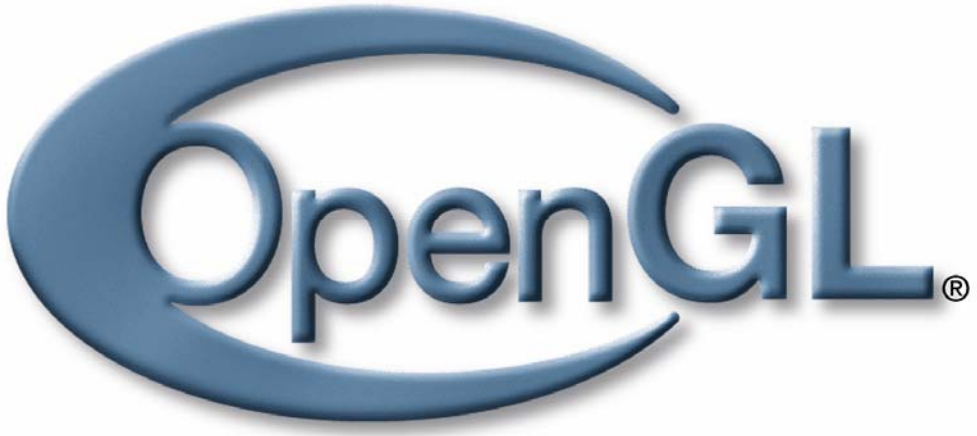


New OpenGL Features

Evan Hart
NVIDIA

© Copyright Khronos Group, 2006 - Page 1

Roadmap

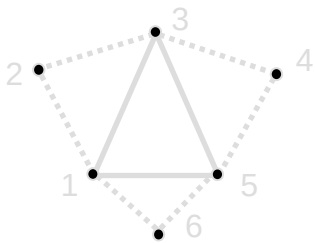
- Geometry Shaders
- Transform Feedback
- New HDR formats
- sRGB Framebuffers

© Copyright Khronos Group, 2006 - Page 2

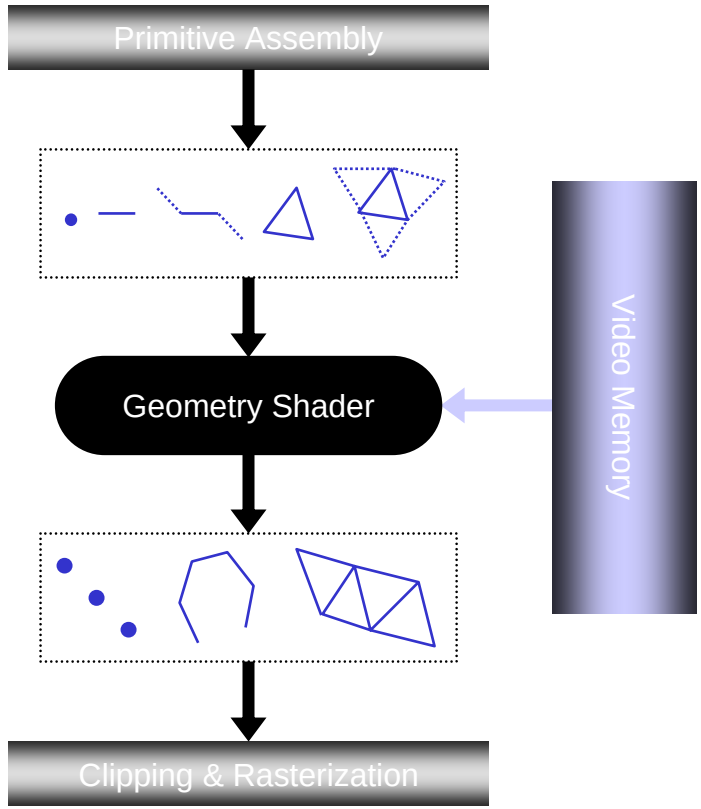
Geometry Shader Basics

- Input
  - Standard primitives
    - point, line, triangle…
  - New primitive types include neighboring vertices
    - Line with adjacency

    - Triangle with adjacency



- Output
  - Unique output type (independent from input type)
  - Points, line strips or triangle strips
  - Can output zero or more primitives
  - Generated primitive stream is in the same order as inputted



