Pushing the Boundaries of View Extrapolation with Multiplane Images

Pratul P. Srinivasan[1] Richard Tucker[2] Jonathan T. Barron[2]
Ravi Ramamoorthi[3] Ren Ng[1] Noah Snavely[2]
[1]UC Berkeley, [2]Google Research, [3]UC San Diego

Abstract

We explore the problem of view synthesis from a narrow baseline pair of images, and focus on generating high-quality view extrapolations with plausible disocclusions. Our method builds upon prior work in predicting a multiplane image (MPI), which represents scene content as a set of RGBα planes within a reference view frustum and renders novel views by projecting this content into the target viewpoints. We present a theoretical analysis showing how the range of views that can be rendered from an MPI increases linearly with the MPI disparity sampling frequency, as well as a novel MPI prediction procedure that theoretically enables view extrapolations of up to $4\times$ the lateral viewpoint movement allowed by prior work. Our method ameliorates two specific issues that limit the range of views renderable by prior methods: 1) We expand the range of novel views that can be rendered without depth discretization artifacts by using a 3D convolutional network architecture along with a randomized-resolution training procedure to allow our model to predict MPIs with increased disparity sampling frequency. 2) We reduce the repeated texture artifacts seen in disocclusions by enforcing a constraint that the appearance of hidden content at any depth must be drawn from visible content at or behind that depth.

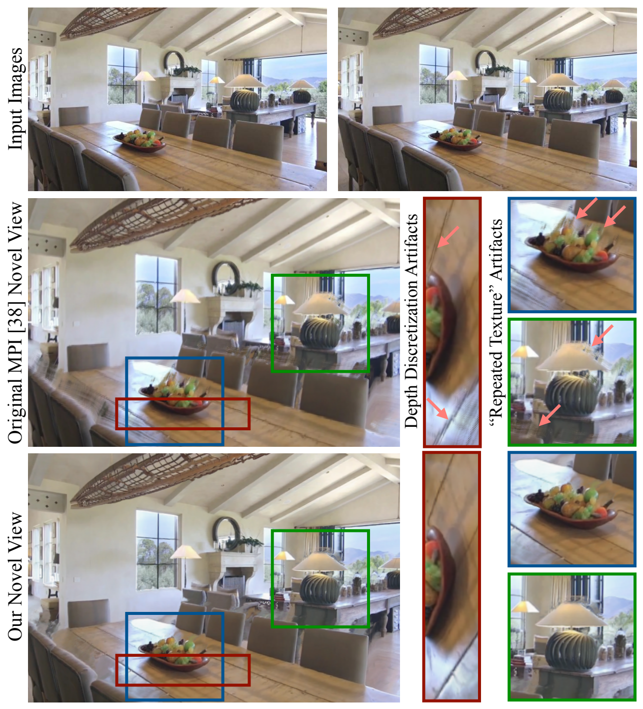
Input Images
Original MPI [38] Novel View
Depth Discretization Artifacts
"Repeated Texture" Artifacts
Our Novel View

Figure 1. Given two input images taken from nearby viewpoints, our algorithm predicts an MPI scene representation that can render view extrapolations with disocclusions. Our model improves upon prior work in two specific ways: 1) We reduce depth discretization artifacts due to insufficient depth sampling, as seen in the red zoom of the wood table. 2) We mitigate the repeated texture artifacts produced by prior methods by predicting plausible hidden scene content, as shown in the blue and green zooms where we predict realistic textures behind the fruit bowl and lamp.

1. Introduction

View synthesis, the problem of predicting novel views of a scene from a set of captured images, is a central problem in computer vision and graphics. The ability to render nearby views from a single image or a stereo pair can enable compelling photography effects such as 3D parallax and synthetic defocus blur. Furthermore, given a collection of images of a scene taken from different viewpoints, view synthesis could enable free-viewpoint navigation for virtual and augmented reality.

However, there is still a long way to go. State-of-the-art view synthesis algorithms use their input images to estimate a 3D scene representation, which can then be reprojected to render novel views. This approach works well for content visible in the input images, but the quality of novel views degrades rapidly as the target viewpoint moves further away from the input views, thereby revealing more previously-occluded scene content. In this work, we study the problem of view extrapolation where regions of the rendered images observe disoccluded content, and focus specifically on demonstrating view synthesis from a stereo input.

We build upon a state-of-the-art deep learning approach for view synthesis [38] that predicts a scene representation called a multiplane image (MPI) from an input narrow-baseline stereo pair. An MPI consists of a set of fronto-parallel RGBα planes sampled within a reference view camera frustum, as illustrated by Figure 2. Diffuse volumetric scene representations such as the MPI are becoming increasingly popular for view synthesis for a number of reasons: 1) They can represent geometric uncertainty in ambiguous regions as a distribution over depths, thereby trading perceptually-distracting artifacts in those ambiguous regions for a more visually-pleasing blur [17, 21]. 2) They are able to convincingly simulate non-Lambertian effects such as specularities [38]. 3) They are straightforward to represent as the output of a CNN and they allow for differentiable rendering, which enables us to train networks for MPI prediction using only triplets of frames from videos for input and supervision [38]. In this work, we extend the MPI prediction framework to enable rendering high-quality novel views up to $4\times$ further from the reference view than was possible in prior work. Our specific contributions are:

Theoretical analysis of MPI limits (Section 3.2). We present a theoretical framework, inspired by Fourier theory of volumetric rendering and light fields, to analyze the limits of views that can be rendered from diffuse volumetric representations such as the MPI. We show that the extent of renderable views is limited by the MPI's disparity sampling frequency, even for content visible in both the input and rendered views, and that this "renderable range" increases linearly with the MPI's disparity sampling frequency.

Improved view extrapolation for visible content (Section 3.3). View extrapolation in previous work on MPIs is limited in part by a network architecture that fixes the number of disparity planes during training and testing. Increasing the renderable range of an MPI by simply increasing its fixed number of planes during training is not computationally feasible due to the memory limits of current GPUs. We present a simple solution that increases disparity sampling frequency at test time by replacing the previously used 2D convolutional neural network (CNN) with a 3D CNN architecture and a randomized-resolution training procedure. We demonstrate that this change reduces the depth discretization artifacts found in distant views rendered by prior work, as shown in Figure 1.

Predicting disoccluded content for view extrapolation (Section 4). We observe and explain why MPIs predicted by prior work [38] contain approximately the same RGB content at each plane, and differ only in α. This behavior results in unrealistic disocclusions with repeated texture artifacts, as illustrated in Figure 1. In general, the appearance of hidden scene content is inherently ambiguous, so training a network to simply minimize the distance between rendered and ground truth target views tends to result in unrealistic hallucinations of this occluded content. We propose to improve the realism of predicted disocclusions by constraining the appearance of occluded scene content at every depth to be drawn from visible scene points at or beyond that depth, and present a two-step MPI prediction procedure that enforces this constraint. We demonstrate that this strategy forces predicted disocclusions to contain plausible textures, alleviates the artifacts found in prior work, and produces more compelling extrapolated views than alternative approaches, as illustrated in Figures 1 and 5.

2. Related Work

Traditional approaches for view synthesis. View synthesis is an image-based rendering (IBR) task, with the goal of rendering novel views of scenes given only a set of sampled views. It is useful to organize view synthesis algorithms by the extent to which they use explicit scene geometry [24]. At one extreme are light field rendering [11, 18] techniques, which require many densely sampled input images so that they can render new views by simply slicing the sampled light field without relying on accurate geometry estimation. At the other extreme are techniques such as view dependent texture mapping that rely entirely on an accurate estimated global mesh and then reproject and blend the texture from nearby input views to render new views [5].

