OGC® DOCUMENT: 22-035

External identifier of this OGC® document: http://www.opengis.net/doc/PER/T18-D026

Open Geospatial Consortium

TESTBED-18: 3D+ DATA STREAMING ENGINEERING REPORT

ENGINEERING REPORT

PUBLISHED

Submission Date: 2023-05-18
Approval Date: 2023-06-08
Publication Date: 2023-09-01
Editor: Jérôme Jacovella-St-Louis

Note

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

I KEYWORDS

The following are keywords to be used by search engines and document catalogues.

3D, space, streaming, GeoPose, moving features, orbit, OGC API, glTF, 3D Tiles, BCRS, NOVAS, SOFA, Solar System, ISS, Voyager, special relativity, general relativity

II ABSTRACT

This OGC Testbed 18 *3D Plus Data Standards and Streaming* Engineering Report (ER) reviews existing specifications that support interoperable descriptions of orbital and non-orbital space-based assets, objects, and observations as well as terrestrial observations. The ER suggests a framework consolidating these specifications as a foundation for modeling, representation, and serialization from space-based assets operating at any location in our solar system (3D+ data). This framework enables the streaming of 3D+ data to visualization devices (displays, AR, VR) for presentation.

III SUMMARY

This Engineering Report presents the results of OGC Testbed 18 research performed as a component of the *3D+ Data Standards and Streaming* thread. A review of existing standards pertaining to the efficient streaming and visualization of spatiotemporal 3D and 4D data was performed. This review also considered use cases in Space, including objects referenced relative to a spherical body other than Earth, in orbit, or in free flight within the Solar System, the Milky Way, or outside our galaxy.

The relevant standards and candidate standards identified include multiple *OGC APIs*, *Khronos® glTF™* and *OGC GeoPose*. The *OGC GeoPose* and *OGC API — Moving Features* were reviewed, resulting in some recommendations to those standards. Existing coordinate systems for referencing objects in Space were reviewed and conversions were performed between the Barycentric Celestial Reference System (BCRS), the Geocentric Celestial Reference System (GCRS), Ecliptic coordinate systems (heliocentric and geocentric), the International Terrestrial Reference System (ITRS), and GeoPose. The NOVAS C library was used for these conversions and for developing a Solar System visual simulation. The demonstration also tracked and displayed the International Space Station (ISS) as well as the Voyager 1 space probe, integrating glTF™ models provided by NASA and the Gaia Sky in Colour from ESA. The applicability of the theory of special and general relativity to potential scenarios was explored. The concept of time dilation, the Lorentz factor, and Schwarzschild's exact solution to Einstein's field

equations describing time dilation resulting from both velocity and gravitational fields are briefly introduced.

In addition to the recommendations specific to particular OGC standards that were provided, as an outcome of this work participants recommend the following for the OGC community:

- learn more about the different inertial celestial reference systems standardized by the International Astronomical Union (IAU), such as the BCRS and GCRS, and define entries for them within the OGC CRS register under the IAU namespace (which currently only includes planetary CRSs);
- perform additional practical experiments using software sanctioned by the IAU to manipulate these coordinate systems;
- aim to better understand how to use these libraries correctly; and
- validate results across different implementations, particularly in terms of obtaining high precision results and taking into consideration relativistic effects.

1

INTRODUCTION

1 INTRODUCTION

This OGC Testbed 18 Engineering Report presents the result of research performed as a component of the *3D+ Data Standards and Streaming* thread. A review of existing standards pertaining to the efficient streaming and visualization of spatiotemporal 3D and 4D data was performed. Use cases in space, including objects referenced relative to a spherical body other than Earth, in orbit, or in free flight within the Solar System, the Milky Way, or outside our galaxy were also considered.

The relevant standards and candidate standards identified include the following.

- Multiple OGC APIs
  - *OGC API — 3D GeoVolumes* for streaming 3D content referenced to a spherical body
  - *OGC API — Tiles* for streaming arbitrary content referenced to a spherical body
  - *OGC API — Moving Features* for streaming the changing positions and properties of objects over time
  - *OGC API — Connected Systems* for streaming the positions of many objects or of objects moving rapidly
- The *OGC 2D Tile Matrix Set and Tileset Metadata* as the foundation for *OGC API — Tiles* as well as the fixed tiling of 3D space requirements class of *3D GeoVolumes*
- *OGC 3D Tiles* and *Khronos® glTF™* as payload for 3D content
- *Ericsson Texture Compression 2* (ETC2), *Khronos® KTX* texture formats, *PNG* and *JPEG* to encode textures
- *OGC GeoPose* as a convenient way to encode both position and orientation information
- *ISO 19111 (OGC Abstract Topic 2)* as the foundation for referencing by coordinates

As part of reviewing the GeoPose standard, issues were documented in the GeoPose GitHub repository. The relevant capabilities that would greatly improve the usability of the standard only exist in *advanced targets* of the standard which were found to be cumbersome and not a natural extension of the core capabilities of the *simple target* or of the JSON encoding itself. These capabilities include associating time stamps and identifiers with an individual pose, defining sequences of multiple GeoPoses, and defining a pose relative to a parent pose. A recommendation was made to consider defining these capabilities as extensions preserving the basic form of the simple target. Two space scenarios illustrate what this could look like in a JSON encoding. In one scenario, passengers are onboard a spaceship that is in free flight from Earth to Mars, and in the other, passengers are onboard an express train on the surface of Mars.

As part of reviewing the *OGC API — Moving Features* standard, issues were documented in the GitHub repository, including clarifying the possible synergy with the *OGC GeoPose* and *OGC API — Connected Systems*, and use cases for high performance queries.

The research also reviewed existing coordinate systems for referencing objects in Space and experimented with conversions between:

- the equatorial Barycentric Celestial Reference System (BCRS) and Geocentric Celestial Reference System (GCRS);
- ecliptic coordinate systems (heliocentric and geocentric);
- the International Terrestrial Reference System (ITRS) which forms the basis of the EPSG:4978 (Cartesian) and EPSG:4979 (ellipsoidal) 3D coordinate systems; and
- GeoPose (WGS84 / Local Tangent Plane — East/North/Up).

For several of these conversions, as well as for calculating the positions of the Sun, Earth, Moon, and planets in our Solar System at a given time, the NOVAS C library was used. A table summarizing the functions of the library used in this experiment is presented. The SOFA library is also mentioned as a potential alternative. The BCRS is described as the *global standard reference system for objects located outside the gravitational vicinity of Earth* on Wikipedia.

glTF™ models provided by NASA as well as the Gaia Sky in Colour from ESA were used to provide a situational awareness in space reflecting the camera position and orientation in a visual demonstration. A scenario tracking the position of the International Space Station (ISS) from a sequence of positions returned by a Web API was also demonstrated, using a detailed glTF™ model of the space station provided by NASA. Another scenario which displays and tracks the position of the Voyager 1 space probe along its journey across our Solar System since 1977 illustrates a use case extending deeper into space, again using a NASA glTF™ model as well as historical data.

Finally, the applicability of the theory of special and general relativity to potential scenarios is explored, also noting cases where it may not be applicable. The concept of time dilation is introduced and illustrated with the thought experiment of a light clock, from which the Lorentz factor is derived simply with the well-understood Pythagoras theorem. The scenario of Global Positioning Systems (GPS) is discussed, where the velocity and gravitational fields have opposite effects on time dilation for the on-board clocks compared to clocks on Earth. Schwarzschild's exact solution to Einstein's field equations is also presented, describing with a simple equation time dilation resulting from both velocity and gravitational fields.

2

TERMS, DEFINITIONS AND ABBREVIATED TERMS

2 TERMS, DEFINITIONS AND ABBREVIATED TERMS

No terms and definitions are listed in this document.

2.1. Abbreviated terms

API	Application Programming Interface
AR	Augmented Reality
AU	Astronomical Unit (149,597,870,700 meters, the distance from the Sun to Earth’s orbit)
BCRS	Barycentric Celestial Reference System
BCT	Barycentric Coordinate Time
BVH	Bounding Volume Hierarchy
CRS	Coordinate Reference System
ENU	East, North, Up (axes)
ER	Engineering Report
ESA	European Space Agency
ETC2	Ericsson Texture Compression 2
GCRS	Geocentric Celestial Reference System
GPS	Global Positioning System
glTF™	GL Transmission Format
IAU	International Astronomical Union
ISS	International Space Station
ITRF	International Terrestrial Reference Frame
ITRS	International Terrestrial Reference System
KTX	Khronos Texture

LTP	Local Tangent Plane
NASA	National Aeronautics and Space Administration
NOMAD	Naval Observatory Merged Astrometric Dataset
NOVAS	Naval Observatory Vector Astrometry Subroutines
OGC	Open Geospatial Consortium
SOFA	Standards of Fundamental Astronomy
VR	Virtual Reality
2DTMS	2D Tile Matrix Set

3

STREAMING STATIC DATA REFERENCED TO SPHERICAL BODY

3 STREAMING STATIC DATA REFERENCED TO SPHERICAL BODY

The majority of data collected concerns observations and measurements on or near the surface of a spherical body, in particular planet Earth. A large portion of that data is also static. This static data may provide useful situational context for visualizing both other static content, as well as tracking dynamic objects (e.g., visualizing the position of a satellite in orbit above satellite imagery of the Earth). This section explores data access specifications supporting the interoperable selection and transmission of such static information referenced to a particular spherical body, including Earth, but also other spherical bodies such as the Moon or Mars.

3.1. Streaming data with OGC API — 3D GeoVolumes

The *OGC API — 3D GeoVolumes* (draft) Core conformance class defines a mechanism to retrieve 3D content of different resolutions for flexible bounding volumes hierarchies, encoded using OGC specifications such as 3D Tiles or I3S.

Another conformance class supports the retrieval of data based on an implicit tiling scheme, such as a 2D Tile Matrix Set extended to three or more dimensions, based on the method described in the informative annex J of the 2D TileMatrix Set and TileSet Metadata Standard (*2DTMS & TSMD*).

3.2. Flexible Bounding Volume Hierarchy

Defining a flexible bounding volume hierarchy enables equal distribution of the amount of data in different tiles based on the data density at different locations. In this case, space partitioning depends upon the data. A disadvantage of this approach compared to implicitly defining the space occupied by tiles (partitioning the space irrespective of its content) is that it is then not possible to determine which tiles cover a pre-determined space volume, such as a view frustum (the region of space in the modeled world visible on the display device).

3.3. Implicit Tiles

With the implicit tiling scheme for 3D content in GeoVolumes, the orientation and positioning of the content for a particular tile of data is implied from the tile identifier, and a global multi-resolution tile set can be described using the TileSet metadata. The informative description from the *2DTMS & TSMD* annex J to extend this TileSet metadata to additional dimensions will be standardized in the draft GeoVolumes standard for describing 3D content.

Three different approaches are supported as to how to handle the vertical dimensions:

- extending the vertical dimension infinitely from the center of the spherical body to infinity outwards;
- sub-dividing the vertical dimension based in fixed number of divisions at all resolution levels; and
- progressively sub-dividing the vertical dimension into more divisions at higher resolution levels.

The OGC 3D Tiles Next Community Standard introduces support for implicit tilesets. However, the 3D Tiles implicit tiling mechanism is not fully aligned with the approach proposed in the 2D Tile Matrix Set and TileSet Metadata standard. This is because an implementation of 3D Tiles cannot express variable width tile matrix capabilities, still relies on complex hierarchies of tileset metadata for explicitly declaring tiles availability, and does not follow the convention of aligning local East, North, Upwards (ENU) axes with a plane tangent to the centroid of a tile. However, it is possible to define a 1.0 or 1.1 3D Tiles tileset declaring those transformations, so that the same 3D models whose local coordinate system is aligned with the ENU axes can be used with both the implicit tiling approach following a TileMatrixSet (including variable widths), as well as a TileSet defining a flexible Bounding Volume Hierarchy.

3.4. Batched 3D Models vs. points referencing 3D models

Two common approaches exist for storing detailed 3D model content within tiles. One approach is to encode the entire content of each tile as a single 3D model object. This improves streaming performance by avoiding several round-trips to the server when requesting a large number of models. Another approach is to encode for each tile a reference point for everywhere a model is positioned. This reference point can additionally specify an optional orientation and/or scaling parameter to transform the model. This second approach is most useful for instantiating the same model many times such as building a forest from just a few tree models. Another advantage is that these reference points can easily be clamped to the terrain elevation, including when multiple resolutions of terrain are used. This ensures that the models will always be positioned on the ground. However, this clamping mechanism may also be implemented with the first approach if separate object nodes are used within a higher-level object node.

Figure 1 — San Diego CDB from CAE, Elevation Model from Viewfinder Panoramas (SRTM)

3.5. Multi-resolution content positioned on spherical bodies

Any large data set covering a large surface of a spherical body would benefit from a streaming mechanism supporting the selection of a resolution of interest. This applies to vector features, satellite imagery, point features, other type of gridded data (including tiled gridded elevation coverage data for digital terrain models), as well as 3D models, for which a more generalized overview is preferable when seen at a scale where smaller details would not be relevant to the current visualization or analysis.

Most of the OGC API data access standards (draft and approved) already include core capabilities or define (or plan to define) optional conformance classes supporting the selection of a spatiotemporal subset and lower resolution of a particular collection of geospatial data. As an example, visualizing a whole hemisphere of Mars as seen from far away will require a lower resolution than visualizing the view from a rover on its surface.

