

OmniPlane: A Recolorable Representation for Dynamic Scenes in Omnidirectional Videos

Simin Kou , Fang-Lue Zhang* , Member, IEEE,
Jakob Nazarenus , Reinhard Koch , and Neil A. Dodgson 

Abstract—Consumer-level omnidirectional video offers an economically viable means to create virtual reality (VR) assets, enabling users to explore and interact within a fully immersive visual environment. However, editing such videos, particularly those with 360° views and dynamic objects, poses significant challenges. Existing approaches to representing and manipulating omnidirectional content—whether designed for typical 2D perspective imagery or panoramas—often fail to adequately capture the complex spatiotemporal relationships crucial for producing high-quality, editable outputs in dynamic, panoramic settings. To overcome these challenges, we introduce OmniPlane, a novel method that leverages spherical spatiotemporal feature grids to empower the representation and editability of real-world dynamic omnidirectional environments casually captured by commodity omnidirectional cameras. OmniPlane computes spatiotemporal features by fusing vectors or matrices from each learnable spatial and spatiotemporal feature plane within a spherical coordinate system, complemented by a specifically designed weighted sampling strategy respecting the inherent spherical distribution of omnidirectional content. These learned feature planes can be flexibly decomposed into palette-based color bases. This innovative method not only enhances the representation capability of omnidirectional content and dynamics but also enables the recoloring of omnidirectional videos. Extensive experiments and a dedicated user study validate the superior performance of our proposed method in facilitating recolorable representations of dynamic omnidirectional environments.

Index Terms—Neural representation, 360° video editing, scene reconstruction, virtual reality.

I. INTRODUCTION

OMNIDIRECTIONAL videos, also known as 360° videos, offer an economically viable means to create virtual reality (VR) assets from surroundings in the physical world. As a crucial component of the rapidly evolving metaverse, these videos are expected to provide highly interactive and vivid experiences. Recent advances have thrust the importance of 360° video analysis and processing to the forefront, sparking extensive research [1], [2]. However, existing research for handling omnidirectional inputs [3]–[7] predominantly assumes static scenes when reconstructing inherent 3D scenes, and thus falls short in scenarios that require dynamic content adaptation and high interactivity, essential for enhancing realism and

Simin Kou, Fang-Lue Zhang, and Neil A. Dodgson are with the School of Engineering and Computer Science, Victoria University of Wellington, Wellington 6012, New Zealand (e-mail: simin.kou@vuw.ac.nz; fan-glue.zhang@vuw.ac.nz; neil.dodgson@vuw.ac.nz).

Jakob Nazarenus and Reinhard Koch are with the Department of Computer Science, Kiel University, Kiel 24118, Germany (email: jna@informatik.uni-kiel.de; rk@informatik.uni-kiel.de).

*Fang-Lue Zhang is the corresponding author.

user engagement in VR environments. Moreover, such approaches do not support the manipulation of dynamic scenes, limiting their ability to edit and personalize content within VR environments, which are essential for an immersive and interactive VR experience. Our research aims to address these limitations by developing a novel approach that enhances both the handling of dynamic elements and the editability of content in omnidirectional videos, enriching the VR experience by making it more interactive and customizable.

Recent advances in Neural Radiance Fields (NeRF) [8] and 3D Gaussian Splatting (3D-GS) [9] have spurred the exploration of dynamic and editable representations based on typical 2D perspective images. Dynamic NeRFs use deformable fields to model motions [10]–[12] or incorporate scene flows [13], [14] to describe movements within the scene, thus enhancing temporal coherence in the scene reconstructions. To address the high computational costs of the above methods, voxel-based [15], [16] and feature plane-based solutions [17], [18] are employed to encode temporal changes within the dynamic scene. One HexPlane-based approach [18] has also been combined with 3D-GS [19] to predict Gaussian deformations at novel timestamps. Despite these advances, existing methods are inadequate for handling outward-facing 360° videos, as they do not fully consider the difficulties brought by the spherical properties of 360° videos, such as the distortions in their equirectangular projection (ERP)-based frame representation and the uneven sparse point cloud reconstruction quality in the pre-processing step. The previous attempts at editing inherent 3D scenes [20]–[22] from a set of 2D perspective images, which relied on similar NeRF-based representations, have also demonstrated limited capabilities in processing the dynamics and complexities of omnidirectional video content.

Our work focuses on enhancing the capability of representing dynamic content and the editability of 360° video representations. We introduce OmniPlane, a recolorable spatiotemporal omnidirectional representation, which captures dynamic spherical relationships in 360° videos by combining spherical spatial and spatiotemporal feature planes and also achieves recoloring capabilities through palette-based color decomposition on the learned feature planes. Inspired by feature grid-based dynamic scene representations [17], [18], OmniPlane decomposes a designed 4D spacetime spherical grid into six learnable spherical feature planes, representing dynamic environments in 360° videos. Each of the feature planes spans a pair of spherical coordinate axes (longitude Θ , latitude Φ , radial R) or includes a temporal axis T (e.g., $\Theta\Phi$, RT , ΘR , ΦT). To respect the inherent distribution of such spherical content on ERP video

frames, ray sampling across the feature planes that include the Φ axis is guided by a pre-computed weight map. This map allocates different weights across latitudes, with lower weights near the poles and higher weights near the equator. OmniPlane computes a fused feature vector for each 4D spherical point in spacetime by projecting it onto each feature plane. This vector serves as the density information of each 4D point and is put through a small MLP to predict its color. We further propose to decompose OmniPlane into palette-related bases to facilitate 360° video recoloring. The dynamic nature of 360° videos, crucial for VR applications, requires motion-color coherence across spherical content—a challenge not addressed by existing static scene-focused methods [21], [23]. To address this, we encode 4D spherical spatiotemporal positions to learn a specialized palette that captures global dominant colors, complemented by bases for pixel-level adjustments. This design ensures color consistency during motion, enhancing recoloring quality in dynamic omnidirectional environments. Furthermore, we introduce a novel hue-focused regularization to enhance palette representativeness, overcoming the issue of limited palette color diversity of previous methods and enabling more effective user-driven customization. Extensive experiments and a user study were conducted to demonstrate the superiority of our method over existing approaches in both dynamic scene representation and editability.

The main contributions of this work can be summarized as:

- We propose OmniPlane, a novel representation tailored for omnidirectional videos with dynamic objects. It involves spherical feature planes across spatial and temporal dimensions, with weighted sampling to capture the spherical distributions of 360° videos, providing an effective representation of dynamic omnidirectional scenes.
- We design a palette decomposition-based learning framework that empowers OmniPlane to deliver natural and motion-consistent recoloring of dynamic scenes in 360° videos. It uniquely tackles the challenges of maintaining global color coherence in spherical video content.
- We introduce a novel hue-focused regularization to enhance palette representativeness, enabling intuitive and effective user control beyond prior palette-based methods.
- We establish a new real-world 360° video dataset with various dynamic objects, camera motions, and their spherical camera poses, providing a valuable resource for advancing 360° video research.
- We conduct a dedicated user study to evaluate the performance of dynamic scene representation and recoloring quality in 360° videos, offering practical insights into their applicability in VR environments.

II. RELATED WORK

A. Neural Representation in Videos

Neural representation methods excel in video processing due to their non-linear learnable capacity, enabling them to adeptly manage the inherent complexity of videos. This ability allows them to effectively understand and interpret both the spatial details and temporal changes, resulting in robust performance across complex scenarios.

Recent advances in neural video representation—such as frame-based [24], [25], patch-based [26], and pixel-based [27] approaches—have shown promise in video compression, inpainting, and editing. The field has also expanded into novel applications like text-driven video stylization [28], [29], video deflickering [30], and face video manipulation [20]. However, challenges remain in 360° videos, especially in modeling omnidirectional motion due to limitations in capturing inherent spatial relationships between spherical pixels.

Neural Radiance Fields (NeRF) [8] have revolutionized photorealistic rendering by using implicit continuous functions, namely multi-layer perceptron (MLPs), to transform positional coordinates into color and volume density. To achieve more efficient storage and computation, tri-plane representations have been introduced in NeRF-like models, mitigating the need to query dense 3D grids. TensoRF [31], as the pioneering work, significantly reduces training time and inference costs for static scenes. This feature plane-based approach has since been extended to dynamic scene representations [17], [18], [32] through the integration of temporal encoding. More recently, techniques such as 3D Gaussian Splatting (3D-GS) [9], [33] have been introduced to facilitate real-time rendering using a 3D Gaussian-based scene representation. To enhance the editability of such neural representations, various methods have been developed [34]–[36], enabling modifications to learned volumetric representations. Nevertheless, the current editable NeRF methods, primarily designed for static scenes from multi-view typical 2D perspective images, struggle to effectively represent outward-facing captured omnidirectional inputs with a non-converged camera setup, featuring a non-grid structure and non-uniform sampling properties.

B. Omnidirectional Representation

To effectively utilize the omnidirectional data that is crucial for representing 360° content, methods have been developed that go beyond traditional perspective images. Multi-Sphere Images (MSI) stand out in 360° view synthesis, offering rapid rendering [37]–[39]. However, MSIs require substantial memory, are limited by a narrow field-of-view (FoV), and depend on specifically constrained small camera paths relative to the scene size for effective light field capture. In contrast, casual capture using commodity omnidirectional cameras, such as those with dual fish-eye lenses, is more practical. Technologies like OmniPhotos [40] and egocentric approaches [3] use a deformed geometric proxy with pre-computed optical flow and depth estimation. Yet, their effectiveness heavily relies on the performance of pre-trained models.