© Copyright Khronos Group, 2006 - Page 3

Geometry Shader Applications

- Better point sprites
  - Rotation, non-square, motion blur
- Simple subdivision
- Single pass cube map creation
- Automatic stencil shadow polygon generation
- Fur rendering
  - Fin generation
- Curve rendering
  - 2D rendering, hair/fur
- GPGPU
  - Data amplification – variable number of outputs

© Copyright Khronos Group, 2006 - Page 4

EXT_geometry_shader

- New link-time parameters
  - Primitive input and output type
  - Max vertices output
- New shader variables
  - gl_VerticesIn – number of input vertices
  - gl_Layer – texture array layer target
  - gl_PrimitiveID – Set primitive ID seen by the fragments
  - gl_PrimitiveIDIn – Primitive ID based on input prims

© Copyright Khronos Group, 2006 - Page 5

EXT_geometry_shader code

```
varying in vec3 eyeNormal[gl_VerticesIn];

varying out vec3 oEyeNormal;

for ( int i = 0; i < gl_VerticesIn; i++) {
    oEyeNormal = eyeNormal[i];
    //causes all output varying to submit
    EmitVertex();
}

//Start a new strip
RestartPrimitive();
```

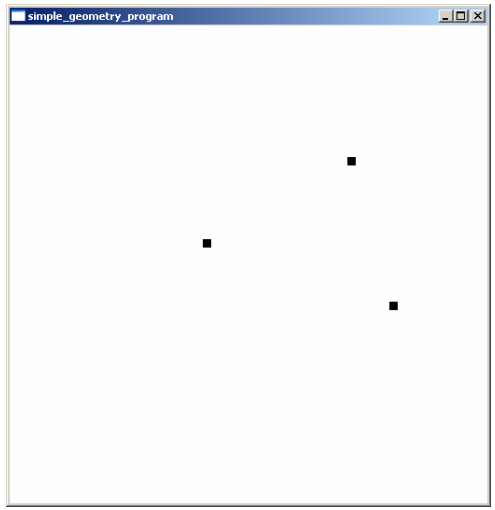
© Copyright Khronos Group, 2006 - Page 6

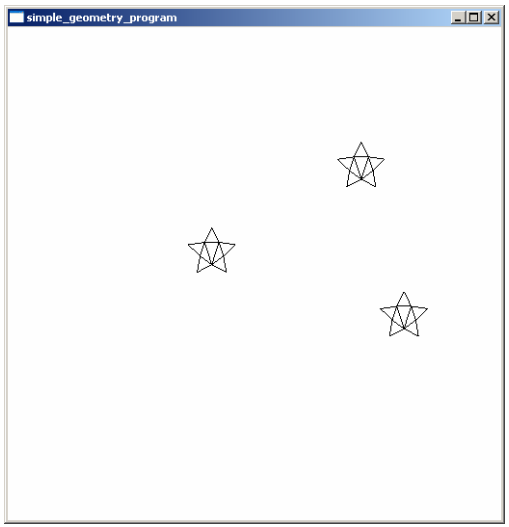
Example: Point Sprites on Steroids

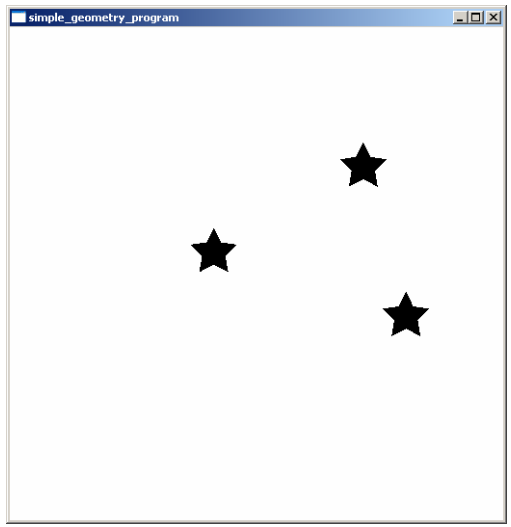
- ARB_point_sprite
  - Limited utility
  - Simple coordinate expansion
  - Always square
  - Always screen aligned
- Geometry shader point sprites
  - Explicit expansion in the geometry shader
  - Any size / orientation
  - Any shape
  - Arbitrary coordinate mapping
  - Extra cost