Many successful modern approaches to view synthesis [3, 13, 21, 40] follow a strategy of computing detailed local geometry for each input view followed by forward projecting and blending the local texture from multiple input views to render a novel viewpoint. This research has traditionally focused on interpolating between input views and therefore does not attempt to predict content that is occluded in all input images. In contrast, we focus on the case of view extrapolation, where predicting hidden scene content is crucial for rendering compelling images.

Deep learning approaches for view synthesis. Recently, a promising line of work has focused on training deep learning pipelines end-to-end to render novel views. One class of methods focuses on the challenging problem of training networks to learn about geometry and rendering from scratch and synthesize arbitrarily-distant views from such limited input as a single view [7, 20, 39]. However, the lack of built-in geometry and rendering knowledge limits these methods to synthetic non-photorealistic scenarios.

Other end-to-end approaches have focused on photorealistic view synthesis by learning to model local geometry from a target viewpoint and using this geometry to backwards warp and blend input views. This includes algorithms for interpolating between views along a 1D camera path [9], interpolating between four input corner views sampled on a plane [15], and expanding a single image into a local light field of nearby views [28]. These methods separately predict local geometry for every novel viewpoint and are not able to guarantee consistency between these predictions, resulting in temporal artifacts when rendering a sequence of novel views. Furthermore, the use of backward projection means that disoccluded regions must be filled in with replicas of visible pixels, so these techniques are limited in their ability to render convincing extrapolated views.

The most relevant methods to our work are algorithms that predict a 3D scene representation from a source image viewpoint and render novel views by differentiably forward projecting this representation into each target viewpoint. This approach ensures consistency between rendered views and allows for the prediction of hidden content. Tulsiani *et al*. and Dhamo *et al*. predict a layered depth image (LDI) representation [6, 31], but this approach is unable to approximate non-Lambertian reflectance effects. Furthermore, training networks to predict LDIs using view synthesis as supervision has proven to be difficult, and the training procedure requires a regularization term that encourages hidden content to resemble occluding content [31], limiting the quality of rendered disocclusions. Zhou *et al*. proposed the MPI scene representation [38], where novel views are rendered by forward projecting and alpha compositing MPI layers, and a deep learning pipeline is used to train an MPI prediction network using held-out views as supervision. They demonstrated that the MPI scene representation can convincingly render parallax and non-Lambertian effects for a small range of rendered views. We build upon this work and present a theoretical analysis of limits on views rendered from MPIs as well as a new MPI prediction framework that is able to render more compelling view extrapolations with disocclusions.

Inpainting occluded content. Predicting the appearance of content hidden behind visible surfaces can be thought of as 3D scene inpainting. The problem of inpainting in 2D images has an extensive history [12], ranging from early propagation techniques [2] to modern CNN-based inpainting [36]. However, such algorithms must be applied separately to each rendering and therefore do not ensure consistency between different views of the same occluded content.

A few recent works [1, 14, 22, 29] focus on multi-view inpainting, i.e. removing objects and inpainting the resulting empty pixels in a collection of multiple input images. This strategy operates on input image collections instead of scene representations, so it cannot be used to predict occluded content that only appears during view extrapolation.

Finally, a recent line of work [8, 27, 34] focuses on scene shape completion. These methods require an input depth image and only focus on inpainting the shape and semantics of hidden content and not its appearance, so the predicted scenes cannot be used for rendering novel views. In contrast to prior methods, our work addresses the problem of jointly inpainting the geometry, color, and opacity of hidden content in scenes to render convincing disocclusions.

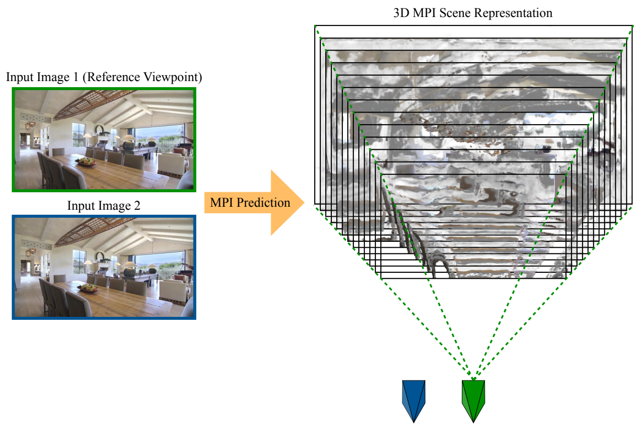
3D MPI Scene Representation
Input Image 1 (Reference Viewpoint)
Input Image 2
MPI Prediction

Figure 2. **MPI scene representation.** Our work builds on the MPI scene representation and prediction procedure introduced in [38]. We train a deep network that takes two narrow-baseline images of a scene as input (captured at the blue and green camera poses shown above), and predicts an MPI scene representation, consisting of a set of fronto-parallel RGBα planes within a reference camera frustum (signified by the green camera above). Novel views are rendered by alpha compositing along rays from the MPI voxels into the novel viewpoint.

3. View Extrapolation for Visible Content

3.1. MPI scene representation

The multiplane image (MPI) scene representation, introduced by Zhou *et al*. [38] and illustrated in Figure 2, consists of a set of fronto-parallel RGBα planes within a reference camera's view frustum. An MPI can be thought of as a frustum-shaped volumetric scene representation where each "voxel" consists of a diffuse RGB color and opacity α. Novel views are rendered from an MPI by alpha compositing the color along rays into the novel view using the "over" operator [17, 23], which is easily implemented as homography-warping each MPI plane onto the sensor plane of the novel view (see Equation 2 in Zhou *et al*. [38]), and alpha compositing the resulting images from back to front.

3.2. Theoretical signal processing limits for rendering visible content

Perhaps surprisingly, there is a limit on views that can be rendered with high fidelity from an MPI, even if we just consider mutually-visible content, i.e., content visible from all input and target viewpoints. Rendering views beyond this limit results in depth discretization artifacts similar to aliasing artifacts seen in volume rendering [17].

We formalize this effect in the context of MPI renderings, and make use of Fourier theory to derive a bound on viewpoints that can be rendered from an MPI with high fidelity. Our model of rendering mutually-visible content from an MPI is conceptually similar to Frequency domain volume rendering [30] using a shear-warp factoriza-

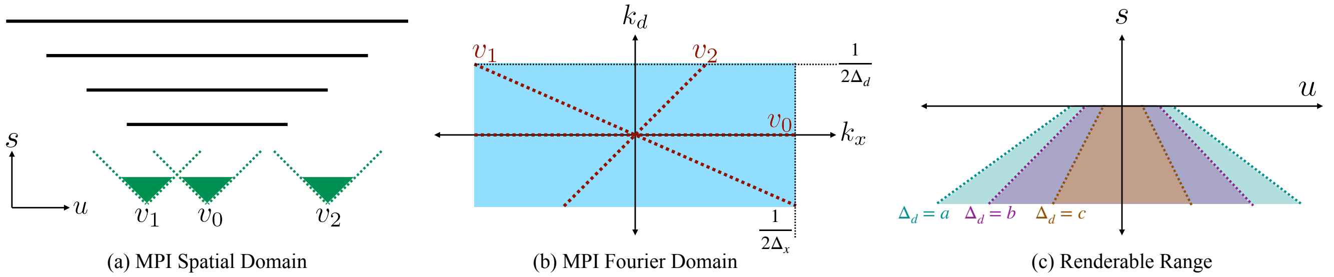

Figure 3. **Viewpoint limits for rendering visible content from an MPI.** Views rendered from an MPI without occlusions can be expressed as sheared integral projections of that MPI. (a) Here, we visualize a 2D slice from an MPI, where the y dimension is constant and only the x and z dimensions vary. This MPI is in the reference viewpoint v_0. (b) In the frequency domain, rendered views are equivalent to 1D slices of the 2D MPI spectrum, where views further from the reference viewpoint correspond to Fourier slices at steeper slopes. The MPI spectrum is bandlimited due to its spatial and disparity sampling frequencies, so there is a range of viewpoints outside which rendered views will have a lower spatial bandwidth than the original MPI plane images. Viewpoint v_1 represents the maximum extent of this "renderable range", and v_2 represents a viewpoint outside this range. (c) The renderable range of views is shaped like a truncated cone, and we visualize how the range of renderable views shrinks linearly as we increase the disparity sampling interval Δ_d from $a < b < c$.

