Tiled Multiplane Images for Practical 3D Photography

Numair Khan
numairkhan@meta.com

Lei Xiao
lei.xiao@meta.com

Douglas Lanman
doug.lanman@meta.com

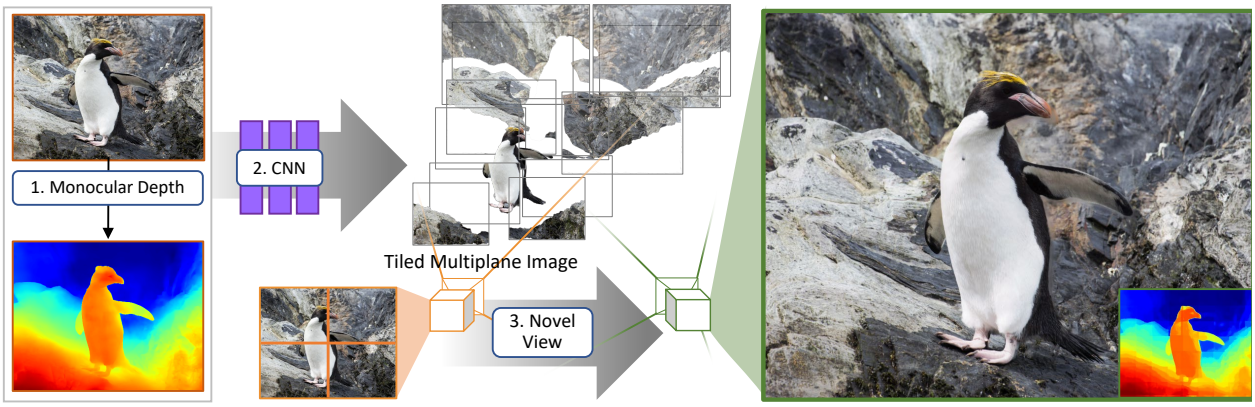

Figure 1: An overview of our 3D photography method. **Left:** Using a single RGB image and an estimated monocular depth map, the scene is recreated as a tiled grid of many small MPIs. **Middle:** A visualization of a 2×2 tiled grid of MPIs, each with three RGBA layers. **Right:** Novel views are rendered by warping and compositing each MPI tile into the target camera's frustum. The 768×1152 pixel image shown is generated using a 7×11 grid of 4-layer MPIs.

Abstract

The task of synthesizing novel views from a single image has useful applications in virtual reality and mobile computing, and a number of approaches to the problem have been proposed in recent years. A Multiplane Image (MPI) estimates the scene as a stack of RGBA layers, and can model complex appearance effects, anti-alias depth errors and synthesize soft edges better than methods that use textured meshes or layered depth images. And unlike neural radiance fields, an MPI can be efficiently rendered on graphics hardware. However, MPIs are highly redundant and require a large number of depth layers to achieve plausible results. Based on the observation that the depth complexity in local image regions is lower than that over the entire image, we split an MPI into many small, tiled regions, each with only a few depth planes. We call this representation a Tiled Multiplane Image *(TMPI). We propose a method for generating a TMPI with adaptive depth planes for single-view 3D photography in the wild. Our synthesized results are comparable to state-of-the-art single-view MPI methods while having lower computational overhead.*

1. Introduction

The novel view synthesis (NVS) problem involves using a set of input images to generate views from new and unseen camera positions, allowing three dimensional interaction with photos. This is a long-studied problem, with early work relying on interpolation within dense structured image sets [20, 11, 8]. The specialized rigs commonly required to capture the large number of images restricted these methods to lab settings [46]. However, the potential applications offered by novel view synthesis on modern mobile and VR devices has kindled wide interest in the problem, and encouraged researchers to seek methods that make the technology more accessible. The term *3D photography* refers to the use of novel view synthesis in everyday capture settings, often from a single image.

Over the past few years a number of proposed scene representations have leveraged the great strides being made in learning-based techniques to achieve more accurate synthesis with fewer constraints. The most recent of these are neural radiance fields (NeRFs) [26, 50] which represent the scene as multi-layer perceptrons. Their results define the high bar of novel view synthesis. However, this high quality has a significant data and computational cost.

An alternate representation, a multiplane image (MPI), defines the scene as a stack of fronto-parallel RGBA planes that can be warped and rendered into novel viewpoints [53, 26, 9]. An MPI offers the advantage of rendering speed, and suffers from less aliasing than mesh or point-based methods [38, 47, 29]. The latter characteristic is important for applications that require temporal stability. However, an MPI is a highly redundant scene representation: the number of RGBA planes required to capture and reconstruct all the depth variation in a scene can be quite high. Since most scenes have a larger amount of free space than occupied, most of the planes in an MPI are very sparse. This makes them inefficient as a representation [2], and expensive to generate, transmit, and store.

In this paper we propose to address these shortcomings of multiplane images and develop a lightweight solution to the 3D photography problem that can be practically implemented on mobile and VR devices. Some examples of the applications we envision are 3D video conferencing, telepresence, and VR passthrough [49]. We show how subdividing the image plane into many small MPIs with only a few planes in each, provides a more efficient representation from a computational and memory perspective. However, the naive approach of directly using existing MPI methods with tiles creates boundary artifacts in the novel views. This happens because the commonly used fixed spacing of MPI planes fails to capture the full depth range of a tile when the number of planes is small. Furthermore, it is sensitive to outliers in small regions. We propose a clustering-based approach using learnt confidence weights to predict per-tile MPI planes that better represents local depth features. Our method is lightweight and generates results comparable to the state-of-the-art in MPI-based 3D photography.

