Alright, hi everyone! My name is Omar, I’m a graphics programmer at Cesium. I mostly work on developing our open source tools & supporting the ecosystem. I want to share some of the progress we’ve made over the past year towards this goal of making the vast amounts of 3D data out there accessible and usable through the web, as well as talk a bit about where we’re headed in the future.
So for those who don’t know, Cesium is an end-to-end platform for 3D geospatial data. We’re most commonly known for CesiumJS, the open source JavaScript library for 3D globes and maps on the web. But we also do a lot of work with open standards, we created 3D Tiles and co-created glTF, which together make it easier to stream massive 3D datasets. And then there’s Cesium ion, a subscription service for finding 3D content, as well as tiling and hosting your own. This is where the satellite imagery for most Cesium applications is hosted for example.
We’re going to start with a 3D Tiles update, then look at some performance improvements in Cesium, as well as some visual improvements, before we see what’s ahead.
The big piece of news is that 3D Tiles is now officially an open standard of the Open Geospatial Consortium as of February 5th! We’re pretty excited about this because it means we’re moving towards a world where it's much easier to share and work with 3D geospatial data.
The idea is that regardless of where your 3D data comes from, whether captured from drones, LIDAR, or if it's a design model of a building that will exist in the future, it can all be converted into this open format.
Sharing 3D insights is as easy as sharing a link

Once converted to 3D Tiles, it becomes really easy to stream it over the web, because you can request just the data you need. This is really cool because it works without having to install anything. You don’t need to send someone over 100 GB of data just to show them this one neighbourhood or one particular insight.
For example, this is a high resolution dataset of the entire city of Melbourne Australia, captured by Aerometrex. The whole thing is around 11 GB. So it would take around an hour to download at our office’s WiFi, and then maybe you need some kind of desktop software to open it. And of course, you gotta make sure you have enough RAM!

With 3D Tiles, you can send someone a link to a web app with just this view, which only takes just 30 mb. So it goes from an hour to just a few seconds to share this. And because it’s web based, you could look at this on your phone, or really any device with a web browser!
Performance Improvements in Cesium

Now a lot of what we’ve been working on has been finding ways to make this streaming even faster in our open source CesiumJS library.
One of the biggest improvements has been adding support for WebP images. You can see the scene on the left loads a lot faster using WebP, compared to the JPEG version on the right. This is a photogrammetry dataset of New York captured by Nearmap.
WebP is generally 30% smaller

<table>
<thead>
<tr>
<th>Lossless Compression</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>PNG</td>
<td>63 MB</td>
</tr>
<tr>
<td>WebP Lossless</td>
<td>43 MB</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lossy Compression</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>JPEG</td>
<td>6.93 MB</td>
</tr>
<tr>
<td>WebP</td>
<td>5.30 MB</td>
</tr>
</tbody>
</table>

cesium.com/blog/2019/02/12/faster-3d-tiles-streaming-webp

WebP is not a new image format, but it’s newly been getting browser support. It generally provides 30% compression at equivalent quality to JPEG or PNG. This is a big boost for tilesets that are textured since textures often make up the largest % of data by filesize.

What’s really cool about WebP is that it supports transparency with lossy compression. So before, if you had a tileset with windows that were transparent, you were forced to use PNG, which made it 10x heavier. This is currently only supported in Chrome & Firefox.
Another big win is support for Draco compression. Draco is an open source library developed by google for compressing 3D geometry. Here we're looking at a tileset of CityGML buildings in New York. On the left is the Draco compressed version, which is a total of 100 mb. On the right is without Draco compression, and it's about 700 mb. This doesn't just make things load faster, it uses significantly less data.

The reason it's only 45% faster is because there is a cost to decoding it, but it's still overall faster than without compression. You can apply this to anything that has 3D geometry, like models of buildings, as well as point clouds!
For point clouds, this is a dataset with 185 million points with color, intensity and classification attributes. This was captured by Trimble. You can see it’s dramatically smaller, it goes from 1.8 GB to 0.6 GB.
Draco also allows you to configure the precision you care about during compression. Here’s what it looks like at millimeter, centimeter and decimeter precisions.
33% Faster Terrain & Imagery

We’ve also recently done a big overhaul of our terrain and imagery system to make it load a lot faster and use less data. The before is on the left. The big difference is that before, to load a given view, it had to load all levels of detail one by one up to this view. Now it can skip levels, so it’s faster, but also uses a lot less data. This technique of skipping levels of detail is something we already do for 3D Tiles, but now terrain & imagery can benefit from it too.
Visual Improvements in Cesium

Now in addition to all of the performance work, we’ve also been working on some exciting visual improvements.
A big one is clamping polylines to the underlying terrain. The goal was to get something that was fast, visually precise, and easy to use.
It works pretty well on mid-range hardware even with thousands of polylines. There’s a lot you can do with this, for example...
Here’s an example visualizing ski trails in Switzerland.
It also supports different materials like this glow line, or dashed line. You could also create your own custom materials and apply it as well.
We also added support for polylines on 3D Tiles. This now means that any 3D data you can visualize on the globe on Cesium can have a visually accurate line drawn through it. This is important because one of our values is that features should work seamless across datatypes. You don't have to worry about whether polylines are supported for city models vs point clouds etc. If you can visualize it in Cesium, then it works.

