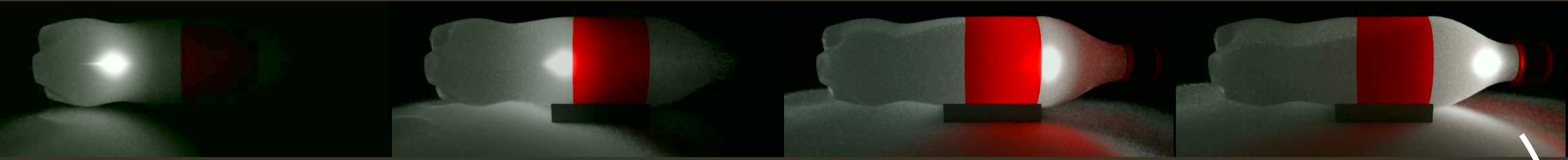




Pulse of light going through a bottle at one trillion frames per second (Camera Culture Group, MIT Media Lab)



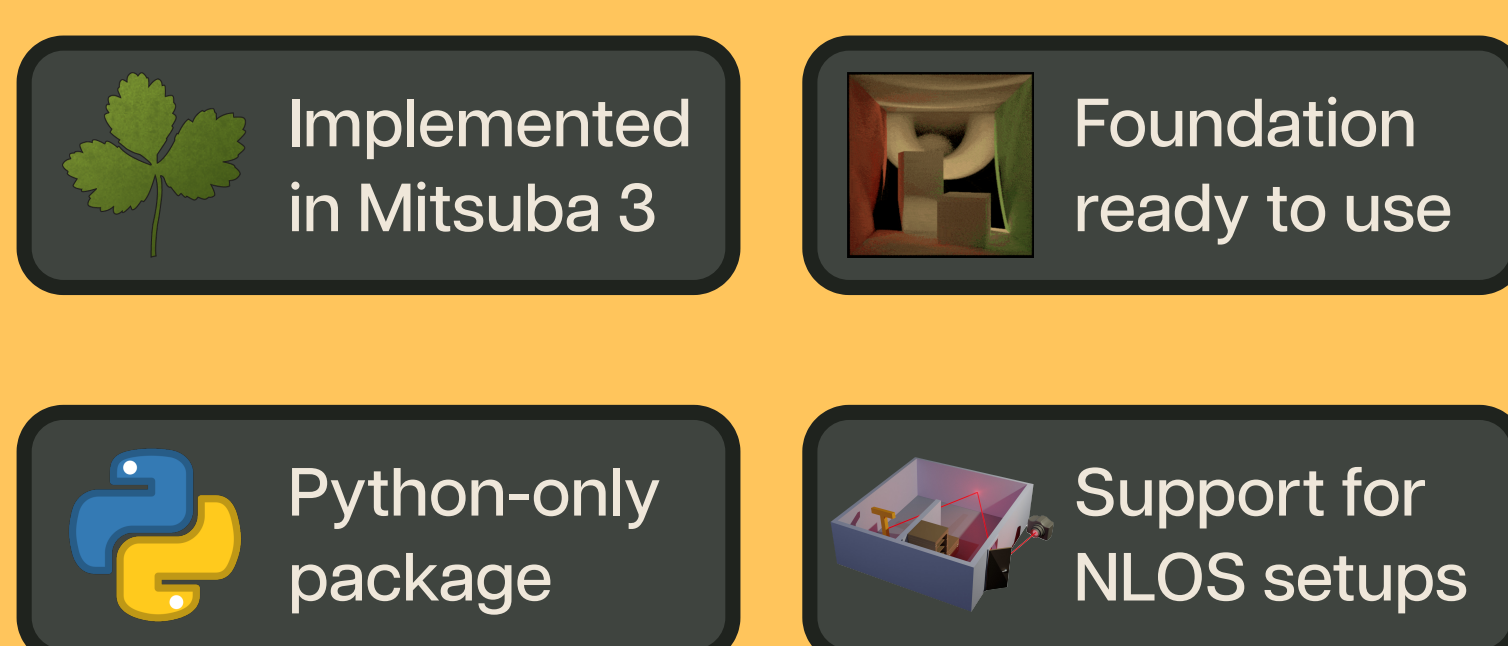
Our recreation of the bottle scene, rendered using MITRANSIENT

