D-LaMa: Depth Inpainting of Perspective-Occluded Environments

Keywords: D-LaMa, Depth Inpainting, Occlusion Problem

Abstract: Occlusion is a common problem in computer vision where backgrounds or objects are occluded by other objects in the foreground. Occlusion affects object recognition or tracking and influences scene understanding with the associated depth estimation and spatial perception. To solve the associated problems and improve the detection of areas, we propose a pre-trained image distortion model that allows us to incorporate new perspectives within previously rendered point clouds. We investigate approaches in synthetically generated use cases: Masking previously generated virtual images and depth images, removing and painting over a provided mask, and the removal of objects from the scene. Our experimental results allow us to gain valuable insights into fundamental problems of occlusion configurations and confirm the effectiveness of our approaches. Our research findings serve as a guide to applying our model to real-life scenarios and ultimately solve the occlusion problem.

1 INTRODUCTION

In numerous application fields, such as 3D modeling, mixed reality, autonomous vehicle systems, and robotics, depth perception and the associated spatial perception are essential for localization, navigation, obstacle avoidance, and 3D mapping. For example, a moving robot needs to understand the full geometry of surrounding objects and scenes to make accurate predictions and decisions. The ability to navigate and interact in an environment requires depth perception that considers occlusion, relative height, relative size, perspective convergence, texture, and shadow gradients. Retrieving depth or geometry information from a single image is a challenge owing to the loss of depth that occurs when a 3D scene is projected onto a 2D image (Aharchi and Ait Kbir, 2020).

The problem of occlusion occurs when objects or parts of objects are obscured from the view of cameras or sensors by other elements in a scene (see Fig. 1). This challenge affects numerous computer vision tasks, including object recognition and detection, object tracking, scene understanding, and depth estimation. The respective occlusions impede understanding the scene by preventing the accurate perception of spatial relationships between objects.

Considerable efforts have been made to solve the occlusion problem and the resulting limitations in computer vision. On account of their precision and efficiency, neural networks such as Neural Rendering Field (NeRF) are a popular approach. For example, the NeRF model can be trained to learn the radiance (color and opacity) at each point of a 3D scene. Despite the considerable success of this method, multiple images from different viewpoints are often required, and the model must be retrained for each new presented scene. Additionally, the required training process is regularly lengthy and complex (Mildenhall et al., 2020; Munkberg et al., 2023).

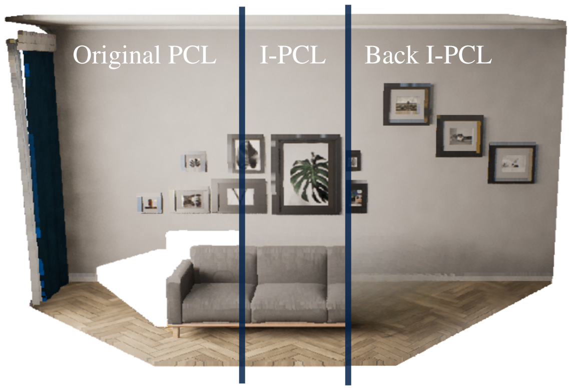

Figure 1: **Rendered Point Cloud of 3D Scene with Occlusion, Non-Occlusion and Reconstructed Background:** The original point cloud (PCL), on the left, contains white areas which convey the occlusion problem. The center illustrates the result of the inpainted point cloud (I-PCL) reconstructed using our D-LaMa model. The sofa can be hidden by combining PCL and I-PCL. The reconstructed backside is demonstrated on the right.

An alternative approach utilizes neural-based generation of semantic segmented 3D scenes using the image or depth map as input (Song et al., 2017). However, the color information is neglected since only the geometry and segmentation class of the scene are used for the rendering of the final 3D reconstruction. Similar approaches have been proposed to complete the 3D scene without the segmentation class information.

But until now, these likewise do not consider the color information (Firman et al., 2016).

A straightforward but also promising method for the occlusion problem is image inpainting, where the missing information within the image is masked and filled in (Yu et al., 2019; Nazeri et al., 2019; Yan et al., 2018). The mask content itself can be created from the visual context of hidden regions by deriving the color information from surrounding counterparts.

Fusing photogrammetry techniques, depth cameras, and neural network models can lead to promising approaches for the reconstruction of hidden geometries within an image scene. Particularly the potential of neural networks lends itself to the solution of occlusions. The specific requirements for such a model and related methods have yet to be determined. Therefore, our paper provides foundational insight into the associated requirements, limitations, and challenges to guide future research by providing a workable strategy to solve the problem of completing occlusions by image inpainting using a neural network model.

Our evaluation reveals the feasibility and transferability of our novel D-LaMa Model for reconstructing perspective-occluded environments, where a pretrained deep learning-based image inpainting model is utilized. Our findings extend research on the benefits of depth inpainting and contain the following major contributions:

- D-LaMa model for the completion of occluded surfaces and object-hidded backgrounds in point clouds
- Evaluation of depth image similarity by metrics of SSIM, LPIPS, and MSE for
  - Removal Virtual Projected Inpainting (VPI)
  - Object Removal Inpainting (ORI)
  - Stereoscopic Image Inpainting (SII)

Our results show the effectiveness of the D-LaMa model on the occlusion problem through practical applications. In VPI, we reveal problems caused by the properties of point clouds and lack of real-world attributes, such as reflections. In ORI, we refer to the effects on model results due to mask expansion during the inpainting process. Adverse artifacts occurred especially when the mask did not completely cover the object. The SII demonstrates remarkable proficiency in inpainting 3D point clouds within a stereoscopic setup. We identified an incoherence of the estimated depth image caused by the separate inpainting of the left and right images.

Our empirical examination is based on synthetically generated data from Unreal Engine. By opting for synthetic over real-world data, a better understanding of the challenges and influencing factors of our approaches in a controlled environment is ensured.

2 Occlusion Handling in Computer Vision

Over the last decade, significant research has been conducted in the field of computer vision to construct and render increasingly detailed 3D scenes. The challenges for reconstructing individual objects and overcoming data gaps are caused not only by occlusion but also by sensor and hardware limitations as well as the resulting noise (Müller, S., and Kranzlmüller, D., 2021; Müller, S., and Kranzlmüller, D., 2022).

