Open Geospatial Consortium

Submission Date: <yyyy-mm-dd>

Approval Date: <yyyy-mm-dd>

Publication Date: <2021-11-01>

External identifier of this OGC® document: http://www.opengis.net/doc/EG/{standard}/{m.n}

Internal reference number of this OGC® document: YY-nnnrx

Version: 0.4

Category: OGC® Engineering Guidance

Editor: C. Perey, J. Morley, J. Lieberman, M. Salazar, C. Smyth

OGC GeoPose Reviewers Guide

Warning

This document provides guidance for reviewers of the OGC GeoPose Standard. This document is a non-normative resource and not an official position of the OGC membership. It is subject to change without notice and may not be referred to as an OGC Standard. In addition to this guide, developers, implementers and reviewers may wish to study the OGC GeoPose Users Guide. The guidance provided in this document is not to be referenced as required or mandatory technology in procurements.

Document type:	OGC®Engineering Guidance
Document subtype:	
Document stage:	Draft
Document language:	English

Table of Contents

i. Abstract

The GeoPose Reviewers Guide is a public resource structured to provide quick answers to questions which a reviewer may have about the OGC GeoPose specification. It is provided to support professionals who need to understand OGC GeoPose and/or are reviewing the GeoPose draft standard but do not wish to implement it.

GeoPose 1.0 is an OGC Implementation Standard for exchanging the position and orientation of real or virtual geometric objects (Poses) within reference frames anchored to the earth's surface (Geo) or within other astronomical coordinate systems. The standard specifies two Basic forms with no configuration options for common use cases, an Advanced form with more flexibility for more complex applications, and five composite GeoPose structures that support time series plus chain and graph structures.

ii. Keywords

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

GeoPose, ogcdoc, OGC document, OGC Implementation Standard, Geospatially-anchored position and orientation, pose, reviewers

iii. Preface

This version of the GeoPose Reviewers Guide is limited in scope to the draft implementation specification for GeoPose 1.0. Content of this document will be updated when relevant information and feedback to the OGC GeoPose 1.0 SWG is provided and the standard finalized. The Open Geospatial Consortium shall not be held responsible for the accuracy or completeness of this reviewers guide.

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.

iv. Submitting organizations

The OGC GeoPose Standards Working Group submitted this document for publication by the Open Geospatial Consortium (OGC).

v. Submitters

The OGC GeoPose Standards Working Group submitted this document for publication by the Open Geospatial Consortium (OGC).

Introduction

Chapter 1. What is GeoPose?

There are two answers to this question: a conceptual answer and an answer that explains what the standard provides. Conceptual Answer: When a real or digital object’s pose is defined relative to a geographical frame of reference it will be called a [make sure that there’s 100% alignment between this text and the glossary. Link to the glossary definition?]"geographically-anchored pose." All physical world objects inherently have a geographically-anchored pose. Digital objects may be associated with a geographically-anchored pose (for example, in a real-world overlay or on a stage).

What the Standard Provides: The OGC GeoPose standard defines a conceptual model, a logical model and encodings for the position and orientation of a real or a digital object in a machine-readable form in real world coordinates, a geographically-anchored pose.

Chapter 2. Why is Another Standard Needed?

A new standard is required to facilitate the seamless interchange of position and orientation information between proprietary systems and any systems that implement to existing standards. The **review of standards related to GeoPose** demonstrates that there are many relevant specifications that could use GeoPose, however, do not provide the elements.

Chapter 3. How Does OGC GeoPose Address Diverse Requirements?

The draft **OGC GeoPose 1.0 standard** defines data structures for the interoperable exchange of the position and orientation of real or virtual geometric objects (Poses) within reference frames anchored to the earth's surface (Geo). In developing this standard, the SWG sought to use a single conceptual model to address requirements ranging from common use case that benefit from low complexity, and low optionality ("without optional parameters"), to those complex use cases needing high flexibility.

In order to meet the wide range of requirements, the OGC GeoPose specifies:

1. Two basic forms with no configuration options for common use cases,
2. An advanced form with more flexibility for more complex applications, and
3. Five composite structures to support time series plus chain and graph structures.

GeoPose 1.0 is the derived OGC Implementation Standard for exchanging GeoPoses.

Chapter 4. How Did we Define the GeoPose v 1.0 Scope?

While the earth is the focus of the GeoPose 1.0, the specification could also be used in conjunction with other astronomical bodies than the earth.

In the course of developing GeoPose v1.0 and in order to focus on the key objectives of the standard, it was decided to that the following considerations would be out of scope for the v1.0:

- details of any frame transformations (e.g., the radius of the Earth),
- differential properties (i.e., acceleration and velocity) and other physical properties of objects that could be associated with a GeoPose,
- concepts of uncertainty (accuracy and precision),
- camera models or view frustums,
- scaling and other non-rigid transforms,
- interpolation methods in case of complex targets, and
- [FILL IN ANYTHING ELSE WE CAN THINK OF]

NOTE	we could divide up and present this scope differently. We could also say that any of the above could be presented in parallel with GeoPose. Many of the aspects which were excluded could could be introduced in parallel as more properties in a schema.

Chapter 5. Who Will Use the OGC Reviewers Guide?

The GeoPose Reviewers Guide is a resource for those who seek to understand the key concepts used in OGC GeoPose, the requirements it meets and the data structures it specifies.

We intend this guide to be useful for reviewers of the standard as well as decision makers seeking to understand the relevance of this standard in their use cases.

Chapter 6. How To Use This Resource

The GeoPose Reviewers Guide is not intended to be read from start to finish. Rather, it is a resource structured to provide quick answers to questions which a reviewer may have about the OGC GeoPose specification. It is provided to support professionals who need to understand OGC GeoPose and/or are reviewing the GeoPose draft standard but do not wish to implement it.

In addition, this guide can provide insights to professionals considering adopting GeoPose for their projects and products.

The GeoPose Reviewers Guide contains hyperlinks which can be used to navigate directly to relevant sections of the guide as well as to sections of the draft GeoPose specification and the GeoPose User Guide.

Reviewers Guide Scope

The GeoPose Reviewers Guide introduces the the **key concepts used in GeoPose** to its **target audiences**.

To identify broadly applicable requirements for GeoPose, the SWG solicited use cases and chose five that were agreed to be representative in nature. To understand the ways in which GeoPose can be used and how it meets requirements identified, this guide can be used in conjunction with the OGC GeoPose use cases section of the standard.

The choices of https://data.ogc.org/geopose-swg/pdf/geopose_standard.pdf#standardization_targets [standardization targets] made in the GeoPose SWG during standard development are explained in **this section** of the present guide.

Finally, this guide explains how GeoPose fits in the landscape of geospatial computing. It compares GeoPose with approaches that have been **taken in other standards**, and in **open source projects and libraries, and commercial products** for encoding geospatially-anchored position and orientation with six degrees of freedom.

Chapter 7. References

TO BE UPDATED WHEN WE HAVE ALL DONE

The following documents contain provisions that, through reference in this text, constitute provisions of this Reviewers Guide. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the document referred to applies.

