Game Developers Conference

Terrain Rendering using GPU-Based Geometry Clipmaps

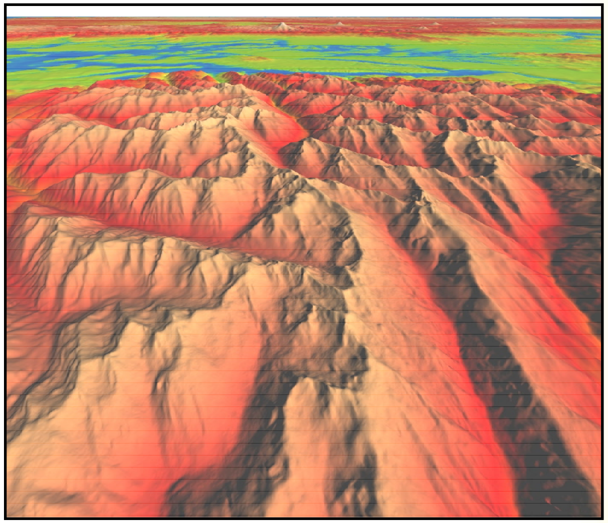
Arul Prakash
University of Utah

Hugues Hoppe
Microsoft Research

SAN FRANCISCO CA MAR 7-11 GDC 05

Terrain Rendering Challenges

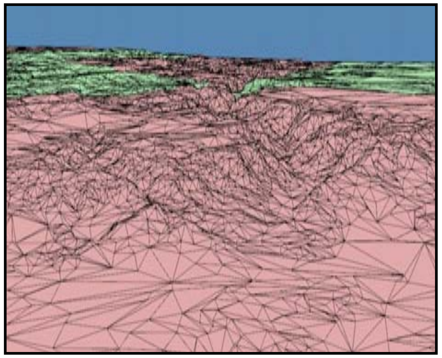
- Regular grid (image) of height values

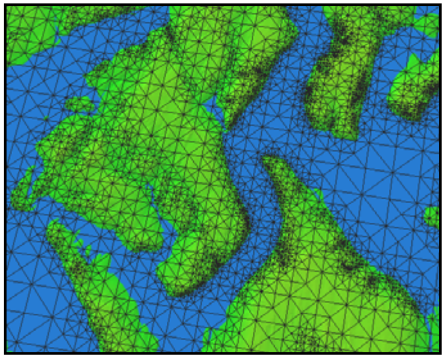


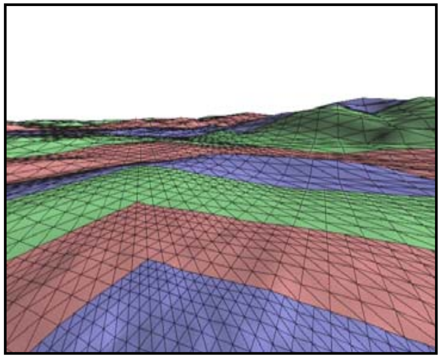
- Concise storage → No paging hiccups
- Real-Time frame rates → 60 fps
- Visual continuity → No temporal pops

A Change of Focus

- Hoppe 1998 – Highly irregular Connectivity
- Lindstrom 1996 – Semi-regular Connectivity
- Losasso & Hoppe 2004 – Totally regular Connectivity

