Robust Sampling for Progressive Global Illumination

Iliyan Georgiev   Philipp Slusallek
Saarland University
DFKI

Outline

1. Motivation
   a) Progressive rendering
   b) Importance (of) sampling
2. Importance sampling of virtual point lights
3. Importance caching for complex illumination

Ultimate goal

performance

“Grand Theft Auto”

Game

Time: 0.03 sec

○ Performance

↗ Realism

Goal!

○ Realism

↗ Performance

“The Wet Bird”

Internet Ray Tracing Competition

Time: 21 hours

photo-realism

Progressive rendering

- A decent solution
- Quickly gaining popularity
  - ✓ Progressively increasing quality (while still)
  - ✓ Low-latency interaction
  - ✗ Difficult to reuse samples
- Need algorithms that
  - Converge ← *ultimate quality*
  - Have fixed memory footprint ← *limited memory*
  - Are well parallelizable ← *parallel hardware*

Importance (of) sampling

- Only classic brute-force algorithms used in practice
  - ✓ Fulfill requirements
  - ✗ Slow… convergence…
- Tremendous improvements by smarter sampling
  - Importance sampling
  - Multiple importance sampling (MIS)
  - Adaptive sampling

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

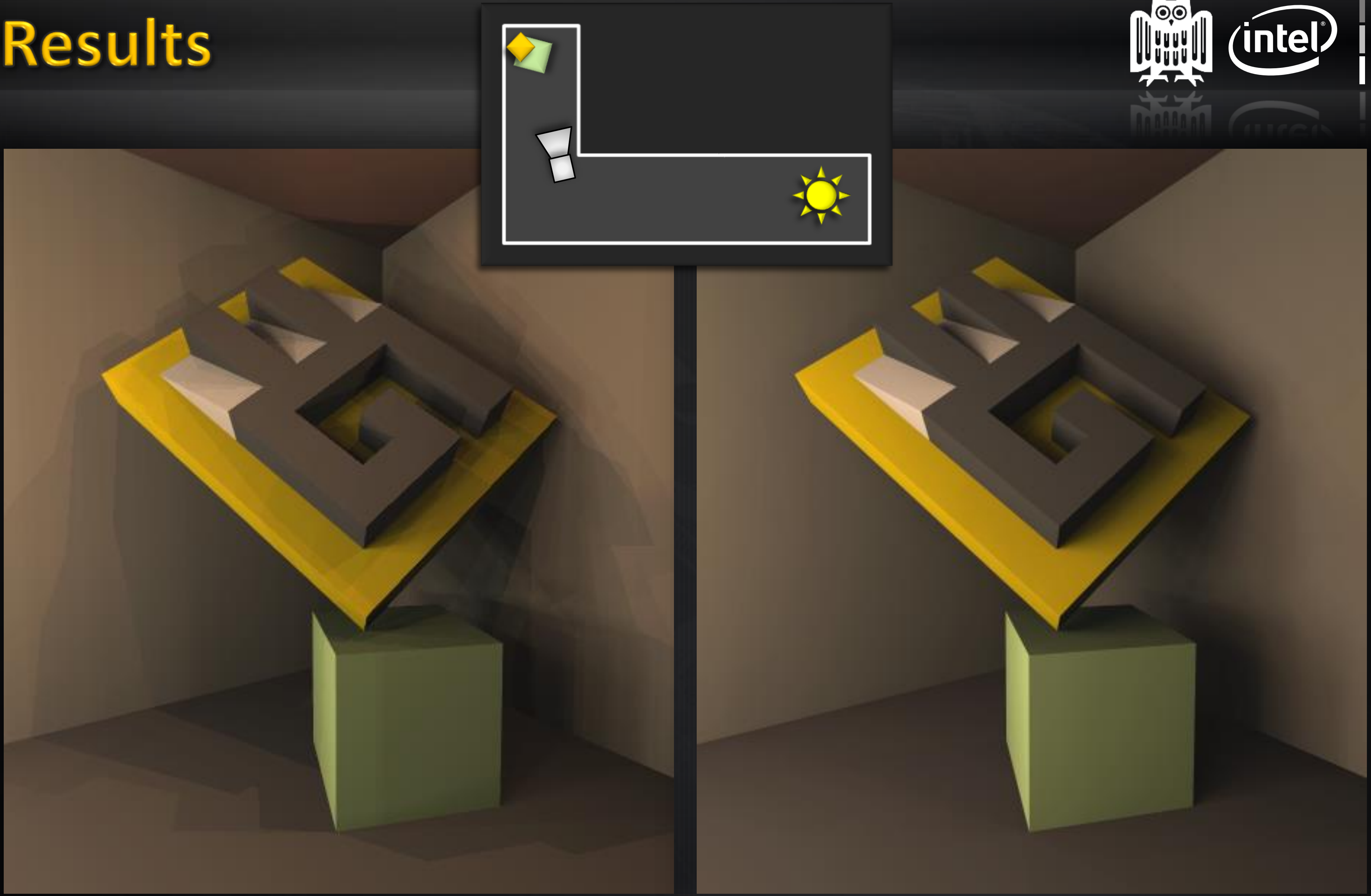
Choosing the acceptance probability

- 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



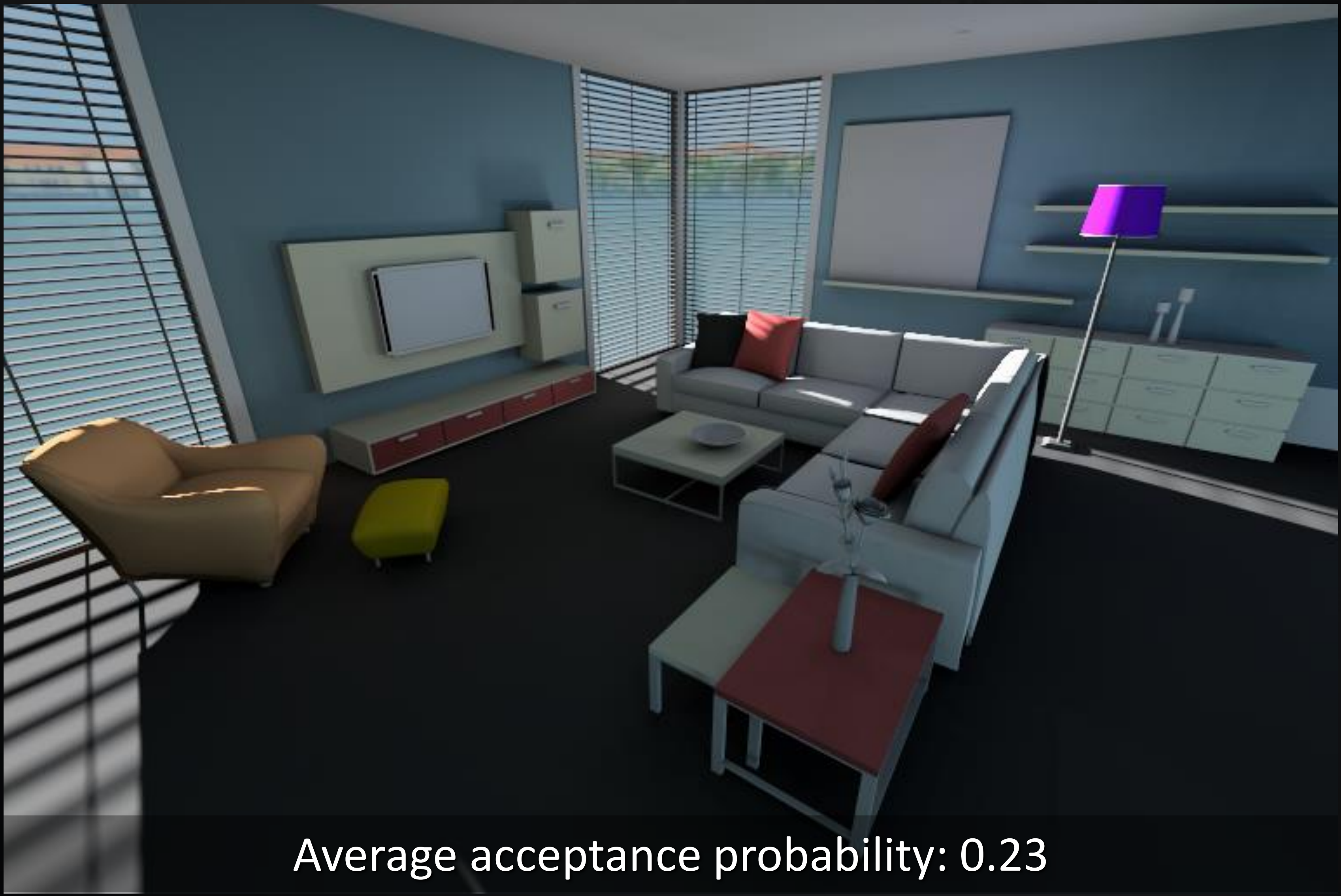
Instant Radiosity

Our Extension (0.07 acceptance)

