

Master's Thesis

Learning-Based Approach for Effective Image
Depth Peeling

Jaemin Park

The Graduate School

Sungkyunkwan University

Department of Computer Science and Engineering

M a s t e r ' s T h e s i s

Learning-Based Approach for Effective Image

Depth Peeling

2 0 2 0

Jaeh Park

Master's Thesis

Learning-Based Approach for Effective Image
Depth Peeling

Jaemin Park

The Graduate School

Sungkyunkwan University

Department of Computer Science and Engineering

Learning-Based Approach for Effective Image Depth Peeling

Jaein Park

A Master's Thesis Submitted to the Department of Computer Science and
Engineering and the Graduate School of Sungkyunkwan University in partial
fulfillment of the requirements for the degree of Master of Science in Engineering

October 2025

Supervised by

Sungkil Lee

Major Advisor

This certifies that the Master's Thesis
of Jaein Park is approved.

_____ [signature]
Committee Chair : Jaemin Jo

_____ [signature]
Committee Member : Jaepil Heo

_____ [signature]
Major Advisor : Sungkil Lee

The Graduate School
Sungkyunkwan University

December 2025

Table of Contents

Chapter 1. Introduction	1
Chapter 2. Related Work	3
2.1. Single-Image Novel View Synthesis	3
2.2. Layered Scene Representations	4
2.3. Kernel-driven Image Reconstruction	5
Chapter 3. Virtually Visible Hidden Volume (VVHV)	6
Chapter 4. Method	8
4.1. Input Feature Gathering	9
4.2. Structural Feature Extractor	11
4.3. Differentiable Spatial Sampling	12
4.4. Kernel Weight Prediction via Attention	14
4.5. Loss Function	15
4.6. Compute Shader Integration	18
4.7. Image-based ray tracer for view warping	20
Chapter 5. Comparisons and Results	22
5.1. Dataset and Implementation	22
5.2. Comparison with Previous Methods	23
5.3. Ablation Studies	26
Chapter 6. Conclusion	27
References	28

List of Tables

Table 1.	24
----------	-------	----

List of Figures

Figure 1.	1
Figure 2.	8
Figure 3.	11
Figure 4.	25
Figure 5.	26

Abstract

Learning-Based Approach for Effective Image Depth Peeling

We present a real-time method for multiview synthesis from a single input image. Our approach predicts a depth-peeling-like layered scene representation directly from the image, enabling view-dependent reconstruction without explicit 3D geometry. Previous layer-based approaches attempt to recover full scene geometry or hallucinate entire backgrounds, which are computationally expensive for real-time applications. We define the concept of virtually visible hidden volumes (VVHVs) that geometrically delimit the hidden regions in the source view that become visible in the bounded target views, serving as an effective visibility domain for reconstruction. The layered representation is inferred using a kernel-predicting neural network, which reconstructs occluded regions as a weighted aggregation of visible areas in the input view. Furthermore, we employ a multi-view image-warping scheme optimized for both the layered scene representation and the bounded set of novel views, enabling fast and consistent multi-view synthesis. Our real-time solution achieves image quality comparable to that of offline single-image novel view synthesis methods, demonstrating both high efficiency and visual fidelity.

Keywords : Depth Peeling , Visibility , View Synthesis

Chapter 1. Introduction

Synthesizing novel views from a single input image has long been a fundamental challenge at the intersection of computer vision and graphics. The single-image setting is particularly under-constrained: while monocular depth estimators can provide approximate geometry, the truly disoccluded regions remain invisible and must be inferred. Existing approaches often attempt to reconstruct full scene geometry or hallucinate entire backgrounds, but such strategies are typically too heavy to achieve real-time performance. Recent advances in explicit scene-level representations have demonstrated strong rendering quality by modeling detailed geometric structures from multiple views. However, these approaches generally rely on dense multi-view observations and extensive optimization to recover complete scene geometry, which limits their applicability in the single-image setting and in real-time scenarios.

Our key insight is that novel view synthesis requires reconstructing only the disoccluded regions that become visible within a given set of target views. Thus, the problem can be reformulated as identifying these regions and inferring which source pixels should serve as references for their reconstruction. To this end, we introduce the concept of *Virtually Visible Hidden Volumes (VVHV)*, which generalizes Potentially Visible Hidden Volumes (PVHV) [Kim and Lee(2023)] to the single-image setting. VVHV provides a principled way to localize disocclusion regions and select reference pixels, yielding a memory-efficient scene representation in the form of depth peeling layers. Building on this idea, we propose *Neural Depth Peeling (NDP)*, a lightweight network that synthesizes hidden layers by expressing each occluded fragment as a weighted combination of visible pixels in the source view.

This paper presents a real-time pipeline that integrates VVHV detection with kernel-based reconstruction. Input features are gathered directly from shaders, structural cues are extracted using pixel-adaptive convolution, and attention-based kernels are employed to selectively synthesize new depth peeling layers. To efficiently render multiple viewpoints, we further introduce an optimized multi-view warping scheme that jointly refines reprojection across bounded target views while maintaining temporal and spatial consistency. With only about 10K trainable parameters, our design supports batch inference over a set of target views at real-time rates.

- We introduce **Virtually Visible Hidden Volume**, extending PVHV to the single-image case to localize disocclusion regions and their reference pixels.
- We propose **Neural Depth Peeling**, a novel single-image view synthesis framework that reconstructs hidden layers as weighted combinations of visible pixels.
- We design an **efficient multi-view warping scheme** that enables consistent and high-quality rendering across a bounded range of target viewpoints in real time.