- IETF: RFC 2045 & 2046, Multipurpose Internet Mail Extensions (MIME). (November 1996),
- IETF: RFC 3986, Uniform Resource Identifier (URI): Generic Syntax. (January 2005)
- INSPIRE: D2.8.III.2 Data Specification on Buildings – Technical Guidelines. European Commission Joint Research Centre.
- ISO: ISO 19101-1:2014, Geographic information - Reference model - Part 1: Fundamentals

NOTE	Each reference has an anchor. That allows users to jump to this citation from any hyperlinked reference in the text. The second part of the anchor is the text that will be displayed such as RFC 2045

Chapter 8. Terms and Definitions

For the purposes of this document, the following additional terms and definitions apply.

For GeoPose, specifically, there is a visual glossary of frequently used terms. To consult the visual GeoPose glossary, visit the GeoPose wiki.

NOTE	Which of these boilerplate terms/definitions do we want to keep in the GeoPose Reviewer's Guide?

What is a reviewers guide? What is a users guide? What is a standard?

2D data
geometry of features is represented in a two-dimensional space
NOTE In other words, the geometry of 2D data is given using (X,Y) coordinates.
[INSPIRE D2.8.III.2, definition 1]

2.5D data
geometry of features is represented in a three-dimensional space with the constraint that, for each (X,Y) position, there is only one Z
[INSPIRE D2.8.III.2, definition 2]

3D data
Geometry of features is represented in a three-dimensional space.
NOTE In other words, the geometry of 2D data is given using (X,Y,Z) coordinates without any constraints.
[INSPIRE D2.8.III.2, definition 3]

application schema
A set of conceptual schema for data required by one or more applications. An application schema contains selected parts of the base schemas presented in the ORM Information Viewpoint. Designers of application schemas may extend or restrict the types defined in the base schemas to define appropriate types for an application domain. Application schemas are information models for a specific information community.
OGC Definitions Register at http://www.opengis.net/def/glossary/term/ApplicationSchema

codelist
A value domain including a code for each permissible value.

conceptual model
model that defines concepts of a universe of discourse
[ISO 19101-1:2014, 4.1.5]

conceptual schema

1. formal description of a conceptual model
   [ISO 19101-1:2014, 4.1.6]
2. base schema. Formal description of the model of any geospatial information. Application schemas are built from conceptual schemas.

OGC Definitions Register at http://www.opengis.net/def/glossary/term/ConceptualSchema

Implementation Specification
Specified on the OGC Document Types Register at http://www.opengis.net/def/doc-type/is

NOTE	Notice that each definition has an anchor. Anchor text would also be a good idea which we may include latter. Terms used within a definition should be cross-linked to their definition if it is included in this document (see Application Schema for an example.

Chapter 9. GeoPose Glossary

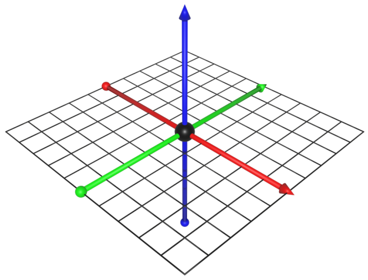
Conceptual Overview

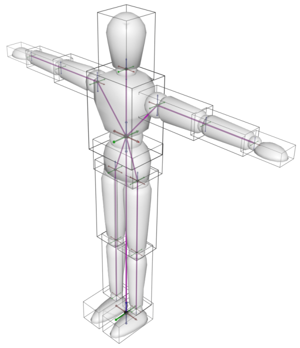
This overview was originally created to facilitate the communication between GeoPose SWG members to reach a shared understanding of concepts related to GeoPose. It has been integrated into this guide to provide readers with a better understanding of the conceptual foundations on which the standard is based.

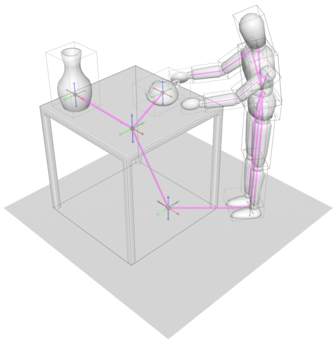
The definition of each term is accompanied by a visual representation of the concept, a more general explanation and examples of how it may be applied. Additionally, the text includes hyperlinks to connect the different concepts to one another and to illustrate how they are used in other standards and real world applications.

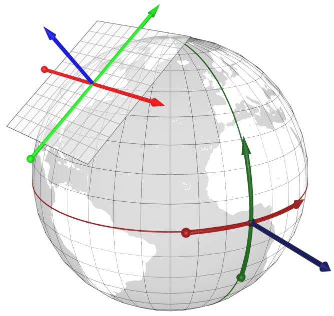
Chapter 10. Key Concepts

The following list of terms covers the conceptual underpinnings for the GeoPose standard while offering examples of how they are actually employed in different scientific fields and industry sectors.

(Reference) Frame  Frame.blend	**As defined in ISO 19111, a reference Frame is a parameter or set of parameters that realize the position of the origin, the scale, and the orientation of a coordinate system.** Another, more practical definition from Wikipedia is: *"In physics, a (reference) frame consists of an abstract coordinate system and the set of physical reference points that uniquely fix (locate and orient) the coordinate system and standardize measurements within that frame."* A Reference Frame can be visualized as a subspace with its own coordinate system (represented with an arrow for each axis and a 2D grid to facilitate measurements) and a point of origin (represented as a sphere at the point where the arrows intersect). Generally, to simplify calculations, the associated coordinate systems are Cartesian in nature, but in the case of non-euclidean spaces, they require more advanced definitions (like the geographical coordinate systems used in global positioning systems).

Pose  Pose.blend	**"A Pose determines the position and orientation of an object (or group of objects) relative to a coordinate system.”** In Robotics and Human-Computer Interaction, this concept is often referred to as Six Degrees of Freedom (6DoF) and it is used to specify the position and orientation of an object with great precision (generally, by combining a **Local Positioning System** for the translation and an **Aircraft Principal Axes Orientation System** for the rotation). In this way, it is possible to define complex kinematic systems that allow the creation of advanced robotic arms and exoskeletons. From the point of view of computer graphics, however, a Pose can be seen as a simplified version of a "Transform class" (without the **Scale**-related fields/properties). This direct correspondence implies that a Pose is associated to a single node of an Scene Graph (i.e., the "hand" has a Pose, the "arm" has another Pose, etc.), but, when 3D artists and engineers talk about "Poses", they are generally referring to the combined Pose data of all the components of an entity (i.e., the Pose of a dummy, as seen in the illustration). In fact, most 3D editing software solutions include a separate "Pose mode" so that artists can independently work on the geometry of the different 3D models and on their animation.

