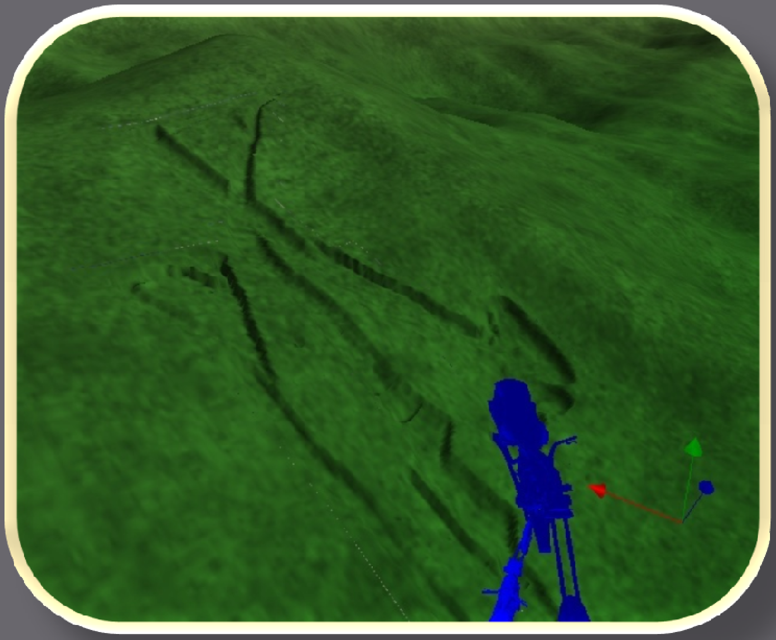
Deforming Terrain Using Rigid Bodies



Adil YALÇIN

Progress By Comparison

Term I (CS565)

- All fixed-function pipeline
- Height position updates on CPU
- Initial usage of GPU for collision detection.
- Two-level system: Patch-Subpatch (lowres-highres) structures defined and managed.

Term II (CS568)

- All programmable pipeline
- Height position updates on GPU
- Limited compression
- Supports multiple object collision per patch.
- Terrain data unified / refined
  - Vertex Texture Lookups (SM3)
  - Shared position data.
  - 2D Heightmap used in all steps
- Skybox rendering

Progress By Comparison (Cont’d)

Term I (CS565)

- No physics contact data generated from contacts.

Term II (CS568)

- Contact data fed into a separate 3D rigid body physics engine (ODE)
  - Note: Parameter tweaking required for stable animations.
- Various improvements
  - Index data can be shared between patches (index manager)