Results

Average acceptance probability: 0.28

Results



Average acceptance probability: 0.23

Wrap Up

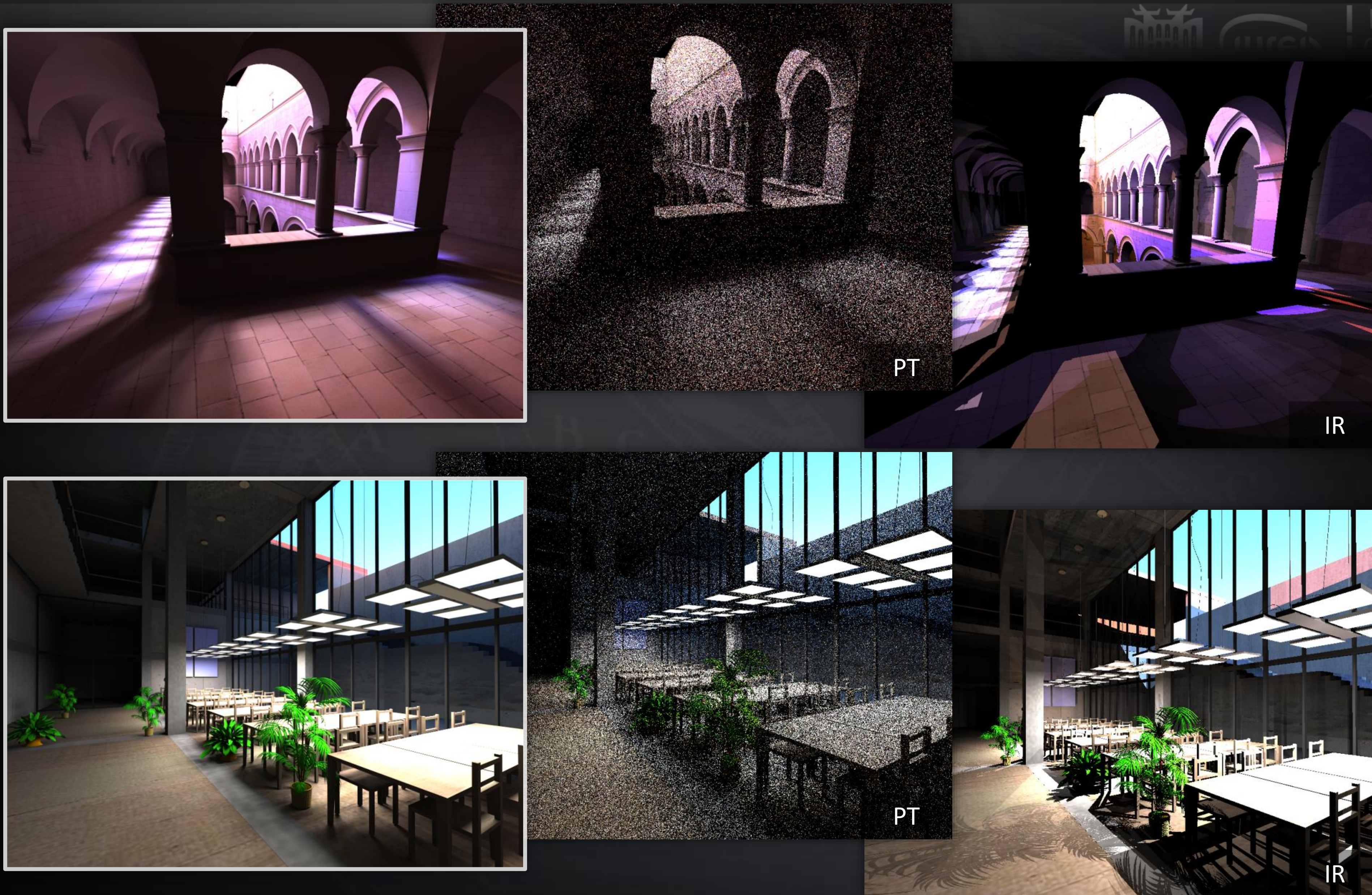
- Simple extension of IR
  - Generate VPLs from light sources only
- Probabilistically accept VPLs on the fly
  - Fixed minimal additional storage
  - Easy to parallelize
- Two parameters
  - $\varepsilon_{\mathrm{p}} = 0.05$
  - Number of camera samples, e.g. 100
- "One-pixel image" assumption

