(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

CORRECTED VERSION

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
14 March 2024 (14.03.2024)

WIPO | PCT

(10) International Publication Number
WO 2024/054804 A9

(51) International Patent Classification:
H04N 19/30 (2014.01) *H04N 19/597* (2014.01)
H04N 19/46 (2014.01)

(21) International Application Number:
PCT/US2023/073486

(22) International Filing Date:
05 September 2023 (05.09.2023)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
63/404,885 08 September 2022 (08.09.2022) US

(71) Applicant: **DOLBY LABORATORIES LICENSING CORPORATION** [US/US]; 1275 Market Street, San Francisco, California 94103 (US).

(72) Inventors: **SU, Guan-Ming**; c/o Dolby Laboratories, Inc., 1275 Market Street, San Francisco, California 94103 (US). **YIN, Peng**; c/o Dolby Laboratories, Inc., 1275 Market Street, San Francisco, California 94103 (US). **CHOUDHURY, Anustup Kumar Atanu**; c/o Dolby Laboratories, Inc., 1275 Market Street, San Francisco, California 94103 (US). **LU, Taoran**; c/o Dolby Laboratories, Inc., 1275 Market Street, San Francisco, California 94103 (US).

(74) Agent: **KONSTANTINIDES, Konstantinos** et al.; DOLBY LABORATORIES, INC., Intellectual Property Group, 1275 Market Street, San Francisco, California 94103 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO,

(54) Title: SCALABLE 3D SCENE REPRESENTATION USING NEURAL FIELD MODELING

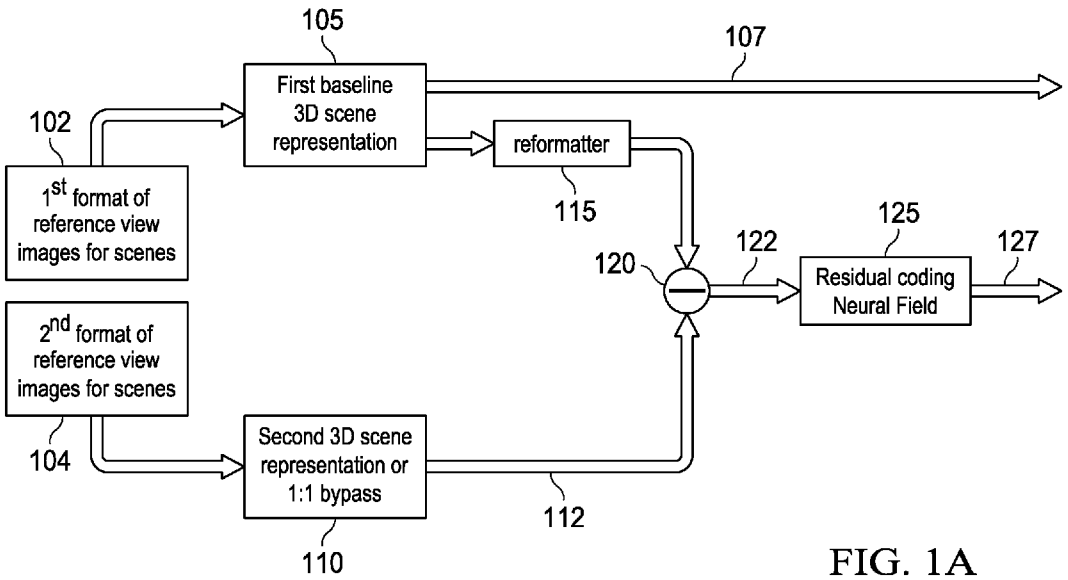

FIG. 1A

(57) **Abstract:** Methods, systems, and bitstream syntax are described for a scalable 3D scene representation. A general framework presents a dual-layer architecture where a base layer provides a baseline scene representation, and an enhancement layer provides enhancement information under a variety of scalability criteria. The enhancement information is coded using a trained neural field. Example systems are provided using a PSNR criterion and a baseline multi-plane image (MPI) representation. Examples of bitstream syntax for metadata information are also provided.

WO 2024/054804 A9

[Continued on next page]

RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

(48) Date of publication of this corrected version:
18 April 2024 (18.04.2024)

(15) Information about Correction:
see Notice of 18 April 2024 (18.04.2024)

SCALABLE 3D SCENE REPRESENTATION USING NEURAL FIELD MODELING

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of priority from U.S. Provisional Application Serial No. 63/404,885 filed on 8 September 2022, which is incorporated by reference herein in its entirety.

TECHNOLOGY

[0002] The present document relates generally to images. More particularly, an embodiment of the present invention relates to a scalable 3D scene representation using a dual layer approach where information of an enhancement layer is modeled using a neural field.

BACKGROUND

[0003] In recent years there has been an increased interest for the efficient modeling and representation of 3D scenes. 3D scenes may be used in a variety of applications, including volumetric imaging, virtual reality, or augmented reality. Deep learning techniques have shown promising results in 3D scene representation and reconstruction; however, not all devices can handle the computation load associated with such approaches. As appreciated by the inventors here, it is desirable to provide scalable 3D scene representation under a variety of scalability criteria, thus improved techniques for 3D scene representation are described herein.

[0004] The term "metadata" herein relates to any auxiliary information transmitted as part of a coded bitstream and assists a decoder to render a decoded image or a 3D scene. Such metadata may include, but are not limited to, color space or gamut information, reference display parameters, camera parameters, neural network parameters, and the like.

[0005] The approaches described in this section are approaches that could be pursued, but not necessarily approaches that have been previously conceived or pursued. Therefore, unless otherwise indicated, it should not be assumed that any of the approaches described in this section qualify as prior art merely by virtue of their inclusion in this section. Similarly, issues identified with respect to one or more approaches should not assume to have been recognized in any prior art on the basis of this section, unless otherwise indicated.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] An embodiment of the present invention is illustrated by way of example, and not in way by limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:

[0007] FIG. 1A depicts an example of an encoder for a scalable 3D scene representation under a general scalability framework according to an embodiment of this invention;

[0008] FIG. 1B depicts an example of decoder for a scalable 3D scene representation under a general scalability framework according to an embodiment of this invention;

[0009] FIG. 1C depicts an example of an encoder for a scalable 3D scene representation under a PSNR criterion according to an embodiment of this invention;

[00010] FIG. 1D depicts an example of decoder for a scalable 3D scene representation under a PSNR criterion according to an embodiment of this invention;

[00011] FIG. 2A depicts an example of an encoder for a scalable 3D scene representation under a PSNR criterion and a multi-plane image (MPI) representation according to an embodiment of this invention; and

[00012] FIG. 2B depicts an example of decoder for a scalable 3D scene representation under a PSNR criterion and an MPI representation according to an embodiment of this invention.

DESCRIPTION OF EXAMPLE EMBODIMENTS

[00013] Example embodiments that relate to a scalable 3D-scene representation are described herein. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the various embodiments of present invention. It will be apparent, however, that the various embodiments of the present invention may be practiced without these specific details. In other instances, well-known structures and devices are not described in exhaustive detail, in order to avoid unnecessarily occluding, obscuring, or obfuscating embodiments of the present invention.

SUMMARY

[00014] Example embodiments described herein relate to scalable 3D-scene representation. In an embodiment, in an encoder, to generate a scalable 3D scene representation, a processor:

accesses a first set of images in a first format (102) for a scene;

generates a first 3D scene representation (107) for the scene based on the first set of images;

accesses a second set of images in a second format (104) for the scene;

generates a second 3D scene representation (112) for the scene based on the second set of images, wherein the second 3D representation is better than the first 3D scene representation according to one or more quality criteria;

using a set of original viewing positions and a set of novel viewing positions, generates output image residuals (122) based on the first 3D scene representation and the second 3D scene representation;

trains a residual neural field network (125) using the output image residuals to generate predicted residual images approximating the output image residuals;

transmits the first 3D scene representation (107) for the scene as a base layer; and transmits information about the trained residual neural field network as an enhancement layer.

[00015] In an embodiment, in a decoder, to generate an output 3D scene, a processor:

receives a base layer bitstream (107) comprising a first 3D scene representation (107) for a scene;

receives an enhancement layer bitstream (127) comprising information to reconstruct a trained residual neural field network;

given a viewer position:

generates a first 3D output (132) of a scene based on the first 3D scene representation;

generates image residuals (145) using the viewer position and the trained residual neural field network; and

combines the first 3D output of the scene and the image residuals to generate an enhanced 3D output of the scene.

3D SCENE REPRESENTATIONS AND NEURAL FIELDS

Introduction

[00016] There are multiple 3D scene representation models, including the multi-view plus depth (MVD) method (Ref. [6]), multi-plane imaging (MPI) (Ref. [5]), and neural radiance field (NeRF) (Ref. [2]) representation. Among all of those methods, there are three major evaluation criteria: (1) their computation complexity at training and testing time, (2) bit size (bandwidth requirement) of scene representation and model size, and (3) 3D scene reconstruction quality. In practice, there are multiple end-devices, and applications need to address the computation capability and required 3D reconstructed quality of the end-application while preserving communication bandwidth. Some devices can only afford a low computation-load, but their users can accept lower quality. For high-end devices, adding more computation to achieve better quality is feasible. To cover a wide spectrum of needs and requirements, embodiments herein propose a dual-layer system with a base layer (BL) to satisfy a baseline set of requirements and an enhancement layer (EL) to enhance user experience. The proposed framework can also incorporate a variety of scalability criteria based on peak signal to noise ratio (PSNR), dynamic range, color gamut, spatial resolution, temporal frame rate, and the like.

[00017] As an example, for the base layer one may adopt the MPI representation, due to its ultra-low decoding computation. Such a base layer would ensure a broad deployment of the encoded bit stream to multiple devices, and it would maintain a baseline quality. However, MPI lacks the ability to provide lots of specular highlights (non-Lambertian; for example, transparent materials belong to the non-Lambertian family). To provide those specular highlights, one can encode the difference between a 3D scene with specular highlights and MPI in the enhancement layer using neural field coding. The base layer can be coded (compressed) using conventional codec techniques, such as AVC, HEVC, VVC, AV1, and the like, while the enhancement layer can carry neural-network coefficients representing the neural field. Once a device has more computation power, one can decode the enhancement and add it on top of the base layer to provide better rendering quality.