Chapter 2. Related Work

2.1. Single-image Novel View Synthesis

Single-image NVS has been approached via warping, geometry-based rendering, and strong generative priors. Warping methods predict dense correspondences to directly remap input pixels into the target view, e.g., Appearance Flow [Zhou et al.(2016)]. Geometry-driven approaches reconstruct an explicit proxy (e.g., a point cloud from monocular depth) for differentiable rendering and refinement, as in SynSin [Wiles et al.(2020)]. Recent works leverage powerful priors to hallucinate unseen content conditioned on the target pose, including GAN-based scene synthesis [Koh et al.(2023)] and pose-guided diffusion models [Tseng et al.(2023)]. Transformers and diffusion have further advanced single-image NVS with stronger long-range reasoning and multi-view consistency (e.g., NViST [Jang and Agapito(2024)], MultiDiff [Muller et al.(2024)]). These directions validate the feasibility of single-image NVS but typically infer full-scene content or depend on heavy generative priors. In contrast, we target real-time efficiency by selectively reconstructing only disoccluded regions via neural depth peeling.

2.2. Layered Scene Representations

Layered scene models—especially multiplane images (MPI)—offer an efficient trade-off between quality and speed for view synthesis from sparse or single views. The first single-view MPI predictor [Tucker and Snavely(2020)] established the paradigm of regressing layered color–alpha planes from a single image. Subsequent methods improved plane allocation and content completion: AdaMPI [Han et al.(2022)] adapts plane placements, SinMPI [Pu et al.(2023)] augments MPIs with synthesis-aware outpainting, and TMPI [Khan et al.(2023)] tiles MPIs for practical scalability. Self-improving strategies (SIMPLI [Solovev et al.(2023)]) and high-efficiency layered depth systems (Quark [Flynn et al.(2024)]) further enhance robustness and throughput.

Beyond MPI, layered reasoning has also been studied from the perspective of *depth peeling*, which interprets multi-view warping through successive occlusion layers. For example, the PVHV formulation [Kim and Lee(2023)] defines hidden volumes that become visible only under certain novel view directions, effectively treating depth peeling as a selective multi-view culling process. This line of work highlights that layered scene representations need not reconstruct the entire volume; instead, peeling can focus on fragments that are virtually visible. Our approach follows this principle, but extends it to the single-image setting by introducing the concept of Virtually Visible Hidden Volumes (VVHV), and performing selective inference with neural depth peeling.

2.3. Kernel-driven Image Reconstruction

Our method represents invisible occluded regions as weighted combinations of visible pixels in the source view. A number of works reconstruct missing or refined content by explicitly modeling each output pixel as a weighted sum of nearby source pixels. *Shepard Convolutional Neural Networks* [Ren et al.(2015)] generalized classical Shepard interpolation by learning spatially adaptive kernels, enabling effective handling of irregular sampling and image completion. *Kernel-Predicting Neural Shadow Maps* [Hu et al.(2025)] applied a similar philosophy in the rendering domain: per-pixel kernels are predicted to blend neighboring samples, producing realistic soft shadows with high efficiency.

Chapter 3. Virtually Visible Hidden Volume (VVHV)

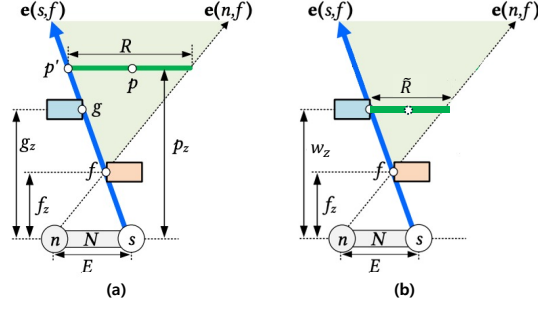


Figure 1: (a) PVHV, (b) VVHV

One of our primary objectives is to achieve real-time scene reconstruction with minimal memory overhead. Rather than reconstructing the entire hidden geometry, we adopt the idea of Potentially Visible Hidden Volumes (PVHV) [Kim and Lee(2023)], which restricts reconstruction to fragments that may actually appear in novel views, thereby pruning redundant geometry and reducing computation.

PVHV is defined through the notion of a visibility radius. For a hidden fragment p occluded by a visible fragment f , the radius is given as

$$R(p, f) = \left(\frac{p_z - f_z}{f_z} \right) E, \quad \text{where } p_z > f_z, \quad (1)$$

where p_z and f_z denote the depths of p and f , respectively, and E denotes the novel-view bound radius. A fragment belongs to PVHV if it satisfies the visibility test [Kim and Lee(2023)]:

$$H_s(N) = \{p \in W \mid p_z > f_z, |p - p'| < R(p, f), (p - p') \cdot (n - s) < 0\}, \quad (2)$$

where W is the set of candidate hidden fragments, p' is the projection of p in the novel view direction, $n \in N$ is a novel view in the target set, and s is the source view.

In our single-image setting, however, the true hidden depth p_z of occludee fragments is not available. We therefore extend the concept to *Virtually Visible Hidden Volumes (VVHV)*, which represent the regions hidden in the source view but hypothesized to be visible in a novel view.

Suppose that a sampled pixel w , obtained around an occlusion edge, is judged to belong to the same underlying object surface as a hidden fragment p . In this case, w can serve as a VVHV reference for p . Such a relationship provides object-aware cues: the features of w can be exploited when reconstructing p .

Moreover, this assumption justifies treating the depth of w (w_z) as a plausible estimate of the hidden depth p_z . Although p_z cannot be directly observed in our single-image setting, the geometric consistency between w and p supports the approximation $w_z \approx p_z$.

Formally, let Ω denote the set of candidate samples around occlusion edges, and let $w \in \Omega$. The virtual visibility radius is then defined as

$$\tilde{R}(f, w) = \left(\frac{w_z - f_z}{f_z} \right) E, \quad (3)$$

In this way, without explicit scene geometry, VVHV not only specifies where disocclusion must be reconstructed, but also identifies which pixels can provide valid features to guide the reconstruction. This dual role enables selective and object-aware disocclusion inference while reducing both runtime and memory for real-time novel view synthesis.

