

Paving Procedural Roads with Pixel Shaders

Jörn Loviscach
Hochschule Bremen
Bremen, Germany

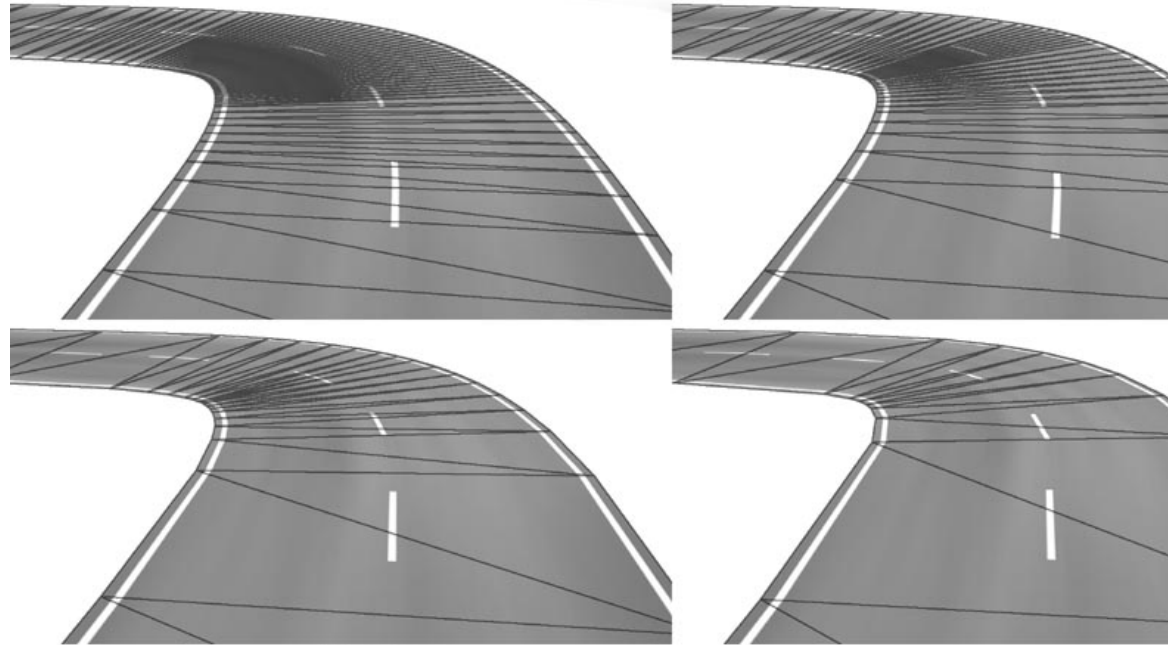
Outline

- Polygonal roads vs. procedural roads
- Implicit fat curves
- Cookie-cutting
- Performance, strengths, weaknesses
- Conclusion, outlook

Polygonal Roads vs. Procedural Roads

Polygonal Roads vs. Procedural Roads

- VR driver training, 3D games:
lots of roads
- Polygons tend to show in boundaries
and textures of roads
- Smooth-looking
polygonal roads
demand
level-of-detail
management