Figure 2 — Mars glTF model from NASA, Gaia Sky in Colour from ESA

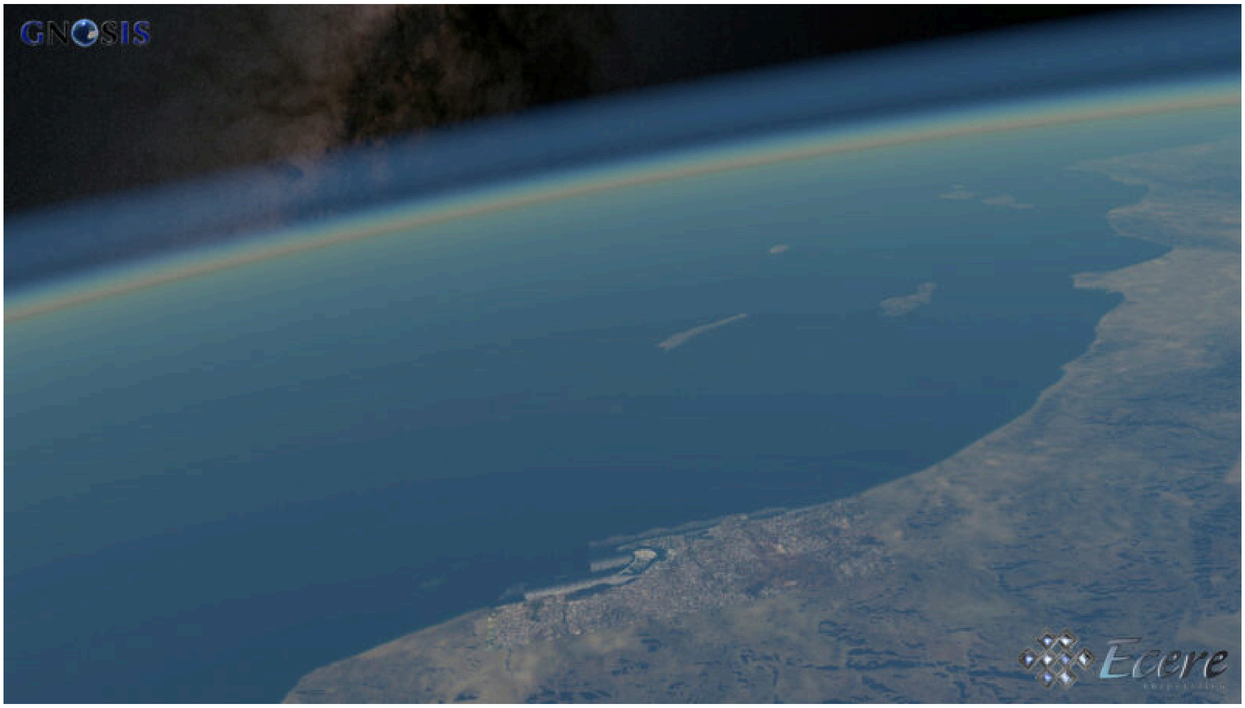

Figure 3 — Blue Marble Next Generation from NASA,
San Diego CDB from CAE, Gaia Sky in Colour from ESA

4

STREAMING 3D MODELS

4 STREAMING 3D MODELS

The ability to stream 3D models enables a number of visualization use cases. Models can either be unique and specific to the geospatial location where they are used, or shared and re-used to represent objects in different positions. They can range from small objects such as people or cars, to medium and larger objects such as trees, buildings, or satellites, or all the way to entire low-resolution models of entire planets.

The *OGC API — 3D GeoVolumes* candidate standard supports streaming models either as the payload for integrated mesh tiles in the Bounding Volume Hierarchy approach (using either *i3s* or *3D Tiles*), or as referenced 3D models. These referenced 3D models can be instantiated by points vector tiles, or through identifiers that could be properties of other APIs for streaming features such as *OGC API — Features* or *OGC API — Moving Features*.

4.1. Model formats

The payload for the models can be encoded in a variety of model formats, such as glTF™ and COLLADA™ defined by the Khronos® Group. While COLLADA™ was meant as an interoperable format based on XML, glTF™ is a more compact representation based on JSON, including support for encoding some of the data in binary buffers. Several other model formats exist, including E3D which is actively developed by Ecere and used during the OGC Testbed 14 for the *CityGML and AR* task. E3D is being proposed as a simple yet complete fully binary encoding that already includes built-in support for 16-bit vertex quantized meshes, which is similar to the Draco extension (`KHR_draco_mesh_compression`) for glTF. A useful resource is the Asset-Importer-Lib which supports transcoding between several different 3D model formats.

4.1.1. glTF™

glTF™ is a popular interoperable format which is widely used as part of the *3D Tiles* specifications and can also be used directly with *OGC API — GeoVolumes*. Binary glTF (extension `.glb`, media type `model/gltf-binary`) is the more practical option than the JSON alternative where binary buffers must either be external reference or base-64-encoded.

Figure 4 — Hubble Space Telescope glTF model from NASA, Gaia Sky in Colour from ESA

4.2. Textures

Models can either embed textures or reference external textures via a URI. Both approaches are possible with glTF. Standardizing a mechanism for listing and retrieving available textures to be re-used by multiple models is being considered for the *OGC API — 3D GeoVolumes specification* as part of its *referenced models* requirements class.

For detailed photorealistic models, textures often represent the larger proportion of the total data volume, more so than the geometry. Compressed textures are often used to reduce the memory footprint. Graphical Processing Units (GPUs) directly support some compressed formats in memory. The Ericsson Texture Compression 2 (ETC2) format is supported by all OpenGL ES 3.0 and OpenGL 4.3 implementations and there are very fast encoding implementations available that support compressing textures in real-time. There are formats dedicated to compressed textures targeting GPUs such as the KTX texture container format for streaming textures, but traditional image formats such as PNG (lossless) and JPEG (if lossy is acceptable) remain viable options as well.

4.3. Animated / Articulated Models

Model formats support defining object node hierarchies, where each node's transformation / pose is relative to its parent, as well as bones / skinning system allowing the attachment of a skeleton defined using such a node hierarchy to a mesh in order to deform it. This approach is used to better represent the movement of objects that are not rigid. A skin associates each vertex of a mesh with one or more bones together with a weight representing the relative importance of that bone in deforming the vertex. Animations are defined as a timeline along which different transformations (e.g., rotation, scale and translation) are applied to a nodes' transformations.

4.3.1. Animated / Articulated models in glTF and COLLADA

glTF™ supports the definition of object node hierarchies, transformations, skeletons / skins, as well as animations.

COLLADA™ supports these capabilities as well.

These capabilities provide the ability to define complex poses of complex systems such as spaceships with internal moving parts and human bodies.

4.3.2. Streaming inner poses of articulated objects

As described in Section 5, in the approved GeoPose 1.0 Standard (draft), the use of inner poses requires the more advanced encoding syntaxes.

However, if the simple syntax with `position` was extended to support not only latitude and longitude CRS, but arbitrary CRS, simple poses could easily be nested as shown in the "train on Mars" scenario. The ability to specify an ID which identifies which node of an object the pose applies to.

This is closer in concept to the ability to define poses and animations in 3D computer graphics, as found in glTF and COLLADA.

With the ability to retrieve a single pose hierarchy with nested poses, or a sequence of these, such a simple JSON encoding, or a binary variant of it for efficiency, would support streaming complex inner poses of articulated objects, including ships with moving parts or human actors. Another alternative may be to stream a glTF payload containing only node hierarchies without the mesh geometry.

5

STREAMING POSITIONS OF MOVING OBJECTS

5 STREAMING POSITIONS OF MOVING OBJECTS

Streaming the positions of moving objects usually implies a data sequence of spatial coordinates accompanied by a time coordinate in a temporal reference system, enabling accuracy regardless of issues such as latency, as well as the ability to include historic or predicted positions, and interpolate positions between two items of a sequence. In addition to a position, it may also be useful to include the orientation of objects at each moment in time, fully describing its six degrees of freedom *pose*. Several OGC standards relate to this capability, most notably OGC API — Moving Features. *OGC API — Moving Features* is an extension of *OGC API — Features* extended with the notions of temporal geometry and temporal properties. Another relevant API is the OGC API — Connected Systems, also designed as an extension to OGC API — Features. GeoPose is also relevant as a representation combining both the position and the orientation temporal property.

5.1. Objects moving on the surface of spherical body

The following JSON pose sequence example illustrates a scenario locating people in different wagons in a train on the surface of Mars. Their position and orientation are expressed relative to the wagons, with the position and orientation of the wagons themselves specified relative to a coordinate reference system for Mars.

```
{
   "crs": [
      "link:++http://www.opengis.net/def/crs/IAU/2015/49901"++[],
      "link:++http://www.opengis.net/def/crs/IAU/0/CoordinatedMarsTime"++[]
   ],
   "sequence": [
      {
         "time": 2053283731.311,
         "id": "link:++http://example.com/nodes/MarsExpress/1"++[],
         "nodes": [
            {
               "id": "link:++http://example.com/nodes/MarsExpress/1/Wagons/1"+
+[],
               "position" : { "lat" : 30, "lon": 50, "h": 25 },
               "angles" : { "yaw": 30, "pitch": 5, "roll" : -0.5 },
               "nodes" : [
                  {
                     "id": "link:++http://example.com/nodes/MarsExpress/1/
Passengers/Josh"++[],
                     "position" : { "x": 2.0, "y": 0.8, "z": -6 },
                     "angles": { "yaw": 1, "pitch": 2, "roll" : 0 }
                  },
                  {
                     "id": "link:++http://example.com/nodes/MarsExpress/1/
Passengers/Jerome"++[],
                     "position" : { "x": 2.2, "y": 0.8, "z": -7 },
                     "angles": { "yaw": -1, "pitch": 1, "roll" : 0 }
```

```
                }
            ]
        },
        {
            "id": "link:++http://example.com/nodes/MarsExpress/1/Wagons/3"+
+[],
            "position" : { "lat" : 29.998, "lon": 50.004, "h": 24.5 },
            "angles" : { "yaw": 31, "pitch": 5.2, "roll" : -0.4 },
            "nodes" : [
                {
                    "id": "link:++http://example.com/nodes/MarsExpress/1/
Passengers/Steve"++[],
                    "position" : { "x": -5, "y": 0.82, "z": 6 },
                    "angles": { "yaw": -2, "pitch": 1.5, "roll" : 0 }
                }
            ]
        }
      ]
    }
  ]
}
```

Figure 5 — Expressing a scenario with people aboard wagons of an express train on Mars as a sequence of hierarchical poses:

5.2. Objects orbiting around a spherical body

Ecere experimented with visualizing the position of the International Space Station (ISS) orbiting Earth, based on the position sequence returned by the API endpoint https://www.astroviewer.net/iss/ws/orbit.php?sat=25544 (where satellite ID 25544 corresponds to the ISS Zarya module).

A sample partial JSON output is shown below:


```
{
   "sat": 25544,
   "tRef": 1681189743,
   "orbitData": [
      { "t": 1681189740, "ln": 77.570, "lt": 45.452, "h": 424, "v": 7.662, "s":
 true },
      { "t": 1681189800, "ln": 82.368, "lt": 47.149, "h": 424, "v": 7.662, "s":
 true },
      { "t": 1681189860, "ln": 87.477, "lt": 48.610, "h": 424, "v": 7.662, "s":
 true },
      { "t": 1681189920, "ln": 92.876, "lt": 49.809, "h": 424, "v": 7.662, "s":
 true },
      { "t": 1681189980, "ln": 98.526, "lt": 50.719, "h": 424, "v": 7.663, "s":
 true },
      ...
   ]
}
```


Figure 6 — Sample JSON output from the astroviewer ISS orbit API:

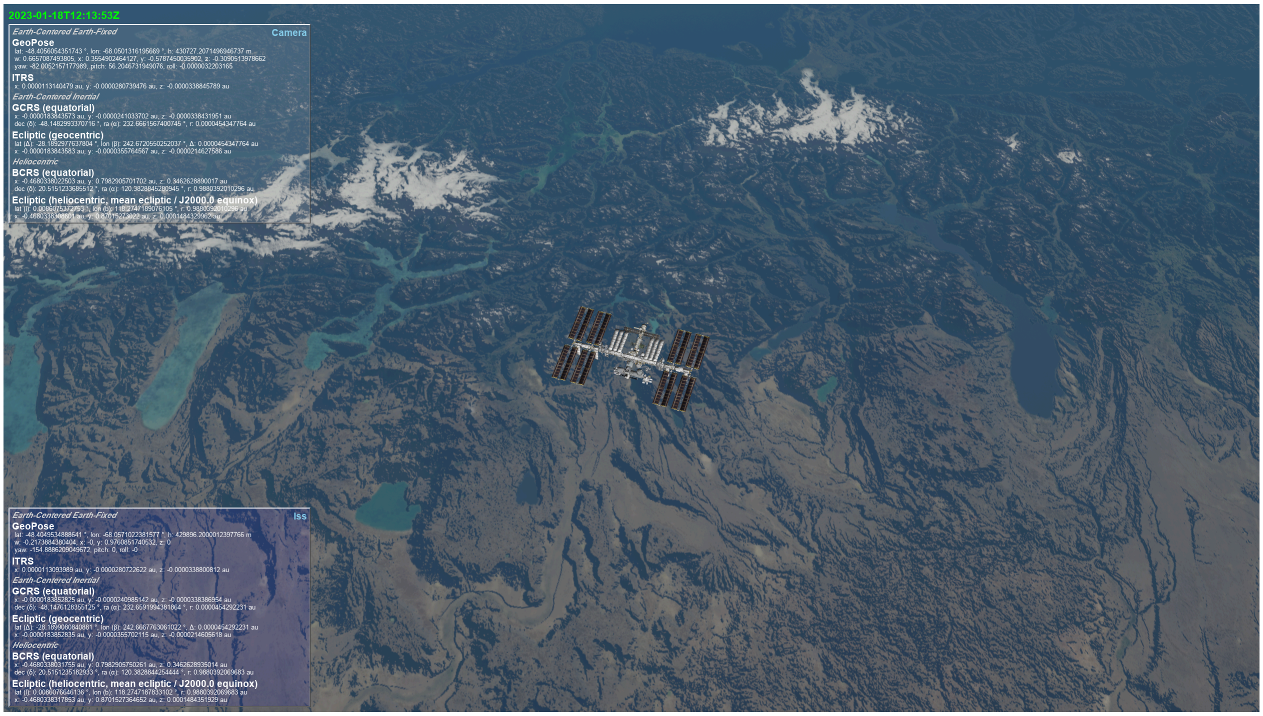

Figure 7 — International Space Station real-time position visualized in Ecere's 3D+ demonstration client (seen from above) with NASA's Blue Marble Next Generation and Viewfinder Panoramas terrain elevation (glTF Model from NASA)

