

# Per-Pixel Displacement Mapping with Distance Functions

William Donnelly  
University of Waterloo  
(Presented by Matthias Wloka)



## Last Year's Big Thing

- Parallax mapping
  - T. Kaneko et al. “Detailed Shape Representation with Parallax Mapping.” In *Proceedings of the ICAT 2001 (The 11th International Conference on Artificial Reality and Telexistence)*, Tokyo, Dec. 2001.
- Valid for smoothly varying height fields
  - No occlusion
  - No large displacements
  - No high frequency features



## Demo



## Advantages

- Handles high frequencies
  - For example, slats or text
- Correctly resolves self-occlusion



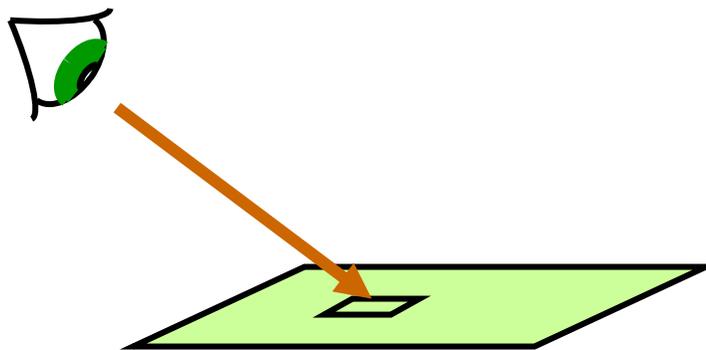
## Potential Applications

- Metal-grate walkways
- Chain-link fences
- Everywhere?
  - In addition to bump mapping
- Geometry can be painted on
  - Need not be modeled



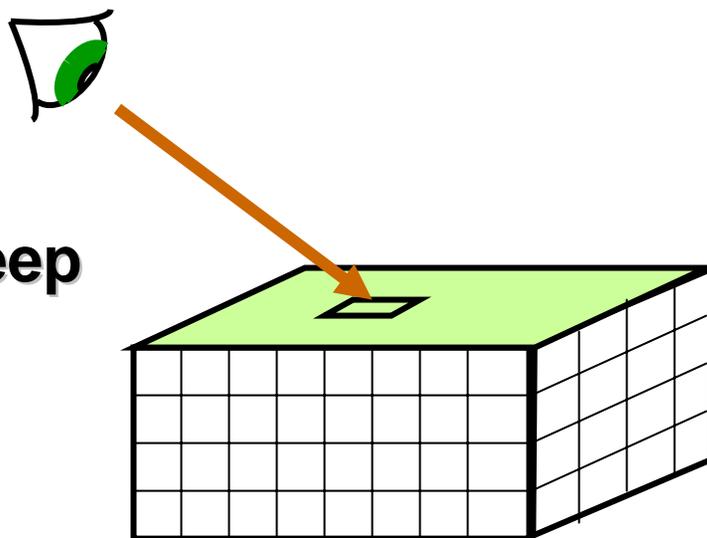
## How Does It Work?

- Transform eye ray into tangent space
- Ray trace!?



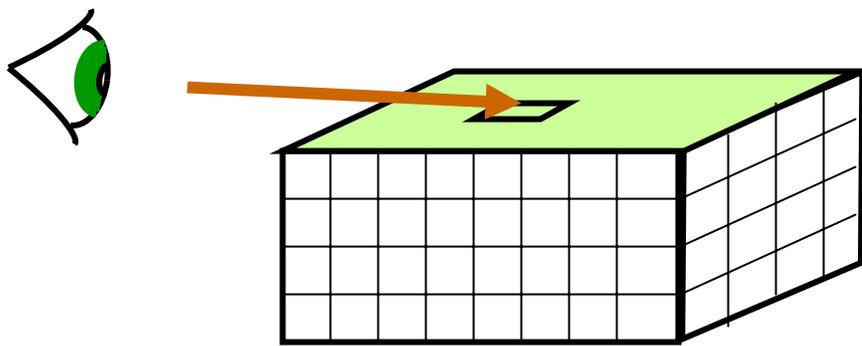
## Step Along Eye Ray

- Volume texture encodes
  - ‘Filled’ or
  - ‘Empty’ material
- ‘Thin’ texture
  - Up to 16 voxels deep
  - E.g., 256x256x16



