Realistic Image Synthesis

- Instant Global Illumination -

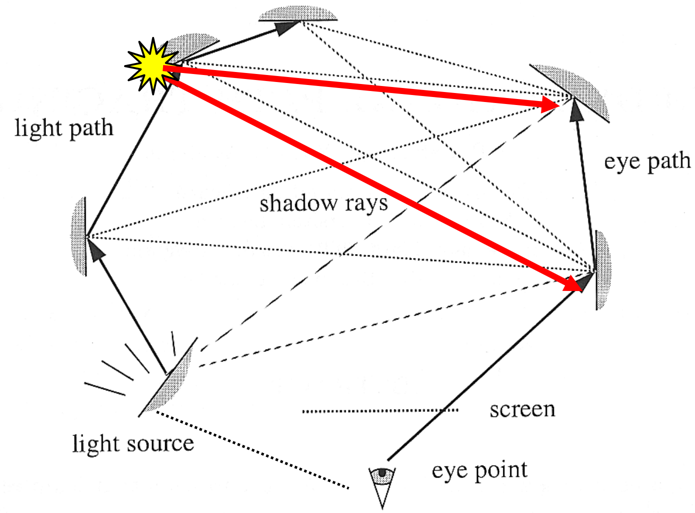
Philipp Slusallek
Karol Myszkowski
Gurprit Singh

Overview of MC GI methods

- **General idea**
  - Generate samples from lights and camera
  - Connect them and transport illumination along paths
- **Path Tracing**
  - For each pixel, generate random path from camera
  - Generate light sample on lights and connect for direct lighting
- **BiDir-Path Tracing**
  - For each pixel, generate random paths from camera and from lights
  - Connect them (in multiple ways) and transport light
- **Instant Radiosity/Global Illumination**
  - In preprocessing, generate fixed set of samples from lights (VPLs)
  - During rendering, connect to all of them and transport light
- **Lightcuts**
  - Do not connect to all VPLs, but use importance sampling
  - Create a hierarchical structure to efficiently select VPLs

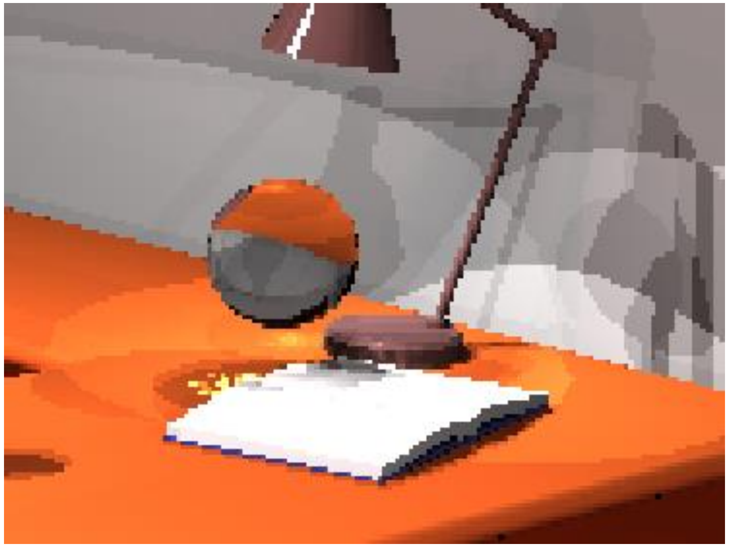
Reuse of Light Paths

- **Bidirectional path tracing**
  - Starts a new light path for every eye sample
    - Many new path being traced
  - No correlation between samples → noise
- **Idea: Reuse light samples**
  - Generate random light samples in a preprocessing pass
  - Each light sample becomes a Virtual Point Light (VPL) illuminating the entire scene (and not just one pixel)
    - Significantly reduces light samples and required tracing of rays
  - Generates correlated errors across entire image
    - Not unbiased -- but *consistent*



Instant Radiosity [Siggraph97]