NeRF-based methods, like OmniNeRF [41] and EgoNeRF [5], lead scene reconstruction from omnidirectional images by integrating spherical data from fish-eye projections to synthesize novel views. Otonari et al. [42] introduce distortion-aware and content-aware ray sampling into the vanilla NeRF framework, to improve the novel view synthesis performance on 360° images. Additionally, approaches like 360Roam [4], 360FusionNeRF [43], and PanoGRF [44] focus on roaming applications by learning omnidirectional radiance fields. Recent developments in 3D-GS [9], such as 360-GS [6]

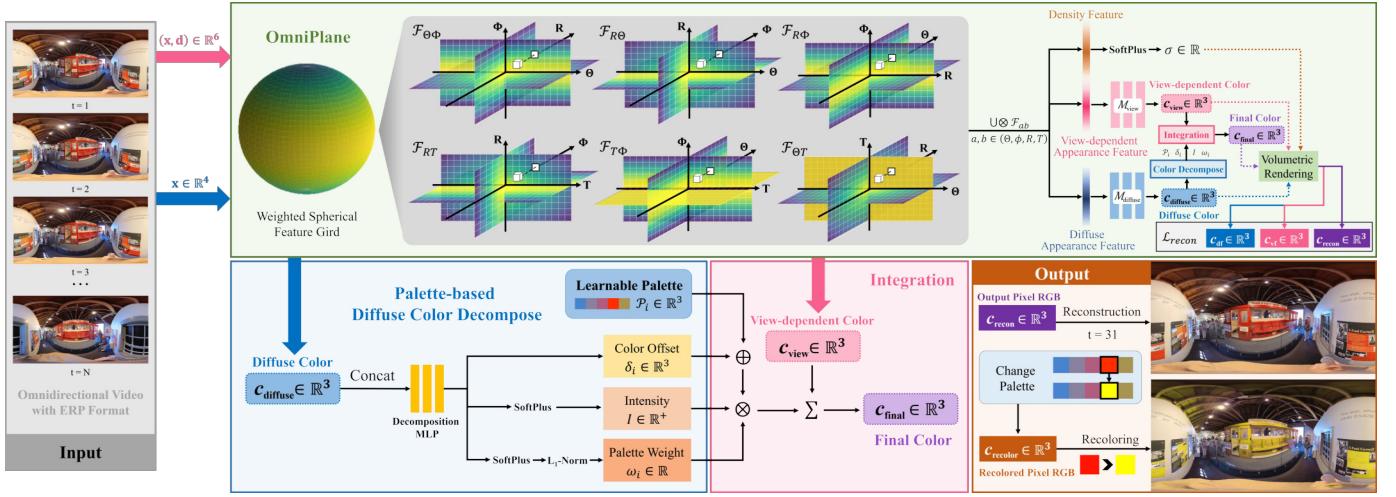


Fig. 1. An overview of the proposed OmniPlane. OmniPlane takes an omnidirectional video in ERP format as input, reconstructs it, and then decomposes it into editable components. OmniPlane consists of six pairs of spherical feature planes, employing uniform ray sampling across the Θ and T axes (e.g., the all-yellow feature plane $\mathcal{F}_{\Theta T}$ with uniform weights) and cosine-weighted non-uniform sampling for the Φ axis to align with ERP properties (e.g., the feature plane $\mathcal{F}_{T\Phi}$ displaying weights from yellow to blue to represent diminishing importance of ray sampling along the Φ axis). Exponential sampling is used along the radial R axis to encourage the consistent performance of such a spherical appearance (e.g., the feature plane $\mathcal{F}_{R\Theta}$ depicting radial weight distribution from the center outward). OmniPlane is trained to derive diffuse and view-dependent RGB values from spatiotemporal positions $\mathbf{x} = (\theta, \phi, r, t) \in \mathbb{R}^4$ and combinations $(\mathbf{x}, \mathbf{d}) \in \mathbb{R}^6$ of positions \mathbf{x} with ray directions \mathbf{d} , respectively. To facilitate color editing, a palette-based module decomposes the diffuse color $\mathbf{c}_{diffuse} \in \mathbb{R}^3$ into several components: learnable palette $\mathcal{P}_i \in \mathbb{R}^3$, color offset $\delta_i \in \mathbb{R}^3$, intensity $I \in \mathbb{R}^+$, and palette blending weights $\omega_i \in \mathbb{R}$. These components and the view-dependent color $\mathbf{c}_{view} \in \mathbb{R}^3$ are integrated to generate the final RGB colors $\mathbf{c}_{final} \in \mathbb{R}^3$. The \mathbf{c}_{view} , $\mathbf{c}_{diffuse}$, and \mathbf{c}_{final} represent color components of a sampled point \mathbf{x} along the rays, can be further rendered into pixel-wise colors \mathbf{c}_{vf} , \mathbf{c}_{df} , and \mathbf{c}_{recon} through volumetric rendering with obtained density σ for optimization and reconstruction. After training, this model enables the recoloring of omnidirectional videos by adjusting their learned palettes.

and OmniGS [7], have expanded real-time omnidirectional rendering applications. However, these NeRF-based and 3D-GS-based representations struggle with dynamic objects and lack user-friendly editing capabilities.

C. Omnidirectional Content Manipulation

Zhu et al. [45] introduced an inpainting method for 360° images, which addresses distortions through structure-rectifying warping and fills gaps via a 2D completion technique [46]. Xu et al. [47] further advanced the field by proposing a coarse-to-fine optimization approach, iteratively restoring missing pixels and motion information while accounting for the geometric properties of spherical images. Efforts to enhance visual comfort in 360° videos have used video stabilization techniques [48], [49], considering content distortion within the equirectangular representation. Zhang et al. [50] proposed edit propagation methods for 360° panoramas to perform the global color change. Li et al. [51] introduced bullet comments into 360° videos, while Wong et al. [52] designed a solution for view-adaptive asymmetric detail enhancement, and Vermast et al. [53] claimed that spherical and cube-shaped 3D thumbnails provide a better user experience when searching for details in 360° videos. Huang et al. [54] proposed a 360° image composition method, leveraging 3D point clouds and a per-view projection technique to insert objects into stereo 360° panoramas with accurate depth perception and reduced ghost artifacts. Xu et al. [55] presented a decomposition method for the illumination of omnidirectional images, facilitating the modification of reflectance or shading effects. Ai et al. [56] introduced a transformer-based 360° image outpainting framework, enabling 360° panorama generation from narrow FoV images. While these approaches have demonstrated promise

in handling specific tasks with omnidirectional data, they fall short of providing a representation that maintains temporal coherence across all elements of a dynamic scene, essential for supporting comprehensive 360° video editing tasks.

III. METHODOLOGY

A. Overview

To develop a novel framework for representing dynamic scenes in omnidirectional videos and facilitating appearance edits, we propose OmniPlane. It is based on the following three key observations: (1) Equirectangular projection is the predominant format used by consumer-level omnidirectional cameras (e.g., Insta360) and HMDs (e.g., Meta Quest). We prioritize solutions that utilize the ERP format, as it is more convenient and manageable for user interactions and processing. (2) Identifying representative elements within the omnidirectional video streamlines the editing process, making it more accessible and user-friendly. (3) Real-life 360° videos are commonly captured casually by users with handheld cameras, inherently limiting the spatial range of available viewpoints. This situation, combined with the omnidirectional nature that captures every angle of the scene in each frame, maintains a relatively consistent color distribution throughout the video.

Fig. 1 gives an overview of our proposed OmniPlane representation model. Our spatial feature grids are defined in a spherical coordinate system, where the position of a given point in space is specified by the longitude (Θ), the latitude (Φ), and its radial distance from the origin (R). In our representation, using multiple feature planes, the spatial feature planes span the following pairs of spherical coordinate axes: $\Theta\Phi$, ΘR , and ΦR . To encode motion information, we use feature planes spanning a temporal axis (T) and each of the

spatial axes (Θ, Φ, R), forming three spatiotemporal feature planes: ΘT , ΦT , and RT . These spatial and spatiotemporal feature planes collectively compose the OmniPlane, which serves as our representation capturing omnidirectional density and appearance features. View-dependent colors, c_{view} , are rendered through a lightweight MLP $\mathcal{M}_{\text{view}}$ based on learnable feature planes \mathcal{F}_{ab} ($a, b \in \{\Theta, \Phi, R, T\}$) of each involved 4D spatiotemporal position $\mathbf{x} = (\theta, \phi, r, t)$ along with the ray direction \mathbf{d} , and it is defined as follows:

$$c_{\text{view}} = \mathcal{M}_{\text{view}}(\mathcal{T}(\mathcal{F}_{ab}, \mathbf{x}), \mathbf{d}), \quad (1)$$

where \mathcal{T} denotes a bilinear interpolation for querying the corresponding feature for each position on each plane. Diffuse colors, $c_{\text{diffuse}} = \mathcal{M}_{\text{diffuse}}(\mathcal{T}(\mathcal{F}_{ab}, \mathbf{x}))$, are produced by another MLP $\mathcal{M}_{\text{diffuse}}$ using only 4D spatiotemporal positions \mathbf{x} . These diffuse colors are then decomposed into several learnable components—an extracted palette \mathcal{P}_i , (where i is the number of the base colors in the palette) for the entire scene, along with each position's color offsets δ_i from base colors, intensity I , and palette blending weights ω_i —enhancing the editability of the scene in the given omnidirectional video. Users can adjust the learned palette \mathcal{P}_i to recolor the omnidirectional scene, and the integration of these palette-based components with view-dependent colors predicts the final colors c_{final} for each position. The final reconstructed colors of pixels c_{recon} are generated through volumetric rendering based on the above colors of the sampled points in the volume together with their obtained densities $\sigma = \text{softplus}(\mathcal{T}(\mathcal{F}_{ab}, \mathbf{x}))$.

B. OmniPlane: Omnidirectional Feature Planes

A dynamic omnidirectional scene could be naturally represented as a spherical 4D volume, comprising static 3D volumes at each frame $t \in 1, 2, \dots, T$. However, such a representation is exceedingly memory-intensive, requiring $O(N^3TF)$ space, where N , T , and F denote spatial resolution, temporal resolution, and feature size (F is 1 for the density feature grid and 27 for the appearance feature grid). Furthermore, modeling separate 3D volumes for each time step contradicts the inherent entanglement of space and time observed in the real world. To compute with manageable computational overhead and memory usage while facilitating information sharing across different times, we build our omnidirectional representation based on HexPlane [18]. Hexplane considers a dynamic scene as a 4D feature volume, denoted by $V = XYZT \in \mathbb{R}^F$, employing a factorization technique over each learned feature plane M^{AB} , where $A, B \in \{X, Y, Z, T\}$. However, the method cannot be trivially applied to spherical images by simply replacing the spatial Cartesian coordinate system XZY with the spherical coordinate system $\Theta\Phi R$. In Hexplane, rays and 3D positions are sampled uniformly when querying within feature planes, which is not suitable for sampling $\Theta\Phi R$ space, because the visual feature distribution is not balanced along the axes of Θ , Φ , and R . To accurately model omnidirectional images, we need a new approach to accommodate the spatial visual feature distribution in the spherical coordinate system.