MITRANSIENT is a physically-based light transport tool for time-gated and time-resolved cameras, and more

Imaging hardware



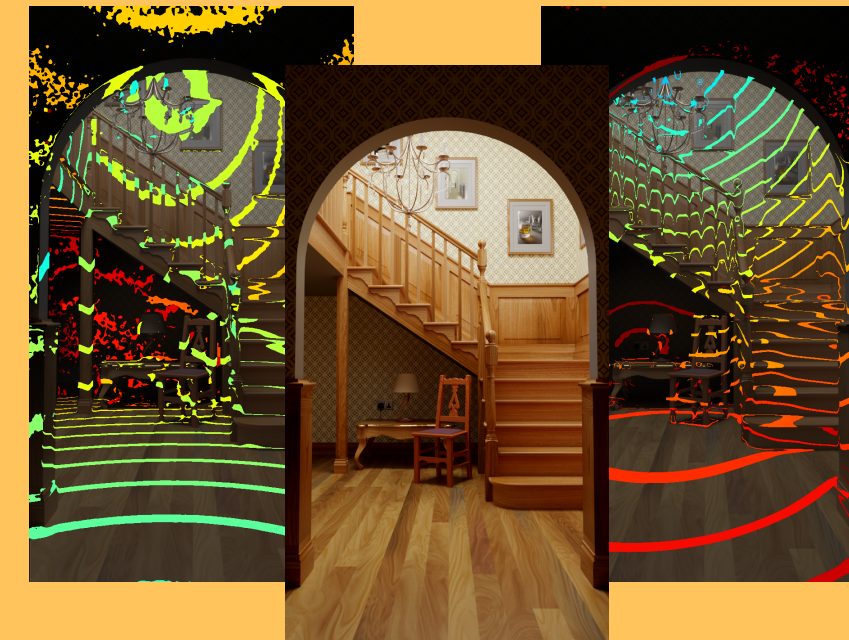
Main features



And many more!

- Spectral rendering
- Volumetric rendering
- Transient polarization
- Transient derivatives for inverse rendering
- Spatiotemporal filters
- Tools for NLOS setups