The completion of object shapes in 3D reconstruction research has evolved from early interpolation (Edelsbrunner and Mücke, 1994; Bajaj et al., 1995; Chen and Medioni, 1995; Curless and Levoy, 1996; Amenta et al., 1998; Bernardini et al., 1999; Davis et al., 2002) and energy minimization (Sorkine and Cohen-Or, 2004; Kazhdan et al., 2006; Nealen et al., 2006) techniques to data-driven approaches leveraging symmetry (Pauly et al., 2008; Sipiran et al., 2014; Sung et al., 2015) and databases for geometric priors (Pauly et al., 2005; Shen et al., 2012; Li et al., 2015; Rock et al., 2015; Li et al., 2016).

With the advancement of neural networks, various 3D data representations such as voxels (Yan et al., 2016; Girdhar et al., 2016; Tatarchenko et al., 2017; Wu et al., 2016; Brock et al., 2016), point clouds (Fan et al., 2017; Yang et al., 2018; Lin et al., 2018; Insafutdinov and Dosovitskiy, 2018; Achlioptas et al., 2018), meshes (Groueix et al., 2018; Wang et al., 2018a; Chen and Zhang, 2019), implicit function representation (Park et al., 2019; Chen and Zhang, 2019; Mescheder et al., 2019; Peng et al., 2020), and structure-based representation (Zou et al., 2017; Li et al., 2017; Wu et al., 2020) have been explored to handle the object completion task.

Despite the promising results, challenges, such as low resolution of voxels, lack of geometric details in structure-based representations, complex learning processes for neural networks in point clouds and mesh topology, as well as additional post-processing stages for implicit representations still remain. Their applicability for dealing with occlusions within a 3D scene is limited, as additional processes are required to identify and generate the complete geometry for each object, resulting in a high computational cost.

Multiple studies recognized missing regions in a 3D scene as part of the task of scene completion. The proposed methods behind these studies use neural net-

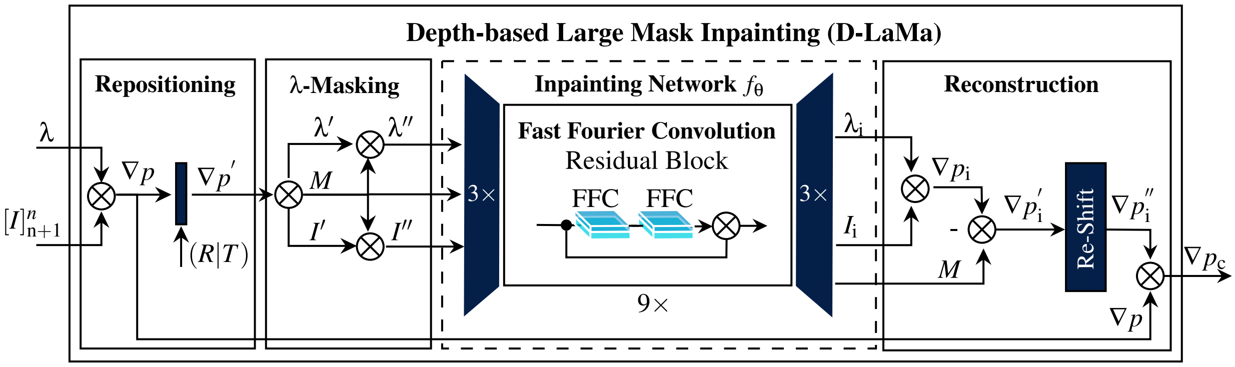

Figure 2: **Pipeline of Depth based LaMa-Architecture (D-LaMa):** The architecture consists of 4 components: In Repositioning, a new virtual point cloud is reprojected from a different perspective. λ-Masking generates a virtual projected mask by positional change of perspective. The network inpaints the masked disparity map and image in order to derive the occluded surface. By reconstructing the position and combining the inpainted point cloud (I-PCL) with the previous point cloud (PCL), a complete scene without occlusion can be reconstructed.

works to learn the complete scene geometry through voxel data representation (Song et al., 2017; Chen et al., 2019; Firman et al., 2016; Dai et al., 2018). However, the inclusion of color information is often neglected, resulting in a lack of visual fidelity. To overcome this limitation, further research has been conducted that utilizes the image inpainting approach by considering color and depth information for an effective scene completion with stereoscopic setup characteristics (Wang et al., 2008; Hervieu et al., 2010). A single view (He et al., 2011; Doria and Radke, 2012) is used to remove or inpaint occluding objects and to complete the occluded background region. Since these approaches rely on traditional inpainting methods with explicit mathematical operations and do not incorporate neural networks, they are computationally expensive and often exhibit an inferior quality.

In the field of novel view synthesis, research has been conducted to generate or render images from new perspectives. One popular approach utilizes NeRF to achieve realistic renderings of complex scenes (Mildenhall et al., 2021; Reiser et al., 2021; Müller et al., 2022; Rosinol et al., 2023). However, this method requires sophisticated hardware and requires complex post-processing steps to retrieve the final scene as a 3D model. Other alternatives, such as 3D warping methods acknowledging the above limitations offer straightforward options that are easier to implement. For instance, depth information can be used to render novel view images from a new viewpoint (Mark et al., 1997; Li et al., 2013; Li et al., 2018; Mori et al., 2009; Yao et al., 2019; Huang and Huang, 2020). Moreover, the concept of 3D warping combined with traditional image inpainting can be employed to fill in the holes in the novel view (Mori et al., 2009; Huang and Huang, 2020; Yao et al., 2019). Despite its promising conceptualization, the utilization of multiple initial viewpoints from the same scene limits the applicability of a generated scene, where only a single viewpoint is used.

Traditional methods of completing missing regions in 3D scenes are computationally expensive and often yield suboptimal results. The inclusion of neural networks shows promising results, but is often limited by neglected color information, leading to unrealistic 3D reconstructions. This behavior is noticeable in the novel view synthesis task, where the learning-based approach excels in quality but is complex. The utilization of multiple initial viewpoints is required to solve the novel view synthesis task in general. Despite this, the use of inpainting emerges as a promising solution to bridge this gap.

By applying the inpainting process to both color and depth images, we can effectively address the occlusion problem. However, relying solely on traditional inpainting methods without neural networks will result in inferior performance. In contrast to the previous approaches, our approach utilizes the simplicity of inpainting methods as well as the capability of neural network models to solve the problem of occlusion completeness.

3 Depth based LaMa-Architecture

Our concept (see Fig. 2) centers on the LaMa inpainting model (Suvorov et al., 2022), which uses advanced algorithms and neural network architectures to fill in missing or damaged parts in images.