[00018] Some other benefits compared to using single-layer solutions neural network, such as NeRF, to code a 3D scene directly include the following. Neural network solutions, such as NeRF, or generally speaking, an MLP (a Multilayer Perceptron), require scene specific training which can be an issue for some application. In contrast, MPI can use a pretrained

network. If MLP is only used for residue, the neural network (NN) can be greatly simplified, and training time should be dramatically reduced.

[00019] For an MLP, such as NeRF, the model size is about 5 Mbytes per image scene. A straightforward transmission of such a model for a video sequence can be a big burden to the network. Furthermore, the compressibility for such a NN representation is still under investigation. If MLP is used for the residue layer, the transmission bitrate can be dramatically reduced.

[00020] In certain embodiments, the enhancement layer is out of the coding loop. Thus, one can offer a quality enhancement by simply adding NN residual information to the scene rendered using just the base layer. In addition, the out of the coding loop operation does not require a bit-exact process. The platform can select either floating point, or fixed point operations to fit its computational environment.

[00021] In an embodiment, without limitation, the NN coefficients can be carried within the bitstream or downloaded from external means, for example, using syntax defined in Ref.[13] (see also Ref. [4]).

[00022] Scalability allows one to apply for a variety of diverse quality criteria to generate the enhancement layer, including:

- PSNR: the 3D rendering quality can be improved by adding enhancement layer residuals on top of a lower quality base layer;
- Dynamic range: one can enhance the dynamic range of the rendered image by adding an enhancement layer residual on top of a standard dynamic range (SDR) 3D scene to generate a high dynamic range (HDR) 3D scene;
- Color gamut: one can enhance the color gamut by adding an enhancement layer residual on top of a narrower SDR color gamut 3D scene to generate a wider color gamut 3D scene;
- Spatial resolution: using an upscaled version of the base layer, one can add the enhancement layer information to enhance the details of a final scene at a higher resolution than the base layer resolution;
- Temporal frame rate: one can apply a frame rate interpolation on the base layer, then add the neural-field residual to generate an output at a higher frame rate;
- Any combinations of the above scalability criteria

Neural Fields

[00023] The term 'neural fields' denotes coordinate-based, fully-connected neural networks (Ref. [1]). A neural network connects many layers of artificial neurons to learn to non-linearly map a fixed-size input to a fixed-size output. A multi-layer perceptron (MLP) neural network can approximate any function through their learned parameters. Thus, a neural field can be built from a multi-layer perceptron (MLP). In the rest of this discussion, the terms MLP and neural field will be used interchangeably.

[00024] An end-to-end MLP network consists of K layers of *weights* $\{\boldsymbol{W}_k\}$ and *bias* $\{\boldsymbol{b}_k\}$ parameters. Denote those parameters as $\Phi = \{\{\boldsymbol{W}_k\}, \{\boldsymbol{b}_k\}\}$. This MLP network takes input $\boldsymbol{x}$ and output $\hat{\boldsymbol{y}}$, where

$$\hat{\boldsymbol{y}} = MLP_{\Phi}(\boldsymbol{x}). \tag{1}$$

Having a ground truth signal $\boldsymbol{y}^{gt}$, the formal problem formulation to optimize the parameter set Φ is given by:

$$\Phi^* = arg\min_{\Phi} D(\hat{\boldsymbol{y}}, \boldsymbol{y}^{gt}), \tag{2}$$

where $D()$ denotes a loss/error function.

[00025] In some 3D scene representations, such as NeRf (see Ref. [2]), the input $\boldsymbol{x}$ consists of spatial locations (x, y, z) and the viewing direction (θ, ϕ), and outputs the volume density (σ) and view dependent emitted radiance (r, g, b) at those coordinates.

Positional encoding

[00026] Neural fields suffer from a loss of frequency details. To address this issue, applying positional encoding is a common solution. In positional encoding, the network inputs are mapped to a higher dimensional space. This is because neural networks are more biased towards learning lower frequency functions. Thus, a typical neural network is not able to represent high frequency variations in color and geometry. For neural scene representation, the performance of a neural network is significantly improved by mapping the position coordinates p from R to R^{2L} where L is the number of frequencies. A typical mapping γ acting on a coordinate p can be represented as:

$$\gamma(p) =$$
$$[\sin(2^{l_0}\pi p)\ \cos(2^{l_0}\pi p)\ \sin(2^{l_1}\pi p)\ \cos(2^{l_1}\pi p)\ \ldots\ \sin(2^{l_{L-1}}\pi p)\ \cos(2^{l_{L-1}}\pi p)\ ], \quad (3)$$

where $\{l_0, l_1, \ldots, l_{L-1}\}$ are integers. In a typical setting, $l_k = k$.

[00027] Alternatively, one may apply parametric encoding, that is, arrange additional trainable parameters (beyond weights and biases) in an auxiliary data structure: such as grid, or a tree, and to look-up and (optionally) interpolate these parameters depending on the input vector.

Additional solutions to help alleviating the high frequency modelling, include a using periodic function as the activation function (see SIREN in Ref. [7]).

Forward mapping

[00028] In some applications, the output from an MLP is not the direct required result and needs another mapping. For example, in NeRF, the output from MLP is (σ, r, g, b) at the coordinate query point (x, y, z, θ, ϕ). To construct a projected 2D image, a volume rendering is needed by querying all particles along each ray and computing the final rendered RGB value.

In the proposed embodiments, the output from the neural residual network is already the rendered RGB residual. The RGB residual can be directly added on top of the rendered novel view from the base layer. Next, different architecture designs will be discussed.

A general framework for scalable 3D representation

[00029] Consider a set of images, $\{\boldsymbol{I}^g_{(t^g)}\}$, capturing the same scene from several different viewing positions, denoted as $\{t^g\}$. In an embodiment, the collected image set can be used to construct a first 3D scene representation algorithm $R^b_{\Phi_b}$ (with parameter Φ_b) to be used as base layer. Given a query viewing position, one then can render an image of the scene at the original viewing positions $\{t^g\}$ and novel viewing positions $\{t^n\}$. Denote the rendered image at $\{t^g\}$ as $\{\hat{\boldsymbol{I}}^b_{(t^g)}\}$, and at $\{t^n\}$ as $\{\hat{\boldsymbol{I}}^b_{(t^n)}\}$. The base layer should provide the minimal (base level) quality of the 3D representation, suited for a typical decoding environment.

[00030] Next, one can use a second 3D scene representation algorithm that can offer an increased level of quality over the base level. As discussed before, and will be discussed in more details later, such increased level of quality may include improved PSNR, higher bit depth, wider color gamut, and the like. Depending on the scalability criterion, one may apply the same training dataset or a different training data set to get a model $R^s_{\Phi_s}$ (with parameters Φ_s). As before, one can render the image at the original viewing positions $\{t^g\}$ and novel viewing positions $\{t^n\}$ using $R^s_{\Phi_s}$. Denote the rendered image at $\{t^g\}$ as $\{\hat{\boldsymbol{I}}^s_{(t^g)}\}$, and at $\{t^n\}$ as $\{\hat{\boldsymbol{I}}^s_{(t^n)}\}$.

[00031] In an embodiment, the residual image can be generated by taking the rendering difference from the first base 3D scene representation $R^b_{\Phi_b}$ and the second 3D scene representation $R^s_{\Phi_s}$. At original viewing positions $\{t^g\}$ and novel view positions $\{t^n\}$, one has

$$\hat{\boldsymbol{I}}^e_{(t^g)} = \hat{\boldsymbol{I}}^s_{(t^g)} - \hat{\boldsymbol{I}}^b_{(t^g)}, \qquad (4)$$
$$\hat{\boldsymbol{I}}^e_{(t^n)} = \hat{\boldsymbol{I}}^s_{(t^n)} - \hat{\boldsymbol{I}}^b_{(t^n)}.$$

[00032] In an embodiment, both sets of residual images, $\{\hat{\boldsymbol{I}}^e_{(t^g)}\}$ and $\{\hat{\boldsymbol{I}}^e_{(t^n)}\}$ are used to train a third neural residual network MLP $R^r_{\Phi_r}$ (with parameter Φ_r). Note that in this case, the MLP takes an image coordinate (x, y) with positional encoding and viewing position t as input; and outputs RGB values for pixel locations (x, y) as $\hat{\boldsymbol{I}}^r_{(t)}(x, y)$.

$$\hat{\boldsymbol{I}}^r_{(t)}(x, y) = MLP_{\Phi_r}(\gamma(x), \gamma(y), t)\,, \qquad (5)$$

where $\gamma()$, as discussed earlier, denotes a positional encoding function.

[00033] Unlike NeRF, which needs volume rendering, the neural residual does not need forward mapping to obtain the rendered 2D image. The output from the MLP is already in the RGB domain. The main goal of the neural residual network is to take any viewing position $\{t\}$ and output the predicted residual image $\{\hat{\boldsymbol{I}}^r_{(t)}\}$. The optimization process can be formulated as follows:

$$\Phi_r{}^* = arg \min_{\Phi_r} D(\hat{\boldsymbol{I}}^r_{(t)}, \hat{\boldsymbol{I}}^e_{(t)}). \qquad (6)$$

In an embodiment, the base model parameter set, Φ_b, and the residual model parameter set, Φ_r, can be separately compressed by MPEG NNC (Ref. [4]). Other embodiments may use 3D

representations that don't involve neural networks. For example, the base model parameter set, Φ_b, may represent multiview texture (MVC), multiview texture plus depth (MVC+D or MVD), or an MPI format. Those formats can be used to render a 3D scene and can be compressed using existing single-layer or multi-layer codecs, like AVC, HEVC, VVC, MIV (MPEG Immersive Video), and the like.