Chapter 4. Method

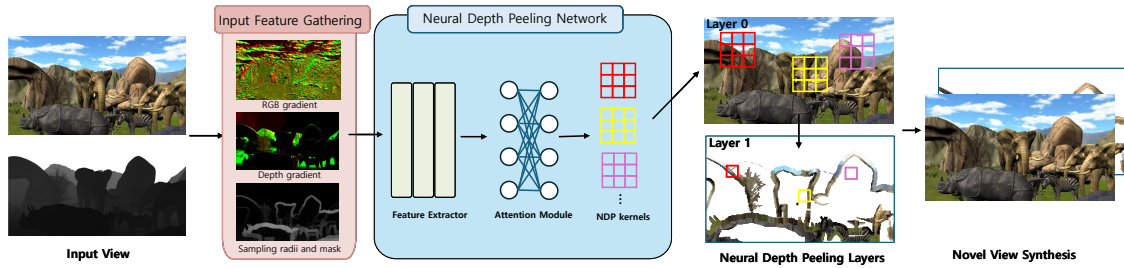


Figure 2: The Overall Framework of Neural Depth Peeling (NDP).

In this section, we describe the Neural Depth Peeling (NDP) pipeline in detail. Figure 2 illustrates the overall framework. The key idea is to represent invisible occluded regions as weighted combinations of visible pixels in the source view. We first gather input features directly from shaders and use them to construct structural features for each target pixel to be depth-peeled. Based on these features, the network predicts attention weights over neighboring samples, which are trained to reconstruct the corresponding ground-truth peeled layer by minimizing a reconstruction loss between the synthesized and reference views. The weighted aggregation guided by the learned attention produces the second depth peeling layer. Overall, the network is trained end-to-end with this objective and forms a lightweight architecture with only about 10K trainable parameters.

4.1. Input Feature Gathering

We construct three types of feature maps from a single RGB-D input, together with a VVHV mask:

- an RGB-D map (I, D) , which serves as the first layer in our depth peeling pipeline. I denotes the raw color image and D is the depth map estimated by the state-of-the-art monocular depth estimator Depth Anything V2 [Yang et al.(2024)].
- an RGB-D gradient map G , which captures local structure and texture. Unlike conventional gradient operators that produce edges on both sides of discontinuity, we assign both RGB and depth edges only to the pixel with smaller depth, i.e., the occluder side. The motivation is to encourage the network to interpret edges as belonging to the foreground object, rather than mistakenly associating them with background features. For the RGB edge, we first convert the image to grayscale and then compute a two-channel edge map.
- an search radius map R , which encodes the search radius for VVHV sampling. Since the true hidden depth p_z is unknown in the VVHV formulation, we take the local maximum w_z obtained from the N-buffer [D  coret(2005)] as a substitute, which provides the most conservative bound.
- an VVHV mask M , which identifies regions requiring disocclusion. The network performs inference only on masked pixels, significantly reducing memory consumption.
- a novel view set radius E , which scales the VVHV test for sampled pixels within the network.

All the above feature maps are computed in the shader stage and then passed to the neural depth

peeling network as inputs.

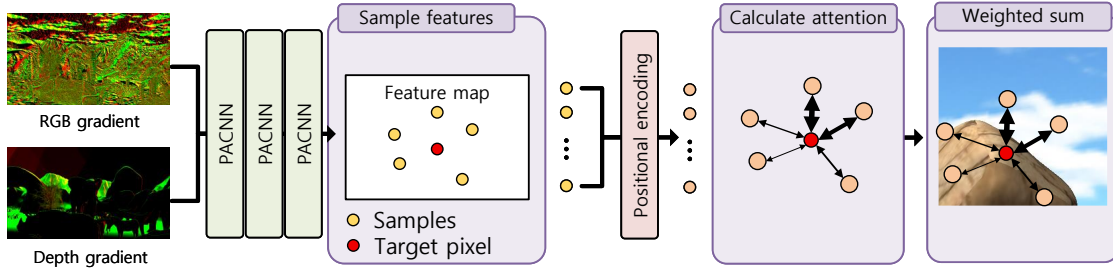


Figure 3: The structure of the NDP Network.

4.2. Structural Feature Extractor

As illustrated in Fig. 3, the feature extractor takes RGB-D edge maps as input and employs a pixel-adaptive convolutional neural network (PACNN) [Su et al.(2019)]. The motivation for adopting PACNN is that convolution weights can adapt not only to spatial proximity in screen space, but also to the distance between pixels along the depth dimension, thereby preventing the gradients of geometrically distant pixels from being inadvertently mixed. We intentionally exclude raw RGB-D values and instead use only edge maps, so that the network remains invariant to color variations and focuses purely on structural cues. Gradients from each channel are processed independently by applying groupwise convolutions. The receptive field is adapted to the radius of the target novel view set by controlling both the number of stacked layers and the dilation used in PACNN. Outputs from all layers are concatenated channel-wise, allowing features at multiple receptive field sizes to be aggregated. Finally, these structural features are concatenated with the raw RGB-D input to form the final feature embedding F .

4.3. Differentiable Spatial Sampling

Our method reconstructs occluded regions by predicting a per-pixel reconstruction kernel. However, directly considering all valid sampling regions would require memory proportional to the number of pixels multiplied by the size of the valid area, which is computationally prohibitive. To address this issue, we collect a fixed number of samples within the valid region and use them as reconstruction candidates.

Sampling inherently poses challenges for gradient propagation, making it difficult to train the sampling process directly. Inspired by the variational autoencoder (VAE) [Kingma and Welling(2022)] framework, we design a learnable sampler that introduces a differentiable stochastic sampling mechanism.