In summary, the main contribution of this work are,

1. The demonstration of *tiled multiplane images* as a practical representation for view synthesis tasks.

2. A learning framework for generating tiled multiplane images from a single RGB input for 3D photography.

3. A novel approach to adaptive MPI plane positioning.

2. Related Work

The progenitors of current 3D photography were the early works on image-based rendering (IBR). These methods usually relied on interpolation within the convex hull of a large set of images to generate novel views. Levoy *et al*. [20] and Gortler *et al*. [11] proposed the canonical two-plane parameterization of light fields that renders novel views by quadrilinear interpolation. Gortler's method also provided an early demonstration of the use of geometric proxies to improve rendering quality. Davis *et al*.'s [8] work extended interpolation-based view synthesis to unstructured images. The excellent analysis of plenoptic sampling done by Chai *et al*. [5] proved, however, that for large distances the number of images required for view synthesis by interpolation was impractically high. Consequently, the large majority of recent view synthesis methods have relied on learnt priors to overcome the high sampling requirements.

One of the corollaries of Chai *et al*.'s analysis was that the sampling requirements for view-interpolation are inversely related to the geometric information of the scene. Thus, many subsequent methods have relied on coarse geometric proxies to improve view synthesis quality [34, 35, 13]. In Mildenhall *et al*.'s [26] method this proxy takes the form of a multiplane image (MPI), which they use to achieve interpolation-based view synthesis that overcomes Chai *et al*.'s sampling limits. MPIs were first proposed by Zhou *et al*. [53] who used them for extrapolating novel views outside the convex hull of the input stereo cameras (Tucker and Snavely [42] observe that an MPI can be considered as an instance of Szeliski and Golland's [40] earlier "stack of acetates" volumetric model). Mildenhall *et al*. [26] and Srinivasan *et al*. [39], respectively, provide a theoretical analysis of the limits of view interpolation and extrapolation using MPIs. Flynn *et al*. [9] use learnt gradient descent to generate MPIs from multiple views. Attal *et al*. [1] and Broxton *et al*. [3] extend the MPI concept to concentric RGBA spheres that can be used for view synthesis in 360 degrees. Wizadwongsa *et al*. [48] and Li *et al*. [22] replace the discrete RGBA planes of MPIs with continuous neural surfaces to achieve higher quality results. Tucker and Snavely [42] use a scale-invariant method that allows them to learn strong data priors that can generate MPIs from a single view. Their approach is in a line of recent work dubbed *3D photography* that aims for novel views from in-the-wild, single-view images. Li *et al*. [23], Han *et al*. [12], and Luvizon *et al*.'s [25] methods are MPI-based examples of this approach. An MPI, however, is an over-parameterized scene representation. Recent methods [24, 10, 25, 12] have sought to overcome this shortcoming to some extent through a more judicious placement of depth planes. Nonetheless, their high level of redundancy dilates their memory and computational footprint, and limits their wider adoption in mobile and AR/VR applications.

For such use cases, a more efficient approach to novel view synthesis involves depth-based warping [6, 49], often followed by inpainting [18, 19]. Shih *et al*. [38] propose to guide inpainting in disoccluded regions of the warped view using a layered depth representation. Li *et al*. [21] apply this general approach to 360-degree input. Wiles *et al*. [47] use a depth map to generate a point cloud of neural features which can be projected and rendered in novel views using a generative network. Choi *et al*. [7] estimate a probability volume instead of a single depth map to handle uncertainty in difficult regions. Niklaus *et al*. [29] use seg-

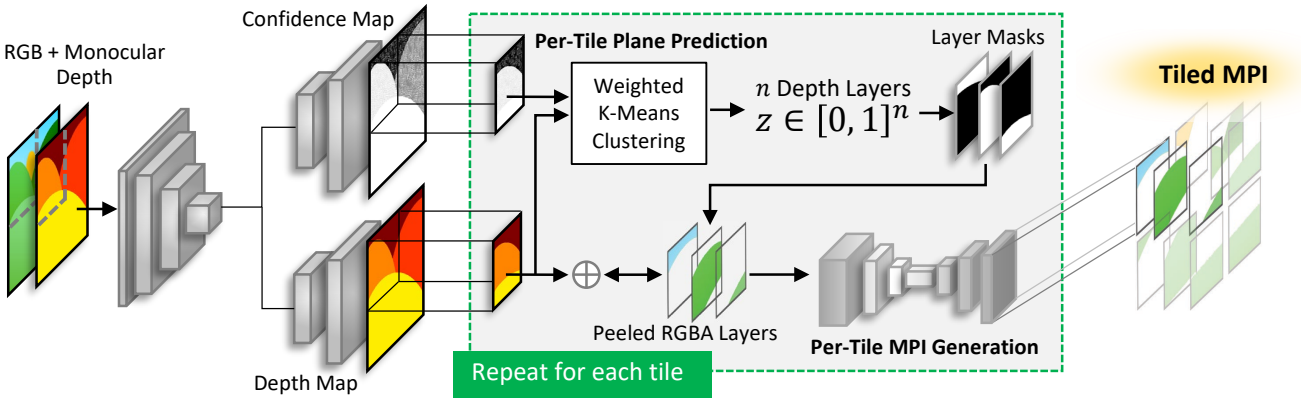

Figure 2: Tiled multiplane image generation for single-view 3D photography. Given an RGB image and depth from a monocular estimator, our method first generates a pixel-wise confidence map and a pre-processed depth map. These are used to predict n depth planes per tile through weighted k-means clustering (with $k = n$). The layer masks defined by pixel labels are used to peel RGBA layers, yielding a rough initial MPI. This is concatenated ($\bigoplus$) to the depth and passed to a refinement network that generates the RGBA images of the final per-tile MPI.

mentation to remove the geometric and semantic distortions from depth that often impair the rendering quality of this approach. Nonetheless, depth-based warping suffers from hard boundaries and is over-sensitive to errors in the depth estimate. While Jampani *et al*. [14] propose soft layering with alpha mattes to address this shortcoming, MPIs are inherently capable of handling such artifacts via blending.

Finally, our review of view synthesis would not be complete without mentioning neural radiance fields (NeRFs) [27, 50] which have recently burgeoned in popularity. While great strides are being made in improving the time and data requirements of NeRFs [51, 28], they remain expensive for interactive applications. Some recent work [33, 41, 43] shows that decomposing a single large radiance field into smaller sub-components can improve efficiency. This is similar to the approach we adopt for multiplane images.

