Importance Sampling of Virtual Point Lights

Eurographics 2010

short paper

Motivation

- Instant Radiosity (IR) – two-pass
  - Cheap pre-processing
  - Expensive rendering
- Previous approaches
  - Bidirectional/Metropolis Instant Radiosity [Segovia et al.]
    - Difficult to implement
    - Multiple sampling strategies
    - Many parameters
    - Difficult to stratify
  - "One-pixel image" assumption

Our method

- Simple extension of IR
  - Generate VPLs from light sources only
- Probabilistically accept VPLs
  - Proportionally to total contribution
  - All VPLs bring the same power to the image
  - ⇨ "One-pixel image" assumption
- Minimum importance storage
  - Filter VPLs on the fly

Probabilistic VPL acceptance

- VPL energy

$$L_i = \frac{L_i}{p_i} p_i = \frac{L_i}{p_i} \int_0^1 \chi_{[0,p_i]}(t)dt$$

- One-sample Monte Carlo integration with ξ

$$\widehat{L_i} = \begin{cases} \frac{L_i}{p_i}, & \xi < p_i \\ 0, & \text{else} \end{cases}$$

- Allows to control VPL density

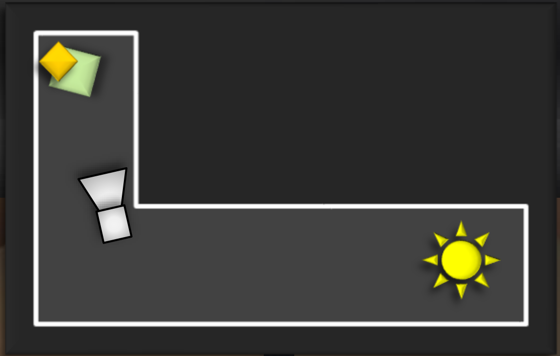
Choosing the acceptance probability

- Want N VPLs with equal total contribution
  - $\Phi_v = \frac{\Phi}{N}$
- For each VPL candidate i with energy L_i
  - Estimate total contribution Φ_i
  - Russian roulette decision with $p_i = \min\left(\frac{\Phi_i}{\Phi_v} + \varepsilon_\mathrm{p}, 1\right)$
    - Accept with energy $\frac{L_i}{p_i}$
    - Discard

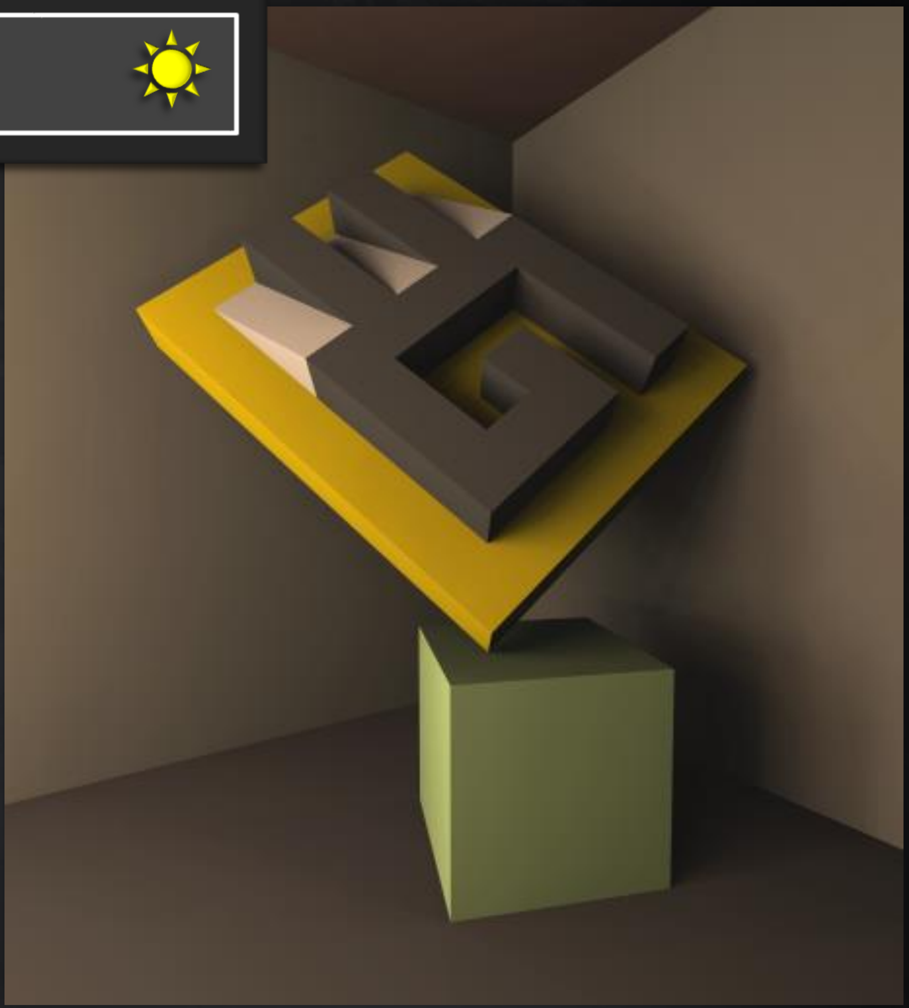
Estimating Image Contribution

- Computing Φ_i
  - Create a number of samples from camera rays
    - Analogs of importons
  - Connect VPLs to camera samples
- Computing Φ
  - Progressively
    - Set $\Phi = 0$
    - Loop
      - Render frame, compute Φ^i
      - Accumulate $\Phi = \left(1 - \frac{1}{i}\right)\Phi + \frac{1}{i}\Phi^i$
  - In a single pass – path tracing, using VPLs, etc.

Results



Instant Radiosity

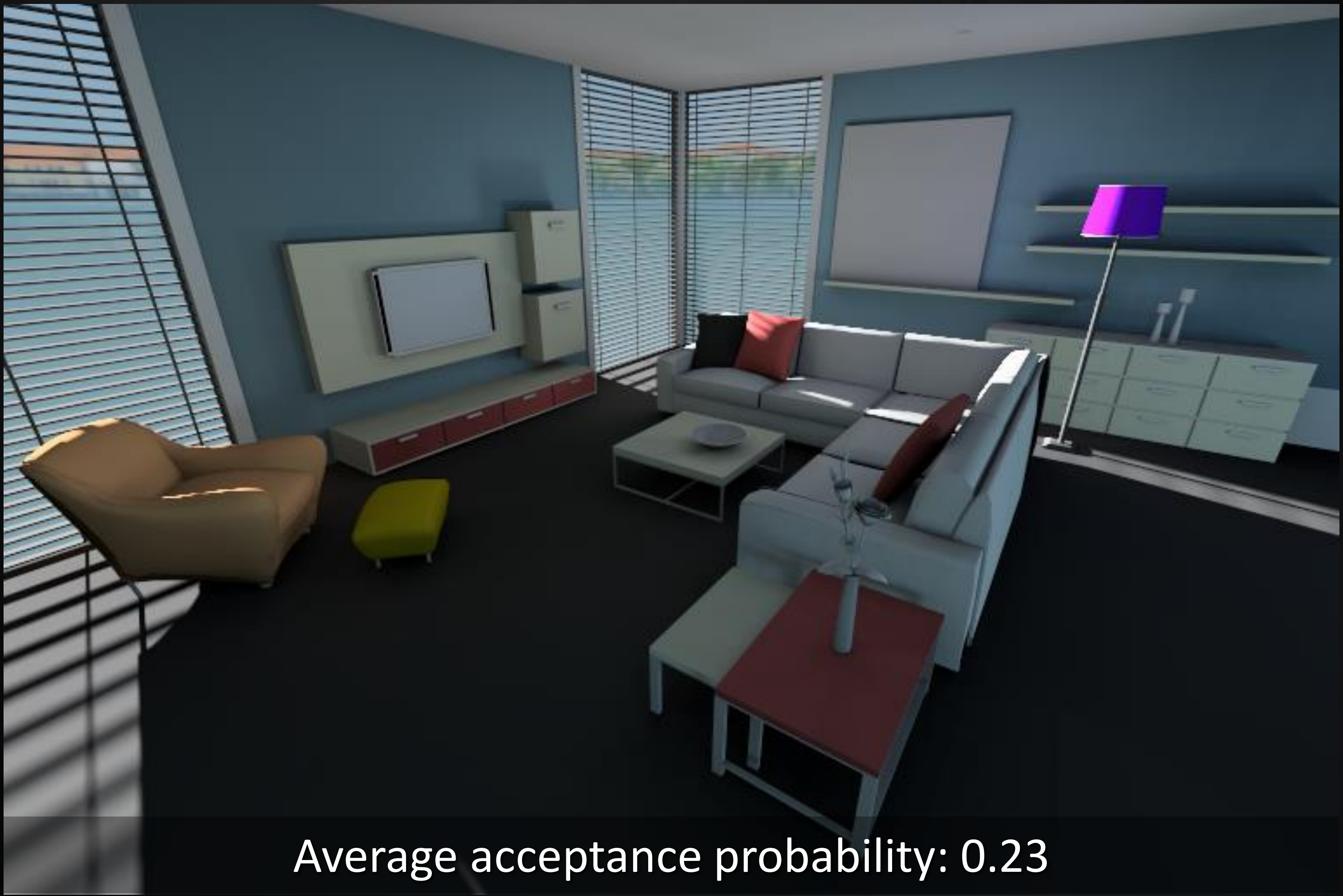


Our Extension (0.07 acceptance)

