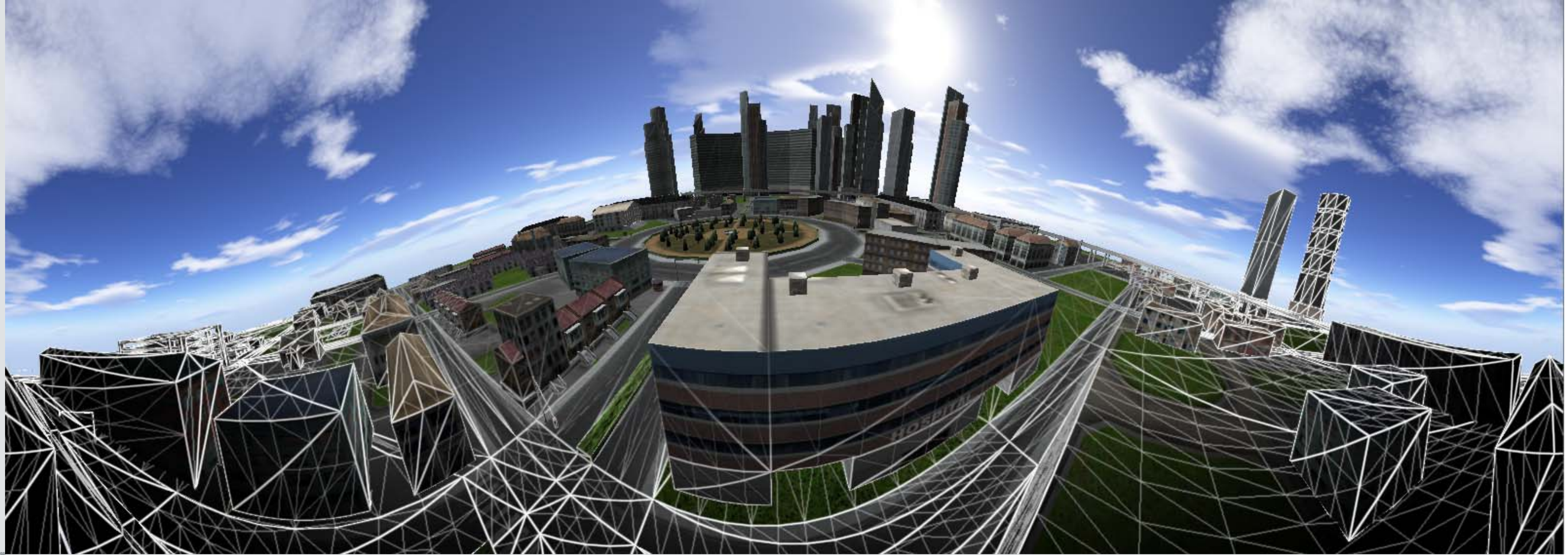


Dynamic Mesh Refinement on GPU using Geometry Shaders

Haik Lorenz and Jürgen Döllner
Hasso-Plattner-Institute
University of Potsdam, Germany

Outline

1. **Introduction**
2. Dynamic Mesh Refinement
3. Results
4. Conclusions

Introduction – About me

Research assistant at Hasso-Plattner-Institute, University of Potsdam, Germany

Member of the research group "3D Geoinformation"

Focus subject: **Textured virtual 3D city models**

- **Creation**: Automatic texturing using oblique aerial imagery
- **Storage**: Open Geospatial Consortium (OGC) standard "CityGML" for rich city models
- **Visualization**: Effective presentation of large-scale virtual 3D city models

Introduction – Motivation

- Enable non-planar projections & deformations in object space for arbitrary objects
  - Provide better quality than screen space operations
  - Require sufficient vertex density in screen space
- Basic operation: (View dependent) mesh refinement

Closeup of a screen-space cylindrical projection

[Trapp & Döllner 2008: "A Generalization Approach for 3D Viewing Deformations of Single-Center Projections"