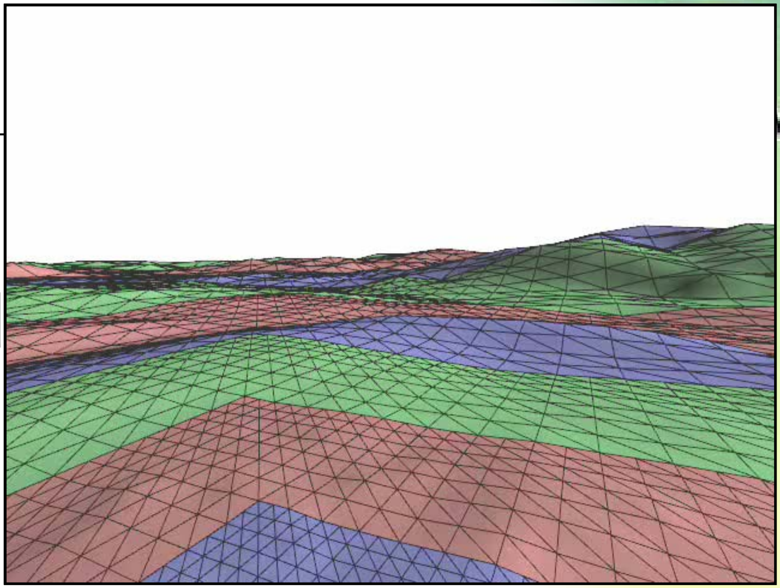


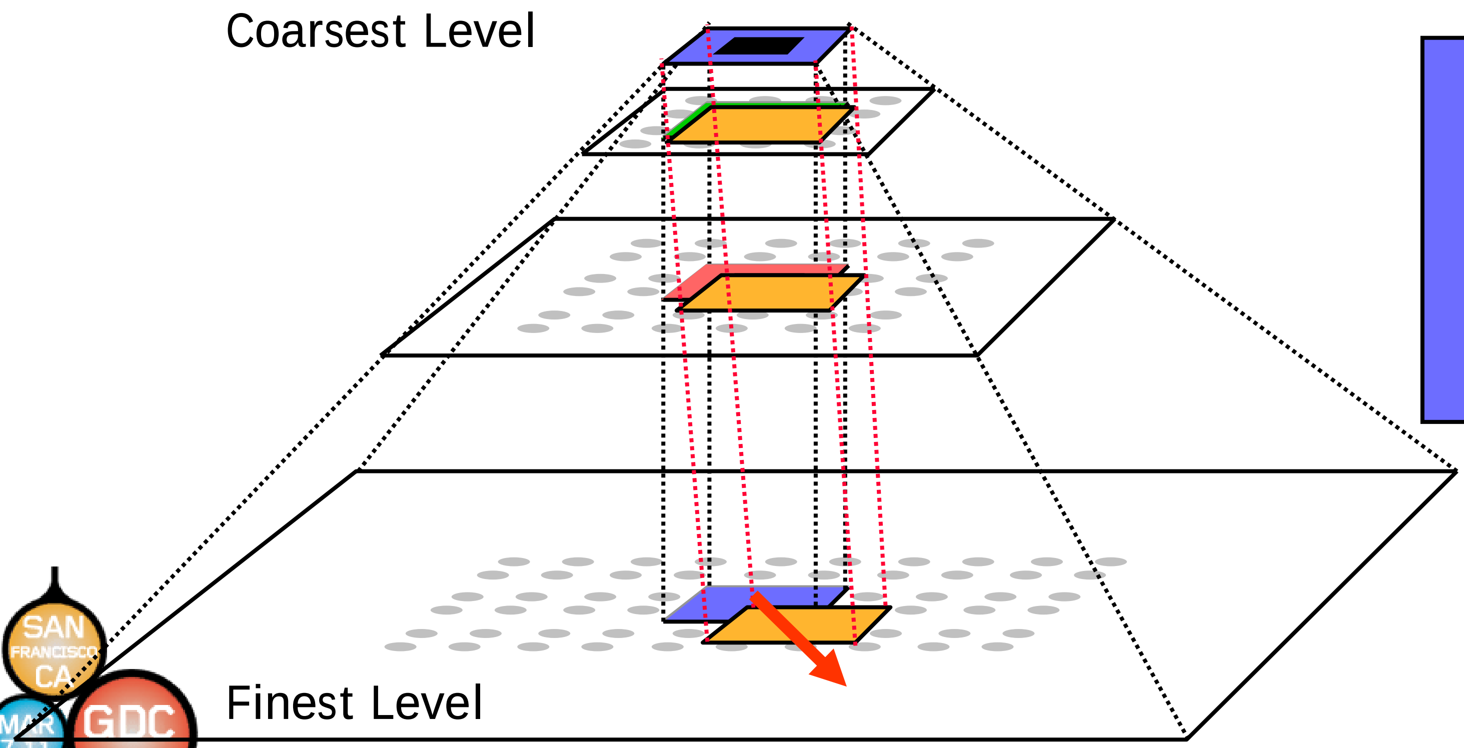


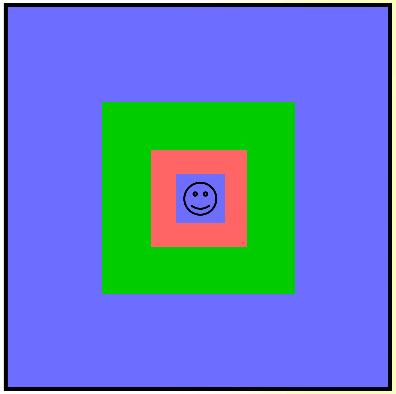


Geometry Clipmaps

- Terrain as mipmap pyramid
- LOD using nested grids





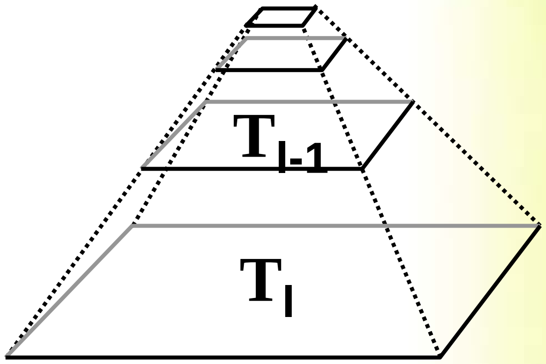


Terrain Compression

- **Store coarsest level + inter-level residuals**

$$R_l = T_l - U(T_{l-1})$$

$$\tilde{R}_l = compress(R_l)$$



- **Reconstruction**

$$T_l = U(T_{l-1}) + decompress(\tilde{R})$$

Why GPU?