Pose Chain  SceneGraph.blend	**"A Pose Chain is a sequence of Poses that is used to determine the position and orientation of objects across multiple Frames."** When dealing with multiple reference Frames (specially, when they are of different types), it is necessary to perform several complex transformation operations to convert between the different Coordinate Reference Systems. To facilitate this time-consuming process and ensuring that the operations are applied in the right order, the different poses are often organized into one-dimensional sequences of Poses called "Chains". In many applications, where a large number of entities have similar Pose Chains, it is common to reorganize this structure into a Graph, so that it is possible to more effectively apply the transformation operations (by storing the result of the repeated operations). In Computer Graphics, this concept is the basis of the "Scene Graph", in which the Transform class instances of the different entities are connected together using a hierarchical graph through (single) parent-children relationships. In this way, the Pose of each entity of the graph is calculated in a depth-first manner that minimizes the number of required calculations. NOTE: The definition of GeoPose needs to say that "a GeoPose" is something that complies with the specification but a "geographically-anchored pose" is a concept which can be encoded in proprietary manners or in manner compliant with the standard

OGC GeoPose  GeoPose.blend	**"A GeoPose is a Pose in relation to a real-world object (by default, Earth), typically, via a Geographical Reference Frame."** The goal of the standardization of GeoPose is to establish a single mechanism to encode the position and orientation of an real/virtual objects within a geographical (ellipsoidal) Frame, greatly simplifying the determination of the pose of entities relative to an astronomical object and, consequently, relative to each other. In practical terms, a GeoPose facilitates the application of a transformation operations to properly position and orientate modes in a Scene Graph (generally, the root one) on a realistic 3D map and/or in an Augmented/Mixed Reality environment.

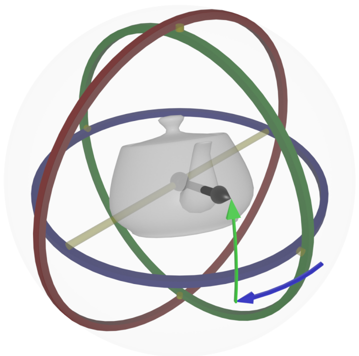
Location	**"Not to be confused with the term Position, a location is a distinctive region of a space."** While a Position merely specifies a single point within a space (relative to a Frame), a Location defines a limited region within a space (often with its very own Frame). A location generally receives a unique name within a context (or within another location) to easily identify it among others and, if necessary, it can also have a additional semantic information to further differentiate it (e.g., "Washington D.C.", "Washington State" and "Washington City, Utah"). There are as many types of locations as there are types of spaces (e.g., "the bottom of screen", "behind the car", "Earth's orbit", etc.), but it is specially relevant in the context of Geography, where the *Geo*location ("In which street/road/town I am?") is often much more important than the actual *Geo*position ("what are my GPS coordinates?"). The boundaries of Locations are often defined using either dimensional properties (i.e., width, height and depth) or specific shapes (most notably, 2D projections in a geographical space called Geofences). However, when there are a large amount of locations or these are constantly changing, the boundaries are defined by proximity to the closest point in the topological skeleton or by the minimum number of logical connections.

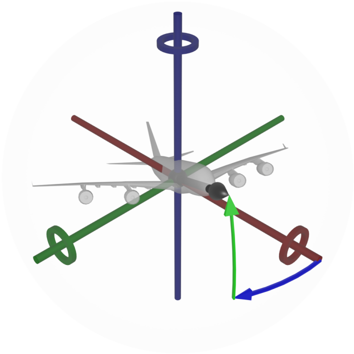
Chapter 11. Orientation Systems

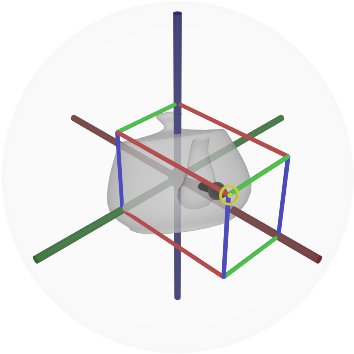
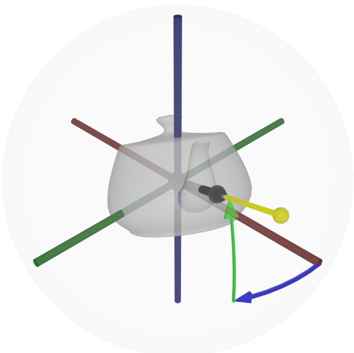
This section covers a system to accurately describe the orientation of an object relative to the associated reference frame. As with positioning counterparts, these systems provide a vector that needs to be consistently interpreted (so that it can be converted into a rotation matrix), not only to describe the numeric values in the proper sequence, but also to note the units and to specify the type of orientation system to which the values refer.

MOVE THIS REMARK TO ANOTHER PLACE: While orientation systems also have to account for inaccuracies, they are considered less important because they can generally be corrected afterwards (or recalculated using two position vectors) with relative ease.

Not all applications require orientation information to define a Pose.

Euler Angles  EulerAngles.blend	**"The Euler angles are three angles introduced by Leonhard Euler to describe the orientation of a rigid body with respect to a fixed coordinate system."** (from Wikipedia) The angles are generally specified in degrees (in a similar way to longitude/latitude values in Global Positioning Systems), although the internal operations on computers are almost always performed in radians, one axis after another. Usually, the application order of rotation transformations is the inverse of the specification (the Z angle is applied first, then the Y on, and finally, the X one), but it might vary depending on the use case or the hand rule employed. It is important to note that, since the rotation operations are applied globally and linked together by the application order, if two axes are driven into a parallel configuration, it can generate a Gimbal Lock, resulting in the loss of one degree of freedom. This is usually not a problem when defining simple, discontinuous orientations, but when the use case involves overlapping or interpolating between multiple Euler angles, the result can be negatively affected.

Aircraft Principal Axes  AircraftPrincipledAxes.blend	**"Aircraft Principal Axes define the relative (local) rotation of an object, using a plane flying in the +X axis as a reference."** This system defines the orientation of an object as a combination of rotations along the three axes of an aircraft: yaw (or normal) axis, pitch (or transverse) axis and roll (or longitudinal) axis. These axes have a direct correspondence with the X, Y and Z axis of the (right-handed) Euler system and are defined with the same units. However, since these axes are applied locally (i.e., rotating the coordinate system of the object at each step), the order of application becomes irrelevant. The Aircraft Principled Axes system has the advantage of being very intuitive (once the users fully understand what axis is related to the terms yaw, pitch and roll) and of not being susceptible to Gimbal Lock issues. On the other hand, this orientation system often requires additional computational power to generate the intermediate rotation matrices (one for each axis with a value different from 0) and, while the image of an airplane can be projected onto other vehicles, it might be difficult to do so with other static elements (e.g., it doesn't make sense to define the "roll angle" of buildings or any other static structures).

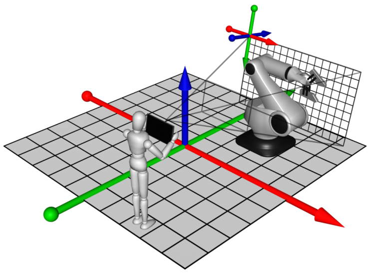
Quaternions*  Quaternion.blend	**"In computer graphics, a Quaternion is 4D complex number that is used to specify orientations in a 3D space."** In essence, a Quaternion is a mathematical construct with 3 imaginary components (i, j and k) and one real component. When it is used to describe 3D orientations, each imaginary component describes the dimension of a **unitary** vector in a particular axis of the space (i, j and k are assigned to the X, Y and Z axis, respectively) whereas the real component expresses **half** of the rotation around that unitary vector, expressed in radians. However, to ensure that the resulting axial vector is unitary (i.e., its length equals one), a quaternion has to follow the fundamental formula $i^2 = j^2 = k^2 = i \cdot j \cdot k = -1$. Due to their mathematical construction and the possibility to interpolate them both linearly and spherically, most 3D engines use Quaternions internally to orientate and animate compound objects. However, the complex nature and the interdependence between imaginary components, make Quaternions very difficult for most human beings to understand and use, so they are rarely employed in user interfaces or human-readable documents.
Look-At Systems  LookAtSystem.blend	**"A Look-At system enables the definition of orientation relative to another object or position vector."** Similar to Quaternions, Look-At systems use a vector (obtained from the difference of the target position minus the current position) to determine the orientation of an object. After normalizing the vector, it is possible to use its components to create a quaternion and apply the same mathematical operations. However, to properly determine the rotation angle around the axial vector (the real component of a quaternion), it is necessary to also provide an "up" vector to serve as a pivot.