[00034] FIG. 1A depicts an example processing pipeline for encoding a scalable 3D representation using a generic framework that supports a variety of scalability criteria, such as:

a. PSNR scalability
b. Dynamic range scalability
c. Color gamut scalability
d. Spatial resolution scalability
e. Temporal frame rate scalability

[00035] As depicted in FIG. 1A, the base layer comprises a first unit (105) to generate a first baseline 3D scene representation (107). Input to this unit is a first set of reference input images (102) for a scene, in a first format. This 3D scene representation may be further compressed using either traditional image and video coding tools or alternative NN-representation coding tools (not shown).

[00036] To generate the enhancement layer (127), a second set of reference input images (104) for the same scene, but in a second format, is fed to a second unit (110) which will generate a second 3D representation (112). For example, depending on the scalability criterion and without loss of generality, the two sets of reference images (102, 104) may represent:

a. PSNR scalability: the first set of images is the same as the second set of images;
b. Dynamic range scalability: the first set of images are in SDR, and the second set of images are in HDR;
c. Color gamut scalability: the first set of images are in R.709, and the second set of images are in R.2020;
d. Spatial resolution scalability: the first set of images are in 1080p, and the second set of images are in 2160p;
e. Temporal frame rate scalability: the first set of images are in 24 fps, and the second set of images are 48fps;

In unit 110, for the second 3D scene representation, a 1:1 bypass might be used if the rendered scene is in an original camera position where ground-truth images are available. In other words, using the 1:1 bypass, one can take advantage of having the ground truth image at pose $\{t^g\}$ by directly using the ground truth image to generate the residual, instead of using the second model (110), whose output might still contain artifacts/distortion.

[00037] As depicted in FIG. 1A, in some embodiments, a reformatter (115) may be needed when there is spatial and/or temporal misalignment between the base layer and the enhancement layer outputs (107 and 112) (e.g., in cases d) and e) discussed above). For spatial resolution scalability, the reformatter may perform spatial up-scaling or down-scaling. For temporal frame rate scalability, the reformatter may drop frames or perform inter-frame interpolation. This reformatter is used in both encoder and decoder (see FIG. 1B). In some embodiments, the reformatter may be employed in the enhancement layer, after the second/enhancement layer representation unit (110).

[00038] Given the two scene representations, a residual (122) is generated by residual generator (120), representing their difference. All residuals from different views are encoded by neural field (125). The neural-network representation of residual neural field (125) is compressed and transmitted as neural network residual bitstream output (127).

[00039] At the decoder side, as depicted in FIG. 1B, a decoder receives bitstreams (107) and (127) representing the baseline and enhancement information. Note that if bitstream (107) was compressed prior to transmission, it should also be suitably decompressed in the decoder (not shown). Some decoders may simply use only the baseline information and ignore any enhancement information. As depicted in FIG. 1B, given a user's specified viewing position to render a scene, a base layer unit (130) will reconstruct a rendered baseline view (132) . Depending on the scalability criterion, as discussed earlier, if the decoder will use residual information, then the baseline view (132) may need to be processed by the reformatter (115). The enhancement layer bitstream (127) will be decoded along with the user's viewer position input to render the residual (145) generated using neural field (140). The output from the reformatter will be added to the residual to generate the refined novel view (150).

[00040] In an embodiment, one may desire to reduce the computational complexity of generating the neural field (125) and/or reducing the neural-field model size, for example, by training neural field 125 using input residuals (122) of lower spatial resolution. This step of reducing the spatial resolution of the residuals can be a separate processing unit (not shown)

positioned after the residual generator (120) and before the residual neural field (125), or it can be absorbed by the structure of the residual field (125). In the decoder, one can add a spatial-upscaling unit after neural field 140. Alternatively, since a neural field is a continuous function block, during inferencing, one can query higher resolution outputs even if the neural residual network is trained using lower resolution grid data. Thus, the residual decoding neural field (140) can absorb the spatial interpolation operation and there is no need for a separate spatial/temporal interpolation module.

Scalability using a PSNR criterion

[00041] FIG. 1C and FIG 1D depict a simplified version of FIG. 1A and FIG. 1B when the scalability criterion is PSNR. As depicted in FIGs 1C and 1D, the reformatter (115) is removed and the encoder is trained based on a single set of reference views and scenes (108).

[00042] In FIG. 1D, given a novel viewing position, t, the base 3D scene representation $R_{\Phi_b}^b$ will output the base image (132) as $\hat{\boldsymbol{I}}_{(t)}^b$, and the neural residual model (140) will output the predicted residual (145) as $\hat{\boldsymbol{I}}_{(t)}^r$. The final refined rendered image (150) will be the combination of the two images:

$$\hat{\boldsymbol{I}}_{(t)}^f = \hat{\boldsymbol{I}}_{(t)}^b + \hat{\boldsymbol{I}}_{(t)}^r. \tag{7}$$

[00043] In an embodiment, the baseline representation may be based on the multi-view and depth (MVD) format. In such a scenario, the original view images are the input encoded images, and the novel view images are the Depth Image Based Rendering (DIBR) generated images (Ref. [6]).

[00044] In another embodiment, the baseline representation may be based on the MPI representation (Ref. [5]). It is noted that the term "MPI" is typically used to handle face-forward scenes. When dealing with 360 degree video, the term MSI (Multi-sphere Imaging) is used (Ref. [8]). But the concept is the same. Next, as an example, and without limitation, additional details will be provided for the MPI representation, assuming a face-forward scene, where all camera poses are in the same plane. As an example, only image scenes will be considered, but a similar concept can be applied to video scenes as well.

[00045] FIG. 2A depicts an example embodiment of an encoder for a scalable 3D scene representation under a PSNR criterion and an MPI representation. Given reference multi-camera captured images (202), the base layer (207) contains MPI bitstreams. To reduce the burden at the decoder, for each camera position (a camera pose or view) $t \in \{t^g\}$, in unit

(205), a pretrained NN may be used to convert an image ($\boldsymbol{I}^{g}_{(t^g)}$) into D-layer MPI format ($\boldsymbol{C}_{\boldsymbol{i}}^{(s)}, \boldsymbol{A}_{\boldsymbol{i}}^{(s)}$) for $i = 0, \ldots, D-1$, where $\boldsymbol{C}_{\boldsymbol{i}}^{(s)}$is the i-th texture layer and $\boldsymbol{A}_{\boldsymbol{i}}^{(s)}$ is the i-th transparent layer. The pre-processing steps to convert MPIs from multi-camera poses to fit into conventional codecs such as AVC, HEVC, and VVC, and the like are not shown in this diagram. For each novel view position $t \in \{t^n\}$, a pre-defined number (typically it's 4 or 8) of nearest original camera views are selected for interpolation and the set is denoted as T_t. Each selected original view, s, will be warped to novel view position, t, as a new MPI representation: ($\boldsymbol{C}_{\boldsymbol{i}}^{(s\to t)}, \boldsymbol{A}_{\boldsymbol{i}}^{(s\to t)}$) for $i = 0, \ldots, D-1$. The warping process for each layer ($\boldsymbol{C}_{\boldsymbol{i}}^{(s)}, \boldsymbol{A}_{\boldsymbol{i}}^{(s)}$), from the current viewpoint position s to new viewpoint position t may be expressed as:

$$\boldsymbol{C}_{i}^{(s\to t)} = T_{s,t}(\sigma d_i, \boldsymbol{C}_i), \qquad (8)$$
$$\boldsymbol{A}_{i}^{(s\to t)} = T_{s,t}(\sigma d_i, \boldsymbol{A}_i).$$

The warping function, $T_{s,t}()$, may be represented as

$$\begin{bmatrix} u_s \\ v_s \\ 1 \end{bmatrix} = \boldsymbol{K}_s(\boldsymbol{R} - \frac{\boldsymbol{t}\boldsymbol{n}^T}{\boldsymbol{a}})(\boldsymbol{K}_t)^{-1}\begin{bmatrix} u_t \\ v_t \\ 1 \end{bmatrix}, \qquad (9)$$

where (u_s, v_s) is the pixel coordinate at pose s and (u_t, v_t) is the pixel coordinate at pose t. $\boldsymbol{K}_s$ and $\boldsymbol{K}_t$ are the camera intrinsic camera models for reference view and target view. **R** and **t** are the extrinsic camera models for rotation and translation. **n** is the normal vector [0 0 1]$^{\mathrm{T}}$. a is the distance to a plane that is front-to-parallel to the source camera at depth σd_i.

[00046] The rendered image from s to t, $\boldsymbol{I}^{(s\to t)}$, can be computed as the warped texture and alpha channel:

$$\boldsymbol{W}_{i}^{(s\to t)} = (\boldsymbol{A}_{i}^{(s\to t)} \cdot \prod_{j=i+1}^{D-1}(1 - \boldsymbol{A}_{j}^{(s\to t)})), \qquad (10)$$
$$\boldsymbol{I}^{(s\to t)} = \sum_{i=0}^{D-1} \boldsymbol{C}_{i}^{(s\to t)} \boldsymbol{W}_{i}^{(s\to t)}.$$

One can also sum up all transparent layers, which will be used in the fusion process.

$$\boldsymbol{A}^{(s\to t)} = \sum_{i=0}^{D-1} \boldsymbol{A}_{\boldsymbol{i}}^{(s\to t)}. \qquad (11)$$

The novel view is the weighted combination from the warped selected neighbors T_t is given by (Ref. [9]):

$$\hat{\boldsymbol{I}}^{b}_{(t)} = \frac{\sum_{s\in T_t} \alpha^{(s\to t)} \boldsymbol{A}^{(s\to t)} \boldsymbol{I}^{(s\to t)}}{\sum_{s\in T_t} \alpha^{(s\to t)} \boldsymbol{A}^{(s\to t)}}. \quad (12)$$

The weighting factor is expressed as

$$\alpha^{(s\to t)} = e^{-\frac{f}{Dd_N}\|p_s - p_t\|_2}, \quad (13)$$

where f here represents the camera focal length and p_s and p_t represent the poses of camera s and the novel view t.

