A Deep Learning-based 6DoF Video Synthesizing Method Using Instant-NGPs

Jaeyeol Choi[1], Jong-Beom Jeong[2], JunHyeong Park[3], Eun-Seok Ryu[3]
[1]*Department of Applied Artificial Intelligence, Sungkyunkwan University (SKKU), Seoul, Republic of Korea*
[2]*Department of Computer Science Education, Sungkyunkwan University (SKKU), Seoul, Republic of Korea*
[3]*Department of Immersive Media Engineering, Sungkyunkwan University (SKKU), Seoul, Republic of Korea*
E-mail: {jaychoi, uof4949, joke0702, esryu}@skku.edu

***Abstract*—This paper introduces a new method for synthesizing six degree of freedom (6DoF) videos using neural radiance fields, which allows training from plain 2D images to render 3D scene at arbitrary viewports. Neural network model representing a previous timepoint is fine-tuned to train models for subsequent timepoints. Additionally, instant neural graphics primitives (Instant-NGP) is applied for speed improvement. The proposed method achieved both improved objective quality for same number of training iterations and enhanced consistency between frames. Furthermore, it shows superiority over other methods for generating 6DoF videos, in terms of quality and time efficiency.**

***Index Terms*—Deep learning, Virtual reality, NeRF, Instant-NGP, MIV, 6DoF, Immersive video**

I. Introduction

Neural radiance field (NeRF) [1] represents three-dimensional scene by parameters of a neural network. Once the NeRF model is trained, it can render images from arbitrary viewpoints. This capability makes it suitable for generating realistic views for head-mounted display (HMD) devices.

In videos, significant correlations exist between successive frames. Therefore, training separate NeRF models for consecutive frame reveals remarkable similarities in training sets. This suggests the potential benefits of cross-model parameter sharing, which can preserve model performance while reducing training iteration. Theoretically, initializing the weights of neural network from those learned for a similar scene can lead to faster convergence to an optimization point than initializing it randomly. Moreover, the method could mitigate spatial consistency issues arising when training each frame independently.

This paper explores the effects of parameter sharing across multiple NeRF models and presents a deep learning-based pipeline for synthesizing six degree of freedom (6DoF) videos from videos captured by multiple cameras. The conceptual diagram is illustrated in Fig. 1. The proposed method also incorporates Instant-NGP [2], which enhances time efficiency via layer reduction and parametric encoding. Additionally, the comparison between the proposed method and the state-of-art 6DoF immersive video compression standard is introduced.

This research was supported by Culture, Sports and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism in 2023 (Project Name: Development of content streaming copyright core technology on metaverse platform, Project Number: RS-2023-00223812, Contribution Rate: 100%)

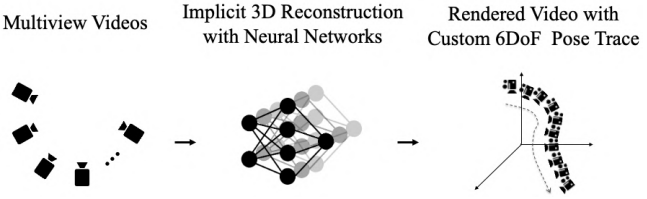


Fig. 1: The concept of encoding multiview videos by neural networks

II. Related Work

A. MPEG Video Coding Standard for 6DoF Immersive Video

6DoF videos provide high interactivity by allowing users to move physically within the scene as well as rotate their heads when using an HMD. [3] suggests a technique for capturing, generating, and rendering 6DoF videos using a single device. A prevalent strategy for 6DoF video creation is the video encoding-based compression scheme introduced by the moving picture experts group (MPEG) immersive video (MIV) coding standard [4]. This standard employs a distinct approach to eliminate redundant pixels and stitch patches in both texture and geometry videos acquired from multiple cameras. [5], [6] focus on enhancing and applying MIV for more efficient compression and transmission of 6DoF videos. One limitation of MIV is its necessity for both the texture (color) video and the corresponding depth map video of each view. This requirement poses a challenge for non-expert users due to the need for additional layers that might not be readily available or simple to produce.

