GONE IN 60 NANOSECONDS

東京理科大学
TOKYO UNIVERSITY OF SCIENCE

(Adventures in Transient Rendering & Stuff)

Roland Schregle
RS SciComp

roland.schregle@gmail.com

Nozomu Yoshizawa,
Department of Architecture
Tokyo University of Science, Japan

yosizawa@rs.noda.tus.ac.jp

19th International Radiance Workshop

18-20 de Agosto, 2021
Bilbao, España

Overview

1. Motivation
2. Steady state vs. Transient state
3. Transient photon map
4. Transient (4D) kd-tree
5. Video: Cornell box
6. Video: Villa Müller
7. Conclusions and future work

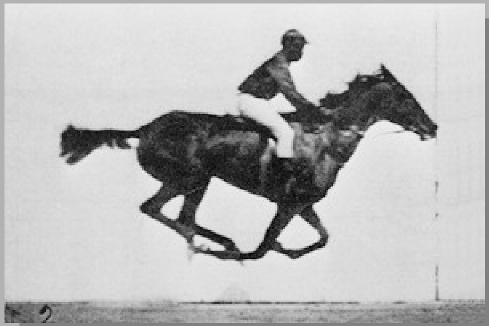
Motivation

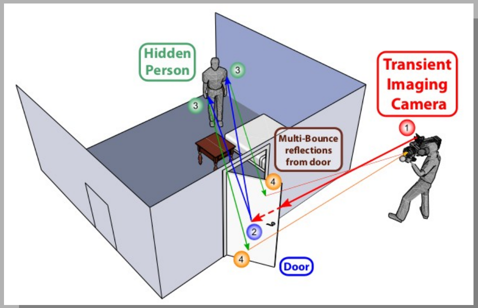
Develop a prototype that simulates propagation of light in extreme slow motion to:

- Gain insight into the light transport process (“light flow”)
- Assess directionality which may be subdued in steady state
- Assess influence of individual fenestrations, skylights, scattering surfaces in temporal isolation
- Assess colour bleeding effects before they are neutralised in steady state
- Simulate & visualise the imperceptible!
- Blow ya mind!

Motivation: High Speed Imaging

- High-speed photography
  - Muybridge racehorse (1848, Δt~14.5 ms)
  - Air gap flash (Edgerton, 1960s, Δt<1 μs)
- Femtophotography
  - Streak camera + pulsed laser (Velten, Raskar et al, 2010s, Δt<2 ps)
  - Non-line of sight imaging (Kirmani, Velten, Raskar et al, 2010s)