Specifically, in the spherical coordinate system, using uniform grids along both Θ and Φ causes high-valence vertices

near the poles. While this arrangement uniformly covers all the pixels of the ERP and is easy to implement, it creates an unbalanced distribution of the spherical content, causing oversampling at the regions near the poles and undesirable artifacts. Solutions like EgoNeRF [5], which introduced a yin-yang spherical grid, have been developed to mitigate these issues. However, EgoNeRF introduces discontinuities between the two hemispheres of the grid, particularly impacting editing applications. To preserve a unified spherical representation while solving the oversampling issue, we design a soft-weighted ray sampling strategy that considers the actual data distribution. This strategy favors areas near the equator with a higher probability of sampling and reduces the probability near the poles. The sampling probability weight, W_ϕ , is defined as:

$$W_\phi = \frac{\cos\phi \cdot \lambda_\phi + 1}{\sum(\cos\phi \cdot \lambda_\phi + 1)}, \quad (2)$$

where λ_ϕ is a soft factor for the latitude importance sampling. W_ϕ is applied along the Φ axis and coupled with uniform weights along the Θ or T axis to create a weight map, as illustrated in Fig.2. This map guides the sampling of rays passing through the feature plane $\mathcal{F}_{\Theta\Phi}$ or $\mathcal{F}_{T\Phi}$. Since Θ and T axes exert a uniform influence, we maintain uniform sampling of rays intersecting the feature plane $\mathcal{F}_{\Theta T}$, as depicted by the map in Fig.1, which shows a uniform yellow color.

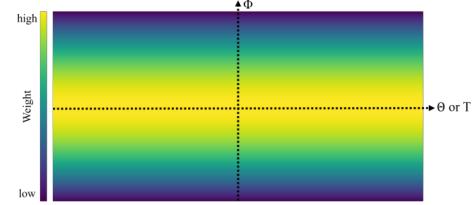


Fig. 2. Illustration of a weight map to guide the sampling of rays passing through the feature plane $\mathcal{F}_{\Theta\Phi}$ or $\mathcal{F}_{T\Phi}$.

Using the above ray sampling weight map, we randomly generate rays and use their pixel values to supervise the learning of the feature planes and their mappings to density and color. This design makes the learned feature planes have a balanced feature distribution corresponding to the actual spherical visual content. Moreover, omnidirectional data inherently captures more details of those objects nearer to the center of the sphere, while more distant objects occupy proportionally smaller areas on the projected images. To accommodate this property, where feature grid cells with the same volume contain less content as distance increases radially, we use exponentially increasing sampling [5] intervals along the radial R axis within our designed 4D spherical feature grid. For the R axis, which lacks a predefined range unlike the Θ and Φ axes that correlate directly with the angular space of $360^\circ \times 180^\circ$, we need to establish an appropriate range $r_R \in [r_{\text{near}}, r_{\text{far}}]$. Here, r_{near} and r_{far} represent the nearest and farthest distances along the R axis. As depicted in Fig. 3 (a), we sample five points (illustrated in blue) along a ray (represented by a gray dashed line), where the interval between each adjacent pair of points increases exponentially with the radius. The interval r_q along the R axis is defined as:

$$r_q = r_0 \tau^{N_q - 1}, \quad (3)$$

where r_0 is the minimum interval between the starting point of each sampled ray r_{near} and the first sampled point, and N_q denotes the number of sampled points along each ray. The exponential increase ratio, τ , is defined by:

$$\tau = e^{\frac{\log(\frac{r_{\text{far}} - r_{\text{near}}}{r_0})}{N_q - 1}}. \quad (4)$$

Based on the above sampling scheme, we sample points along each ray guided by weight maps when querying the feature plane $\mathcal{F}_{R\Theta}$ or \mathcal{F}_{RT} (Fig. 3(b)) and $\mathcal{F}_{R\Phi}$ (Fig. 3(c)). Our point sampling setup effectively adapts to the varying levels of detail across the radius in omnidirectional scenes.

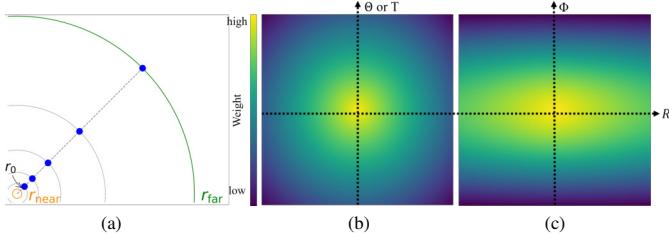


Fig. 3. Illustration of exponentially increasing sampling for points along the radial R axis and weight maps when R couples with other axes. (a) Exponentially increasing intervals between adjacent sampled points along the R axis. (b) Weight map for point sampling across the feature plane $\mathcal{F}_{R\Theta}$ or \mathcal{F}_{RT} . (c) Weight map for point sampling across the feature plane $\mathcal{F}_{R\Phi}$.

C. Palette-based Recolorable OmniPlane

The implicit neural features within the 4D spherical volume are inadequate for flexible editing operations. We enable appearance editing on OmniPlane by decomposing the representation into editable components. Drawing inspiration from the static scene recoloring method PaletteNeRF [21], we develop a palette-based, recolorable OmniPlane. This novel design facilitates spatiotemporally consistent recoloring of dynamic scenes in omnidirectional videos. It incorporates the learning of palette and palette-related bases for each spherical pixel into the OmniPlane optimization process, enabling user-friendly, immersive editing by providing a representative palette.

Given an input omnidirectional video, N_p base colors can be extracted as the initial palette \mathcal{P}_i , where $i \in \{0, 1, \dots, N_p - 1\}$ using the method proposed by Tan et al. [23], alongside the palette blending weights ω_i for each sampled point throughout the scene. To mitigate the influence of view-dependent effects on the color decomposition of each point, such as specular reflections, we employ a decomposition learning approach focused solely on the diffuse color c_{diffuse} derived from 4D spatiotemporal positions \mathbf{x} . The view-dependent color c_{view} is processed by a separate branch and is derived from a combination of 4D spatiotemporal position and ray direction inputs. As depicted in Fig. 1, within the decomposition module, the learned diffuse color is decomposed into three learnable bases δ_i , ω_i , and I , coupled with a learnable palette \mathcal{P}_i , and trained using a decomposition MLP. The three bases are integrated with the learned palette \mathcal{P}_i and view-dependent color c_{view} to compute the final color $c_{\text{final}} \in \mathbb{R}^3$, as defined by:

$$c_{\text{final}} = c_{\text{view}} + I(\mathbf{x}) \sum_{i=1}^{N_p} \omega_i(\mathbf{x})(\mathcal{P}_i + \delta_i(\mathbf{x})), \quad (5)$$

where the palette blending weights ω_i are normalized to ensure accurate color reconstruction. Based on the final colors c_{final} of each sampled point, we reconstruct the final appearance of each pixel on the given omnidirectional video frames through the following volumetric rendering equation:

$$c_{\text{recon}} = \sum_{q \in Q} \left(\left(\prod_{j=1}^{q-1} e^{-\sigma_j r_q} \right) (1 - e^{-\sigma_q r_q}) c_{\text{final}}^q \right), \quad (6)$$

where q denotes each sampled point and Q is a point set including all sampled points per ray.

By decomposing diffuse colors into multiple learnable components and integrating them with view-dependent colors, our OmniPlane becomes recolorable, enabling intuitive appearance editing by manipulating the learned base colors in the palette.

D. Training for OmniPlane

Hierarchical Sampling along Radius. Efficient point sampling along rays is crucial for the performance of volumetric rendering. Our OmniPlane features a unique property: all rays in the spherical coordinate system align with the radial R axis, resulting in exponentially increasing intervals between adjacent sampled points along this axis. While feature grid-based methods typically adjust the classical coarse-to-fine scheme by progressively increasing feature grid resolution to expedite training, this adaptation demands substantial memory, particularly for spherical grids. To address this, we introduce a hierarchical sampling approach leveraging OmniPlane's unique property. Firstly, we use exponential sampling (Section III-B) to sample coarse points along rays. We then predict density values for these coarse points, providing an estimate of the overall point distribution along each ray. Using this density information, we generate fine sampling points by using midpoints between adjacent coarse points and filtering them based on the probability distribution function of weights along the ray. The final set of sampled points for training comprises the sorted concatenation of coarse and fine sampling points. This hierarchical sampling strategy optimizes memory usage while maintaining accuracy in point representation along rays.

Initialization. Although every component in our OmniPlane can undergo optimization during training, Kuang et al. [21] suggest that using an extracted palette in RGB space yields superior results compared to random initialization. Leveraging this finding, we extend upon its palette initialization approach for the palette and blending weights within our framework.

Optimization. Our OmniPlane becomes recolorable through diffuse color decomposition and view-dependent color regression. These processes are optimized jointly to minimize reconstruction loss between rendered and input video frames. For each spatiotemporal spherical point (θ, ϕ, r, t) , OmniPlane queries density, diffuse appearance, and view-dependent appearance features. The appearance features are used to regress diffuse and view-dependent RGB values c_{diffuse} and c_{view} for each sampled point along each ray using separate MLP-based renders $\mathcal{M}_{\text{diffuse}}$ and $\mathcal{M}_{\text{view}}$. Decomposing the diffuse color c_{diffuse} into bases, the final color c_{final} for each sampled point is obtained by combining these bases with the view-dependent color c_{view} using Equation 5. By volumetric rendering, OmniPlane

produces the final diffuse color c_{df} , view-dependent color c_{vf} , and combined color c_{recon} for each pixel. It is essential to note that the sum of the non-decomposed pixel-wise diffuse color c_{df} and the view-dependent color c_{vf} ideally equals c_{recon} . Thus, the reconstruction loss \mathcal{L}_{recon} is defined as:

$$\mathcal{L}_{recon} = \|c_g - c_{recon}\|_2^2 + \|c_g - (c_{df} + c_{vf})\|_2^2, \quad (7)$$

Here, c_g represents the color of the input video frames.

