

Web3D 2026

13-15 October 2026
Doha, Qatar

OGC 3D Tiles implementation for X3DOM

Andreas Plesch*
Harvard University
Cambridge, MA, USA
EarthStruct, LLC
Waltham, MA, USA
andreas_plesch@harvard.edu

Anita Havele
Management
Web3D consortium
Salinas, CA, USA
anita.havele@web3d.org

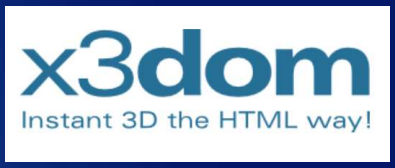


Figure 1: Sibbe OGC 3D Tileset superimposed on Google Photorealistic 3D Tileset in X3DOM scene

Background: 3D Web Geospatial Ecosystem

- The **X3D** Web3D Consortium specification
  - Interactive 3D scenes on the Web in real-time.
  - Supports a wide range of 3D graphics features.
  - Support for multiple coordinate projections, double-precision data types.
- The OGC **3D Tiles** specification
  - Efficient delivery and visualization of massive geospatial datasets on the web.
- We provide **interoperability** between these web technologies.











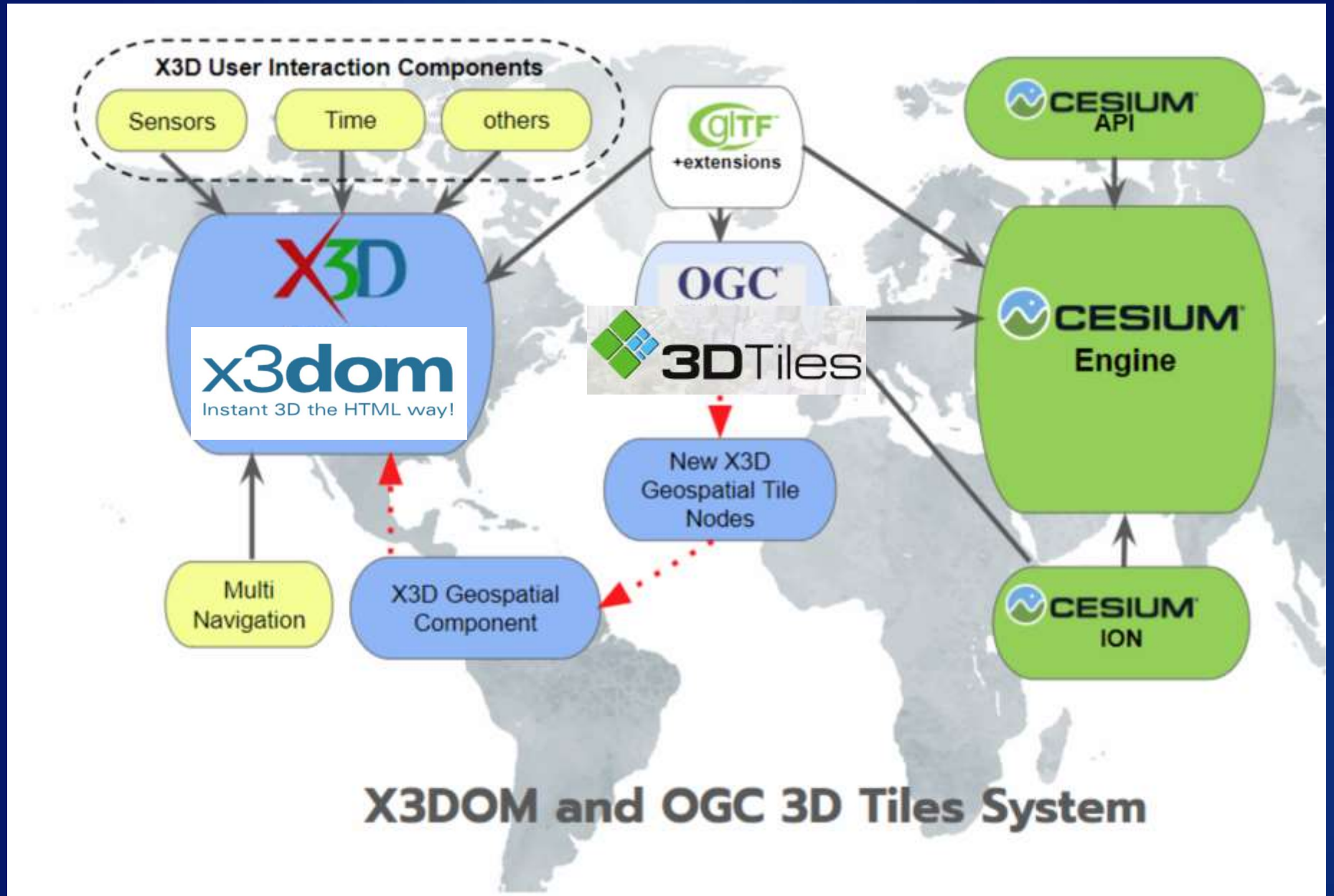
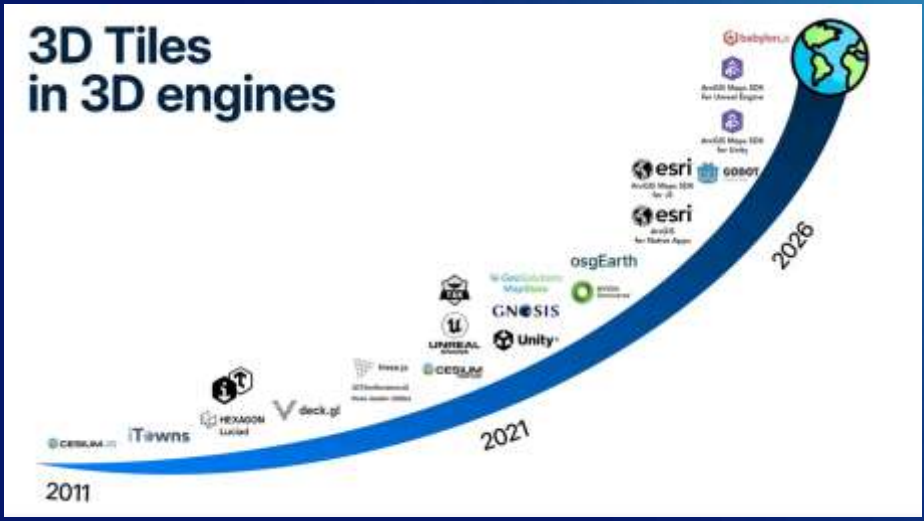


Figure 1: Relations between X3D, X3DOM, Cesium systems, glTF and OGC 3D Tiles.

