*Sensors and Materials*, Vol. 36, No. 12 (2024) 5491–5505
MYU Tokyo

S & M 3881

# Research on a Three-dimensional Reconstruction Algorithm for Power Grid Tower Poles Based on a Foreground-excluded Neural Radiance Field

Jiyong Zhang,[1] Aiyuan Zhang,[2*] Jingguo Lv,[2]
Donghui Liu,[1*] Xiaohu Sun,[1] and Bing Wu[3]

[1]State Grid Economic and Technological Research Institute Co., Ltd.,
Building A, 5th and 6th Floors, No. 18 Binhe Avenue, Future Science and Technology City, Changping District,
Beijing City, Beijing 102200, China
[2]Beijing University of Civil Engineering and Architecture,
No. 15 Yongyuan Road, Daxing District, Beijing City, Beijing 102616, China
[3]Economic Research Institute of State Grid ZheJiang Electric Power Company,
No. 1 Nanfu Road, Shangcheng District, Hangzhou City, Shuicheng Building, Hangzhou 310020, China

(Received August 27, 2024; accepted December 6, 2024)

***Keywords:*** NeRF, DPT, 3D reconstruction, dynamic foreground culling, differentiable sampling, power grid towers

Neural radiance field (NeRF) has emerged as a cutting-edge approach in neural rendering for 3D scene representation. Addressing the limitations of conventional NeRF in handling dynamic foregrounds and depth discrimination in complex power grid scenes, in this study, we introduce an enhanced NeRF algorithm leveraging the dense prediction transformer (DPT). Our method employs DPT to eliminate dynamic foreground elements across viewpoints and integrates a dual-sphere reparameterization with a differentiable sampling strategy within the NeRF model to apply blur constraints to distant scenes. This approach notably improves the reconstruction quality of power grid towers. Experiments show that our algorithm surpasses traditional NeRF variants in capturing details, rendering backgrounds, and overall visual quality, with improved performance in peak signal-to-noise ratio mean structural similarity index measure, and multi-scale structural similarity index measure (MS-SSIM) metrics compared with state-of-the-art models.

## 1. Introduction

Power grid towers are crucial structures in the electrical transmission network, and their precise 3D models are essential during the surveying phase. 3D reconstruction technology provides detailed spatial data and visualization, which is irreplaceable for planning transmission lines, assessing the interaction between towers and the environment, and optimizing the tower layout. Accurate 3D models allow engineers to foresee and resolve potential spatial conflicts, evaluate the feasibility of different tower designs, and make more rational path selections in

[*]Corresponding author: e-mail: 2108570023107@stu.bucea.edu.cn

https://doi.org/10.18494/SAM5348

complex terrains. Additionally, 3D reconstruction helps reduce the time and cost of field surveys, enhancing the safety and efficiency of surveying work. As 3D reconstruction technology continues to evolve, its application in the surveying phase will become more widespread, providing strong technical support for innovation and development in the power industry.

3D reconstruction is a technique that uses computer vision to extract features from images taken from multiple angles and reconstruct the 3D structure of objects. The main 3D reconstruction algorithms are divided into traditional geometric and deep learning-based methods. Traditional methods include active and passive geometric approaches. Active methods use sensors to emit signals and analyze the reflected signals to obtain 3D structural information. Passive methods rely on multi-view images and multi-view geometry principles to calculate disparities and reconstruct 3D shapes.(1) Others have used SLAM and GNSS information for localization and mapping in complex scenes, but these methods have limitations in generalization. For scenes with no texture, complexity, and lack of GNSS signals, some researchers(2–4) have designed collection devices to recover structures, but complex equipment system design has led to weaker generalization capabilities. Structured light scanning technology, such as Azure Kinect,(5) obtains the depth value of each pixel through a depth camera, is not limited by texture conditions, and further improves accuracy. However, as the distance increases, the accuracy decreases, making it difficult to meet the needs of large-scale scenes.(6) Others have enhanced reconstruction in textureless scenes with random light sources, but this method is limited in strong light environments.

With the significant progress of deep learning algorithms in the field of computer vision, traditional geometric algorithms improved by deep learning have gradually emerged. In 2016, Revaud *et al.* first proposed an unsupervised learning framework.(7) Although unsupervised learning may not be efficient, scholars have significantly improved the accuracy of feature matching with end-to-end models such as SuperPoint,(8) SuperGlue,(9) and LoFTR,(10) especially in weak texture areas, surpassing traditional methods.

Deep learning applications in the field of 3D reconstruction leverage its powerful data processing and complex pattern learning capabilities. Deep-learning-based methods are mainly divided into (1) depth-based methods such as DeepMVS(11) and MVSNet,(12) which use convolutional neural networks and stereo matching techniques to predict depth maps from multi-view images; (2) voxel-based methods such as 3D ShapeNets(13) and VoxNet,(14) which reconstruct 3D voxel models from a single image through deep belief or 3D convolutional neural networks; (3) point-cloud-based methods such as PointSetGeneration(15) and FoldingNet,(16) which directly generate 3D point clouds from a single image, handling complex scenes and generating high-fidelity point clouds; (4) mesh-based methods such as Pixel2Mesh;(17) and (5) implicit-function-based methods such as DeepSDF,(18) which represent and reconstruct 3D shapes through graph convolutional networks and continuous signed distance functions, respectively. These methods demonstrate the advantages of deep learning in automatic feature extraction and improving generalization capabilities. The introduction of PIFuHD,(19) IM-NET,(20) and occupancy networks(21) has further improved the accuracy and efficiency of deep learning in the field of 3D reconstruction.