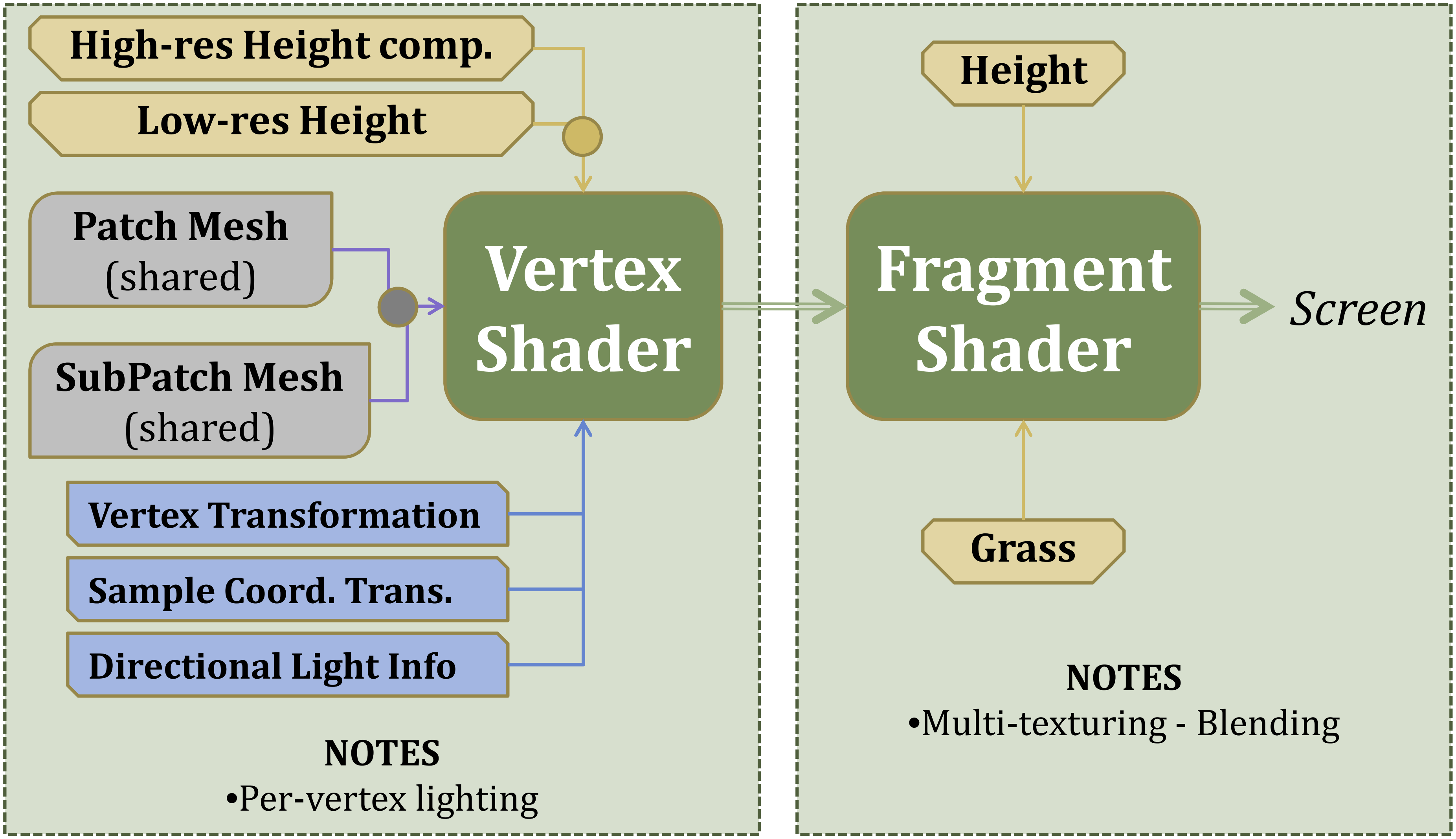
RESULT

All collision / compression / decompression is ready to be managed on the GPU.

Unified Terrain Renderer

- Fixed texture coordinate generation problem
  - The problem was:
    - The modelView matrix holds camera-space orientations.
    - Texture coordinate uses world-space orientations.
    - Separated patchTransformation matrix.
- Sub-Patch rendering implemented
  - SubPatch and Patch share shaders.
    - They generally follow the same path. Path is based on uniform data.

Unified Terrain Renderer



NOTES

- Per-vertex lighting

NOTES

- Multi-texturing - Blending

Height Resolution Enhancer

Aim

- Achieve the same topology of rendered data. High resolution will be used for collision detection, compression and sub-patch rendering.

