**3GPP TSG-SA WG4 Meeting #128 S4-241065**

**Jeju,South Korea 4**

|  |
| --- |
| *CR-Form-v12.2* |
| **CHANGE REQUEST** |
|  |
|  | **26.927** | **pCR** |  | **rev** |  | **Current version:** | **0.7.0** |  |
|  |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: comprehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* |
|  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME | **x** | Radio Access Network |  | Core Network | **x** |

|  |
| --- |
|  |
| ***Title:***  | pCR Crowd sourced media capture and synthesis |
|  |  |
| ***Source to WG:*** | Nokia  |
| ***Source to TSG:*** | S4 |
|  |  |
| ***Work item code:*** | AI4Media |  | ***Date:*** | 14th May 2024 |
|  |  |  |  |  |
| ***Category:*** | B |  | ***Release:*** | Rel-19 |
|  | *Use one of the following categories:****F*** *(correction)****A*** *(mirror corresponding to a change in an earlier release)****B*** *(addition of feature),* ***C*** *(functional modification of feature)****D*** *(editorial modification)*Detailed explanations of the above categories canbe found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | *Use one of the following releases:Rel-8 (Release 8)Rel-9 (Release 9)Rel-10 (Release 10)Rel-11 (Release 11)…Rel-16 (Release 16)Rel-17 (Release 17)Rel-18 (Release 18)Rel-19 (Release 19)* |
|  |  |
| ***Reason for change:*** | Adding details to Clause 4.2.4 on crowd sourcing  |
|  |  |
| ***Summary of change:*** | Addition of a scenario on crowdsourced media capture, processing and synthesis in clause 4.2.4 using implicit neural representations |
|  |  |
| ***Consequences if not approved:*** | The TR does not capture use of implicit neural representation in crowdsourced media capture and synthesis |
|  |  |
| ***Clauses affected:*** | 4.2.4 |
|  |  |
|  | **Y** | **N** |  |  |
| ***Other specs*** |  | **x** |  Other core specifications  | TS/TR ... CR ...  |
| ***affected:*** |  | **x** |  Test specifications | TS/TR ... CR ...  |
| ***(show related CRs)*** |  | **x** |  O&M Specifications | TS/TR ... CR ...  |
|  |  |
| ***Other comments:*** |  |
|  |  |
| ***This CR's revision history:*** |  |

Introduction

Implicit Neural Representations (INR) of scenes such as Neural Radiance Fields (NeRF) represent 3D scenes as continuous and differentiable functions. INRs have achieved state of the art results on 3D scene view synthesis on consumer grade hardware in a relatively short time. Owing to their success and potentiall applicability in a wide range of use cases, contributions about INRs have been presented and discussed in SA4 in recent meetings [S4-240083](https://www.3gpp.org/ftp/TSG_SA/WG4_CODEC/TSGS4_127_Sophia-Antipolis/Docs/S4-240083.zip), [S4-240459](https://www.3gpp.org/ftp/TSG_SA/WG4_CODEC/TSGS4_127_Sophia-Antipolis/Docs/S4-240459.zip), [S4-240499](https://www.3gpp.org/ftp/TSG_SA/WG4_CODEC/TSGS4_127_Sophia-Antipolis/Docs/S4-240499.zip), [S4-240080](https://www.3gpp.org/ftp/TSG_SA/WG4_CODEC/TSGS4_127_Sophia-Antipolis/Docs/S4-240080.zip), The discussions targeted INR use cases like AR scene representations in [S4-240083](https://www.3gpp.org/ftp/TSG_SA/WG4_CODEC/TSGS4_127_Sophia-Antipolis/Docs/S4-240083.zip), [S4-240459](https://www.3gpp.org/ftp/TSG_SA/WG4_CODEC/TSGS4_127_Sophia-Antipolis/Docs/S4-240459.zip), [S4-240499](https://www.3gpp.org/ftp/TSG_SA/WG4_CODEC/TSGS4_127_Sophia-Antipolis/Docs/S4-240499.zip) and Avatar representations [S4-240080](https://www.3gpp.org/ftp/TSG_SA/WG4_CODEC/TSGS4_127_Sophia-Antipolis/Docs/S4-240080.zip). TR 26.998 now documents INRs such as NeRFs as possible AR scene representation in Clause 4.4.4 while the Permanent Document of FS\_AVATAR study documents INR representations of Avatars in Clause 3.2

Capturing scenes using Neural Radiance Field or implicit neural representations is an active area of research. We propose a scenario for deployment of such an application in 3GPP networks where the application provider leverages crowdsourcing enabled by the network.