Closeup of an object-space cylindrical projection

Introduction – Examples

360° cylindrical projection in object space

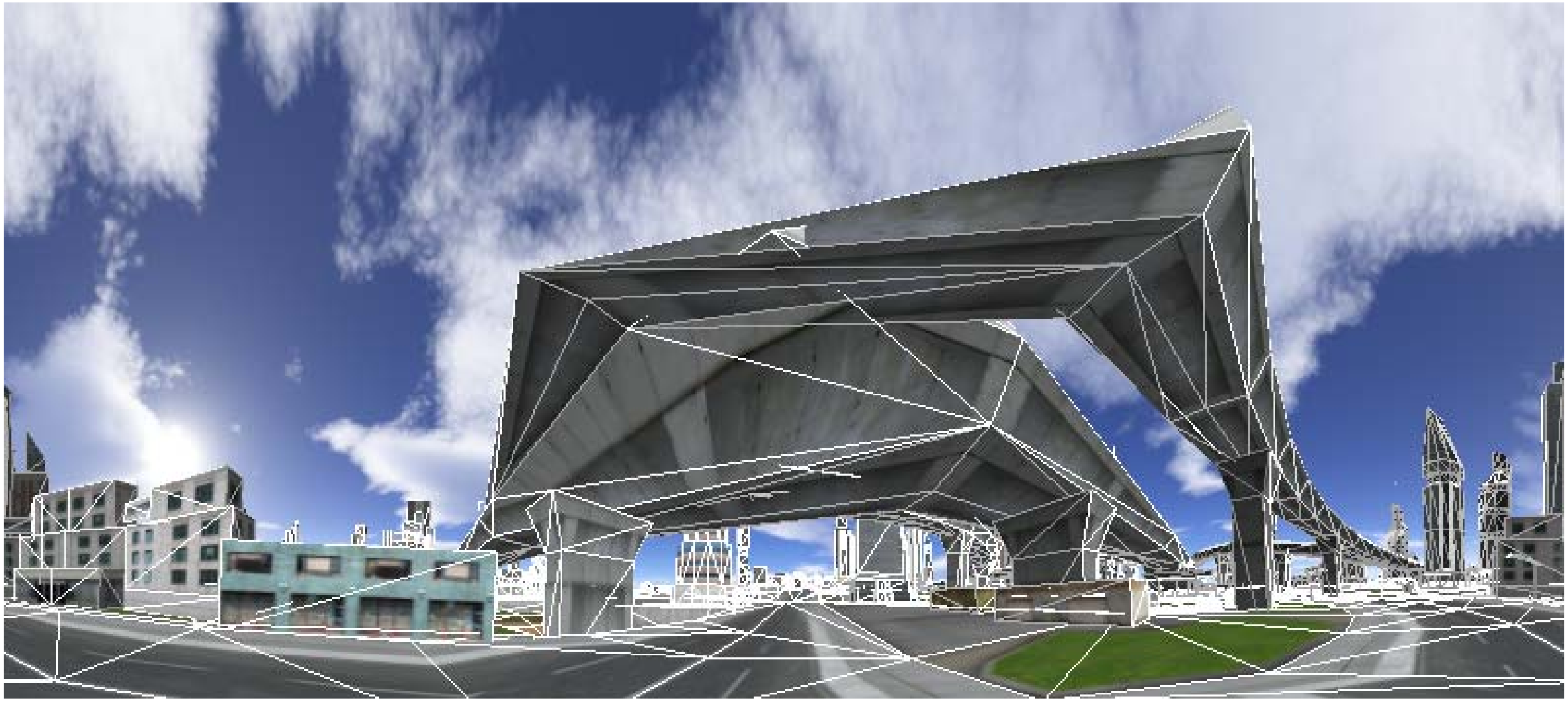
Introduction – Examples

360° cylindrical projection in object space

Original mesh – 34596 triangles

Introduction – Examples

360° cylindrical projection in object space



Original mesh – 34596 triangles

Introduction – Examples

