Interacting with Heightfields

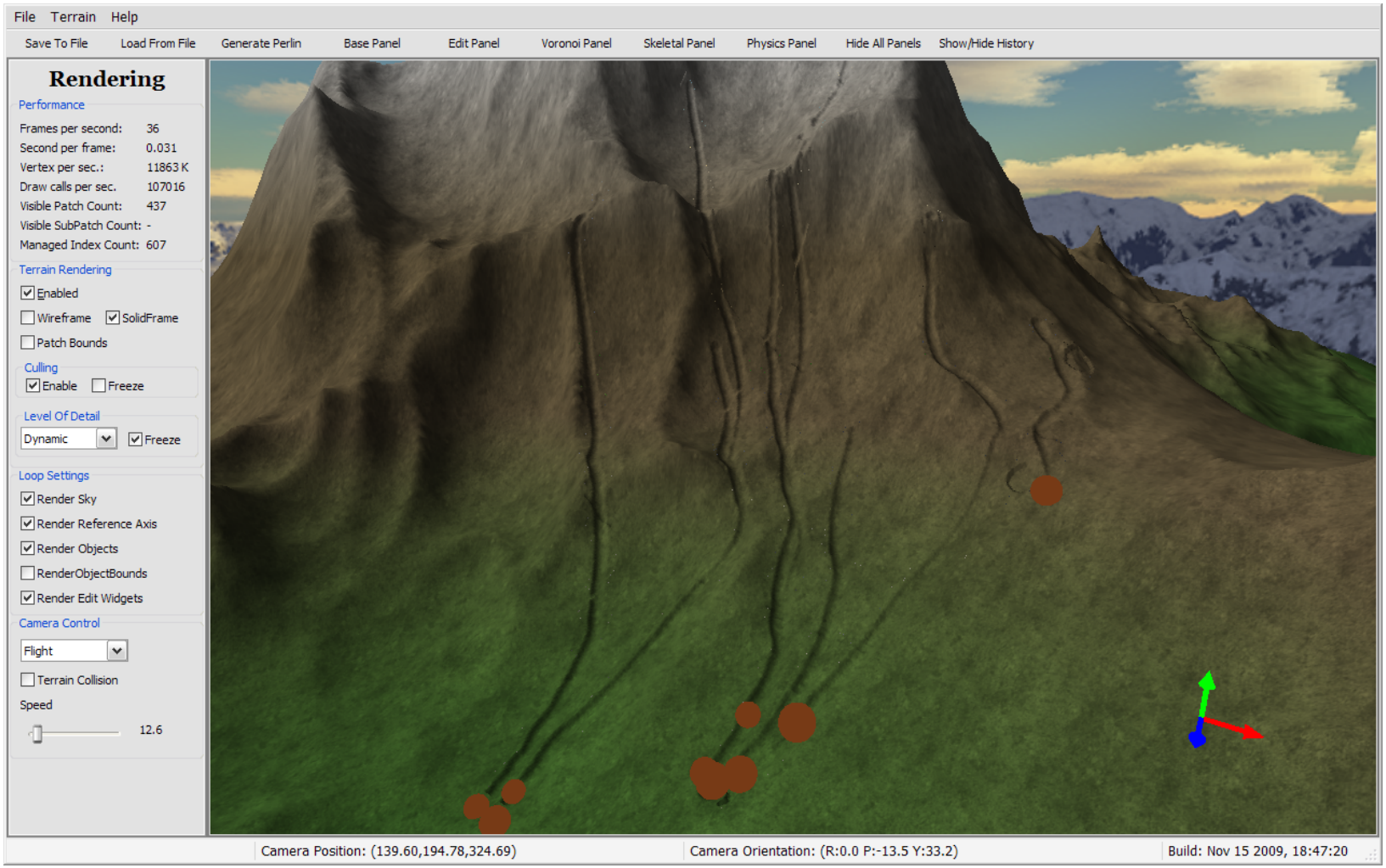
M. Adil Yalçın
yalcin@cs.bilkent.edu.tr

Advisor: Tolga Çapın
Co-advisor: Bülent Özgüç

Bilkent University

CS590 - 16th October, 2009

Figure: Terrain deformations...



Where are we heading?

- Physical interaction with terrain is an *interesting* feature

Where are we heading?

- Physical interaction with terrain is an *interesting* feature
- It can be done in *real-time* for multiple dynamic bodies

Where are we heading?

- Physical interaction with terrain is an *interesting* feature
- It can be done in *real-time* for multiple dynamic bodies
- It is *applicable*.

Where are we heading?

- Physical interaction with terrain is an *interesting* feature
- It can be done in *real-time* for multiple dynamic bodies
- It is *applicable*.
- And other sub-problems exists:

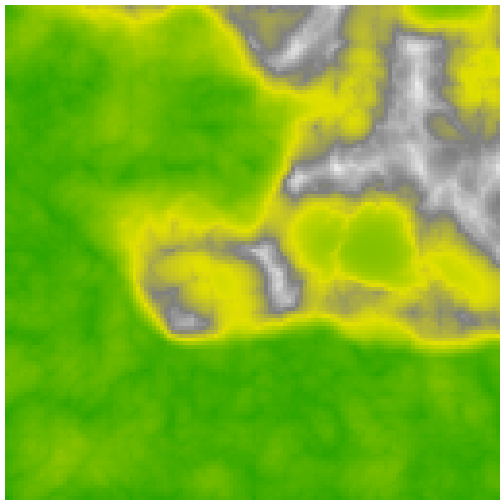
Where are we heading?

- Physical interaction with terrain is an *interesting* feature
- It can be done in *real-time* for multiple dynamic bodies
- It is *applicable*.
- And other sub-problems exists:
  - Editing
  - Procedural generation

Where are we heading?

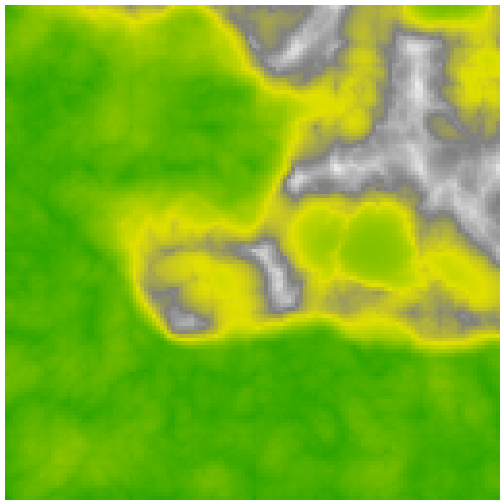
- Physical interaction with terrain is an *interesting* feature
- It can be done in *real-time* for multiple dynamic bodies
- It is *applicable*.
- And other sub-problems exists:
  - Editing
  - Procedural generation
- Introducing OpenREng...
  - And GPUs as general-purpose hardware

Terrain ⇒ Heightfield , Heightmap



Heightfield : 2D grid structure, cells store height values.

