**3GPP TSG-SA WG4 Meeting #130S4-241846r1**

**US, Orlando, 18 – 22 November 2024**

**Source: Orange**

**Title: Pseudo-CR on Spatial Description Functions**

**Spec: 3GPP TR 26.819 v0.1.0**

**Agenda item: 9.9**

**Document for: Agreement**

**1. Introduction**

The Study on Spatial Computing for AR Services [FS\_ARSpatial] was approved during SA#104 meeting. The core spatial computing functions have been documented in TR 26.819 in the SA4-129e meeting. Spatial computing includes functions such as world tracking, relocalization, anchoring, 3D model construction, collider generation, segmentation and label, light extraction. The resulting output of spatial computing is a set of spatial mapping information that is organized in a data structure called the XR Spatial Description for storing and exchanging the information. This contribution adds details about Trackables in the anchor function and details how they can be leveraged in the context of relocalization.

**2. Reason for Change**

Anchoring content to the real world is often ensured through the detection and tracking of given object in the real world which can be a predefined image or mesh, a body part of the user, or a flat surface in the user surrounding.

**3. Proposal**

It is proposed to agree the following changes to 3GPP TR 26.819 v0.1.0.

\* \* \* First Change \* \* \* \*

### 4.2.1 General

XR spatial computing encompasses a set of functions which process sensor data to generate information about the world 3D space surrounding the AR user. These include functions such as relocalization (establishing the position of users and objects within that space), anchoring, 3D Model construction, segmentation and labeling (semantic perception) and light extraction, are achieved through a combination of advanced sensors, cameras, and algorithms that enable devices to understand and interact with the three-dimensional space around them.

This requires accurately localizing the AR device worn by the end-user in relation to a spatial coordinate system of the real-world space. Vision-based localization systems reconstruct a sparse spatial mapping of the real-world space in parallel (e.g., SLAM). Beyond the localization within a world coordinate system, which is usually based on a sparse spatial map, dense spatial mapping of objects is also essential in order to place 3D objects on real surfaces and provides the ability to occlude objects behind surfaces, do physics-based interactions based on surface properties, provide navigation functions, or provide a visualization of the surface.

For the purpose of understanding and perceiving the scene semantically, machine-learning and/or artificial intelligence may be used to provide context for the observed scene.

The output of spatial computing is spatial mapping information that is organized in a data structure called the *XR Spatial Description* for storing and exchanging the information. Some spatial computing functions may also take an XR spatial description and may result in updates to the XR spatial description. Spatial computing functions typically include data exchange and require a network architecture.

An AR device may provide sensor data to the spatial computing function to create or update the spatial mapping information. The device may also access the spatial computing function to retrieve different spatial mapping information depending on the needs of the XR application.

The spatial computing functions can run locally on the AR device or can be executed remotely on the cloud or on the edge in a “spatial computing server” accessed through dedicated interfaces as detailed in TR 26.298 [2].

The main functions provided by a spatial computing service are given in Figure 4.2.1-1and explained in the following subclauses.

\* \* \* Next Change \* \* \* \*

### 4.2.3 Relocalization

Relocalization is a function that is used to estimate the pose of the AR device at initialization, when tracking is lost, or regularly to correct the drift of the tracking (TR 26.998 [3] clause 4.2.3). Cameras capture the real world, while sensors (accelerometers, gyroscopes, and depth sensors) contribute additional data for mapping and positioning. Computer vision algorithms process this data to determine the location and orientation of the device relative to its environment.

Relocalization estimates the pose of the AR device according to a known real environement.

SLAM, Visual Localization, e.g., Visual SLAM (vSLAM), or Visual Positioning System (VPS) are all algorithms that can be used for mapping unknown environments while also maintaining the localization of the device/user within that environment, as explained in TR 26.928 [2] clause 4.1.4. Combining a spatial description of the real world with the estimation of the pose of different trackables can also be used for localization as proposed with the World Storage and World Analysis components of the ETSI ARF architecture [21].

For relocalization, the following input data can be used:

- Sensor data:

- images (for SLAM)

- Depth maps

- GNSS data

- local pose of AR Device

The output of the relocalization function is a pose, which is defined by a position and an orientation.

