



# Wuppertal 3D

Vorstellung: Hintergrund und Projektidee

## Zur Person



- **2005 - 2010**
  - Studium Geodäsie und Geoinformation in Bonn
  - BSc, MSc Thesis: Laserscan Punkwolken (Prof. Plümer)
- **2010 – 2014**
  - Senior Software Engineer CPA Systems/CPA ReDev (Siegburg)
    - 3D Stadtmodelle (SGJ3D, CityGML) und (militärische) 3D Simulation
    - Datenbanken, OGC Konformität, Modellierung
- **Seit 2014**
  - Fachreferent und Teamleiter bei der Stadt Wuppertal
    - „Inhouse Consultant“ für Geodaten
    - Weiterentwicklung Wuppertaler GDI (WuNDa)
    - Teamleitung Stadtkartographie (Stadtplanwerk, Projektgeschäft)
    - Projektarbeit: Wuppertal 3D

# Inhalt

- **Hintergrund**
  - Derzeit kein produktives 3D-Stadtmodell im Einsatz
  - Wuppertal 3D ist in der Konzeptionsphase
- **Hier und heute: Vorstellung der Ideen zu Wuppertal 3D**
  - Rahmenbedingungen
  - Priorisierung
  - Aktueller Stand
  - Zeitplan

# Hintergrund Wuppertal 3D

- **Projekt Wuppertal 3D**
  - Personelle, finanzielle Ressourcen für Wuppertal 3D äußerst gering
  - Jede Art von Fortführung von 3D-Daten erzeugt personellen, finanziellen Aufwand (MitarbeiterInnen, Software, Wartung, etc.) => nicht realisierbar
  - **Idee Minimalstrategie: Wuppertal 3D 1.0 (>80% Eigenentwicklung)**
    - reine Viewerlösung vorhandener 3D Daten
      - DGM aus Bruchkanten und Punktraster selber erzeugen (mit vorhandener Software)
      - Gebäudemodelle (vom Land) visualisieren
      - DGM Texturierung mit Luftbild
    - Integration in Wuppertaler GDI (kein *Sekundärsystem* beschaffen)
    - Minimale Benutzerinteraktion
      - Keine Selektion, kein Export, kein Import
    - Keine Fortführungslogik!
      - Daten so wie sie sind verwenden
    - Keine CityGML Datenhaltung
      - Einmalige Aufbereitungsprozess: CityGML (Gebäude vom Land) => Visualisierungsformat

# Anforderungen Wuppertal 3D

## • 3D-Viewer

- a.) Dynamische perspektivische Betrachtung von kommunalen Geodaten
  - Freie Kamera
  - Freie Positionierung
- b.) Skalierbarkeit
  - vom Flug durch die Wuppertaler Talachse
  - bis zur Betrachtung einzelner Objekte
- c.) Integration in WuNDa (Wuppertaler GDI)
  - „3D-Fenster“ in WuNDa
  - Sekundärsysteme nach Möglichkeit vermeiden
- d.) Sonstiges:
  - performant, MultiUser fähig, installationsfrei, wartungsarm etc.

## • 3D Editor

- Da Fortführung personell , finanziell nicht realisierbar => nicht benötigt

## • 3D Datenhaltung (CityGML)

- Da weder Fortführung- , Import- und Export- Möglichkeiten realisierbar sind, wird **keine Datenbanklösung** benötigt
- Stattdessen: Ausschließlich dateibasiertes Viewing

# Priorisierung Wuppertal 3D

- **Prio 1: Statischer Viewer**

- **Schwerpunkt Geländedarstellung: Raster, TIN, Bruchkanten, Umgebung**
- Einfache Texturierung: Luftbild 2014 (Nicht konfigurierbar!)
- Einfache Objektgeometrie: Visualisierung Gebäudemodelle des Landes
- **Ergebnis:** Wuppertal 3D 1.0

- **Prio 2: Dynamisierung durch Konfigurierbarkeit**

- Integration von Umweltdaten (Laufende Forschungsprojekte)
- Integration von Laserscanpunktwolken
- Konfigurierbarkeit der DGM Texturierung

- **Prio 3: Langfristige Erweiterungen**

- Schnittstelle Liegenschaftskataster
  - Gebäudemodelle aus Liegenschaftskataster (nur wenn zu 100% automatisierbar!)
- Weitere Daten und Projekte