Visualization utilities



Check it out



Project page
Tutorials



Coca-Cola light
Full video



github.com/diegoroyo/mitransient

\$ pip install mitransient



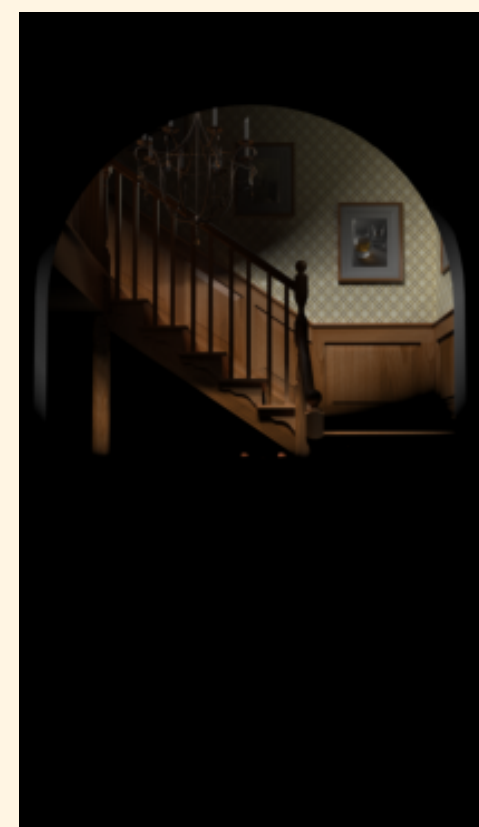
mitransient.readthedocs.io

Example: Staircase scene

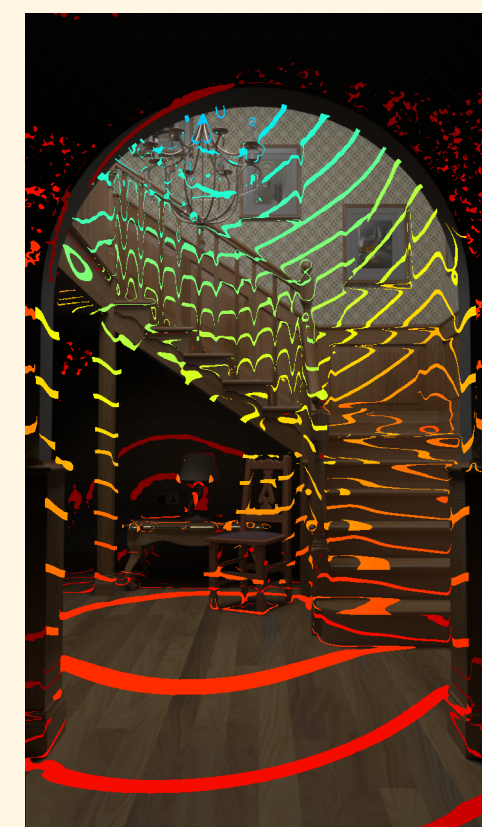
Steady state



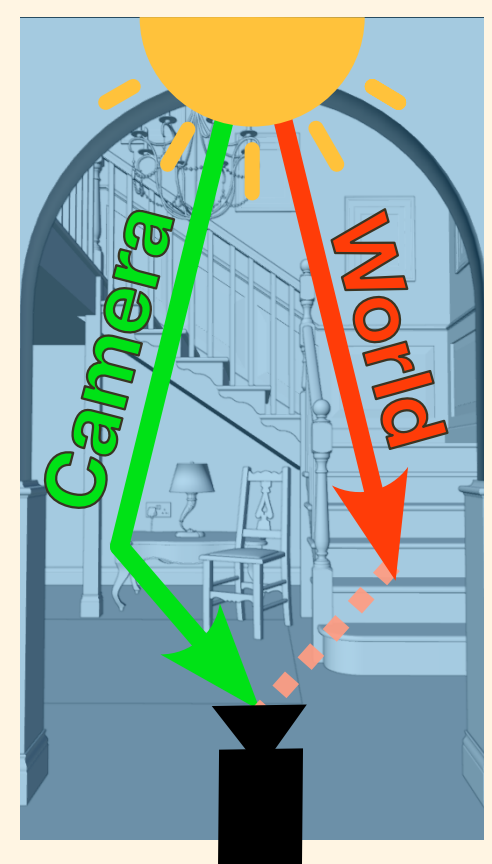
World time



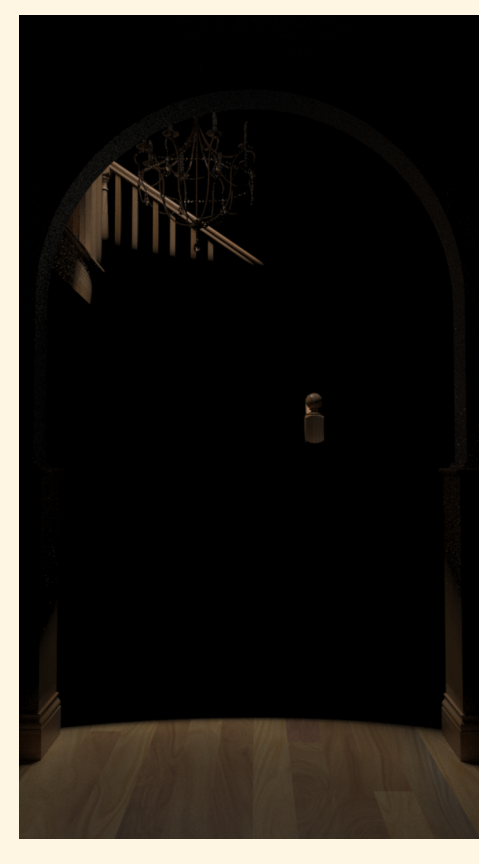
Rainbow vis.