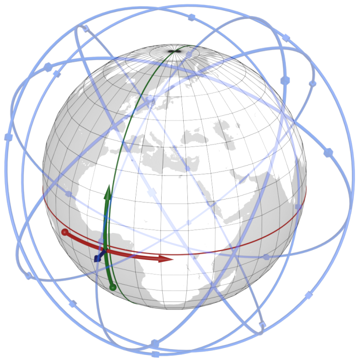
Chapter 12. Positioning Systems

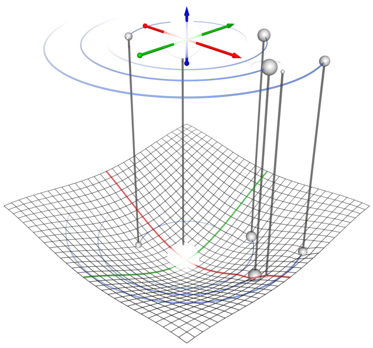
This section covers a system to accurately describe the position of an object relative to the associated reference frame. The position of an object is a single point in space relative to the origin point of the reference frame. This point is usually placed in the center of mass of the object and provides a simple way to perform other spatial calculations (i.e., distances between objects, travel time, collision detection/avoidance, etc.). In contrast, with a camera, the point could be its focal point.

The objective of positioning systems is to provide a position vector (that is often converted into a translation matrix) as precise as possible within a reference frame. These vectors are often accompanied with an (in)accuracy value to be able to establish an uncertainty threshold for different tasks.

We distinguish the concepts of position and location. Location is a broader term that can be represented in many forms, including textual forms such as the address. While it is possible to define a position using a known location (e.g., "This building is in Tokyo"), this practice often results in imprecise information.

Local Systems  LocalPositioning.blend	**"A Local Positioning System defines the location of objects relative to a Frame based on Euclidean spaces."** These frames are often defined with an arbitrary point of origin depending on the nature of the subspace to represent. When working with physical spaces or virtual 3D scenes, the point of origin is generally situated in the center of the space (to avoid the limitations of single-precision floating-point numbers). However, when operating within 2D spaces with clear boundaries, like screens or documents, the point of origin is normally placed in the top-left corner (due to the graphic systems based on the left-to-right, top-to-bottom writing direction used in most western countries). Needless to say, this makes the transformations between the different frames (i.e., determining what 3D object has the user selected on a 2D screen) a very cumbersome process. In recent years, advancements in Indoor Positioning Systems technologies have enabled the precise tracking of real-world elements in relatively large **local** subspaces. Yet, when a use case has more than a single location or the curvature of the Earth becomes an important factor to take into account, using a Global Location System is generally a better option.

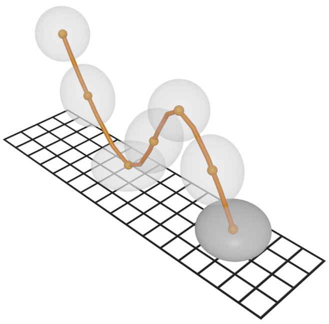
Global Systems  GlobalPositioning.blend	**"A Global Positioning System defines the location of objects relative to a Frame based on spherical spaces, generally associated with the planet Earth."** As its name implies, a Global Positioning System is defined by a spherical object that establishes the properties of a **Frame** itself (even if the astronomical body is not a perfect sphere). The resulting subspace is delimited in the two horizontal dimensions of the surface of a sphere (**longitude** should only have values between -180 and 180 degrees, whereas the **latitude** values should be within a -90 to 90 degrees) and partially delimited in the vertical dimension (**altitude** values should never be lower than the negative value of the radius of the sphere). As for the point of origin of this subspace, for practical reasons, it is generally placed on an arbitrary point alongside the equator of the sphere (thus, also defining an offset value for the altitude values). Traditionally, determining the position of an object on the surface of the Earth with any degree of accuracy required cumbersome tools and manual calculations. Fortunately, nowadays, the existence of satellite-based radio-navigation systems like GPS, Galileo or NavIC has greatly simplified the process and -as long as the devices can establish visual contact with three satellites- it is possible to track the position of all kinds of objects on Earth.

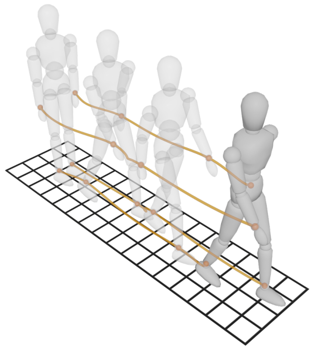
Universal Systems  UniversalPositioning.blend	**"A Universal Positioning System defines the location of objects relative to an Inertial Frame based on Minkowski spaces."** These systems aim to provide a more precise and truly unique positioning for objects relative to an inertial point in the physical Universe (generally, the barycenter of the Solar System). However, since the Minkowski spaces are defined by the rules of Special Relativity, the very subspaces are "curved" by Lorentz Transformations and any traversal is limited by the Speed of Light. Needless to say, the extremely complex calculations that these positioning systems require relegating them to applications related with interplanetary travel and communications. For example, the SPICE framework created by the NASA's Planetary Science Division operates with both the J2000 and ICRF Inertial Frames to calculate transfer orbits between bodies of the Solar System.

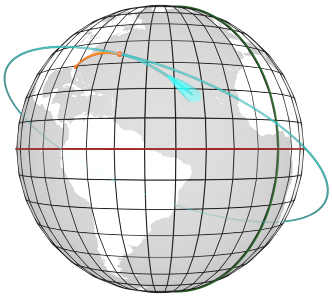
Chapter 13. Time-related Concepts

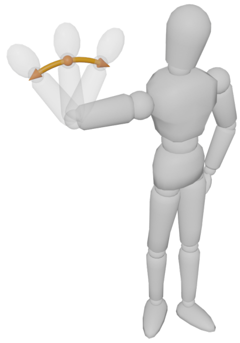
While, by definition, a pose defines the position and orientation of an object at a particular point in time, there are many applications that require the use of temporal sequences of poses. Properly taking into account the temporal dimension requires consideration of additional factors like (moment of) inertia, mass, changes in shape and, in extreme cases, even the variations in the Earth's gravity field.

To help represent this complexity, the following list of terms is intended to offer a basic introduction to the topic.

