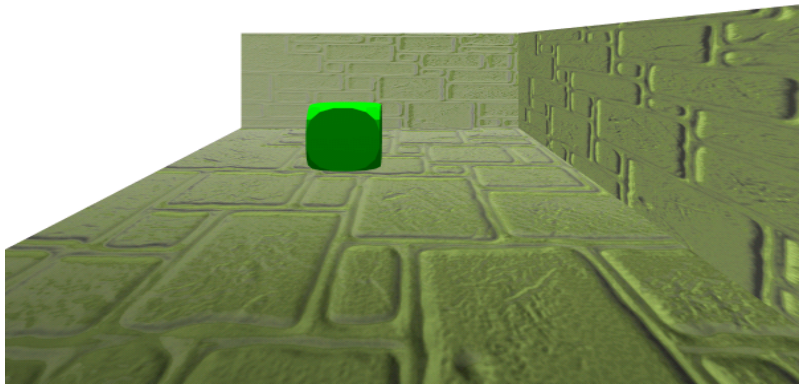
Generation of Shadows in Scene Graphs based VR

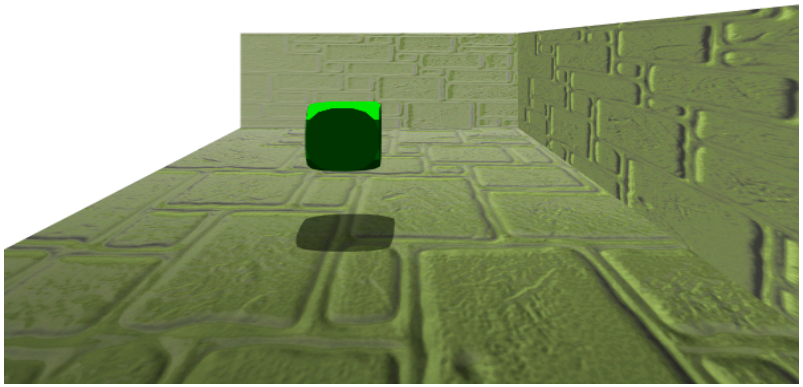
Björn Kühl, Kristopher J. Blom, Steffi Beckhaus

interactive media / virtual environments

University of Hamburg, Germany

The importance of shadows





Shadows in games

Shadows can be found in almost every new game

Skilled/Specialized modelers and designers

Two common methods for shadow generation are used:

- Shadow Mapping
  - Need for Speed
  - Prince of Persia
- Stencil Shadow Volumes
  - Doom3
  - F.E.A.R
  - Prey

