About Me

Started Cesium

Books on Virtual Globes, WebGL, OpenGL

Contribute to Open Standards and Formats

Teach Computer Graphics

Open-Source Geospatial Community Service

http://www.seas.upenn.edu/~pcozzi/
A personal history
2008 – MS Thesis: Visibility Driven Out-of-Core HLOD Rendering
2011 – Streaming imagery in Cesium

Without tile processing
Without tile processing
With unbounded cache
Without tile priority details
Without layers

texture pool

Walk tile tree

image queue

Tile tree

Throttled texture creation

Update

Throttled HTTP request
Async response

Render
- draw back-to-front
- clear render queue

With anisotropic filtering

OGC®
2014 – Point cloud prototype
2015 – 3D Tiles

- Geospatial Datasets
  - 3D Tiles
    - Cesium 3D Tiles
      - Optimized for streaming
      - and other tile formats
  - 3D Engines

OGC
Submitted 3D Tiles to enter the OGC Community Standard process.

- Submission team from government, commercial, research, and academia
  - Analytical Graphics, Inc.
  - Bentley Systems, Inc.
  - Fraunhofer-Gesellschaft
  - Hochschule für Technik Stuttgart
  - US National Geospatial-Intelligence Agency (NGA)
  - virtualcitySYSTEMS GmbH

- Vote ends today
  - Reached quorum
  - Your feedback is deeply valued
    - Through OGC channels, or
3D Tiles

• Spec
  – Spatial data structure defined in JSON
  – Tile formats: binary with embedded JSON
    • Ready to render
  – Declarative styling

• Software ecosystem
  – Open-source Cesium implementation far along
  – Just kicked off open-source debugging, analysis, and validation tools with University of Pennsylvania
  – Several significant adoptions
3D Tiles

• Spatial data structure supports many tiling approaches
  – Server/tool can decide what is optimal for a dataset
• Client/runtime traverses generic spatial data structure
tile payloads

- Efficient JSON + binary runtime 3D model open standard from The Khronos Group
  - Open-source exporters, converters, pipelines, validators, loaders
- 3D Tiles normatively references glTF

Publicly Stated Support for glTF

khronos.org/gltf
• Use per-point properties in expressions to compute color, transparency, and show/hide.
  – Evaluated in parallel per point on the GPU

```json
{
    "color" : {
        "conditions" : {
            "$\{temperature\} < 0.1" : "color('#000099')",
            "$\{temperature\} < 0.2" : "color('#00cc99', 1.0)",
            "$\{temperature\} < 0.3" : "color('#66ff33', 0.5)",
            "$\{temperature\} < 0.4" : "rgba(255, 255, 0, 0.1)",
            "$\{temperature\} < 0.5" : "rgb(255, 128, 0)",
            "$\{temperature\} < 0.6" : "color('red')",
            "$\{temperature\} < 0.7" : "color('rgb(255, 102, 102)')",
            "$\{temperature\} < 0.8" : "hsl(0.875, 1.0, 0.6)",
            "$\{temperature\} < 0.9" : "hsla(0.83, 1.0, 0.5, 0.1)",
            "true" : "color('#FFFFFF', 1.0)"
        }
    }
}
```
3D Tiles Showcases

https://youtu.be/KoGc-XDWPDE
http://cesium.entwine.io/?resource=nyc

- 4.7 billion points in NYC. Post-Sandy collection from USGS
- Processed in two hours using Amazon’s 30 core machine
- Entwine is open-source: entwine.io
Point Cloud + 3D Buildings
Point Cloud + 3D Buildings
Point Cloud + 3D Buildings
Point Cloud + 3D Buildings

Copyright © 2016 Open Geospatial Consortium
Fully time-dynamic 3D Tiles kicking off soon. Join us!

Join us!