360° cylindrical projection in object space

View-dependently refined mesh – 39721 triangles

Introduction – Examples

360° cylindrical projection in object space

View-dependently refined mesh – 39721 triangles

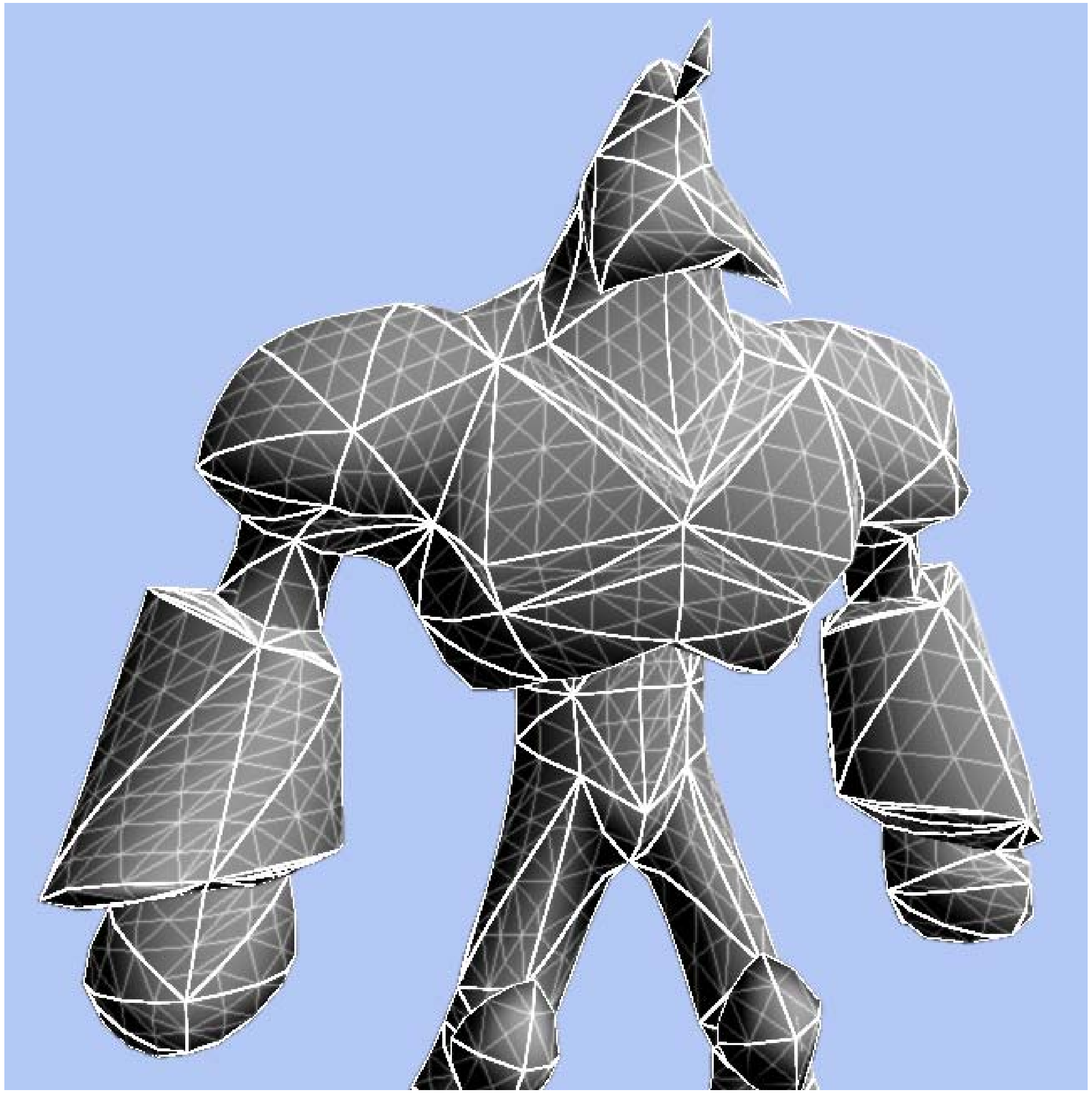
Introduction – Examples

360° cylindrical projection in object space

Comparison between original and refined mesh (34596 vs. 39721 triangles)

