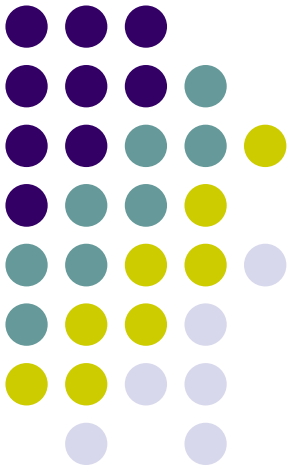
Advanced Computer Graphics
CS 563: *VPL-based Real-Time GI Algorithm*
Light Propagation Volumes

Xin Wang

Computer Science Dept.

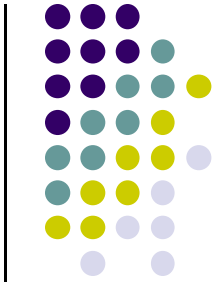
Worcester Polytechnic Institute (WPI)



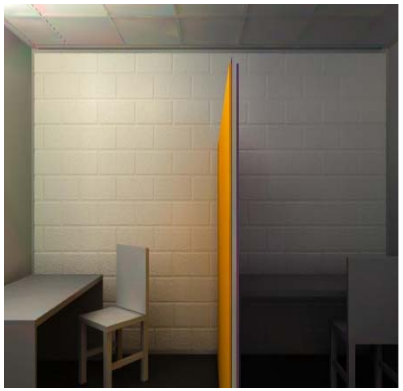
Overview

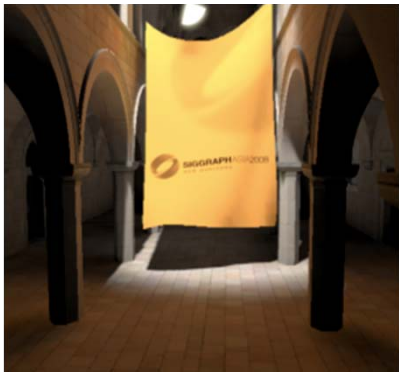

Multiple bounces
Completely dynamic
(cameras, lights,
scene)
Indirect Occlusion
Glossy reflections
Real-time, suitable for
game production

Previous Work

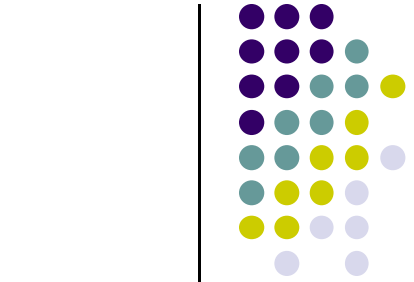


- Irradiance Volumes[Greger et al. 1997]/SH Irradiance Volumes[Tatarchuk 2004]
- VPL Visibility[Laine et al. 2007]
- Lattice-Boltzmann Lighting[Geist et al. 2004]
- Light Propagation Maps[Fattal 2009]

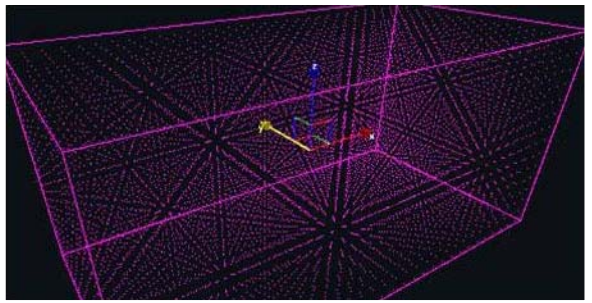


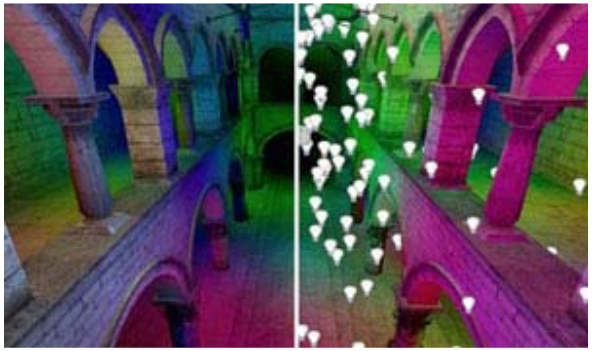



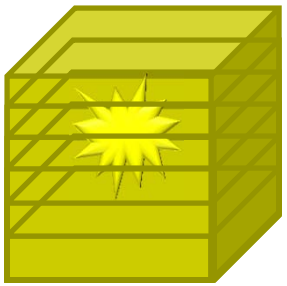
Massive Lighting



- Render sliced unwrapped light box into LPV (spatial overdraw vs screen-space, maximum 1024x32 pixels)
- Convert light's radiant intensity into SH
- Shadows are not supported