Input

- Low resolution height texture
  - Ex: 2^5+1 = 32+1 = 33

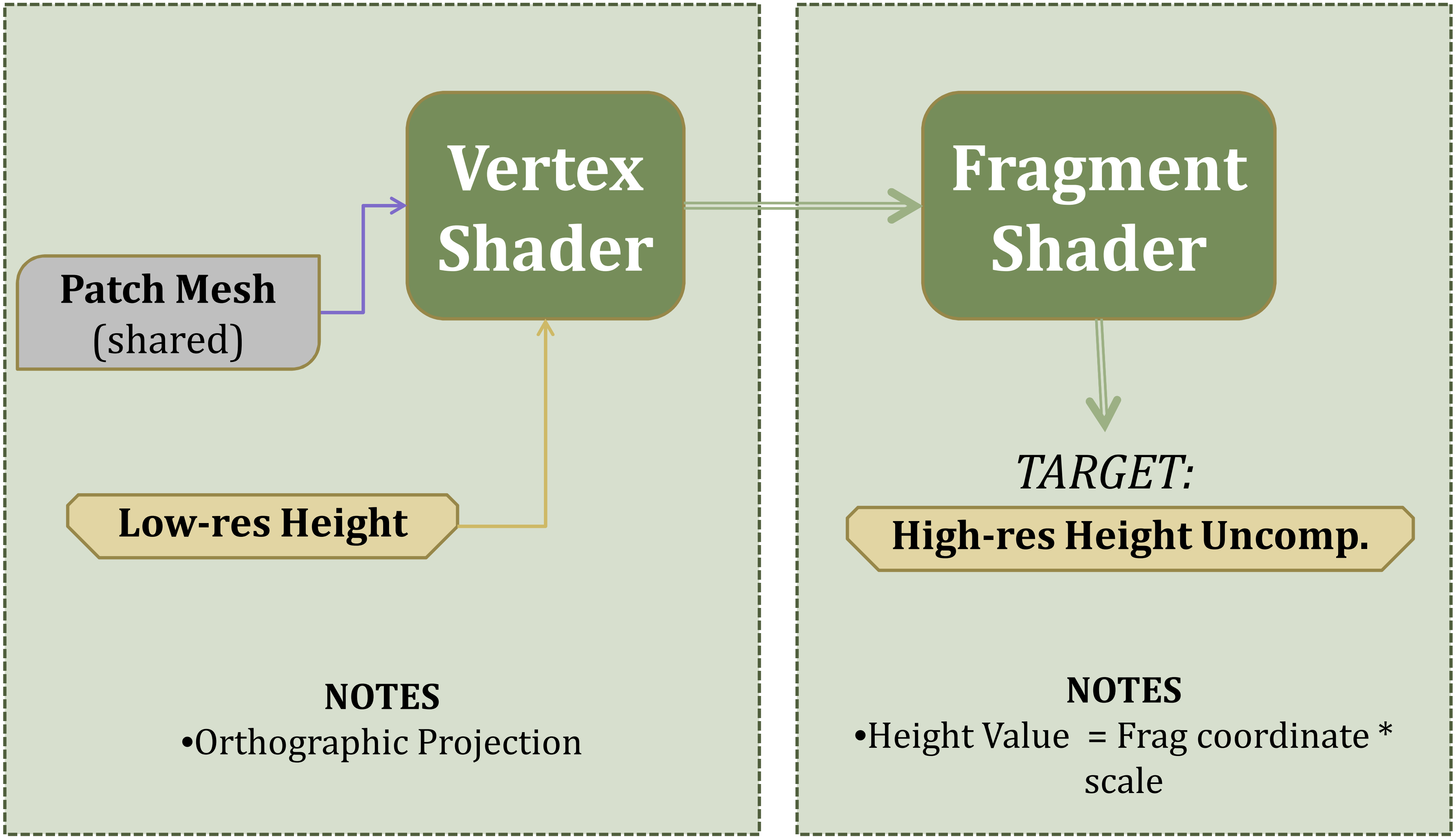
Output

- High resolution height texture
  - Ex: 2^8+1 = 256+1 = 257

Current Problem

- Sampling does not produce expected results.