tion [16]. Additionally, our derivation of an MPI's "renderable range" is inspired by derivations for a 3D display's depth-of-field [41] and light field photography's "refocusable range" [19]. Our main insight is that the 2D Fourier Transform of a view rendered from an MPI can be considered as a 2D slice through the 3D Fourier Transform of the MPI. An MPI is bandlimited by its fixed sampling frequency, so there exists a range of viewpoints outside of which rendered views will have a smaller spatial frequency bandwidth than the input images, potentially resulting in aliasing artifacts. We cover the main steps of this derivation below. Please refer to our supplementary materials for detailed intermediate steps and diagrams.

Let us consider rendering views from an MPI in the simplified case where (a) the camera is translated but not rotated, and (b) there is no occlusion, so all content is equally visible from every viewpoint. The rendered view $r_{\mathbf{u},s}(\mathbf{x})$ at a lateral translation $\mathbf{u}$ and axial translation s relative to the reference camera center can then be expressed as:

$$r_{\mathbf{u},s}(\mathbf{x}) = \sum_{d \in \mathcal{D}} c(\mathbf{x}', d) = \sum_{d \in \mathcal{D}} c\left((1 - sd)\,\mathbf{x} + \mathbf{u}d, d\right) \quad (1)$$

where $c(\mathbf{x}, d)$ is the pre-multiplied RGBα at each pixel coordinate $\mathbf{x}$ and disparity plane d within the set of MPI disparity planes $\mathcal{D}$. Note that $\mathbf{u}$ and s are in units of pixels (such that the camera focal length $f = 1$), and we limit s to the range $-\infty < s < 1/d_{max}$ because renderings are not defined for viewpoints within the MPI volume. Additionally, note that the disparity d is in units $1/pixel$.

To study the limits of views rendered from an MPI, let us consider a worst-case MPI with content in the subset of closest planes, for which we make a locally linear approximation to the coordinate transformation $(\mathbf{x}, d) \to (\mathbf{x}', d)$:

$$r_{\mathbf{u},s}(\mathbf{x}) = \sum_{d \in \mathcal{D}} c\left((1 - sd_{max})\,\mathbf{x} + \mathbf{u}d, d\right) \quad (2)$$

where d_{max} is a constant. Now, we have expressed the rendering of mutually-visible content as a sheared and dilated integral projection of the MPI. We apply the generalized Fourier slice theorem [19] to interpret the Fourier transform of this integral projection as a 2D slice through the 3D MPI's Fourier transform. The resulting rendered view is the slice's inverse Fourier transform:

$$r_{\mathbf{u},s}(\mathbf{x}) = \mathcal{F}^{-1}\left\{ C\left(\frac{k_{\mathbf{x}}}{1 - sd_{max}}, \frac{-\mathbf{u}k_{\mathbf{x}}}{1 - sd_{max}} \right) \right\} \quad (3)$$

where $\mathcal{F}^{-1}$ is the inverse Fourier transform and $C(k_{\mathbf{x}}, k_d)$ is the Fourier transform of $c(\mathbf{x}, d)$.

An MPI is a discretized function, so the frequency support of C lies within a box bounded by $\pm 1/2\Delta_{\mathbf{x}}$ and $\pm 1/2\Delta_d$, where $\Delta_{\mathbf{x}}$ is the spatial sampling interval (set by the number of pixels in each RGBα MPI plane image) and Δ_d is the disparity sampling interval (set by the number of MPI planes within the MPI disparity range).

Figures 3a and 3b illustrate Fourier slices through the MPI's Fourier transform that correspond to rendered views from different lateral positions. Rendered views further from the reference view correspond to slices at steeper slopes. There is a range of slice slopes within which the spatial bandwidth of the rendered views is equal to that of the MPI, and outside of which the spatial bandwidth of the rendered views decreases linearly with the slice slope.

We can solve for the worst-case "renderable range" by determining the set of slopes whose slices intersect the box in Figure 3b at the spatial frequency boundary $\pm 1/2\Delta_{\mathbf{x}}$. This provides constraints on camera positions $(\mathbf{u}, s)$, within which rendered views enjoy the full image bandwidth:

$$s \leq 0, \quad |\mathbf{u}| \leq \frac{\Delta_{\mathbf{x}}\,(1 - sd_{max})}{\Delta_d} \quad (4)$$

Figure 3c plots the renderable ranges with varying disparity intervals Δ_d, for an MPI with disparities up to d_{max}.

The allowed lateral motion extent increases linearly as the target viewpoint moves further axially from the MPI, starting at the reference viewpoint. Decreasing Δ_d linearly increases the amount of allowed lateral camera movement. Intuitively, when rendering views at lateral translations from the reference viewpoint, the renderable range boundary corresponds to views in which adjacent MPI planes are shifted by a single pixel relative to each other before compositing.

3.3. Increasing disparity sampling frequency with 3D CNN and randomized-resolution training

Section 3.2 establishes that additional MPI planes increases the view extrapolation ability, and that this relationship is linear. Accordingly, the range of extrapolated views rendered by the original MPI method [38] is limited because it uses a 2D CNN to predict a small fixed number of planes (32 planes at a spatial resolution of 1024×576). Simply increasing this fixed number of planes in the network is computationally infeasible during training due to GPU memory constraints. Additionally, training on smaller spatial patches to allow for increased disparity sampling frequency prevents the network from utilizing larger spatial receptive fields, which is important for resolving depth in ambiguous untextured regions.

We propose a simple solution to predict MPIs at full resolution with up to 128 planes at test time by using a 3D CNN architecture, theoretically increasing the view extrapolation ability by $4\times$. The key idea is that because our network is fully 3D convolutional along the height, width, and depth planes dimensions, it can be trained on inputs with varying height, width, and number of depth planes. We use training examples across a spectrum of MPI spatial and disparity sampling frequencies that fit in GPU memory, ranging from MPIs with low spatial and high disparity sampling frequency (128 planes) to MPIs with high spatial and low disparity sampling frequency (32 planes). Perhaps surprisingly, we find that the trained network learns to utilize a receptive field equal to the maximum number of spatial and disparity samples it sees during training, even though no individual training example is of that size.

Our MPI prediction network takes as input a plane-sweep-volume tensor of size $[H, W, D, 3N]$, where H and W are the image height and width, $D = |\mathcal{D}|$ is the number of disparity planes, and N is the number of input images ($N = 2$ in our experiments). This tensor is created by reprojecting each input image to disparity planes $\mathcal{D}$ in a reference view frustum. We use a 3D encoder-decoder network with skip connections and dilated convolutions [35] in the network bottleneck, so that the network's receptive field can encompass the maximum spatial and disparity sampling frequencies used during training. Please refer to our supplementary materials for a more detailed description of our network architecture and training procedure.

4. View Extrapolation for Hidden Content

In the previous section, we described how view extrapolation is limited by the disparity sampling frequency, which is a fundamental property of the MPI scene representation. View extrapolation is also limited by the quality of hidden content, which is instead a property of the MPI prediction model. Models that train a CNN to directly predict an MPI from an input plane-sweep-volume (which contains homography-warped versions of the same RGB content at each plane) learn the undesirable behavior of predicting approximately the same RGB content at each MPI plane with variation only in α (see Figure 5 in Zhou *et al.* [38]). We observe that this behavior is consistent for models that use either the original 2D CNN architecture or our 3D CNN architecture (Section 3.3). Copies of the same RGB content at different MPI layers lead to "repeated texture" artifacts in extrapolated views, where disoccluded content contains repeated copies of the occluder, as visualized in Figure 1.