Importance Caching for Complex Illumination

Eurographics 2012

full paper

Motivation



Motivation

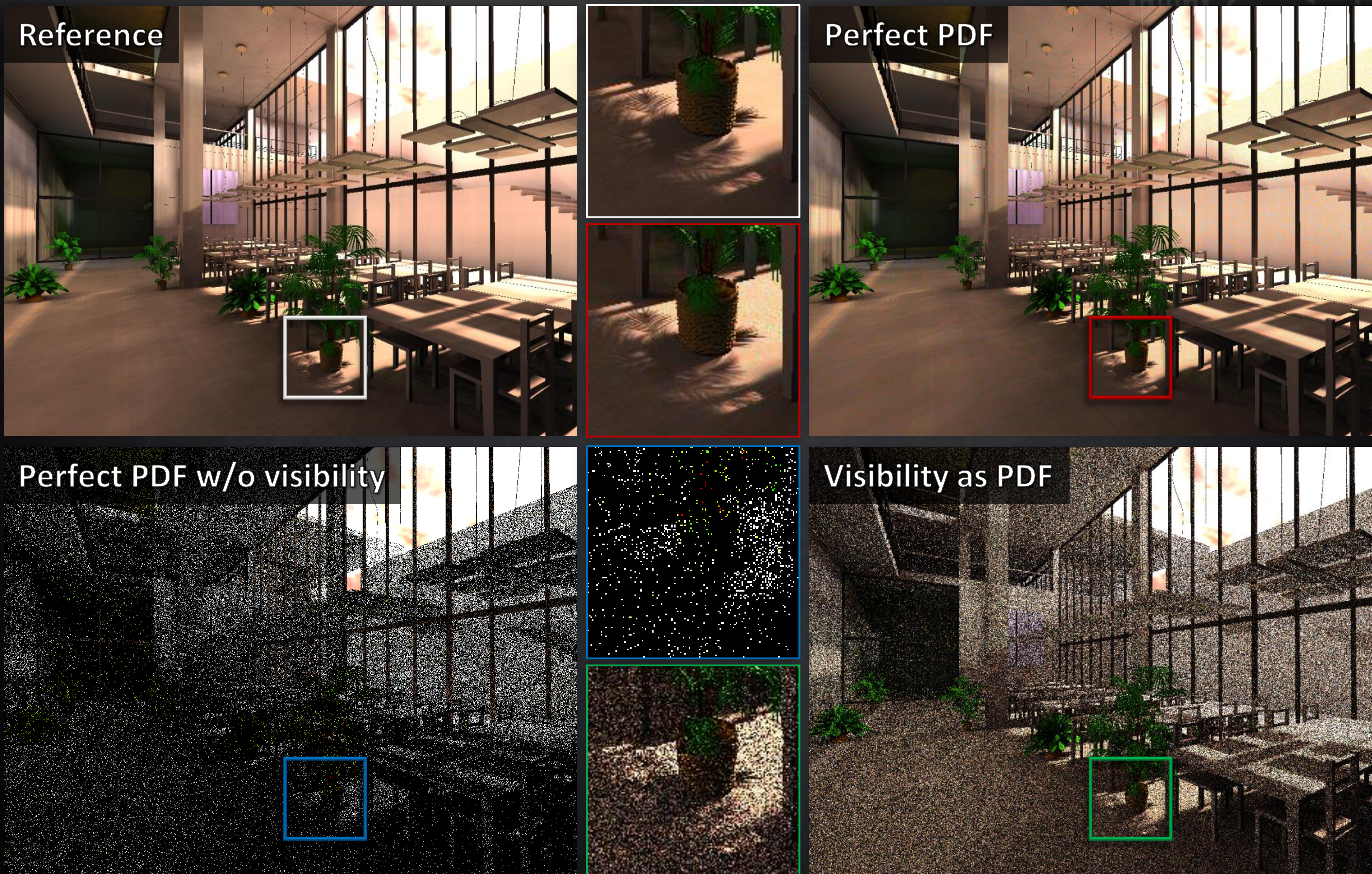
- Global illumination still very costly
  - Indirect illumination
  - Even direct illumination – environment, area lights
- Two basic algorithmic improvements
  - Importance sampling
    - Better sample distribution (ideally proportional to integrand)
    - Higher quality with fewer samples
  - Exploiting 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)



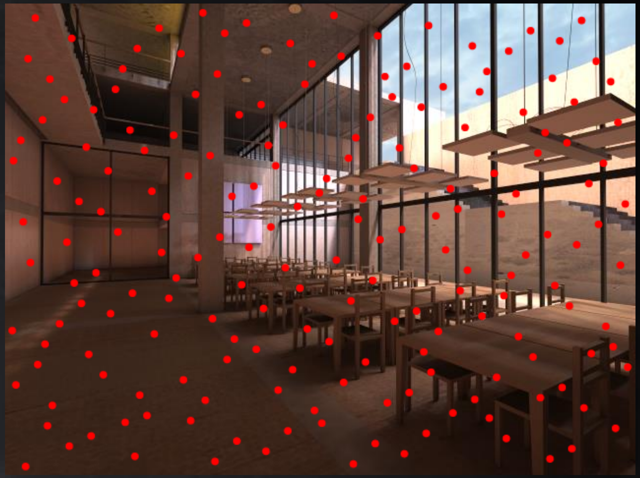
Background

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



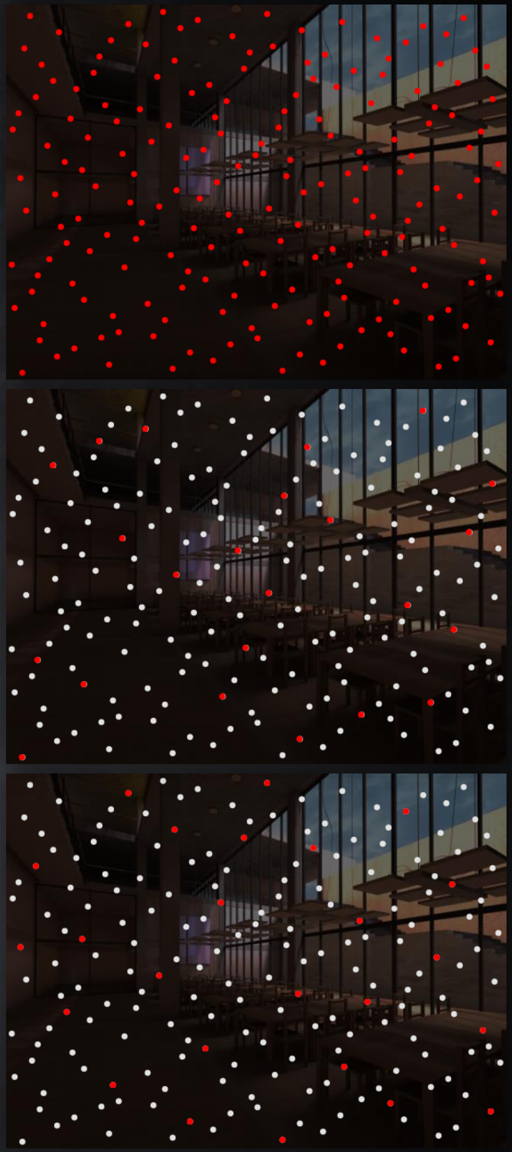
- Idea – combine all three
  - Unbiased VPL sampling framework
  - Shade only few most relevant VPLs
- Approach
  - Consider full integrand (visibility)
  - Shade all VPLs at few locations
  - Reuse VPL evaluations as importance at other locations
- Issue – illumination discontinuities
  - Additional 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