NeRF, a differentiable rendering method based on neural implicit solid representation, has emerged as a novel approach for high-quality view synthesis. However, its application in complex outdoor environments, such as power grid scenes with dynamic foregrounds and distant, unbounded backgrounds, presents challenges. In this paper, we address these challenges by proposing an improved 3D reconstruction method for power grid towers based on the DPT network's ability to identify dynamic foreground elements. The overall architecture of the model is shown in Fig. 1. Our contributions include the following:

(1) The algorithm introduces a DPT semantic segmentation and depth estimation network with a transformer structure, which eliminates dynamic foreground interference in images, ensuring the continuity and integrity of power facility reconstruction.

(2) At the same time, the improved sampling strategy and dual-sphere reparameterization method optimize the ray tracing process, enhance the focus on the foreground of power facilities, and reasonably handle distant scene information, reducing the impact of blur on reconstruction quality.

(3) Comparative experiments are conducted to compare the algorithm in this paper with traditional NeRF and NeRF-W algorithms in terms of detail capture ability, background rendering ability, and overall visual effect.

## 2. Related Work

### 2.1 Three-dimensional reconstruction of power grid towers

The three-dimensional reconstruction of power grid towers is essential in electrical system infrastructure, traditionally achieved through stereo vision techniques such as SFM and MVS,

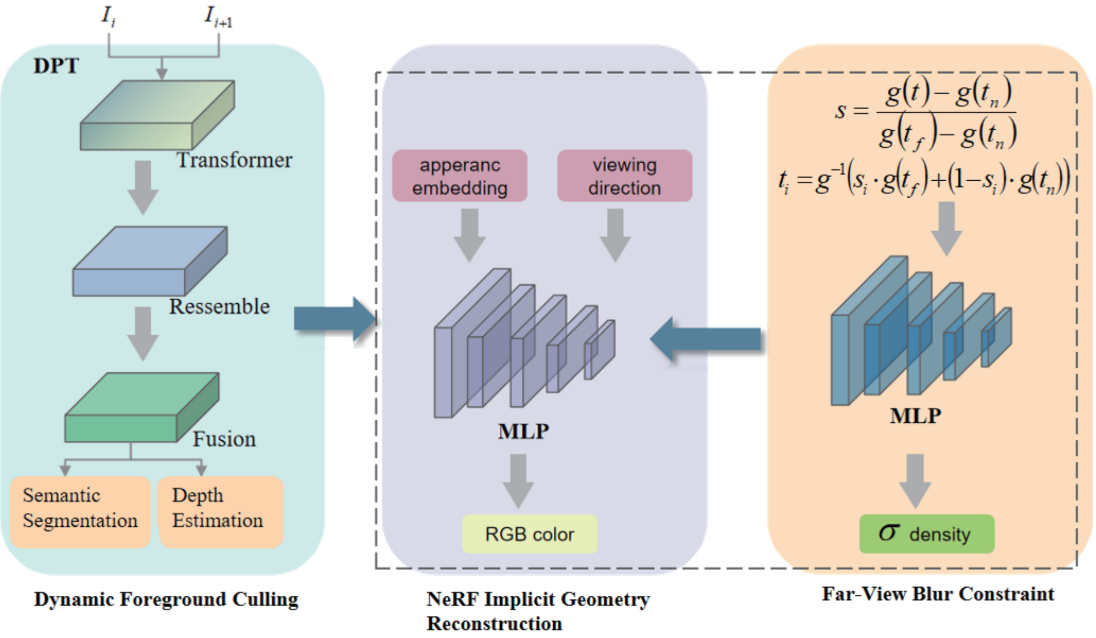

Fig. 1. (Color online) Technical approach for model construction.

which construct 3D models from 2D images by analyzing disparities. Despite their prevalence, these methods struggle with efficiency and accuracy in complex and large-scale scenarios.

Advancements in technology have made LiDAR a preferred method for its precision and efficiency, especially in challenging environments. Integrating multisensor data, such as photogrammetry with LiDAR, has also improved reconstruction accuracy and robustness.

Challenges persist, such as obstructions from birds, nests, pedestrians, and vehicles, which can degrade reconstruction quality. Additionally, unbounded backgrounds such as vegetation and buildings significantly impact the precision of tower reconstruction.

In this study, we assessed the existing 3D reconstruction techniques and proposed innovative solutions to address these challenges, striving for a significant advancement in the field of power grid tower reconstruction.

The 3D reconstruction of power grid towers is essential for electrical infrastructure, yet traditional methods such as SFM and MVS struggle with efficiency and accuracy in complex scenarios. While LiDAR offers precision, challenges remain with dynamic elements and unbounded backgrounds affecting reconstruction quality. In this study, we address these limitations by proposing solutions that enhance accuracy, handle dynamic obstructions, and improve large-scale reconstruction efficacy, aiming to advance the field and bolster the reliability of power grid infrastructure.