Height Resolution Enhancer



NOTES

- Orthographic Projection

NOTES

- Height Value = Frag coordinate * scale

Deformation - Phase 1 - **Generate Object Height**

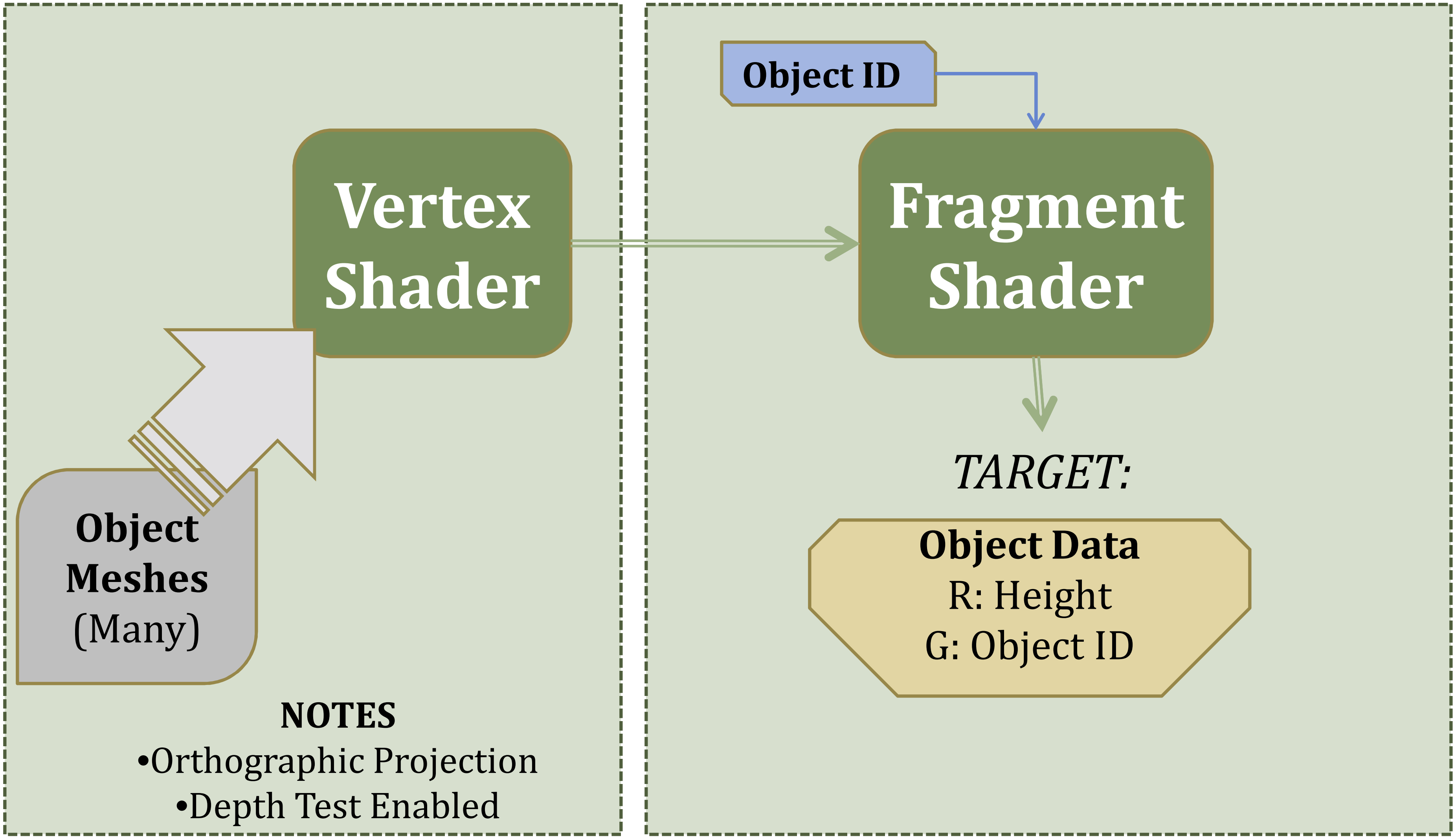
Aim

- Generate object height data to a reusable texture

Output

- Each collision patch stores the depth & ID of objects in each high-res sample point.
- The output of this phase is fed into next phase : Collision/Compression

Deformation - Phase 1 - **Generate Object Height**



Object ID

Vertex Shader

Fragment Shader

Object Meshes
(Many)

TARGET:

Object Data
R: Height
G: Object ID

NOTES
•Orthographic Projection
•Depth Test Enabled

Deformation - Phase 2 – **Collision/Compression**

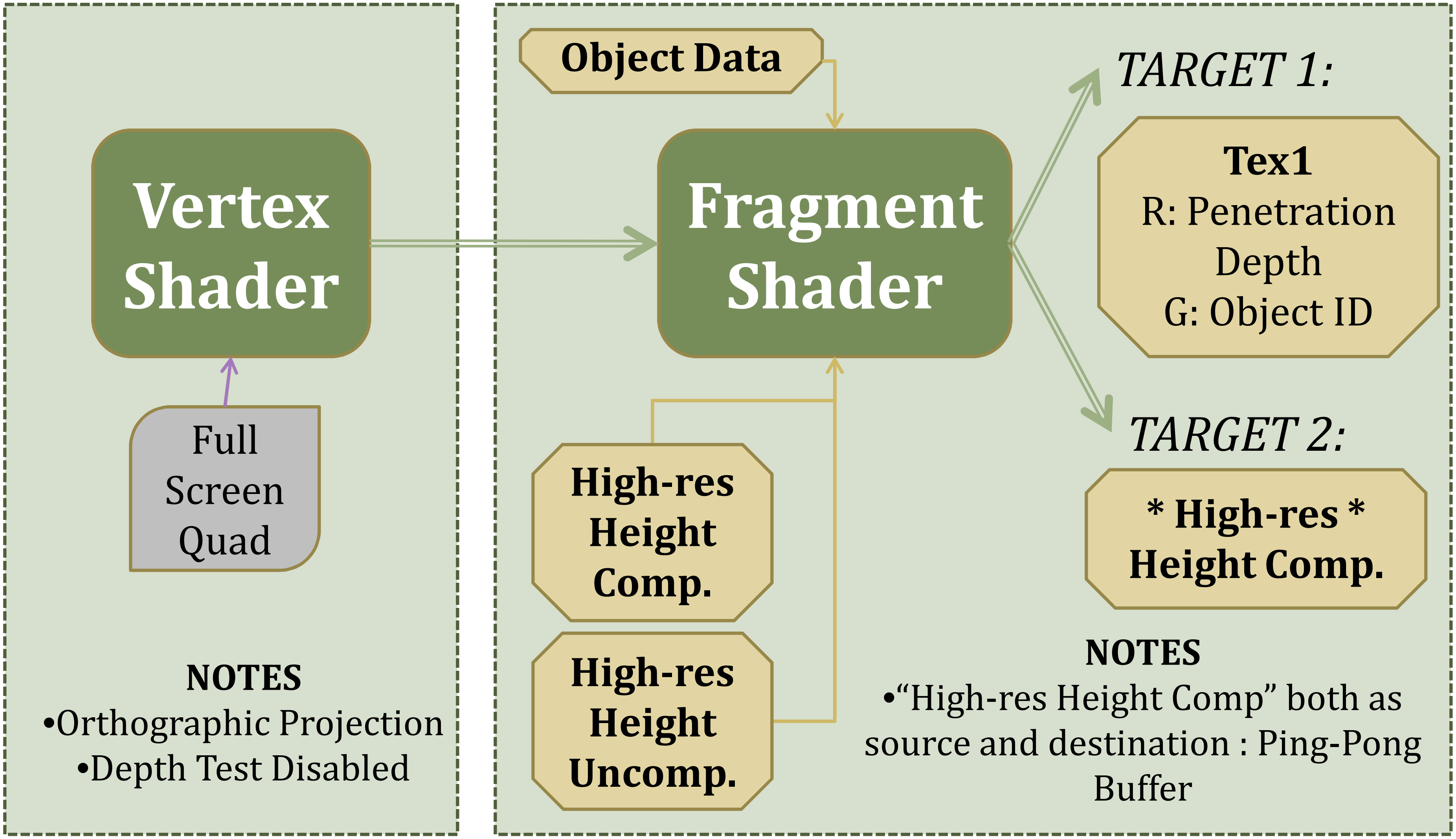
Aim

- Generate object height data to a reusable texture

Output

- Each collision patch stores the depth & ID of objects in each high-res sample point.
- The output of this phase is fed into next phase : Collision/Compression

Deformation - Phase 2 – **Collision/Compression**



Vertex Shader

Full Screen Quad

Object Data

Fragment Shader

High-res Height Comp.

High-res Height Uncomp.

TARGET 1:

Tex1
R: Penetration Depth
G: Object ID

TARGET 2:

*** High-res * Height Comp.**

NOTES

- Orthographic Projection
- Depth Test Disabled

NOTES

- “High-res Height Comp” both as source and destination : Ping-Pong Buffer

Further Details

- Height data is 16bit float in entire system.
- Penetration and ObjectID's are not permenant. They are stored in collision frame buffers (CFB).
  - CFB's can hold different patches in each iteration.

Future Work

- Automatic Decompression
  - Compressed ground slowly recovers its original state.
- Ground displacement
  - Currently, volume is not preserved.
    - Compressed ground is gone.
    - Can be suitable for some materials if compression is low (mud).
  - Distance tranformation can be used to displace compressed material: Flooding can be implemented in GPU.
    - Jump Flooding algorithm (Voronoi construction) is a candidate for this phase.
- Sampling related problems are critical and should be solved.

Future Work

- Adding “Object Weight Based” computations
  - Heavier objects will compress the ground more.
  - Based on physical interaction.
    - Current approaches only consider geometric properties of meshes.
    - Soft-body extensions ???
  - Computational model not yet done.
    - Simple approach: Store object weight (or object mass) along in object height texture
- Improvements using Multi-Sampling

Questions…

Thank you for your attention.

yalcin@cs.bilkent.edu.tr