Introduction – Examples

Geometry synthesis using PN-triangles



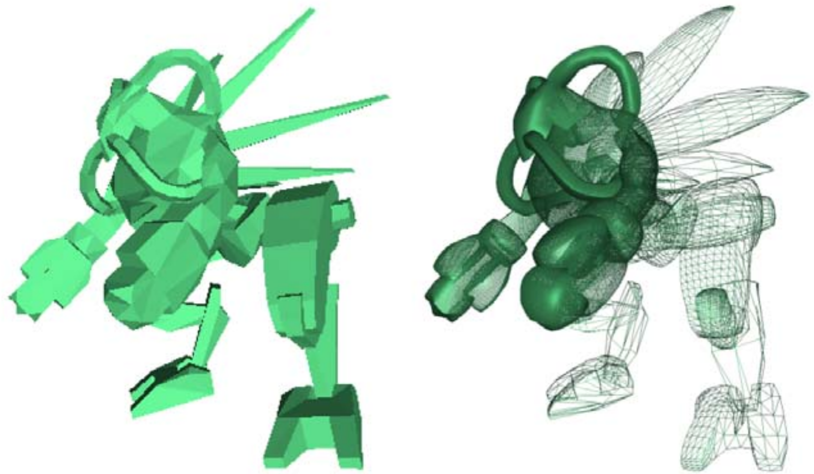
Introduction – Existing techniques

Adaptive tessellation of subdivision surfaces [Bunnell '05]

- For subdivision surfaces only

Adaptive Mesh Refinement [Boubekeur and Schlick '07]

- Barycentric refinement patterns replace input triangles
- Low technical requirements (VS1)
- One shader setup + render call per input triangle
- Focusing on geometry synthesis with large refinement ratio



(a) CPU (b) GPU

Outline

1. Introduction
2. **Dynamic Mesh Refinement**
3. Results
4. Conclusions

Dynamic Mesh Refinement – Overview

Input: Triangles without specific connectivity or topology

Design goals

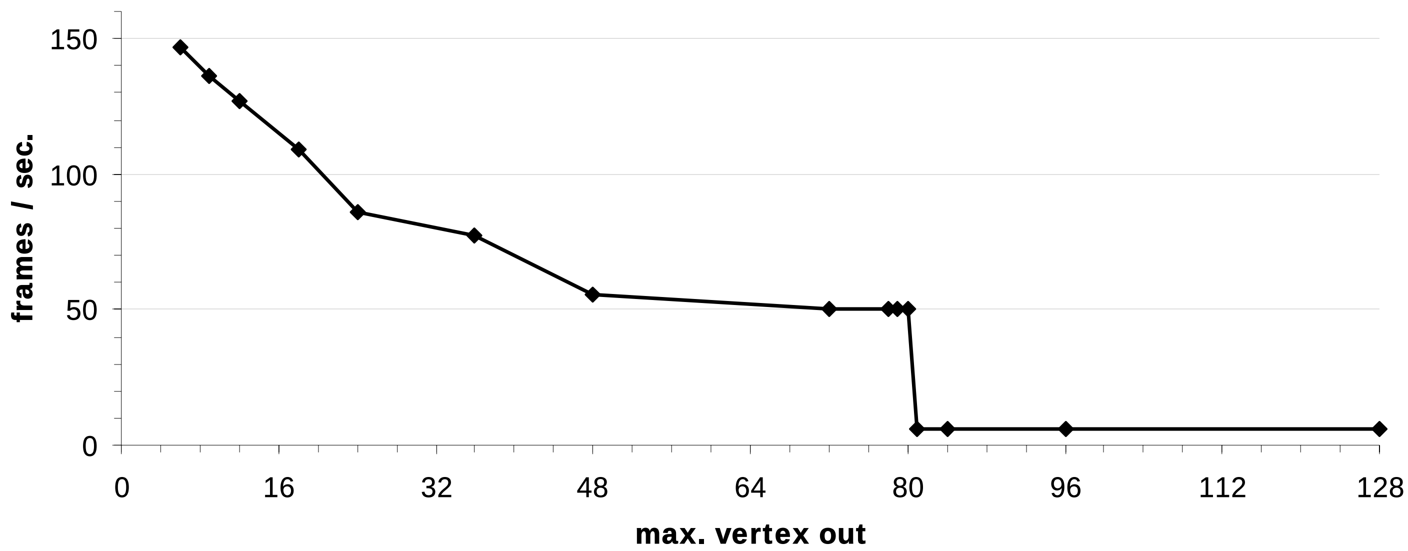
1. View-dependency
   - Overall low refinement ratio, but “unlimited” peak ref.
   - Per-frame refinement changes
2. Substantial input triangle count
   - Minimize communication between CPU and GPU