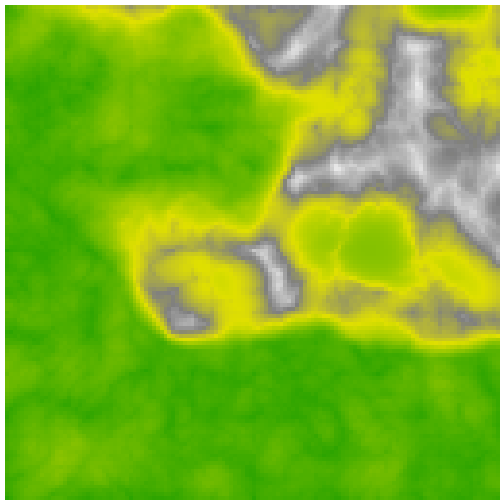
Terrain ⇒ Heightfield , Heightmap



Heightfield : 2D grid structure, cells store height values.

Large-scale : 1024x1024 (1M samples) and above

Terrain ⇒ Heightfield , Heightmap



Heightfield : 2D grid structure, cells store height values.

Large-scale : 1024x1024 (1M samples) and above

Level of detail : "Quality where it can be appreciated"

What is done?

Trying to identify a basic problem, including many missing parts...

What is done? (Basic Layer)

CS503 Modeling and Simulation

What is done? (Basic Layer)

CS503 Modeling and Simulation

- Hierarchical large-scale structure: Quad-tree ▸ Info

What is done? (Basic Layer)

CS503 Modeling and Simulation

- Hierarchical large-scale structure: Quad-tree ▸ Info
- Level-Of-Detail: Geo-Mipmapping ▸ Info

What is done? (Basic Layer)

CS503 Modeling and Simulation

- Hierarchical large-scale structure: Quad-tree ▸ Info
- Level-Of-Detail: Geo-Mipmapping ▸ Info
- Culling: non visible object detection ▸ Info

What is done? (Basic Layer)

CS503 Modeling and Simulation

- Hierarchical large-scale structure: Quad-tree ▸ Info
- Level-Of-Detail: Geo-Mipmapping ▸ Info
- Culling: non visible object detection ▸ Info
- Rendering using fixed-function GPU pipeline

What is done? (Editing)

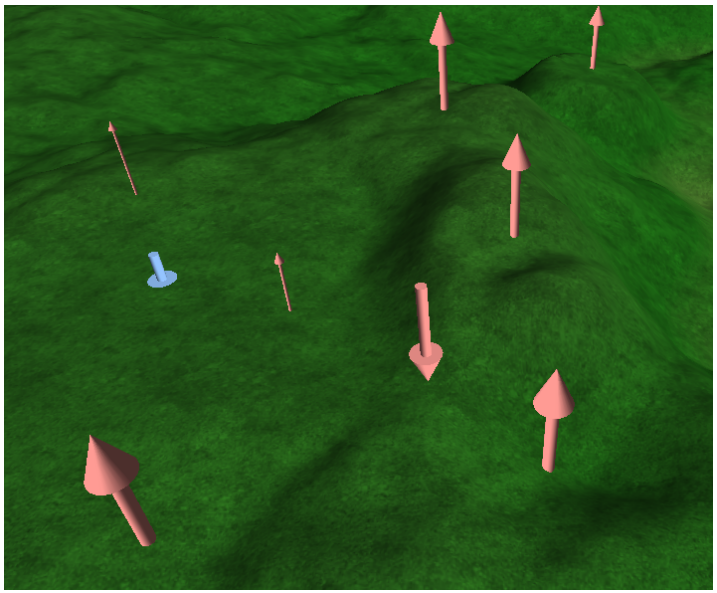
CS566 User Interface Design

What is done? (Editing)

CS566 User Interface Design

Problem : Visualizing Editing Operations

Solution : Using 3D Widgets



Published:

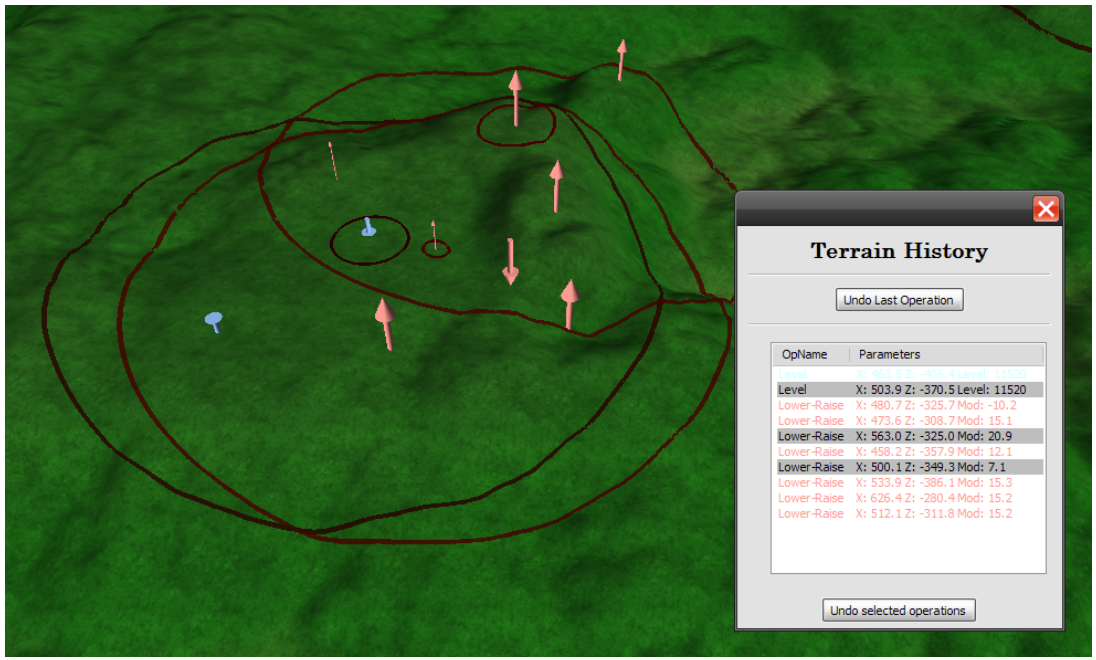
Editing Heightfield using 3D Widgets and History Management, Adil Yalcin, Tolga Capin, ISCIS '09

What is done? (Editing)

Problem : Undo only latest operation at one time.