[00047] In an embodiment, the enhancement layer (227) contains a neural network coding bitstream which comprises NN MLP model parameters (e.g., for model 225). The input to the NN MLP (225) is (x, y, m, n), where (x, y) is pixel location of an image and (m, n) denotes the pose coordinates. The output of NN MLP is RGB value for any given (x, y, m, n). At the encoder side, one needs to train the MLP per NN residue scene. The training residue images (220) may be generated (e.g., in unit 215) using reference views and novels views as follows:

- First, apply a second 3D scene representation algorithm (e.g., NeRF 210) using the same dataset to get the model $R^{s}_{\Phi_s}$ (with parameter Φ_s). One can render the image at the original viewing positions $\{t^g\}$ and novel viewing positions $\{t^n\}$ using $R^{s}_{\Phi_s}$. Denote the rendered image (212) at $\{t^g\}$ as $\{\hat{\boldsymbol{I}}^{s}_{(t^g)}\}$, and at $\{t^n\}$ as $\{\hat{\boldsymbol{I}}^{s}_{(t^n)}\}$.
- for original camera pose t^g, the residue image (220) ($\hat{\boldsymbol{I}}^{e}_{(t^g)}$) = original camera captured image $\boldsymbol{I}^{g}_{(t^g)}$ – rendered image by camera MPIs ($\hat{\boldsymbol{I}}^{b}_{(t^g)}$), i.e., $\hat{\boldsymbol{I}}^{e}_{(t^g)} = \boldsymbol{I}^{g}_{(t^g)} - \hat{\boldsymbol{I}}^{b}_{(t^g)}$
- for novel pose t^n, the residue image (220) ($\hat{\boldsymbol{I}}^{e}_{(t^n)}$)= rendered image by NeRF model (210) ($\hat{\boldsymbol{I}}^{s}_{(t^n)}$) – rendered image by multi-camera MPIs ($\hat{\boldsymbol{I}}^{b}_{(t^n)}$), i.e., $\hat{\boldsymbol{I}}^{e}_{(t^n)} = \hat{\boldsymbol{I}}^{s}_{(t^n)} - \hat{\boldsymbol{I}}^{b}_{(t^n)}$

The residue images (220) $\{\hat{\boldsymbol{I}}^{e}_{(t)}\}$ are then used for NN 225 to train MLP function $R^{r}_{\Phi_r}$ (with parameter Φ_r):

$$\hat{\boldsymbol{I}}^{r}_{(t)}(x, y) = MLP_{\Phi_r}(x, y, m, n). \quad (14)$$

The network parameters can be found via the optimization

$$\Phi_r{}^{*} = arg \min_{\Phi_r} D(\hat{\boldsymbol{I}}^{r}_{(t)}, \hat{\boldsymbol{I}}^{e}_{(t)}). \quad (15)$$

[00048] This residue images generation does not take compression artifact into consideration. If compression is taken into consideration, one more parameter can be added into the NN MLP input function, for example, (x, y, m, n, Qp), where, for example, Qp is the average quantization parameter used for coding the base layer. Additional parameters can be used to indicate quality levels too. When generating the training data, the MPI rendered image can be replaced by compressed MPI rendered images.

[00049] FIG. 2B depicts an example embodiment of the corresponding decoder. Given baseline input 207, for any given novel pose (m, n), the base layer NN (230) decodes the required multiple (for example, four) camera pose MPIs and renders a base layer image (232). For the enhancement layer, given input 227, the residue image (245) is generated using a trained MLP (240). Then, the base layer (232) and enhancement layer (245) are added to generate the final image (250).

[00050] An example of an MLP function is shown as follows using PyTorch code:

Table 1. Example of neural field for residue coding

```
import torch
from torch import nn


class MLP(nn.Module):

  '''
  Multilayer Perceptron for regression.
  '''
```

```
def __init__(self, in_dim, out_dim):
    super().__init__()
    self.layers = nn.Sequential(
      nn.Linear(in_dim, 256),
      nn.ReLU(),
      nn.Linear(256, 128),
      nn.ReLU(),
      nn.Linear(128,64),
      nn.ReLU(),
      nn.Linear(64, 32),
      nn.ReLU(),
      nn.Linear(32, 16),
      nn.ReLU(),
      nn.Linear(16, out_dim),
      nn.Sigmoid() # Compresses the input to range [0,1]
    )


def forward(self, x):
    '''
      Forward pass
    '''
    return self.layers(x)
```

[00051] During the training, the loss function can be defined using the normalized root mean squared error. Other loss functions can also be applied.

```
def loss_function(targets,outputs): # Normalized root mean squared error
  # Source: https://pytorch.org/docs/stable/generated/torch.linalg.norm.html
        return torch.linalg.norm(targets-outputs)/torch.linalg.norm(targets)
```

As an example, in an embodiment, for the training parameters: learning rate is 1e-3, and Adam optimization (a replacement optimization algorithm for stochastic gradient descent for training deep learning models) is used.

[00052] In an alternative embodiment, the base layer 3D representation can be a baseline NeRF with a smaller model size, and the enhancement layer can be created via a more advanced (or higher precision) NeRF.

[00053] The input $\boldsymbol{x}$ of each NeRF consists of the spatial location (x, y, z) and the viewing direction (θ, ϕ), and outputs the volume density (σ) and view dependent emitted radiance (r, g, b) at those coordinates.

[00054] For each viewing direction, one generates the rendered base image using a smaller model NeRF with parameter Φ_b:

$$\hat{\boldsymbol{I}}^{b}_{(t)}(x,y) = MLP_{\Phi_b}(x,y,z,\theta,\phi).$$

(16)

Using a smaller model NeRF with parameter Φ_s one can render a better quality image as

$$\hat{\boldsymbol{I}}^{s}_{(t)}(x,y) = MLP_{\Phi_s}(x,y,z,\theta,\phi).$$

(17)

The residual (e.g., 220) can be generated using original and novel views as follows:

- For original camera pose t^g, the residue image ($\hat{\boldsymbol{I}}^{e}_{(t^g)}$) = original camera captured image $\boldsymbol{I}^{g}_{(t^g)}$– rendered image by NeRF model $MLP_{\Phi_b}(\hat{\boldsymbol{I}}^{b}_{(t^g)})$, i.e., $\hat{\boldsymbol{I}}^{e}_{(t^g)} = \boldsymbol{I}^{g}_{(t^g)} - \hat{\boldsymbol{I}}^{b}_{(t^g)}$
- for novel pose t^n, the residue image ($\hat{\boldsymbol{I}}^{e}_{(t^n)}$)= rendered image by NeRF model MLP_{Φ_s} as $\hat{\boldsymbol{I}}^{b}_{(t^n)}$ – rendered image by NeRF model MLP_{Φ_b} as ($\hat{\boldsymbol{I}}^{s}_{(t^n)}$), i.e., $\hat{\boldsymbol{I}}^{e}_{(t^n)} = \hat{\boldsymbol{I}}^{s}_{(t^n)} - \hat{\boldsymbol{I}}^{b}_{(t^n)}$

Note: Using the 1:1 bypass, one can take the advantage of having the ground truth image at pose $\{t^g\}$ by directly using the ground truth image to generate the residual, instead of using the bigger NeRF model, which might still contain artifacts/distortion.

[00055] The neural residual network, MLP_{Φ_r}, can be trained using a method similar to the one mentioned for the MPI base layer.

[00056] In another embodiment, the base 3D representation can be generated using a scene-independent NeRF, such as PixelNeRF (Ref. [3]). The training procedure for such a scenario will be the same as the one for the scene-dependent NeRF discussed earlier.

Messaging Considerations

[00057] The proposed approach is out of the coding loop. In an embodiment, syntax related to the system parameters may be communicated using metadata, such as supplementary enhancement information (SEI) used in the MPEG video coding standards. The syntax can also be carried as part of the video program sequence (VPS), the slice

program sequence (SPS), the picture program sequence (PPS), the picture header, the slice header, and the like.

[00058] For example, an SEI message can carry information for the camera, the base layer, and the enhancement layer. The camera information should include camera parameters and camera position. For the base layer, since the base layer bitstream is the codec bitstream, one only needs to signal the extra information that the codec bitstream does not carry. Such information may include the base layer representation, such as MPI, MVD, and the like. For each input representation, some additional information may be needed. For example, for MPI, one may need to communicate the number of cameras, number of MPI layers, and the tiles-assembly. For the enhancement layer, syntax elements need to specify the NN MLP parameters. One can carry the NN parameters by external means or by using neural network represented by the ISO/IEC 15938-17 bitstream. As in NNPFC SEI (Ref. [10]), the enhancement layer information can include the input and output formatting information.

[00059] Table 2 provides an example SEI message to communicate syntax parameters related to a neural field used in a scalable 3D scene representation. To avoid duplication, only additional information is listed, that is, information not carried in the NNPFC SEI. Descriptor information (e.g., ue(v), u(n), and the like) is defined the same as in NNPFC SEI.

