SIGGRAPH2010

The People Behind the Pixels

Real-time Diffuse Global Illumination in CryENGINE 3

Anton Kaplanyan
antonk@crytek.de

Crytek GmbH

- 10 years in game development
- ~650 employees in 5 offices across Europe
- Multicultural company with 30+ languages
- Shipped:
  - FarCry on CryENGINE 1 in 2001 (PC only)
  - Crysis and Crysis Warhead on CryENGINE 2 in 2007-8 (PC only)
- Multi-platform consoles-ready CryENGINE 3
- Currently working hard on Crysis 2…
  - Q4 2010

Global Illumination in games

Why dynamic Global Illumination?

- Most games use precomputed indirect lighting (Lightmaps, PRT etc.)
  - Means static scene/lighting
- CryENGINE 3® includes following features:
  - Dynamic deferred lighting
  - Objects' breakability as a part of game-play
- That cancels out all precomputed GI methods
  - We've tried out most of it (including Lightmaps, PRT, RAM etc)
- But we came up with a solution….

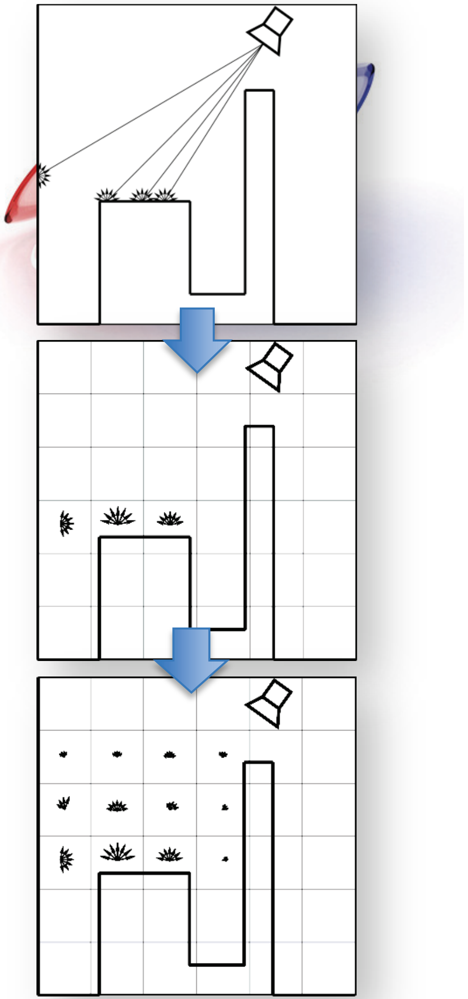
Diffuse Global Illumination in Crysis 2™

Diffuse Global Illumination in Crysis 2™

CASCADED LIGHT PROPAGATION VOLUMES

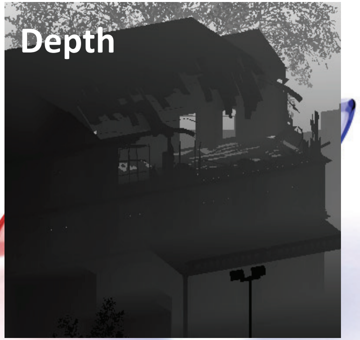
Core Idea

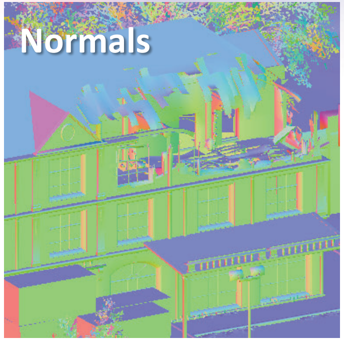
1. Sample lit surfaces
   – Treat them as secondary light sources
2. Cluster samples into a uniform coarse 3D grid
   – Sum up and average radiance in each cell
3. Iteratively propagate radiance to adjacent cells, works only for diffuse
4. Lit the scene with the resulting grid