3. Method

A traditional MPI represents the scene as a set of N fronto-parallel planes in the camera frustum of a reference view $\mathcal{I}$. Each plane is associated with an RGBA image. While it is possible to place the planes at any depth, they are usually arranged linearly in disparity (inverse depth) [53, 9, 26, 42]. A novel view $\mathcal{I}_t$ is rendered by warping the planes into the target camera's image space via a homography, and compositing them front-to-back using the *over* [31] operator:

$$\mathcal{I}_t = \sum_{i=1}^{N} \left(\alpha_i c_i \prod_{j=i+1}^{N} (1 - \alpha_i) \right) \tag{1}$$

where α_i and c_i are the warped alpha and color channels, respectively, of the i^{th} plane.

Both warping and compositing can be done very efficiently on graphics hardware, allowing real-time rendering of novel views [26]. Moreover, the alpha channel at each plane allows MPIs to represent soft edges and anti-alias any errors in the scene reconstruction, leading to fewer perceptually objectionable artifacts than depth-based warping methods. However, the number of planes N required to capture all the depth variation in a scene is usually large, even though most planes are very sparse. We propose to overcome this shortcoming by representing the reference image $\mathcal{I}$ as a tiled grid of many small MPIs (Figure 3). Given $\mathcal{I}$ and its depth map from a monocular depth estimator, our method predicts the placement of $n \ll N$ depth planes within each tiled region and uses this prediction to generate the RGBA images of the MPI in a single forward pass.

3.1. Tiled Multiplane Image Representation

Our scene representation is based on a set of m tiles, each representing a square sliding block of size h at 2D pixel locations $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m$ in the source image $\mathcal{I}$. The locations $\mathbf{x}_k$ lie on a regular grid with spacing determined by some stride r. Each tile consists of n front-parallel RGBA planes, the depth placement of which is not fixed but varies across tiles. We let α_j^i, c_j^i and d_j^i denote, respectively, the alpha channel, the color channel, and depth of the i-th plane in the j-th tile. Then the *tiled multiplane image* (TMPI) representation $\Gamma(\mathcal{I})$ of the image is defined as:

$$\Gamma(\mathcal{I}) = \left\{ (\alpha_j^i, c_j^i, d_j^i, \mathbf{x}_j) \right\} \tag{2}$$

for all $i = 1, 2, ..., n$ and $j = 1, 2, ..., m$. Further, we define an ordering on this set of 4-tuplets as,

$$\left(\Gamma(\mathcal{I}), \leq \right) = \left\{ (\alpha, c, d, \mathbf{x})_{k=1,...,mn} \mid d_k \leq d_{k+1} \right\} \tag{3}$$

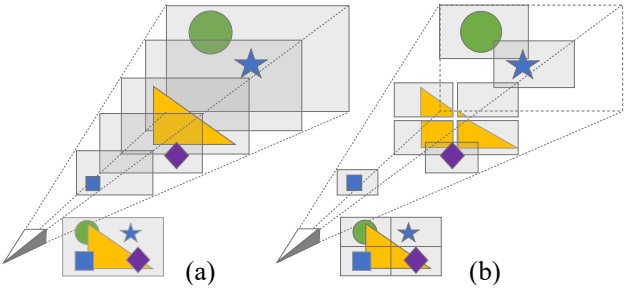

Figure 3: **(a)** A traditional MPI uses five planes per pixel to represent this toy scene, even though no region has more than two overlapping objects. **(b)** A TMPI exploits the low local depth complexity and has only two planes per pixel.

As with traditional MPIs, novel views are rendered in a differentiable manner by warping all planes into the target camera's image space. However, as the depth of the planes varies across the tiles and, hence, across the pixels of $\mathcal{I}$, a planar inverse warp on the target image plane cannot be directly computed via a homography. Instead, each plane must be warped in *tile space* via a homography computed using a shifted intrinsic matrix, and all the warped planes composited sequentially in the target view at their respective tile locations (Algorithm 1). While this makes the rendering of tiled multiplane images less efficient and somewhat less elegant than MPIs (Equation 1), this is only true during training when differentiability is required. At inference, the tiles can be rendered as textured quads using hardware-accelerated rasterization making this stage of the pipeline as efficient as traditional MPIs and with lower texture memory requirements. Thus, their compact form makes TMPIs well-suited for rendering over networks, or on mobile devices and VR headsets. The blending of many MPIs locally is related to Mildenhall *et al.*'s [26] light field fusion. However, their method renders each MPI separately before blending the results with scalar weights. Our method operates at a much finer scale, and composites the planes of all MPIs together to synthesize a novel view.

3.2. Single-View 3D Photography

We now describe our approach to generating tiled multiplane images from a single RGB input (Figure 2). Broadly, our method splits the image plane into a regular tiled grid of learnt confidence and outlier-corrected depth. For each tile, the placement of n fronto-parallel depth planes is determined by clustering pixel depth values weighted by the predicted confidence estimates. This latter step is motivated by the fact that with a small depth plane budget n, the commonly used equal spacing in disparity [53, 9, 26, 42] is wasteful. Thus, the goal is to predict the planes that optimally represent all depth variation within a tile. Using the

Algorithm 1: Differentiable view synthesis using TMPIs. Angled brackets denote pixel indexing.