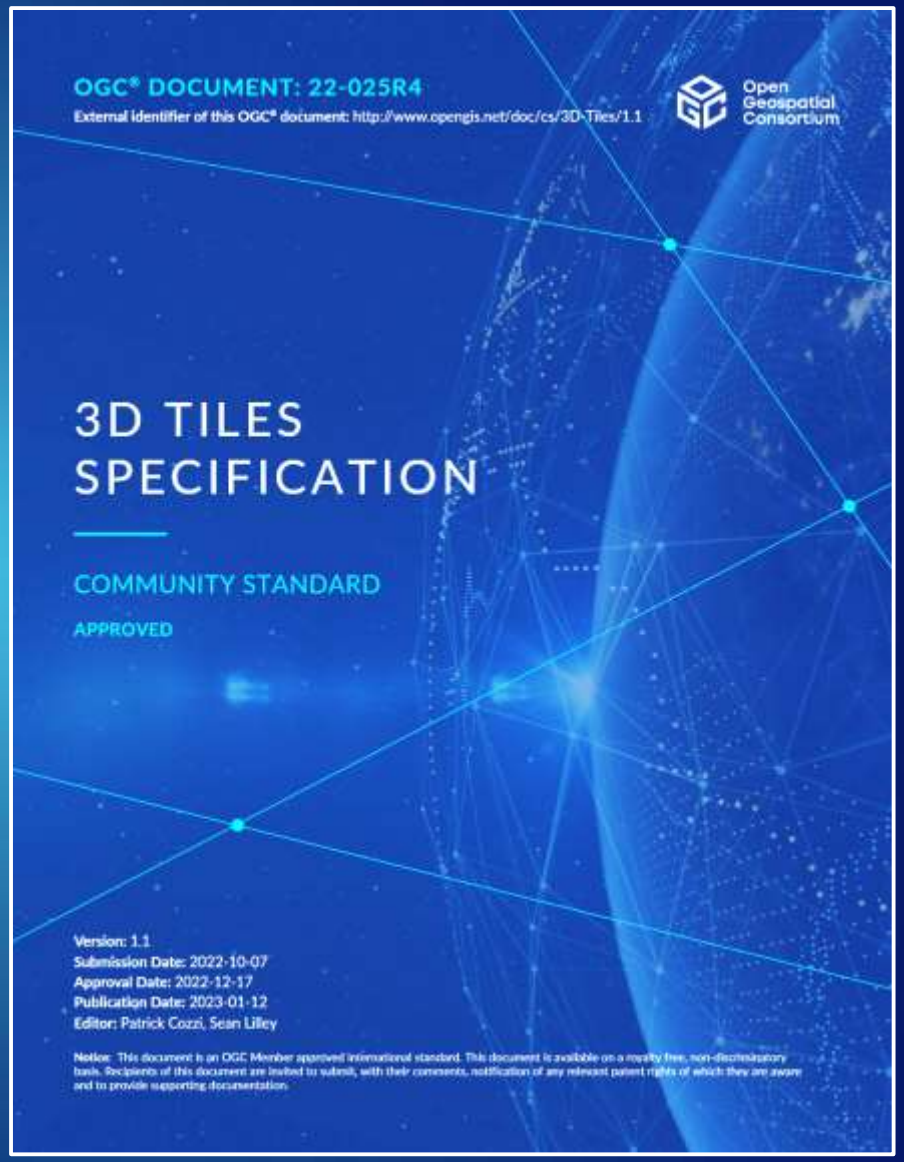
Havele et al. (2024)

OGC 3D Tiles Summary

- **Open** Specification
  - For streaming massive heterogeneous 3D geospatial datasets
- **Spatial** Data Structure
  - Enables Hierarchical Level of Detail (HLOD)
- **Efficient** Level of Detail
  - Based on selection by geometric error from screen space (not distance)
- **Adaptive**
  - Optimized for rendering by **leveraging glTF** as content formt
  - 3D Tiles 2.0 will be integrated into glTF as a large extension