Animation  Animation.blend	**"Animation is a method to create the illusion of movement through a collection of still images."** In computer graphics, animation is generally simulated by modifying the properties of an object (including its position, rotation, scale, material colors, etc.) over a period of time (or timeline). Since complex deformations require a large number of physics calculations, most animation systems either operate with properties independently (i.e, separating position, orientation and scale) or define skeleton-based systems to alter the geometry of the object in a controllable way. In recent years, however, many 3D engines incorporate procedural subsystems to simulate advanced animations (such as explosions, wind systems, collision deformations, etc) in real-time, with relatively high degree of precision. Internally, each change in properties is stored into an "animation key", and when there is more than one key over time, an "animation curve" is generated to specify how to interpolate between them. A collection of curves is called an "animation clip" and can be associated to a given action or event.

Motion  Motion.blend	**"A Motion is a type of Animation that only defines the movement (position+orientation keys) of objects."** In many use cases, it is possible to simplify complex animations by segmenting an object (generally, using a skeleton structure) and reducing the types of keys to just position and orientation (creating animation curves known as Motion Paths). This idea of only taking into consideration the movement of different parts of an object is the basis of motion capture techniques; a series of technologies that, nowadays, can even record subtle facial expressions of human beings. Another interesting point of view is that motions can also be defined as transitions between different poses of an object. In this way, instead of having to define a single sequence of animation keys, it is possible to "jump" between different poses by merely interpolating between them whenever necessary (something especially useful when dealing with real-time camera -motion- tracking systems). It is important to note that a motion between two poses also generates a direction vector and an attitude (orientation relative to that direction vector).

Trajectory  Trajectory.blend	**"A trajectory is a precalculated Motion Path that attempts to describe the future movement of an object."** While motion paths describe the actual movements over time, in many use cases it is imperative to also be able to predict the future movement of an object (e.g., to avoid collisions or reach a location/orbit with a very specific speed and orientation). These special motion paths are called **Trajectories** and are usually calculated by simply taking into consideration the inertia of an object. This is especially important for objects placed in a stable trajectories relative to others objects (usually, astronomical bodies), because if those trajectories are entirely predictable, it is possible to create an Ephemeris system to rapidly calculate their position at any point in time. Additionally, if a predictable trajectory is regularly repeated over time, it can also be called a stable Orbit.

Gesture  Gesture.blend	**"A Gesture is a Pose or Motion used to convey a non-vocal message."** Gestures are a form of communication that involve the positioning/movement of a part of the body in a predetermined way to express a particular concept. And although this type of communication is used by many living beings, humans have used them to develop complete sign languages or to serve as a method of identification. Some basic gestures even allow people to overcome linguistic barriers and create a culture around specific gestures. Nevertheless, since messages conveyed by gestures require both the issuer to be able to make the gestures and the receiver to be able to interpret them correctly, both the communication channel (the physical space) and the meaning of each gesture has to be previously defined to avoid misunderstandings. It is important to clarify that gestures can also be conveyed by a visual representations (e.g., the Unicode symbol U+1F44C "□") or with robotic systems that emulate the body structure and movements of the living beings (which is one of the basis of the Animal-Robot Interaction field).

The GeoPose Standard

This section describes the key elements of the GeoPose standard, especially the conceptual and logical models, and the implementation targets which have been derived from the logical model. The development of the standard has been led by a number of use cases and use case domains, as summarised in the diagram below.

INSERT the figure and explain it here or simply point the reader of the reviewers guide to that section?

Chapter 14. Conceptual Model

The GeoPose Conceptual Model consists of linked definitions of terms denoting concepts expressed in the GeoPose Logical Model and structural data unit specifications for the implementation targets.

The Conceptual Model is defined in this section of the GeoPose Standard.

NOTE	Change the link above to an anchor, not to the .adoc file

Chapter 15. Logical Model

While the Conceptual Model outlines the component terms and relationships in the standard, the Logical Model precisely models the classes and their relationships.

The Conceptual Model describes a (non-normative) domain of discourse for terms used in defining a precise Logical Model (normative) expressed as a Unified Modeling Language (UML) [ref] class diagram.

The scope of the Logical Model is a subset of the scope of the Conceptual Model. The scope of the implementation targets is a subset of the scope of the Logical Model.

The Implementation Targets are mutually independent implementations of subsets of the Logical Model.

Chapter 16. Encodings for Implementation

Christine proposes we use some or all of this text (from the spec) as an introduction to this chapter.

The core abstraction in the OGC GeoPose Standard is the Frame Transform. This is a representation of the transformation taking an Outer Frame coordinate system to an Inner Frame coordinate system. This abstraction is constrained in GeoPose v 1.0 to only allow transformations involving translation and rotation. The intention is to match the usual concept of a pose as a location and orientation. The formalism that expresses a GeoPose Frame Transform is a pair of Reference Frames, Outer and Inner, each defined by a Frame Specification.

Implementation of position and orientation encodings that are compliant with the OGC GeoPose standard is accomplished by following the standard's requirements for one or more of the eight data objects referred to in the specification as the "Standardization Targets".

Summary of the Eight OGC GeoPose Standardization Targets:

- The two Basic GeoPose targets are simple and concise – no options. They satisfy most use case requirements and assume a local tangent plane ENU frame derived from WGS84.
- An Advanced GeoPose target supports more complex use cases where the outer geographic reference frame is not LTP-ENU and/or a valid time is needed. Composite:
- The Chain GeoPose target provides additional flexibility with multiple intermediate frames or specific coordinate reference systems as needed.
- The Frame Graph target supports the full structure need to represent networks of reference frames that arise with the use of multiple and linked location technologies.
- The three (Time) Sequence targets support the packaging of fixed-length time series of GeoPoses and the payload data objects for open-ended GeoPose streams.

Depending on the use case and requirements, a developer can implement support for one or more of these targets.

There are automated tests that can be used to determine whether an actual JSON-encoded GeoPose data object conforms to the standard.

NOTE This is where we explain the rationale for the eight standardization targets that were chosen.

In the majority of use cases, requirements are met by using either:

- **Basic-Euler** GeoPose encoding when orientation is in Euler angles
- **Basic-Quaternion** GeoPose encoding when orientation is in quaternion

The reason that there are two versions of Basic GeoPose is that some developers prefer a human-readable encoding. In these cases, the Basic-Euler is chosen. In other use cases, orientation by quaternion is preferred.

The two differences between Basic GeoPose and Advanced GeoPose is that, first, in Advanced GeoPose, the developer can specify an outer geographic reference frame other than LTP-ENU. In addition, in the Advanced GeoPose encoding, a single time stamp can be provided. The time stamp pertains only to the single encoding so one of the Composite GeoPose encodings is not needed.

The three Composite GeoPose encodings are designed for when there are linked and sequential GeoPoses. The Chain GeoPose is a linear set of linked poses. The Graph GeoPose is used when there are linked poses but not necessarily in a linear sequence.

For time series poses with constant time spacing, the developer will choose to use the Regular Timeseries GeoPose encoding. When there is a per-GeoPose time stamp that is not at a regular interval, the developer will choose to use the Irregular Timeseries GeoPose encoding. Finally, in use cases that do not have a pre-defined end time (also referred to as an open-ended sequence of time-stamped GeoPoses), the developer will specify the Stream GeoPose encoding.