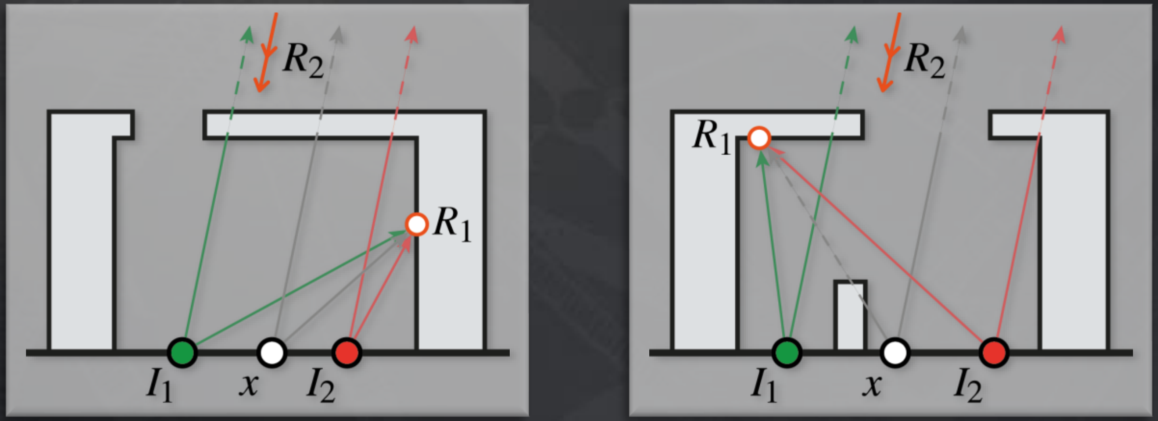
- For each pixel shading point
  - Find nearest IRs
  - Use IR distributions defined for VPL sampling
- Robust sampling if at least one IR correlates
- Increased variance when all IRs irrelevant
  - Identify causes for VPL contribution changes
  - Additional, increasingly conservative distributions
-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

