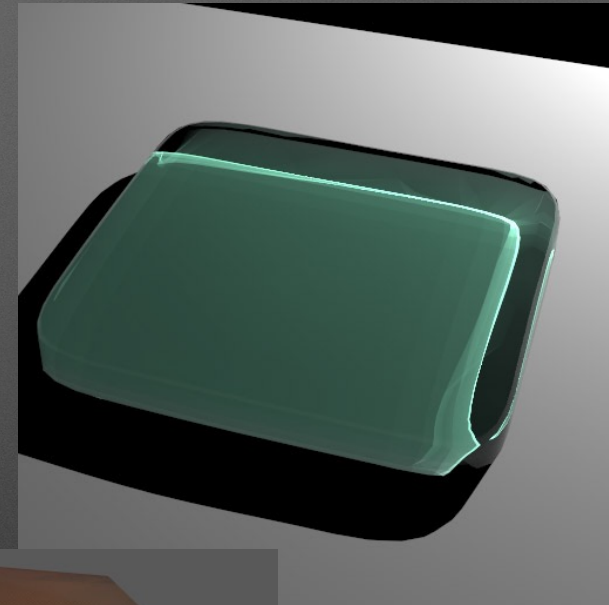
**Eikonal
rendering
[Ihrke et al. 2010]**



**Voxelisation
+relighting
[Sun et al. 2008]**

Path connecting to the camera

- Trace camera ray inside object
- Sample points along camera ray
- For each: find path connecting back to light source
- Newton-Raphson iteration with pseudo-inverse
- [Walter et al. 2009]

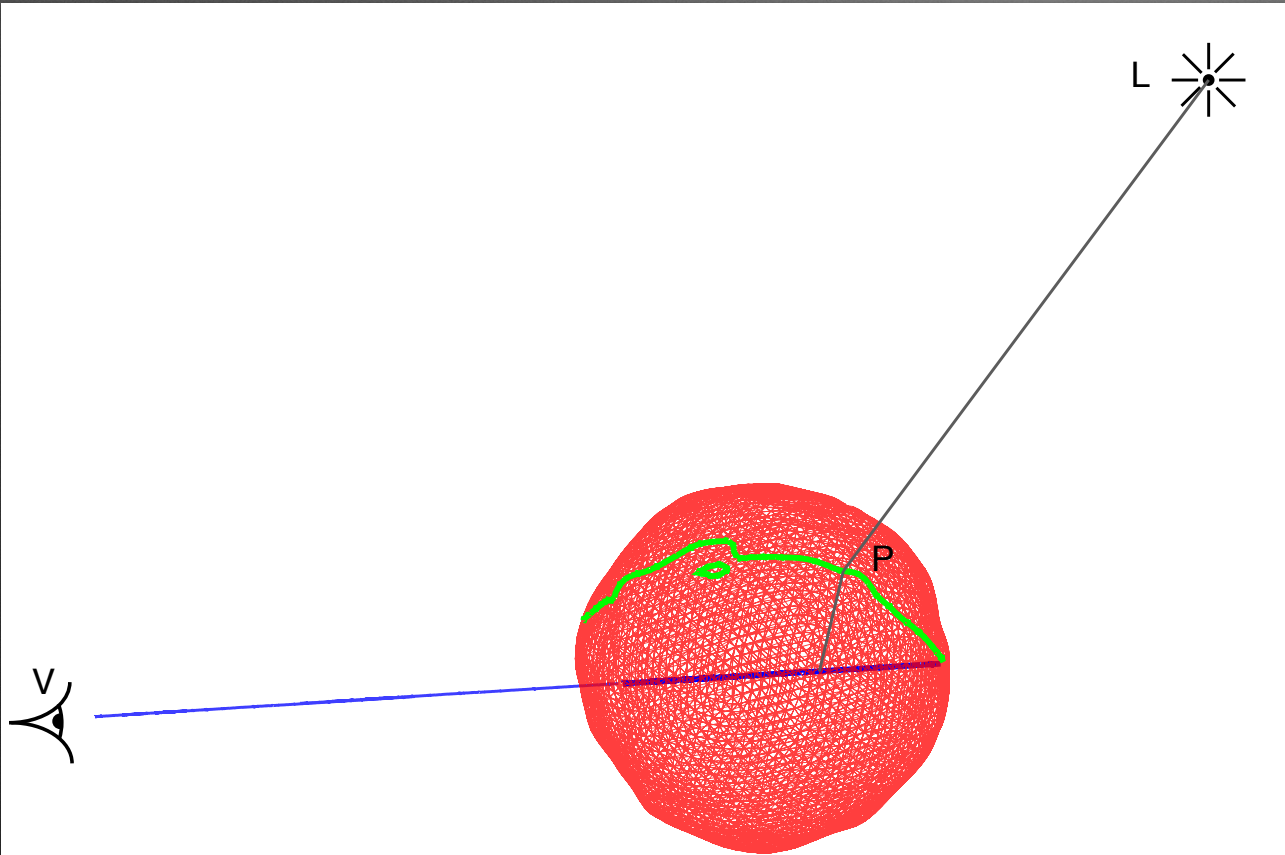


Photon beams

- Multipurpose solution
- Extension of Photon Mapping
- Trace beams into media
- Also multiple scatter
 - [Jarosz et al. 2008]
 - [Jarosz et al. 2011]²
 - [Křivánek et al. 2014]

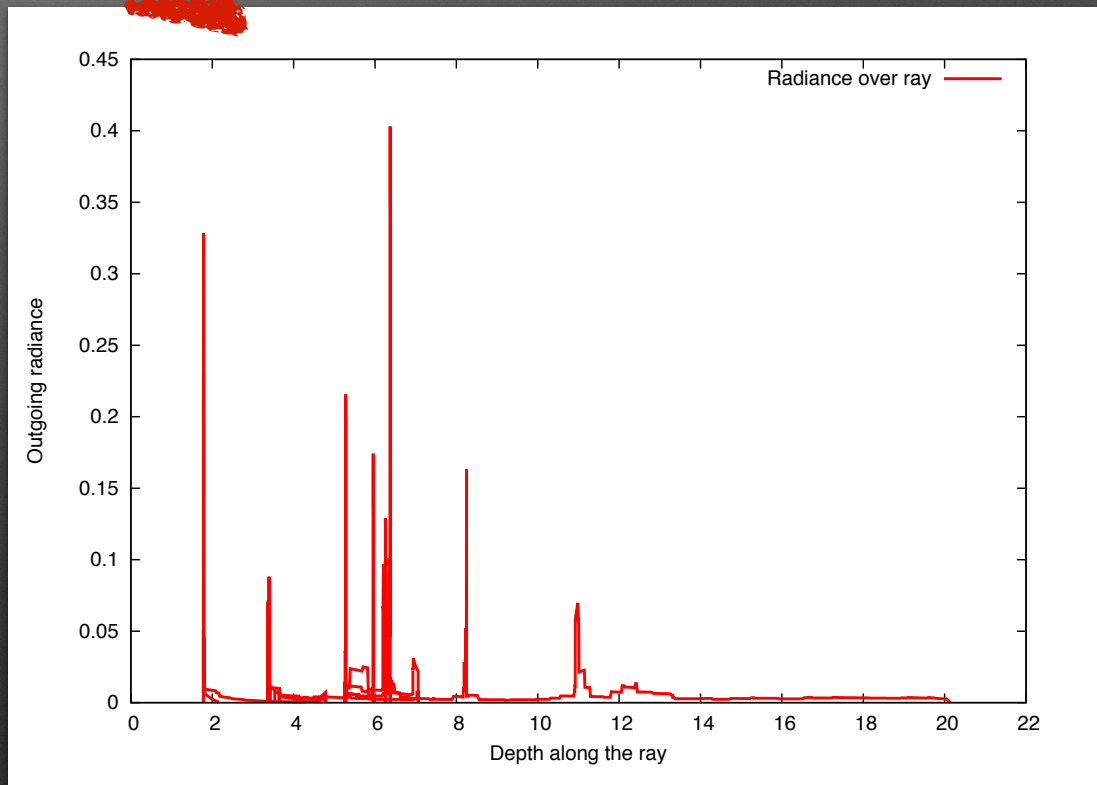
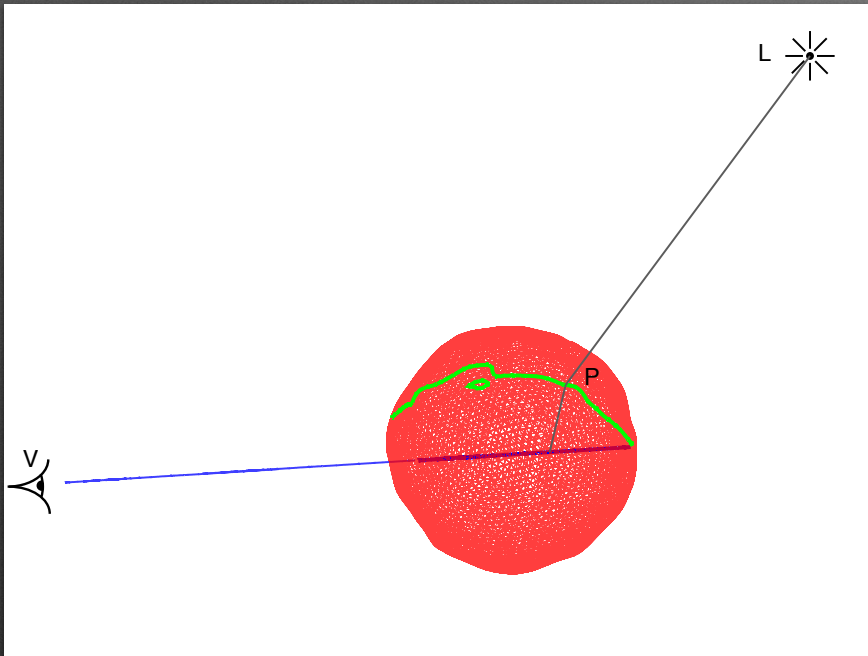