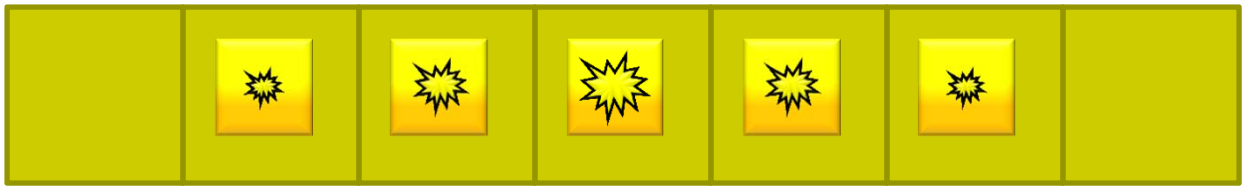




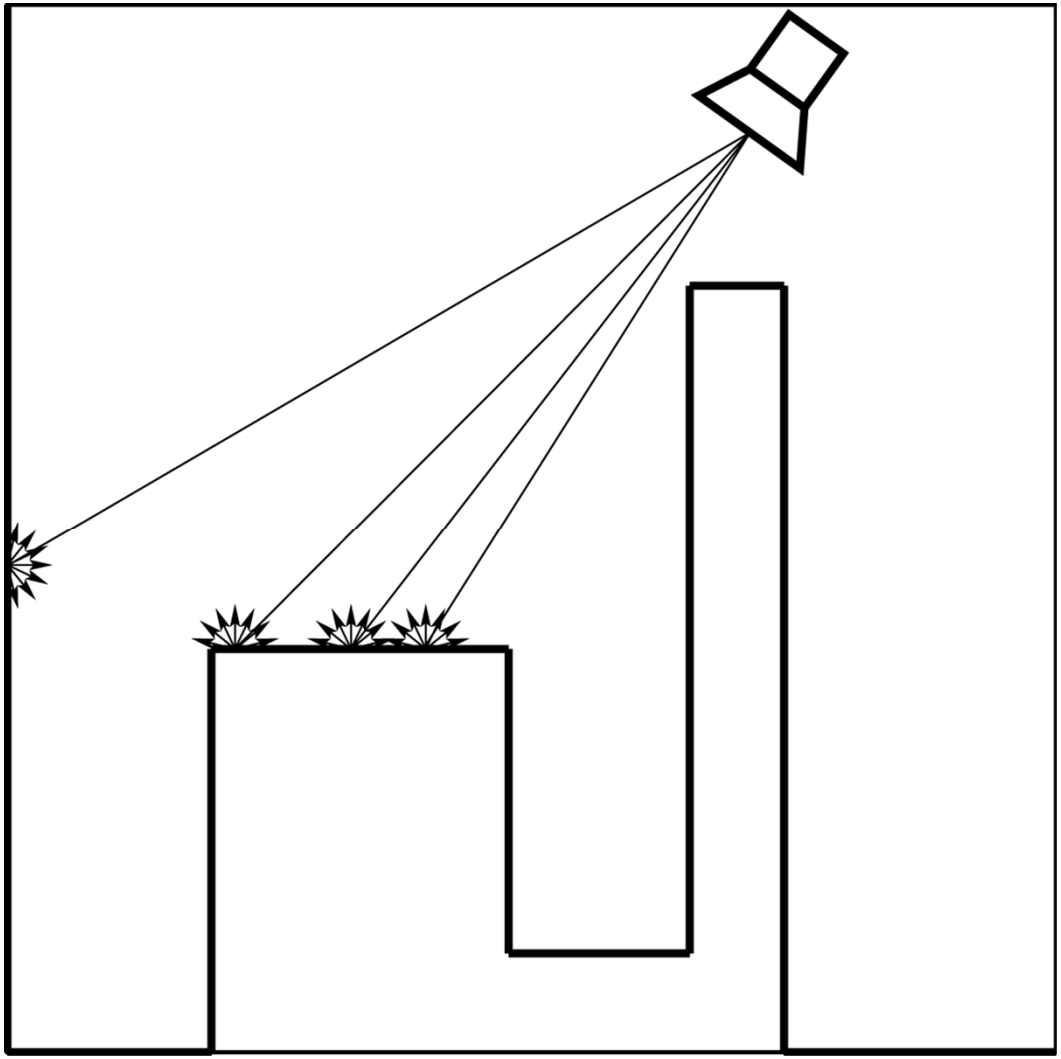


Light in the Light Propagation Volume

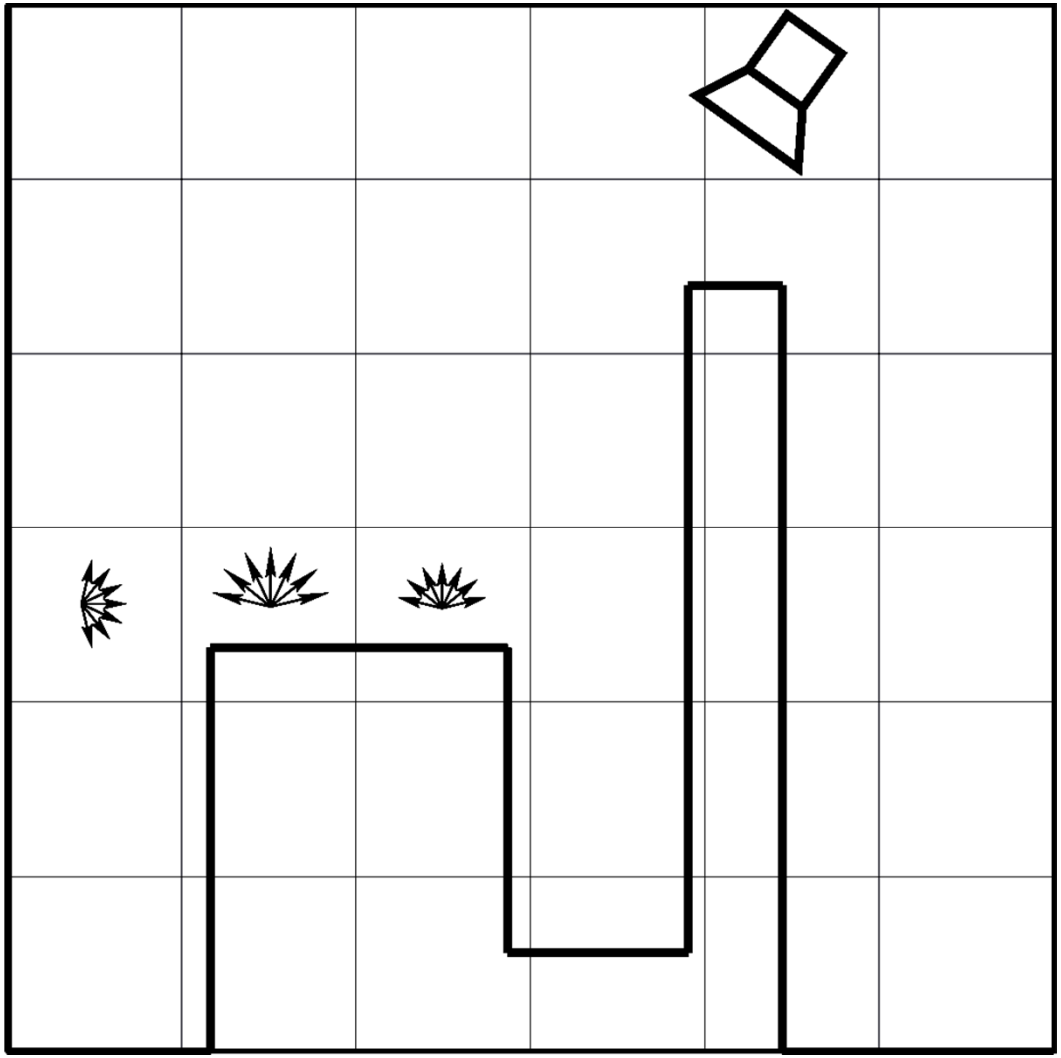
Coverage in unwrapped render target



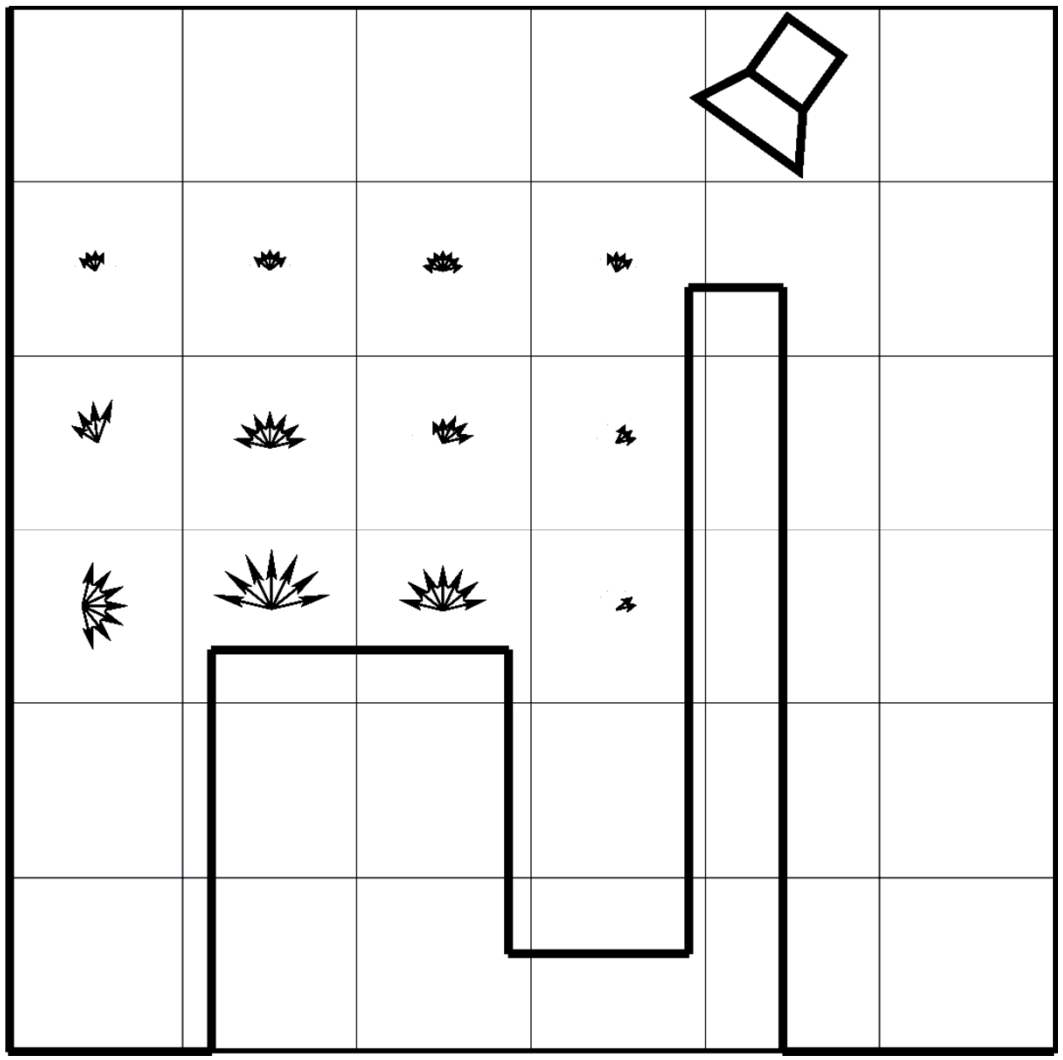
Basic Idea



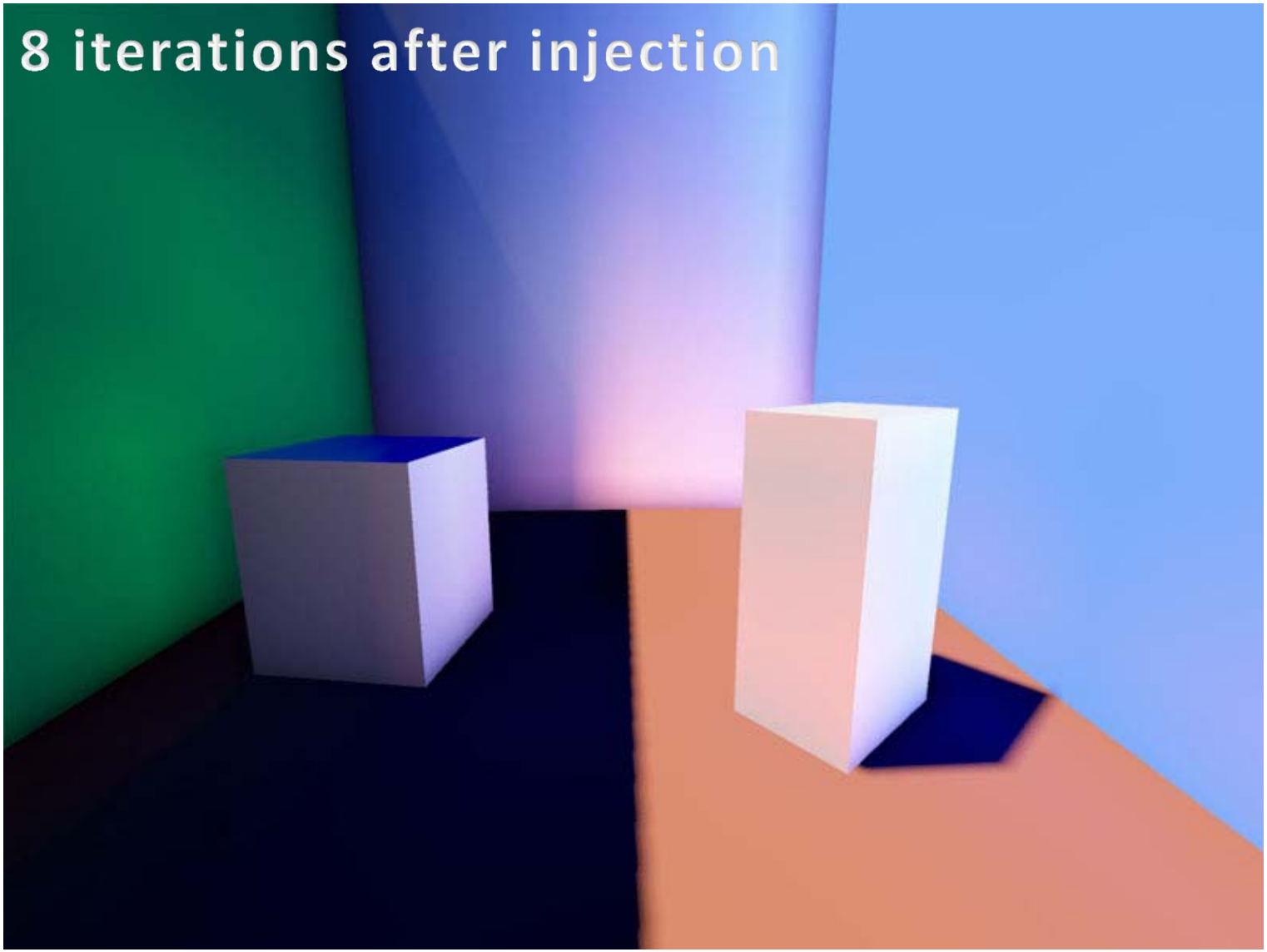
Basic Idea



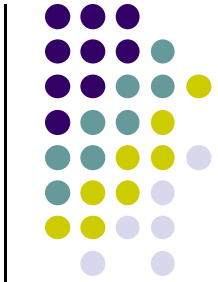
Basic Idea