RenderTMPI $\left(\left(\Gamma(\mathcal{I}), \leq\right), \boldsymbol{R}, \boldsymbol{t}, \boldsymbol{K}\right)$
- **Input :** $\left(\Gamma(\mathcal{I}), \leq\right)$: ordered TMPI planes
  - $\mathbf{R}$: relative rotation of novel view
  - $\mathbf{t}$: relative translation
  - $\mathbf{K}$: camera intrinsics
- **Output:** Novel view $\mathcal{I}_t \in \mathbb{R}^{3\times H\times W}$
- $\mathcal{I}_t \leftarrow \mathbb{0}^{3\times H\times W}$; $\mathcal{T}_t \leftarrow \mathbb{1}^{H\times W}$;
- **foreach** $(\alpha, c, d, \boldsymbol{x}) \in \left(\Gamma(\mathcal{I}), \leq\right)$ **do**
  - $\hat{\mathbf{K}} \leftarrow \mathbf{K} - \begin{bmatrix}\mathbf{I} & \mathbf{x}\\ 0 & 1\end{bmatrix}$;
  - **foreach** $\boldsymbol{u} \in [1, ..., s] \times [1, ..., s]$ **do**
    - $\mathbf{n} \leftarrow [0, 0, 1]^T$;
    - $\begin{bmatrix}\mathbf{u}_s\\ 1\end{bmatrix} \leftarrow \hat{\mathbf{K}}\left(\mathbf{R} - \mathbf{t}\mathbf{n}^T/d\right)\hat{\mathbf{K}}^{-1}\begin{bmatrix}\mathbf{u}\\ 1\end{bmatrix}$;
    - $w \leftarrow \alpha\langle\mathbf{u}_s\rangle\, \mathcal{T}_t\langle\mathbf{u}+\mathbf{x}\rangle$;
    - $\mathcal{I}_t\langle\mathbf{u}+\mathbf{x}\rangle \leftarrow \mathcal{I}_t\langle\mathbf{u}+\mathbf{x}\rangle + w\, c\langle\mathbf{u}_s\rangle$;
    - $\mathcal{T}_t\langle\mathbf{u}+\mathbf{x}\rangle \leftarrow \mathcal{T}_t\langle\mathbf{u}+\mathbf{x}\rangle\left(1 - \alpha\langle\mathbf{u}_s\rangle\right)$;
  - **end**
- **end**
- **return** $\mathcal{I}_t$

end

predicted planes, a fully convolutional network generates the n RGBA images that constitute a per-tile MPI. Unlike Han *et al.*'s [12] adaptive plane method, we generate the RGBA images in a single forward pass. The resulting TMPI is rendered as a set of textured quads using the rasterization pipeline of graphics hardware.

In more detail, given the source image $\mathcal{I}$, we first obtain a depth map $\mathcal{Z}$ for it using a monocular depth estimator. A two-headed U-Net $\Theta(\cdot)$ then predicts a confidence map $\mathcal{C}$ along with denoised depth $\mathcal{D}$. The goal is to learn a representation that ameliorates the sensitivity of the subsequent k-means clustering step to outliers. The joint prediction of confidence and depth is similar to the depth-routing of Weder *et al.* [45] and the aleotoric uncertainty estimation of [15]. The predicted depth and confidence, and the original color image are then unfolded into a set of m square sliding blocks of size h and stride r: $\{(\mathcal{D}^i, \mathcal{C}^i, \mathcal{I}^i)_{i=1,...,m}\}$.

Running $\Theta(\cdot)$ on $\mathcal{I}$ and $\mathcal{Z}$ rather than individual tiles allows it to consider non-local features and avoid undesirable tiling artifacts. Additionally, setting $r < h$ in the unfolding step allows neighboring tiles to overlap. This prevents gaps along tile boundaries and also regularizes per-tile operations across neighbors. However, it also increases the total number of tiles and, thus, the computational requirements. We empirically determine a stride value of $r = h - h/8$ for a good balance between quality and computational efficiency.

Per-Tile Planes Prediction: Next, we predict the n depth planes $\{z^i_{j=1,\dots,n}\}$ that optimally represent the features of the i-th tile. The common approach of spacing the planes linearly in disparity grows inaccurate as n becomes small. Luvizon *et al*. [25] place the planes at depth discontinuities identified via the histogram of depth values. However, their approach is sensitive to parameter settings and fails for smooth surfaces which have no discontinuities. A learning-based approach is adopted by Han *et al*. [12] and Li *et al*. [23]. The former use multi-headed self-attention to adjust a linear placement. While capable of modeling inter-plane interactions, their method is computationally expensive (Table 2). The latter uses a CNN to directly predict the planes. However, we found that without strong regularization a direct approach lacks topological order and has a strong bias towards a fixed placement. An adversarial loss helps improve this but makes the training more unstable.

We observe that as depth is known, plane positioning can be posed as a simple clustering problem. Thus, we predict $\{z^i_j\}$ using k-means clustering on the depth in each tile $\mathcal{D}^i$.

Standard k-means is sensitive to outliers and can generate significantly different plane predictions across neighboring tiles causing artifacts in novel views. Advantageously, along with the n depth planes, clustering also assigns a label to each input pixel, thereby generating a label map that represents discretized depth. Furthermore, the cluster centers of k-means are differentiable with respect to the input samples. Thus, we address the outlier problem by training $\Theta(\cdot)$ to filter the input through a self-supervised reconstruction loss on the discretized depth map generated by a weighted k-means. In weighted k-means, the cluster centers are updated each iteration using the confidence-weighted mean of the constituent samples. Since we do not directly supervise the depth output of $\Theta(\cdot)$, the network can go beyond outlier-filtering to learn any modifications that improves the discrete reconstruction, and consequently optimizes the placement of the n depth planes within each tile.

Per-Tile MPI Generation: Given $\{z^i_j\}$ and the discrete depth map, we estimate a preliminary MPI per tile by peeling RGBA layers from $\mathcal{I}^i$ using the discrete labels as an alpha mask. The masked RGB regions of each plane are inpainted by upsampling valid values from a Gaussian pyramid. A second network $\Psi(\cdot)$ then refines these estimates to generate the final n RGBA images of the MPI for each tile. Following Zhou *et al*. [53] and Tucker and Snavely [42], we represent the RGBA output as a pixel-wise blend of the input image $\mathcal{I}^i$ and a learnt background. However, unlike these works we predict a background image $\mathcal{B}^i_j$ per plane:

$$\mathcal{W}^i_j = \prod_{k>j}(1-\alpha^i_k), \tag{4}$$

$$\mathcal{I}^i_j = \mathcal{W}^i_j \mathcal{I}^i + (1-\mathcal{W}^i_j)\mathcal{B}^i_j \tag{5}$$

Where α^i_k is the predicted alpha value for each plane.