Given a feature map $F(x, y)$ and a sampling radius $R(x, y)$, the sampler predicts four parameters for each pixel: sampling offsets $(\Delta x, \Delta y)$ and their uncertainty scales (σ_x, σ_y) . These parameters are computed as:

$$(\Delta x, \Delta y, \sigma_x, \sigma_y) = R(x, y) \cdot (\tanh(f_x(F)), \tanh(f_y(F)), \text{sigmoid}(f_{\sigma_x}(F)), \text{sigmoid}(f_{\sigma_y}(F))) \quad (4)$$

Here, $f_x(\cdot)$, $f_y(\cdot)$, $f_{\sigma_x}(\cdot)$, and $f_{\sigma_y}(\cdot)$ denote small MLPs that project the local feature vector $F(x, y)$ into scalar fields corresponding to the x-offset, y-offset, and their respective variance parameters. The hyperbolic tangent (\tanh) limits the offsets within $[-1, 1]$, while the sigmoid (σ) constrains the variance to $[0, 1]$. The radius term $R(x, y)$ acts as a scaling factor that prevents the

network from sampling excessively far or sparsely.

Following the VAE-inspired reparameterization idea, the model samples stochastic offsets from a uniform distribution within $[-1, 1]$ during each iteration and computes the final sampling coordinates centered at (x, y) as:

$$(\tilde{x}, \tilde{y}) = (x, y) + (\Delta x, \Delta y) + (\sigma_x \odot \epsilon_x, \sigma_y \odot \epsilon_y), \quad \epsilon_x, \epsilon_y \sim \mathcal{U}(-1, 1) \quad (5)$$

This stochastic formulation enables differentiable gradient flow through the sampling process, allowing the network to learn adaptive and spatially coherent sampling strategies while keeping the sampling range bounded.

In summary, the learnable sampler effectively balances efficiency and adaptability, providing a trainable mechanism to select meaningful sampling points for high-quality reconstruction without excessive memory overhead.

4.4. Kernel Weight Prediction via Attention

Once the structural feature F is obtained, we concatenate it with the raw RGB-D input and sample pixels around each target location (e.g., 64 candidates). These sampled pixels are potential references that may contribute to reconstructing disoccluded regions. Pixels w that do not pass the VVHV test, determined by comparing depths, are masked out and excluded from serving as references.

For each remaining sample, a feature vector is extracted and projected into query, key, and value representations. The query is obtained by projecting the feature of the target pixel, while the keys are generated from the projected features of the sampled pixels, and the values correspond to their raw RGB-D values.

When the number of valid samples is insufficient, noisy patterns may appear. To mitigate this, we first compute a weighted sum with the available samples. The same weights are then applied to combine their feature vectors, which are subsequently used for a second sampling step with a smaller kernel size.

4.5. Loss Functions

Our training objective consists of multiple complementary losses that jointly supervise both the reconstruction network and the learnable sampler. Specifically, we employ pixel-wise losses (MSE and SSIM), a perceptual loss (VGG), an edge consistency loss, and a visibility consistency loss. To further encourage stable learning of the sampler, these losses are applied to both the kernel-weighted reconstruction results and the uniformly weighted reconstructions, where all sample weights are equal.

Overall Objective. The total loss is defined as

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{pix}} + \mathcal{L}_{\text{perc}} + \lambda_{\text{edge}} \mathcal{L}_{\text{edge}} + \lambda_{\text{vis}} \mathcal{L}_{\text{vis}}, \quad (6)$$

where λ_{edge} and λ_{vis} control the relative weights of the corresponding regularization terms.

Pixel-wise Losses. For each reconstruction type $v \in \{\text{ker}, \text{uni}\}$, we compute a combination of mean squared error (MSE) and structural similarity (SSIM) losses with respect to the ground-truth image:

$$\mathcal{L}_{\text{pix}} = \frac{1}{2} \sum_{v \in \{\text{ker}, \text{uni}\}} \left(\mathcal{L}_{\text{MSE}}^{(v)} + \lambda_{\text{ssim}} \mathcal{L}_{\text{SSIM}}^{(v)} \right), \quad (7)$$

where $\mathcal{L}_{\text{MSE}}^{(v)} = \|\hat{I}^{(v)} - I^*\|_2^2$ and $\mathcal{L}_{\text{SSIM}}^{(v)}$ measures local structural similarity.

Perceptual Loss. To enhance perceptual fidelity and global texture consistency, we compute a VGG-based feature loss on both reconstruction types:

$$\mathcal{L}_{\text{perc}} = \frac{1}{2} \sum_{v \in \{\text{ker}, \text{uni}\}} \sum_l \|\phi_l(\hat{I}^{(v)}) - \phi_l(I^*)\|_1, \quad (8)$$

where $\phi_l(\cdot)$ denotes the activation of the l -th layer of a pretrained VGG network.

Edge Consistency Loss. The edge consistency loss enforces structural continuity across occlusion boundaries by encouraging gradient flux to be preserved, so that the amount of gradient flowing into a pixel matches the amount flowing out:

$$\mathcal{L}_{\text{edge}} = \frac{1}{|\Omega|} \sum_{p \in \Omega} \rho_p \|\nabla \hat{I}_p^{\text{in}} - \nabla \hat{I}_p^{\text{out}}\|_2^2. \quad (9)$$

Since discontinuities are most visible near disocclusion boundaries, ρ_p emphasizes such regions and is estimated as the ratio of VVHV samples among all samples cast around pixel p . Occluder pixels are excluded from gradient computation to avoid spurious edges.