Results

Average acceptance probability: 0.28

Results



Average acceptance probability: 0.23

Wrap Up

- Simple extension of IR
  - Generate VPLs from light sources only
- Probabilistically accept VPLs on the fly
  - Fixed minimal additional storage
  - Easy to parallelize
- Two parameters
  - $\varepsilon_{\mathrm{p}} = 0.05$
  - Number of camera samples, e.g. 100
- “One-pixel image” assumption

UNIVERSITÄT DES SAARLANDES

intel

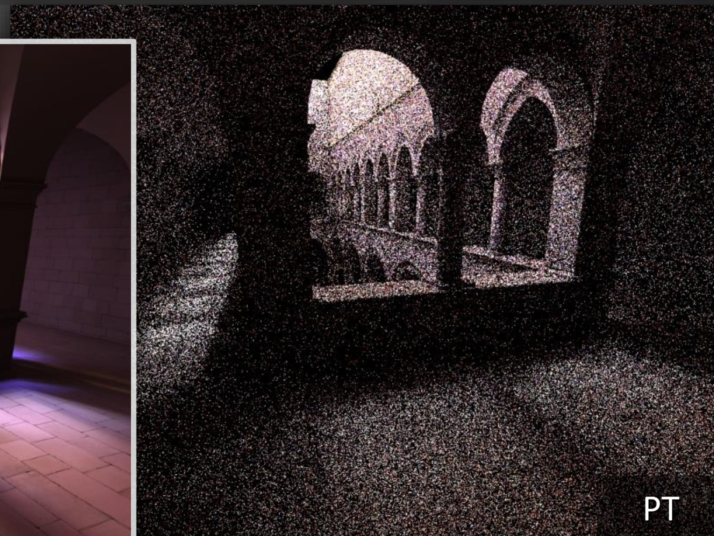
VISUAL COMPUTING INSTITUTE

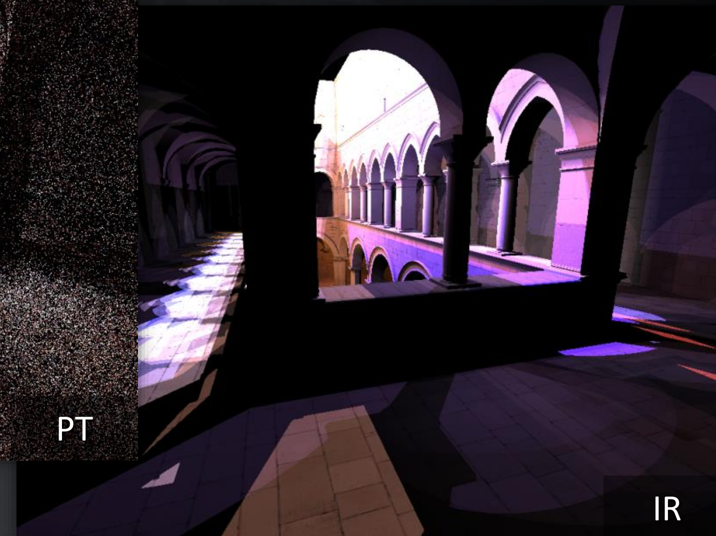
Importance Caching for Complex Illumination