3.1 Virtual Repositioning of Point Cloud

The first step of our Depth-based Large Mask Inpainting Architecture (D-LaMa) is the repositioning of rendered point clouds ∇p to identify spatially occluded surfaces. We achieve a virtually repositioned point cloud $\nabla p'$ by changing the perspective view of the original point cloud ∇p, as shown in Fig. 3.

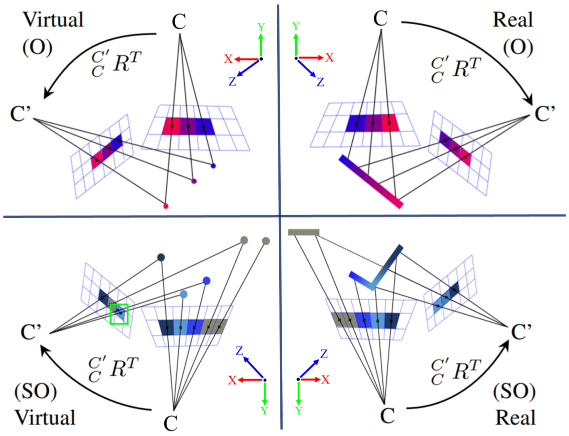

Figure 3: **Virtual and Real Projected Cases:** Optimal case (O): a virtual projection determines the same depth distances as a real, positionally unchanged projection. Suboptimal case (SO): several 3D points are contained in one pixel as depth information.

The four areas in Fig. 3 describe the optimal (O) and suboptimal (SO) cases of environmental projection. The structures of the physical scenes (top and bottom right) can be assigned explicitly to each pixel. In contrast, virtual projections of point clouds can contain pixels that are assigned to several environmental points in a suboptimal case. This single-point cloud indicates properties that favor the occlusion problem. By repositioning the point cloud, we obtain further information about existing occlusions, overlapping textures, and colors. The positional change of ∇p from C to C', described by Eq. 1, helps to achieve an unambiguous assignment.

$$\nabla p'_{(\mathrm{I}',\lambda')} := \nabla p_{(\mathrm{I},\lambda)} \cdot (^{C'}_{C}R^T T) \tag{1}$$

The image $[I]^n_{\mathrm{n+1}}$ and disparity map λ form a 4-channel configuration which can be used to extract a point cloud $\mathrm{P}(\nabla p) := P(I,\lambda)$. Thereby, the Image notation $[I]^n_{\mathrm{n+1}}$ is utilized to generalize the scene reconstruction process, typically involving multiple images for depth estimation. We generate a virtual replication $\nabla p'$ of ∇p from a new perspective (see Fig. 3 and Fig. 4).

The expected results of point cloud repositioning are shown in Fig. 4. We employ $(R \mid T)$ as a parameter for the intrinsic camera pose matrix in order to transform the position of the original camera C to the virtual camera C' position.

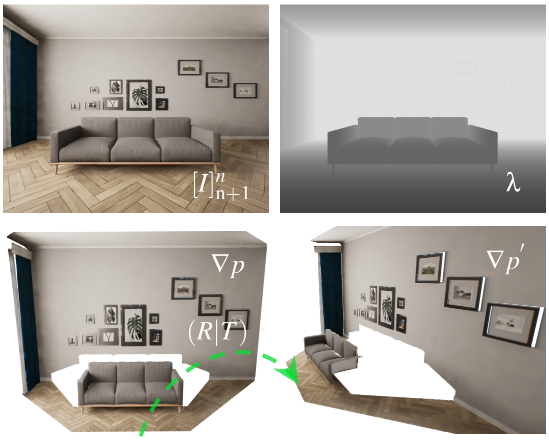

Figure 4: **Point Cloud Repositioning:** The point cloud ∇p can be extracted from the original image $[I]^n_{\mathrm{n+1}}$ and disparity map λ. A new perspective virtual point cloud $\nabla p'$ can be generated by translational and rotational shift $(R|T)$ of ∇p.

3.2 λ-Masking

The inpainting mask M plays a pivotal role in image processing, particularly in the inpainting domain. We construct a binary or greyscale image from the areas of the original image. While the pixels to be painted over are assigned the value 1, the value 0 represents untouched areas. A mask, as illustrated in Fig. 5, serves as a guide and instructs the inpainting algorithms to selectively focus their efforts and ensure precise application of the inpainting process.

The mask M_0, if provided e.g. by user input or object recognition and segmentation, enables the removal of objects by the inpainting process. This mask forms a general case with no positional change where the image pixel will be inpainted and assigned to the value 1. We apply the positionally changed point cloud $\nabla p'$ in a 3D to 2D projection-based function:

$$P(\nabla p') := P'(I', \lambda', M_\tau) \tag{2}$$

We can derive positionally changed image I', disparity map λ' and the corresponding virtual projected mask M_τ. In the case of M_0 being provided, we can readily derive M_λ, where the object of interest is inpainted over by applying the following function:

$$P(I'_\lambda, \lambda'_\lambda, M_\lambda) := (\nabla p'_{(\mathrm{I}',\lambda')} \wedge M_0) \tag{3}$$

$$P(M_\lambda) := (M_\tau \wedge M_0) \tag{4}$$

Due to imperfect masking, where the mask does not cover the object or occlusion region completely, the integration of morphological operations, especially

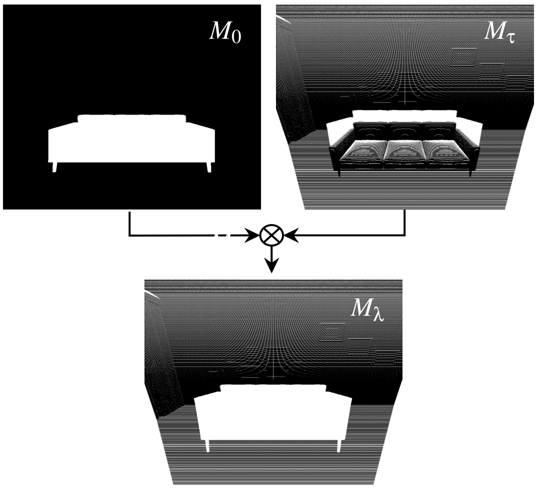

Figure 5: **Mask Generation:** M_0 illustrates a mask derived from the original image position. M_τ shows a virtual projected mask by positional change of perspective. M_λ results by combining M_0 and M_τ.

dilation and erosion, into image processing can reduce the degree of incomplete coverage. These operations, commonly used in tandem, dynamically alter the shape and size of the missing area. Dilation, serving as an expansive force, collaborates with a specified kernel to augment disjointed regions within the mask, facilitating a seamless transition between inpainted and non-inpainted areas. Conversely, erosion, functioning as a refining force, contracts the boundaries of the mask, confining the region designated for inpainting. The D-LaMa model exploits the synergy between dilation and erosion within morphological processing, which refines the inpainting mask.