Visibility Consistency Loss. To ensure that reconstructed pixels correspond to geometrically valid regions, we introduce a visibility consistency loss that penalizes samples reconstructed outside the Virtually Visible Hidden Volume (VVHV). The set of valid pixels for a surface fragment f with normal n and view direction s is defined as

$$H_s(N) = \{p \in W \mid p_z > f_z, |p - p'| < R(p, f), (p - p') \cdot (n - s) < 0\}, \quad (10)$$

where p' denotes the projected position of the reference fragment, and $R(p, f)$ is the local sampling radius. Pixels that do not satisfy this condition are assigned a penalty proportional to their geometric violation:

$$\mathcal{L}_{\text{vis}} = \frac{1}{|W|} \sum_{p \in W} \max(0, \alpha - \psi(p, f, n, s)), \quad (11)$$

where $\psi(p, f, n, s)$ measures the signed distance of pixel p from the valid VVHV domain (e.g., based on the left-hand side of the inequalities in Eq. (9)), and α controls the tolerance margin. This term suppresses spurious reconstructions outside the view-dependent visibility range, stabilizing both the sampler and the kernel predictor.

Multi-resolution Supervision. All loss terms are evaluated at multiple spatial resolutions, and the final objective is obtained by averaging across scales:

$$\mathcal{L}_{\text{final}} = \frac{1}{S} \sum_{s=1}^S \mathcal{L}_{\text{total}}^{(s)}. \quad (12)$$

This hierarchical supervision improves stability and promotes both coarse and fine-scale structural consistency.

4.6. Compute Shader Integration

The feature extractor in our framework follows a conventional U-Net architecture, which can be efficiently deployed using existing inference frameworks such as TensorRT and directly embedded into the rendering pipeline. However, the subsequent stages—sampling and reconstruction—are executed only for pixels that require inpainting within the Virtually Visible Hidden Volume (VVHV) region. Since these operations involve sparse and spatially varying workloads, they are not well suited for standard tensor-based inference engines. To address this, we embed the sampling and MLP computation processes directly into a GPU compute shader.

Pipeline Overview. We first identify the regions that require reconstruction by performing a VVHV test entirely on the GPU. Pixels that pass this test are marked as reconstruction candidates. Then, we launch an *indirect compute dispatch* that dynamically allocates one workgroup per valid pixel. This per-pixel dispatch strategy ensures that compute resources are used only where necessary, avoiding idle execution on fully visible regions.

Per-Pixel Workgroup Computation. Within each workgroup, the following sequence is executed:

1. **Sampler MLP evaluation.** The per-pixel MLP corresponding to the learnable sampler predicts sampling offsets and uncertainties using the local feature vector from the feature extractor.
2. **Poisson sampling.** A Poisson-disc pattern is generated within the valid sampling radius to ensure spatially uniform and non-overlapping samples around the pixel center.

3. **Parallel sample processing.** Each sample is assigned the same number of threads within the workgroup, allowing per-sample operations such as feature lookup and kernel weight computation to run fully in parallel.
4. **Kernel accumulation.** The weighted contributions from all samples are aggregated to reconstruct the target pixel intensity.

Advantages. This design enables fully parallelized reconstruction on the GPU while avoiding the overhead of dense tensor evaluation. By integrating the learnable sampler and kernel predictor directly into a compute shader, we achieve high efficiency and minimal latency, supporting real-time inference even for view-dependent reconstruction tasks.

4.7. Image-based ray tracer for view warping

We extend the concept of lens-based ray tracing proposed by [Lee et al.(2010)] and reformulate it as an image-based ray tracer for view warping rather than optical simulation. In our formulation, each lens sample corresponds to a ray originating from the novel-view camera position, and these rays traverse the layered depth images to reconstruct view-dependent pixel colors. Instead of computing physical refraction through a geometric lens, our method performs ray-layer intersections that capture the geometric relationships implied by the layered depth representation. This enables high-quality view reconstruction comparable to volumetric ray tracing while preserving the computational efficiency of texture-space processing.

Each target pixel in the novel view emits a virtual ray that advances step by step in screen space while sampling from the multi-layer depth texture. At each step, the ray tests whether it intersects one of the available layers and projects the potential hit point back to image space to verify spatial consistency with the current pixel coordinate. Once a valid intersection is confirmed, the color associated with that layer is fetched to reconstruct the warped pixel.

Discrete traversal alone can miss thin or highly slanted surfaces between neighboring pixels. To handle such discontinuities, we introduce a cross-layer correction that compares adjacent rays. When two neighboring rays lie on opposite sides of the same layer’s depth, the missing intersection is interpolated between them, effectively filling cracks and maintaining continuity across occlusion boundaries.

During each iteration, the algorithm evaluates the available layers in front-to-back order, skipping empty or invalid entries to reduce computation. When a valid hit is found, the corresponding color

is retrieved from the RGBZ texture and the ray terminates. Because each ray is independent, the entire process is highly parallelizable and maps efficiently to compute-shader execution, achieving real-time performance.

Chapter 5. Comparisons and Results

5.1. Dataset and Implementation

Dataset. We collected our virtual dataset from scenes composed of 3D models obtained from rendering resource websites. For each scene, we created 100 different settings. Each rendering setting was defined by a randomly sampled camera transform, with constraints ensuring that the scene geometries were neither too close nor too far from each other. This was important because each scene needed to allow the occluded regions to be inferred from the visible regions.

For every rendering setting, we generated both the input features and the ground-truth image. The input features were produced through the rendering pipeline using shader-based feature extraction, while the ground-truth images were generated as the second layer of Effective Depth Peeling, following the method of [Kim and Lee(2023)]. Since the dataset is constructed from synthetic scenes, a domain gap with respect to real-world images remains, which may affect generalization performance and is discussed as a limitation of our approach.

Implementation. The shaders used for rendering, as described in Section 3, were implemented using OpenGL. Our feature extractor and kernel-predicting network were trained with PyTorch using the Adam optimizer. After training, the feature extractor was deployed with TensorRT for fast inference, and half-precision computation was employed for further acceleration. As mentioned earlier, the kernel-predicting network was embedded through a compute shader.