Polygonal Roads vs. Procedural Roads

Procedurally-built roads

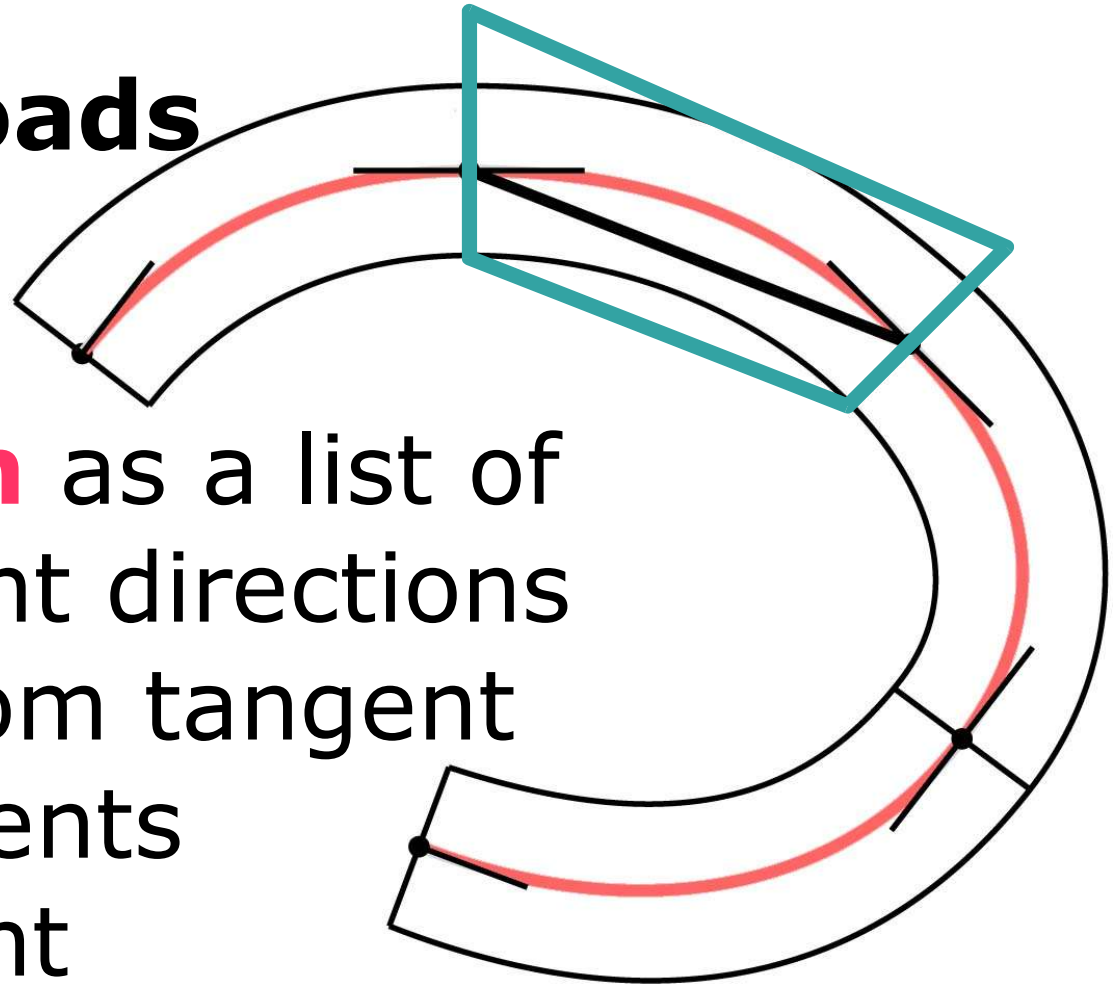
- are by themselves pixel-precise
- minimize data storage and transfer
(high-level geometry; no LoD datasets)

Do that completely on the GPU!

Polygonal Roads vs. Procedural Roads

Process:

- Read the **median** as a list of points and tangent directions
- Form the road from tangent continuous segments
- For every segment construct an oversized **quadrangle**
- Discard off-road pixels in pixel shader