So in this example we're using this line to mark a path an autonomous vehicle might take to park in that spot.
We can also do things like mark a particular set of pipes that are due for inspection, or mark it by the type of fluid going through it.
A related new feature is the ability to drape textured materials on 3D Tiles. This is really powerful because it means you could have an untextured capture of a field or an area, and then overlay that with satellite imagery that changes every day, or that comes from different sources. And again, there’s no difference between 3D Tile and terrain here, a single material can be draped seamless across all data types.
Another update to materials is the ability to do procedural shading on terrain. This makes it really easy to highlight things like elevation or slope, as well as dynamically visualize contour lines. Not only that, but since this is all open source, it’s not too hard to create your own custom shading off of this.
In fact, one external contributor recently opened a pull request to add aspect shading to Cesium. This makes it easy to see what cardinal direction the surface normal points to.
Another new feature that’s really exciting is Classification. Cesium’s classification system makes it really easy to dynamically highlight features in 3D Tiles. The idea is that the intersection between any two datasets can define a highlighted volume. Here we’ve got the photogrammetry tileset of Melbourne we’ve seen earlier, combined with a GeoJSON vector dataset I got from a publically available government website. Here the colors show us the different types of roads. Like orange is a main road, and red I believe are private roads.

What’s really powerful about this is we can now say, I want to only see the area in the photogrammetry, that’s covered by the GeoJSON.
Classification

Here’s what that looks like. We can instantly see just the roads highlighted. The fact that you can take one tileset, and used to classify or highlight another tileset in real time like this means you can update one or the other independently. And instead of spending a lot of time annotating the roads in your drone capture, you can instantly fuse them like this.
Classification

It's also interactive! So you can easily add dynamic annotations, like when you click on a road, or here, a certain structure in the building, you can have it display some metadata about it.
Cesium has had clipping planes for a while, but we’ve made a lot of improvements to them to make them easier to use 3D Tiles. Here’s you can see how you can interactively use it slice through a point cloud to see its internals, and then take some measurements.
It again, works across any data type that you can visualize in Cesium, including the globe itself. Here’s an example showing how you can cut out a section of terrain by defining these clipping planes.
Clipping Planes

So you can use it to cut out an area and replace it with a high resolution drone or LIDAR capture.
Clipping Planes

You can also flip this, to cut out everything but what’s inside. Here I’m using it to isolate a section of the Grand Canyon.
Lastly, we’ve made a lot of improvements to post processing in Cesium. There’s now a lot of built in effects like ambient occlusion to make it easier to see surface definition in indoor environments, and just cool things like lens flare, but we now have a flexible post processing system so you can create your own custom effects and combine them as layers.

The buildings on the left have their geometry batched together to make rendering really fast, but you can still apply effects to individual buildings since Cesium keeps track of which vertices belong to which building. This lets you do things like dynamically outline any particular building or feature.
This does help a lot in making things look nicer. This is one of my favorite shots of the Earth from space in Cesium, that has a little bit of bloom applied as a post process to give that glowing feel.
Summary

- 3D Tiles is now an OGC community standard 🎉
- Performance Improvements in Cesium
  - WebP images make tilesets 30% smaller
  - Draco compression makes geometry 80% smaller and load 45% faster
  - Terrain & imagery now load 33% faster and use less data
- Visual Improvements in Cesium
  - Polylines on terrain & 3D Tiles
  - Textured materials on 3D Tiles
  - Procedural terrain shading such as slope, elevation & contour lines.
  - Dynamic highlighting with classification
  - Clipping planes
  - Post Processing
- And more! See cesium.com/blog

So to summarize, <read slide>. And this is just the highlights!
What’s Next?
A big theme is analysis tools. These tools already exist, but we’re working to bring them to more users by offering them on Cesium ion, so you can upload your terrain, and then do this kind of visibility analysis with your drone capture combined with the Cesium World Terrain.
Viewsheds and Line of Sight
Find visible and occluded regions in any 3D data

You can also see the visibility from one particular spot.
I hinted at this earlier with the clipping planes, but being able to take these measurements on any data. Volume is also something we’re actively working on.
More support for 3D Tiles!

We’d also love to see 3D Tiles supported in more engines. STK, Systems Toolkit, is used for modelling and simulations in aerospace and defense. They just added support for importing 3D Tiles. The more software starts to support 3D Tiles the closer we get to this vision of being able to take your data seamlessly from one program to another. Here, the application is often doing some kind of analysis locally on your desktop, and then sharing the results on the web with CesiumJS.
Looking forward

- REST API for Cesium ion
- Terrain & imagery in 3D Tiles
- Vector 3D Tiles

The REST API for Cesium ion is something that’s coming very soon. I’d love to see terrain & imagery folded into 3D Tiles. It is on our radar, as well as further developing the vector format in 3D Tiles.
Thanks!

omar@cesium.com

@CesiumJS
cesium.com

If you have any questions about 3D Tiles or Cesium, I’m always happy to chat!