We believe that this undesirable learned behavior is due to both the inductive bias of CNNs that directly predict an MPI from a plane-sweep-volume and the output uncertainty for disocclusions. The probability distribution over hidden scene content, conditioned on observed content, is highly multimodal—there may be many highly plausible versions of the hidden content. As a result, training a network to minimize the distance between rendered and ground truth views produces unrealistic predictions of disocclusions that are some mixture over the space of possible outputs.

We propose to reduce the output uncertainty by constraining the predicted hidden content at any depth, such that its appearance is limited to re-using visible scene content at or behind that depth. This effectively forces the network to predict occluded scene content by copying textures and colors from nearby visible background content. One possible limitation is that this constraint will have difficulty predicting the appearance of self-occlusions where an object extends backwards perpendicular to the viewing direction. However, as argued by the generic viewpoint assumption [10], it is unlikely that our reference viewpoint happens to view an object exactly at the angle at which it extends backwards along the viewing direction. In general, the majority of disoccluded pixels view background content instead of self-occlusions.

We enforce our constraint on the appearance of occluded content with a two-step MPI prediction procedure. The first step provides an initial estimate of the geometry and appearance of scene content visible from the reference viewpoint. The second step uses this to predict a final MPI where the color at each voxel is parameterized by a flow vector that points to a visible surface's color to copy.

In the first step, an input plane-sweep volume p is constructed by reprojecting j input images $i_{\mathbf{v}_j}$, each captured at a viewpoint $\mathbf{v}_j$, to disparity planes $d \in \mathcal{D}$. The 3D CNN Φ_1

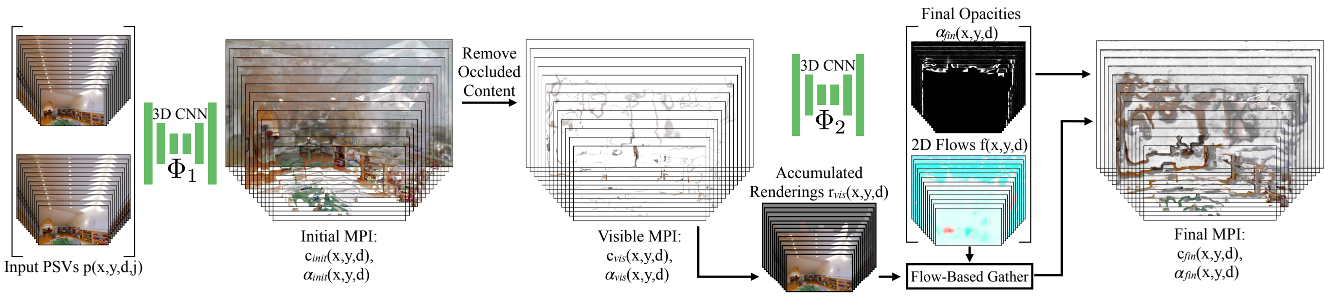

Figure 4. **Two-step MPI prediction pipeline.** We propose a two-step procedure to predict convincing hidden content in an MPI for view extrapolation. In the first step, a 3D CNN predicts an initial MPI from the input images' plane-sweep-volumes. Next, occluded content in this MPI is softly removed, resulting in a "first-visible-surface" MPI. In the second step, another 3D CNN predicts final MPI opacities and a 2D flow vector for each MPI voxel. The final MPI RGB colors are computed by using these predicted flows to gather RGB colors from back-to-front cumulative renderings of the visible content. This encourages hidden content at any depth to be synthesized by copying textures of visible content at or behind the same depth, which reduces the output space uncertainty for hidden content and thereby enables convincing view extrapolation with realistic disocclusions.

of Section 3.3 takes this plane-sweep volume and predicts an initial MPI's RGB and α values, c_{init} and α_{init}:

$$c_{init}(x,y,d), \alpha_{init}(x,y,d) = \Phi_1\big(p(x,y,d,j)\big). \quad (5)$$

This initial MPI typically contains repeated foreground textures in occluded regions of the scene. In the second step of our procedure, we aim to preserve the predicted geometry and appearance of the first visible surface from the initial MPI while re-predicting the appearance and geometry of hidden content and enforcing our flow-based appearance constraint. We softly remove hidden RGB content from this initial MPI by multiplying each MPI RGB value by its transmittance t relative to the reference viewpoint $\mathbf{v}_0$:

$$t_{\mathbf{v}_0}(x,y,d) = \alpha_{init}(x,y,d) \prod_{d' > d} \left[1 - \alpha_{init}(x,y,d')\right] \quad (6)$$

$$\begin{aligned} c_{vis}(x,y,d) &= c_{init}(x,y,d) t_{\mathbf{v}_0}(x,y,d) \\ \alpha_{vis}(x,y,d) &= t_{\mathbf{v}_0}(x,y,d) \end{aligned} \quad (7)$$

where c_{vis} and α_{vis} are the MPI RGBα planes from which content that is occluded from the reference view has been softly removed. Intuitively, a voxel's transmittance (Equation 6) describes the extent to which an MPI voxel's color contributes to the rendered reference view.

A second CNN Φ_2 takes this reference-visible MPI, consisting of c_{vis} and α_{vis}, as input and predicts opacities $\alpha_{fin}(x,y,d)$ and a 2D flow vector for each MPI voxel $f(x,y,d) = [f_x(x,y,d), f_y(x,y,d)]$:

$$\alpha_{fin}(x,y,d), f(x,y,d) = \Phi_2\big(c_{vis}(x,y,d), \alpha_{vis}(x,y,d)\big). \quad (8)$$

The final MPI's colors $c_{fin}(x,y,d)$ are computed by using these predicted flows to gather colors from renderings of the visible content at or behind each plane $r_{vis}(x,y,d)$:

$$r_{vis}(x,y,d) = \sum_{d' \leq d} \left[c_{vis}(x,y,d')\right] \quad (9)$$

$$c_{fin}(x,y,d) = r_{vis}\left(x + f_x(x,y,d), y + f_y(x,y,d), d\right).$$

We gather the color from r_{vis} using bilinear interpolation for differentiability. This constraint restricts the appearance of hidden content at each depth to be drawn from visible scene points at or beyond that depth.

5. Training Loss

As in Zhou *et al.* [38], we train our MPI prediction pipeline using view synthesis as supervision. Our training loss is simply the sum of reconstruction losses for rendering a held-out novel view r_{gt} at target camera pose $\mathbf{v}_t$, using both our initial and final predicted MPIs. These MPIs are predicted from input images $i_{\mathbf{v}_0}$ and $i_{\mathbf{v}_1}$. We use a deep feature matching loss $\mathcal{L}_{VGG}$ for layers from the VGG-19 network [26], using the implementation of Chen and Koltun [4]. The total loss $\mathcal{L}$ for each training example is:

$$\begin{aligned} \mathcal{L} = &\mathcal{L}_{VGG}(r_{init}(i_{\mathbf{v}_0}, i_{\mathbf{v}_1}, \mathbf{v}_t), r_{gt}) + \\ &\mathcal{L}_{VGG}(r_{fin}(i_{\mathbf{v}_0}, i_{\mathbf{v}_1}, \mathbf{v}_t), r_{gt}) \end{aligned} \quad (10)$$

where r_{init} and r_{fin} are rendered views from the initial and final predicted MPIs.

