I3S & glTF in Geospatial

Tamrat Belayneh
tbelayneh@esri.com
Snr. Software Engineer / Principal Technology Evangelist
@esri
Indexed 3D Scene Layers (I3S)

I3S is an OGC Community Standard
http://www.opengeospatial.org/standards/i3s

The open community GitHub version of this Standard:
https://github.com/Esri/i3s-spec
I3S | Indexed 3D Scene Layers  
Support multiple Profiles and Layer Types

3D Objects
- 3D Shapes
- Trees
- Buildings
- Infrastructure

Point Scene Layer
- Point locations
- Symbolize with 3D Object Styling
- Can visualize attributes by size and color

Integrated Mesh Scene Layer
- Skin of the Earth
- Textured with imagery
- Captured by Drone2Map or other methods

Point Cloud Scene Layer
- LiDAR
- Photogrammetric points
- Style by elevation, color, classification

Building Scene Layer
- Detailed building models
- Typically Sourced from format such as Revit/IFC
- Supports Filter categories and floors

OGC I3S Version 1.2

OGC I3S Version 1.3
OGC I3S version 1.2 brought increased Performance for 3D Object and Integrated Mesh Scene layers:

- Better material support – gltf feature compatible
- Paged Node access pattern
- More compact Geometry – supports using Draco compression for the binary geometry content
- Kb2 (basis) compressed texture support
- More optimal Selection Strategy - standardizes on Oriented Bounding Boxes (OBBs) for node selection
A major improvement in OGC I3S 1.2 is node paging

Previously, clients received one node per request, starting I3S 1.2 nodes may be grouped into pages (64 by default)
I3S | Enhancements

Geometry Compression

- Support for Draco geometry compression
  https://github.com/google/draco

- Draco an open source (Apache 2.0) library for compressing 3D geometries

- Draco supports compressing geometry attributes creating more compact nodes, which in turn provides a smaller payload, increasing performance
A geometry payload that’s up to ~85% smaller compared to OGC I3S 1.1 CS

<table>
<thead>
<tr>
<th>Series</th>
<th>Test</th>
<th>I3S 1.1</th>
<th>I3S 1.2</th>
<th>Improvement (reduction) Ratio of I3S 1.2 Over I3S 1.1</th>
<th>No of Requests</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IM_Vricon_Textured_SP_BK_1</td>
<td>61.00</td>
<td>56.00</td>
<td>85.52%</td>
<td>4.27%</td>
</tr>
<tr>
<td>2</td>
<td>IM_Vricon_Textured_SP_BK_2</td>
<td>81.00</td>
<td>60.00</td>
<td>95.13%</td>
<td>14.89%</td>
</tr>
<tr>
<td>3</td>
<td>IM_Vricon_Textured_SP_BK_3</td>
<td>93.00</td>
<td>64.00</td>
<td>95.53%</td>
<td>18.47%</td>
</tr>
<tr>
<td>4</td>
<td>IM_Vricon_Textured_SP_BK_4</td>
<td>215.00</td>
<td>173.00</td>
<td>91.99%</td>
<td>10.82%</td>
</tr>
<tr>
<td>5</td>
<td>IM_Vricon_Textured_SP_BK_5</td>
<td>307.00</td>
<td>259.00</td>
<td>90.98%</td>
<td>8.48%</td>
</tr>
<tr>
<td>6</td>
<td>3D_Object_UnTextured_BK_1</td>
<td>73.00</td>
<td>73.00</td>
<td>87.08%</td>
<td>0.00%</td>
</tr>
<tr>
<td>7</td>
<td>3D_Object_UnTextured_BK_2</td>
<td>100.00</td>
<td>90.00</td>
<td>88.55%</td>
<td>5.26%</td>
</tr>
<tr>
<td>8</td>
<td>3D_Object_UnTextured_BK_3</td>
<td>120.00</td>
<td>102.00</td>
<td>89.05%</td>
<td>8.11%</td>
</tr>
<tr>
<td>9</td>
<td>3D_Object_UnTextured_BK_4</td>
<td>156.00</td>
<td>120.00</td>
<td>90.54%</td>
<td>13.04%</td>
</tr>
<tr>
<td>10</td>
<td>3D_Object_UnTextured_BK_5</td>
<td>193.00</td>
<td>130.00</td>
<td>92.00%</td>
<td>19.50%</td>
</tr>
<tr>
<td>11</td>
<td>3D_Object_Textured_SF_BK_1</td>
<td>14.00</td>
<td>11.00</td>
<td>87.42%</td>
<td>12.00%</td>
</tr>
<tr>
<td>12</td>
<td>3D_Object_Textured_SF_BK_2</td>
<td>32.00</td>
<td>27.00</td>
<td>86.29%</td>
<td>8.47%</td>
</tr>
<tr>
<td>13</td>
<td>3D_Object_Textured_SF_BK_3</td>
<td>53.00</td>
<td>45.00</td>
<td>83.71%</td>
<td>8.16%</td>
</tr>
<tr>
<td>14</td>
<td>3D_Object_Textured_SF_BK_4</td>
<td>85.00</td>
<td>68.00</td>
<td>83.04%</td>
<td>11.11%</td>
</tr>
</tbody>
</table>
I3S | Enhancements