Spaces Dataset				
Method	PSNR↑	SSIM↑	LPIPS↓	L1↓
SVMPI [42]	25.42	0.748	0.210	0.040
VMPI [23]	22.37	0.636	0.268	0.057
MINE [22]	24.02	0.702	0.229	0.048
AdaMPI [12]	26.17	0.703	0.229	0.047
Ours	24.93	0.750	0.175	0.037
Tanks & Temples Dataset				
Method	PSNR↑	SSIM↑	LPIPS↓	L1↓
SVMPI	17.85	0.530	0.370	0.082
VMPI	16.32	0.463	0.395	0.103
MINE	17.23	0.506	0.366	0.088
AdaMPI	18.62	0.565	0.270	0.073
Ours	18.69	0.569	0.267	0.073

Table 1: Quantitative evaluation of view synthesis results on the *Spaces* and *Tanks and Temples* multi-view datasets. SVMPI, MINE and AdaMPI use 32 MPI depth planes; VMPI uses 8; our approach uses 4 planes per image tile.

4. Training Procedure

We train $\Theta(\cdot)$ in a self-supervised manner by minimizing the L1 loss between the input depth $\mathcal{Z}$ and the folded discrete depth maps produced per tile by weighted k-means clustering. Then, we freeze $\Theta(\cdot)$ and train the MPI generation network $\Psi(\cdot)$ on a novel view synthesis task by following Han *et al*.'s [12] warp-back strategy to generate pseudo ground truth multi-view training data. This involves warping single-view images into a target camera using monocular depth, and inpainting disocclusion holes with a specially trained network. The view-synthesis training objective is a combination of VGG, structural similarity [44], and L1 losses on the synthesized color image $\mathcal{I}_t$, weighed as 0.1, 0.25, and 1.0 respectively. For both networks, we use the 111K images of the COCO dataset [4] and the monocular depth method of Ranftl *et al*. [32].

5. Experiments

5.1. Implementation

Our method is implemented in PyTorch and trained on eight Nvidia Tesla V100 GPUs. We use 256×384 images and accumulate gradients across eight mini-batches of 16 samples each. For both networks $\Theta(\cdot)$ and $\Psi(\cdot)$, we use the Adam optimizer with a learning rate of 1×10^{-3} and a cosine annealing schedule with restarts every 200 epochs. A vectorized implementation of the differentiable TMPI renderer (Algorithm 1) runs at ∼100ms / mini-batch allowing efficient parallel computation.

	Params. (M) ↓	GMAC ↓	Runtime (ms) ↓	Peak (GB) ↓	Space (MB) ↓
DPT*	123.0	110	31	4.39	–
SVMPI	43.5	58.0	111	4.53	26.9
MINE	38.1	250	110	4.91	107
VMPI†	4.31	52.3	96.0	4.57	13.5
AdaMPI†	19.0	288	350	5.94	107
Ours†	6.43	57.0	91.6	3.20	5.25

*Monocular depth method. † Method uses monocular depth.

Table 2: Run-time, memory and complexity evaluation of all methods with a single 350×630 RGB image on an NVIDIA GeForce RTX 3080 GPU. The choice of resolution is dictated by the baseline methods which run out of memory for larger images on the specified hardware. GMACS are Giga Multiply-Accumulate ops./second.

5.2. Baselines

We compare our approach to four state-of-the-art single-view 3D photography methods based on multiplane images: Tucker *et al*. [42] (SVMPI), Li *et al*. [23] (VMPI), Li *et al*. [22] (MINE) and Han *et al*. [12] (AdaMPI). Like us, VMPI and AdaMPI use a monocular estimator to recover depth as the first step of their pipeline. The input to all methods is the same, however — a single unconstrained RGB image — and so we evaluate them as end-to-end 3D photography approaches. Nonetheless, we do note the additional depth estimation step when evaluating computational and memory performance (Table 2). We use $N = 32$ planes for all baselines except VMPI, which is designed for $N = 8$.

We do not compare to single-view methods based on neural radiance fields [51] or the recent work of Nicklaus *et al*. [29] and Shih *et al*. [38] as these are too computationally intensive for our intended use cases on mobile and VR devices. While recent work on NeRFs has demonstrated impressive rendering speeds, training remains expensive, and further, requires a large number of input views and a static scene. Jampani *et al*.'s [14] method, though not MPI-based, is related. But the authors have not released their code.

5.3. Testing Datasets

We test all methods on the *Spaces* [9], and *Tanks and Temples* [17] datasets. *Spaces* consists of 100 indoor and outdoor scenes captured using a purpose-built, 16-camera rig. For *Tanks and Temples* we use the *Intermediate* split which has uniformly sampled frames from eight high-resolution videos of more challenging outdoor environments. We compute camera poses and depth maps for all scenes using COLMAP [37, 36]. The depth maps are required to resolve the scale ambiguity of monocular depth for correct reprojection to target views. We randomly select 1000 source views from each dataset scaled to 350×630 and use the next image in capture sequence as the target for view synthesis. This choice of resolution is dictated by the baseline methods which run out of memory for larger images on the specified hardware. We present high resolution results for our method on the Davis dataset [30] in Figure 5.

Spaces Dataset			
Variant	MAE ↓	MSE ↓	Q25 ↓
Vanilla k-means	35.3	2.00	14.0
Linear plane spacing	48.7	3.80	19.3
Ours	27.1	1.20	10.6
Tanks& Temples Dataset			
Variant	MAE ↓	MSE ↓	Q25 ↓
Vanilla k-means	35.7	2.00	14.2
Linear plane spacing	46.0	3.40	18.1
Ours	27.0	1.20	10.6

Table 3: Evaluating the reconstruction error of different depth discretization approaches. All values are $\times 10^{-3}$.