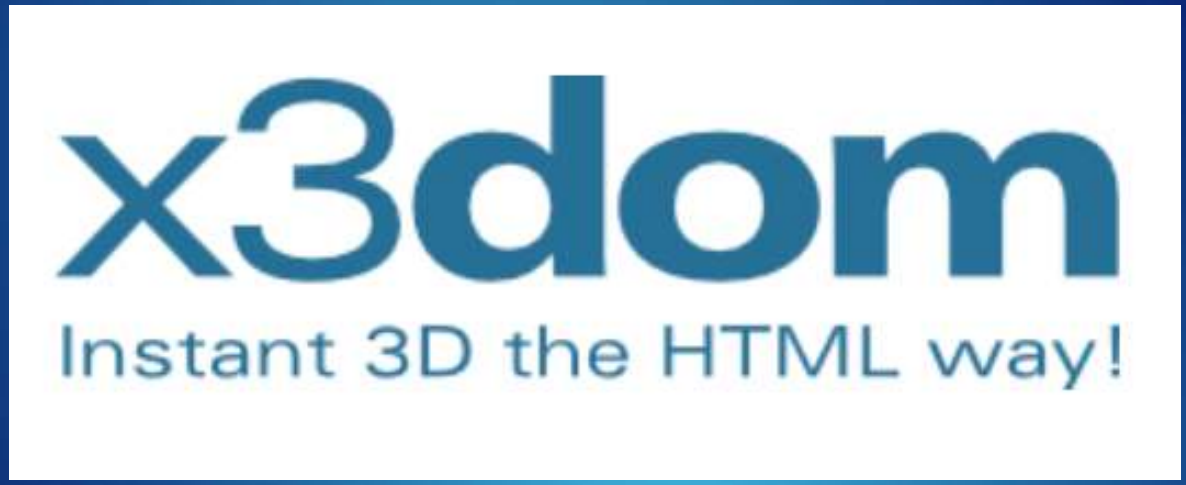


Cozzi (2026): 3D Tiles adoption



OGC (2022)