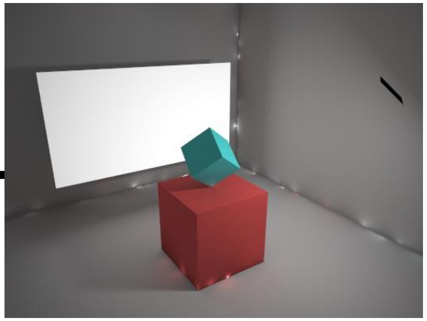
- **Trace few (10-20) rays from (area) lights**
  - These are the light paths from BiDir path tracing
  - Each contains a fraction of the energy of a light
- **Use them to generate 'virtual point lights' (VPLs)**
  - VPLs placed at every hit point along the path
    - Termination via Russian Roulette (or fixed path length (biased))
  - Trace shadow rays to all of them during rendering
  - Contains both direct and indirect diffuse illumination
- **Inherently smooth, except for sharp shadow boundaries**
  - Shadow artifacts in case of few VPLs
  - But converges consistently

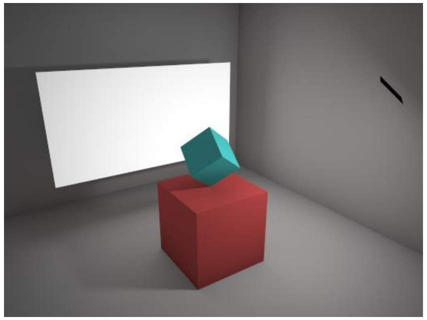


Instant Radiosity: Remarks

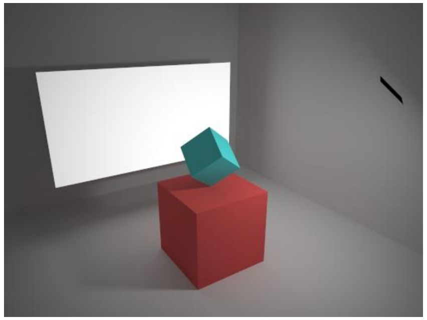
- **Approximation of illumination**
  - VPLs provide an approximation of the light distribution in a scene
  - Converge to real distribution with larger number of VPLs
- **Dealing with non-diffuse surfaces**
  - Consider BRDF when reflecting photons and during illumination
  - OK for mostly diffuse: Highly glossy surfaces would reveal VPLs
- **Shadow ray can be traced coherently**
  - Select VPLs in a coherent way (e.g. by clustering)
  - Shoot packets of ray to VPL clusters

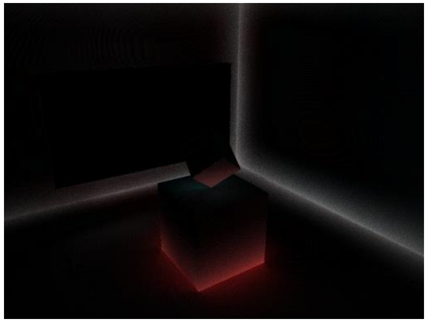
Instant Radiosity: $1/r^2$ Bias





- **Illumination Singularity**
  - Scene would be way to bright near a VPL
    - Illumination can become arbitrarily large ($1/r^2$)
    - Would be averaged out eventually
    - But may need arbitrary large number of VPLs
  - Possible Solution
    - Limit contribution such that it will not be too bright
      - Limit the $1/r^2$ term by some constant b
    - → Introduces bias: too dark, specifically in corners





- **Bias Compensation [Kollig&Keller'06]**
  - Add back missing contribution through MC sampling
  - Continue the eye paths randomly
    - Check bias: $1/r^2$-b
    - If non-negative: Scale contribution by $(1/r^2-b)/(1/r^2)$
  - Optimization
    - Limit ray length to critical region: $r < 1/\mathrm{sqrt}(b)$

Also see: Simon Brown (http://sjbrown.co.uk/2011/05/09/virtual-point-light-bias-compensation/