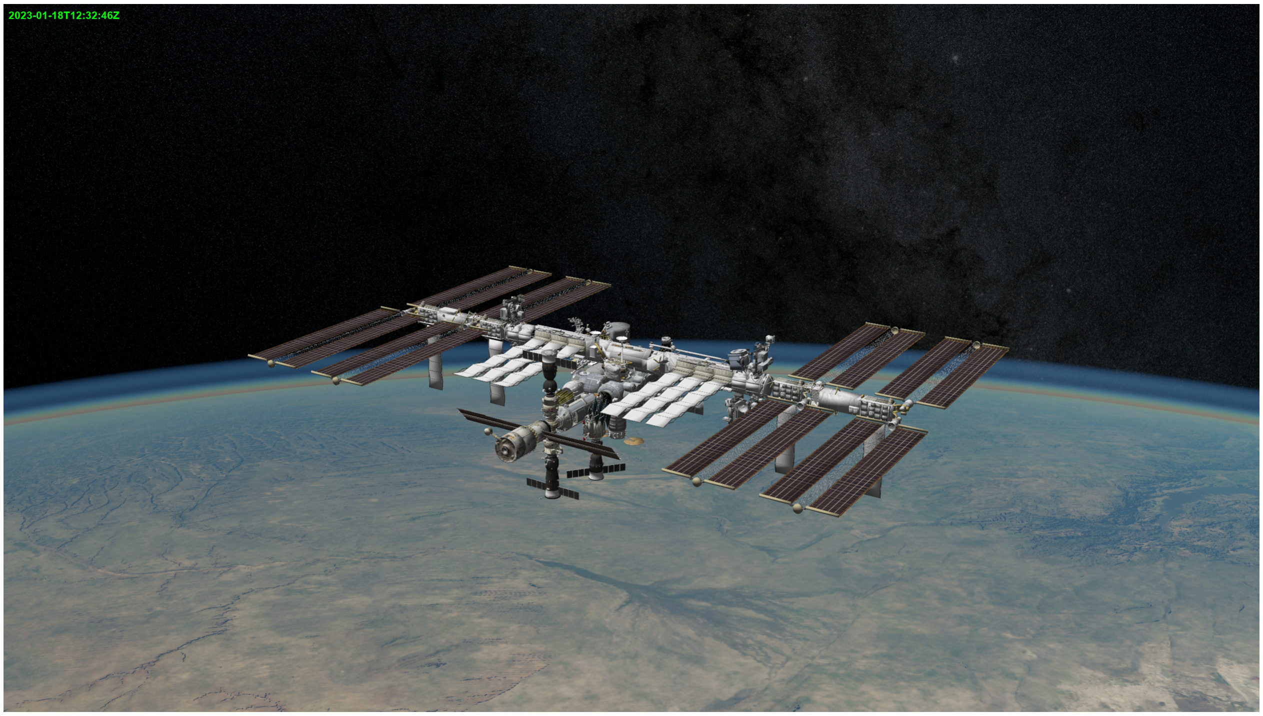

Figure 8 — International Space Station real-time position visualized in Ecere's 3D + demonstration client, with NASA's Blue Marble Next Generation, ESA's Gaia Sky in Colour and Viewfinder Panoramas terrain elevation (glTF Model from NASA)

5.3. Streaming rapidly moving data

The draft OGC API — Connected Systems Standard extends *OGC API — Features*, mainly focused towards static features, with the ability to efficiently retrieve a large number of dynamic features and associated properties using mechanisms such as MQTT and WebSocket.

5.4. Streaming data with OGC API — Moving Features

OGC API — Moving Features defines retrieving both temporal geometry (`/items/{itemId}/tgeometries`) and temporal properties (`/items/{mFeatureId}/tproperties`) for a particular feature. The API supports selecting a time range of interest using a `datetime` parameter.

Some questions and issues (with their GitHub repository issue numbers linked below) were raised by participants to try to better assess how suitable the standard would be for 3D+ data, including many rapidly moving features, as well as to clarify a potential relationship with GeoPose as follows.

- Past, Current, Predicted trajectories (#25)
- Retrieving temporal geometries for multiple features (#26)
- Retrieving temporal geometries and properties together (#27)
- Scales within orientations (#28)
- Base property used to refer to 3D models (#29)
- GeoPose as a representation of *OGC API — Moving Features* temporal geometry (#31)
- Clarify the relationship with *OGC API — Connected Systems* (#33)

5.5. Representing positions and orientations with GeoPose

The OGC GeoPose Draft Standard specifies requirements for defining a "pose", consisting of a position and orientation, in a convenient and standardized manner. In its simplest encoding form, a position is expressed in WGS84 latitude, longitude, and height above the ellipsoid, as well as an orientation relative to the local tangent plane (LTP), with axes oriented towards East, North and Up (ENU).

Additional more advanced encodings are also specified in GeoPose, including the ability to include the time at which it is valid and define graphs involving multiple poses and the transformation between an outer and inner frame within that graph, as well as the ability to specify an outer frame other than WGS84 / LTP-ENU.

Unfortunately, these advanced capabilities come at a steep increase in complexity for the encoding, as well as a complete break from the simple basic encoding form. Some of that syntax also breaks significantly from common JSON practice, requiring an additional parsing pass. Related issues (with their GitHub repository issue numbers linked below) were filed with the GeoPose Standard Working Group as follows.

- Advanced GeoPose JSON encoding not JSONic (#69)
- Time with basic GeoPose JSON encodings (#70)
- GeoPose as a representation of OGC API — Moving Features temporal geometry (#71)
- UNIXTime with different epoch / different celestial body (#72)
- Simple encoding for GeoPose sequences (#73)
- Much more powerful existing inner frame animation capabilities in 3D model formats (#74)

Furthermore, as discussed in the Testbed 18 Reference Frame Transformation ER (22-038), the OGC Abstract Topic 2 / ISO 19111 — *Referencing by Coordinates Standard* fully covers the capabilities of GeoPose, including support for the advanced capabilities.

5.6. Objects referenced in engineering CRS

The ability to specify a transformation between frames of reference allows defining an engineering CRS. In the “train on Mars” scenario above, the positions of the individual passengers can be seen as being defined in the engineering CRS local to the individual wagons.

6

STREAMING DATA FOR THE SOLAR SYSTEM AND BEYOND

6 STREAMING DATA FOR THE SOLAR SYSTEM AND BEYOND

Several coordinate systems have been defined to conveniently position objects within the Solar System and beyond. In order to learn more about some of these coordinate systems and get a better sense of how data using them could be streamed, Ecere developed an experimental Solar System visual simulation. The simulation positions 3D models of the Sun, the eight planets, Pluto, and the Moon based on the current simulation time. The orientation of these celestial bodies is also set based on their angular velocities and the tilt of their rotation axes. In addition, the trajectory of the Voyager 1 probe is represented based on historical and predicted data. The GeoPose corresponding to each of these objects as well as for the current 3D view's camera is displayed together with the position in each of the additional coordinate systems selected for the experiment.

6.1. Astronomical coordinate systems

Geographic coordinate reference systems, such as the International Terrestrial Reference System and Frame (ITRS/ITRF) used by the EPSG:4978 (Cartesian) and EPSG:4979 (ellipsoidal) 3D coordinate systems, are fixed and centered to the Earth and therefore rotate with the Earth's spin, allowing the use of the same coordinates to denote a position on Earth regardless of the time of day and day of the year. For positioning objects away from Earth, it is desirable to use an inertial reference frame fixed with respect to the distant stars.

Among these inertial reference frames, some have as their origin the Earth's center of mass (geocentric), the Sun's center or mass (heliocentric), or the Solar System's center of mass (barycentric). The Solar System's center of mass can be away from the Sun's Center of mass by approximately twice the radius of the Sun, or about 1 million kilometers.

These inertial reference frames also vary in terms of the reference planes, which can be the ecliptic (the trajectory of the Earth orbiting around the sun), or the celestial equator (an outwards extension of the Earth's equatorial great circle). The angular difference between these two reference planes, the ecliptic and the celestial equator, is called the obliquity of the ecliptic, and is approximately 23.4 degrees. For both ecliptic and equatorial reference planes, the primary direction (reference longitude) is the (March/vernal) equinox — the intersection of the ecliptic and equatorial planes where the Sun moves North through the celestial equator (the Sun's ascending node).

For equatorial reference frames, the orientation of the celestial sphere varies based on Earth's tilted spinning rotation axis which changes over time with the effects of *precession* (a long term variation with a period of around 26 thousand years) and *nutation* (a short term effect with a period of 18.6 years), both resulting from gravitational forces of the Sun, Moon, and planets. Therefore, the equinox of equatorial reference frames are qualified as being the true equinox of a particular date (including the short term effect of nutation), the mean equinox of date (excluding nutation), or the mean equinox of a standard epoch (e.g., J2000.0 for noon on January 1, 2000 Terrestrial Time, corresponding to 11:58:55.816 UTC).

Coordinates in both equatorial and ecliptic reference frames can be expressed using either spherical coordinates or 3D rectangular (Cartesian) coordinates. Whereas the terms *ecliptic latitude* (β) and *ecliptic longitude* (λ) are used for spherical coordinates, *declination* (δ, latitude) and *right ascension* (α, longitude) are used to denote spherical coordinates in an equatorial reference frame.

The International Celestial Reference System (ICRS) is standardized by the International Astronomical Union (IAU) based on an equatorial reference frame with an origin at the barycenter of the Solar System.

The term Barycentric Celestial Reference System (BCRS) is sometimes used interchangeably with ICRS. The Geocentric Coordinate Reference System (GCRS) can be thought of as a geocentric version of the ICRS. (see also Wikipedia articles about the BCRS and GCRS and about the ICRS/ICRF).

Although they were not covered in the Testbed 18 experiments, reference frames are also defined relative to the galactic plane (inclined 60.2 degrees relative to the ecliptic), and the supergalactic plane (inclined 84.5 degrees to the galactic plane, based on the supercluster of galaxies containing the Milky Way). Another type of coordinate not part of these experiments is horizontal coordinate, which specifies an altitude from the horizon and an azimuth (usually relative to true north).

The following table summarizes these major types of astronomical coordinate reference systems.

Table 1 — Summary of astronomical coordinate reference systems

TYPE	ORIGIN	REFERENCE PLANE	PRIMARY DIRECTION
Equatorial (GCRS)	Earth's center of mass	Celestial equator	(March) Equinox
Equatorial (BCRS)	Barycenter of Solar System	Celestial equator	(March) Equinox
Ecliptic (geocentric)	Earth's center of mass	Ecliptic	(March) Equinox
Ecliptic (heliocentric)	Sun's center of mass (or Solar System's)	Ecliptic	(March) Equinox
Galactic	Sun's center of mass	Galactic plane	Galactic center
Supergalactic	Sun's center of mass	Supergalactic plane	Intersection of galactic and supergalactic plane
Horizontal	Observer's location	Horizon	North of horizon

6.2. Astronomy Software libraries

Two software libraries have been sanctioned by the IAU for manipulating and performing transformations between the BCRS and other reference systems. These are the Standards of Fundamental Astronomy (SOFA) system and the Naval Observatory Vector Astrometry Subroutines (NOVAS). Ecere cursorily reviewed the documentation for both libraries and settled on the use of the NOVAS C library for the purpose of its Solar System visual simulation experiments as it proved easier to identify the required functionality from its API and documentation.

The following table lists the NOVAS methods used in the experiment and provides a brief description of the purpose for which they were used.