X3DOM: X3D Open-Source Implementation for the web



X3DOM.org

Javascript Engine

X3D Geospatial Profile

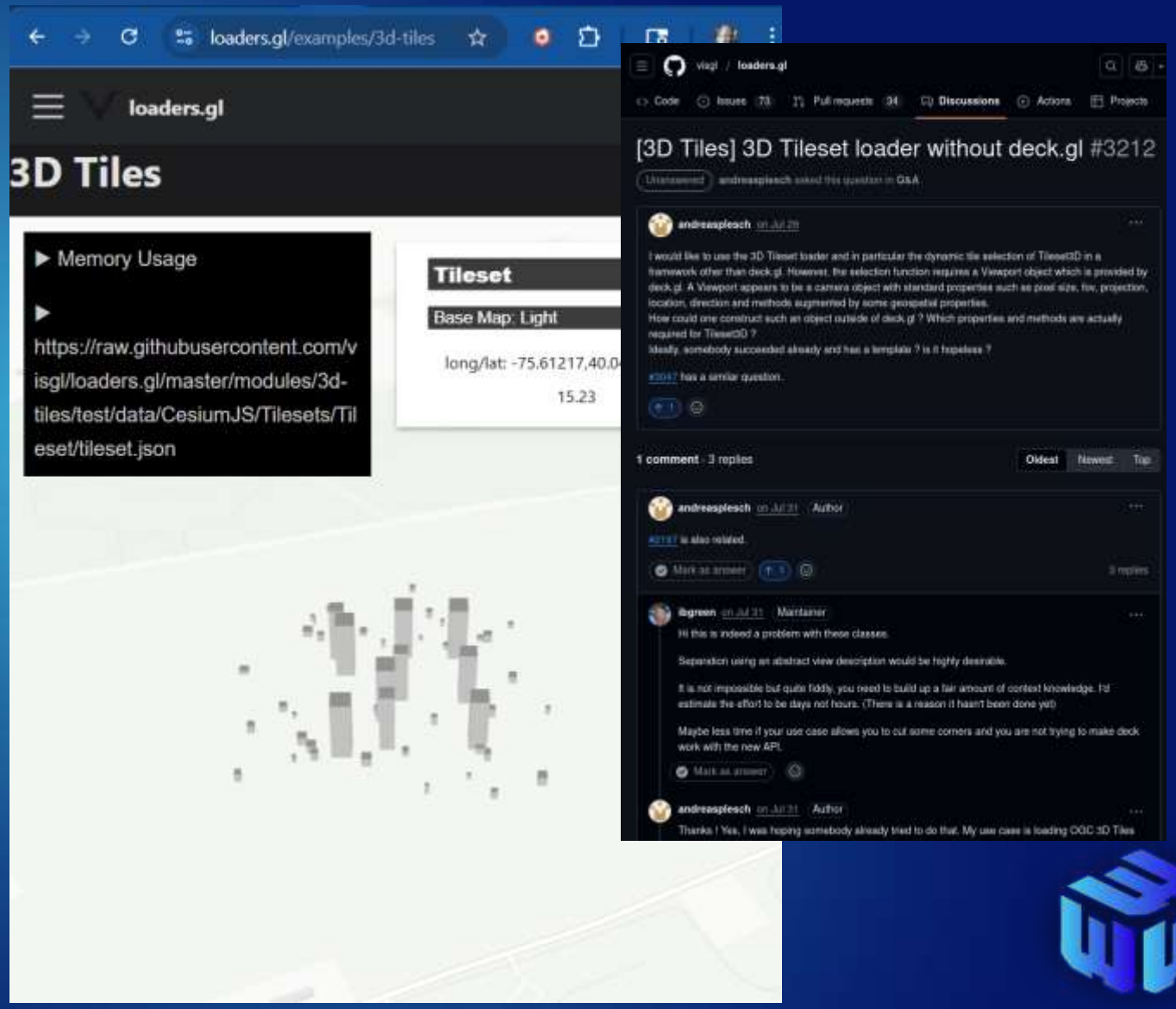
fully supported

Latest Release

On github repo

3D Tiles Code Base for web

- a few js implementations for web browsers
- **CesiumJS** (reference implementations), tile loader and traversal tightly coupled to engine
- loaders.gl : for mapbox, Uber; is derived from older Cesium; map centric for traversal tile selection
- **3DTilesRendererJS** : NASA-AMMOS, Cesium ecosystem grantee, uses threejs
- others (NYTimes, JDUltra)



3D Tiles examples

- comprehensive **collection of 137 test examples** in separate project repo: https://github.com/andreasplesch/3d-tiles-examples/
- new **reference 3D Tiles viewer** in example repo using CesiumJS and 3DTilesRendererJS: https://andreasplesch.github.io/3d-tiles-examples/
- viewer accepts any 3D Tiles url and shows json for quick reference