## Step Size Problems

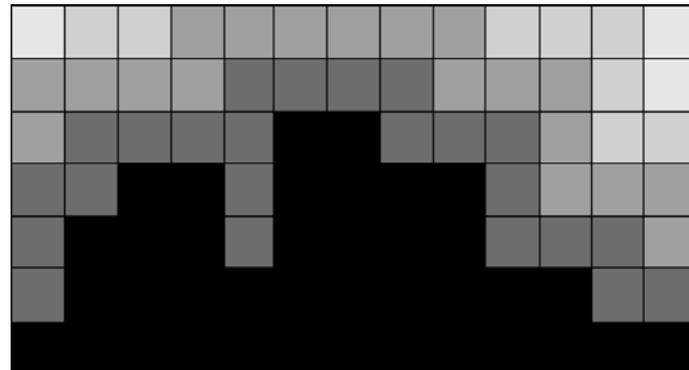
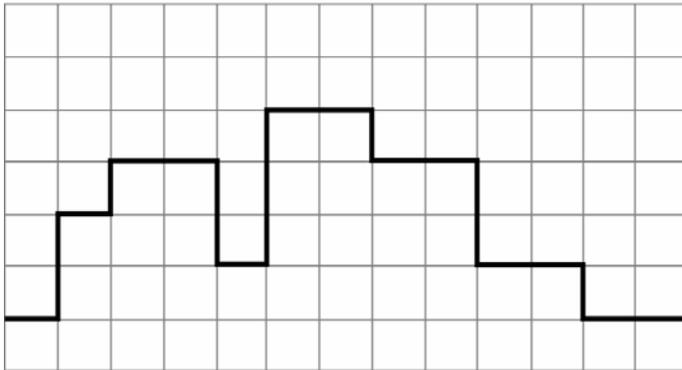
- Fixed size?
  - Too big: Misses high frequency detail
  - Too small: Wasted performance
- Vary with incident angle?
  - Still intractable at grazing angles



# Volume Texture Stores Distance Map

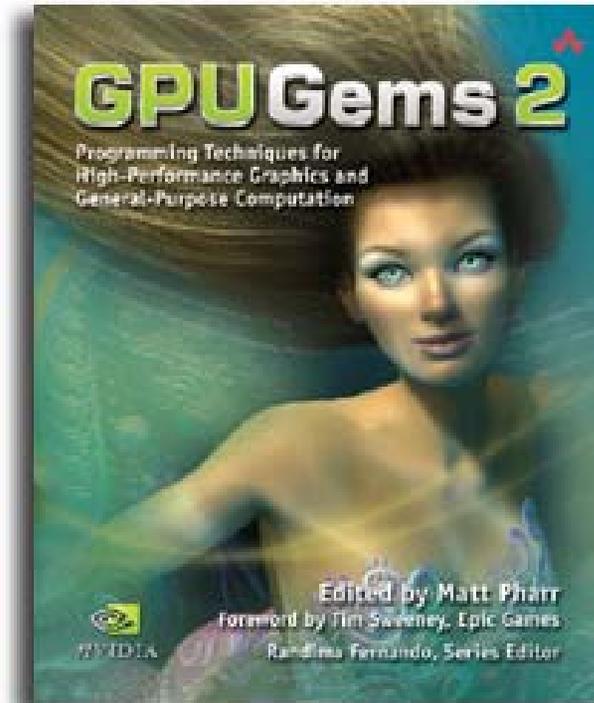
- Voxel stores:
  - How far am I from closest surface?