6. Results

The following section presents quantitative and qualitative evidence to validate the benefits of our method. Please view our supplementary video for rendered camera paths that demonstrate our predicted MPIs' ability to render high quality extrapolated views that are consistent with a 3D scene representation and contain realistic disocclusions.

6.1. Experiment details

We train and evaluate on the open-source YouTube Real Estate 10K dataset [38][1], which contains approximately 10,000 YouTube videos of indoor and outdoor real estate

[1]`https://google.github.io/realestate10k/`

Algorithm	SSIM$_{\text{fov}}$	SSIM$_{\text{occ}}$	NAT$_{\text{occ}}$
Original MPI [38]	0.838	0.803	0.805
Our r_{init}	0.858	0.811	0.904
r_{init} + Adversarial Disocclusion	0.853	0.791	0.849
Disocclusion Inpainting [36]	0.808	0.691	0.227
Our r_{fin}	0.853	0.814	0.931

Table 1. **Quantitative evaluation.** Images rendered from our predicted MPIs are quantitatively superior to those rendered from the original MPI model [38]. Furthermore, our method predicts disocclusions that are both closer to the ground truth hidden content and more perceptually plausible than alternative methods.

scenes along with computed camera poses for each video frame. We generate training examples on the fly by sampling two source frames and a target frame from a randomly chosen video, so that the target image is not in between the source images (and therefore requires view extrapolation, not view interpolation) for $\sim$87% of the training examples.

The dataset is split into 9,000 videos for training and 1,000 for testing, where the test set videos do not overlap with those in the training dataset. From these test videos, we randomly sample 6,800 test triplets, each consisting of two input frames and a single target frame.

6.2. Evaluation metrics

We use three metrics for our quantitative comparisons:
SSIM$_{\text{fov}}$: To evaluate the overall quality of rendered images, we use the standard SSIM [32] metric computed over the region of the target image that views all MPI planes.
SSIM$_{\text{occ}}$: To specifically assess the accuracy of predicted disocclusions, we evaluate SSIM over the subset of pixels that were not visible from the input reference viewpoint. We determine whether a pixel in a rendered target image is disoccluded by examining the MPI voxels that contribute to the rendered pixel's value, and thresholding the maximum change in transmittance of these contributing voxels between the reference and target viewpoint. Similarly to Equation 6, we can compute the transmittance of each MPI voxel from a target viewpoint $\mathbf{v}_t$ as:

$$t_{\mathbf{v}_t}(x,y,d) = \alpha_{\mathbf{v}_t}(x,y,d) \prod_{d'>d} [1 - \alpha_{\mathbf{v}_t}(x,y,d')] \quad (11)$$

where $\alpha_{\mathbf{v}_t}$ is an MPI α plane homography-warped onto the sensor plane of viewpoint $\mathbf{v}_t$. We consider a pixel (x,y) in the target rendered view as a member of the disoccluded pixels set $\mathcal{H}$ if the transmittance t of any contributing MPI voxel is some threshold value greater than the same voxel's transmittance when rendering the reference viewpoint:

$$\mathcal{H} = \left\{ (x,y) : \max_d \left(t_{\mathbf{v}_t}(x,y,d) - t_{\mathbf{v}_0 \to \mathbf{v}_t}(x,y,d) \right) \geq \epsilon \right\} \quad (12)$$

where $t_{\mathbf{v}_0 \to \mathbf{v}_t}$ is the transmittance relative to the reference viewpoint, warped into the target viewpoint so that both transmittances are in the same reference frame. We compute disoccluded pixels using α_{init} for all models, to ensure that each model is evaluated on the same set of pixels. We set $\epsilon = 0.075$ in our experiments. Please see our supplementary materials for visualizations of disoccluded pixels.
NAT$_{\text{occ}}$: To quantify the perceptual plausibility of predicted disoccluded content, we evaluate a simple image prior over disoccluded pixels. We use the negative log of the Earth Mover's (Wasserstein-1) distance between gradient magnitude histograms of the rendered disoccluded pixels and the ground-truth pixels in each target image. Intuitively, realistic rendered image content should have a distribution of gradients that is similar to that of the true natural image [25, 33], and therefore a higher NAT$_{\text{occ}}$ score.

6.3. Comparison to baseline MPI prediction

We first show that renderings from both our initial and final predicted MPIs (r_{init} and r_{fin}) are superior to those from the original MPI method [38], which was demonstrated to significantly outperform other recent view synthesis methods [15, 37]. The increase in SSIM$_{\text{fov}}$ from "Original MPI" (Table 1 row 2) to "Our r_{init}" (row 3) demonstrates the improvement from our method's increased disparity sampling frequency. Furthermore, the increase in SSIM$_{\text{occ}}$ and NAT$_{\text{occ}}$ from "Original MPI" (row 2) to "Our r_{fin}" (row 6) demonstrates that our method predicts disoccluded content that is both closer to the ground truth and more plausible. Figure 5 qualitatively demonstrates that renderings from our method contain fewer depth discretization artifacts than renderings from the original MPI work, and that renderings from our final MPI contain more realistic disocclusions without "repeated texture" artifacts.

6.4. Evaluation of hidden content prediction

We compare occluded content predicted by our model to the following alternative disocclusion prediction strategies:
Our r_{init}: We first compare renderings "Our r_{fin}" from our full method to the ablation "Our r_{init}", which does not enforce our flow-based occluded content appearance constraint. The improvement in SSIM$_{\text{occ}}$ and NAT$_{\text{occ}}$ from Table 1 row 3 to row 6 and the qualitative results in Figure 5 demonstrate that our full method renders disocclusions that are both closer to the ground truth and more perceptually plausible with fewer "repeated texture" artifacts.
"r_{init} + Adversarial Disocclusions": Next, we compare to an alternative two-step MPI prediction strategy. We use an identical Φ_1 to predict the initial MPI, but Φ_2 directly predicts RGBα planes instead of α and flow planes. We apply an adversarial loss to the resulting rendered target image to encourage realistic disocclusions (additional details in our supplementary materials). Table 1 row 4 demonstrates that this strategy renders disocclusions that are less accurate but more perceptually plausible than the original MPI method,