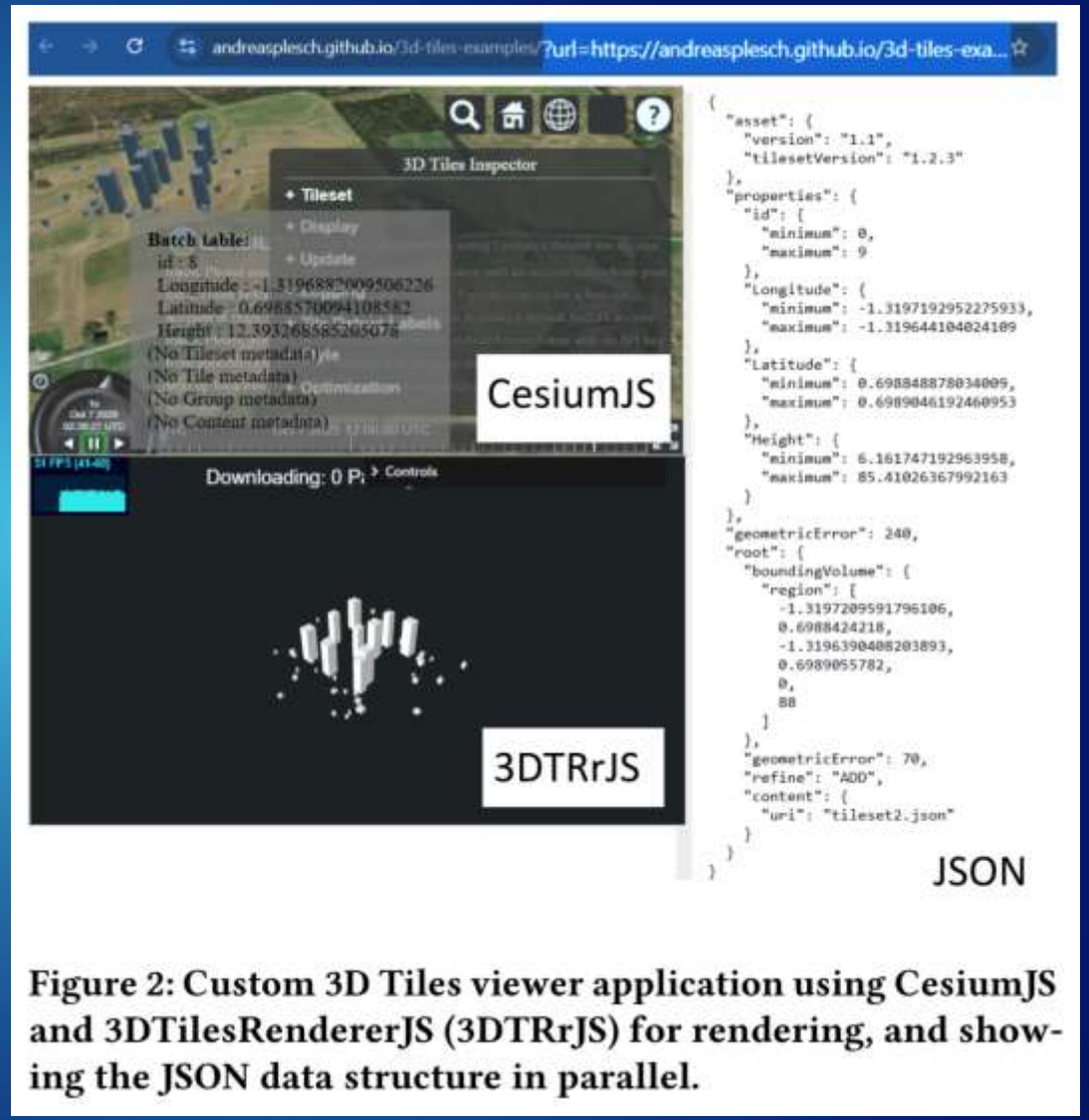


Figure 2: Custom 3D Tiles viewer application using CesiumJS and 3DTilesRendererJS (3DTRrJS) for rendering, and showing the JSON data structure in parallel.

Implementation Design

- **Translation** of 3D-tiles into X3D(OM) nodes
- **Inline** for loading glTF content
- **Transform** for required glTF rotation
- **MatrixTransform** for 3D Tile positioning, **encapsulates** one tile or one tileset
- **GeoTransform** for **GeoOrigin** transform (x3dom specific)
  allows for multiple **child tiles**
- enables **integration in geospatial X3D** including GeoOrigin support

```
3  <X3D profile='Immersive' version='3.3' xmlns:xsd='http://www.w3.org/2001/XMLSchema-
4      <Scene>
5          <GeoTransform globalGeoOrigin="true">
6              <GeoOrigin DEF='GO' geoCoords="40.029 -75.6148 0" rotateYUp="true" />
7              <MatrixTransform>
8                  <Transform rotation='1 0 0 1.5708'>
9                      <Inline url='"https://andreasplesch.github.io/3d-tiles-examples
10                 </Transform>
11             </MatrixTransform>
12         </GeoTransform>
13     </Scene>
14 </X3D>
```

Figure 3: Principal X3D representation of tile content showing glTF loading and appropriate Transform hierarchy.