# Zeitplan Wuppertal 3D

## • Prio 1: Statischer Viewer

- **Schwerpunkt Geländedarstellung: Raster, TIN, Bruchkanten, Umgebung**
- Einfache Texturierung: Luftbild 2014 (Nicht konfigurierbar!)
- Einfache Objektgeometrie: Visualisierung Gebäudemodelle des Landes
- **Ergebnis:** Wuppertal 3D 1.0

## • Prio 2: Dynamisierung durch Konfigurierbarkeit

- Integration von Umweltdaten (Laufende Forschungsprojekte)
- Integration von Laserscanpunktewolken
- Konfigurierbarkeit der DGM Texturierung

## • Prio 3: Langfristige Erweiterungen

- Schnittstelle Liegenschaftskataster
  - Gebäudemodelle aus Liegenschaftskataster (nur wenn zu 100% automatisierbar!)
- Weitere Daten und Projekte

Vergangenheit

2014

2015

2016

2017/18

Zukunft

# Rückblick 2014 und Aktuelle Arbeiten

- **Forschung und Entwicklung (in Arbeit!)**

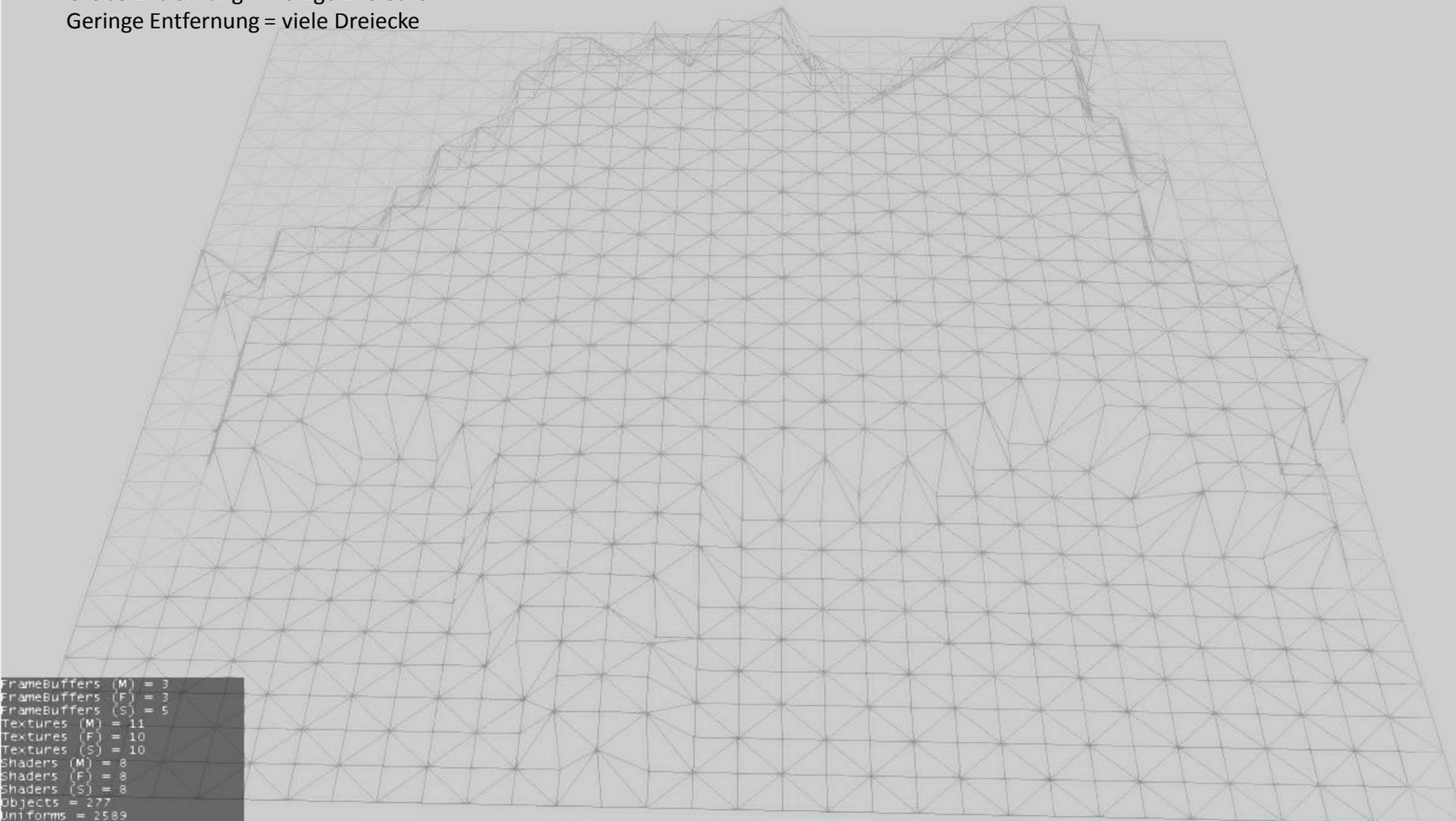
*Basis aller 3D Visualisierungen ist eine Geländedarstellung (texturiertes DGM). Dazu wurde/wird ein eigener Viewer konzeptionell entwickelt um anschließend in die Wuppertaler GDI integriert zu werden.*