Let's take a closer look



- Ray from camera
- Many sample points
- Find path to camera
- Compute radiance
- Visualize

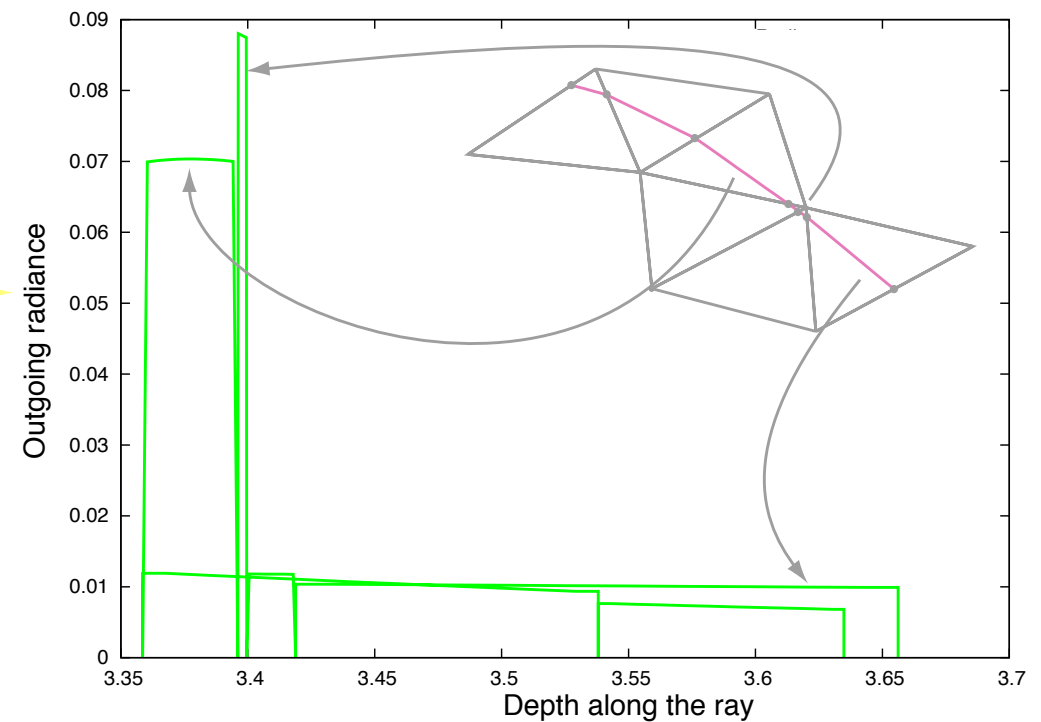
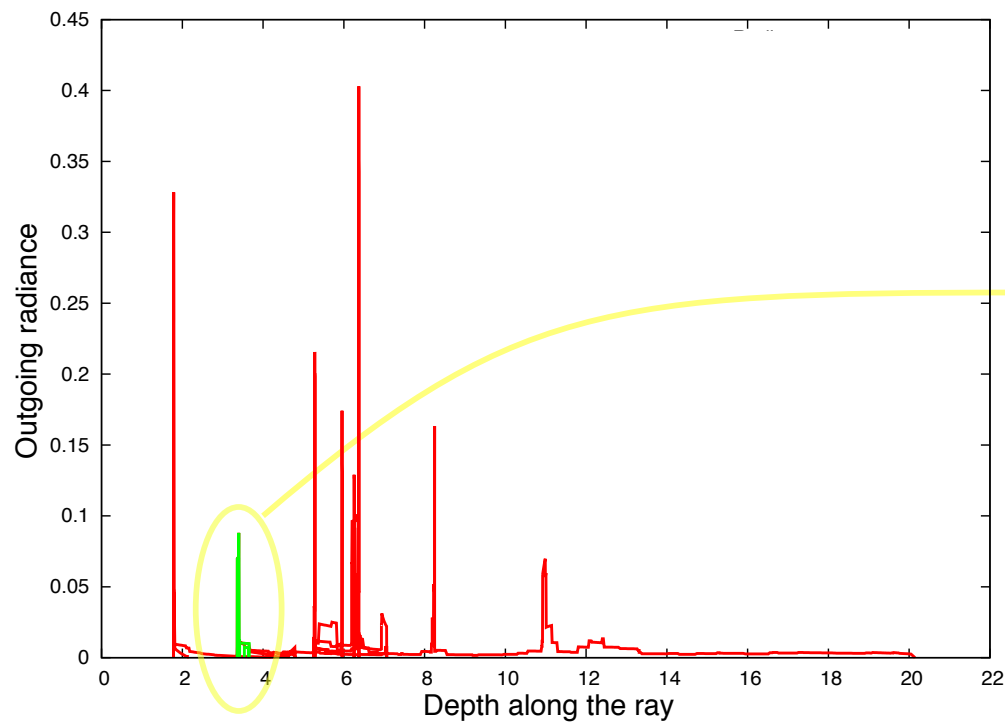
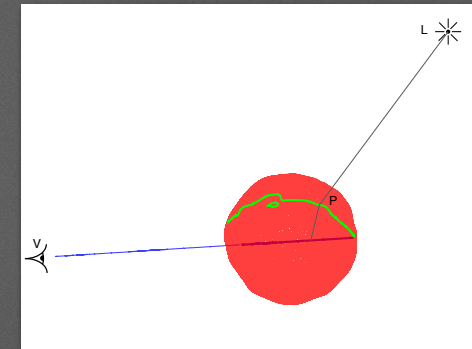
Let's take a closer look



Radiance inside object

- Irregular function
- Very sharp peaks
- Must sample them all
- How?

Let's take a closer look



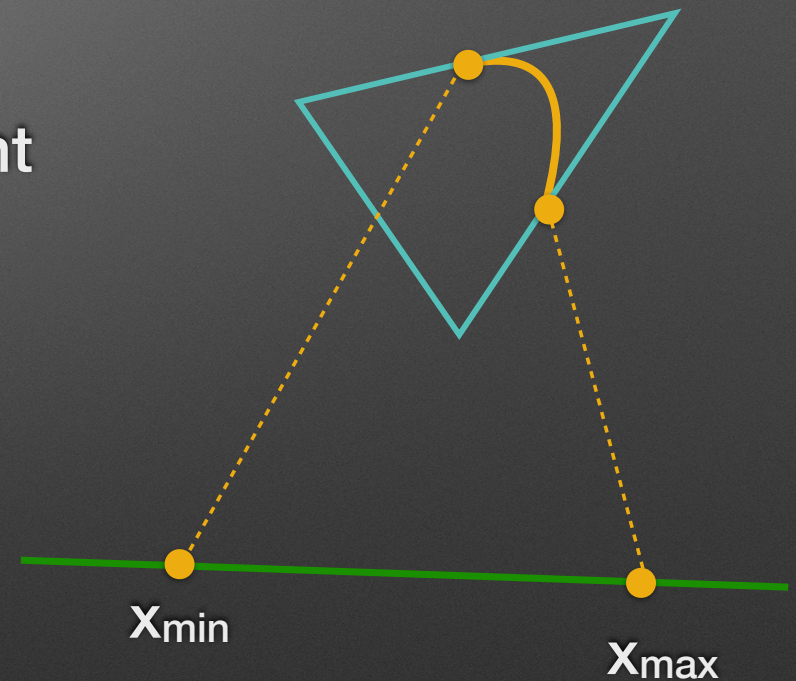
For each surface triangle: smooth function

Smooth & irregular function

- Globally irregular
- Locally (for each triangle) smooth
- Discontinuities = triangle boundaries
 - curve-triangle intersection

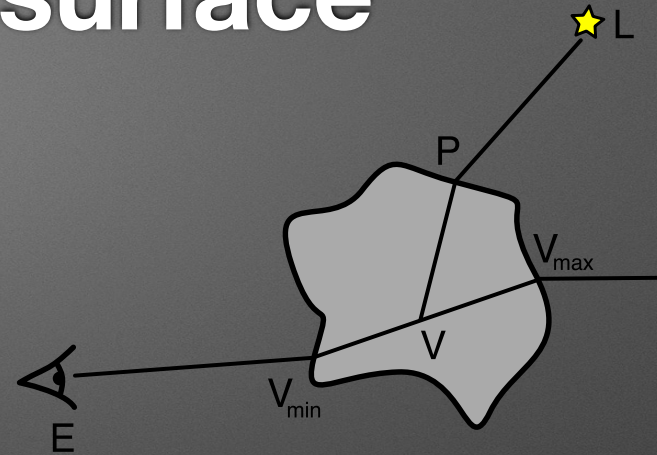
Algorithm

- for each image sample
 - trace ray, refract at surface, find exit point
 - for each triangle:
 - compute intersection with curve
 - interval $[x_{\min}, x_{\max}]$ on the ray
 - sample regularly inside the interval



Curve on the surface

- Half-vector, h
- $f = h + n$
- parameters: x along the ray, (α, β) on triangle
- zeros of f = curve on the triangle plane
- Finding $[x_{\min}, x_{\max}]$ = zeros of f on the triangle edges