Instant Global Illumination

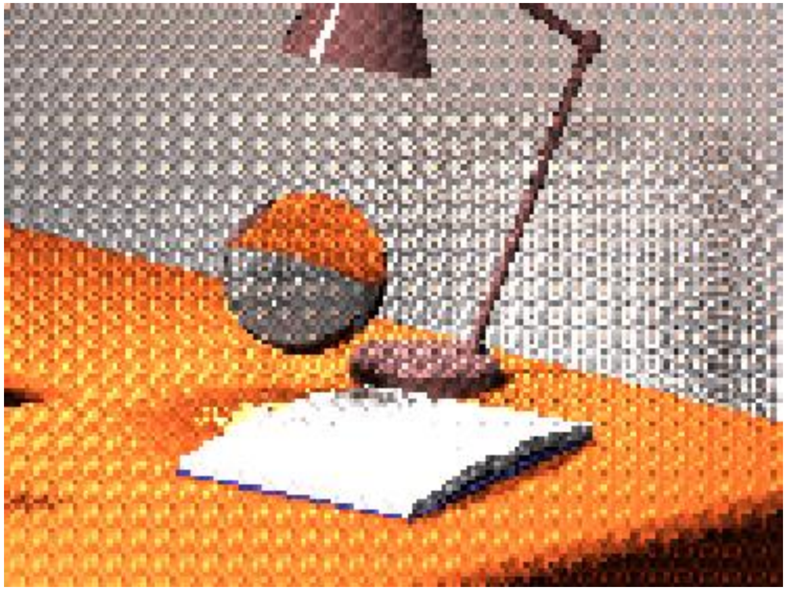
- **Idea: Use coherent form of bi-directional path tracing**
  - Speed up the computations (assumes cluster setup)
    - Ensure that shadow rays are coherent for packet ray tracing
    - Require no communication between rays
  - Combine advantages of several different algorithms
    - Instant Radiosity: smooth diffuse lighting
    - Ray Tracing: reflections, refractions, visibility testing
    - Interleaved Sampling (ILS): better quality, easy to parallelize
    - Discontinuity Buffer: removes ILS artifacts
  - Cluster limitations
    - Streaming computations
      - Master sends stream of jobs to clients
      - They eventually return the results – while already working on the next job(s)
      - Achieves almost perfect speedup (pipelining)
    - Cannot communicate between nodes (too high latency)

More Samples, (Almost) For Free

- **IR: Way to few samples for good results**
  - Hard shadow borders
- **Idea**
  - Use different sets of VPLs for different pixels
    - E.g.: 9 set of VPLs
  - Every 3x3th (or 5x5) pixel uses same set of VPLs
- **Better quality:**
  - 9 times as many VPL *per image* than without
- **Easily parallelizable**
  - Each node computes pixels with
  - same VPL id
  - Nicely scales with #nodes
- **But: Massive aliasing**
  - Can obviously see 2D grid …
  - Could be avoided if all samples are used within a pixel (supersampling) – but we aim at speed here!

1	2	3	1	2
4	5	6	4	5
7	8	9	7	8
1	2	3	1	2
4	5	6	4	5

VPL sets used per pixel



Remove Aliasing

- **Idea: Discontinuity Buffer [Keller, Kollig]**
  - Filter irradiance among neighboring pixels
    - Use the same filter width (3x3)
    - Smoothing/removal of ILS-artifacts
    - Like irradiance caching, but more stable
- **Only filter in smooth regions**
  - Must detect discontinuities
  - Criterion: normal & distance
    - Ignore pixels that differ too much
- **Problem: Clients don’t have access to neighboring pixels!**
  - Filtering has to run on the server
  - High server load
- **Server has to get additional data**
  - Normal, irradiance, distance
  - High network bandwidth !

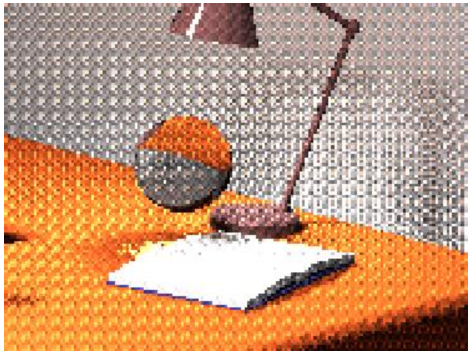
1	2	3	1	2
4	5	6	4	5
7	8	***9***	7	8
1	2	3	1	2
4	5	6	4	5