### 2.2 Neural radiance fields

NeRF technology, introduced in 2020, revolutionizes 3D scene representation and synthesis. It uses neural networks to model 3D space's volume and creates realistic views from limited 2D images. NeRF operates on the premise of a 5D function correlating spatial positions and viewing directions to color and density outputs.

NeRF represents a continuous 3D scene as a 5D vector-valued function, where each 5D coordinate input includes a position vector $x = (x, y, z)$ in the target space and a viewing direction vector $d = (\theta, \varphi)$. By inputting the 5D coordinates of points along the viewing ray into a multilayer perceptron (MLP) network, NeRF approximates the continuous 5D scene and optimizes the weights $F_\theta$: $(x, d) \rightarrow (c, \sigma)$ to map out the color vector $c = (r, g, b)$ and volumetric density $\sigma_0$ emitted by the target space point in the direction opposite to the viewing ray. Assuming a viewing ray $r(t) = o + td$ with a near boundary $t_n$ and a far boundary $t_f$, the formula for calculating the color value is

$$C(r) = \int_{t_n}^{t_f} e^{-\int_{t_n}^{t} \sigma(o+sd)ds} \cdot \sigma(o+td) \cdot c(o+td, d)\, dt. \tag{1}$$

## 3. 3 D reconstruction algorithms for key power grid facilities

In this paper, we propose a NeRF algorithm based on foreground exclusion for the 3D reconstruction of power grid towers. Initially, image data of the power grid towers are collected from multiple viewpoints and preprocessed to enhance image quality. Subsequently, the dense prediction transformer (DPT) network is utilized for semantic segmentation and depth estimation to identify and remove dynamic foreground elements from the images. Then, a dual-sphere reparameterization method is applied to optimize the spatial representation of the scene, focusing on near-field details and appropriately blurring the distant background. Additionally, a differentiated sampling strategy is introduced to increase the sampling density in near-field areas and reduce the number of sampling points in distant areas, thereby improving the quality of reconstruction.

### 3.1 Dynamic foreground extraction based on DPT networks

In the 3D reconstruction of power grid towers, images are gathered from multiple perspectives, which are then preprocessed to enhance processing accuracy. Employing a DPT network, we performed semantic segmentation to distinguish towers from dynamic foreground elements, while depth estimation provides additional scene information. This facilitates the removal of dynamic elements, ensuring the clarity and accuracy of the tower reconstruction.

As shown in Fig. 2, the model assembles feature representations into an image-like form through a decoder and processes tokens at different resolutions. These feature representations are gradually fused, ultimately contributing to the generation of dense predictions. Notably, a simple three-stage reassembly operation is introduced, aimed at recovering an image-like representation from the arbitrary-layer output tokens produced by the transformer encoder.

$$\mathrm{Re}\,assemble_s^{\hat{D}}(t)=\left(\mathrm{Re}\,assemble_s\circ Concatenate\circ\mathrm{Re}\,ad\right)(t) \tag{2}$$

Here, $s$ represents the scale of the reconstructed representation compared with the input image and $D$ is the dimension of the output features.

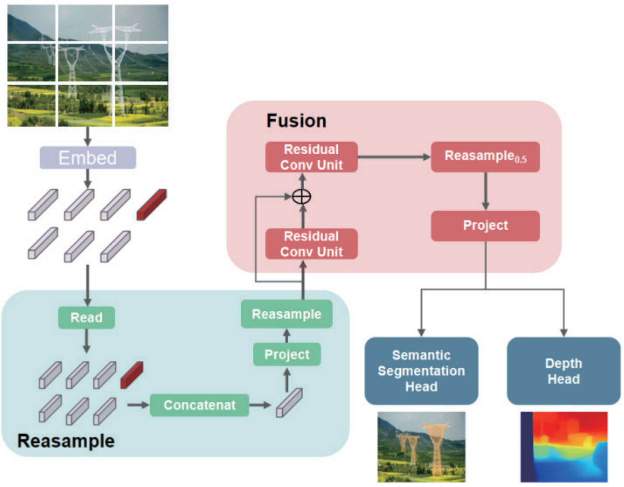

Fig. 2. (Color online) DPT network model architecture diagram.

The process begins by mapping $N_p$ + 1 tokens to $N_p$ tokens, creating an image-like representation through spatial concatenation.

$$\mathrm{Re}ad:\mathbb{R}^{N_p+1}\rightarrow\mathbb{R}^{N_p\times D} \tag{3}$$

This operation primarily manages the readout tokens effectively. Although readout tokens do not have a clear purpose for dense prediction tasks, they may still aid in capturing and distributing global information. Thus, the model evaluates three different variants of this mapping.

$$\begin{aligned}&\mathrm{Re}ad_{ignore}(t)=\left\{t_1+t_0,\ldots,t_{N_p}+t_0\right\}\\&\mathrm{Re}ad_{proj}(t)=\left\{mlp\left(cat\left(t_1,t_0\right)\right),\ldots,mlp\left(cat\left(t_{N_p},t_0\right)\right)\right\}\end{aligned} \tag{4}$$