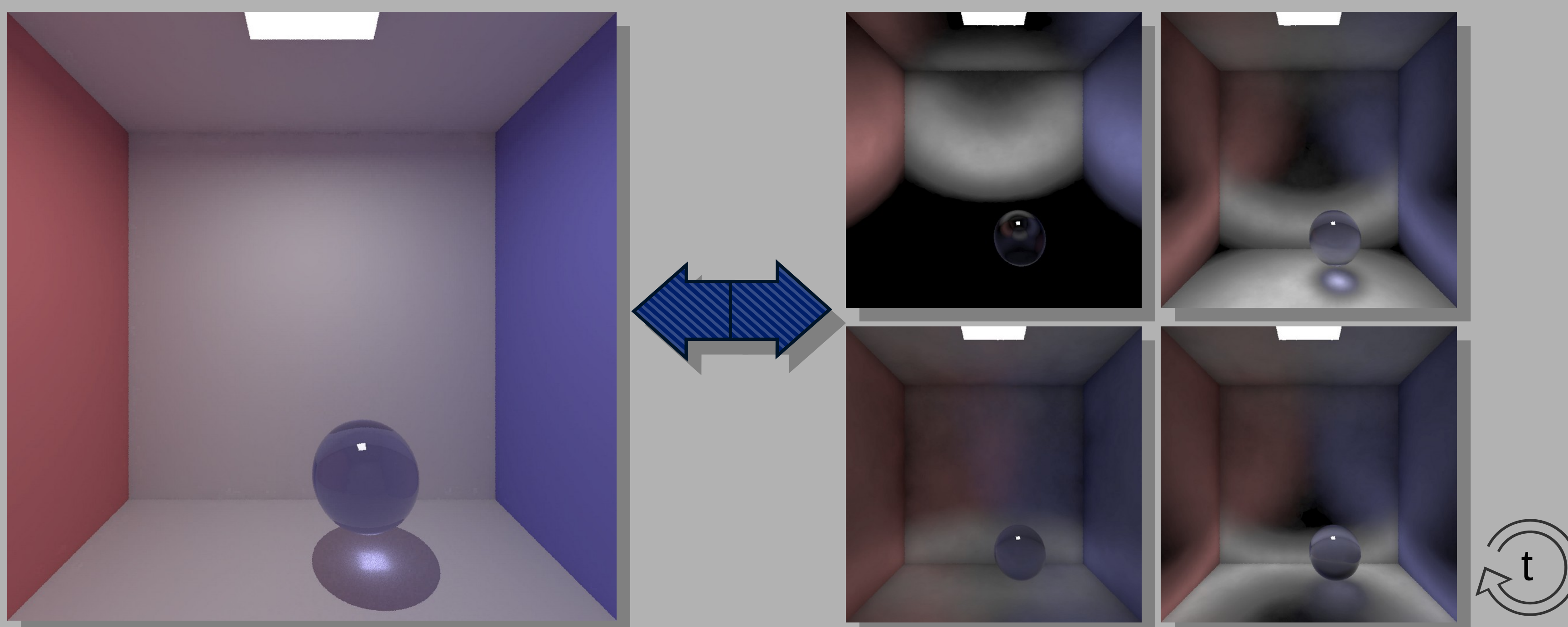




Source: Wikipedia, Michael Hoppe Gallery, MIT Media Lab

Motivation: Steady State vs. Transient Rendering

- We experience, simulate and analyse light in steady state
- Steady state implies a temporal equilibrium in lighting distribution
- <u>But</u>: steady state may obscure light flow visible in transient state



Motivation: Steady State vs. Transient Rendering

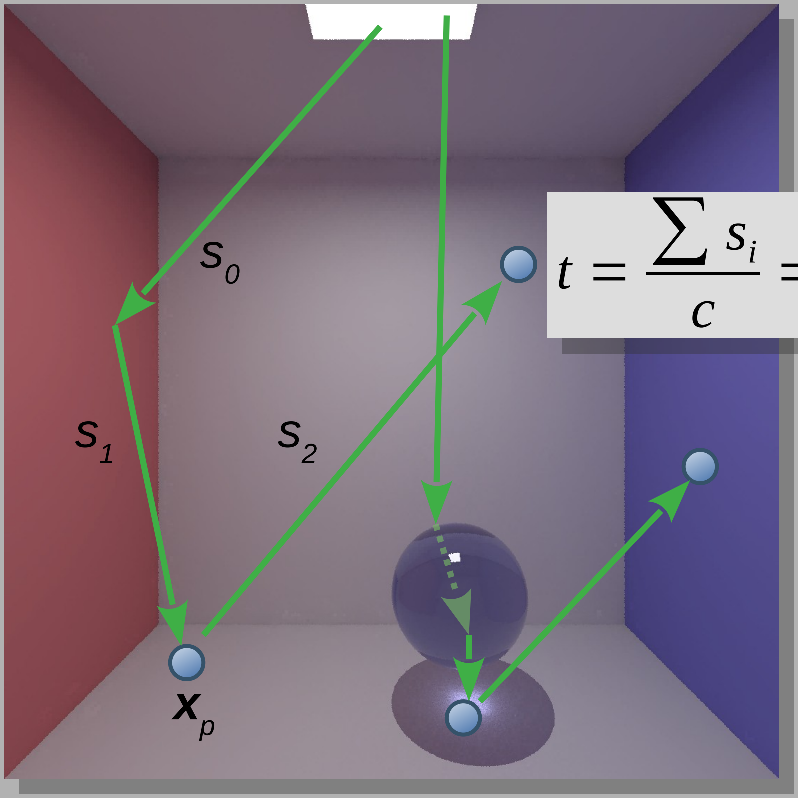
- Solve transient rendering equation; Radiance is function of time t:

$$L_r(\mathbf{x},\boldsymbol{\omega}_r,t) = L_e(\mathbf{x},\boldsymbol{\omega}_r,t) + \int_\Omega L_i(\mathbf{x},\boldsymbol{\omega}_i,t)\, f_r(\mathbf{x},\boldsymbol{\omega}_i,\boldsymbol{\omega}_r)\cos\theta_i \, d\omega$$

- Light propagates at c = 299 792 458 m/s in vacuum
- Simplifying assumptions:
  - No relativistic effects (Lorentz contraction, doppler effect)
  - Propagation time over distance s: $t(s) = \int_s \frac{\eta(\mathbf{s})}{c} ds$
  - Assume vacuum, constant index of refraction $\eta = 1$
    → Ignore effects of media (e.g. dielectrics, interfaces)
    → Propagation distance over time t: $s(t) = ct$

Transient Photon Map

- Approximate transient rendering equation with modified photon map
- Scatter/deposit "light particles" emitted from sources → precomp. ambient lighting
- User specifies <u>constant</u> speed of light *c* <u>scaled to geometry</u> → η ignored in dielectric, etc. → photon time of flight *t* from total path length s_p
- Store photon position $\boldsymbol{x}_p$, flux Φ_p, path length s_p