- Verwendung der JMonkey 3D-Engine
  - OpenSource, gut dokumentiert, JAVA-basiert
- Datenaufbereitung Geländemodell
  - ArcGIS 5m Punktraster + Bruchkanten => GML TIN => Eigener Importer => JMonkeyEngine
- Prototypische Implementierungen und Untersuchungen
  - Dynamische Generalisierung (GeoMipMapping)
  - Integration eines statischen „Umgebungs DGM“
  - Erste Tests zur Darstellung der Gebäudemodelle (Land)

Experimentelle Visualisierung des Wuppertaler DGM (Punkte-Kanten-Darstellung)

Große Entfernung = wenige Dreiecke

Geringe Entfernung = viele Dreiecke



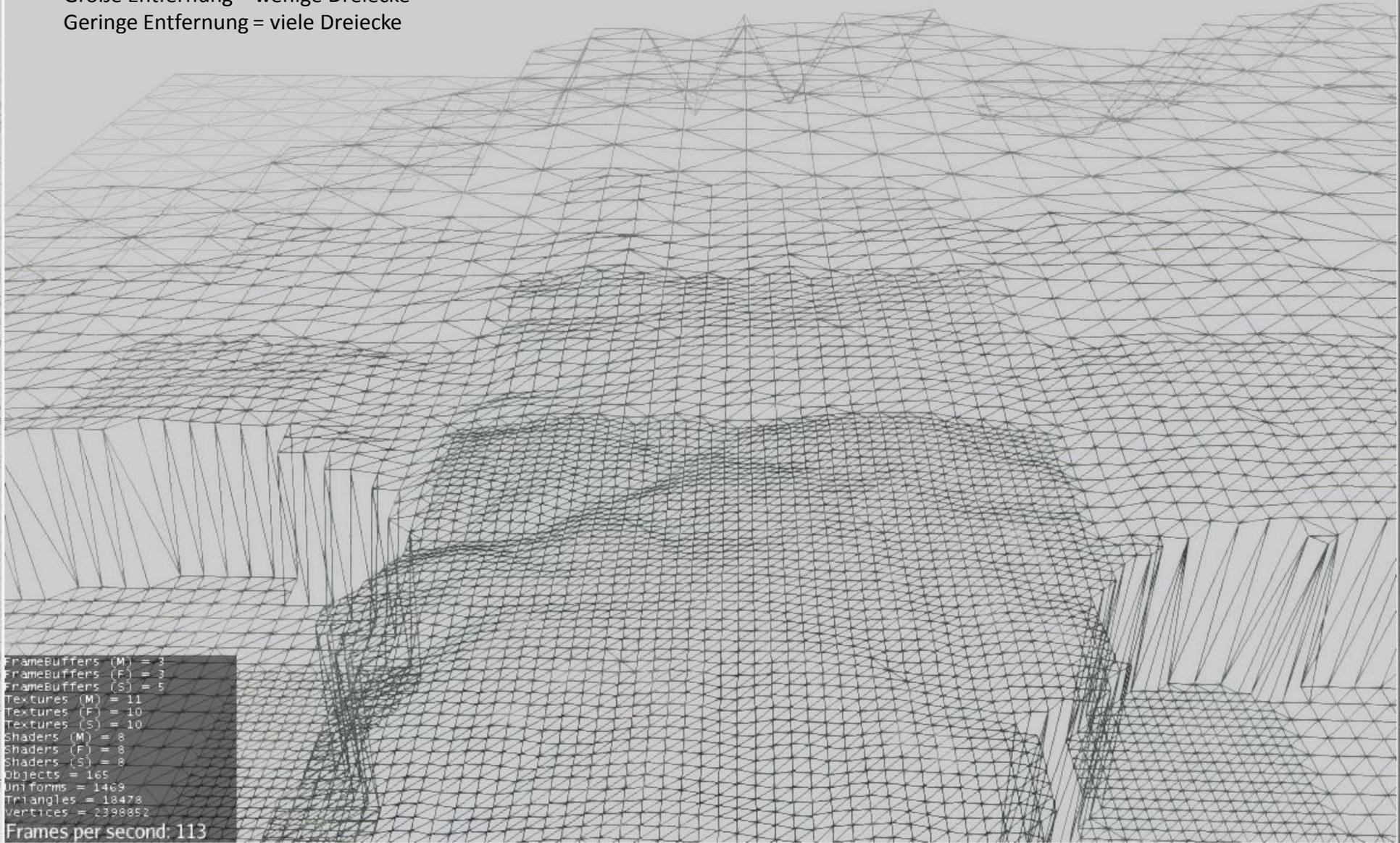
```
FrameBuffers (M) = 3
FrameBuffers (F) = 3
FrameBuffers (S) = 5
Textures (M) = 11
Textures (F) = 10
Textures (S) = 10
Shaders (M) = 8
Shaders (F) = 8
Shaders (S) = 8
Objects = 277
Uniforms = 2589
Triangles = 9420
Vertices = 4262640
```