Table 2. Example of SEI messaging for scalable 3D scene representation

3D_nn_residue_dual_layer_info(payloadSize) {	**Descriptor**
nnr_purpose_idc /*specified for output: 5 bit mapping: 0th bit: PSNR quality; 1st bit: dynamic range mapping; 2nd bit: color gamut; 3rd bit: spatial resolution 4th bit: temporal frame rate */	ue(v)
if (nnr_purpose_idc&0x08) {	
nnr_output_pic_width_in_luma_samples	ue(v)
nnr_output_pic_height_in_luma_samples	ue(v)
}	
nnr_output_colour_description_present_flag /*dual layer output colour space where addition happens*/	u(1)
if(nnr_output_colour_description_present_flag) {	
nnr_colour_primaries	u(8)
nnr_transfer_characteristics	u(8)
nnr_matrix_coeffs	u(8)
}	
/* camera view port information: camera parameters and camera position parameters */	
viewport_camera_info_present_flag	u(1)
if (viewport_camera_info_present_flag) {	
nnr_num_cameras_minus1	ue(v)
nnr_camera_view_point_info(nnr_num_cameras_minus1)	

}	
/* nnr_base layer information*/	
nnr_bl_idc /*BL is MPI or MVD etc*/	ue(v)
if (nnr_bl_idc = = 0 || nnr_bl_idc = = 1) { /* MPI or MSI: they can use the same tiling but it affects the input of NN MLP*/	
nnr_mpi_layer_minus1	ue(v)
mpi_tiling_info(nnp_mpi_layer_minus1) /*How MPI is tiled in pseudo video sequence*/	
/*view rendering required parameters*/	
nnr_sf_value /*σ parameter for view rendering using forward mapping, in units of 0.001*/	u(32)
depth_representation_info( ) /*signal depth min/max information required for view rendering. This SEI is in VSEI, HEVC and VVC*/	
} elseif ((nnr_bl_idc = = 2) { /* MVD */	
…	
… /* MVD-specific parameters */	
}	
/* nnr_enhancement layer neural network residue information*/	
nnr_mode_idc /*carry NNC bitstream or external means*/	ue(v)
nnr_input_dimension_minus3 /*MVC case, 1D array, input dimension is 3, MPI 2D array, input dimension is 4, MSI: input dimension is 5, MVD 6DoF: imput dimension is 6*/	ue(v)
for (i = 0; i < nnr_input_dimension_minus3 + 3; i++)	
nnr_position_encoding_freq[i] /*L: number of freq used for input position coding*/	ue(v)
/*residue could be negative value: normalize data in [0 1], the syntax in units of 0.001 */	
nnr_normalized_weight	u(32)
nnr_abs_normalized_offset	u(32)
if (nnr_abs_normalized_offset != 0)	
nnr_sign_normalized_offset	u(1)
/*note: optionally, below, one can include input and output formatting information in NNPFC SEI (note included in this example)*/	
…	
/* Enhancement layer NNR bitstream specified or updated by ISO/IEC 15938-17 bitstream */	
if(nnr_mode_idc = = 1) {	
while(!byte_aligned( ))	
nnr_reserved_zero_bit	u(1)
for(i = 0; more_data_in_payload( ); i++)	
nnr_payload_byte[i]	b(8)
}	
}	

[00060] For the camera viewpoint information, depending on the setup, for a 1D setup, one can use the Multiview acquisition information SEI and Multiview view position SEI in VSEI, HEVC, and AVC. For general setup, one can use the Viewport camera parameters SEI and viewport position SEI in Visual Volumetric Video-based Coding (V3C) (Ref. 11]) and MPEG Immersive video (MIV) (Ref. [12]). An example is shown below in Table 3. It is noted that the following SEIs: multiview_acquisition_info( ), multiview_view_position( ), viewport_camera_parameters (), and viewport_position( ) do not need to be included in the camera_viewport_info() SEI message depicted in Table 3. They can also be sent outside of the camera_viewport_info() SEI message. In another embodiment, because a 6DoF (six degrees of freedom) setup includes a 1D setup, one can just always use the 6DoF setup case, but it might require more bits.

Table 3. Example camera viewport SEI message

camera_viewport_info(nnr_num_cameras_minus1) {	**Descriptor**
cv_idc /*0: 1D setup; 1: 6DoF setup*/	ue(v)
if (cv_idc == 0) { /*the following two SEIs are in VSEI, HEVC, and VVC spec*/	
multiview_acquisition_info( )	
multiview_view_position( )	
}	
else if(cv_idc == 1) { /*the following two SEIs are in ISO/IEC 23090-5 spec*/	
for (i = 0; i <= nnr_num_cameras_minus1; i++) {	
viewport_camera_parameters ()	
viewport_position( )	
}	
}	
}	

cv_idc specifies the setup of camera viewpoint information as shown in the following Table.

cv_idc	camera viewport setup
0	1D Horizontal
1	6 DoF or general setup

nnr_purpose_idc specifies the purpose of the neural network residual output. It's a 5-bit on-off signalling, where: the 0^{th} bit signals the PSNR quality enhancement; the 1^{st} bit signals the dynamic range enhancement; the 2^{nd} bit signals the colour gamut enhancement; the 3^{rd} bit signals the spatial resolution enhancement, and the 4^{th} bit signals the temporal frame rate. Each bit can be 0 or 1 and in total there are 2^5 combinations.

nnr_output_pic_width_in_luma_samples specifies the width, in units of luma samples, of the output picture referring to the SEI.

nnr_output_pic_height_in_luma_samples specifies the height, in units of luma samples, of the output picture referring to the SEI.

Note: the base layer decoded picture resolution can be different from the final output resolution using this SEI.

[00061] The following syntax elements are for the colour space where 3D neural network residue layer system can be applied in different colour space than for the coded layer video sequence (CLVS) layer. For example, the bitstream is in YCbCr colour space, where dual layer system can operate on RGB colour space.

nnr_output_colour_description_present_flag equal to 1 indicates that a distinct combination of colour primaries, transfer characteristics, and matrix coefficients for the output picture resulting from the SEI is specified in the SEI message syntax structure. nnr_output_colour_description_present_flag equal to 0 indicates that the combination of colour primaries, transfer characteristics, and matrix coefficients for the output picture resulting from the SEI is the same as indicated in VUI parameters for the CLVS.

nnr_colour_primaries has the same semantics as specified in clause 7.3 for the vui_colour_primaries syntax element, except as follows:

– nnr_colour_primaries specifies the colour primaries of the picture resulting from applying the SEI message, rather than the colour primaries used for the CLVS.

– When nnr_colour_primaries is not present in the SEI message, the value of nnr_colour_primaries is inferred to be equal to vui_colour_primaries.

nnr_transfer_characteristics has the same semantics as specified in clause 7.3 for the vui_transfer_characteristics syntax element, except as follows:

– nnr_transfer_characteristics specifies the transfer characteristics of the picture resulting from applying the SEI message, rather than the transfer characteristics used for the CLVS.

– When nnr_transfer_characteristics is not present in the SEI message, the value of nnr_transfer_characteristics is inferred to be equal to vui_transfer_characteristics.

nnr_matrix_coeffs has the same semantics as specified in clause 7.3 for the vui_matrix_coeffs syntax element, except as follows:

– nnr_matrix_coeffs specifies the matrix coefficients of the picture resulting from applying the SEI message, rather than the matrix coefficients used for the CLVS.

– When nnr_matrix_coeffs is not present in the SEI message, the value of nnr_matrix_coeffs is inferred to be equal to vui_matrix_coeffs.

– The values allowed for nnr_matrix_coeffs are not constrained by the chroma format of the decoded video pictures that is indicated by the value of ChromaFormatIdc for the semantics of the VUI parameters.

nnr_num_cameras_minus1 plus 1 specifies the number of viewport cameras.

[00062] Below are the semantics for the base layer related information.

nnr_bl_idc specifies the base layer signal input format for a 3D scene representation as follows:

nnr_bl_idc	base layer input format
0	MPI
1	MSI
2	MVC or MVD

nnr_mpi_layer_minus1 plus 1 specifies the number of MPI layers of the MPI representation.

nnr_sf_value specifies the scaling factor using for view rendering in units of 0.001.

It is noted that for depth_representation_info( ), it does not need to signal inside the proposed SEI. It can be signaled outside of the current SEI.

[00063] Below are the semantics used for the enhancement layer NNR.

nnr_mode_idc equal to 0 specifies that the neural network residue is determined by external means not specified in this Specification. nnr_mode_idc equal to 1 specifies that that the neural network residue is a neural network represented by the ISO/IEC 15938-17 bitstream contained in this SEI message. nnr_mode_idc equal to 2 specifies that the neural network residue is a neural network identified by a specified tag Uniform Resource Identifier (URI) (nnr_uri_tag[i]) and neural network information URI (nnr_uri[i]). The value of nnr_mode_idc shall be in the range of 0 to 255, inclusive. Values of nnr_mode_idc greater than 2 are reserved for future specification by ITU-T | ISO/IEC and shall not be present in bitstreams conforming to this version of this Specification. Decoders conforming to this version of this Specification shall ignore SEI messages that contain reserved values of nnr_mode_idc.

nnr_input_dimension_minus3 plus 3 specifies the input signal dimension.

base layer input format	nnr_input_dimension
MPI 1D	3
MPI 2D	4
MSI or 3DoF	5
MVD	6

nnr_position_encoding_freq [i] specifies the frequency of input position encoding for i-th dimension.

nnr_normalized_weight specifies the weight value in units of 0.001.

nnr_abs_normalized_offset specifies the absolute offset value in units of 0.001.

nnr_sign_normalized_offset specifies the sign of the offset value.
nnr_normalized_offset = ((nnr_sign_normalized_offset >=0)? 1:-1) *
nnr_abs_normalized_offset
The residue input x to the neural networks is scaled to y in [0 1]: y = (nnr_normalized_weight * x + nnr_normalized_offset) * 0.001.

[00064] For unbounded scene, it may necessary also to specify the min and max value of pose coordinators and/or depth information (for zoom-in/out effect) to avoid over-rendering.

In one example, let the reference camera poses been expressed by (x_{si}, y_{si}, z_{si}) in the camera-to-world coordinating system where *i* is the index of reference cameras. The reference camera poses form a 3D volume in the 3D space and the min and max values can be calculated by:

$x\_min = \min(x_{si});$ $x\_max = \max(x_{si});$

$y\_min = \min(y_{si});$ $y\_max = \max(y_{si});$

$z\_min = \min(z_{si});$ $z\_max = \max(z_{si});$

When a novel view (x_t, y_t, z_t) is to be rendered, the camera pose of the novel view needs to be bounded within the 3D volume by the min and max values of the reference poses. i.e.,

$x_{tb} = \min(x\_max, \max(x\_min, x_t));$

$y_{tb} = \min(y\_max, \max(y\_min, y_t));$

$z_{tb} = \min(z\_max, \max(z\_min, z_t));$

In another example, scaling factors may be further applied to the min and max values of the reference camera poses to adjust the bounded area.

