Efficient GPU Path Tracing

Jerry van Angeren

Jorim Geers

7 March 2013

Content

- Introduction
- Path Tracing
- Bi-directional Path Tracing
- Conclusion

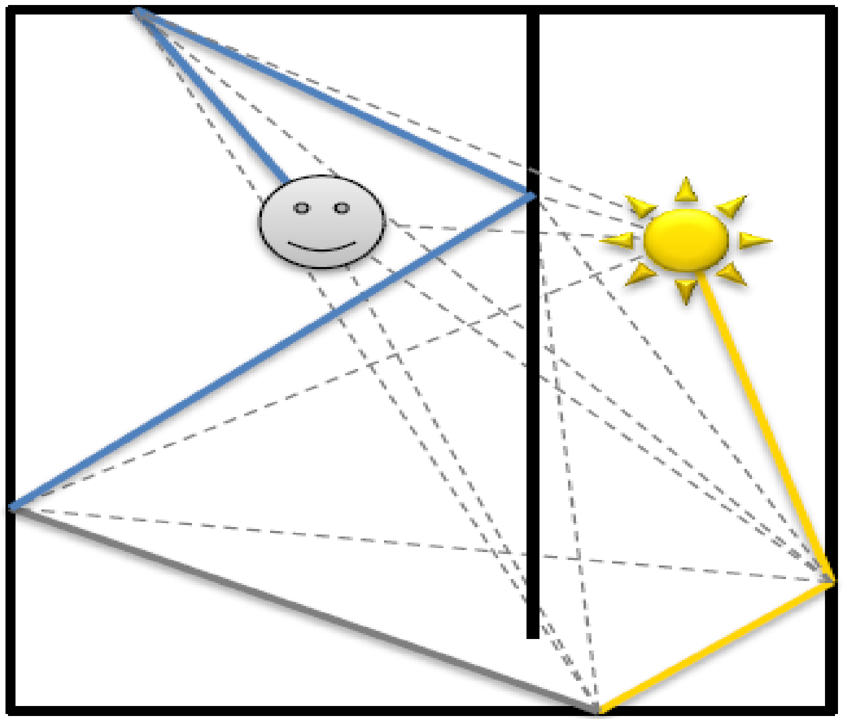
Introduction

- Stochastic termination reduces SIMD efficiency
- Proposal:
  - Stream compaction
  - Sample regeneration

Antwerpen, D., van (2011). Improving SIMD efficiency for parallel Monte Carlo light transport on the GPU. In *Proceedings of the ACM SIGGRAPH Symposium on High Performance Graphics* (HPG '11), 41-50.

Short Recap

- Monte Carlo Light Transport
- Russian Roulette
- Bi-directional Path Tracing



Path Tracing

- Keeping SIMD efficiency high is a major complication
- Immediately regenerate new samples
- Primary ray coherence
- Streaming Path Tracing (SPT)