- Much less CPU utilization
- Very little AGP/PCIe bus utilization
- Small system memory requirement
- Small video memory requirement
- Significant rendering speedup

GPU Implementation Overview

- DirectX 9.0c - support for Shader Model 3.0
- HLSL code available on book's CD

- Store data as textures (in video memory)
  - Elevation data – 32-bit 1-channel texture
  - Residual data – 32-bit 1-channel texture
  - Normal data – 8-bit 4-channel texture
- Update parts of texture that change

System Desig

> **Vertex Textures**
>
> **Supported in DirectX 9.0c & Shader Model 3.0 on NVIDIA GeForce 6800**

Update

- **Upsample coarser level elevation texture**
- **Add residuals**
- **Update normal map texture**

ering

Vertex Shader

-Use (x, y) to lookup z value in elevation texture

-Compute α

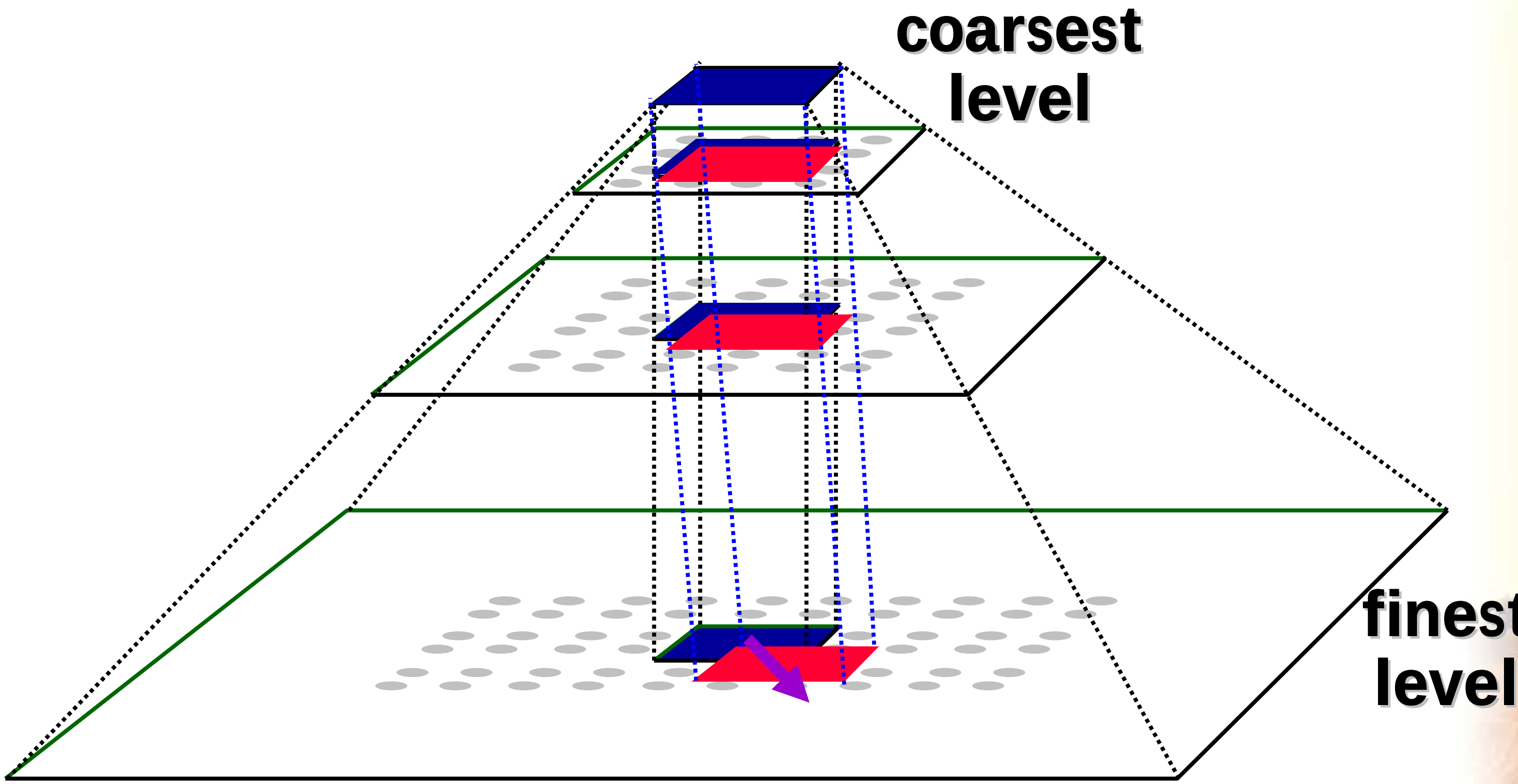
- Blend Geometry

Pixel Shader

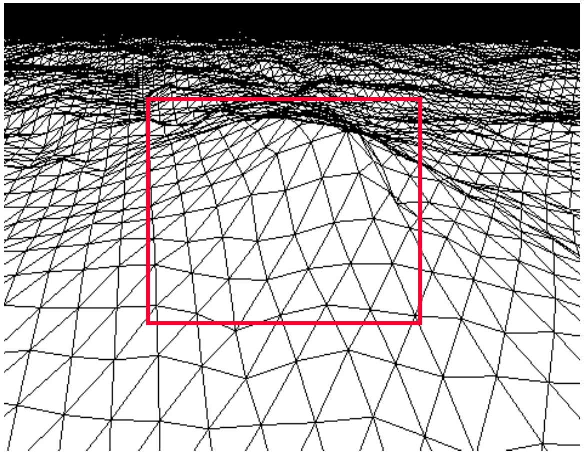
- Blend normals
- Does the shading
- Texture lookup