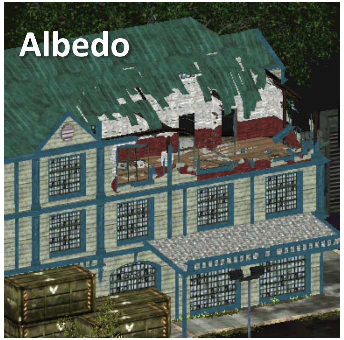


Sampling the scene for GI

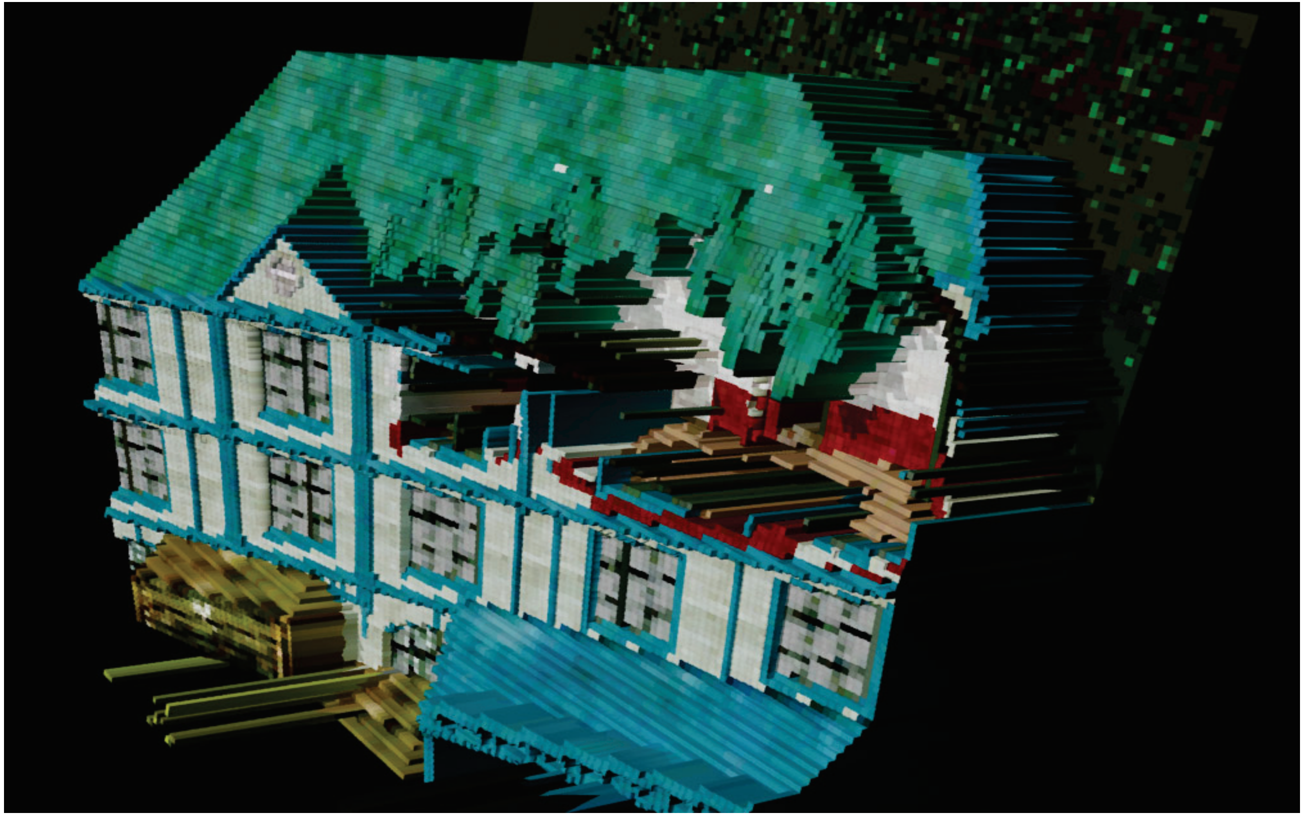
- We use *surfels* (aka “points”, “disks”)
  - Surfel == surface element
- All lit surfels can be flattened into 2D map in light’s space
- Reflective Shadow Maps [DS05]
  - Fastest way to sample lit surfels on GPU
  - Even excessively