GeoPose in Context

This section we discuss the place of GeoPose in the world - how GeoPose is situated in the landscape of standards, and the opportunities for GeoPose to standardise the interaction between standards which manage some representation of geographically anchored position and orientation.

In this section of the reviewer’s guide, we list the Standards Development Organizations (SDOs) and standards that specify [refer to the GeoPose implementation targets in the draft spec][a GeoPose implementation target] as a data format. As a convention for the remainder of this section, we use the term "GeoPose" as a shorthand for any implementation target.

The key question this section will answer is where and how GeoPose fits into the landscape of standards. By examining some of the relevant standards| this section illustrates the gaps which GeoPose fills.

There are many existing and emerging standards for position and orientation information. They have emerged from requirements defined in different industries: aviation, planetary sciences, maritime, robotics, autonomous driving, satellite positioning and aerospace, to name a few. There is good practice in commercial and other domains for expressing the position and orientation of entities in all these fields. These existing standards cover different scales, for different purposes, different information environment: sometimes graphical, sometimes geospatial.

Chapter 17. GeoPose connects objects to their representation

There are conceptual and actual data pipelines that connect the real world and the objects in it with the representations in information and graphics systems. A number of the critical standards define how those pipelines are developed and interoperate or fit together.

In a standard Web browser, information is displayed on a plane. There is a need for a standard to represent information retrieved from the Web on the 2D display plane in a manner that is sufficiently fast to provide XR experiences. W3C WebXR focuses on this need.

In order to establish this data flow, there is a need in real graphics systems to locate those objects within a 3D local stage (or scene) so that when the user’s head or perceived objects move within that “stage”, the graphics systems can address those movements and changes correctly. Khronos OpenXR focuses on that stage of the pipeline.

On the other side, the OGC has been working on how sensor systems measure these changes in position and orientation. How does a system model the sensor that’s capturing all the data about the features (e.g. objects) in its environment, including where they are and how the system represents them in a local 2D or 3D environment? OGC Sensor Web Enablement, particularly Observations & Measurements and SensorML, addresses these needs.

Each of these stages in the pipeline need to exchange data between themselves. How do you get the position and orientation of anything from anywhere to anywhere, into and out of these interactive environments? That’s where GeoPose comes in. GeoPose defines the data structure(s) to pass position and orientation information between elements in the pipeline.

Chapter 18. GeoPose in the Landscape of Standards

The primary source of inspiration for GeoPose was the NASA SPICE framework because it is able to cover any scale from interplanetary and interstellar to specific local objects. NASA designed SPICE to address significant challenges in looking at both ephemeris objects (fixed, or on predictable paths) and objects that have changeable positions and orientations such as satellites, cameras, and other sensors. Representing different frames for these objects and being able to transform between them is really useful. SPICE is a formalism that is much larger than one needs for a simple or basic implementation| but an incredibly appropriate foundation from which the SWG was inspired.

OGC Moving Features was also taken into account when GeoPose was designed. Although it describes object position and orientation, Moving Features (MF) is focused on a particular set of use cases with an emphasis on sensor streams, digital exhausts| and location information (usually GPS) coming off of vehicles in a municipal environment. It accomplishes this compactly so that the data can be easily incorporated into analytical and visual applications.

OGC Moving Features and GeoPose have distinct roles and are complementary. Moving Features is focused on a local, municipal scale, and rapid streams of measurements. Getting observations in and out of municipal management and other platforms easily and efficiently is one of the roles that GeoPose plays.

The OGC Sensor Web Enablement suite of standards deals with how to work with sensors and getting useful observations out of them.

image::images/landscape-standard-ef0fe.png On the right of this figure is a schematic showing the workflows that are enabled in Observations and Measurements, in Semantic Sensor Networks, and with encodings such as SWE Common that enable transport of sensor outputs (observations). Another important part of these standards, SensorML, models the sensor process itself, from an initial environmental stimulus to how a measurement is recorded as an observation and recognized as an observed property of a real world object.

If this is a directed sensor (e.g. camera), orientation is an essential aspect of the sensor model. The result potentially includes the positions and orientations of both the sensor / platform and any observed entities in the world, although the sensor position and orientation may be secondary to the primary objective of observing the positions and orientations of these observed entities, e.g. cars. GeoPose permits systems to get the position and orientation in and out of SWE-compliant platforms or devices without losing any resolution or introducing delays.

Pose is essential for combining physical and digital entities in visual scenes for them to be used by a service or a user, particularly if entities are being brought into a scene from very different source contexts, and regardless of where they are in space.

Khronos OpenXR handles poses, particularly within specific spaces (frames of reference). It defines a set of reference spaces (view, local and stage); and specifies the model in which graphics hardware can use the pose in rendering objects. Within a particular graphical system, it is effective but GeoPose adds the capability to bring in a pose from any source in any frame of reference.

GeoPose can also relate the frame of reference of a Web browser window to a virtual world or the real world.

Geocentric (earth-based) position and orientation are the basis for all these integrations. OGC GeoPose provides that usable common ground, both the geospatial expertise that OGC has cultivated for many years and digital representation of physical space as the most common denominator among all these systems and representations.

To summarize, there are a number of well-developed standards for position and orientation. What these lack is a means for position and orientation information to be passed between them in a manner that is independent of graphical system, applications scene, frame of reference, and technology. OGC GeoPose offers portability of information between all these domains and systems.

The approaches to this issue that have been published in other standards prior to introduction of GeoPose appear in the tables below.

Chapter 19. Standards which Could Reference GeoPose in Normative Clauses

19.1. OGC Standards

The OGC has many positioning and location standards, some also express orientation. They do so in different scales and with different global and/or local coordinate reference systems. Some also deal with different time scales. However, these standards are not designed for sharing position and orientation.

Some standards (such as OGC CDB) deal with fixed infrastructure, or with somewhat more specialized information, such as KML and IndoorGML. Some deal with expressing location and orientation in very dynamic and real time scales, such as Sensor Web Enablement and Moving Features.

Standard	**Graphic/Virtual Context**	**Local SRS**	**Geodetic SRS**	**6DOF: as entity or attributes?**	**Temporality**	**Remark**
Moving Features	?	Y	Y	Attributes of temporal geometry	Y	
Sensor Web Enablement (SWE)						
CityGML	Y	Y	Y	Y (?)	Y	
IndoorGML		Y	Y			
"CDB (Common Database)"	?	?	?	?	?	
KML			Y			
Observations and Measurements	?	?	?	?	?	
SensorThings API	?	?	?	?	?	
IMDF	?	?	?	?	?	

Standard	Graphic/Virtual Context	Local SRS	Geodetic SRS	6DOF: as entity or attributes?	Temporality	Remark
3D Tiles	Y	Y	Y	"x,y,z+normal"	Y	"3D Tiles is basically a binary, encapsulated glTF with georeferencing. There are efforts to make glTF more ""geospatially friendly"". → include glTF (Khronos Group) in the list."