B. NeRF and Instant Neural Graphic Primitives

The key advantage of NeRF [1] is its ability to learn from a set of 2D texture images, obviating the need for depth maps mentioned in II-A. However, the required training time is a downside of NeRF that potentially limits its practical application. Instant-NGP [2] provides a solution that yields performance comparable to NeRF with significantly reduced training time by utilizing multiresolution hash encoding and minimizing the number of MLP layers. This parametric encoding maps input coordinates to a table of trainable features at each resolution. As the parameters of hash-encoded features avoids fully connected relations, an increase in total parameters does not proportionally inflate the training time.

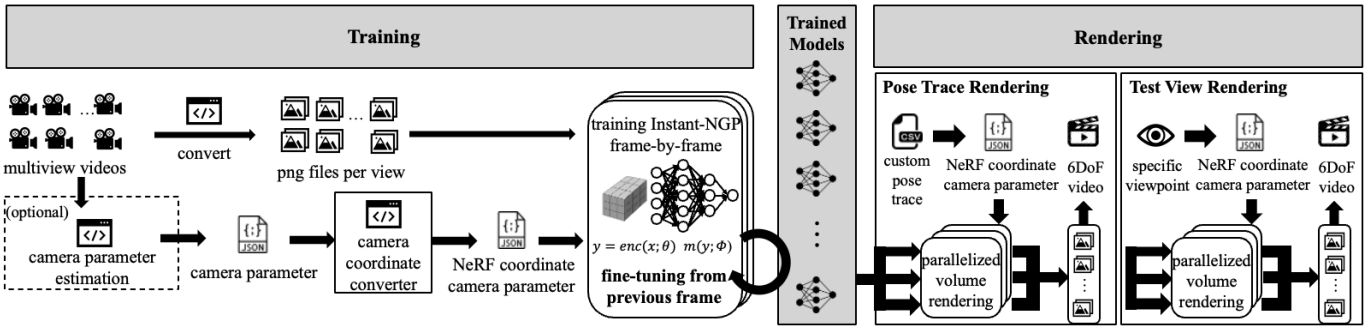


Fig. 2: System architecture

C. Dynamic Scene Representation with Neural Networks

The aforementioned models have a limitation that they can only represent static scenes. Recent research [7]–[14] has developed neural network models that encodes the movement of the scenes. Specifically, Neural Volumes [7] employs voxel grid and encoder-decoder network to represent dynamic 3D scene. D-NeRF [10] introduces a two-stage network. Initially, it translates time-specific coordinates into a difference from canonical space, and subsequently predicts color and density from this canonical space to represent object motion. Its limitation arises from mapping all movements to this specific space, which complicates the representation of objects that appear or disappear over time. On the other hand, DyNeRF [11] presents time-conditioned neural radiance fields that utilizes a time-variant code in addition to location and viewing direction to encode dynamic scenes from multi-view videos. However, despite its performance, DyNeRF's requirement of 1.3k GPU hours for a mere 10-second clip poses a substantial challenge in terms of training duration.

Improved NeRF models for dynamic scenes encapsulate information from various timeframes into a single model, enhancing model size efficiency. However, their complex neural network structures inevitably lengthen training times, and their comprehensive temporal representation may be redundant for individual frame processing. To address these problems, this paper utilizes Instant-NGP [2], a NeRF model designed for static scenes and tailored for quicker training and efficient per-frame inference. Notably, Mixvoxels [13] and K-planes [14] introduces more explicit method for dynamic scene representation. Mixvoxels represent scenes using separate static and dynamic voxels. K-Planes transforms high-dimensional scenes —those altering in appearance or motion over time— into multiple 2D planes. By leveraging only a linear computation for specific queries, K-planes achieves faster operations compared to typical dynamic NeRF methods. This paper also presents experiments that compare the proposed Instant-NGP-based method with K-planes.

D. Parameter Sharing and Transfer Learning

In deep learning, parameters are often shared within different sections of a model or among multiple models [15]. Transfer learning denotes utilizing features learned from one task to improve learning in another, which is beneficial when dealing with scarce but related data across models [16]. Fine-tuning is the method for transfer learning that adjusts the parameters from pre-trained model, tuning it for the new task. Recognizing the inherent similarity between consecutive frames, this paper introduces an approach to apply fine-tuning across neural radiance fields models representing sequential scene.