Finding zeros of f

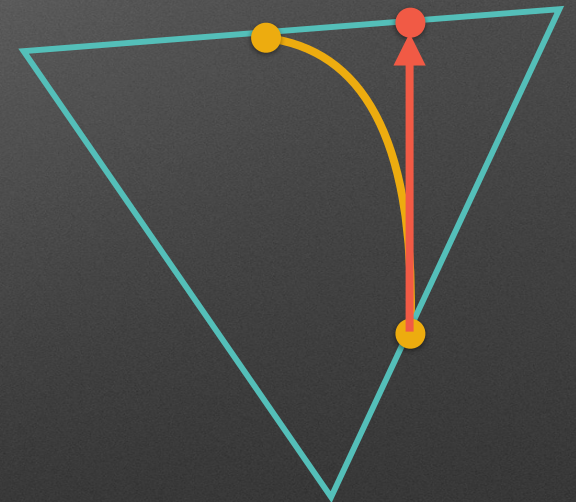
- Linearization: $f(p+\delta p) \simeq f(p) + J \delta p$
- Newton-Raphson method:
 - starting point p_0
 - $p_{n+1} = p_n - J^{-1} f(p_n)$
 - Problem: 1 fixed coordinate, 2 varying

Pseudo-Inverse Newton

- With 1 fixed coordinate:
 - $f: \mathbb{R}^2 \rightarrow \mathbb{R}^3$
 - J is not invertible (not even square)
- Pseudo-inverse : $J^+ = (J^T J)^{-1} J$
- $p_{n+1} = p_n - J^+ f(p_n)$
- Amazingly stable/efficient

Finding a good starting point

- Quadratic convergence *if* close to solution
- *If* we know a point on the curve:
 - point + curve gradient = tangent to curve
 - intersect with triangle edge: starting point
- Point on curve:
 - intersection ray/triangle plane. Or previous point

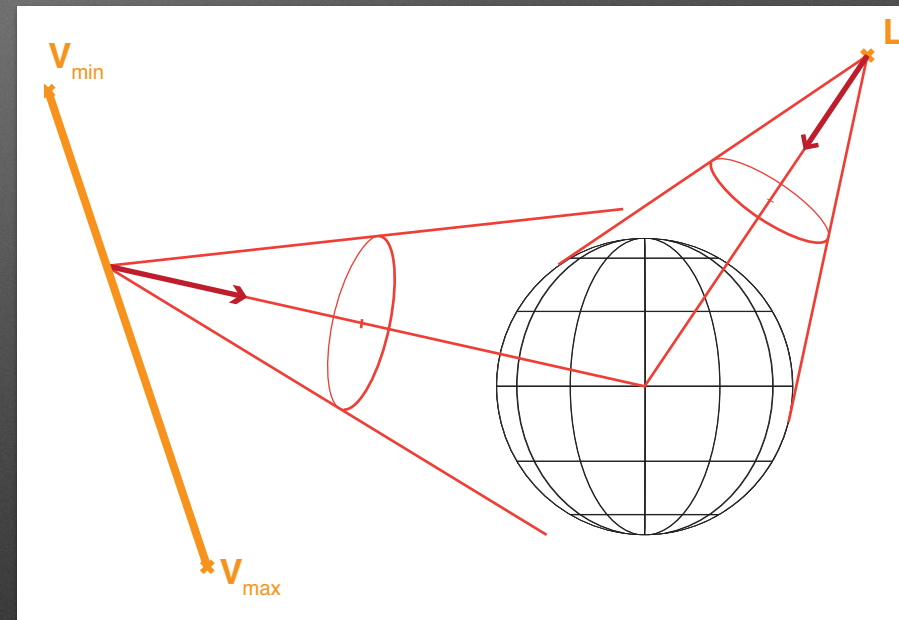


Curve gradient

- Known: Jacobian J of f , at point where $f = 0$
- Need: curve gradient
- $J = (l_1, l_2, l_3)$ (line vectors) ; rank = 2
- $g = l_i \wedge l_j$

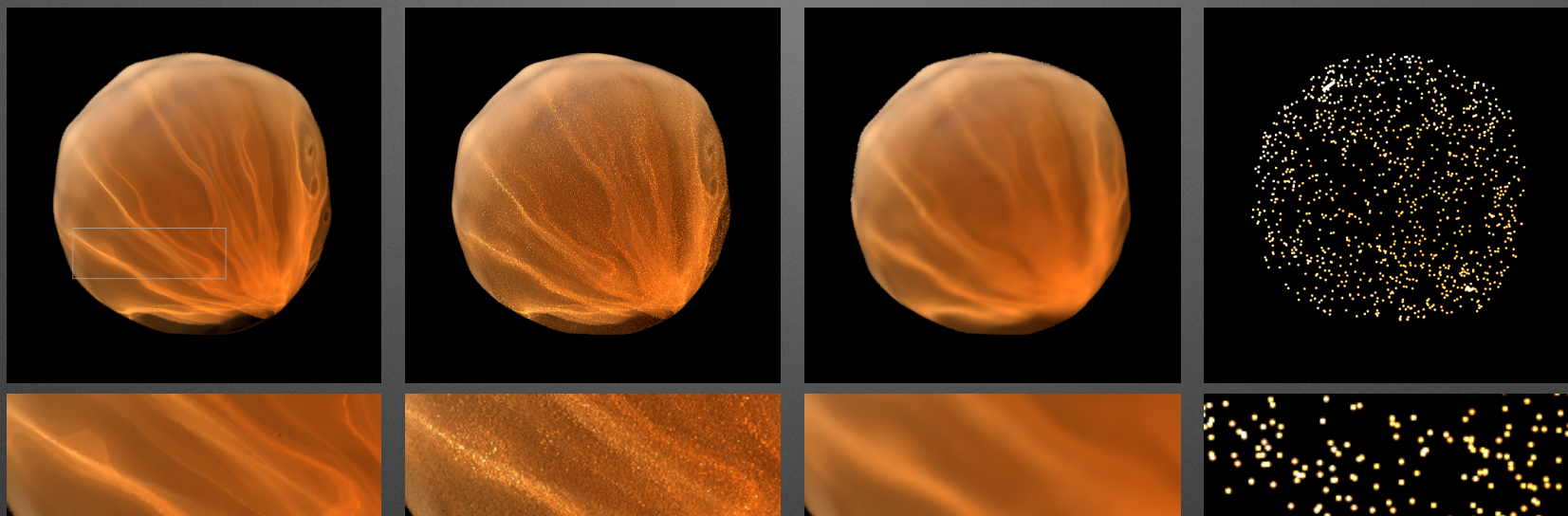
Culling triangles

- Use existing BVH to cull triangles
- Geometric condition for refraction:
 - Triangle must be inside spindle shape
 - Bounding sphere for BVH cell
 - Bounding cones for both rays
 - Condition for bounding cones
- Normal-based condition for individual triangles



Results and comparison

Equal-time comparison



Our method
(169 s)

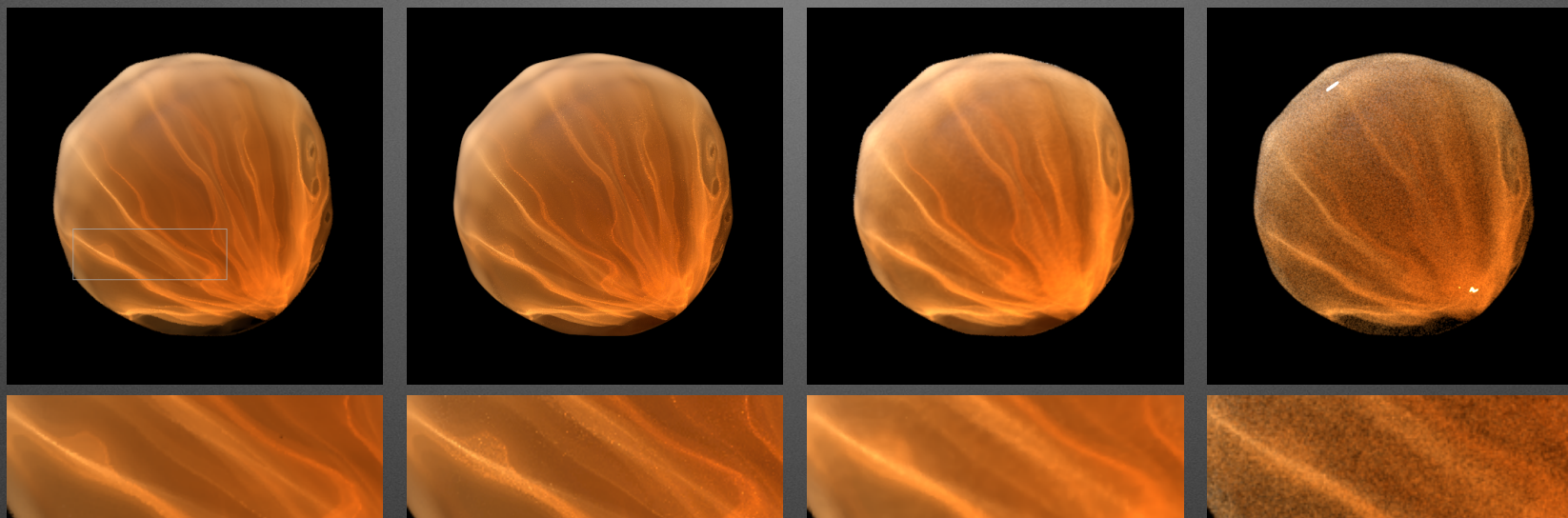
Walter et al.
(27 spp, 170 s)

Photon map
(650 K, 167 s)

BDPT
(520 spp, 171 s)