Sampling distributions

- Four sampling distributions at each IR

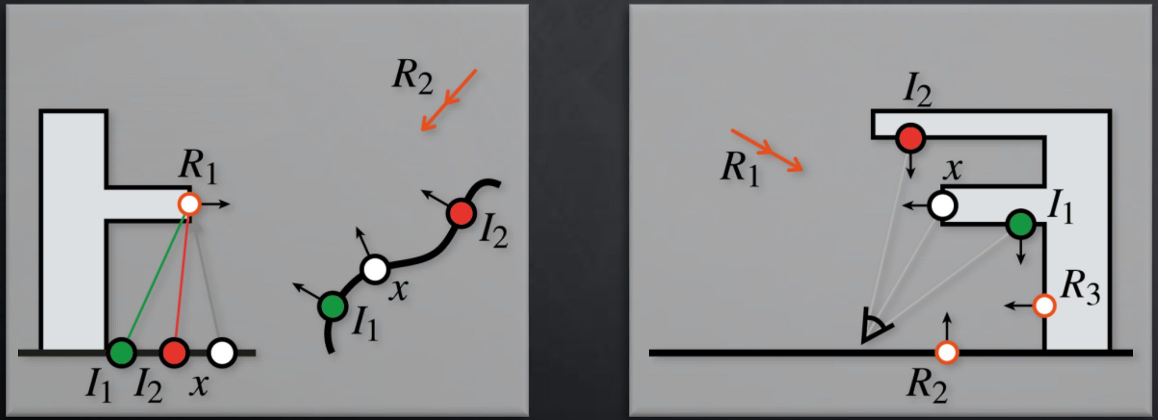
$\mathcal{F}$: Full

$\mathcal{U}$: Unoccluded



$\mathcal{B}$: Bounded

$\mathcal{C}$: Conservative



Distribution Combination

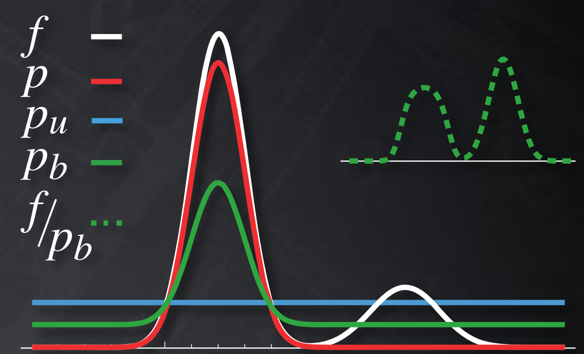
Horizontal Combination

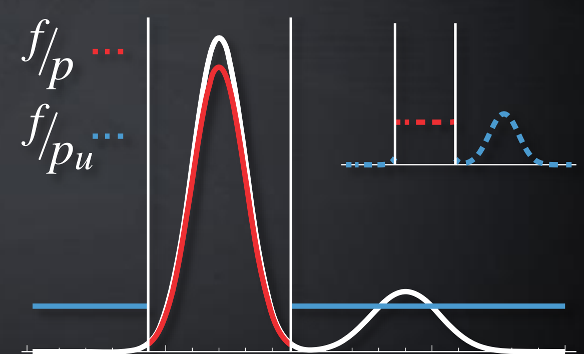
- Matrix structure
- Distributions often correlate among IRs
  - Combine first horizontally
    - Balance heuristic
    - Corresponds to mixture
      - Directly sample mixture
  - Collapse columns into one

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

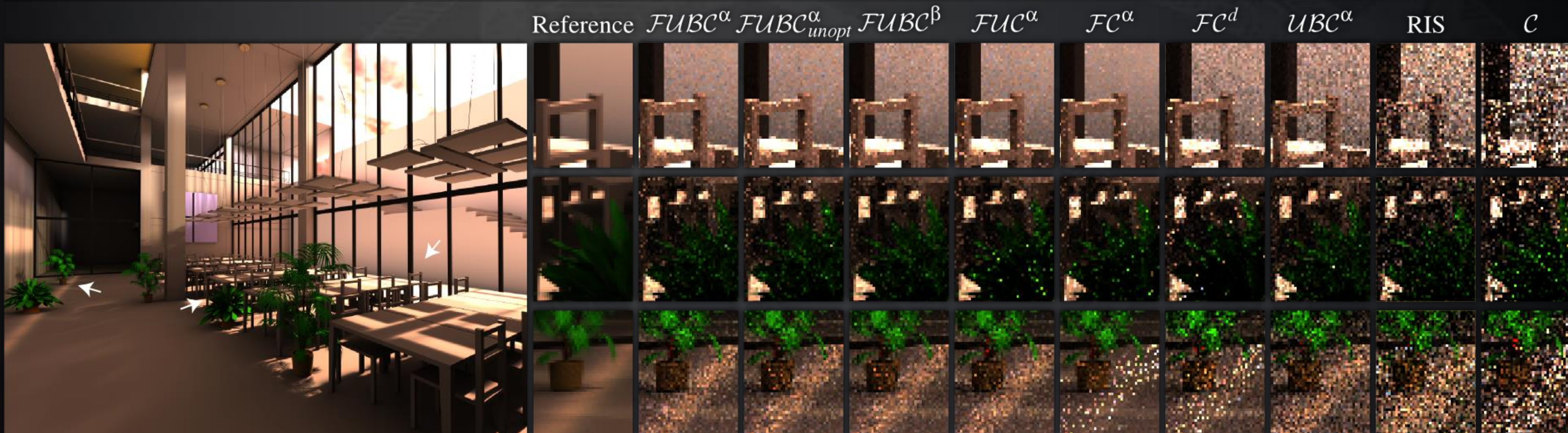




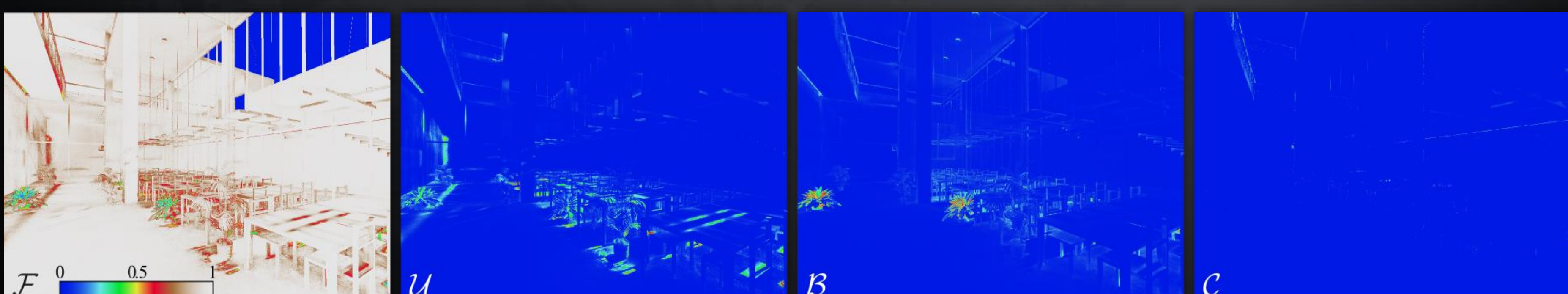
Results

Study Hall (diffuse)