The operation involves passing information from readout tokens to all other tokens by adding representations, and then connecting the readout to all other tokens. This is followed by projecting the tokens back to the original feature dimension $D$ using a linear layer, and then applying a GELU nonlinearity to disseminate the information to other tokens. After the Re*ad* module, the resulting $N_p$ tokens are reshaped into an image-like representation based on their initial positions in the image. Formally, a spatial cascading operation is applied to obtain a feature map of size $H$ × $W$ with $D$ channels.

$$\begin{aligned}&Concatenate:\mathbb{R}^{N_p\times D}\rightarrow\mathbb{R}^{\frac{H}{p}\times\frac{W}{p}\times D}\\&\mathrm{Re}\,sample_s:\mathbb{R}^{\frac{H}{p}\times\frac{W}{p}\times D}\rightarrow\mathbb{R}^{\frac{H}{s}\times\frac{W}{s}\times\hat{D}}\end{aligned} \tag{5}$$

Finally, this representation is passed to a spatial reception layer, which scales the representation to x times, ensuring that each pixel has $H/S$ × $W/S$ features. The final representation is half the resolution of the input image. It is then fed into a dual-output head network for semantic segmentation.

First, the semantic segmentation module of the DPT network identifies dynamic foreground elements in the image, such as vehicles and birds. This module outputs a probability map C. To better distinguish between the foreground and the background, depth estimation is performed using the DPT network. Depth estimation involves predicting the distance of each pixel in the image relative to the camera.

$$C_{(x,y)}=f_{seg}\left(I;\theta_{seg}\right),\ D_{(x,y)}=f_{depth}\left(I;\theta_{depth}\right) \tag{6}$$

Here, $C_{(x,\ y)}$ represents the probability that the pixel at location ($x$, $y$) belongs to the dynamic foreground and $D_{(x,\ y)}$ is the estimated depth of the pixel at ($x$, $y$) from the camera. $f_{seg}$ and $f_{depth}$ are the semantic segmentation functions, I is the input image, and $\theta_{seg}$ and $\theta_{depth}$ are the parameters of the semantic segmentation network.

Ultimately, combining the results of semantic segmentation and depth estimation allows for the removal of the dynamic foreground. The specific steps are as follows:

(1) Foreground Mask Generation

A binary mask is generated from the probability map C by setting a threshold T, indicating whether each pixel belongs to the dynamic scene.

(2) Depth Weighting

To more accurately remove the foreground, especially dynamic objects close to the camera, depth information is used to weight the mask. This ensures that closer dynamic objects are prioritized for removal.

$$M_{weighted}\left(x,y\right)=M\left(x,y\right)\cdot\frac{D_{max}-D\left(x,y\right)}{D_{max}} \tag{7}$$

Here, $D_{max}$ represents the maximum depth value in the scene.

(3) Image Reprojection

Finally, the original image $I$ is reprojected using the weighted mask $M_{weighted}$ to generate an image $I_{filtered}$ that has the dynamic foreground removed.

$$I_{filtered}\left(x,y\right)=\left(1-M_{weighted}\left(x,y\right)\right)\cdot I \tag{8}$$

By employing the aforementioned methods, dynamic foregrounds can be effectively removed in the 3D reconstruction of power grid towers, resulting in the creation of high-quality 3D models.

## 3.2 Boundaryless far-view blur constraint

### 3.2.1 Dual-spherical parameterization

In the 3D reconstruction of power grid towers, in addition to the interference from dynamic foregrounds, another critical challenge is addressing the blurriness of unbounded distant views. Traditional 3D reconstruction methods often result in blurred distant areas, leading to a loss of detail and reduced accuracy in the reconstruction.

To tackle the issue of distant blur, in this paper, we introduce an enhanced NeRF algorithm specifically tailored for the reconstruction of critical facilities such as power grid towers. The approach primarily involves two innovations: (1) a dual-spherical reparameterization technique and (2) a differential sampling method.

The dual-spherical reparameterization technique focuses on confining the geometric structure of the scene within a manageable space, allowing for a more effective handling of both near and distant views. In the context of power grid scenarios, this helps to concentrate on the details of the towers while applying a blur to the distant views.

First, two spherical surfaces are defined to constrain the reconstruction space: (1) the near plane: a sphere with radius $r = 1$, focused on capturing foreground elements such as power grid towers, and (2) the far plane: a sphere with radius $R \geq 1$, used to handle distant view information. Then, for each camera ray, on the basis of its relationship with the unit sphere, a reparameterization function called contract is applied. The role of this function is to map the ray onto the near or far spherical surface.

$$contract(r) = \begin{cases} r, & \text{if } \|r\| \leq 1 \\ \dfrac{R - \|r\|}{\|r\|} \cdot r, & \text{if } \|r\| > R \end{cases} \tag{9}$$

Here, $r$ represents the coordinate of the ray and $\|r\|$ is its Euclidean norm.

### 3.2.2 Differentiated sampling

By defining a dual-sphere parameterization method from the near plane to the far plane, each ray is parameterized accordingly. Subsequently, an multi-layer perceptron (MLP) is initialized, preparing it for learning the implicit representation of the scene. On the basis of the parameterized rays, a differentiable sampling strategy is implemented, achieving a sampling scheme where the foreground is densely sampled and the background is sparsely sampled.