To handle imperfect camera poses in real-world omnidirectional video inputs, we incorporate Total Variations (TV) loss [5], [18], [31] for our spherical feature grid, encouraging spatiotemporal continuity in density and appearance features. Furthermore, given the inherently ill-posed nature of decomposing scene appearances into multiple components, we adopt several losses for regularization: view-dependent color loss \mathcal{L}_s , color offset loss \mathcal{L}_{offset} , and blending weight sparsity loss \mathcal{L}_{sp} , as introduced in PaletteNeRF [21]. We also incorporate $\mathcal{L}_{palette}$ and \mathcal{L}_{weight} for supervising the learnable palette and blending weights. While the above losses have shown promise when learning a reasonable palette, they do not consider the representativeness and diversity of the extracted base colors. Since the diversity predominantly relies on the hue channel in the HSV space, we introduce a novel hue-separation loss \mathcal{L}_{hue} to enhance hue diversity within the learnable palette \mathcal{P}_i . To ensure that the extracted colors are representative and well separated regarding the hues of base colors, we define a threshold k for the hue-channel difference when assessing the hue similarity between base colors in \mathcal{P}_i . If the minimum difference between any pair of base colors is smaller than the threshold k , we encourage a larger difference; otherwise, the penalty is minimal to maintain stability. \mathcal{L}_{hue} is defined as:

$$\mathcal{L}_{hue} = 1 - \frac{1}{1 + e^{\frac{h_{min}-k}{scale}}}, \quad (8)$$

where h_{min} denotes the minimum hue difference between any color pairs in the learned palette \mathcal{P}_i . The $scale$ adjusts sensitivity, smoothing the loss to encourage h_{min} values above threshold k and minimizing penalties once exceeded. Finally, our model is jointly optimized by the weighted sum of all the above loss functions, defined as:

$$\begin{aligned} \mathcal{L} = & \mathcal{L}_{recon} + \lambda_p \mathcal{L}_{palette} + \lambda_w \mathcal{L}_{weight} + \\ & \lambda_s \mathcal{L}_s + \lambda_h \mathcal{L}_{hue} + \lambda_{offset} \mathcal{L}_{offset} + \lambda_{sp} \mathcal{L}_{sp}, \end{aligned} \quad (9)$$

where λ_p , λ_w , λ_s , λ_h , λ_{offset} , and λ_{sp} are trade-off coefficients.

E. Recoloring Omnidirectional Videos via OmniPlane

Our OmniPlane model can seamlessly facilitate recoloring operations by adjusting the values of its learned base colors. This process allows for the immersive modification of color schemes based on user-defined palettes. Initially, when users provide novel palettes or alter any base colors in the current palette, we compute their changes from the original optimized palette within the HSV color space. This involves calculating the differences in the hue (H) values, as well as the scaling for the saturation (S) and value (V) components. Subsequently, these calculated alterations are applied to the per-point soft

color, denoted as $c_s = \mathcal{P}_i + \delta_i(\mathbf{x})$, resulting in the transformed color c'_s . Finally, the model automatically generates the recolored omnidirectional video by computing:

$$c_{recolor} = c_{view} + I(\mathbf{x}) \sum_{i=1}^{N_p} (\omega_i(\mathbf{x}) \times c'_s). \quad (10)$$

IV. EXPERIMENTS

A. Implementation Details

Configurations. Our framework (Fig. 1) consists of three MLPs: two render MLPs for diffuse and view-dependent color regressing, and the third for color decomposition. Each render MLP has a single hidden layer with 128 channels, followed by an output layer with 3 channels for RGB values. The input layer varies depending on the presence of ray directions d and the positional encoding (PE) technique used. We employ PE akin to how it is used in NeRF-like methods, defined as $PE(\mathbf{x}) = \sin(2^l \pi \mathbf{x}), \cos(2^l \pi \mathbf{x})$, with $l \in \{0, 1, 2\}$. The decomposition MLP comprises two parts: the first part has a single hidden layer with 64 channels connecting to an intermediate output layer with 13 channels. This intermediate output is then decoded by two output layers: one with $N_p \times 3$ channels for color offset δ_i , and another with $N_p + 1$ channels for a combination of blending weights ω_i and intensity I . We optimize on multiple losses with empirical trade-off coefficients: $\lambda_p = 0.001$, $\lambda_w = 0.05$, $\lambda_s = 0.1$, $\lambda_h = 0.0002$, $\lambda_{offset} = 0.03$, and $\lambda_{sp} = 0.0002$. For \mathcal{L}_{hue} , we set $k = 15$ and $scale = 5$. All experiments detailed in this paper were conducted on an NVIDIA RTX 3090Ti GPU. Our method takes an average of 3.5 hours to train each video at a resolution of 1920×960 , and around 15 seconds to render each frame.

Dataset. In the absence of publicly available real-world omnidirectional video datasets featuring significant dynamic objects and camera motions, we developed a new dataset named ‘DyOmni’ for evaluating our approach and baseline methods. This dataset comprises sixteen real-world omnidirectional videos capturing diverse indoor and outdoor dynamic scenes captured with a handheld camera. The videos include scenarios involving the videographer’s presence or absence within a large area, along with other dynamic entities in both nearby and distant locations. Some videos also include an additional videographer instead of just one. All videos were recorded with a casually moving 360° camera to closely mimic real-world user capture conditions and facilitate the extraction of camera parameters. Thirteen of the videos were casually captured using commodity omnidirectional cameras—eleven with an Insta360 X3 360 Action Camera and two with a Ricoh Theta Z1—both mounted on a selfie stick. The remaining three videos, ‘GreatWall’, ‘City’ and ‘Ayutthaya’ were sourced externally with permission for research purposes. Each video contains between 80 and 100 frames. We used a Structure-from-Motion (SfM) library OpenMVG [57] to estimate spherical camera parameters from all these videos in the equirectangular projection format. We will release this dataset following the publication of this paper.

Evaluation Metrics. To ensure a comprehensive evaluation of omnidirectional video reconstruction, we employ different

metrics that consider various aspects of fidelity and quality. The unique spherical properties of omnidirectional data necessitate specialized metrics beyond traditional PSNR, SSIM, and LPIPS for accurately assessing perceived image quality in 360° videos, particularly when viewed on virtual reality (VR) displays with non-uniform pixel densities. To handle this, we use weighted-to-spherically uniform evaluation metrics, WS-PSNR [58] and WS-SSIM [59]. They adjust the importance of image regions based on expected visibility to a viewer. For instance, regions more likely to be in the viewer's direct line of sight (near the equator) are weighted more heavily than peripheral regions (near the poles). Additionally, we compute CPP-PSNR [60], to evaluate quality based on resampled pixels on the Craster Parabolic Projection, which preserves the solid area of each spherical surface pixel for a more accurate assessment respecting the omnidirectional content. We further measure the average SSIM and LPIPS across all faces of the CubeMap projection, due to its mitigation of polar distortion and narrow field-of-view that is similar to perspective images, ensuring reasonably reliable perceptual quality assessments. Alongside these specialized metrics, we also calculate and report scores using original PSNR, SSIM, and LPIPS in ERP format. Furthermore, we perform LPIPS using both AlexNet [61] and VGG-Net [62], denoted as LPIPS-a, LPIPS-v, Cube-LPIPS-a, and Cube-LPIPS-v, respectively. Moreover, we conduct a user study to quantitatively evaluate recolored videos of our method and other state-of-the-art (SOTA) methods.

Baseline Methods. To evaluate video reconstruction performance, we chose EgoNeRF [5], HexPlane [18], and 4D Gaussian Splatting (4D-GS) [19] to compare with. EgoNeRF is designed to extract a NeRF-based representation from an omnidirectional video, while HexPlane and 4D-GS are specialized for representing dynamic scenes from perspective images, leveraging feature grids and Gaussian splatting, respectively. For video recoloring, we compare with PaletteNeRF [21], which excels in recoloring static scenes using multiple 2D perspective images, and a palette-based spatiotemporal recoloring method designed for typical perspective videos (4D Geometric Palette) [63]. Additionally, we compare with an alternative that combines EgoNeRF with our color decomposition module (EgoNeRF+Palette) for omnidirectional scene recoloring.

Configurations for Comparative Methods. HexPlane [18] and 4D-GS [19], designed for typical perspective images, rely on camera pose extraction using SfM tools based on pinhole camera assumptions. Further, 4D-GS requires initialization with an SfM-derived point cloud. To ensure fair comparisons, we adapt HexPlane and 4D-GS to omnidirectional videos by projecting the frames into the CubeMap format and applying their pipelines to perspective images on cube faces. The resultant reconstructed videos are then mapped back to ERP frames through post-processing. PaletteNeRF, originally designed for perspective images with a limited field of view (FoV), is incompatible with omnidirectional data without modifications. To adapt it, we calculate ray directions using spherical coordinates for each ERP frame, ensuring full coverage of the 360° horizontal and 180° vertical FoV. While CubeMap projection can align with PaletteNeRF's default settings, it introduces inconsistencies in recoloring, especially in dynamic

regions and at the seams between faces. Since PaletteNeRF is built on the Instant-NGP backbone [64], we adjust the world coordinates from the NeRF convention (X-right, Y-forward, Z-up) to the format expected by Instant-NGP (X-right, Y-up, Z-forward), following their official implementation. These adjustments based on ERP inputs ensure spatial consistency and fairness in recoloring comparisons with PaletteNeRF.

B. Omnidirectional Video Reconstruction

As shown in Fig. 4, Our OmniPlane exhibits superior performance in the qualitative evaluation of real-world omnidirectional video reconstruction. It captures finer details of dynamic objects and significantly reduces artifacts across the scene, outperforming EgoNeRF [5]. Compared to HexPlane [18] and 4D-GS [19], OmniPlane demonstrates more consistent performance and maintains more high-frequency details in dynamic omnidirectional scenes. From the results, we can see that although HexPlane demonstrates good performance on most dynamic objects due to their tailored dynamic designs, it often struggles with maintaining high-frequency details in regions with complex textures throughout the video, such as trees, letters, or thin lines on ceilings. Additionally, it tends to produce blurred content in distant areas, particularly in outdoor scenes. In comparison, our OmniPlane reconstructs videos with sharper details at both near and far distances, effectively handling both static and dynamic regions. Moreover, HexPlane requires training six cube faces per scene, thereby taking longer than our approach to achieve the results presented here. 4D-GS often exhibits floaters in dynamic regions due to imperfect depth estimation, along with seam artifacts, as it omits Gaussians near the edges of the view frustum to maintain a 99% confidence interval. This exclusion leads to visible artifacts near-cube seams in ERP reconstructions. In contrast, Our OmniPlane, specifically optimized for spherical inputs, avoids these issues and consistently delivers superior reconstructions of omnidirectional videos.

The quantitative assessment, presented in Table I and II, provides average scores for each metric across omnidirectional videos captured in indoor and outdoor dynamic scenes, respectively. These results underscore our superior video reconstruction performance compared to baseline methods, suggesting our better representation capacity for dynamic scenes in omnidirectional videos. The performance of our method in indoor scenes is usually greater than in outdoor scenes. This arises because outdoor scenes often feature extensive static regions with smooth, low-frequency textures, such as the sky and ground, which are inherently easier to reconstruct and yield smaller space to improve than indoor scenes.