Technique comparison

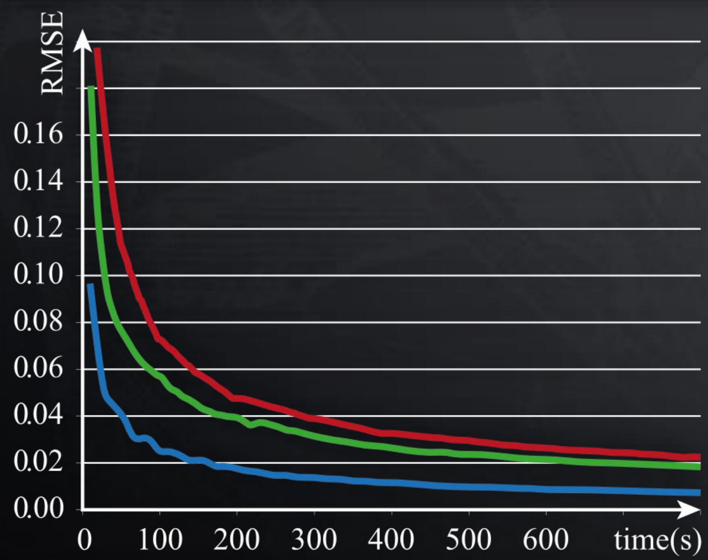


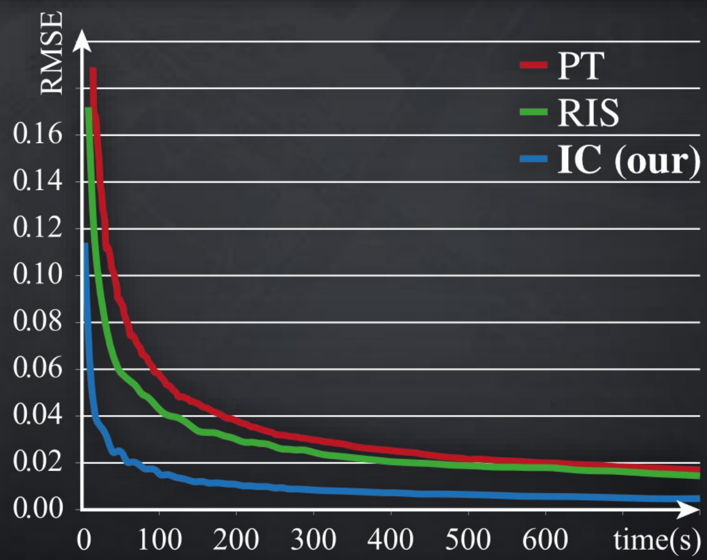
$\mathcal{FUBC}^{\alpha}$ fractional contributions