Finally, Add QMC

- **Use Randomized Quasi Monte Carlo [Keller et al.]**
  - Used for generating the light sources
  - Faster convergence, especially for small sampling rates
  - Can be combined easily with interleaved sampling
- **Plus: ‘Technical’ advantages of QMC**
  - Fast random number generation (table lookup + bit-ops)
  - Can reproduce any sequence of samples based on single seed value
    - Can easily synchronize different clients on same data
    - Each client can reproduce the sample set of any other client
  - Avoid ‘jumping’ of VPLs:
    - Just start with same seed every frame
  - For progressive convergence, just advance the seed value…
    - QMC sequences perfectly combine into the future…

Summary

- **Base ingredient**
  - Instant Radiosity + Ray Tracing
  - Plus fast caustic photon maps
- **Combine with Interleaved Sampling**
  - Better quality
  - Parallelizable
- **Remove artifacts with Disco-Buffer**
  - Faster convergence
  - Better parallelizability
- **Use randomized QMC**
  - Low sampling rates, parallelizability
- **Result: Definitely not perfect**
  - But not too bad for *only ~20 rays/pixel* !

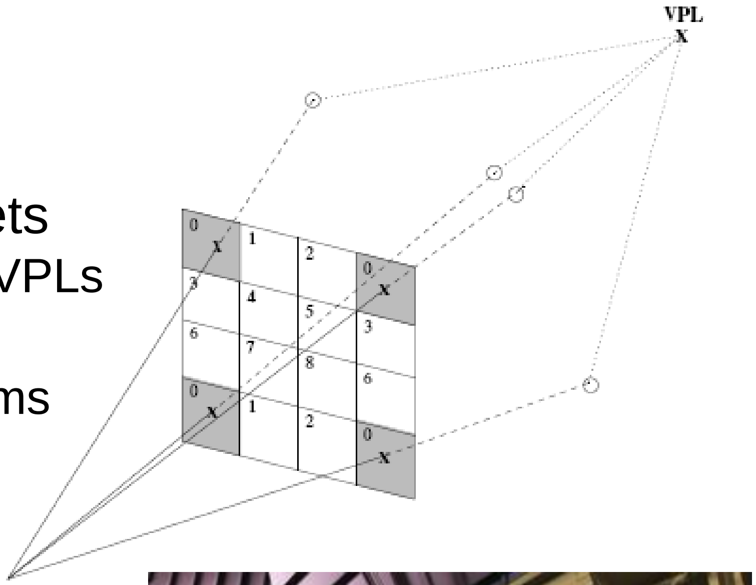


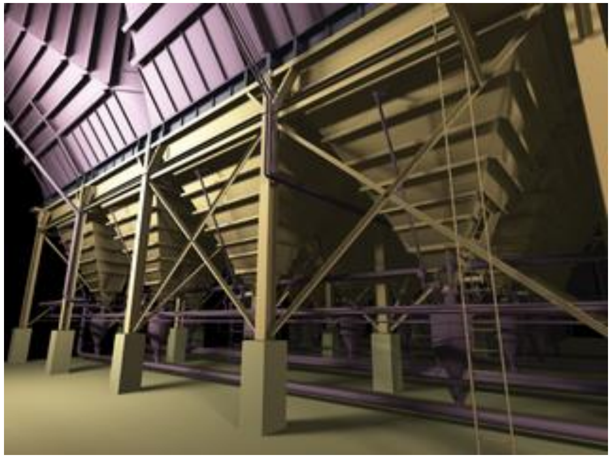
Remaining Issues

- **Missing scalability**
  - Render nodes each compute some tiles of the entire image
    - Dynamically selected at runtime (load balancing)
  - All data must be send to master for filtering
    - Colors, normals, depths
  - Approach limited by network bandwidth
- **Filtering on the clients**
  - Requires information from neighboring pixes
  - All clients need to compute all VPLs
    - Not great, but doable
    - They often did compute multiple sets anyway
      - Because of dynamic load balancing
- **Filtering overhead was too costly**
  - Repeatedly test and sum up blocks of 3x3 pixels

Scalable IGI

- **Approach**
  - Assign tiles of pixels to clients
    - Must include border for filtering
    - Overhead is ~10% for 40x40 pixels
  - Trace ray in interleaved coherent packets
    - Use SIMD across pixels with the same VPLs
  - Filtering of tile can be done on client
    - Constant time filtering using running sums
    - Add/subtract only at border of domain
    - Must only send final color
  - Low-cost antialiasing
    - Nx supersampling: Assign VPLs into N groups
    - Trace different primary ray for every group
    - Connect this hit point to VPLs of group and average (again interleaved sampling)
  - RQMC sampling of light sources
  - SIMD shader interface