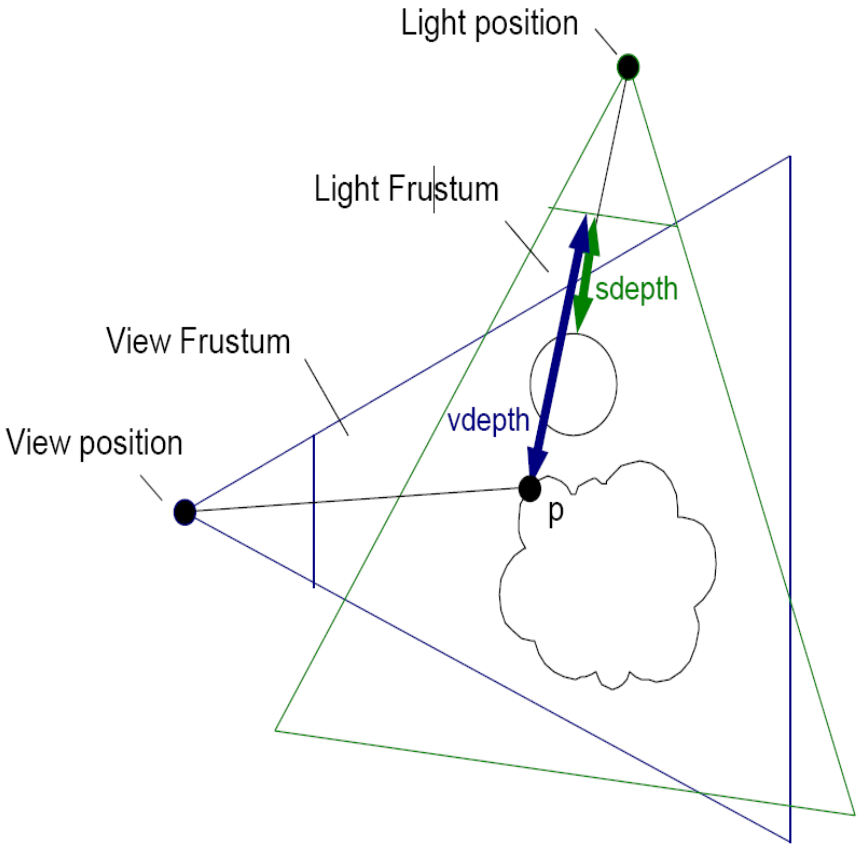
Shadow Mapping

1 pass: light view

- store depth values in shadow map

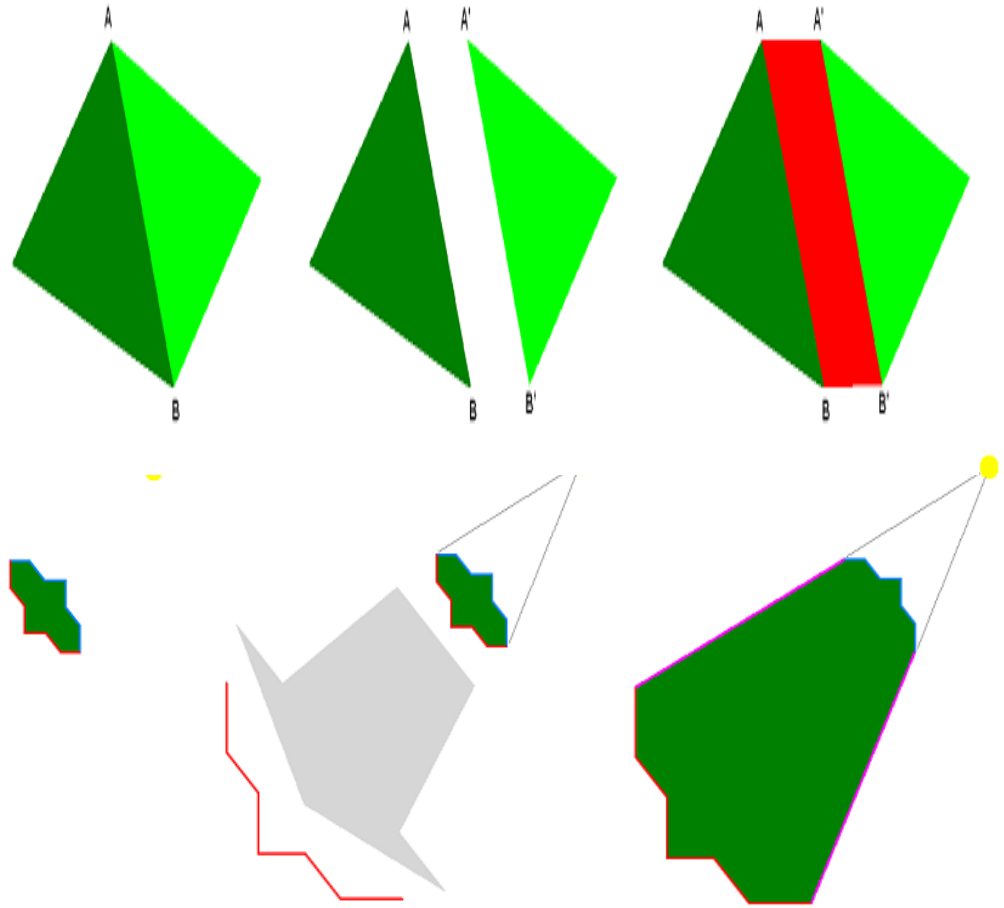
2 pass: camera view

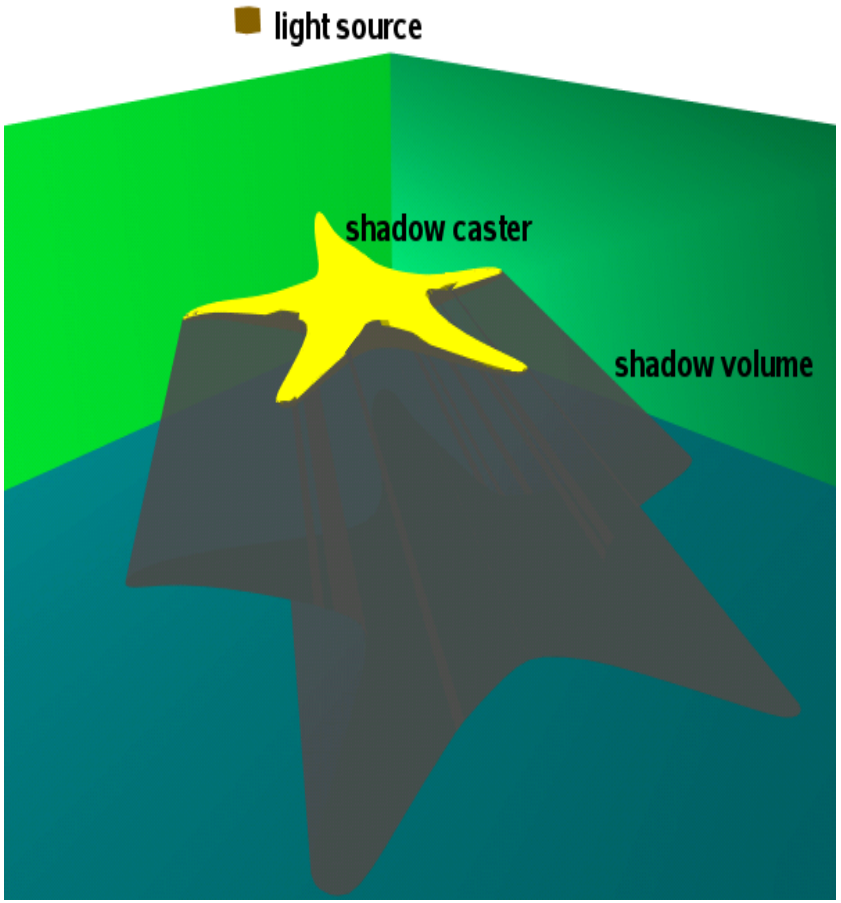
- transform camera's eye coords in light's clip coords
- compare transformed depth value with indexed shadow map position