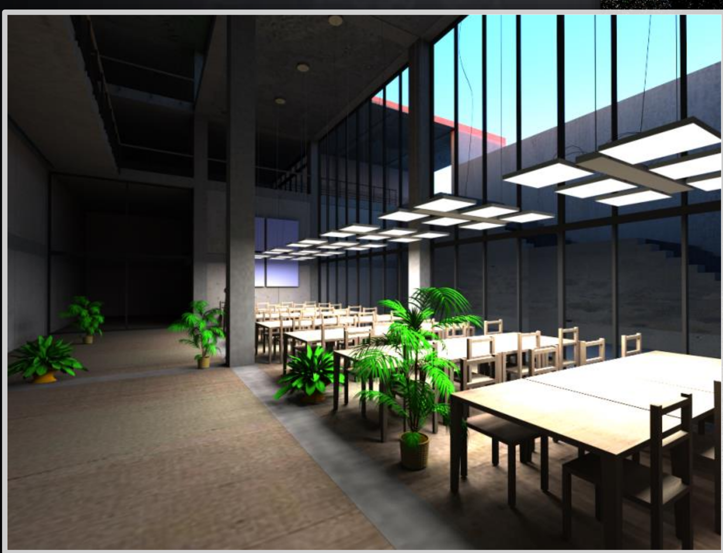
Eurographics 2012

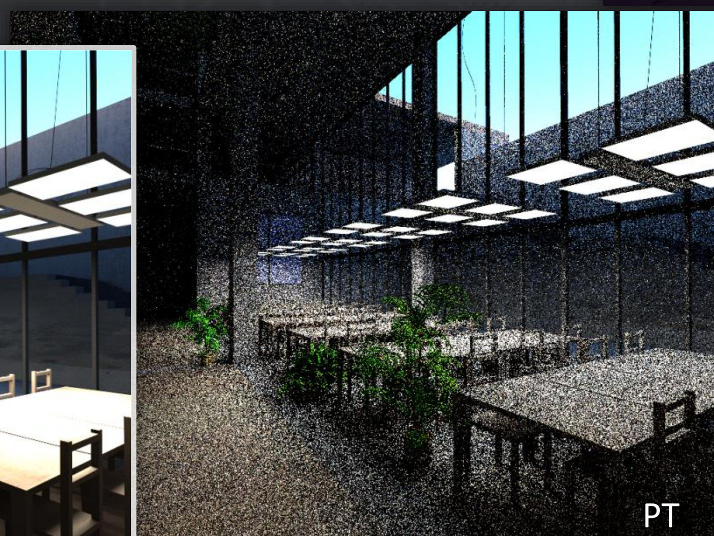
full paper

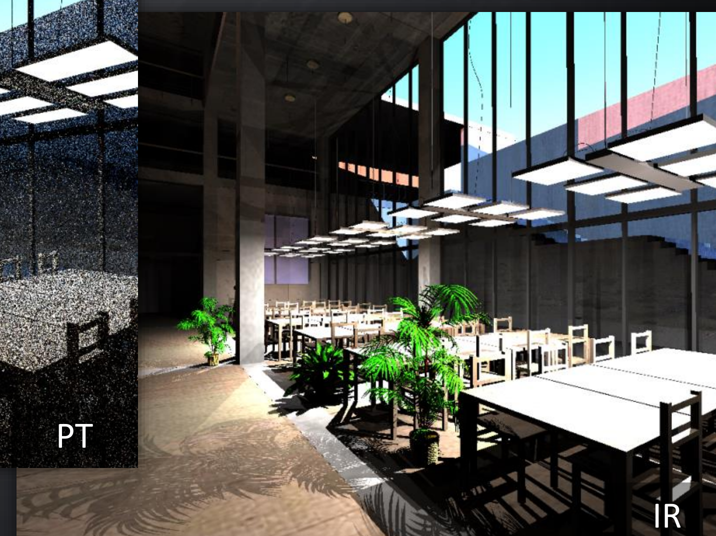
Motivation











Motivation

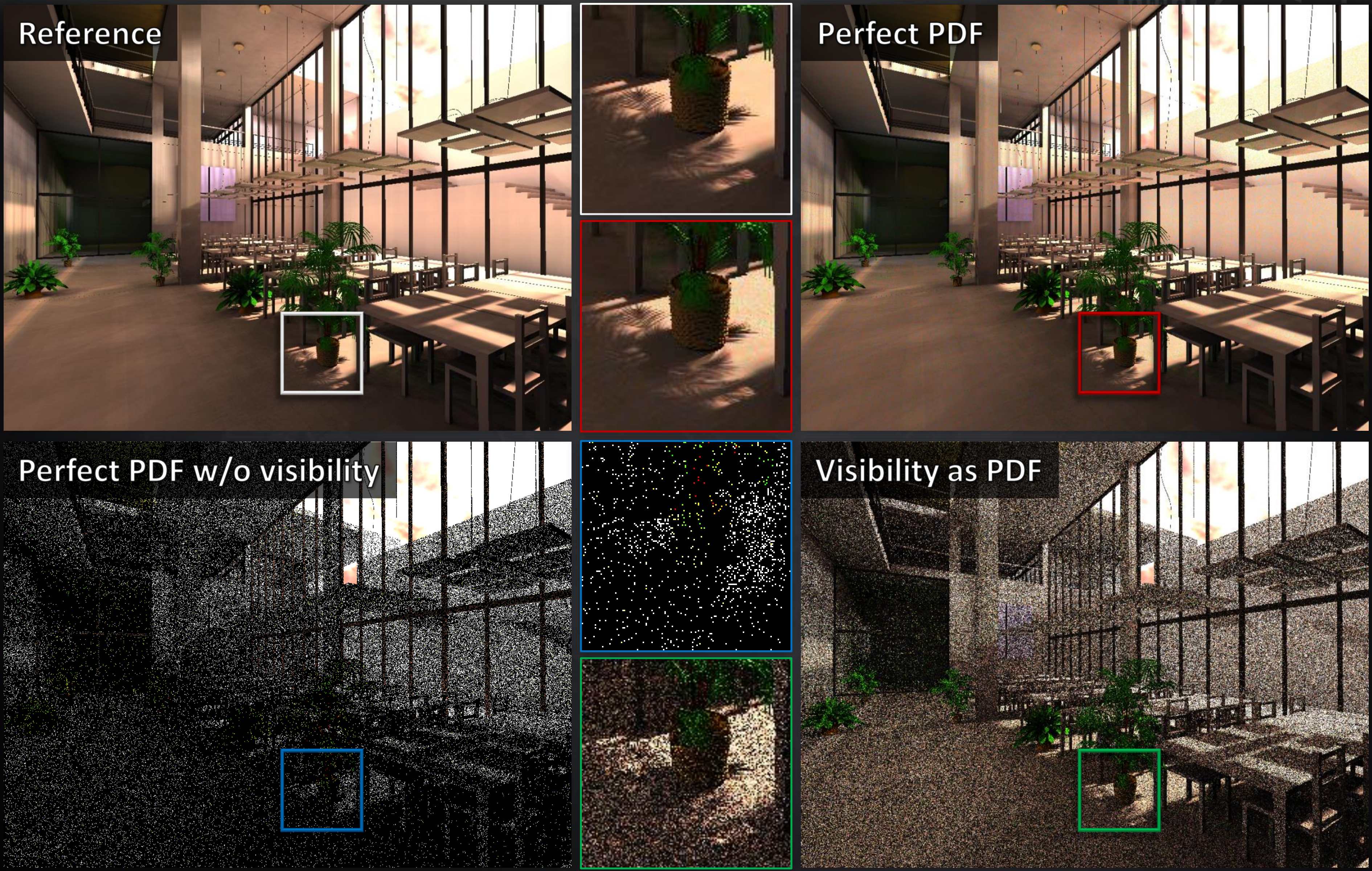
- Global illumination still very costly
  - Indirect illumination
  - Even direct illumination – environment, area lights
- Two basic algorithmic improvements
  - Importance sampling
    - Better sample distribution (ideally proportional to integrand)
    - Higher quality with fewer samples
  - Exploiting coherence
    - Pixel integrands are often highly correlated
    - Amortize sampling effort among pixels
    - Fast!

Background

Importance Sampling

- Global – virtual point lights (VPLs)
  - Importance-driven sample generation/filtering
    - Find relevant VPLs for the current view point (one-pixel image)
  - ✓ Fast – few VPLs
  - ✗ Suboptimal – VPL importance varies across pixels
- Local (per pixel)
  - Construct product PDF specialized for integrand
  - ✓ Robust – PDF often matches integrand well
  - ✗ Not in the presence of occlusion
  - ✗ Costly – per-pixel PDF construction (BRDF pre-processing)

Motivation (Single Sample per Pixel)



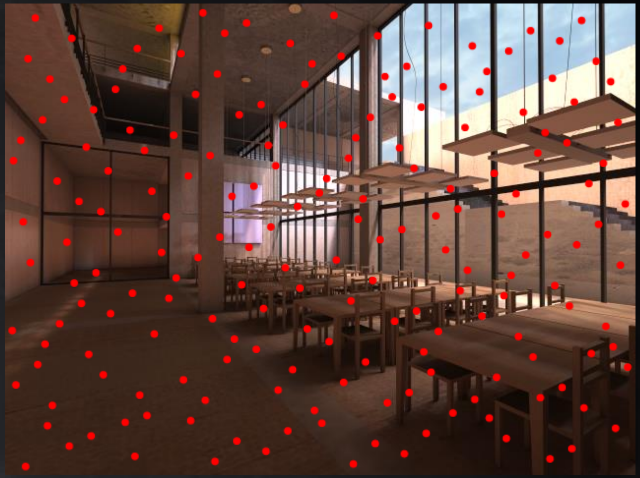
Background

Exploiting Coherence

- Illumination is often smooth
  - Especially indirect
  - ➢ Correlated pixel integrals
- Filtering
  - Idea – share samples among integrals
  - Reuse samples by interpolation/filtering
    - Irradiance caching, photon mapping
    - Preserve discontinuities
  - Smooth, low-variance results
  - Biased, smeared edges → indirect only
  - Slow convergence, increased memory usage

Images by J. Křivánek and P. Gautron

Algorithm Overview



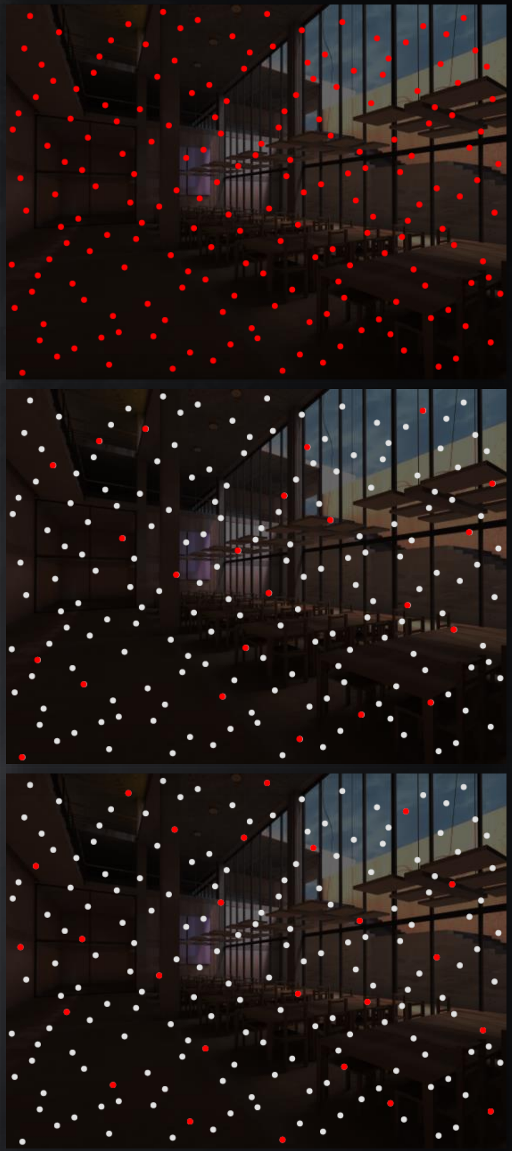
- Idea – combine all three
  - Unbiased VPL sampling framework
  - Shade only few most relevant VPLs
- Approach
  - Consider full integrand (w/ visibility)
  - Shade all VPLs at few locations
  - Reuse VPL evaluations as importance at other locations
- Issue – illumination discontinuities
  - Additional more conservative distributions
  - Efficient MIS combination at shading points

Algorithm Outline

- Progressive rendering
  - Interactive feedback, fixed-memory convergence
- For each frame
  1) Create importance records (IR) from camera
  2) Create virtual point lights (VPLs)
     - Probabilistic rejection (global)
  3) Store VPL distributions at each IR (local)
  4) Render
     - Borrow nearby IR distributions for VPL sampling (coherence)

Preprocess

- VPLs – on light sources and indirect
- IRs store VPL contributions
  - Accumulated during VPL generation
- Discard VPLs irrelevant for the image
  - Immediately after generation
  - Subset of IRs for contribution estimate
    - Halton sequence periodicity
- Accumulate VPL contribution to IRs