C. Omnidirectional Video Recoloring

In Fig. 5, we present a qualitative comparison of recoloring for real-world omnidirectional videos. PaletteNeRF's recolored results exhibit noticeable artifacts, primarily due to its use of Instant-NGP as the backbone. Instant-NGP performs well on static scenes captured with perspective frames, where rays are sampled on regular 2D grids. However, ERP video frames have non-uniform spatiotemporal distributions, leading

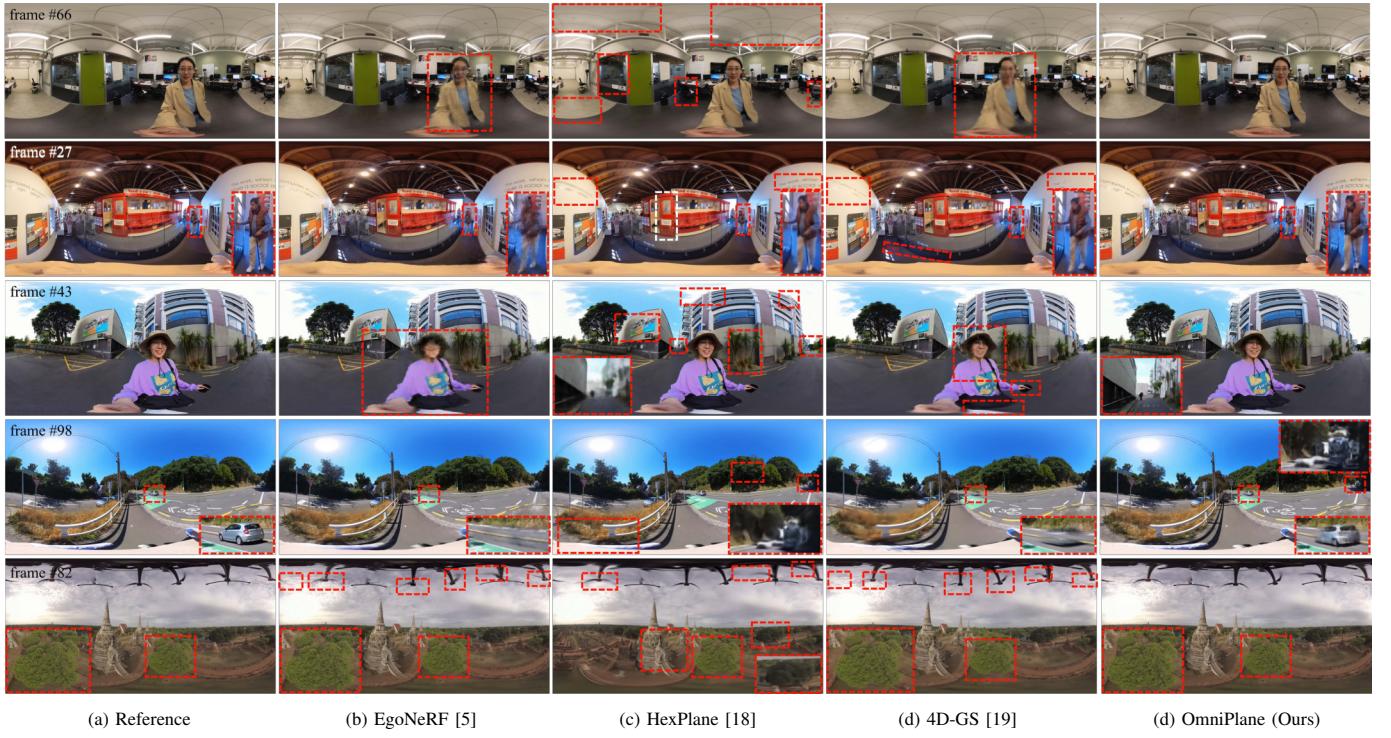


Fig. 4. Qualitative comparisons of reconstruction results on real-world omnidirectional videos in our ‘DyOmni’ dataset.

TABLE I

QUANTITATIVE COMPARISONS OF REAL-WORLD OMNIDIRECTIONAL VIDEO RECONSTRUCTION RESULTS IN EIGHT INDOOR DYNAMIC SCENES.

Method	PSNR ↑	SSIM ↑	LPIPS-a ↓	LPIPS-v ↓	WS-PSNR ↑	WS-SSIM ↑	CPP-PSNR ↑	Cube-SSIM ↑	Cube-LPIPS-a ↓	Cube-LPIPS-v
EgoNeRF [5]	28.58	0.879	0.148	0.243	30.04	0.874	29.01	0.893	0.131	0.228
HexPlane [18]	29.96	0.905	0.114	0.192	31.35	0.897	30.74	0.919	0.095	0.165
4D-GS [19]	29.32	0.897	0.132	0.212	31.09	0.896	30.23	0.923	0.097	0.180
OmniPlane	31.94	0.920	0.100	0.196	33.46	0.916	32.83	0.931	0.084	0.177

TABLE II

QUANTITATIVE COMPARISONS OF REAL-WORLD OMNIDIRECTIONAL VIDEO RECONSTRUCTION RESULTS IN EIGHT OUTDOOR DYNAMIC SCENES.

Method	PSNR ↑	SSIM ↑	LPIPS-a ↓	LPIPS-v ↓	WS-PSNR ↑	WS-SSIM ↑	CPP-PSNR ↑	Cube-SSIM ↑	Cube-LPIPS-a ↓	Cube-LPIPS-v
EgoNeRF [5]	27.65	0.847	0.160	0.234	29.37	0.838	28.55	0.853	0.148	0.228
HexPlane [18]	27.88	0.827	0.203	0.260	29.65	0.810	29.12	0.829	0.193	0.257
4D-GS [19]	26.44	0.816	0.204	0.264	28.69	0.805	27.95	0.864	0.159	0.232
OmniPlane	29.87	0.866	0.147	0.218	31.53	0.858	31.21	0.864	0.145	0.217

to mismatches in Instant-NGP’s hash grid encoding and interpolation. Moreover, the absence of temporal adjustments exacerbates these issues over time, especially in dynamic regions. In contrast, our OmniPlane significantly reduces these artifacts, delivering more natural results and providing a palette that more accurately represents the desired recoloring. Unlike the video recoloring method for perspective videos developed by Du et al. (4D Geometric Palette) [63], OmniPlane requires fewer base colors in the palette, enhancing user-friendliness during editing. Additionally, our model effectively prevents color leaking, ensuring precise recoloring without affecting areas outside the regions of target objects. When compared with EgoNeRF [5] coupled with the color decomposition module, our approach offers more accurate recoloring results and superior performance in dynamic regions. For instance, in the first row, our results demonstrate comprehensive recoloring across various green elements in the scene, including the doors, the light green wall, and the small bag on the distant table. In the second row, our result exhibits consistent

recoloring from red to yellow and meanwhile yields a more realistic outcome that respects the original video. For example, the recolored regions on the ceiling preserve their brightness, ensuring overall visual coherence. In the third row, our model seamlessly transforms the blue light on the road into red, maintaining natural edits along the cloud edges by applying a sheer red without any original blue residual. The fourth row showcases a more natural green reflection on the glasses of our results. In the fifth row, our result shows precise recoloring of green areas, such as the nuanced gradient effect on a distant tree, without color leakage to adjacent non-green zones.

User Study. To assess OmniPlane’s effectiveness, we conducted a user study with twenty participants, including thirteen with prior VR experience and seven novices. We tested five indoor and five outdoor scenes from our dataset. In each scene, we manually adjusted a base color in the palette to a target color, such as converting blue to red. To ensure an immersive VR experience, participants viewed original and recolored videos using an HMD. The study was divided into



Fig. 5. Qualitative comparisons of recolored results on real-world omnidirectional videos in our developed ‘DyOmni’ dataset.

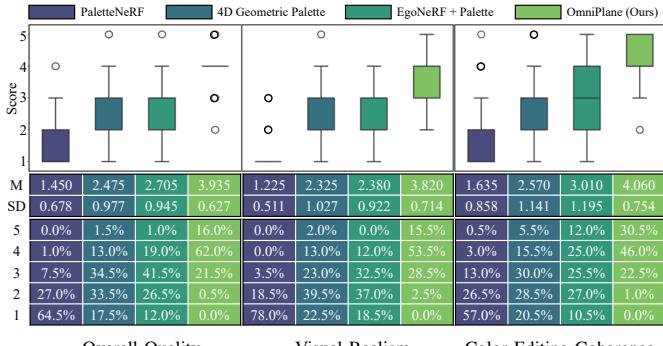


Fig. 6. Box plots of user study scores (1–5) from 20 users across 10 scenes, per method, for overall quality, visual realism, and color editing coherence, along with their mean (M), standard deviation (SD), and proportions of scores.

ten sessions per participant. Each session started with the original video, followed by recolored versions processed by three baseline methods and our model, presented in random order to avoid bias. Participants rated each recolored video

on three dimensions: overall quality, visual realism, and color editing coherence. The ratings used a five-point scale, with 1 for very poor and 5 for exceptional performance. Each participant completed the study individually, without access to other participants’ score sheets, ensuring unbiased evaluations. Specifically, overall quality encompasses the video’s naturalness, plausibility, and the viewer’s comfort while watching. Visual realism pertains to how realistic the scene and dynamic objects appear, with attention to the presence or absence of blurriness and visual artifacts, while disregarding color changes. Color editing coherence evaluates the spatiotemporal consistency and harmony of the recoloring. Fig. 6 and Table III present the user study results across the three evaluation dimensions. Our method achieves the highest mean scores, with most ratings concentrated in the 3–5 range, no scores of 1, and very few scores of 2, indicating a strong user preference. In contrast, other methods exhibit distributions skewed toward lower scores (1–3) with greater variability.

TABLE III
T-TEST ANALYSIS OF COMPARATIVE USER STUDY RESULTS.

Comparison	Overall Quality t-Value ↑ p-Value ↓	Visual Realism t-Value ↑ p-Value ↓	Color Editing Coherence t-Value ↑ p-Value ↓
OmniPlane (Ours) vs PaletteNeRF	38.061 2.631E-134	41.300 9.767E-139	30.024 1.266E-103
OmniPlane (Ours) vs 4D Geometric Palette	17.791 1.766E-50	16.902 1.944E-47	15.406 4.021E-41
OmniPlane (Ours) vs EgoNeRF+Palette	15.344 6.828E-41	17.463 2.197E-50	10.510 1.587E-22



Fig. 7. A recolored result of a single omnidirectional image selected from our dataset, using a 360° edit propagation method [50] compared to ours.