Source: M. Lipp: *A Survey of Real-Time Shadow Mapping Techniques*

Stencil Shadow Volumes





Source: *MSDN: Shadow Volume Sample*

(http://msdn.microsoft.com/library/default.asp?url=/library/en-us/directx9-c/ShadowVolume-Sample.asp)

Shadows in VR?

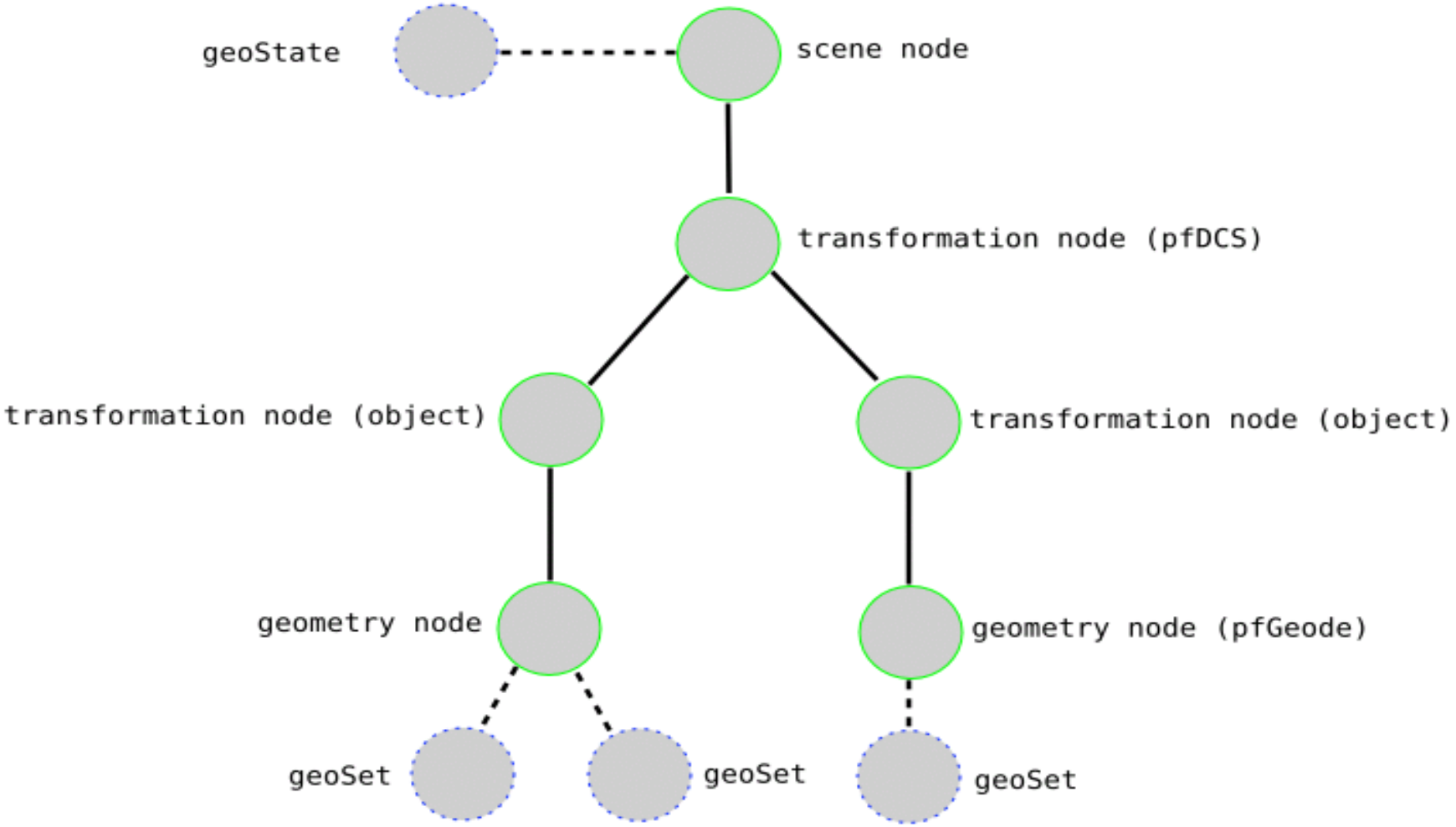
VR is highly focused on immersion and the feeling of presence.

High quality images can enhance this feeling of presence.