• **Fall 2016 / Spring 2017**
  – Discuss your use cases
  – Put finishing touches on spec
  – Finish open-source Cesium implementation and validator
  – Move 3D Tiles through OGC community standard process

• **Spring 2017 / Summer 2017**
  – Time-dynamic 3D Tiles

https://github.com/AnalyticalGraphicsInc/3d-tiles

(spec, examples, tutorials, etc.)
https://github.com/AnalyticalGraphicsInc/3d-tiles

(spec, examples, tutorials, etc.)

Contact

Patrick Cozzi
pcozzi@agi.com
@pjcozzi
Bonus Slides
• Open-source JavaScript library for world-class 3D globes and maps
• Millions of end users
• Lots of Google Earth migrations to Cesium
• Widely used by NASA

cesiumjs.org
3D geospatial datasets: Terrain, imagery, buildings, point clouds, trees, vector data, massive models, …

Massive heterogeneous 3D geospatial datasets  

3D engines, e.g., Cesium

OGC®
**3D geospatial datasets**: Terrain, imagery, buildings, point clouds, trees, vector data, massive models, …

Open specification for streaming massive heterogeneous 3D geospatial datasets
2D tiling has limited use in 3D

- OK for imagery
- OK-ish for terrain
- What about 3D buildings, point clouds, vector data, and massive models?

Image from virtualglobebook.com
2D tiling has limited use in 3D

- Sub-optimal subdivisions for non-uniform datasets
- No 3D subdivision, e.g., for point clouds
- Doesn't easily handle objects that overlap two tiles
- 3D requires multiple LODs in the same view
  - Need error metric for LOD selection
  - Need to avoid cracking visual artifacts
• The foundations for 3D tiling were created by
  – Graphics research
  – Movie industry
  – Game industry
• 3D Tiles brings these techniques to geospatial
3D Tiles in the ecosystem

- 3DPS
- CDB
- CityGML
- KML, GML, COLLADA, LAS...
- KHronos
- glTF
- WMS

- data discovery
- payload can be 3D Tiles
- retrieve and convert to 3D Tiles for streaming massive datasets
- preserve attributes
- convert to 3D Tiles for streaming
- individual 3D models
- can be payload in a 3D Tiles tile
- massive 3D datasets with attributes and styles
- optimized for runtime streaming
- 3D Tiles complement existing 2D standards

3D engines, e.g.,

OGC
Refinement: replacement vs. additive

Replacement

\(O(n \log n)\) tileset

Additive

\(O(n)\) tileset

visit(tile)
{
    if (computeSSE(tile) <= pixel tolerance)
        render(node);
    else
        foreach (child in tile.children)
            visit(child);
}

GPU-Friendly Compressed Tiles

- Optional **GPU-friendly compression**
  - **Quantized positions** – 16-bits per x, y, and z
  - **Oct-encoded normals** – 16-bit normals
  - **RGB565 colors** – 16-bit per color (point clouds)

- **Benefits**
  - No CPU decode cost
  - Stays compressed in GPU memory
  - Cheap or free decode in parallel on GPU

Copyright © 2016 Open Geospatial Consortium

Image by Kayvon Fatahalian, CMU
Selected 3D Tiles examples
Work in progress. Data thanks to the Federal Office of Topography swisstopo!
NYC: CityGML and OSM

- ~1.1 million buildings
- Mouse over highlight
- Dynamic styling with per-building attributes
- Separate imagery layer
- 12.84 GB CityGML converted to 1.85 GB 3D Tileset (727 MB gzipped)

CityGML: .json / COLLADA

OSM extract: .json / COLLADA

cesiumjs.org/NewYork/
- High-resolution terrain, buildings, and imagery
- Runtime annotation

cesiumjs.org/2016/03/08/Vricon-and-3D-Tiles/
• Textured buildings
• Translucent windows
• Derive LOD from CityGML LOD
• Full access to CityGML semantics
• Point Clouds
• Combine with terrain and imagery
Solar potential of each building in Berlin

http://bit.ly/2cCh4Jy
MicroStation export

- Massive CAD models
- Exteriors and interiors

Work in progress