Propagation Example

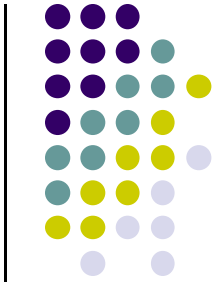


Light Propagation Volume Outline



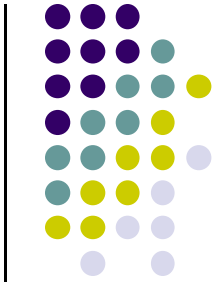
- Use **many-lights** approach to capture sources of indirect lighting
- **Sample** directly lit surfaces and **initialize** 3D grid
- Represent directional distribution with **Spherical Harmonics**
  - Inspired by **SH Irradiance Volumes** [Tatarchuk04]
- **Iterative, local propagation**: cell-to-cell

Secondary Light Sources



- Sample lit surface elements
- Grid initialization
- Light propagation in the grid
- Scene illumination with the grid

Reflective Shadow Maps

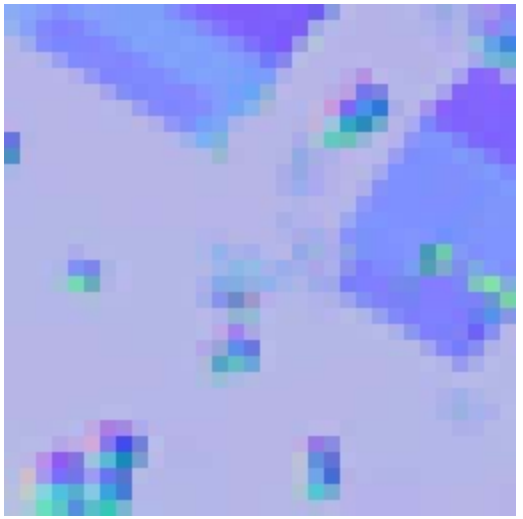


- Reflective Shadow Map – efficient VPL generator
- Shadow map with MRT layout: depth, color and normal