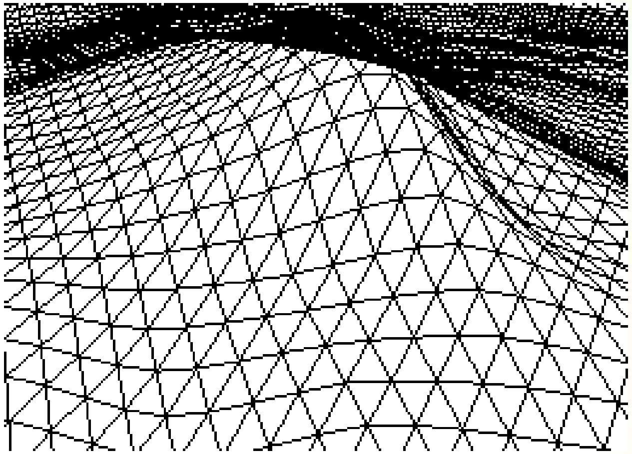
Clipmap Update

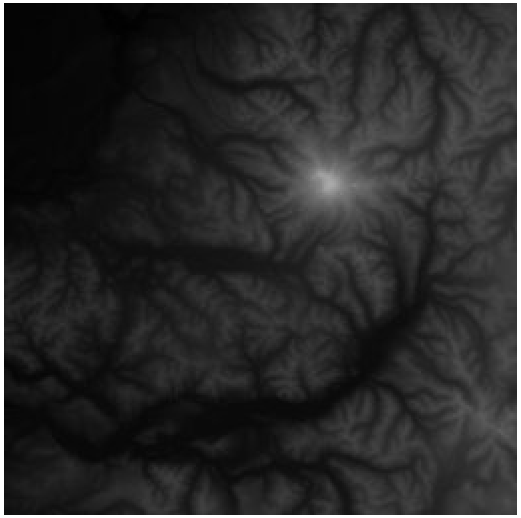
- Shift clipmap levels as user moves



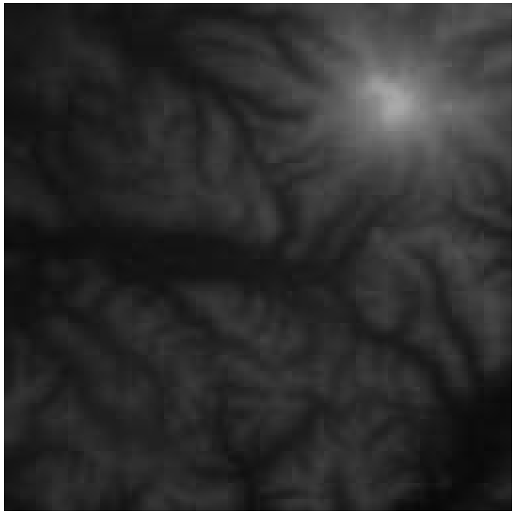
Upsample



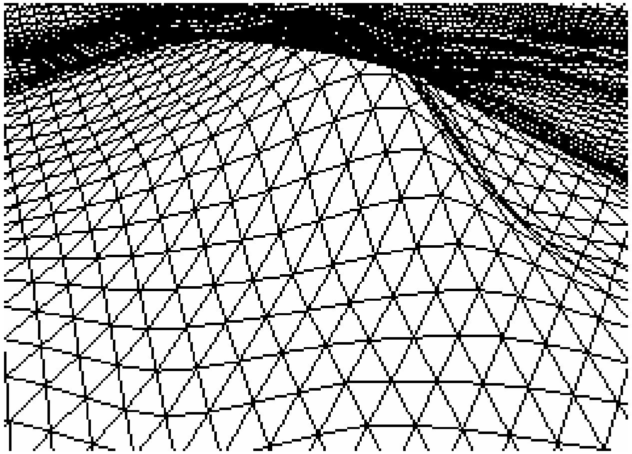