Node Design

single GeoTileset node

- **url field** for 3D Tiles json
- **geoOrigin field** for potential scene GeoOrigin
- **no geoSystem field** since 3D Tileset content is expected to be in ECEF coordinates (‘GC’)
- internal GeoTransform field to hold constructed node structure
- **maximumScreenSpaceError field** for LOD selection tuning
- no other LOD selection related fields since all defined in 3D Tiles json
- **standard metadata field** to hold any metadata

GeoTileset node

```
<X3D profile='Immersive' version='3.3' xmlns:xsd='http://www.w3.org/2001/XMLSchema-
    <Scene>
        <GeoTransform globalGeoOrigin="true">
            <GeoOrigin DEF='GO' geoCoords="40.029 -75.6148 0" rotateYUp="true" />
            <MatrixTransform>
                <Transform rotation='1 0 0 1.5708'>
                    <Inline url='"https://andreasplesch.github.io/3d-tiles-example
                </Transform>
            </MatrixTransform>
        </GeoTransform>
    </Scene>
</X3D>
```

outer X3D Geospatial Scene

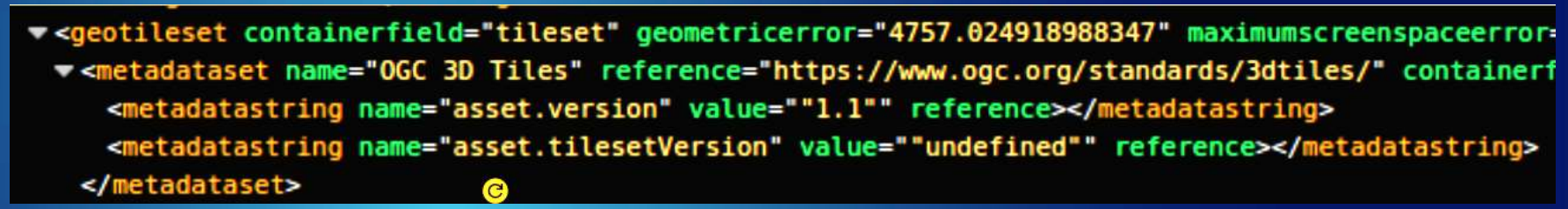
Tile node

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.3//EN" "http://www.web3d.org
<X3D profile='Immersive' version='3.3' xm
instance' xsd:noNamespaceSchemaLocation=
3.0.xsd'>
  <Scene>
    <GeoViewpoint position="40.029 -75.6148 20">
      <GeoOrigin DEF='GO' geoCoords="40.029 -75.6148 0" rotateYUp="true"/>
    </GeoViewpoint>
    <Inline url='"basemap.x3d"'/>
    <GeoTileset url='"./Tilesets/Tileset/tileset.json"'>
      <GeoOrigin USE='GO'/>
    </GeoTileset>
  </Scene>
</X3D>
```

Node Design

single GeoTileset node

- **metadata field** is populated by node with a **Metadataset node** which holds extracted 3D Tiles metadata. The node has a reserved **name “OGC 3D Tiles”**.



Node implementation flow

- use loaders.gl to fetch and parse tileset json
- and to traverse and select tiles depending on view (state)
- expects mapbox type view (state) not gl style view matrix
- loader accepts callbacks called after downloading of content for tiles selected for current view

```
{
    tilesetJsonPromise = this._load(
        this._vf.rootUrl[0], this._Tiles3DLoader, { '3d-tiles': {isTileset: tru
}
var that = this;
tilesetJsonPromise.then(
    function fullfilled ( tilesetJson )
    {
        console.log( tilesetJson );
        const tileset3d = new that._Tileset3D( tilesetJson,
            {
                throttleRequests: false,
                onTileLoad: that._onTileLoad.bind( that )
            });
        let rt = that._nameSpace.doc._x3dElem.runtime;
        const viewportOpts = {
            width: rt.getWidth(),
            height: rt.getHeight(),
            latitude: tileset3d.cartographicCenter[1],
            longitude: tileset3d.cartographicCenter[0],
            pitch: 2, // from vertical
            bearing: 10, // from N ccw
            zoom: 14, // size=360/2^zoom; 360/size=2^zoom; zoom=ln2(360/size);
            // nearZ: 0.59679,
            // farZ: 5967.85292
            // projectionMatrix: rt.projectionMatrix().toGL()
        }
        that.viewport = new that._WebMercatorViewport( viewportOpts );
        tileset3d.update ( that.viewport );
        console.log ( tileset3d );
        return tileset3d.selectTiles ( that.viewport );
    },
```



Node implementation flow

- onLoad callback then constructs X3D node structure and adds it to the parent GeoTileset X3D node
- caching by Loader itself and by x3dom for glTF content management
- other callbacks for deselection or unloading of previously loaded tiles
  - leads to making corresponding X3D node invisible or complete removal depending on memory

x3dom / src / nodes / Geospatial / **GeoTileset.js**

lines (329 loc) - 15.6 KB