5.4. Evaluation Metrics

We evaluate the rendered views quantitatively on four metrics: Peak-Signal to Noise Ratio (PSNR), Structural Similarity [44] (SSIM), Learned Perceptual Image Patch Similarity [52] (LPIPS) with a VGG-16 backbone, and the mean absolute error (L1). Following previous work [12, 42, 23], we crop 15% of the image around the edge to account for disocclusions. Further, if more than 15% of the remaining pixels are blank in a view synthesized by any method, we discard the result across all baselines.

5.5. Results

Table 1 presents quantitative evaluation of all methods on the two test sets. Qualitative results are shown in Figure 4 for *Tanks and Temples* and Figure 9 for *Spaces*. Our approach uses $n = 4$ depth planes with a tile size of $h = 64$. The competitive performance of our method despite having much fewer planes per tile can be attributed to the adaptive placement of depth planes in each tile which allows it to effectively cover a larger depth range than the monolithic MPIs of the baselines, Memory and computational performance is evaluated in Table 2. VMPI, AdaMPI and our method have the additional overhead of monocular depth estimation using Ranftl *et al*.'s [32] DPT. Our approach has lower runtime, peak memory and space requirements than AdaMPI while achieving similar quality results.

We evaluate our plane placement strategy in Table 3. We measure the reconstruction quality of the discretized depth

Figure 4: Comparing the novel view synthesis results of the baseline methods and our approach on the *Tanks and Temples* dataset. Our results are better than SVMPI, VMPI and MINE, and comparable to AdaMPI while using far fewer depth planes.

Figure 5: View synthesis results of our method on the HD (1080×1920) Davis [30] dataset. Original views are inset.

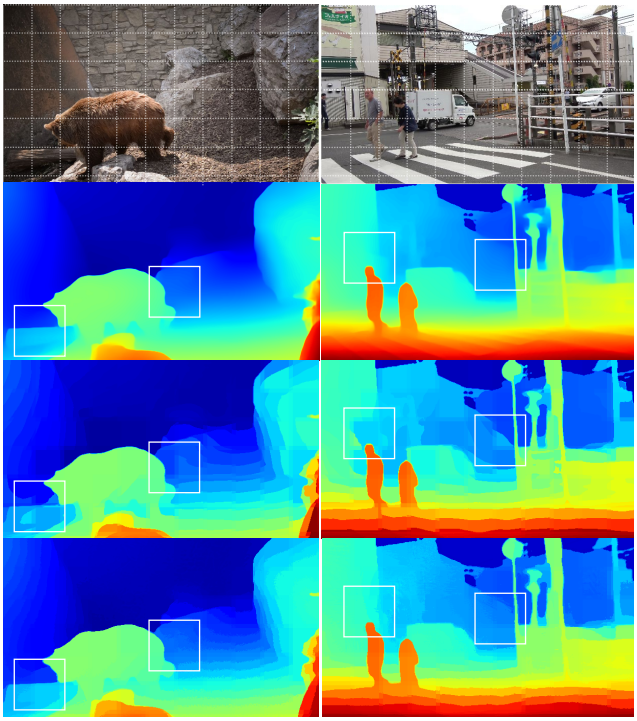

Figure 6: **Top to bottom:** Input image (Davis dataset); inverse monocular depth from DPT [32]; naive discretization by spacing four planes linearly in the inverse-depth range of each tile; our weighted clustering-based discretization with four planes. Our representation shows finer variation on receding surfaces and suffers fewer tiling artifacts.

map defined by the planes in each tile on a monocular depth input that is perturbed by a small amount of Gaussian noise ($\lambda = 0, \sigma^2 = 1 \times 10^{-3}$). We compare our approach to naive linear spacing of depth planes in inverse disparity space, and to vanilla unweighted k-means. Our method is robust to outliers and yields a much better reconstruction. Figure 6 provides qualitative comparison with linear spacing on samples from the Davis dataset [30].

Figure 7 evaluates the effect of tile size and number of planes on the quality of view synthesis. In general, the results uphold the intuition that small tiles and a large number of planes improve quality. This trend is less clear, however, for number of planes > 8. This would seem to support the observation of Khakhulin *et al*. [16] and Hu *et al*. [13] that reductive models are unable to handle redundant geometry effectively. Moreover, the model's complexity increases proportionately with the number of depth planes, leading to slower convergence for the same number of training steps.

6. Limitations

As previously observed, a tiled multiplane image cannot exploit the elegant warping and compositing equations of a traditional MPI for differentiable rendering during training. Furthermore, in some cases our method fails to reconstruct thin features consistently across tiles (Figure 8).

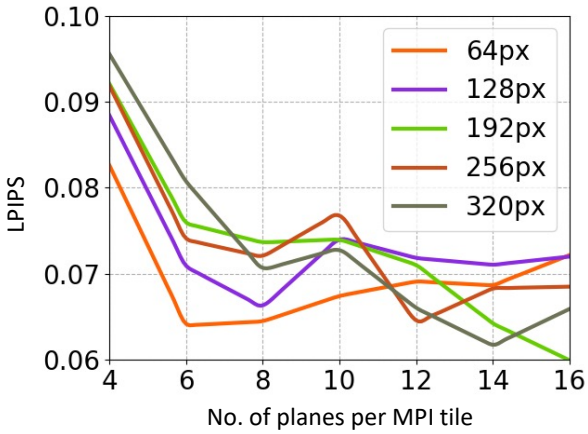

Figure 7: Evaluating the effect of tile size and number of depth planes on view synthesis quality. In general, the results uphold the intuition that small tiles and a large number of planes improve quality. The trend, however, is less clear when the number of planes > 8, indicating that the model may be unable to handle redundant geometry effectively.

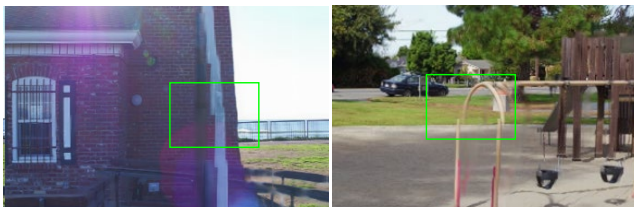

Figure 8: Failure cases of our approach: in some cases, fine features are inconsistently reconstructed across tiles.