Demo ●

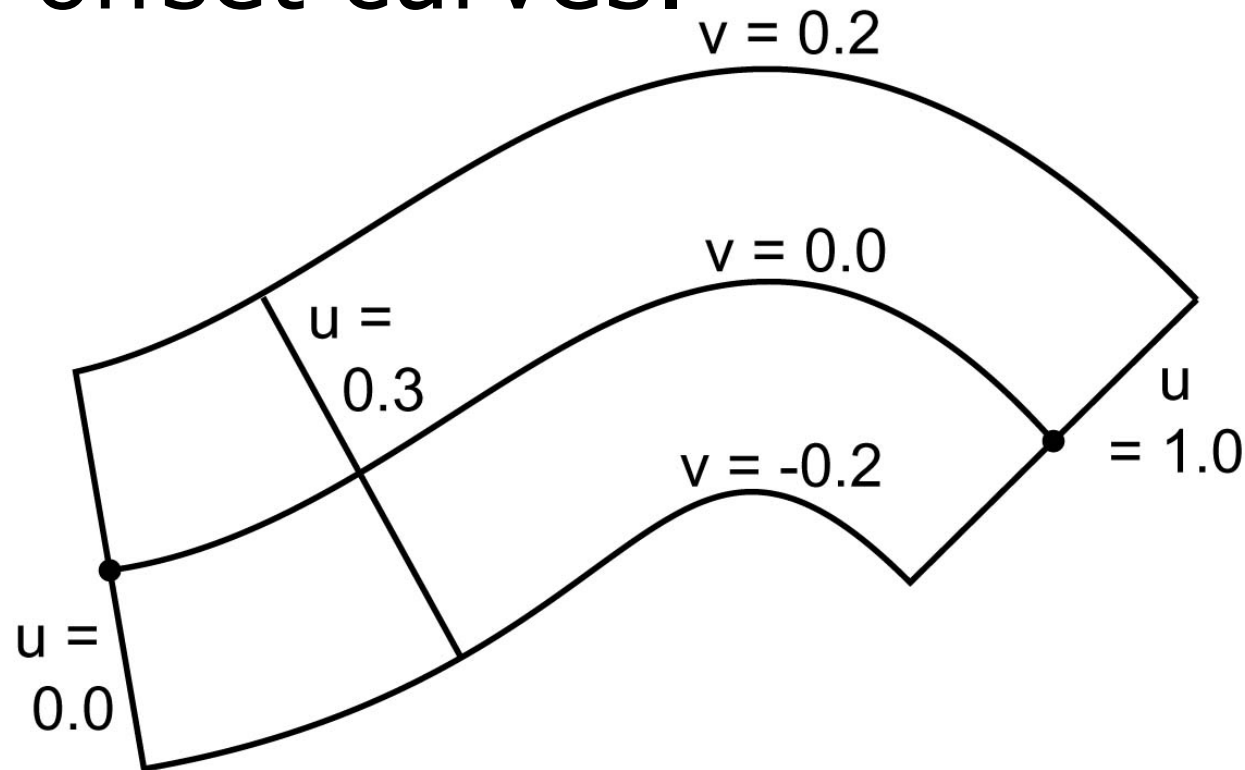
Implicit Fat Curves

Implicit Fat Curves

How to create a segment of the road?

- Model the median as a cubic function.
- Create the full width of the road analogously to offset curves.

Curve is *not* computed explicitly:
Given (x, y)
find (u, v) .



Implicit Fat Curves

Exact solution not feasible.

Approximate solution (see paper):

- only solve a quadratic equation
- boundary conditions (positions, tangents) precisely met

Demo ●

Cookie-Cutting

Cookie-Cutting

Initialization:

- Build oversized quadrangles (see demo)
- Equip all four vertices of a single quadrangle with 2D coordinates etc.