GPU

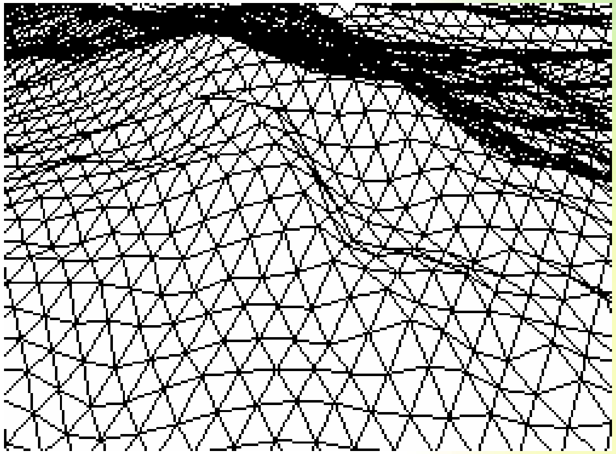


Add residuals

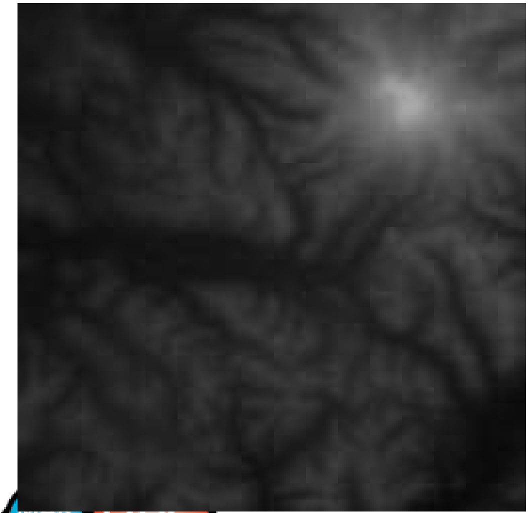


System Memory

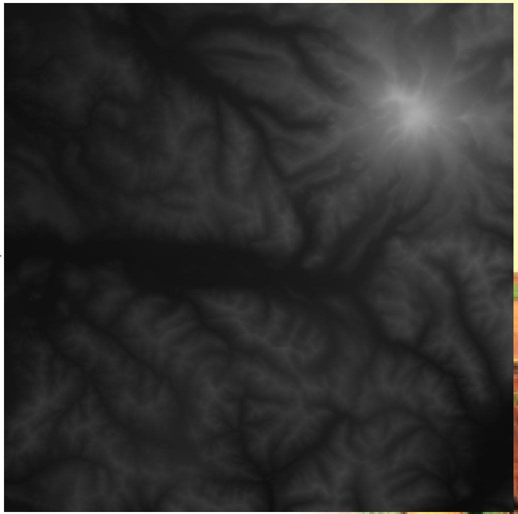
Compressed residuals (350MB for US)



ROI decompression (CPU)