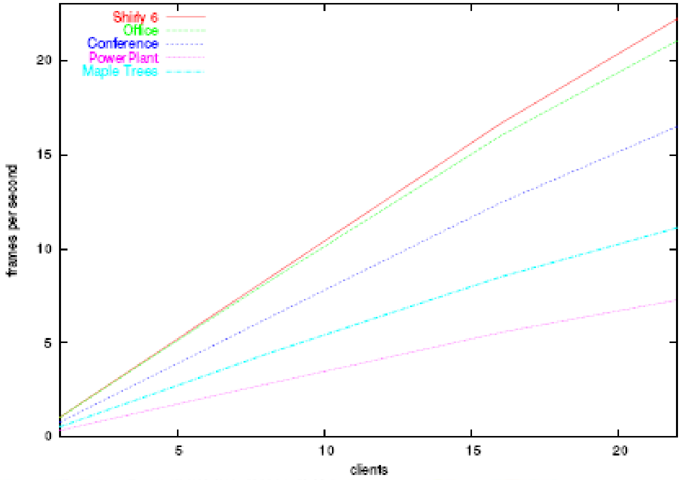


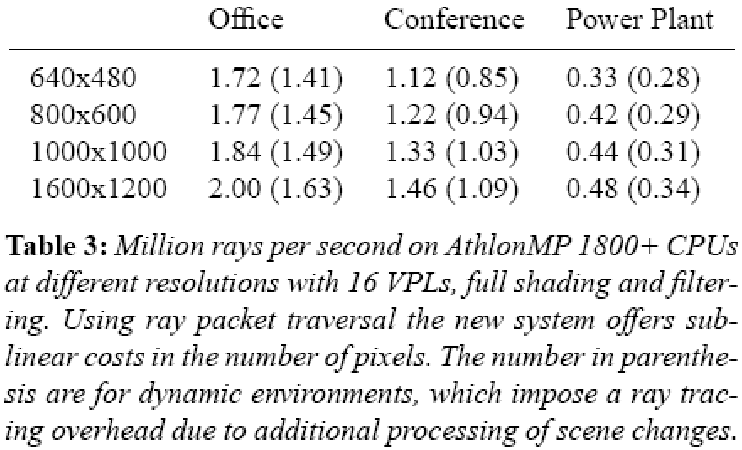


IGI with 50 Mio polygons

Results

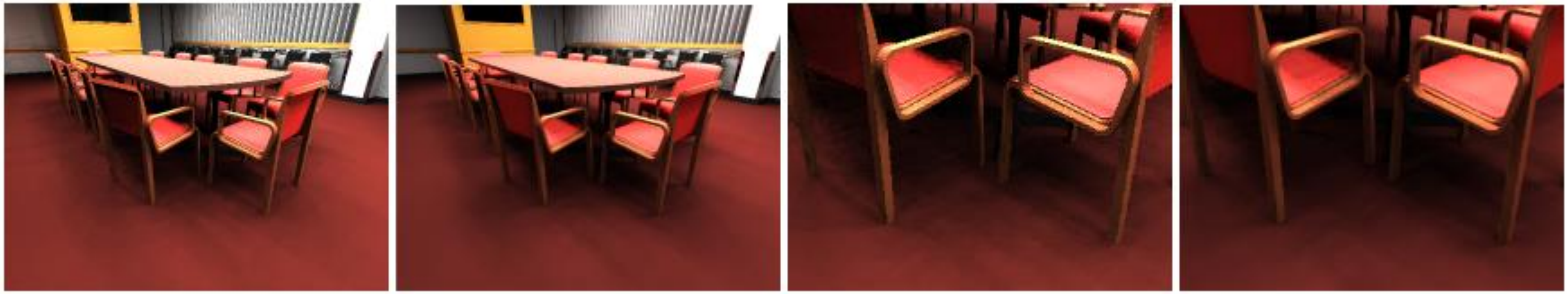
- **Performance**
  - Faster by 2.5-3x
  - Almost perfect scalability (> 20 fps) plus good use of coherence