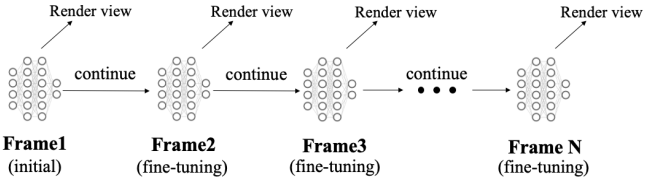


Fig. 3: The concept of parameter sharing through time scale

III. Proposed Method

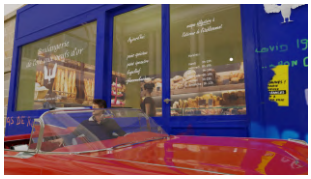
A. Fine-tuning the Instant-NGP Models through Time

This paper propose the method of applying parameter sharing on Instant-NGP [2] by fine-tuning the models at each frame. Specifically, the proposed approach involves utilizing the parameters learned from the previous frame to continue the training of the model for the subsequent frame, as shown in Fig. 3. Using weights from a similar scene to initialize a neural network can lead to faster training to optimal weights than random initialization. Afterward, 6DoF video reflecting the rotation and movement of user is generated by concatenating frame-wise images rendered from a customized pose trace. Training a neural radiance model with parameter sharing offers two primary benefits:

- Enhanced image rendering quality at same training iterations for each frame.
- Improved frame-to-frame consistency in the resulting video.

B. System Architecture

This section presents the proposed deep learning-based pipeline that transforms multi-view videos into 6DoF immersive video based on custom pose, as shown in Fig. 2. Initially, camera parameters are extracted from the multi-view video using structure from motion (SfM) algorithms [17], [18]. Camera parameters are mapped to NeRF coordinate using *camorph* [19], and videos are converted to images with



(a)

(b)

(c)

Fig. 4: Sample images of MIV test sequences– (a) *Cadillac* [21], (b) *Fan* [22], (c) *CBABasketball* [23]

ffmpeg [20] for training. Sequential Instant-NGP training is performed by fine-tuning each frame based on the previous one. Training progresses frame-by-frame, while rendering can be done in parallel. Two rendering options are available: 6DoF pose trace rendering and fixed viewpoint rendering, with the latter being used in upcoming experiments to compare image quality against ground truth from identical viewpoints. Training step excludes videos from the rendering viewpoint for a fair comparison.

C. Environment and Datasets

This section details the experimental setup. The MIV standard test sequences [24] are chosen as the dataset to facilitate comparison with MIV benchmarks. Three test sequences are selected: *Cadillac*, *Fan*, and *CBABasketball*. *Cadillac* sequence [21], shown in Fig.4(a), is taken by a 5×3 camera grid. Each frame has a resolution of 1920×1080 pixels. The challenge of *Cadillac* is maintaining the reflections on the car and glass. *Fan* sequence [22], shown in Fig.4(b), has its primary challenge being the accurate capture of the fan's thin wires. The *CBABasketball* sequence [23], displayed in Fig.4(c), contains 300 frames of a live basketball game, captured by 30 fisheye cameras, each frame being 2048×1088 pixels. View 6 (v6) and v8 are used as test set for *Cadillac* and *Fan*, while v14 and v25 are used as test set for *CBABasketball*. For the experiments, NVIDIA RTX3080 GPU was used for training, achieving computation speed of 95.7 iterations per second with Instant-NGP. The rendering process was parallelized across four identical GPUs.

IV. Experimental Results

A. Effectiveness of Parameter Sharing on Instant-NGP

The experiment of this section evaluates the quality of rendered videos under two conditions: independently training each frame and applying of parameter sharing with fine-tuning. After the process of conducting 1500 iterations for each frame in the *Cadillac*, *Fan*, and *Basketball* sequences, the averaged RGB peak signal-to-noise ratio (PSNR) of all frames is used for quality assessment. Table I indicates that the adapting fine-tuning increased the average PSNR across the three sequences

(a)

(b)

Fig. 5: Rendered images from models for custom pose trace– (a) without applying parameter sharing, (b) applied parameter sharing through time scale