Table 2 — Summary of NOVAS C library functions used in experiments

NOVAS FUNCTION	PURPOSE
`julian_date()`	Calculating a double precision fractional Julian day from year, month, day, hour, and fractional seconds
`radec2vector()`	Converting spherical coordinates to rectangular
`vector2radec()`	Converting rectangular coordinates to spherical
`cel2ter()`	Converting from GCRS to ITRS (for a given fractional Julian day)
`ter2cel()`	Converting from ITRS to GCRS (for a given fractional Julian day)
`ecl2equ_vec()`	Converting (heliocentric) ecliptic to equatorial (e.g., ICRS) coordinates (rectangular)
`equ2ecl_vec()`	Converting equatorial (e.g., ICRS) to (heliocentric) ecliptic coordinates (rectangular)
`solarsystem()`	Calculating GCRS coordinates (in astronomical units) of solar system bodies (1: Mercury, 2: Venus, 3: Earth, 4: Mars, 5: Jupiter, 6: Saturn, 7: Uranus, 8: Neptune, 9: Pluto, 10: Sun, and 11: Moon) at a given fractional Julian day
`ephem_open()`	Initializing ephemerides data file
`ephem_close()`	Releasing memory associated with ephemerides data file

In order to use the `solarsystem()` function to compute the position of celestial bodies other than the Earth and the Sun, the library must be configured to use an ephemerides data file (version 441 available from `ftp://ssd.jpl.nasa.gov/pub/eph/planets/Linux/de441/`). This is done by compiling the `solsys3.c` source file (instead of e.g., `solsys1.c`) invoking the `ephem_open()` function with the filename, and using the corresponding `ephem_close()` during termination. In addition, the `eph_manager.c` file had to be modified to recognize version 441, assuming that the `RECORD_LENGTH` should be 8144 as with recognized version 421.

6.3. Visualizing Voyager 1 trajectory experiment

An objective of this experiment was to demonstrate the ability to describe and exchange the position of objects from a location on Earth to interstellar space. Visualizing the trajectory of the Voyager 1 space probe presented an interesting scenario. Information on Voyager 1 mission was found at https://nssdc.gsfc.nasa.gov/nmc/spacecraft/display.action?id=1977-084A . Daily positions of the spacecraft were found available at https://pds-ppi.igpp.ucla.edu/data/VG1-SS-POS-6-1DAY-V1.0/ (from 1977 to 2005). A similar set of positions was used, but including the predicted trajectory until the end of 2023 (possibly retrieved from https://omniweb.gsfc.nasa.gov/helios/heli.html , which now appears to be offline). An excerpt from this dataset is shown below. The Solar Ecliptic Coordinate System (SE) is used, with `RAD_AU` representing the distance from the Sun:

Table 3 — Excerpt from Voyager 1 trajectory data

YEAR	DAY	RAD_AU	SE_LAT	SE_LON
1977	252	1.01	0.1	347.3
1977	253	1.01	0.1	348.6
1977	254	1.01	0.1	349.9
1977	255	1.01	0.2	351.2
1977	256	1.02	0.2	352.5

The launch of Voyager 1 can be seen in this YouTube video.

Voyager 1 was launched from Cape Canaveral Launch Complex 41 on September 5, 1977 at 12:56:00 UTC aboard a Titan IIIE launch system. A properly georeferenced 3D model of the complex and the Titan IIIE integrated in the demonstration is seen below.

Figure 9 — Cape Canaveral Launch Complex 41 seen with a Titan IIIE rocket

NASA also has its own Voyager tracker at https://voyager.jpl.nasa.gov/mission/status/ .

Information on the Voyager 2 mission is also found at https://nssdc.gsfc.nasa.gov/nmc/spacecraft/display.action?id=1977-076A . Daily positions for Voyager 2 are available from https://pds-ppi.igpp.ucla.edu/data/VG2-SS-POS-6-1DAY-V1.0/ .

Figure 10 — Voyager 1 as it approaches Jupiter (glTF Models from NASA, Gaia Sky in Colour from ESA)

Figure 11 — Voyager 1 soon after its closest approach of Saturn (glTF Models from NASA, Gaia Sky in Colour from ESA)

6.4. Solar System visualization

The position and orientation of the Sun, Moon, and major planets, along with the Earth's proper position and orientation were all represented in the visual simulation experiment.

The ability to convert ICRS coordinates into ITRS coordinates with the NOVAS library allowed defining the position of GeoPoses, whereas the axis tilt relative to its orbit and the rotational velocity of each celestial body allowed computing the orientation. The ability to convert positions in space to a GeoPose facilitates the visualization within a geospatial visualization toolkit, such as the GNOSIS SDK used for this demonstration. This allows positioning presentation objects using geographic 3D coordinate reference systems such as EPSG:4978 (Cartesian) and EPSG:4979 (ellipsoidal).

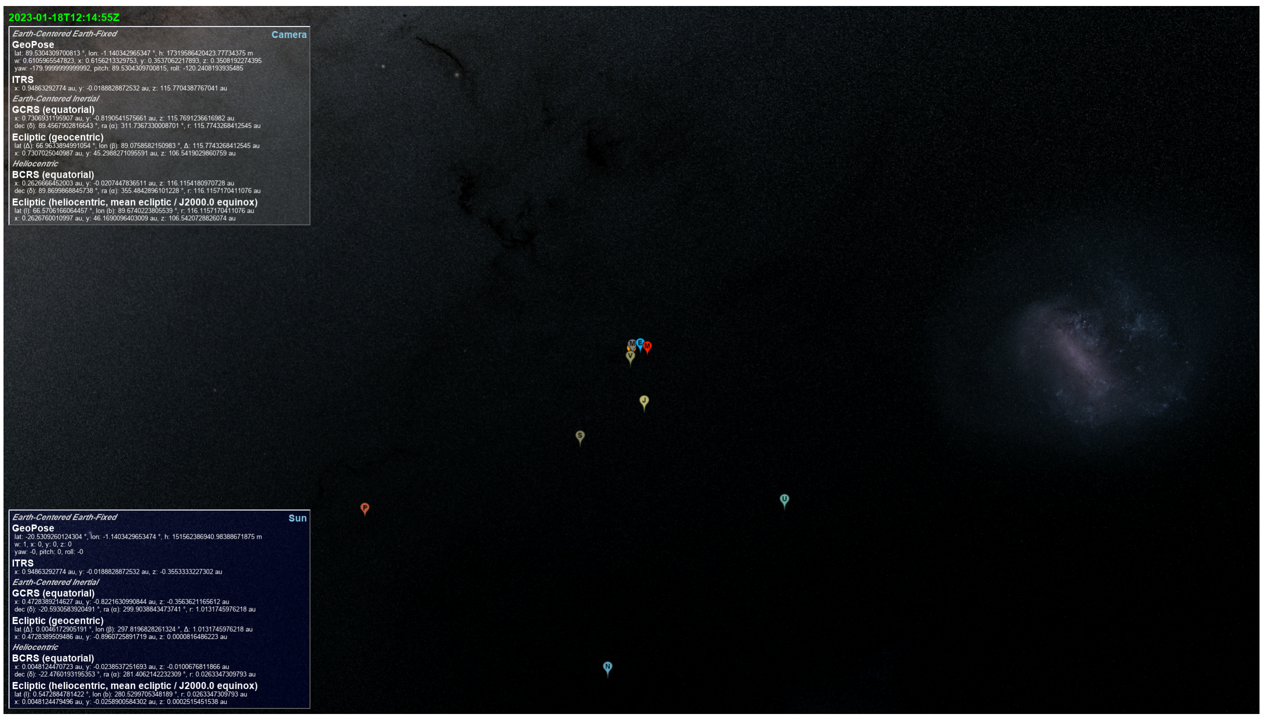

Figure 12 — A view of the Solar System, centered on the Sun

3D models were sourced from glTF models provided by NASA (available from https://solarsystem.nasa.gov/resources/), and scaled to the correct size. The following table summarizes the radius, axis tilt, and rotational velocity at the equator of each celestial body represented.

Table 4 — Radius, axis tilts and angular velocity of celestial body

CELESTIAL BODY	RADIUS (KILOMETERS)	ROTATIONAL VELOCITY AT EQUATOR (M/S)	AXIS TILT (DEGREES)
Sun	696342	1997	0 (7.25 to ecliptic)

CELESTIAL BODY	RADIUS (KILOMETERS)	ROTATIONAL VELOCITY AT EQUATOR (M/S)	AXIS TILT (DEGREES)
Mercury	2439.7	3.026	2.04
Venus	6051.8	1.81	177.36
Earth	6378.137	465.1	23.4392811
Mars	3396.2	241	25.19
Jupiter	71492	12600	3.12
Saturn	60268	9870	26.73
Uranus	25559	2590	97.77
Neptune	24764	2680	28.32
Pluto	1188.3	13.106	122.53
Moon	1738.1	4.627	1.5424