1	3	4	4	2	5	5	4	4	2	2	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---

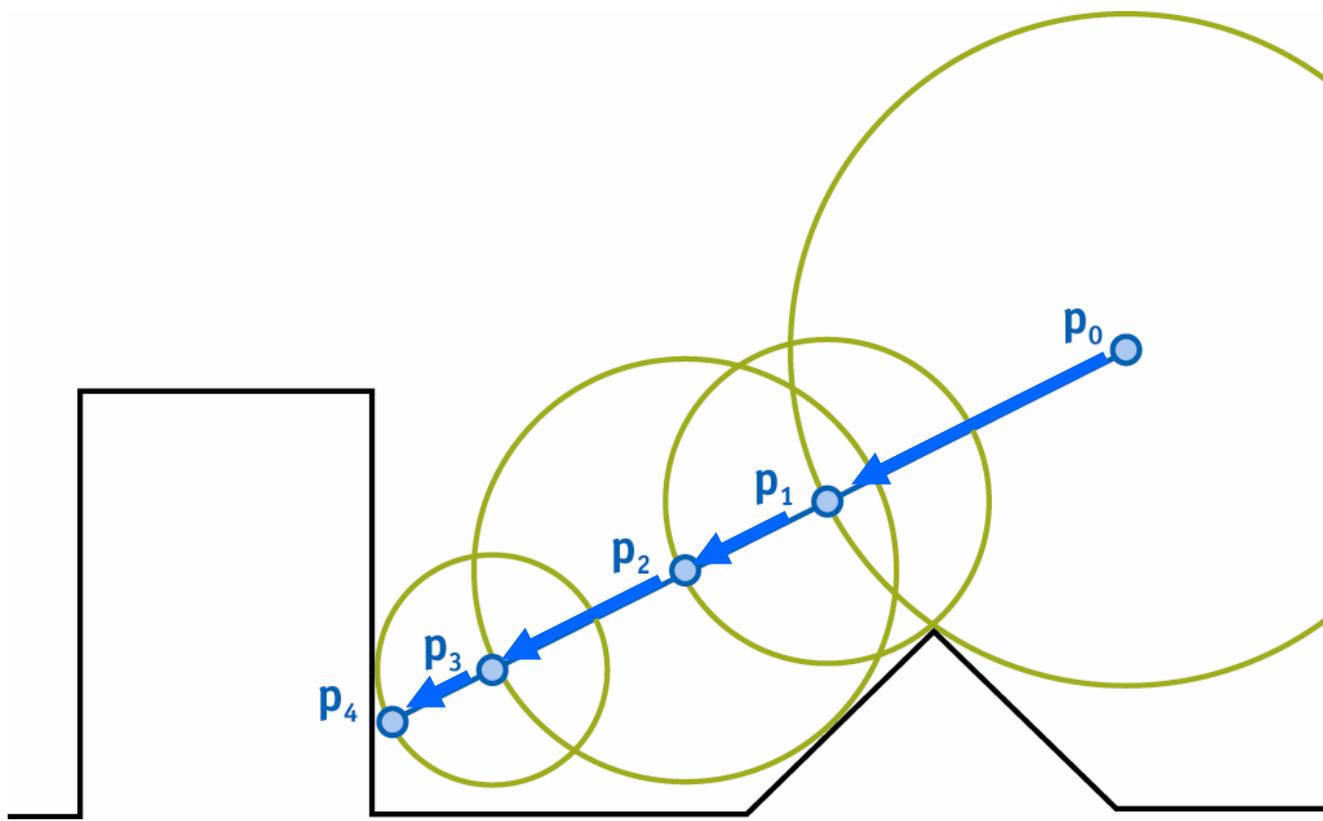


## Efficient Computation of Distance Maps

- See Section 8.4
- Tool with source on book's CD
- Based on:
  - Danielsson, Per-Erik. 1980. "Euclidean Distance Mapping." *Computer Graphics and Image Processing* 14, pp. 227–248.