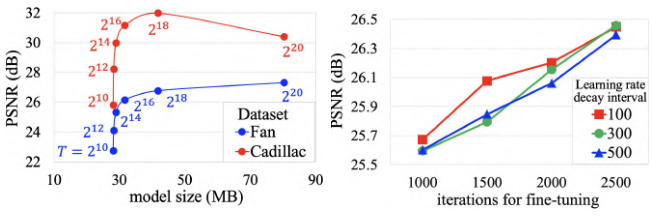
TABLE I: The average PSNR with or without parameter sharing on same number of training iterations (unit: dB)

Experiement	W/o Parameter Sharing	W/ Parameter Sharing
Cadillac (v6)	29.16	**31.37**
Cadillac (v8)	28.71	**30.67**
Fan (v6)	23.07	**26.00**
Fan (v8)	22.80	**25.65**
CBABasketball (v14)	19.47	**22.41**
CBABasketball (v25)	21.18	**25.64**
Average	24.07	**26.96**

from 24.07dB to 26.96dB at same training iterations. Fig. 5 offers a side-by-side subjective quality assessment: without parameter sharing (a) and with parameter sharing (b). When every frames were trained independently, the presence of artifacts and flickering led to inconsistent scene reconstructions, highlighted by the yellow dashed line. In conclusion, applying fine-tuning on Instant-NGP enhanced the quality of the rendered images and improved the subjective consistency when combining these images into a video.

B. Hyperparameters

In Instant-NGP [2], the hyperparameter table size denotes the number of indices that can be allocated in the hash table for each level during multiresolution hash encoding. Increasing the table size lead to an increase in model size but improves performance by reducing collisions. The goal of this experiment is to determine the optimal table size for representing a dynamic scene using multiple Instant-NGP



(a) (b)

Fig. 6: (a) Adjusting the table size, (b) Adjusting the learning rate decay interval when fine-tuning

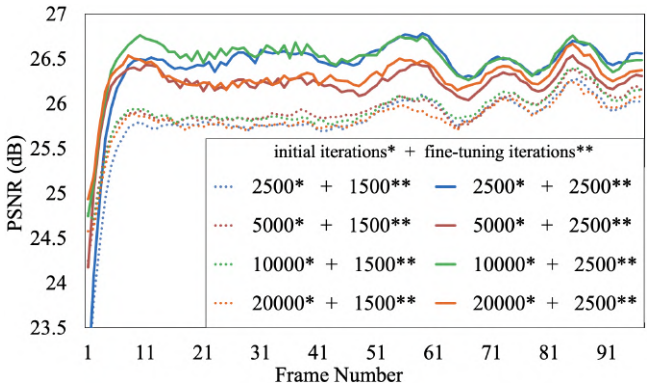


Fig. 7: PSNR per frame when cumulative training is conducted with fine-tuning

models. Fig. 6(a) displays the relationship between average PSNR and model size when adjusting the table size, with a consistent iteration count of 2,000 for the *Cadillac* and *Fan* sequences. As the table size increases from 2^{10} to 2^{16}, the PSNR improves without a significant increase in model size. Beyond 2^{18}, however, the growth in model size becomes significant without a corresponding improvement in quality. A table size of 2^{16} or 2^{18} are considered for optimal balance between model capacity and performance.

The proposed method employed the Adam optimizer [25] with $\beta_1 = 0.9$, $\beta_2 = 0.99$, and $\epsilon = 10^{-15}$, consistent with the original Instant-NGP. Although the original Instant-NGP initiates learning rate decay after 20,000 iterations at intervals of 10,000, these values are adjusted in the this study due to the characteristics of the fine-tuning task, i.e., conducting a very limited number of fine-tuning iterations from previously trained scenes. The starting point of decay is set to 0, and intervals of 100, 300, and 500 were explored. Results presented in Fig. 6(b) suggest that reducing the learning rate decay interval leads to more effective outcomes, attributable to the subtle gradient change caused by rapid decrease in learning rate.