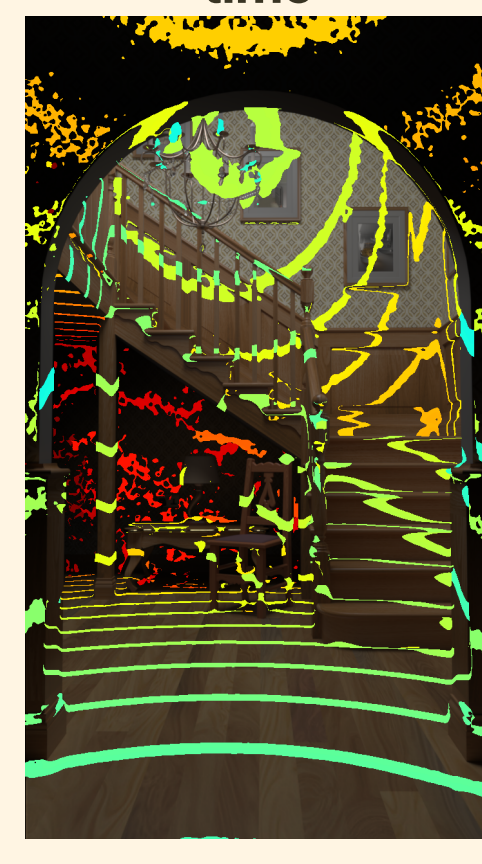
Scene model



Camera time



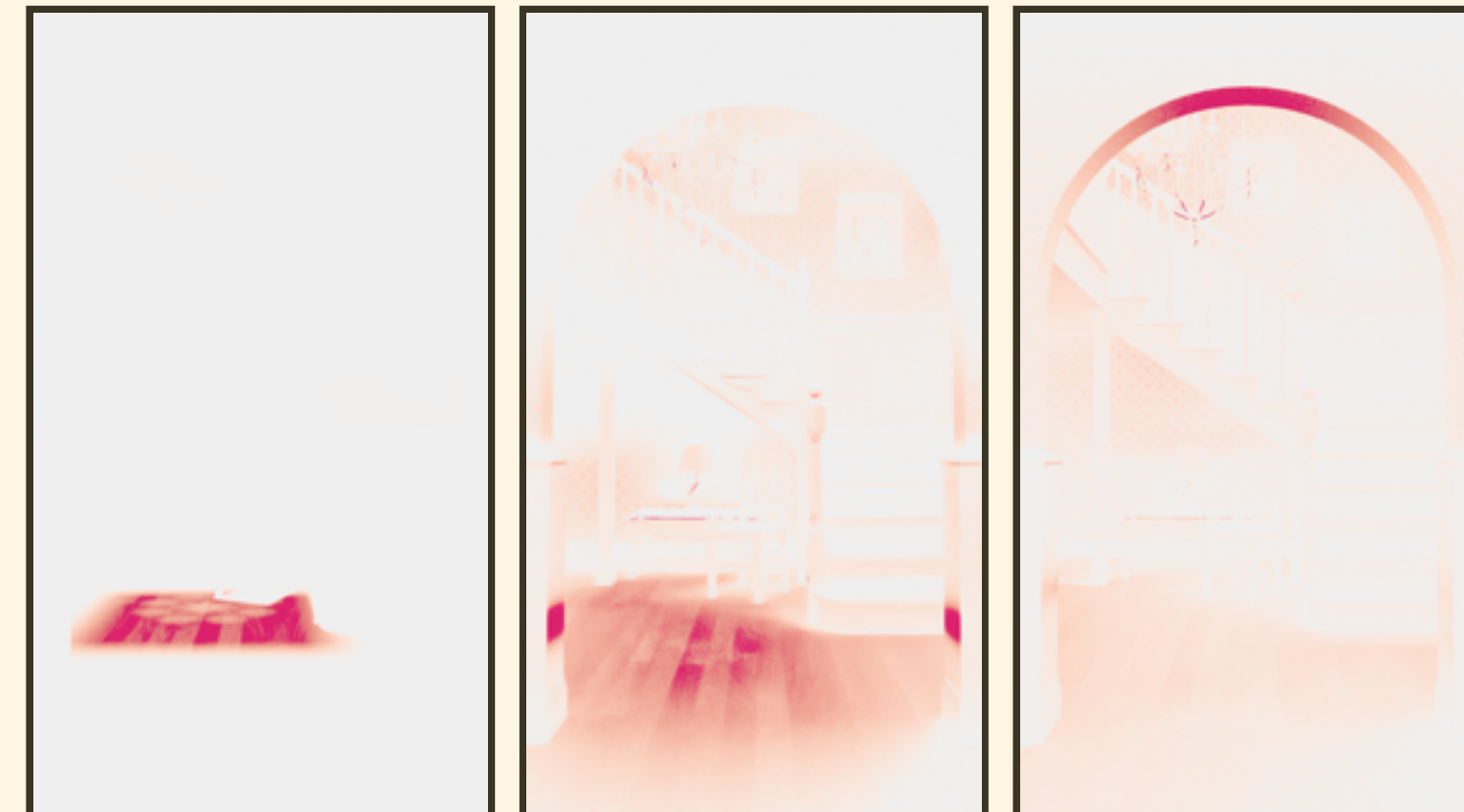
Time-resolved estimator



Transient derivatives

The effect of floor reflectance on the image pixels, over time

$dI(x, y, t; \theta) / d\theta_{\text{floor}}$



transient_path

```
<integrator
type="transient_path">
<bool
name="camera_unwarp"