...but shadows aren't supported in most VR systems

- Due to the underlying rendering system
- Multi-pipe (graphic cards)
- Distributed systems
- Head-tracked

Scene Graphs Based VR Systems: Performer example

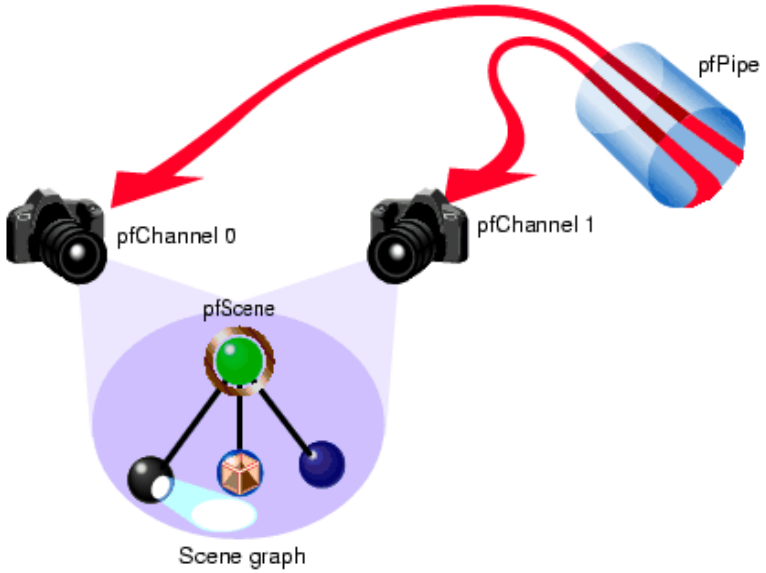


Shadow Mapping: Implementation for Performer

The scene is viewed through a channel

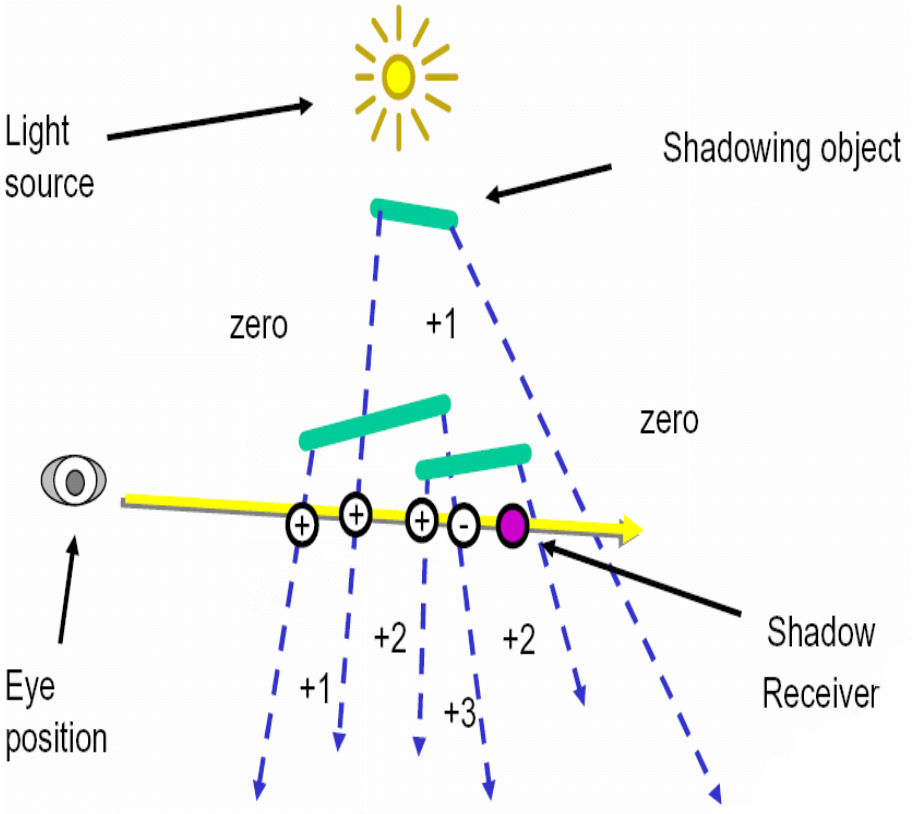
1st channel:
light's view

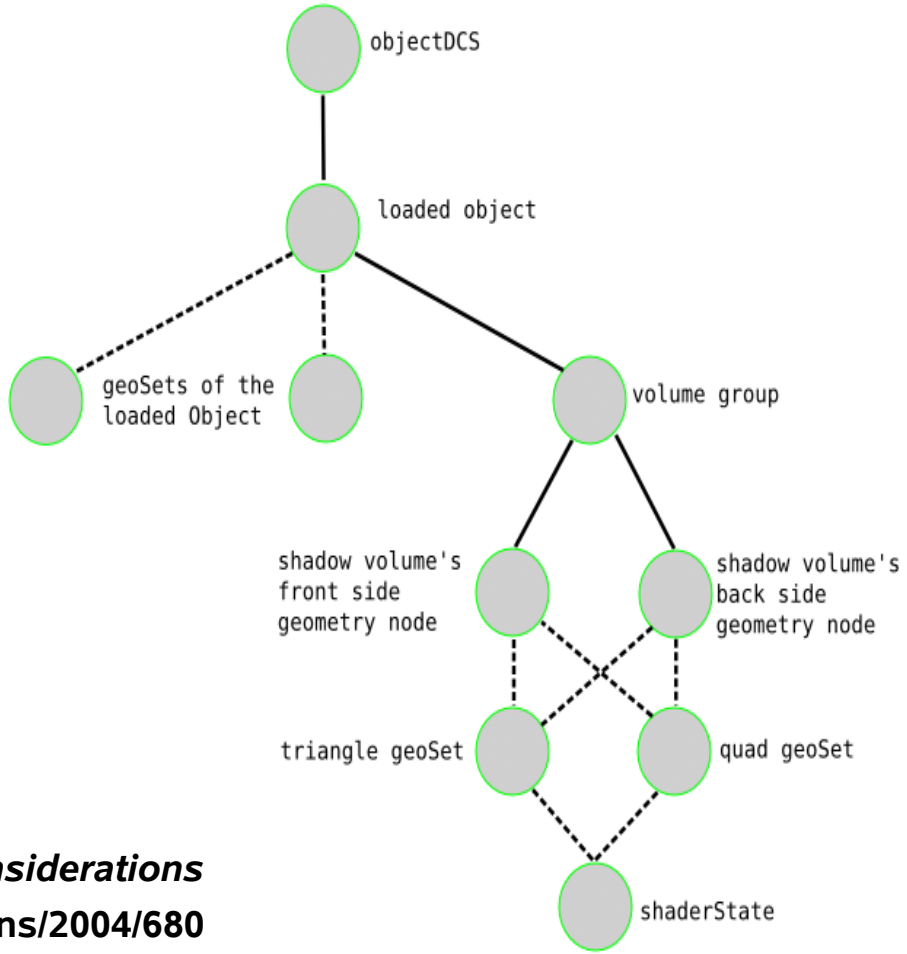
2nd channel:
camera's view



Source: *OpenGL Performer Programmer's Guide*
(http://www.sgi.com/products/software/performer/manuals.html)