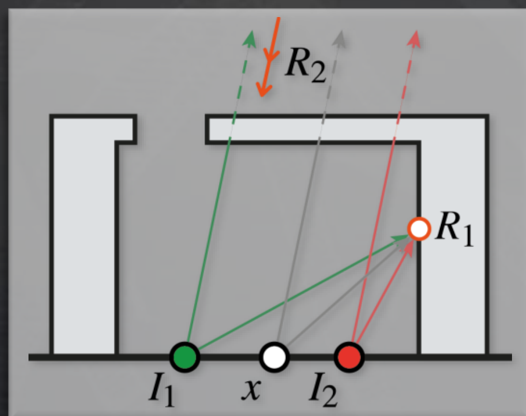
Rendering

- For each pixel shading point
  - Find nearest IRs
  - Use IR distributions defined for VPL sampling
- Robust sampling if at least one IR correlates
- Increased variance when all IRs irrelevant
  - Identify causes for VPL contribution changes
  - Additional, increasingly conservative distributions
- Many strategies – combine efficiently
  - Bilateral MIS combination framework

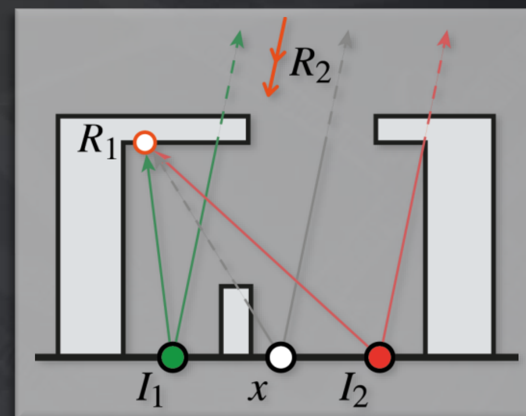
Sampling distributions

* Four sampling distributions at each IR

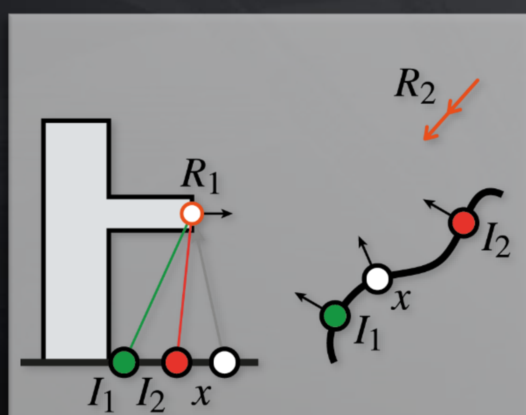
$\mathcal{F}$: Full



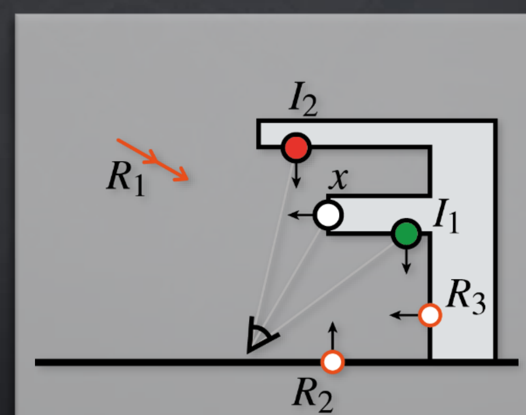
$\mathcal{U}$: Unoccluded



$\mathcal{B}$: Bounded



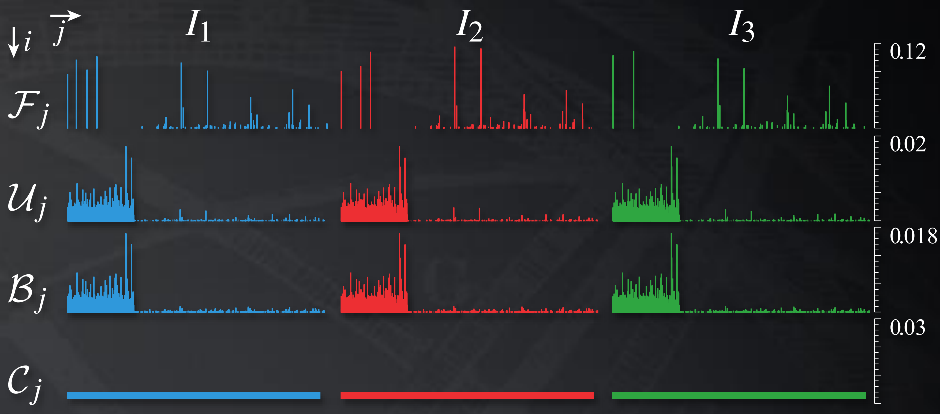
$\mathcal{C}$: Conservative



Distribution Combination

Horizontal Combination

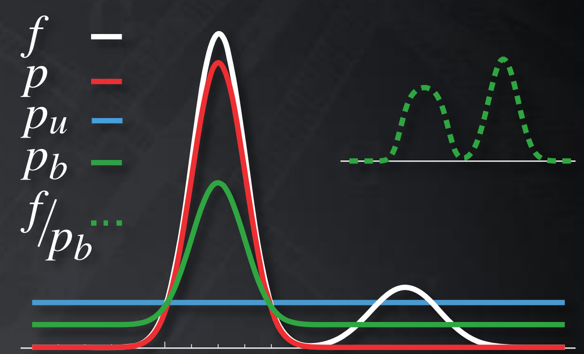
- Matrix structure
- Distributions often correlate among IRs
  - Combine first horizontally
    - Balance heuristic
    - Corresponds to mixture
      - Directly sample mixture
  - Collapse columns into one

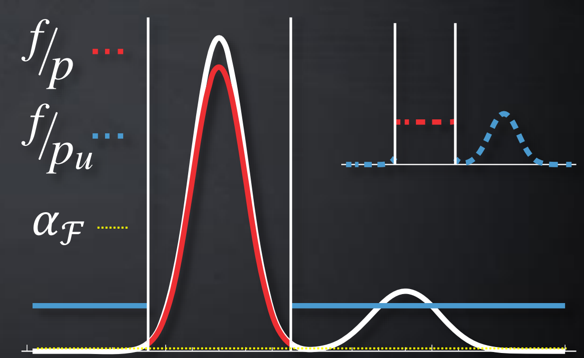


Distribution Combination

Vertical Combination

- Balance/power heuristics suboptimal
- Novel α-max combination heuristic
  - Prioritize distributions: $\mathcal{F}, \mathcal{U}, \mathcal{B}, \mathcal{C}$
  - Define confidences: $\alpha_{\mathcal{F}}, \alpha_{\mathcal{U}}, \alpha_{\mathcal{B}}, \alpha_{\mathcal{C}}$
  - Discard low-probability samples
    - If $p_{\mathcal{F}}(x) < \alpha_{\mathcal{U}} p_{\mathcal{U}}(x)$
- Distribution optimization
  - Apply heuristic at each IR
  - ➢ Exactly one distribution is non-zero for each VPL