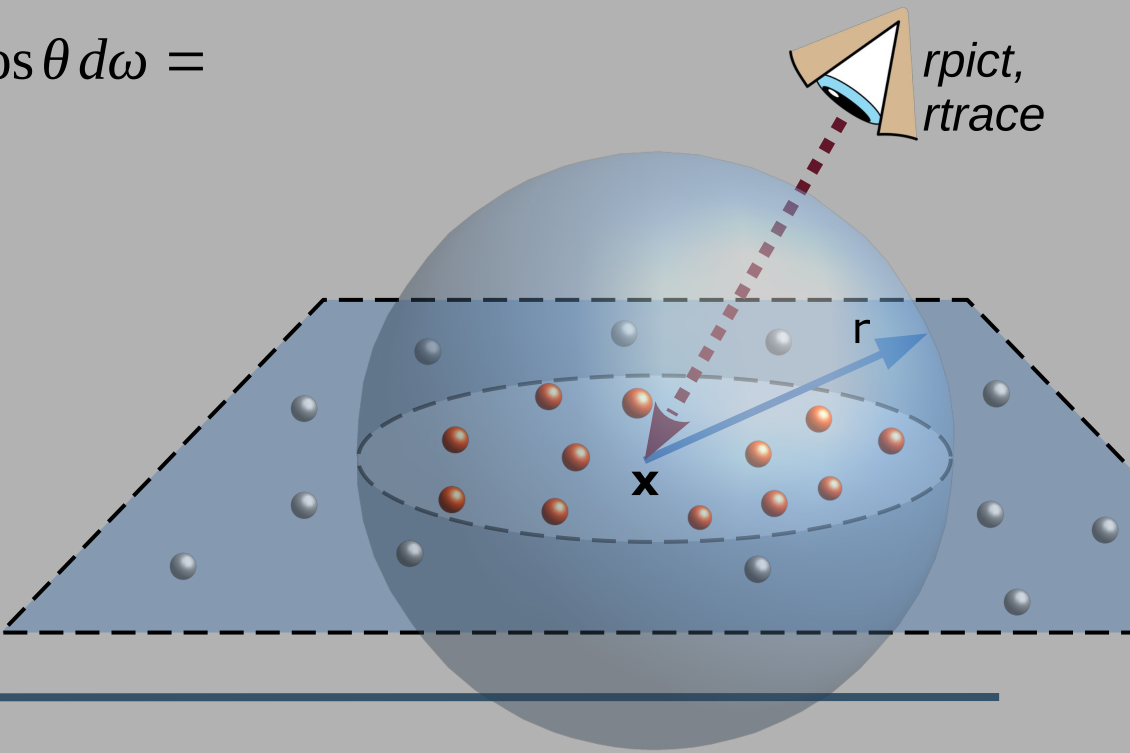


$$t = \frac{\sum s_i}{c} = \frac{s_p}{c}$$

Transient Photon Map: Density Estimate

- *rpict/rtrace* looks up photons to evaluate irradiance
  → couples to photon paths → bidirectional raytracer
- Irradiance *E(**x**, t)* proportional to photon density around ***x***, scaled by flux *Φ*

$$E(\mathbf{x},t) = \int_{\Omega} L_i(\mathbf{x},\boldsymbol{\omega},t)\cos\theta\, d\omega = \int_{\Omega} \frac{d^2\Phi(\mathbf{x},\boldsymbol{\omega},t)\cos\theta}{\cos\theta\, d\omega\, dA}\, d\omega$$