Approach

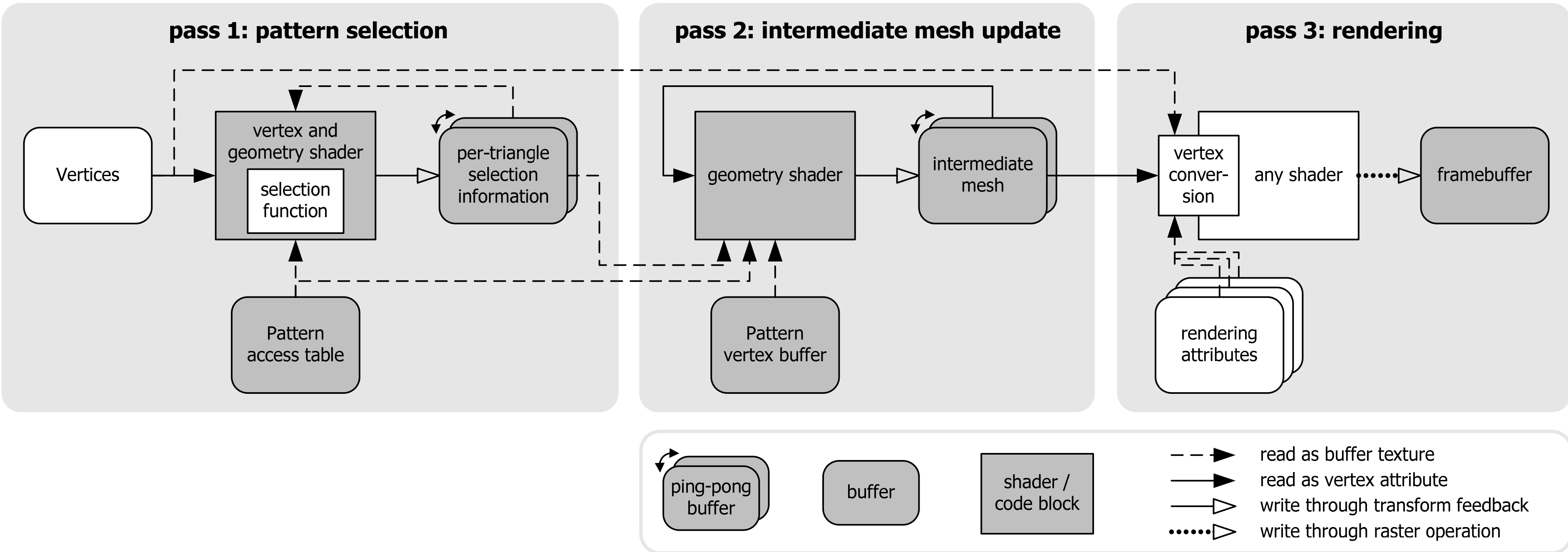
- Use barycentric refinement patterns
- Use geometry shaders for “pattern instantiation” on GPU
  - Implies storage of an intermediate refined mesh