	Office	Conference	Power Plant
640x480	1.72 (1.41)	1.12 (0.85)	0.33 (0.28)
800x600	1.77 (1.45)	1.22 (0.94)	0.42 (0.29)
1000x1000	1.84 (1.49)	1.33 (1.03)	0.44 (0.31)
1600x1200	2.00 (1.63)	1.46 (1.09)	0.48 (0.34)

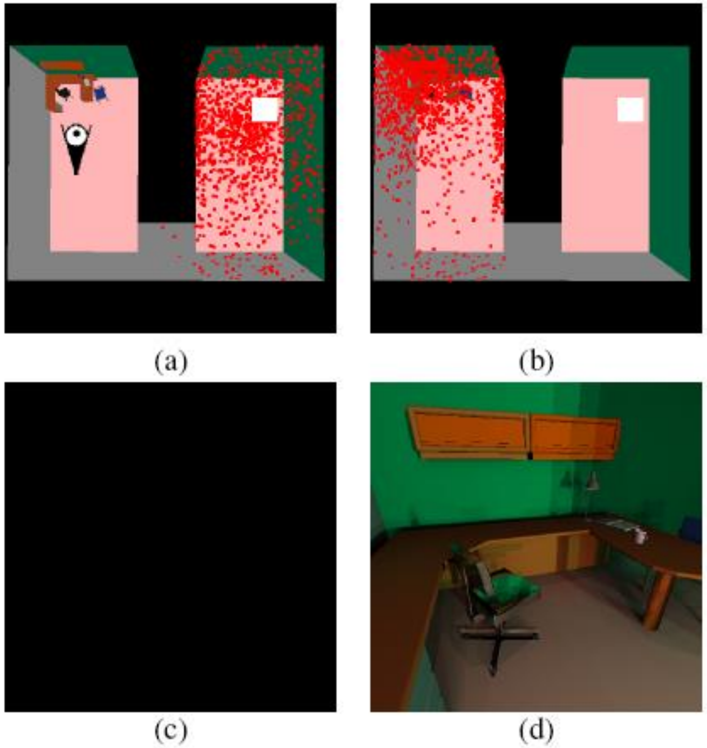
Table 3: *Million rays per second on AthlonMP 1800+ CPUs at different resolutions with 16 VPLs, full shading and filtering. Using ray packet traversal the new system offers sublinear costs in the number of pixels. The number in parenthesis are for dynamic environments, which impose a ray tracing overhead due to additional processing of scene changes.*