Solution : Out-of-order undoing

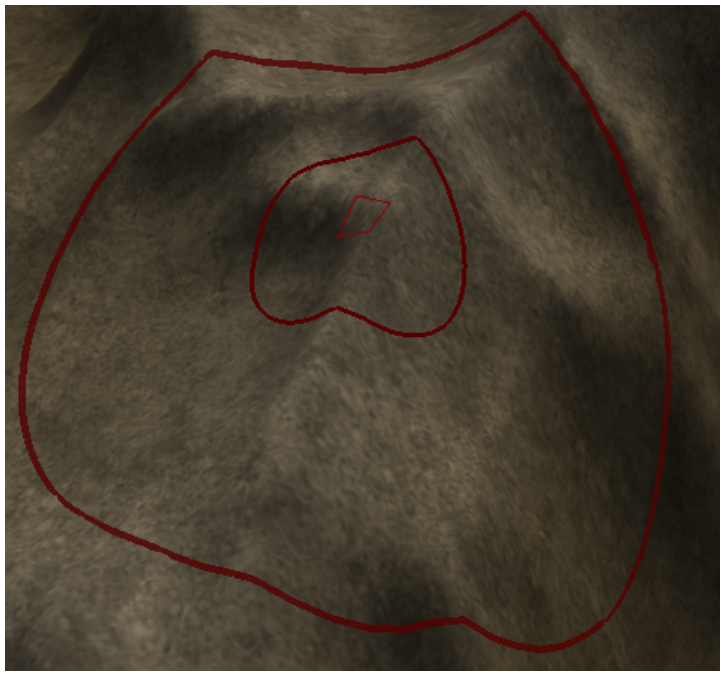
- Note: A region is affected by multiple op's.



What is done? (Editing)

Problem : Visualizing selection (Discrete vs Continuous)

Solution : Discrete Grid, Continuous Circle



What is done? (Collision detection)

CS565 Application of Computer Graphics

- Initial collision detection using old GPU pipeline

What is done? (Collision detection)

CS565 Application of Computer Graphics

- Initial collision detection using old GPU pipeline
  - Without GPU programs, with stencil buffers.

What is done? (Collision detection)

CS565 Application of Computer Graphics

- Initial collision detection using old GPU pipeline
  - Without GPU programs, with stencil buffers.
- Physical interaction layer : Detail layer

What is done? (Collision detection)

CS565 Application of Computer Graphics

- Initial collision detection using old GPU pipeline
  - Without GPU programs, with stencil buffers.
- Physical interaction layer : Detail layer
- Blending between detail and basic layer

What is done? (Improved collision, physics)

CS568 Advanced Topics in Computer Graphics

- Collision and unified rendering using programmable OpenGL pipeline.

What is done? (Improved collision, physics)

CS568 Advanced Topics in Computer Graphics

- Collision and unified rendering using programmable OpenGL pipeline.
  - Multiple body collision per patch

What is done? (Improved collision, physics)

CS568 Advanced Topics in Computer Graphics

- Collision and unified rendering using programmable OpenGL pipeline.
  - Multiple body collision per patch
  - Compression limiting

What is done? (Improved collision, physics)

CS568 Advanced Topics in Computer Graphics

- Collision and unified rendering using programmable OpenGL pipeline.
  - Multiple body collision per patch
  - Compression limiting
- Managing terrain height on the GPU

What is done? (Improved collision, physics)

CS568 Advanced Topics in Computer Graphics

- Collision and unified rendering using programmable OpenGL pipeline.
  - Multiple body collision per patch
  - Compression limiting
- Managing terrain height on the GPU
  - Stored as a 2D GPU Texture data structure.

What is done? (Improved collision, physics)

CS568 Advanced Topics in Computer Graphics

- Collision and unified rendering using programmable OpenGL pipeline.
  - Multiple body collision per patch
  - Compression limiting
- Managing terrain height on the GPU
  - Stored as a 2D GPU Texture data structure.
- Generating contacts, physical simulation support

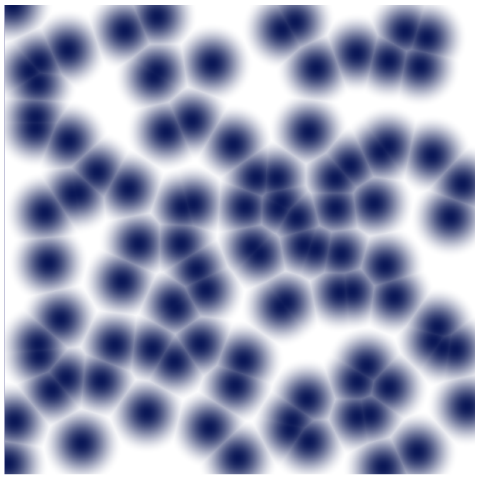
What is done? (Improved collision, physics)

CS568 Advanced Topics in Computer Graphics

- Collision and unified rendering using programmable OpenGL pipeline.
  - Multiple body collision per patch
  - Compression limiting
- Managing terrain height on the GPU
  - Stored as a 2D GPU Texture data structure.
- Generating contacts, physical simulation support
  - Using ODE: OpenDynamicsEngine (www.ode.org)

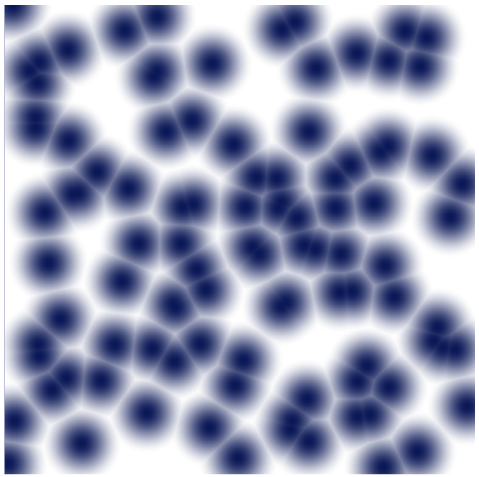
What is done? (Other course-work)

- CS564 Computational Geometry
  - Generating Distance Transform / Voronoi on GPU (to be used in collision erosion / displacement)



What is done? (Other course-work)

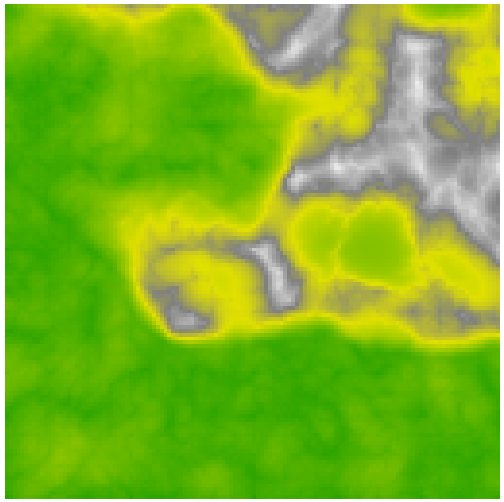
- CS564 Computational Geometry
  - Generating Distance Transform / Voronoi on GPU (to be used in collision erosion / displacement)



- CS567 Animation
  - A character walking on terrain, leaving footsteps behind
  - Aim: Completely physical-driven walk behaviour
  - Work In Progress...