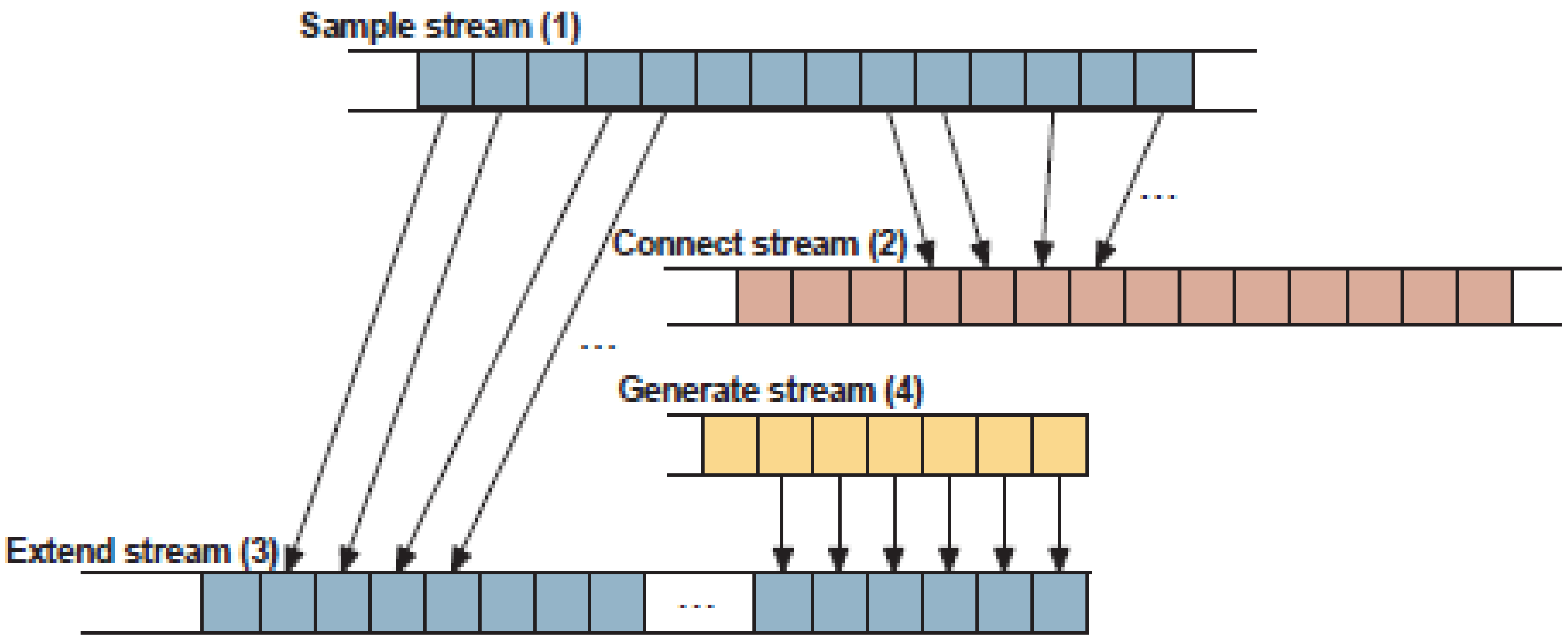
Stream Compaction

- Remove terminated/inactive samples
  - Russian Roulette
- Atomic instructions
  - Counter for written elements
  - Atomically increased by batch size
  - Colliding Atomic instructions

Sample Regeneration

- Shorter stream left after stream compaction
- Fill empty places with regenerated samples
- Primary ray coherence