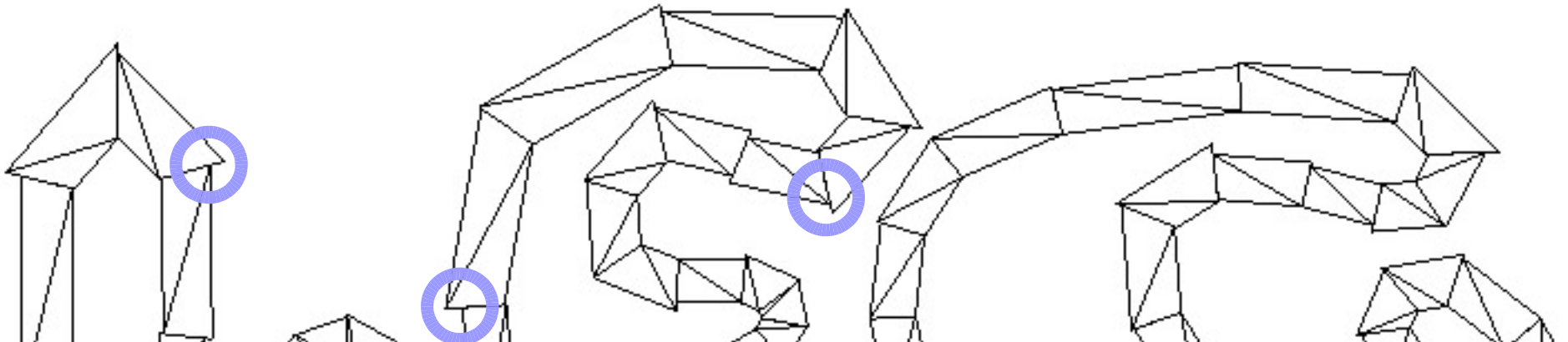
Rendering:

- Evaluate the implicit fat curve in the pixel shader
- Discard off-road pixels
(HLSL: `clip`, DX Assembler: `texkill`)
- Retrieve texture color

Cookie-Cutting

T-junctions:

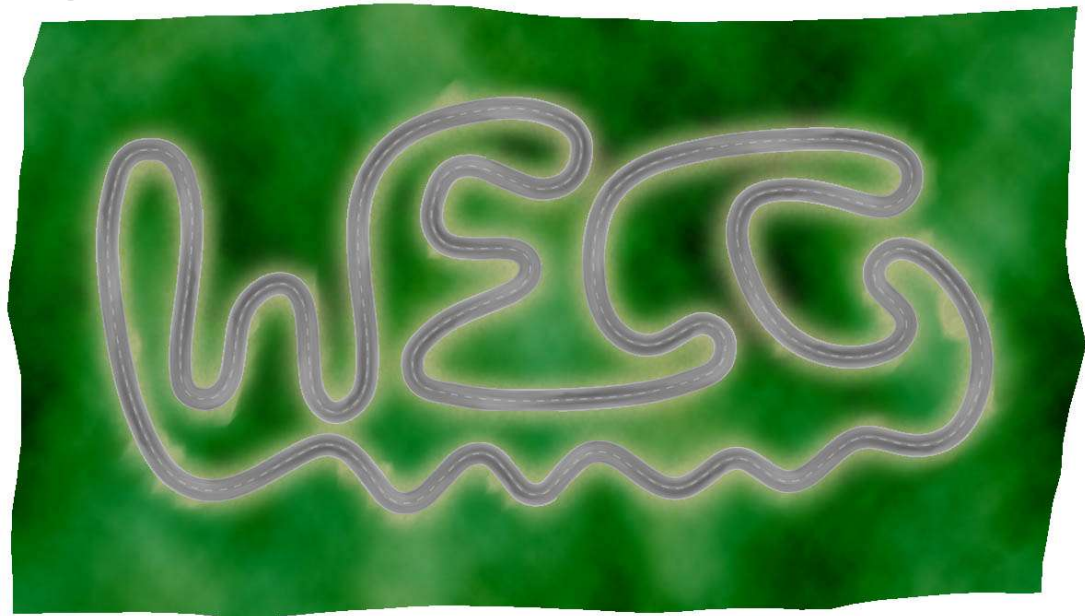
- No missing pixels with Nvidia hardware; elsewhere use tiny overlap
- Cannot be abandoned easily: Vertices have to be equipped with values of local coordinates
- Help to minimize rendered area