TABLE IV
QUANTITATIVE RESULTS OF ABLATIONS ON VARIOUS FEATURE GRIDS BASED ON DIFFERENT SPHERICAL COORDINATE REPRESENTATIONS.

Ablation	Indoor Dynamic Scenes					Outdoor Dynamic Scenes				
	WS-PSNR ↑	WS-SSIM ↑	CPP-PSNR ↑	Cube-SSIM ↑	Cube-LPIPS-v ↓	WS-PSNR ↑	WS-SSIM ↑	CPP-PSNR ↑	Cube-SSIM ↑	Cube-LPIPS-v ↓
CPP	31.64	0.887	31.58	0.907	0.232	30.28	0.827	30.32	0.840	0.262
Yin-Yang	32.67	0.904	31.97	0.922	0.198	31.29	0.851	30.95	0.859	0.225
Pure ERP	32.98	0.909	32.32	0.926	0.188	31.18	0.848	30.84	0.857	0.226
OmniPlane	33.46	0.916	32.83	0.931	0.177	31.53	0.858	31.21	0.864	0.217

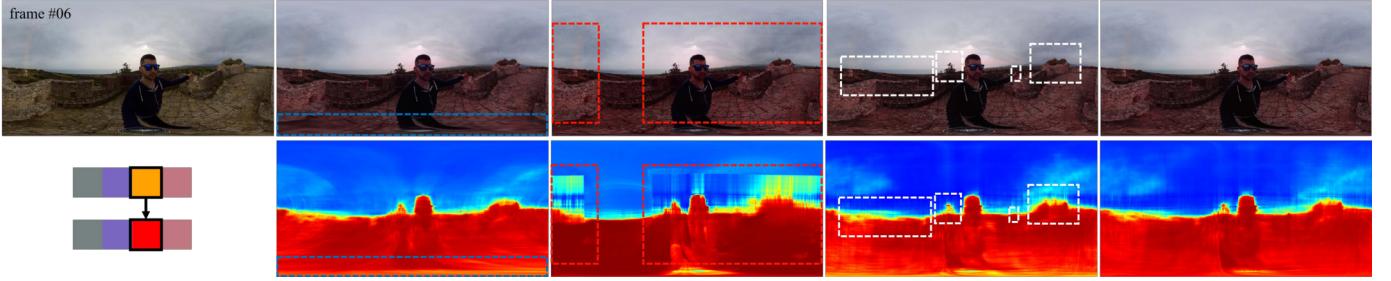


Fig. 8. Qualitative results of an ablation study on diverse spherical feature grids. The left column displays a selected original video frame alongside the color palette used for recoloring. The right four columns exhibit the recolored frames and corresponding depth maps generated using different feature grids.

Pairwise t-tests in Table III highlight our method's advantages, showing significantly high t-values and low p-values. These results highlight our effectiveness in handling dynamic content and achieving natural recoloring in real-world 360° videos.

Comparison with a 360° Image Edit Propagation Method. We explored the feasibility of applying a 360° image edit propagation method [50] to recolor 360° videos. Our implementation revealed that it was unsuitable for 360° video sequences, primarily due to its inability to preserve temporal coherence for dynamic content. A recoloring comparison of a single 360° image is illustrated in Fig. 7. Their recolored result reveals color leakages (e.g., on the cloth and trees) and inaccurate color changes (e.g., on the billboard). In contrast, our recolored image avoids color leakages and achieves natural changes, with the reds accurately represented rather than appearing as magenta in their results. Due to its inability to generate plausible visual quality in video recoloring, we opted not to include the results of their method in our user study.

D. Ablation Studies

Our OmniPlane is dedicated to achieving the dual objectives of dynamic scene representation and recolorable capacity

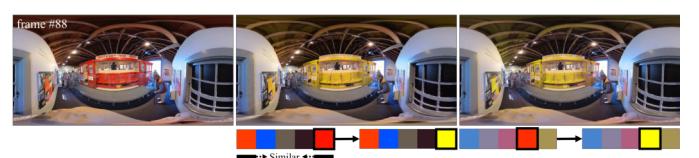


Fig. 9. Ablation study results on w/o and w hue regularization \mathcal{L}_{hue} .

within omnidirectional videos. We conducted ablation studies to evaluate its effectiveness, exploring diverse feature grids using various spherical coordinate representations, and assessing the impact of removing our novel regularization term \mathcal{L}_{hue} .

Various Feature Grids. Given the importance of precise appearance reconstruction in representations, we explored alternative strategies to spherical coordinates. Specifically, we replace spherical coordinates coupled with our weighted sampling strategy by using Yin-Yang coordinates as detailed in [5]. Additionally, we investigate the use of resampled spherical coordinates on the Craster Parabolic Projection (CPP) plane and pure spherical coordinates on the ERP. Table IV presents quantitative ablation results, highlighting the superior performance of our designed spherical grid in conjunction with our tailored sampling strategy. Fig. 8 illustrates the qualitative

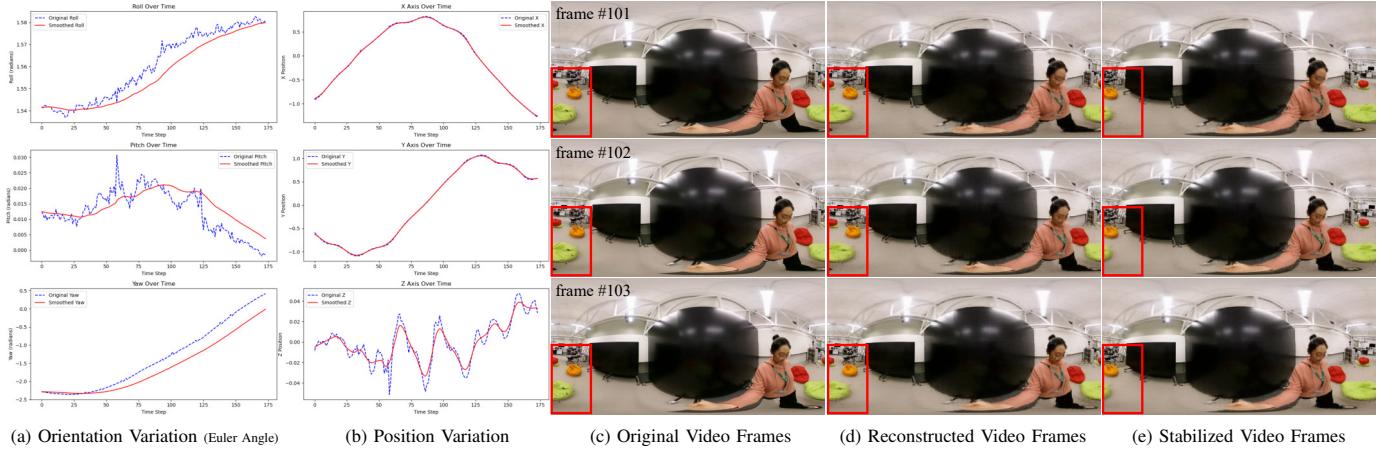


Fig. 10. Results of omnidirectional video stabilization using our OmniPlane. The first two columns show comparison plots of camera orientations (Euler angles) and positions before and after stabilization. The remaining columns display the original, reconstructed, and stabilized video frames, respectively.

outcomes of these ablation studies, revealing evident artifacts and floaters in the results of the CPP-based grid, along with discontinuous depth and appearance details in the Yin-Yang grid results. The CPP-based grid exhibits artifacts due to resampling, while the Yin-Yang grid presents discontinuity issues between its two balanced spherical parts. Moreover, the ERP projection without weighted sampling oversamples near the poles, leading to underperformance near the equator.

Novel Regularization on the Hue Channel. Fig. 9 presents the qualitative ablation results for a scene named ‘Shop’, comparing scenarios with and without the designed hue regularization \mathcal{L}_{hue} . The omission of hue regularization \mathcal{L}_{hue} leads to overly similar base colors in the learned color palette, such as dark orange and red. This similarity can diminish the palette’s usefulness to users by reducing its representativeness, ultimately impacting the effectiveness of recoloring applications.

E. Discussions and Limitations

Omnidirectional Video Stabilization Application. While the primary focus of our OmniPlane is recoloring omnidirectional videos, it can also be extended to other tasks. One natural extension is omnidirectional video stabilization, thanks to its ability to represent dynamic scenes captured by omnidirectional videos. Once the OmniPlane is trained, it generates a dynamic scene capable of rendering novel omnidirectional frames from camera poses beyond those captured in the input frames. Stabilization is achieved by smoothing the camera poses and rendering novel frames from the smoothed poses. We smooth the camera positions using a standard 1D Gaussian filter with a kernel size of 13 and a standard deviation $\epsilon = 9.0$, empirically chosen to balance jitter reduction and natural camera motion preservation. For smoothing camera orientations, we use spherical linear interpolation (Slerp) between quaternions to maintain valid rotational consistency, ensuring orthogonality. We further refine transitions between frames by combining Slerp with an exponential moving average (EMA). The interpolation between two quaternions is defined as:

$$S(\alpha, m_i, m_{i+1}) = \frac{\sin((1-\alpha)\psi)}{\sin(\psi)}m_i + \frac{\sin(\alpha\psi)}{\sin(\psi)}m_{i+1}, \quad (11)$$

where $\psi = \cos^{-1}(m_i \cdot m_{i+1})$ is the angle between the two quaternions m_i and m_{i+1} , and $\alpha \in [0, 1]$ is the interpolation factor controlling the weight distribution between the quaternions. In our experiments, we set α to 0.05. A stabilization result is shown in Fig. 10, where the camera path is effectively smoothed. As indicated by the red boxes, the sofas in the original frames shift noticeably left and right between consecutive frames, while in the stabilized frames, they remain nearly stationary, demonstrating the stability achieved by our model. For video results and additional examples, please refer to our supplemental videos.