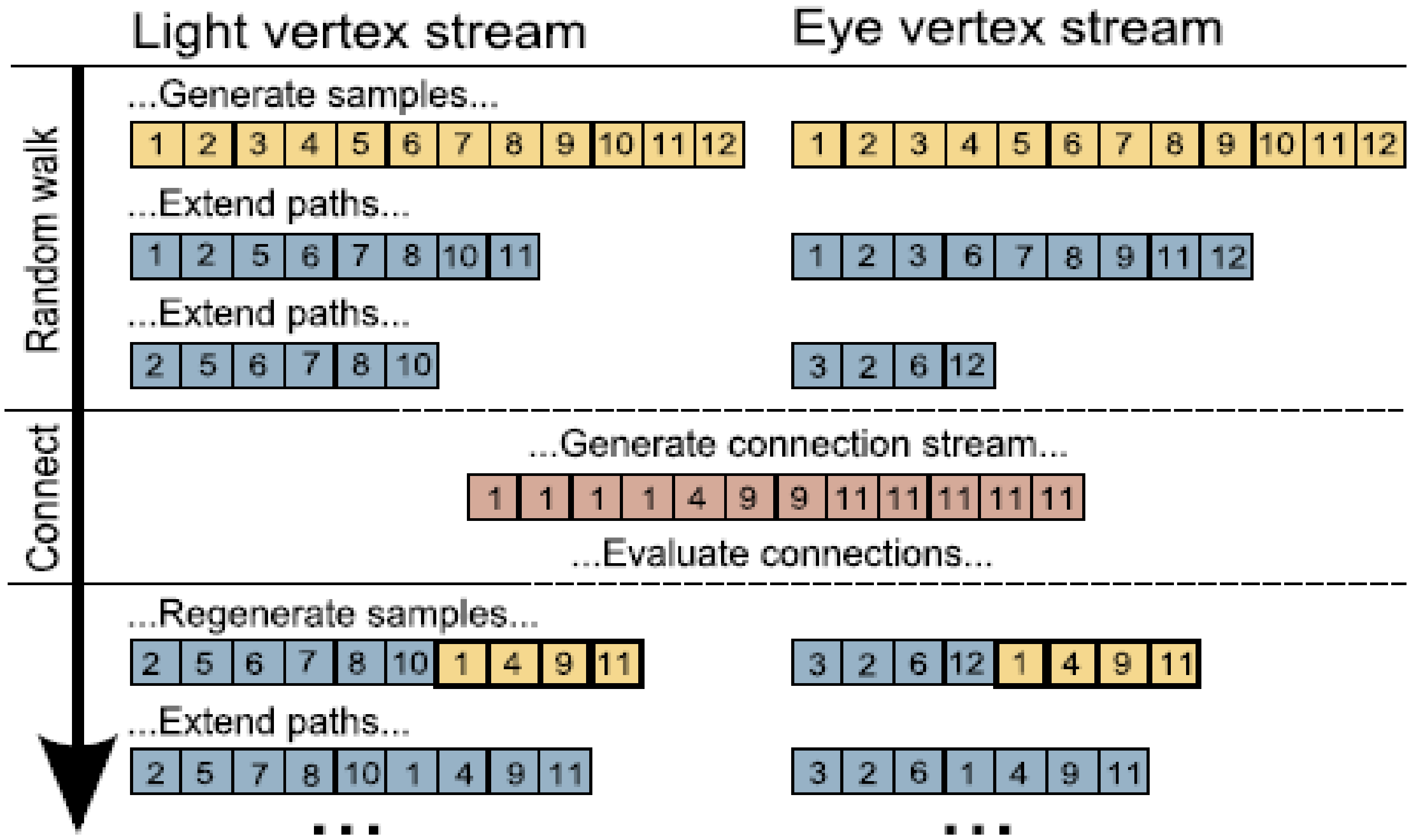
Overview SPT



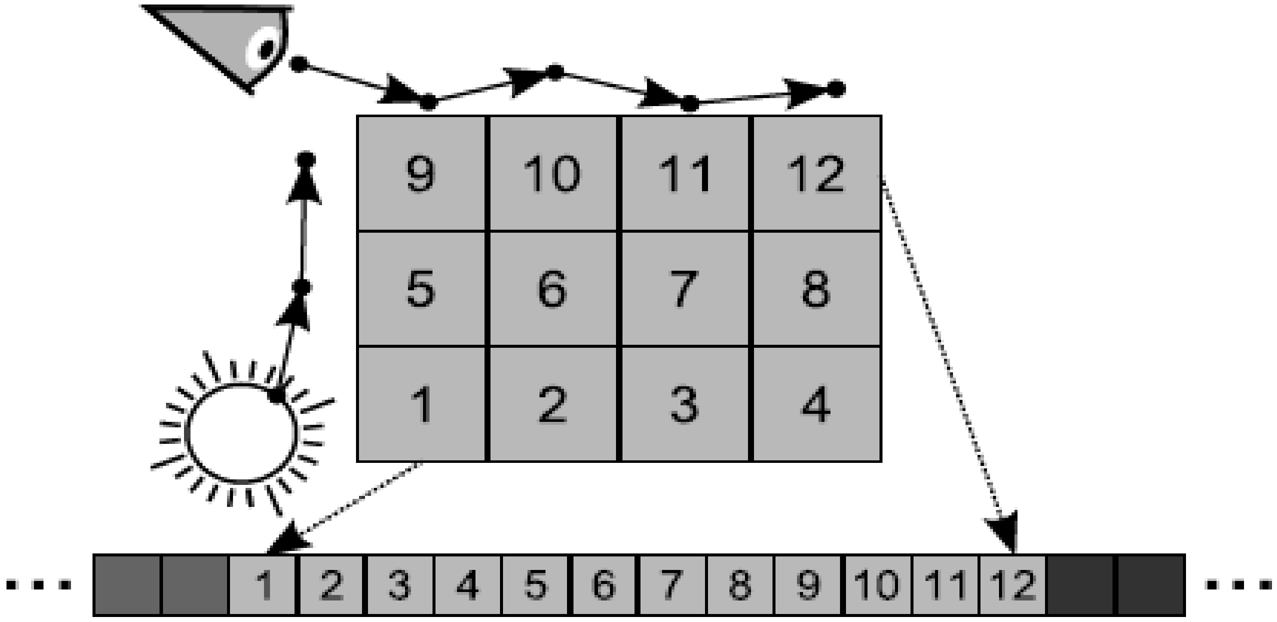
Bi-Directional Path Tracing (BDPT)