Frames per second: 131

### Experimentelle Visualisierung des Wuppertaler DGM (Punkte-Kanten-Darstellung)

Große Entfernung = wenige Dreiecke

Geringe Entfernung = viele Dreiecke

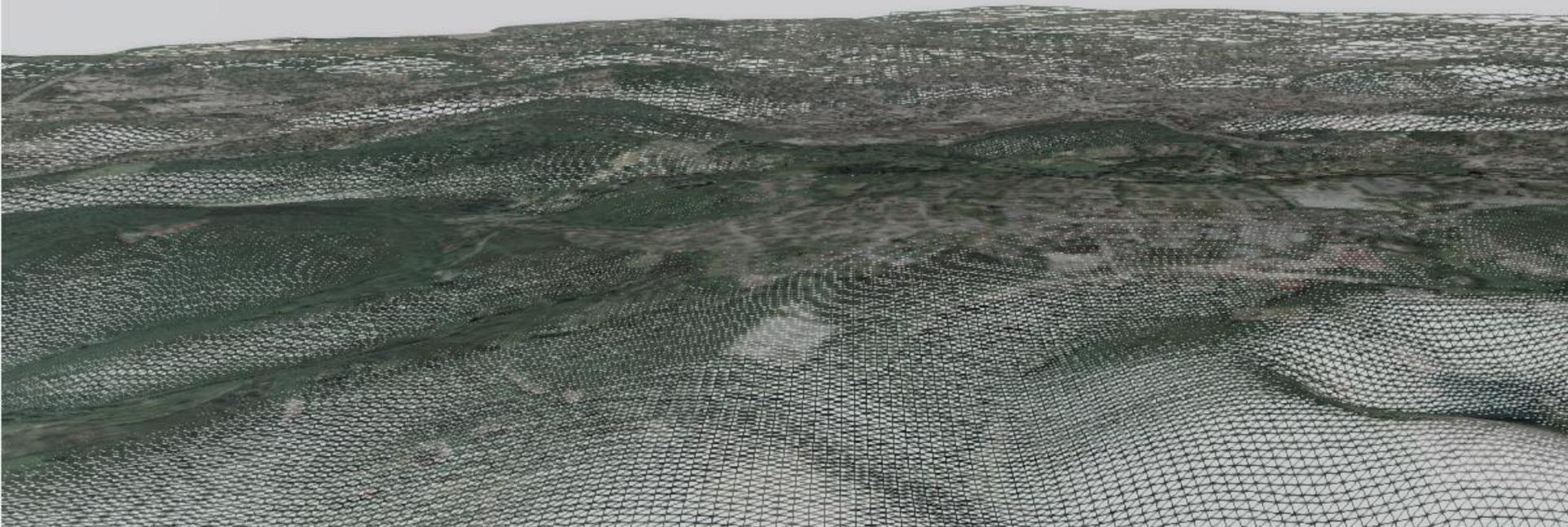


```
Framebuffers (M) = 3  
Framebuffers (F) = 3  
Framebuffers (S) = 5  
Textures (M) = 11  
Textures (F) = 10  
Textures (S) = 10  
Shaders (M) = 8  
Shaders (F) = 8  
Shaders (S) = 8  
Objects = 165  
Uniforms = 1469  
Triangles = 18478  
Vertices = 2398852  
Frames per second: 113
```

