Hi, my name is Thomas Ludwig, I'm with Glare Technologies, developers of Indigo Renderer

We're a small company of 3 fulltime employees, plus some friends and contractors helping out.

I've been with Glare for 10 years now, though the history of Indigo stretches back further than that as a hobby project of Nicholas Chapman.

OVERVIEW

- History and context
- Motivation
- Difficulty of caustics and indirect lighting
- From a user's point of view…
- From a developer's point of view…
- Avoid complex algorithms
- GPU rendering benefits and challenges
- Conclusion

I'll start with a brief history of Indigo Renderer and our market context, and then go over some motivating examples for some of the design decisions.

Difficult indirect lighting, especially caustics, is not a focus of most rendering systems so I'll go into detail about that, followed by user and developer perspectives for using bidirectional algorithms.

HISTORY AND CONTEXT

- Basis is Veach's thesis [1], inspired by Maxwell Render
- Need for specialised product, “max quality” implementation
  - Make it accessible to non-CG specialists

Basis is of course Veach’s thesis, and Maxwell early pioneers in physically-based MC

Non-CG specialists e.g. architects, CAD designers, people with primary job in design, want good results easily

SketchUp and Revit users for archviz, C4D for productviz, Blender CAD

HISTORY AND CONTEXT

- Basis is Veach's thesis [1], inspired by Maxwell Render
- Need for specialised product, “max quality” implementation
  - Make it accessible to non-CG specialists
- Indigo Renderer
  - Emphasis on quality and simplicity
  - (Volumetric) unidirectional and bidirectional path tracing
  - Optional Kelemen PSS-MLT [2] on top for most difficult scenes
  - Truly unbiased - 10k path depth, bidir on by default
  - Mainly archviz and productviz customers

Indigo Renderer places great emphasis on image quality, and simplicity.

MOTIVATION

GENERATIONS / VANCOUVER 12-16 AUGUST
SIGGRAPH2018

- Archviz / productviz has different requirements and allowances
  - Can assume scene fits in memory, allows bidir methods and GPU
  - Demand highest final quality, quick previews
- Keep algorithms simple as possible
  - Need to exploit huge GPU resources
  - GPU unidir is already quite complex!

Unidir vs bidir (2.5 mins on AMD ThreadRipper), scene by Giorgio Luciano & polygonmanufaktur.de

Biggest enabling assumption of viz: scenes can fit in memory

Relaxing this constraint allows powerful bidirectional methods

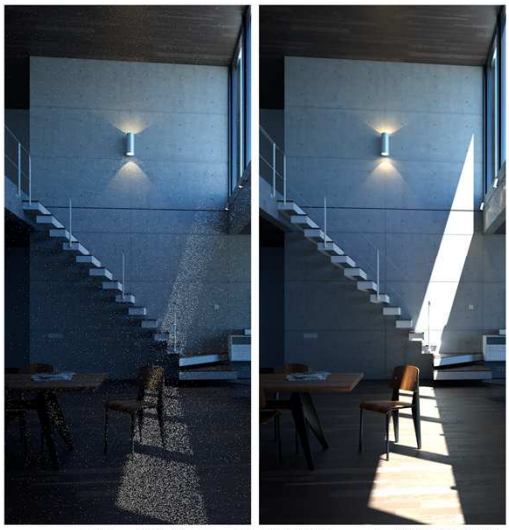
Pronounced advantage over unidir for rendering caustics

Fast early convergence big practical benefit of MLT, useful for previews

MOTIVATION

GENERATIONS / VANCOUVER 12-16 AUGUST
SIGGRAPH2018

- Archviz / productviz has different requirements and allowances
  - Can assume scene fits in memory, allows bidir methods and GPU
  - Demand highest final quality, quick previews
- Keep algorithms simple as possible
  - Need to exploit huge GPU resources
  - GPU unidir is already quite complex!
- Interior renders much more efficient
  - Difficult to sample localised reflected light with eye paths
  - Light paths induce perfect distribution