Stencil Shadow Volumes: Implementation for Performer

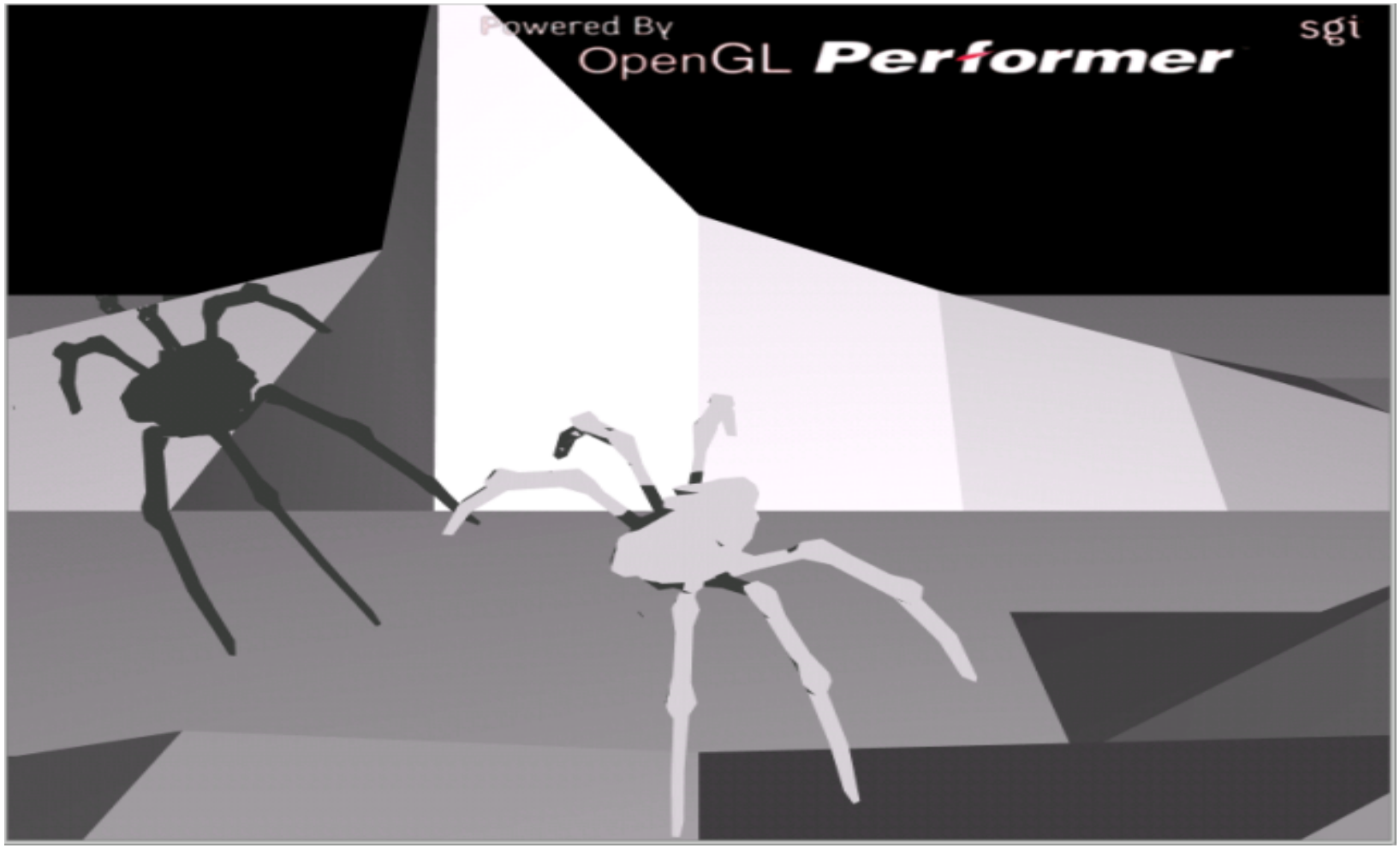




Source: ***6800 Leagues Under The Sea: Shadow Considerations***

(http://download.nvidia.com/developer/presentations/2004/6800_Leagues/\\6800_Leagues_Shadows.pdf

Stencil Shadow Volumes: Implementation for Performer



AVANGO

Object-oriented, distributed VR-Framework

“Performer with fields”

- Built as a layer on top of Performer
- Orthogonal dataflow Field Container concept

Abstraction from underlying devices

Classes are coded in C++

Scheme scripting and run-time command execution