value="true"/>
<integer
name="max_depth"
value="15"/>
<string
name="temporal_filter"
value="gaussian"/>
</integrator>
```

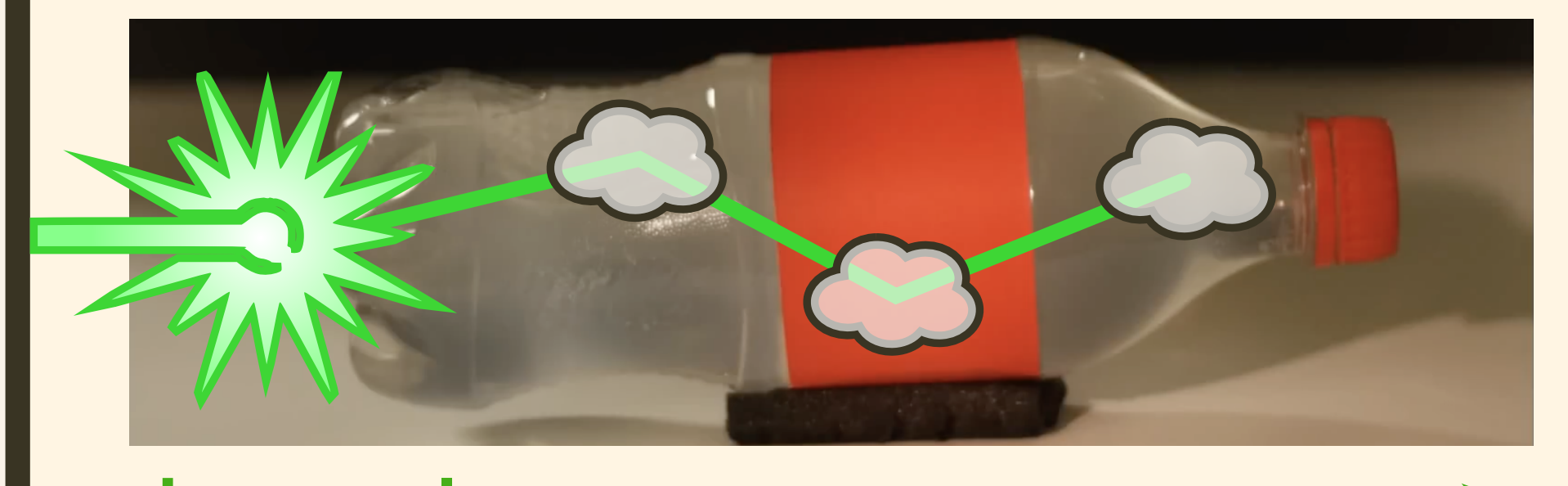
Video result

transient_hdr_film

```
<film
type="transient_hdr_film">
<integer
name="temporal_bins"
value="450"/>
<float
name="bin_width_opt"
value="0.02"/>
<float
name="start_opt"
value="0.0"/>
</film>
```