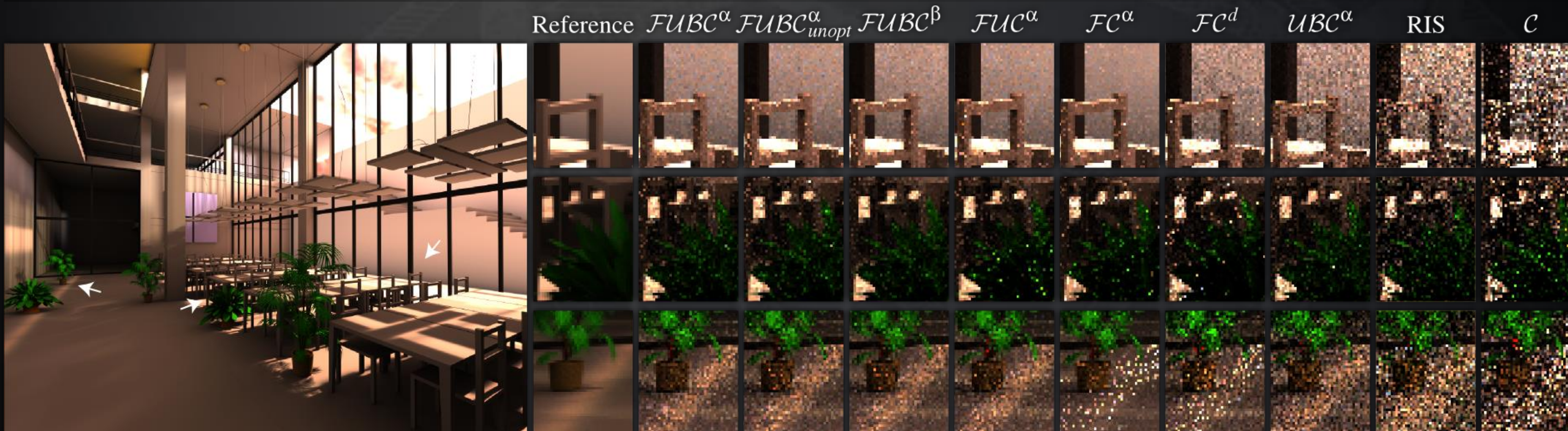




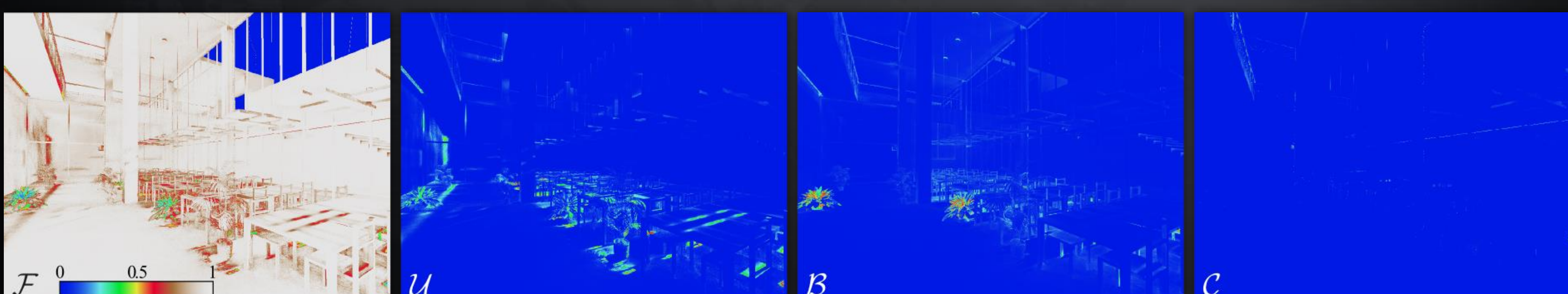
Results

Study Hall (diffuse)

Technique comparison

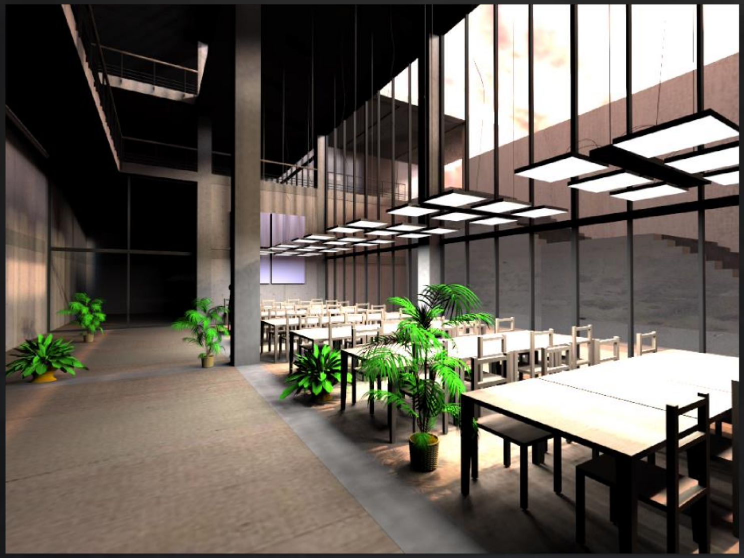


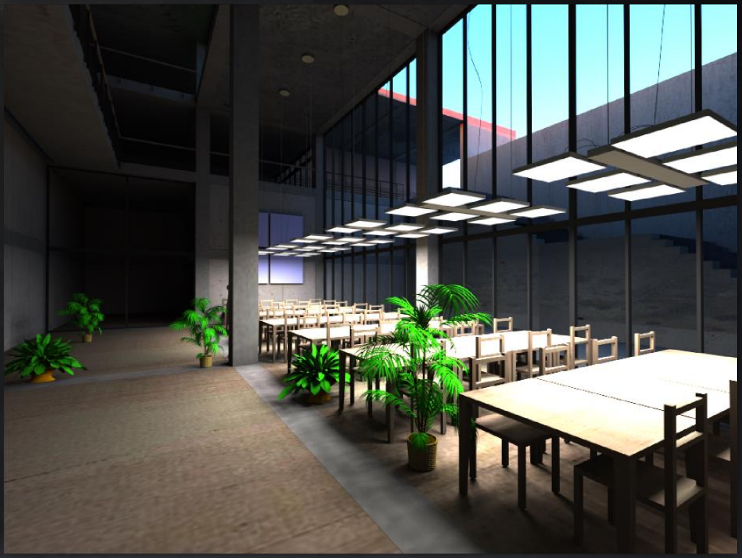
$\mathcal{FUBC}^{\alpha}$ fractional contributions