5.2. Comparison with Previous Methods

Baselines. We compare our approach with reproducible state-of-the-art methods for single-image novel view synthesis, including AdaMPI [Han et al.(2022)], TMPI [Khan et al.(2023)], and SinMPI [Pu et al.(2023)]. A common characteristic shared by these methods, including ours, is that the final output is generated based on the result of monocular depth estimation.

Datasets. We evaluate the quality of the rendered novel views on the Local Light Field Fusion (LLFF) dataset [Mildenhall et al.(2019)]. Since this dataset provides multi-view data, it enables quantitative comparisons using the provided ground-truth images.

Evaluation Protocol. To ensure a fair comparison, all methods—including ours—use input images of the same resolution, and their depths are estimated with the same pre-trained model. Similar to SinNeRF [Xu et al.(2022)], the depth maps for the LLFF dataset were predicted using a pre-trained 3D Gaussians [Kerbl et al.(2023)]. Although all methods are capable of generating novel views using monocularly predicted depths, such as those produced by DPT [Ranftl et al.(2020)] or DepthAnythingv2 [Yang et al.(2024)], accurate evaluation requires that the predicted depth be consistent with the true scene scale, ensuring that the synthesized views are properly aligned with the ground-truth images.

Evaluation Metrics. We evaluate performance using standard image quality metrics commonly employed in prior work, including SSIM, PSNR, and LPIPS [Zhang et al.(2018)].

5.2.1 Quantitative Comparison

Table 1: Quantitative comparison on the LLFF dataset.

Method	SSIM \uparrow	PSNR \uparrow	LPIPS \downarrow	FPS
TMPI	to be measured			
SinMPI	to be measured			
AdaMPI	0.384	14.77	0.28	4
Ours	0.369	15.70	0.271	49

Table 1 compares to quantitative results of three different methods evaluated on the LLFF dataset. The reported values were measured only for the images shown in Fig. 3 and will be supplemented in future work. As shown in the table, our method achieves real-time frame rates while producing quality comparable to, or even surpassing, that of offline methods.

5.2.2 Qualitative Comparison

Figure 4 presents a qualitative comparison between our method and AdaMPI. While MPI-based methods can synthesize plausible views, they often suffer from artifacts such as layer blending errors, depth discretization, and ghosting near occlusion boundaries due to their reliance on fixed planar representations. In contrast, our approach more accurately handles continuous geometry and fine-grained occlusions, resulting in sharper reconstructions and more consistent parallax across viewpoints.

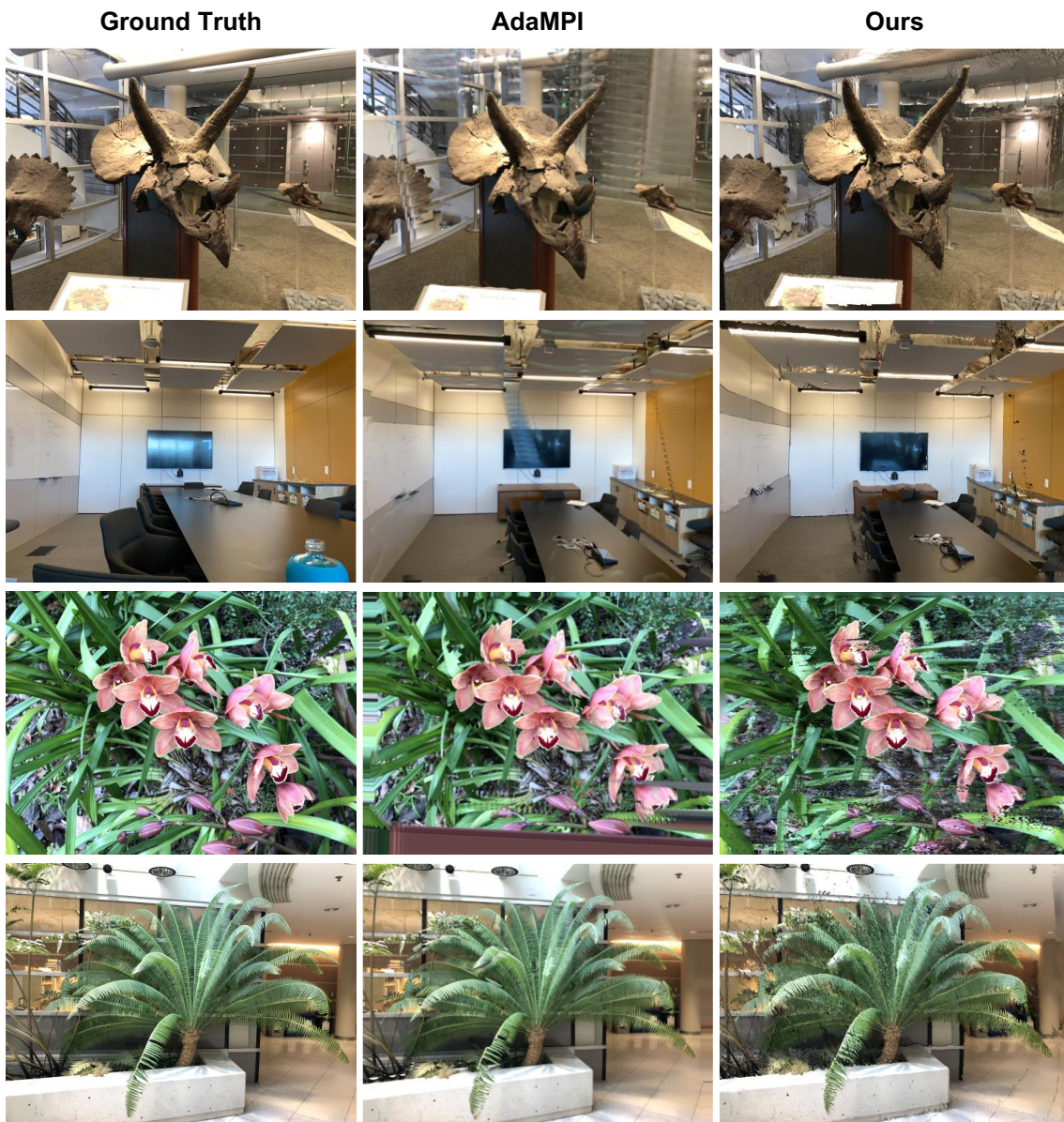


Figure 4: Qualitative comparison.

