



3DTiles

An open specification for
streaming massive 3D
geospatial scenes

Matt Amato
[@matt_amato](https://twitter.com/matt_amato)

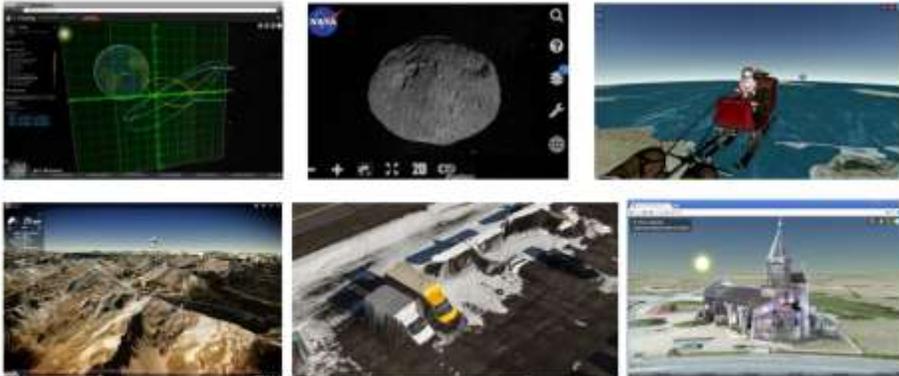


CESIUM  agi

Thanks for taking the time to check out my slides. This presentation is an overview of Cesium 3D Tiles, why we need them, what they can do, and an overview of the ecosystem of who is currently using them.

Cesium

- Open-source JavaScript library for world-class 3D globes and 2D maps
- Millions of end users across a variety of industries
- Widely used by NASA, NGA, state & local governments, and more
- Lots of Google Earth migrations to Cesium



cesiumjs.org

2 of 17

In case you're not familiar with it, Cesium is an open-source browser-based geospatial visualization engine for 3D globes and maps. It uses the Apache 2 license and because it's built with browser-based technology can run almost anywhere. It's grown into a large open source community project, with over 80 contributors and a vibrant forum with over 900 members.

Cesium provides 3D, 2D, and 2.5D (what we call Columbus View), all through a single API. It's not just open source, it was also built for and with open standards in mind. It's highly customizable with a flexible API. Finally, in addition to 3D, one of the things that sets it apart is our handling of time-dynamic data, almost everything in Cesium can be animated over time without losing interactivity. Cesium was originally built with aerospace use cases in mind, so it's incredibly precise and accurate in its visualization and can go from ground level to outer space from within the same map.

If you're new to Cesium I definitely encourage you to check out the many demos and presentations available on cesiumjs.org.



When we started Cesium, the last thing we wanted to do was get involved in formats, standards, and specifications. After all, we knew there was already a lot of data out there, and we simply wanted to make the most robust geospatial visualization engine to use them.

Of course what we quickly realized is that there are two major problems with this approach.

First, most of the existing standards and techniques for 2D data visualization do not translate well into 3D. This applies to both vector and raster data. The catchphrase I like to use for this is that 3D is a lot more than just 2D + 1. The third dimension requires a fundamental different approach to data visualization in general, for example, 2D maps have a zoom level that applies to the current view, but that concepts doesn't apply to 3D because tilted views have a continuous change in zoom level as you move away from the camera. Because of this, traditional 2D tiling schemes doesn't work well in 3d.

Second, existing 3D data formats are more about data interoperability and storage and not data visualization. These formats need a lot of processing, either offline or at runtime in order to make them usable inside of a 3D engine.

This means there also not built for streaming, particularly over the web, or usable on underpowered devices, such as tablets and smart phones. We need a standardized format for real-time visualization, interrogation, and styling. This includes traditional datasets like vectors, imagery, and terrain, but also buildings, massive 3D models, point clouds, weather and many other forms of 3D data.

Cesium 3D Tiles



3D geospatial datasets: Terrain, imagery, buildings, point clouds, vector data, trees, airspace, weather, massive models, ...

Massive heterogeneous
3D geospatial datasets



3D engines, e.g.,

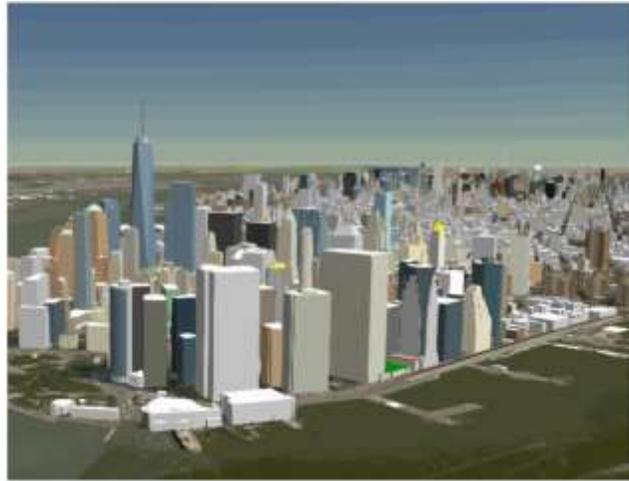


Open specification for streaming massive
heterogeneous 3D geospatial datasets

4 of 17

3D Tiles builds on techniques from the movie and game industry, as well as graphics research papers and applies them specifically to geospatial visualization.