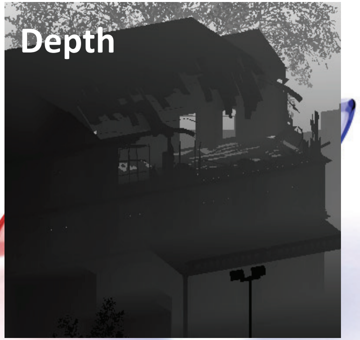


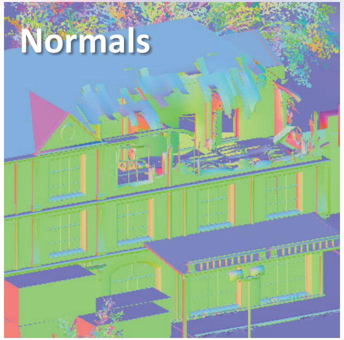


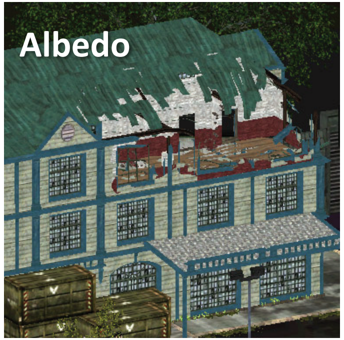


Sampling the scene for GI









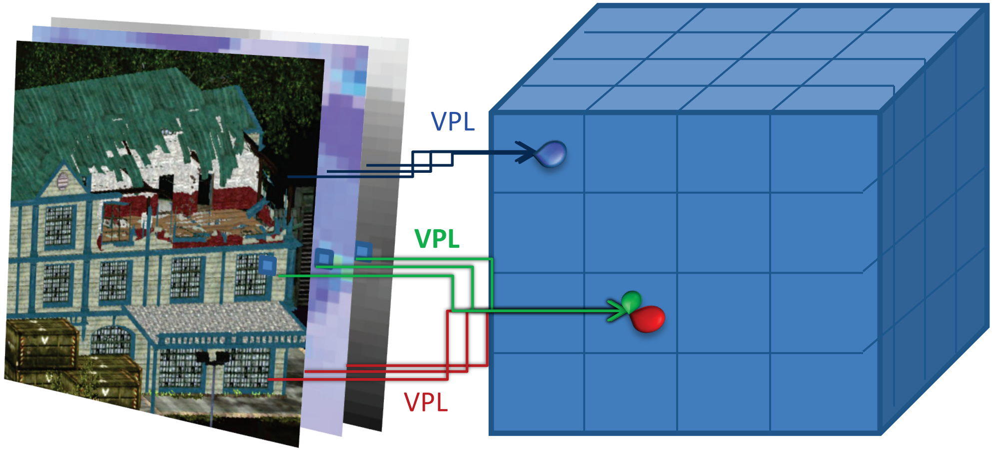
Clustering Surfels

- Lit surfels represented as *Virtual Point Lights*
  - Comes from Instant Radiosity approach [Keller97]
- Distribute each surfel into the closest grid cell
  - Similar to PBGI, light-cuts and radiosity clustering
- Convert all VPLs into outgoing radiance distribution
  - Represent in Spherical Harmonics with lower bands
  - Sum it up in the center of owner grid cell
  - Done completely on GPU using rasterization