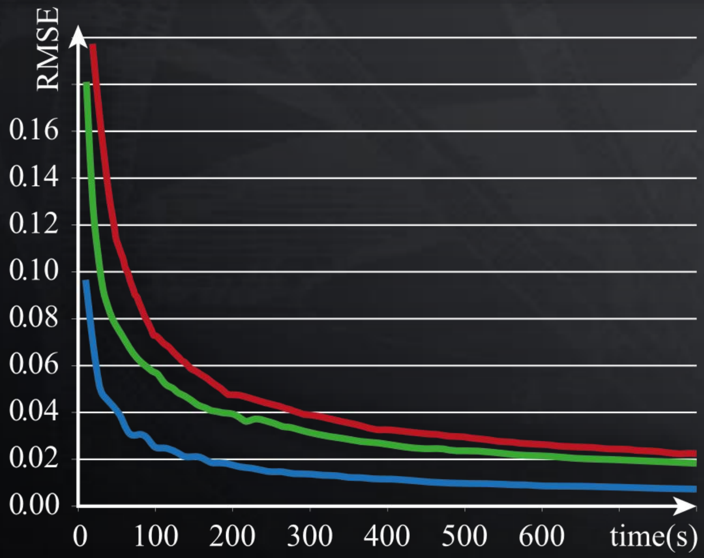


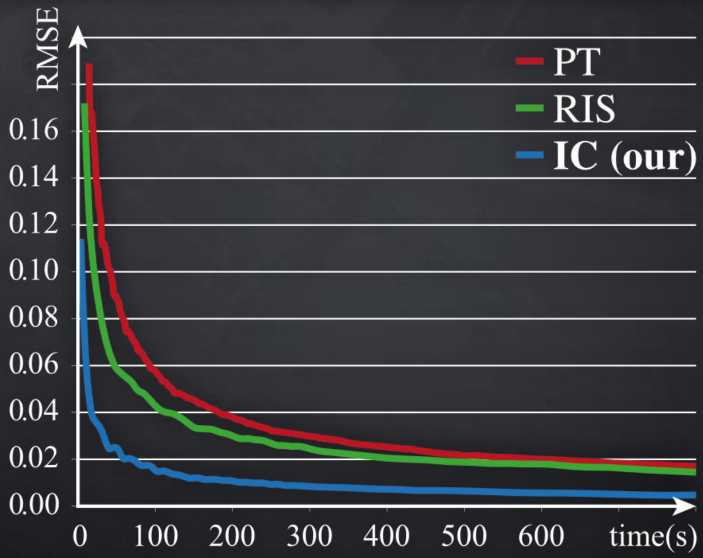
Results

Numerical tests



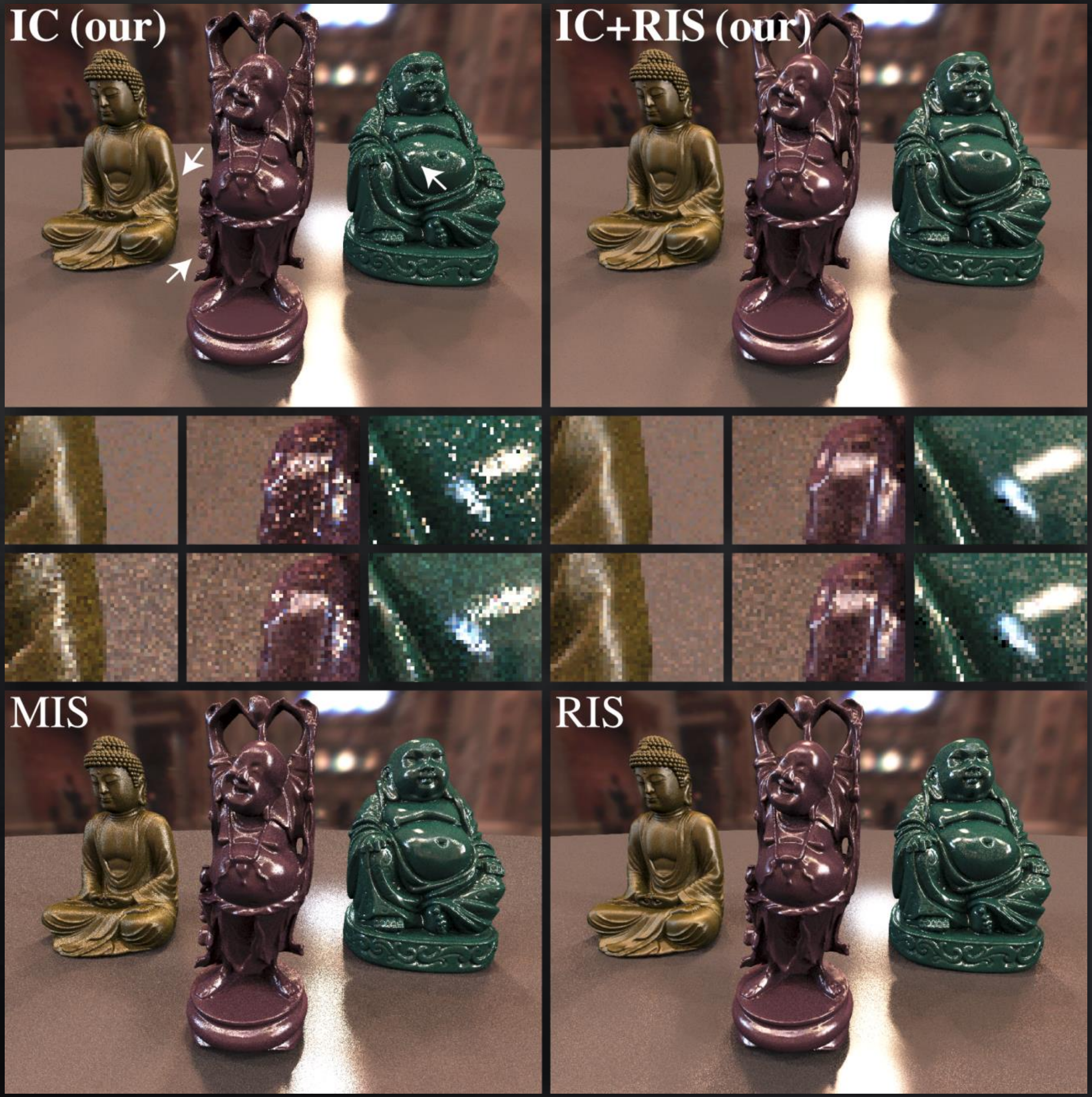






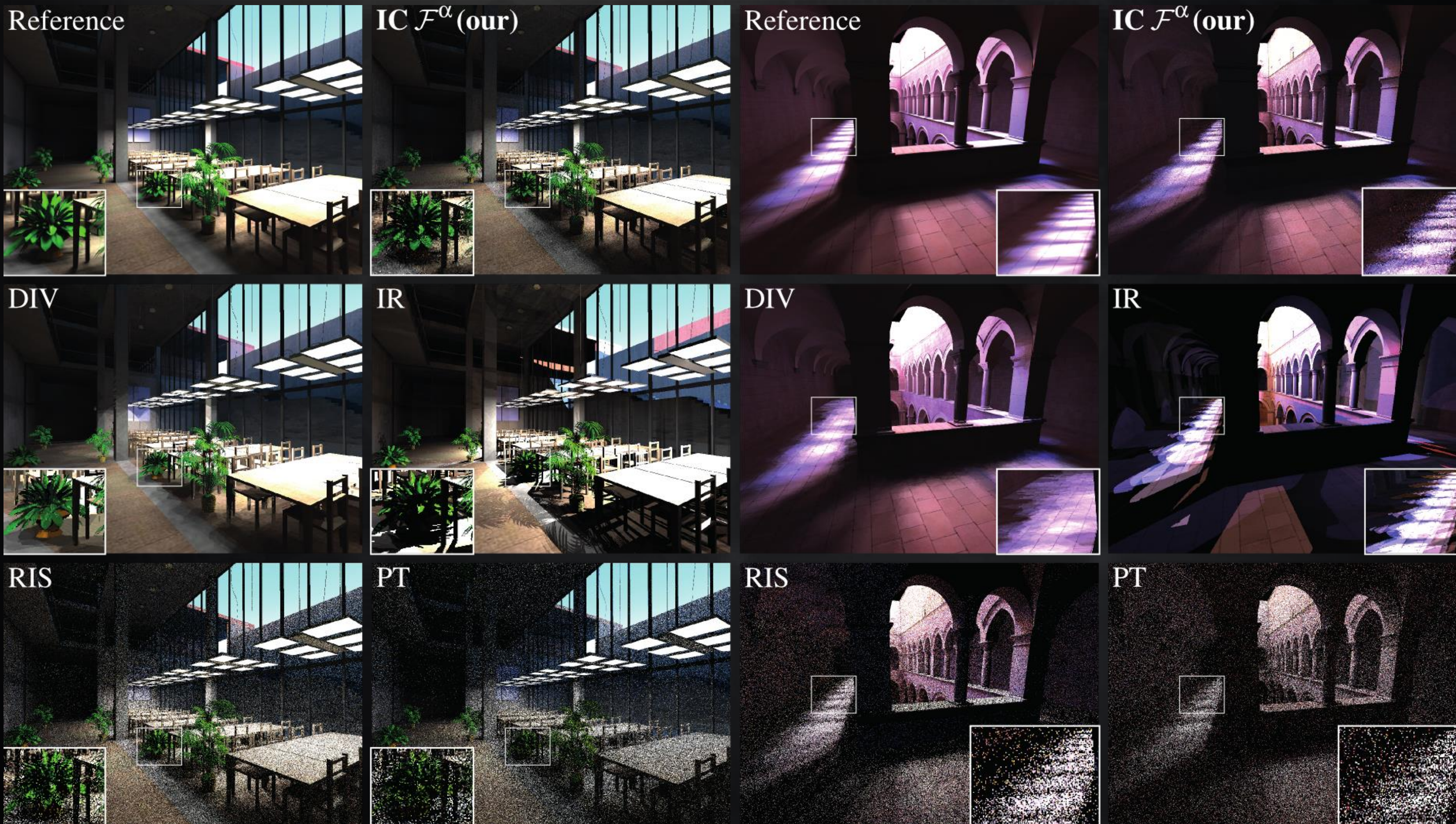
Results

Glossy



Results

Preview quality (0.5 FPS)



Summary

- Exploiting coherence in an unbiased way
  - Can capture discontinuities
    - Only error is noise (and VPL clamping)
  - Specialized sampling techniques
- All VPL types handled simultaneously
- Progressive rendering
  - First good approximation within a second
  - Full convergence with fixed memory footprint

Efficient Direct Lighting

Philipp Slusallek

Efficient Sampling of Direct Lighting

- **How to handle huge numbers of dynamic light sources at interactive rates (in an unbiased way)?**