More than a pretty picture



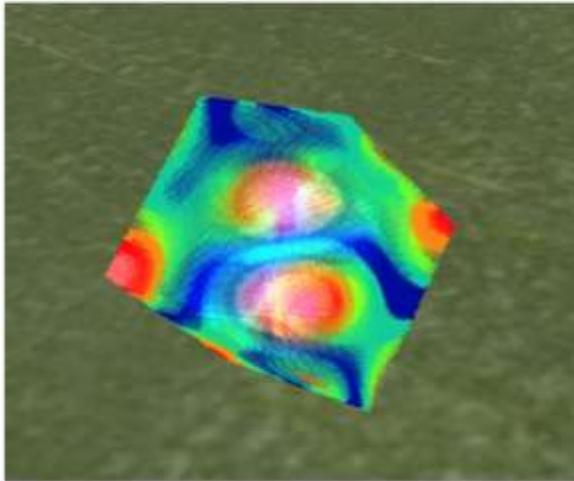
5 of 17

Now you might have noticed I just said, visualization, interrogation, and styling. If all we were after is visualization, the end result wouldn't be very useful. It has to be more than a pretty picture and we need to retain interactivity and query-ability.

For example, take this animated gif of Manhattan (sorry it doesn't translate to print), from a tileset of about 1 million buildings. As we mouse over the buildings they light up and show their name. Clicking on a building will actually bring up the metadata associated with the specific building.

This metadata can then be used to style the buildings based on any parameters we wanted; such as height, zoning, energy usage. Any type of feature data you have can be fed back into the styling system. You can even show and hide buildings dynamically.

Unified styling



Use **per-point properties** in expressions to compute **color, transparency, and show/hide**.

6 of 17

Of course the previous slide discussed buildings, but as I stated earlier, 3D Tiles is heterogeneous, which means the same exact styling language and metadata discovery applies to all types of payloads, such as this point cloud image showing per-point temperature styling. So whether you have rgb, temperature, chemical composition, or other metadata, they can be fed into the declarative styling system.

In Cesium, we do most of the styling on the graphics card, which makes it incredibly fast and even suitable for time-dynamic updates.



- khronos.org/gltf
- Efficient 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

7 of 17

The main approach we take in Cesium is to use open standards and specifications wherever possible, but we create open standards and specifications when they are needed. You can't talk about 3D Tiles without mentioning glTF. 3D Tiles is not the first spec we've written. Just like there was no good runtime standard for 3D geospatial data, there wasn't a standard for individual 3D models as well. Think of representing a single building or tree model rather than an entire city.

So a couple of years ago the Cesium team, lead by Patrick Cozzi, worked with The Khronos Group to come up with glTF. For those of you who aren't aware, Khronos is a standards body primarily known in the graphics industry. They have created things like OpenGL, WebGL, and Collada.

glTF has caught on big and these are just some of the companies that are using it in their products or have otherwise publicly stated support.

The reason I'm bringing this up now is because the 3D Tiles spec normatively references the glTF format for 3D model representation, so you should just be aware of what it is and why it exists.

OGC standardization

- **3D Tiles** has entered 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
- Voting ended September 22nd, reached quorum

8 of 17

And speaking of standardization, We are making 3D Tiles more than an open spec. We've already started the process of becoming an OGC Community Standard. Adoption of 3D Tiles has been so rapid, that we were able to form a submission team from government, commercial, research, and academia.

Selected examples



9 of 17

So that's the mile high view of what 3D Tiles is, why we need it, and what we are doing to make sure it is an open standard that can be adopted throughout the industry. I already mentioned that the rate of 3D Tiles adoption has been incredibly fast. For the next part of the talk, I want to go over some showcases from various companies and groups working with 3D Tiles.