9680 triangles

Equal quality comparison



Our method
169 s

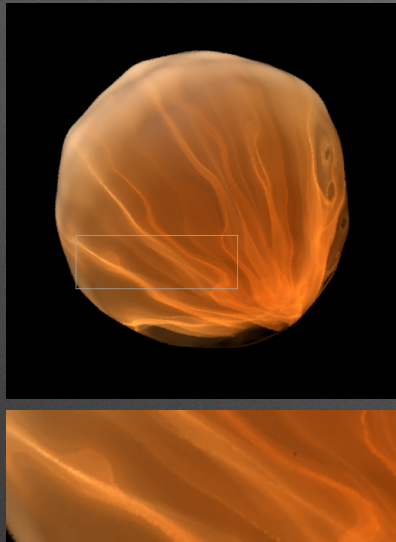
Walter et al.
512 spp
3188 s

Photon map
15M
2940 s

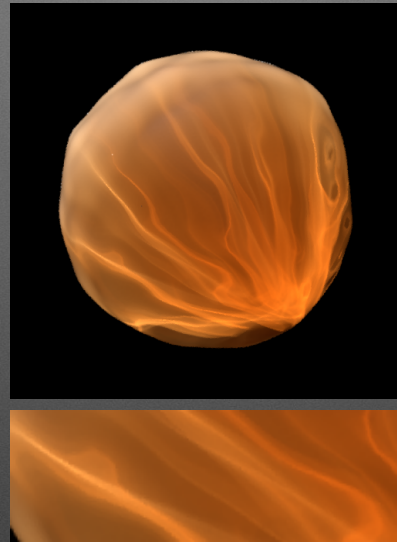
BDPT
500 K spp
331776 s

(15M photons is the maximum on this computer)

Validation

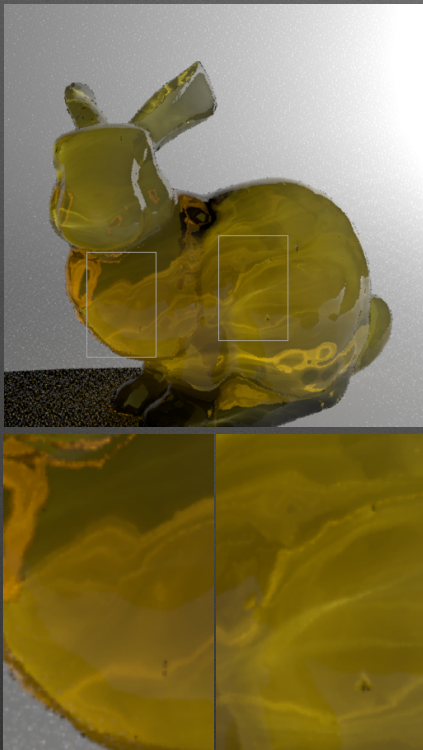


Our method
(169 s)

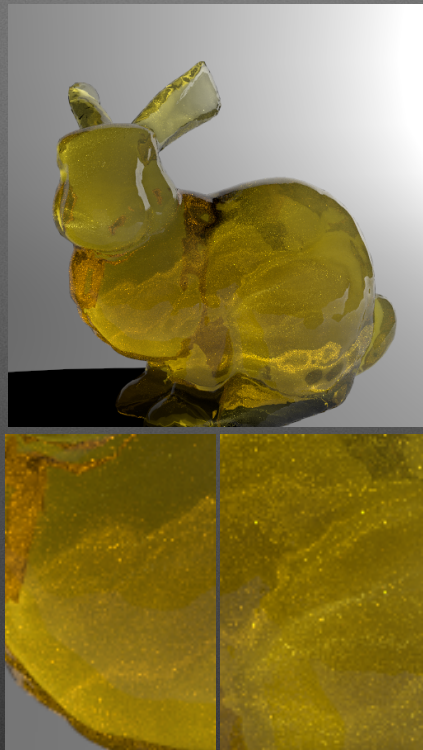


150M photons
(3.53 h on 8 core cluster)

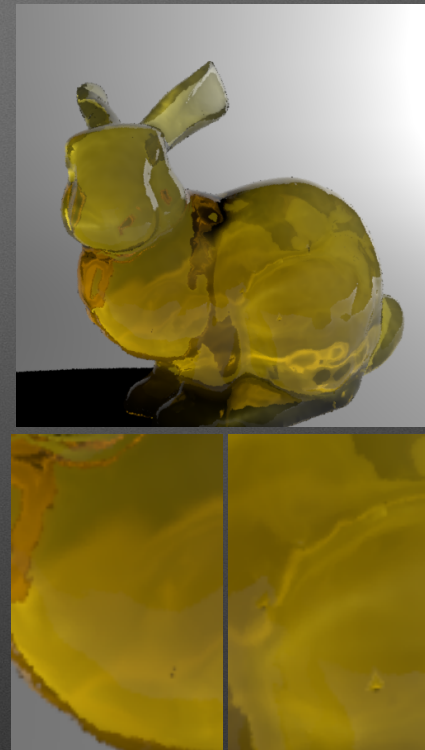
Equal time comparison (2)



Our method
5.8 mn
16301 triangles

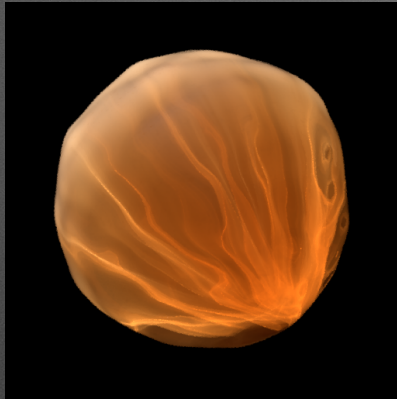


Walter et al.
64 spp
6.2 mn

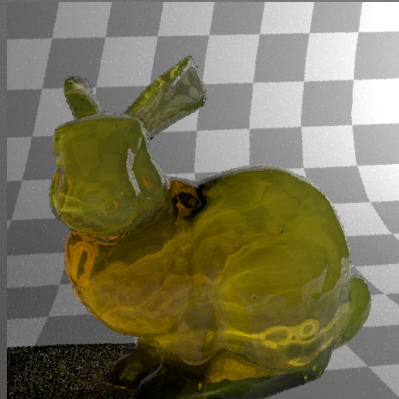


Photon map
1.8M photons
5.6 mn

Multiple test scenes



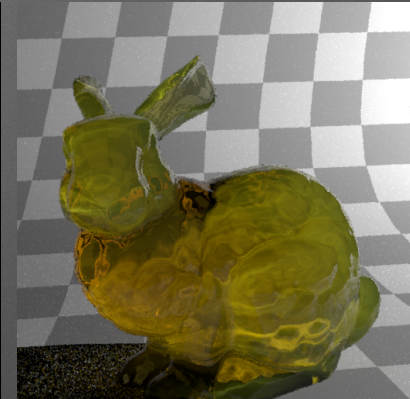
9680 t.



16301 t.



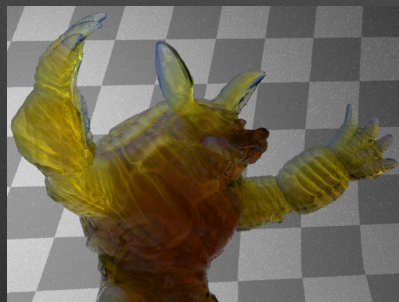
50 K t.



69451 t.



310K t.



346K t.