19.2. Other SDOs

There are other standards development organizations (SDO's) that deal with location and orientation for graphics. They are compiled in the tables below. Work done in the W3C defines how systems express location and orientation for browsers. The Motion Imagery Standards Board (MISB) has standards for moving cameras. ISO also has sections of its standards in SC 24| such as the X3D standards| that encode orientation and position in graphics. In the Khronos Group| there are standards such as OpenXR and glTF that specify how to form digital assets that encode position and orientation

Khronos Group

Standard	Graphic/Virtual Context	Geographically-referenced Local SRS	Geodedic CRS	6DOF as entity or attribute?	Temporality
glTF	?	?	?	?	?
OpenXR	?	?	?	?	?
OpenVX	?	?	?	?	?

|This OpenXR Extension for Microsoft Spatial Anchors allows an application to create a spatial anchor| an arbitrary freespace point in the user's physical environment that will then be tracked by the runtime. The runtime should then adjust the position and orientation of that anchor's origin over time as needed| independently of all other spaces and anchors| to ensure that it maintains its

original mapping to the real world.

W3C

Standard	Graphic/Virtual Context	Geographically-referenced Local SRS	Geodedic CRS	6DOF as entity or attribute?	Temporality
Geolocation API	?	?	?	?	?
Browser Sensor Interfaces	?	?	?	?	?
Immersive Web WebXR Device API	?	?	?	?	?

XRSpace and XR Pose |An XRSpace represents a virtual coordinate system with an origin that corresponds to a physical location. Spatial data that is requested from the API or given to the API is always expressed in relation to a specific XRSpace at the time of a specific XRFrame. Numeric values such as pose positions are coordinates in that space relative to its origin. The interface is intentionally opaque.

Motion Imagery Standards Board (MISB)

Standard	Graphic/Virtual Context	Geographically-referenced Local SRS	Geodedic CRS	6DOF as entity or attribute?	Temporality
MISB ST 0601	?	?	?	?	?
MISB ST 0801.5	?	?	?	?	?

BuildingSmart

Standard	Graphic/Virtual Context	Geographically-referenced Local SRS	Geodedic CRS	6DOF as entity or attribute?	Temporality
IFC	Y	?	Y	No	?

IfcSite and other IfCProducts permits topologic orientation, but not 6DOF. IFCSite lets users provide the WGS84 location (lat,lng,alt) of "the single geographic reference point for this site " http://standards.buildingsmart.org/MVD/RELEASE/IFC4/ADD2_TC1/RV1_2/HTML/schema/ifcproductextension/lexical/ifcsite.htm For orientation they refer to the concept of "true north": "The world coordinate system, established at the IfcProject.RepresentationContexts, may include a definition of the true north within the XY plane of the world coordinate system, if provided, it can be obtained at IfcGeometricRepresentationContext.TrueNorth."

ASTM

Standard	Graphic/Virtual Context	Geographically-referenced Local SRS	Geodedic CRS	6DOF as entity or attribute?	Temporality
E57	defines fifteen features that cover the core capabilities of the E57 format	?	?	?	?

There are also specifications (standards) that are developed for and used by industries/domains.

19.3. Space

The Observation Geometry System NASA uses for Space Science Missions is called SPICE. A tutorial presentation about SPICE is available here.

NASA

Standard	Relevant Section	Quote the Text
SPICE	Frame Kernel	

Also I must create a table dedicated to IEEE Standards. What are the IEEE standards?

What about ISO standards?

This URL is a convenient place to view many space data standards URL: http://spacedatastandards.org/

Chapter 20. Software that Use or Generate Geospatially-anchored Position and Orientation

This section of the reviewer’s guide captures information about commercial software and open source software and libraries that use or generate geospatially-anchored poses. It illustrates the diverse ways that geospatially-anchored poses are stored and represented.

The exercise shows the high potential to increase interoperability between a wide range of existing solutions when a developer chooses to implement the OGC GeoPose standard.

The table below documents, in a structured format, the features and functions of solutions identified at the time of publication of this guide.

Since new software is being introduced to address the requirements of use cases covered by GeoPose, this section could be maintained/updated periodically. If you have information to contribute to this section, or have recommendations and questions, please create an issue in this repo.

Company or Open Source Project Name	Product or Service Name	Open Source?	"When this solution offers, generates or uses pose, is the pose (A) geospatially-anchored or (B) does it use an internally-defined (local) Frame of Reference?"			Does this solution use a static spatial reference system? Can the adopter specify their SRS?		Please provide details about the spatial reference system used in this system		"In this solution, how is position represented?"			"In this solution, how are orientation and rotation represented?"		"In this solution, is temporal information associated with poses?"	"If temporal information is associated, provide details"	Comments

			GeoSpatially-anchored FoR	Internally-defined (local) FoR						Globally (latitude and longitude)	Locally				yes=Green		
Away Team Software Ltd	Web Video Map Tracks		Both are possible - either camera heading or moving object heading with camera yaw (from heading)			Static spatial reference system		WGS84		Globally (latitude and longitude)			Heading pitch roll			Camera orientation is sampled periodically and intermediate values can be calculated by interpolation	

Magic Leap	ML1		?????			Static spatial reference system							Quaternions				
HERE Technologies	LiveSight					Static spatial reference system				Globally (latitude and longitude)			"Yaw, pitch and roll"			timestamp	

FlightSafety International	Professional Pilot Training					Static spatial reference system		WGS84		"Globally (latitude and longitude), Some positions are relative to objects that are moving (based on user controls)"			"Yaw, pitch and roll, Quaternions, Euler Angles"		"Poses change with time, but the temporal information is not saved"		

Epic Games	Unreal Engine		Both : A local frame of reference which is itself geospatially anchored			Spatial reference system is defined for each use		"Users can choose the CRS or their choice, as long as they have a WKT or EPSG code."					"Yaw, pitch and roll, Quaternions, Euler Angles"				
Arvizio Inc.	Arvizio Immerse 3D		It could be both depending on data type and specific project			Spatial reference system is defined for each use		The products contains extendable database of projections and geoids		Could be both			"Yaw, pitch and roll"		May be in certain situations since the product supports animation	Using animation one can support changing of object(s) poses	

"Cesium, Inc."	Cesium ion and CesiumJS using 3D Tiles					Static spatial reference system		WGS84		"EPSG:4978 (earth-centered/earth-fixed). For precision reasons, keep both local and local-to-global transform data (location and rotation)."			"3D Tiles data captures the information using the standard graphics approach - transform matrices. Additional options beyond transform matrices (e.g., quaternions, heading			"Cesium JS can display time-dynamic data provided as CZML or KML or via API. The data contains samples of position over time and CesiumJS uses interpolation to create the complete path.	