Equal compute time images comparing old and new scalable approach

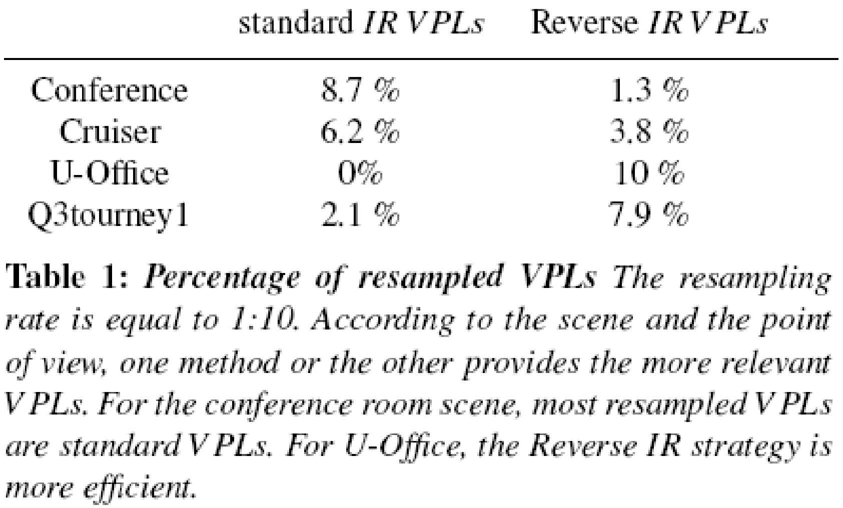
Bidirectional Instant Radiosity

- **Idea [Segovia, EGSR 06]**
  - Generate VPLs where they matter
  - Each VPL should have equal contribution to *image*
- **Approach**
  - Generate “VPLs” from light AND camera
    - N/2 samples each
  - Estimate illumination at reverse VPLs using Instant Global Illumination
    - Shoot some paths each (e.g. ~5)
    - Gather light from forward/light VPLs
    - Reverse VPLs act as proxies
  - Estimate importance using M (e.g. 5-10) paths each from camera (length 2)
  - Resample VPLs (e.g. select 10%) according to contribution to camera
  - Estimate accurate pdf for VPLs using more camera path (e.g. 50)
  - Use selected VPLs during rendering



200 selected VPLs

Results



	standard *IR VPLs*	Reverse *IR VPLs*
Conference	8.7 %	1.3 %
Cruiser	6.2 %	3.8 %
U-Office	0%	10 %
Q3tourney1	2.1 %	7.9 %

Table 1: ***Percentage of resampled VPLs*** *The resampling rate is equal to 1:10. According to the scene and the point of view, one method or the other provides the more relevant VPLs. For the conference room scene, most resampled VPLs are standard VPLs. For U-Office, the Reverse IR strategy is more efficient.*