3.3 Inpainting Network

Our concept employs the receptive field of a three-layer 2D convolutional network. The receptive field itself influences specific neurons of a network unit to produce specific features. As data traverses the network layers, the receptive field of neurons in deeper layers expands, encompassing information from a broader input data region. In Fig. 6 we illustrate the inpainting network of D-LaMa based on Fast Fourier Convolution (FFC).

FFC (Heusel et al., 2017; Suvorov et al., 2022) uses channel-wise Fast Fourier Transformation in a comprehensive image-wide receptive field consisting of global and local convolution kernels. By integrating the global context, the inadequacies of smaller convolution kernels (3×3) can be compensated. The D-LaMa model processes an input image and a mask as a combined 4-channel tensor and generates the final 3-channel RGB image through a fully convoluted approach.

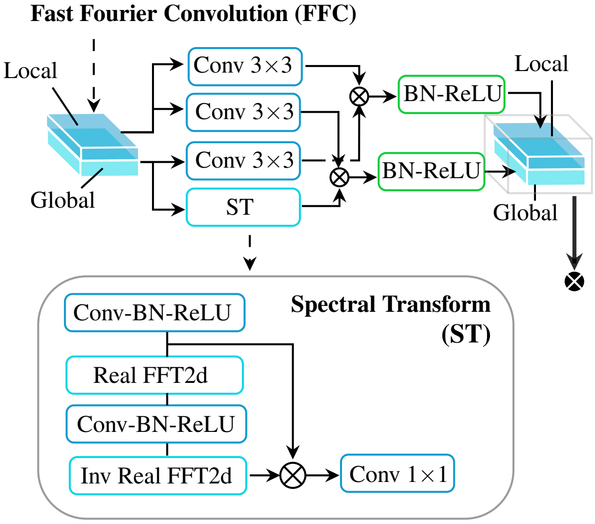

Figure 6: **Architecture of Fast Fourier Convolution (FFC) and Spectral Transform (ST) (Suvorov et al., 2022):** The architecture contains local and global parts consisting of 3×3 convolution kernel and spectral transform.

Loss Function: Deliberate and systematic strategies are used to integrate a range of loss functions. Each loss function has a specific role in improving the overall performance of the model.

High Receptive Field Perceptual Loss: The D-LaMa model incorporates the so-called High Receptive Field Perceptual Loss (HRF PL), which utilizes a base model $\phi_{HRF}(\cdot)$. The HRF PL between the input image x and the resulting inpainted image $\hat{x}$ is formulated as follows:

$$\mathcal{L}_{HRFPL}(x,\hat{x}) = \mathcal{M}([\phi_{HRF}(x) - \phi_{HRF}\hat{x}]^2) \qquad (5)$$

Eq. 5 signifies an element-wise operation with $\mathcal{M}$ as a sequential two-stage mean operation (interlayer mean of intra-layer means). $\phi_{HRF}(x)$ can be implemented using Fourier or Dilated convolutions.

Adversarial Loss: The adversarial loss ensures that the LaMa model $f_\theta(x')$ generates local details that appear natural. A discriminator $D_\xi(\cdot)$ operates at a local patch level (Isola et al., 2017), distinguishing between "real" and "fake" patches. The non-saturating adversarial loss is defined as:

$$\mathcal{L}_D = -\mathbb{E}_x[\log D_\xi(x)] - \mathbb{E}_{x,m}[\log D_\xi(\hat{x}) \odot m]$$
$$-\mathbb{E}_{x,m}[\log(1 - D_\xi(\hat{x})) \odot (1-m)] \qquad (6)$$
$$\mathcal{L}_G = -\mathbb{E}_{x,m}[\log D_\xi(\hat{x})] \qquad (7)$$
$$\mathcal{L}_{Adv} = \mathrm{sg}_\theta(\mathcal{L}_D) + \mathrm{sg}_\xi(\mathcal{L}_G) \rightarrow \min_{\theta,\xi} \qquad (8)$$

x represents a sample from a dataset, m is a synthetically generated mask, $\hat{x} = f_\theta(x')$ is the inpainting result for $x' = \mathrm{stack}(x \odot m, m)$, the stops gradients operator sg_{var} with respect to the variable var, and $\mathcal{L}_{Adv}$, which stands for the combined loss used for optimization.

Final Loss Function: The LaMa model additionally utilizes loss functions by $R_1 = \mathbb{E}_x \|\nabla D_\xi(x)\|^2$ for gradient penalty as proposed by (Mescheder et al., 2018; Ross and Doshi-Velez, 2018; Esser et al., 2021), and a discriminator-based perceptual loss, or feature matching loss, denoted as $\mathcal{L}_{DiscPL}$ (Wang et al., 2018b). $\mathcal{L}_{DiscPL}$ stabilizes training and occasionally improves performance slightly. The final loss function $\mathcal{L}_{final}$ for the LaMa inpainting model can therefore be denoted as the weighted sum of the previously mentioned losses.

$$\mathcal{L}_{final} = \kappa \mathcal{L}_{Adv} + \alpha \mathcal{L}_{HRFPL} + \beta \mathcal{L}_{DiscPL} + \gamma R_1 \quad (9)$$

$\mathcal{L}_{Adv}$ and $\mathcal{L}_{DiscPL}$ contribute to the generation of naturally looking local details, while $\mathcal{L}_{HRFPL}$ ensures the supervised signal and consistency of the global structure. In our experiments, the hyperparameters $(\kappa, \alpha, \beta, \gamma)$ are determined via the coordinate-wise beam-search strategy, resulting in the weight values $\kappa = 10$, $\alpha = 30$, $\beta = 100$, and $\gamma = 0.001$ (Suvorov et al., 2022).

Previous work evaluated LaMa variants containing a ResNet-like architecture (He et al., 2016), consisting of three downsampling blocks, 6-18 residual blocks with integrated FFC, and three upsampling blocks. We use a “Big LaMa” variant, which employs eight residual blocks and is trained exclusively on low-resolution 256 × 256 crops extracted from approximately 512 × 512 images. The used variant of this paper is trained on eight NVIDIA V100 GPUs for approximately 240 hours (Suvorov et al., 2022).