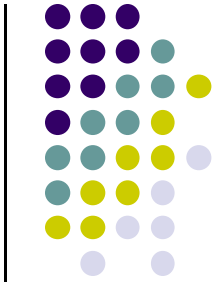
Depth

Flux

Normal



Injection



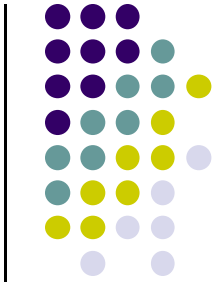
Sample lit surface elements

Grid initialization

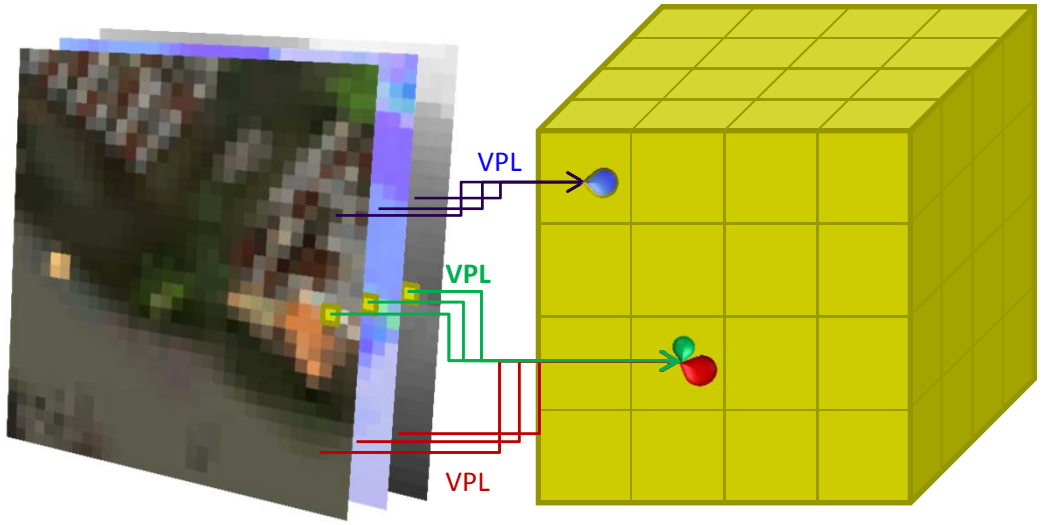
Light propagation in the grid

Scene illumination with the grid

Rendering Pipeline – Injection



Reflective shadow maps



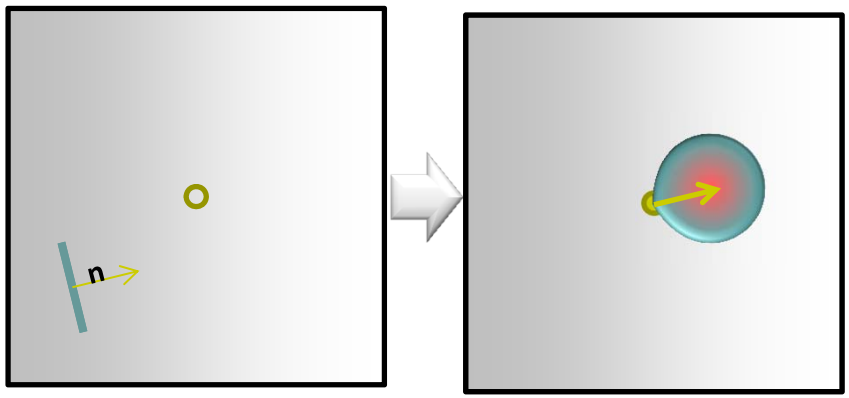
A set of regularly sampled VPLs of the scene from light position

Radiance volume gathering

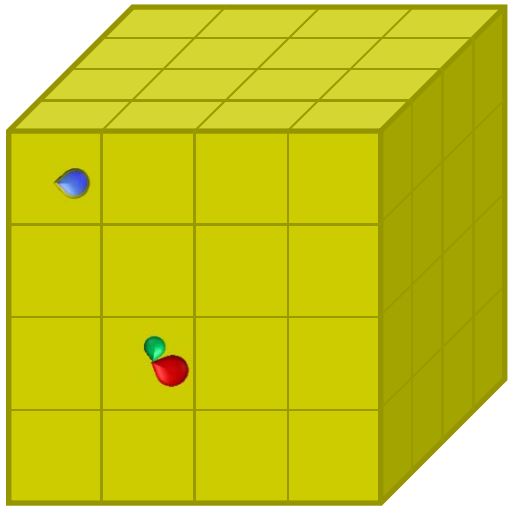
Discretize initial VPL distribution by the regular grid and SH

?

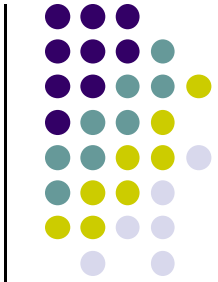
Light Injection into the Volume



- Every element of Reflective Shadow Map is a **secondary lights**
- Render as a **point primitive** into 3D grid
  - Represent flux in **Spherical Harmonics**
- Accumulate all VPLs into the grid
- The 3D grid is **initialized** with initial reflected light in the end

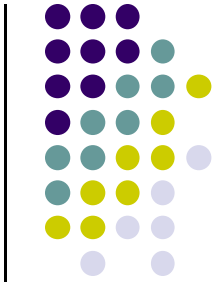


Light Propagation

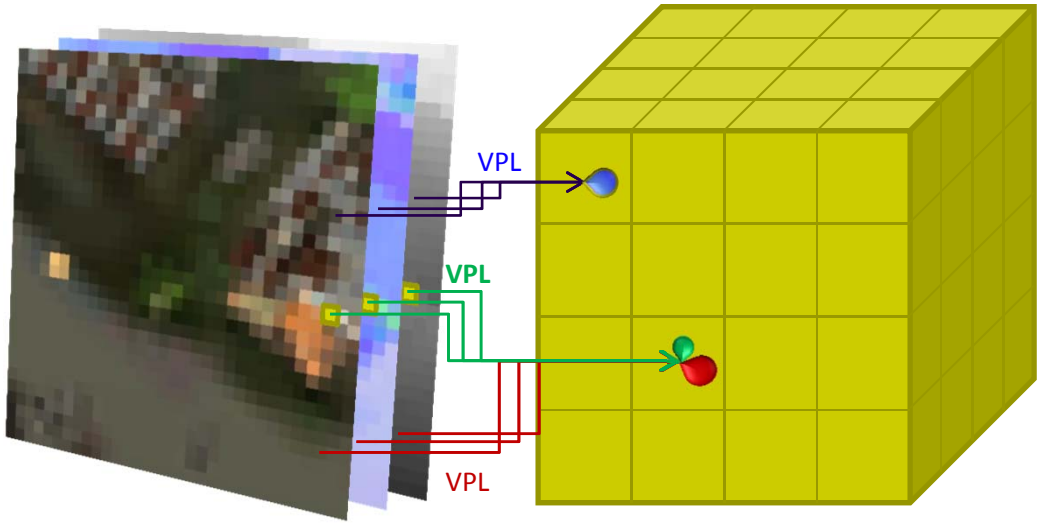


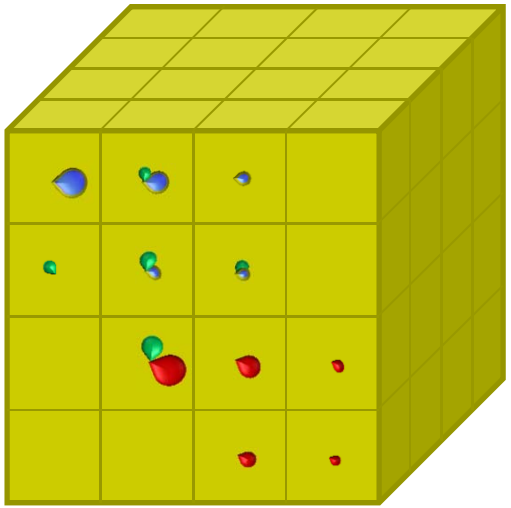
- Sample lit surface elements
- Grid initialization
- **Light propagation in the grid**
- Scene illumination with the grid