Dynamic Mesh Refinement – Geometry Shaders

- Geometry shaders can change, drop, or amplify primitives between vertex processing and before rendering
- Ideal for mesh refinement: Simply tessellate triangles on GPU on the fly...
- BUT: Output limited and extremely output sensitive



Dynamic Mesh Refinement – Sketch

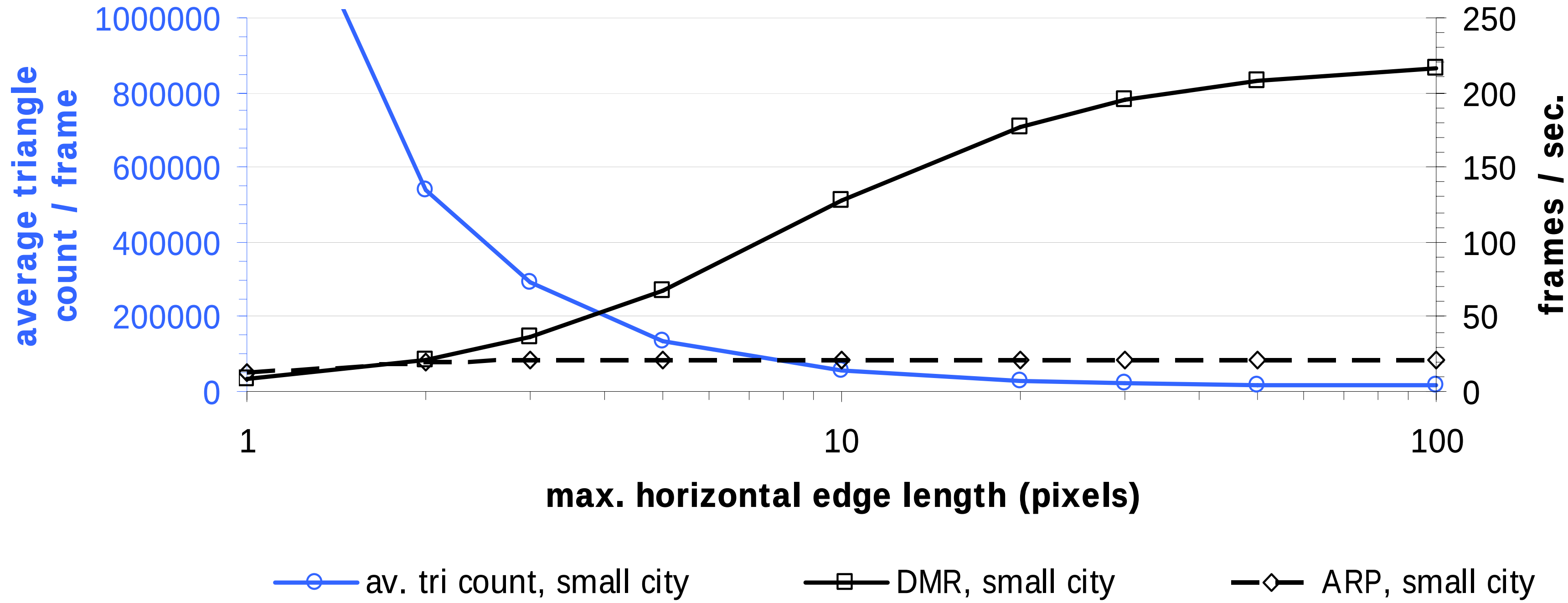


- **Key idea**: Incrementally update intermediate mesh

Dynamic Mesh Refinement – Incremental update

- Replaces repeated refinement passes
  - One update pass per frame
- Intermediate mesh concatenates pattern copies
- For each input triangle:
  - Goal: Overwrite existing pattern copy with new one
  - Each existing subtriangle is replaced by a few subtriangles of the new refinement pattern
  - Excess subtriangles are dropped
- Allows for arbitrary pattern size, but limits per-frame pattern growth