What is done? (Generating Terrain)

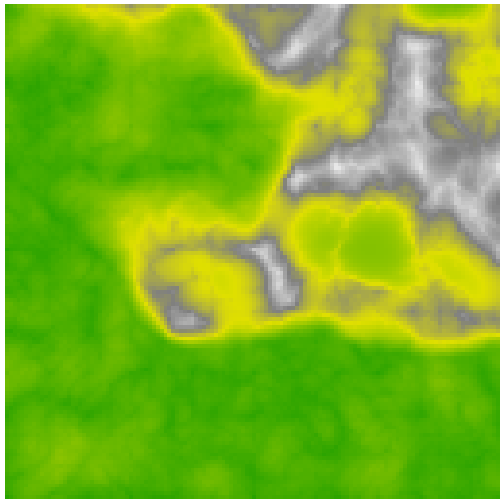
Requirement: Natural results...



What is done? (Generating Terrain)

Requirement: Natural results...
Sub-requirements:

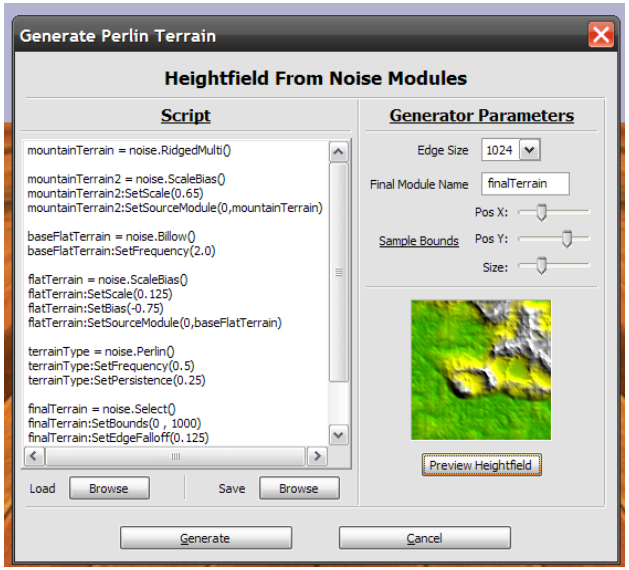
- Pseudo-random
- Band-limited spatial frequency (smoothness)
- No repeating patterns



What is done? (Generating Terrain)

Solution:

- Noise source: Perlin Noise
- Mixing - blending noise sources (modules) together
  - Using libnoise (libnoise.sourceforge.net/)
- Scripting through Lua language
  - Define / preview / create natural noise on run-time



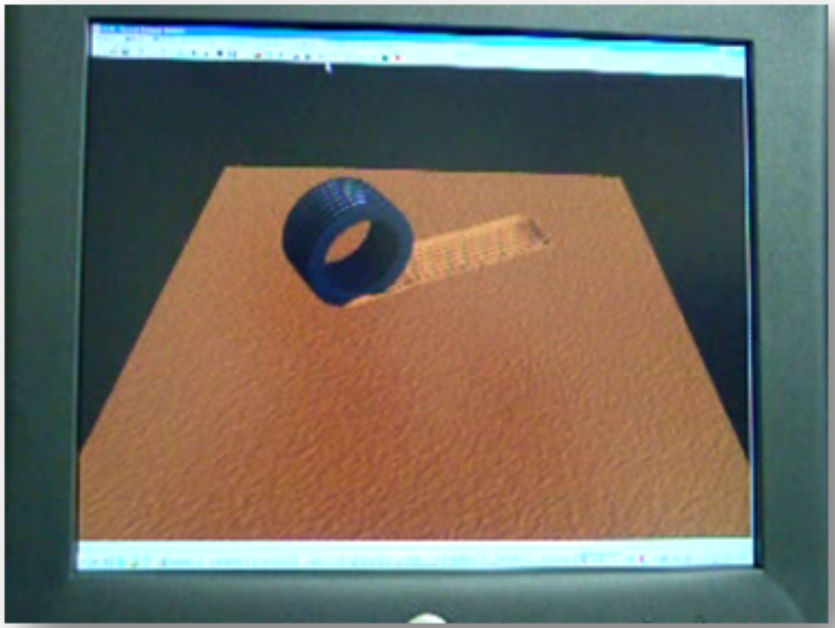
Physical Interaction

Selected Previous Work

- Animating Sand, Mud and Snow (1997)
  - The basic method for collision detection and displacement

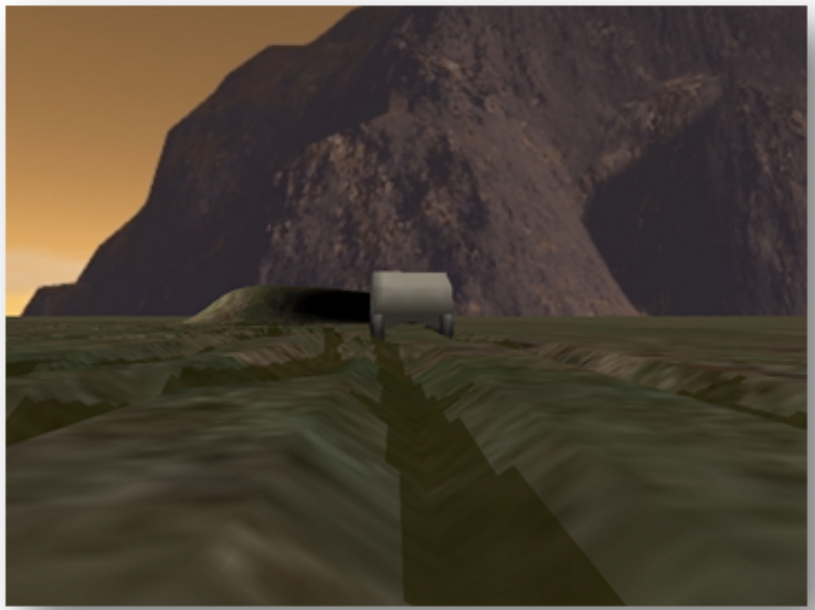
Selected Previous Work

- Displaying Marks on Soft Grounds Created by Objects
  - Collision detection on GPU (old pipeline)



Selected Previous Work

- Real-Time GPU-Based Simulation of Dynamic Terrain
  - Heightmap as a 2D GPU Texture. No erosion.