- Two phases
  - Random walk phase ~50%
  - Connection phase ~50%
- Stochastic termination results in uneven workload
- Same idea as SPT

Bi-Directional Path Tracing (BDPT)



Mapping to connection stream

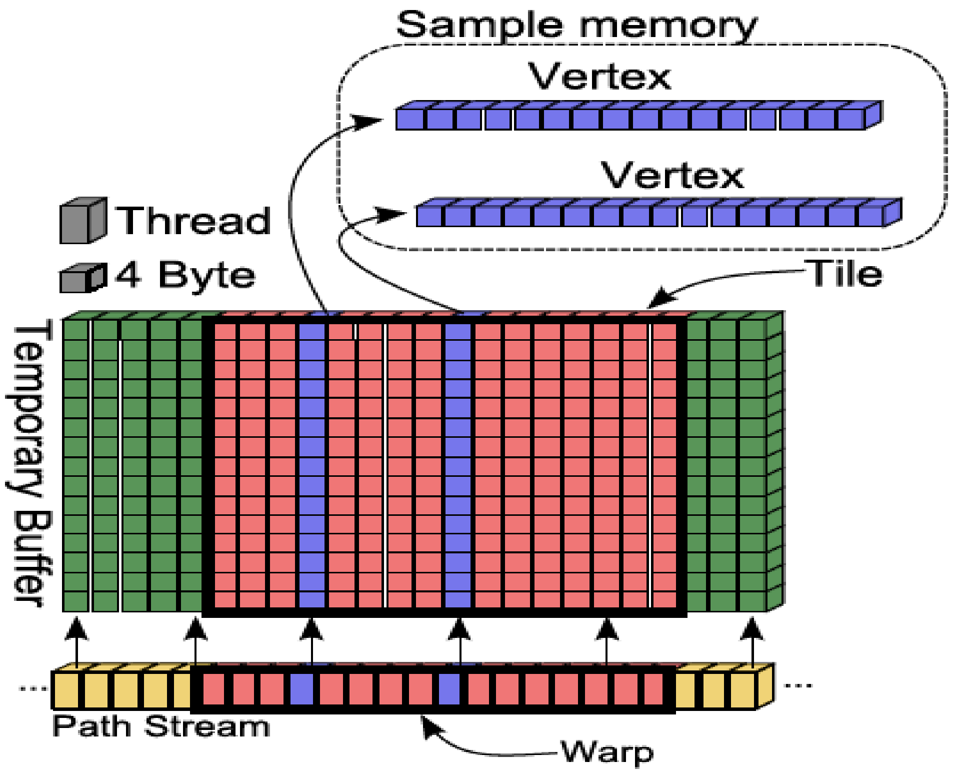


Parallel connect

- Mapping
- Parallel scan
  - Obtain sequence of prefix sums
- Evaluate all connections in parallel
  - With use of Binary Search
  - Inverse mapping

