**3GPP TSG-SA WG4 Meeting #132S4-250859**

**Fukuoka, Japan, 19 – 23 May 2025**

**Source: Tencent**

**Title: [FS\_ARSpatial] pCR on Semantic perception**

**Agenda item: 9.8**

**Document for: Agreement**

**1. Introduction**

The present contribution proposes to address the semantic perception aspects in clause 4.2.9 currently empty and associated proposed syntax in clause 4.2.11.

**2. Proposal**

It is proposed to agree the following changes to the 3GPP draft TR 26.819 v0.4.0

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

### 4.2.9 Semantic perception

#### 4.2.9.1 Introduction

The semantic perception refers to the use of artificial intelligence and machine learning (AI/ML) techniques to interpret and understand the real-world environment captured by AR devices. This function enables the identification, classification, and contextual understanding of objects, surfaces, and scenes in both 2D and 3D spaces.

The semantic perception builds upon the outputs of other spatial computing functions such as segmentation and labelling (as defined in clause 4.2.6), and 3D model reconstruction (as defined in clause 4.2.5). It enhances the XR experience by enabling applications to reason about the environment, support intelligent interactions, and adapt content based on the recognized context. It may be performed locally on the AR device or remotely on edge/cloud servers, depending on the computational complexity and latency requirements. The output of this function may be integrated into the XR Spatial Description and used by other spatial computing functions or the scene manager.

#### 4.2.9.2 Input data

- Sensor data:

- Images captured by AR device

- Depth maps

- Pose of AR device

- 3D models (meshes or point clouds)

- Predefined label sets or ontologies (optional)

- Output data:

- Output data of the segmentation & labelling function

- Semantic representation (representation of objects and their semantic or spatial/temporal relationships (e.g., adjacency, containment)

~~- Semantic Map: A structured representation of recognized objects and their spatial relationships.~~

~~- Labels: Semantic categories assigned to detected objects or regions (e.g., “chair”, “wall”, “screen”).~~

~~- Scene Graph: A graph-based representation of objects and their semantic or spatial relationships (e.g., adjacency, containment).~~

\* \* \* Second Change \* \* \*

### 4.2.11 Summary of spatial description formats

This section describes the common output data formats for the spatial computing functions defined in clause 4.2. Table 1 provides a list of the output data for each spatial computing function and the corresponding format.

Table – Output data of spatial computing functions.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function** | **Data** | **Description** | **Format Example** | **Possible Representation** |
| World Tracking | Feature Map | Features to compute a pose in XR Space. | Array of features |  |
| Feature | 3D point associated with a descriptor, the descriptor format depends on the feature type. | SIFT (Scale Invariant Feature Transform) | Typically, a vector of 128 floats. |
| ORB – Oriented FAST Rotated BRIEF (Binary descriptor) | Typically, a 32-bit descriptor. |
| BRIEF – Binary Robust Independent Elementary Features) | Typically, 32-bit descriptor. |
| FREAK – Fast Retina Keypoint | Typically, 64-bit descriptor. |
| SURF – Speeded-Up Robust Features | Typically, vector of 64-bit or 128-bit floats. |
| Relocalization | Pose | Pose of the AR device in XR Space | Position: 3D vector  Orientation: quaternion | Position: 3 floating point values.  Oriantation: 4 floating point values |
| XR\_space\_id | An identifier for the XR Space used as a reference. | String |  |
| Anchoring | Id | An identifier of the new trackable (when the anchoring function is invoked by a producer). | String |  |
| Pose | Pose of the anchor in relation to the device according to a trackable (when the anchoring function is invoked by a consumer). | Position: 3D vector  Orientation: quaternion | Position: 3 floating point values.  Oriantation: 4 floating point values |
| 3D Model Construction | Model | 3D model(s) of surrounding space. | Non-segmented model (e.g., point cloud, mesh with/without attributes) | OBJ, STL, PLY [41], FBX, glTF |
| Segmented model | glTF, USD |
| Semantic Perception | Semantic Map | A structured representation of recognized objects and their spatial relationships | Graph, list or array of labelled objects | Each object includes a label (string), position (3D vector), and optionally a confidence score. |
| Labels | Set of semantic labels assigned to detected objects or regions | String array | ["chair", "table", "wall", "car", "door", ...] |
| Scene Graph | Graph of objects with semantic relationships (e.g., adjacency, containment) | Nodes represent objects; edges represent spatial or semantic relationship | JSON, RDF, or MPEG-I SD |
| Collider Generation | Colliders | Set of colliders (not combined with associated objects). | Array of colliders |  |
| Collider | A collider object. | Primitive (e.g., sphere with radius) |  |
| Mesh | OBJ, STL, FBX |
| Colliders | Part of description of the 3D Model (when combined with associated objects). | Hierarchical node graph | MPEG-I SD, USD [46] |
| Segmentation and Labeling | Objects | A list of 3D objects resulting from the segmentation and labeling function. | Array or a hierarchical node graph. |  |
| Object | A 3D object in the captured world. |  |  |
| Mesh | Segmented object. | Mesh |  |
| Label | Label of segmented object. | String |  |
| Light Extraction | Lights | Set of extracted lights | Array |  |
| Light | A description of a light source that includes a set of parameters that depend on the type of the light. Possible types include: point, directional, area, spot, texture-based, or image-based light. | Object | For point light, the parameters include: pose, intensity, color, and range. |
| Real Object Removal | Objects | A set of textured 3D objects. | Mesh + texture | OBJ, STL, PLY, FBX, glTF |
| Images | Images with transparency. |  | PNG, TIFF, TGA, BMP |

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