Dynamic Mesh Refinement – Comparison



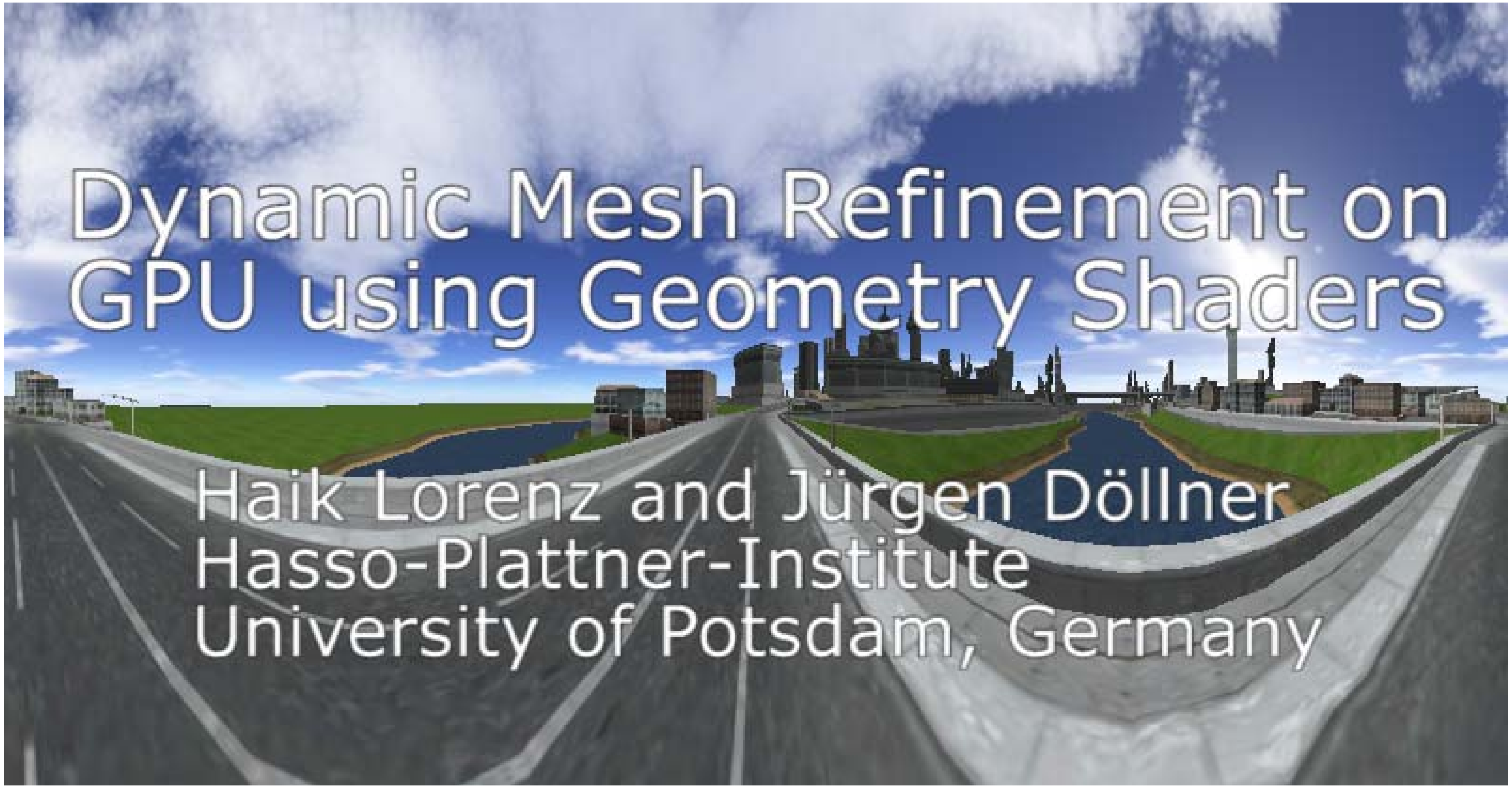
Comparison to [Boubekeur & Schlick ‘07] for cylindrical projection (~13500 triangles in mesh, ~1050 fps w/o refinement)