# Equivalent to Sphere Tracing



## Distance Maps Are General

- NOT limited to height fields
- Encode arbitrary geometry in volume
  - Chain mail
  - Overlapping slats
  - Chain-link fence
  - ...



# Pixel Shader

```
float distance;  
float3 offset = normalize(v2f.tanEyeVec);  
float3 texCoord = v2f.texCoord;  
  
// March the ray  
for (int i = 0; i < NUM_ITERATIONS; i++)  
{  
    distance = tex3D(distanceTex, texCoord).x;  
    texCoord += distance * offset;  
}
```



## Performance

- Each iteration is {tex; mad;}
  - Single cycle on GeForce FX and GeForce 6
- Number of iterations depends on
  - Volume texture resolution
  - Smoothness of data
  - 16 iterations plenty for shown 256x256x16
- 90M pixels/s observed throughput
  - GeForce 6800 GT
  - Inclusive lighting



## Questions?

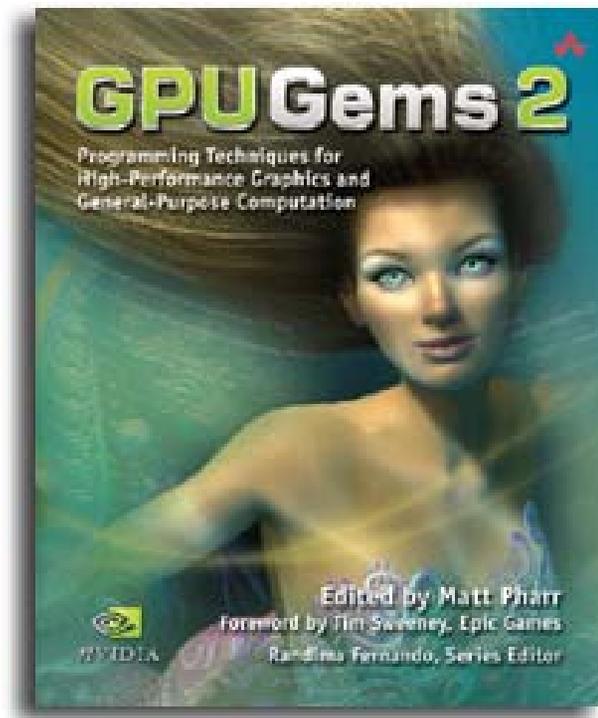
- See GPU Gems 2, Chapter 8
- <http://developer.nvidia.com>  
The Source for GPU Programming
- [mwloka@nvidia.com](mailto:mwloka@nvidia.com)
- Slides available online



# GPU Gems 2

## Programming Techniques for High-Performance Graphics and General-Purpose Computation

- 880 full-color pages, 330 figures, hard cover
- \$59.99
- Experts from universities and industry



“The topics covered in *GPU Gems 2* are critical to the next generation of game engines.”

— Gary McTaggart, Software Engineer at Valve, Creators of *Half-Life* and *Counter-Strike*

“*GPU Gems 2* isn’t meant to simply adorn your bookshelf—it’s required reading for anyone trying to keep pace with the rapid evolution of programmable graphics. If you’re serious about graphics, this book will take you to the edge of what the GPU can do.”

—Rémi Arnaud, Graphics Architect at Sony Computer Entertainment

# The Source for GPU Programming

[developer.nvidia.com](http://developer.nvidia.com)

- Latest News
- Developer Events Calendar
- Technical Documentation
- Conference Presentations
- GPU Programming Guide
- Powerful Tools, SDKs and more ...



Join our FREE registered developer program for early access to NVIDIA drivers, cutting edge tools, online support forums, and more.

**nVIDIA**

[developer.nvidia.com](http://developer.nvidia.com)

©2004 NVIDIA Corporation. NVIDIA, and the NVIDIA logo are trademarks and/or registered trademarks of NVIDIA Corporation. Nalu is ©2004 NVIDIA Corporation. All rights reserved.