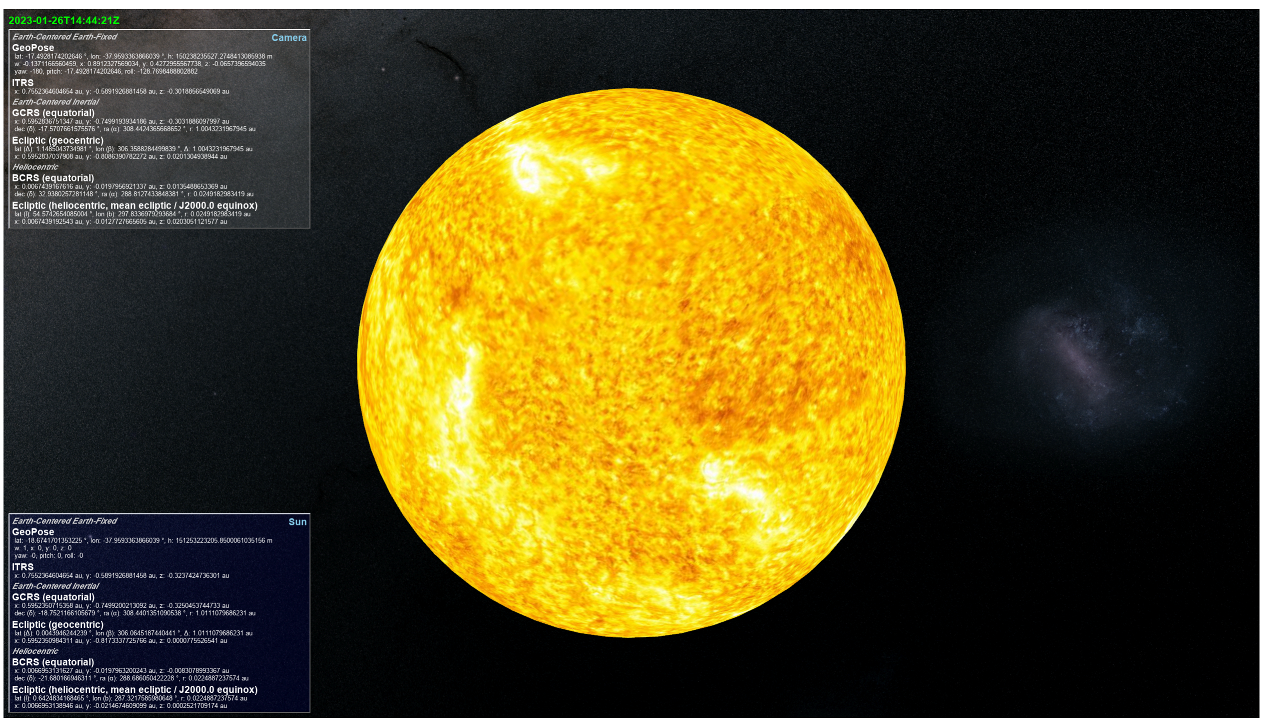

Figure 13 — A closer view of the Sun

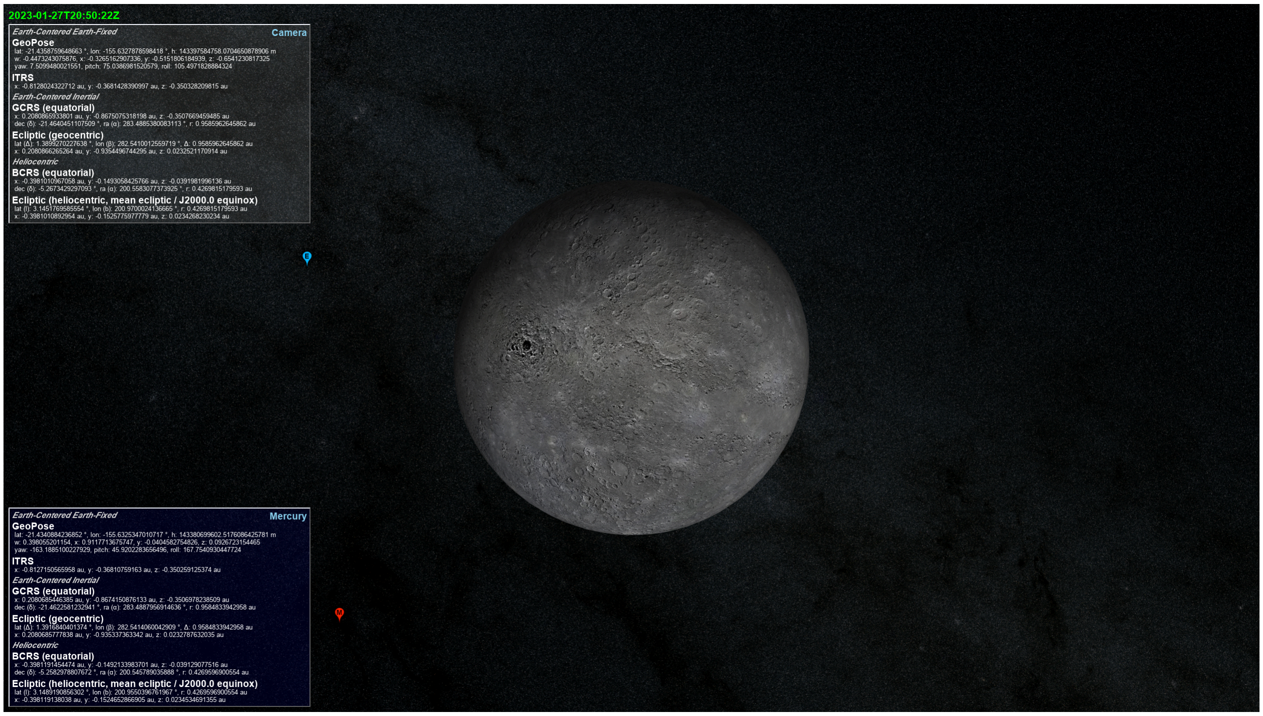

Figure 14 — A view of Mercury

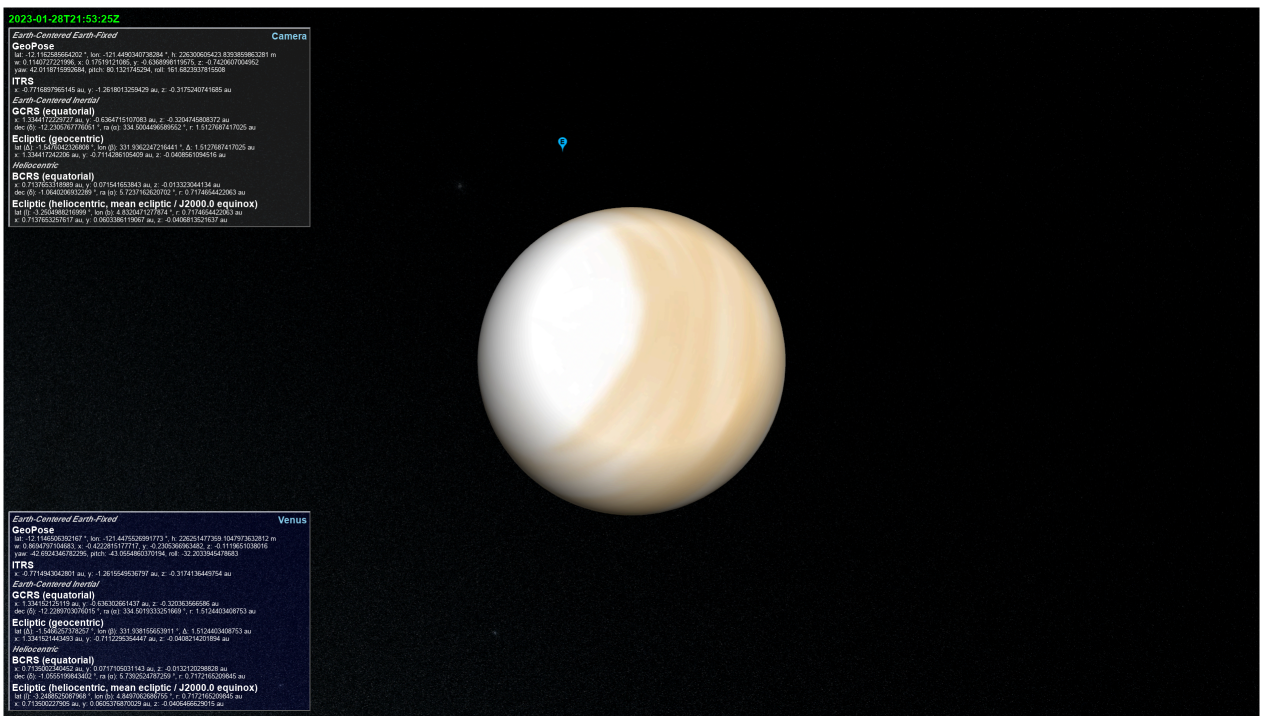

Figure 15 — A view of Venus

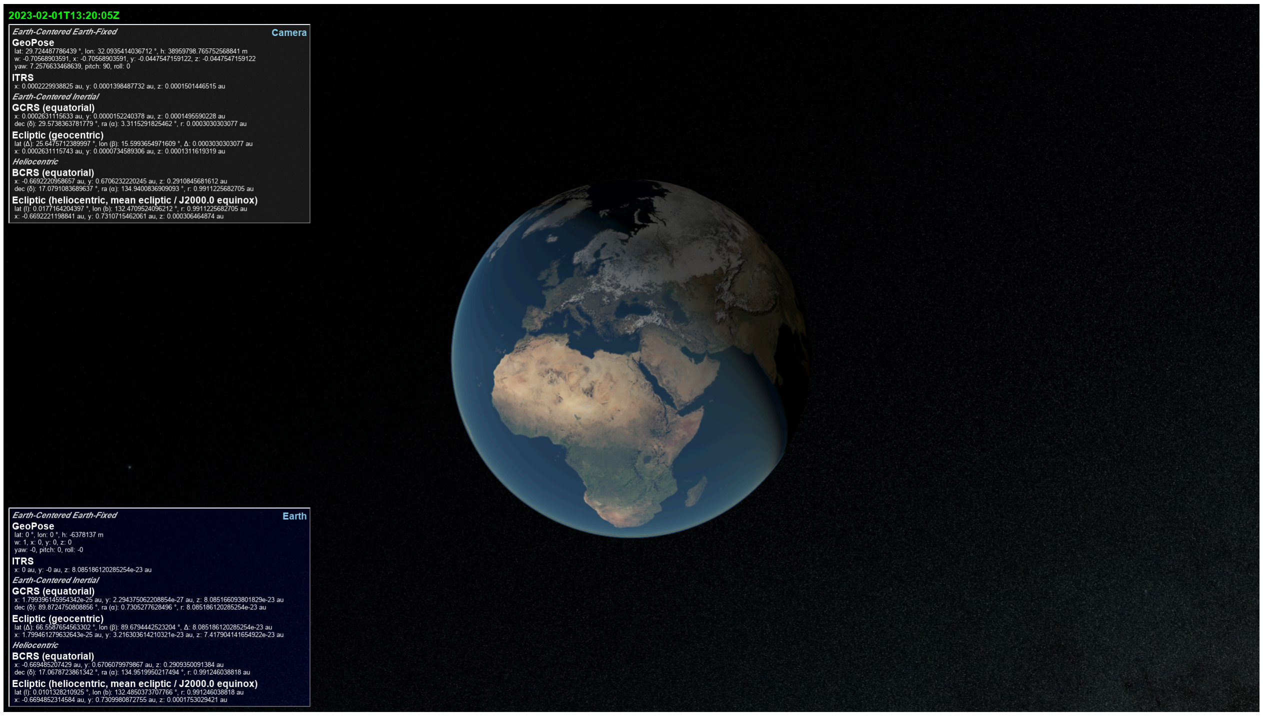

Figure 16 — A view of Earth

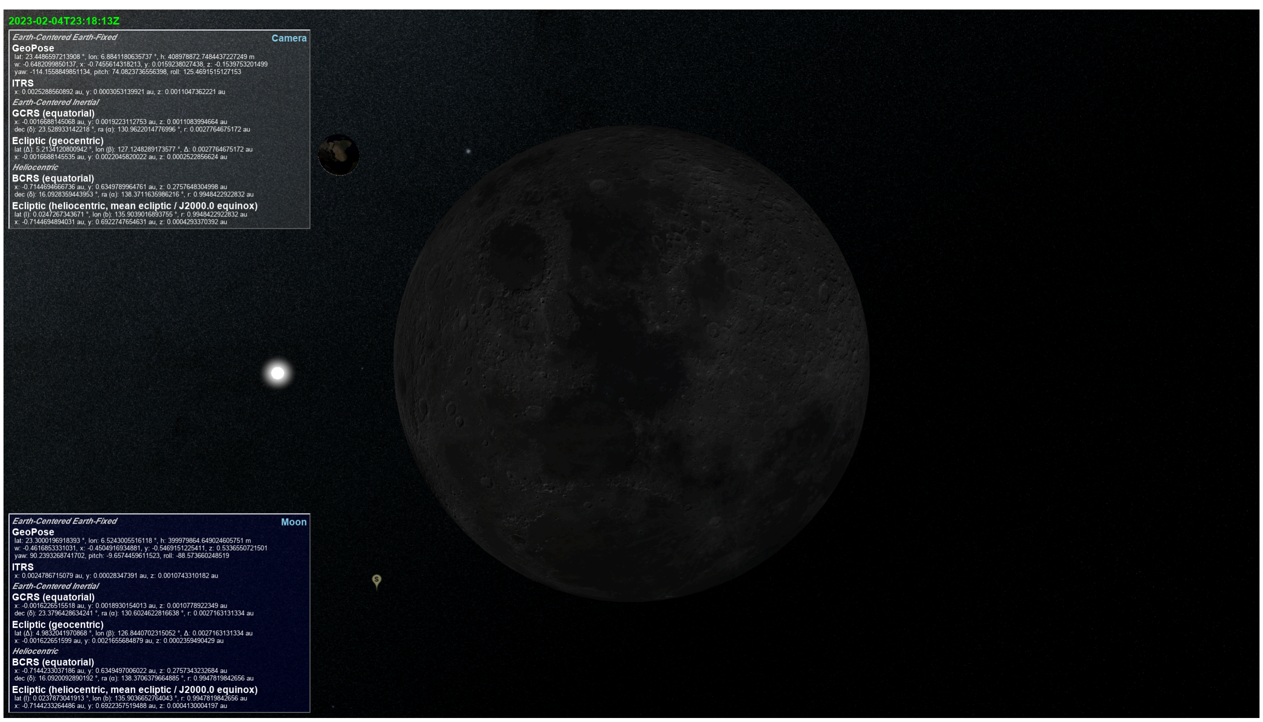

Figure 17 — A view of the Moon

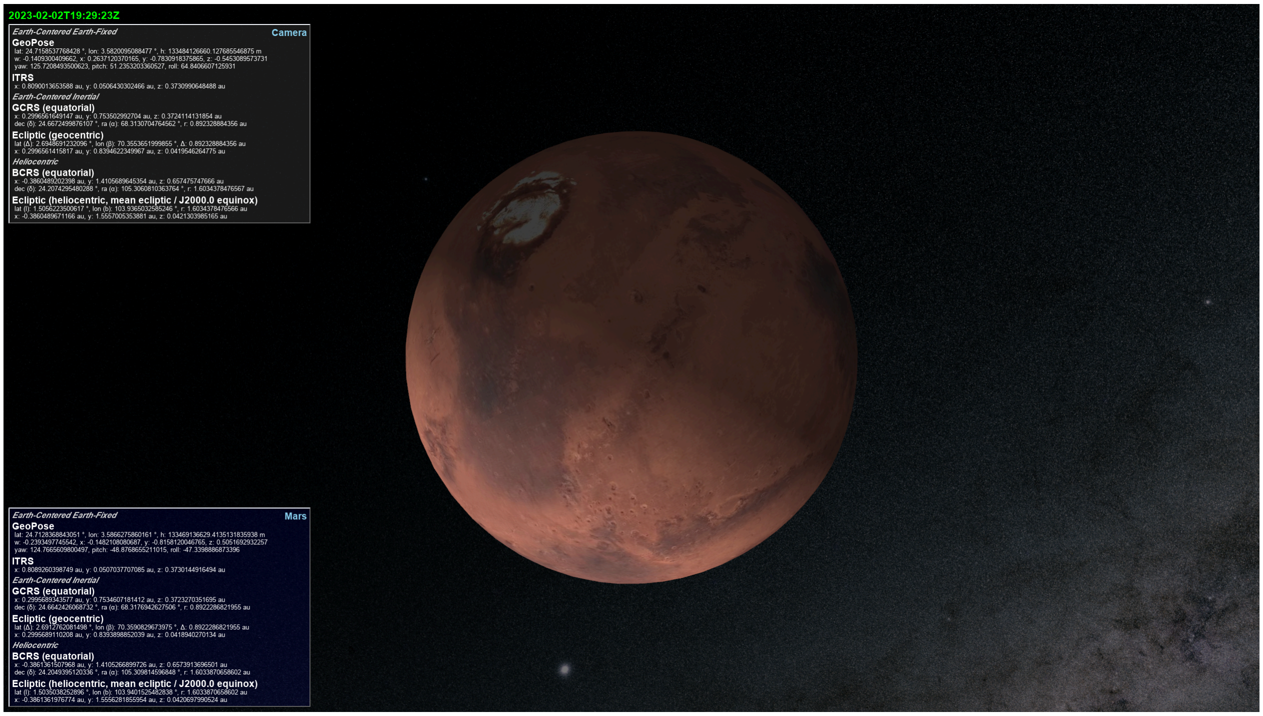

Figure 18 — A view of Mars

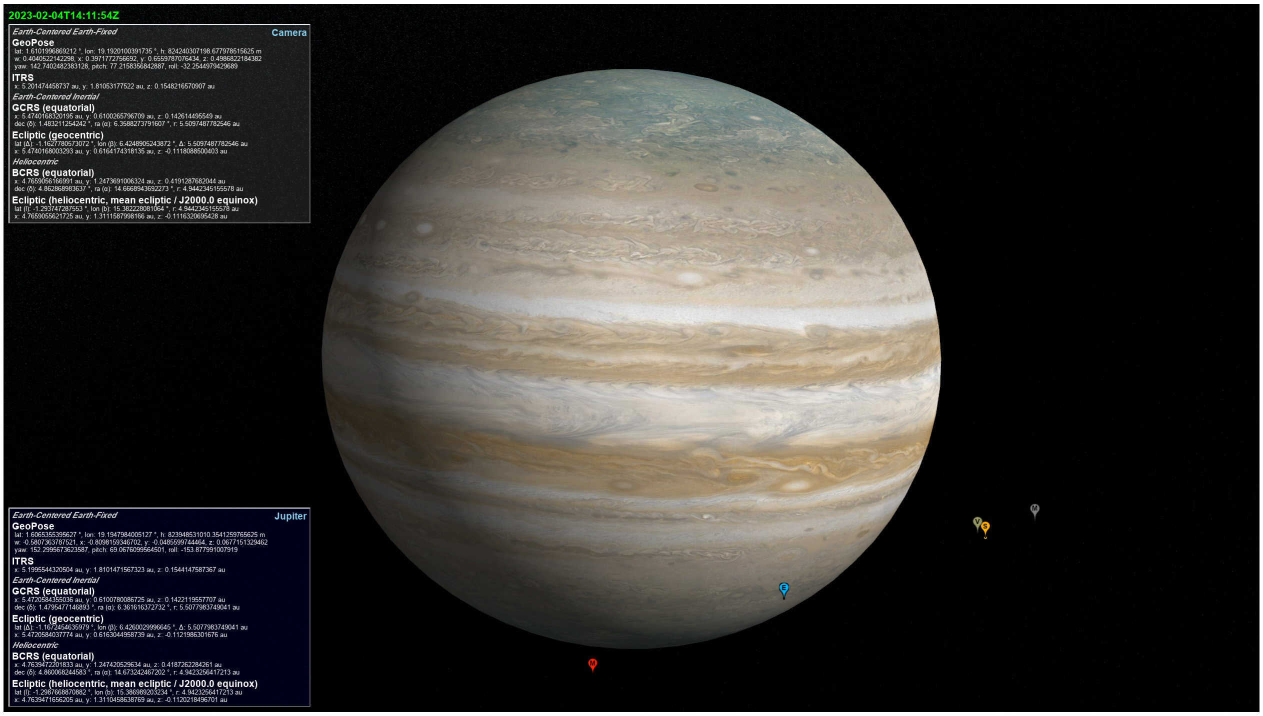