Shadow Mapping: Implementation for AVANGO

Unfortunately, the approach for Performer can't be used for AVANGO (and also for other VR systems)

Reasons:

- Channels are hidden from the user
- Channels have to share data

Solution:

- A light source exists in Performer capable of producing shadows
- A encapsulated version exists in AVANGO!

Stencil Shadow Volumes: Implementation for AVANGO

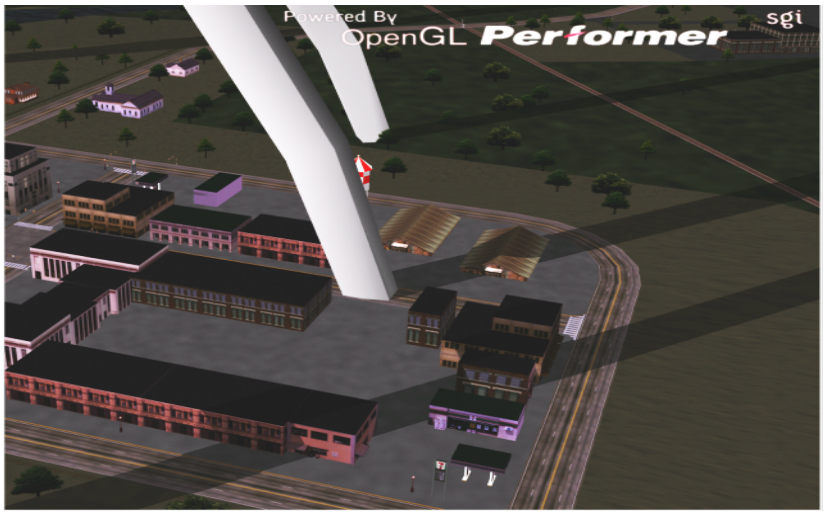
New node: fpStencilShadow

Inherited from fpDCS (transformation node)

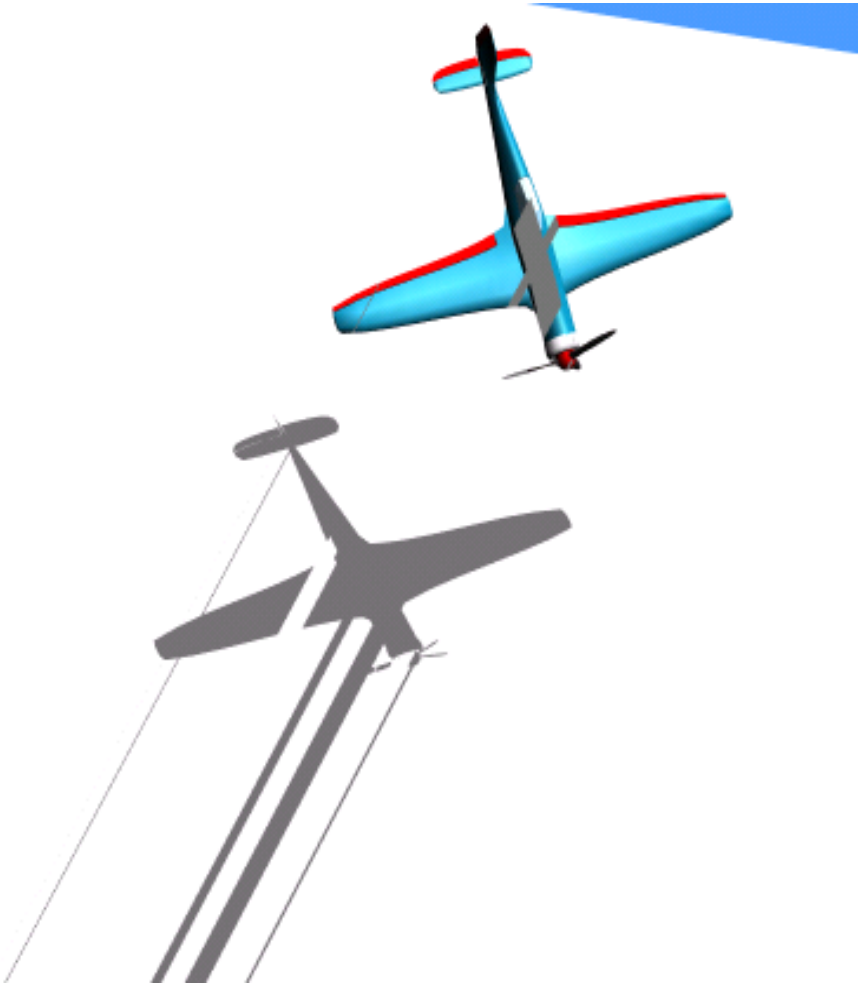
Requires an object (tree) and a light number