Suitable for view-dependent refinement and large number of triangles

Outline

1. Introduction
2. Dynamic Mesh Refinement
3. **Results**
4. Conclusions

Results - Movie



Outline

1. Introduction
2. Dynamic Mesh Refinement
3. Results
4. **Conclusions**

Conclusions

Generic technique for dynamic mesh refinement

Excellent for view-dependent rendering techniques

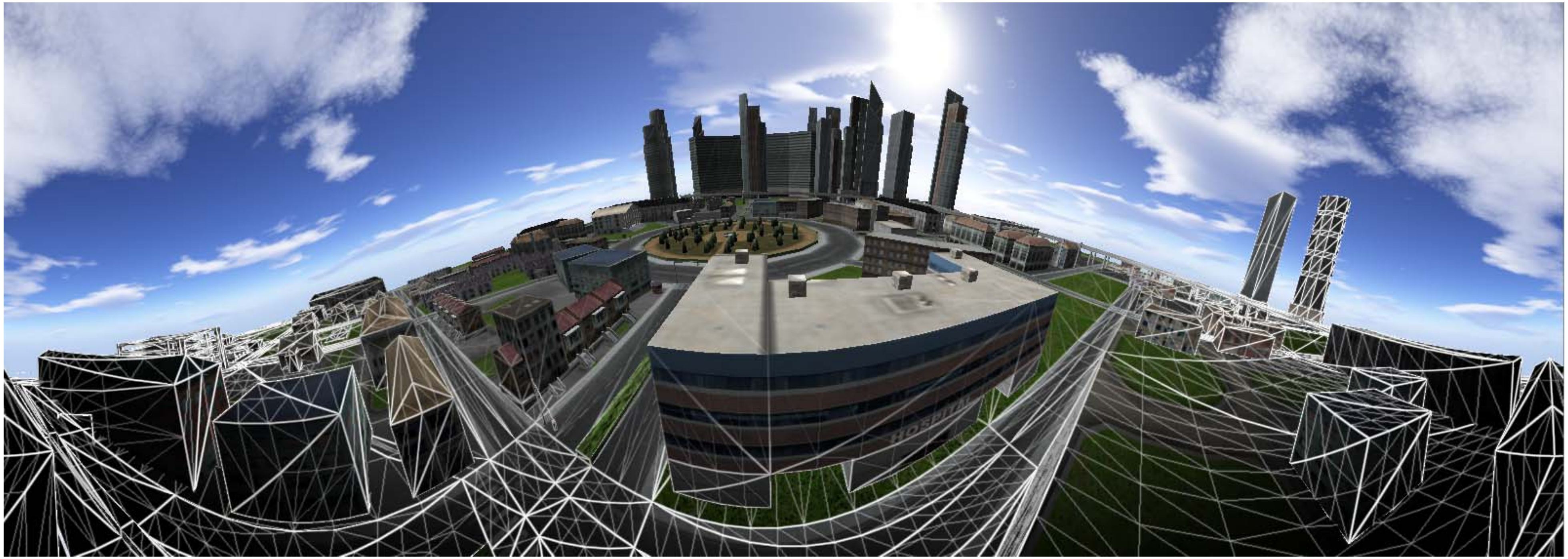
Runs on the GPU exclusively

Applicable to any mesh

Future work:

- Applications for mesh refinement:
  - General non-planar projections in object space
  - Per-vertex global illumination with guaranteed vertex density in image space
- Unified approach to mesh refinement and simplification

The End



Thank you for your attention!

Contact:

haik.lorenz@hpi.uni-potsdam.de

Research group “3D Geoinformation”: www.3dgi.de

supported by:

Bundesministerium für Bildung und Forschung