Figure 19 — A view of Jupiter

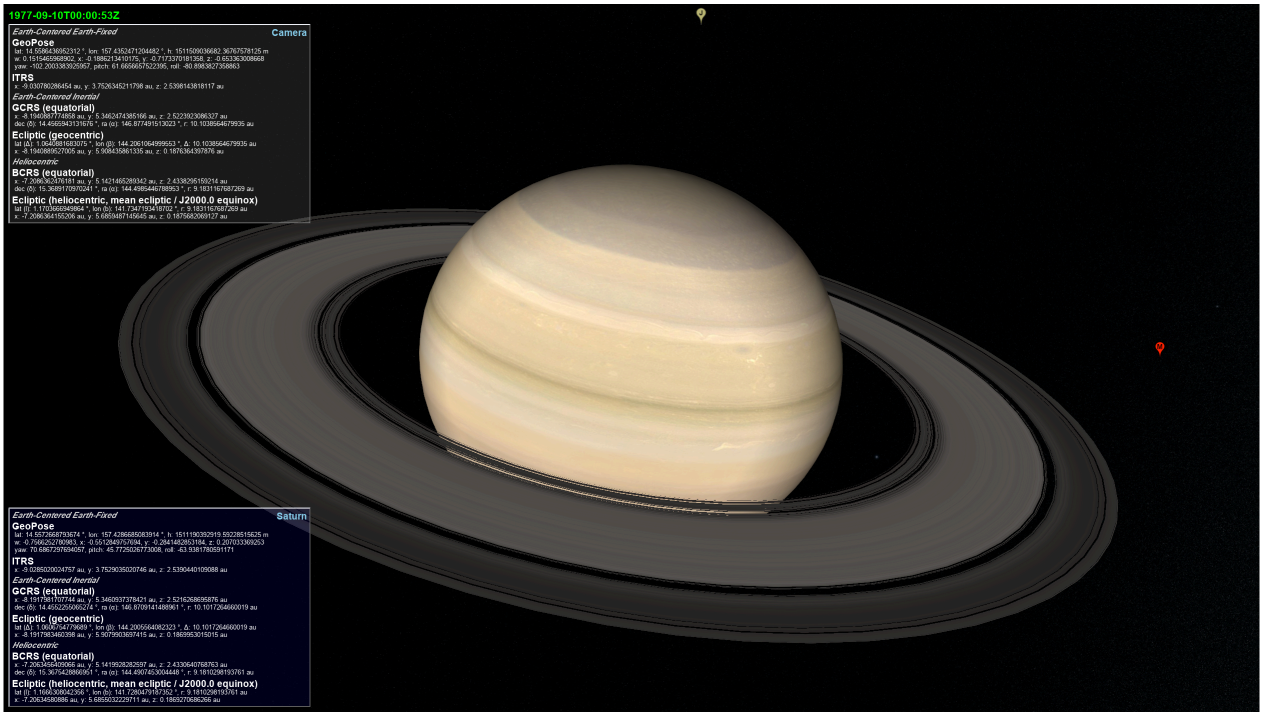

Figure 20 — A view of Saturn

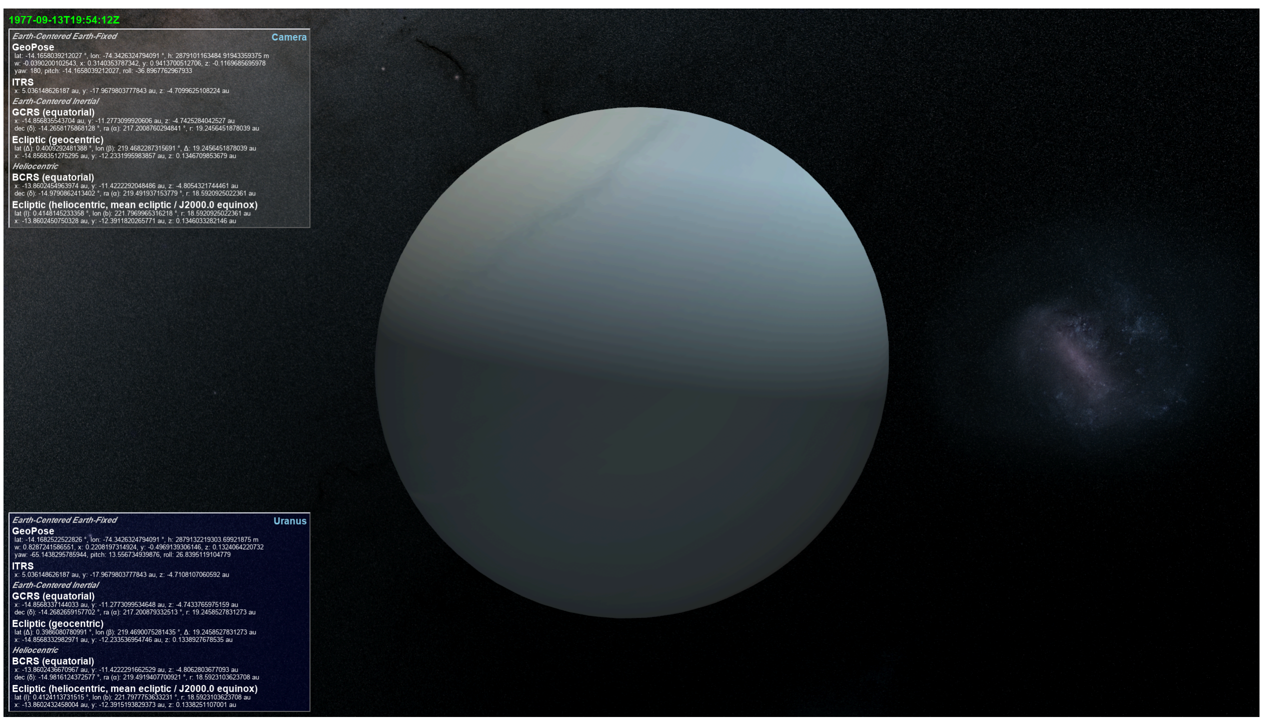

Figure 21 — A view of Uranus

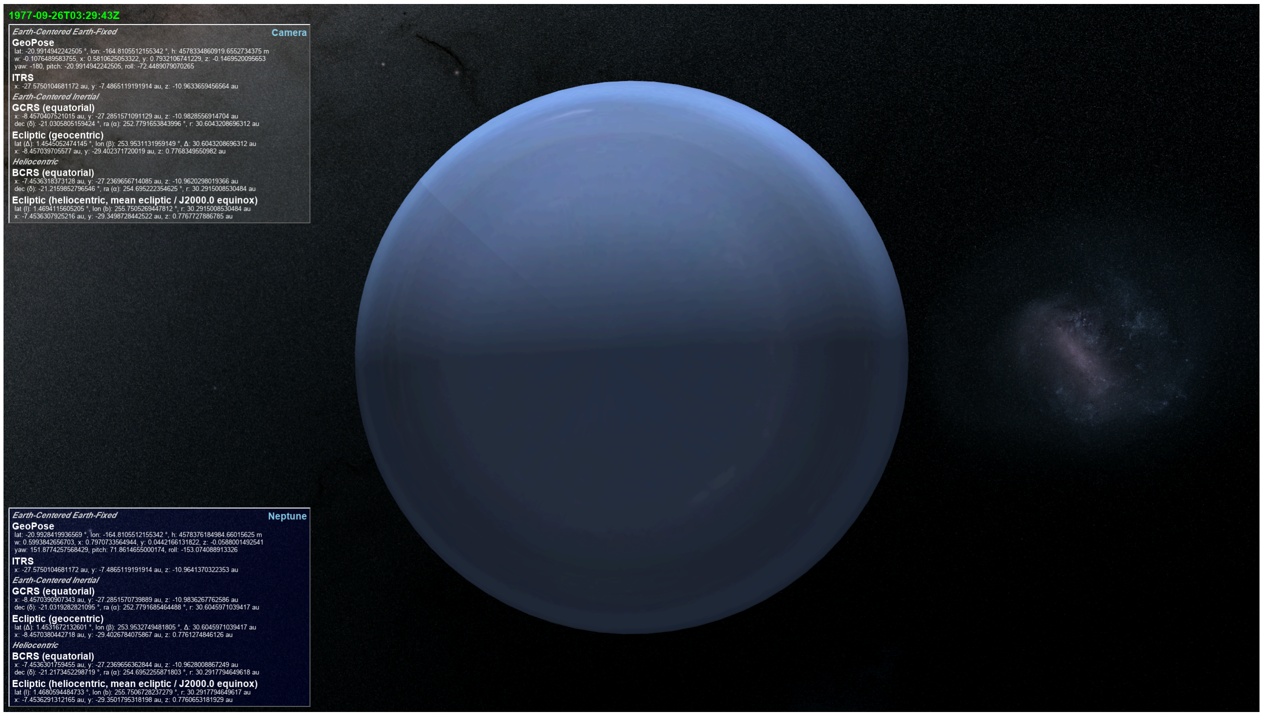

Figure 22 — A view of Neptune

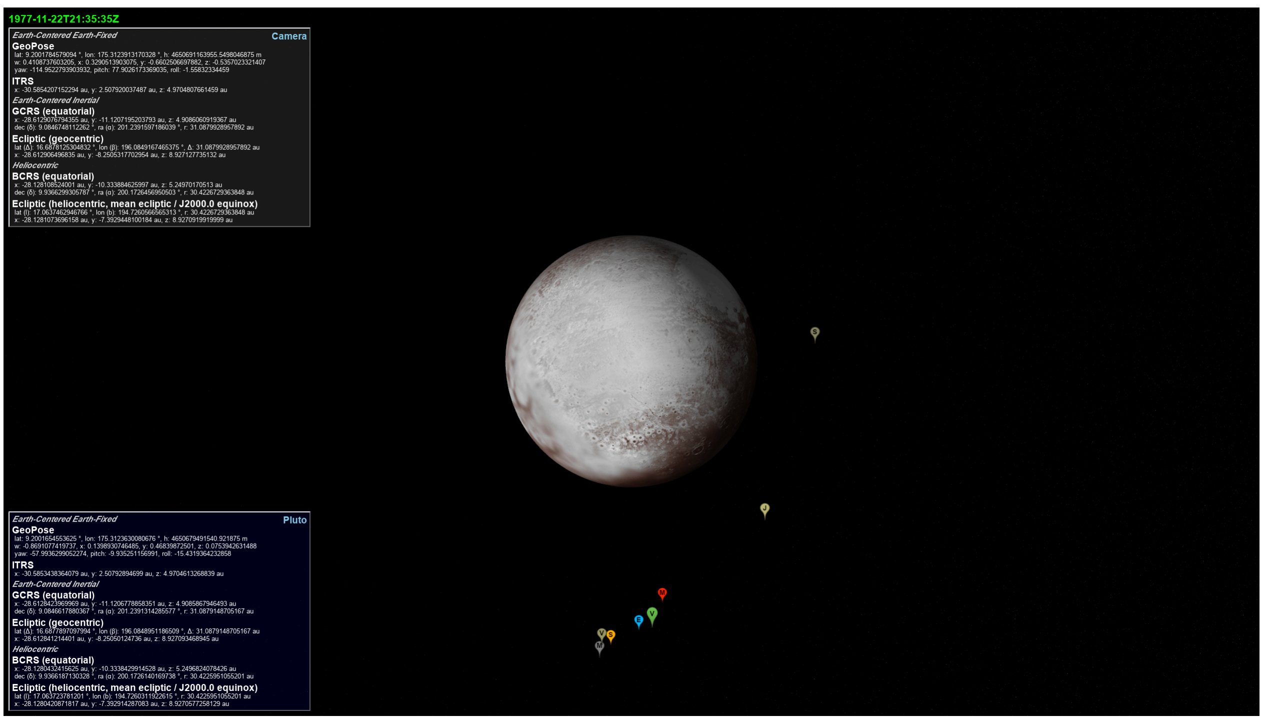

Figure 23 — A view of Pluto

NASA also has a Solar System visualization tool at https://eyes.nasa.gov/apps/solar-system/ which also features the trajectory of the Voyager spacecrafts.