Implicit Neural Networks, such as NeRFs can be computationally complex, consequently, the inference on NeRFs i.e. novel view synthesis may be carried out in a network inference engine at the request of a UE. Recent developments in NeRF research such as “[Real-Time Neural Light Field on Mobile Devices (CVPR 2023)](https://snap-research.github.io/MobileR2L/)” has shown that appropriately designed NeRF models can be run even in real time on cosumer UEs. Capablities of the target deployment UEs would need to be considered during the design of such NeRFs tagerted towards inference on UEs.

|  |
| --- |
| 1st Change |

### 4.2.4 Crowd-sourcing media capture

#### 4.2.4.1 Introduction

This use case and its corresponding scenarios are based on clause 6.2 of TR 22.874 [aa]. A set of users attending a live concert and capturing the event on their UEs, use a shared (or a set of shared) DNN model(s) to process and improve their respective captured video and/or audio. Audio and video data may be captured in a noisy environment or an environment with poor lighting conditions. Multiple tasks may then be performed on the processed video and/or audio for media content analysis, e.g. to extract lyrics, annotate the video, improve audio and video quality, translate language, anonymize a face, etc. One such task may be creating Implicit Neural Representations (INR) of the event. The UEs share image or video data with a model training server in the network which trains and generates an implicit neural representation, such as a Neural Radiance Field (NeRF), of the event. The UEs request novel or enhanced views from the NeRF. The network may also train a set of NeRFs for the event suitable for different UEs which vary in their inference capabilities and platform support. Device appropriate NeRF models may be downloaded by the UEs and used for view synthesis locally.

This use case involves two different scenarios based on either a device inference or a network inference.

#### 4.2.4.2 Device inference

The main scenario is to improve the media capture of each UE by using an up-to-date model adapted to the context event.

This scenario may involve the distribution of multiple models to a large number of UEs in a short period of time. The UEs are heterogeneous, running with different types of operating systems (e.g., Android or iOS), supporting different AI/ML engines/frameworks or having different GPU/CPU/NPU and RAM capabilities available for running the AI/ML service on the UE. This will need the distribution of a huge amount of various AI/ML models adapted to the different device capabilities. Depending on each user’s UE, the UE may request the download of a set of DNN models for device inference.

Moving or changing the environment (localization, energy, processing unit, memory, etc.) may need AI/ML model updates, where the DNN models stored in the network may be adapted or updated during the service.

The AI/ML application may optimize the end-to-end latency (e.g., to achieve latency below 1s) or the expected accuracy level of the inference result (e.g., to achieve image recognition precision of 99%) by modifying the model. The desired latency and/or accuracy level can therefore impact the size of the AI/ML model to be distributed. This can be done by:

- optimizing the model accuracy and latency for on-device execution. The model accuracy and execution latency are known, and the optimization may result in bandwidth saving.

- compressing the model for reducing the bandwidth usage and improving the delivery latency. This may affect the accuracy of the model.

If an uncompressed model is sent, accuracy is not affected but delivery latency would depend on the size of the model and the network bandwidth.

The distribution of the AI/ML models for a large number of UEs at the same time may also need to serve the models from different endpoints (e.g., cloud, edge, or other UEs), and may use several or different communication links (e.g. unicast, multicast or broadcast).

#### 4.2.4.3 Network inference

The main scenario may be the sharing of the input media from multiple sources for network inference, as well as the selection of suitable DNN models according to the UE and/or task.

This scenario requests the UE to upload the media data for network inference. Similarly, to the UE inference, DNN models stored in the network may be adapted or updated during the service for network inferences.

|  |
| --- |
| End of change |

|  |
| --- |
| 2nd Change |

#### 4.2.4.4 Scenario: Crowdsourced Implicit Neural Representation

4.2.4.4.1 Crowdsourcing content

**Figure 4.2.4.4-1: Crowdsourced NeRF creation**

In a crowdsourced NeRF generation scenario, an application provider creates NeRF(s) of an event, such as a music concert or a sports match. To train the NeRF(s), the service provider uses training engine(s) in the network. The training data, for example, images or videos with associated metadata such as location and pose of the UE, is received from UEs present in the event (UE1 , UE2 ,….UEN ). The training data is preprocessed and used for training the NeRF models. Multiple NeRFs, targeting different UE or Server hardwares, inference capabilities and platforms, may be trained.



4.2.4.4.2 Content Synthesis 

Figure 4.2.4.4-2 NeRF and synthesized view distribution

In the UE inference scenario of clause 4.2.4.2, a UE with on-device inference capabilities (UEC) requests NeRF model appropriate for its inference capablities from the network, downloads the model and runs the inference engine to synthesize novel views locally on the device.

In the network inference scenario of clause 4.2.4.3, a resource-constrained UE (UED) may request a particular view of the event from the network, the network inference engine renders the view and it is delivered to the UE.

|  |
| --- |
| End of change |