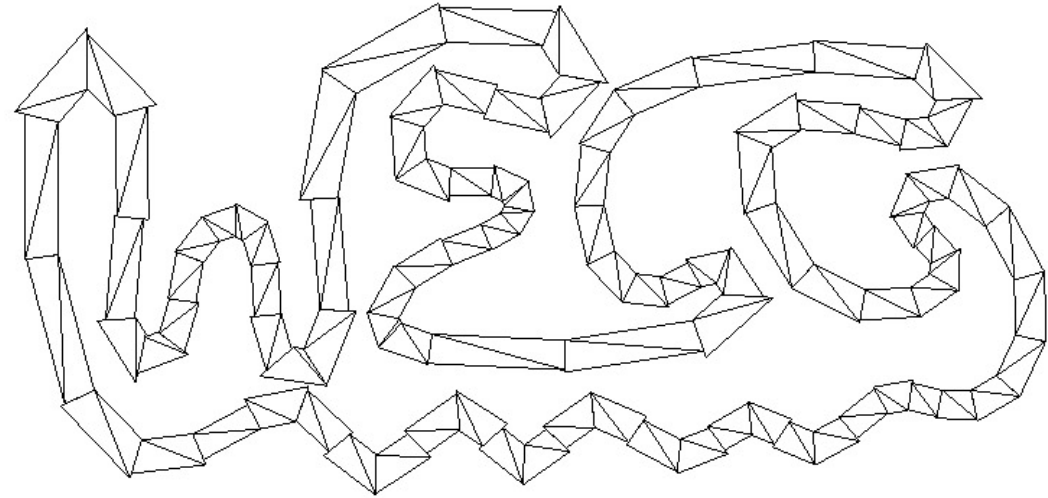
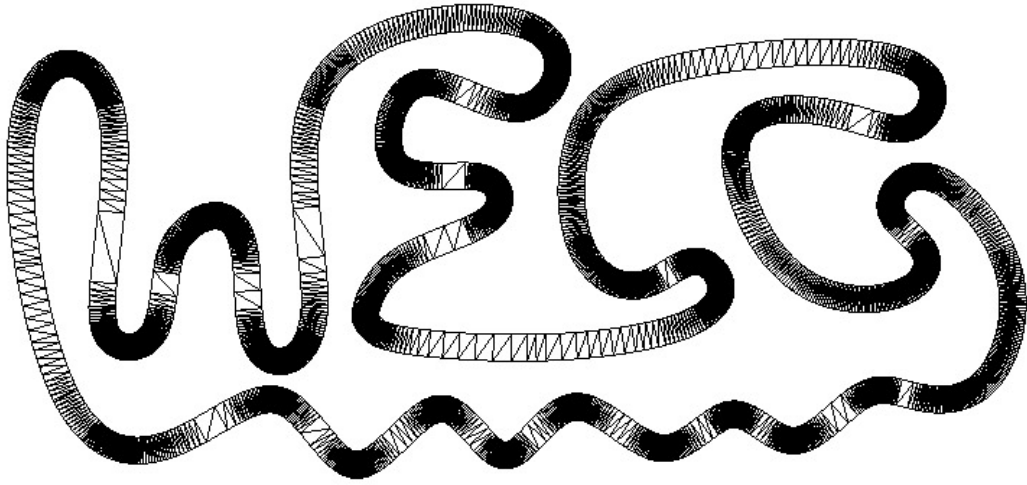
Performance, Strengths, Weaknesses

Performance, Strengths, Weaknesses

- small scene of ~ 100 segments:
three times slower than high-resolution
polygonal geometry (~ 5000 tris)
- break-even with hi-res
polys at $\sim 350k$ tris
- break-even with
mid-res polys
at $\sim 2M$ tris



Performance, Strengths, Weaknesses



Strengths:

- Pixel-scale precision without LoD
- Amount of data strongly reduced

Weaknesses:

- Roads over hills and through valleys?
- Curbs? Intersections?

Conclusion, Outlook

Conclusion, Outlook

- Procedural roads can be rendered efficiently by the GPU
- General road layout (intersections etc.): combine procedural geometry and polygons (fully specified behavior at boundary!)
- Generalize to handle some amount of curvature in height, too?
- Similar method to render tubes using billboards?

Questions?