5.3. Ablation Studies

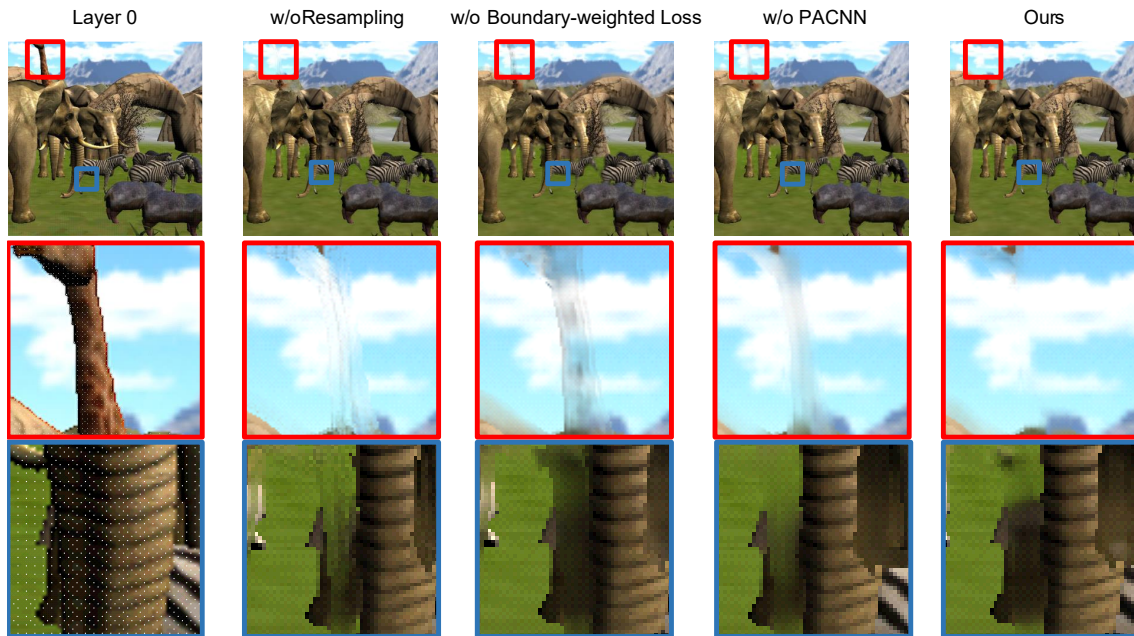


Figure 5: The ablation study on synthesized scene with 3D models.

Figure 5 illustrates the results of our ablation study. Resampling helps reduce aliasing artifacts in the reconstructed regions, while the boundary-weighted loss prevents the model from getting trapped in local minima. Additionally, the PACNN component contributes to a better understanding of the scene’s structural layout, leading to more coherent and stable reconstructions.

Chapter 6. Conclusion

We have introduced a real-time single-image multiview synthesis method that reconstructs novel views through a layered scene representation guided by virtually visible hidden volumes (VVHVs). By explicitly defining the visible domain for occluded regions, our approach connects geometry-aware formulations with image-based reconstruction, enabling efficient novel-view synthesis without explicit 3D modeling. While the model is trained using a combination of synthetic and real images, a remaining domain gap between training data and real-world inputs is acknowledged, and its impact on reconstruction quality and generalization is discussed; future improvements may be achieved by supervising the model directly on rendered novel views rather than intermediate depth peeling layers, and by incorporating losses based on pretrained feature extractors to better capture real-image statistics. A detailed quantitative comparison with recent generative approaches on real-image benchmarks is left as an important direction for future work. The proposed kernel-predicting reconstruction and multi-view warping scheme ensure spatial consistency across views, and the formulation naturally extends to frame-wise processing, suggesting applicability to video-based multiview synthesis. Overall, this work demonstrates that interactive single-image multiview synthesis with reasonable geometric plausibility is feasible, providing a practical foundation for lightweight real-time graphics and vision applications.

References

- [Décoret(2005)] Xavier Décoret. 2005. N-Buffers for Efficient Depth Map Query. *Computer Graphics Forum* 24 (09 2005). <https://doi.org/10.1111/j.1467-8659.2005.00864.x>
- [Flynn et al.(2024)] John Flynn et al. 2024. Quark: Scalable, Real-Time Neural View Synthesis with Layered Depth. *ACM Trans. Graph.* (2024).
- [Han et al.(2022)] Long Han, Kanglei Lin, et al. 2022. Single-view view synthesis in the wild with learned adaptive multiplane images. In *ACM SIGGRAPH Conference Proceedings*. ACM.
- [Hu et al.(2025)] Xuejun Hu, Jinfan Lu, Kun Xu, et al. 2025. Kernel Predicting Neural Shadow Maps. In *ACM SIGGRAPH Conference Proceedings*. ACM.
- [Jang and Agapito(2024)] Wonbong Jang and Lourdes Agapito. 2024. NViST: Transformer-based Single-Image Novel View Synthesis. In *CVPR*.
- [Kerbl et al.(2023)] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* 42, 4 (July 2023). <https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/>
- [Khan et al.(2023)] Numair Khan, Douglas Lanman, and Lei Xiao. 2023. Tiled Multiplane Images for Practical 3D Photography. *arXiv preprint arXiv:2309.14291* (2023).
- [Kim and Lee(2023)] Kim and Lee. 2023. Potentially visible hidden volumes for novel view synthesis. In *ACM SIGGRAPH Conference Proceedings*. ACM.