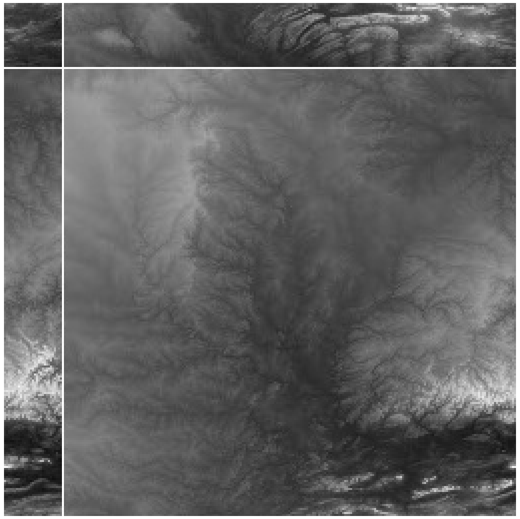


GPU +

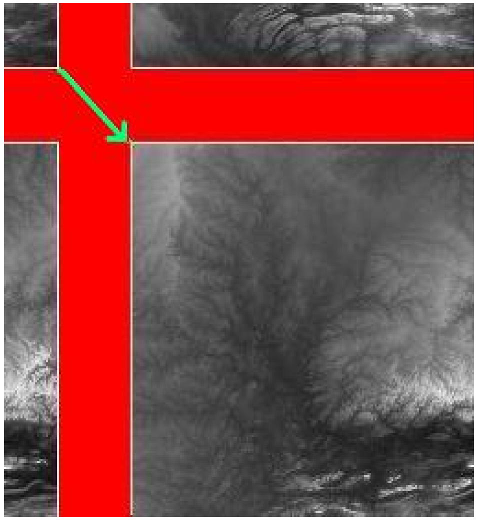


Residual Image in video memory

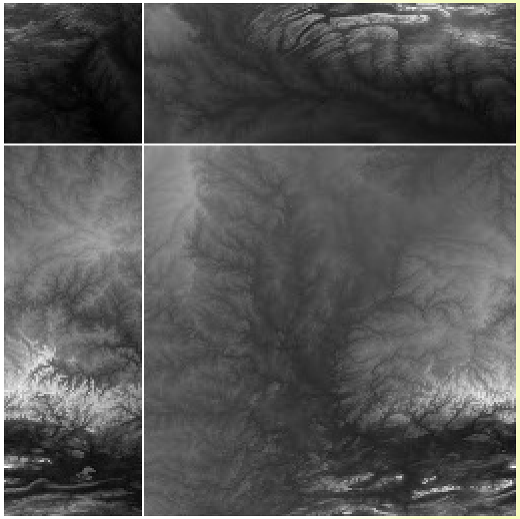
Incremental update



Before update



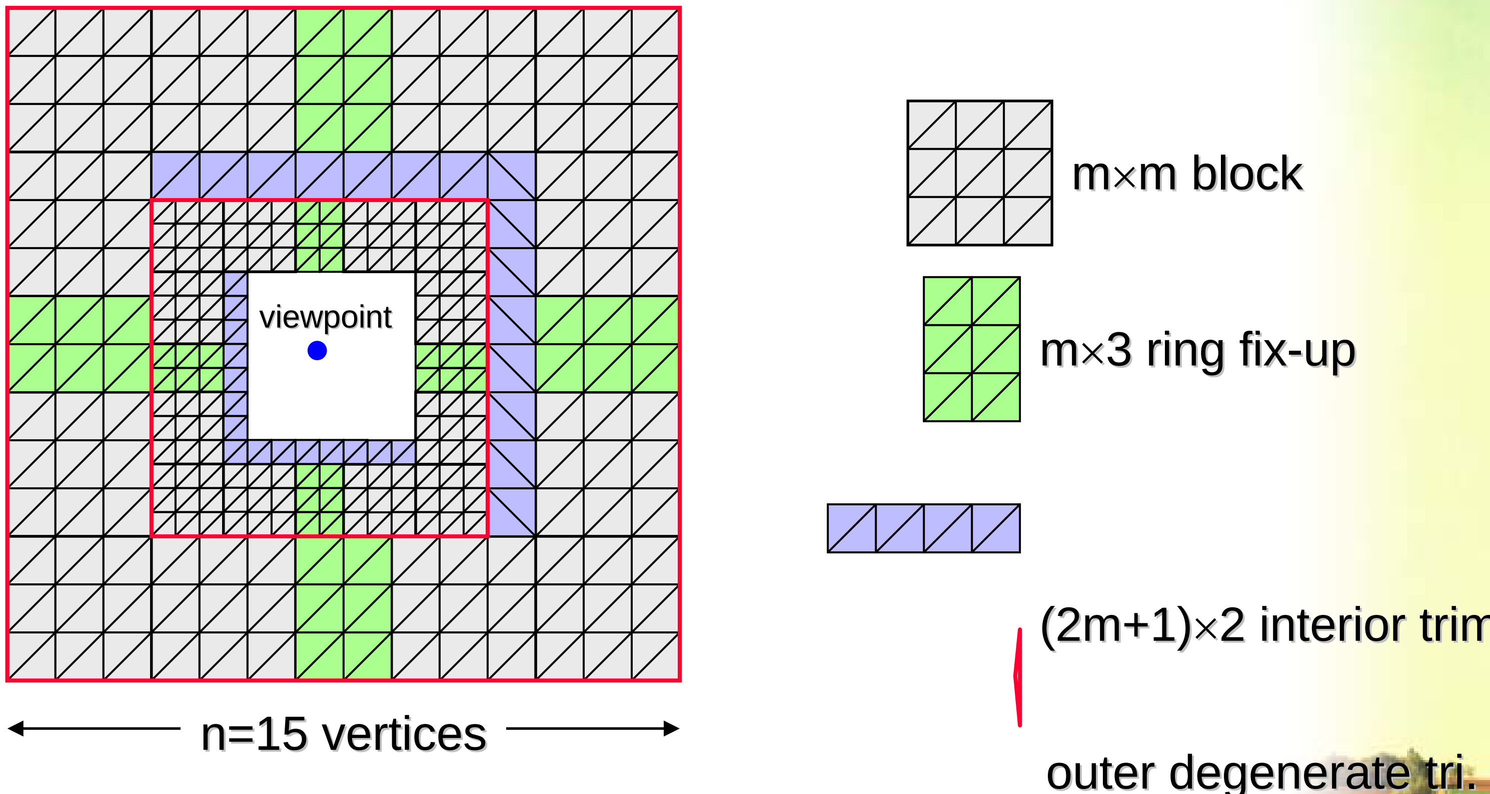
Update region



After update

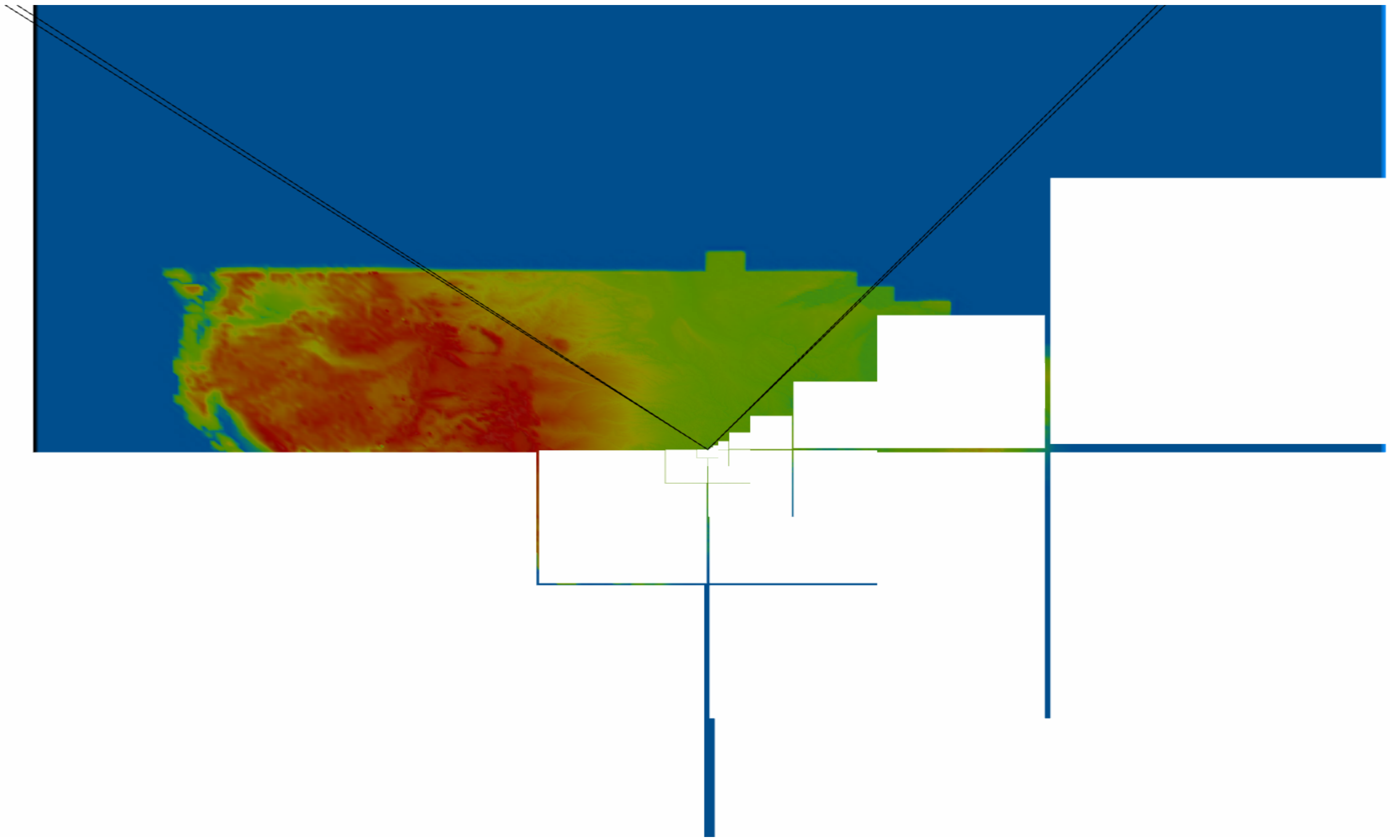
Individual Clipmap Levels

<u>Example</u>: n=15, m=4



- **See Section 2.3.2 in book**

View-frustum culling



- Culling done at block level on CPU
- 2-3x speedup

Timing Results

	Previous Implementation	Current Implementation
Upsampling	3 ms	1.3ms
Decompression	8 ms	8 ms
Normal Map Computation	11 ms	0.6 ms