\* \* \* Next Change \* \* \* \*

### 4.2.4 Anchoring

As defined in TS 26.119, a Trackable is a real-world object that can be tracked by the XR runtime. Each trackable provides a local reference space, also known as a trackable space, in which an anchor can be expressed. Examples of trackables that AR runtimes can track include:

* **Image markers:** This includes fiducial markers (such as QR codes) and natural image markers
* **Meshes:** These can be scanned with a dedicated tool or stored in specific formats (CAD for instance)
* **Body parts:** This encompasses user skeletons and facial bones.
* **Planes:** Flat surfaces in the physical environment that come with both tracking and shape information.
* **Geo Trackables:** A Geo Trackable is a position and orientation on earth in a geodetic referential.

ARKit and ARCore all support image, geotrackable, body and plane detection and tracking. Example of body tracking (here the face) with ARCore is given in Figure 1. ARKit also supports meshes as demonstrated in Figure 2.

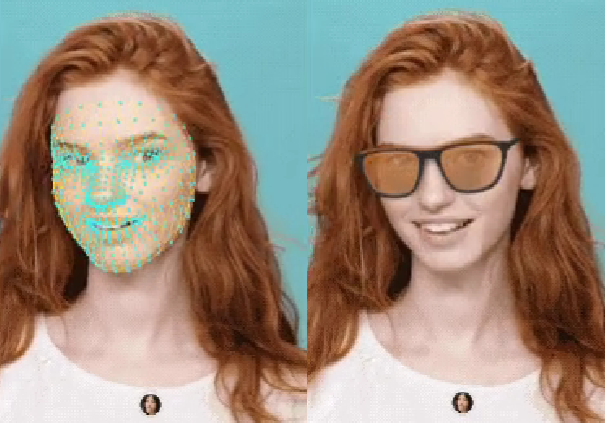


Figure 4.2.3-1 ARCore Face Tracking



Figure 4.2.3-2 ARKit 3D mesh tracking

Like Trackables, Spatial Anchors are a concept in Augmented Reality which serve as fixed reference points that AR devices can detect and use to position virtual content accurately relative to the real-world coordinates. Virtual objects or information are anchored to specific locations in physical space. But Spatial Anchors have additional functionalities enabling the persistence and stability of virtual content in the real world.

A spatial anchor acts as a marker or reference point in the real world that AR devices can recognize and track. Spatial anchors are created by leveraging spatial mapping techniques, which involve capturing and analyzing the physical features of the surrounding environment. This can be done through depth-sensing cameras, LiDAR scanners, or other sensors to understand the geometry and spatial characteristics of the space. These techniques help applications identify stable, distinctive features in the space, allowing precise placement of anchors that can persist and be tracked across different sessions.

Applications can use one anchor per virtual object, or choose to have multiple virtual objects use the same anchor.

Once the spatial mapping is performed, spatial anchors are placed at desired locations within the mapped environment. Anchor can be persistent for the entire duration of the session.

The three main AR SDK (Meta, Google and Apple) offer anchoring solution:

- ARCore, Cloud Anchor [6]

- ARKit, World Map [7]

- Meta Quest, Spatial Anchor [8]

For creating spatial anchors, the following input data can be used:

- Anchor pose for creation

- Sensor data:

- images

- pose of AR device

- depth sensor (e.g. LiDAR) data

The Anchoring function provides several services:

* Creation: when a new trackable is created, information is extracted from the trackable and an anchor is created with a relative pose to the trackable. A unique identifier of the newly created trackable is returned.
* Consumption: from a trackable ID, the function retrieves the stored trackable and all of its information. Sensor data and the features of the trackable are used to compute the pose of the anchor(s) associated with the trackable. When the trackable is detected, the anchors are positioned, and their poses are returned.
* Synchronization: This function enables spatial anchors to be synchronized across multiple AR devices, allowing for shared AR experiences, e.g. for collaborative applications where multiple users interact with the same virtual content.
* Persistence: This function enables spatial anchors to be saved and retrieved across sessions. This means that virtual objects can remain in the same physical location even after the AR application is closed and reopened.

\* \* \* End of Changes \* \* \* \*