A living, breathing specification
I3S - Texture support including various compressions

Some empirical Testing results:

- ETC2 textures are generally sized somewhere between 4-6x that of the JPG (similar to DDS)
- Depending on the texture size (square textures for example) ETC2 textures might be bigger than DDS (gzip compressed)
GPU Compressed Textures

GPUs can directly process compressed textures in memory if held in specialized block-based compressed formats designed for
Fast Decoding Speed and Random Access
Reduces GPU memory size and bandwidth = faster rendering

Encoding to GPU compressed formats is compute-intensive and so converting uncompressed images, JPGs or PNGs to GPU compressed textures at runtime is typically not feasible

So, why not just use GPU compressed textures in glTF 3D asset files...
...because of industry fragmentation!

---

| GPU Compressed Texture Format Fragmentation |
| Multiple GPU Compressed Texture formats with varying levels of support across diverse platforms |

<table>
<thead>
<tr>
<th></th>
<th>AMD</th>
<th>Apple A12</th>
<th>Apple M1</th>
<th>ARM</th>
<th>Intel</th>
<th>NVIDIA Desktop</th>
<th>NVIDIA Tegra</th>
<th>Qualcomm</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASTC</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>BC1, BC3</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>BC7</td>
<td>☑️</td>
<td>☑️</td>
<td>☑️</td>
<td>✗</td>
<td>☑️</td>
<td>✓</td>
<td>☑️</td>
<td>✗</td>
</tr>
<tr>
<td>ETC1</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>☑️</td>
</tr>
<tr>
<td>ETC2</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>☑️</td>
</tr>
<tr>
<td>PVRTC1</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>☑️</td>
<td>☑️</td>
<td>☑️</td>
<td>☑️</td>
<td>☑️</td>
</tr>
</tbody>
</table>

---

Source: The Khronos Group. Ktx2.0 Launch
Esri is a member of the Open Geospatial Consortium (OGC®), and has been delivering geospatial solutions for over 5 decades, investing significant brain power to solve one vexing problem in the geospatial world: making massive geospatial content available across platforms and devices, including web browsers, mobile devices, and desktop applications. In response to that challenge, the Indexed 3D Scene Layer (I3S) specification was created about a decade ago. Esri worked with the community to release the specification under an Apache license in 2015. The OGC adopted I3S in 2017 as its first 3D streaming Community Standard.

I3S was created to enable streaming and storage of massive 3D geographic data. An I3S dataset, called a Scene Layer, can contain millions of discrete 3D objects, each with their own attributes, integrated surface meshes, point cloud data, etc. I3S allows various types of geospatial content to be efficiently encoded and transmitted as a single scene in order to create an interactive visualization experience that is accessible across a wide variety of devices. Recently, OGC adopted the latest version of I3S, OGC I3S 1.2, which incorporates advanced physically based materials that are feature-compatible with the Khronos® glTF™ 3D asset format standard, and supports referencing of glTF 3D models in I3S Scene Layers.