Transient Photon Map: Density Estimate

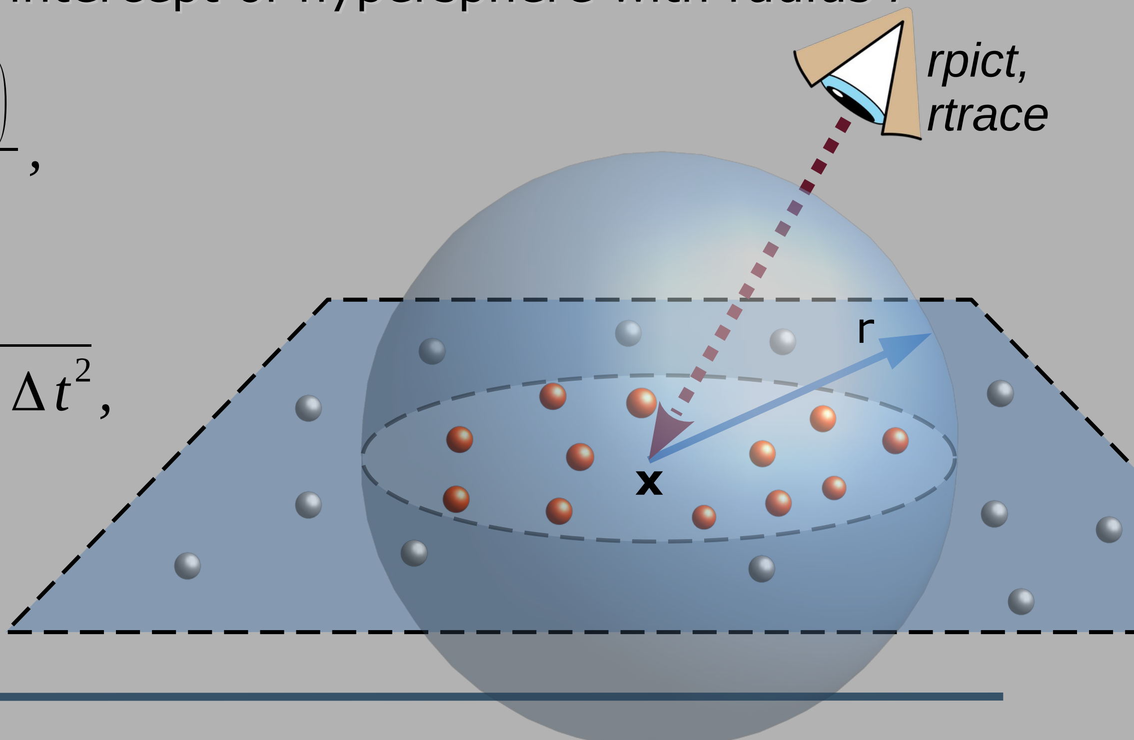
- Locate N photons by position and path length s_p (propagation distance over time t) → spatially consistent distance metric
- Bandwidth N defines spatiotemporal radii Δt, r
- Sum $\Delta\Phi_p$, normalise by intercept of hypersphere with radius r

$$E(\boldsymbol{x}, t) \approx \sum_{p=1}^{N} \frac{\Delta\Phi_p(\boldsymbol{x_p}(t))}{\pi r^2},$$

$$\{p : \|\boldsymbol{x_p}(t), \boldsymbol{x}\| \le r\},$$

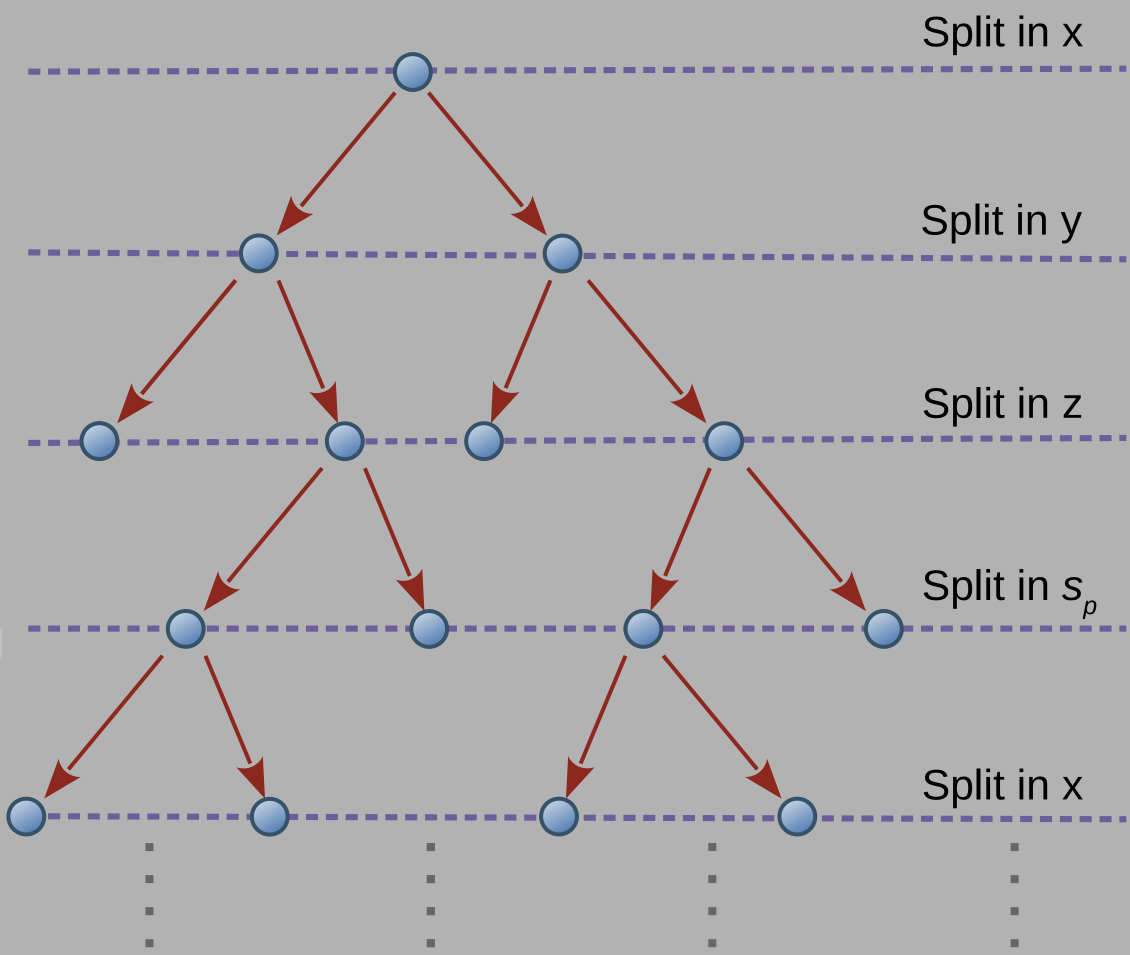
$$\|\boldsymbol{x_p}(t), \boldsymbol{x}\| = \sqrt{(\boldsymbol{x_p} - \boldsymbol{x})^2 + \Delta t^2},$$