Results

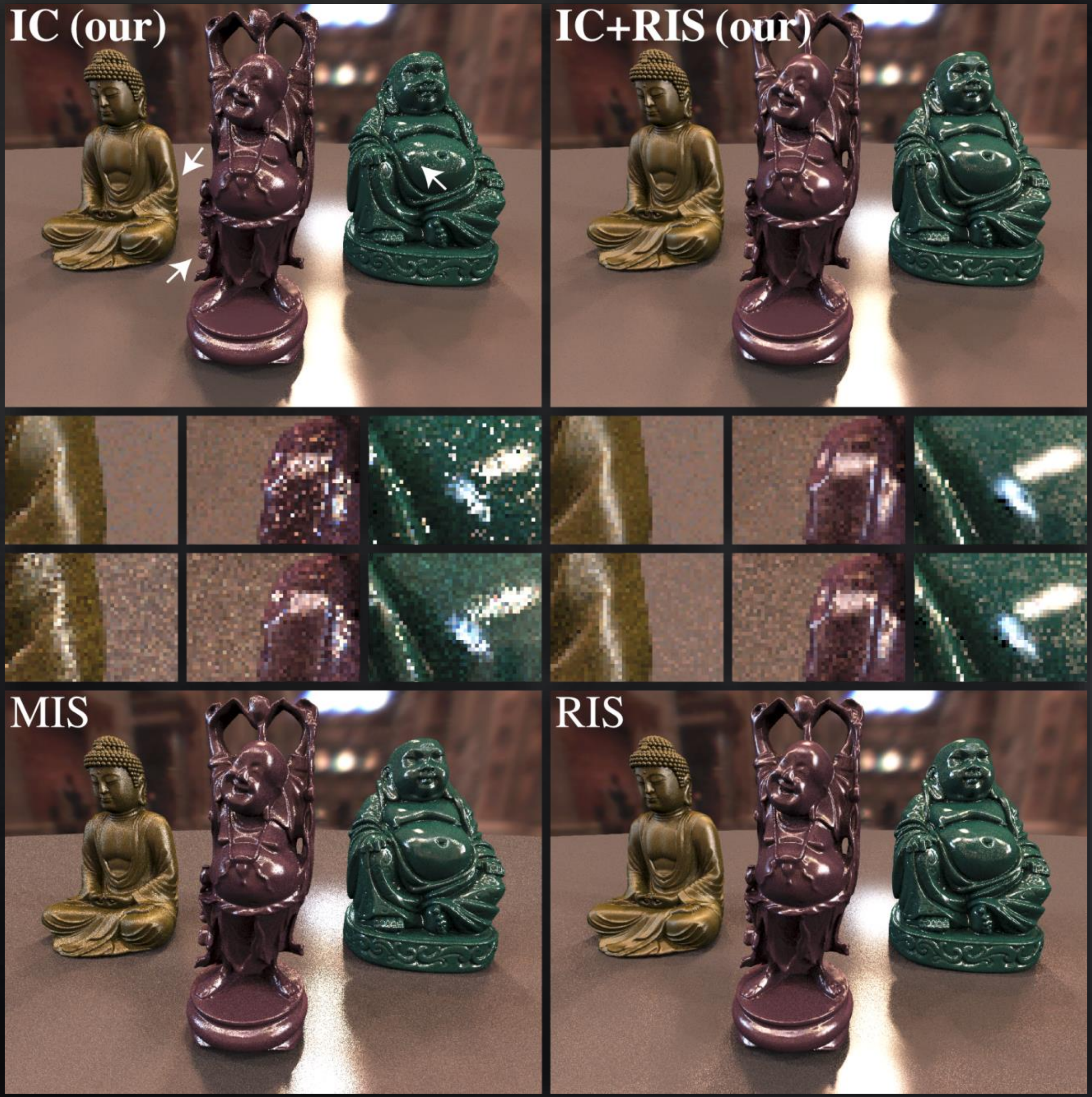
Numerical tests





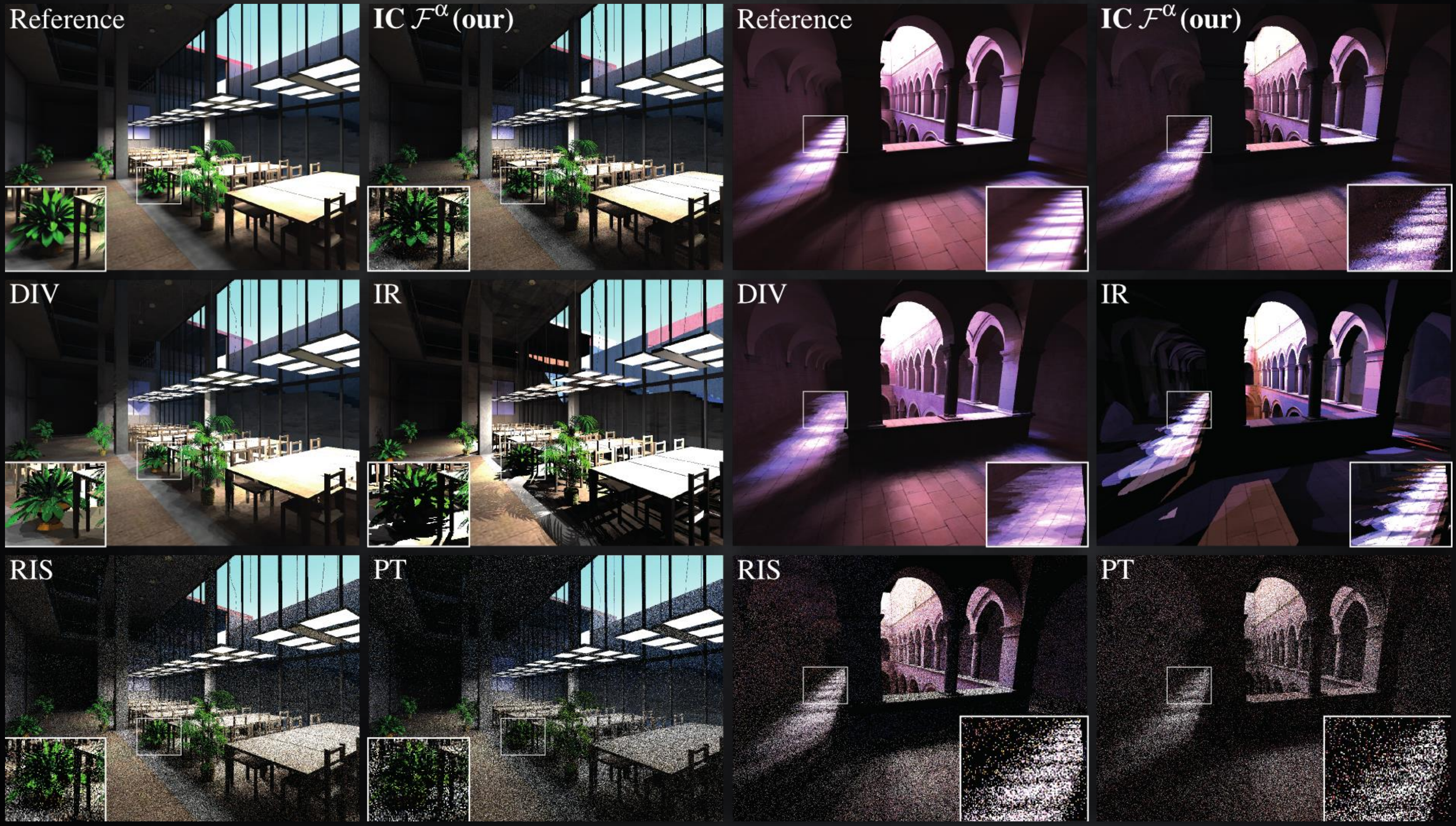
Results

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