Vertex Scattering

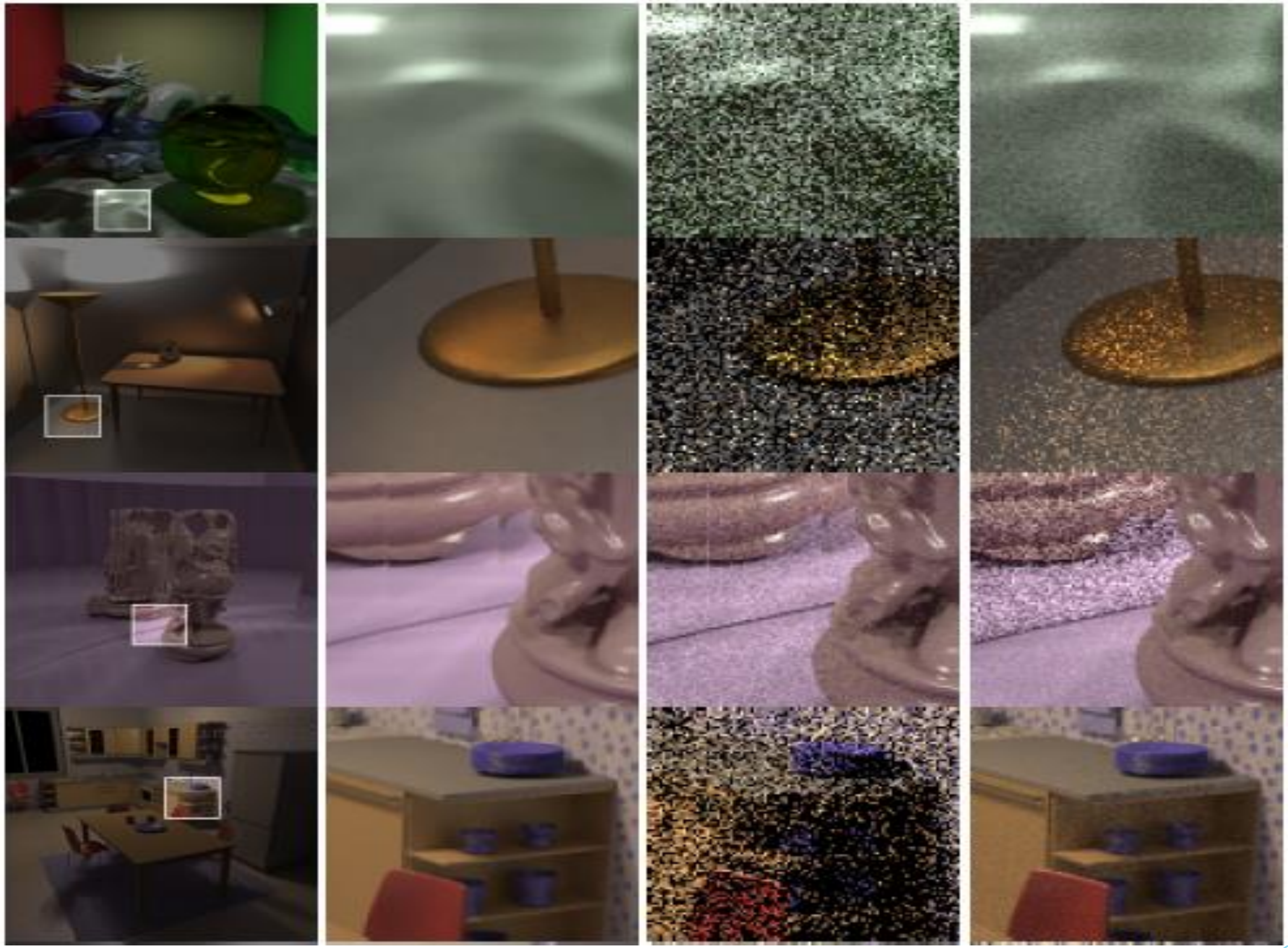
- Multiple threads in warp work together
- All threads write path vertex to temporary 2D buffer
- Transpose tile in shared memory
- Each warp thread writes 4 bytes of every vertex to sample memory



Multiple Importance Sampling

- Contribution of each bidirectional connection
- Random walk phase
  - Calculate probabilities
  - Calculate quantities using these probabilities
- Connection phase
  - Evaluate balance heuristic weights

Results



Results

	PT			BDPT		
	CPU	GPU	Speed up	CPU	GPU	Speed up
Cornell box	0.62	7.1	11.5x	0.28	2.93	10.4x
Glass egg	1.74	19.8	11.4x	0.38	3.07	8.1x
Buddha	1.18	12.3	10.4x	0.32	3.10	9.7x
Kitchen	1.12	13.5	12.1x	0.27	3.03	11.3x

Performance in 10^6 samples/s

Conclusion

- High SIMD efficiency
  - Due to sample regeneration and stream compaction
  - Parallel connect in BDPT
- High effective bandwidth
  - Due to vertex scattering
- Only tested with Blinn-Phong shading model