© Copyright Khronos Group, 2006 - Page 7

Example: Point Sprites







© Copyright Khronos Group, 2006 - Page 8

Geometry Shader Dangers

- Floating point math is inacurate
- Geometric correctness relies on repeatable computations
- Pre-geometry shaders
  - Single vertex program executed independently on each vertex
  - Identical operations on each vertex ensures correctness
    - Index 0 is index 0 no matter what
- Post-geometry shaders
  - Developer needs to take care
  - Edges are shared between primitives, but traversal is different
  - Face normal will differ slightly based on which vertex the computation starts with
  - Problems are tricky to solve and easy to overlook
- Strategies
  - Preprocess data to ensure consistency
  - Recast algorithm to remove duplicate computations

© Copyright Khronos Group, 2006 - Page 9

EXT_transform_feedback

- Allows storing output from a vertex program or geometry program to buffer object
- Enables multi-pass operations on geometry, e.g.
  - Store results of animation (skinning) to buffer, reuse for multiple lights
  - Recursive subdivision
- Provides queries for number of primitives generated by geometry program
- Not standardized yet
  - Work in progress

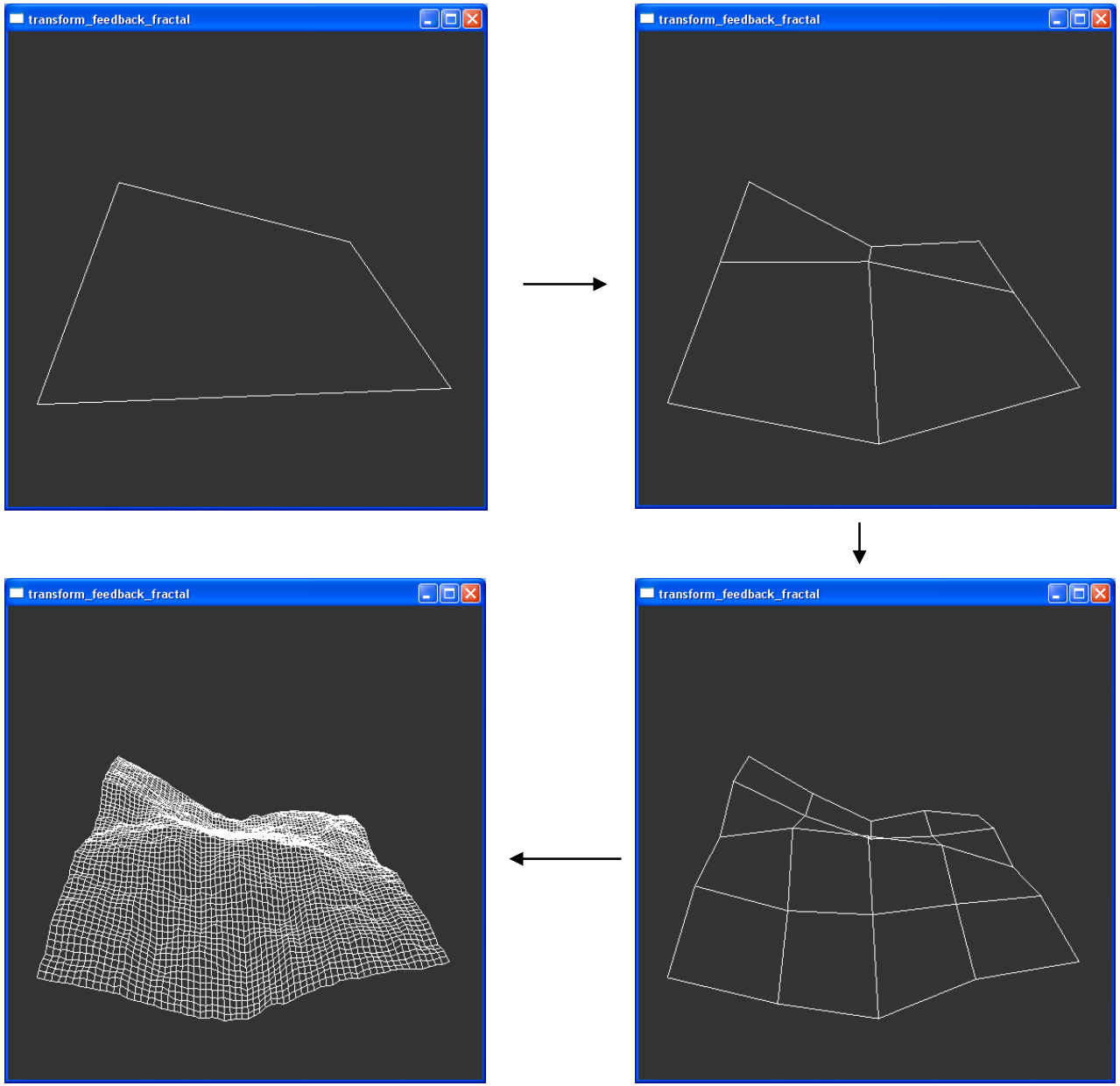
© Copyright Khronos Group, 2006 - Page 10