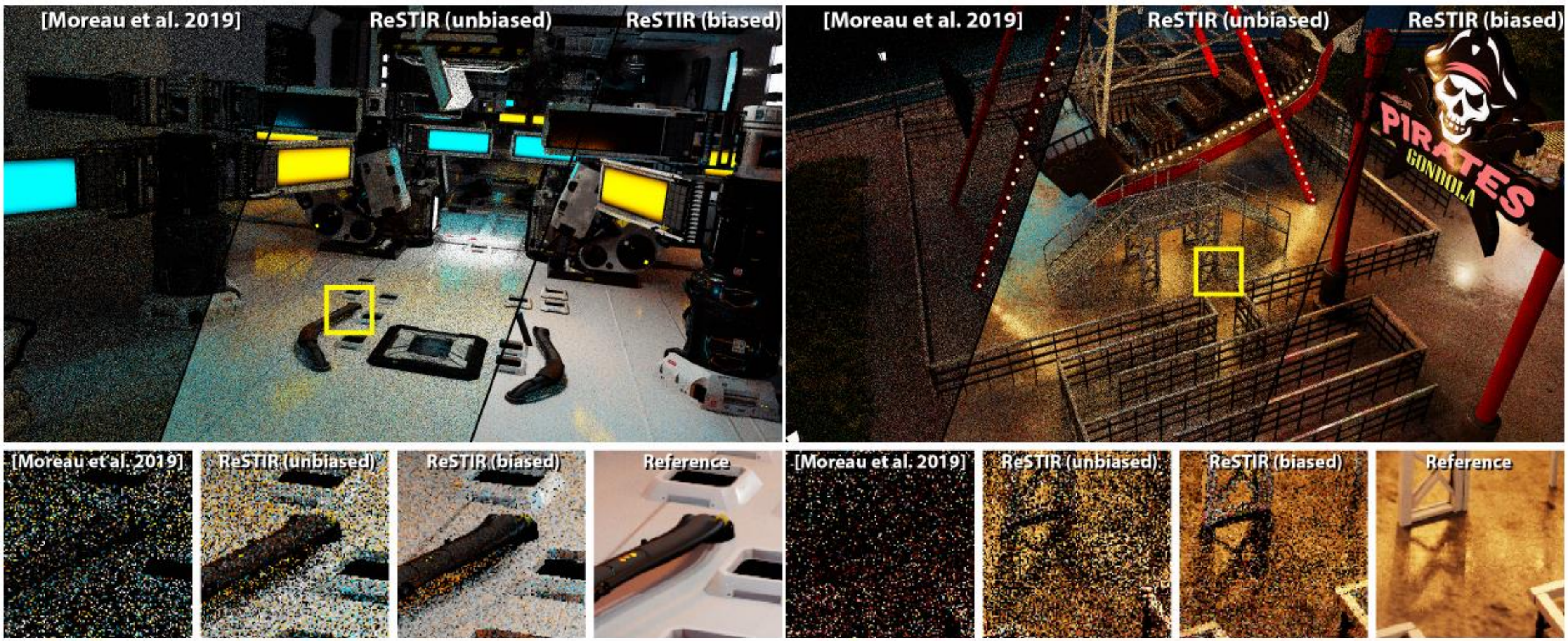


Fig. 1. Two complex scenes ray traced with direct lighting from many dynamic lights. (Left) A still from the ZERO DAY video [Winkelmann 2015] with 11,000 dynamic emissive triangles. (Right) A view of one ride in an AMUSEMENT PARK scene containing 3.4 million dynamic emissive triangles. Both images show three methods running in equal time on a modern GPU, from left to right: Moreau et al. [2019]'s efficient light-sampling BVH, our new unbiased estimator, and our new biased estimator. The ZERO DAY image is rendered in 15 ms and AMUSEMENT PARK in 50 ms, both at 1920 × 1080 resolution. ZERO DAY ©beeple, Pirate Ship

See Bitterli et al., Siggraph 2020

Resampled Importance Sampling (RIS)

- **Paper by Talbot [EGSR 2005]**
  - Randomly select a candidate set of M VPLs (e.g. according $p \sim L_e$)
  - Per pixel q compute probability of each VPL p_q (e.g. according to irradiance)
  - Select from list according to via CDF over candidate set

$$p(z|x) = \frac{w(x_z)}{\sum_{i=1}^{M} w(x_i)}, \qquad \text{with } \mathrm{w(x)} = \frac{p_q(x)}{p(x)}$$

  - Compute contribution

$$< L >_{RIS} = \frac{f(y)}{p_q(y)} \left(\frac{1}{M} \sum_{j=1}^{M} w(x_j) \right)$$

  - Second term correct for the fact that the sampling is from a subset
- **Can also be combined with MIS**
  - Select candidates from N distributions with MIS weights as p
  - Cost increases quadratically with N for computing weights ☹

Reservoir Sampling

- **Choosing N samples from a stream of items**
  - Length and content of stream may be unknown
  - Select probability of replacing an item based on weight to relative to already seen items
  - Randomly replace one of the existing items
  - Replacement happens with desired probability $p = \frac{w(x_{m+1})}{\sum_{j=1}^{m+1} w(x_j)}$
  - Ensures others in previous samples are OK

$$\frac{w(x_i)}{\sum_{j=1}^{m} w(x_j)}\left(1 - \frac{w(x_{m+1})}{\sum_{j=1}^{m+1} w(x_j)}\right) = \frac{w(x_i)}{\sum_{j=1}^{m+1} w(x_j)}$$

  - Algorithm of Chao:

```
(*
  S has items to sample, R will contain the result
  S[i].Weight contains weight for each item
 *)
WeightedReservoir-Chao(S[1..n], R[1..k])
  WSum := 0
  // fill the reservoir array
  for i := 1 to k
      R[i] := S[i]
      WSum := WSum + S[i].Weight
  for i := k+1 to n
    WSum := WSum + S[i].Weight
    p := k * S[i].Weight / WSum // probability for this item
    j := random();           // uniformly random between 0 and 1
    if j <= p                // select item according to probability
        R[randomInteger(1,k)] := S[i]  //uniform selection in reservoir for replacement
```

  - Greatly simplifies for N=1
    - No random selection

Streaming RIS

- **Generating M random samples over all light sources**
- **Select N via Streaming RIS (via Reservoir Sampling)**
  - Proportional to unoccluded contribution
- **Computing shadows only for selected N samples**

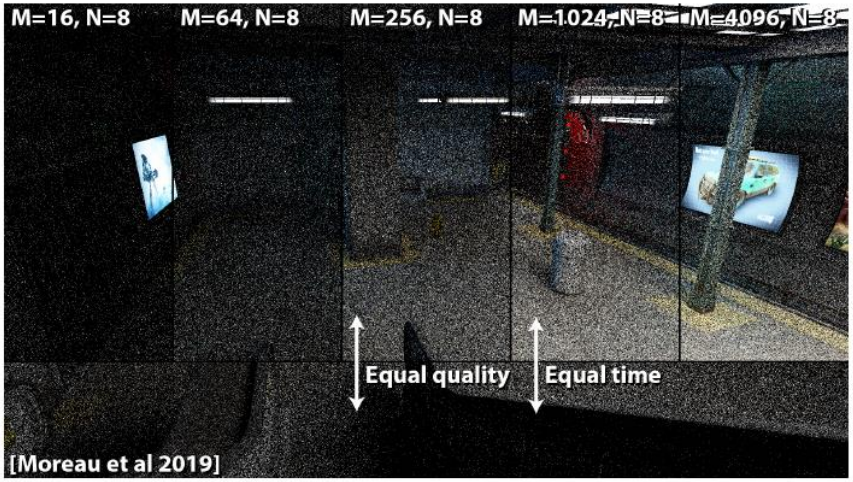
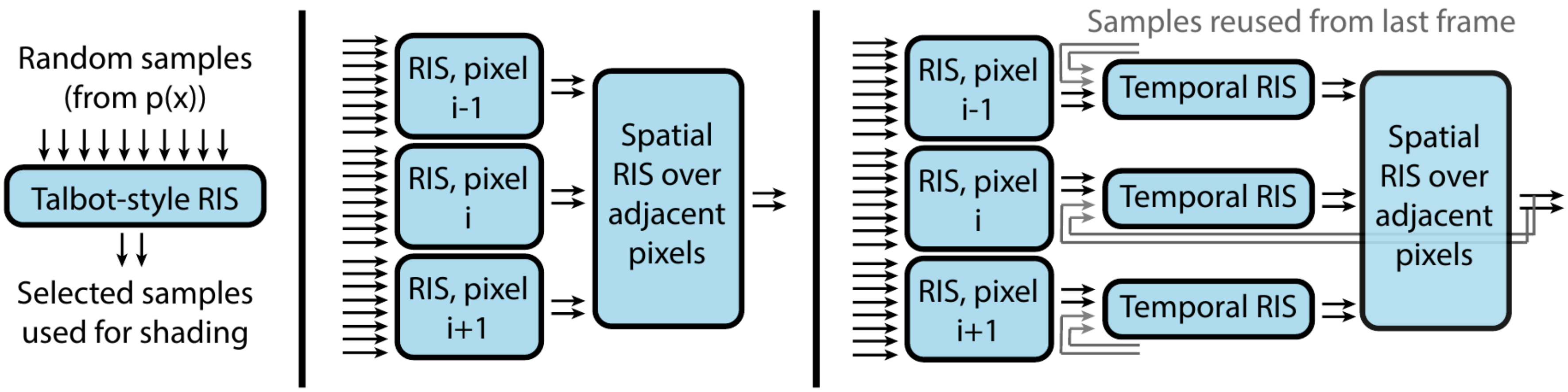


Fig. 4. Streaming RIS quality improves with increased M (candidates) and N (samples for shading). Here we show the effect of increasing M in the multi-room SUBWAY scene with 23,000 textured emissive triangles. Tracing 8 shadow rays costs 6 ms; selecting those samples costs (left to right) 1.0, 2.5, 10.1, 42, and 168 ms. Moreau et al. [2019]'s total cost is 48 ms when shooting 8 rays, comparable to $M = 1024$, but with quality comparable to $M = 256$. SUBWAY ©silvertm

Extending RIS

- **RIS can be extended both spatially and temporally**
  - Jointly increases sample count by orders of magnituse
  - With very little extra work