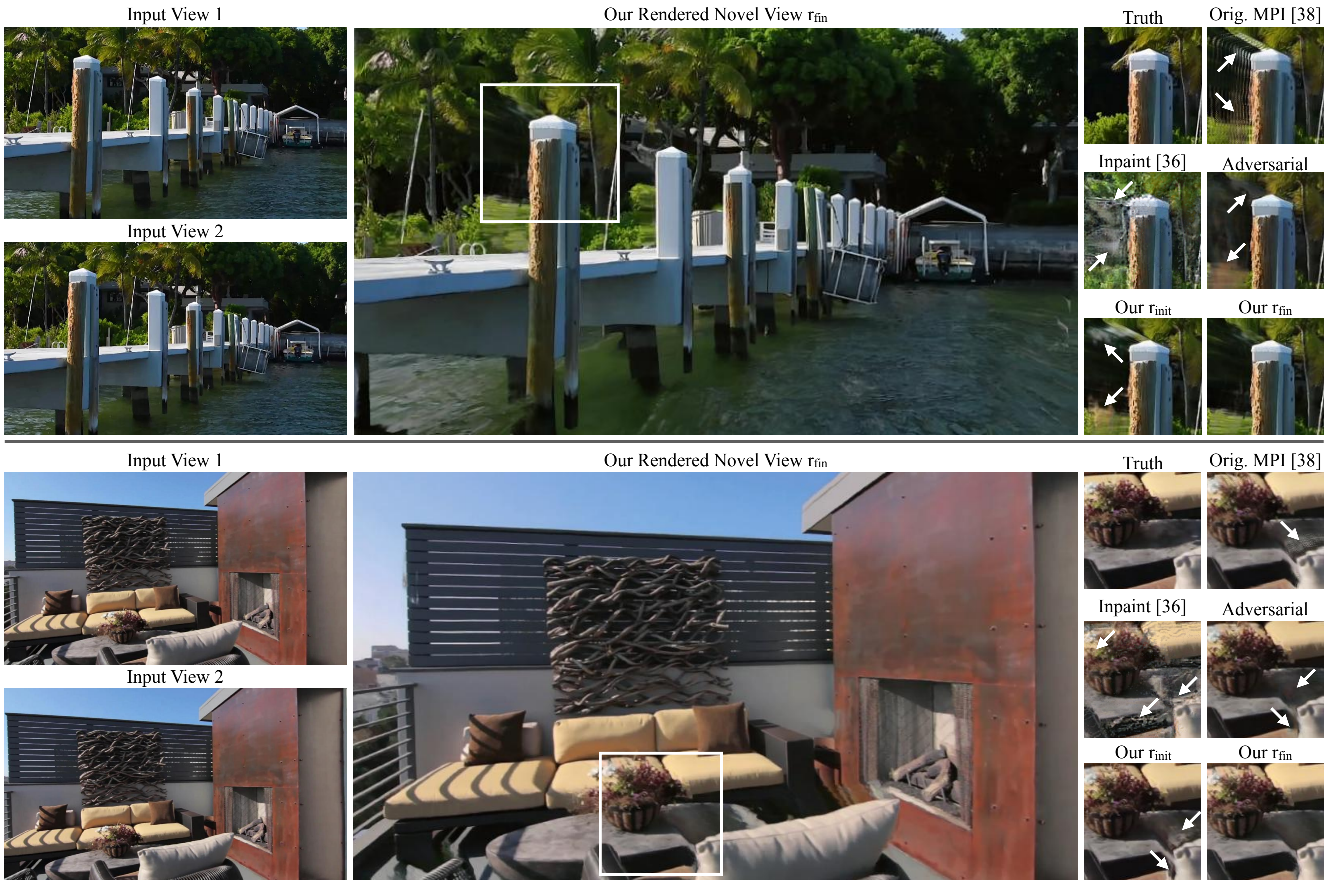

Figure 5. **Qualitative comparison of rendered novel views.** Our method predicts MPIs with convincing hidden content, as demonstrated by the disoccluded foliage textures to the left of the wooden pole in the top example, and the disoccluded region to the left of the grey pillow in the bottom example. Renderings from alternative methods contain depth discretization artifacts, implausible colors, blurry textures, and repeated textures in disoccluded regions.

due to the adversarial loss. However, Figure 5 demonstrates that the renderings from our full method contain sharper content and more accurate colors than those of the "r_{init} + Adversarial Disocclusions" strategy. We hypothesize that this is due to the difficulty of training a discriminator network when the number and location of "fake" disoccluded pixels varies drastically between training examples.

"Disocclusion Inpainting": Finally, we compare to an image-based disocclusion prediction strategy. We remove the disoccluded pixels from our final MPI renderings and re-predict them using a state-of-the-art deep learning image inpainting model [36]. Table 1 row 5 shows that this strategy results in an overall quality reduction, especially for the accuracy and plausibility of disoccluded regions. Figure 5 visualizes the unrealistic inpainting results. Furthermore, as shown in our video, predicting disocclusions separately for each rendered image creates distracting temporal artifacts in rendered camera paths because the appearance of disoccluded content changes with the viewpoint.

7. Conclusion

We have presented a theoretical signal processing analysis of limits for views that can be rendered from an MPI scene representation, and a practical deep learning method to predict MPIs that theoretically allow for $4\times$ more lateral movement in rendered views than prior work. This improvement is due to our method's ability to predict MPIs with increased disparity sampling frequency and our flow-based hidden content appearance constraint to predict MPIs that render convincing disocclusion effects. However, there is still a lot of room for improvement in predicting scene representations for photorealistic view synthesis that contain convincing occluded 3D content and are amenable to deep learning pipelines, and we hope that this work inspires future progress along this exciting research direction.

Acknowledgments We thank John Flynn and Ben Mildenhall for fruitful discussions, and acknowledge support from ONR grant N000141712687, NSF grant 1617234, and an NSF fellowship. This work was done while PPS interned at Google Research.

References

[1] Seung-Hwan Baek, Inchang Choi, and Min H. Kim. Multiview image completion with space structure propagation. *CVPR*, 2016.

[2] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of SIGGRAPH*, 2000.

[3] Gaurav Chaurasia, Sylvain Duchêne, Olga Sorkine-Hornung, and George Drettakis. Depth synthesis and local warps for plausible image-based navigation. In *ACM Transactions on Graphics (SIGGRAPH)*, 2013.

[4] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. *ICCV*, 2017.

[5] Paul Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In *Proceedings of SIGGRAPH*, 1996.

[6] Helisa Dhamo, Keisuke Tateno, Iro Laina, Nassir Navab, and Federico Tombari. Peeking behind objects: Layered depth prediction from a single image. *arXiv:1807.08776*, 2018.

[7] S. M. Ali Eslami, Danilo J. Rezende, Frederic Besse, Fabio Viola, Ari S. Morcos, Marta Garnelo, Avraham Ruderman, Andrei A. Rusu, Ivo Danihelka, Karol Gregor, et al. Neural scene representation and rendering. In *Science*, 2018.

[8] Michael Firman, Oisin Mac Aodha, Simon Julier, and Gabriel J. Brostow. Structured prediction of unobserved voxels from a single depth image. *CVPR*, 2016.

[9] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. DeepStereo: Learning to predict new views from the world's imagery. *CVPR*, 2016.

[10] William T. Freeman. Exploiting the generic viewpoint assumption. In *IJCV*, 1996.

[11] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *Proceedings of SIGGRAPH*, 1996.

[12] Christine Guillemot and Olivier Le Meur. Image inpainting: Overview and recent advances. In *IEEE Signal Processing Magazine*, 2014.

[13] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. In *ACM Transactions on Graphics (SIGGRAPH Asia)*, 2018.

[14] Joel Howard, Bryan S. Morse, Scott Cohen, and Brian L. Price. Depth-based patch scaling for content-aware stereo image completion. *WACV*, 2014.

[15] Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. Learning-based view synthesis for light field cameras. In *ACM Transactions on Graphics (SIGGRAPH Asia)*, 2016.

[16] Philippe Lacroute and Marc Levoy. Fast volume rendering using a shear-warp factorization of the viewing transformation. In *Proceedings of SIGGRAPH*, 1994.

[17] Marc Levoy. Display of surfaces from volume data. In *IEEE Computer Graphics and Applications*, 1988.

[18] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of SIGGRAPH*, 1996.

[19] Ren Ng. Fourier slice photography. In *ACM Transactions on Graphics (SIGGRAPH)*, 2005.

[20] Eunbyung Park, Jimei Yang, Ersin Yumer, Duygu Ceylan, and Alexander C. Berg. Transformation-grounded image generation network for novel 3D view synthesis. *ECCV*, 2016.

[21] Eric Penner and Li Zhang. Soft 3D reconstruction for view synthesis. In *ACM Transactions on Graphics (SIGGRAPH Asia)*, 2017.

[22] Julien Philip and George Drettakis. Plane-based multi-view inpainting for image-based rendering in large scenes. *I3D*, 2018.

[23] Thomas Porter and Tom Duff. Compositing digital images. *SIGGRAPH*, 1984.

[24] Heung-Yeung Shum and Sing Bing Kang. A review of image-based rendering techniques. In *Visual Communications and Image Processing*, 2000.

[25] Eero Simoncelli. Statistical models for images:compression restoration and synthesis. *Asilomar Conference on Signals, Systems, and Computers*, 1997.