I3S Scene Layers often contain significant amounts of texture data, consuming large amounts of storage and processing power. Their size makes it a challenge to maintain desired levels of interactivity. High resolution texture files are integral to the realism and legibility of a 3D object - but uncompressed, they're huge. Textures using JPEG or PNG compression are typically software-decoded, eating CPU cycles and resulting in slow loading. Additionally, they must be held as large

KTX 2.0 Universal Textures in OGC’s I3S v1.2: Puts the Whole World in Your Hands

February 22, 2022 by Tamrat Belayneh (Senior software engineer at Esri) and the Khronos 3D Formats Working Group

KTX 2.0 Universal Textures using Basis Universal Supercompression

KTX 2.0, version 1.2, adds support for KTX 2.0 textures, together representing a win on every front: memory consumption, image quality, and encoding time. A Scene Layer using KTX 2.0 loads memory and power, and is practically identical in terms of image quality.
I3S | Esri-Binomial Collaboration

- Reduce the encoding and supercompressing speed of the binomial library
- Create a ready to use web/c++ API to output and consume basis in Ktx2 format

Basis encoding Version 1.12 compared to Version 1.13: Plots Quality (measured as a function of a Y-PSNR (db), Encoding times (seconds) and Bits Per Texel (basis file size in bits/Texel Count*Num Channels) – where Texel count is texture.width*texture.height for each of the 7 use cases and the basis encoder setting for quality was set to 128 and compression level to 1. Note the difference between Basis Ver. 1.12 Encoding Time (in seconds – orange bars) vs Basis Ver 1.13 Encoding Time using the Optimized code base Version 1.13 (In seconds – yellow bars).
I3S | Esri-Binomial Collaboration

<table>
<thead>
<tr>
<th>Image_name</th>
<th>Encoding Time (Secs) Basis 1.13 (SSE 4.1: 1, Multithreading: 1 - comp_level 1)</th>
<th>Encoding Time (Secs) Basis 1.13 (SSE 4.1: 1, Multithreading: 1 - comp_level 0)</th>
<th>Encoding Time (Secs) Basis 1.12 (Multithreading: 1 - comp_level 1)</th>
<th>Improvement Factor (Using Comp_level_1)</th>
<th>Improvement Factor (Using Comp_level_0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IM_Frankfurt_4K</td>
<td>5.12</td>
<td>4.62</td>
<td>14.23</td>
<td>2.78</td>
<td>3.08</td>
</tr>
<tr>
<td>IM_1K</td>
<td>0.30</td>
<td>0.26</td>
<td>0.91</td>
<td>3.01</td>
<td>3.57</td>
</tr>
<tr>
<td>IM_NYC_2k</td>
<td>1.48</td>
<td>1.33</td>
<td>5.22</td>
<td>3.51</td>
<td>3.91</td>
</tr>
<tr>
<td>3D_Object_SanFran_4k</td>
<td>8.68</td>
<td>6.45</td>
<td>23.46</td>
<td>2.70</td>
<td>3.64</td>
</tr>
<tr>
<td>IM_chateau_1k</td>
<td>0.72</td>
<td>0.63</td>
<td>1.96</td>
<td>2.72</td>
<td>3.13</td>
</tr>
<tr>
<td>IM_Rancho_IM_4k</td>
<td>6.46</td>
<td>5.82</td>
<td>23.23</td>
<td>3.60</td>
<td>3.99</td>
</tr>
<tr>
<td>3D_Object_Singapore_2k</td>
<td>1.08</td>
<td>0.89</td>
<td>3.36</td>
<td>3.11</td>
<td>3.78</td>
</tr>
</tbody>
</table>

Compares the Encoding Optimization gained for various I3S datasets: Basis 1.13 added a new geospatial focused compression level 0.
I3S | Basis in KTX 2.0 Container Supported

- Table 4 compares the encoding optimization gained for the texture resources of an I3S dataset by leveraging the GPU using OpenCL™ to orchestrate parallel computation as well as CPU improvements gained in version 1.16 vs. 1.15 of the Basis Universal encoder libraries.