The Method

- Broad-phase collision detection
- Narrow-phase collision detection
- Contact-data and sub-patch generation

Broad-phase collision detection

- Collide object bounding box vs terrain bounding box hierarchy

Broad-phase collision detection

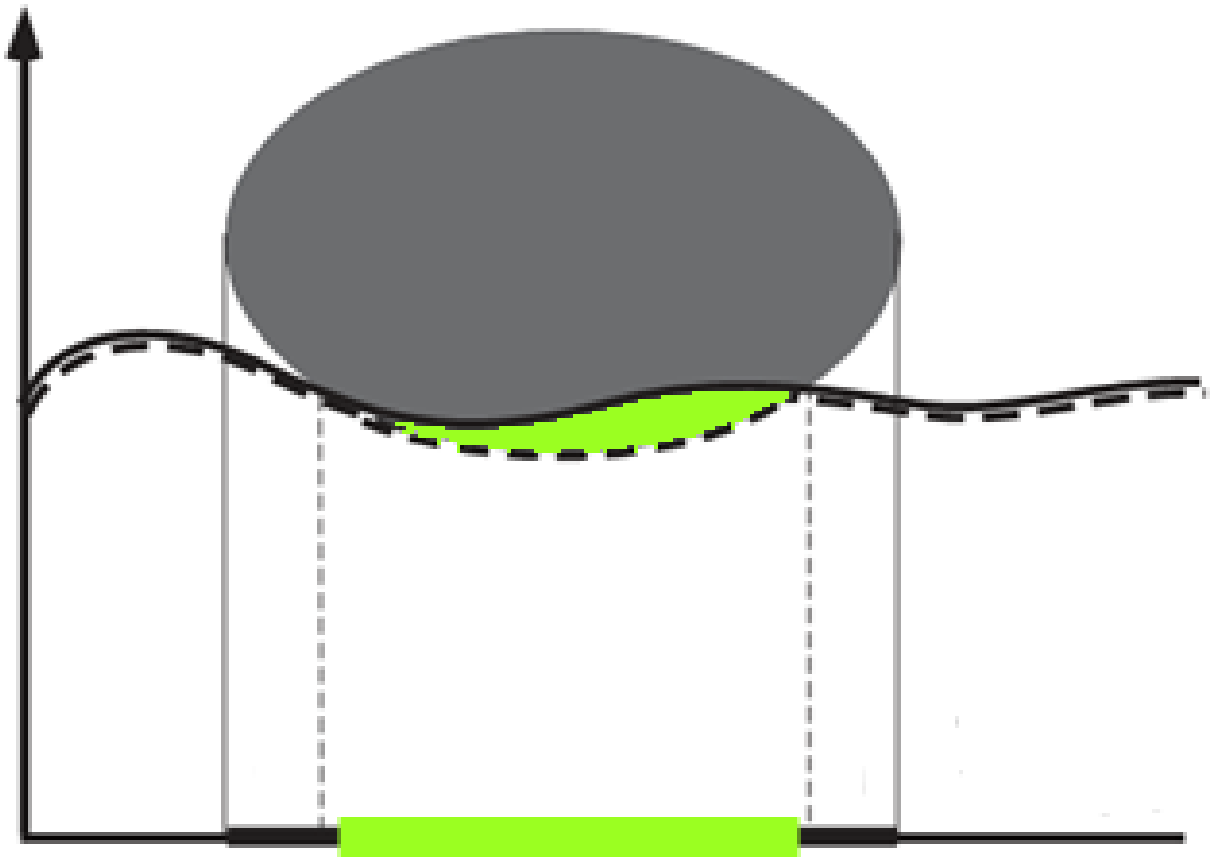
- Collide object bounding box vs terrain bounding box hierarchy
- Result:

Broad-phase collision detection

- Collide object bounding box vs terrain bounding box hierarchy
- Result:
  - Object-terrain patch pairs that will be collided in narrow-phase