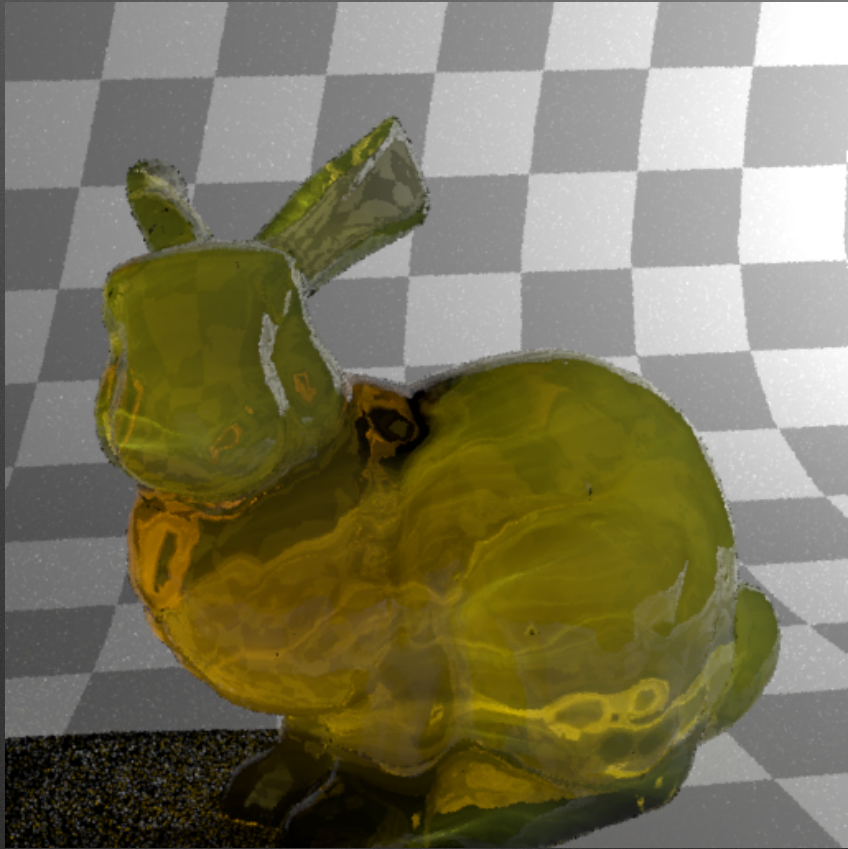


500 K t.

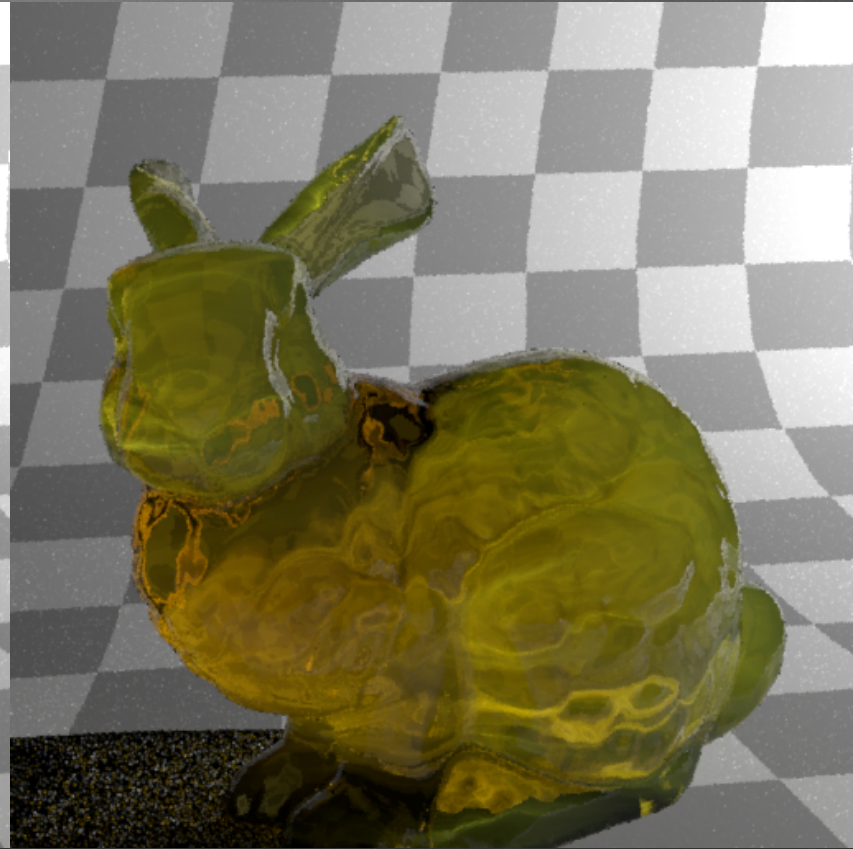


1M t.

Visual comparison

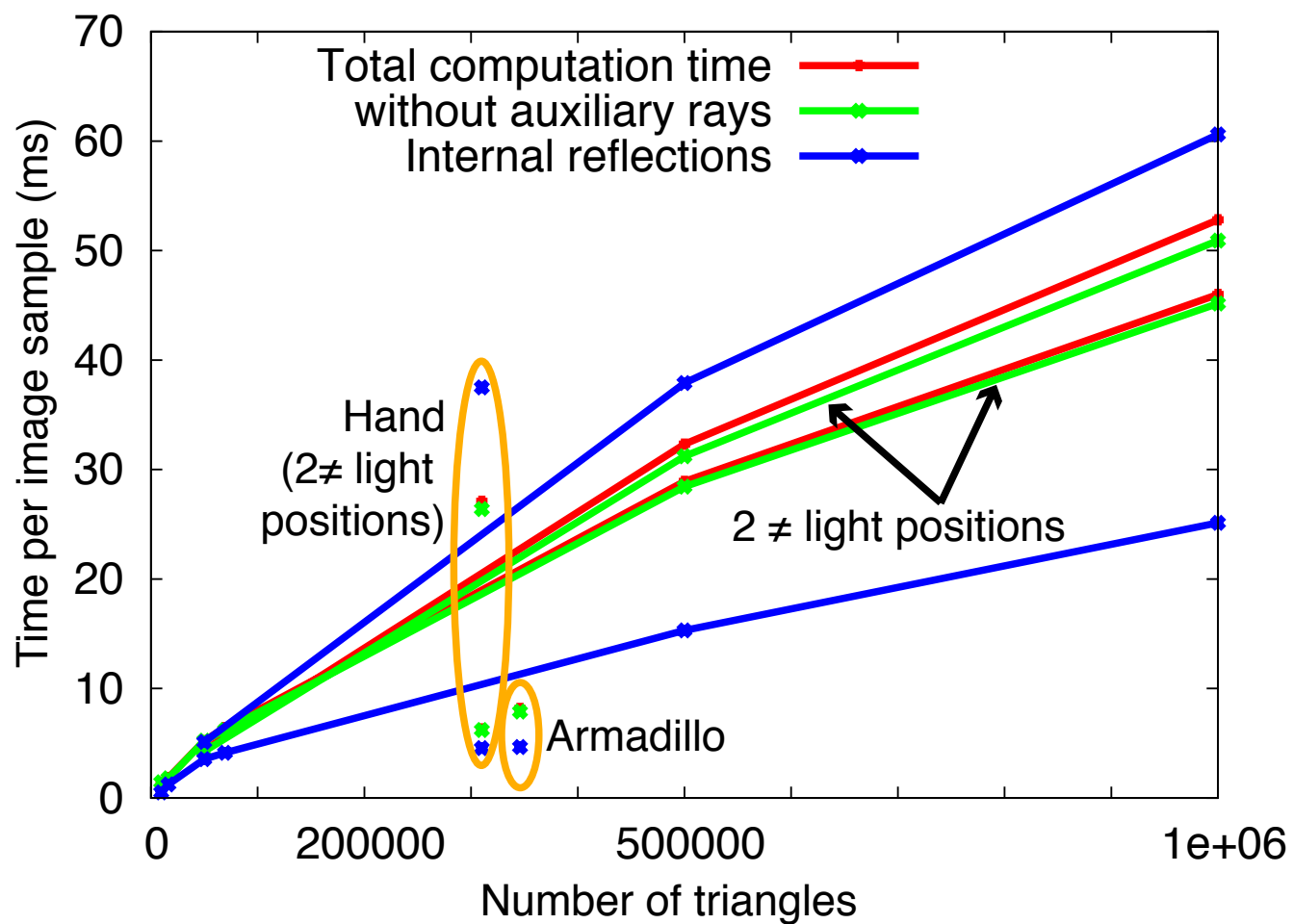


16301 t.



69451 t.

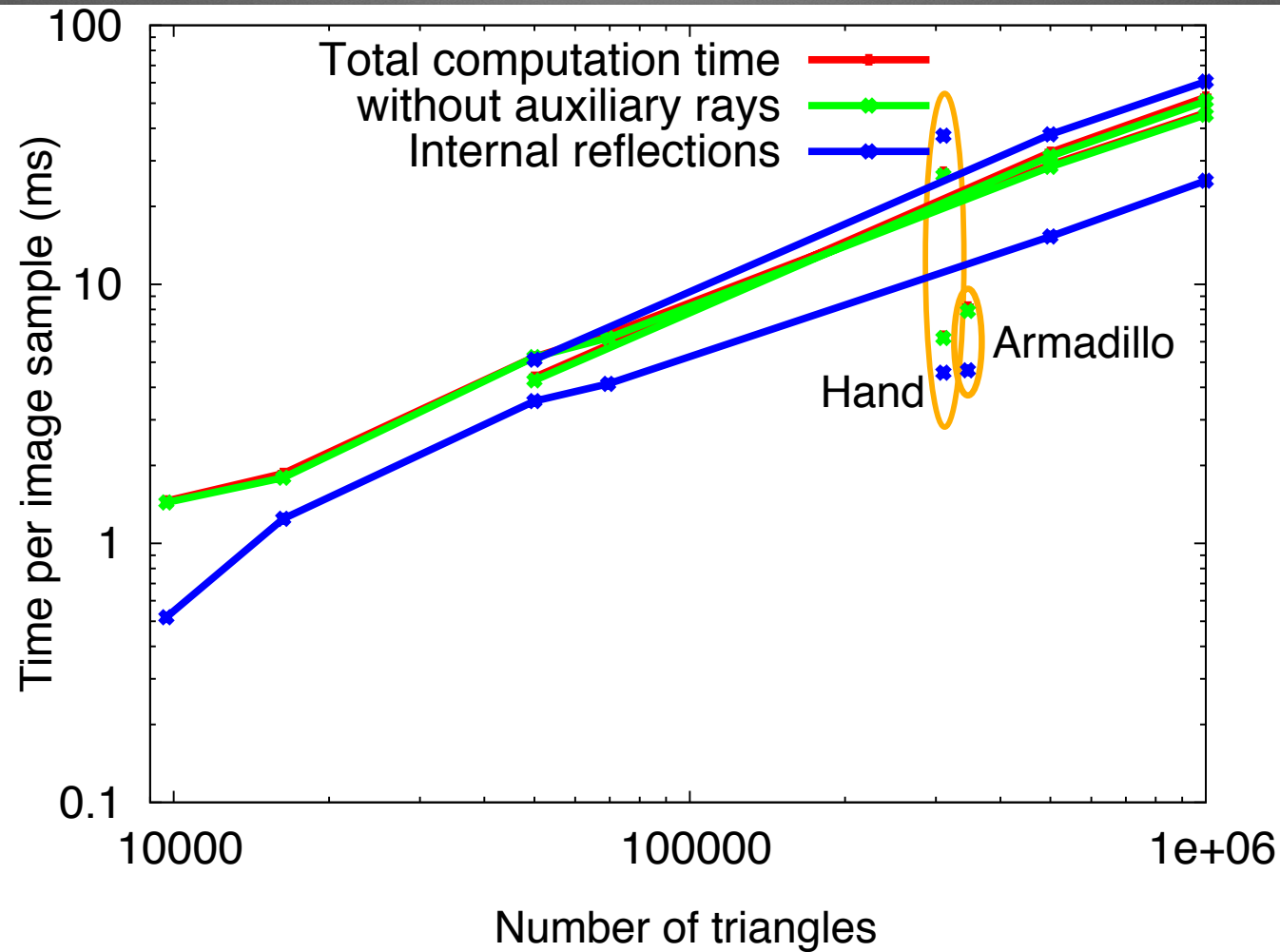
Timings



Timings — Analysis

- Most of computation time: finding zeros of f
- “compact” objects: time increases with num. triangles
 - little variations with light position
- “branching” objects: more variation with light source
- Internal reflections: roughly similar to direct lighting