1. Quadro RTX 5000 @ 1.545 GHz, 1545 Mhz, 3072 CUDA Cores, Memory data rate: 14.00 Gbps, Memory Interface: 256-bit, Memory bandwidth: 448.06 GB/s, Total available graphics memory: 81779 MB, Dedicated video memory: 16384 MB GDDR6, Driver version: 472.42

2. Quadro RTX A6000 @ 1.800 GHz, 1800 Mhz, 10752 CUDA Cores, Memory data rate: 16.00 Gbps, Memory Interface: 384-bit, Memory bandwidth: 768.106 GB/s, Total available graphics memory: 81826 MB, Dedicated video memory: 49140 MB GDDR6, Driver version: 462.31

3. Intel(R) Xeon(R) W-10885M CPU @ 2.40GHz, 2400 Mhz, 8 Core(s), 16 Logical Processor(s), Total Physical Memory: 128 GB, running Windows 10 Pro for Workstations 64-bit

4. AMD Ryzen Threadripper PRO 3995WX 64 Cores, 2695 Mhz, 64 Core(s), 128 Logical Processor(s), Total Physical Memory: 128 GB, running Windows 10 Pro for Workstations 64-bit

<table>
<thead>
<tr>
<th>GPU</th>
<th>Texture assets (count)</th>
<th>Factor over v1.15 using parallel GPU Encoding time (Secs.)</th>
<th>Factor over v1.15 using sequential GPU Encoding time (Secs.)</th>
<th>Factor over v1.15 using parallel CPU Encoding time (Secs.)</th>
<th>v1.15 encoding time using parallel CPU (Secs.)</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quadro RTX 5000¹</td>
<td>1,126</td>
<td>9.9X</td>
<td>2.0X</td>
<td>1.8X</td>
<td>1,515</td>
<td>Intel(R) Xeon(R) W-10885M³</td>
</tr>
<tr>
<td>RTX A6000²</td>
<td>10,121</td>
<td>4.5X</td>
<td>2.2X</td>
<td>2.2X</td>
<td>15,648</td>
<td>AMD Ryzen Threadripper PRO 3995W⁴</td>
</tr>
</tbody>
</table>
I3S | Support in the community

- The Vis.gl open framework now has support for I3S consumption (IM & 3D Object layers) using deck.gl & loaders.gl modules.

Vis.gl
Frameworks for WebGL-powered large-scale data visualization

https://loaders.gl/examples/i3s
I3S | loaders.gl natively supports consuming I3S 1.6 and I3S 1.7

- Loaders.gl & deck.gl natively supporting consumption of 3D Object and IntegratedMesh layers in I3S 1.6 & 1.7 format

https://loaders.gl/examples/i3s
I3S | loaders.gl natively supports consuming Building Scene Layers (OGC I3S 1.3/I3S 1.8)

- Loaders.gl & deck.gl natively supporting consumption of Building Scene Layer in I3S 1.8 (OGC I3S 1.3) format

Using Loader.gl sample application

Code sample
I3S | Loaders.gl has advanced (semantic validation) + Visual Debugger

- Add I3S validators to perform not only syntactic level checking of nodes but also Visual debugging.

Vertex Normals

Bounding box debug views
I3S | 3DTiles ⇔ I3S Conversion

- A loaders.gl based converter between I3S and 3D tiles tilesets.

- Supports the following types of conversion (2-way)
  - 3DTiles Batched3DModel ⇔ I3S IntegratedMesh Scene Layer
  - 3DTiles Batched3DModel + Hierarchy extension ⇔ I3S 3D Object Scene Layer

- Ability to batch convert a 3D Tiles content to to Scene Layer Package (SLPK) and Indexed 3D Scene Layer REST (i3sREST)
  - Directly from cesium ion - via its REST api
  - From a file-based 3D Tiles content
I3S | 3DTiles ⇔ I3S Conversion cont’d

- Ability to batch convert an I3S OGC 1.3 in Scene Layer Package (SLPK) and Indexed 3D Scene Layer REST (i3sREST) format to 3D Tiles

- Support for: 3DTiles version 1.0 and 3D Tiles Version 1.1 (3D Tiles Next) (with glTF version 2.0)
  - Has experimental support for glTF 1.0 as it still seems prevalent in the marketplace

- Coordinates: supports WGS84 as that’s the only supported 3D Tiles CRS

- Handles correct gravity models by converting between orthometric to ellipsoidal
I3S | Support of I3S 1.7 in Cesium.JS

- Develop capability for reading and viewing I3S leveraging CesiumJS framework. Supporting IntegratedMesh and 3D Object Scene Layers in I3S OGC 1.3 format.