Federal Office of Topography swisstopo



- 1.5 million buildings with 3D Tiles
- Buildings clamped to terrain
- Separate terrain and imagery layers
- 22.2 GB KML/COLLADA converted to 1.5 GB 3D Tileset (0.97 GB gzipped)

KML + COLLADA



Work in progress. Data thanks to the Federal Office of Topography swisstopo!

agi NYC: CityGML and OSM

cesiumjs.org/NewYork/

- ~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 → 3DTiles

OSM extract → .json / COLLADA → 3DTiles

11 of 17

We took the New York City OpenStreetMap extract from Mapzen and turned it into 3D Tiles.

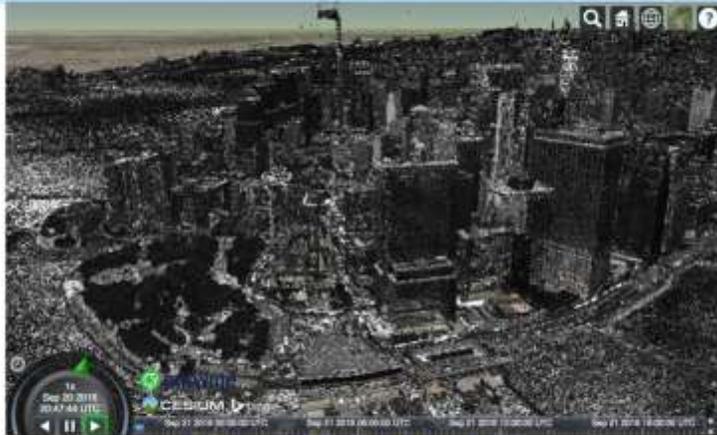
Not only are we visualizing over 1.1 million buildings, but you can mouse over each building to see what it is and click to get more information. Keeping this level of interactivity in tact was very important to us. We didn't just want to turn the data into a pretty picture, we wanted to avoid losing any information.

Here are some highlights:

1. You can select various points of interest in NYC using the combo box in the upper left
2. Clicking on a building gets you the OSM metadata, including links back to the same feature on OSM
3. You can color buildings by their height (this styling is computed on the fly and doesn't have to be "baked in" to the dataset.) You can also select different color palletes.
4. You can also use the slider to hide buildings below a certain height.
5. The "Show broken OSM relations" checkbox will be explained in a future

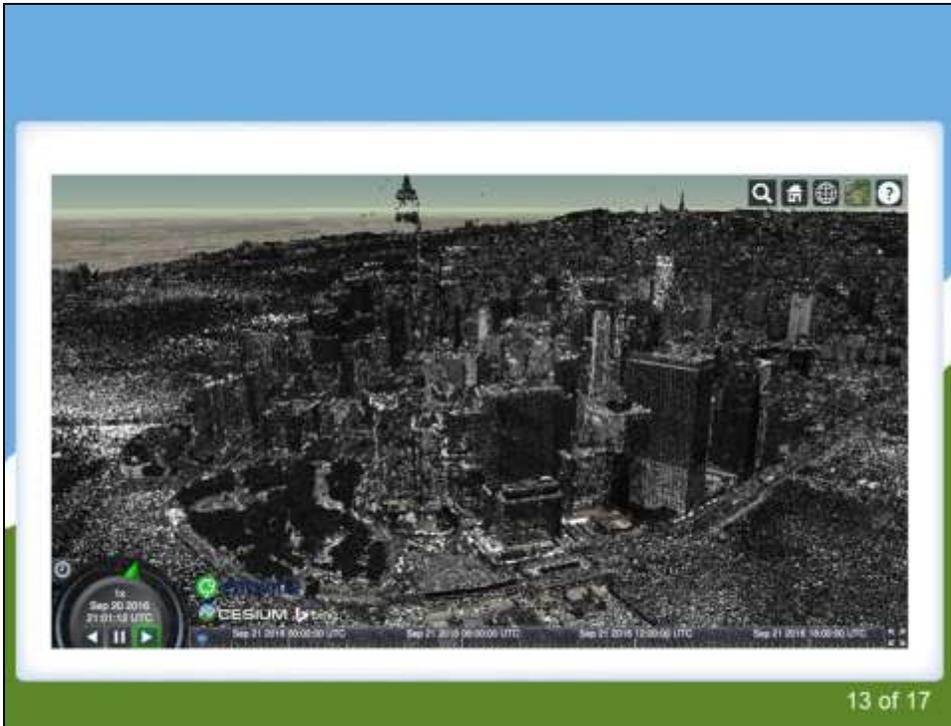
slide.