Results

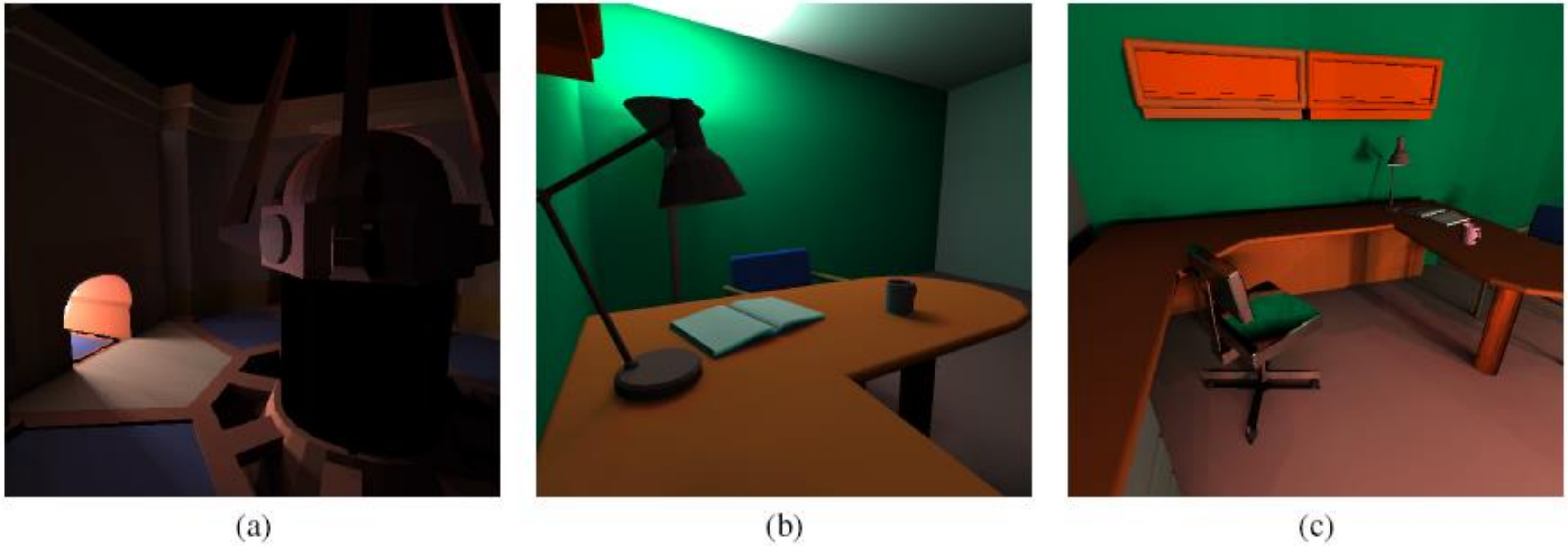


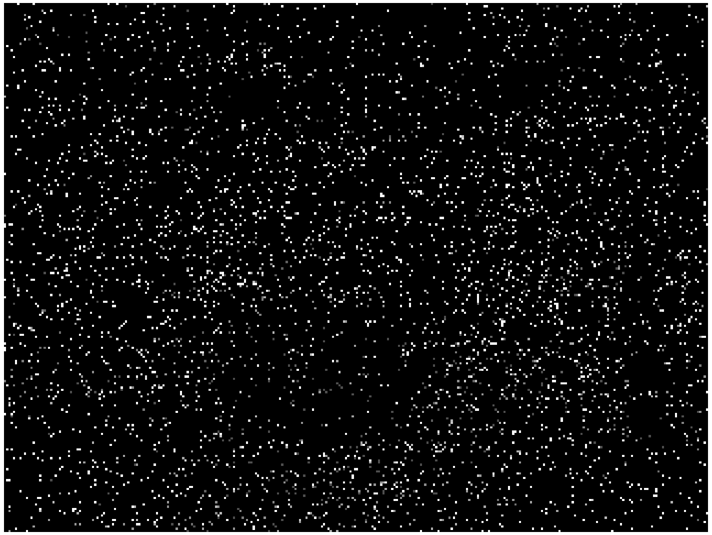
Figure 6: *Sampling Performances.* *Several pictures computed with the sampling techniques described in the article. Bidirectional sampling finds the relevant V PLs no matter the visibility layout. (a) A part of q3tourney1 (courtesy of ID software) indirectly lit trough a small corridor. Lights make at least 3 bounces before coming into the room. Most of the V PL are found with reverse IR strategy. (b) A simple office indirectly lit by a halogen lamp easily handled with Standard Instant Radiosity. (c) The U Office (raytraced). The scene is completely indirectly lit through a small corridor. Bidirectional IR quickly finds the relevant V PLs.*

Dealing With Many Lights [EGSR'03]

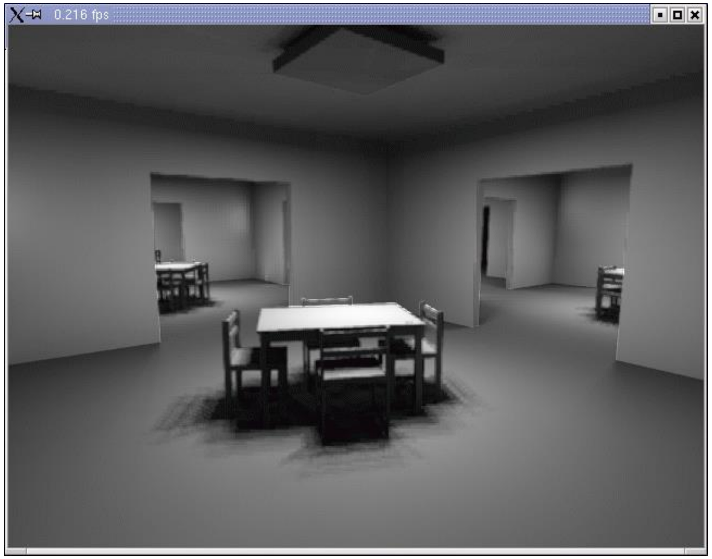
- **Problem**
  - Efficiency drops severely in highly occluded environments
    - Think: Large building with many (illuminated) rooms
  - Probability of light being visible is low
    - Must generate many VPLs to get a good statistics, at all
    - Must send many shadow rays to VPLs that are almost certainly occluded
- **Idea**
  - Ignore any lights that do not contribute illumination
  - Avoid computing from lights, would load data for entire (huge) scene
- **Solution**
  - Estimate importance of lights using path tracing (1 path per pixel)
    - Gives ~1-2 million samples, could additionally average of last few frames
  - Use importance sampling to distribute VPLs from lights
- **Issues**
  - Average is over entire image (might miss lights illuminating small area)
  - Can cause temporal aliasing (flickering) due to randomness of VPLs
    - Somewhat offset by deterministic QMC sampling (mostly same VPLs/light)

Estimate Importance of Lights

- **Example: “Shirley10”**



Estimate (1 sample/pix)



Real scene (10x10 rooms, 1 light each)

  – Path traced image hardly recognizable ...
    - But: Estimate correct up to a few percent
  – Used for importance sampling of lights