Propagation

Reflective shadow maps

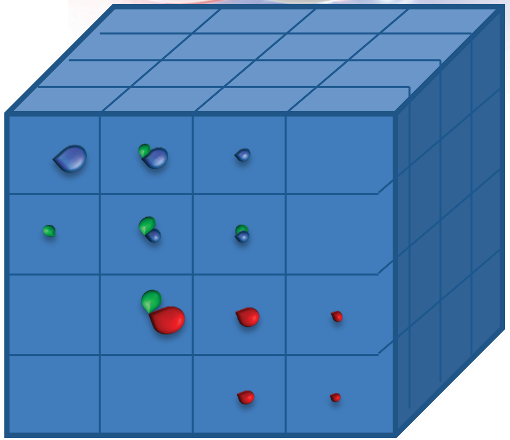
Radiance volume gathering



A set of regularly sampled VPLs of the scene from light position

Discretize initial VPL distribution by the regular grid and SH

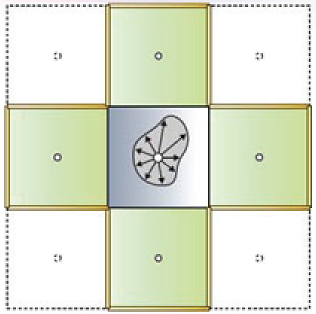
Iterative propagation



Propagate light iteratively going from one cell to another

Propagation, cont'd

- Local cell-to-cell propagation across the 3D grid
  - Similar to SH Discrete Ordinate Method for participating media illumination [GRWS04]
- 6 axial directions with contour faces as a propagation wave front
- Accumulate the resulting SH coefficients into the destination cell for next iteration

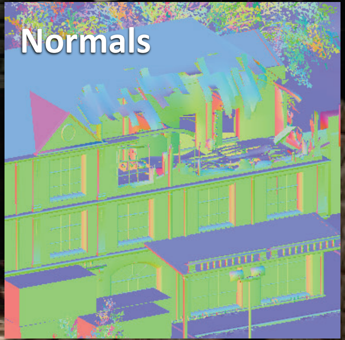


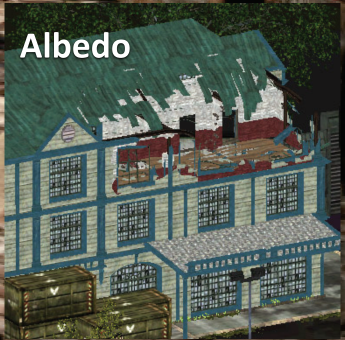
propagation along axial directions

Final scene rendering with LPV

- Look-up resulting grid 3D texture at certain position with h/w trilinear interpolation
- Convolve the irradiance with cosine lobe of surface's normal being illuminated
- Apply dampening factor to avoid self-bleeding
  - Compute directional derivative towards normal
  - Dampen based on gradient deviation from the intensity distribution direction