6. Flyover mode is just a “demo” mode that disables mouse control and takes

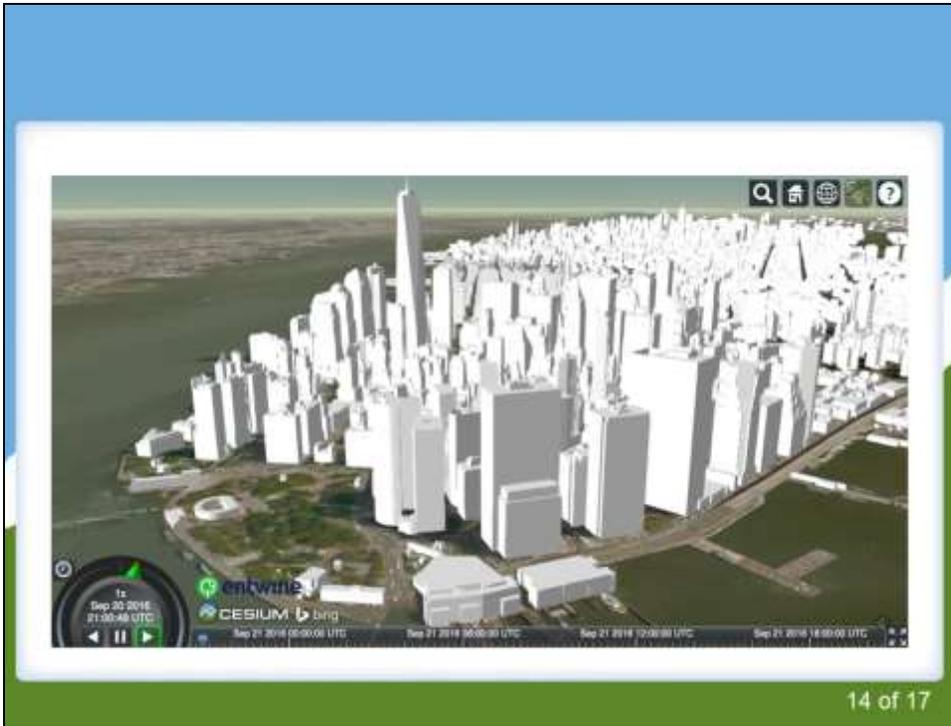


<http://cesium.entwine.io/?resource=nyc>

- 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



Since the NYC and Entwine demos both showcase New York datasets, we couldn't help but to create a few slides of them being combined together. This is just the point cloud.



The same view with the official NYC CityGML dataset.



And finally, both of them together.

VRICON



The screenshot shows a 3D rendering of a city, likely Rio de Janeiro, with a UI overlay. The UI includes a search bar, a 'Bookmarks' panel with several thumbnails, and a '+' icon. The city is rendered with high-resolution terrain, buildings, and imagery.

- High-resolution terrain, buildings, and imagery
- Runtime annotation

cesiumjs.org/2016/03/08/Vricon-and-3D-Tiles/

16 of 17

Check out <http://cesiumjs.org/2016/03/08/Vricon-and-3D-Tiles/> for more info on Vricon's use of 3D Tiles.



~540K full textured LOD2 buildings.

Create a three-level tree (TMS levels 15-17) with different LODs and texture resolutions.

<http://www.virtualcitysystems.de/en/>

Screenshot: Berlin

Dataset:

Berlin 3D city model

539,481 fully textured LoD2 buildings

890 km² digital terrain model

Approx. 370 LoD3 buildings and 3 LoD4 buildings (interior details)

City trees (SolitaryVegetationObject), 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)

Demo (with just open data):

<http://hosting.virtualcitysystems.de/demos/berlin/vcm/index.html?lang=en>



Bentley Context Capture

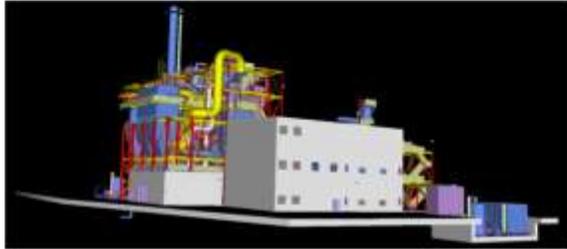


ContextCapture reality mesh export



- Buildings, terrain, trees, etc.
- Publish data and app to the Web
- Overlay vector data

<https://www.linkedin.com/pulse/contextcapture-web-publishing-cesium-aude-camus>



Work in progress

MicroStation export



- Massive CAD models
- Exteriors and interiors



https://sample.aero3dpro.com.au/NightCliffe_2016/App/index.html

Fully time-dynamic 3D Tiles kicking off soon.

Roadmap

- **Fall 2016 / Spring 2017**

- 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

Questions?

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

Contact

Matthew Amato

mamato@agi.com

@matt_amato

Showcases thanks to



22 of 17

Thanks again for checking out my slides. Please don't hesitate to contact me or the Cesium team if you have any questions or comments.