6.5. Passengers aboard interplanetary spaceship scenario

The following JSON hierarchical pose sequence example was put together to illustrate a scenario locating a spaceship on its way from Mars to Earth, with the positions and orientation of passengers onboard relative to the spaceship's own position and orientation.


```
{
   "crs": [
      "link:++http://www.opengis.net/def/crs/OGC/0/BarycentricCelestial"++[],
      "link:++http://www.opengis.net/def/crs/OGC/0/BarycentricTime"++[]
   ],
   "sequence": [
      {
         "time": 2053283731.311,
         "id": "link:++http://example.com/nodes/StarShip/1"++[],
         "position": { "x": 0.5633130330055, "y": 1.181739084022, "z":
0.4698052612411 },
         "quaternion": { "x": 0.5131, "y": 0.14325, "z": 0.65354, "w":
0.5376739680327 },
         "nodes": [
            {
               "id": "link:++http://example.com/nodes/StarShip/1/Passengers/
Josh"++[],
               "position" : { "x": 10, "y": 40, "z": 4 },
               "angles": { "yaw": 170, "pitch": 2, "roll" : 0 }
            },
            {
               "id": "link:++http://example.com/nodes/StarShip/1/Passengers/
Jerome"++[],
               "position" : { "x": 12, "y": 42, "z": -6 },
               "angles": { "yaw": -20, "pitch": 1, "roll" : 0 }
            }
         ]
      }
   ]
}
```


Figure 24 — Expressing a scenario with people aboard a spaceship to Mars as a sequence of hierarchical poses:

6.6. James Webb Space Telescope scenario

The James Webb Space Telescope orbits the Sun near the second Lagrange point (L2) of the Earth-Sun system, around 1.5 million kilometers away from Earth. This location is where gravity from the Sun and Earth balance the centripetal force required for a comparatively small object to move with them, minimizing the amount of energy necessary to maintain its position. The telescope's position could be described using any of the coordinate systems mentioned in this section and converted to any other, but heliocentric ecliptic proves to be most convenient.

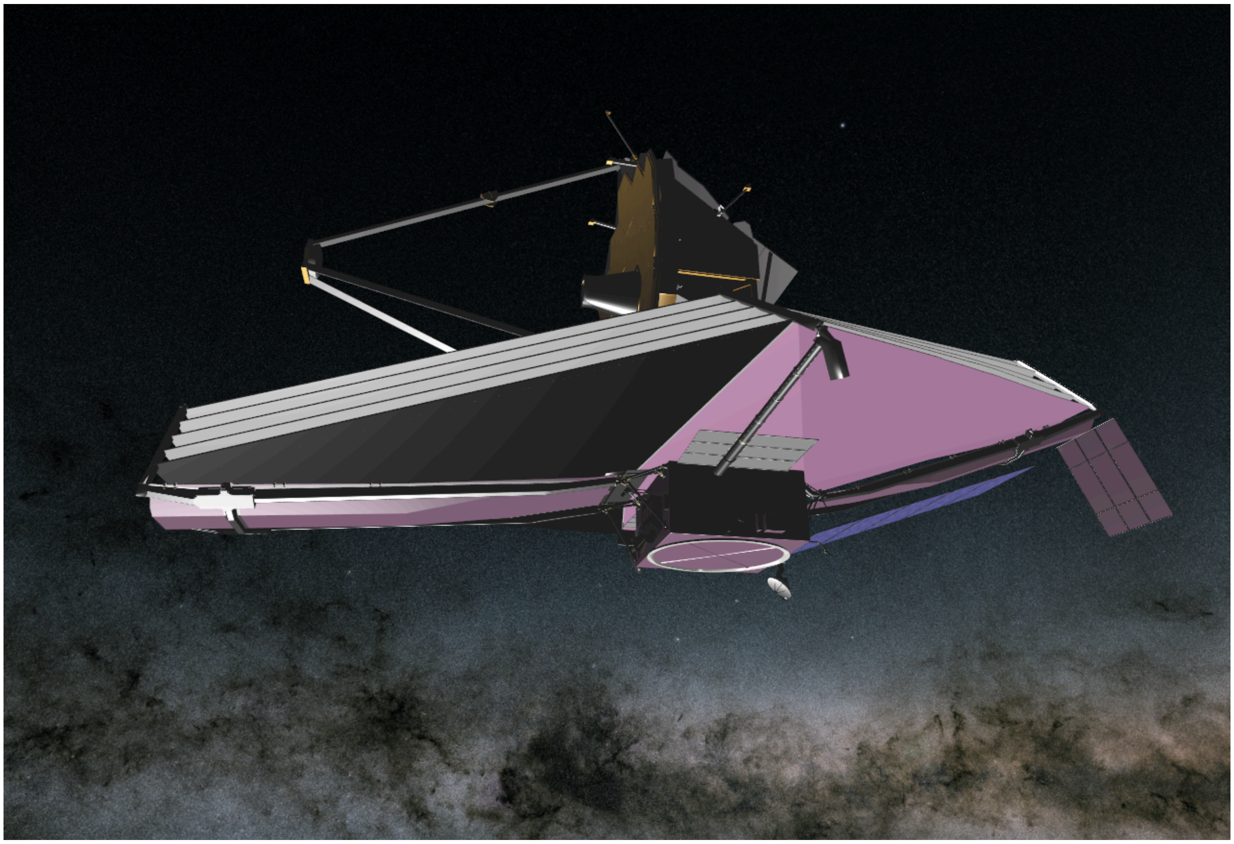

Figure 25 — James Webb Space Telescope glTF model from NASA, Gaia Sky in Colour from ESA

6.7. Star catalogs

Catalogs identifying a very large number of stars, including position and magnitude information, are available. Such a star database can be used to provide situational awareness. Star positions could be efficiently streamed using filters balancing spatial proximity and brightness together with available bandwidth and computational capabilities. Structures such as octrees extending out into the celestial sphere from a geographic grid could provide one mechanism for how such data could be accessed and cached, following the approach described in the OGC 2D Tile Matrix Set and Tileset metadata standard Annex J.1.3. Several catalogs, originating from the original KStars free software, including all stars up to magnitude 8 from the Naval Observatory Merged Astrometric Dataset (NOMAD) catalog, can be downloaded from this link:

https://free-astro.org/download/kstars-siril-catalogues/ .

VizieR is a comprehensive resource providing access to several astronomical catalogs accessible using multiple interfaces and query tools to filter data of interest.

6.8. Special and general relativity considerations

As the scope of this Testbed's *3D+ Data Standards and Streaming* thread included evaluating the ability to reference and stream data with temporal aspects where special relativity might be relevant, participants attempted to familiarize themselves with basic notions of special relativity and attempted to identify concrete use cases where the effects of special relativity might be significant. Since relevant scenarios involved general relativity as well, that topic was also explored.

6.8.1. Non-relativistic scenarios

For observations made according to a local coordinate reference system related to the known positions of major astronomical bodies at a certain point in time, pairing a temporal component with those observations should enable streaming that information and to precisely identify a position in four-dimensional space. Regardless of the time it takes to receive the message, the spatiotemporal position can then be accurately converted to any reference frame based on that information by using the accurately calculated motions of those reference astronomical bodies for the recorded timestamp. The localization and tracking of objects far away from Earth within their local coordinate reference frame may not necessarily require relativistic considerations. Whether special relativistic effects are significant depends on the degree of extreme accuracy required and how much the relative speeds between different reference frames involved in the calculations approach the speed of light. General relativity becomes relevant when dealing with very strong gravitational fields such as near massive objects. The requirement to keep a record of the time of observations as part of a coordinate system and stream this temporal information together with spatial coordinates, to accurately reconstruct the reference frame of those observations, is a requirement that should not be confused with the requirement to take relativistic effects into consideration.

6.8.2. Special relativity

Special relativity describes effects that become apparent for objects traveling at speeds approaching the speed of light. For such objects, time will slow down relative to stationary observers. An example used to illustrate special relativity is a theoretical light clock, where a ball of light bounces against two parallel mirrors. If this light clock is traveling at a certain speed (v) relative to a stationary observer, the light will also be traveling in a horizontal direction in addition to the vertical bounce, forming a pair of right triangles across a single up-down tick. For a local observer traveling with the clock, during this tick, the light will have traveled twice the vertical length (a) of the clock, at the speed of light **c**, whereas for the stationary observer, the light will have traveled twice the length of the hypothenuses (h) of those right triangles. However, the speed of light is constant, and therefore the light of the clock does not travel faster because the clock is already traveling at a certain speed. Instead, the discrepancy is explained by time dilation, whereas a different amount of time elapses for the local observer moving with the clock compared to the stationary observer.

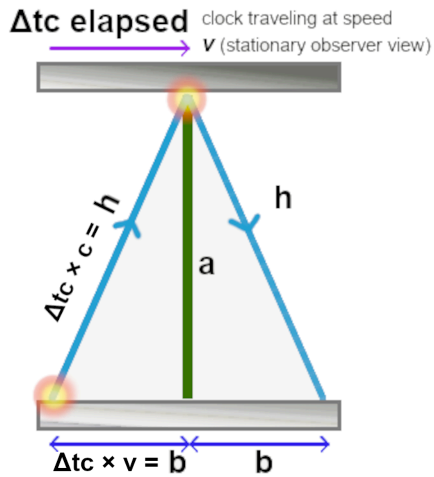

Figure 26 — Light clock moving at speed (v) as observed by a stationary observer, with light traveling distance (h) in time (Δtc)

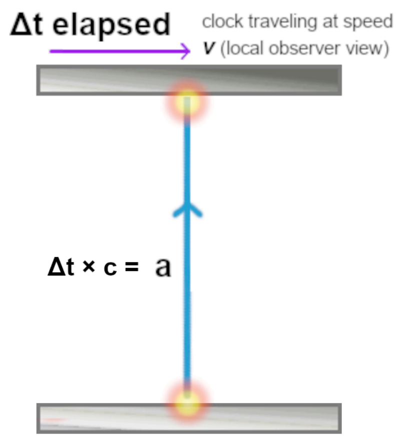

Figure 27 — Light clock as observed by a local observer traveling with the clock at speed (v), with light traveling distance (a) in time (Δt)

If the distance traveled by the moving clock during a full tick is twice the base (b) of those right triangles, the Pythagorean theorem states that $h^2 = a^2 + b^2$. If the clock is traveling a distance (b) at speed (v) during a half tick of the clock taking time (Δtc) according to the stationary observer, the distance (b) can be expressed as Δtc × v, and the distance (h) can be expressed as Δtc × c giving $\Delta tc^2\ c^2 = a^2 + \Delta tc^2\ v^2$. For the local observer however, the light has only traveled across the vertical distance (a), also at the constant speed of light **c**. Therefore for this local observer, a quantity of time Δt = a/c has elapsed, and distance (a) can be expressed as Δt × c. Relating the two equations results in $\Delta tc^2\ c^2 = \Delta t^2\ c^2 + \Delta tc^2\ v^2$, from which we can isolate the ratio Δt/Δtc = $\sqrt{(1 - (v^2 / c^2))}$, which gets smaller as speed (v) approaches the speed of light **c**, causing time for the local observer in motion to advance more slowly compared to the stationary observer. This ratio is the reciprocal of the Lorentz factor (γ). The time (Δt) measured in local observer's own reference frame is called the *proper time*, whereas the time (Δtc) for the stationary observer is called the *coordinate time*.

Scenarios where special relativity is relevant include those involving objects moving at speeds approaching the speed of light, or across very long distances or periods of times over which cumulative effects become significant. To put things into perspective based on some example values of the Lorentz factor:

- for an observer moving at 5% the speed of light (~54 million km/h), time passes 0.1% slower compared to a stationary observer;
- for an observer moving at 10% the speed of light (~108 million km/h), time passes 0.5% slower compared to a stationary observer;
- for an observer moving at 60% the speed of light (~648 million km/h), time passes 25% slower compared to a stationary observer; and
- for an observer moving at 86.66% the speed of light (~935 million km/h), time passes 200% slower compared to a stationary observer.