References

Each one of the references listed herein is incorporated by reference in its entirety.

[1] Yiheng Xie et al., "Neural Fields in Visual Computing and Beyond," Eurographics 2022/ CGF, State-of-the-Art Report, Volume 41 (2022), No. 2, 2022.

[2] Ben Mildenhall et al., "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis," ECCV 2020, also in arXiv:2003.08934v2, 5 April 2022.

[3] Alex Yu et al., "pixelNerF: Neural Radiance Fields from One or Few Images," CVPR 2021, also in arXiv:2012.02190v3, 30 May 2021.

[4] Heiner Kirchhoffer et al, "Overview of the Neural Network Compression and Representation (NNR) Standard," IEEE Trans. on Circuits and Systems for Video Technology, Vol. 32, No. 5, May 2022, pp.3203~3216.

[5] Richard Tucker and Noah Snavely, "Single-view view synthesis with multiplane images," CVPR 2020.

[6] "Test Model 11 of 3D-HEVC and MV-HEVC," JCT3V-K1003, Geneva, CH, Feb. 2015.

[7] Vincent Sitzmann et al., "Implicit Neural Representations with Periodic Activation Functions," NeurIPS 2020, also in arXiv:2006.09661v1, 17 June, 2020.

[8] Benjamin Attal et al., "MatryODShka: Real-time 6DoF Video View Synthesis using Multi-Sphere Images," (ECCV) 2020.

[9] Ben Mildenhall et al., "Local Light Field Fusion: Practical View Synthesis with Prescriptive Sampling Guidelines", ACM Transactions on Graphics, Vol. 38, No. 4, Article 29, July 2019.

[10] Sean McCarthy et al., "Additional SEI messages for VSEI (Draft 2)", JVET-AA2006v2, JVET 27th meeting, 13-22 July 2022.

[11] ISO/IEC 23090-5, Information technology — Coded Representation of Immersive Media — Part 5: Visual Volumetric Video-based Coding (V3C) and Video-based Point Cloud Compression (V-PCC).

[12] ISO/IEC 23090-12, Information technology — Coded representation of immersive media — Part 12: MPEG Immersive video.

[13] ISO/IEC 15938-17:2022, "MPEG NNC specification: Information technology — Multimedia content description interface — Part 17: Compression of neural networks for multimedia content description and analysis."

EXAMPLE COMPUTER SYSTEM IMPLEMENTATION

[00065] Embodiments of the present invention may be implemented with a computer system, systems configured in electronic circuitry and components, an integrated circuit (IC) device such as a microcontroller, a field programmable gate array (FPGA), or another configurable or programmable logic device (PLD), a discrete time or digital signal processor (DSP), an application specific IC (ASIC), and/or apparatus that includes one or more of such systems, devices or components. The computer and/or IC may perform, control, or execute instructions relating to a scalable 3D scene representation, such as those described herein. The computer and/or IC may compute any of a variety of parameters or values that relate to a scalable 3D scene representation described herein. The image and video embodiments may be implemented in hardware, software, firmware and various combinations thereof.

[00066] Certain implementations of the invention comprise computer processors which execute software instructions which cause the processors to perform a method of the invention. For example, one or more processors in a display, an encoder, a set top box, a transcoder, or the like may implement methods related to a scalable 3D scene representation as described above by executing software instructions in a program memory accessible to the processors. Embodiments of the invention may also be provided in the form of a program product. The program product may comprise any non-transitory and tangible medium which carries a set of computer-readable signals comprising instructions which, when executed by a data processor, cause the data processor to execute a method of the invention. Program

products according to the invention may be in any of a wide variety of non-transitory and tangible forms. The program product may comprise, for example, physical media such as magnetic data storage media including floppy diskettes, hard disk drives, optical data storage media including CD ROMs, DVDs, electronic data storage media including ROMs, flash RAM, or the like. The computer-readable signals on the program product may optionally be compressed or encrypted. Where a component (e.g. a software module, processor, assembly, device, circuit, etc.) is referred to above, unless otherwise indicated, reference to that component (including a reference to a "means") should be interpreted as including as equivalents of that component any component which performs the function of the described component (e.g., that is functionally equivalent), including components which are not structurally equivalent to the disclosed structure which performs the function in the illustrated example embodiments of the invention.

EQUIVALENTS, EXTENSIONS, ALTERNATIVES AND MISCELLANEOUS

[00067] Example embodiments that relate to a scalable 3D scene representation are thus described. In the foregoing specification, embodiments of the present invention have been described with reference to numerous specific details that may vary from implementation to implementation. Thus, the sole and exclusive indicator of what is the invention, and what is intended by the applicants to be the invention, is the set of claims that issue from this application, in the specific form in which such claims issue, including any subsequent correction. Any definitions expressly set forth herein for terms contained in such claims shall govern the meaning of such terms as used in the claims. Hence, no limitation, element, property, feature, advantage or attribute that is not expressly recited in a claim should limit the scope of such claim in any way. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

CLAIMS

What is claimed is:

1. In an encoder, a method to generate a scalable 3D scene representation, the method comprising:

accessing a first set of images in a first format (102) for a scene;

generating a first 3D scene representation (107) for the scene based on the first set of images;

accessing a second set of images in a second format (104) for the scene;

generating a second 3D scene representation (112) for the scene based on the second set of images, wherein the second 3D representation is better than the first 3D scene representation according to one or more quality criteria;

using a set of original viewing positions and a set of novel viewing positions, generating output image residuals (122) based on the first 3D scene representation and the second 3D scene representation;

training a residual neural field network (125) using the output image residuals to generate predicted residual images approximating the output image residuals;

transmitting the first 3D scene representation (107) for the scene as a base layer; and

transmitting information about the trained residual neural field network as an enhancement layer.

2. The method of claim 1, further comprising reformatting outputs of the first 3D scene representation or the second 3D scene representations before generating the image residuals.

3. The method of claim 2, wherein reformatting comprises image upscaling, image downscaling, frame dropping, frame interpolation, or dynamic range/colour gamut extension.

4. The method of any one of claims 1 to 3, wherein the one or more quality criteria include PSNR scalability, dynamic range scalability, color gamut scalability, spatial resolution scalability, and temporal frame-rate scalability.

5. The method of any one of claims 1 to 4, wherein the first set of images is identical to the second set of images.

6. The method of any one of claims 1 to 4, wherein the first set of images differs from the second set of images in terms of dynamic range or bit-depth, color gamut, spatial resolution, or frame rate.

7. The method of any one of claims 1 to 6, wherein a 3D scene representation may be one of multiview plus depth (MVD) representation, a multi-plane imaging (MPI) representation, or a neural radiance field (NeRF) neural network representation.

8. The method of claim 5, wherein the first 3D scene representation comprises a first NeRF model and the second 3D scene representation comprises a second NeRF model, wherein the second NeRF model renders better quality images than the first NeRF model, and generating the output image residuals comprises:

computing first image residuals $\hat{\boldsymbol{I}}^{e}_{(t^g)} = \boldsymbol{I}^{g}_{(t^g)} - \hat{\boldsymbol{I}}^{b}_{(t^g)}$; and

computing second image residuals $\hat{\boldsymbol{I}}^{e}_{(t^n)} = \hat{\boldsymbol{I}}^{s}_{(t^n)} - \hat{\boldsymbol{I}}^{b}_{(t^n)}$,

wherein, t^g denotes an original camera pose, t^n denotes a novel camera pose,

$\hat{\boldsymbol{I}}^{b}_{(t)}(x, y) = MLP_{\Phi_b}(x, y, z, \theta, \phi)$ and

$\hat{\boldsymbol{I}}^{s}_{(t)}(x, y) = MLP_{\Phi_s}(x, y, z, \theta, \phi)$

denote images rendered based on the first NeRF model and second NeRF model respectively for spatial location (x, y, z) and viewing direction (θ, ϕ), and $\boldsymbol{I}^{g}_{(t^g)}$ denotes an image in the first set of images.

9. The method of claim 8, wherein during training, parameters of the residual neural field network are generated by optimizing

$$\Phi_r^{*} = arg \min_{\Phi_r} D(\hat{\boldsymbol{I}}^{r}_{(t)}, \hat{\boldsymbol{I}}^{e}_{(t)}),$$

wherein Φ_r^{*} denotes an optimum set of the parameters of the residual field network, $\hat{\boldsymbol{I}}^{r}_{(t)}(x, y)$ denotes an output of the trained residual field network at a view t , $\hat{\boldsymbol{I}}^{e}_{(t)}$ denotes an image residual at view t, and $D()$ denotes a loss function to be minimized during training.

10. In a decoder, a method to generate an output 3D scene, the method comprising:

receiving a base layer bitstream (107) comprising a first 3D scene representation (107) for a scene;

receiving an enhancement layer bitstream (127) comprising information to reconstruct a trained residual neural field network;

given a viewer position:

generating a first 3D output (132) of a scene based on the first 3D scene representation;

generating image residuals (145) using the viewer position and the trained residual neural field network; and

combining the first 3D output of the scene and the image residuals to generate an enhanced 3D output of the scene.

11. The method of claim 10, further comprising reformatting the first 3D output of the scene or the image residuals before combining them.

12. The method of claim 11, wherein reformatting comprises image upscaling, image downscaling, frame dropping or frame interpolation.