Timings (log scale)



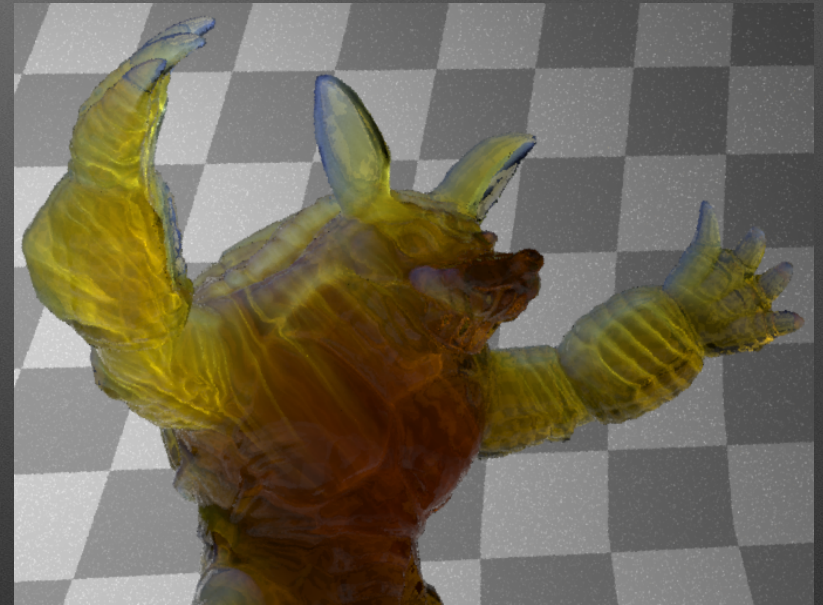
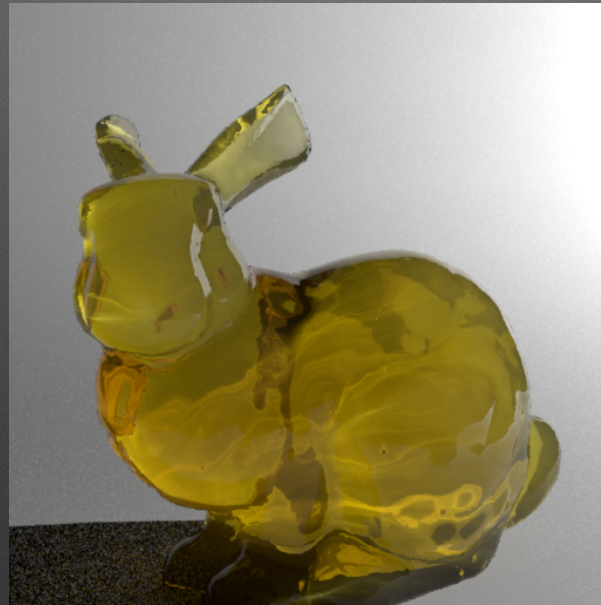
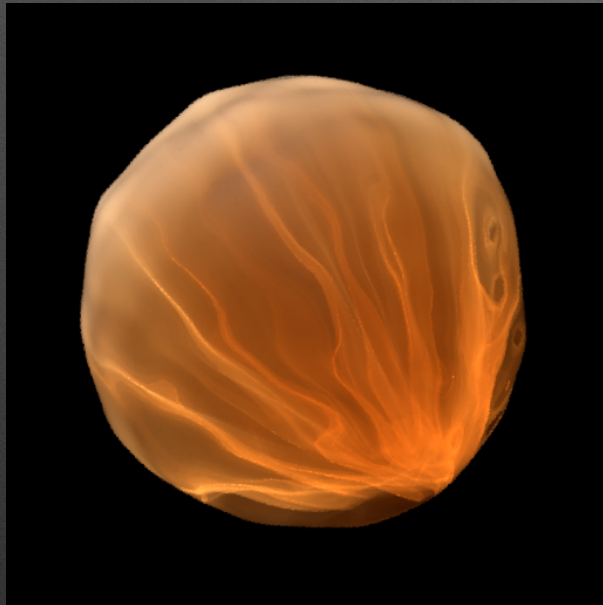
Timings — analysis

- Time per pixel sample \approx number of polygons $^{\wedge} 0.75$
- Constant = light source position

Conclusion and future work

- Scattering + 2 refractive boundary
- Fast and high quality
- Source code available : <https://hal.inria.fr/hal-01083246>
- Future work:
 - Cache results from neighbouring pixels
 - Multiple refractive layers

Thank you

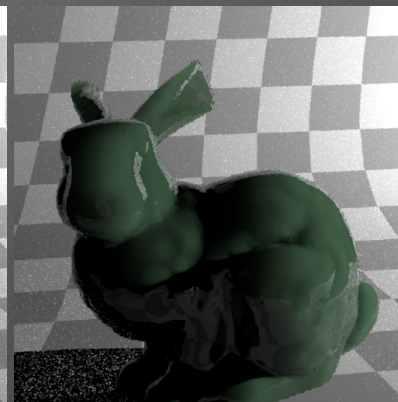
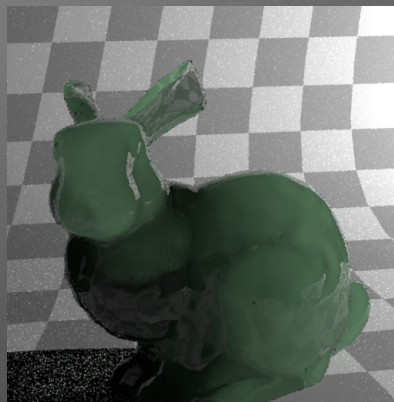
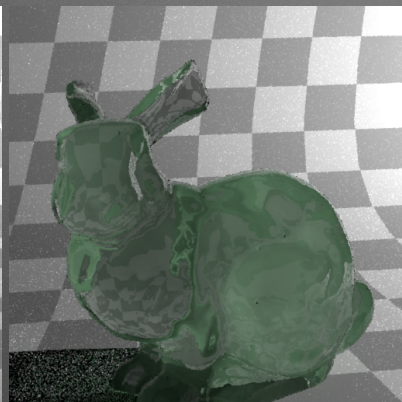
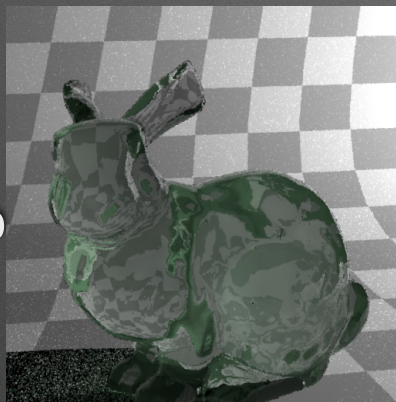


Extra slides

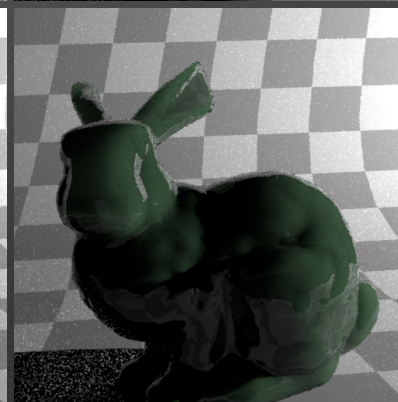
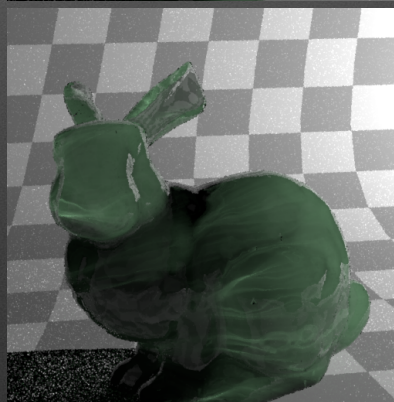
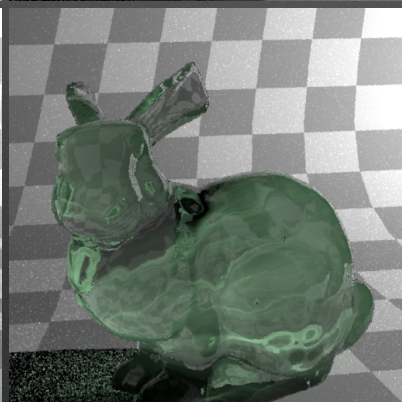
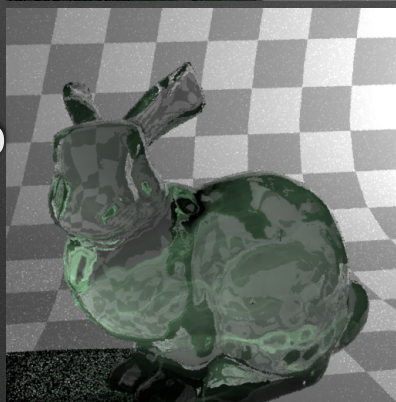
- Density variation
- Previous work
- BDPT
- Why so difficult?