7. Conclusion

We present a method for estimating *tiled multiplane images* from a single RGB input for 3D photography. This includes a novel approach to adaptively spacing a small number of depth planes within an MPI tile to better represent local features. Our method is lightweight, and points a path to realizing novel view synthesis on mobile and VR devices.

References

[1] Benjamin Attal, Selena Ling, Aaron Gokaslan, Christian Richardt, and James Tompkin. MatryODShka: Real-time 6DoF video view synthesis using multi-sphere images. In *European Conference on Computer Vision (ECCV)*, Aug. 2020. 2

[2] Horace B Barlow et al. Possible principles underlying the transformation of sensory messages. *Sensory communication*, 1(01):217–233, 1961. 2

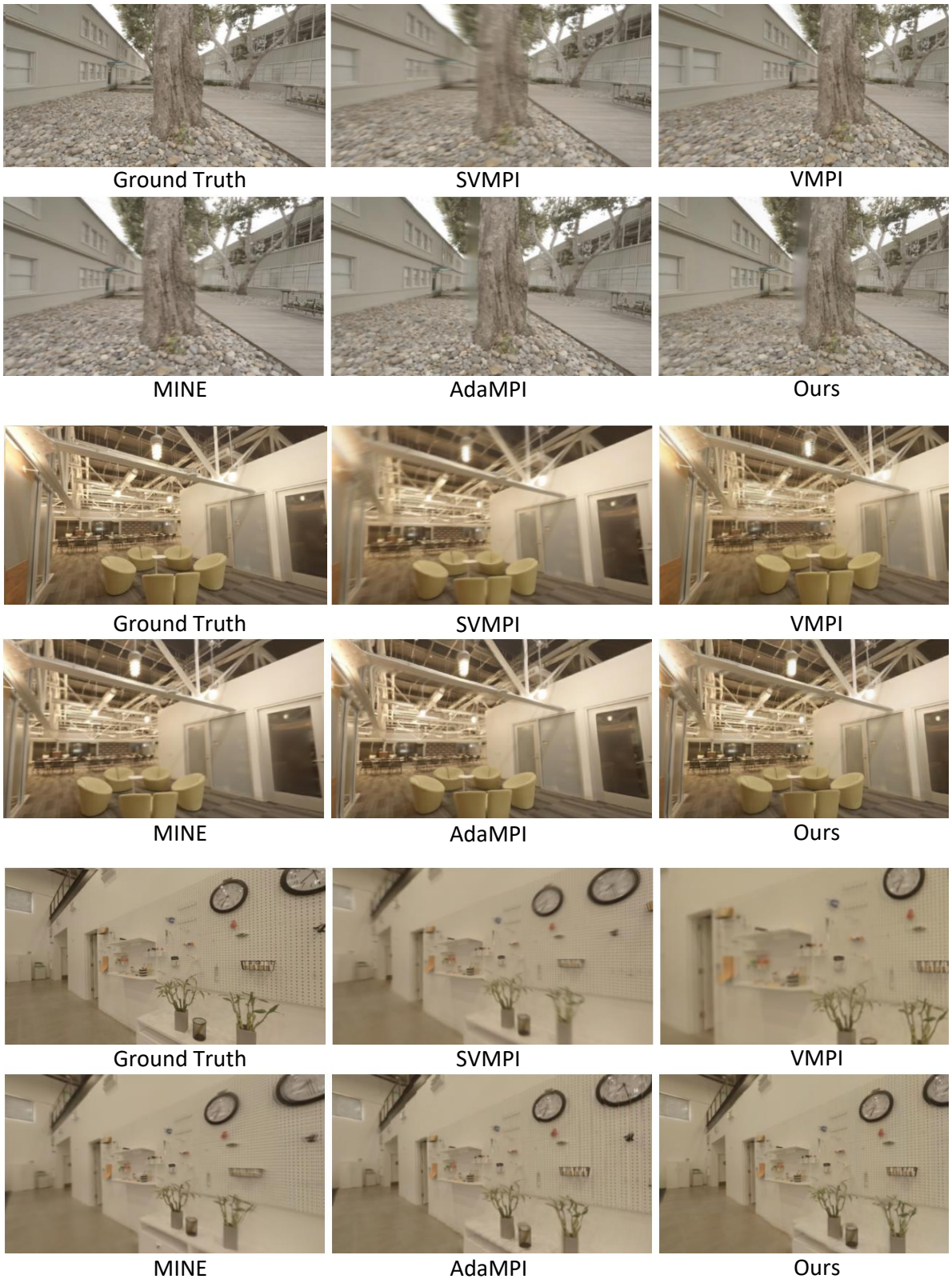

Figure 9: Comparing the novel view synthesis results of the baseline methods and our approach on the *Spaces dataset*.

[3] Michael Broxton, John Flynn, Ryan Overbeck, Daniel Erickson, Peter Hedman, Matthew Duvall, Jason Dourgarian, Jay Busch, Matt Whalen, and Paul Debevec. Immersive light field video with a layered mesh representation. *ACM Transactions on Graphics (TOG)*, 39(4):86–1, 2020. 2

[4] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018. 5

[5] Jin-Xiang Chai, Xin Tong, Shing-Chow Chan, and Heung-Yeung Shum. Plenoptic sampling. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 307–318, 2000. 2

[6] Gaurav Chaurasia, Arthur Nieuwoudt, Alexandru-Eugen Ichim, Richard Szeliski, and Alexander Sorkine-Hornung. Passthrough+ real-time stereoscopic view synthesis for mobile mixed reality. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 3(1):1–17, 2020. 2

[7] Inchang Choi, Orazio Gallo, Alejandro Troccoli, Min H Kim, and Jan Kautz. Extreme view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7781–7790, 2019. 2

[8] Abe Davis, Marc Levoy, and Fredo Durand. Unstructured light fields. In *Computer Graphics Forum*, volume 31, pages 305–314. Wiley Online Library, 2012. 1, 2