Unidir vs bidir (8 mins on Intel i7-8700K), scene by MAD IMAGERY

For unidir to sample localised reflections, needs path guiding methods

When there is realistically modelled glass in front of emitter, you need bidir methods

MOTIVATION

GENERATIONS / VANCOUVER 12-16 AUGUST
SIGGRAPH2018

- No contest in optical simulations

Lenses Experiment by Raphael Rau / Silverwing VFX, rendered with bidir MLT @ 4K, clean image in 45 mins on 4 GHz 8-core

Scenes as complex as this are not the norm, but same high accuracy engine rendering any archviz or productviz scene

DIFFICULTY OF CAUSTICS AND INDIRECT LIGHTING

- Caustic and indirect illumination common in archviz
  - Glass bulbs, lampshades, mirrors and windows
  - Illumination from IES profiles and detailed lights
- Often approximated
  - Biased methods, limited path depth, increasing roughness with glossy scatters

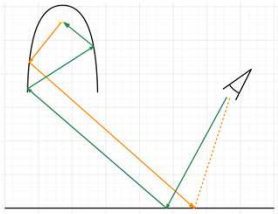
Dowling by Aaron Crozier / bubs

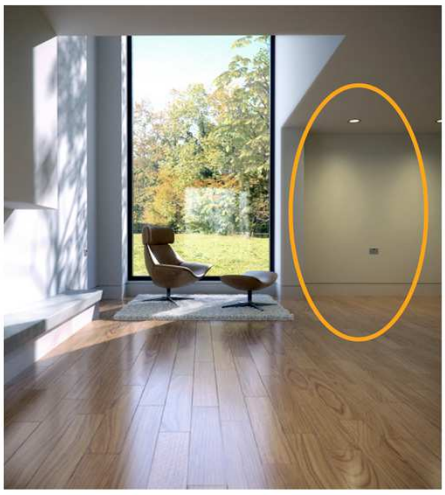
Many realistic scenes almost entirely illuminated by indirect light

Narrow IES lights particularly tricky for unidir

DIFFICULTY OF CAUSTICS AND INDIRECT LIGHTING

- Caustic and indirect illumination common in archviz
  - Glass bulbs, lampshades, mirrors and windows
  - Illumination from IES profiles and detailed lights
- Often approximated
  - Biased methods, limited path depth, increasing roughness with glossy scatters
- But what if it's modelled accurately?
  - Difficult to sample from eye paths:





Dowling by Aaron Crozier / bubs

Green light path starts from the camera, hits the floor or wall, fails to make a direct lighting connection and strikes light through series of low probability scatters, producing firefly

Orange light paths start from emitter, reflect inside fixture, then strike floor or wall and connect to camera → perfect importance sampling

FROM A USER'S POINT OF VIEW...

- Compute power gets cheaper over time, human time does not
- Fast on modern hardware
- Bidir by default means less to think about
  - Safest default without scene-based optimisation

Unidir vs bidir (5 mins), scene by Filippo Scarso / pibuz

Some users willing to trade rendering speed / hw costs for simple setup, high quality final results

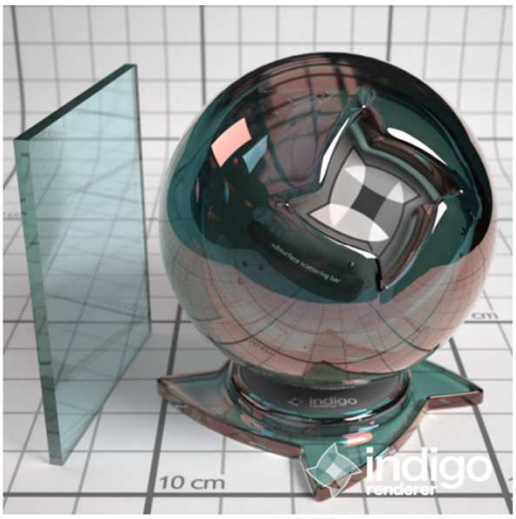
Nevertheless, all A/B comparisons in these slides rendered in 5 mins or less on desktop CPU

Bidir at best much more efficient, at worst not much worse (MIS), thus safest default without changing rendering modes

FROM A USER'S POINT OF VIEW...