C. Changes of Model Quality Across Successive Training

The experiment aims to determine the optimal number of training iterations when sequentially training multiple Instant-NGP models with shared parameters. Using the *Fan* sequence as a dataset, different iteration counts were tested for the initial frame (2500, 5000, 10000, 25000) and the subsequent frames (1500, 2500). The reason for the distinct number of iterations for the first frame was to prevent a significant decrease in rendering quality of the early frames, as depicted by the blue graphs in Fig. 7. The data from Fig. 7 indicates that while quality generally improved with cumulative training, it began to fluctuate after the 11th frame. This fluctuation can be attributed to quality degradation when the differences between successive frames, such as object movement, become more pronounced. Increasing the number of fine-tuning iterations from 1500 to 2500 typically led to higher quality. This effect seemed to diminish as the total amount of accumulated training iterations increased.

TABLE II: Comparison between TMIV, K-Planes, and Our Method (PSNR unit: dB, time unit: hh:mm:ss, size unit: MB)

(a) *Cadillac* sequence (97 frames)

	TMIV	K-Planes	Our Method
Avg. PSNR (v6, v8)	30.86	23.05	**31.83**
training(encoding) duration	02:05:51	01:52:24	**00:40:06**
rendering(decoding) duration	01:50:46	01:22:05	**00:13:45**
total model(atlas) size	5149	**580**	4049

(b) *Fan* sequence (97 frames)

	TMIV	K-Planes	Our Method
Avg. PSNR (v6, v8)	**27.62**	23.70	27.07
training(encoding) duration	03:03:00	01:53:18	**00:43:11**
rendering(decoding) duration	01:46:33	01:24:25	**00:10:54**
total model(atlas) size	5149	**580**	4049

(c) *Basketball* sequence (300 frames)

	TMIV	K-Planes	Our Method
Avg. PSNR (v14, v25)	**26.36**	23.31	24.03
training(encoding) duration	41:34:44	01:29:32	**01:10:25**
rendering(decoding) duration	15:54:04	01:48:47	**00:37:31**
total model(atlas) size	48129	**580**	12528

D. Performance Analysis

This section evaluates the proposed method by comparing it with existing techniques for encoding 6DoF videos. The test model for immersive video (TMIV) [26] was utilized in its main anchor mode, concurrently with the K-Planes [14], a model capable of representing dynamic scene. For the proposed method, the table size is set to 18, with the learning rate decay interval of 100. However, achieving an entirely fair comparison presents challenges due to differing conditions between methods such as the use of either deep learning or video encoding or the usage of GPUs. Table II presents the experimental results. The proposed method significantly reduced the time needed for both training a model and rendering a video compared to alternative methods. For scene captured in confined spaces like *Cadillac* and *Fan*, the proposed method showed higher PSNR comparable to other models, but for larger scene datasets, it fell short of the PSNR achieved by TMIV. Regarding bitrate, the proposed method showed higher values compared to K-Planes, which represents dynamic scenes with a single model. However, this can potentially be improved in further studies by applying techniques such as [27] and [28].

V. Conclusion

This paper investigates the effect of parameter sharing among neural radiance fields models when training sequential frames. Fine-tuning Instant-NGP models has improved video rendering quality, showing an advantage over randomly initializing parameters at each frame. By employing this technique, a new pipeline was developed to produce 6DoF immersive videos, achieving temporal advantage compared to existing methods. Additionally, the proposed method enhanced the quality of rendered images in specific datasets. Future studies may explore model compression techniques for the presented method or adapt it for real-time rendering applications.

REFERENCES

[1] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis," in *European Conference on Computer Vision (ECCV)*, 2020.

[2] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics (ToG)*, vol. 41, no. 4, pp. 1–15, 2022.

[3] A. P. Pozo, M. Toksvig, T. F. Schrager, J. Hsu, U. Mathur, A. Sorkine-Hornung, R. Szeliski, and B. Cabral, "An integrated 6DoF video camera and system design," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, pp. 1–16, 2019.

[4] J. M. Boyce, R. Doré, A. Dziembowski, J. Fleureau, J. Jung, B. Kroon, B. Salahieh, V. K. M. Vadakital, and L. Yu, "MPEG immersive video coding standard," *Proceedings of the IEEE*, vol. 109, no. 9, pp. 1521–1536, 2021.

[5] J.-B. Jeong, S. Lee, and E.-S. Ryu, "Sub-bitstream packing based lightweight tiled streaming for 6 degree of freedom immersive video," *Electronics Letters*, vol. 57, no. 25, pp. 973–976, 2021.