```
_onTileLoad : function ( tile )
    const tileCtx =
    {
        doc       : this._ctx.doc,
        runtime   : this._ctx.runtime,
        xmlNode   : this._ctx.xmlNode.cloneNode( true ), // overwritten
        nameSpace : this._ctx.nameSpace
    };

    tileCtx.xmlNode = document.createElement('Transform');
    let glTFTransform = new x3dom.nodeTypes.Transform( tileCtx );
    glTFTransform._vf.rotation = x3dom.fields.Quaternion.parseAxisAngle( "1 0 0 " + Math.PI/2 );
    glTFTransform.fieldChanged("rotation"); //applies to trafo

    tileCtx.xmlNode = document.createElement('Inline');
    let glTFInline = new x3dom.nodeTypes.Inline( tileCtx );
    glTFInline._vf.url = x3dom.fields.MFString.parse( tile.contentUrl );
    glTFInline.nodeChanged(); //is necessary and loads the inline scene

    glTFTransform.addChild( glTFInline );
    glTFTransform.nodeChanged(); //is necessary

    tileCtx.xmlNode = document.createElement('MatrixTransform');
    let tileTransform = new x3dom.nodeTypes.MatrixTransform( tileCtx );
    tileTransform._vf.matrix = x3dom.fields.SFMatrix4f.fromArray( tile.computedTransform ).transpose();
    tileTransform.fieldChanged("matrix"); //applies to trafo
    //tileTransform._trafo.setFromArray ( tile.computedTransform ); //shortcut works but does not update
    tileTransform.addChild( glTFTransform );
    tileTransform.nodeChanged();

    this._geoOriginTransform.addChild( tileTransform );
    this._geoOriginTransform.nodeChanged(); //is necessary and loads the inline scene
```



Summary of supported 3D Tiles features

- 3D Tiles is a large spec. with a lot of functionality
- a substantial subset is supported which allows for many applications
- support is a function of existing glTF extensions support in x3dom and
- existing 3D Tiles feature support in loaders.gl libraries

- glTF 2.0 as the only tile content format
- A list of glTF 2.0 extensions:
  - KHR_mesh_quantization
  - EXT_meshopt_compression
  - KHR_draco_mesh_compression
  - KHR_materials_pbrSpecularGlossiness
  - KHR_materials_unlit
  - KHR_texture_transform
  - KHR_lights_punctual
  - KHR_materials_emissive_strength
  - EXT_texture_webp
  - MSFT_texture_dds
- full traversal of a tile set
- LOD selection through geometric error and tunable maximum screen space error
- REPLACE and ADD refinement strategies
- tile sets of tile sets
- tile sets with empty roots
- implicit tile sets with quadtrees and octrees

DEMO

https://earthstruct.github.io/x3dom/test/functional/3dtiles/tests.html



Figure 5: The Bay area in the Google Photorealistic 3D Tileset rendered together with geospatially positioned, and animated X3D and Web3D logos loaded from the X3D example archive (https://www.web3d.org/x3d/content/examples/Basic/) demonstrating full integration. Access to the Tileset requires a key. The information panel shows current heading, pitch, zoom level, loaded tiles and memory usage.



Thank You!

Web3D 2026

Doha, Qatar