Variation with density

Straight line



Our algo



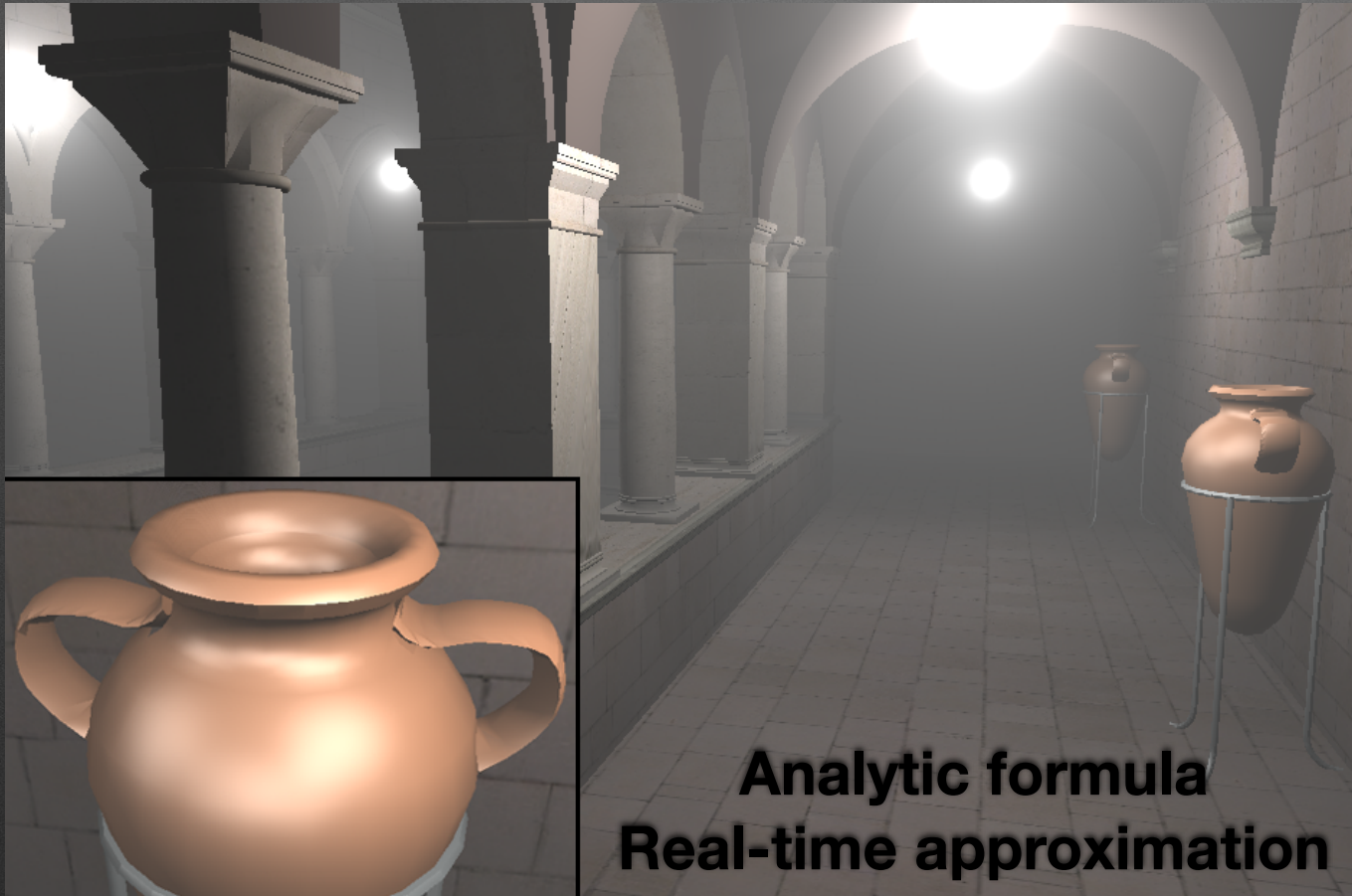
Scale=2

Scale=8

Scale=32

Scale=128

Single scattering (no boundaries)



[Sun et al. 2005]

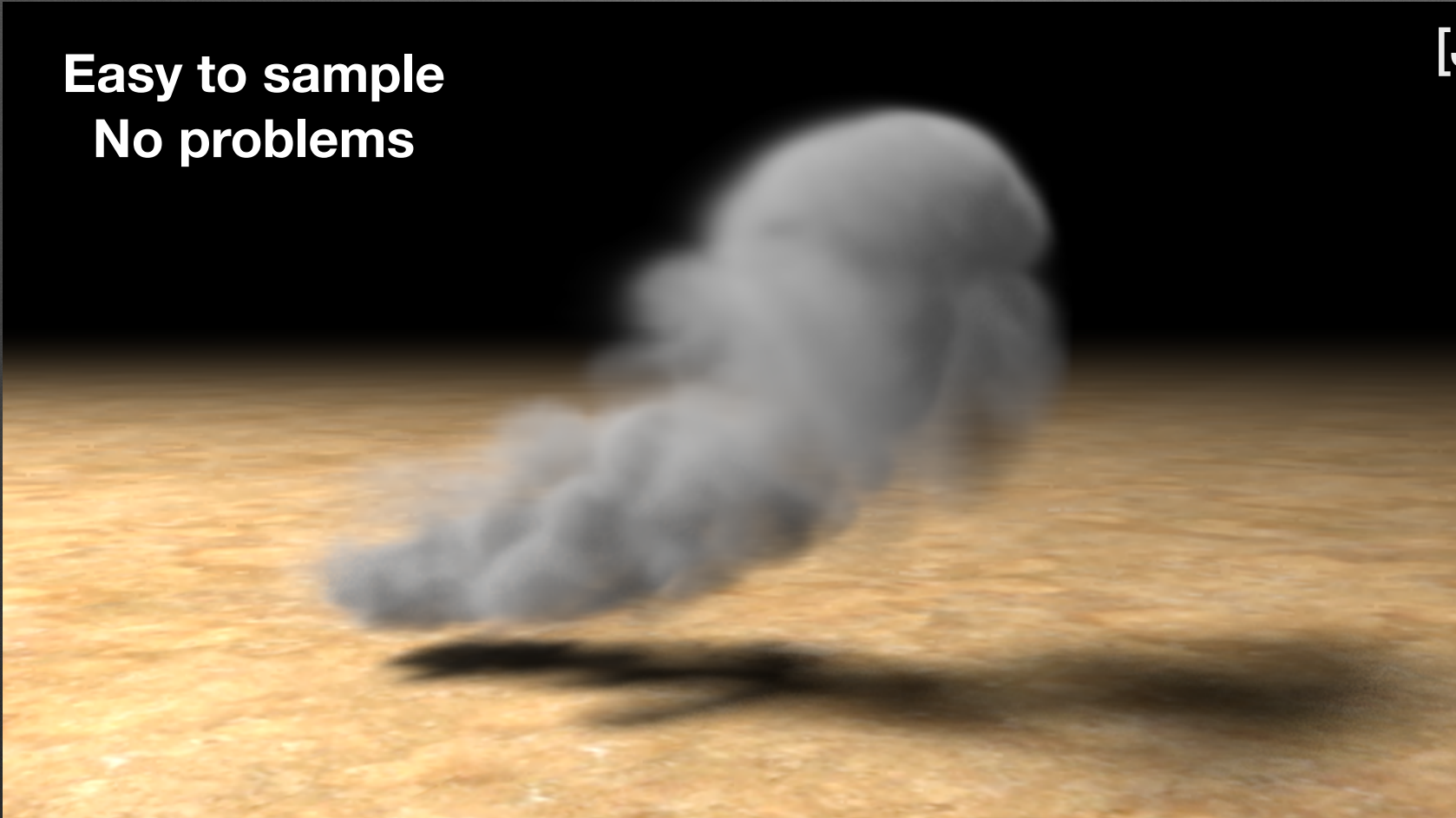
[Perogaro & Parker 2009]