## Experimentelle Visualisierung des Wuppertaler DGM (Punkte-Kanten-Darstellung)

Große Entfernung = wenige Dreiecke

Geringe Entfernung = viele Dreiecke



```
FrameBuffers (M) = 3  
FrameBuffers (F) = 3  
FrameBuffers (S) = 5  
Textures (M) = 11  
Textures (F) = 10  
Textures (S) = 10  
Shaders (M) = 8  
Shaders (F) = 8  
Shaders (S) = 8  
Objects = 97  
Uniforms = 789  
Triangles = 213582  
Vertices = 1267256
```

Frames per second: 77

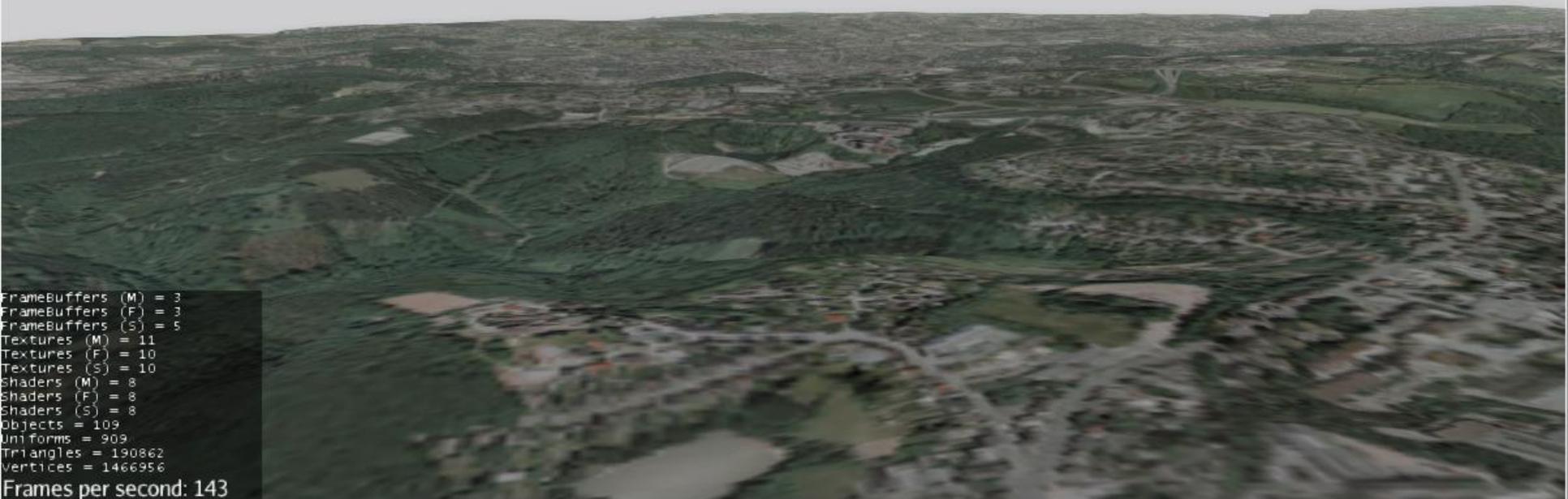
Experimentelle Visualisierung des Wuppertaler DGM (**Flächendarstellung**)

Große Entfernung = wenige Dreiecke

Geringe Entfernung = viele Dreiecke

```
FrameBuffers (M) = 3  
FrameBuffers (F) = 3  
FrameBuffers (S) = 5  
Textures (M) = 11  
Textures (F) = 10  
Textures (S) = 10  
Shaders (M) = 8  
Shaders (F) = 8  
Shaders (S) = 8  
Objects = 109  
Uniforms = 909  
Triangles = 190862  
Vertices = 1466956
```

Frames per second: 143



# Vielen Dank