Volumetric rendering: Coca-cola light scene

Scene setup



Laser pulse

Volumetric integrator

transient_prbvolpath

```
<integrator
type="transient_prbvolpath">
<bool
name="camera_unwarp"
value="false"/>
<string
name="temporal_filter"
value="gaussian"/>
...
</integrator>
```

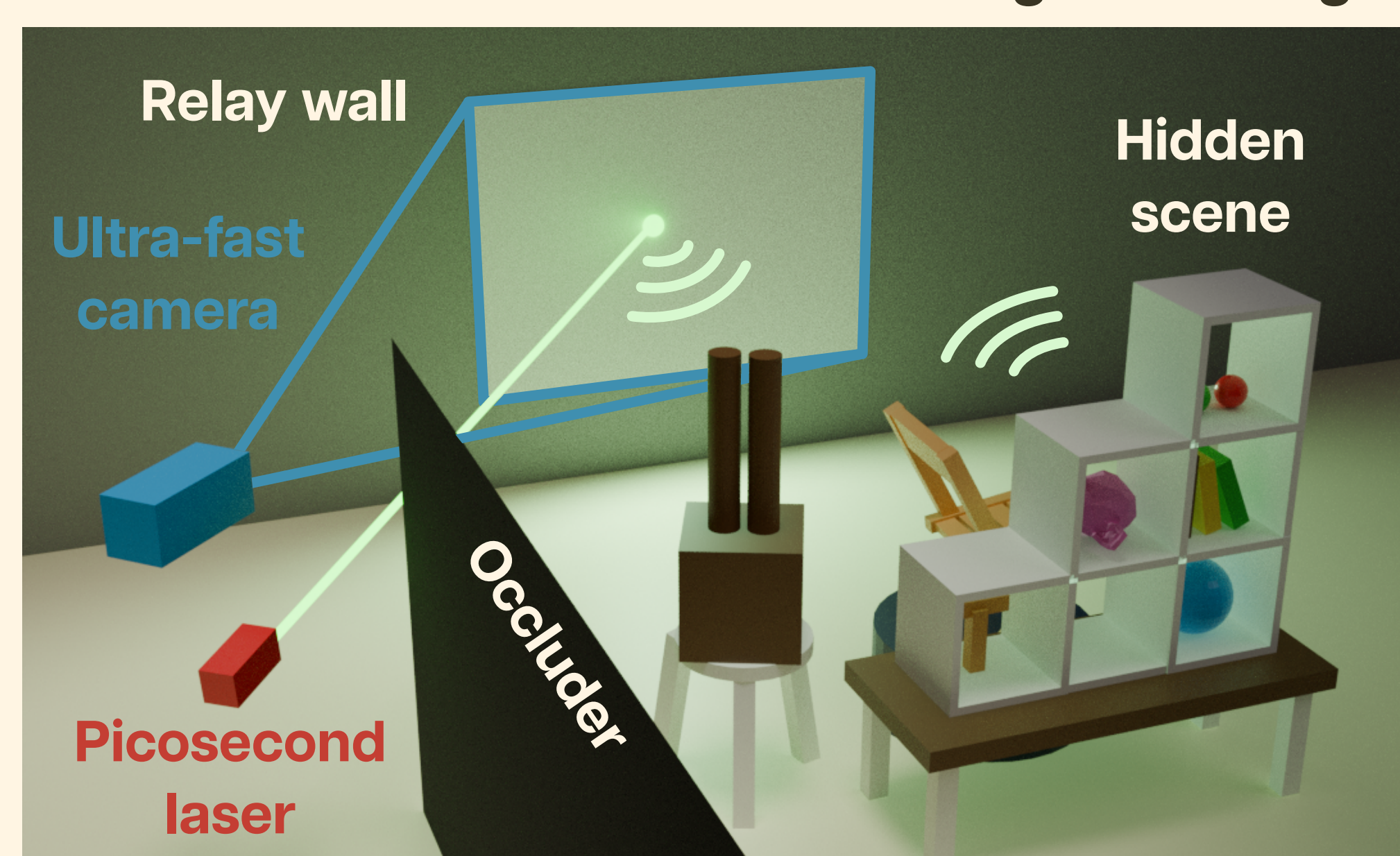
Homogeneous medium

homogeneous

```
<medium type="homogeneous">
<float name="albedo"
value="0.4"/>
<float name="sigma_t"
value="0.3"/>
<phase type="hg">
<float name="g"
value="0.7"/>
</phase>
</medium>
```

Non-line-of-sight imaging: office scene

Goal: reconstruct hidden scene using indirect light



Running is easy
with our tool Y-TAL

\$ tal render office

office/office.yaml

```
num_bins: 3000
bin_width_opt: 0.003
start_opt: 0.0

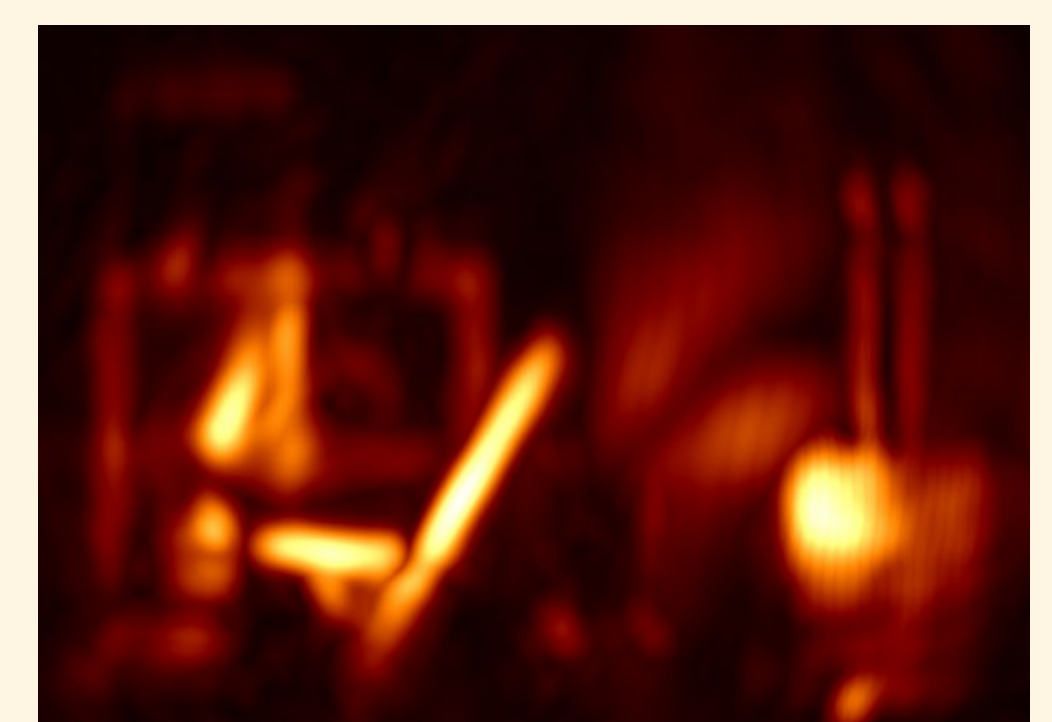
scan_type: single
sensor_width: 180
sensor_height: 130
laser_lookat_x: 90
laser_lookat_y: 65

geometry:
- mesh: # hidden scene
type: obj
filename: office.obj
- mesh: # relay wall
type: rectangle
```

Hidden scene



Reconstruction



[Liu et al. 2019]

Our simulation



Get Y-TAL



Depth &
normals