Example: Terrain Subdivision

- Takes quad as input (actually line_adj primitive)
- Geometry program subdivides into 4 new quads using diamond-square subdivision
- Uses two VBOs, reads from one, writes to the other using transform feedback
- Then swap

© Copyright Khronos Group, 2006 - Page 11

Terrain Subdivision



© Copyright Khronos Group, 2006 - Page 12

EXT_texture_buffer_object

- Bind buffer object as a texture
- Jumbo 1D texture
  - Larger size limit (128 Mtexels)
- Does not support filtering
- Addressed by element
  - Not normalized [0-1]
- Limited format support
  - No RGB support (RGBA is OK)
  - Minimum of 1 byte per component
  - No compressed formats

© Copyright Khronos Group, 2006 - Page 13

Texture buffer usage

- Large constant store
  - Significantly larger than EXT_bindable_uniform
  - Jumbo bone list for skinning
- Instancing data
  - Transform matrices
  - Materials
- Custom indexing
  - Separate ‘index’ for position, normal, texture coordinate
  - Less efficient than normal vertex fetching
    - Not as coherent
  - Can be useful interactive editing due to reduced sw cost

© Copyright Khronos Group, 2006 - Page 14

HDR Texture Format Extensions

- EXT_packed_float
  - Space-efficient float format
  - Relatively low precision
    - More than good enough most times
- EXT_texture_shared_exponent
  - Space-efficient float format
  - Variable accuracy
    - Can be more or less accurate than packed float
    - Also more than good enough

© Copyright Khronos Group, 2006 - Page 15

Packed Float Textures

- EXT_packed_float
  - 11/11/10 floating point format
  - 5 bit exponent per component
    - Bias of -15
  - Only supports positive values (no sign bit)
  - Can be used as framebuffer format
  - Max values
    - R/G – 65024
    - B – 64512
  - Size advantage can make it much faster than float16

© Copyright Khronos Group, 2006 - Page 16

RGBE / Shared Exponent textures

- EXT_texture_shared_exponent
  - 9/9/9/5 RGBE format
  - Similar to Radiance 8/8/8/8 RGBE format
  - Shared 5 bit exponent (bias of -15)
  - Source texture format only (not renderable to)
  - Only supports positive values (no sign bit)

© Copyright Khronos Group, 2006 - Page 17

EXT_framebuffer_sRGB

- Allows framebuffer to be stored in sRGB space
  - Provides perceptual color compression
    - 8 bits sRGB is similar to 10 bits linear RGB
  - Converts to/from sRGB on access to framebuffer
  - Implements a gamma of 2.2
    - This matches most monitors relatively well
- Controlled via a simple enable
  - Not necessarily available on all formats

© Copyright Khronos Group, 2006 - Page 18

Thanks

- Simon Green
- Samuel Gateau
- NVIDIA DevTech team
- Barthold Lichtenbelt

© Copyright Khronos Group, 2006 - Page 19