$$\Delta t = d_p - ct$$



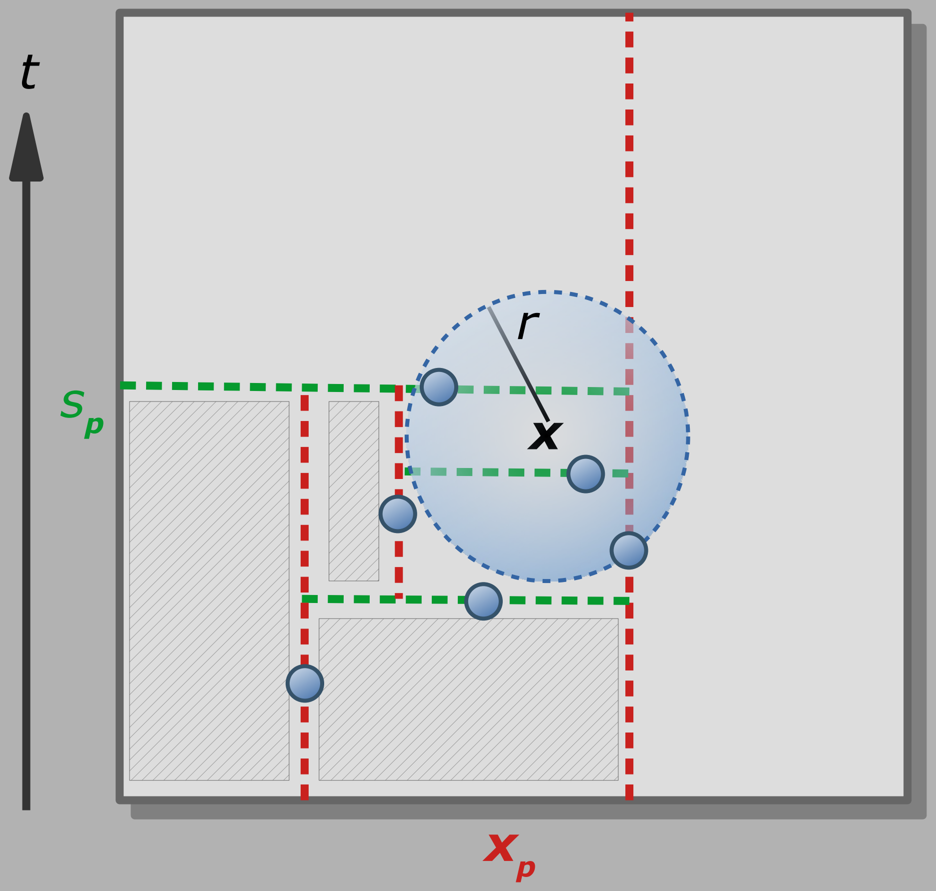
Transient Photon Map: 4D kd-tree

- Binary tree facilitates locating photons in space + time = 4D
- Nodes correspond to photons
- 4D keys [position $\boldsymbol{x}_p$, path length s_p] in consistent units
- Space subdivided along alternating axes when descending tree



Transient Photon Map: 4D kd-tree

- Space subdivided along alternating axes when descending tree
- Regions not overlapped by search radius *r* are culled