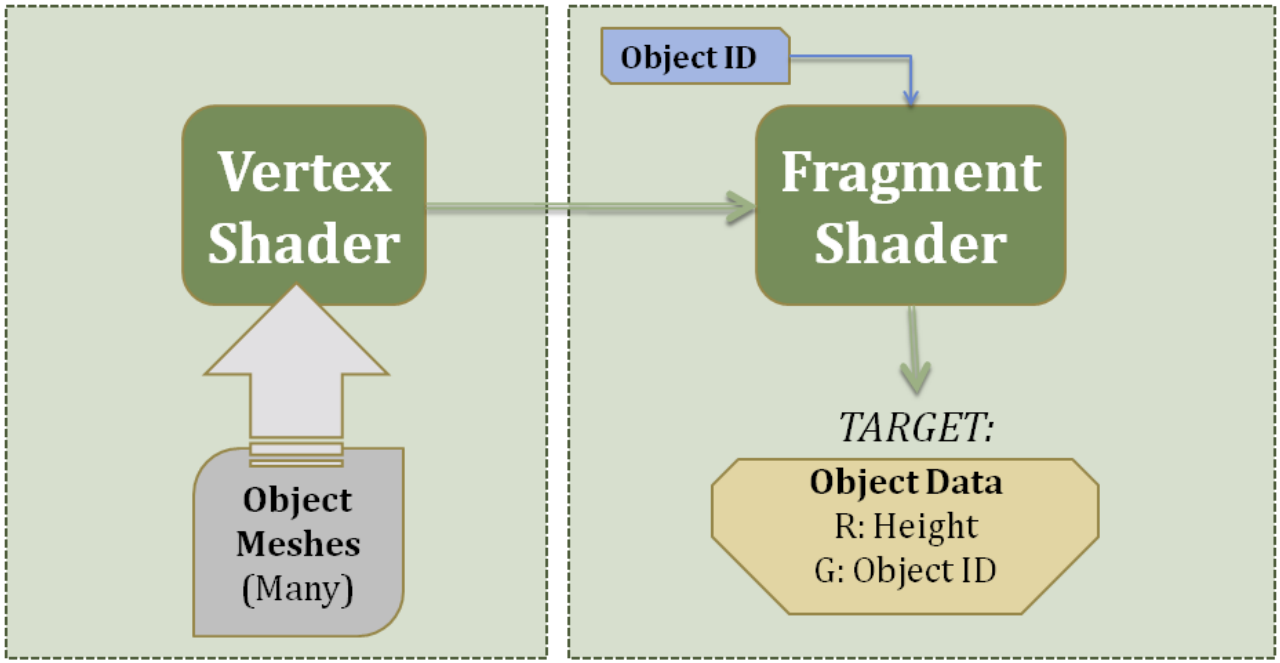
Narrow-phase collision detection

Figure: The basic idea



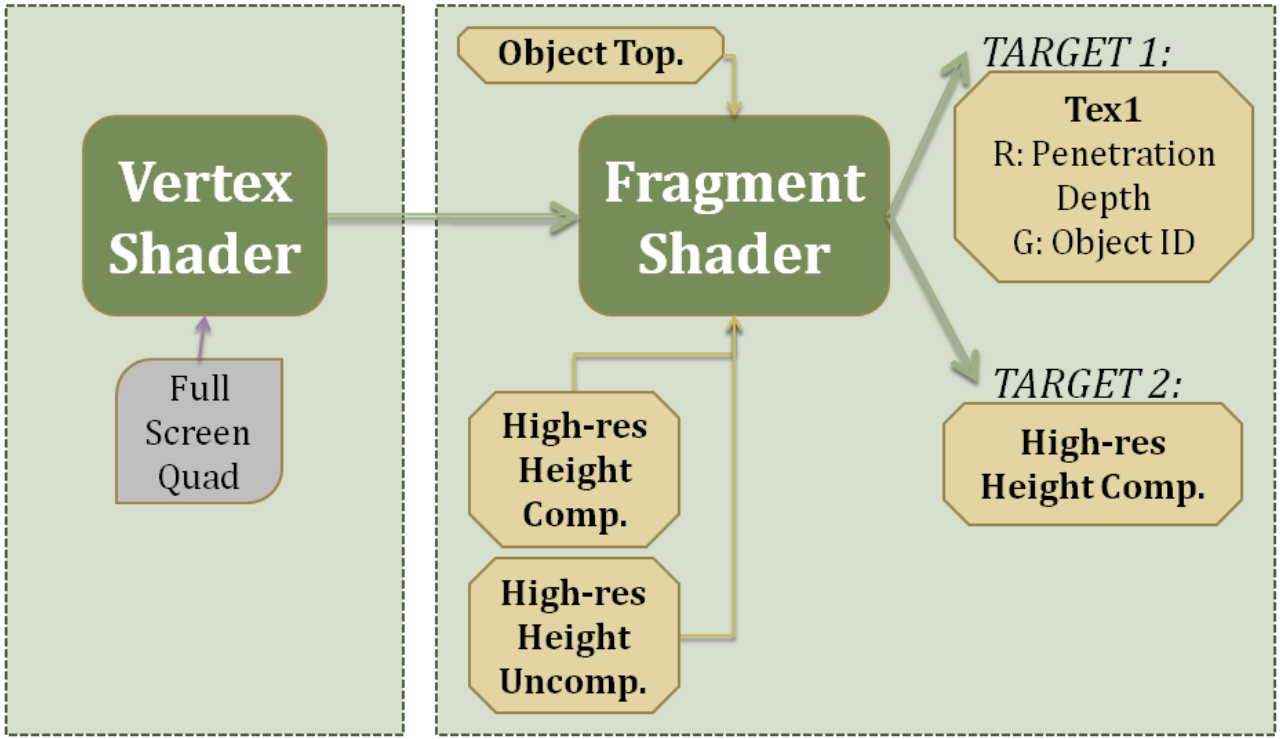
Narrow-phase collision detection

Figure: 1. Generating object topology



Narrow-phase collision detection

Figure: 2. Collision And Compression



Contact-data and sub-patch generation

- Related data structures managed on CPU.

Contact-data and sub-patch generation

- Related data structures managed on CPU.
- Each object is simulated using its own contact data.

Contact-data and sub-patch generation

- Related data structures managed on CPU.
- Each object is simulated using its own contact data.
- For rendering:

Contact-data and sub-patch generation

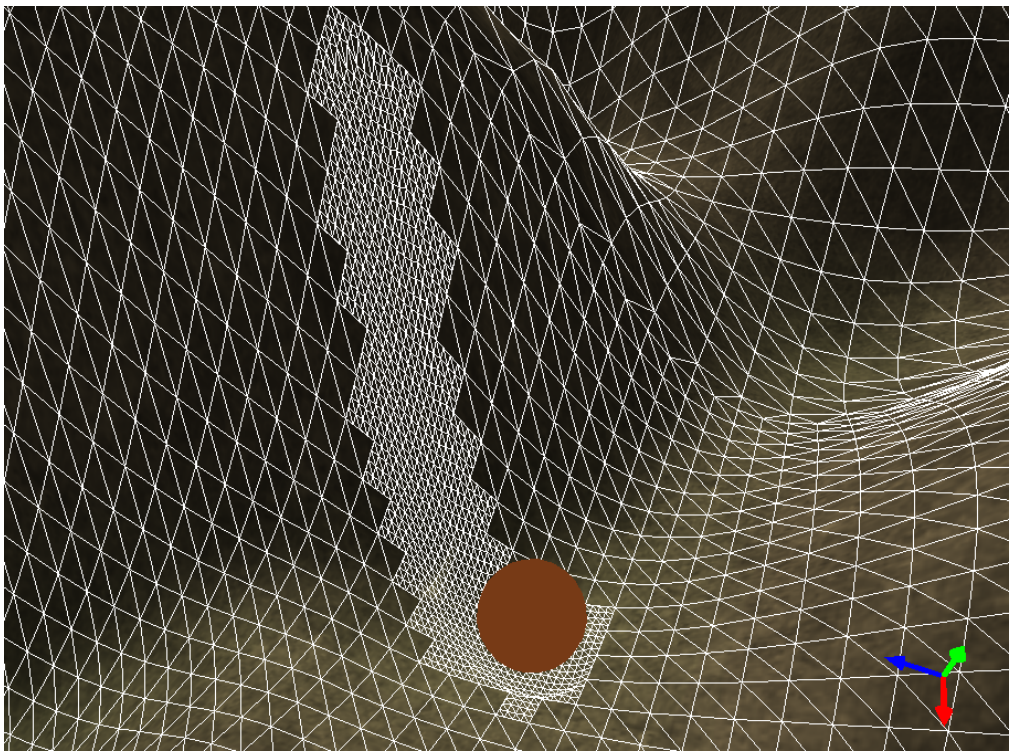
- Related data structures managed on CPU.
- Each object is simulated using its own contact data.
- For rendering:
  - Each patch is notified of collided regions

Contact-data and sub-patch generation

- Related data structures managed on CPU.
- Each object is simulated using its own contact data.
- For rendering:
  - Each patch is notified of collided regions
  - Sub-patch structures attached

Result

Figure: A sphere object fell down from a hill, deforming the geometry...



Introducing OpenREng

Part of the 3DPHONE Project.

Introducing OpenREng

Part of the 3DPHONE Project.

Approach

- Open-source (hosted @ SourceForge)

Introducing OpenREng

Part of the 3DPHONE Project.

Approach

- Open-source (hosted @ SourceForge)
- C++ / Object Oriented / 21 KLOC / 8 months

Introducing OpenREng

Part of the 3DPHONE Project.