Hexagon	LuciadFusion 2020.1 and LuciadRIA 2020.1					Spatial reference system is defined for each use		"We support any spatial reference system for models. For the world, we typically use EPSG:4978 (geocentric reference)"		Globally (latitude and longitude)			Euler Angles				

Leica Geosystems AG part of Hexagon	"Leica Imaging Totalstation / Multistation, Tilted & Imaging GNSS"		Both depending on the solution (example:for total station imaging camera is local)			Static spatial reference system		ECF or Local		Globally (latitude and longitude)			"Quaternions, Euler Angles"			Timestamp	

Hexagon AB / myVR Software AS	myVR XRToolkit (SDK)					Spatial reference system is defined for each use		"Multiple systems depending on use case, either data dependent or application defined."		Globally (latitude and longitude)			"Yaw, pitch and roll, Quaternions"		Application defined	Application defined	

Esri	Oriented Imagery					Spatial reference system is defined for each use		It can be any spatial reference system.		Globally (latitude and longitude)			"Yaw, pitch and roll, Euler Angles, Euler Angles with two rotations about z axis and one about x axis in order z-x-z"			Acquisition Date parameter in Oriented Imagery Schema stores the temporal information	

Ecere	GNOSIS Cartographer		"Normally geospatially anchored, but local transformations can be anchored to those geospatial anchors"			"Normally WGS84 is used, but we want to improve support for different epochs / realizations of WGS84, and other CRS can be converted to our internal WGS84 representation as		WGS84		Globally (latitude and longitude)			"Yaw, pitch and roll, Quaternions, The question is tricky as Yaw, Pitch, Roll Euler are also called Euler angles. Wikipedia distinguishes between ""Proper Euler angles""		"Not currently, but there certainly is value in doing so, though it could be provided alongside the pose."		

Graphmetrix Inc.	Trinpod					Static spatial reference system		WGS84 - nested objects ultimately have an event based reference back to lat/long/elev		Events are used from start to finish to capture object state and motion using nested oriented reference frames that ultimately resolve to WGS84 at any nesting level			Quaternions			Events with start time/location and end time/location are used for all changes to entities	

Norwegian Mapping Authority	Border Go	Yes	The GeoPose library maintains an estimate of the geospatial position and orientation of a real-world anchored local frame of reference.	"The local cartesian system (frame of reference) in the AR session parallell to the local tangent plane of the WGS84 ellipsoid and has east north up axis, in addition there is		there is a geodesy library that allows the use of geospatial data sets using different SRS’s		WGS84		"latitude, longitude and altitude above/below the reference ellipsoid in meters"	"two cartesian coordinate systems, one that holds the position of the origin in the earth anchored reference frame that is estimated to be at a lat,lng,alt with LTP ENU		Quaternions			GeoPose is estimated continously based on various sensor data (sensor fusion) and a phyiscal model that realies on tampstamps of the meassurements that updates the estimate.	

Fantasmo	Camera Positioning Standard	Proposed															

Cesium	Cesium.js VelocityOrientationProperty	Yes	ECEF	local cartesian				WGS84		"ECEF + lat,lng, alt"	local cartesian frames of reference		Quaternion			the orientation are estimated based on a stream of points that typically are temporally arranged like a flight trajectory	Not an original account by Cesium.js developer. Based on Jan-Erik Vinje reading som of the docs such as this: https://cesium.com/learn/cesiumjs/ref-doc/VelocityOrientationProperty

Robot Operating System (ROS)	geographic_msgs/GeoPose.msg	Yes	GeoPose	One can also use local cartesian frames of reference.		It can at least handle UTM and WGS84		WGS84 + UTM		"WGS84 (Lat, lng, alt) + UTM"	local cartesian frames of reference		Quaternion			Robots are by their nature dynamic and ROS provides mechanism for GeoPose streaming	Not an original account by ROS developer or user. Based on Jan-Erik Vinje reading som of the docs http://docs.ros.org/

Below is a table of companies who publish products about which all relevant information has yet to be captured.

Company	**Product Name**	**Local and/or Geospatial Pose**	**SRS Variable or Static**	**Naming Conventions**	**Data Model Details**
Autodesk	Cell in column 2	local	Cell in column 4	Cell in column 5	Cell in column 6
Bentley	Context Capture	Cell in column 3	Cell in column 4	Cell in column 5	Cell in column 6
Deeyook	Cell in column 2	Cell in column 3	Cell in column 4	Cell in column 5	Cell in column 6

Company	Product Name	Local and/or Geospatial Pose	SRS Variable or Static	Naming Conventions	Data Model Details
Esri	ArcGIS Runtime	Cell in column 3	Cell in column 4	Cell in column 5	Cell in column 6
Esri	ArcGISARView (built on SceneView)	Cell in column 3	Cell in column 4	Cell in column 5	Cell in column 6
Facebook	Scape.io	Cell in column 3	Cell in column 4	Cell in column 5	Cell in column 6
Google	Visual Positioning Service	Cell in column 3	Cell in column 4	Cell in column 5	Cell in column 6
Google	Maps	Cell in column 3	Cell in column 4	Cell in column 5	Cell in column 6
Google	Chrome?	Cell in column 3	Cell in column 4	Cell in column 5	Cell in column 6
HERE Technologies	Visual Positioning Service Proof of Concept with Verizon	Cell in column 3	sub-meter accuracy using an image or video. Proprietary 3D positioning algorithms from HERE analyze images or videos for accurate positioning. https://t.her.is/2GjHvCf	Cell in column 5	Cell in column 6
Immersal (now Hexagon)	Cell in column 2	local	Cell in column 4	Cell in column 5	Cell in column 6
Lyft	BlueVision	Cell in column 3	Cell in column 4	Cell in column 5	Cell in column 6
Microsoft	Azure Spatial Anchors	Cell in column 3	Cell in column 4	Cell in column 5	Cell in column 6
Niantic	Cell in column 2	Cell in column 3	Cell in column 4	Cell in column 5	Cell in column 6
PTC	Vuforia	local	Cell in column 4	Cell in column 5	Cell in column 6

Company	Product Name	Local and/or Geospatial Pose	SRS Variable or Static	Naming Conventions	Data Model Details
SPAR3D	Cell in column 2	Cell in column 3	Cell in column 4	Cell in column 5	Cell in column 6
Sturfee	Cell in column 2	Cell in column 3	Cell in column 4	Cell in column 5	Cell in column 6
Trimble	Cell in column 2	Cell in column 3	Cell in column 4	Cell in column 5	Cell in column 6
Uber	Cell in column 2	Cell in column 3	Cell in column 4	Cell in column 5	Cell in column 6
Verses	Cell in column 2	Cell in column 3	Cell in column 4	Cell in column 5	Cell in column 6
vGIS	Cell in column 2	Cell in column 3	Cell in column 4	Cell in column 5	Cell in column 6
Visometry	Vision Lib	Local	Cell in column 4	Cell in column 5	Cell in column 6
Visualix (now acquired)	Cell in column 2	Local	Cell in column 4	Cell in column 5	Cell in column 6

Annex A: Revision History

Date	Release	Editor	Primary clauses modified	Description
2021-06-15	0.1	C.Perey and J. Morley	all	initial version
2021-09-30	0.9	C.Perey and J. Morley	all	initial version

Annex B: Bibliography

NOTE

Example Bibliography (Delete this note).

The TC has approved Springer LNCS as the official document citation type.

Springer LNCS is widely used in technical and computer science journals and other publications

- For citations in the text please use square brackets and consecutive numbers: [1], [2], [3]

– Actual References:

[n] Journal: Author Surname, A.: Title. Publication Title. Volume number, Issue number, Pages Used (Year Published)

[n] Web: Author Surname, A.: Title, http://Website-Url

[1] OGC: OGC Testbed 12 Annex B: Architecture. (2015).