The differential sampling method in NeRF focuses sampling density based on proximity. It concentrates more samples in the foreground, such as power grid towers, and fewer in the background. This is achieved by parameterizing camera rays to vary sampling density from a near plane to a far plane, resulting in denser sampling near the subject and sparser sampling in the distance.

$$s = \frac{g(t) - g(t_n)}{g(t_f) - g(t_n)} \tag{10}$$

Here, $t$ denotes the distance along the ray, with $\mathrm{t_n}$ and $\mathrm{t_f}$ representing the positions of the near and far planes, respectively. $g(t)$ is a defined invertible monotonically increasing mapping function that maps $t$ to a normalized ray distance $s \in [0, 1]$.

After parameterizing the ray, sampling points in the t space can be generated by uniformly sampling in the s space. This approach ensures a linear interval of sampling points in disparity, capturing more details in the foreground area, as shown in Fig 3.

$$t_i = g^{-1}\left(s_i \cdot g(t_f) + (1 - s_i) \cdot g(t_n)\right) \tag{11}$$

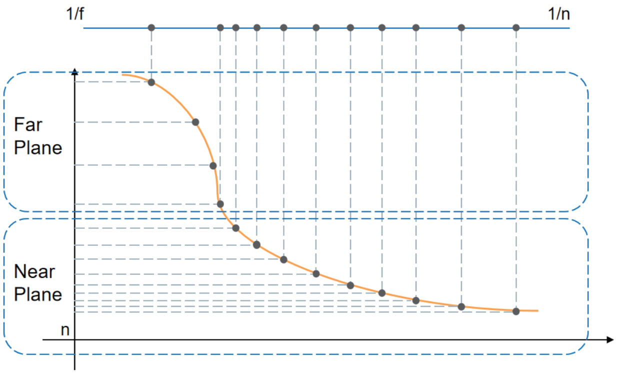

Fig. 3. (Color online) Differential sampling method illustration.

Here, $s_i$ represents the uniformly sampled points in the $s$ space, with $i = 0, 1, ..., K$. The corresponding sampling points $t_i$ in the $t$ space can be calculated using the inverse mapping function $g^{-1}$.

Ultimately, the preprocessed images and differentiable sampling points are fed into the MLP, which is trained through supervised learning. The model's predictions are optimized by minimizing the discrepancies between the model's predictions and the actual observations using a loss function, thereby refining the MLP parameters. With the trained MLP model, volumetric rendering is performed for each ray to synthesize new views. Along the rays, the pixel colors are calculated by integrating based on the volumetric density and emitted luminance output by the MLP, reconstructing the geometric shape of the power grid tower.

## 4. Experimental Design

### 4.1 Experimental data and environmental configuration

The experiment uses the TTPLA and STN PLAD datasets, both rich in high-resolution images for power transmission tower and line detection. The TTPLA dataset has 1100 images at 3840 × 2160 pixels with 8987 annotated towers and lines. The STN PLAD dataset adds 133 images at 5472 × 3078 or 5472 × 3648 pixels, tagging 2409 objects across five categories. The TPLA and STN PLAD datasets differ in image dimensions and annotation categories, allowing for a thorough assessment of the algorithm's adaptability and robustness in diverse scenarios. To maintain consistency and accuracy, all images underwent uniform preprocessing, including scaling, normalization, and enhancement, prior to model input. Network parameters were dynamically adjusted, with convolutional kernel sizes, strides, and paddings modified to suit images of varying resolutions.

Hardware includes an Intel Core i7-11700K CPU with 8 cores, 16 threads, and two NVIDIA GeForce RTX 3090 GPUs (24GB GDDR6X each), along with 64 GB of DDR4 RAM. Software comprises Windows 10, PyTorch 1.8.1, TensorFlow 2.4.1, CUDA 11.2, and cuDNN 8.1.0.

## 4.2 Evaluation metrics

The evaluation metrics for 3D reconstruction results based on NeRF mainly include three indices: PSNR, which is used to assess the quality of image reconstruction, with higher values indicating better reconstruction quality; SSIM and MS-SSIM, which are used to evaluate the visual quality of images, with higher values indicating greater structural similarity; and learned perceptual image patch similarity (LPIPS), which is used to assess the visual realism of image reconstruction, with lower values indicating that the reconstructed image is more similar to the real image. Since SSIM and MS-SSIM yield similar evaluation results, we selected only MS-SSIM as one of the evaluation metrics.

## 4.3 Experimental results and analysis

### 4.3.1 Dynamic foreground removal

The experiments initially focused on verifying the dynamic foreground removal module. By employing a semantic segmentation network, potential dynamic foregrounds were identified and masked. These regions were excluded during model training to eliminate their effect and to synthesize the missing information from different viewpoints. The effectiveness of this approach was evaluated through a comparative analysis of the original and synthesized images, as depicted in Fig. 4.

Figures 4(a) – 4(d) demonstrate that the new perspective images successfully removed pedestrians, as indicated by the red boxes, without compromising the depth structure. An ablation study was then conducted to qualitatively assess the module’s impact on 3D model reconstruction, with results shown in the accompanying Table 1.

The ablation study shows that the dynamic foreground removal module enhances performance across all metrics. With the module enabled, the PSNR improved by 2.22 for set (a) and 2.83 for set (b). The MS-SSIM saw an increase of 0.087 for set (a) and 0.028 for set (b),

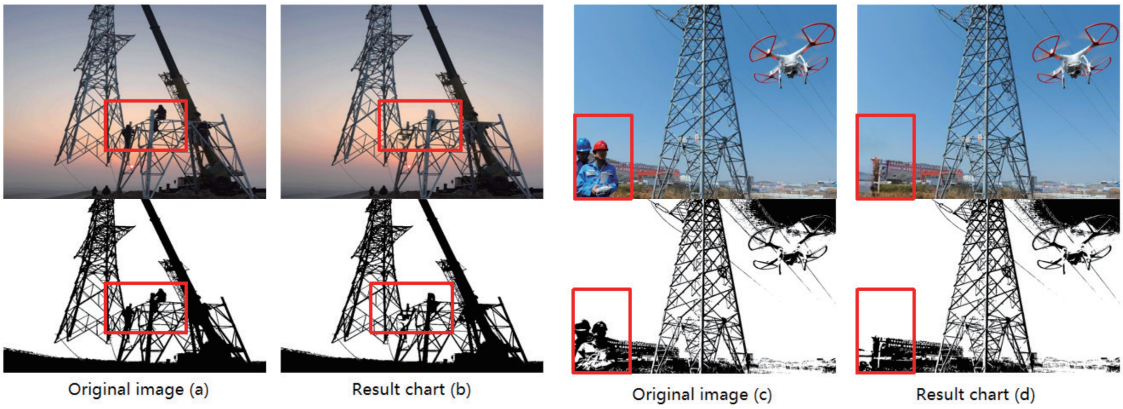

Fig. 4. (Color online) Comparison of depth maps between original and new synthetic perspective images.

Table 1
Evaluation of dynamic foreground removal ablation study results.

| Serial Number | Dynamic Foreground Removal | ↑PSNR | ↑MS-SSIM | ↓LPIPS |
|---|---|---|---|---|
| (a) | × | 15.63 | 0.542 | 0.529 |
| | √ | 17.85 | 0.629 | 0.475 |
| (b) | × | 14.96 | 0.764 | 0.361 |
| | √ | 17.43 | 0.792 | 0.254 |

whereas the LPIPS decreased by 0.054 for set (a) and 0.107 for set (b). This indicates that the module effectively removes dynamic foreground elements, improving the accuracy of the reconstruction.

### 4.3.2 Boundaryless far-view blur constraint

To validate the effectiveness of this module, we used an ablation study method to analyze the experimental results, as shown in Fig. 5 and Table 2.

In Fig. 5, the results with and without the boundaryless far-view blur constraint are compared. The red frames indicate the unoptimized results, whereas the yellow frames show the optimized ones. In group (a), trees in the background are blurred without the constraint, but become clear with it. In group (b), building structures and text are distorted without the constraint but are clearly defined with it.

The data in Table 2 demonstrate significant enhancements in all metrics with the module. Notably, in group (a), the PSNR improvement was substantial at 3.14, likely due to the abundance of background details. The module's use of differential sampling and dual-spherical reparameterization enriches sampling around the target's surface. In group (b), the module's detail-capture capabilities are evident, yielding a PSNR of 28.26, an increase of 3.43. The MS-SSIM and LPIPS values of 0.982 and 0.153, respectively, affirm the module's effectiveness.

## 4.4 Comparison of experimental results and analysis

To confirm the effectiveness of our algorithm, we compared it with the commonly used NeRF and NeRF-W. Figure 6 shows that NeRF struggles with background acquisition and detail extraction in complex scenes. NeRF-W, while an improvement, still exhibits issues with rendering distant backgrounds clearly. Our algorithm, however, captures more detailed reconstruction information and accurately predicts the depth of distant backgrounds, leading to clearer and more accurate background rendering.

In Fig. 6 and Table 3, our algorithm's results are compared with the results of NeRF and NeRF-W, showing our approach's superior image quality metrics. Notably, in the scenario (b) experiment, our algorithm demonstrated the most prominent results, achieving PSNR, MS-SSIM, and LPIPS values of 25.37, 0.642, and 0.299, respectively. Compared with the results of the NeRF algorithm, there were improvements of 10.68 in PSNR and 0.094 in MS-SSIM, whereas the LPIPS value decreased by 0.048. When compared with the results of the NeRF-W

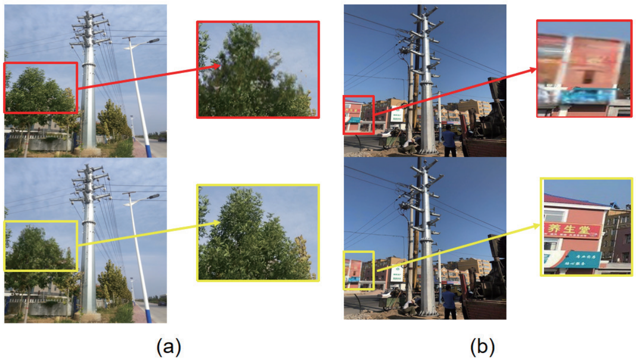

Fig. 5. (Color online) Boundaryless far-view blur constraint strategy ablation study results.

Table 2
Blur constraint ablation study evaluation results.

| Serial number | Dynamic foreground removal | ↑PSNR | ↑MS-SSIM | ↓LPIPS |
|---|---|---|---|---|
| (a) | × | 15.63 | 0.542 | 0.529 |
| | √ | 17.85 | 0.629 | 0.475 |
| (b) | × | 14.96 | 0.764 | 0.361 |
| | √ | 17.43 | 0.792 | 0.254 |

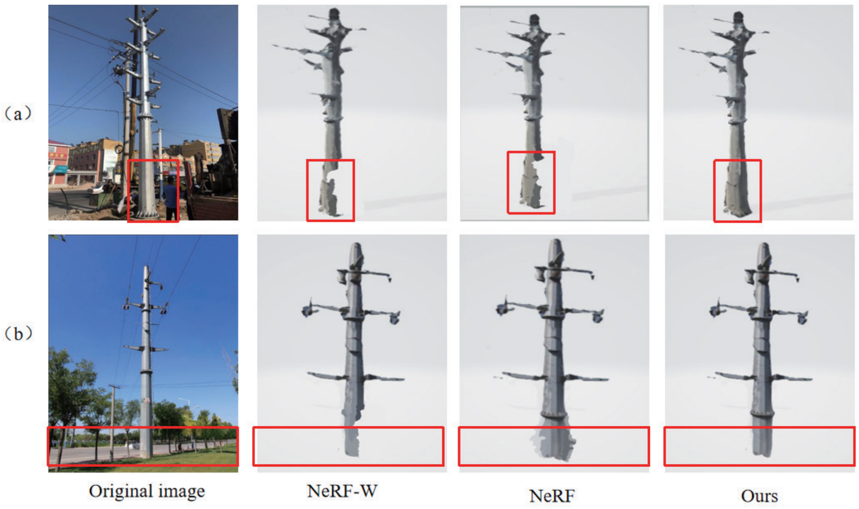

Fig. 6. (Color online) Algorithm comparison experimental results.

Table 3
Overall algorithm comparison experimental evaluation results.

| Scenes | Evaluation metrics | NeRF | NeRF-W | Ours |
|---|---|---|---|---|
| (a) | ↑PSNR | 20.51 | 23.68 | 27.34 |
| | ↑MS-SSIM | 0.692 | 0.605 | 0.582 |
| | ↓LPIPS | 0.535 | 0.674 | 0.512 |
| (b) | ↑PSNR | 14.69 | 21.07 | 25.37 |
| | ↑MS-SSIM | 0.548 | 0.591 | 0.642 |
| | ↓LPIPS | 0.293 | 0.291 | 0.245 |

algorithm, the PSNR and MS-SSIM values improved by 4.30 and 0.051, respectively, and the LPIPS value decreased by 0.046. These results indicate that our algorithm produces high-quality visual effects. The data suggest that our algorithm significantly outperforms both NeRF and NeRF-W, showing superior performance in 3D reconstruction tasks for complex outdoor scenes.

### 4.5 Crossover of experimental results and analysis

In this experiment, we aimed to evaluate the impact of different datasets (TPLA and STN PLAD) on the accuracy of a NeRF-based algorithm for the 3D reconstruction of power grid towers with foreground exclusion and to verify the algorithm's generalization capability. All images underwent a uniform preprocessing step before being input into the model, including scaling to a unified size, normalization, and enhancement, to ensure the consistency of the input data.

From Table 4, it can be observed that when the TPLA dataset is used for training and the STN PLAD dataset for testing, the PSNR is 25.80, the MS-SSIM is 0.864, and the LPIPS is 0.256. This indicates that the model performs relatively well on the STN PLAD dataset, likely due to the similarity in image size and scene complexity between the TPLA and STN PLAD datasets, which enables the model to generalize effectively to new datasets.

Conversely, when the model is trained on the STN PLAD dataset and tested on the TPLA dataset, the PSNR drops to 19.34, the MS-SSIM decreases to 0.783, and the LPIPS increases to 0.307. This suggests that the larger image sizes and more complex scenes in the STN PLAD dataset pose challenges for the model when generalizing to the smaller and less complex scenes of the TPLA dataset.

However, when the model is trained and tested on the same dataset, its performance improves. For instance, the model trained and tested with the STN PLAD dataset achieves a PSNR of 28.16, an MS-SSIM of 0.931, and an LPIPS of 0.239, demonstrating that the model performs optimally when dealing with test data similar to the training data.

Furthermore, the model trained with the TPLA dataset and tested on the STN PLAD dataset shows a PSNR of 25.47, an MS-SSIM of 0.860, and an LPIPS of 0.284, indicating that the model can still maintain a certain level of performance when handling different datasets, although not as high as when trained and tested on the same dataset.

The experimental results demonstrate that, despite differences in image size and annotation categories, our algorithm can still maintain high accuracy and robustness. This is attributed to a series of measures we have taken in the algorithm design, such as the uniform preprocessing and dynamic adjustment of network parameters.

Table 4
Crossover of experimental results.

| Datasets | Training set | Test set | PSNR↑ | MS-SSIM↑ | LPIPS↓ |
|---|---|---|---|---|---|
| TPLA | TPLA | STN PLAD | **25.80** | **0.864** | **0.256** |
| | STN PLAD | TPLA | 19.34 | 0.783 | 0.307 |
| STN PLAD | STN PLAD | TPLA | 28.16 | 0.931 | 0.239 |
| | TPLA | STN PLAD | 25.47 | 0.860 | 0.284 |

## 5. Conclusions

NeRF technology is instrumental in 3D reconstruction for power grid infrastructure, enhancing planning, maintenance, and disaster recovery. It also facilitates immersive training through virtual and augmented reality.

In this research, we introduced an enhanced NeRF algorithm tailored for the complex outdoor reconstruction of power towers. By employing dynamic foreground removal and managing far-view blur without boundaries, the algorithm achieves clearer and more accurate 3D models. It surpasses traditional NeRF and NeRF-W in capturing details and rendering backgrounds, proving effective in complex reconstruction tasks.

Looking ahead, we aim to optimize the algorithm for faster processing and greater precision, especially in larger scenes. We will also test its adaptability across diverse power grid facilities and environments. Future efforts will include real-time reconstruction, multimodal data integration, and improving the machine learning models' adaptability to further elevate the technology in 3D reconstruction for critical power grid infrastructure.

## Acknowledgments

This study was supported by the Science and Technology Project of State Grid Corporation of China: Research and Application of Multi-source Collaborative and Dynamic Data Sharing Technology for Power Grid Engineering Survey Data (5700-202356317A- 1-1-ZN).

## References

1 Y. Zhu, R. Jin, and L. Zhao: Acta Astronaut. **42** (2021) 1271. https://doi.org/10.3873/j.issn.1000-1328.2021.10.008
2 Y. Xue, S. Zhang, M. Zhou, and H. Zhu: Undergr. Space. **6** (2021) 134. https://doi.org/10.1016/j.undsp.2020.01.002
3 R. Zlot and M. Bosse: Proc. IEEE Int. Conf. Field Serv. Robotics **1** (2014) 479. https://doi.org/10.1007/978-3-642-40686-7_32
4 R. S. Pahwa, K. Y. Chan, J. Bai, V. B. Saputra, M. N. Do, and S. Foong: Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS) (IEEE,2019) 7025–7032. https://doi.org/10.1109/IROS40897.2019.8967577
5 J. A. Albert, V. Owolabi, A. Gebel, C. M. Brahms, U. Granacher, and B. Arnrich: Sensors. **20** (2020) 5104. https://doi.org/10.3390/s20185104
6 X. Y and G. Jiang: Remote Sens. **13** (2021) 16. https://doi.org/10.3390/rs13163103
7 J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid: Int. J. Comput. Vis. **120** (2016) 300. https://doi.org/10.1007/s11263-016-0908-3
8 D. DeTone, T. Malisiewicz, and A. Rabinovich: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW) (IEEE,2018). https://doi.org/10.1109/CVPRW.2018.00060
9 P. E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR) (IEEE,2020). https://doi.org/10.1109/cvpr42600.2020.00499
10 J. Sun, Z. Shen, Y. Wang, and X. Zhou: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR) (IEEE,2021). https://doi.org/10.1109/CVPR46437.2021.00881
11 P. H. Huang, K. Matzen, J. Kopf, N. Ahuja, and J. B. Huang: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR) (IEEE,2018) 2821–2830. https://doi.org/10.1109/CVPR.2018.00298
12 Y. Yao, Z. Luo, S. Li, and F. Tian: Lect. Notes Comput. Sci. (ECCV) (IEEE,2018) 785–801. https://doi.org/10.48550/arXiv.1804.02505
13 Z. Wu, S. Song, A. Khosla, F. Yu, L. Z, X. T, and J. X: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) (IEEE,2015) 1912–1920. https://doi.org/10.48550/arXiv.1406.5670

14 D. Maturana and S. Scherer: Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS) (IEEE,2015) 922–928. https://doi.org/10.1109/IROS.2015.7353481

15 H. Fan, H. Su, and L. Guibas: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) (IEEE,2017). https://doi.org/10.48550/arXiv.1612.00603

16 Y. Yang, F. Chen, S. Yiru, and T. Dong: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR) (IEEE,2018) 206–215. https://doi.org/10.48550/arXiv.1712.07262

17 N. Wang, Y. Zhang, Z. Li, Y. Fu, H. Yu, W. Liu, X. Xue, and Y. Jiang: IEEE Trans. Pattern Anal. Mach. Intell. **43** (2021) 3600–3613. https://doi.org/10.1109/TPAMI.2020.2984232

18 J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR) (IEEE,2019) 165–174. https://doi.org/10.1109/CVPR.2019.00025

19 S. Saito, T. Simon, J. M. Saragih, and H. Joo: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR) (IEEE,2020) 81–90. https://doi.org/10.1109/cvpr42600.2020.00016

20 Z. Chen and H. Zhang: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR) (IEEE,2019). https://doi.org/10.1109/CVPR.2019.00609

21 L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR) (IEEE,2019) 4455–4465. https://doi.org/10.1109/CVPR.2019.00459.