Transient Photon Map: Commands

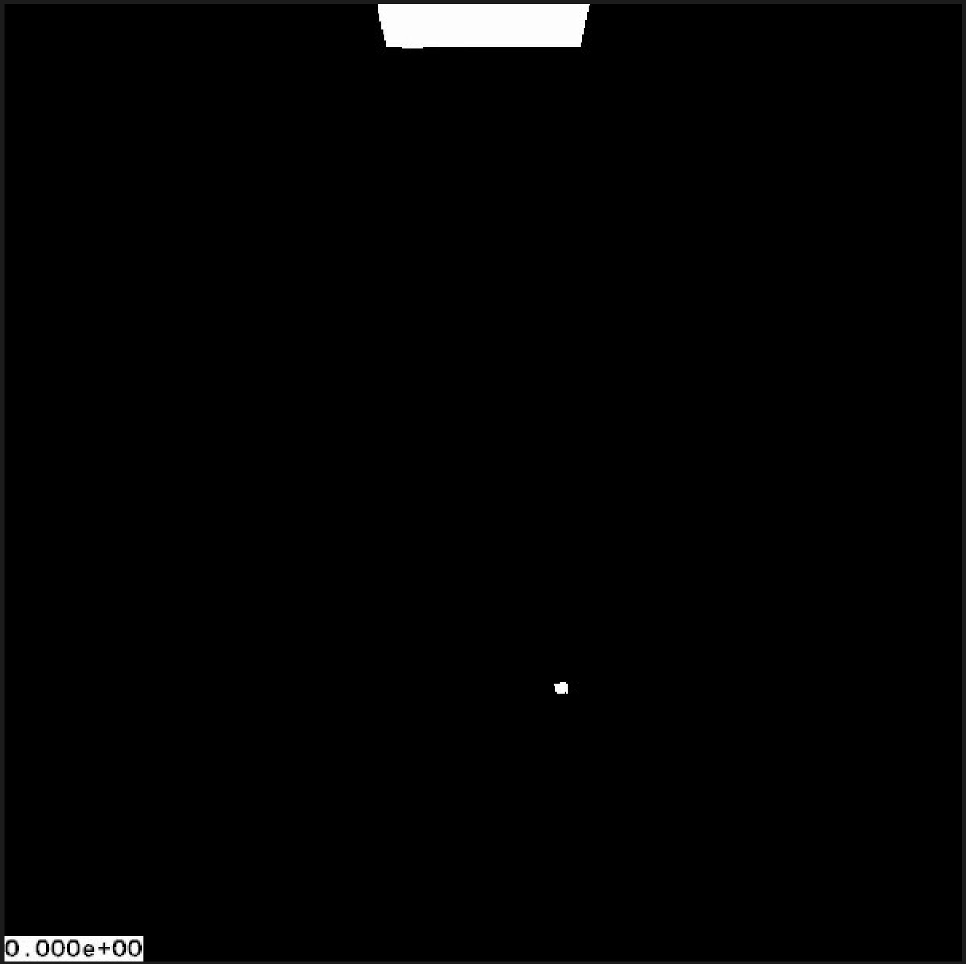
- Generate 100 million transient photons
  (note *c* scaled to mm/sec):

  mkpmap -apt bonzo.tpm 100m <u>2.99792458e11</u> … bonzo.oct

- Render with *rpict* with bandwidth of 100 photons at *t*=5.0 nsec:

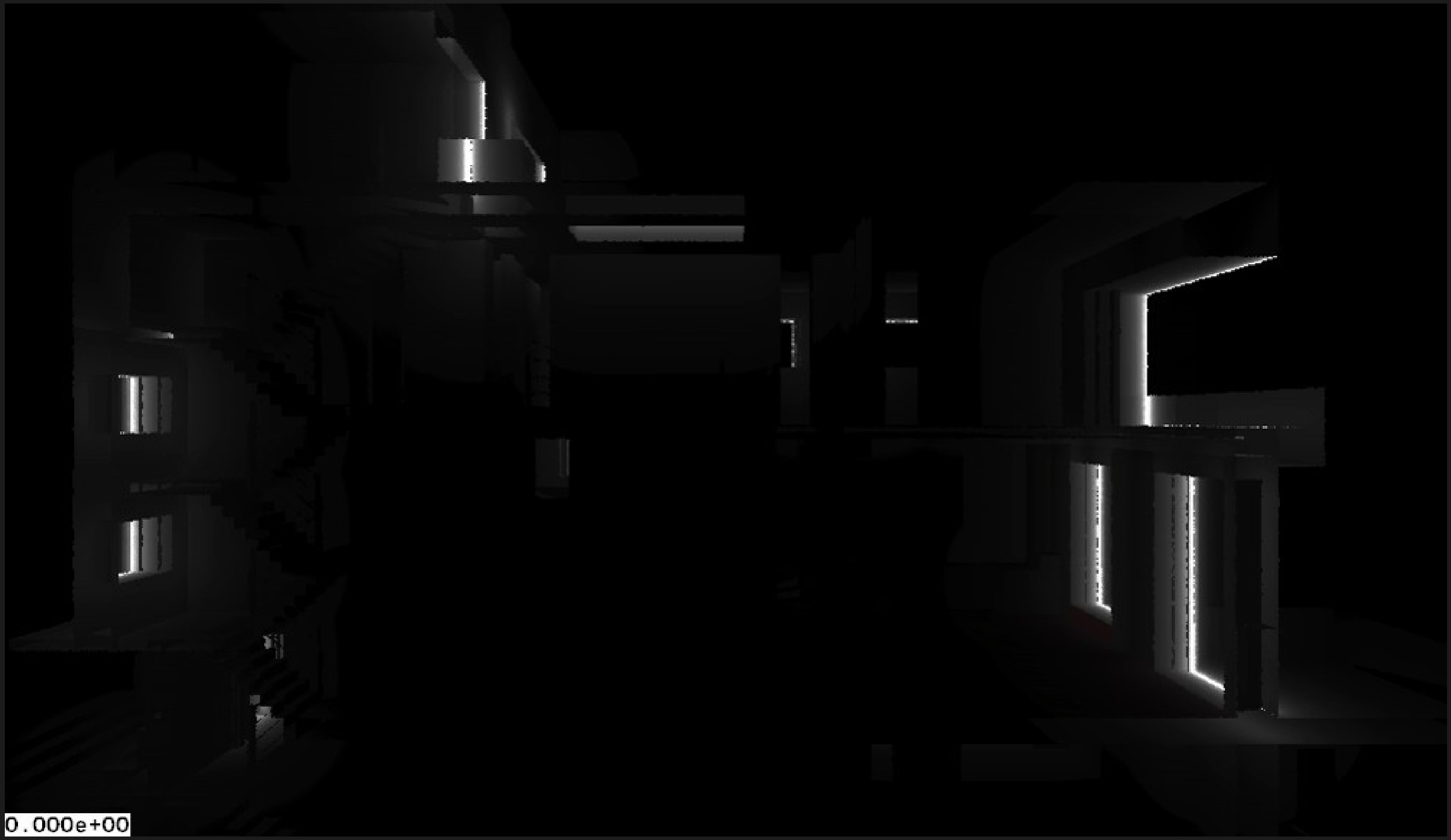
  rpict -ap bonzo.tpm 100 5e-9 -vf bonzo.vf … bonzo.oct