[26] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015.

[27] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. *CVPR*, 2017.

[28] Pratul P. Srinivasan, Tongzhou Wang, Ashwin Sreelal, Ravi Ramamoorthi, and Ren Ng. Learning to synthesize a 4D RGBD light field from a single image. *ICCV*, 2017.

[29] Theo Thonat, Eli Shechtman, Sylvain Paris, and George Drettakis. Multi-view inpainting for image-based scene editing and rendering. *3DV*, 2016.

[30] Takashi Totsuka and Marc Levoy. Frequency domain volume rendering. In *Proceedings of SIGGRAPH*, 1993.

[31] Shubham Tulsiani, Richard Tucker, and Noah Snavely. Layer-structured 3D scene inference via view synthesis. *ECCV*, 2018.

[32] Zhou Wang, Alan Bovik, Hamid Sheikh, and Eero Simoncelli. Image quality assessment: from error visibility to structural similarity. In *IEEE Transactions on Image Processing*, 2004.

[33] Yair Weiss and William T. Freeman. What makes a good model of natural images? *CVPR*, 2007.

[34] Bo Yang, Zihang Lai, Xiaoxuan Lu, Shuyu Lin, Hongkai Wen, Andrew Markham, and Niki Trigoni. Learning 3d scene semantics and structure from a single depth image. *CVPR Workshops*, 2018.

[35] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *ICLR*, 2016.

[36] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. *CVPR*, 2018.

[37] Zhoutong Zhang, Yebin Liu, and Qionghai Dai. Light field from micro-baseline image pair. *CVPR*, 2015.

[38] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *ACM Transactions on Graphics (SIGGRAPH)*, 2018.

[39] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A. Efros. View synthesis by appearance flow. *ECCV*, 2016.

[40] C. Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High-quality video view interpolation using a layered representation. In *ACM Transactions on Graphics (SIGGRAPH)*, 2004.

[41] Matthias Zwicker, Wojciech Matusik, Fredo Durand, and Hanspeter Pfister. Antialiasing for automultiscopic 3D displays. In *Proceedings of Eurographics Symposium on Rendering*, 2006.

A. Supplementary Video

We have included a supplementary video that compares renderings from MPIs predicted by our model to those predicted by the original MPI method [S6] and the "Disocclusion Inpainting" baseline that inpaints disoccluded pixels in each rendering using a state-of-the-art deep learning approach [S5]. We invite readers to view this video for qualitative evidence of our method's ability to produce improved renderings with fewer depth discretization and repeated texture artifacts compared to the original MPI method, and with more convincing temporally-consistent disocclusions compared to the "Disocclusion Inpainting" baseline.

B. Section 3.2 Derivation Details

In this section, we provide additional details for the derivations in Section 3.2 of our main manuscript.

Figure S1 illustrates a 2D slice of the camera setup geometry where we hold the y dimension constant and view the xz-plane. An MPI in the frame of the reference camera (green dot) is viewed by a novel view camera (blue dot) at a translation (u, s) relative to the reference camera. x is the pixel coordinate of the red diffuse scene point on the blue camera's sensor plane, and x' is the pixel coordinate of the red scene point on the visualized MPI plane at disparity d. Note that the MPI plane pixel coordinate x' scales linearly with $1/d$, because each MPI plane RGBα image contains the same number of pixels sampled evenly within the camera frustum. It is straightforward to use the similar triangles of this diagram (above and to the right of the blue dot) to derive Equation 1 of our main manuscript:

$$r_{u,s}(x) = \sum_{d \in \mathcal{D}} c(x', d) = \sum_{d \in \mathcal{D}} c\left((1 - sd)\, x + ud, d\right) \tag{S1}$$

where $c(x, d)$ is the pre-multiplied RGBα at each pixel coordinate x and disparity plane d within the set of MPI disparity planes $\mathcal{D}$. Note that u and s are in units of pixels (such that the camera focal length $f = 1$), and we limit s to the range $-\infty < s < 1/d_{max}$ because renderings are not defined for viewpoints within the MPI volume. Additionally, note that the disparity d is in units $1/_{pixel}$.

We wish to study the limits of views rendered from an MPI, so let us consider a worst-case MPI with content in the subset of closest planes, for which we make a locally linear approximation to the coordinate transformation $(x, d) \rightarrow (x', d)$:

$$r_{u,s}(x) = \sum_{d \in \mathcal{D}} c\left((1 - sd_{max})\, x + ud, d\right) \tag{S2}$$

where d_{max} is a constant. Now, we have expressed the rendering of mutually-visible content as a sheared and dilated integral projection of the MPI. We apply the generalized Fourier slice theorem [S4] to interpret this integral projection of an MPI as a 2D slice through the 3D MPI's Fourier transform. Using operator notation, the generalized Fourier slice theorem can be expressed as:

$$\mathcal{F}^M \circ \mathcal{I}_M^N \circ \mathcal{B} \equiv \mathcal{S}_M^N \circ \frac{\mathcal{B}^{-\mathrm{T}}}{|\mathcal{B}^{-\mathrm{T}}|} \circ \mathcal{F}^N \tag{S3}$$

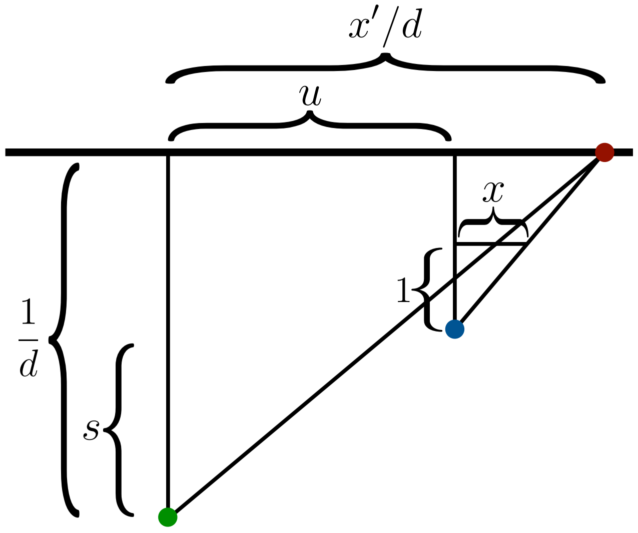

Figure S1. **Camera setup geometry.** An MPI in the frame of a reference camera (green dot) is viewed by a novel camera position (blue dot) at a translation (u, s) relative to the reference camera.

where $\mathcal{F}^M$ is the M-dimensional Fourier transform operator, $\mathcal{I}_M^N$ is the integral projection operator of an N-dimensional function to M dimensions by integrating out the last $N - M$ dimensions, $\mathcal{B}$ is a basis transformation operator (where $|\mathcal{B}^{-\mathrm{T}}|$ is the determinant of the inverse transpose of the transformation matrix), and $\mathcal{S}_M^N$ is the slicing operator that takes an M-dimensional slice from an N-dimensional function by setting the last $N - M$ dimensions to zero. The relevant values of the resulting transformation of MPI's Fourier transform, given the sheared and dilated MPI in Equation S2, are:

$$\begin{aligned} \mathcal{B} &= \begin{bmatrix} (1 - sd_{max}) & u \\ 0 & 1 \end{bmatrix} \\ \mathcal{B}^{-\mathrm{T}} &= \frac{1}{1 - sd_{max}} \begin{bmatrix} 1 & 0 \\ -u & (1 - sd_{max}) \end{bmatrix} \end{aligned} \tag{S4}$$

We use these values to express the Fourier transformation of our sheared and dilated MPI as:

$$C(k_{x'}, k_d) = C\left(\frac{k_x}{1 - sd_{max}}, \frac{-uk_x}{1 - sd_{max}} + k_d\right) \tag{S5}$$

where $C(k_x, k_d)$ is the Fourier transform of $c(\mathbf{x}, d)$. We omit the $1/|\mathcal{B}^{-\mathrm{T}}|$ term since it is simply a scaling factor and can be absorbed into the definition of C. Finally, we compute the resulting rendered view as the inverse Fourier transform of the slice taken from the MPI's Fourier transformation by setting $k_d = 0$:

$$r_{u,s}(x) = \mathcal{F}^{-1}\left\{ C\left(\frac{k_x}{1 - sd_{max}}, \frac{-uk_x}{1 - sd_{max}}\right)\right\} \tag{S6}$$

where $\mathcal{F}^{-1}$ is the inverse Fourier transform. This connects our detailed supplementary derivation back with Equation 3 in the main manuscript.