[9] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2367–2376, 2019. 2, 3, 4, 6

[10] Sushobhan Ghosh, Zhaoyang Lv, Nathan Matsuda, Lei Xiao, Andrew Berkovich, and Oliver Cossairt. Liveview: dynamic target-centered mpi for view synthesis. *arXiv preprint arXiv:2107.05113*, 2021. 2

[11] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54, 1996. 1, 2

[12] Yuxuan Han, Ruicheng Wang, and Jiaolong Yang. Single-view view synthesis in the wild with learned adaptive multiplane images. In *ACM SIGGRAPH*, 2022. 2, 4, 5, 6

[13] Ronghang Hu, Nikhila Ravi, Alexander C Berg, and Deepak Pathak. Worldsheet: Wrapping the world in a 3d sheet for view synthesis from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12528–12537, 2021. 2, 8

[14] Varun Jampani, Huiwen Chang, Kyle Sargent, Abhishek Kar, Richard Tucker, Michael Krainin, Dominik Kaeser, William T Freeman, David Salesin, Brian Curless, and Ce Liu. Slide: Single image 3d photography with soft layering and depth-aware inpainting. In *Proceedings of the IEEE International Conference on Computer Vision*, 2021. 3, 6

[15] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017. 4

[16] Taras Khakhulin, Denis Korzhenkov, Pavel Solovev, Gleb Sterkin, Andrei-Timotei Ardelean, and Victor Lempitsky. Stereo magnification with multi-layer images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8687–8696, 2022. 8

[17] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017. 6

[18] Johannes Kopf, Suhib Alsisan, Francis Ge, Yangming Chong, Kevin Matzen, Ocean Quigley, Josh Patterson, Jossie Tirado, Shu Wu, and Michael F Cohen. Practical 3d photography. In *Proceedings of CVPR Workshops*, volume 1, 2019. 2

[19] Johannes Kopf, Kevin Matzen, Suhib Alsisan, Ocean Quigley, Francis Ge, Yangming Chong, Josh Patterson, Jan-Michael Frahm, Shu Wu, Matthew Yu, et al. One shot 3d photography. *ACM Transactions on Graphics (TOG)*, 39(4):76–1, 2020. 2

[20] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42, 1996. 1, 2

[21] David Li, Yinda Zhang, Christian Häne, Danhang Tang, Amitabh Varshney, and Ruofei Du. Omnisyn: Synthesizing 360 videos with wide-baseline panoramas. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 670–671. IEEE, 2022. 2

[22] Jiaxin Li, Zijian Feng, Qi She, Henghui Ding, Changhu Wang, and Gim Hee Lee. MINE: Towards continuous depth mpi with nerf for novel view synthesis. In *ICCV*, 2021. 2, 5, 6

[23] Qinbo Li and Nima Khademi Kalantari. Synthesizing light field from a single image with variable mpi and two network fusion. *ACM Trans. Graph.*, 39(6):229–1, 2020. 2, 5, 6

[24] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T Freeman. Learning the depths of moving people by watching frozen people. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4521–4530, 2019. 2

[25] Diogo C Luvizon, Gustavo Sutter P Carvalho, Andreza A dos Santos, Jhonatas S Conceicao, Jose L Flores-Campana, Luis GL Decker, Marcos R Souza, Helio Pedrini, Antonio Joia, and Otavio AB Penatti. Adaptive multiplane image generation from a single internet picture. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2556–2565, 2021. 2, 5

[26] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 2019. 1, 2, 3, 4

[27] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 3

[28] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a mul-

tiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 3

[29] Simon Niklaus, Long Mai, Jimei Yang, and Feng Liu. 3d ken burns effect from a single image. *ACM Transactions on Graphics (ToG)*, 38(6):1–15, 2019. 2, 6

[30] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. 6, 7, 8

[31] Thomas Porter and Tom Duff. Compositing digital images. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 253–259, 1984. 3

[32] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12179–12188, 2021. 5, 6, 8

[33] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14335–14345, 2021. 3

[34] Gernot Riegler and Vladlen Koltun. Free view synthesis. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX 16*, pages 623–640. Springer, 2020. 2

[35] Gernot Riegler and Vladlen Koltun. Stable view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12216–12225, 2021. 2

[36] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6

[37] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 6

[38] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 3d photography using context-aware layered depth inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 6

[39] Pratul P Srinivasan, Richard Tucker, Jonathan T Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. Pushing the boundaries of view extrapolation with multiplane images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 175–184, 2019. 2

[40] Richard Szeliski and Polina Golland. Stereo matching with transparency and matting. In *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*, pages 517–524. IEEE, 1998. 2

[41] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8248–8258, 2022. 3

[42] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2, 3, 4, 5, 6

[43] Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12922–12931, 2022. 3

[44] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 5, 6

[45] Silvan Weder, Johannes Schonberger, Marc Pollefeys, and Martin R Oswald. Routedfusion: Learning real-time depth map fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4887–4897, 2020. 4

[46] Bennett Wilburn, Neel Joshi, Vaibhav Vaish, Eino-Ville Talvala, Emilio Antunez, Adam Barth, Andrew Adams, Mark Horowitz, and Marc Levoy. High performance imaging using large camera arrays. In *ACM SIGGRAPH 2005 Papers*, pages 765–776. 2005. 1

[47] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7467–7477, 2020. 2

[48] Suttisak Wizadwongsa, Pakkapon Phongthawee, Jiraphon Yenphraphai, and Supasorn Suwajanakorn. Nex: Real-time view synthesis with neural basis expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8534–8543, 2021. 2

[49] Lei Xiao, Salah Nouri, Joel Hegland, Alberto Garcia Garcia, and Douglas Lanman. Neuralpassthrough: Learned real-time view synthesis for vr. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022. 2

[50] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. In *Computer Graphics Forum*, volume 41, pages 641–676. Wiley Online Library, 2022. 1, 3

[51] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 3, 6

[52] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6

[53] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *SIGGRAPH*, 2018. 2, 3, 4, 5