GENERATIONS / VANCOUVER 12-16 AUGUST
SIGGRAPH2018

- Compute power gets cheaper over time, human time does not
- Fast on modern hardware
- Bidir by default means less to think about
  - Safest default without scene-based optimisation
- Example: partnership with Saint-Gobain Glass
  - Verified spectral model for several glass types
  - Available on the online material database
  - Can now easily be used in high accuracy archviz, both interior and exterior shots



Saint-Gobain Glass Cool-Lite SKN 165

Saint Gobain needed predictive accuracy, measured spectral data for various commercially offered glass types

Indigo material library allows anyone to use extremely realistic glass

FROM A DEVELOPER'S POINT OF VIEW...

- Bidir is more difficult to implement and maintain
  - Unphysical hacks trickier
    - Section planes
    - Shadow catcher

Scene by polygonmanufaktur.de

Unphysical hacks commonly used for viz, e.g. section planes, invisible to cam objects

Section planes can't simply clip off geometry, still need to emit from lights and reflect etc. More special cases for bidir

Shadow catcher planes are another special case, used for compositing

FROM A DEVELOPER'S POINT OF VIEW...

- Bidir is more difficult to implement and maintain
  - Unphysical hacks trickier
    - Section planes
    - Shadow catcher
    - Invisible to camera
  - Non-symmetric scattering, normal smoothing
  - Naive implementation is O(N^4), can be optimised to O(N^2) [3]

Scene by Oscar Johansson

Similarly, more special cases for invisible to cam objects when using bidir

Refraction density different for light paths, interpolated normals need Veach's "smoothing factor"

Fast N^2 implementation complex to implement and debug

AVOID COMPLEX ALGORITHMS

- Complex can mean:
  - Difficult to implement robustly, e.g. Veach-MLT vs PSS-MLT
  - Difficult to understand settings exposed to users, e.g. irradiance caching
  - Difficult to predict behaviour, e.g. flickering in animation

AVOID COMPLEX ALGORITHMS

- Complex can mean:
  - Difficult to implement robustly, e.g. Veach-MLT vs PSS-MLT
  - Difficult to understand settings exposed to users, e.g. irradiance caching
  - Difficult to predict behaviour, e.g. flickering in animation
- Bidir is as powerful as you can get before getting exotic
  - Proven highly efficient mix of direct and indirect techniques
  - Embarrassingly parallel, albeit with incoherent splats for light paths
- Needs to work on GPU eventually too

Vanilla bidir has overhead from many shadow rays, but there are ways to make it more efficient.

Combinatorial bidir, two-way path tracing by Simon Brown are examples

GPU RENDERING BENEFITS AND CHALLENGES

- Huge performance boost for unidir PT (fully) on GPU
  - Tried hybrid CPU+GPU, always bottlenecked
  - Multi GPU performance is incredible
  - Nvidia announced dedicated ray tracing units in GeForce RTX, 10 gigarays / sec!
- GPU bidir in future
  - Requires more stages in wavefront path tracing [4]
- Even more memory limited
  - 12 GB on GPU versus 128 GB on CPU practical in 2017
  - Out-of-core would add large amount of complexity
  - Still want in-memory codepath for simpler scenes

Can't ignore available GPU resources, factor ~10 brute force per GPU, GeForce RTX announced yesterday with dedicated RT silicon

Wavefront unidir PT already complex, bidir roughly doubles it (subpath tracing, MIS combining)

Would like to see more research on GPU out of core rendering, need practical method since complexity strictly additive

THANK YOU

GENERATIONS / VANCOUVER 12-16 AUGUST
SIGGRAPH2018

- References:
  - [1] Veach thesis “Robust Monte Carlo Methods for Light Transport Simulation”
  - [2] Kelemen et al. “Simple and Robust Mutation Strategy for Metropolis Light Transport Algorithm”
  - [3] van Antwerpen thesis “Unbiased physically based rendering on the GPU”
  - [4] Laine et al. “Megakernels Considered Harmful: Wavefront Path Tracing on GPUs”