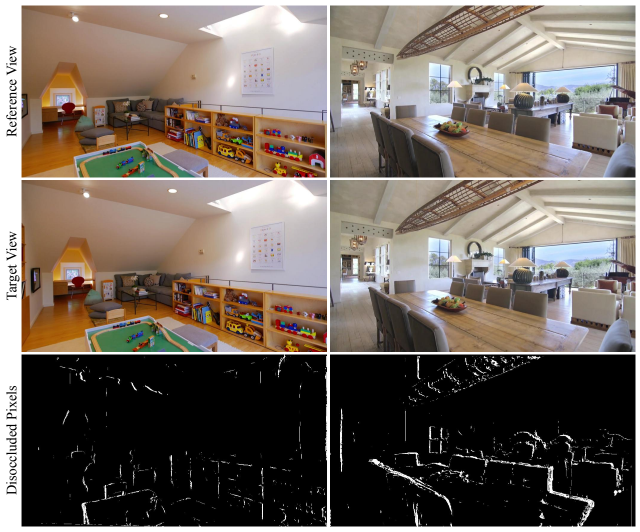

Figure S2. **Computed disocclusion masks.** We visualize example disocclusion masks for target view pixels that are occluded in the reference view, as used for our quantitative evaluations.

	Downsampling	
1-3	$(3\times3\times3$ conv, 8 features$)\times3$	$H\times W\times D\times 8$
4	$3\times3\times3$ conv, 16 features, stride 2	$H/2\times W/2\times D/2\times 16$
5-6	$(3\times3\times3$ conv, 16 features$)\times2$	$H/2\times W/2\times D/2\times 16$
7	$3\times3\times3$ conv, 32 features, stride 2	$H/4\times W/4\times D/4\times 32$
8-9	$(3\times3\times3$ conv, 32 features$)\times2$	$H/4\times W/4\times D/4\times 32$
10	$3\times3\times3$ conv, 64 features, stride 2	$H/8\times W/8\times D/8\times 64$
11-12	$(3\times3\times3$ conv, 64 features$)\times2$	$H/8\times W/8\times D/8\times 64$
13	$3\times3\times3$ conv, 128 features, stride 2	$H/16\times W/16\times D/16\times 128$
14-15	$(3\times3\times3$ conv, 128 features$)\times2$	$H/16\times W/16\times D/16\times 128$
	Bottleneck	
16	$3\times3\times3$ conv, 128 features, dilation rate 2	$H/16\times W/16\times D/16\times 128$
17	$3\times3\times3$ conv, 128 features, dilation rate 4	$H/16\times W/16\times D/16\times 128$
18	$3\times3\times3$ conv, 128 features, dilation rate 8	$H/16\times W/16\times D/16\times 128$
19	$3\times3\times3$ conv, 128 features	$H/16\times W/16\times D/16\times 128$
	Upsampling	
20	$2\times$ nearest neighbor upsample	$H/8\times W/8\times D/8\times 128$
21	concatenate 20 and 12	$H/8\times W/8\times D/8\times(128+64)$
22-23	$(3\times3\times3$ conv, 64 features$)\times2$	$H/8\times W/8\times D/8\times 64$
24	$2\times$ nearest neighbor upsample	$H/4\times W/4\times D/4\times 64$
25	concatenate 24 and 9	$H/4\times W/4\times D/4\times(64+32)$
26-27	$(3\times3\times3$ conv, 32 features$)\times2$	$H/4\times W/4\times D/4\times 32$
28	$2\times$ nearest neighbor upsample	$H/2\times W/2\times D/2\times 32$
29	concatenate 28 and 6	$H/2\times W/2\times D/2\times(32+16)$
30-31	$(3\times3\times3$ conv, 16 features$)\times2$	$H/2\times W/2\times D/2\times 16$
32	$2\times$ nearest neighbor upsample	$H\times W\times D\times 16$
33	concatenate 32 and 3	$H\times W\times D\times(16+8)$
34-35	$(3\times3\times3$ conv, 8 features$)\times2$	$H\times W\times D\times 8$
36	$3\times3\times3$ conv, 4 features (tanh)	$H\times W\times D\times 4$

Table S2. **3D CNN network architecture.** Our initial MPI prediction CNN Φ_1 uses the architecture described in the above table. Our final MPI prediction CNN Φ_2 uses the same architecture without the bottleneck dilated convolutional layers. All convolutional layers in Φ_1 and Φ_2 use a ReLu activation, except for the final layer. Φ_1 applies a tanh to all channels of the final layer, while Φ_2 just applies a tanh on one output channel (corresponding to α) and does not apply an activation to the predicted flows.

C. Network Architecture and Training Details

Table S2 contains precise specifications of the 3D convolutional neural network architecture described in Section 3.3 of the main manuscript.

We implement our system in TensorFlow [S1]. We train using the Adam algorithm [S2] for 300,000 iterations, with a learning rate of 2×10^{-4}, default parameters $\beta_1=0.9$, $\beta_2=0.999$, and a batch size of 1.

For our randomized-resolution training, we uniformly sample input PSV tensors with sizes [height, width, #planes, #channels] of any of the following:

$$\begin{array}{llll}
[\,576, & 1024, & 2^4, & 6] \\
[\,576/2, & 1024/2, & 2^5, & 6] \\
[\,576/4, & 1024/4, & 2^5, & 6] \\
[\,576/4, & 1024/4, & 2^6, & 6] \\
[\,576/4, & 1024/4, & 2^7, & 6] \\
[\,576/8, & 1024/8, & 2^5, & 6] \\
[\,576/8, & 1024/8, & 2^6, & 6] \\
[\,576/8, & 1024/8, & 2^7, & 6]
\end{array}$$

At test time, we apply this network on input PSV tensors with size [576,1024,2^7,6], which has the maximum spatial and depth resolutions seen during training.

D. Disocclusion Mask Examples

Figure S2 visualizes disocclusion masks computed by our method (Equation 12 in the main manuscript) between pairs of reference and target viewpoints.

E. "r_{init} + Adversarial Disocclusions" Details

As described in our main manuscript, "r_{init} + Adversarial Disocclusions" is a two-step MPI prediction strategy we use as a baseline for comparisons. It uses an identical Φ_1 to predict the initial MPI, but Φ_2 directly predicts RGBα layers instead of flows, and we apply an adversarial loss to each final rendered target image to encourage realistic disocclusions. We adopt the SN-PatchGAN discriminator architecture with spectral normalization [S3] and a hinge loss objective function, as proposed in [S5]. We add the adversarial loss to our main objective (Equation 10 in the main manuscript) with a weight $\lambda=5.0$.

References

[S1] Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015.

[S2] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.

[S3] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *ICLR*, 2018.

[S4] Ren Ng. Fourier slice photography. *In ACM Transactions on Graphics (SIGGRAPH)*, 2005.

[S5] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Generative image inpainting with contextual attention. *CVPR*, 2018.

[S6] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *In ACM Transactions on Graphics (SIGGRAPH)*, 2018.