By comparison, the speed of Voyager 1 probe has reached 61,500 km/h, or around 0.0057% the speed of light, and the fastest man-made object is the Parker Solar probe which will reach a speed of 690,000 km/h, or 0.064% the speed of light, at its closest approach to the Sun in 2025.

Based on how the extent of the Solar System is defined, it can range in diameters between ~4.5 billion km (~30 Astronomical Units (AU)) for the aphelion (furthest point in the orbit) of Neptune, to around 1874 AU for the aphelion of the dwarf planet Sedna, or approximately 0.0296 light year. A hypothetical spacecraft traveling at 5% the speed of light (roughly a thousand times faster than Voyager 1) would travel across that distance of the Solar System in approximately 216 days, with only approximately 5 fewer hours elapsing from that object's perspective due to time dilation.

6.8.3. General relativity

General relativity on the other hand describes gravity as the curvature of space and time by the energy and momentum of matter and radiation. As stated by John Wheeler, *spacetime tells matter how to move; matter tells spacetime how to curve*. One effect of general relativity is that despite being massless, light propagating as a wave is bent when passing near a massive object such as a galaxy due to the curvature of spacetime in an effect called gravitational lensing, which can be observed as distant objects appearing as Einstein rings or multiple images such as the Einstein Cross. Another example effect is that a clock will tick faster near a larger mass.

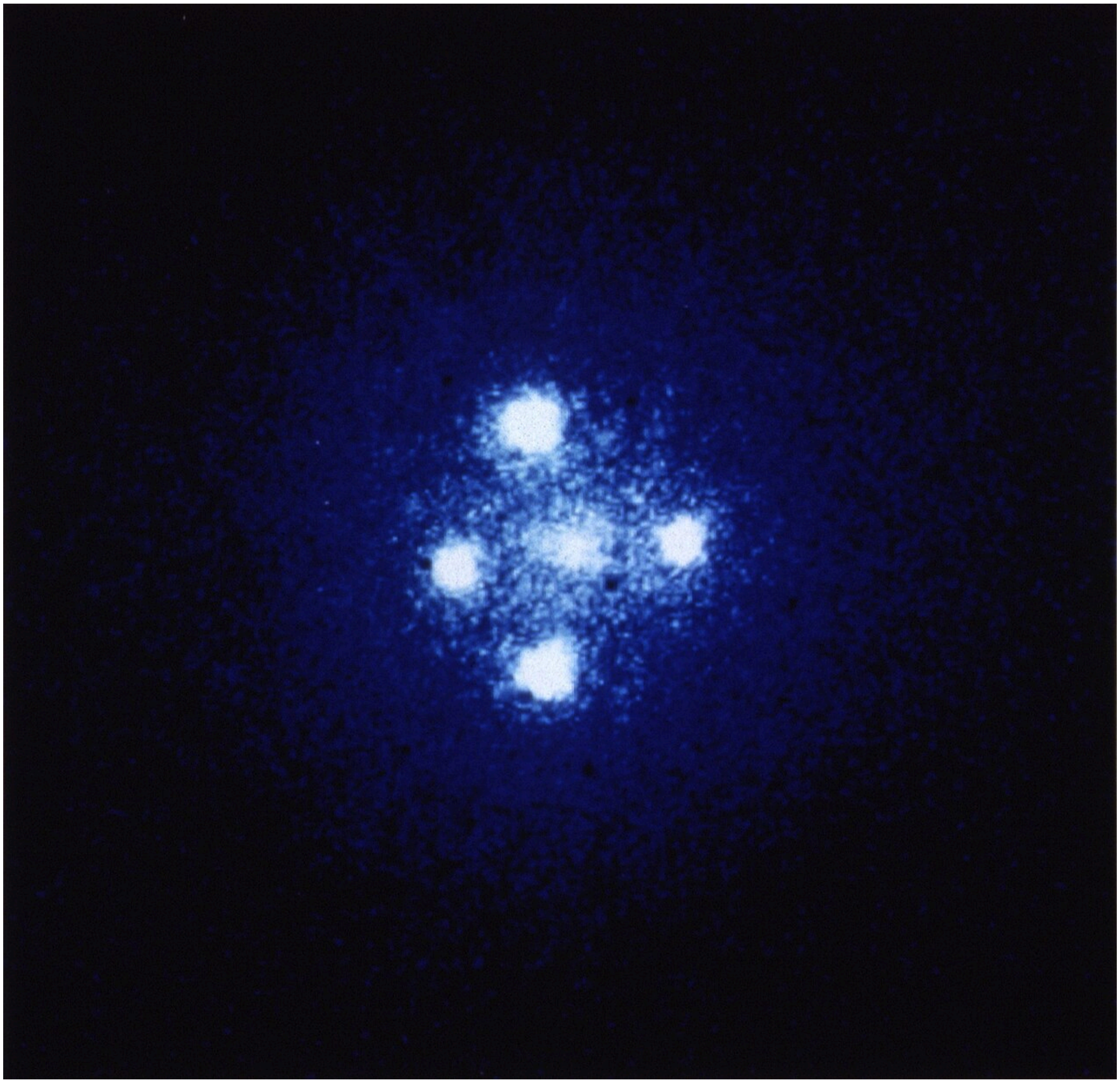

Figure 28 — Einstein cross: Four images of the same distant quasar (NASA, ESA, and STScI)

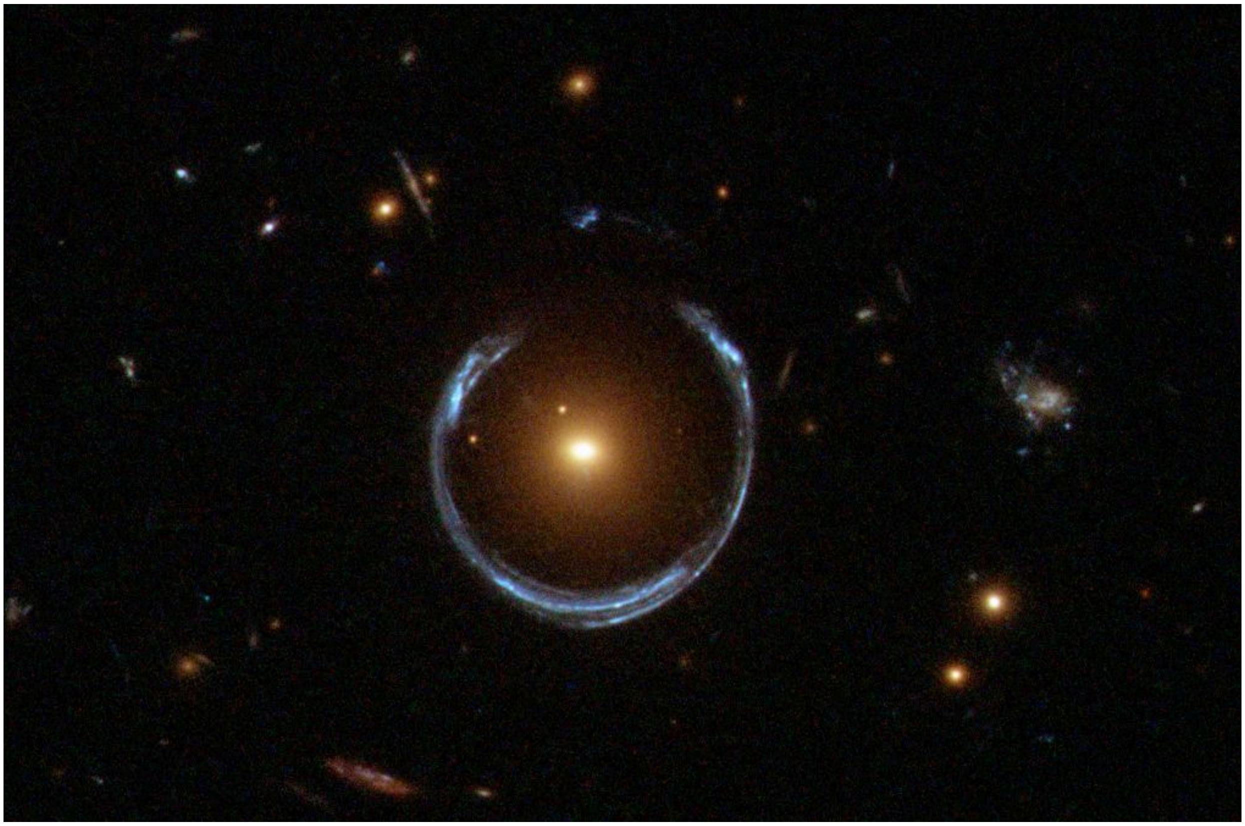

Figure 29 — Einstein ring: a distant blue galaxy is distorted as a ring by the gravity of a luminous red galaxy appearing in its center (NASA)

6.8.4. Relativistic scenarios

The first relativistic scenario that was discussed in Testbed 18 concerned Global Positioning Systems (GPS). In this particular scenario, both special relativity and general relativity have a significant effect, with the general relativity effect having an opposite effect which is actually more significant than that of special relativity. While GPS satellites travel at approximately 14,000 km/h around the Earth, which would cause their clocks to run 7 microseconds (7×10^{-6} seconds) slower per day relative to a stationary clock on the Earth's surface, they are also further away at an altitude of approximately 20,000 km, and therefore are slightly less affected by Earth's gravity, causing the clocks to run 45 microseconds faster per day, resulting in the clocks being 38 microseconds faster per day than a clock on Earth overall.

For several space exploration scenarios, general relativity may contribute more significantly than special relativity, due to the limited speeds at which relevant objects are traveling in space and the presence of massive celestial bodies in their proximity. As one example, due to its close proximity to the Sun, general relativity affects Mercury's orbit significantly.

The use of the BCRS centered on the overall mass of our Solar System and of the Barycentric Coordinate Time (BCT) provides a single consistent frame of reference minimizing the effects of General Relativity for observations within our Solar System, facilitating the exchange and comparison of positions in its vicinity.

Tracking low-Earth-orbit satellites and establishing the relationship between the International Atomic Time and the Barycentric Coordinate Time requires taking into consideration the combined effects of both special and general relativity. The Schwarzschild solution is an exact solution to Einstein's field equations describing a gravitational field around a spherical mass with an assumption that the electric charge and the angular momentum of the mass are zero, providing a simple and precise way to model the time dilation effects for such use cases:

$$\Delta t^2 = (1 - (2GMi / (ri\ c^2)))\ \Delta tc^2 - (1 - (2GMi/(ri\ c^2)))^{-1}\ (\Delta x^2 + \Delta y^2 + \Delta z^2) / c^2$$

where:

- t is the proper time that could be recorded on an atomic clock;
- (x, y, z) are the coordinates of that clock (e.g., in BCRS);
- tc is the coordinate time that would be read on a clock far away from the gravitational masses (e.g., in BCT);
- $\sqrt{(\Delta x^2 + \Delta y^2 + \Delta z^2)} / \Delta tc$ is the velocity of the clock;
- G is the gravitational constant ($6.6743 \times 10^{-11}\ m^3/(kg \times s^2)$);
- c is the speed of light (299,792,458 m/s); and
- Mi are masses in the neighborhood and ri their distances from the clock.

The equation reduces to gravitational time dilation in the absence of motion:

$$\Delta t^2 = (1 - (2GMi / (ri\ c^2)))\ \Delta tc^2$$

$$\Delta t^2/\Delta tc^2 = 1 - (2GMi / (ri\ c^2))$$

$$\Delta t/\Delta tc = \sqrt{(1 - (2GMi / (ri\ c^2)))}$$

The equation also reduces to velocity time dilation in the absence of gravity:

$$\Delta t^2 = \Delta tc^2 - (\Delta x^2 + \Delta y^2 + \Delta z^2) / c^2$$

$$\Delta t^2 = \Delta tc^2 - (v^2\ \Delta tc^2) / c^2$$

$$\Delta t^2 = \Delta tc^2\ (1 - v^2 / c^2)$$

$$\Delta t^2/\Delta tc^2 = 1 - v^2 / c^2$$

$$\Delta t/\Delta tc = \sqrt{(1 - v^2 / c^2)}$$

A

ANNEX A (INFORMATIVE) REVISION HISTORY

A

ANNEX A (INFORMATIVE) REVISION HISTORY

DATE	RELEASE	AUTHOR	PRIMARY CLAUSES MODIFIED	DESCRIPTION
2022-09-30	0.1	J. St-Louis	all	initial version
2023-05-18	0.9	J. St-Louis	all	version for review in preparation for approval vote at 126th MM in Huntsville
2023-07-30	1.0	C. Reed, J. St-Louis	all	applying Carl Reed's review corrections

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Hideki Hayashi, Akinori Asahara, Kyoung-Sook Kim, Ryosuke Shibasaki, Nobuhiro Ishimaru: OGC 16-120r3, *OGC Moving Features Access*. Open Geospatial Consortium (2017). https://docs.ogc.org/is/16-120r3/16-120r3.html.

[2] Kyoung-Sook KIM, Nobuhiro ISHIMARU: OGC 19-045r3, *OGC Moving Features Encoding Extension — JSON*. Open Geospatial Consortium (2020). https://docs.ogc.org/is/19-045r3/19-045r3.html.

[3] OGC: OGC 21-056r9: OGC GeoPose 1.0 Data Exchange Standard, 2022

[4] OGC: OGC 22-029: OGC API — 3D GeoVolumes, 2022

[5] Khronos: Krhonos glTF™ 2.0 Specification, 2021