Results: Cornell Box

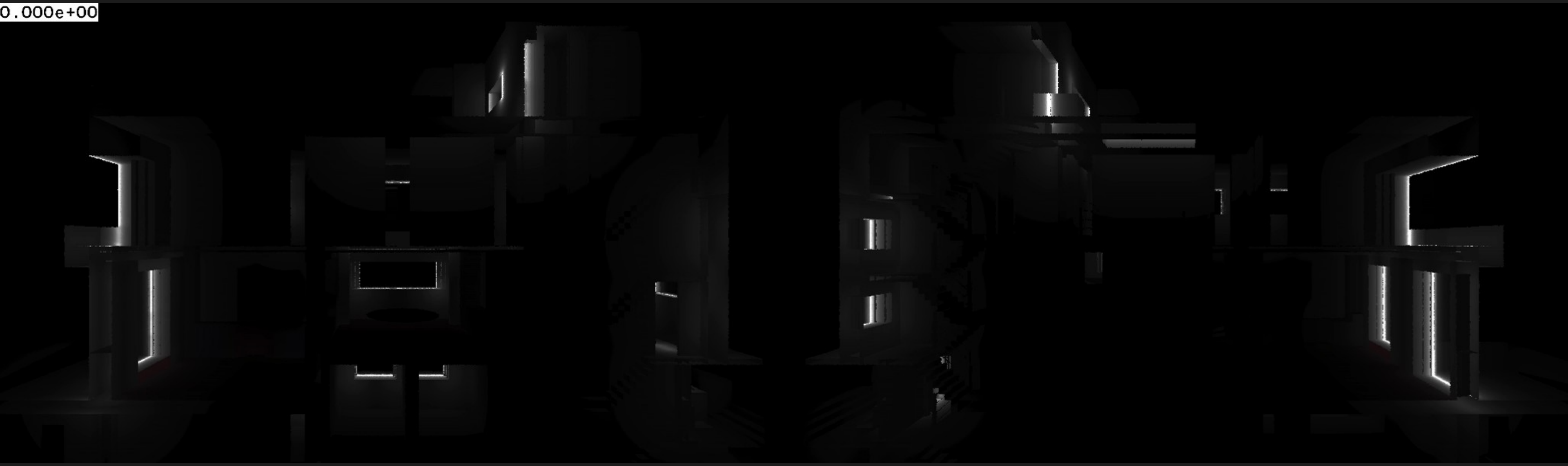


(What's wrong with dis pitcha?)