Analytic formula
Real-time approximation

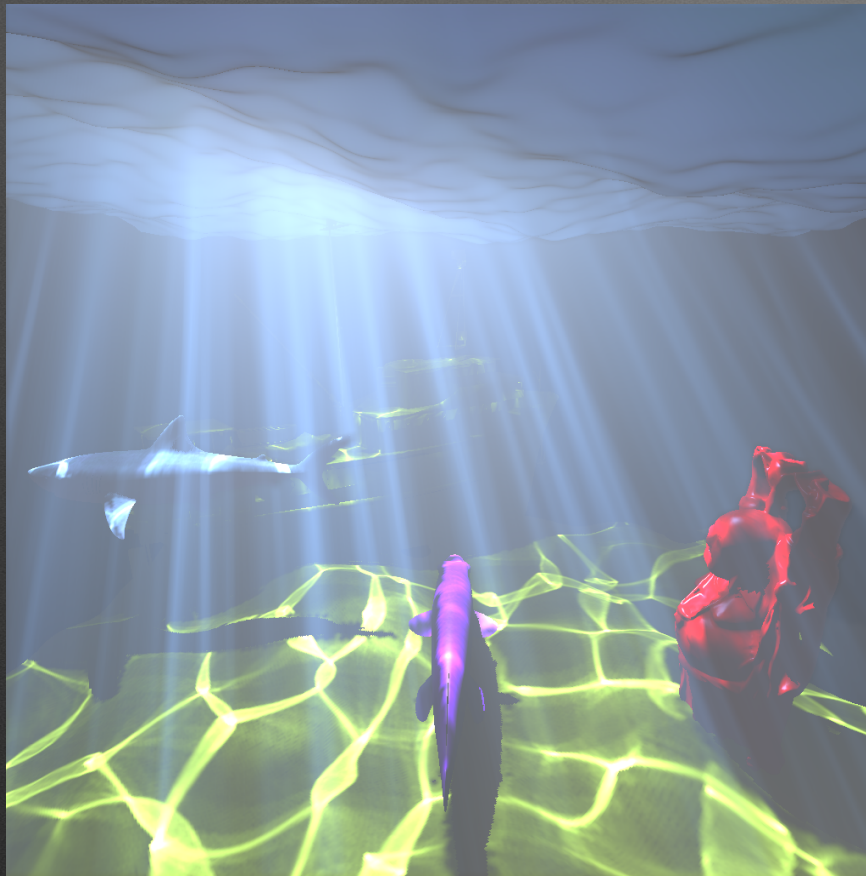
Single scattering (boundaries, no refraction)

Easy to sample
No problems

[Jarosz et al. 2008]



Single scattering (one refractive boundary)



Still relatively easy
Several real-time algorithms:

[Nishita & Nakamae 94]

[Iwasaki et al. 2001]

[Ernst et al. 2005]

[Hu et al. 2010]

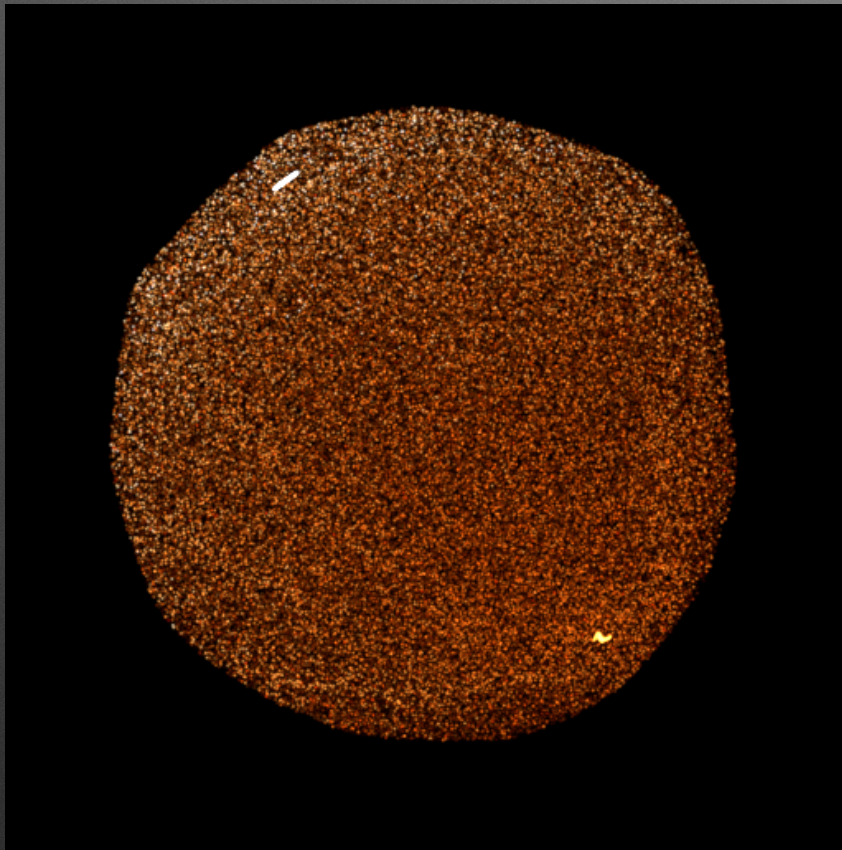
Single scattering (with 2 refractive boundaries)



Scattering *after* the refractions:
can be done in real-time

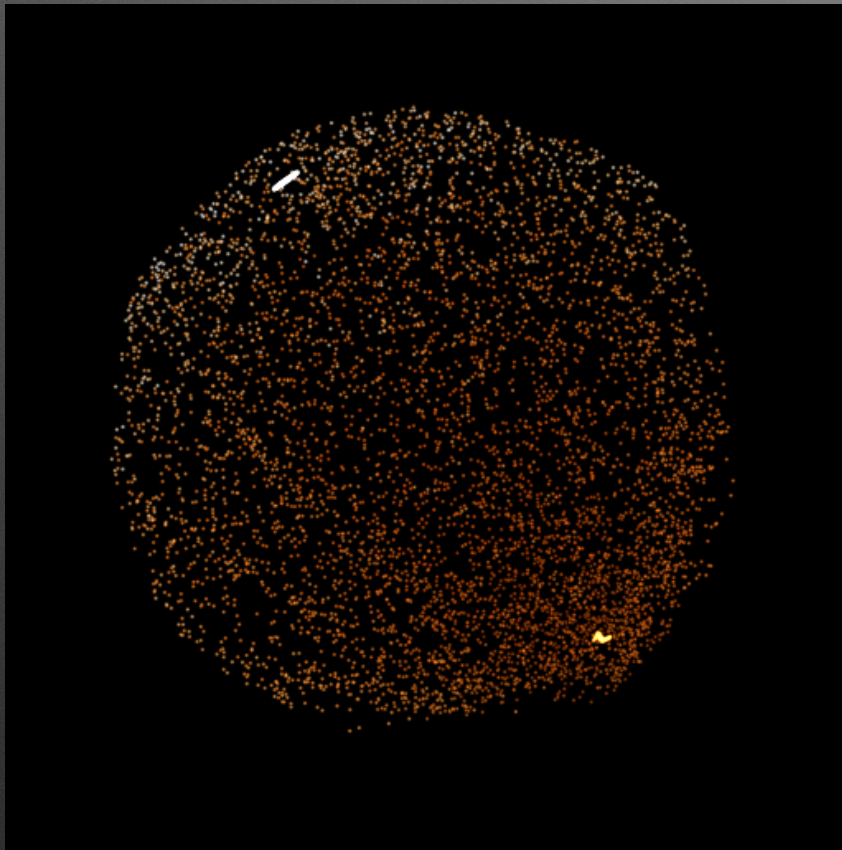
[Hu et al. 2010]

Multiple scattering (refraction + boundary)



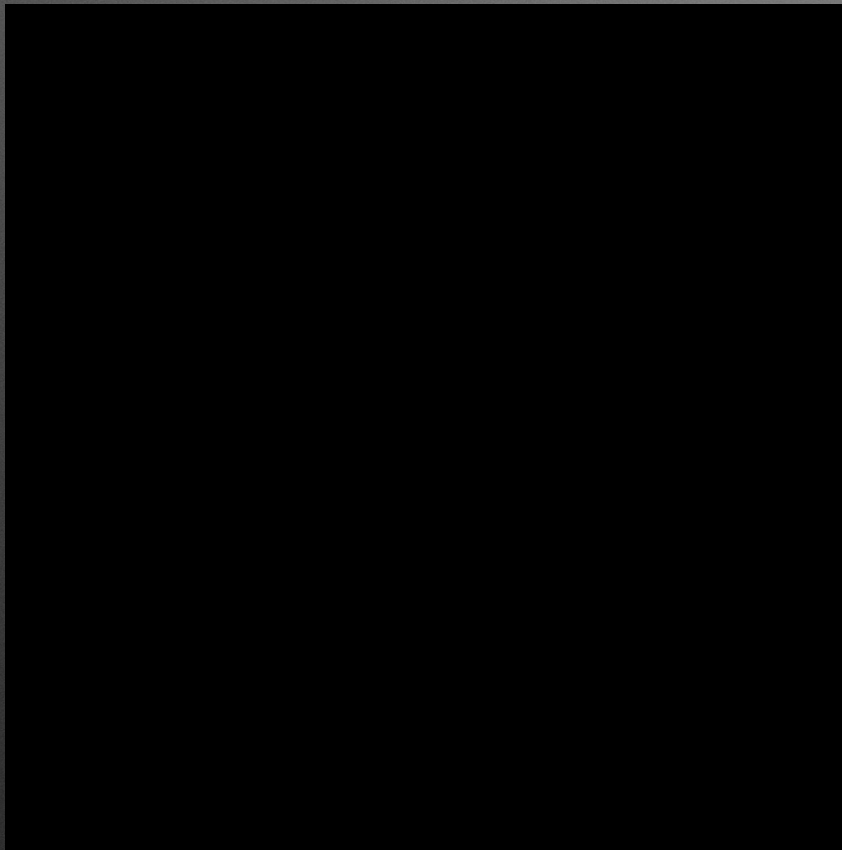
Sphere + BDPT +
spherical light source
+ 2000 spp =
25 mn

Single scattering (refraction + boundary)



Sphere + BDPT +
spherical light source
+ 2000 spp =
15 mn

Single scattering (refraction + boundary)



Sphere + BDPT +
point light source
+ 2000 spp =
10 mn

Why is it so difficult?

Why *was* it so easy?

- (for the other methods, I mean)
- Light: straight line in space
- No refraction: straight line on screen
 - Follow the light, adapt sampling
 - Still possible if single refraction

Why is it so difficult? (2)

- Scattering: not just surface phenomenon
- Sample *in* the volume
- Along the ray
- 3D problem