- the node calculates a shadow volume(s)
- the node configures the rendering to the stencil buffer
- the node configures the semi-transparent film

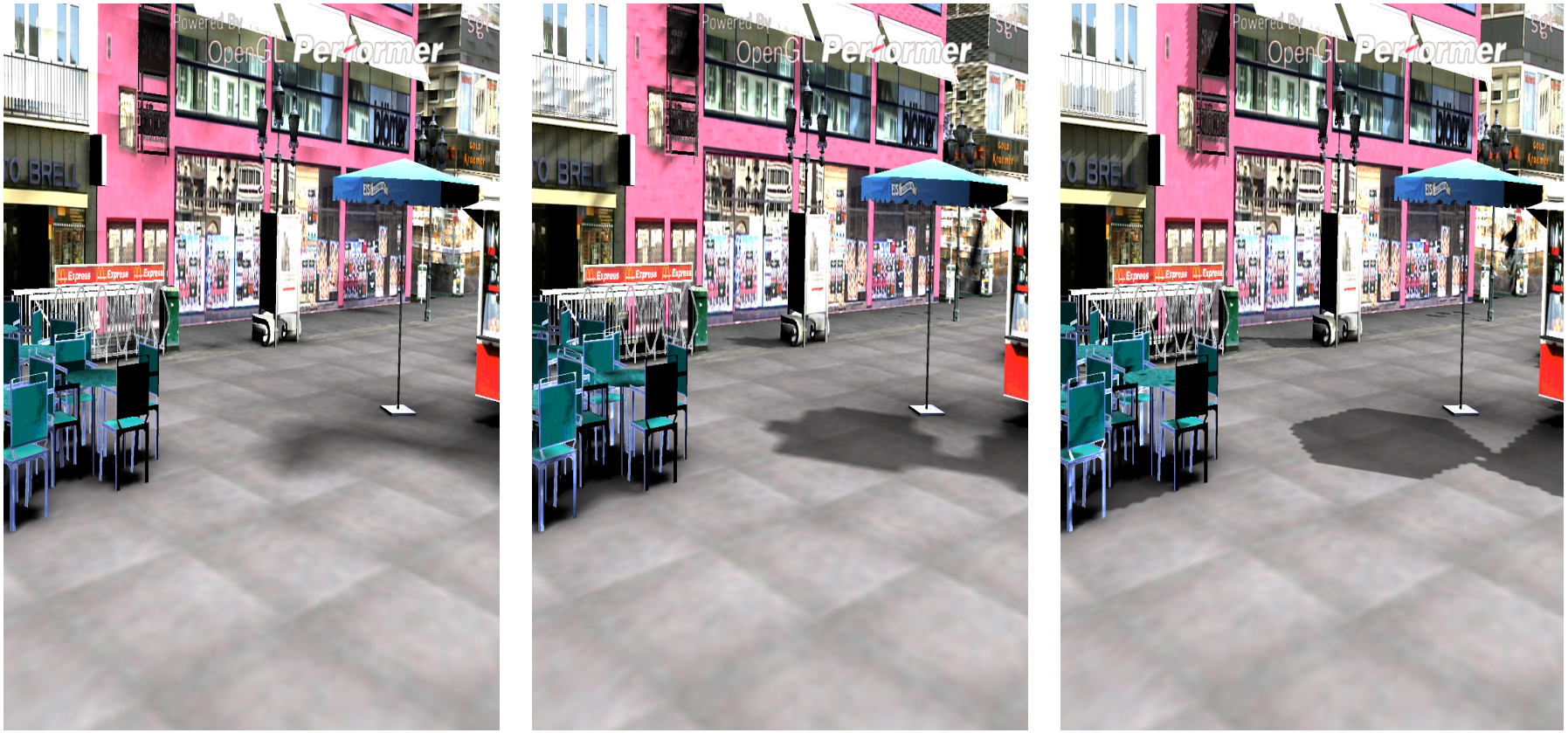
Stencil Shadow Volumes: AVANGO



Stencil Shadow Volumes: Dependancy of faultless models



Shadow Mapping: Choosing an adequate texture resolution

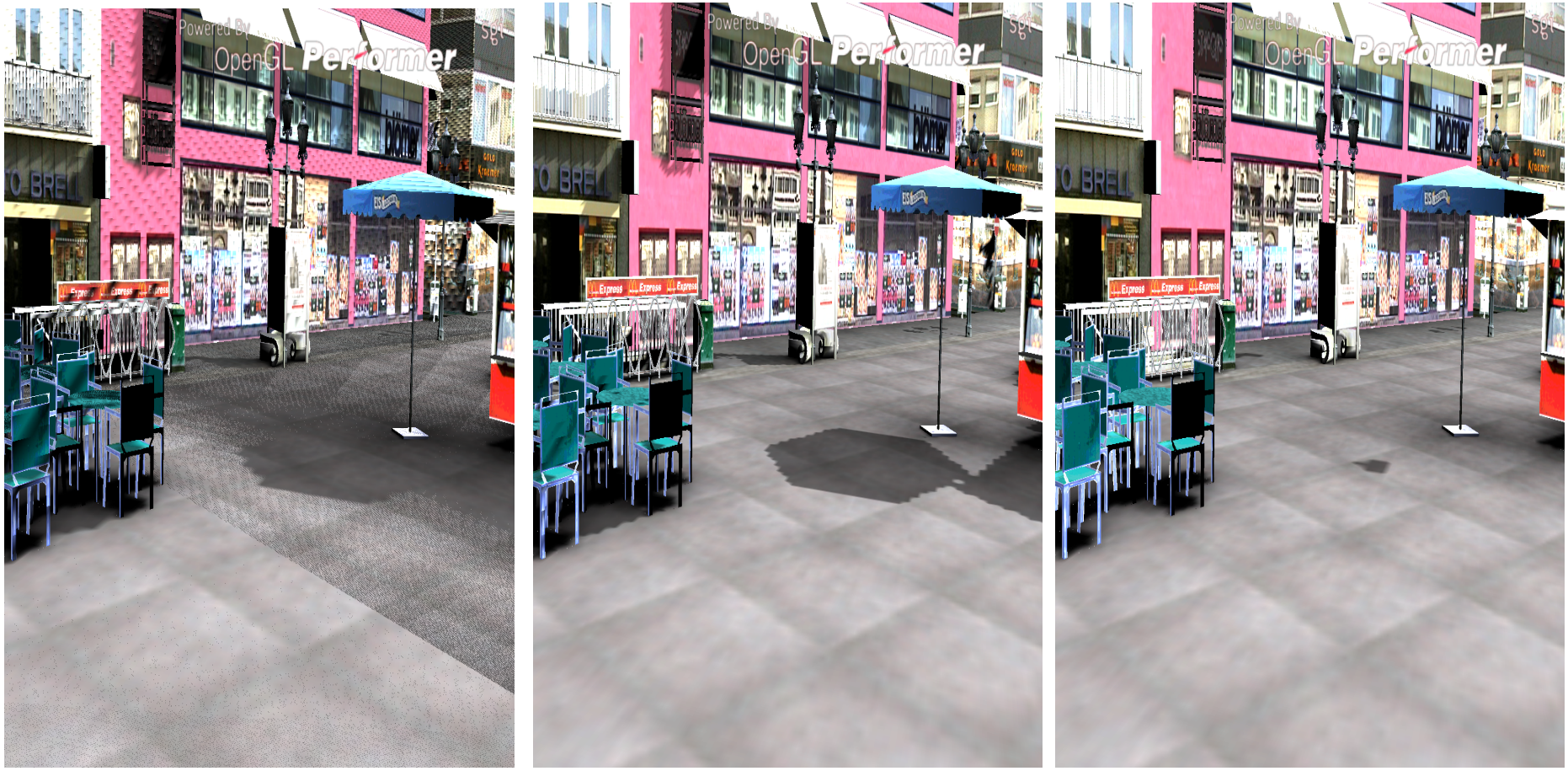


Using different sizes of depth textures (from left to right: 64x64, 256x256, 1024x1024)

Shadow Mapping: Choosing an adequate field of view

Shadows for the whole scene (left), shadow generation limited to a smaller region (right)

Shadow Mapping: Incorrect self-shadowing



Polygon offset: too low (left), just right (middle), too high (right)

Shadow Mapping: Choosing an adequate polygon offset

incorrect self-shadowing

missing shadows

Combined usage of Shadow Volumes & Shadow Mapping

Use tweaked shadow mapping for static geometry
for movable objects shadow volumes should better be used.

Questions?

Mailto:
9kuehl@informatik.uni-hamburg.de
blom@informatik.uni-hamburg.de
steffi.beckhaus@uni-hamburg.de