Results



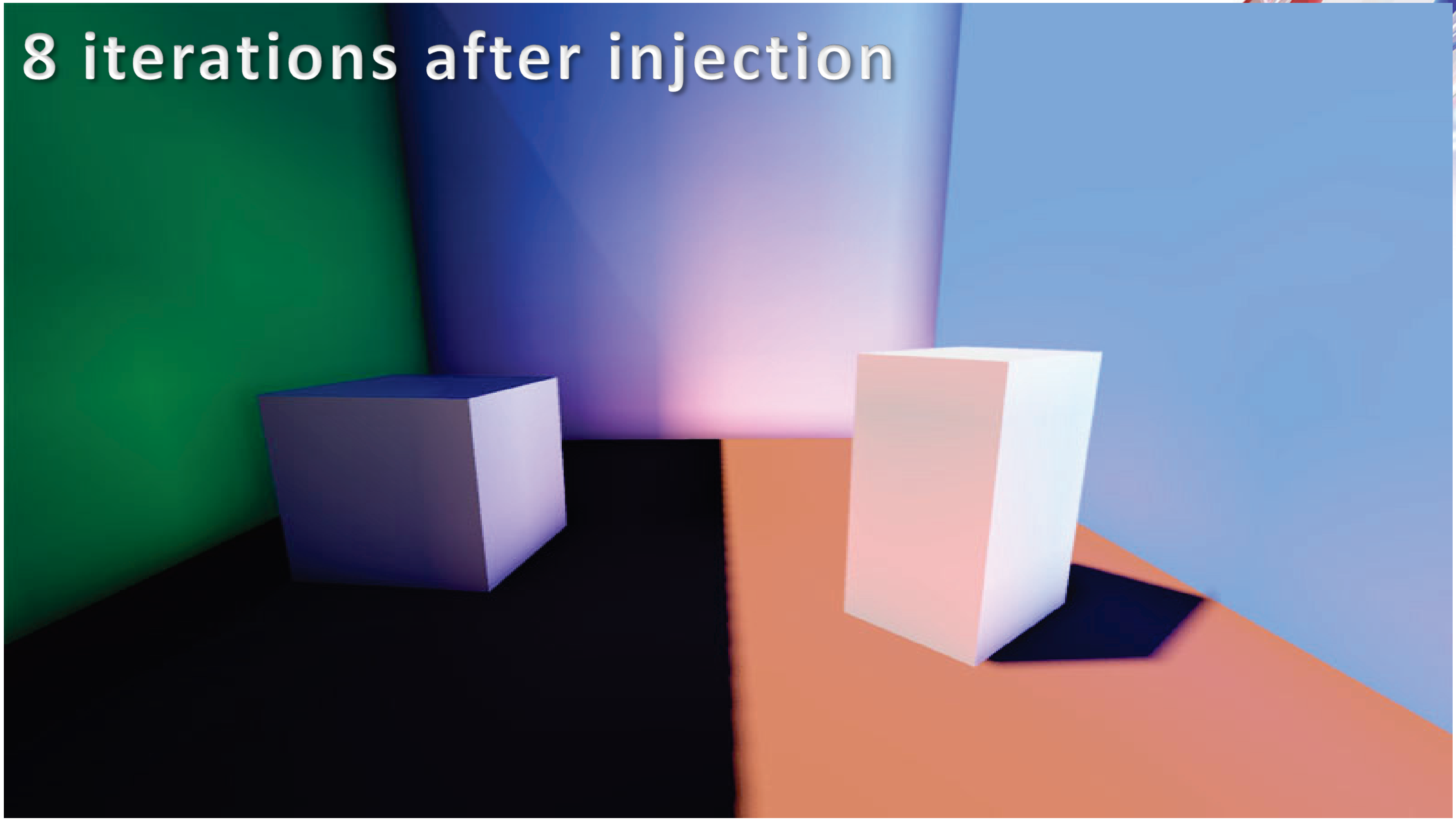


Results

Results

Results

Propagation example

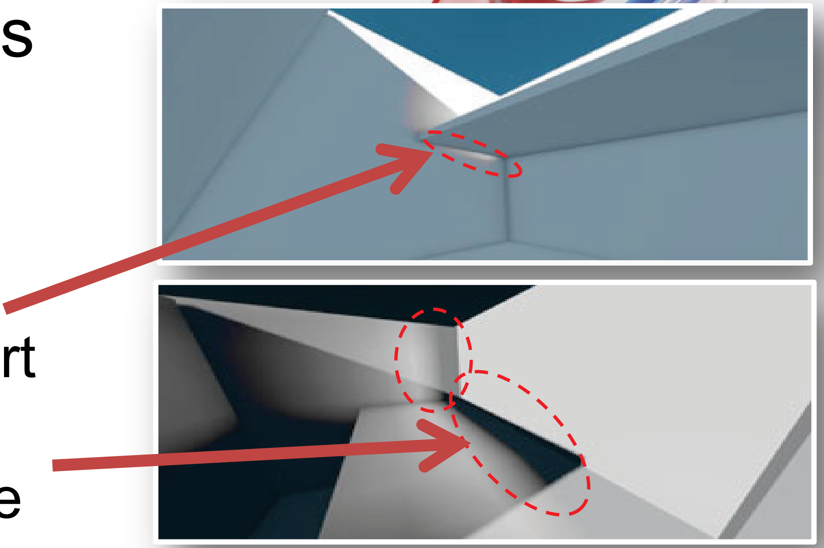


Stabilizing solution

- Spatial stabilization
  - Snap RSM by one pixel for conservative rasterization
  - Snap LPV by one grid cell for stable injection
- Self-illumination
  - Half-cell VPL shifting to normal direction during RSM injection
- Temporal coherence and reprojection
  - Temporal SSAA with reprojection for RSM injection

Limitations of the method

- Only **diffuse** inter-reflections
- Sparse spatial and low-frequency angular **approximations**
  - **Light diffusion:** light transport smears in all directions
  - **Spatial discretization:** visible for occlusion and very coarse grids



- **Incomplete** information for secondary occlusion

Multi-resolution approach

- Render several nested RSMs at different resolutions
  - Inspired by cascaded shadow maps technique
  - Simulates uneven multi-resolution rendering on GPU
  - Distribute objects into **different** RSMs based on their size
- Inject RSMs into corresponding LPVs
  - Create nested LPV grids that bound RSM frustums
  - Do propagation and rendering independently
  - Propagate from inner LPV to outer one

Cascaded Light Propagation Volumes

Global Illumination with 3 Light Propagation Volumes

Extensions

- Transparent objects
- Lighting caching for massive lighting approximation
  - Inject analytical radiance into grid cells covered by light
- Secondary occlusion with additional occlusion grid
  - Multiple bounces possible using the same trick
- Glossy reflections by partial matching in LPV
- Participating media illumination
  - Comes inherently from the propagation process' nature

Global Illumination on particles

Why does it work so good?