3.4 Reconstruction

The reconstruction defines the part of the D-LaMa model in which the point cloud is perspectively rebuilt and re-occluded to complete scenes with objects or to hide objects and reconstruct backgrounds. Fig. 7 illustrates the different variants of the scene that can be reconstructed by combining I-PCL and PCL.

As a result of D-LaMa, we receive the re-occluded mask of point cloud $\nabla p''_{\mathrm{i}}$. By adding $\nabla p''_{\mathrm{i}}$ to the original point cloud ∇p, we can isolate the inpainted regions and reconstruct the complete point cloud $\nabla p_{\mathrm{c}+}$.

$$\nabla p_{\mathrm{c}+} = \nabla p_{(\mathrm{I},\lambda)} + \nabla p''_{(\mathrm{I}'',\lambda'')} \quad (10)$$

$$\nabla p_{\mathrm{c}-} = (\nabla p_{(\mathrm{I},\lambda)} \wedge M_0) + \nabla p''_{(\mathrm{I}'',\lambda'')} \quad (11)$$

In turn, we can hide the object and fill the occluded background area by subtracting $\nabla p''_{\mathrm{i}}$ from ∇p.

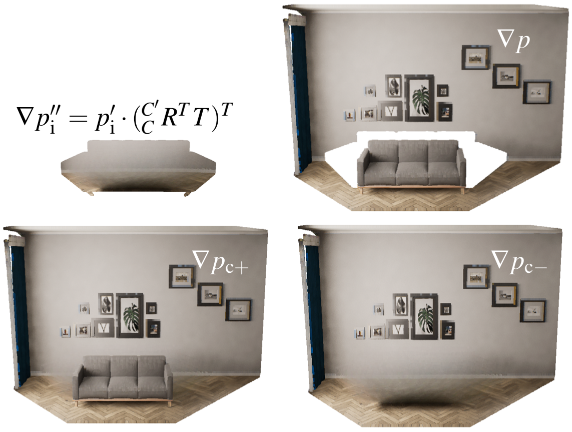
$\nabla p''_{\mathrm{i}} = p'_{\mathrm{i}} \cdot ({}^{C'}_{C}R^T T)^T$
∇p
$\nabla p_{\mathrm{c}+}$
$\nabla p_{\mathrm{c}-}$

Figure 7: **Reconstruction of Inpainted Point Cloud (I-PCL):** Top left shows the re-occluded background of the inpainted point cloud $\nabla p''_{\mathrm{i}}$ generated by the D-LaMa model. The point cloud of the entire scene $\nabla p''_{\mathrm{i}}$ can be reconstructed as an occlusion-free point cloud $\nabla p_{\mathrm{c}+}$ or an occlusion-free point cloud with hidden object $\nabla p_{\mathrm{c}-}$.

4 Experimental Design

Data and metrics: In our experiments, we use synthetically generated data from Unreal Engine 4.27[1] to simulate a real-world outdoor environment. The dataset includes images and object masks in .png and depth information in .exr format with 512x512 px resolution, generated with the easySynth library (Ydrive, 2022). We used the Downtown West Modular Pack example project by Pure Polygons (Polygons, 2020) to replicate a natural outdoor environment scene. Two distinct datasets were generated from the scenes: One where the object of interest is present and one with the object removed.

We conducted a thorough data selection and post-processing phase to ensure both a diverse environment as well as compliance with the experiment requirements. These include depth image conversion, where the depth information is converted to a depth image (by normalizing and limiting the depth to ten meters), and mask enhancement, where the object mask is represented as black-and-white image composite without unwanted artifacts and noise. We used the Python libraries OpenCV (Bradski, 2000), TensorFlow (Developers, 2023), PyTorch (Paszke et al., 2019), and TensorLightning (AI, 2015) for the inpainting process, and Open3D (Zhou et al., 2018) to facilitate visualizing the results.

To assess our empirical experiment, we follow the established practice in image2image literature by using the image similarity metrics Structural Similarity Index (SSIM) (Wang et al., 2003; Wang et al.,

[1] https://www.unrealengine.com

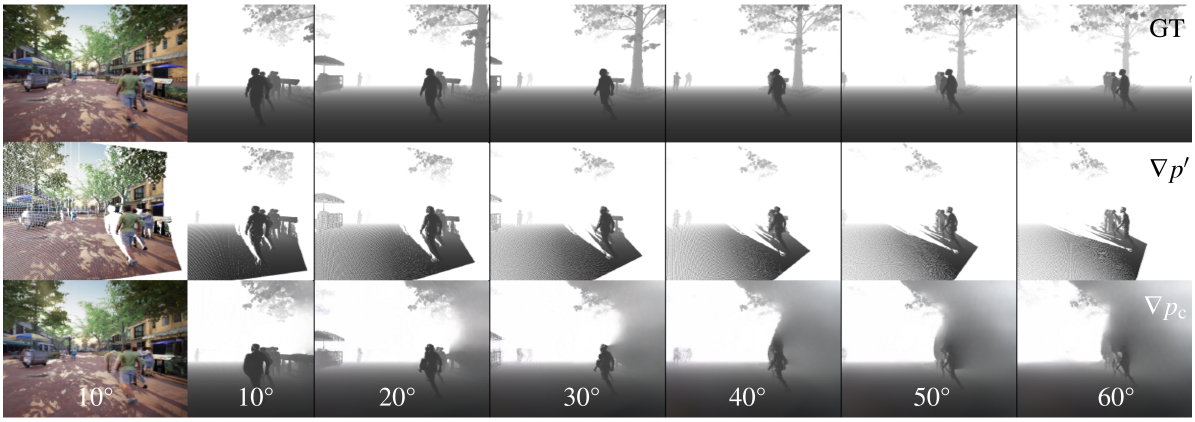

Figure 8: **Results of a Rotated Virtual Projection:** Based on the colored image at top left, the image and the corresponding point cloud are rotated by 10 degrees each. At the top is the ground truth (GT). In the middle is the virtual point cloud $\nabla p'$ with occlusion (visible as a white shadow). Below is the complete scene with fixed occlusion as a ∇p_c.

2004), Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018), and Mean Squared Error (MSE). The values of SSIM and LPIPS range from 0 to 1. A higher SSIM score implies a higher similarity between two images, while higher LPIPS and MSE scores signify greater differences. Python libraries, such as scikit-image (scikit image.org, 2022) are used to assess the SSIM and MSE metrics, while the LPIPS metric is assessed using the library provided by (Zhang et al., 2018).

Evaluation: Our approach involves creating a point cloud from the original viewpoint and projecting it back from a new perspective. However, due to the discrete nature of point clouds and the lack of physical properties in the point clouds, the resulting projection may deviate from the actual 3D scene. Consequently, projecting the original point cloud onto new viewpoints may result in different virtual image projections, where the occluded region is exposed and represented as a void in the new virtual image. Acknowledging this limitation, we evaluate the missing region ratio and the inpainted results on positionally changed image I', the depth image λ' on the corresponding virtual projected mask M_τ via rotational transformation by comparing them with the ground truth images taken from the actual viewpoints. In this case, we consider the general case of our approach, where the object mask is not yet used.

The rotational values defined in the experiment are 10, 20, 30, 40, 50, and 60 degrees to the left and right (±), where we transform our initial point cloud by rotating it about the Y-axis (Yaw) with a defined centroid 250 cm from the camera center along the Z-axis. In addition to the evaluation of inpainted results in general case, we evaluate the effect of object mask M_τ in the process.

5 Results

Larger rotation values result in larger missing or occluded areas in the virtual image and depth image, which are treated as the inpainting mask. By rotating the point cloud and projecting it with a larger rotation value, we further expose missing fields of view, resulting in larger missing areas on each side of the virtual image and depth image. Additionally, the missing areas are compounded by the missing pixel values caused by the occlusion of the foreground object as well as by rounding errors (around 1 px in size) during the point cloud re-projection process.

Rot.	Mask Ratio	SSIM	LPIPS	MSE
±60	68.96	0.4559	0.5651	3284.25
±50	64.87	0.4873	0.5221	2871.92
±40	59.49	0.5292	0.4752	2337.43
±30	52.42	0.5839	0.4035	1657.52
±20	42.37	0.6593	0.3134	1065.91
±10	27.19	0.7617	0.1859	522.05

Table 1: **Inpainting Result of Virtual Image for Select Rotations:** The results refer to Fig. 8 and compares the images by different similarity metrics.

By comparing the inpainting result on the virtual image and depth image (see Tables 1 and 2) we discover that the D-LaMa inpainting model performance degrades as the mask area to be inpainted increases. In most cases, the suboptimal inpainting result is caused by the incapability of the D-Lama model to correctly inpaint the area if multiple distinct objects are present around the missing area. In this case, the D-LaMa model attempts to inpaint the missing area by blending the surrounding objects, resulting in erroneous color and depth information. Additionally, due to anti-aliasing, the resulting virtual image may also be erroneous, particularly around the object edges, since this area contains wrong depth information. This frequently happens in real-world scenarios if the depth information is widely erroneous. Since a straightforward method is used to project the point cloud into the virtual image plane, incorrect pixel values may be assigned to the resulting virtual image and depth image if points that should be obscured from a new viewpoint are exposed. This can contribute to non-optimal inpainting results.

Based on these findings, we conducted a further experiment where an object mask filters the point cloud so that the resulting virtual image and depth image exclude the occlusion directly. This leads to significantly improved inpainting performance if the foreground object is completely removed from the virtual image and depth image. By slightly dilating the mask (e.g., by 3px) proper coverage of the occlusion area can be ensured. To address the rounding error during the point cloud re-projection process and incorrect pixel value assignment in occluded areas, one can, for example, include a translation parameter to zoom out the projection result. A denser point cloud projection can eliminate undefined pixel values or determine and correct missing or incorrect pixel values based on their neighboring pixel value information.

Rot.	Mask Ratio	SSIM	LPIPS	MSE
±60	68.96	0.8506	0.3091	1452.72
±50	64.87	0.8648	0.2727	1074.61
±40	59.49	0.8680	0.2640	868.91
±30	52.42	0.8712	0.2523	657.17
±20	42.37	0.8847	0.2195	393.29
±10	27.19	0.9016	0.1850	186.54

Table 2: **Inpainting Result of Virtual Depth Image for Select Rotations:** The results refer to Fig. 8 and compare the depth maps by different similarity metrics.

As the number of input and output channels of the D-LaMa model is limited to three (RGB), the inpainting processes of the virtual image and depth image need to be performed separately. This may result in offsets between the virtual and depth image inpainting results. Moreover, the inpainting results of the virtual depth images may deviate slightly along their channels. However, due to the small size of the deviation, this outcome can readily be improved by converting the inpainted depth image into a grayscale image.

6 Discussion

Virtual Projected Inpainting: The use of virtual camera projections for inpainting reveals problems caused by the properties of point clouds and their omission of real-world attributes, such as reflections. Consequently, scenes with complicated geometry and reflective objects as well as strict point cloud transformations can degrade the results. Inpainting models with a fixed input and output size can lead to additional limitations. For real-world applications with pixel-by-pixel misalignment, our results highlight the importance of extending the mask to ensure precise coverage of the occlusal region. This provides valuable insight to improve occlusion completion in complex scenarios with misaligned depth and color images.

Object Removal Inpainting: The final result for color and depth inpainting for a given object mask is significantly impacted if the mask does not cover the entire object. Our results reveal that the extraction of occluding objects and subsequent 3D scene reconstruction leads to artifacts related to real-world features such as cast shadows. The persistent shadow of an object that has already been removed can affect realism. Therefore, it could be necessary to reconstruct cast shadows to improve the result.

Stereoscopic Image Inpainting: Complementing the results of virtual projection and object removal, our model shows promising performance when inpainting images in a stereoscopic environment where depth is not explicitly available. Inpainting the left and right halves of stereo images separately leads to potentially disjointed results in the inpainted region, particularly if there are distinct differences in the global and local characteristics between the two images. This incoherence may have detrimental effects on the resulting depth image and prevent perfect pixel matching. In addition, noise and imperfections in the depth image generated by the stereo depth estimation process affect the quality of the resulting point cloud. Alternatively, the stereo image and the object mask for inpainting can be used for object removal during scene reconstruction to improve the overall performance of the model in real-world applications.

7 Conclusion

Occlusion is a significant problem that complicates the understanding of a scene as it prevents the accurate perception of spatial relationships between objects. It impairs depth estimation and leads to inaccuracies in estimating distances of occluded objects. Consequently, mitigating the effects of occlusions is crucial for strengthening the robustness and reliability of computer vision systems.

We demonstrate the effectiveness of the D-LaMa model on the occlusion problem through practical applications. Our proposed methodology proves promising but depends on the use of a robust, pretrained artificial intelligence model for image inpainting tasks. This enables seamless integration and near real-time applicability and allows computational challenges to be met efficiently, while comparable traditional approaches often remain computationally intensive. Our decision to use synthetic data instead of real data for evaluation ensures a thorough understanding of the challenges and influencing factors and establishes our approach as a cutting-edge solution in occlusion completion.

Future Work: In the pursuit of enhancing the robustness and practical applicability of our approach, we outline our future research, encompassing the transition to real-world data and the development of a dedicated model through transfer learning. The training time can be reduced by using existing knowledge. In addition, machine learning models, such as object recognition or segmentation, can be used to generate accurate object masks. While synthetic data provides controlled scenarios, incorporating the refinements of real-world data is critical. Future work could incorporate real data sets to cover the variety and complexity of real environmental conditions, lighting variations, and unforeseen scenarios, providing a more thorough assessment of the D-LaMa performance in practical applications.

Furthermore, investigating the pre-conversion of point clouds into meshes or voxels reprojection can mitigate occlusion-related problems. To overcome the limitations in the size of input and output channels, the implementation of a dedicated inpainting model can be extended with a 5-channel configuration: Three RGB channels, one grayscale channel for masks, and one channel for depth. This improvement intends to inpaint texture and spatial information simultaneously. For stereoscopic input, we suggest a 7-channel setup which aims to produce coherent results for both the left and right images.

8 Acknowledgements

We thank Thomas Odaker and Elisabeth Mayer, who supported this work with helpful discussions and feedback.

REFERENCES

Achlioptas, P., Diamanti, O., Mitliagkas, I., and Guibas, L. (2018). Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR.

Aharchi, M. and Ait Kbir, M. (2020). A review on 3d reconstruction techniques from 2d images. In Ben Ahmed, M., Boudhir, A. A., Santos, D., El Aroussi, M., and Karas, İ. R., editors, *Innovations in Smart Cities Applications Edition 3*, pages 510–522, Cham. Springer International Publishing.

AI, L. (2015). Pytorch lightning. accessed on 01. August 2023.

Amenta, N., Bern, M., and Kamvysselis, M. (1998). A new voronoi-based surface reconstruction algorithm. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 415–421.

Bajaj, C. L., Bernardini, F., and Xu, G. (1995). Automatic reconstruction of surfaces and scalar fields from 3d scans. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 109–118.

Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., and Taubin, G. (1999). The ball-pivoting algorithm for surface reconstruction. *IEEE transactions on visualization and computer graphics*, 5(4):349–359.

Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Brock, A., Lim, T., Ritchie, J. M., and Weston, N. (2016). Generative and discriminative voxel modeling with convolutional neural networks. *arXiv preprint arXiv:1608.04236*.

Chen, Y. and Medioni, G. (1995). Description of complex objects from multiple range images using an inflating balloon model. *Computer Vision and Image Understanding*, 61(3):325–334.

Chen, Y.-T., Garbade, M., and Gall, J. (2019). 3d semantic scene completion from a single depth image using adversarial training.

Chen, Z. and Zhang, H. (2019). Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948.

Curless, B. and Levoy, M. (1996). A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312.

Dai, A., Ritchie, D., Bokeloh, M., Reed, S., Sturm, J., and Nießner, M. (2018). Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans.

Davis, J., Marschner, S. R., Garr, M., and Levoy, M. (2002). Filling holes in complex surfaces using volumetric diffusion. In *Proceedings. First international symposium on 3d data processing visualization and transmission*, pages 428–441. IEEE.

Developers, T. (2023). Tensorflow.

Doria, D. and Radke, R. J. (2012). Filling large holes in lidar data by inpainting depth gradients. In *2012*

IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pages 65–72. IEEE.

Edelsbrunner, H. and Mücke, E. P. (1994). Three-dimensional alpha shapes. *ACM Transactions On Graphics (TOG)*, 13(1):43–72.

Esser, P., Rombach, R., and Ommer, B. (2021). Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883.

Fan, H., Su, H., and Guibas, L. J. (2017). A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613.

Firman, M., Mac Aodha, O., Julier, S., and Brostow, G. J. (2016). Structured prediction of unobserved voxels from a single depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5431–5440.

Girdhar, R., Fouhey, D. F., Rodriguez, M., and Gupta, A. (2016). Learning a predictable and generative vector representation for objects. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14*, pages 484–499. Springer.

Groueix, T., Fisher, M., Kim, V. G., Russell, B. C., and Aubry, M. (2018). A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 216–224.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

He, L., Bleyer, M., and Gelautz, M. (2011). Object removal by depth-guided inpainting. In *Proc. AAPR Workshop*, pages 1–8. Citeseer.

Hervieu, A., Papadakis, N., Bugeau, A., Gargallo, P., and Caselles, V. (2010). Stereoscopic image inpainting: distinct depth maps and images inpainting. In *2010 20th international conference on pattern recognition*, pages 4101–4104. IEEE.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.

Huang, H.-Y. and Huang, S.-Y. (2020). Fast hole filling for view synthesis in free viewpoint video. *Electronics*, 9(6):906.

Insafutdinov, E. and Dosovitskiy, A. (2018). Unsupervised learning of shape and pose with differentiable point clouds. *Advances in neural information processing systems*, 31.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134.

Kazhdan, M., Bolitho, M., and Hoppe, H. (2006). Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, page 0.

Li, D., Shao, T., Wu, H., and Zhou, K. (2016). Shape completion from a single rgbd image. *IEEE transactions on visualization and computer graphics*, 23(7):1809–1822.

Li, D.-H., Hang, H.-M., and Liu, Y.-L. (2013). Virtual view synthesis using backward depth warping algorithm. In *2013 Picture Coding Symposium (PCS)*, pages 205–208. IEEE.

Li, J., Xu, K., Chaudhuri, S., Yumer, E., Zhang, H., and Guibas, L. (2017). Grass: Generative recursive autoencoders for shape structures. *ACM Transactions on Graphics (TOG)*, 36(4):1–14.

Li, S., Zhu, C., and Sun, M.-T. (2018). Hole filling with multiple reference views in dibr view synthesis. *IEEE Transactions on Multimedia*, 20(8):1948–1959.

Li, Y., Dai, A., Guibas, L., and Nießner, M. (2015). Database-assisted object retrieval for real-time 3d reconstruction. In *Computer graphics forum*, volume 34, pages 435–446. Wiley Online Library.

Lin, C.-H., Kong, C., and Lucey, S. (2018). Learning efficient point cloud generation for dense 3d object reconstruction. In *proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pages 7114—-7121.

Mark, W. R., McMillan, L., and Bishop, G. (1997). Post-rendering 3d warping. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 7–ff.

Mescheder, L., Geiger, A., and Nowozin, S. (2018). Which training methods for gans do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR.

Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., and Geiger, A. (2019). Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. (2020). Nerf: Representing scenes as neural radiance fields for view synthesis.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. (2021). Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106.

Mori, Y., Fukushima, N., Yendo, T., Fujii, T., and Tanimoto, M. (2009). View generation with 3d warping using depth information for ftv. *Signal Processing: Image Communication*, 24(1-2):65–72.

Müller, T., Evans, A., Schied, C., and Keller, A. (2022). Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15.

Müller, S., and Kranzlmüller, D. (2021). Dynamic Sensor Matching for Parallel Point Cloud Data Acquisition.

In *29. International Conference in Central Europe on Computer Graphics (WSCG)*, pages 21–30.

Müller, S., and Kranzlmüller, D. (2022). Dynamic Sensor Matching based on Geomagnetic Inertial Navigation. In *30. International Conference in Central Europe on Computer Graphics (WSCG)*.

Munkberg, J., Hasselgren, J., Shen, T., Gao, J., Chen, W., Evans, A., Müller, T., and Fidler, S. (2023). Extracting triangular 3d models, materials, and lighting from images.

Nazeri, K., Ng, E., Joseph, T., Qureshi, F. Z., and Ebrahimi, M. (2019). Edgeconnect: Generative image inpainting with adversarial edge learning.

Nealen, A., Igarashi, T., Sorkine, O., and Alexa, M. (2006). Laplacian mesh optimization. In *Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, pages 381–389.

Park, J. J., Florence, P., Straub, J., Newcombe, R., and Lovegrove, S. (2019). Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Pauly, M., Mitra, N., Wallner, J., Pottmann, H., and Guibas, L. (2008). Discovering structural regularity in 3d geometry. *ACM transactions on graphics*, 27.

Pauly, M., Mitra, N. J., Giesen, J., Gross, M. H., and Guibas, L. J. (2005). Example-based 3d scan completion. In *Symposium on geometry processing*, number CONF, pages 23–32.

Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., and Geiger, A. (2020). Convolutional occupancy networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 523–540. Springer.

Polygons, P. (2020). Downtown west modular pack. https://www.unrealengine.com/marketplace/en-US/product/6bb93c7515e148a1a0a0ec263db67d5b.

Reiser, C., Peng, S., Liao, Y., and Geiger, A. (2021). Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14335–14345.

Rock, J., Gupta, T., Thorsen, J., Gwak, J., Shin, D., and Hoiem, D. (2015). Completing 3d object shape from one depth image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2484–2493.

Rosinol, A., Leonard, J. J., and Carlone, L. (2023). Nerf-slam: Real-time dense monocular slam with neural radiance fields. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3437–3444. IEEE.

Ross, A. and Doshi-Velez, F. (2018). Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, pages 1660–1669.

scikit image.org (2022). scikit-image.image processing in python. https://scikit-image.org/.

Shen, C.-H., Fu, H., Chen, K., and Hu, S.-M. (2012). Structure recovery by part assembly. *ACM Transactions on Graphics (TOG)*, 31(6):1–11.

Sipiran, I., Gregor, R., and Schreck, T. (2014). Approximate symmetry detection in partial 3d meshes. In *Computer Graphics Forum*, volume 33, pages 131–140. Wiley Online Library.

Song, S., Yu, F., Zeng, A., Chang, A. X., Savva, M., and Funkhouser, T. (2017). Semantic scene completion from a single depth image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1746–1754.

Sorkine, O. and Cohen-Or, D. (2004). Least-squares meshes. In *Proceedings Shape Modeling Applications, 2004.*, pages 191–199. IEEE.

Sung, M., Kim, V. G., Angst, R., and Guibas, L. (2015). Data-driven structural priors for shape completion. *ACM Transactions on Graphics (TOG)*, 34(6):1–11.

Suvorov, R., Logacheva, E., Mashikhin, A., Remizova, A., Ashukha, A., Silvestrov, A., Kong, N., Goka, H., Park, K., and Lempitsky, V. (2022). Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2149–2159.

Tatarchenko, M., Dosovitskiy, A., and Brox, T. (2017). Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *Proceedings of the IEEE international conference on computer vision*, pages 2088–2096.

Wang, L., Jin, H., Yang, R., and Gong, M. (2008). Stereoscopic inpainting: Joint color and depth completion from stereo images. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE.

Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., and Jiang, Y.-G. (2018a). Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 52–67.

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., and Catanzaro, B. (2018b). High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612.

Wang, Z., Simoncelli, E. P., and Bovik, A. C. (2003). Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee.

Wu, J., Zhang, C., Xue, T., Freeman, B., and Tenenbaum, J. (2016). Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Advances in neural information processing systems*, 29.

Wu, R., Zhuang, Y., Xu, K., Zhang, H., and Chen, B. (2020). Pq-net: A generative part seq2seq network for 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 829–838.

Yan, X., Yang, J., Yumer, E., Guo, Y., and Lee, H. (2016). Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. *Advances in neural information processing systems*, 29.

Yan, Z., Li, X., Li, M., Zuo, W., and Shan, S. (2018). Shift-net: Image inpainting via deep feature rearrangement.

Yang, Y., Feng, C., Shen, Y., and Tian, D. (2018). Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 206–215.

Yao, L., Han, Y., and Li, X. (2019). Fast and high-quality virtual view synthesis from multi-view plus depth videos. *Multimedia Tools and Applications*, 78:19325–19340.

Ydrive (2022). Easysynth. https://www.unrealengine.com/marketplace/en-US/product/easysynth.

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., and Huang, T. (2019). Free-form image inpainting with gated convolution.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595.

Zhou, Q.-Y., Park, J., and Koltun, V. (2018). Open3d: A modern library for 3d data processing.

Zou, C., Yumer, E., Yang, J., Ceylan, D., and Hoiem, D. (2017). 3d-prnn: Generating shape primitives with recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 900–909.