Discussions on Video Editing Capabilities. Building on the palette-based decomposition approach in OmniPlane, our method enhances 360° video content editing capabilities by leveraging the decomposed components to enable functionalities beyond recoloring. For instance, scaling the view-dependent color component c_{view} and color offsets $\delta_i(\mathbf{x})$ facilitates illumination and texture editing, as shown in Fig. 11 and 12. While these editing capabilities are similar to those of PaletteNeRF [21], OmniPlane uniquely operates on 4D spherical spatiotemporal positions $\mathbf{x} = (\theta, \phi, r, t)$, enabling dynamic scene editing in 360° videos. In contrast, PaletteNeRF is designed for static scenes using 3D Cartesian spatial positions (x, y, z) in typical perspective images. Fig. 11 depicts how scaling c_{view} alters illumination effects, primarily influencing the brightness of highlights and the intensity of gloss reflections. Under different scaling factors, the most noticeable variations appear in high-illumination regions, such as the sunlight surrounding the person. Fig. 12 illustrates how scaling $\delta_i(\mathbf{x})$ affects color textures. Higher scaling factors result in more pronounced changes, mainly in saturation and hue, which are particularly noticeable in examples such as the floor surface. Additionally, OmniPlane can naturally support color-based segmentation for 360° videos through the learned palette blending weights $\omega_i(\mathbf{x})$. Fig. 13 demonstrates soft color segmentation derived directly from $\omega_i(\mathbf{x})$, as well as threshold-filtered hard segmentation, which is obtained by isolating regions where $\omega_i(\mathbf{x})$ for specific palette colors dominate and exceed a threshold (empirically set to 0.5). This segmentation approach can enable localized content editing. In addition, as shown in Fig. 14, OmniPlane facilitates color transfer by



Fig. 11. Results of illumination effect editing by scaling the learned view-dependent color c_{view} . The leftmost column shows the original frame, while the other four columns present the edited frames with c_{view} scaled by 0, 1, 3, and 6, respectively.



Fig. 12. Results of color texture editing by scaling the learned color offsets $\delta_i(\mathbf{x})$. The leftmost column shows the original frame, while the other four columns present the edited frames with $\delta_i(\mathbf{x})$ scaled by 0, 1, 3, and 6, respectively.

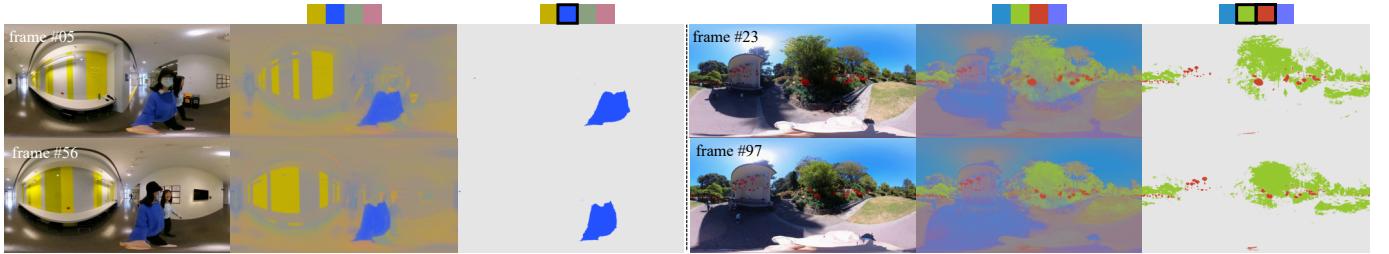


Fig. 13. Results of 360° video segmentation derived from palette blending weights $\omega_i(\mathbf{x})$. There are two instances: an indoor scene (left) and an outdoor scene (right). Each consists of three columns—original frames (left), soft color segmentation across all palette colors (middle), and threshold-filtered hard segmentation highlighting regions dominated by specified colors—blue for the indoor scene and red/green for the outdoor scene (right).



Fig. 14. Results of 360° video color transfer achieved by applying the learned palette from the target video to the reference video.

replacing the palette of a reference video with that of a target video, producing visually coherent and plausible results.

Limitations and Future Work. While OmniPlane enables diverse content editing capabilities for 360° videos through palette-based decomposition, certain limitations remain. Semantic feature-guided editing, while not directly supported, could be achieved by incorporating pre-extracted semantic features from the given 360° video as supervision. This might involve training a semantic feature encoder with OmniPlane's density features to learn a weight map for localized edits. Similarly, although OmniPlane allows direct palette-based color transfer, style transfer requires additional optimization. This could include introducing learnable transformations for decomposed components such as the palette \mathcal{P}_i , intensity $I(\mathbf{x})$, and color offsets $\delta_i(\mathbf{x})$, enabling more variations beyond global replacement or scaling and facilitating stylized edits through iterative fine-tuning. Lastly, OmniPlane's current capabilities are limited to palette-based editing and segmentation, lacking support for physically-based editing (e.g., separating diffuse reflectance and shading components as in intrinsic decomposition [65], [66]) or semantic segmentation.

V. CONCLUSION

In this paper, we propose OmniPlane, a novel method for representing and recoloring dynamic scenes in casual cap-

tured real-world omnidirectional videos. Utilizing a spherical feature grid combined with a soft-weighted sampling technique, OmniPlane effectively calculates omnidirectional density and appearance features across spacetime spherical points via strategic sampling and fusion processes. This innovative method not only ensures high-quality rendering of diverse indoor and outdoor dynamic environments in 360° videos but also supports palette-based user-centric content editing in real-world omnidirectional scenes through integration with color decomposition. Our experiments validate the effectiveness of our framework in omnidirectional video representation and editing operations, confirming its potential as a versatile and practical tool in the realm of immersive media.

ACKNOWLEDGMENTS

This work was supported by Marsden Fund Council managed by the Royal Society of New Zealand under Grant MFP-20-VUW-180 and the grant of Catalyst: Leaders - Julius von Haast Fellowship (23-VUW-019-JVH). Thanks to Zheng-Jun Du and Yun Zhang for providing recolored video frames using our dataset with their methods [63] and [50].

REFERENCES

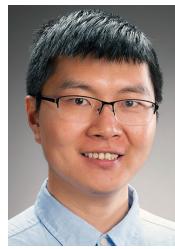
- [1] T. L. da Silveira, P. G. Pinto, J. Murrugarra-Llerena, and C. R. Jung, “3d scene geometry estimation from 360 imagery: A survey,” *ACM Computing Surveys*, vol. 55, no. 4, pp. 1–39, 2022.

- [2] F. Zhang, J. Zhao, Y. Zhang, and S. Zollmann, "A survey on 360 images and videos in mixed reality: Algorithms and applications," *Journal of Computer Science and Technology*, vol. 38, no. 3, pp. 473–491, 2023.
- [3] H. Jang, A. Meuleman, D. Kang, D. Kim, C. Richardt, and M. H. Kim, "Egocentric scene reconstruction from an omnidirectional video," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 4, pp. 1–12, 2022.
- [4] H. Huang, Y. Chen, T. Zhang, and S.-K. Yeung, "Real-time omnidirectional roaming in large scale indoor scenes," in *SIGGRAPH Asia 2022 Technical Communications*, 2022, pp. 1–5.
- [5] C. Choi, S. M. Kim, and Y. M. Kim, "Balanced spherical grid for egocentric view synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 16590–16599.
- [6] J. Bai, L. Huang, J. Guo, W. Gong, Y. Li, and Y. Guo, "360-gs: Layout-guided panoramic gaussian splatting for indoor roaming," *arXiv preprint arXiv:2402.00763*, 2024.
- [7] L. Li, H. Huang, S.-K. Yeung, and H. Cheng, "Omnigs: Omnidirectional gaussian splatting for fast radiance field reconstruction using omnidirectional images," *arXiv preprint arXiv:2404.03202*, 2024.
- [8] 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," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.
- [9] B. Kerbl, G. Kopanas, T. Leimkuehler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics (TOG)*, vol. 42, no. 4, pp. 1–14, 2023.
- [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. 10318–10327.
- [11] K. Park, U. Sinha, P. Hedman, J. T. Barron, S. Bouaziz, D. B. Goldman, R. Martin-Brualla, and S. M. Seitz, "Hypernerf: a higher-dimensional representation for topologically varying neural radiance fields," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 6, pp. 1–12, 2021.
- [12] L. Song, A. Chen, Z. Li, Z. Chen, L. Chen, J. Yuan, Y. Xu, and A. Geiger, "Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 5, pp. 2732–2742, 2023.
- [13] Z. Li, S. Niklaus, N. Snavely, and O. Wang, "Neural scene flow fields for space-time view synthesis of dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6498–6508.
- [14] Y. Du, Y. Zhang, H.-X. Yu, J. B. Tenenbaum, and J. Wu, "Neural radiance flow for 4d view synthesis and video processing," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 2021, pp. 14304–14314.
- [15] L. Liu, J. Gu, K. Zaw Lin, T.-S. Chua, and C. Theobalt, "Neural sparse voxel fields," *Advances in Neural Information Processing Systems*, vol. 33, pp. 15651–15663, 2020.
- [16] W. Gan, H. Xu, Y. Huang, S. Chen, and N. Yokoya, "V4d: Voxel flow for 4d novel view synthesis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 30, no. 2, pp. 1579–1591, 2023.
- [17] 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*, 2023, pp. 12479–12488.
- [18] A. Cao and J. Johnson, "Hexplane: A fast representation for dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 130–141.
- [19] G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and X. Wang, "4d gaussian splatting for real-time dynamic scene rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20310–20320.
- [20] F.-L. Liu, S.-Y. Chen, Y. Lai, C. Li, Y.-R. Jiang, H. Fu, and L. Gao, "Deepfacevideoediting: sketch-based deep editing of face videos," *ACM Transactions on Graphics*, vol. 41, no. 4, p. 167, 2022.
- [21] Z. Kuang, F. Luan, S. Bi, Z. Shu, G. Wetzstein, and K. Sunkavalli, "Palettenerf: Palette-based appearance editing of neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 20691–20700.
- [22] Y.-J. Yuan, X. Han, Y. He, F.-L. Zhang, and L. Gao, "Munerf: Robust makeup transfer in neural radiance fields," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–12, 2024.
- [23] J. Tan, J. Echevarria, and Y. Gingold, "Efficient palette-based decomposition and recoloring of images via rgbspace geometry," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 6, pp. 1–10, 2018.
- [24] H. Chen, B. He, H. Wang, Y. Ren, S. N. Lim, and A. Shrivastava, "Nerv: Neural representations for videos," *Advances in Neural Information Processing Systems*, vol. 34, pp. 21557–21568, 2021.
- [25] H. Chen, M. Gwilliam, S.-N. Lim, and A. Shrivastava, "Hnerf: A hybrid neural representation for videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 10270–10279.
- [26] V. Ye, Z. Li, R. Tucker, A. Kanazawa, and N. Snavely, "Deformable sprites for unsupervised video decomposition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2657–2666.
- [27] Y. Kasten, D. Ofri, O. Wang, and T. Dekel, "Layered neural atlases for consistent video editing," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 6, pp. 1–12, 2021.
- [28] S. Loeschcke, S. Belongie, and S. Benaim, "Text-driven stylization of video objects," in *European Conference on Computer Vision*. Springer, 2022, pp. 594–609.
- [29] O. Bar-Tal, D. Ofri-Amar, R. Fridman, Y. Kasten, and T. Dekel, "Text2live: Text-driven layered image and video editing," in *European conference on computer vision*. Springer, 2022, pp. 707–723.
- [30] C. Lei, X. Ren, Z. Zhang, and Q. Chen, "Blind video deflickering by neural filtering with a flawed atlas," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 10439–10448.
- [31] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, "Tensorf: Tensorial radiance fields," in *European Conference on Computer Vision*. Springer, 2022, pp. 333–350.
- [32] M. Wu, Z. Wang, G. Kouros, and T. Tuytelaars, "Tetirf: Temporal tri-plane radiance fields for efficient free-viewpoint video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 6487–6496.
- [33] J. Luiten, G. Kopanas, B. Leibe, and D. Ramanan, "Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis," in *2024 International Conference on 3D Vision (3DV)*. IEEE, 2024, pp. 800–809.
- [34] Y.-J. Yuan, Y.-T. Sun, Y.-K. Lai, Y. Ma, R. Jia, and L. Gao, "Nerf-editing: geometry editing of neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18353–18364.
- [35] J.-K. Chen, J. Lyu, and Y.-X. Wang, "Neuraleditor: Editing neural radiance fields via manipulating point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12439–12448.
- [36] C. Zheng, W. Lin, and F. Xu, "Editablererf: Editing topologically varying neural radiance fields by key points," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8317–8327.
- [37] B. Attal, S. Ling, A. Gokaslan, C. Richardt, and J. Tompkin, "Matryodshka: Real-time 6dof video view synthesis using multi-sphere images," in *European Conference on Computer Vision*. Springer, 2020, pp. 441–459.
- [38] M. Broxton, J. Flynn, R. Overbeck, D. Erickson, P. Hedman, M. Duvall, J. Dourgarian, J. Busch, M. Whalen, and P. Debevec, "Immersive light field video with a layered mesh representation," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 86–1, 2020.
- [39] T. Habtegebral, C. Gava, M. Rogge, D. Stricker, and V. Jampani, "Somsi: Spherical novel view synthesis with soft occlusion multi-sphere images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 15725–15734.
- [40] T. Bertel, M. Yuan, R. Lindroos, and C. Richardt, "Omniphotos: casual 360 vr photography," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 6, pp. 1–12, 2020.
- [41] K. Gu, T. Maugey, S. Knorr, and C. Guillemot, "Omni-nerf: neural radiance field from 360 image captures," in *2022 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2022, pp. 1–6.
- [42] T. Otonari, S. Ikehata, and K. Aizawa, "Non-uniform sampling strategies for nerf on 360° images," in *2022 British Machine Vision Conference (BMVC)*. BMVA Press, 2022.
- [43] S. Kulkarni, P. Yin, and S. Scherer, "360fusionnerf: Panoramic neural radiance fields with joint guidance," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 7202–7209.
- [44] Z. Chen, Y.-P. Cao, Y.-C. Guo, C. Wang, Y. Shan, and S.-H. Zhang, "Panogr: Generalizable spherical radiance fields for wide-baseline panoramas," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