- Human perception of Indirect Lighting
  - Very sensitive for contact lighting (corners, edges etc.)
  - Indirect lighting is mostly in low frequency
    - Even for indirect shadows
    - Smooth gradients instead of flat ambient in shadow
  - Approximated as diffusion process in participating media
- Cascades: importance-based clustering
  - Emitters are distributed across cascades based on its size

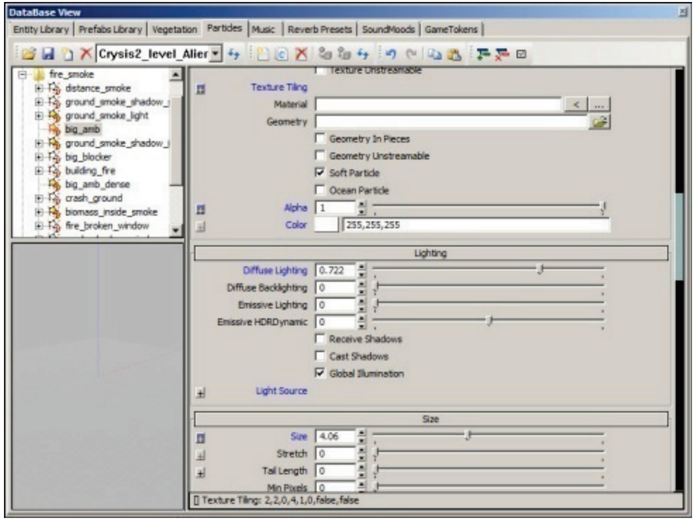
How far are we from ground truth?

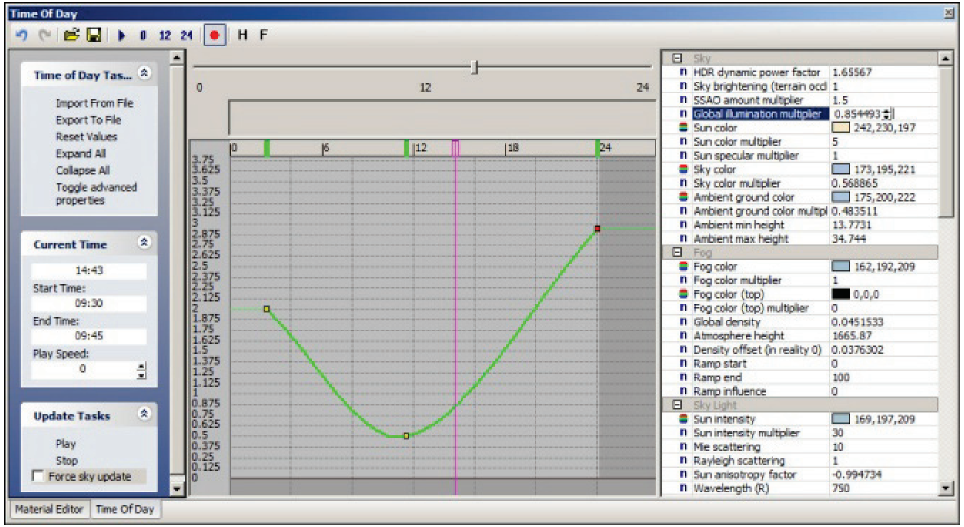
Comparison

	Lightmaps	Precomputed Radiance Transfer	Light Propagation Volumes
Image quality	Very good	Good	Good
Memory budget	Medium	Medium	Low and fixed
Dynamic lighting	-	+	+
Dynamic objects	-	-	+
Secondary Occlusion	+	+	+ (via extensions)
Multiple bounces	+	+	- (unstable solution)
Area covered	Whole scene	Whole scene	Limited (w/o cascades)
Auxiliary data	Moderate	Huge	None
Affecting production	Strong	Strong	Low