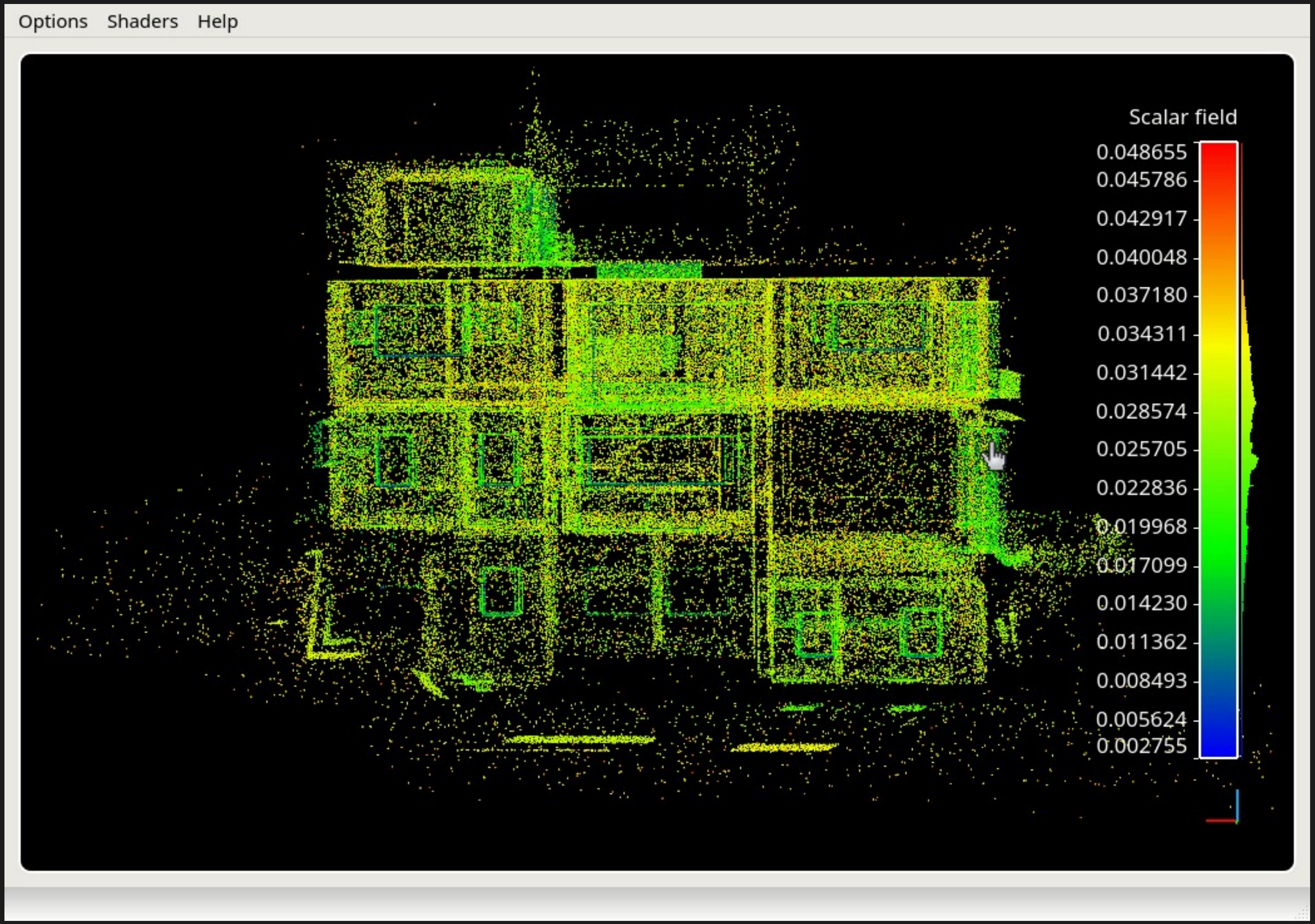
Results: Villa Müller



Results: Villa Müller (Duplex View)

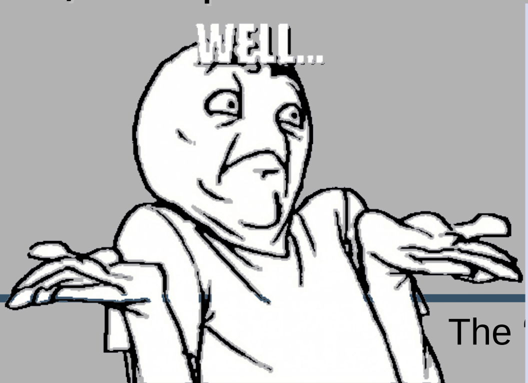


Results: Villa Müller (Point Cloud, Colour = Time)



Conclusions and Future Work

- Currently only prototype, lotsa limitations:
  - ✗ 4D kd-tree is in-core → max ~200M photons
  - ✗ *c* constant, ignores index of refraction (e.g. in dielectric)
  - ✗ Specular → eye/sensor rays still steady state
  - ✗ High spatiotemporal bias, noise → need better sampling in *t* ?
  - ✗ Max search radius adapts poorly to 4D (need *-am* option)
- Adapt out-of-core photon map to 4D (octree → hextree!)
- Photometric relevance/interpretation ?
- **Applications ?**



The "Whatever" Guy™

Thank You for Your Attention!

Project log: https://www.researchgate.net/project/3D-Light-Flow

This research was graciously supported by JSPS KAKENHI Grant Number JP19KK0115.

JAPAN SOCIETY FOR THE PROMOTION OF SCIENCE

日本学術振興会

科研費

KAKENHI