Approach

- Open-source (hosted @ SourceForge)
- C++ / Object Oriented / 21 KLOC / 8 months
- Cross-Op.Sys.: Windows / Linux

Introducing OpenREng

Part of the 3DPHONE Project.

Approach

- Open-source (hosted @ SourceForge)
- C++ / Object Oriented / 21 KLOC / 8 months
- Cross-Op.Sys.: Windows / Linux
- Cross-hardware:

Introducing OpenREng

Part of the 3DPHONE Project.

Approach

- Open-source (hosted @ SourceForge)
- C++ / Object Oriented / 21 KLOC / 8 months
- Cross-Op.Sys.: Windows / Linux
- Cross-hardware:
  - Mobile: OpenGL ES 2.0

Introducing OpenREng

Part of the 3DPHONE Project.

Approach

- Open-source (hosted @ SourceForge)
- C++ / Object Oriented / 21 KLOC / 8 months
- Cross-Op.Sys.: Windows / Linux
- Cross-hardware:
  - Mobile: OpenGL ES 2.0
  - Desktop: OpenGL 3.0 and above

Introducing OpenREng

Approach

- Automated (Retained-mode) rendering

Introducing OpenREng

Approach

- Automated (Retained-mode) rendering
- Low-level and easily extensible

Introducing OpenREng

Approach

- Automated (Retained-mode) rendering
- Low-level and easily extensible
- General-purpose, designed for high performance

Introducing OpenREng

Approach

- Automated (Retained-mode) rendering
- Low-level and easily extensible
- General-purpose, designed for high performance
- Data driven

Introducing OpenREng

Approach

- Automated (Retained-mode) rendering
- Low-level and easily extensible
- General-purpose, designed for high performance
- Data driven
- High-level easy management of 3D scene

A View on Latest Graphics API's

Keypoints:

- GPU is designed for rendering, but not restricted to graphics.

A View on Latest Graphics API's

Keypoints:

- GPU is designed for rendering, but not restricted to graphics.
- GPU for "Parallel/Stream Computing".

A View on Latest Graphics API's

Keypoints:

- GPU is designed for rendering, but not restricted to graphics.
- GPU for "Parallel/Stream Computing".
  - Promoted by OpenCL, NVIDIA's Cuda and ATI's Stream

A View on Latest Graphics API's

Keypoints:

- GPU is designed for rendering, but not restricted to graphics.
- GPU for "Parallel/Stream Computing".
  - Promoted by OpenCL, NVIDIA's Cuda and ATI's Stream

A View on Latest Graphics API's

Keypoints:

- GPU is designed for rendering, but not restricted to graphics.
- GPU for "Parallel/Stream Computing".
  - Promoted by OpenCL, NVIDIA's Cuda and ATI's Stream

The new OpenGL specifications...

- Becomes *"Closer to Hardware"*, provides only low-level primitives

A View on Latest Graphics API's

Keypoints:

- GPU is designed for rendering, but not restricted to graphics.
- GPU for "Parallel/Stream Computing".
  - Promoted by OpenCL, NVIDIA's Cuda and ATI's Stream

The new OpenGL specifications...

- Becomes *"Closer to Hardware"*, provides only low-level primitives
- Shifts *"Rendering management load to application"*

A View on Latest Graphics API's

Keypoints:

- GPU is designed for rendering, but not restricted to graphics.
- GPU for "Parallel/Stream Computing".
  - Promoted by OpenCL, NVIDIA's Cuda and ATI's Stream

The new OpenGL specifications...

- Becomes *"Closer to Hardware"*, provides only low-level primitives
- Shifts *"Rendering management load to application"*
- *"No standard (pre-defined) rendering method enforced"*

A View on Latest Graphics API's

Keypoints:

- GPU is designed for rendering, but not restricted to graphics.
- GPU for "Parallel/Stream Computing".
  - Promoted by OpenCL, NVIDIA's Cuda and ATI's Stream

The new OpenGL specifications...

- Becomes *"Closer to Hardware"*, provides only low-level primitives
- Shifts *"Rendering management load to application"*
- *"No standard (pre-defined) rendering method enforced"*
- And *converges Desktop and Handheld*

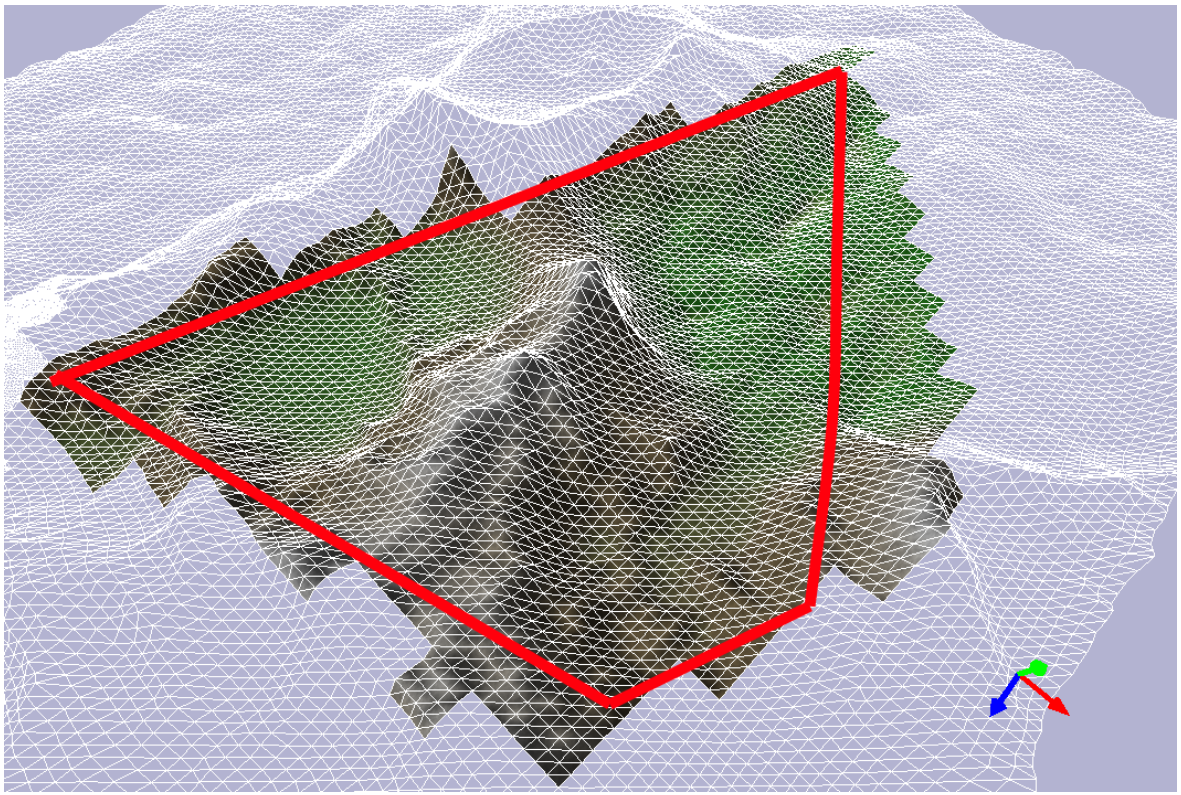
Thank you. Questions are welcome.