- [Kingma and Welling(2022)] Diederik P Kingma and Max Welling. 2022. Auto-Encoding Variational Bayes. arXiv:1312.6114 [stat.ML] <https://arxiv.org/abs/1312.6114>
- [Koh et al.(2023)] Jae-Youn Koh et al. 2023. Simple and Effective Synthesis of Indoor 3D Scenes. *AAAI* 37, 1 (2023), 1169–1178.
- [Lee et al.(2010)] Sungkil Lee, Elmar Eisemann, and Hans-Peter Seidel. 2010. Real-Time Lens Blur Effects and Focus Control. *ACM Trans. Graphics (Proc. SIGGRAPH'10)* 29, 4 (2010), 65:1–7.
- [Mildenhall et al.(2019)] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. 2019. Local Light Field Fusion: Practical View Synthesis with Prescriptive Sampling Guidelines. *ACM Transactions on Graphics (TOG)* (2019).
- [Muller et al.(2024)] Thomas Muller, Zhengqi Chen, et al. 2024. MultiDiff: Consistent multi-view image generation with diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.
- [Pu et al.(2023)] Xiaoyang Pu, Jingyang Zhang, et al. 2023. SinMPI: Novel view synthesis from a single image with expanded multiplane images. In *ACM SIGGRAPH Asia Conference Proceedings*. ACM.
- [Ranftl et al.(2020)] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. 2020. Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2020).

- [Ren et al.(2015)] Jimmy SJ Ren, Li Xu, Qiong Yan, and Wenxiu Sun. 2015. Shepard Convolutional Neural Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 28. 901–909.
- [Solovev et al.(2023)] Petr Solovev et al. 2023. Self-Improving Multiplane-To-Layer Images for Novel View Synthesis. In *WACV*.
- [Su et al.(2019)] Hang Su, Varun Jampani, Deqing Sun, Orazio Gallo, Erik Learned-Miller, and Jan Kautz. 2019. Pixel-Adaptive Convolutional Neural Networks. arXiv:1904.05373 [cs.CV] <https://arxiv.org/abs/1904.05373>
- [Tseng et al.(2023)] Hung-Yu Tseng, Qinbo Li, Changil Kim, Suhub Alsisan, Jia-Bin Huang, and Johannes Kopf. 2023. Consistent View Synthesis with Pose-Guided Diffusion Models. In *CVPR*. 13796~13806.
- [Tucker and Snavely(2020)] Richard Tucker and Noah Snavely. 2020. Single-view view synthesis with multiplane images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 551–560.
- [Wiles et al.(2020)] Olivia Wiles, Forrester Cole, Noah Snavely, and Andrew Zisserman. 2020. SynSin: End-to-end view synthesis from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 7467–7477.
- [Xu et al.(2022)] Dejie Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Humphrey Shi, and Zhangyang Wang. 2022. SinNeRF: Training Neural Radiance Fields on Complex Scenes from a Single Image. *arXiv preprint arXiv:2204.00928*.
- [Yang et al.(2024)] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi

Feng, and Hengshuang Zhao. 2024. Depth Anything V2. arXiv:2406.09414 [cs.CV]
<https://arxiv.org/abs/2406.09414>

[Zhang et al.(2018)] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*.

[Zhou et al.(2016)] Tinghui Zhou, Shubham Tulsiani, Weijie Sun, Jitendra Malik, and Alexei A. Efros. 2016. View synthesis by appearance flow. In *European Conference on Computer Vision (ECCV)*. Springer, 286–301.

논문요약

학습 기반의 효과적인 이미지 뎀스 필링 방법

박재인

소프트웨어학과

성균관대학교

본 논문에서는 단일 입력 이미지로부터 실시간 다중 시점 합성을 수행하는 방법을 제안한다. 제안하는 방법은 입력 이미지로부터 직접 깊이 필링(Depth Peeling) 형태의 다층(scene-layered) 표현을 예측함으로써, 명시적인 3D 기하 복원 과정 없이 시점 의존적인 재구성을 가능하게 한다. 기존의 레이어 기반 접근법들은 전체 장면 기하를 복원하거나 완전히 가려진 배경을 추론하려는 경향이 있으나, 이러한 방식은 실시간 응용에 비해 계산 비용이 매우 크다.

본 연구에서는 가상적으로 보이는 숨겨진 부피(Virtually Visible Hidden Volumes, VVHV) 개념을 정의하였다. 이는 원본 시점에서 가려져 있으나 제한된 범위의 목표 시점에서 드러나는 영역을 기하학적으로 한정하여, 효율적인 재구성 가시성 도메인을 제공한다.

다층 표현은 커널 예측 신경망(Kernel-Predicting Neural Network)을 통해 추론되며, 입력 이미지의 가시 영역으로부터 가려진 영역을 가중 합 형태로 복원한다. 또한, 제안된 방법은 다층 장면 표현과 제한된 목표 시점 집합에 최적화된 다중 시점 이

미지 워핑(Multi-view Image Warping) 방식을 활용하여 빠르고 일관된 시점 합성을 실현한다.

제안된 실시간 방법은 오프라인 단일 이미지 기반 시점 합성 기법들과 비교하여 유사하거나 더 우수한 화질을 달성하며, 높은 효율성과 시각적 품질을 동시에 입증한다.

주제어: 텍스처 필링, 가시성, 시점 합성