13. The method of claim 1, wherein information about the trained residual neural field network comprises one or more of:

a quality parameter specifying the one or more quality criteria (nnr_purpose_idc),

camera viewport information (viewport_camera_info_present_flag parameters),

first model parameters for the first 3D representation model (nnr_bl_idc),

the number of hidden layers in the residual neural field (this relates to NN topology, it can be carried in NNR bitstream or external means based on nnr_mode_idc),

the input position encoding method (nnr_position_encoding_freq [i]),

the activation function (this relates to NN topology, it can be carried in NNR bitstream or external means based on nnr_mode_idc. It can be ReLU in Table 1 or SIREN in Ref. [7], etc),

parameters related to residual rescaling (nnr_normalized_weight/offset, and nnr_sign_normalized_offset)

descriptors of input coordinate parameters (nnr_input_dimension_minus3), and

descriptors of output parameters (nnr_colour_primaries, nnr_output_pic_width_in_luma_samples, nnr_output_pic_height_in_luma_samples etc).

residual rescaling parameters (nnr_normalized_weight, nnr_abs_normalized_offset, and nnr_sign_normalized_offset).

14. The method of claim 13, wherein the information is transmitted as part of supplemental enhancement information messaging.

15. The method of claim 1, wherein training the residual neural field network (125) using the output image residuals is in a first spatial resolution, and further comprising:

training the residual neural field network with input the output image residuals at a second spatial resolution lower than the first spatial resolution, and with output the predicted residual images at the first spatial resolution.

16. A non-transitory computer-readable storage medium having stored thereon computer-executable instructions for executing with one or more processors a method in accordance with any one of the claims 1-15.

17. An apparatus comprising a processor and configured to perform any one of the methods recited in claims 1-15.

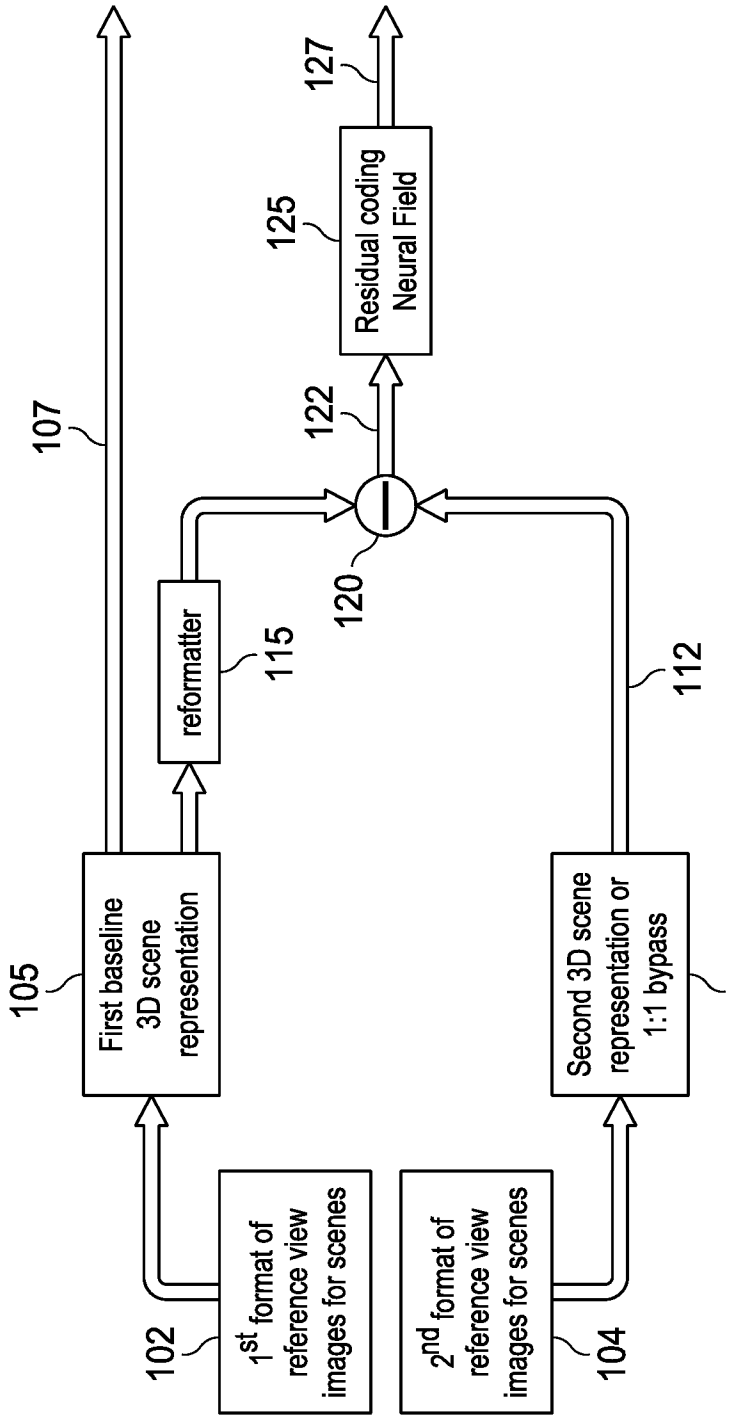
1st format of reference view images for scenes
2nd format of reference view images for scenes
First baseline 3D scene representation
Second 3D scene representation or 1:1 bypass
reformatter
Residual coding Neural Field
102
104
105
110
112
115
120
122
125
127
107

FIG. 1A

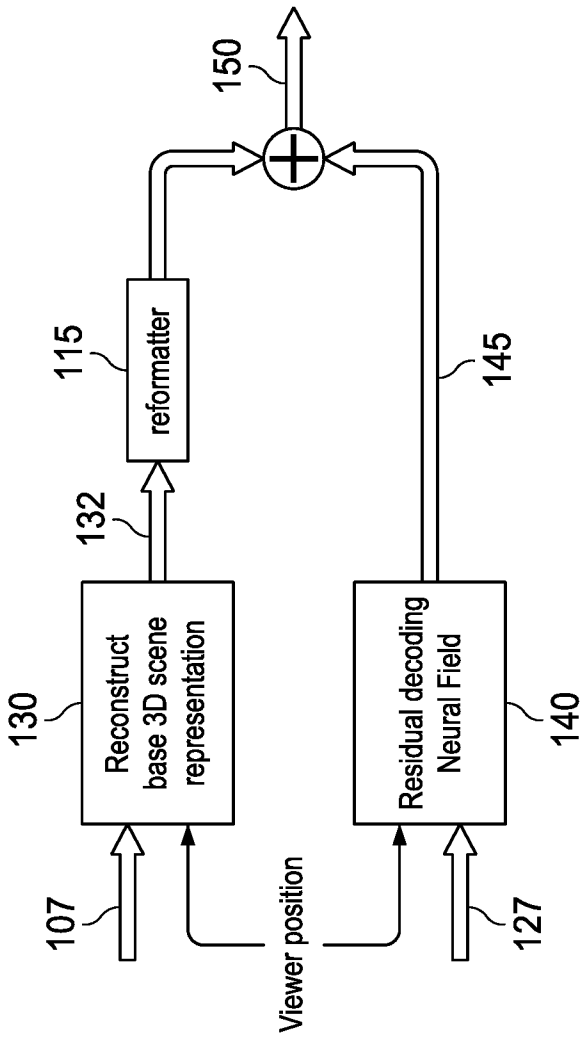
150
115
reformatter
145
132
130
Reconstruct base 3D scene representation
Residual decoding Neural Field
140
107
Viewer position
127

FIG. 1B

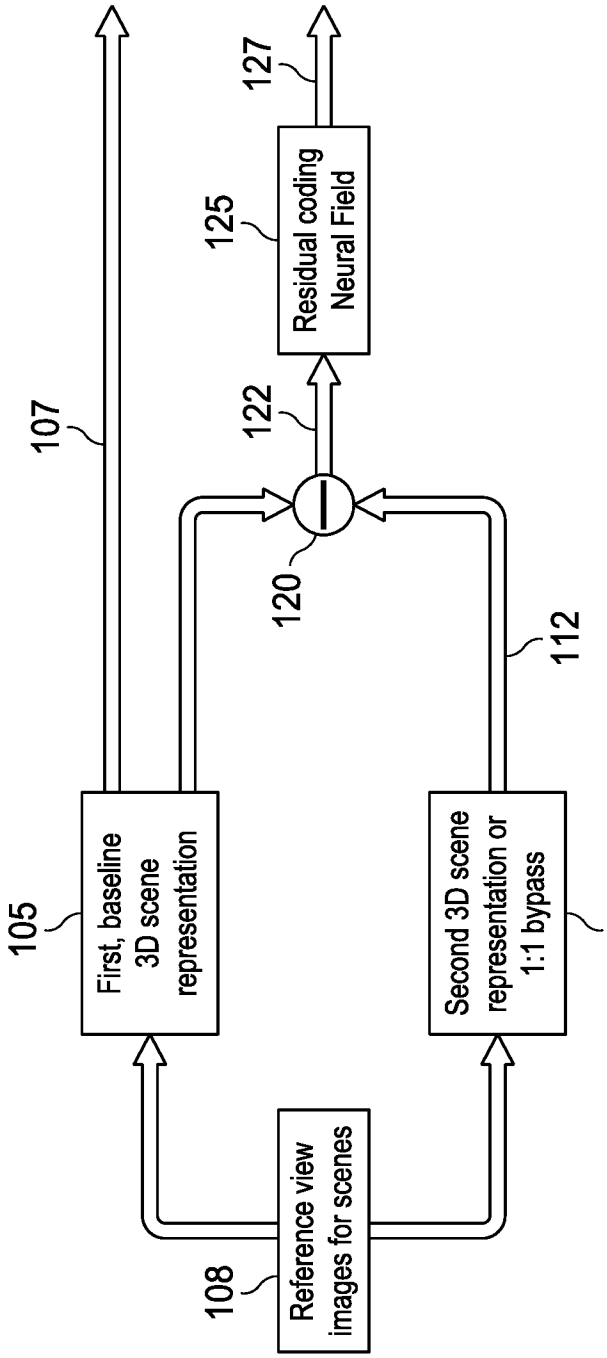
107
127
125
Residual coding Neural Field
122
120
112
105
First, baseline 3D scene representation
Second 3D scene representation or 1:1 bypass
110
108
Reference view images for scenes

FIG. 1C

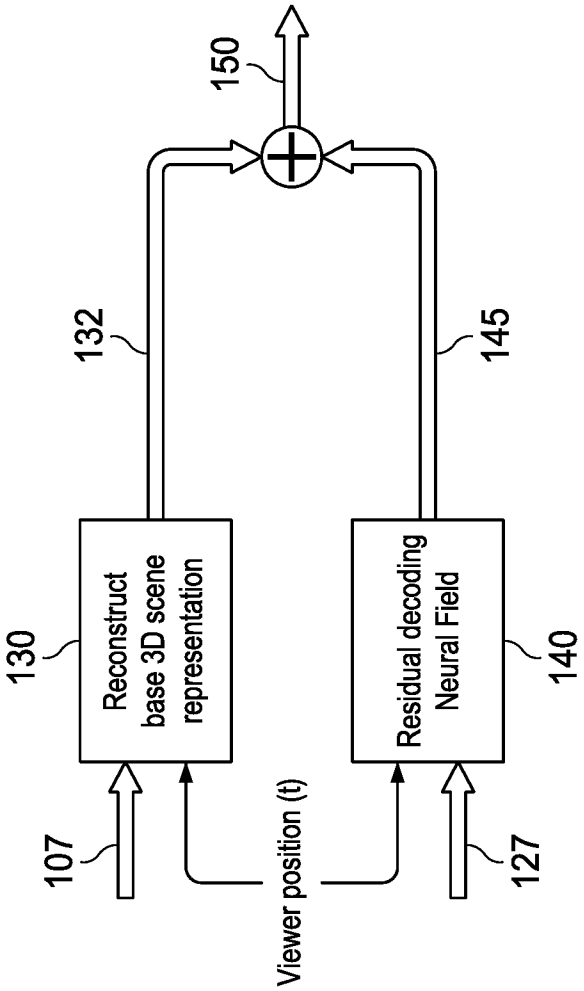
150
132
145
130
Reconstruct base 3D scene representation
Residual decoding Neural Field
140
107
Viewer position (t)
127

FIG. 1D

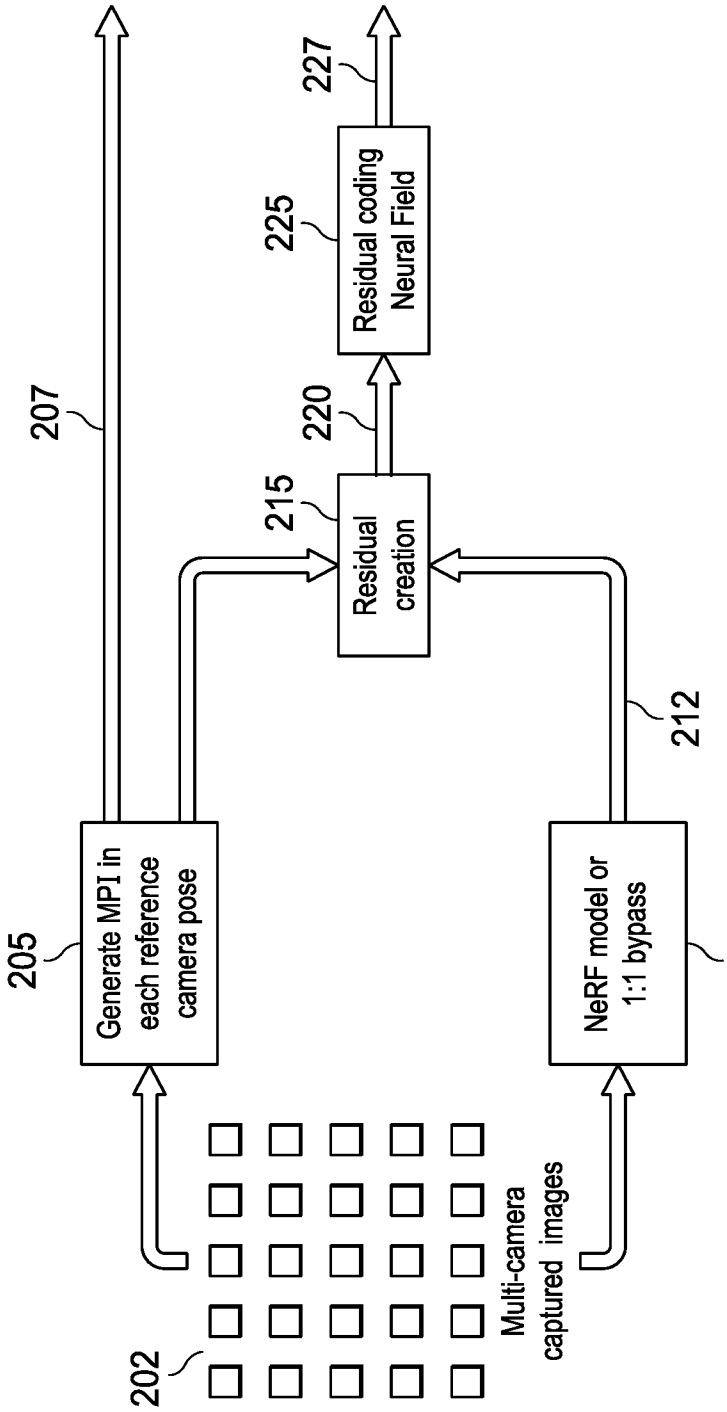
Generate MPI in each reference camera pose
205
NeRF model or 1:1 bypass
210
Residual creation
215
Residual coding Neural Field
225
227
207
220
212
Multi-camera captured images
202

FIG. 2A

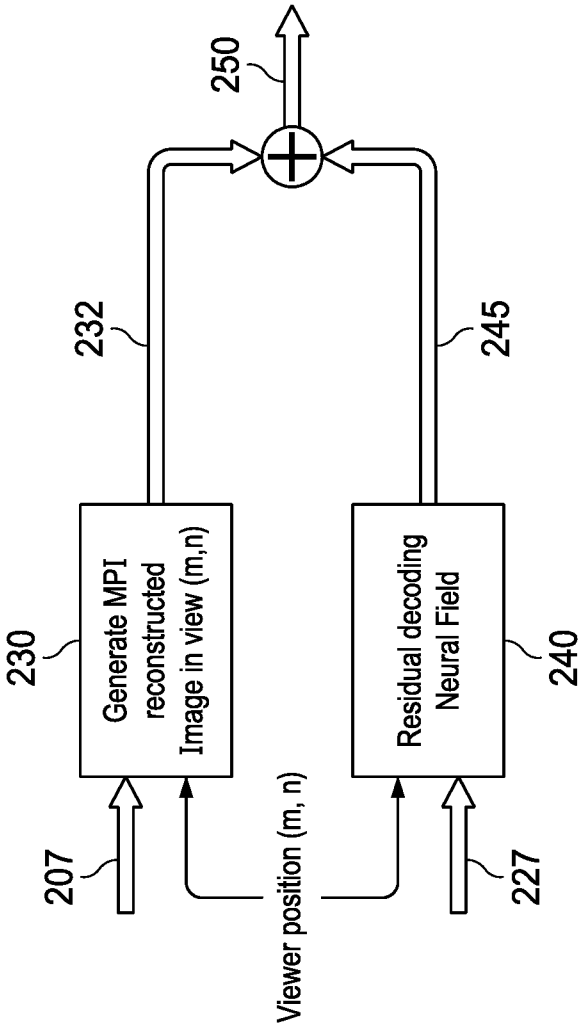
250
232
245
230
Generate MPI reconstructed Image in view (m,n)
Residual decoding Neural Field
240
207
Viewer position (m, n)
227

FIG. 2B

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2023/073486

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04N19/30 H04N19/46 H04N19/597
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	JIAHAO PANG ET AL: "[AI-3DGC][EE13.54-related] Geometric Residual Analysis and Synthesis for PCC", 137. MPEG MEETING; 20220117 - 20220121; ONLINE; (MOTION PICTURE EXPERT GROUP OR ISO/IEC JTC1/SC29/WG11), , no. m58962 12 January 2022 (2022-01-12), XP030299732, Retrieved from the Internet: URL:https://dms.mpeg.expert/doc_end_user/documents/137_OnLine/wg11/m58962-v1-m58962_GRASP.zip m58962_GRASP.docx [retrieved on 2022-01-12] sections 3.1-3.4, 4.1; figures 1-3 ----- -/--	1-17

[x] Further documents are listed in the continuation of Box C.
[ ] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance;; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance;; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
10 November 2023	20/11/2023

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Montoneri, Fabio

1

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2023/073486

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JANUS SCOTT ET AL: "Multi-Plane Image Video Compression", 2020 IEEE 22ND INTERNATIONAL WORKSHOP ON MULTIMEDIA SIGNAL PROCESSING (MMSP), 21 September 2020 (2020-09-21), pages 1-6, XP055944751, DOI: 10.1109/MMSP48831.2020.9287083 ISBN: 978-1-7281-9320-5 sections I, II	1-17
A	MILDENHALL BEN ET AL: "NeRF : representing scenes as neural radiance fields for view synthesis", PROC. OF ECCV 2020, no. 1, 3 August 2020 (2020-08-03), pages 1-25, XP055921628, Retrieved from the Internet: URL:https://arxiv.org/pdf/2003.08934.pdf> cited in the application sections 1, 3-5	1-17
A	MCCARTHY S ET AL: "Additional SEI messages for VSEI (Draft 2)", 27. JVET MEETING; 20220713 - 20220722; TELECONFERENCE; (THE JOINT VIDEO EXPLORATION TEAM OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16), , no. JVET-AA2006 ; m60614 19 August 2022 (2022-08-19), XP030304218, Retrieved from the Internet: URL:https://jvet-experts.org/doc_end_user/documents/27_Teleconference/wg11/JVET-AA2006-v2.zip JVET-AA2006-v2.docx [retrieved on 2022-08-19] sections 8.28.1, 8.28.2	1-17

1