Additional Information:

- www.cs.bilkent.edu.tr/∼yalcin
- http://sourceforge.net/apps/mediawiki/openreng

Frustum Culling

Figure: Red lines denote camera viewpoint.



◂ Return

Hierachical Terrain Patches

Figure: 2D Quad-Tree
(from gamedev.net)

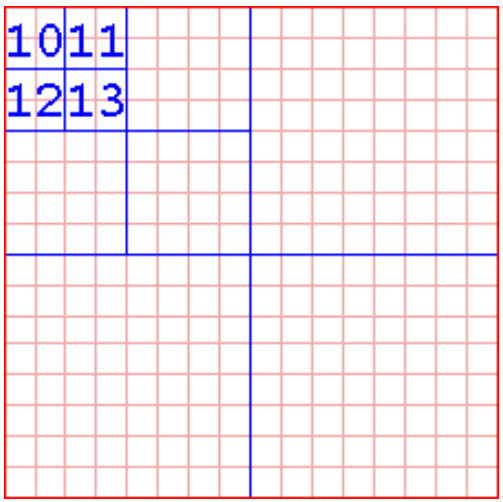
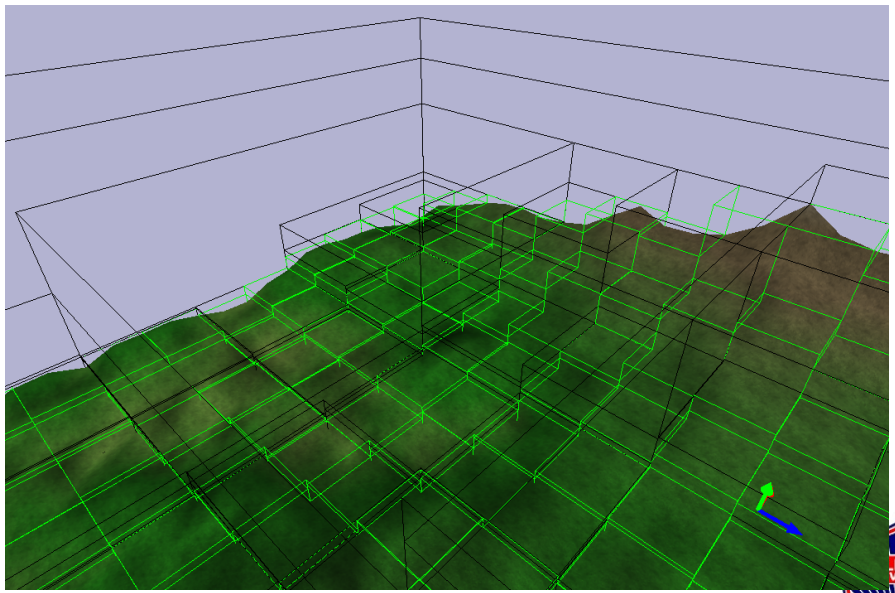


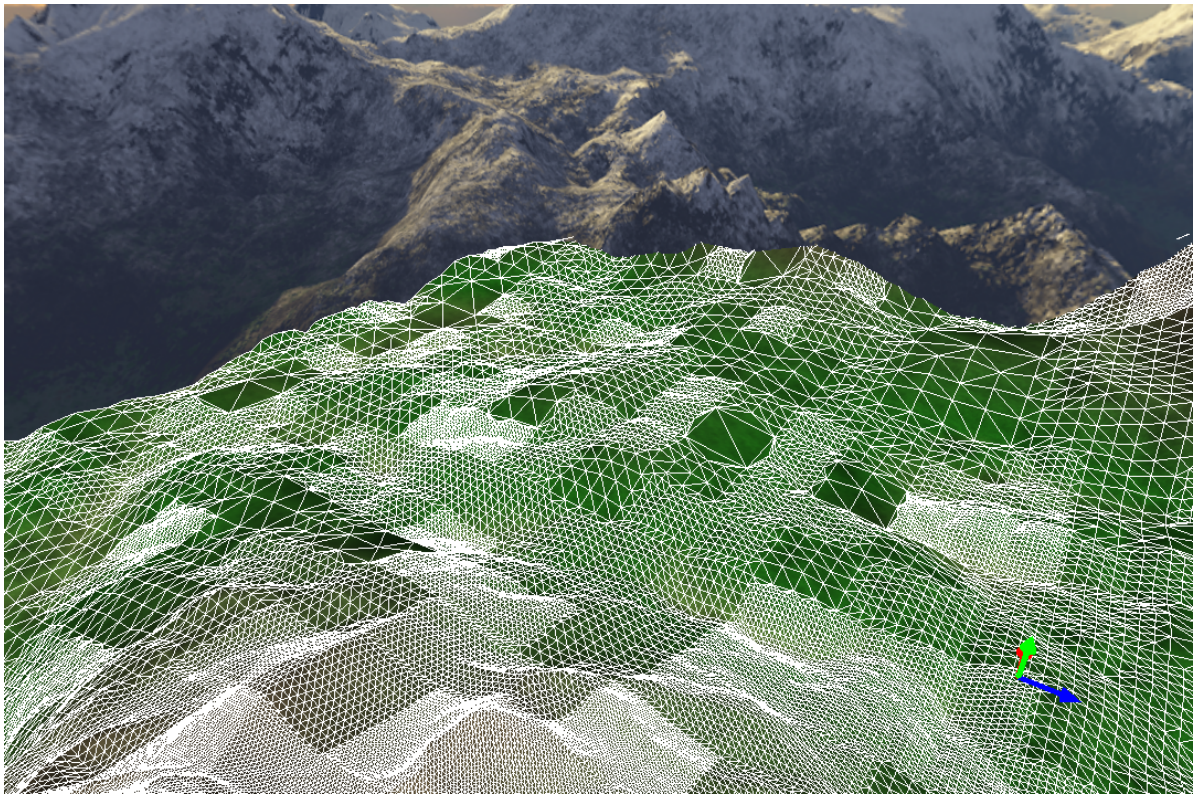
Figure: 3D Quad-Tree



◂ Return

Geo-Mipmapping

Figure: Error metric: Screen-space , non-uniform LoD distribution



Return