Rendering Pipeline – Propagate



Reflective shadow maps **Radiance volume gathering** **Iterative propagation**





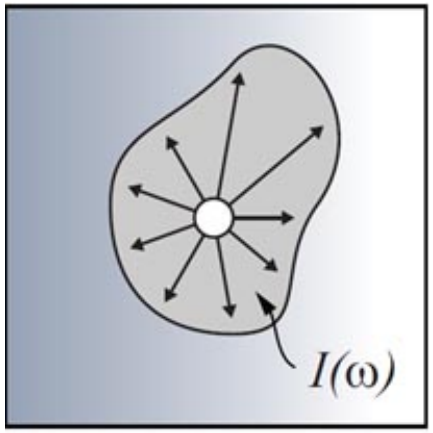
A set of regularly sampled VPLs of the scene from light position

Discretize initial VPL distribution by the regular grid and SH

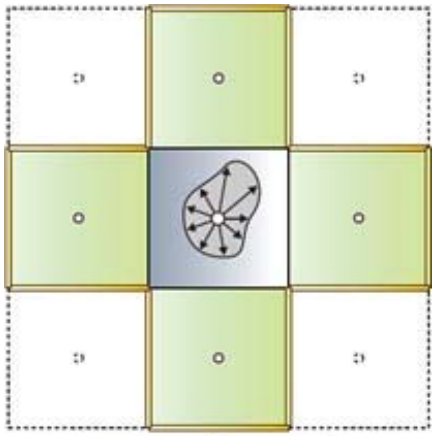
Propagate light iteratively going from one cell to another

Iterative Light Propagation

- **Local cell-to-cell** propagation across the 3D grid
  - Iterate till the light travels through the **entire volume**
  - Similar to **SH Discrete Ordinate Method** (used for participating media illumination)
  - **Number of iterations** depend on the resolution of the grid



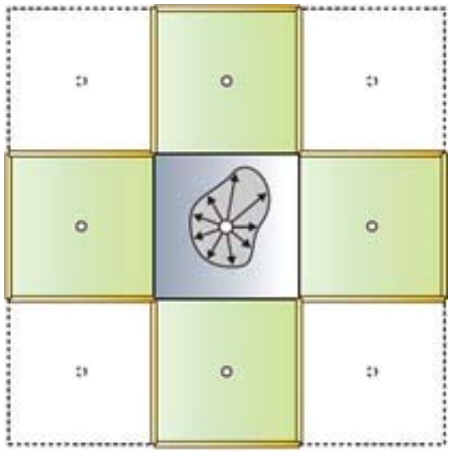
source cell



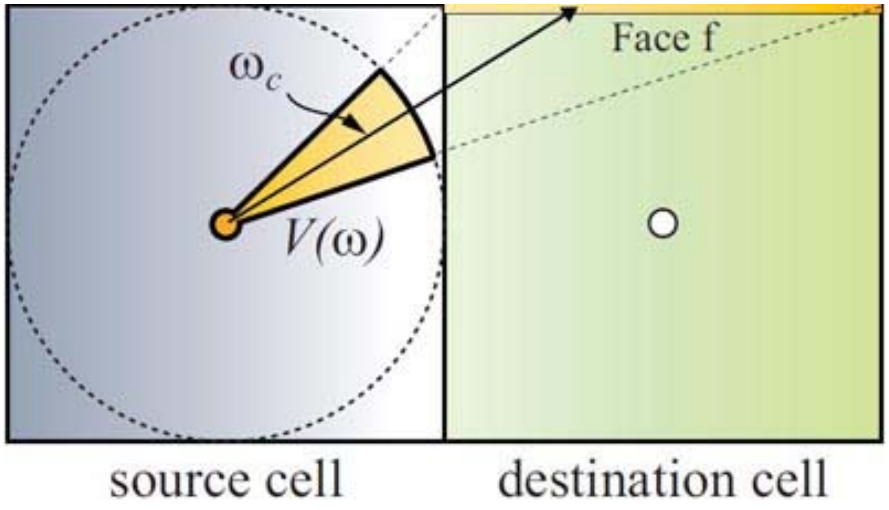
propagation along axial directions

The Propagation Iteration

- **6 axial directions** of propagation
- Use **contour faces** as a propagation wave front
- Integrate source intensity by the solid angle to get incoming **flux** for the face **f**