- Modeled after ArcGISTiledElevationTerrainProvider module that consumes arcgis tiled elevation layers.

- Much thanks to Cesium for reviewing, approving & merging of the PR.

- Sandcastle example is available @ https://sandcastle.cesium.com/ (Select DataSources in the Gallery, and you’ll find 3D Object, IntegratedMesh and 3D Object Feature Picking examples).

- A blog detailing sample usage and framework architecture - OGC Indexed 3D Scene Layers (I3S) Community Standard is now supported in CesiumJS.
Support of I3S 1.7 in Cesium.JS

A 3D Object Scene Layer loaded in cesium.js
Building Scene Layer

- Detailed building models
- Typically Sourced from format such as Revit/IFC
- Supports Filter categories and floors
Building Scene Layers are visually complex. Filtering elements or parts of a building to see only the relevant or important information is very useful.

A building filter enables making elements of a building visible as solid or wireframe. Each building filter contains one or more filter blocks. Filter blocks define what is visible in the BSL.

Complex Building Model

With a filter applied

Filtered for selected plumbing only
BIM Data in Geo Context

e.g. for visualization or facility management
Image and Geometry Compression in Geospatial Live Demo
glTF helps to Address

- Consistency of assets between ArcGIS & 3rd-party applications.
- Materials from 3rd Party tools (i.e. Sketchup, Blender, etc.) import into ArcGIS with inconsistencies (aesthetically look different).
- Consistency of assets between ArcGIS & 3rd-party marketplaces.
- Models from popular 3D marketplaces (i.e. Sketchfab) import into ArcGIS with inconsistencies (aesthetically look different).
- Consistency of AutoCAD 3D models between ArcGIS & 3rd-party applications.
- 3D AutoCAD assets import into ArcGIS with inconsistencies (aesthetically look different).
gltf & 3D GIS | Servers as a glue

General Workflow
<table>
<thead>
<tr>
<th>3D GIS Specialist</th>
<th>City Planner (Graphics staff)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Visualization Specialist</td>
<td>Urban Designer</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GIS Technologist</th>
<th>Architectural Designer</th>
<th>Game Developer</th>
<th>Visual Database Engineer</th>
<th>Modeling / Simulation Analyst</th>
<th>Research Scientist (CFD)</th>
<th>GeoSpatial Intelligence Analyst</th>
<th>3D Modeler</th>
</tr>
</thead>
<tbody>
<tr>
<td>GIS Developer</td>
<td>Architectural 3D Modeler</td>
<td>Game Artist</td>
<td>Modeling/Simulation Developer</td>
<td>Synthetic Environment Designer</td>
<td>Transportation Modeler</td>
<td>3D Artist</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cartographer</th>
<th>Civil Engineer</th>
<th>Environmental Planner</th>
<th>DoD Analyst</th>
<th>Thermal Researcher</th>
<th>Traffic Engineer</th>
</tr>
</thead>
<tbody>
<tr>
<td>GIS Analyst</td>
<td>Plant Designer</td>
<td>BIM Specialist</td>
<td>Army GeoSpatial Engineer</td>
<td>Flood Modeller</td>
<td>Traffic Simulation Specialist</td>
</tr>
</tbody>
</table>
Citizen Engagement
Collaborative planning
Use game engines to enable immersive geospatial experiences

Geospatial Data
- Integrated Mesh
- Imagery and Terrain
- 3D Objects

Developer Tools
- Unreal Engine
- Unity

Effects
- Atmosphere
- Renderers
- Animation

Analysis
- Simulation
- Measurement

XR
- Virtual Reality
- Mixed Reality
- Augmented Reality
ArcGIS Maps SDK

Production releases of ArcGIS Maps SDK for Unity and ArcGIS Maps SDK for Unreal Engine (version 1.0) are now available.

- Delivered as plugins
  - ArcGIS Maps SDK for Unity
  - ArcGIS Maps SDK for Unreal Engine

- UI and APIs to access ArcGIS services and local data

- Supports local and global 3D experiences