Agenda

- Brief history of Cesium and 3D Tiles
- 3D Tiles vision and overview
- State of the 3D Tiles specification and software ecosystem
AGI is a space and defense company. About five years ago, our customers had a need for a lightweight cross-platform virtual globe engine. At the time, WebGL was just emerging so we built a JavaScript library – Cesium – which is now actively developed as open source.

These screenshots show Cesium being used for space situational awareness as part of ComSpOC.

Cesium – cesiumjs.org

ComSpOC - comspoc.com
Cesium has spread far beyond space and defense, into nearly every domain using 3D geospatial.
All of these use cases have common core algorithms for efficiently streaming 3D content based on the view. Each use case has a different data format and we often want to fuse together different formats, e.g., buildings (batched models) and trees (instanced models) on terrain (TIN).

Interactive and styleable – terrain/imagery not as common.
Many datasets, especially buildings, trees, and points clouds, are non-uniform: the features are not evenly distributed nor are the feature’s cost (e.g., number of triangles, size of textures, etc.).

Traditional quadtree subdivision, used in TMS for example, is sufficient for uniform 2D map tiles, but it does not account for 3D (e.g., point clouds) and is suboptimal for non-uniform dataset distributions.
Perfect timing for an open format

- Widespread use of Cesium and WebGL
- 3D geospatial data trends
  - Data acquisition easier than ever
  - Crowdsourcing
  - Open data policies
3D Tiles provides the flexibility needed for the overlapping but diverse 3D use cases.

I alluded to this at Web3D in 2014: http://cesiumjs.org/presentations/CesiumCzmlGltf.pdf

And presented some of the technical approaches at MIT a bit later in 2014: http://cesiumjs.org/presentations/RenderingMassiveGeospatialDatabasesInCesium.pdf
Current 3D Tiles spec: https://github.com/AnalyticalGraphicsInc/3d-tiles

Current Cesium implementation branch:
https://github.com/AnalyticalGraphicsInc/cesium/tree/3d-tiles
In 3D Tiles, a tileset is a set of tiles organized in a spatial data structure, the tree. Each tile has a bounding volume completely enclosing its contents. The tree has spatial coherence; the bounding volume for child tiles are completely inside the parent's bounding volume.
The tree allows for a wide array of spatial data structures so we can create an optimal spatial data structure for the geospatial dataset taking into account the cost of rendering individual models and their distribution, e.g., sparse vs. dense.

Top left: kd-tree
Right: Non-uniform overlapping quadtree (loose-ish quadtree), e.g., for 3D buildings
Bottom left: Octree, e.g., for point clouds

Note that this is different than traditional 2D tiling schemes, such as TMS, which are generally rigid.
Tile Formats

- Batched (b3dm)
- Instanced (i3dm)
- Point Cloud
- Composite
- **Upcoming**: Vector

b3dm and i3dm embed binary glTF:
https://github.com/KhronosGroup/gltF/blob/master/extensions/Khronos/KHR_binary__gltF/README.md
Write short declarative styles using expressions written using a feature’s properties (stored as metadata in the tile) to set a feature’s color (RGB), translucency, and show.

Here, a color ramp is applied based on building height, and the tallest buildings are translucent.

In Cesium, also modify features at runtime, e.g., highlight on mouse over.
3D Tiles and OGC?

- Potential
  - Complement 2D standards: WMS, WMTS
  - Transmission format for 3DPS
  - Streaming for large CityGML datasets
  - Replacement for some KML use cases
- Designed for visualization clients
- Per-feature attributes
Starting to focus on vector data, and then flushing out point cloud and instanced tile formats.
Washington, DC: Over 135,000 structures

Use additive refinement so the “most important” buildings are streamed first and never replaced. Works well for a large number of simple models – would not work for complex models. 3D Tiles provides the flexibility to create an optimize data structure based on the input dataset.
- Drop building “skirts” to clamp to terrain

.csv / .dae / terrain
Demo: http://cesiumjs.org/NewYork/

OSM extract from Mapzen: https://mapzen.com/data/metro-extracts/
57,714 trees using instanced tiles (i3dm).

Could also make a tileset where the leaf tiles are full 3D models (with instance), and interior tiles are vector tiles representing the trees as imposters.

OpenTreeMap data: https://www.opentreemap.org/
http://www.virtualcitysystems.de/en/

Screenshot: Berlin

Demo (with just open data):
Top right: select wall surface of the CityGML building

Bottom left: show building CityGML attributes

Screenshot: Berlin
• Berlin 3D city model
  – 539,401 fully textured LoD2 buildings
  – 890 km² digital terrain model
  – Approx. 370 LoD3 buildings and 3 LoD4 buildings (interior details)
  – City trees ($\text{SollitaryVegetationObject}$), bridges, S-Bahn stations, ...
  – Data volume
    • 16.6 GB CityGML file (XML unzipped)
    • 19.4 GB texture files

• 3D Tiles / b3dm files for the entire model
  – Three TMS layers with different LoDs and texture resolutions
    • Level 15: LoD2 only and low-res textures (2.54 GB)
    • Level 16: LoD2 only and mid-res textures (4.22 GB)
    • Level 17: LoD2-4 and high-res textures (19.5 GB)
http://www.cityzenith.com/ - awesome video on their website

5D Smart CityFeatures Video: https://vimeo.com/157494960

Above: Chicago

Annotation and measurement tool using Cesium entities.
Top left - San Francisco. Prebaked AO
Bottom right – Washington, D.C.
Separate tileset each for buildings, trees, windmills, etc.

Fraunhofer IGD:
• Institute for applied research in Visual Computing which focuses among others on the visualization and management of geoinformations
• Applications:
  • Screenshot: Visualize plannings of e.g. windparks to local communities and to stakeholder of the infrastructure project
  • Visualize huge city models
  • Smart City plattforms, etc.

Problem:
• Currently the citymodels and all other objects are added as gltf models
  • Very basic (self implemented) streaming capabilities

3D-Tiles is currently implemented to have:
• Proper streaming capabilities
• Access to metadata
• Instancing of objects (trees)
• Visualization pointclouds
TIN

http://www.vricon.com/
Runtime annotation gets exact click position (e.g., as opposed to the building’s center) from depth buffer using Cesium’s pickPosition().
Check out 3D Tiles

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

Contact

agi
Patrick Cozzi
pcozzi@agi.com
@pjcozzi

Thanks to
Bonus Screenshots
http://www.virtualcitysystems.de/en/

Screenshot: Berlin
Screenshot: Berlin