- [45] Z. Zhu, R. R. Martin, and S.-M. Hu, "Panorama completion for street views," *Computational Visual Media*, vol. 1, no. 1, pp. 49–57, 2015.
- [46] K. He and J. Sun, "Image completion approaches using the statistics of similar patches," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 12, pp. 2423–2435, 2014.
- [47] B. Xu, S. Pathak, H. Fujii, A. Yamashita, and H. Asama, "Spatiotemporal video completion in spherical image sequences," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2032–2039, 2017.
- [48] J. Kopf, "360° video stabilization," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, pp. 1–9, 2016.
- [49] C. Tang, O. Wang, F. Liu, and P. Tan, "Joint stabilization and direction of 360° videos," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 2, pp. 1–13, 2019.
- [50] Y. Zhang, F.-L. Zhang, Y.-K. Lai, and Z. Zhu, "Efficient propagation of sparse edits on 360° panoramas," *Computers & Graphics*, vol. 96, pp. 61–70, 2021.
- [51] Y.-J. Li, J. Shi, F.-L. Zhang, and M. Wang, "Bullet comments for 360° video," in *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2022, pp. 1–10.
- [52] K.-M. Wong, "View-adaptive asymmetric image detail enhancement for 360-degree stereoscopic vr content," in *2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 2022, pp. 23–26.
- [53] A. Vermast and W. Hürst, "Introducing 3d thumbnails to access 360-degree videos in virtual reality," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 5, pp. 2547–2556, 2023.
- [54] K. Huang, F.-L. Zhang, J. Zhao, Y. Li, and N. Dodgson, "360° stereo image composition with depth adaption," *IEEE Transactions on Visualization and Computer Graphics*, 2023.
- [55] R.-K. Xu, L. Zhang, and F.-L. Zhang, "Intrinsic omnidirectional image decomposition with illumination pre-extraction," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 7, pp. 4416–4428, 2024.
- [56] H. Ai, Z. Cao, H. Lu, C. Chen, J. Ma, P. Zhou, T.-K. Kim, P. Hui, and L. Wang, "Dream360: Diverse and immersive outdoor virtual scene creation via transformer-based 360° image outpainting," *IEEE Transactions on Visualization and Computer Graphics*, vol. 30, no. 5, pp. 2734–2744, 2024.
- [57] P. Moulon, P. Monasse, R. Perrot, and R. Marlet, "Openmvg: Open multiple view geometry," in *Reproducible Research in Pattern Recognition: First International Workshop, RRRP 2016, Cancún, Mexico, December 4, 2016, Revised Selected Papers 1*. Springer, 2017, pp. 60–74.
- [58] Y. Sun, A. Lu, and L. Yu, "Weighted-to-spherically-uniform quality evaluation for omnidirectional video," *IEEE signal processing letters*, vol. 24, no. 9, pp. 1408–1412, 2017.
- [59] Y. Zhou, M. Yu, H. Ma, H. Shao, and G. Jiang, "Weighted-to-spherically-uniform ssim objective quality evaluation for panoramic video," in *2018 14th IEEE International Conference on Signal Processing (ICSP)*. IEEE, 2018, pp. 54–57.
- [60] V. Zakharchenko, K. P. Choi, and J. H. Park, "Quality metric for spherical panoramic video," in *Optics and Photonics for Information Processing X*, vol. 9970. SPIE, 2016, pp. 57–65.
- [61] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.
- [62] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations (ICLR 2015)*. Computational and Biological Learning Society, 2015.
- [63] Z.-J. Du, K.-X. Lei, K. Xu, J. Tan, and Y. Gingold, "Video recoloring via spatial-temporal geometric palettes," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, pp. 1–16, 2021.
- [64] 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.
- [65] G. Ye, E. Garces, Y. Liu, Q. Dai, and D. Gutierrez, "Intrinsic video and applications," *ACM Transactions on Graphics (ToG)*, vol. 33, no. 4, pp. 1–11, 2014.
- [66] E. Garces, J. I. Echevarria, W. Zhang, H. Wu, K. Zhou, and D. Gutierrez, "Intrinsic light field images," in *Computer Graphics Forum*, vol. 36, no. 8. Wiley Online Library, 2017, pp. 589–599.



Simin Kou received her Bachelor's and Master's degrees from Hangzhou Dianzi University, China, in 2019 and 2022, respectively. She is currently working toward a Doctoral degree with the Computer Graphics Research Group at the School of Engineering and Computer Science, Victoria University of Wellington, New Zealand. Her research interests include neural representation, image and video editing, mixed reality, and computer graphics.



Fang-Lue Zhang is currently a senior lecturer with Victoria University of Wellington, New Zealand. He received a Bachelor's degree from Zhejiang University, Hangzhou, China, in 2009, and a Doctoral degree from Tsinghua University, Beijing, China, in 2015. His research interests include image and video editing, mixed reality, and image-based graphics. He is a member of IEEE and ACM. He received the Victoria Early Career Research Excellence Award in 2019. He is on the editorial board of Computer & Graphics. He is a committee member of the IEEE Pacific Graphics 2020 and 2021 and Computational Visual Media 2024.



Jakob Nazarenus received his Bachelor's and Master's degrees from Kiel University, Germany, in 2020 and 2021, respectively. He is currently working toward a Doctoral degree with the Multimedia Information Processing Group at the Department of Computer Science, Kiel University, Germany. His research interests mainly focus on deep learning techniques for object segmentation, detection, reconstruction, and image generation.



Reinhard Koch received his Doctoral degree in Electrical Engineering from the University of Hanover, Germany, in 1996. He then worked as a post-doctoral researcher and leader of the Research Team "Computer Vision and Robotics" in the Group of Prof. Luc van Gool, K.U. Leuven, Belgium, from 1996 to 1999. Since 1999 he has been a professor of Computer Science at Kiel University, Germany, working on Multimedia Information Processing. His research field covers image-based scene reconstruction, 3D-scene capture and reconstruction, light-field and multi-view processing, 3D-TV systems, and underwater computer vision. He has co-authored over 240 papers in peer-reviewed conferences, journals, and book chapters. He received numerous scientific awards, among them the Olympus Award of the DAGM in 1996, and the David-Marr Award of the ICCV in 1998 together with collaborators from K.U.Leuven and University of Oxford. In 2022 he was elected fellow of the International Association on Pattern Recognition (IAPR).



Neil A. Dodgson has a PhD in image processing (1992) and a Doctor of Science degree (2007), both from the University of Cambridge. He worked at the University of Cambridge Computer Laboratory from 1995, where he rose to the position of Professor of Graphics & Imaging. In 2016, he moved to Victoria University of Wellington to head their computer graphics programme. He is currently Professor of Computer Graphics and Dean of Graduate Research. His 150 research publications are in 3D TV, 3D curves and surfaces, and aesthetic imaging. He is a Fellow of Engineering New Zealand (FEngNZ), of the Institution of Engineering & Technology (FIET), and of the Institution for Mathematics & its Applications (FIMA).