Performance

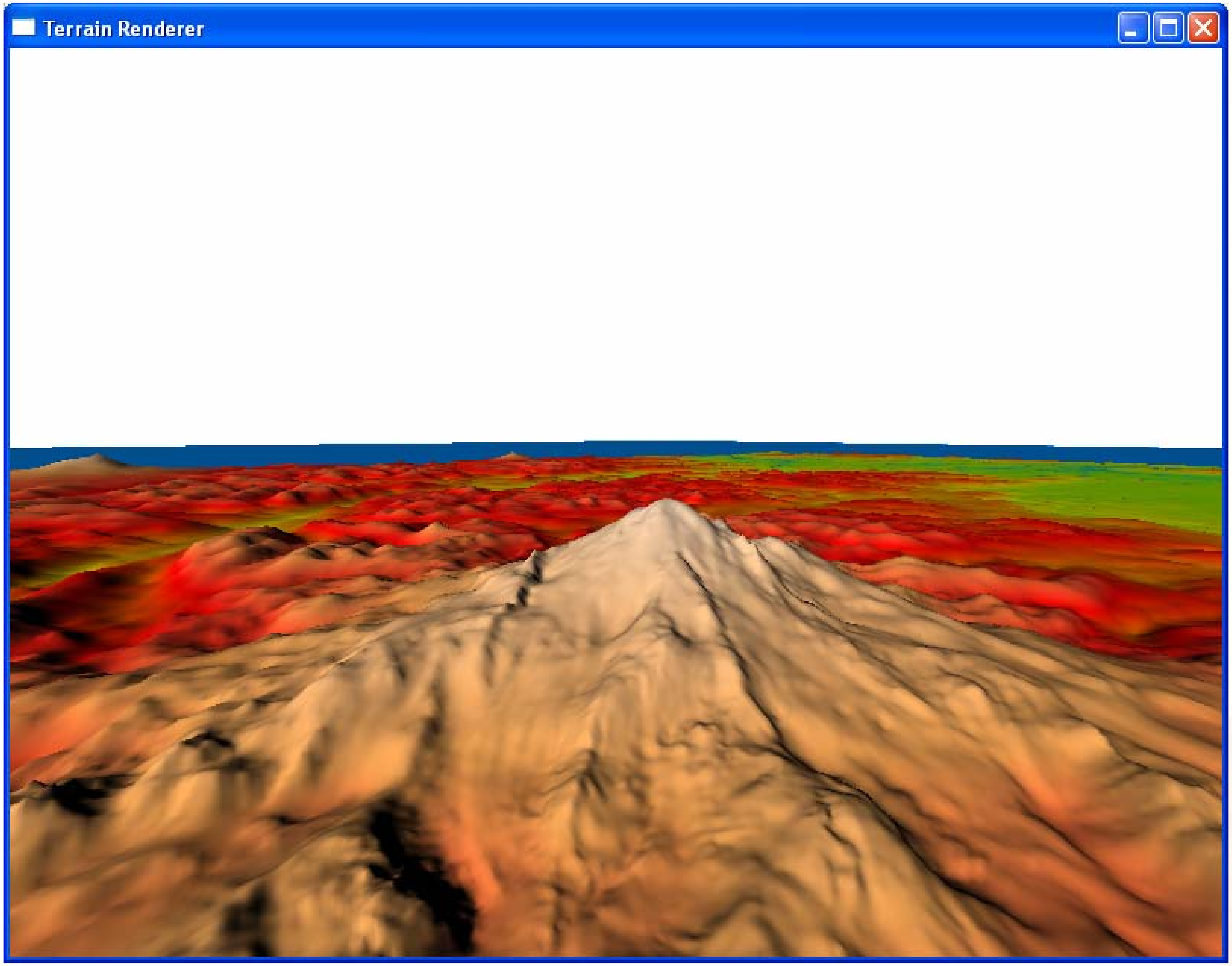
- **Synthesized terrain**
  - 130 frames/second (render-bound)
  - 120 frames/second during user motion
  - 60 million triangles per second
  - CPU utilization: ~0
  - AGP bus utilization: ~0

- **Decompressed terrain**
  - 87 frames/second during viewer motion
  - Decompression on CPU bottleneck

Summary

- Lots of Real-World Applications
  - Games
  - Flight/driving Simulators
  - Virtual Environments
  - Networked Viewer
- Advantages of current framework
  - High compression ratios
  - Terrain synthesis
  - Collision detection within GPU

Demo



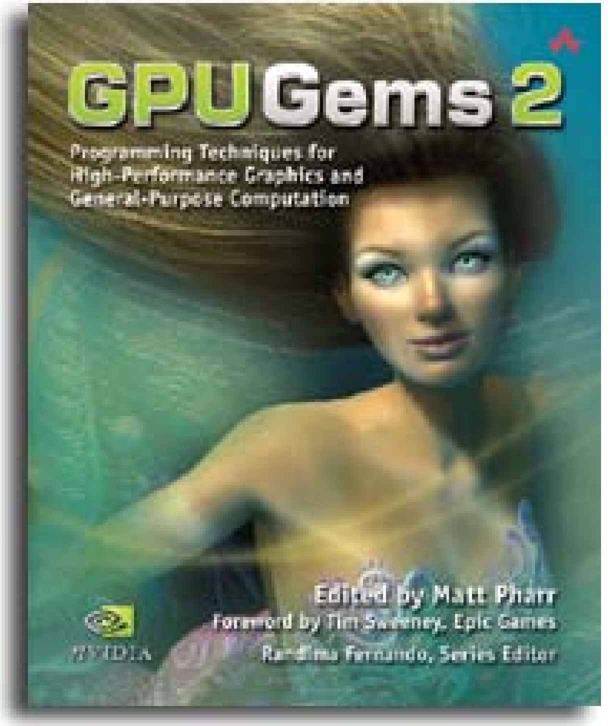
Questions?

- See GPU Gems 2, Chapter 2
- http://developer.nvidia.com
  The Source for GPU Programming
- arul@cs.utah.edu
- 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
• 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
©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.