Tools for game production

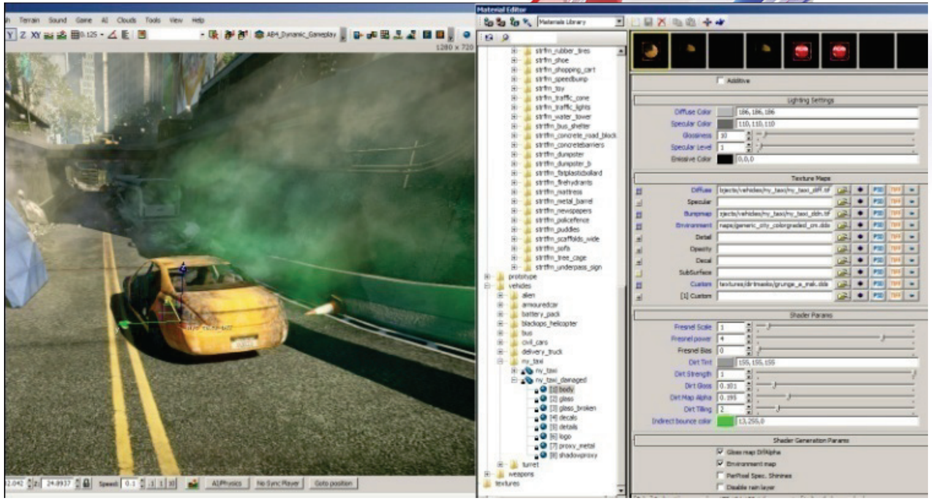
- GI editing tools for artists:
  - GI intensity for each direct light contributing into GI
  - Mark objects as non-casters and/or non-receivers





Tools for game production

- GI tools for artists:
  - Per material indirect color and intensity
  - Optionally apply on any transparent objects and particles



  - Clip areas: provides control over indoors
  - Transition areas: provides smooth GI changes across level areas / game events

Combination with other techniques

- Multiply with SSAO to add micro-occlusion details
- Deferred environment probes
  - Combined to augment for distant GI
- Fill lights and deferred lights
  - Simulating GI with fill lights at some places
  - Important for artists for GI stylization

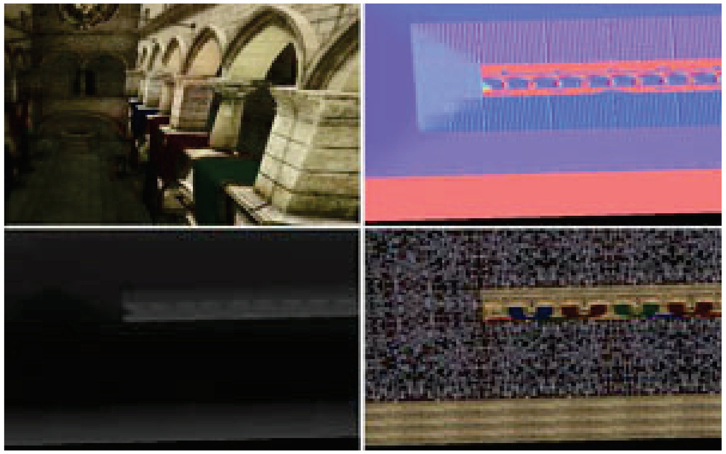
Global Illumination simulated with Deferred Lights