Spatiotemporal Reuse

- **Multiple Reservoirs can be combined into new reservoir**
  - Using Reservoir Sampling with input from each reservoir
  - Using the weight $w_{sum,i}$ of each reservoir i
  - Can be done in *constant time* with access to original input streams
- **Spatial reuse**
  - Store reservoir of M samples for each pixel (in an image)
  - Combine reservoirs from k neighboring pixels
  - Takes differences in lighting between these pixels into account
  - Can be repeated n times for taking k^n pixels into account
- **Temporal reuse**
  - Keep multiple such images around from previous time steps
- **Taking visibility into account**
  - Before spatiotemporal reuse, eliminate occluded samples per pixel
  - Unlikely to be occluded for spatiotemporally neighboring pixels

Spatial Reuse

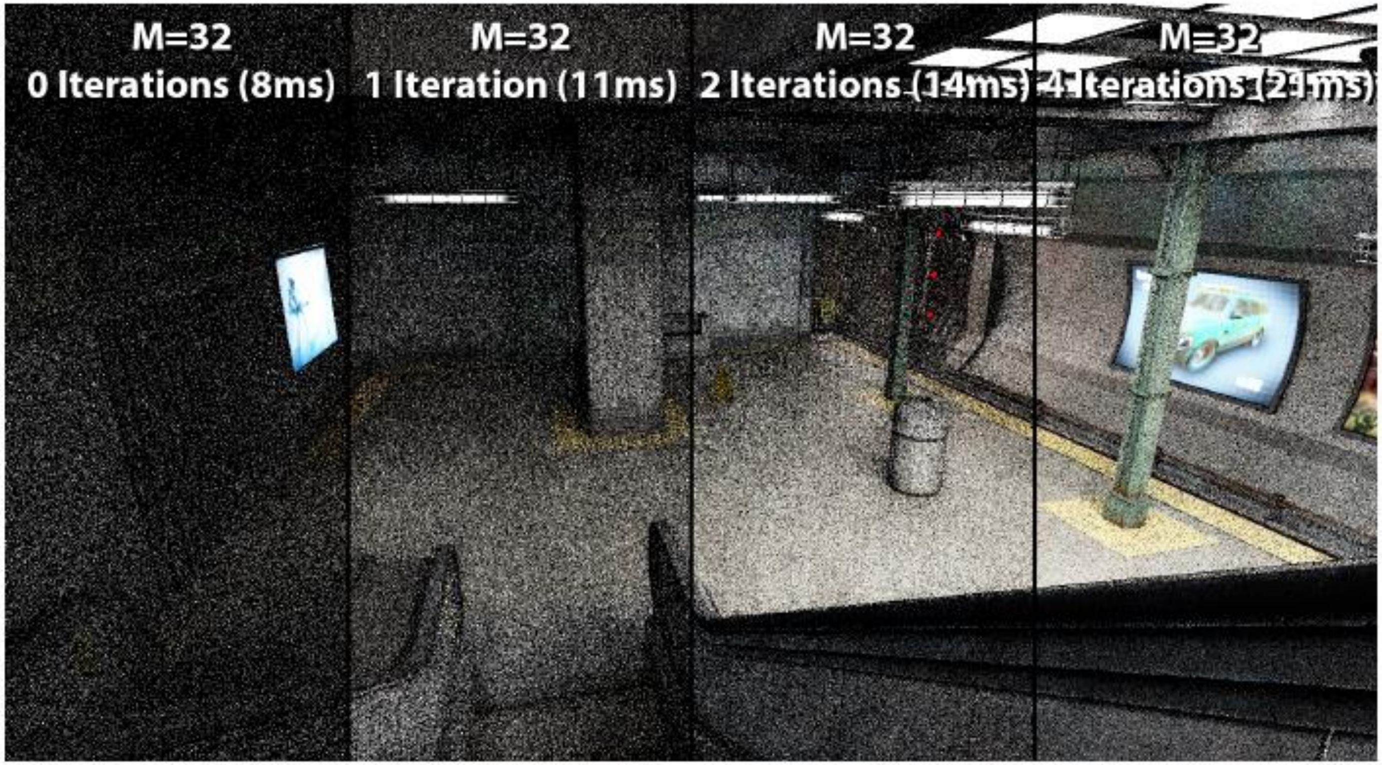


Fig. 5. Starting from $m = 32$ candidates generated by streaming RIS (left), we iteratively apply our spatial reuse operation, gathering $k = 5$ neighbors at each step. The number of repeated applications increase from left to right with 1, 2 and 4 iterations respectively. The image quality increases dramatically without much added cost. SUBWAY ©silvertm

Spatiotemporal Reuse

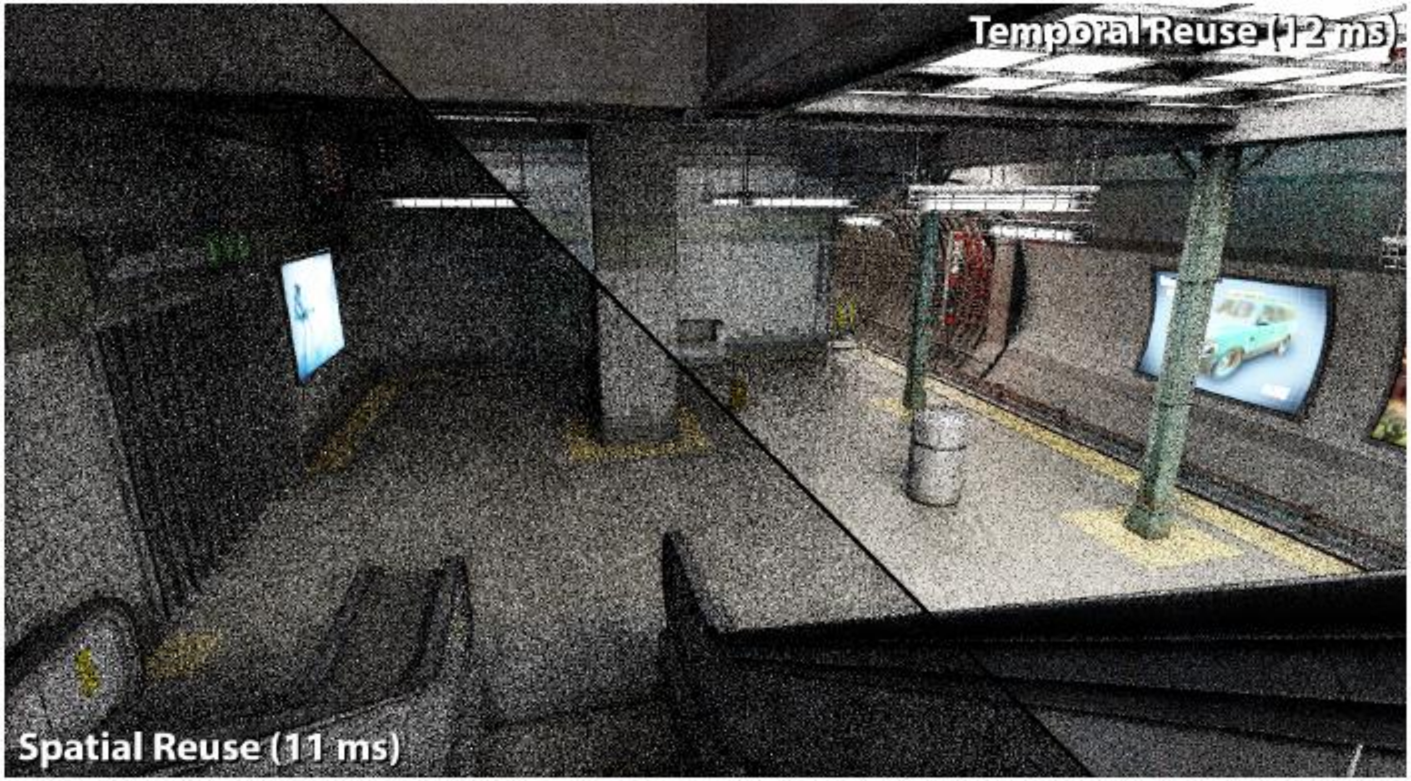


Fig. 6. Compared to one iteration of spatial reuse alone (left, $M = 4$, $k = 5$), adding candidates from previous frames to candidates from the current frame can greatly increase the image quality of streaming RIS (right, after 20 frames) with little added computational cost. SUBWAY ©silvertm

Extensions & Related Work

- **Approach can also be modified to be more efficient but biased**
  - See https://web.cs.dartmouth.edu/news/2020/05/rendering-millions-dynamic-lights-realtime

- **Other Related Work: Bayesien Direct Illumination**
  - See https://cgg.mff.cuni.cz/~jaroslav/papers/2018-bayesianlighting/2018-vevoda-bayesianlighting-slides.pdf