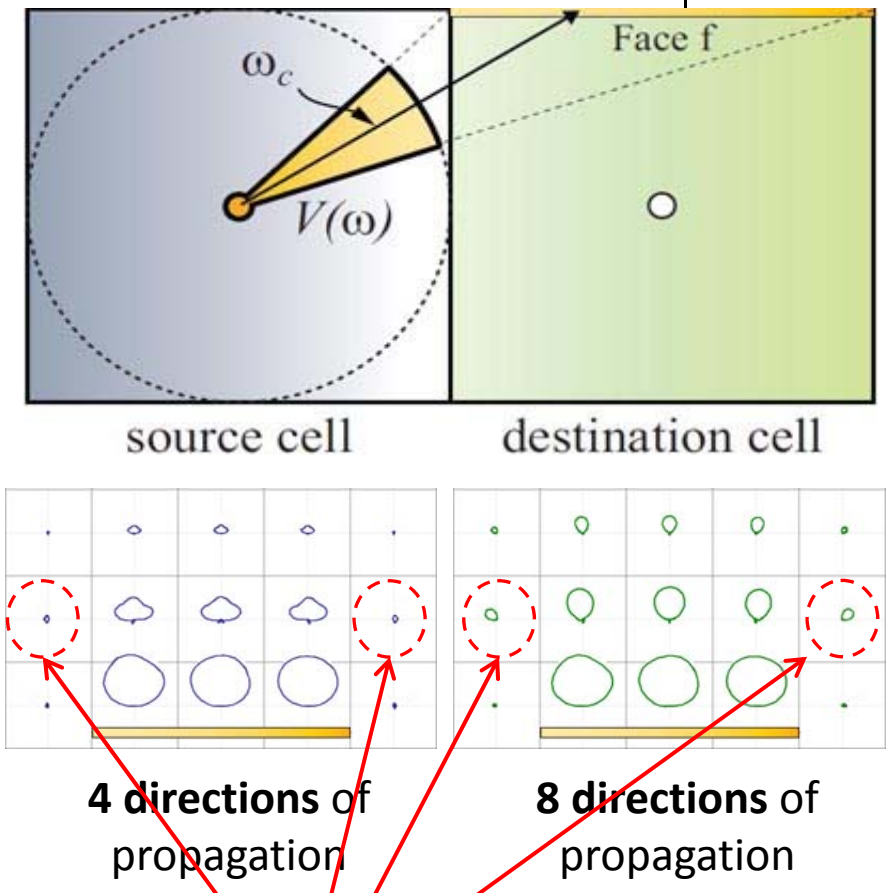


propagation along axial directions



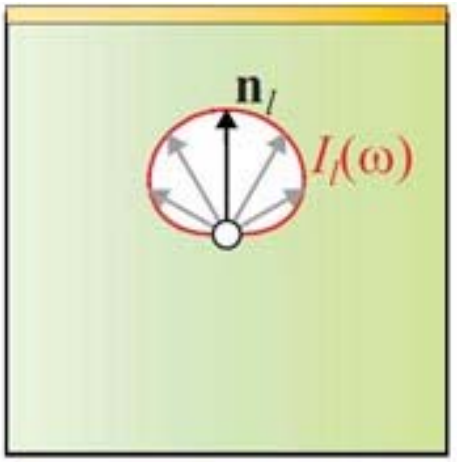
The Propagation Iteration

- Use more than **6 directions**
  - Only **6 direct neighbors**
  - Compute light propagation **to each face** of neighbors' cells
  - **30 virtual directions**
  - SHDOM: **27** neighbor cells **= 27** directions
  - good trade-off of **memory bandwidth**



- **"Ray effect" -** light propagates in a set of fictitious directions

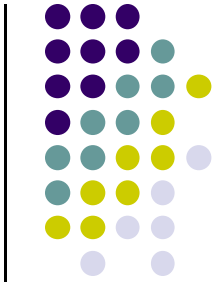
Reprojection



reprojection of the flux into a point light

- Acquire the incident **flux** through the receiving face
- Create a new point light in the center of receiving **cell**
  - Oriented towards the face
  - Causing exactly the same flux as the face received
- Generate **clamped cosine lobe** in SH basis similar to injection stage
- Accumulate the resulting SH coefficients into the destination cell for next iteration

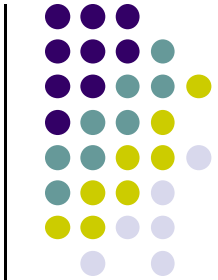
Rendering



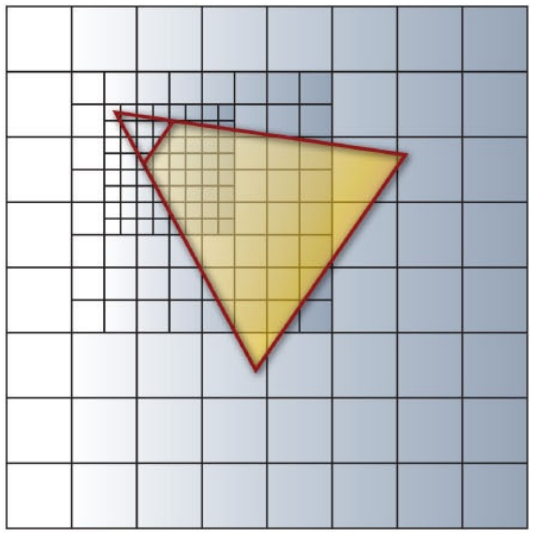
- Look-up grid with trilinear interpolation
- Evaluate the irradiance with cosine lobe of surface's normal
- Apply dampening factor
  - Compute directional derivative towards normal
  - Dampen based on derivative deviation from the intensity distribution direction

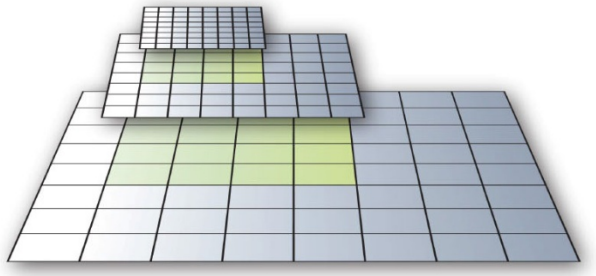
Results of indirect illumination

Cascaded Light Propagation Volumes

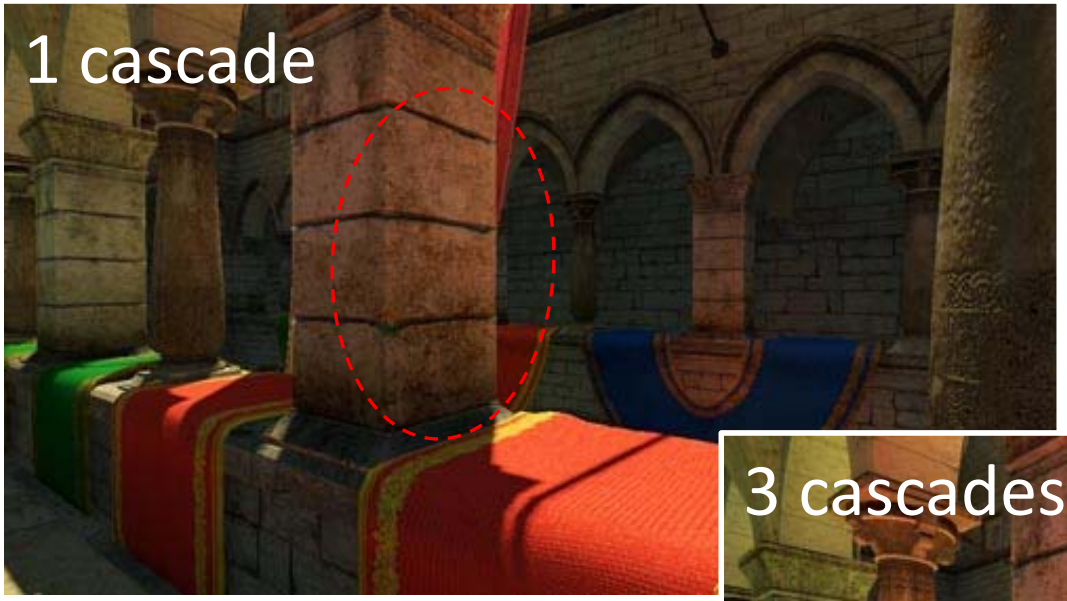


- Motivation: memory and bandwidth cost is **o(N^3)** for increase of LPV **grid**
  - Impossible to support **large scenes**
- Idea: use **multiple nested grids** to refine resolution hierarchically
  - Do **not** consider **small objects** for large sparse grids
- **Transfer** propagated lighting from nested grid to the **parent grid**
- Illuminate scene similarly to *cascaded shadow maps*
- Reduces the **number of iterations** sufficient per cascade



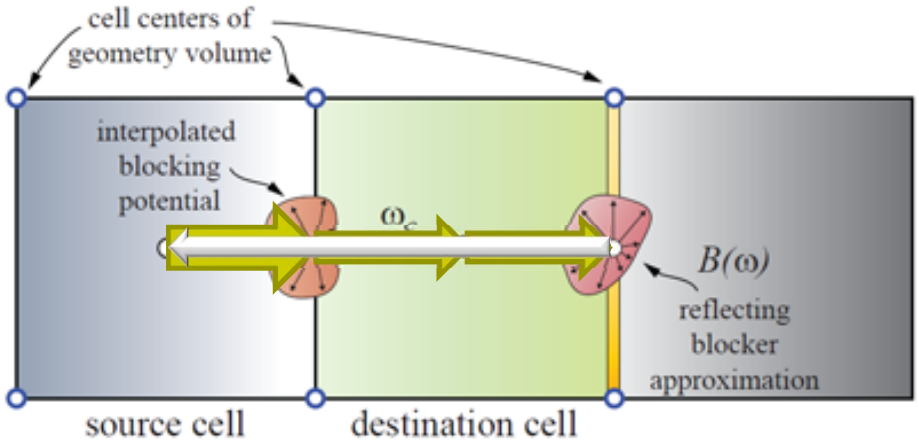


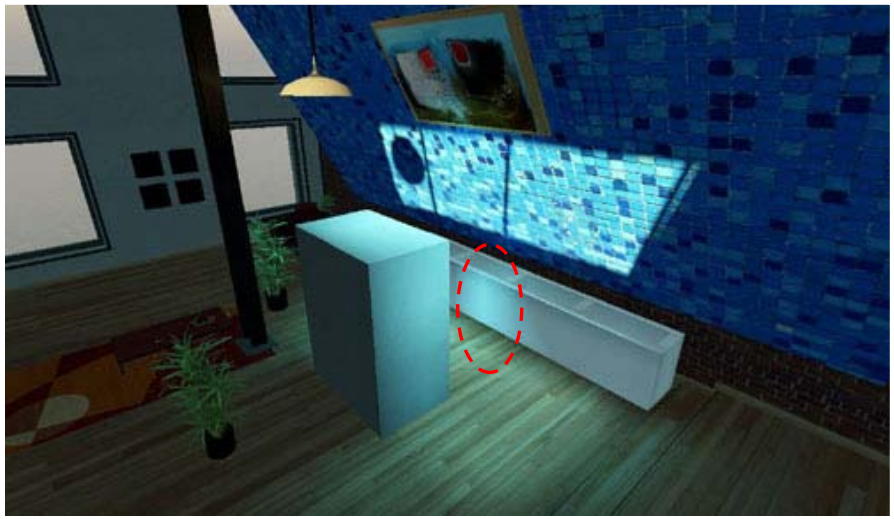
Cascaded Indirect Illumination



Multiple Bounces

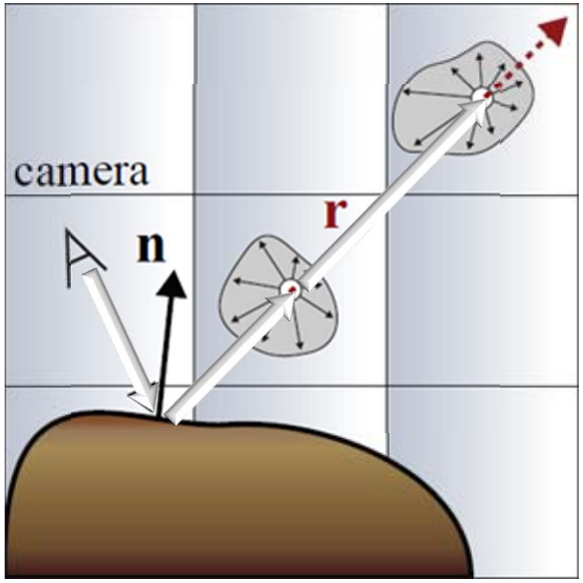
- **Idea**: use information from occlusion grid to compute multiple indirect reflections
- Reflect light during **each** propagation iteration
- Avoid self-illumination by injecting reflected light at **safety-distance**



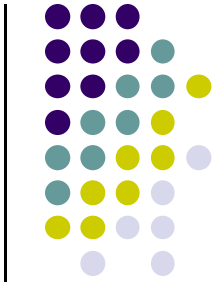


Glossy Reflections

- Idea: Compute incident light from reflection direction by **marching** through LPV grid
- Go **few steps back** in propagation time to reduce **light smearing**
- **4 cells** is sufficient for **moderately glossy** objects
- Lookups into **multiple cells** prevent discontinuities in glossy reflections

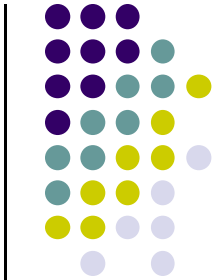


Conclusion



- Full-dynamic: scene, view, lighting changes
- Real-time: GPU- and consoles- friendly
- Production-eligible (simple tweaking)
- Highly scalable
  - proportionally to quality
- Stable, flicker-free
- Supports complex geometry (e.g. foliage)

References



- Cascaded Light Propagation Volumes for Real-Time Indirect Illumination. Anton Kaplanyan, Carsten Dachsbacher. ACM 2010
- Light Propagation Volumes in CryEngine3. Anton Kaplanyan. Siggraph 2009