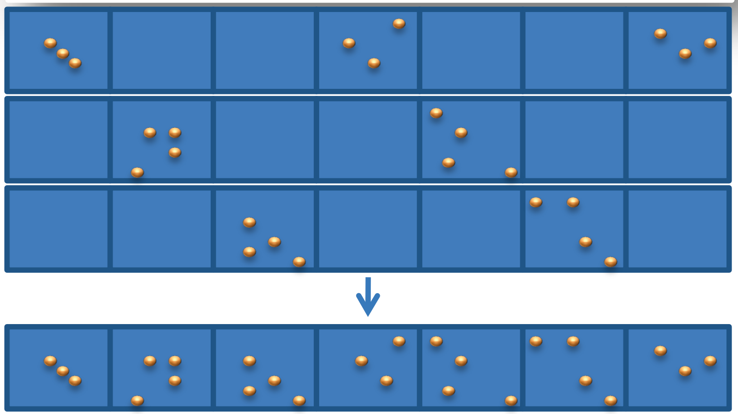
Console optimizations

- For both consoles
  - Store everything in signed **QUVW8 format**, [-1;1] with **scaling factor**
  - Use **h/w 3D textures** and trilinear filtering
- Xbox 360
  - Unwrap RT **vertically** to avoid **bank conflicts** during injection *(next slide)*
  - Use API bug **work-around** to resolve into a 3D slice
- PlayStation 3
  - Use memory aliasing for **render into 3D texture**
  - Use **2x MSAA aliasing** to reduce pixel work twice

Console optimizations , cont’d

- Render Reflective Shadow Map
  - Usually 128 x 128 is ok
- Inject each pixel into unwrapped LPV with a swarm of points
  - 16384 points in one DIP
  - Use vertex texture fetch on X360
  - Use R2VB on PlayStation 3
- Multi-layered unwrapping to avoid bank conflicts during RSM injection
- Combine LPV rendering pass with SSAO to amortize the cost





Performance

Stage	GTX 285, ms	Xbox 360, ms	PS 3, ms
RSM Rendering	0.16 (256^2)	0.5 (128^2)	0.8 (128^2)
VPL Injection	[illegible]	[illegible]	0.4
Occlusion Injection	0.02	0.15	0.15
Propagation	0.5/0.8/1.1	0.5/0.8/1.2	0.5/0.8/1.2
LPV look-up	1.4	0.9	0.9
Total	**2.1/2.4/2.7**	**2.1/2.4/2.8**	**2.6/3.0/3.4**

Depends on scene complexity

32^3 grid size

8 iterations

Depends on image size (1280x720)

Refresh once per 5 frames

Reprojection for camera movement

Performance, cont'd

Stage	GTX 285, ms	Xbox 360, ms	PS 3, ms
RSM Rendering	0.16 (256^2)	0.5 (128^2)	0.8 (128^2)
VPL Injection	0.05	0.2	0.4
Occlusion Injection	0.02	0.15	0.15
Propagation	0.5/0.8/1.1	0.5/0.8/1.2	0.5/0.8/1.2
LPV look-up	1.4	0.9	0.9
Total **(per frame)**	**1.5/1.6/1.7**	**1.1/1.1/1.3**	**1.2/1.3/1.4**

Once per 5 frames

Once per frame

Conclusion

- Full-dynamic approach, changing scene/view/lighting
- GPU- and consoles- friendly
- Extremely fast (takes **~1 ms/frame** on PlayStation 3)
- Production-eligible (rich toolset for real-time tweaking)
- Highly scalable, proportionally to quality
- Stable, flicker-free
  - Supports complex geometry (e.g. foliage)

Q&A

Anton Kaplanyan

antonk@crytek.de

Find the last version of course notes at: http://www.crytek.com/technology/presentations/

References

- [Bunnel05] Bunnel, M. 2005 "Dynamic ambient occlusion and indirect lighting", GPU Gems 2
- [Christensen07] Christensen, P. 2007. "Point-based approximated color bleeding," Tech Memo, Pixar.
- [DS05] Dachsbacher, C., and Stamminger, M. 2005. Reflective shadow maps. In Proc. of the Symposium on Interactive 3D Graphics and Games
- [GRWS04] Geist, R., Rasche, K., Westall, J., and Schalkoff, R. J. 2004. Lattice-boltzmann lighting. In Rendering Techniques 2004 (Proc. of the Eurographics Symposium on Rendering
- [KD10] Kaplanyan A., Dachsbacher C. 2010. Cascaded Light Propagation Volumes, In Proc. of the ACM SIGGraph Symposium on Interactive 3D Graphics and Games
- [KELLER97] Keller, A. 1997. Instant radiosity. In SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques