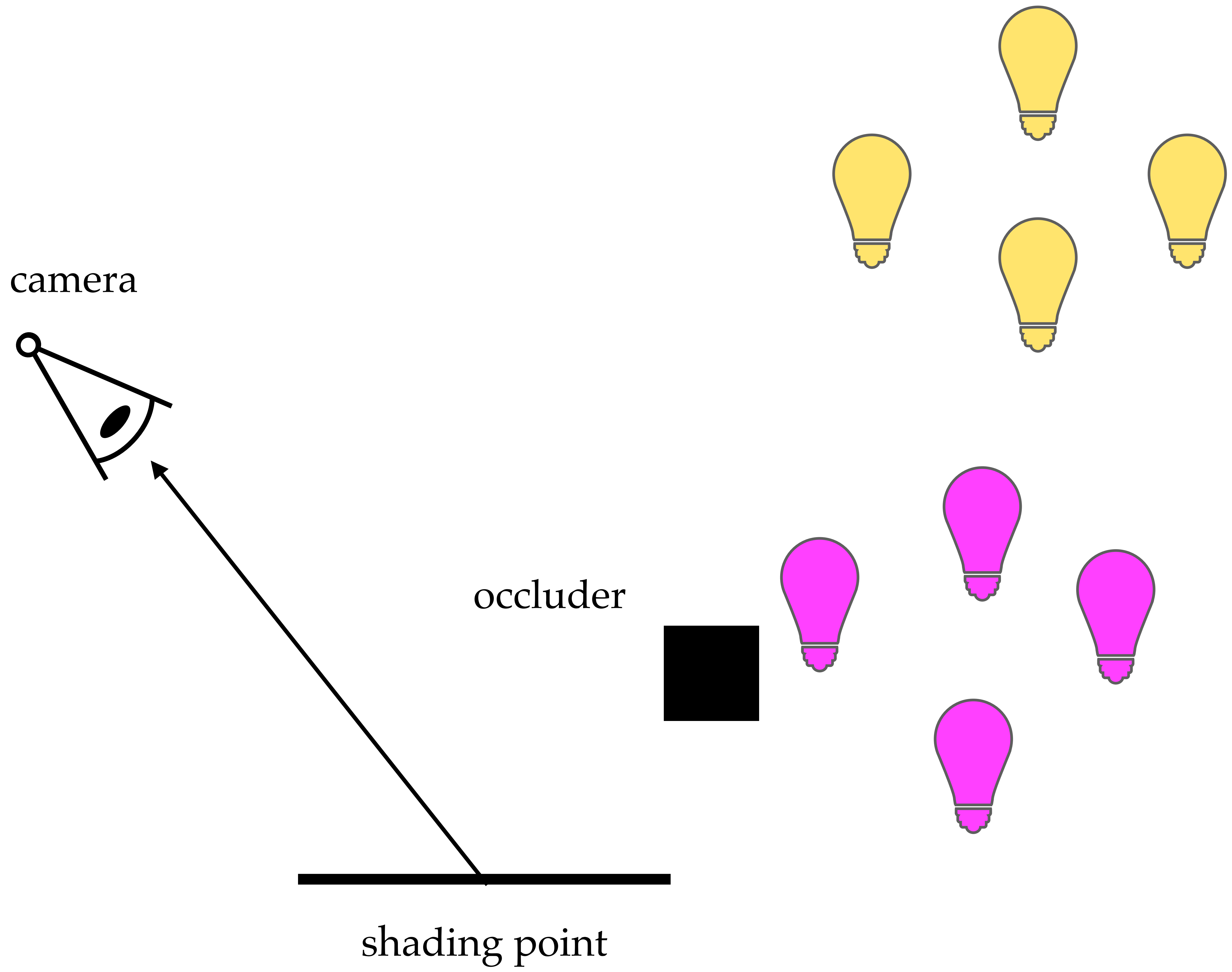
# Many-lights rendering

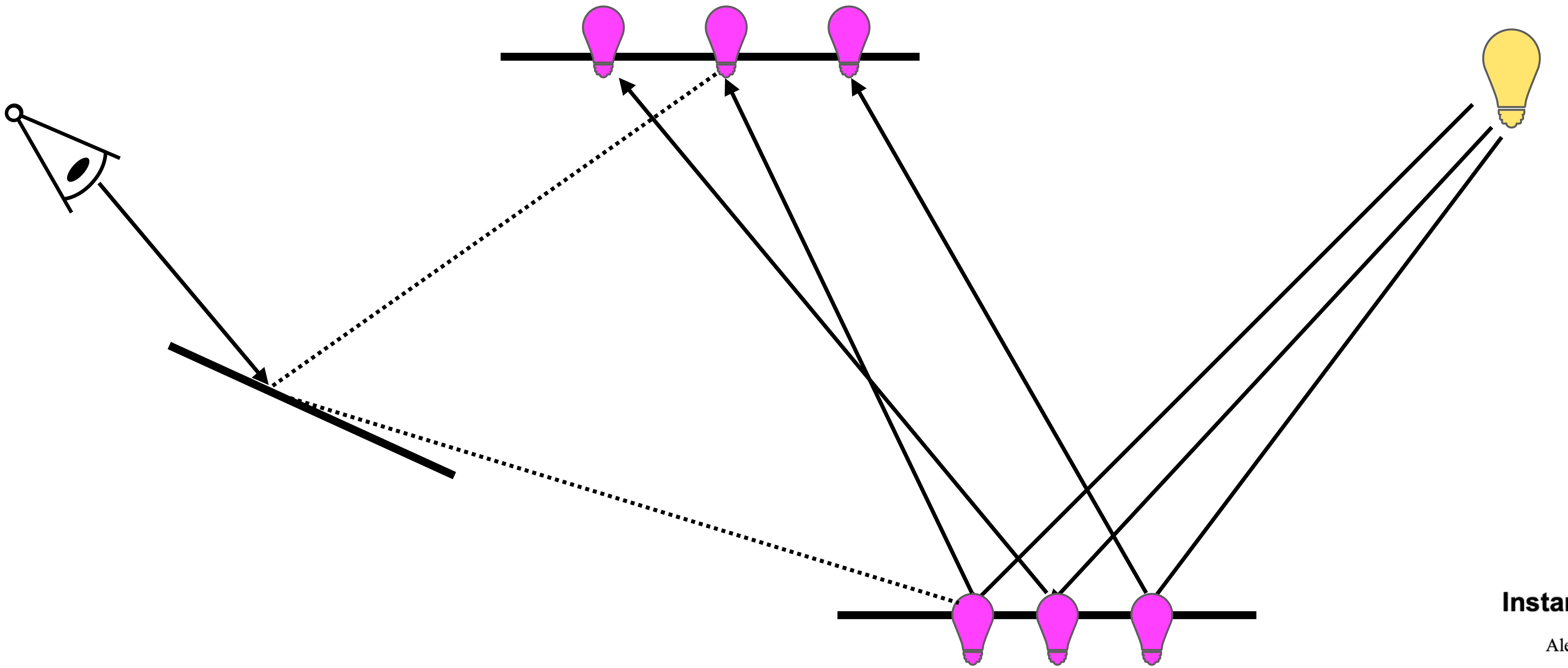
UCSD CSE 272
Advanced Image Synthesis
Tzu-Mao Li

# Rendering with many lights



# Virtual point lights

- we can deposit point lights at light subpaths in bidirectional path tracing



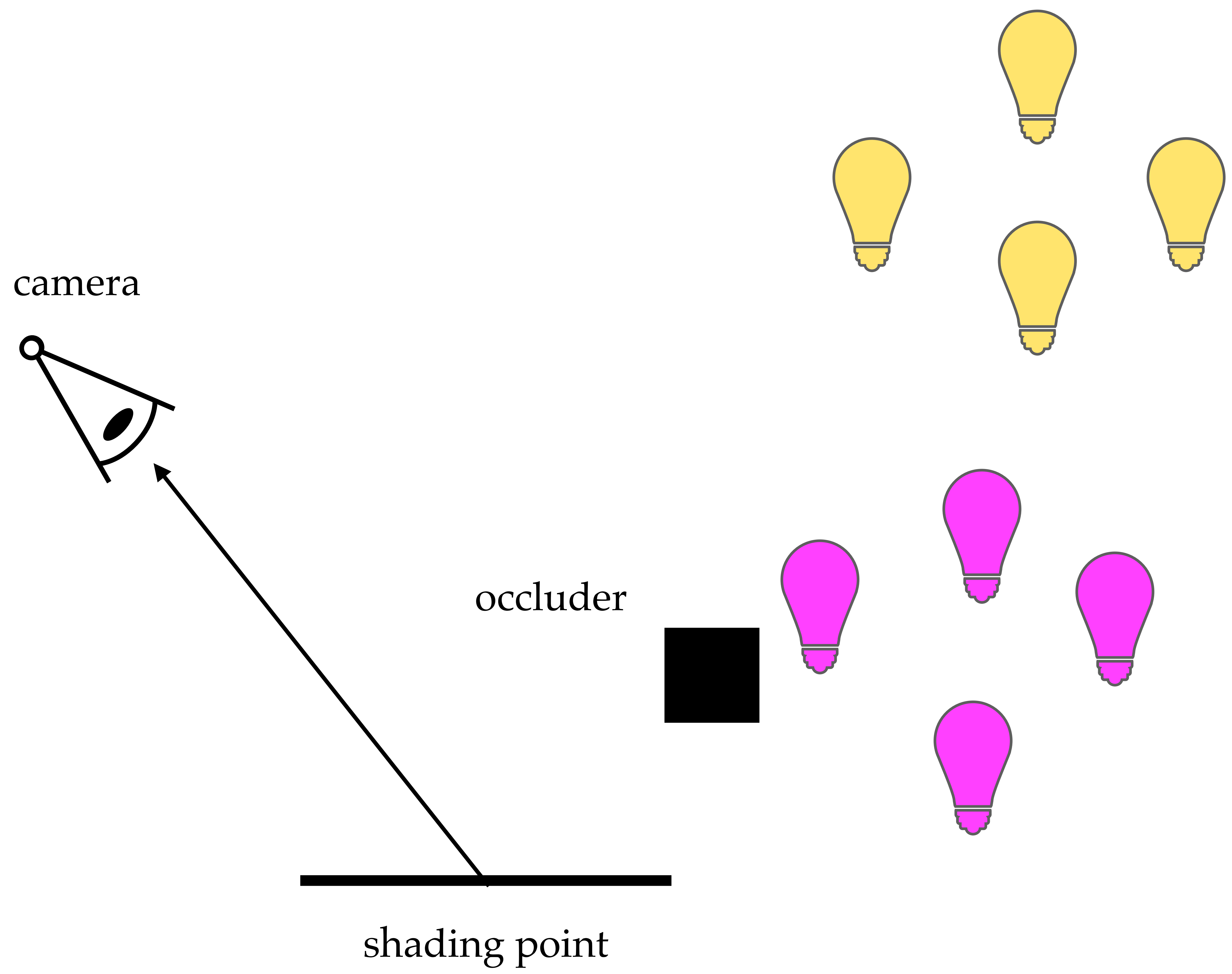
**Instant Radiosity**

Alexander Keller*

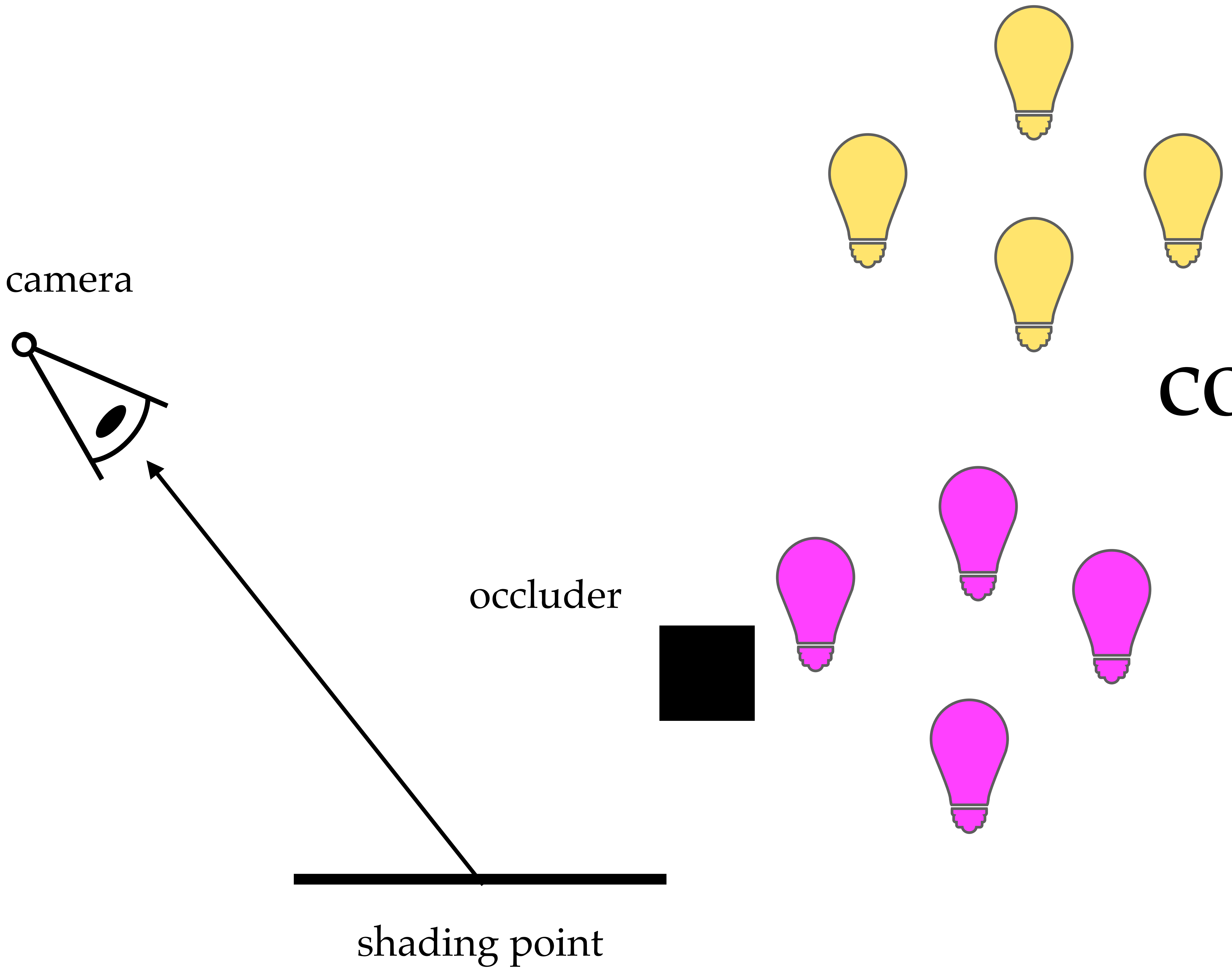
Universität Kaiserslautern

# Goal: importance sample lights

- important lights are:
  - closer
  - at BSDF peak
  - not occluded
  - high intensity



# Naive approach: importance sample light intensity



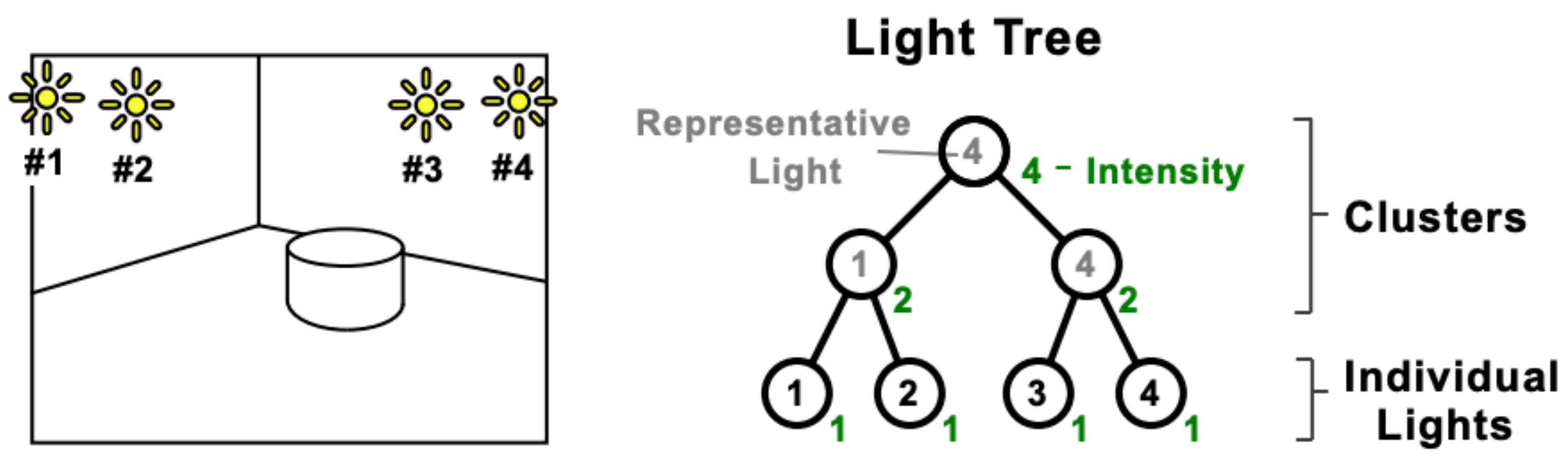

intensity
geometry term

$$\text{contribution} = L \cdot \rho \cdot G \cdot V$$

BSDF
visibility

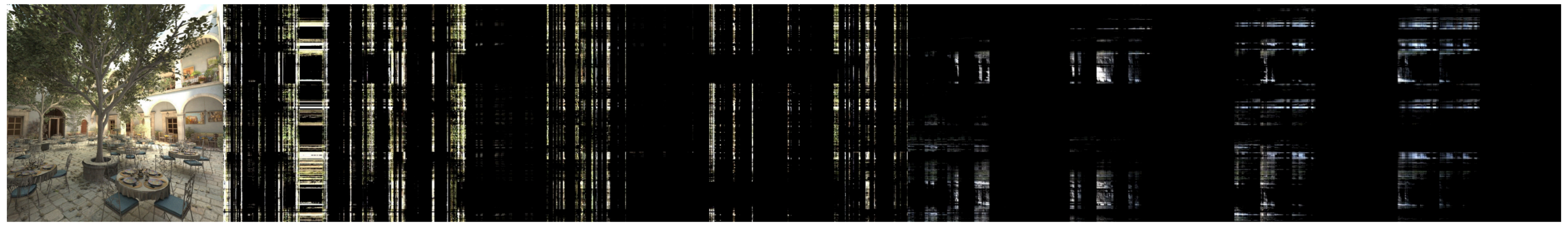
# Ideas

[Walter 2005]

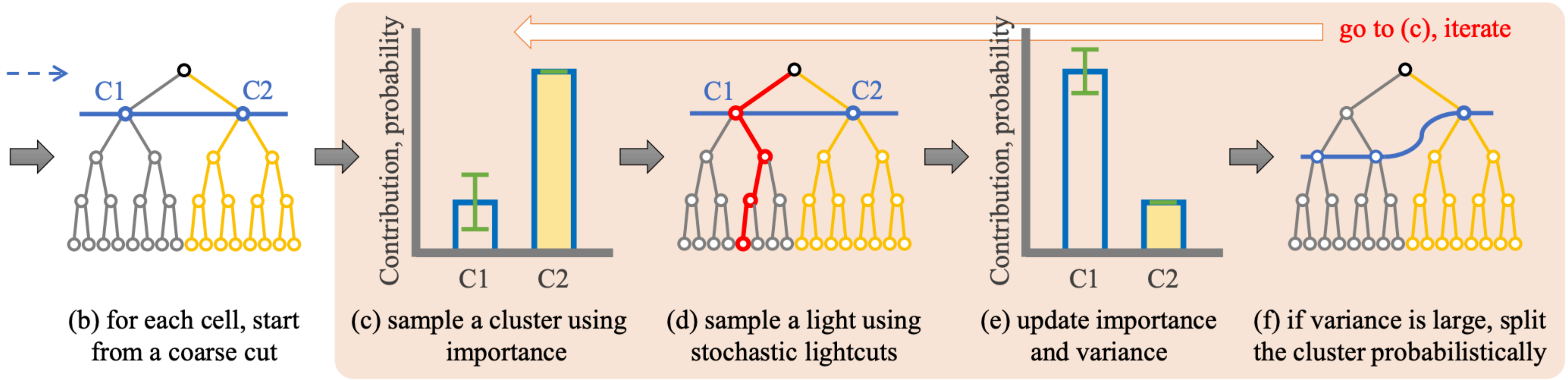



hierarchical clustering
[Shirley 1996,
Paquette 1998,
Walter 2005, …]

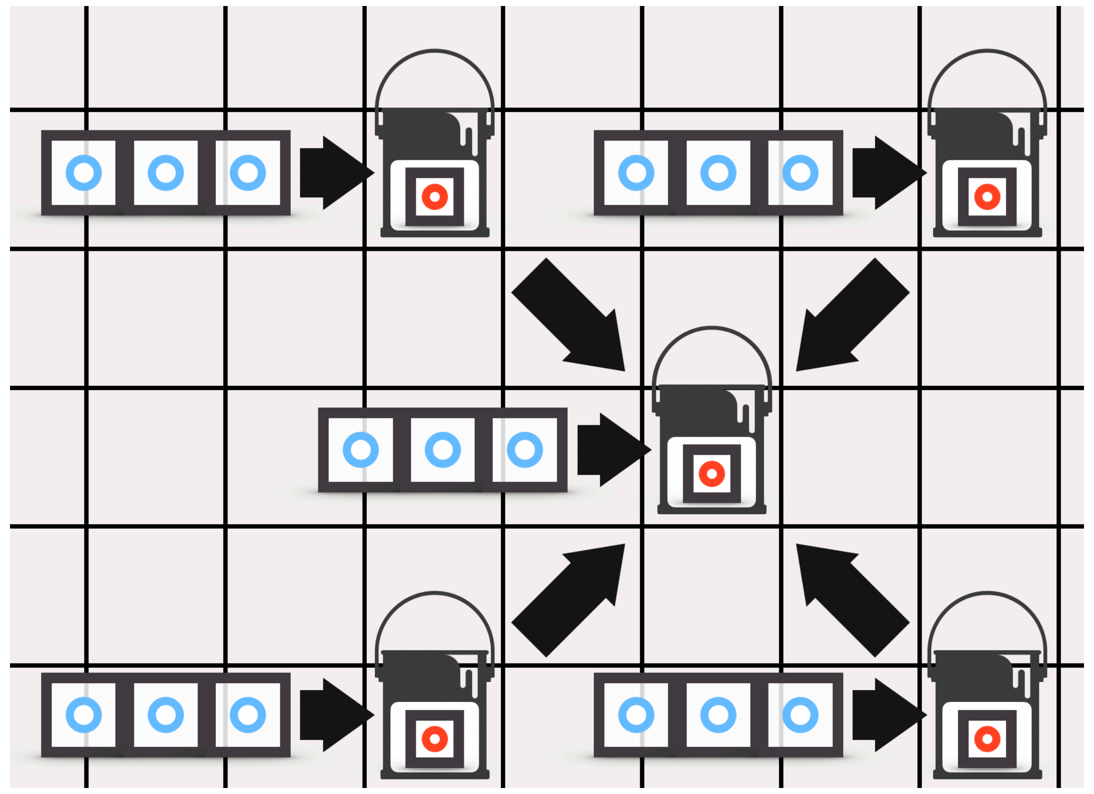
[Ou 2011]



matrix formulation
[Hasan 2007,
Ou 2011,
Huo 2015, ….]



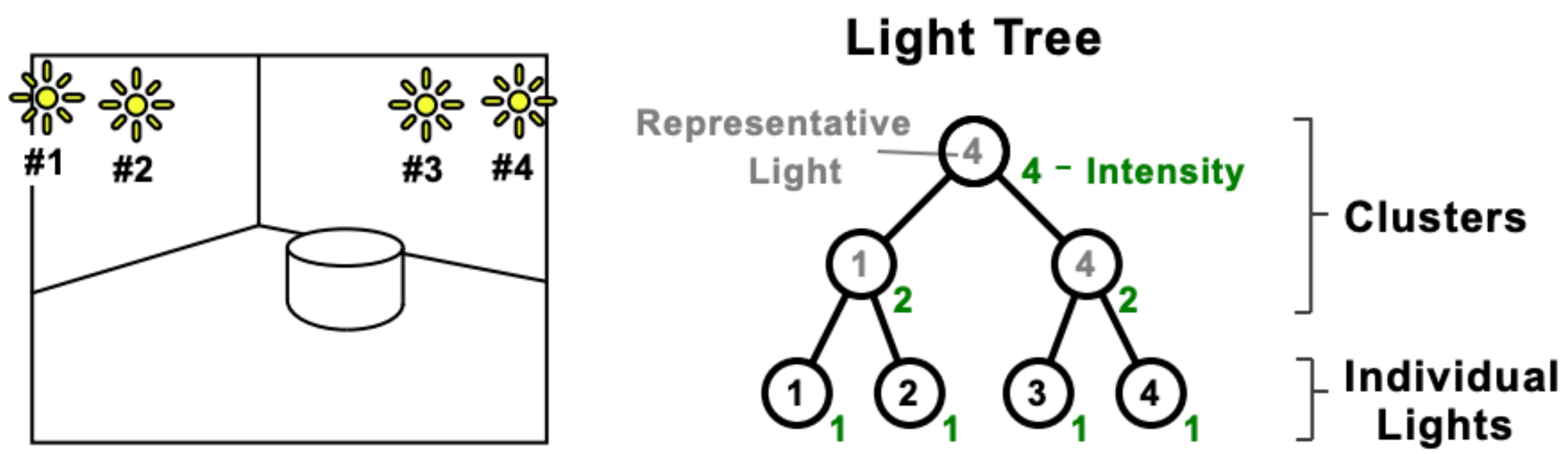

data-driven
[Donikian 2006,
Vevoda 2018,
Wang 2021]



spatial-temporal reuse +
resampling
[Benedikt 2020]

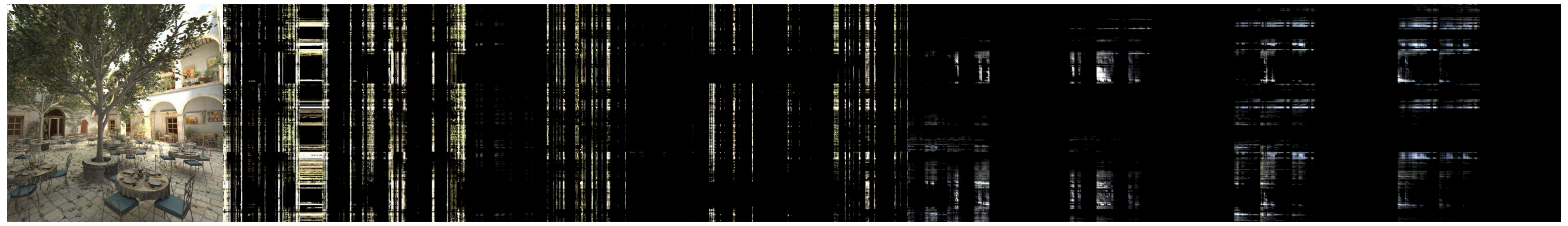
# Ideas

[Walter 2005]

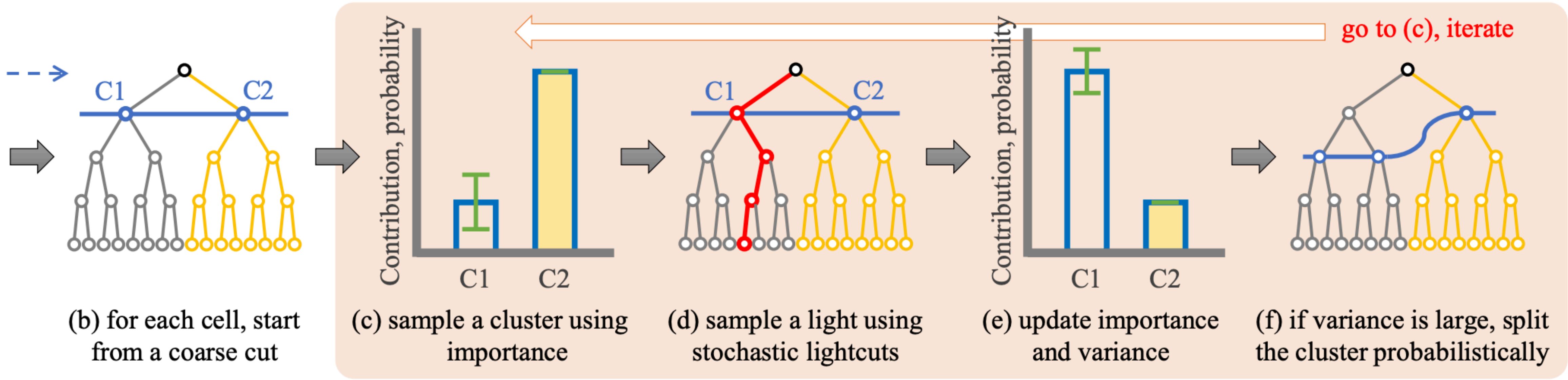



hierarchical clustering
[Shirley 1996,
Paquette 1998,
Walter 2005, …]

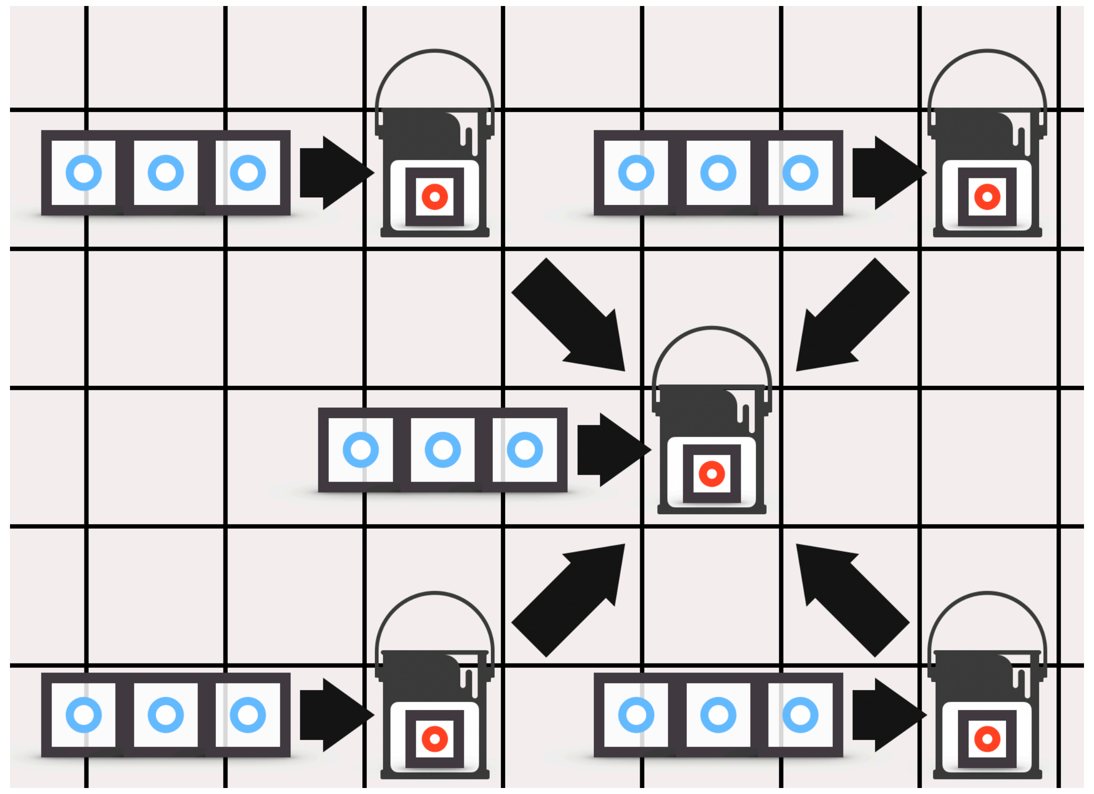
[Ou 2011]



matrix formulation
[Hasan 2007,
Ou 2011,
Huo 2015, ….]




data-driven
[Donikian 2006,
Vevoda 2018,
Wang 2021]

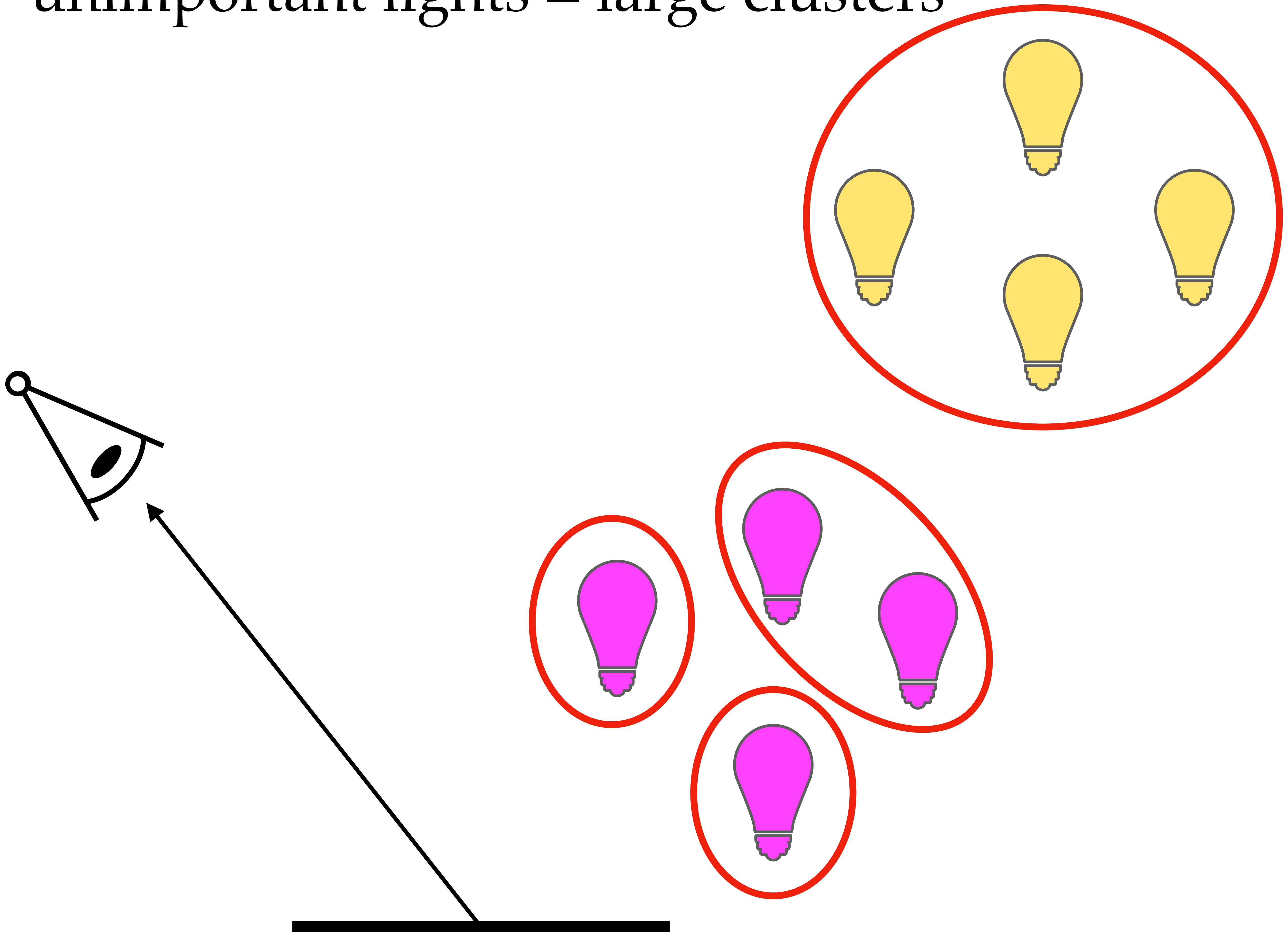


spatial-temporal reuse +
resampling
[Benedikt 2020]

# Idea: group light into clusters

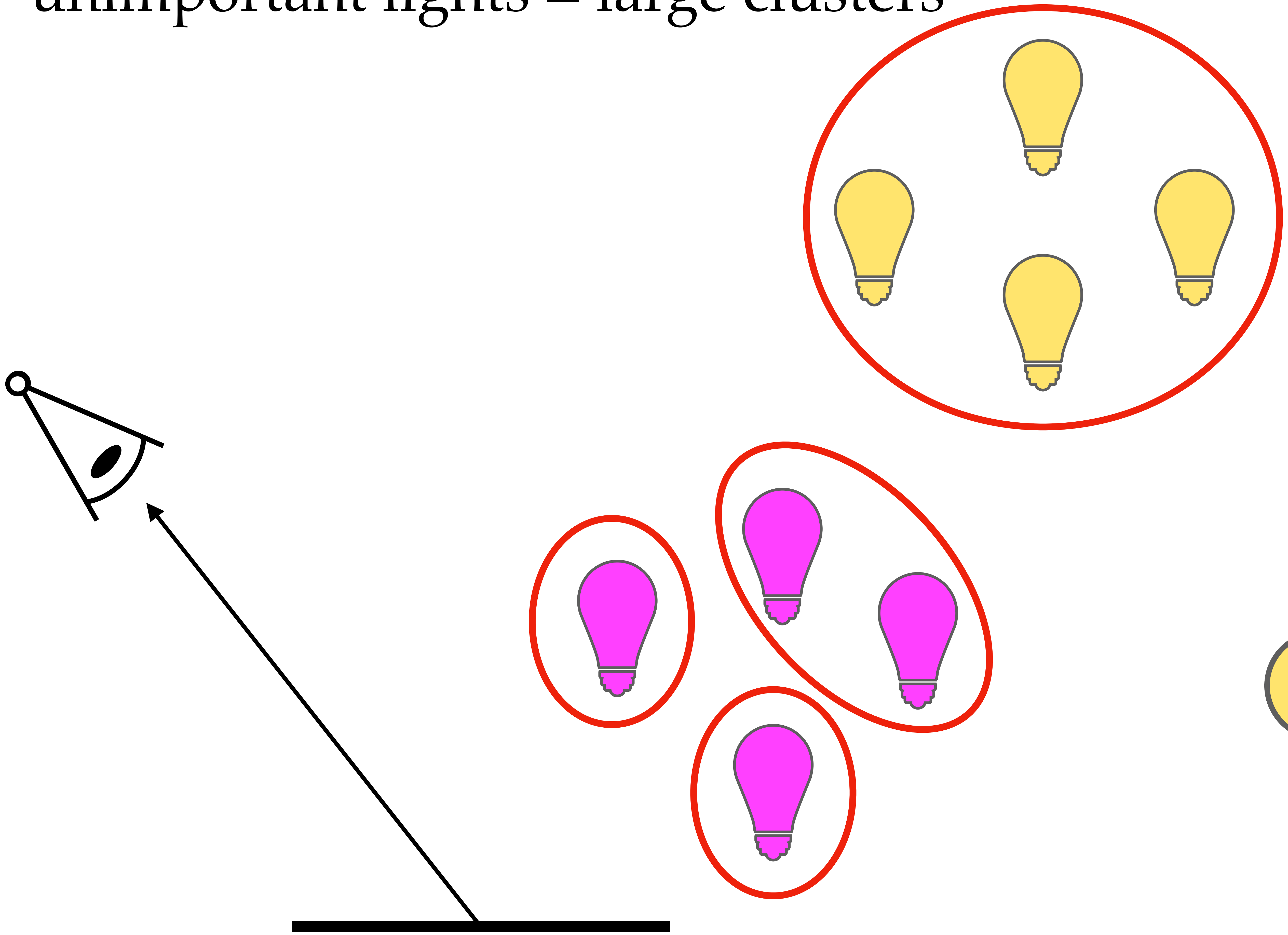
- important lights = small clusters
- unimportant lights = large clusters

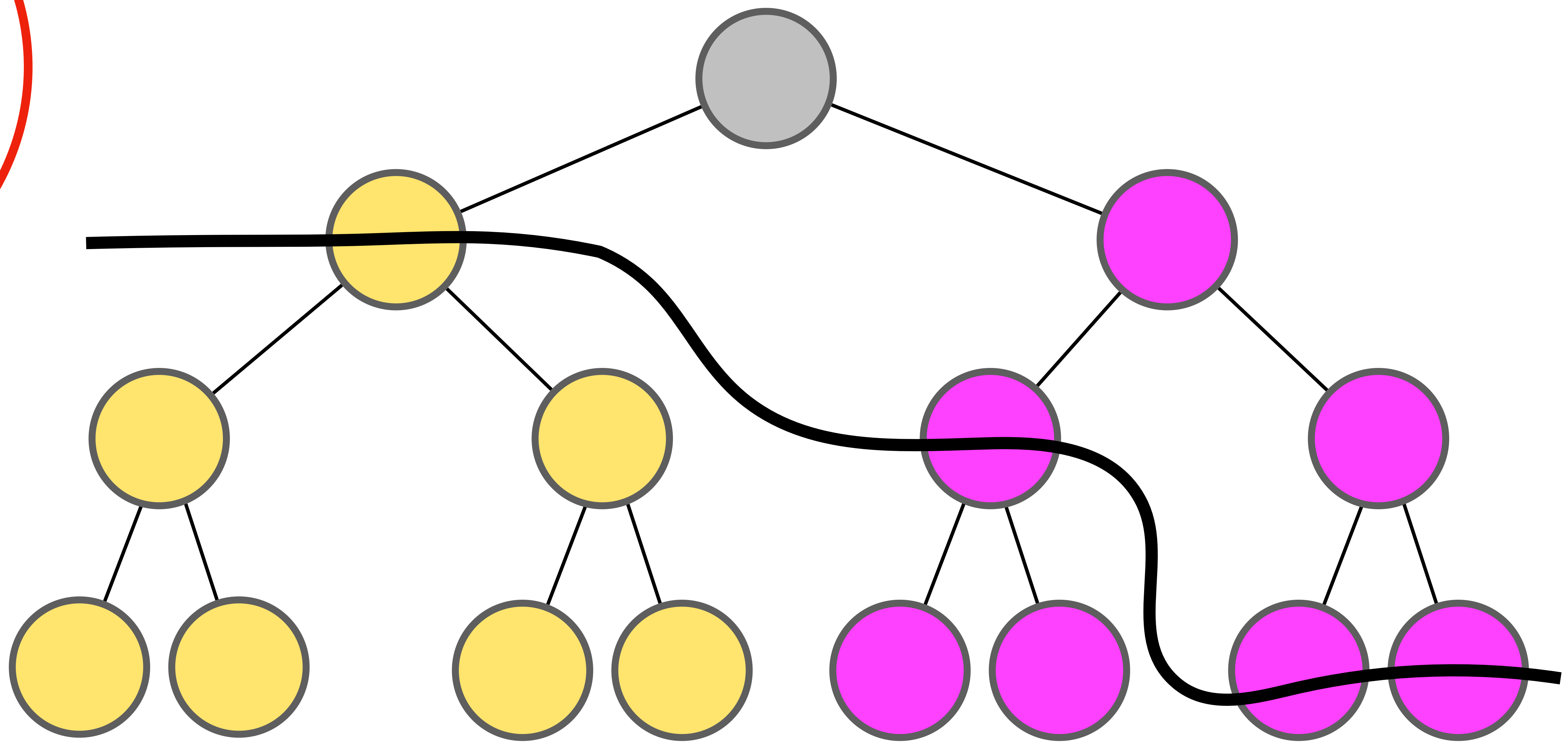
approximate each cluster with one sample



# Use a tree to cluster lights

- important lights = small clusters
- unimportant lights = large clusters





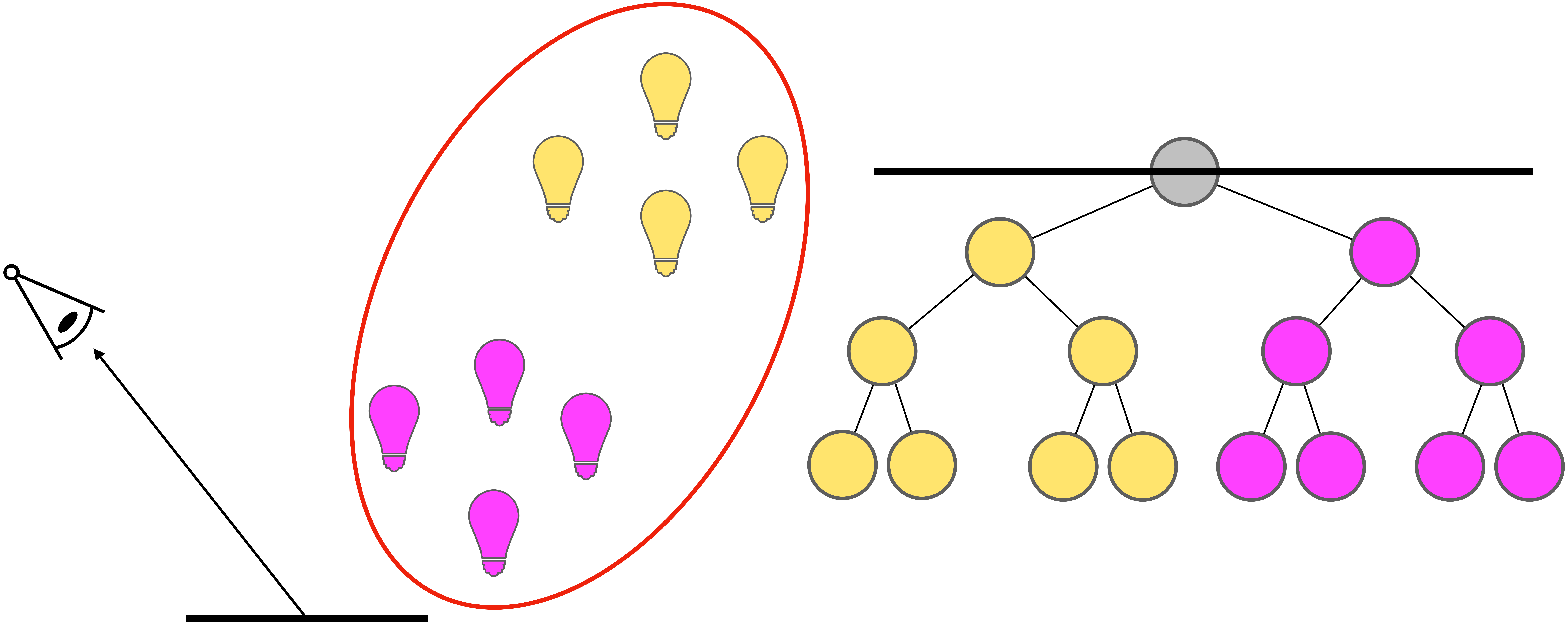
a “lightcut”

**Lightcuts: A Scalable Approach to Illumination**

Bruce Walter    Sebastian Fernandez    Adam Arbree    Kavita Bala    Michael Donikian    Donald P. Greenberg
*Program of Computer Graphics, Cornell University**

# Determining lightcut

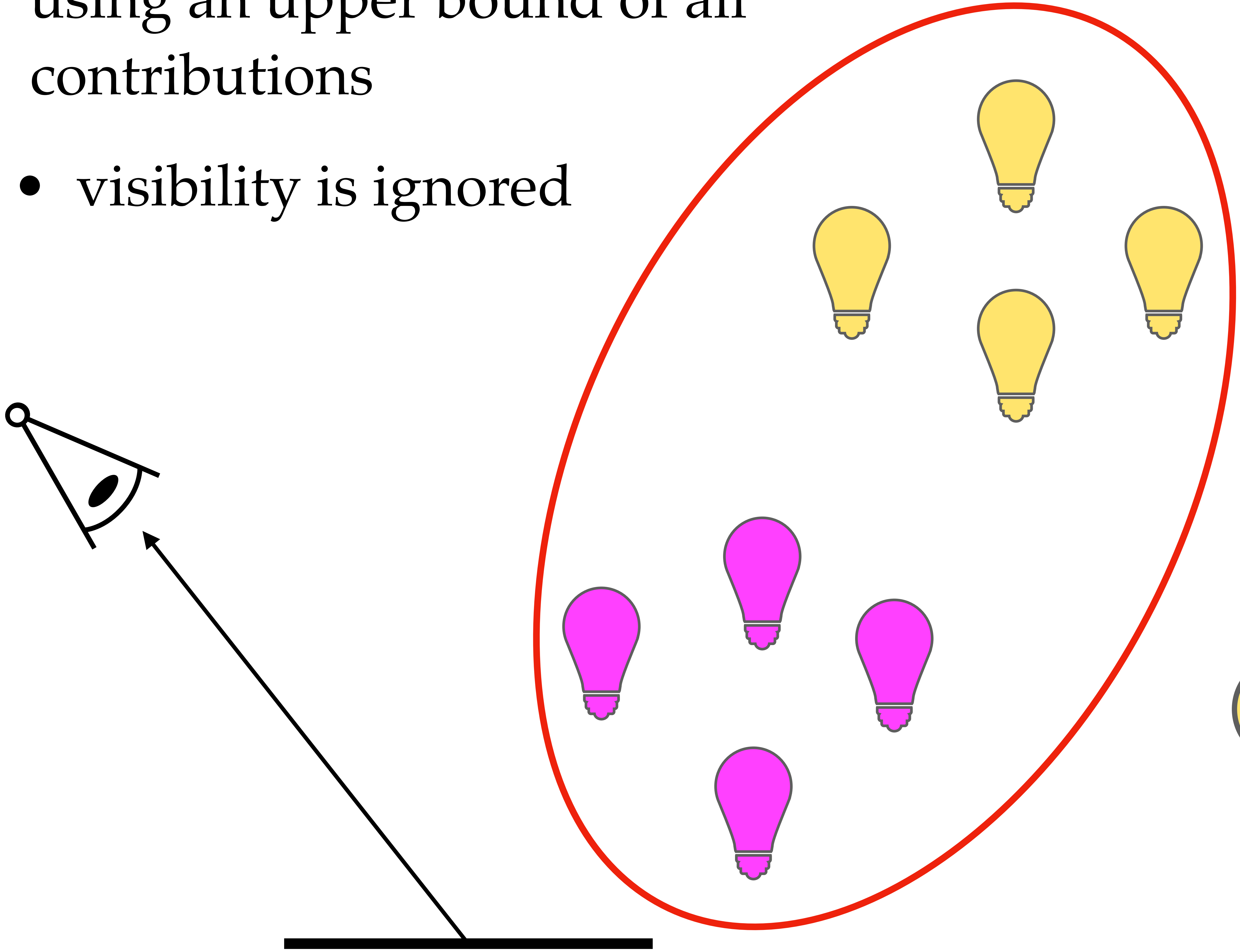
- start at the root

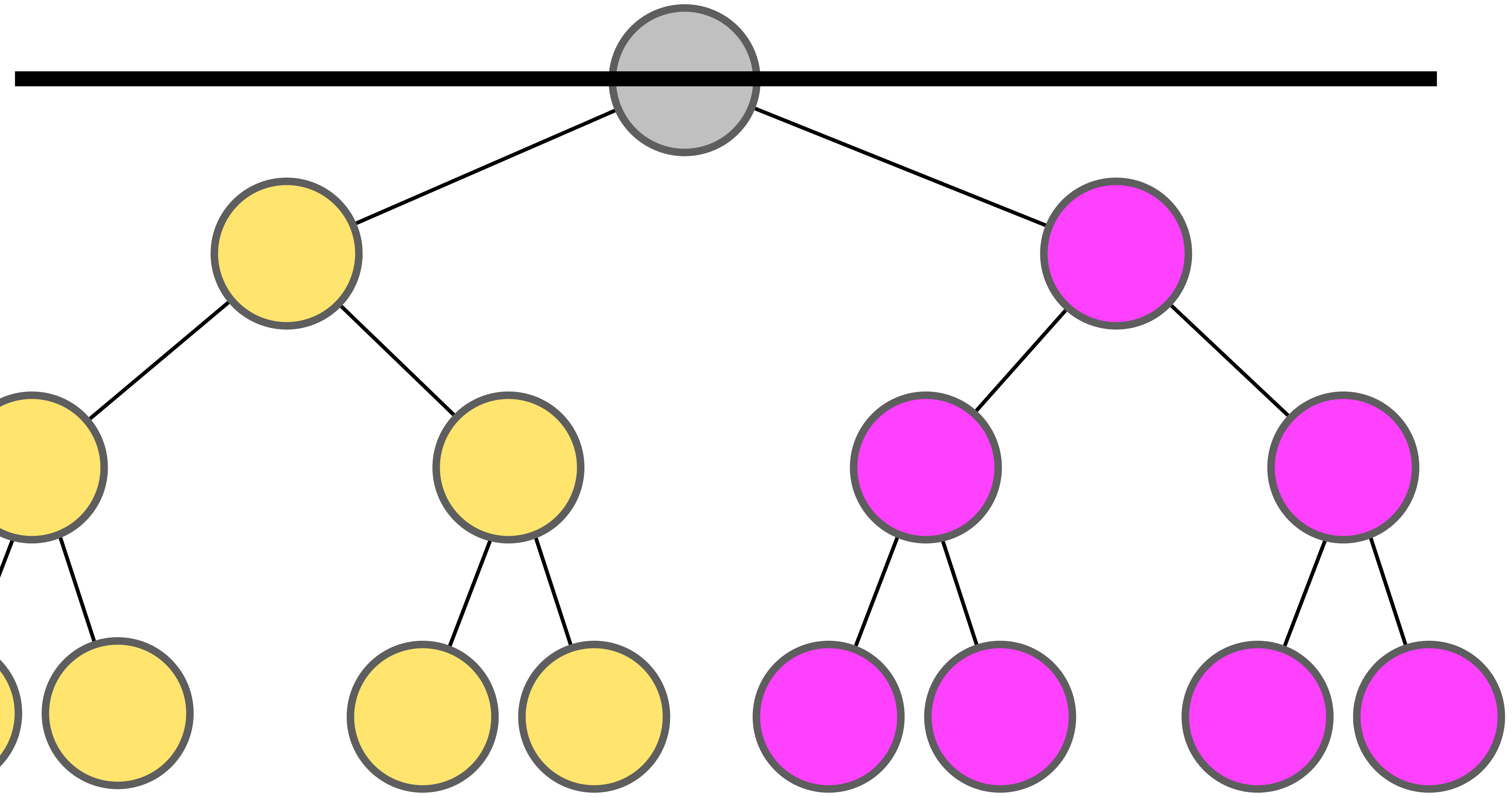


# Determining lightcut

- estimate the importance of the cluster using an upper bound of all contributions
  - visibility is ignored

$$L \cdot \rho \cdot G \cdot V$$

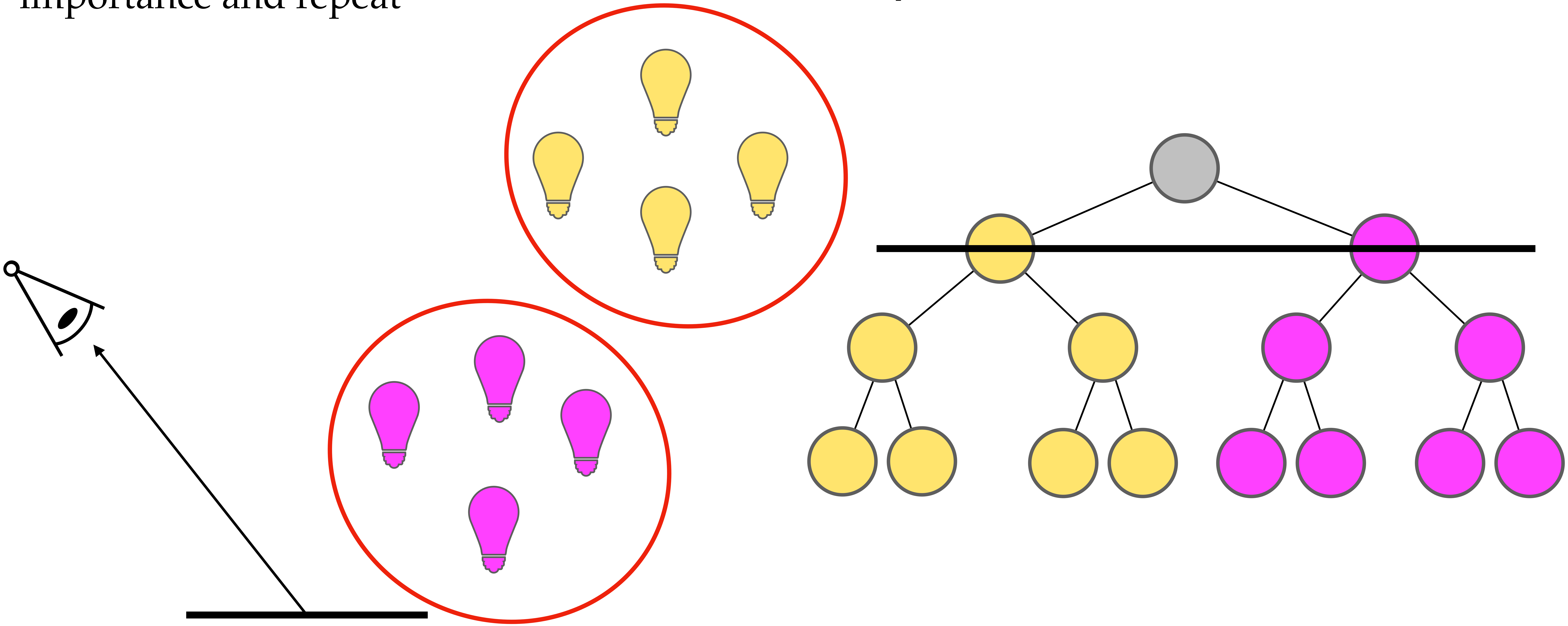




# Determining lightcut

- refine the node with the highest importance and repeat

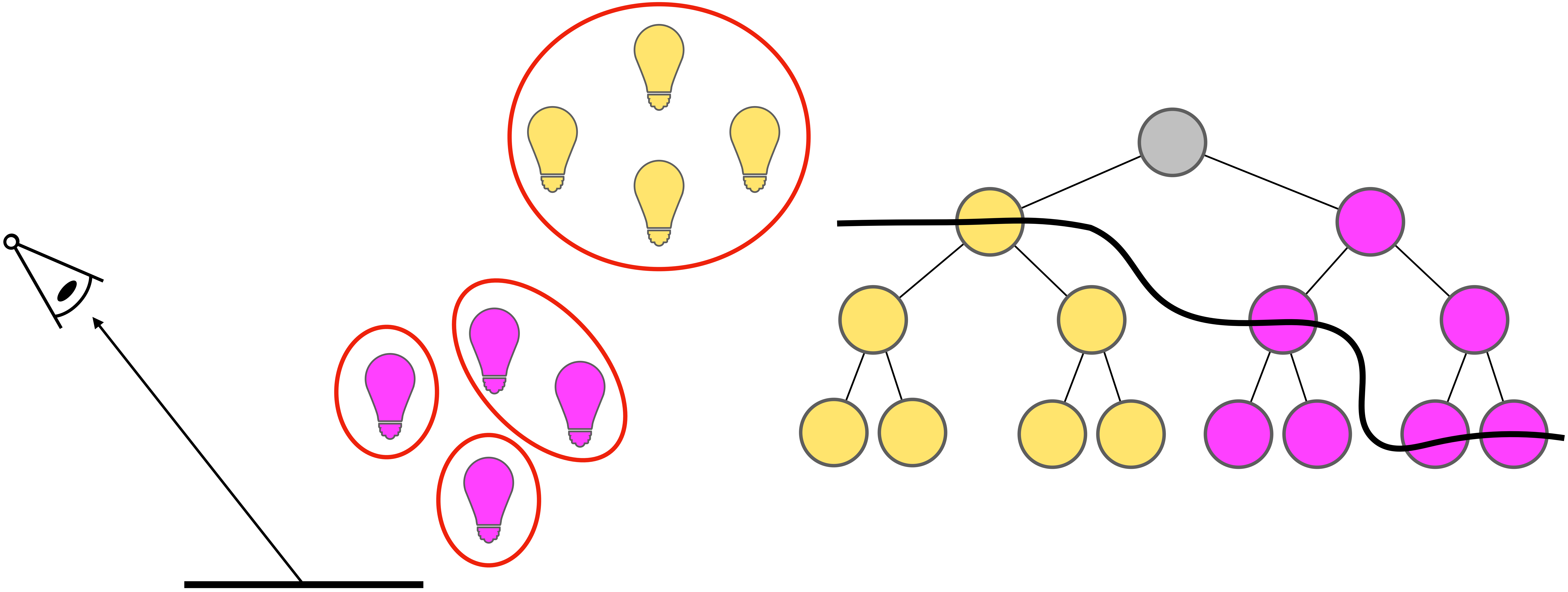
$$L \cdot \rho \cdot G \cdot V$$



# Determining lightcut

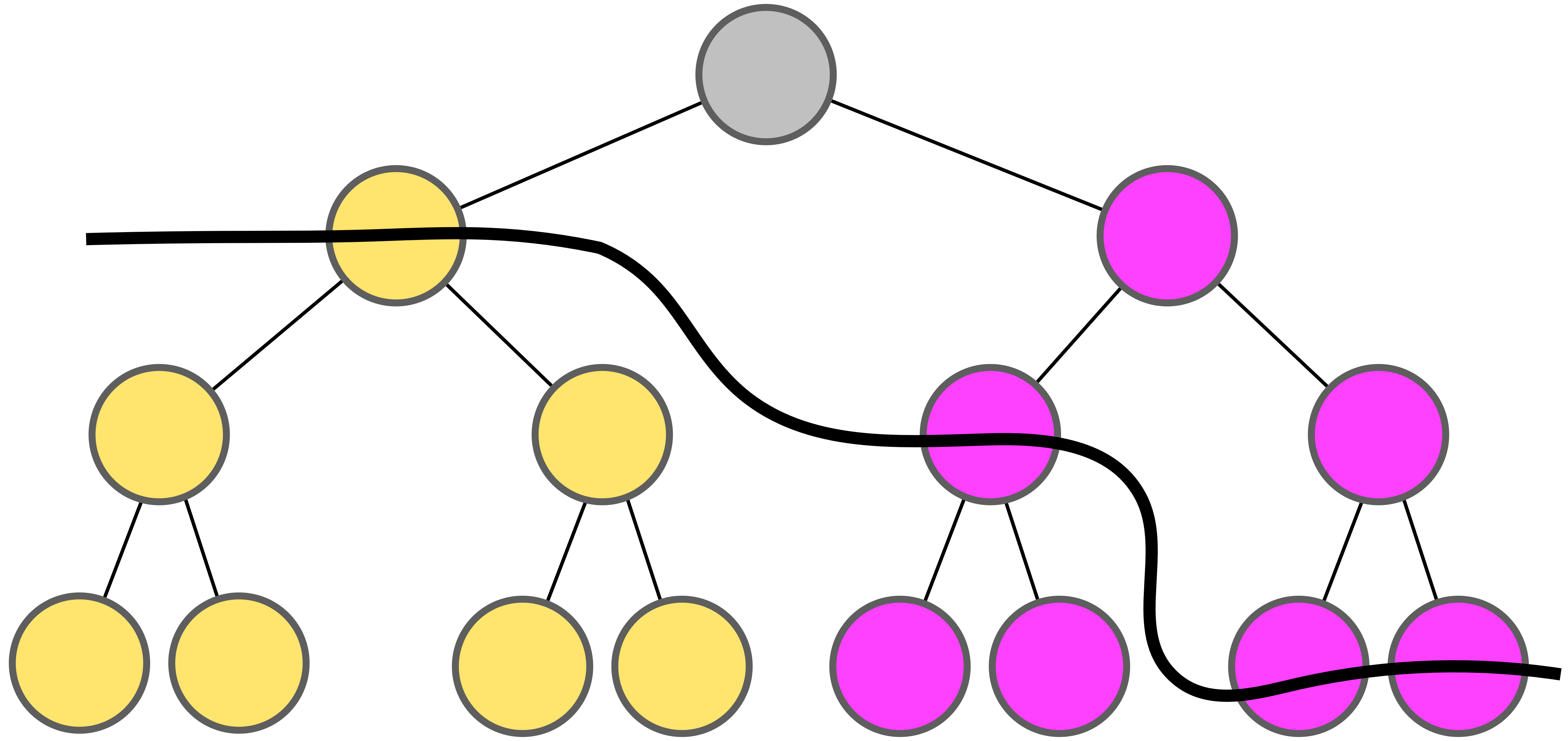
- refine the node with the highest importance and repeat

$$L \cdot \rho \cdot G \cdot V$$



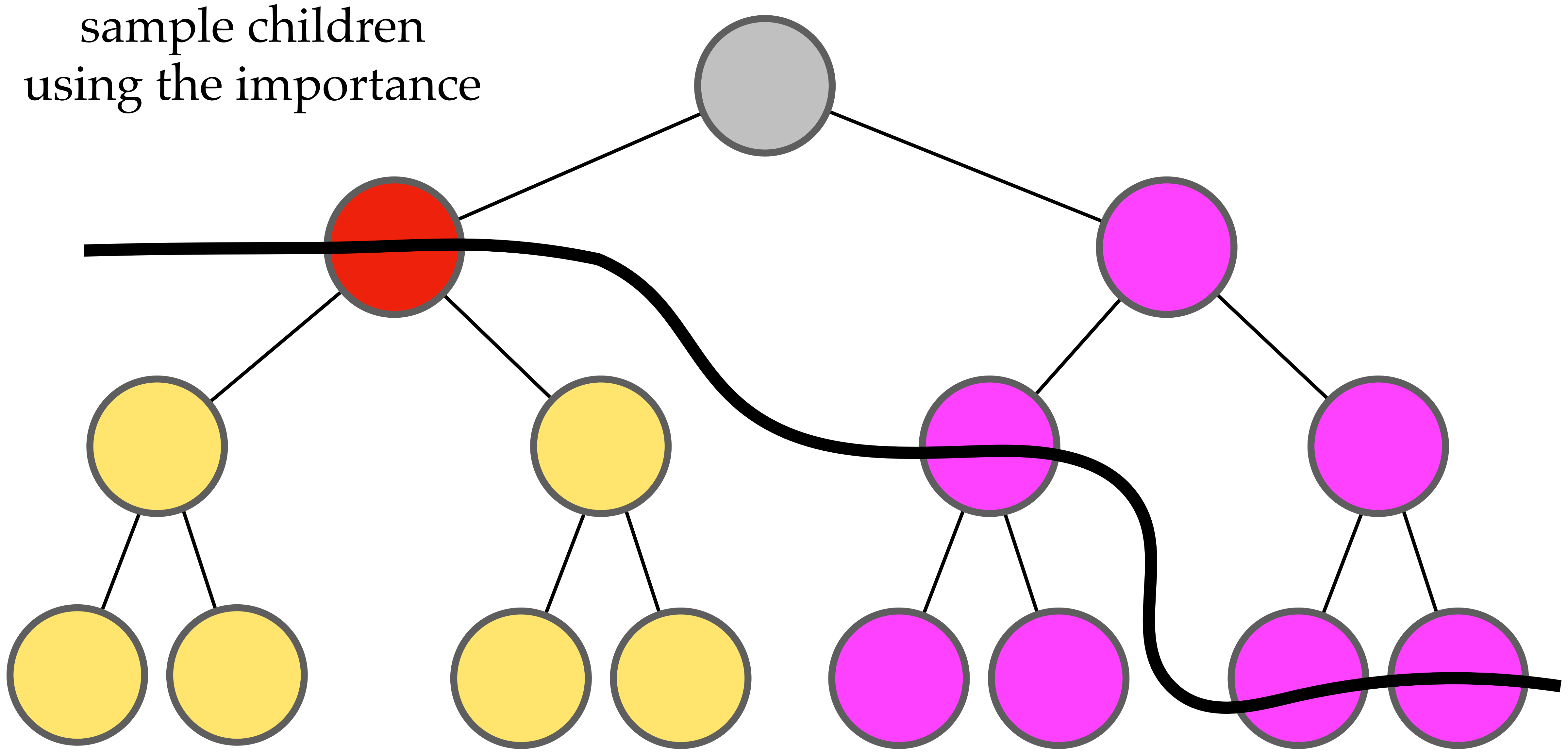
# Sampling light from the lightcut

- uniformly pick a cluster
- sample a light from the cluster by traversing the tree



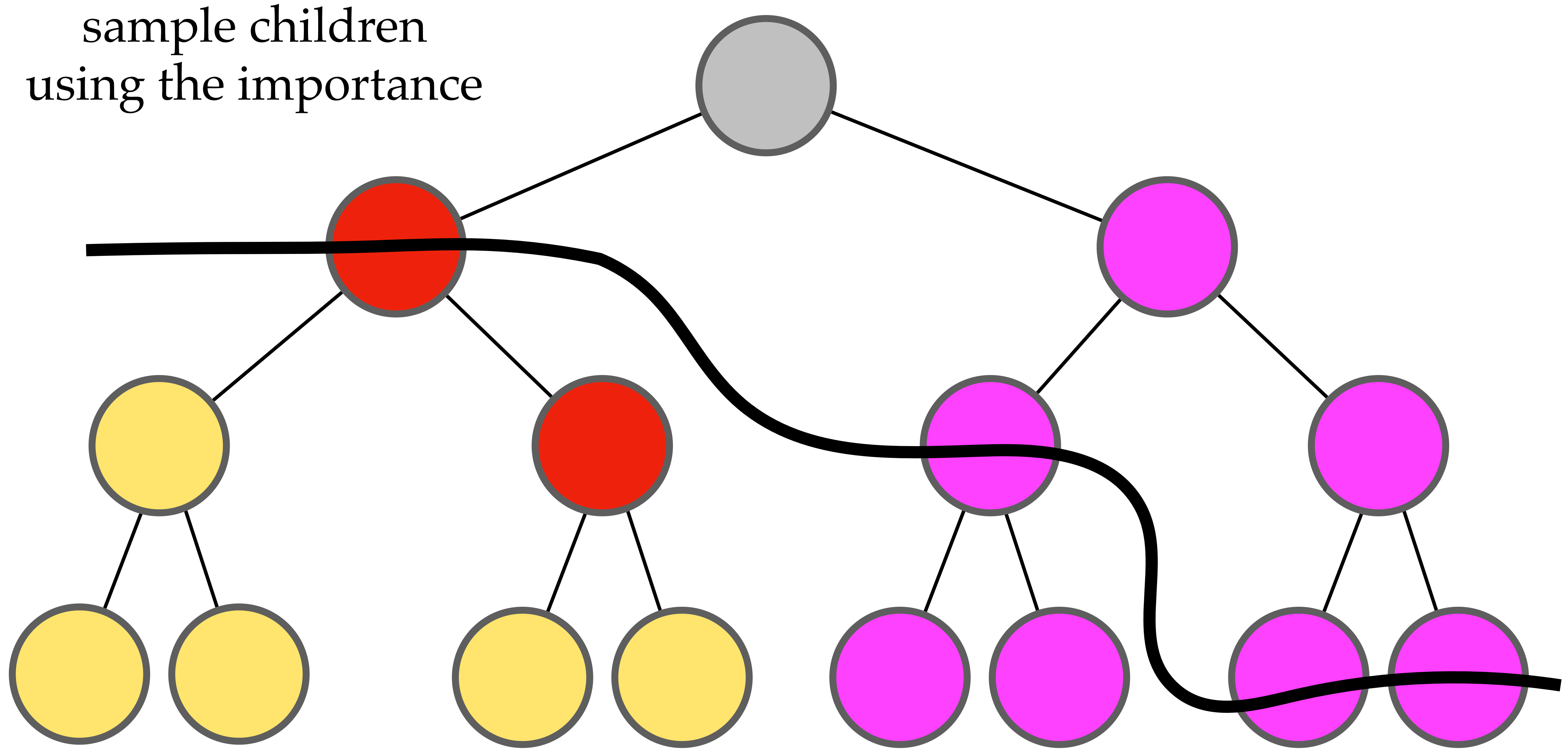
# Sampling light from the lightcut

- uniformly pick a cluster
- sample a light from the cluster
  by traversing the tree



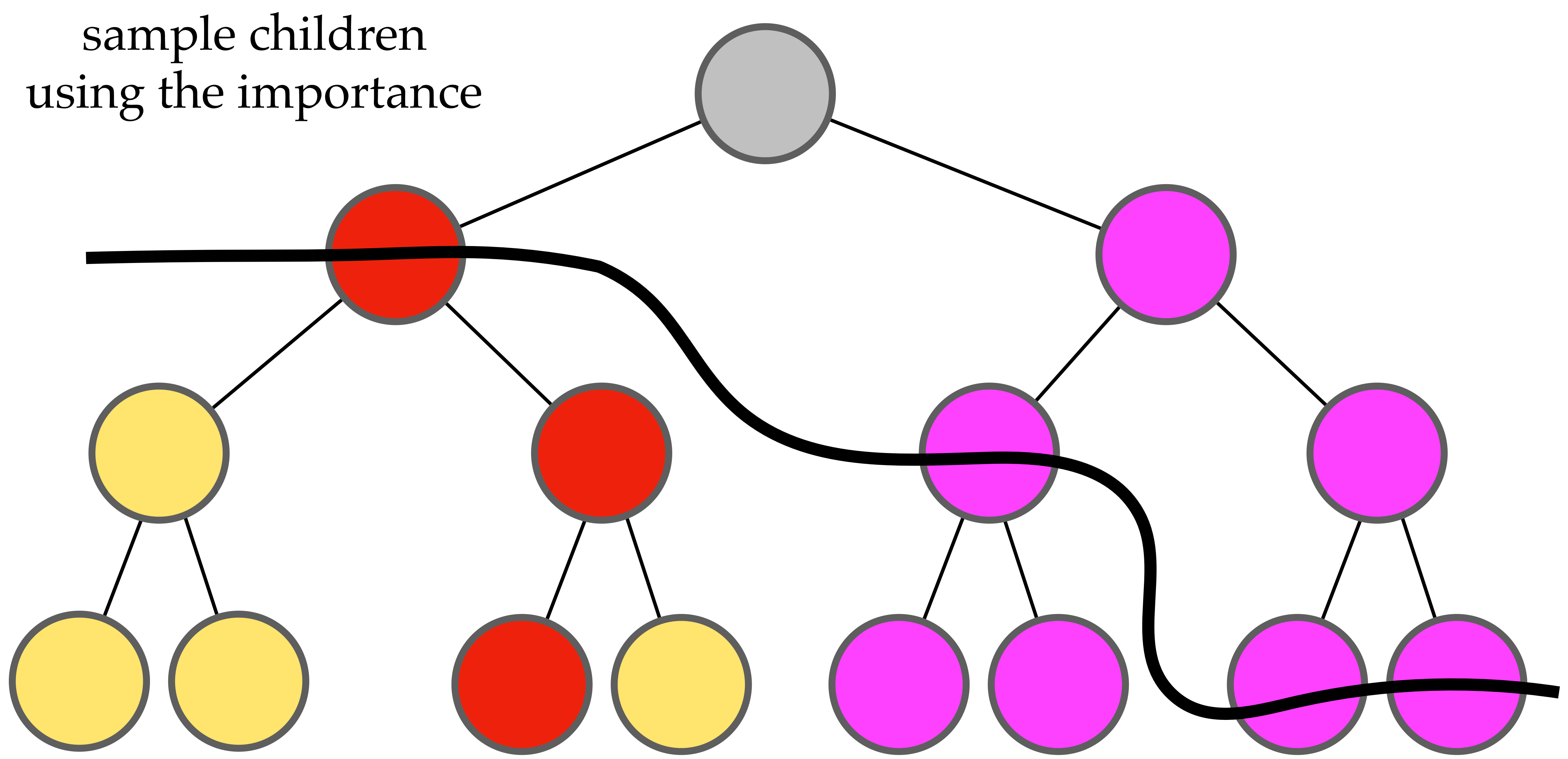
# Sampling light from the lightcut

- uniformly pick a cluster
- sample a light from the cluster by traversing the tree



# Sampling light from the lightcut

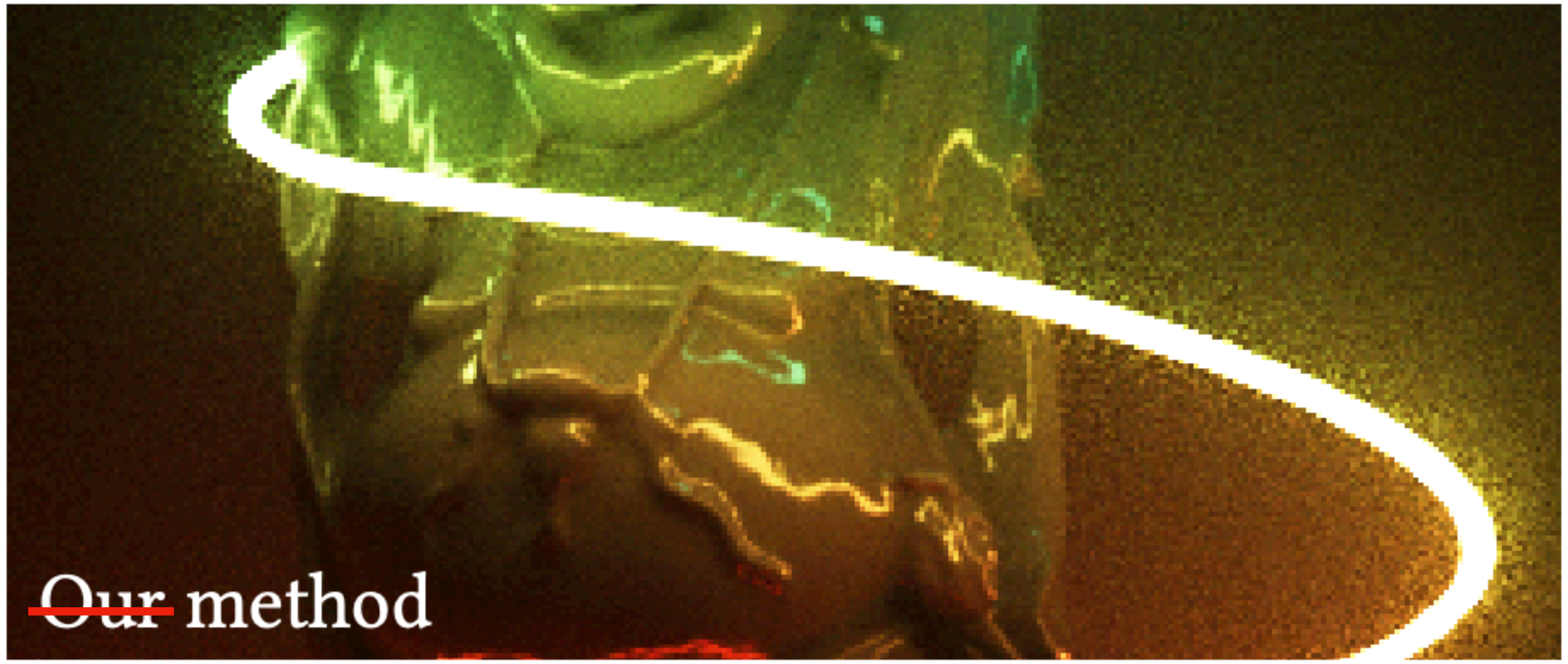
- uniformly pick a cluster
- sample a light from the cluster by traversing the tree

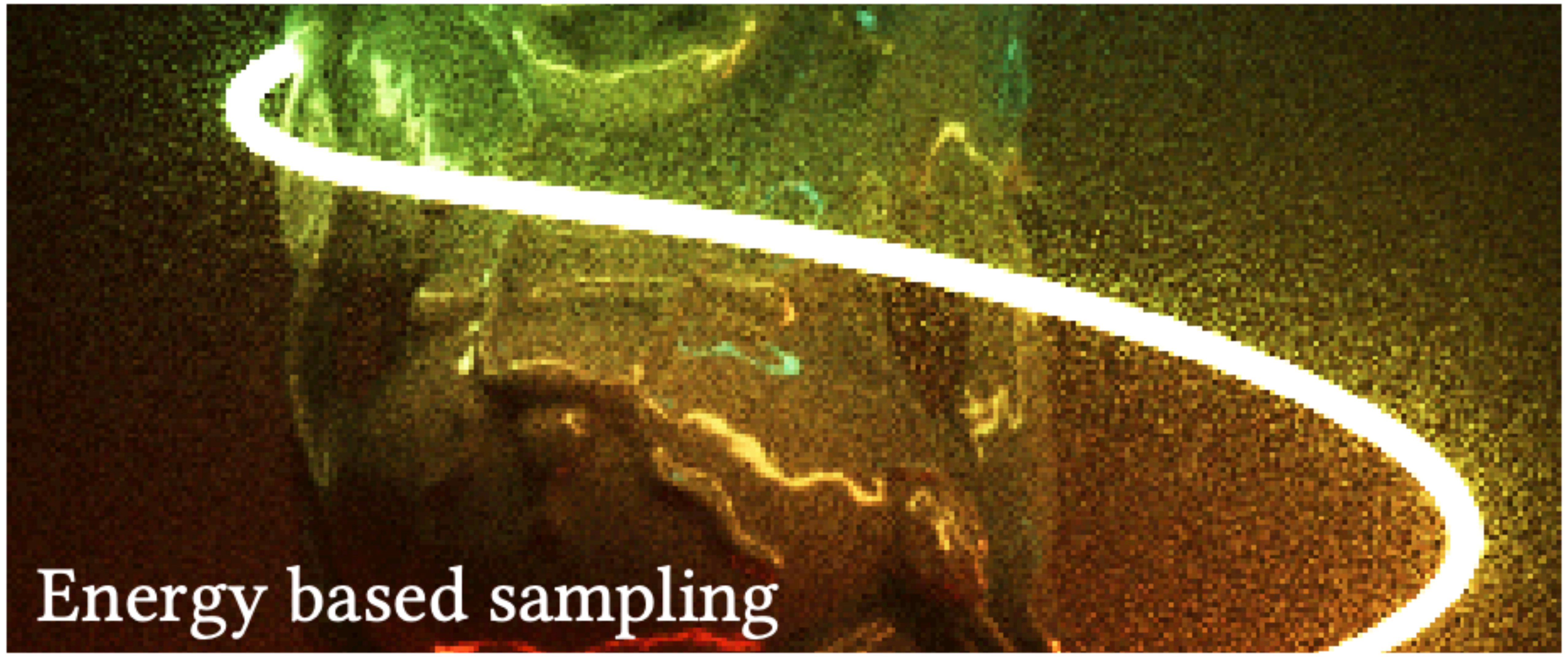



Stochastic Lightcuts

Cem Yuksel
University of Utah, UT, USA

# Lightcuts-like ideas are widely used in practice



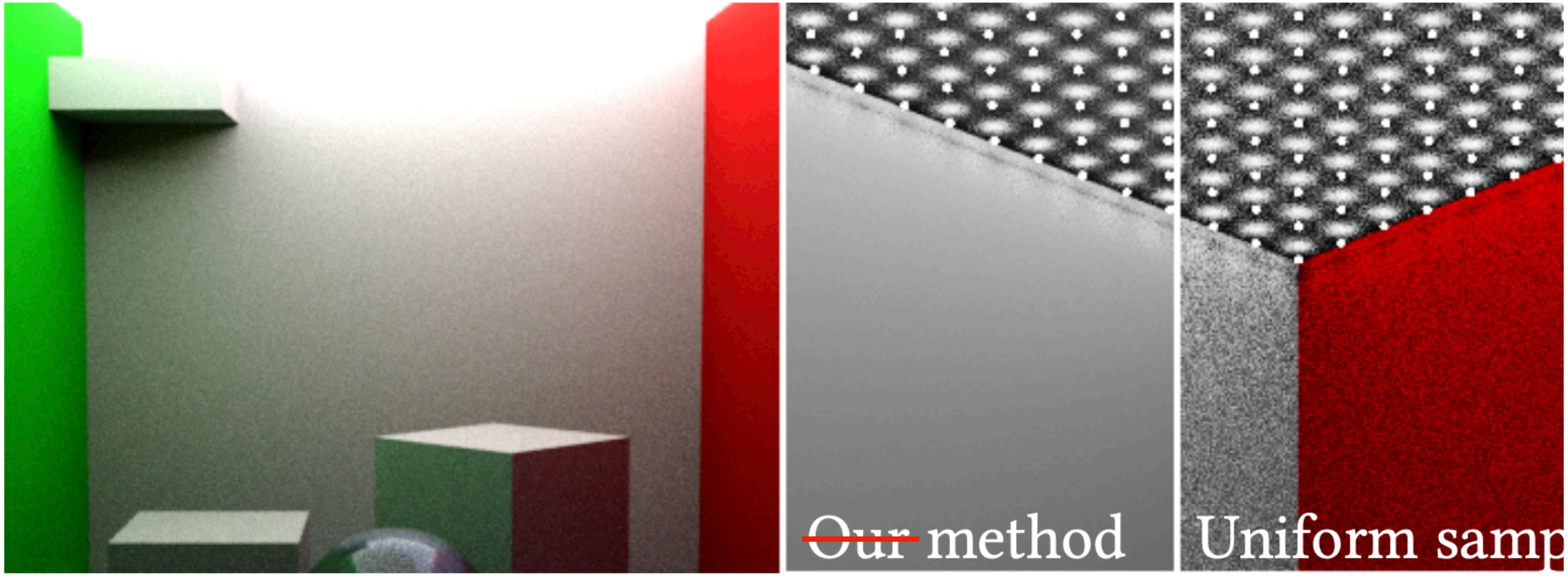




Estevez's

**Importance Sampling of Many Lights with Adaptive Tree Splitting**

ALEJANDRO CONTY ESTEVEZ, Sony Pictures Imageworks
CHRISTOPHER KULLA, Sony Pictures Imageworks

# Lightcuts-like ideas are widely used in practice

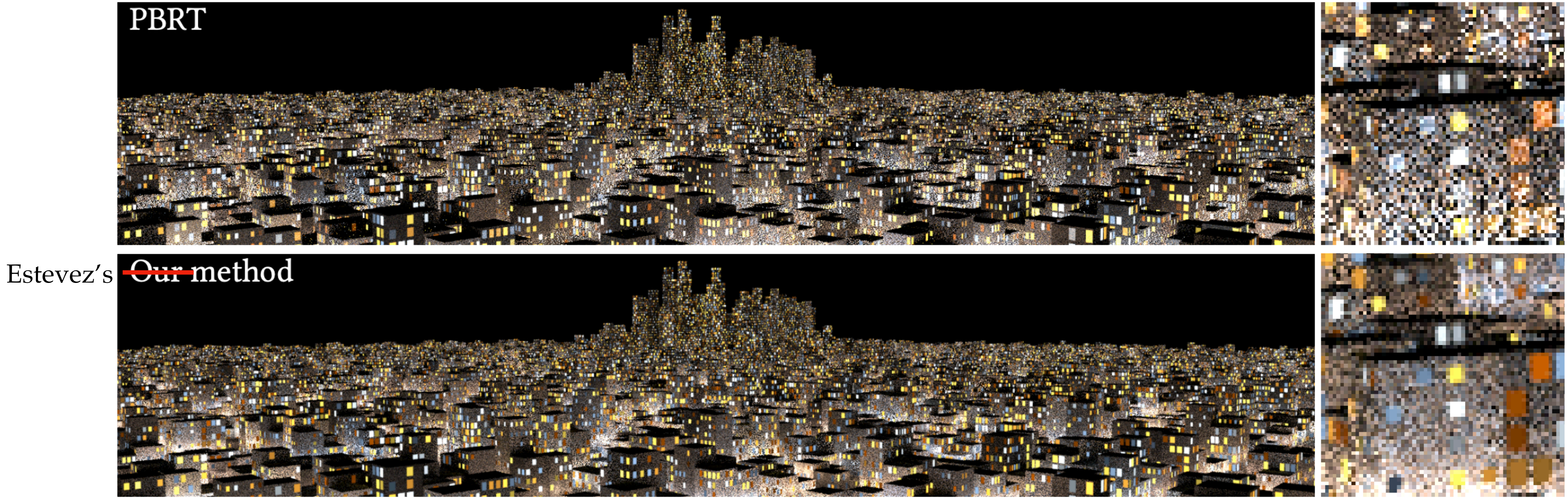



Estevez's

**Importance Sampling of Many Lights with Adaptive Tree Splitting**

ALEJANDRO CONTY ESTEVEZ, Sony Pictures Imageworks
CHRISTOPHER KULLA, Sony Pictures Imageworks

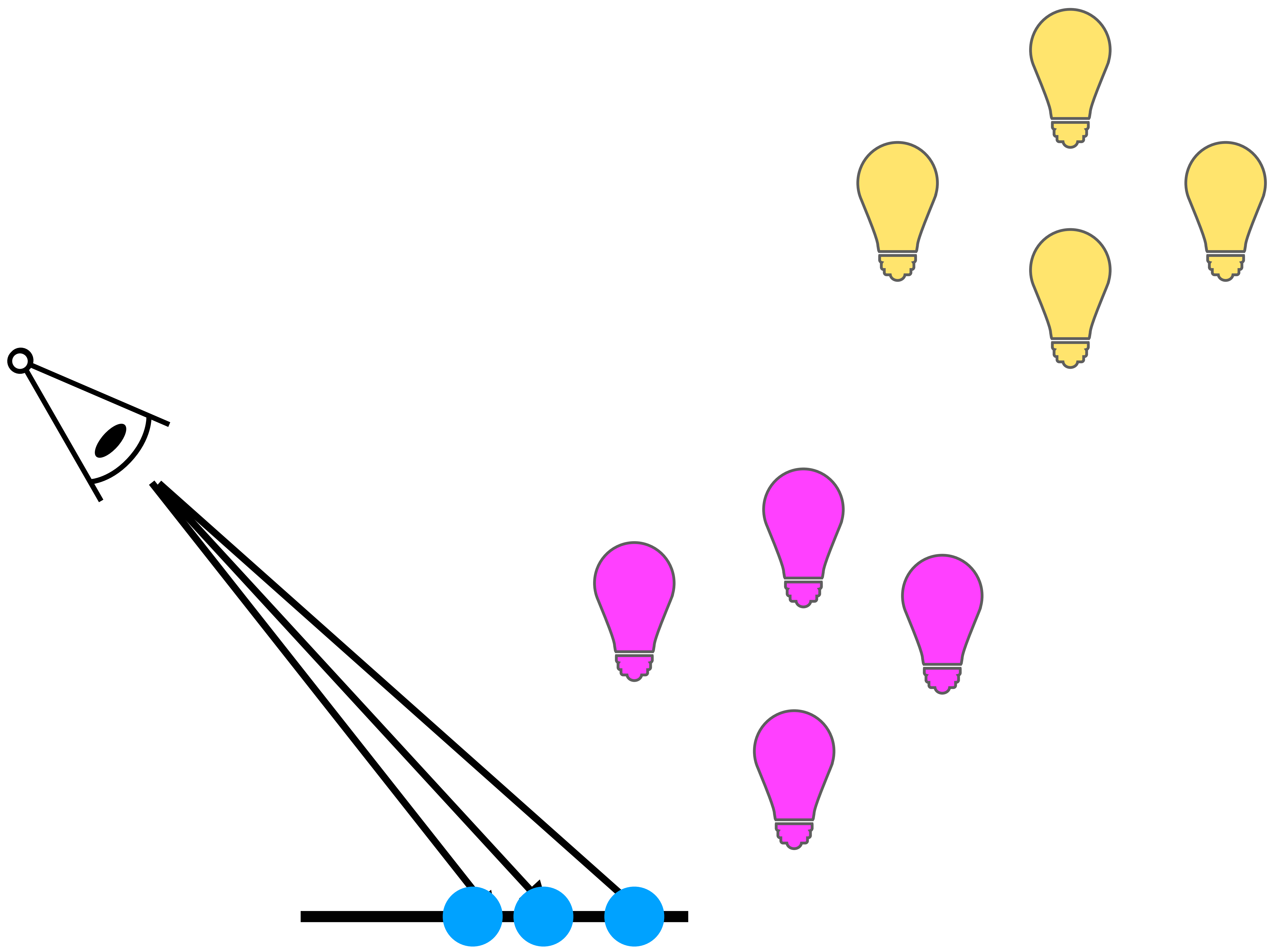
# Lightcuts-like ideas are widely used in practice




**Importance Sampling of Many Lights with Adaptive Tree Splitting**

ALEJANDRO CONTY ESTEVEZ, Sony Pictures Imageworks
CHRISTOPHER KULLA, Sony Pictures Imageworks

# Multi-dimensional lightcuts: also cluster the shading points



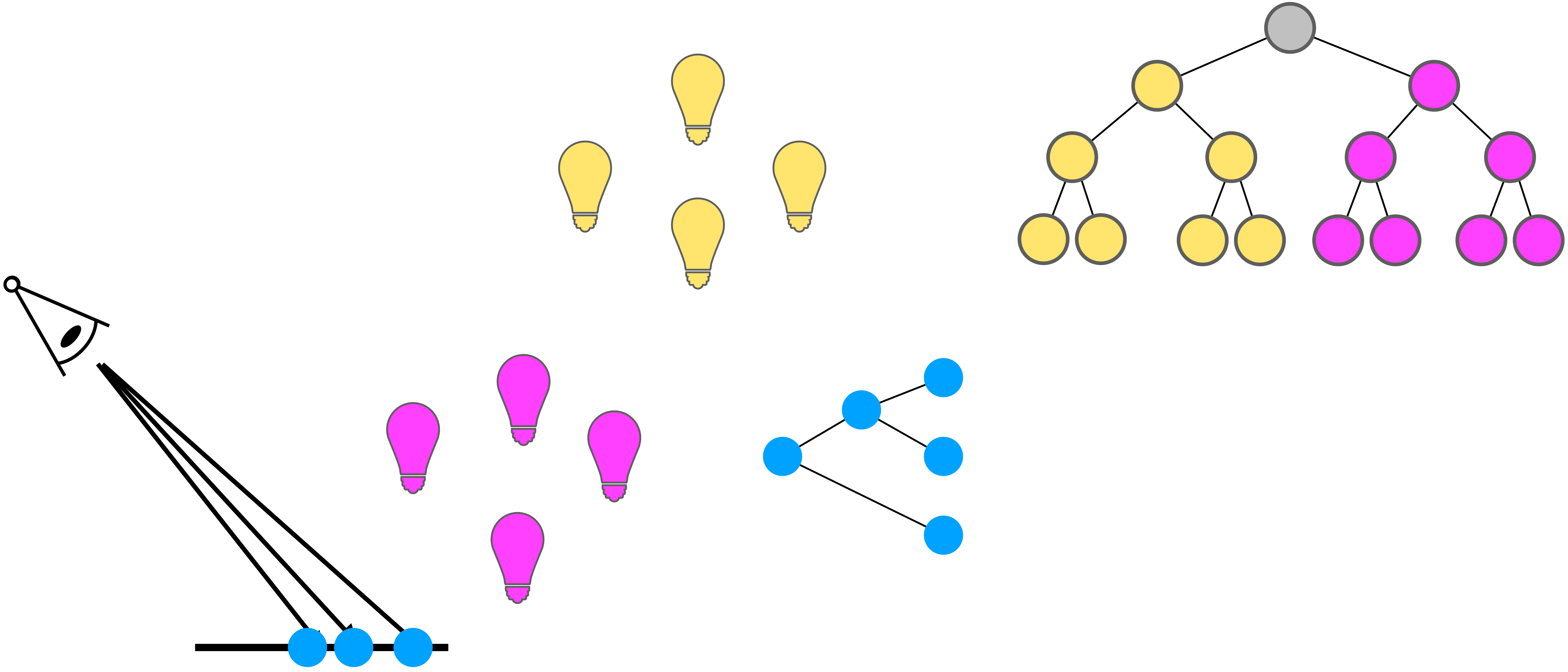
**Multidimensional Lightcuts**

Bruce Walter Adam Arbree Kavita Bala Donald P. Greenberg
Cornell University*

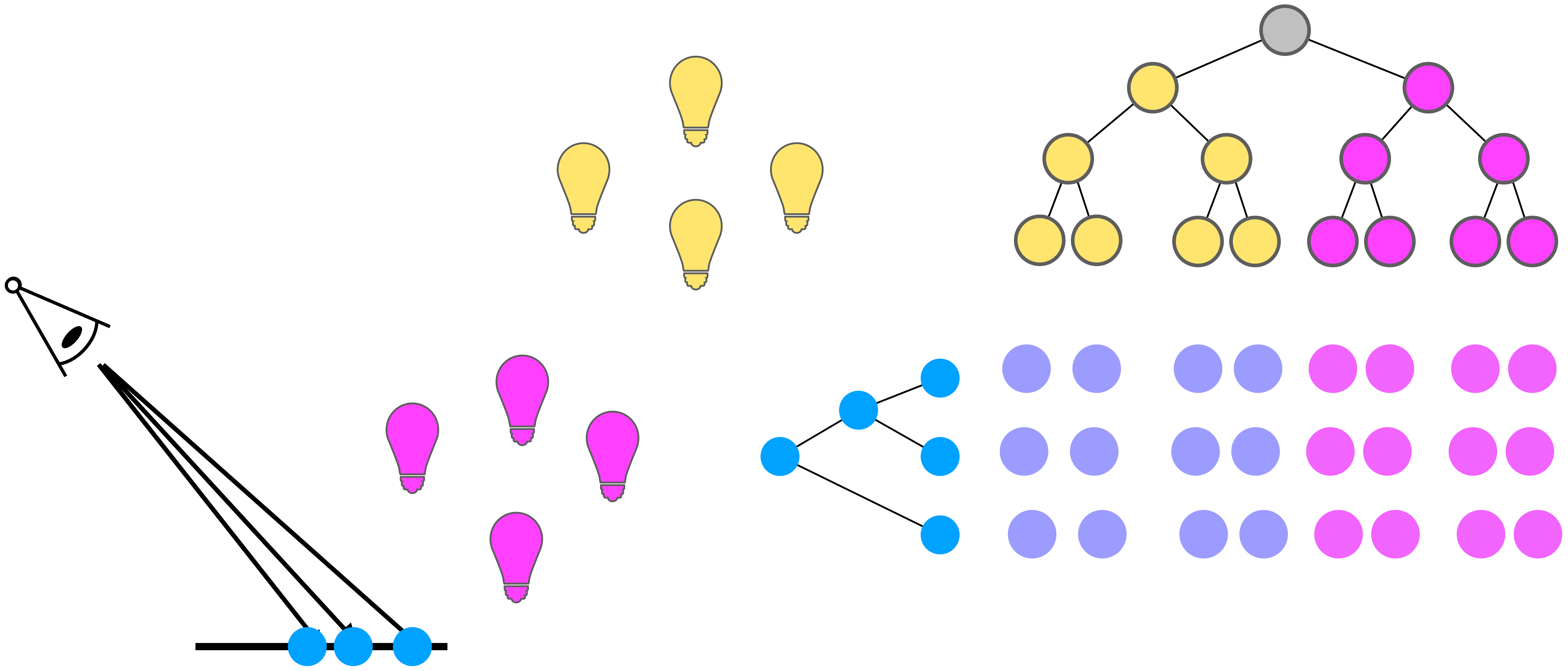
**Bidirectional Lightcuts**

Bruce Walter Pramook Khungurn Kavita Bala
Cornell University*

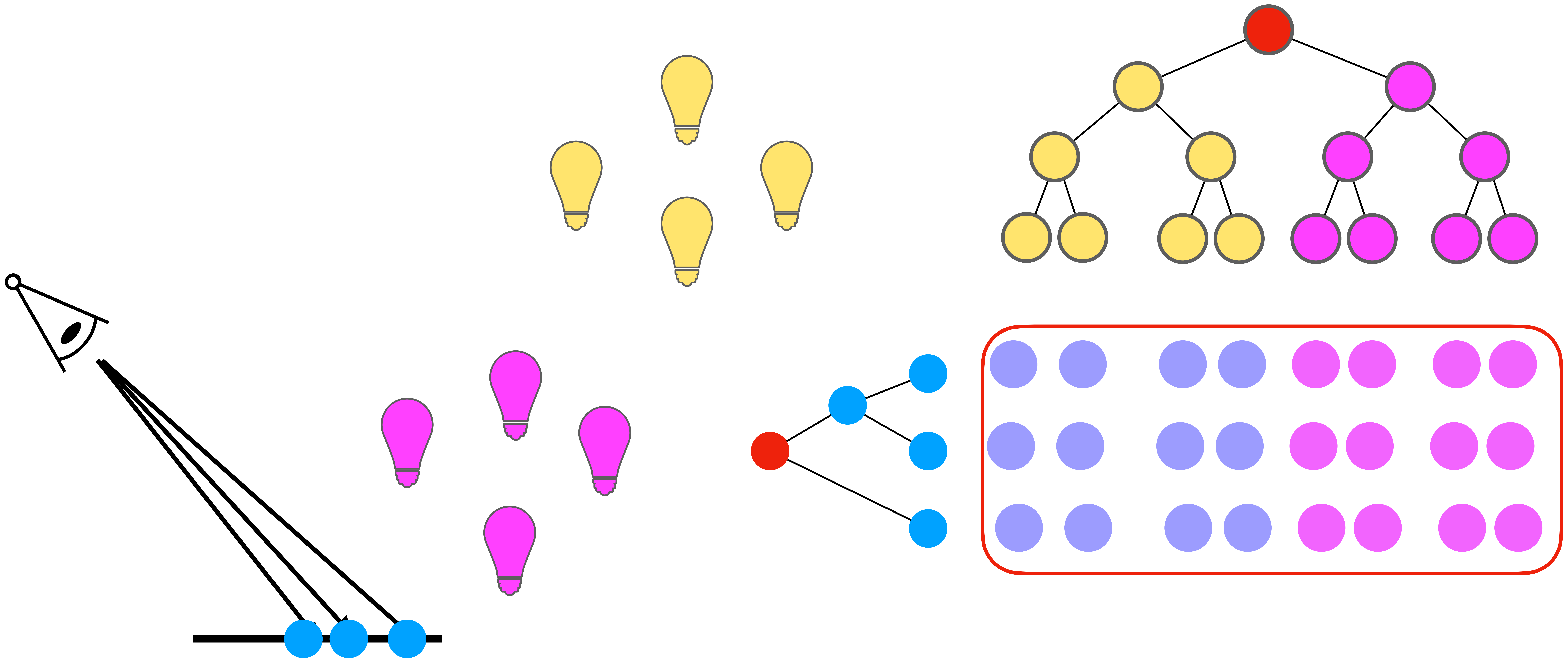
# Multi-dimensional lightcuts: also cluster the shading points



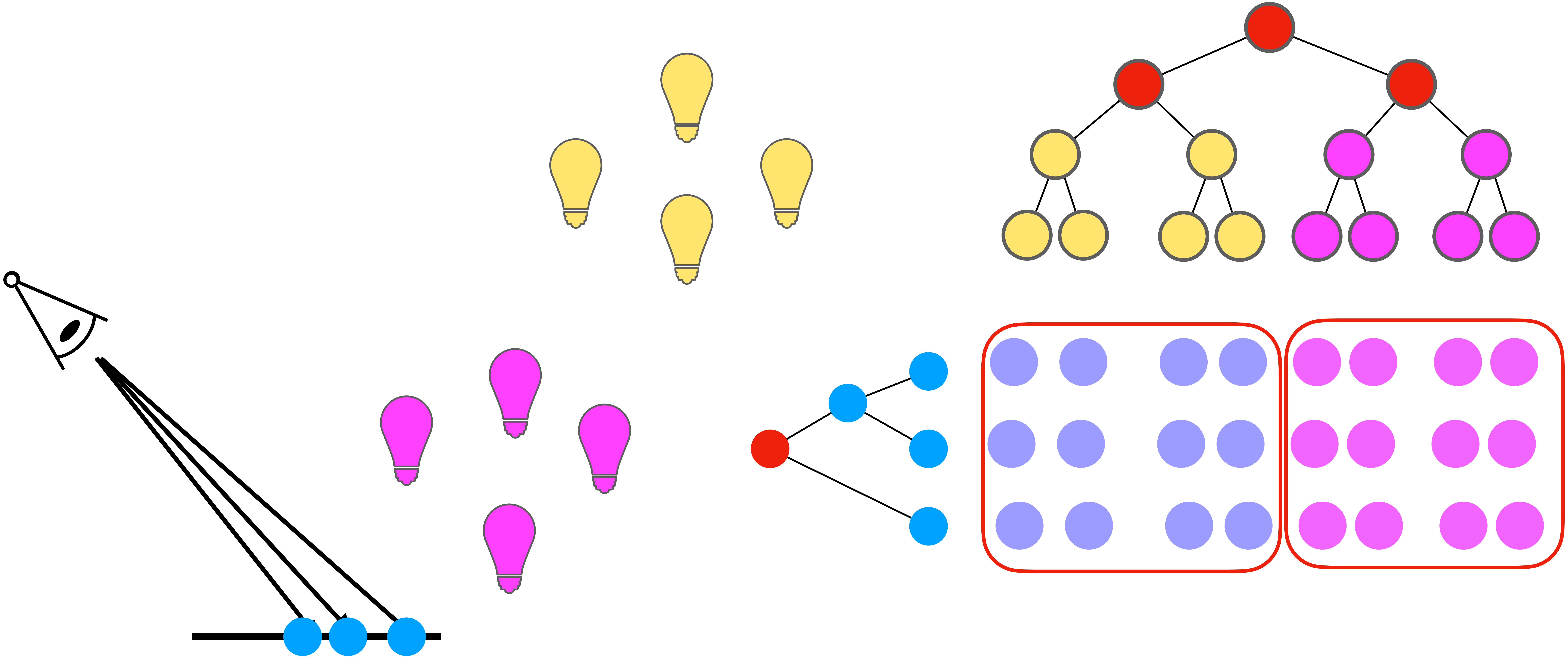
# Multi-dimensional lightcuts: also cluster the shading points



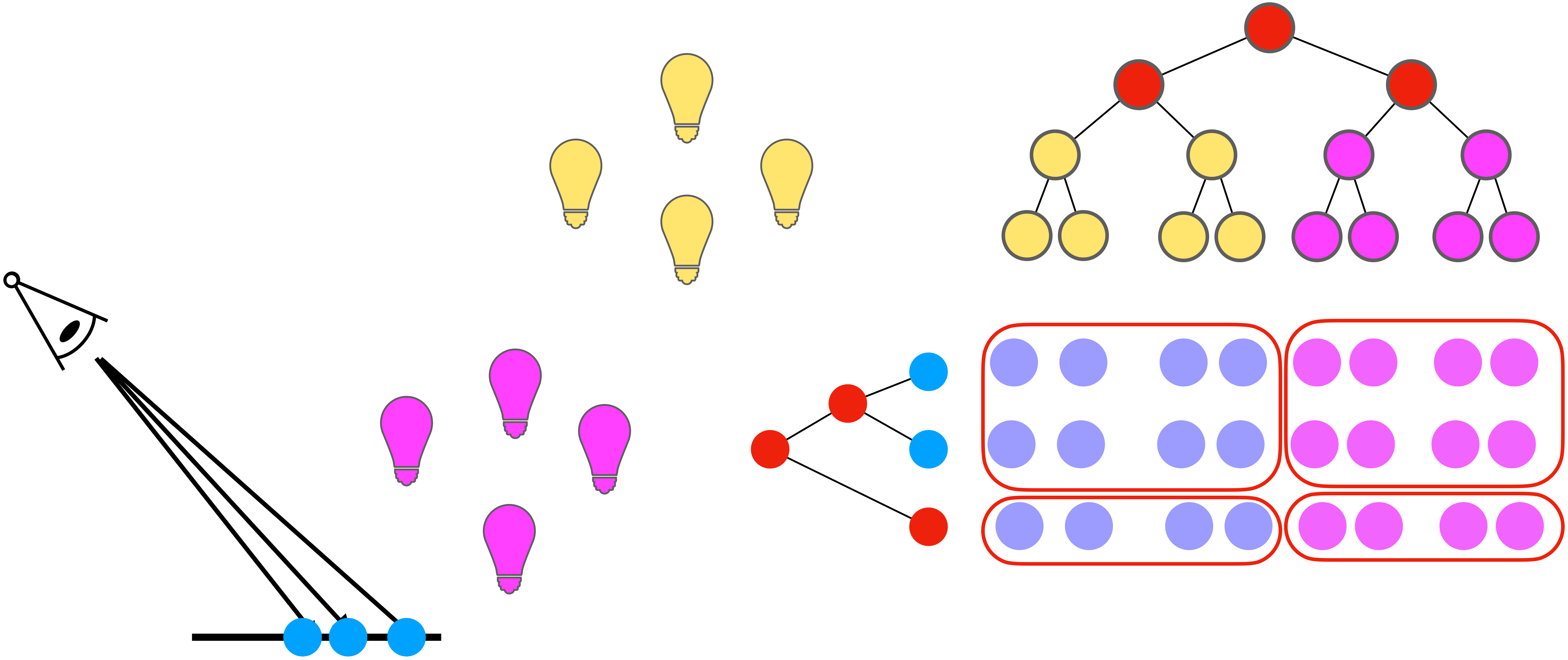
# Multi-dimensional lightcuts: also cluster the shading points



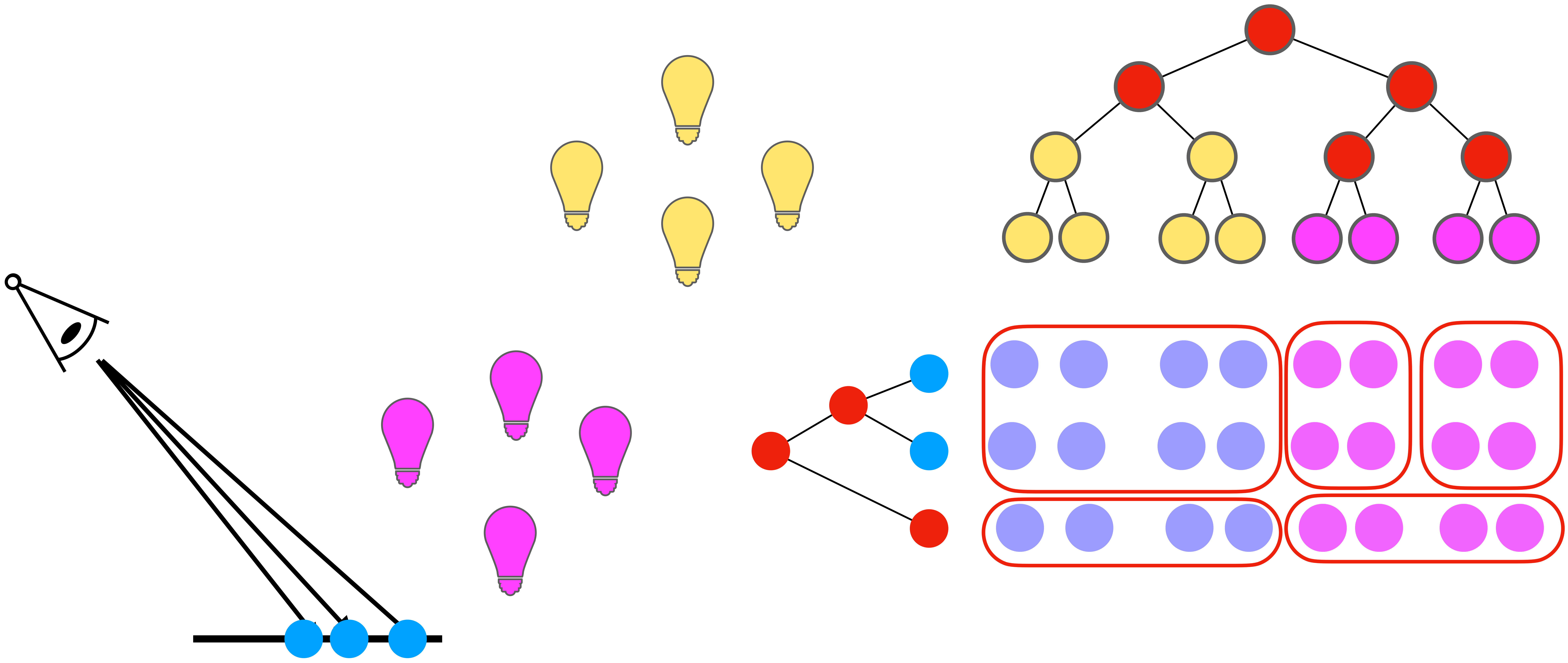
# Multi-dimensional lightcuts: also cluster the shading points



# Multi-dimensional lightcuts: also cluster the shading points



# Multi-dimensional lightcuts: also cluster the shading points



# Multidimensional lightcuts is (was?) used by Autodesk

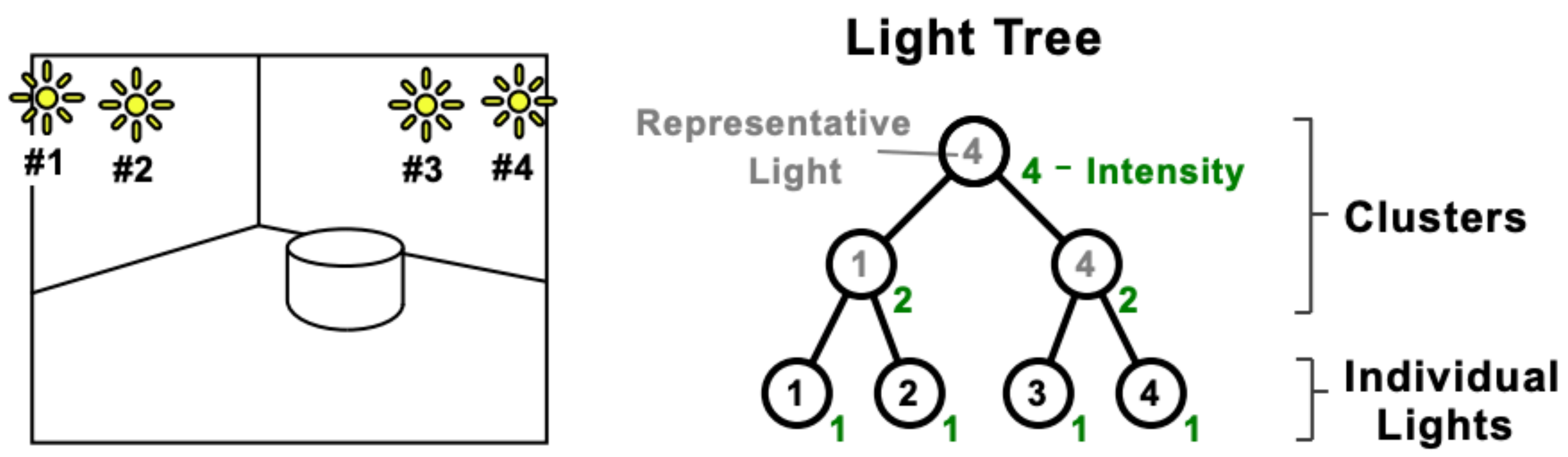
with lightcuts

without lightcuts


https://cgg.mff.cuni.cz/~jaroslav/papers/mlcourse2012/mlcourse2012%20-%2006%20-%20arbree.pdf

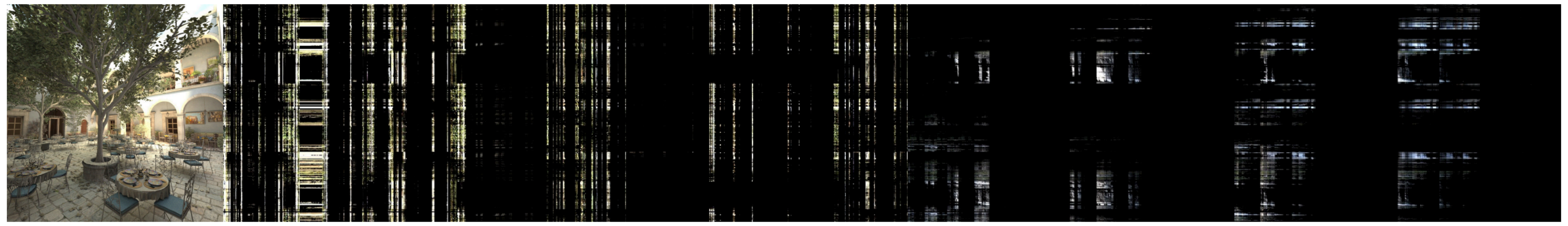
# Ideas

[Walter 2005]

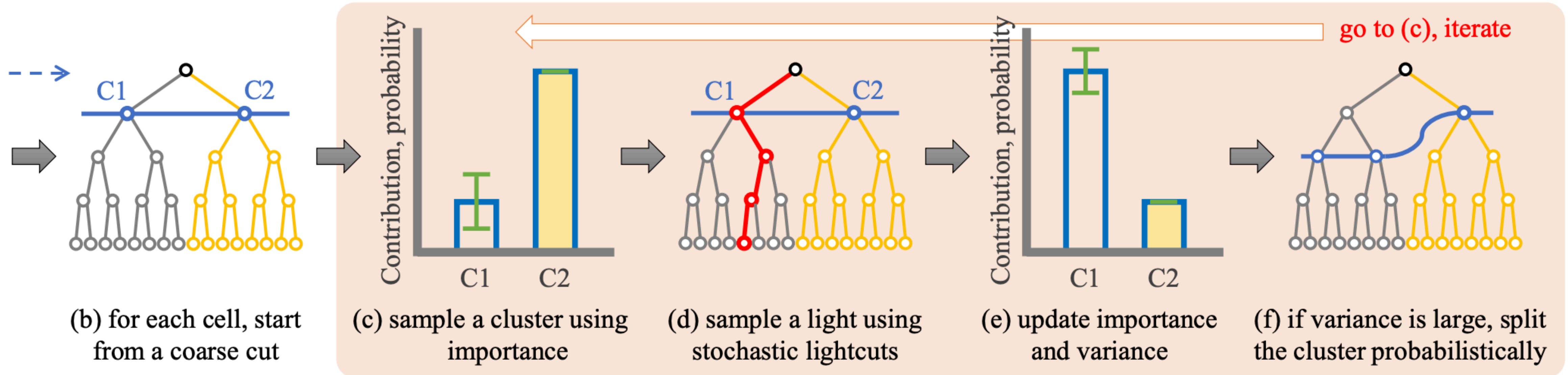



hierarchical clustering
[Shirley 1996,
Paquette 1998,
Walter 2005, …]

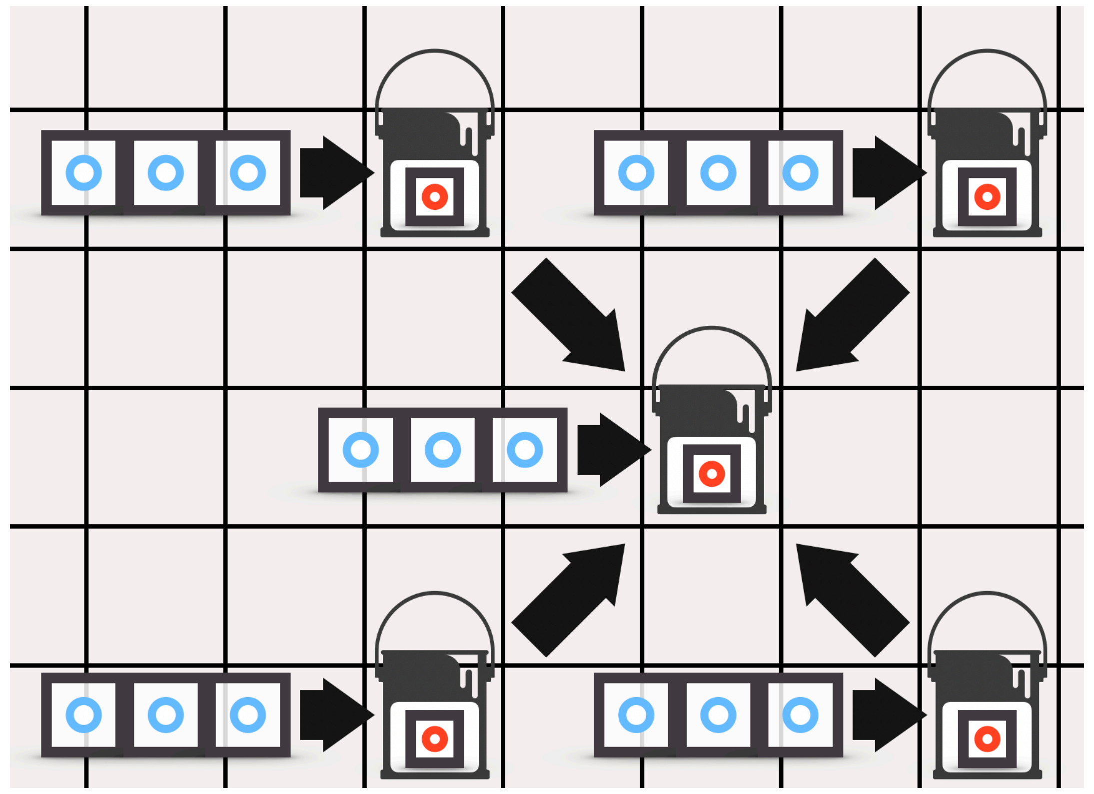
[Ou 2011]



matrix formulation
[Hasan 2007,
Ou 2011,
Huo 2015, ….]

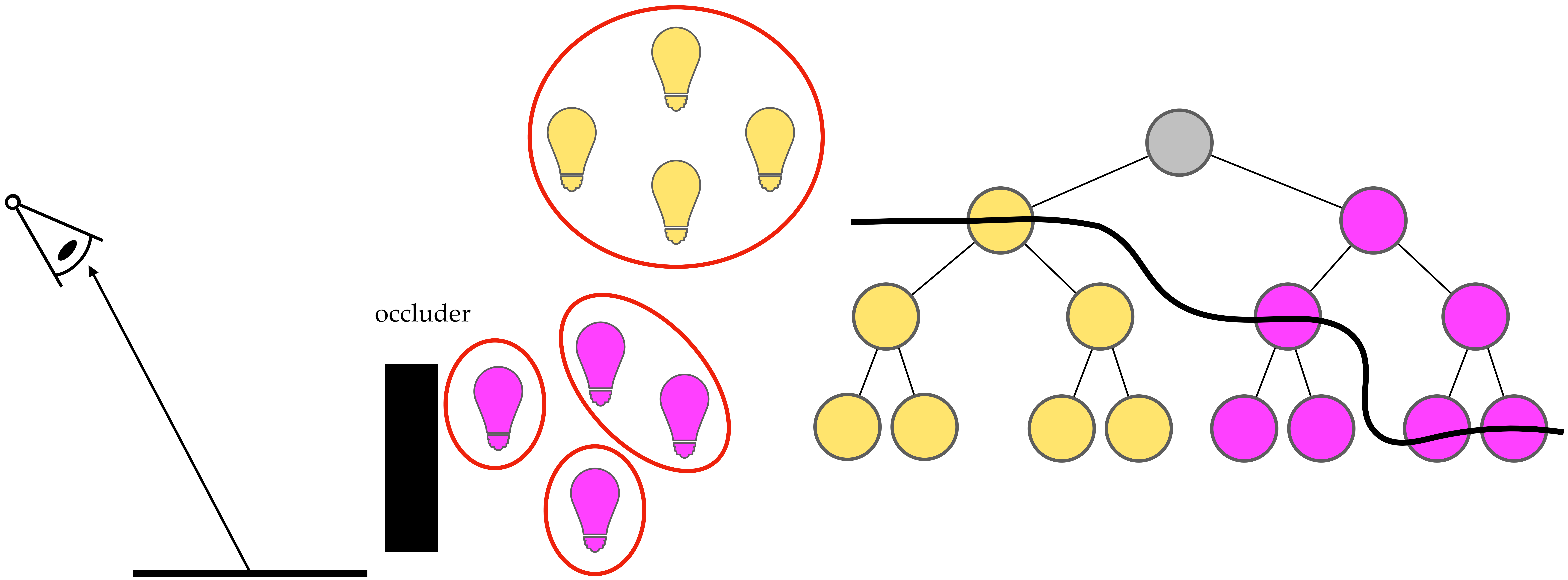



data-driven
[Donikian 2006,
Vevoda 2018,
Wang 2021]

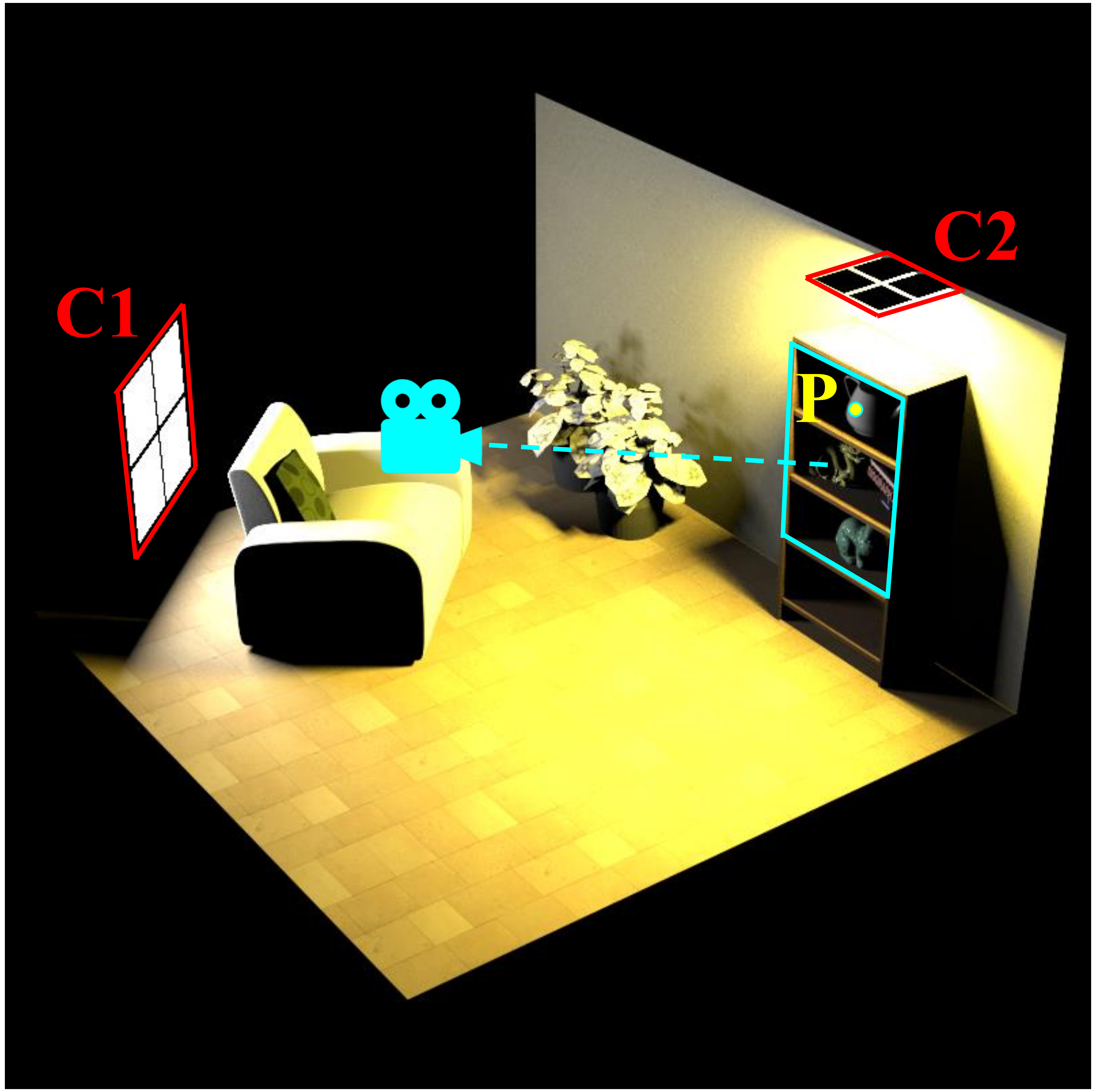


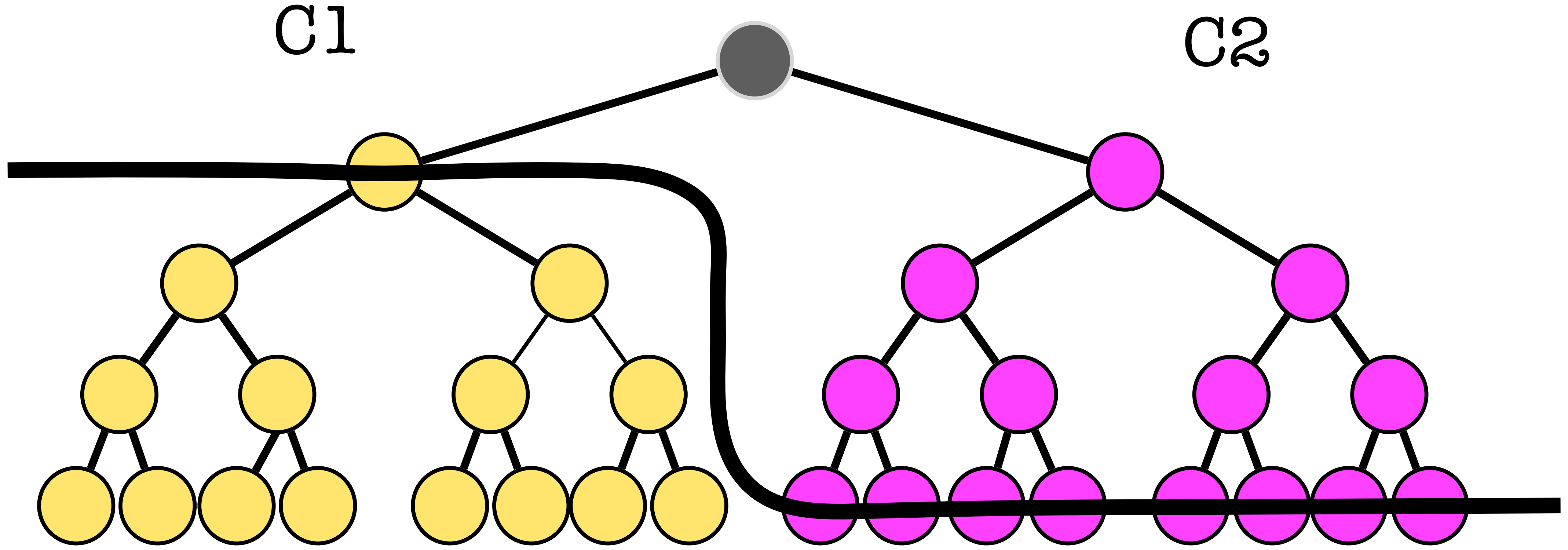
spatial-temporal reuse +
resampling
[Benedikt 2020]

# Lightcuts’ issue: ignore visibility



# A pathological case for lightcuts





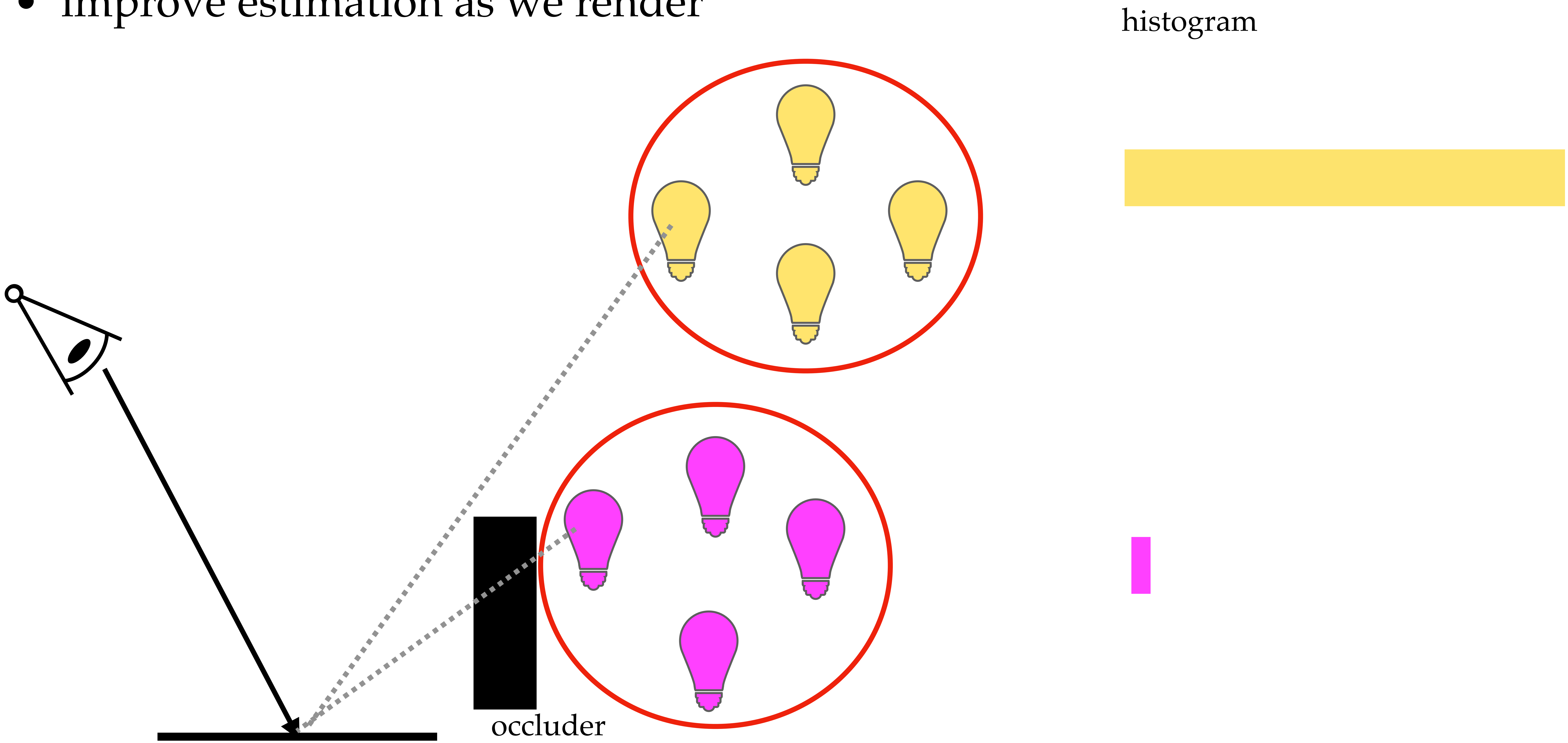
rendered image

Learning to Cluster for Rendering with Many Lights

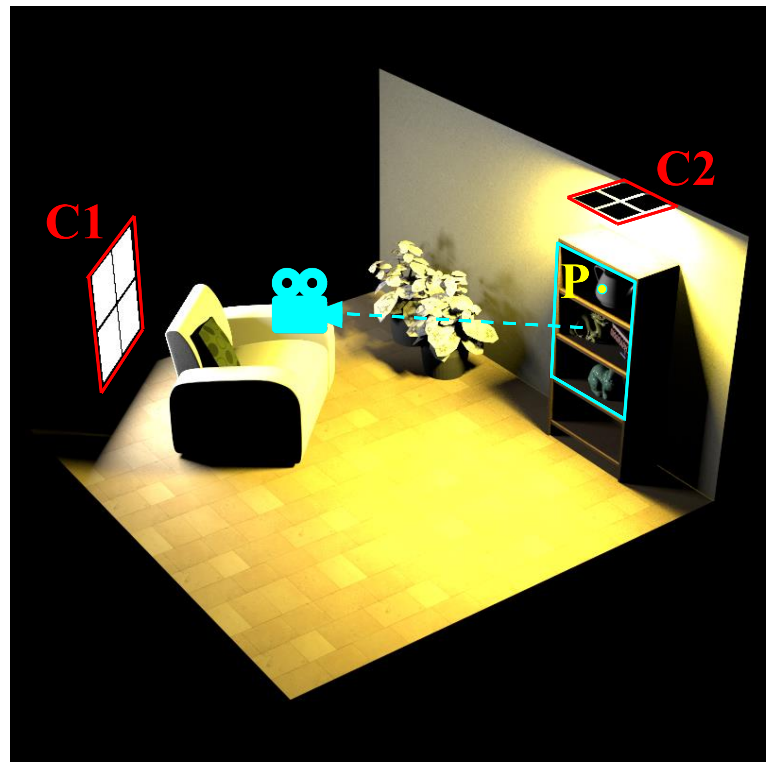
YU-CHEN WANG, National Taiwan University, Taiwan
YU-TING WU, National Taiwan University, Taiwan
TZU-MAO LI, MIT CSAIL & University of California San Diego, United States
YUNG-YU CHUANG, National Taiwan University, Taiwan

# Idea: estimate importance of clusters using samples

- improve estimation as we render



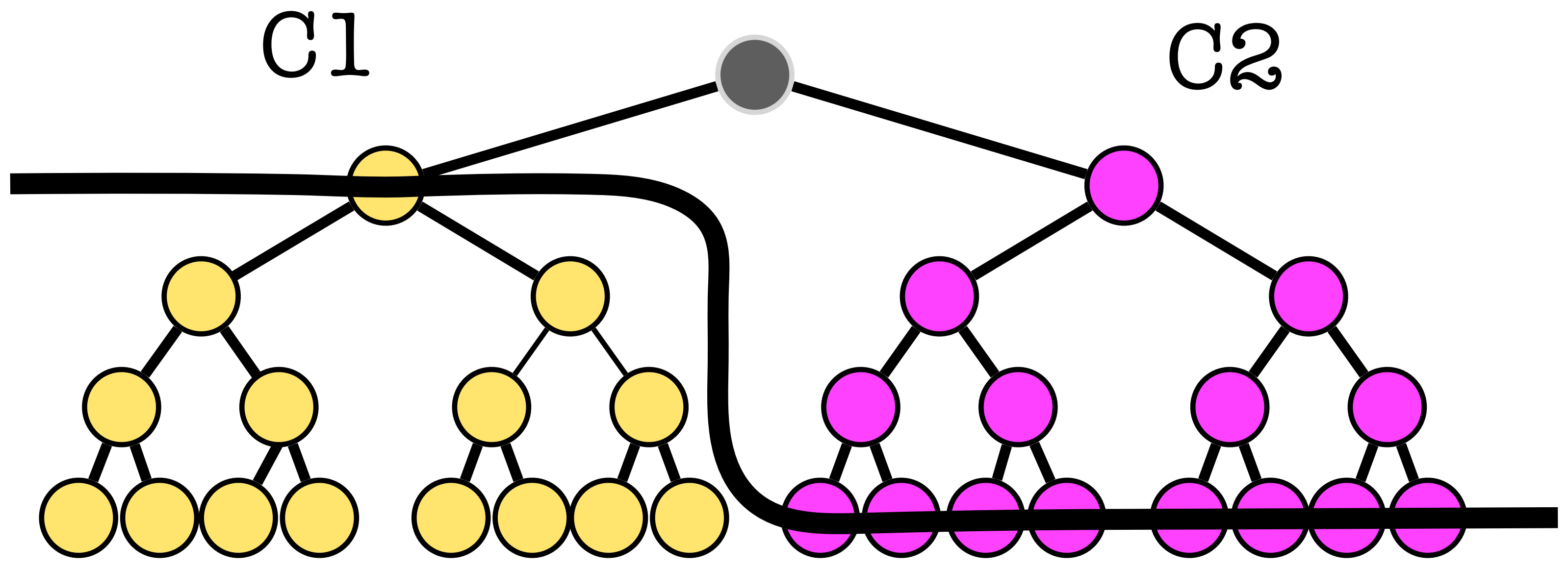
# Inappropriate clustering leads to noisy sampling





[Yuksel 2019] Stochastic Lightcut
(30 sec rendering)

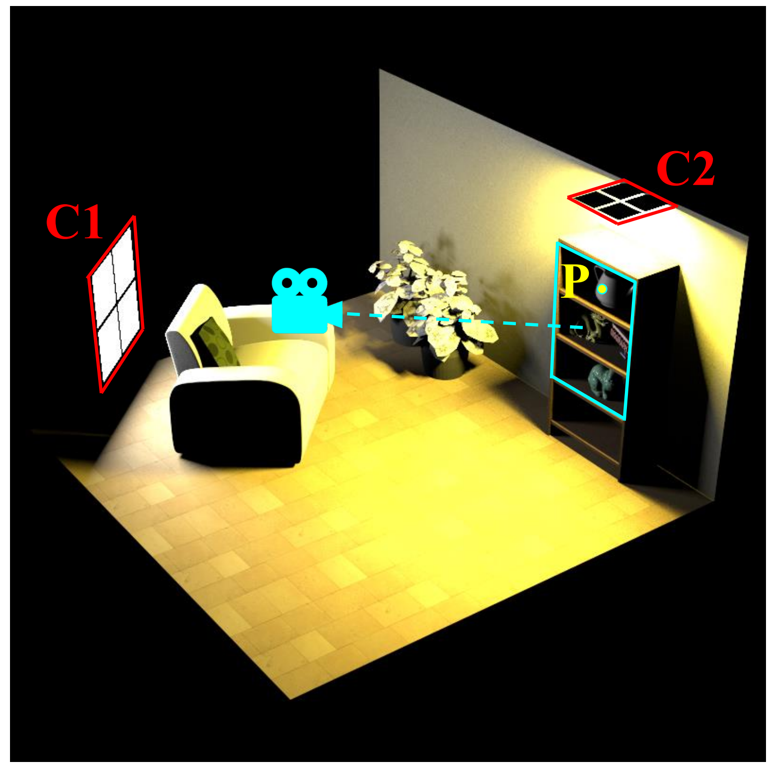

reference

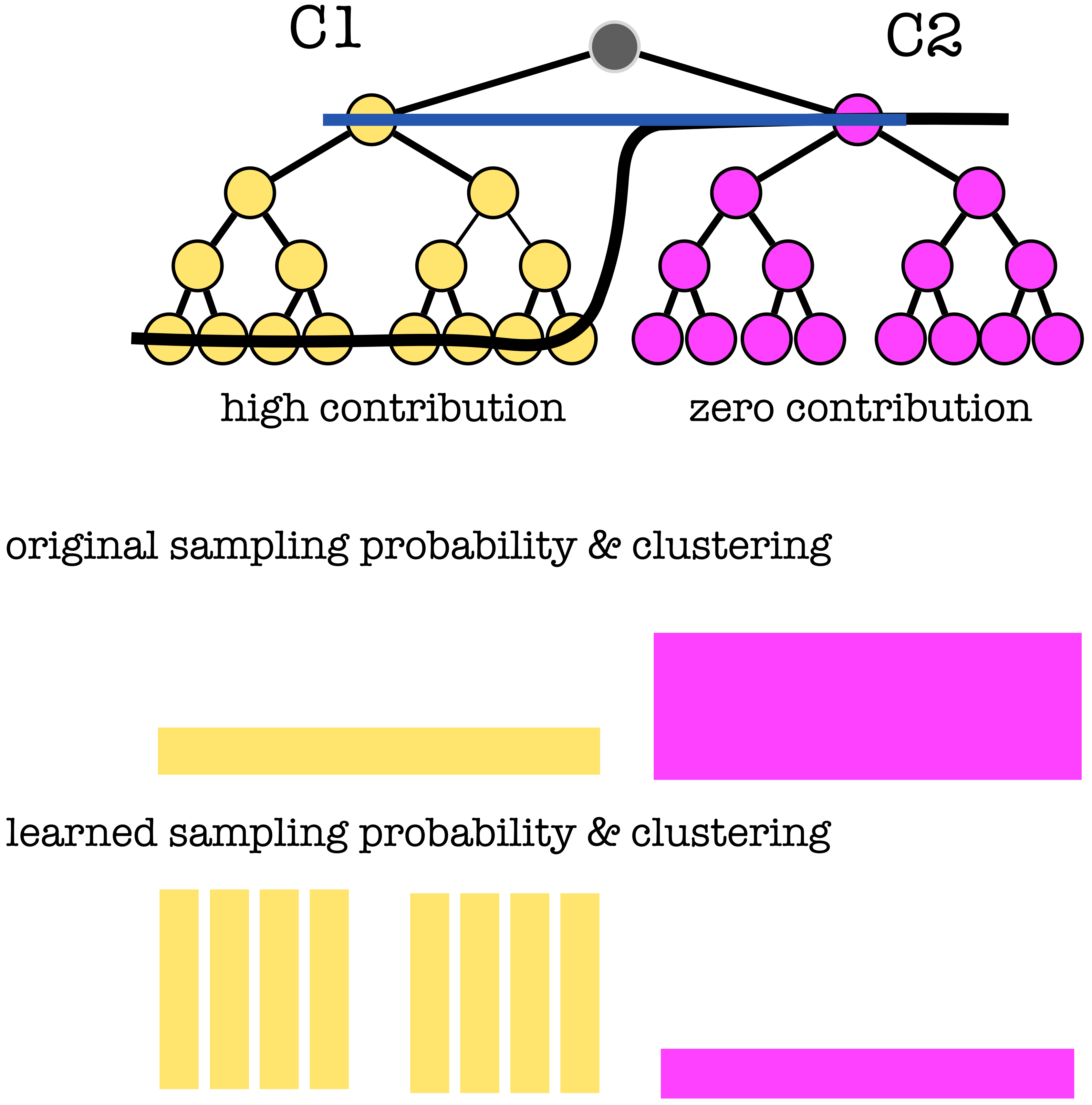


# Our method learns a good light clustering progressively


Ours
(30 sec rendering)


reference



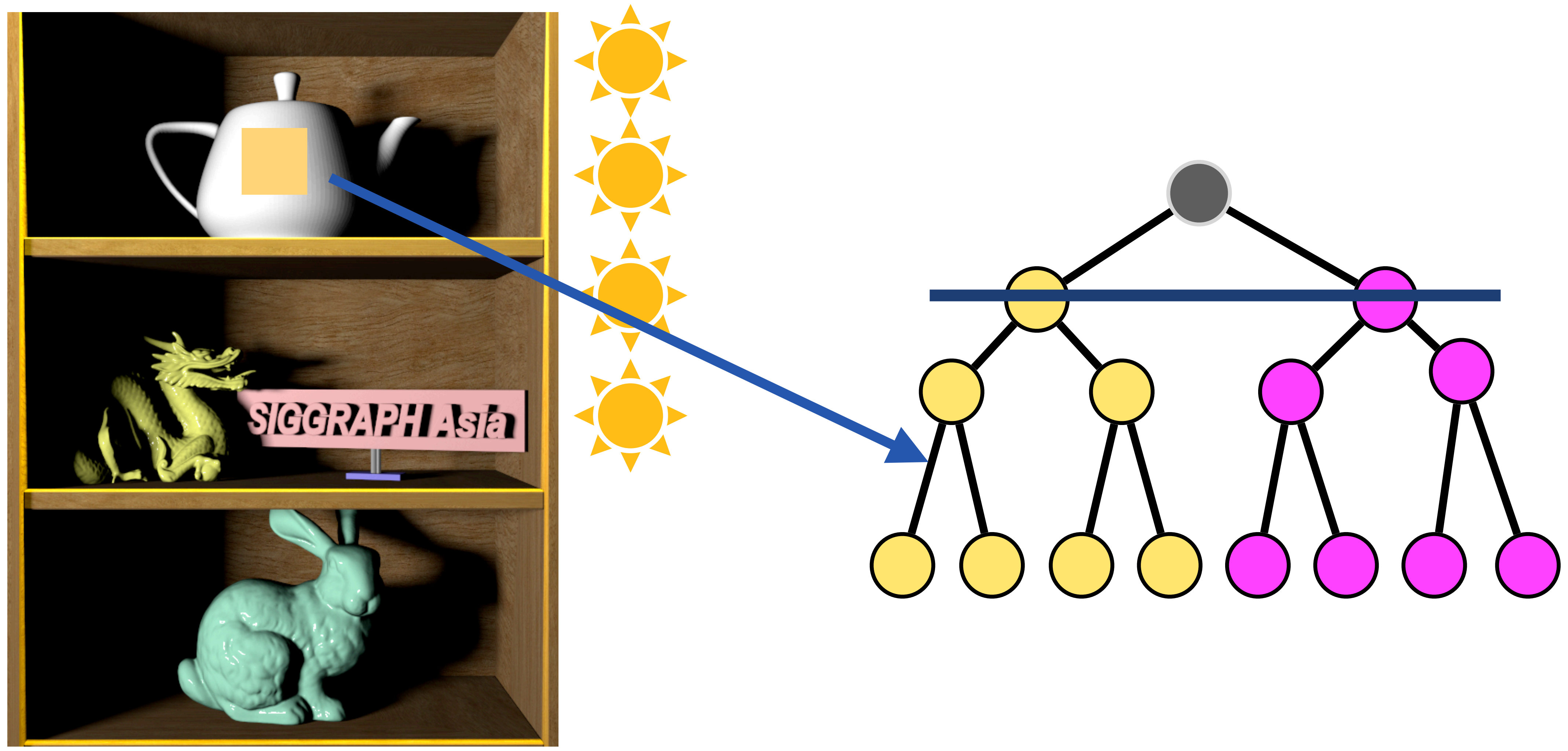



# Algorithm

# Algorithm


group shading
points into cells

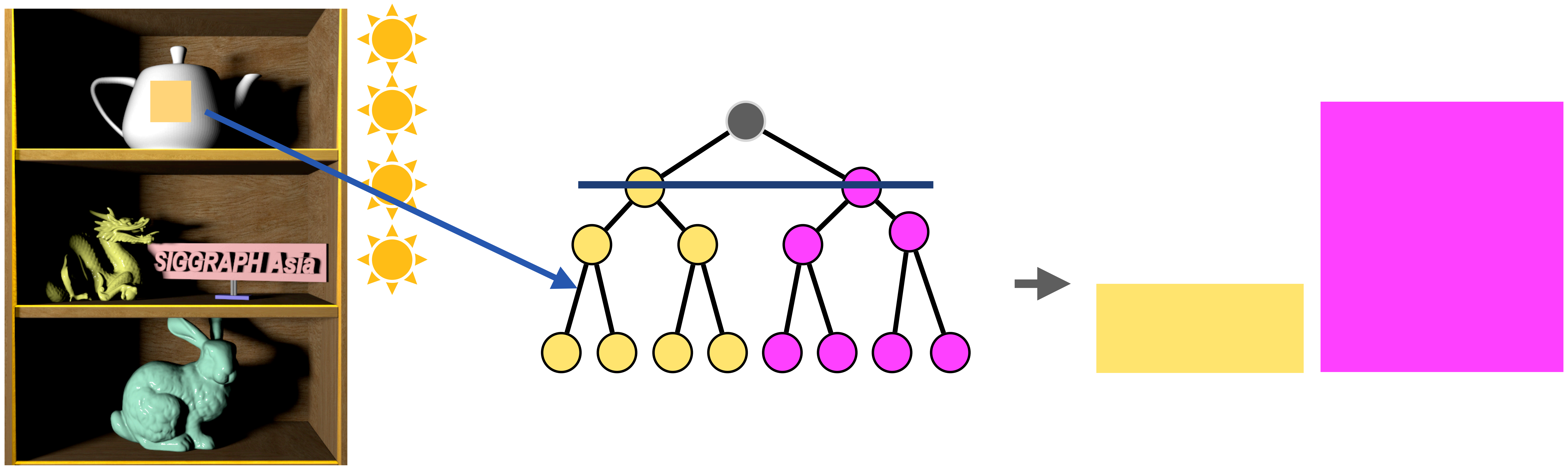
# Algorithm




group shading points into cells

build light hierarchy & init clustering

# Algorithm



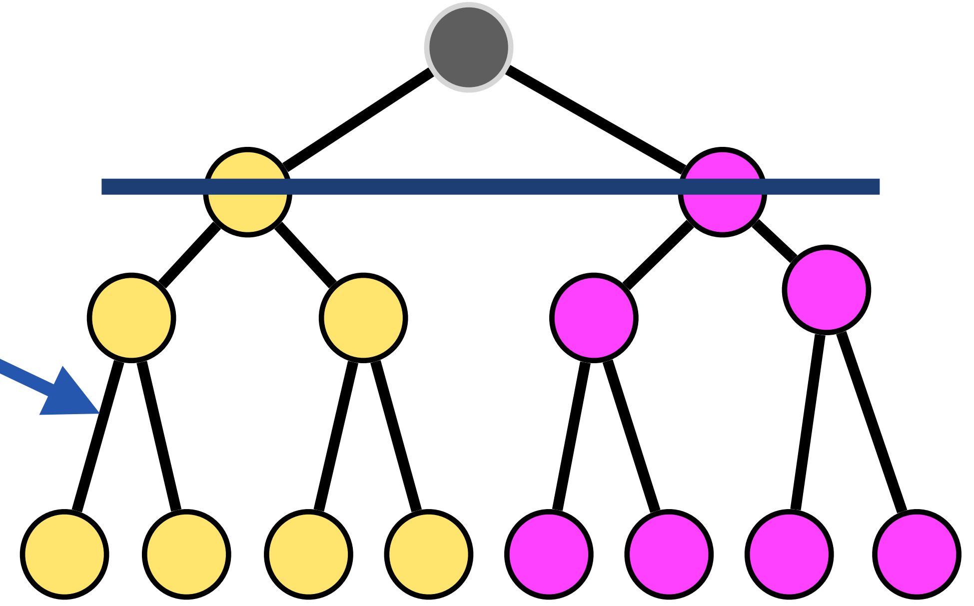

group shading points into cells

build light hierarchy & init clustering

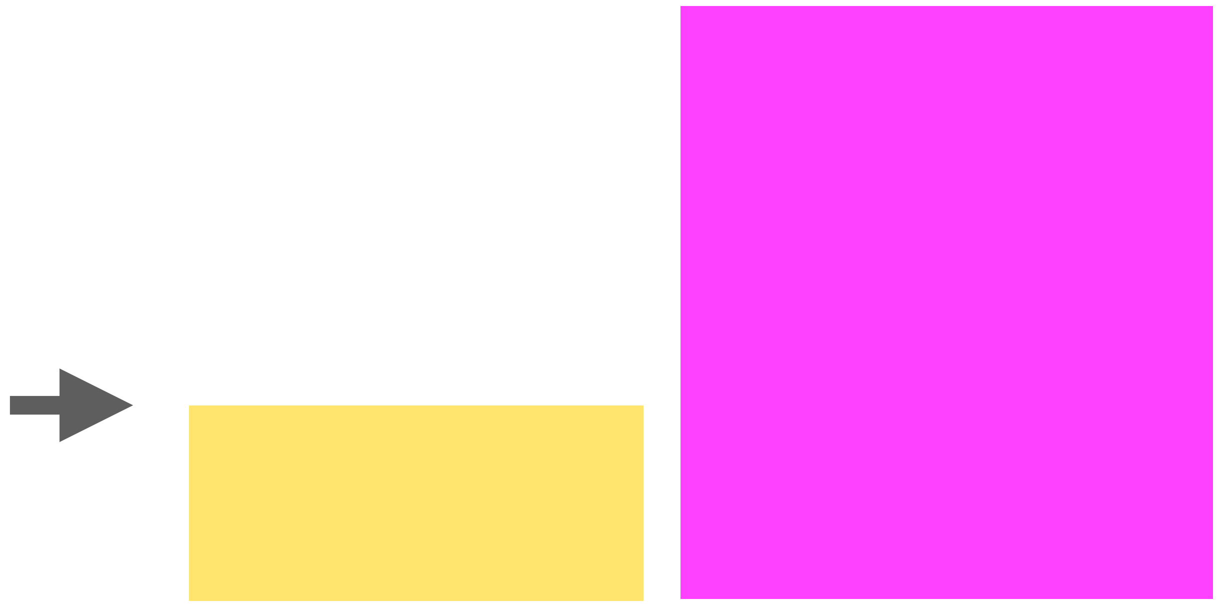
sample a cluster using importance

# Algorithm

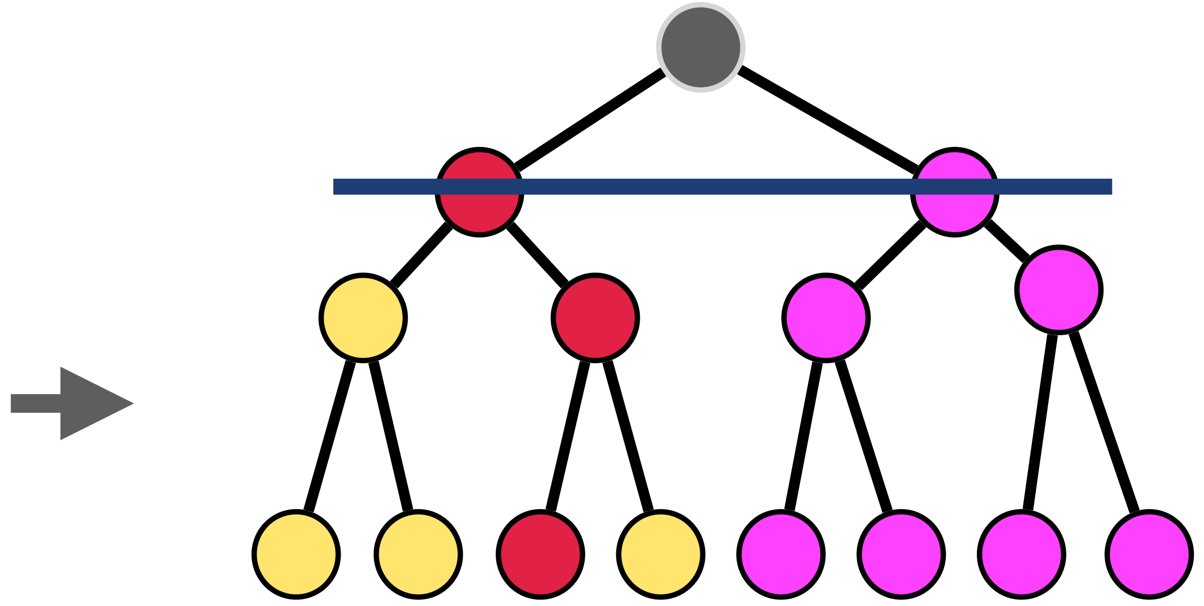

group shading points into cells



build light hierarchy & init clustering



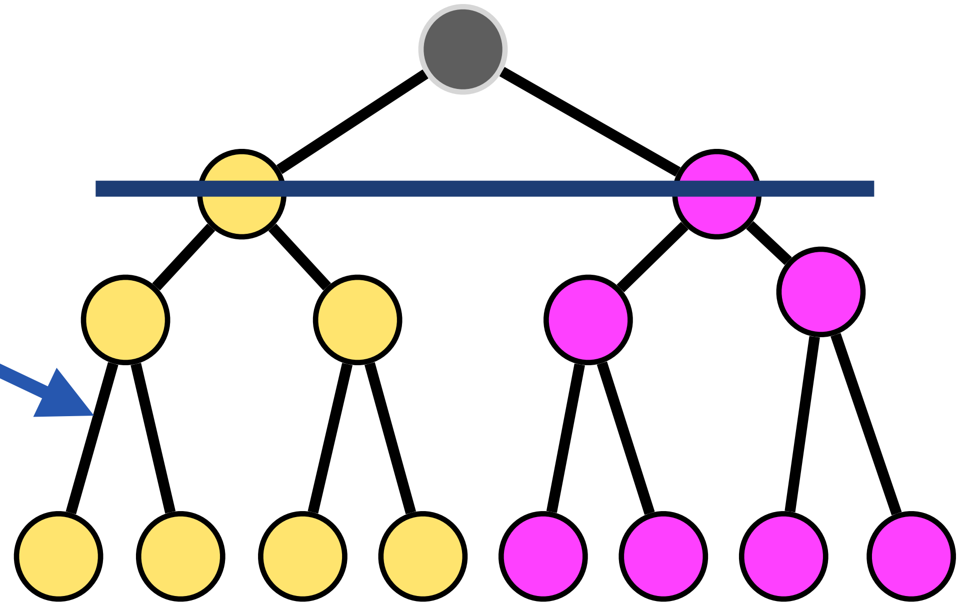
sample a cluster using importance



sample a light within the cluster [Yuksel 2019]

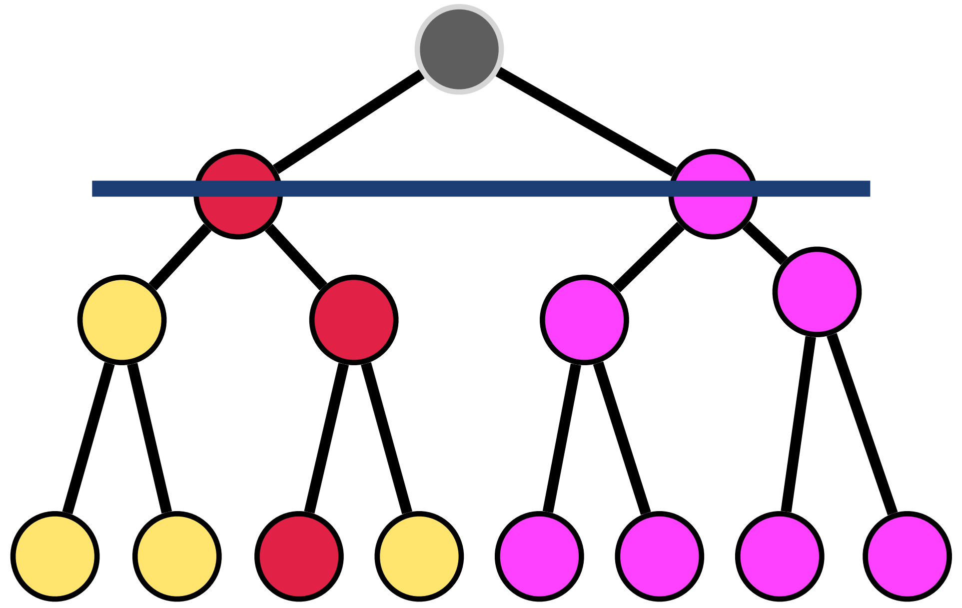
# Algorithm


group shading points into cells



build light hierarchy & init clustering

sample a cluster using importance

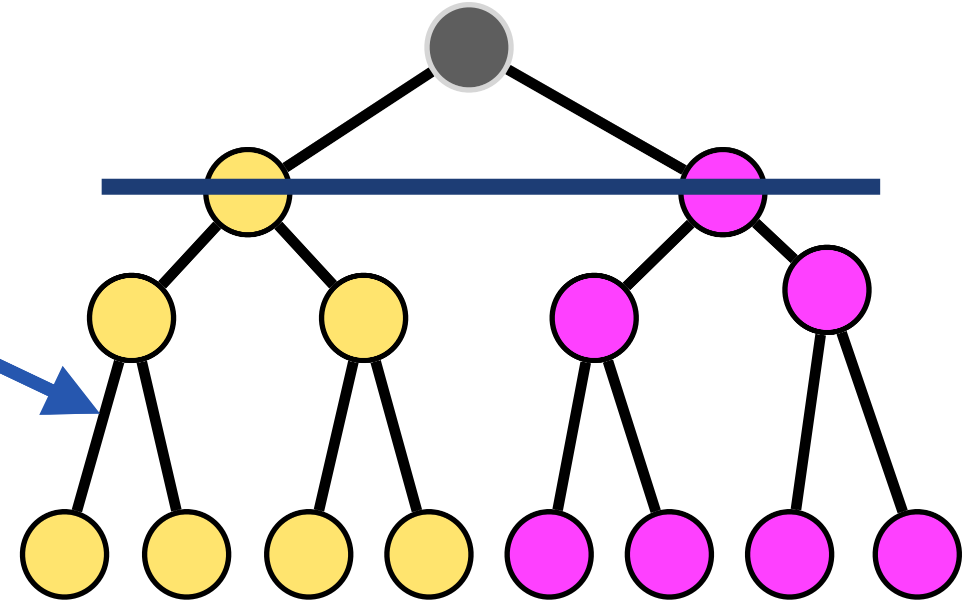


sample a light within the cluster [Yuksel 2019]

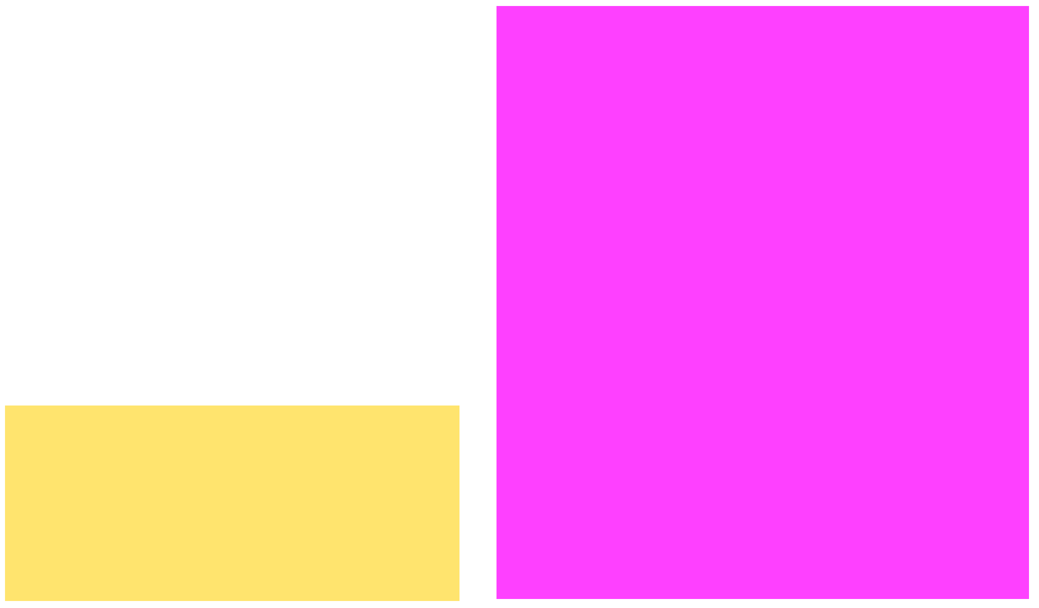
update importance & variance

# Algorithm

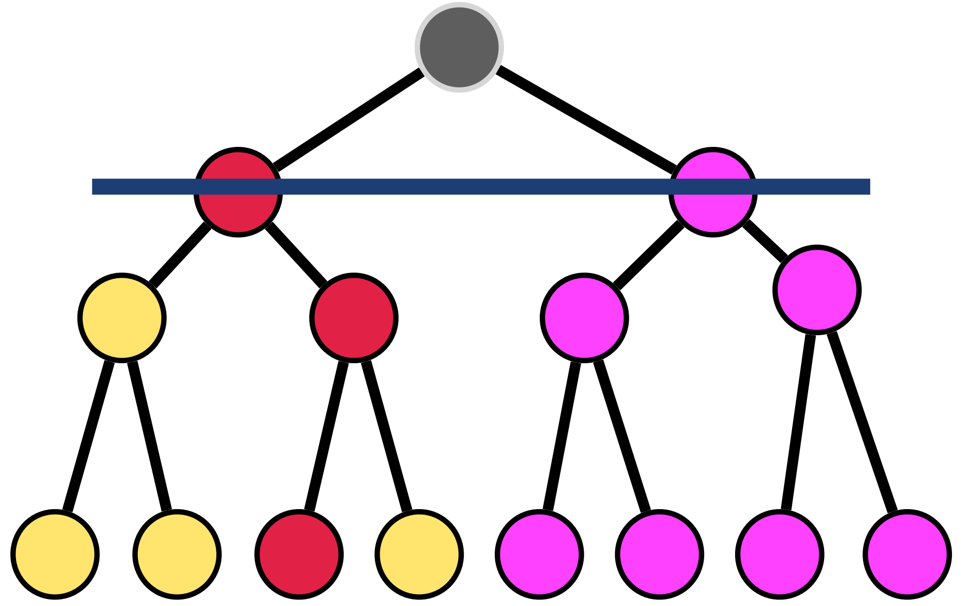

group shading points into cells



build light hierarchy & init clustering

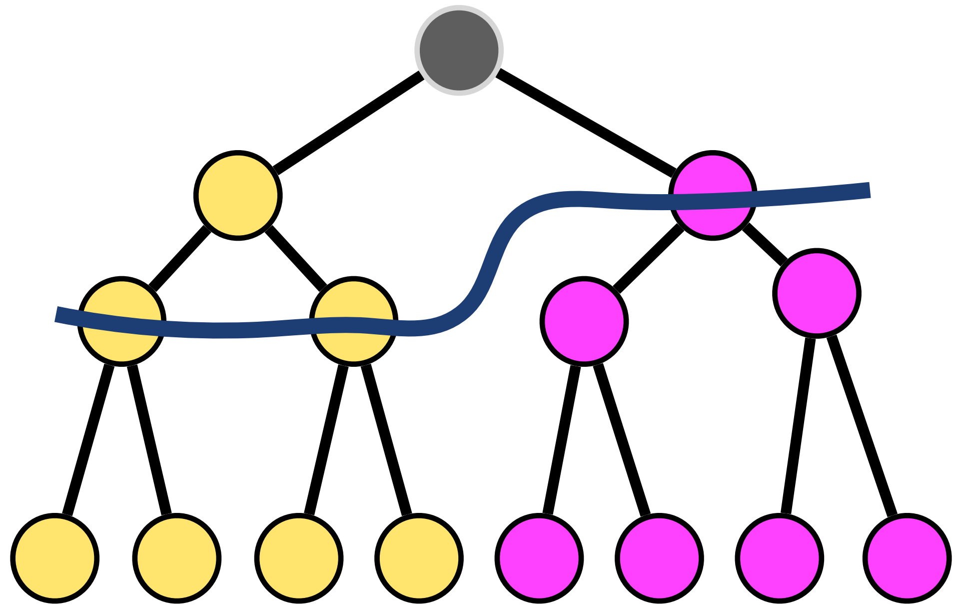


sample a cluster using importance



sample a light within the cluster [Yuksel 2019]

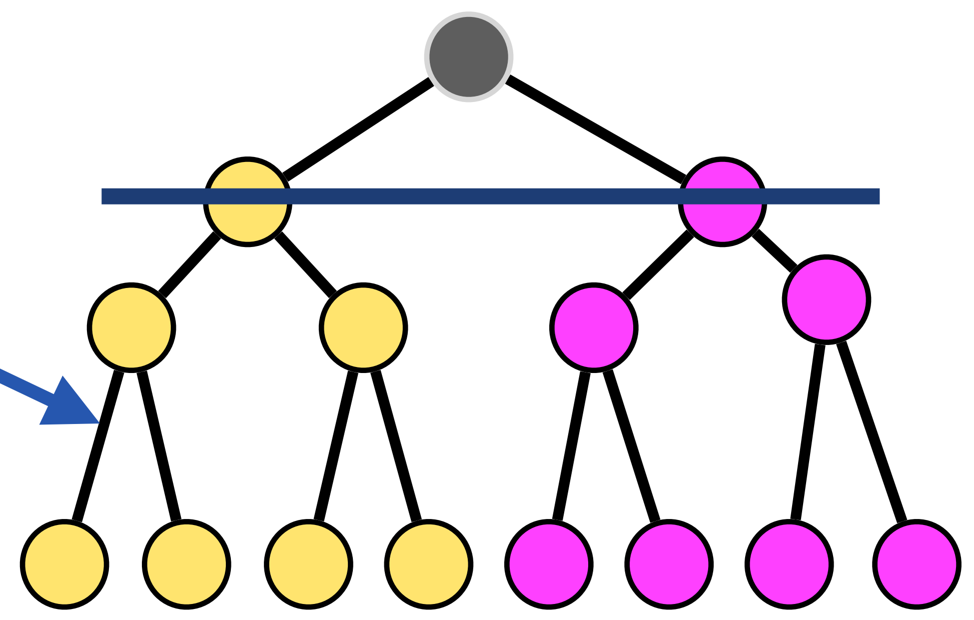
update importance & variance



split cluster if variance is large

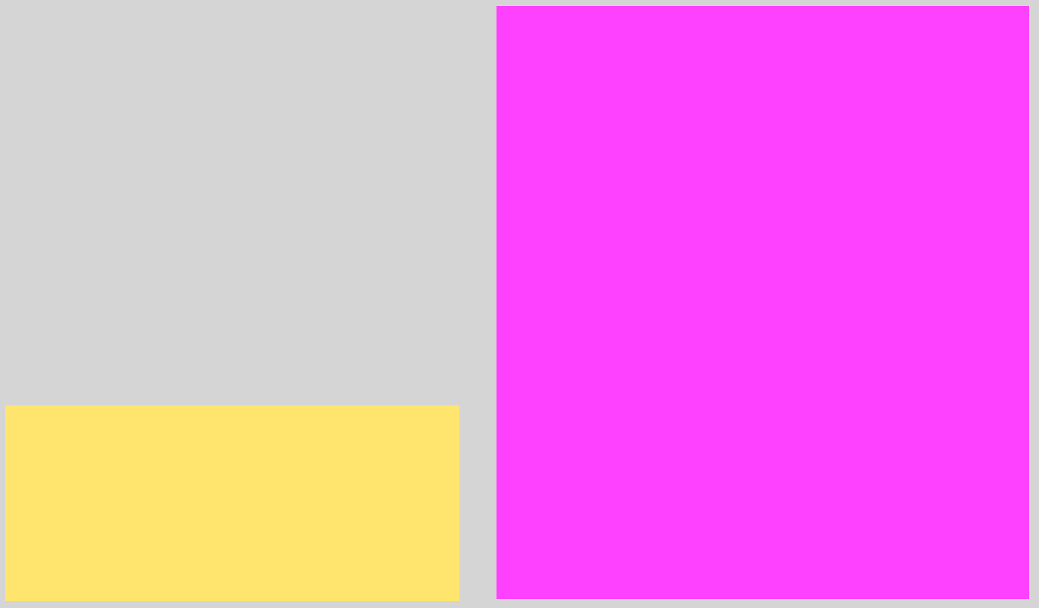
# Algorithm


group shading points into cells

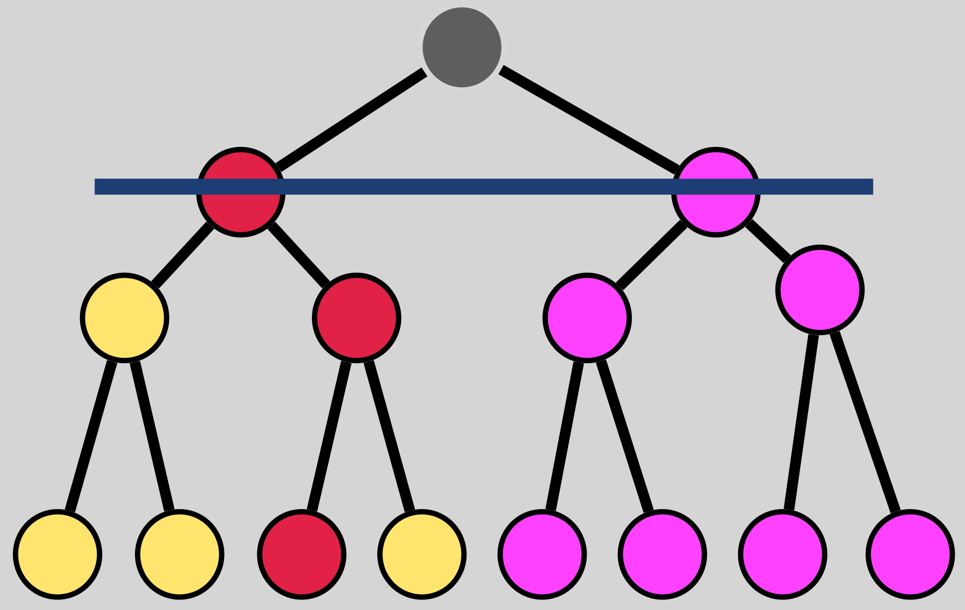


build light hierarchy & init clustering

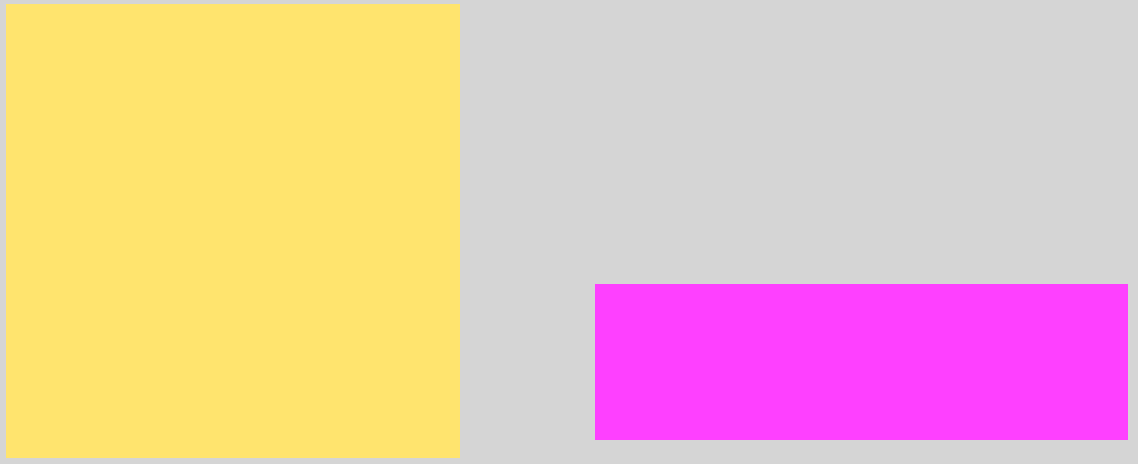
## loop



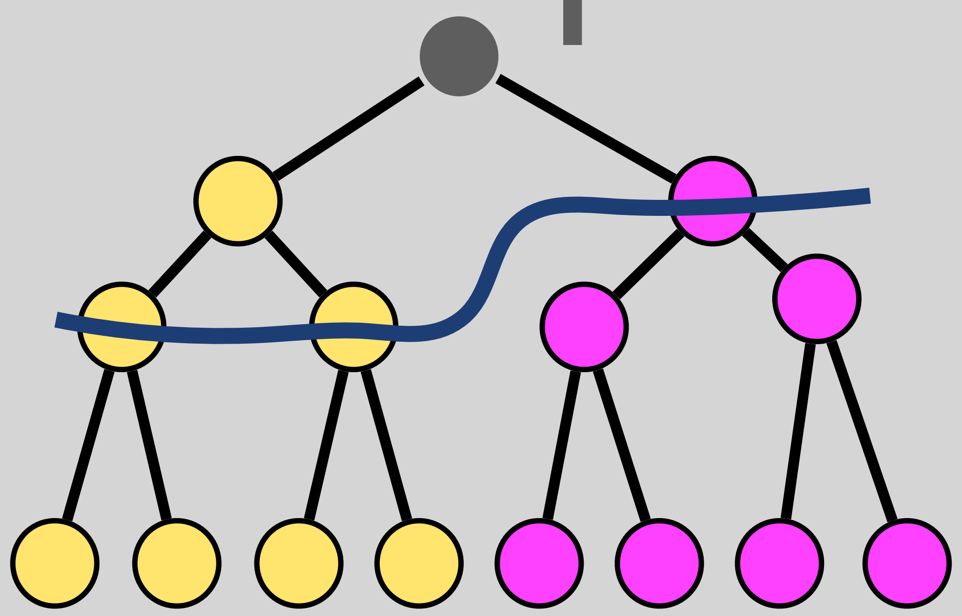
sample a cluster using importance



sample a light within the cluster [Yuksel 2019]



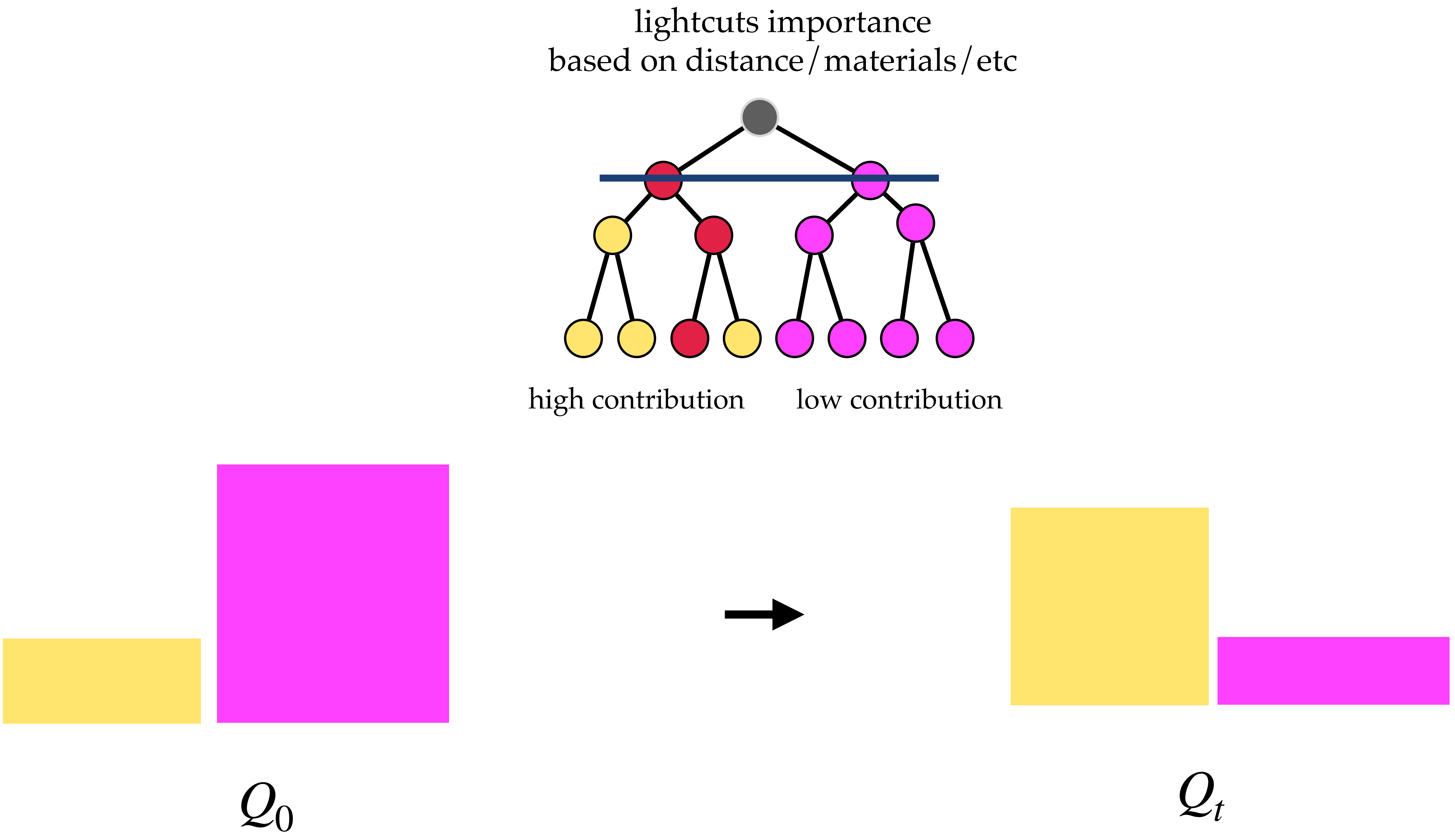
update importance & variance



split cluster if variance is large

# How do we initialize/update the importance?

- key idea: initialize the importance using lightcut upper bound, update with data

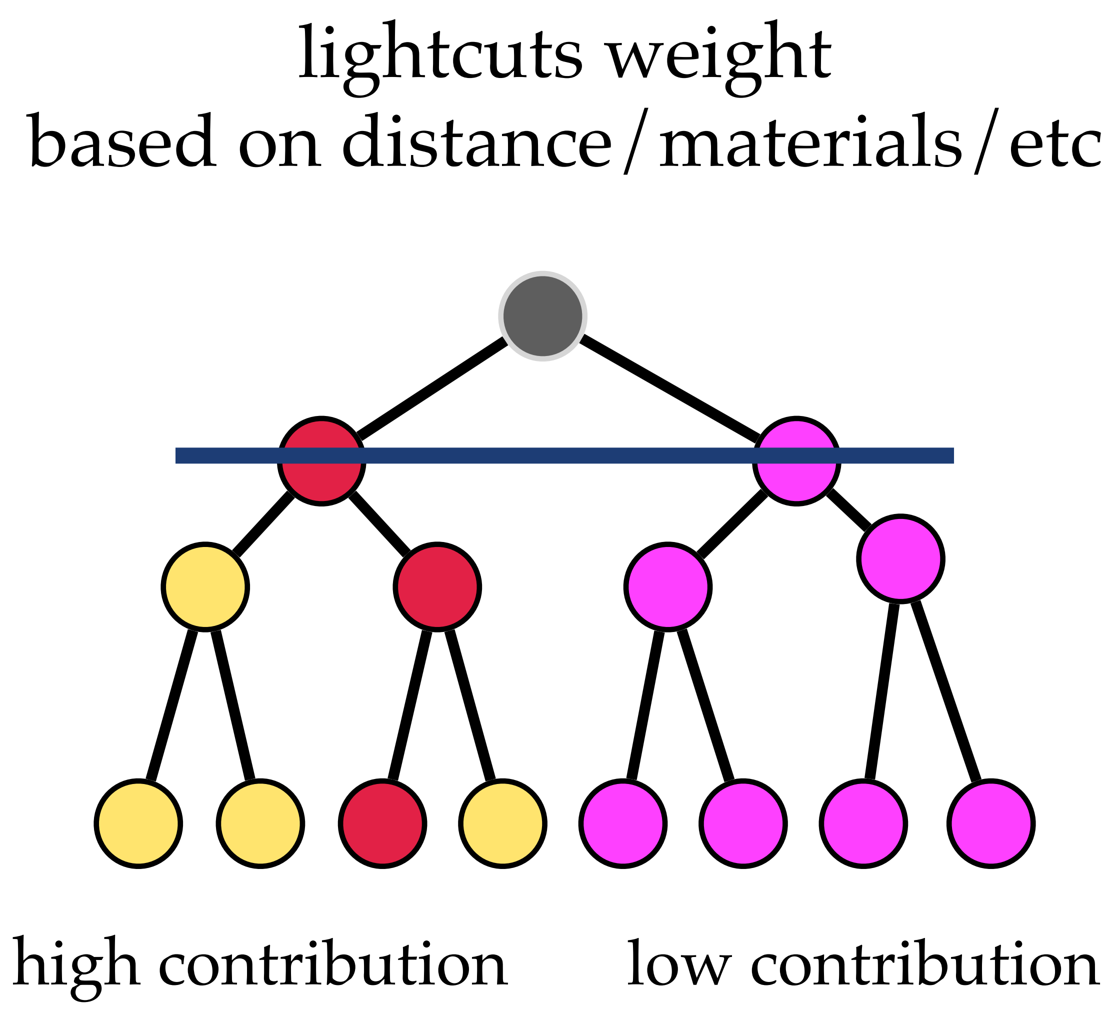


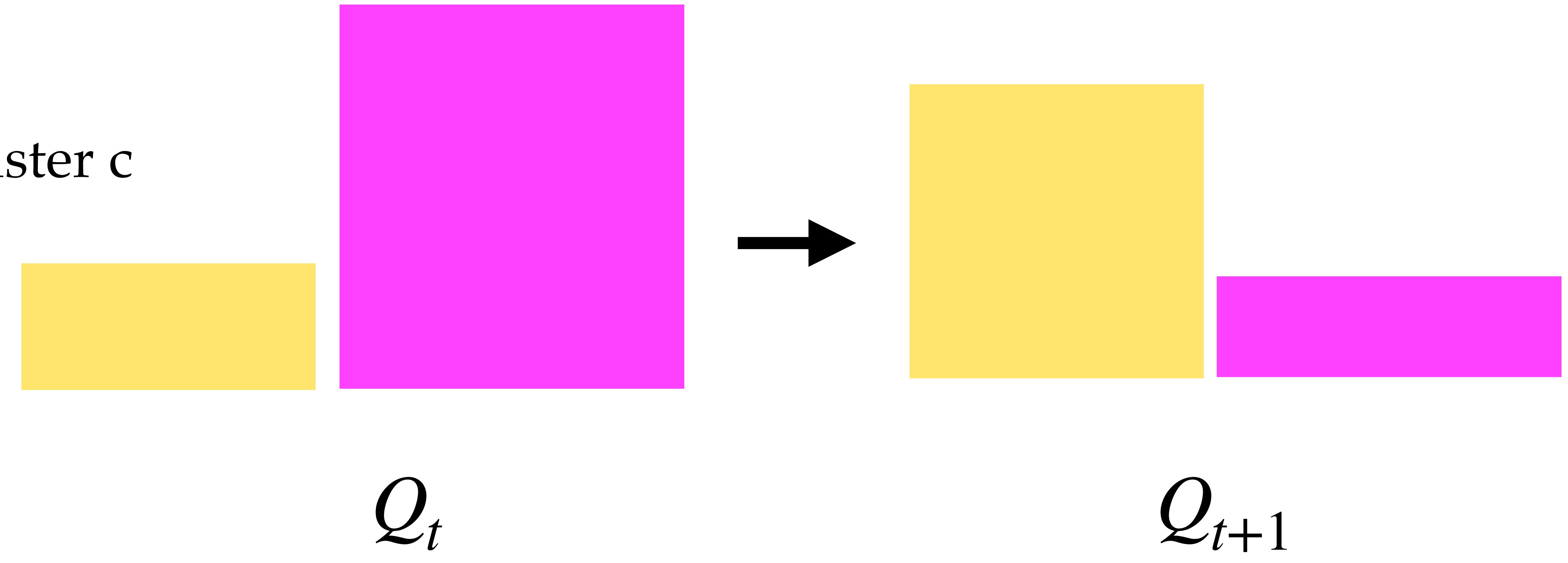
# Key idea: initialize the importance using lightcut upper bound, update with data

$$Q_0(c) = \text{lightcuts importance}$$

$$Q_{t+1}(c) = (1 - \alpha_t)Q_t(c) + \alpha_t(\text{sampling contribution})$$

goal: $Q_t(c)$ converges to the sum of contributions of lights in cluster c



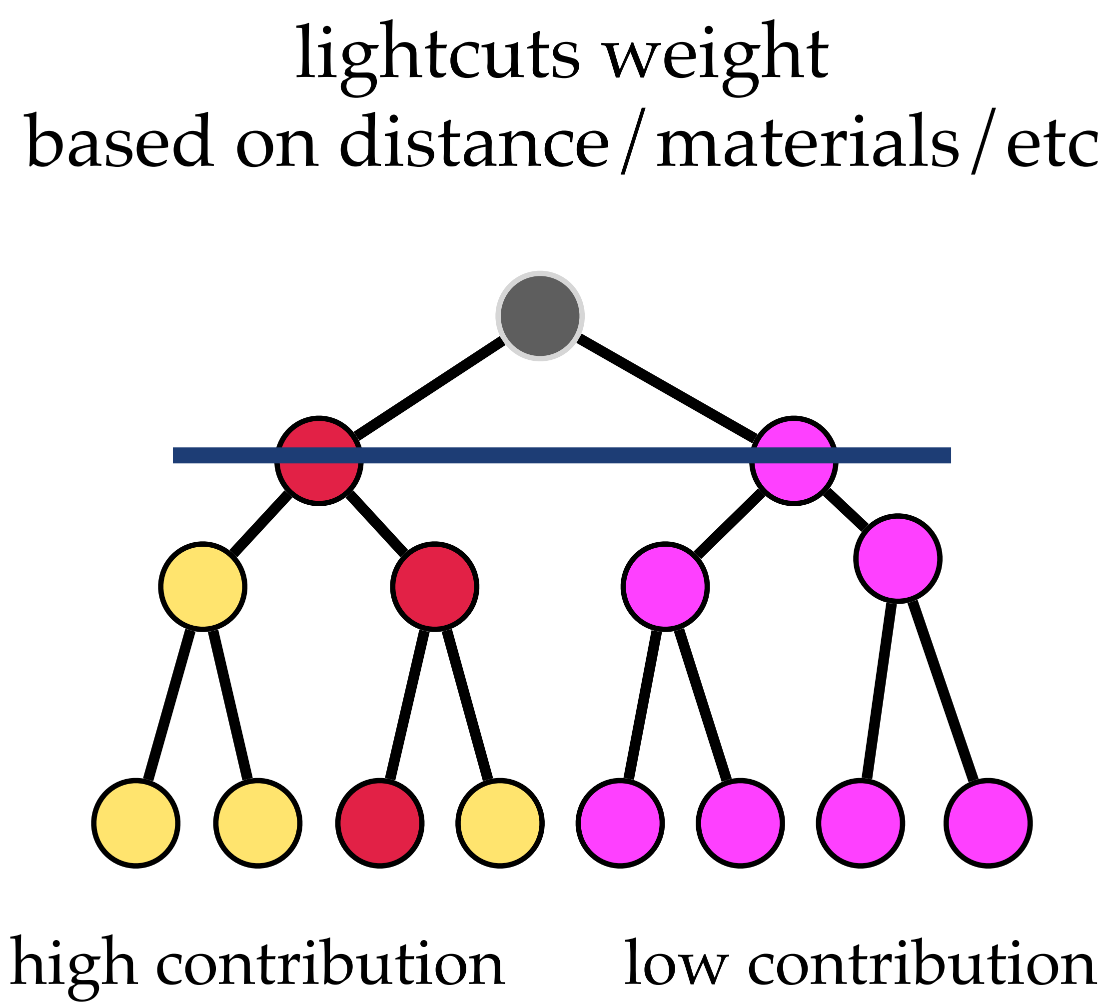



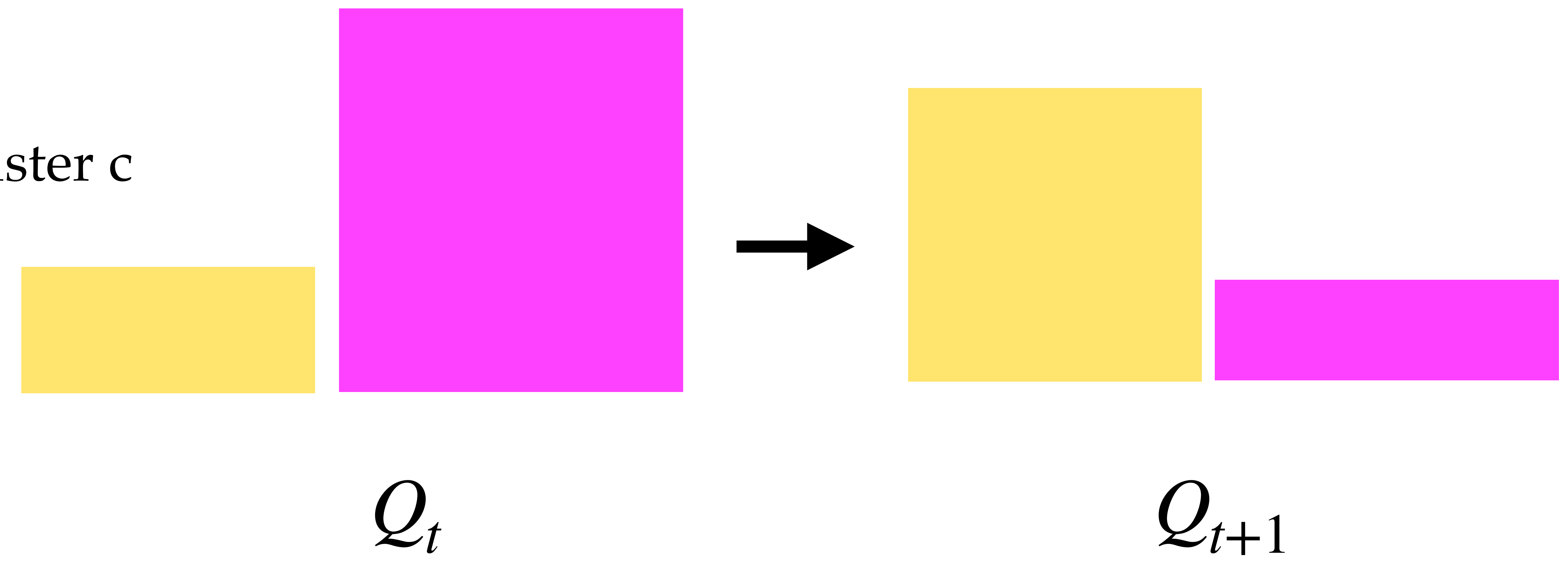
# Key idea: initialize the importance using lightcut upper bound, update with data

$$Q_0(c) = \text{lightcuts importance}$$

$$\boxed{Q_{t+1}(c) = (1-\alpha_t)Q_t(c) + \alpha_t(\text{sampling contribution})}$$

goal: $Q_t(c)$ converges to the sum of contributions of lights in cluster c



# Need to be very careful with the "learning rate" $\alpha_t$

$$Q_{t+1}(c) = (1 - \alpha_t)Q_t(c) + \alpha_t(\text{sampling contribution})$$

converges to the sum of contribution when

$$\sum_{t=1}^{\infty} \alpha_t = \infty \text{ and } \sum_{t=1}^{\infty} \alpha_t^2 < \infty$$

($\alpha_t$ = constant doesn't work!)

Stochastic Approximation
[Robbins and Monro 1951]

# Need to be very careful with the "learning rate" $\alpha_t$

$$Q_{t+1}(c) = (1 - \alpha_t)Q_t(c) + \alpha_t(\texttt{sampling contribution})$$

converges to the sum of contribution when

$$\sum_{t=1}^{\infty} \alpha_t = \infty \texttt{ and } \sum_{t=1}^{\infty} \alpha_t^2 < \infty$$

we pick $\alpha_t = \dfrac{1}{at^b}$

($\alpha_t$ = constant doesn't work!)

Stochastic Approximation
[Robbins and Monro 1951]

# Using a constant $\alpha_t$ can lead to visual artifacts!

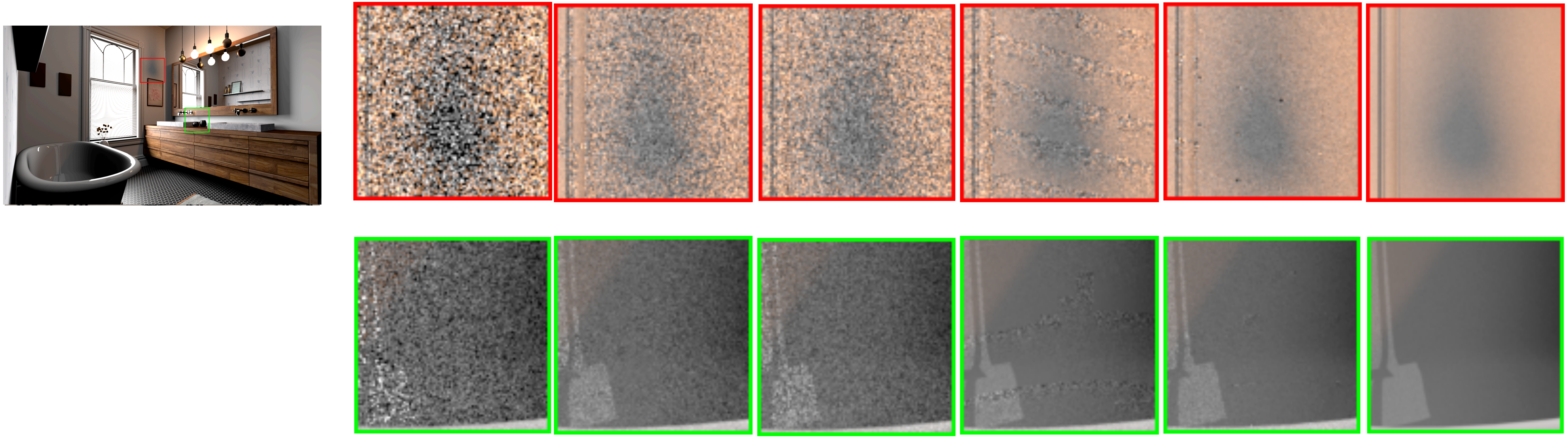
constant $\alpha_t$

ours

$$Q_{t+1}(c) = (1 - \alpha_t)Q_t(c) + \alpha_t(\text{sampling contribution})$$

# We made data-driven methods robust

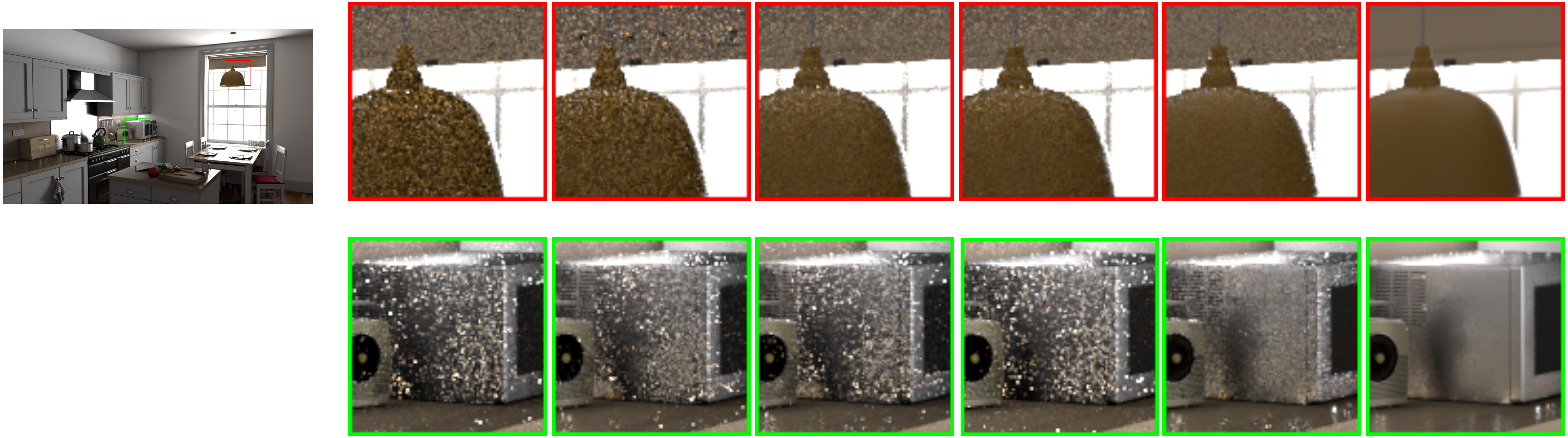
4776 lights, direct lighting only



| | | | | | | |
|---|---|---|---|---|---|---|
| method: | stochastic lightcuts [Yuksel 2019] | Bayesian online [Vevoda 2018] | variance-aware Bayesian [Rath 2020] | reinforcement lightcuts [Pantaleoni 2019] | ours | ref |
| relMSE: | 0.152 | 0.095 | 0.101 | 0.065 | 0.057 | |

# We made data-driven methods robust

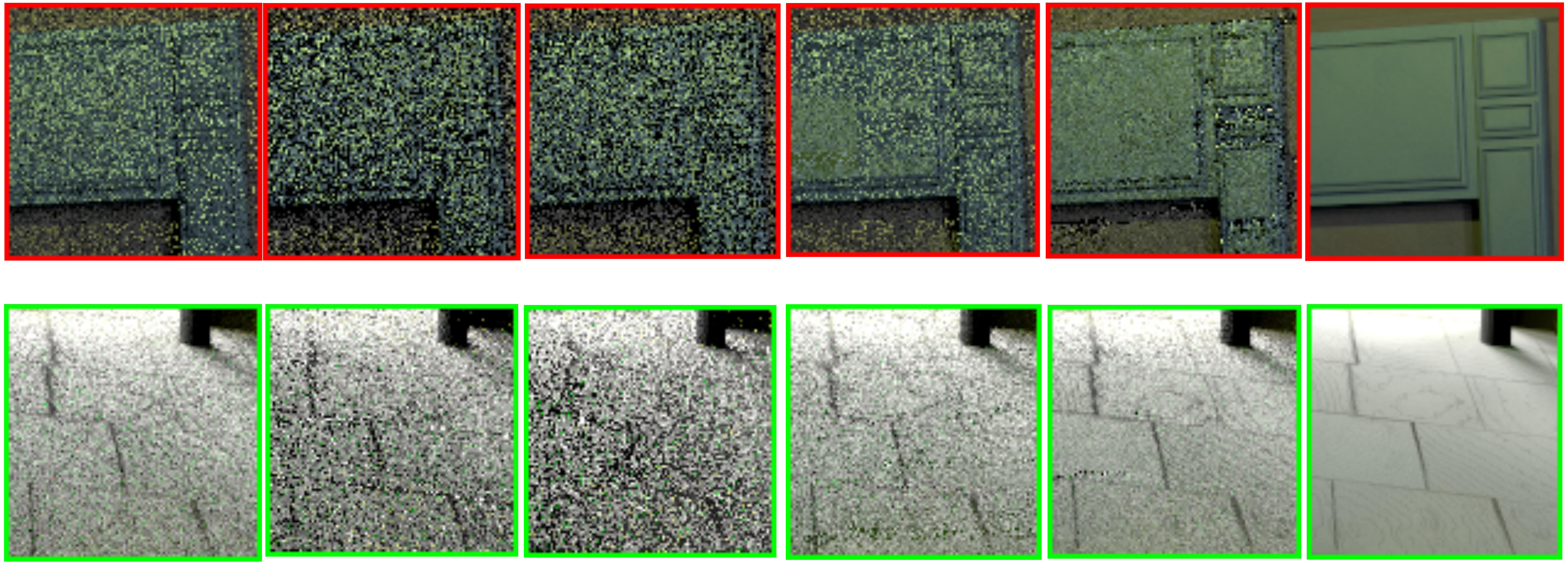
indirect illumination rendered with 71311 virtual point lights



| | | | | | | |
|---|---|---|---|---|---|---|
| method: | stochastic lightcuts [Yuksel 2019] | Bayesian online [Vevoda 2018] | variance-aware Bayesian [Rath 2020] | reinforcement lightcuts [Pantaleoni 2019] | ours | ref |
| relMSE: | 0.352 | 1.034 | 0.476 | 0.404 | 0.050 | |

# We made data-driven methods robust

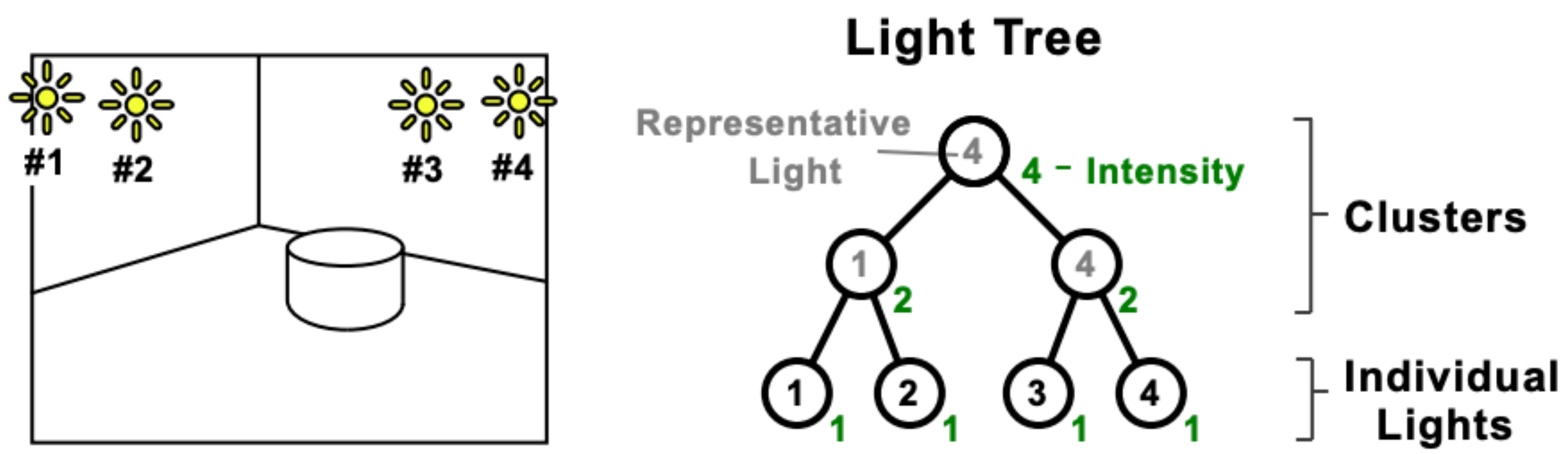
90862 lights, direct illumination only



| | | | | | | |
|---|---|---|---|---|---|---|
| method: | stochastic lightcuts [Yuksel 2019] | Bayesian online [Vevoda 2018] | variance-aware Bayesian [Rath 2020] | reinforcement lightcuts [Pantaleoni 2019] | ours | ref |
| relMSE: | 0.153 | 0.766 | 0.480 | 0.237 | 0.047 | |

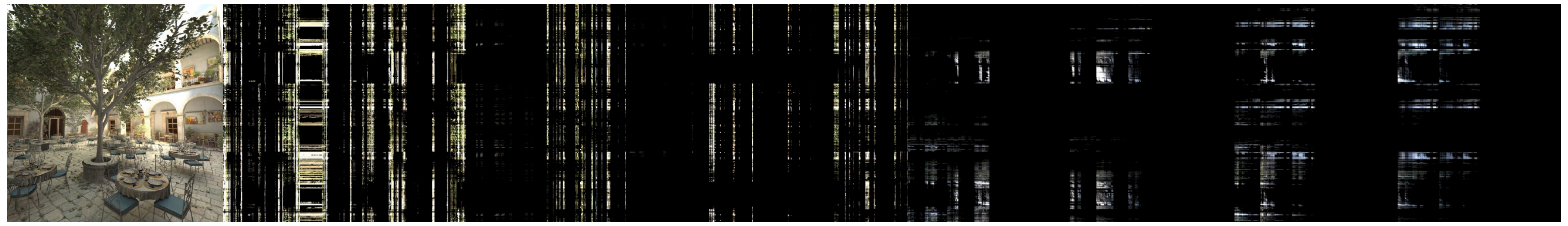
# Ideas

[Walter 2005]

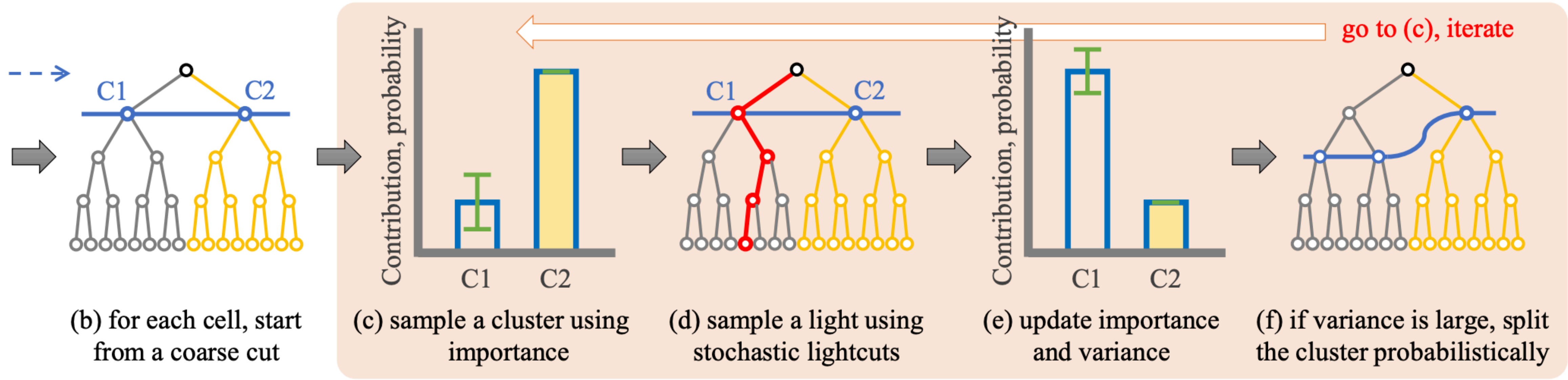



hierarchical clustering
[Shirley 1996,
Paquette 1998,
Walter 2005, …]

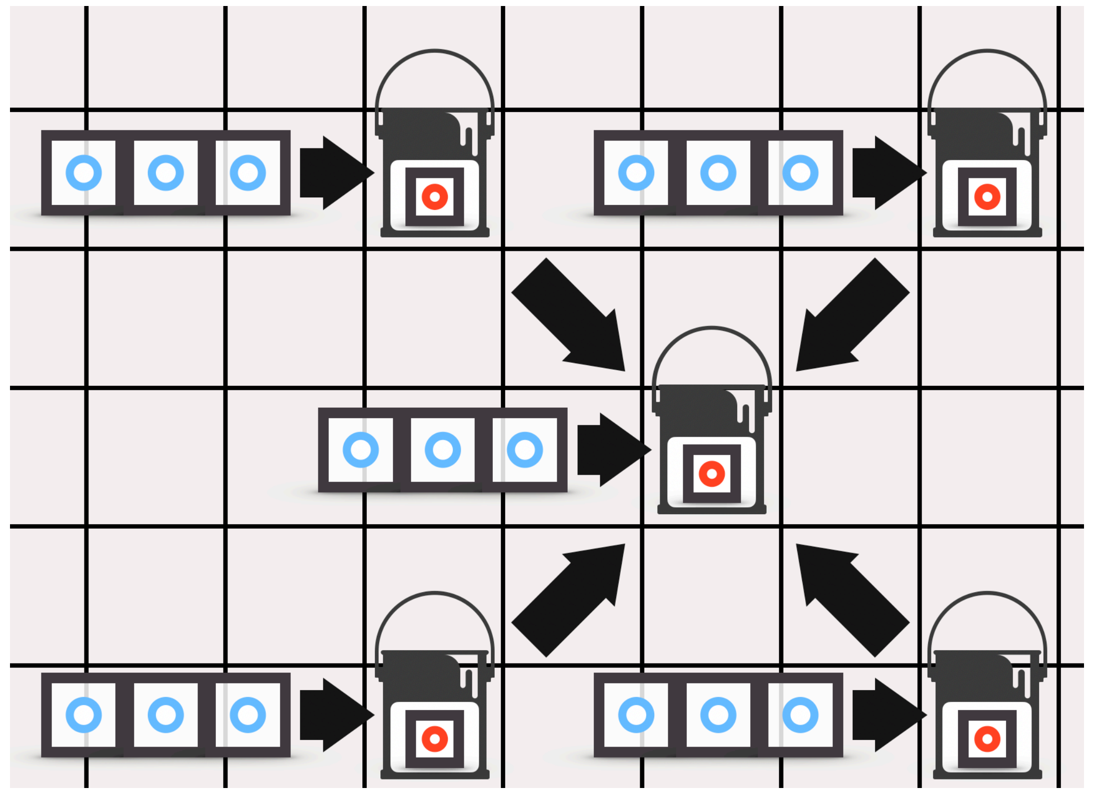
[Ou 2011]



matrix formulation
[Hasan 2007,
Ou 2011,
Huo 2015, ….]



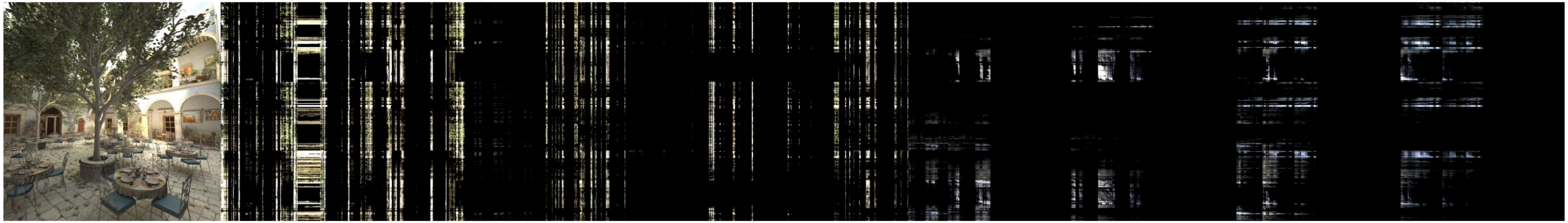

data-driven
[Donikian 2006,
Vevoda 2018,
Wang 2021]



spatial-temporal reuse +
resampling
[Benedikt 2020]

# Many-lights rendering =
# estimating the light transport matrix

lights



pixels

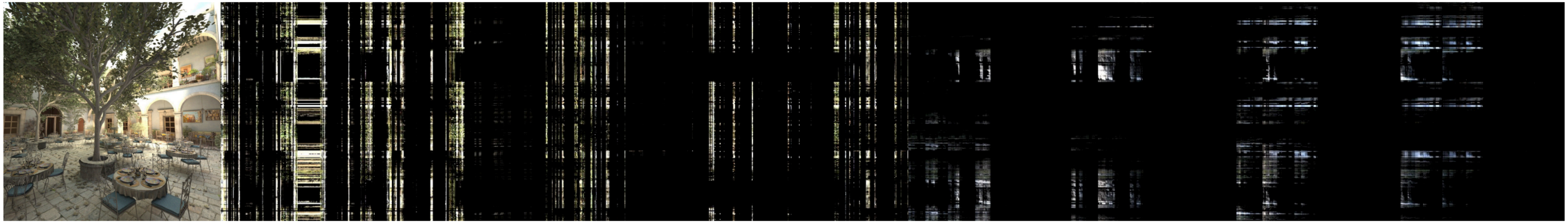
figure from Ou 2011

**Matrix Row-Column Sampling for the Many-Light Problem**


Miloš Hašan* — Cornell University
Fabio Pellacini — Dartmouth College
Kavita Bala — Cornell University

# Many-lights rendering = estimating the light transport matrix

lights



pixels

figure from Ou 2011

observation: the light transport matrix is low-rank!

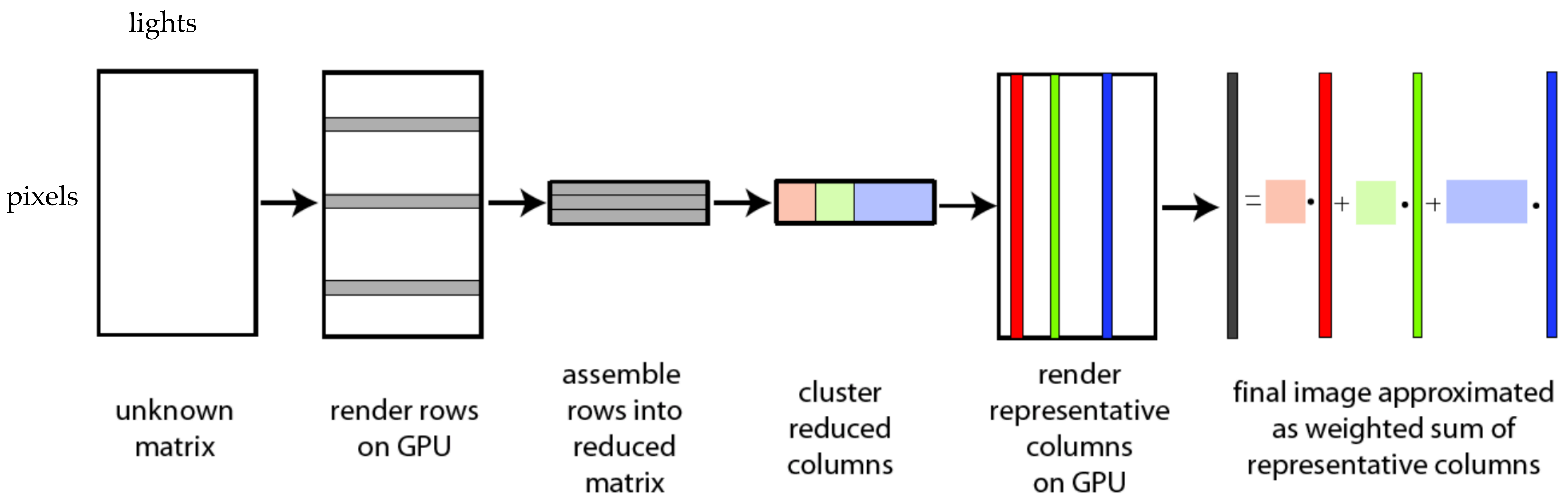
**Matrix Row-Column Sampling for the Many-Light Problem**

Miloš Hašan* Cornell University

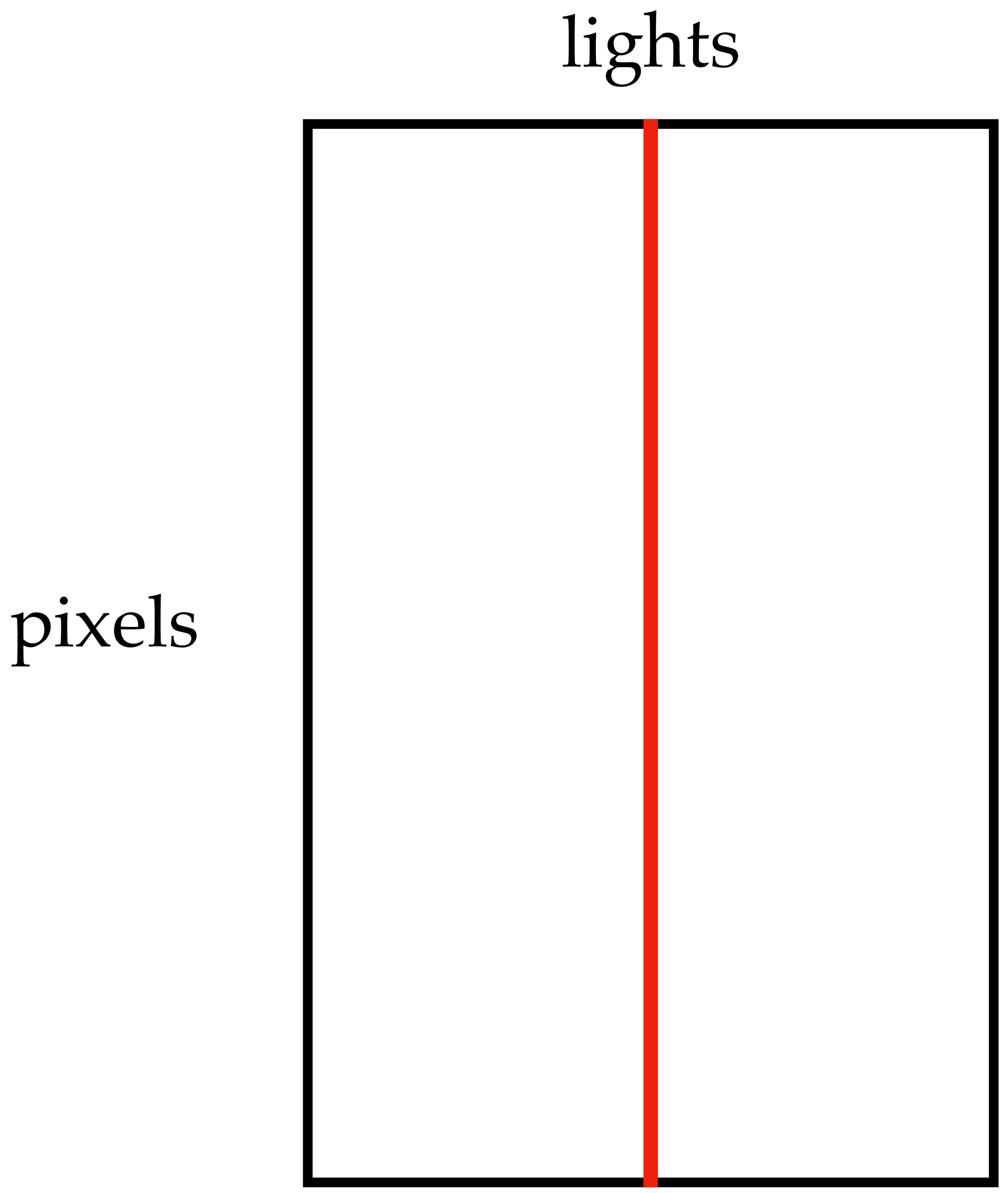
Fabio Pellacini Dartmouth College

Kavita Bala Cornell University

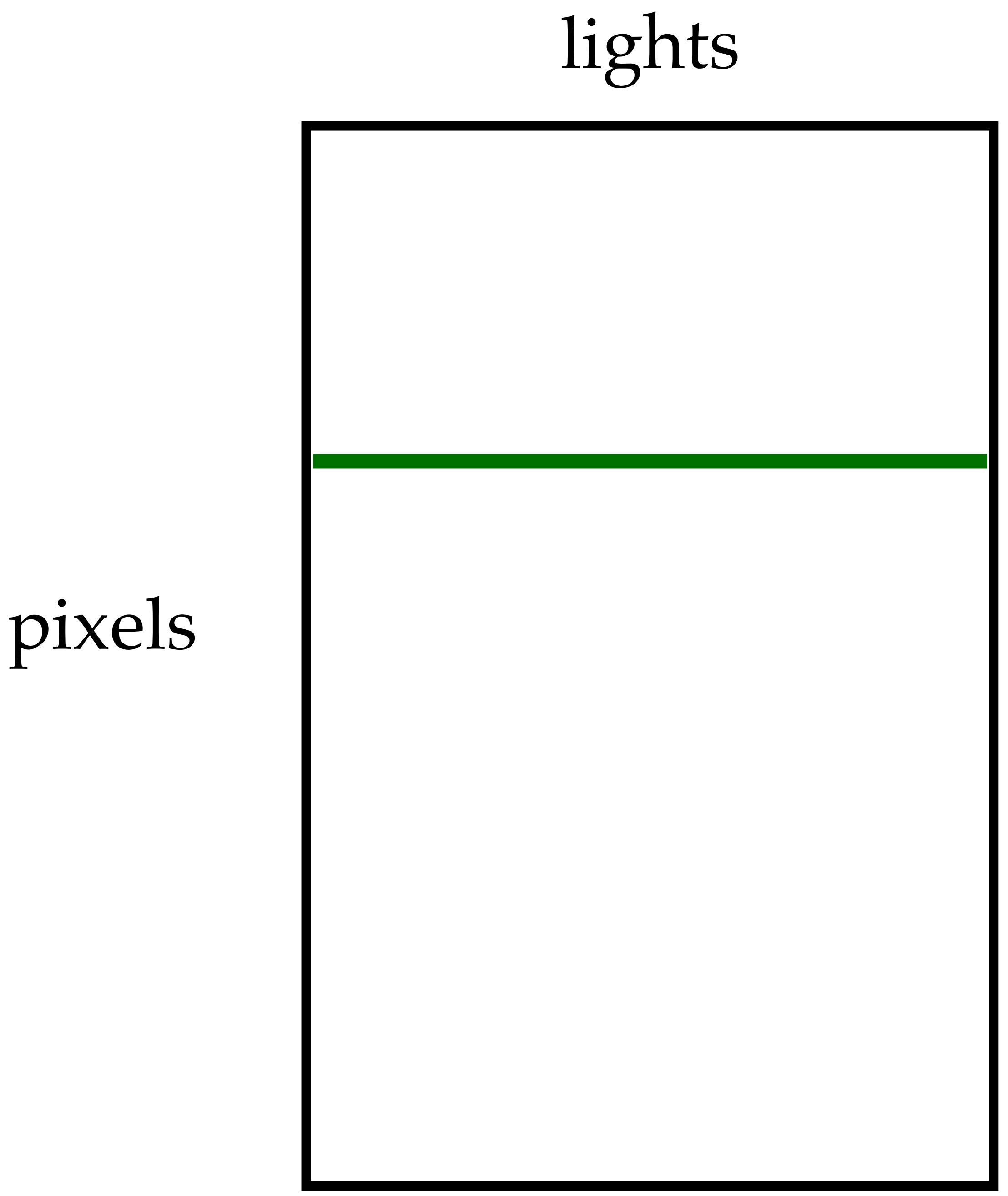
# Idea: reconstruct the light transport matrix by sampling rows and columns



# Row/colummn sampling can be done using rasterization/shadow mapping!




column sampling =
render a point light for all pixels




column sampling =
render a pixel with all lights

# Result: high-quality global illumination only using rasterization!

2.2m triangles: 300 rows, 900 columns, 16.9 s

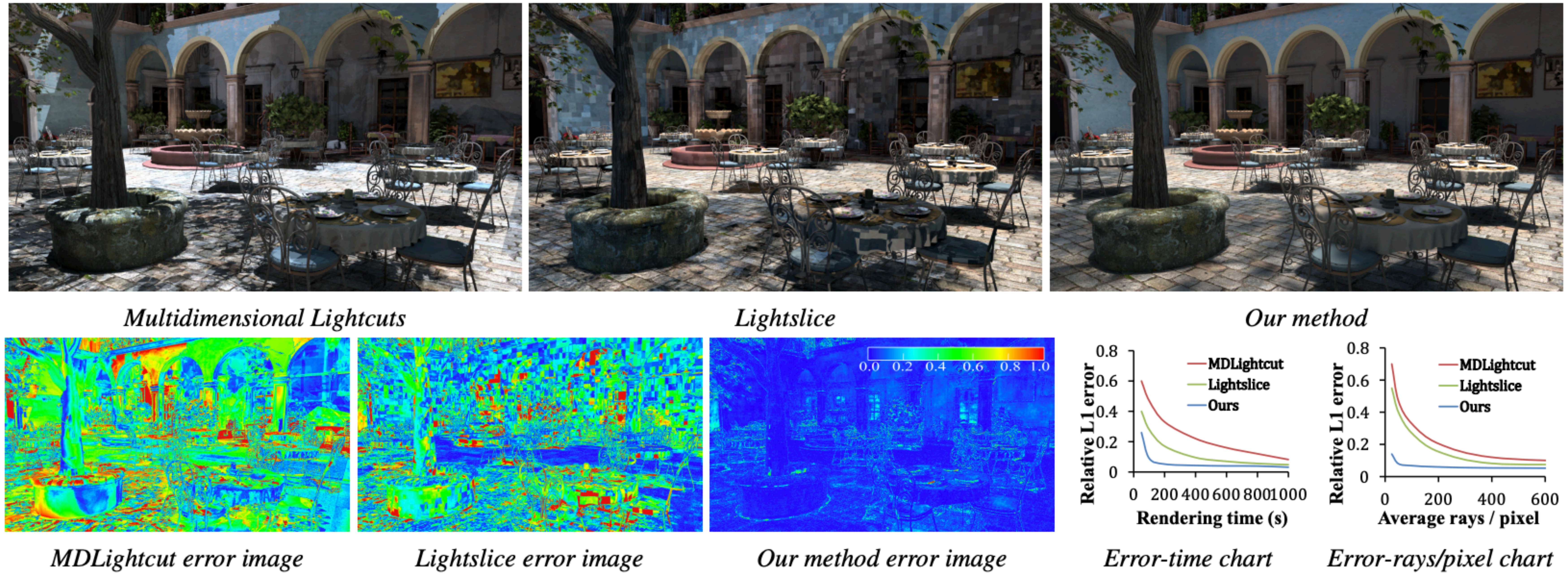
388k triangles: 432 rows, 864 columns, 13.5 s

869k triangles: 100 rows, 200 columns, 3.8 s

# Followup: applying matrix completion algorithms for light transport matrix estimation

## A Matrix Sampling-and-Recovery Approach for Many-Lights Rendering

Yuchi Huo Rui Wang* Shihao Jin Xinguo Liu Hujun Bao*
State Key Lab of CAD&CG, Zhejiang University

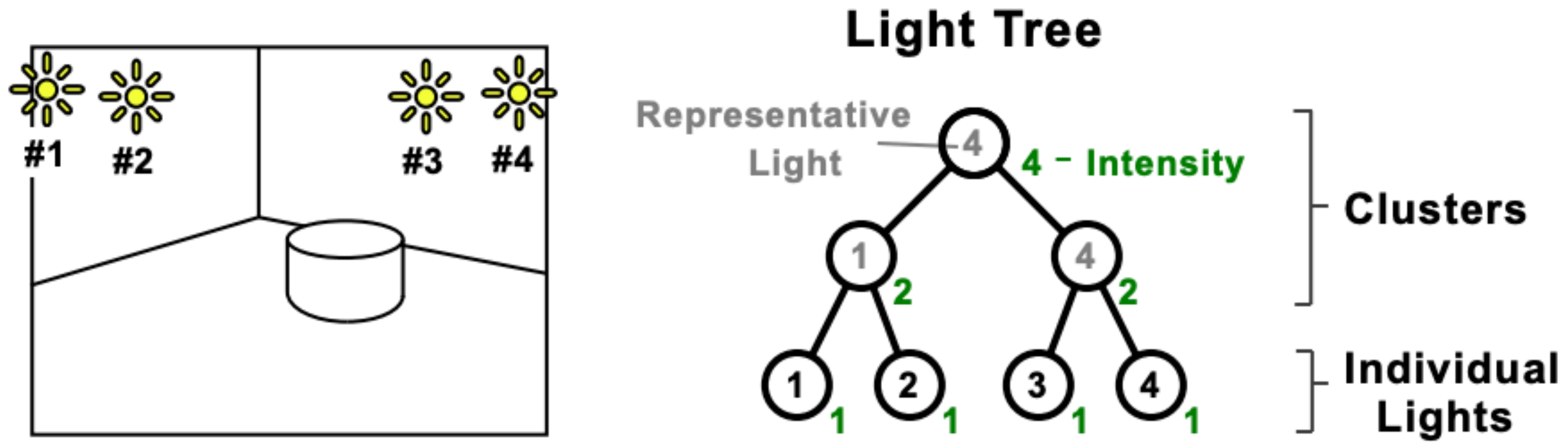



**Matrix Recovery by Matrix Separation.** Matrix separation has been recently developed [Candès et al. 2011; Shen et al. 2014]. Specifically in our scenario, the reduced lighting matrix $\mathbf{L}$ can be separated from the corrupted matrix $\mathbf{D}$ with a sparse error matrix $\mathbf{Z}$, $\mathbf{D} = \mathbf{L} + \mathbf{Z}$, by solving the following minimization:

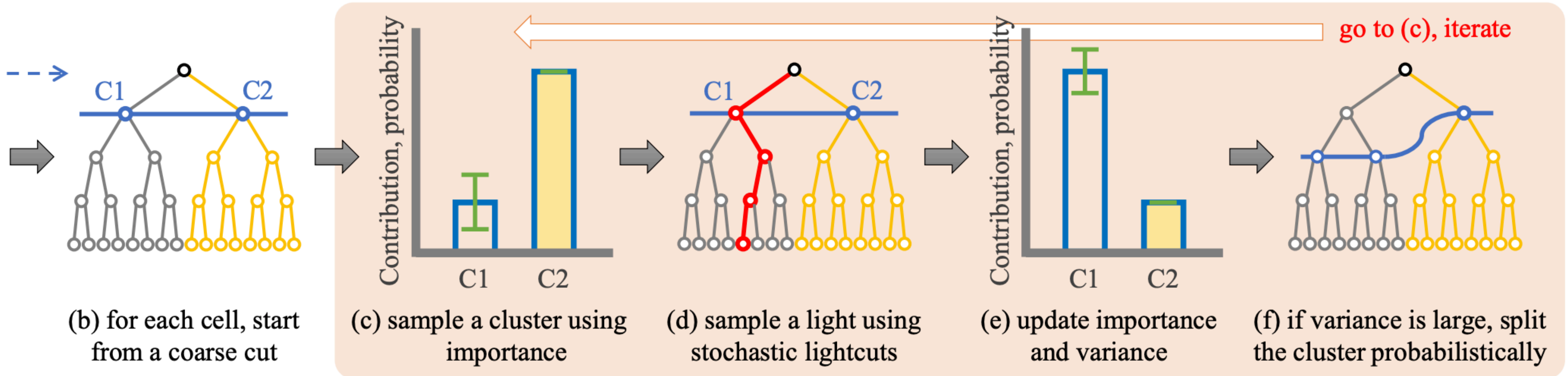
$$\begin{aligned} \min_{\mathbf{L},\mathbf{Z}} \quad & \|\mathbf{L}\|_* + \lambda\|\mathbf{Z}\|_1 \\ s.t. \quad & P_\Omega(\mathbf{L}+\mathbf{Z}) = P_\Omega(\mathbf{D}) \end{aligned} \tag{5}$$

# Ideas

[Walter 2005]

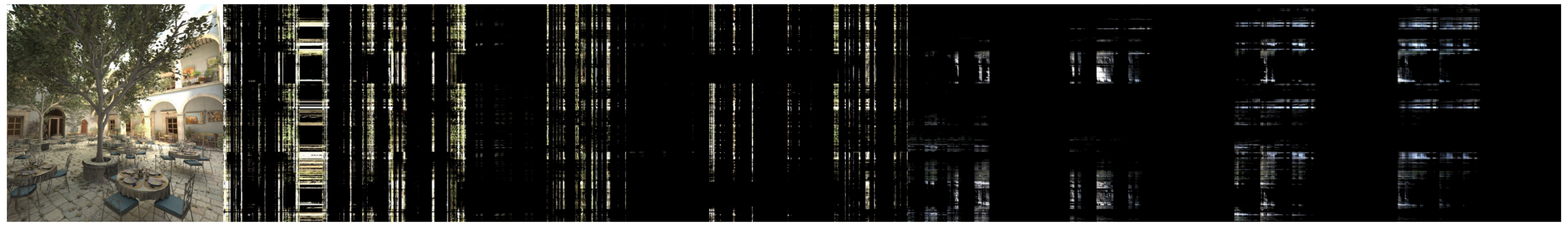



hierarchical clustering
[Shirley 1996,
Paquette 1998,
Walter 2005, …]

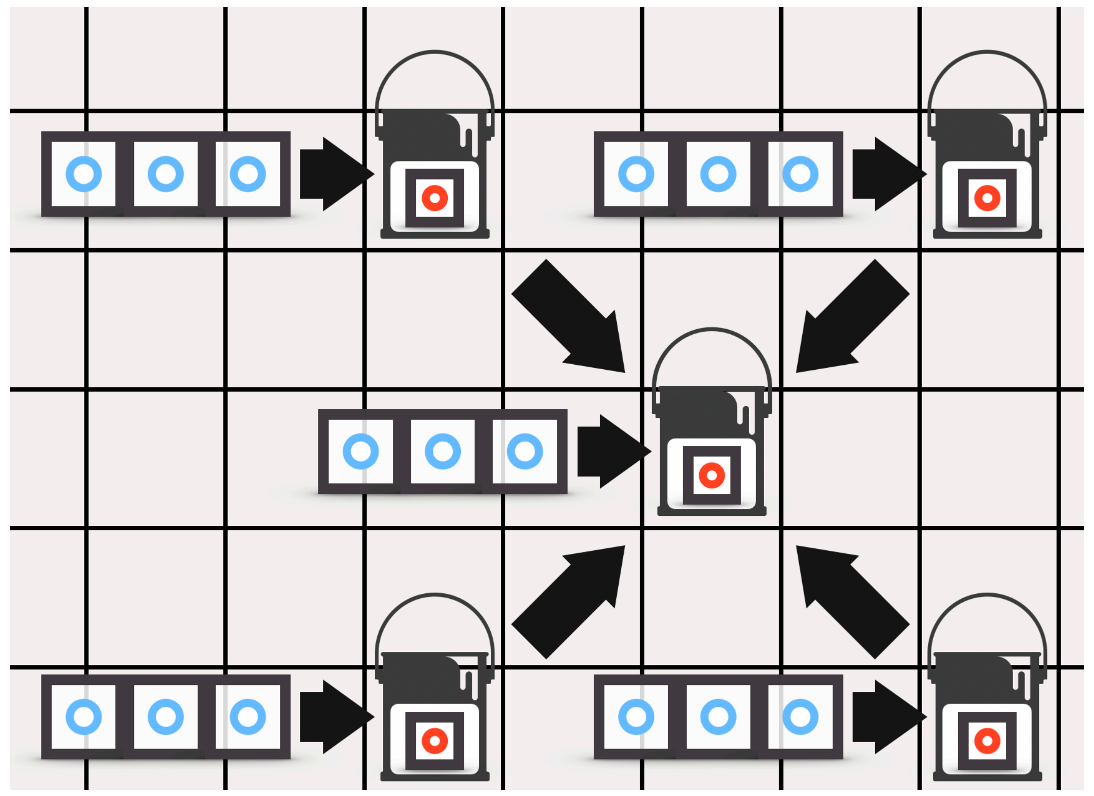



data-driven
[Donikian 2006,
Vevoda 2018,
Wang 2021]

[Ou 2011]



matrix formulation
[Hasan 2007,
Ou 2011,
Huo 2015, ….]

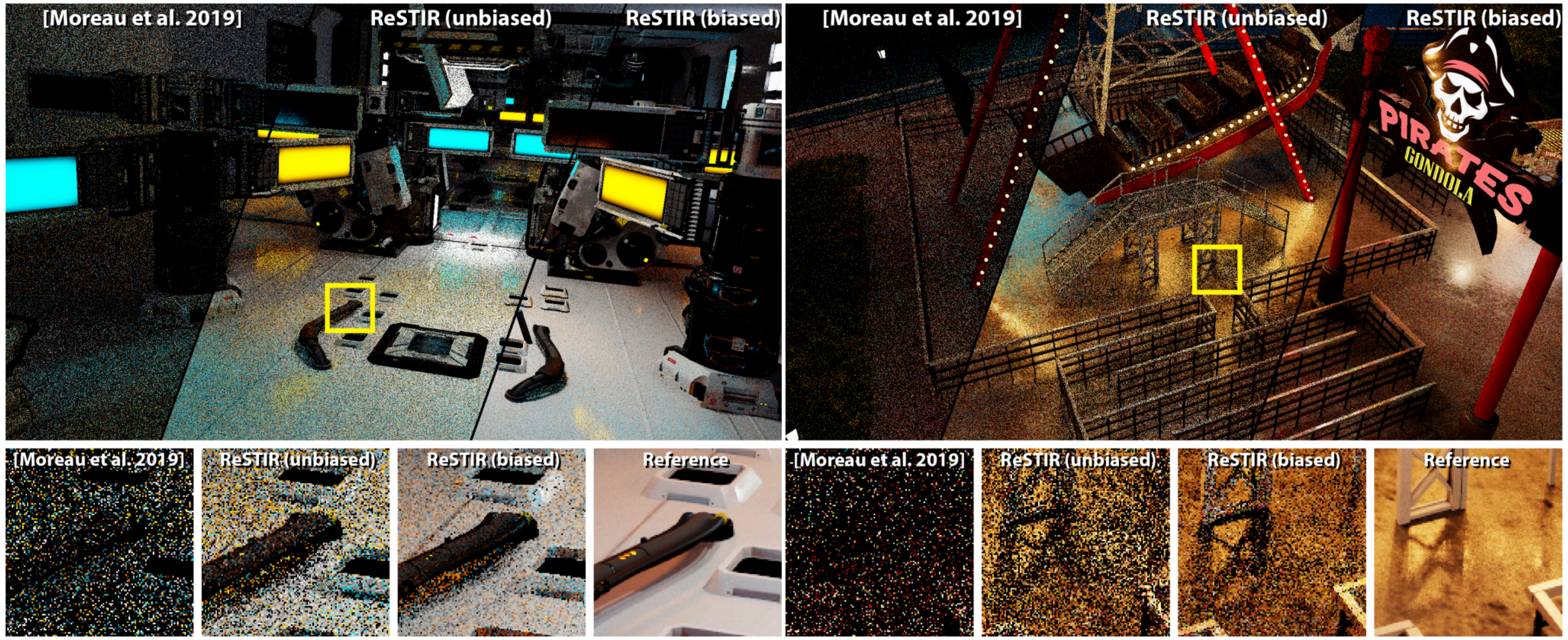


spatial-temporal reuse +
resampling
[Benedikt 2020]

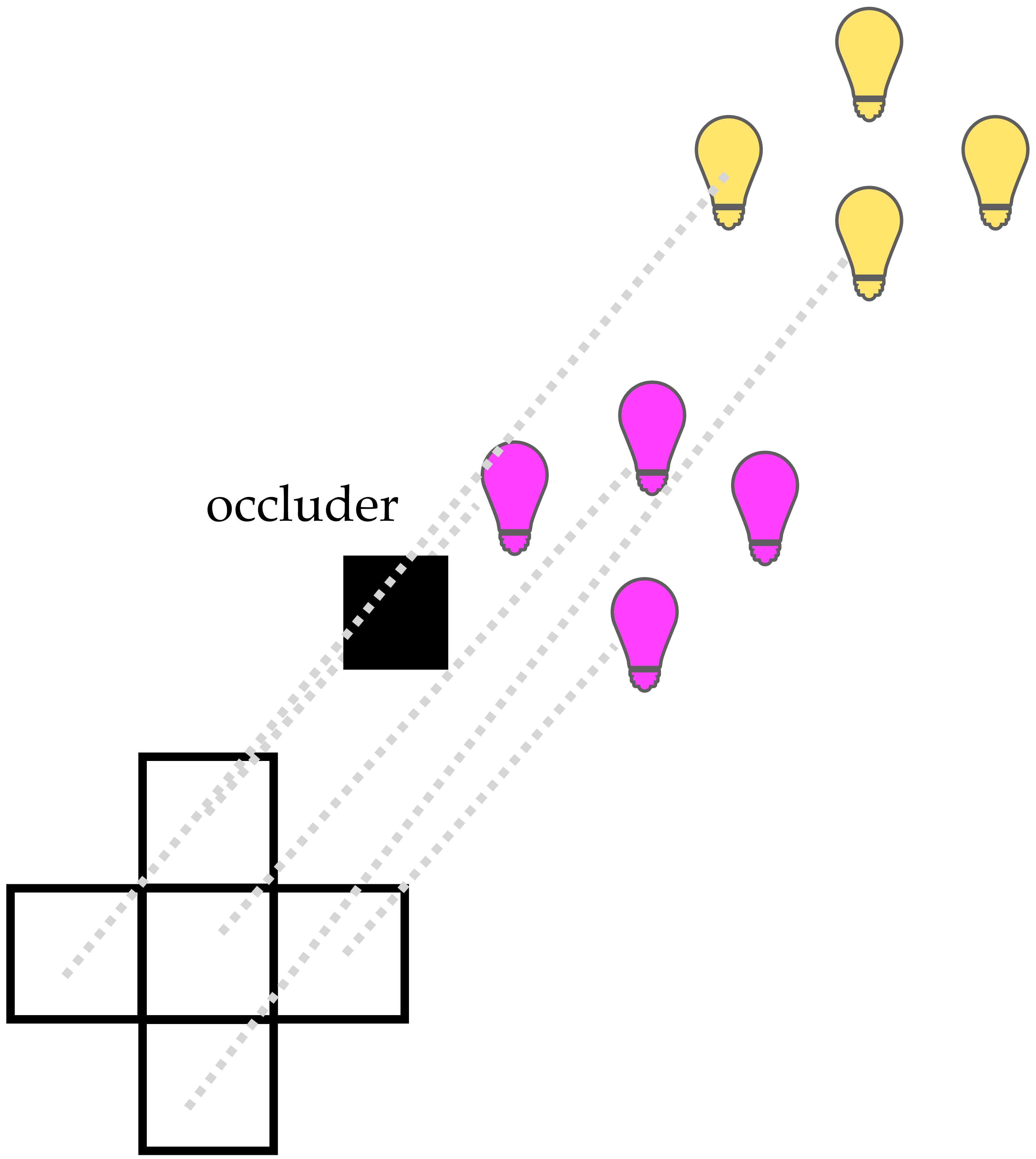
# Motivation: real-time rendering

Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting

BENEDIKT BITTERLI, Dartmouth College
CHRIS WYMAN, NVIDIA
MATT PHARR, NVIDIA
PETER SHIRLEY, NVIDIA
AARON LEFOHN, NVIDIA
WOJCIECH JAROSZ, Dartmouth College

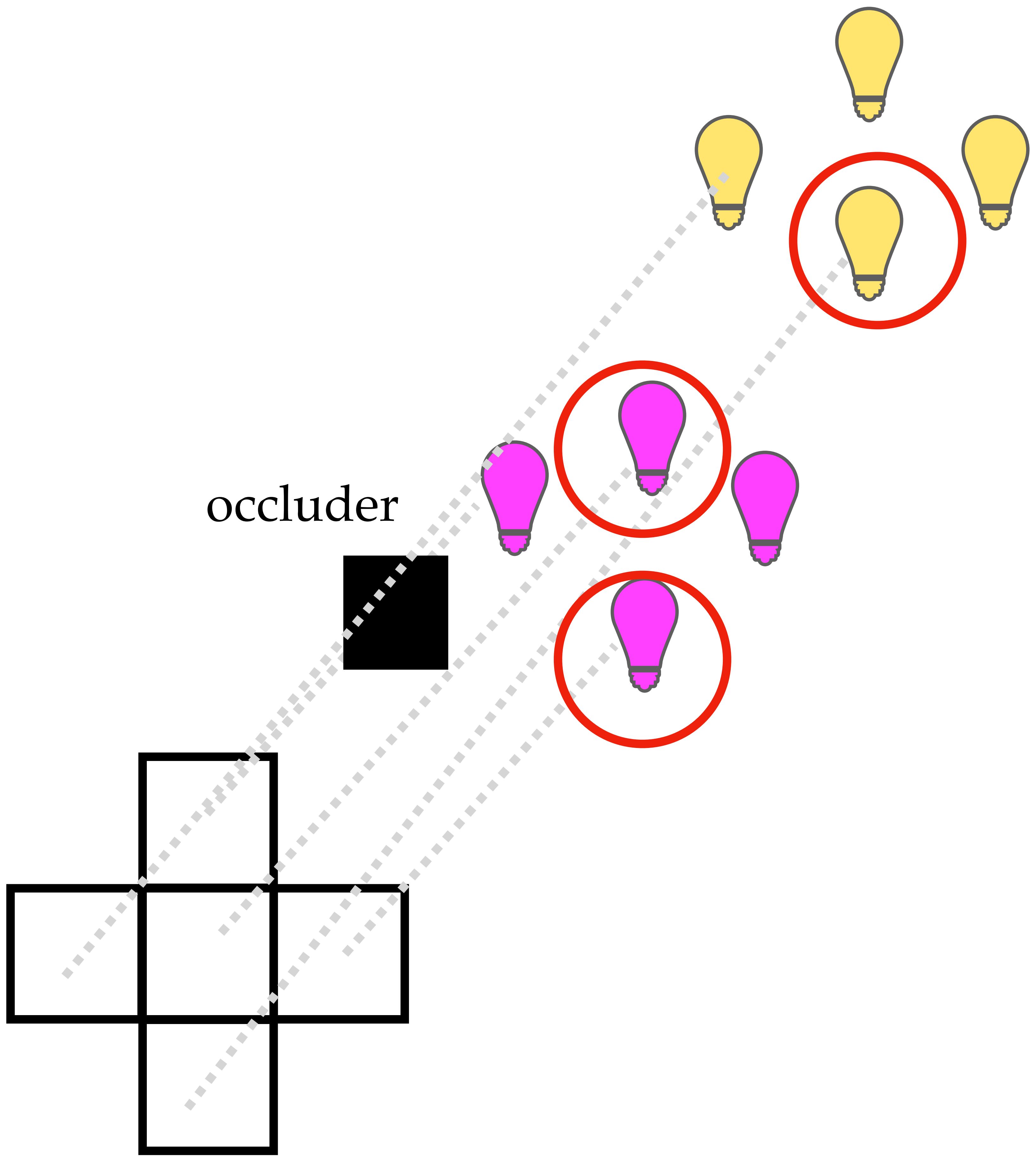


# Idea: reuse neighboring pixels' sampling results



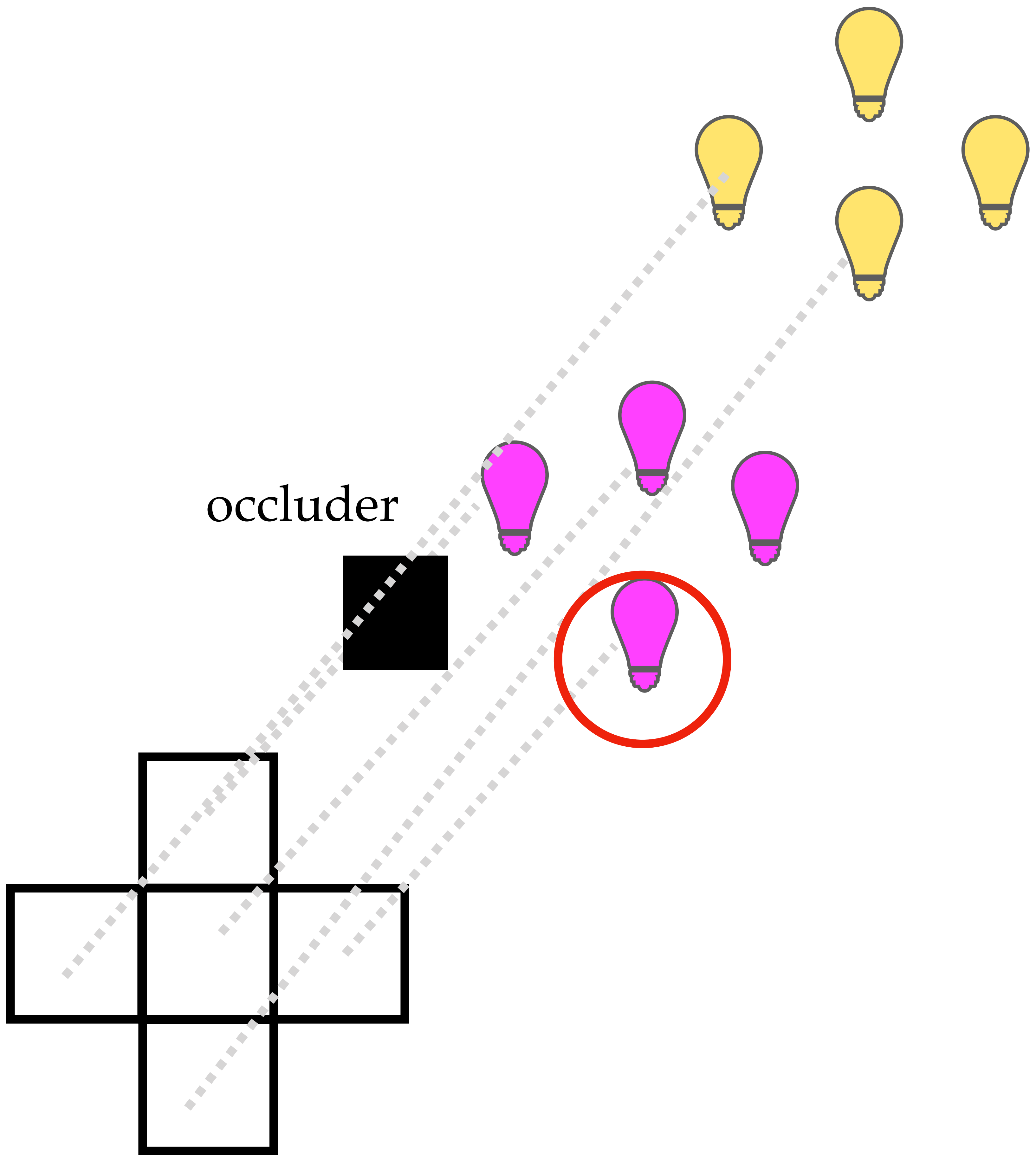

- each pixel starts with a single light sampled (can use lightcuts or whatever)

# Idea: reuse neighboring pixels' sampling results



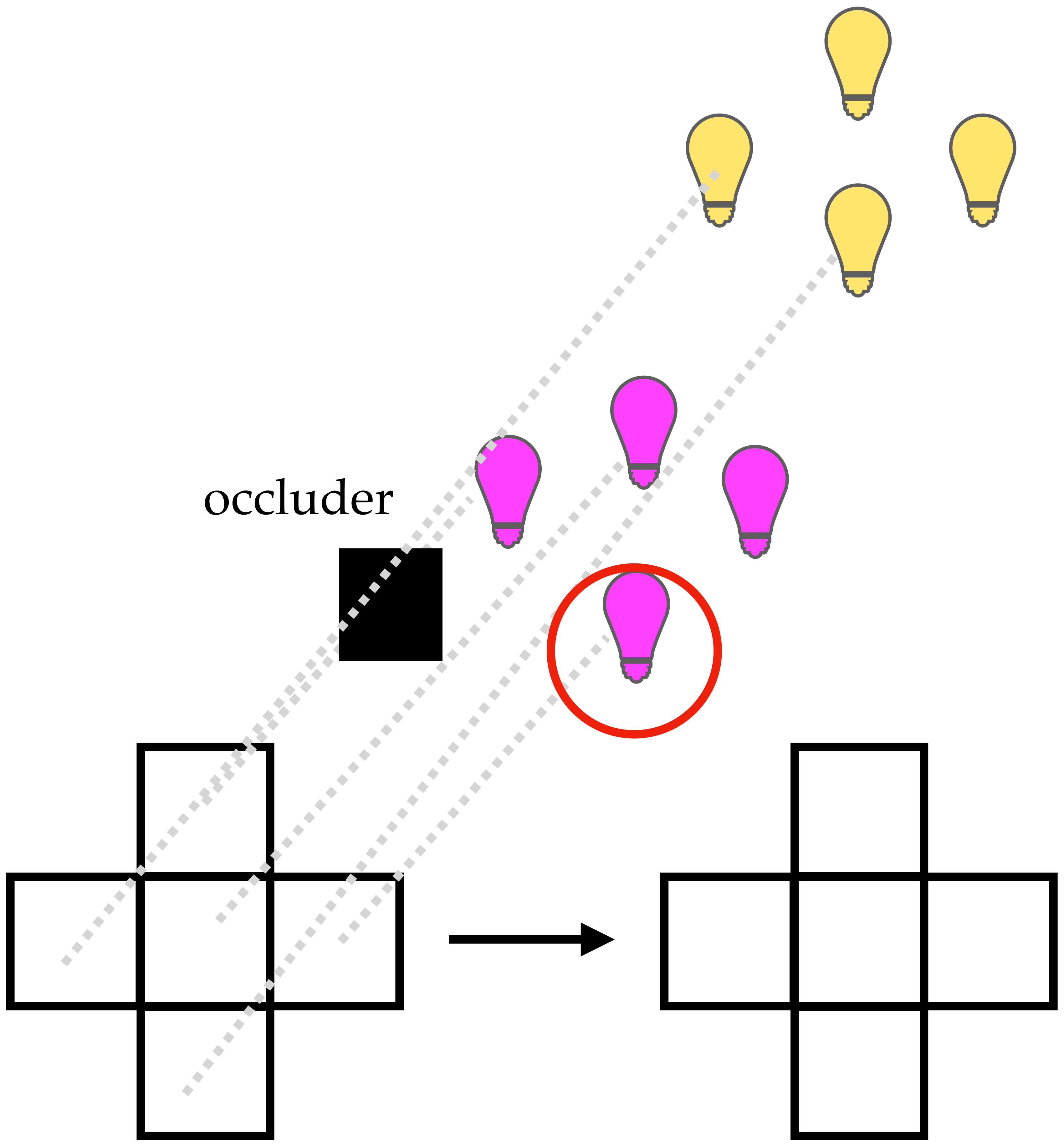

- each pixel starts with a single light sampled (can use lightcuts or whatever)
- for the center pixel, pick the unoccluded lights from neighbor pixels

# Idea: reuse neighboring pixels' sampling results




- each pixel starts with a single light sampled (can use lightcuts or whatever)
- for the center pixel, pick the unoccluded lights from neighbor pixels
- sample from these lights using probability proportional to $L \cdot \rho \cdot G$

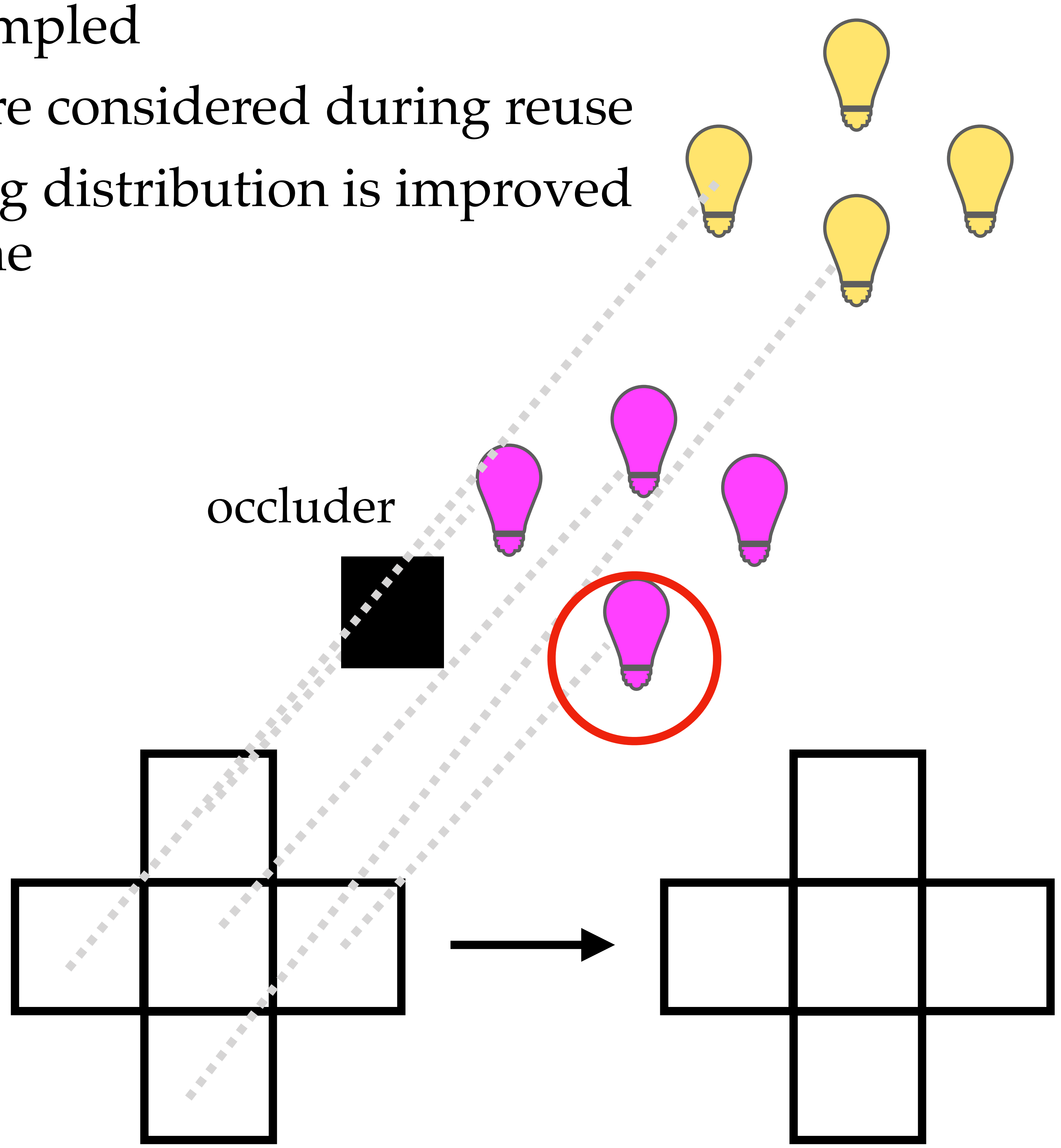
# Idea: reuse neighboring pixels' sampling results




- each pixel starts with a single light sampled (can use lightcuts or whatever)
- for the center pixel, pick the unoccluded lights from neighbor pixels
- sample from these lights using probability proportional to $L \cdot \rho \cdot G$
- can propagate the information to the next frame

# Idea: reuse neighboring pixels' sampling results

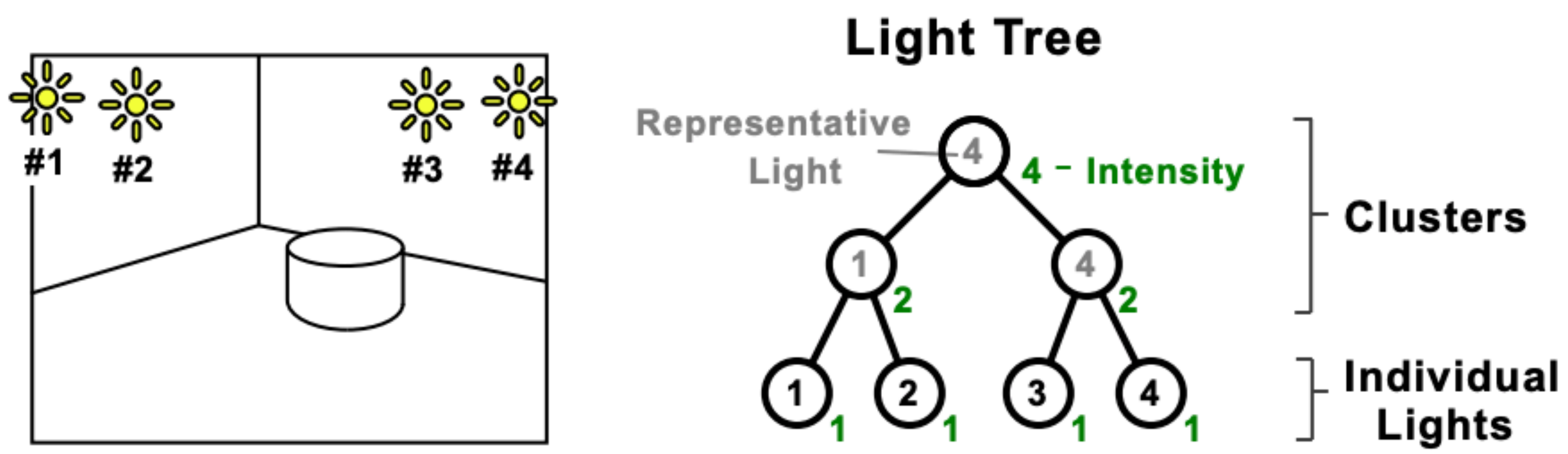
- benefits
  - occluded lights have low probability to be sampled
  - $\rho$ & G are considered during reuse
  - sampling distribution is improved over time




- each pixel starts with a single light sampled (can use lightcuts or whatever)
- for the center pixel, pick the unoccluded lights from neighbor pixels
- sample from these lights using probability proportional to $L \cdot \rho \cdot G$
- can propagate the information to the next frame

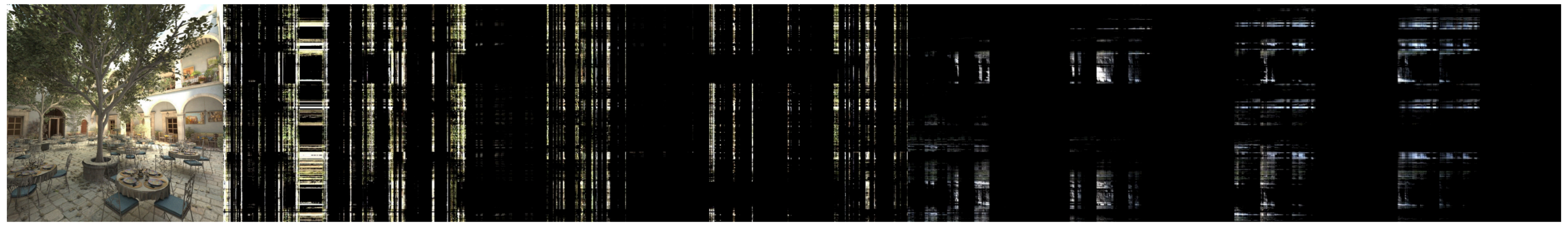
# What are the connections between these ideas?

[Walter 2005]

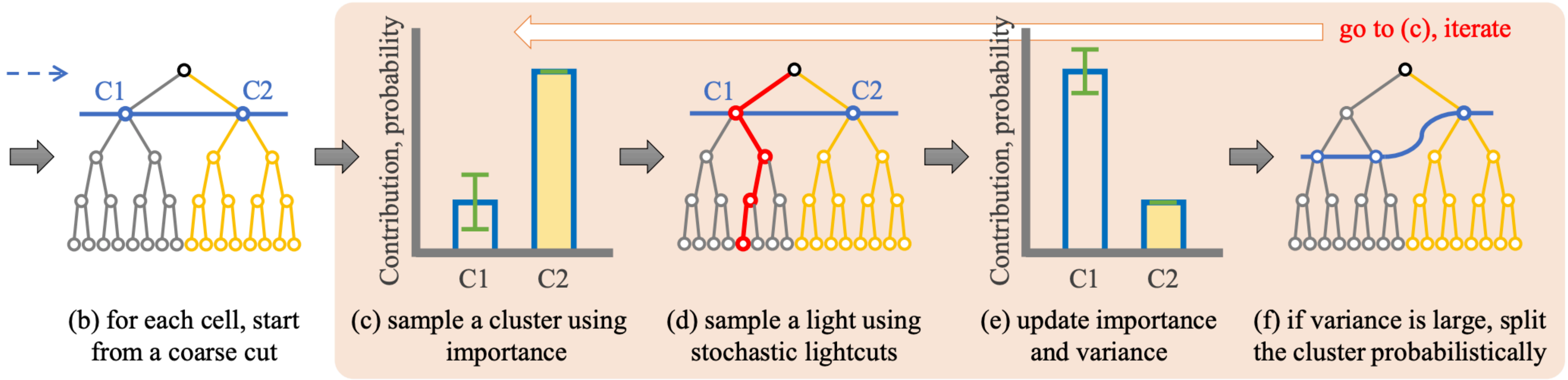



hierarchical clustering
[Shirley 1996,
Paquette 1998,
Walter 2005, …]

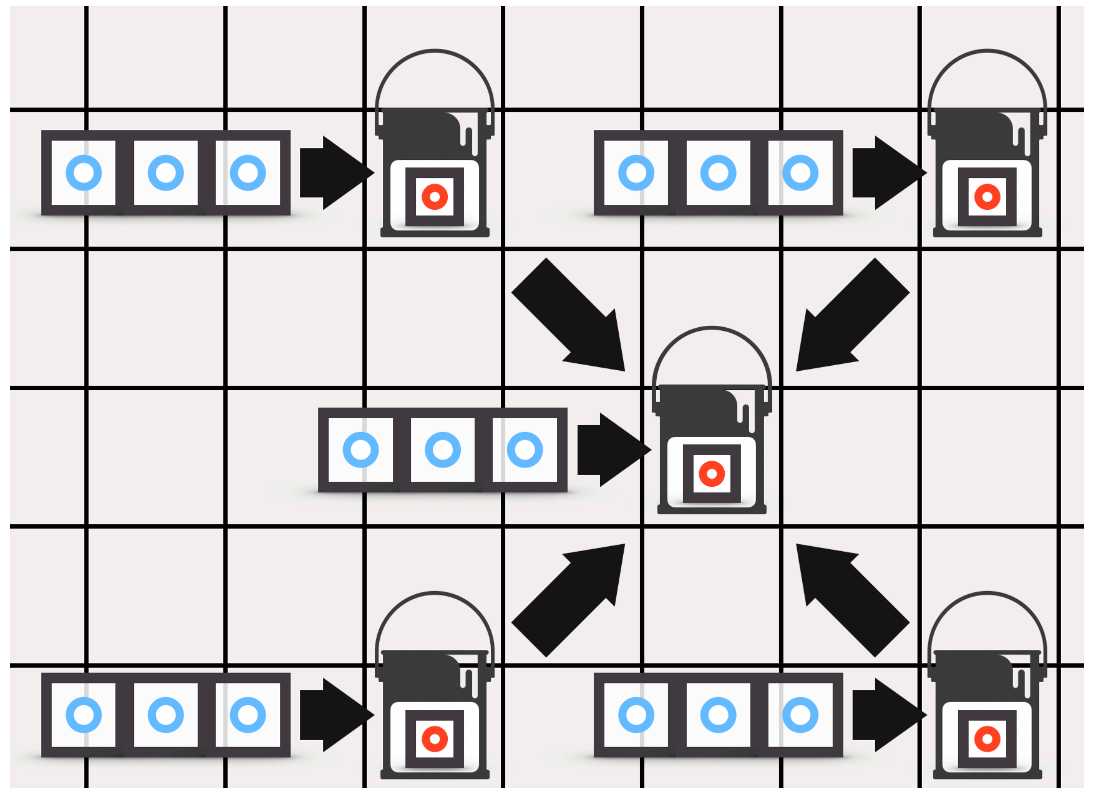
[Ou 2011]



matrix formulation
[Hasan 2007,
Ou 2011,
Huo 2015, ….]




data-driven
[Donikian 2006,
Vevoda 2018,
Wang 2021]



spatial-temporal reuse +
resampling
[Benedikt 2020]

# Next: ReSTIR and Path-reusing