[6] S. Lee, J.-B. Jeong, and E.-S. Ryu, "Efficient Group-Based Packing Strategy for 6DoF Immersive Video Streaming," in *2022 International Conference on Information Networking (ICOIN)*. IEEE, 2022, pp. 310–314.

[7] S. Lombardi, T. Simon, J. Saragih, G. Schwartz, A. Lehrmann, and Y. Sheikh, "Neural volumes: Learning dynamic renderable volumes from images," *arXiv preprint arXiv:1906.07751*, 2019.

[8] Y. Du, Y. Zhang, H. X. Yu, J. B. Tenenbaum, and J. Wu, "Neural radiance flow for 4D view synthesis and video processing," *arXiv preprint arXiv:2012.09790*, 2020.

[9] Z. Li, S. Niklaus, N. Snavely, and O. Wang, "Neural scene flow fields for space-time view synthesis of dynamic scenes," *https://arxiv.org/abs/2011.13084*, 2020.

[10] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-nerf: Neural radiance fields for dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 318–10 327.

[11] T. Li, M. Slavcheva, M. Zollhoefer, S. Green, C. Lassner, C. Kim, T. Schmidt, S. Lovegrove, M. Goesele, R. Newcombe *et al.*, "Neural 3D Video Synthesis From Multi-View Video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5521–5531.

[12] J. Fang, T. Yi, X. Wang, L. Xie, X. Zhang, W. Liu, M. Nießner, and Q. Tian, "Fast dynamic radiance fields with time-aware neural voxels," in *SIGGRAPH Asia 2022 Conference Papers*, 2022, pp. 1–9.

[13] F. Wang, S. Tan, X. Li, Z. Tian, Y. Song, and H. Liu, "Mixed neural voxels for fast multi-view video synthesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 19 706–19 716.

[14] S. Fridovich-Keil, G. Meanti, F. R. Warburg, B. Recht, and A. Kanazawa, "K-planes: Explicit radiance fields in space, time, and appearance," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 12 479–12 488.

[15] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[16] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *Advances in neural information processing systems*, vol. 27, 2014.

[17] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[18] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise View Selection for Unstructured Multi-View Stereo," in *European Conference on Computer Vision (ECCV)*, 2016.

[19] B. Brand, M. Bätz, and J. Keinert, "Camorph: A toolbox for conversion between camera parameter conventions," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2022.

[20] S. Tomar, "Converting video formats with FFmpeg," *Linux Journal*, vol. 2006, no. 146, p. 10, 2006.

[21] R. D. Gérard Briand, Franck Thudor, "Interdigital cadillac synthetic content proposal for advanced miv investigations on transparency and specularity." 135th MPEG meeting of ISO/IEC JTC1/SC29/WG11, MPEG2021/m57186, 2021.

[22] J. F. Renaud Doré, Gérard Briand, "Fan content proposal for MIV." 131th MPEG meeting of ISO/IEC JTC1/SC29/WG11, MPEG2020/m54732, 2020.

[23] Y. Bai, X. Sheng, S. Li, C. Wang, and L. Yu, "Undistorted CBA Basketball Test Sequence for MPEG-I Visual." 137th MPEG meeting of ISO/IEC JTC1/SC29/WG11, MPEG2020/m58500, 2021.

[24] "Common test conditions for MPEG Immersive Video." Standard ISO/IEC JTC1/SC29/WG11, MPEG/n00342, 2023.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[26] "Test Model 15 for MPEG immersive video." Standard ISO/IEC JTC1/SC29/WG4, MPEG/n00271, 2022.

[27] M. Nagel, M. Fournarakis, R. A. Amjad, Y. Bondarenko, M. Van Baalen, and T. Blankevoort, "A white paper on neural network quantization," *arXiv preprint arXiv:2106.08295*, 2021.

[28] H. Kirchhoffer, P. Haase, W. Samek, K. Müller, H. Rezazadegan-Tavakoli, F. Cricri, E. B. Aksu, M. M. Hannuksela, W. Jiang, W. Wang *et al.*, "Overview of the neural network compression and representation (nnr) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 5, pp. 3203–3216, 2021.