TextNeRF: A Novel Scene-Text Image Synthesis Method based on Neural Radiance Fields

Jialei Cui[1], Jianwei Du[2], Wenzhuo Liu[1], Zhouhui Lian[1]*
[1] Wangxuan Institute of Computer Technology, Peking University, China, [2] Southeast University, China

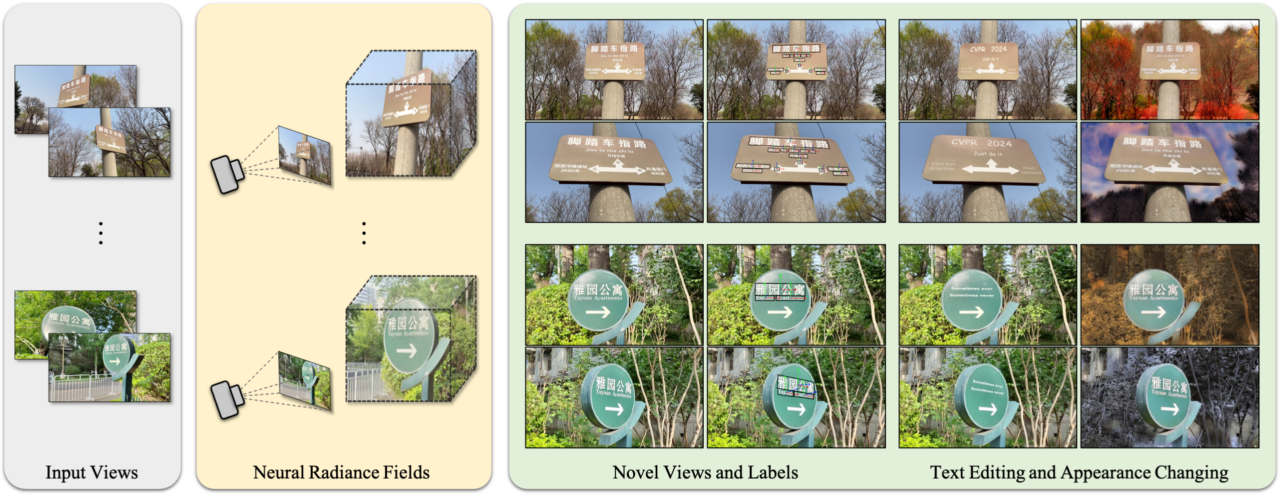


Figure 1. We propose a novel method that leverages NeRF to model real-world scenes and emulate the data collection process by rendering novel views. In addition, detailed annotations including boxes and poses of each text are provided and photo-realistic text editing and appearance changing are also achieved based on our method's powerful geometric modeling capabilities.

Abstract

Acquiring large-scale, well-annotated datasets is essential for training robust scene text detectors, yet the process is often resource-intensive and time-consuming. While some efforts have been made to explore the synthesis of scene text images, a notable gap remains between synthetic and authentic data. In this paper, we introduce a novel method that utilizes Neural Radiance Fields (NeRF) to model real-world scenes and emulate the data collection process by rendering images from diverse camera perspectives, enriching the variability and realism of the synthesized data. A semi-supervised learning framework is proposed to categorize semantic regions within 3D scenes, ensuring consistent labeling of text regions across various viewpoints. Our method also models the pose, and view-dependent appearance of text regions, thereby offering precise control over camera poses and significantly improving the realism of text insertion and editing within scenes. Employing our technique on real-world scenes has led to the creation of a novel scene text image dataset (https://github.com/cuijl-ai/TextNeRF). Compared to other existing benchmarks, the proposed dataset is distinctive in providing not only standard annotations such as bounding boxes and transcriptions but also the information of 3D pose attributes for text regions, enabling a more detailed evaluation of the robustness of text detection algorithms. Through extensive experiments, we demonstrate the effectiveness of our proposed method in enhancing the performance of scene text detectors.

1. Introduction

The detection of text in natural images are pivotal in advancing numerous computer vision applications, including industrial automation, image retrieval, robot navigation, and autonomous driving. Despite the remarkable progress in scene text detection, the inherent complexity of natural scenes and the diverse manifestations of text present are still

*Corresponding author. E-mail: lianzhouhui@pku.edu.cn

This work was supported by National Natural Science Foundation of China (Grant No.: 62372015), Center For Chinese Font Design and Research, and Key Laboratory of Intelligent Press Media Technology.

considerable challenges. These challenges necessitate large amounts of annotated training data for algorithm optimization. While public datasets [8, 9, 21] are available, they often fall short in capturing the eclectic scenarios present in real-world environments. This gap underscores the need for tailored training data collection and annotation, a process that is, however, costly and labor-intensive, particularly for deep learning models.

To address this problem, researchers have turned to synthetic data generation. Existing methods [5, 6, 14, 16, 35] mainly rely on 2D static backgrounds or 3D graphics engines for text image synthesis. Methods based on 2D static backgrounds employ text placement and integration techniques such as region selection, text warping, and color matching but suffer from limited flexibility in data synthesis and challenges in selecting appropriate backgrounds. On the other hand, 3D graphics engine-based methods offer greater control but still struggle to replicate the nuanced complexity of real-world scenes.

Neural Radiance Fields (NeRF) [19], an emerging neural rendering technique, provides an exciting avenue for realistic viewpoint synthesis by correlating spatial coordinates and viewing angles to densities and radiances. NeRF uses multi-layer perceptrons (MLPs) to represent scenes implicitly and enables high-fidelity image synthesis from novel viewpoints using volume rendering. In this paper, we develop a novel scene text image synthesis method that authentically reconstructs scenes with NeRF and mimics the image acquisition process to synthesize scene text images. By leveraging NeRF's geometric modeling capabilities, we can evaluate the structural importance of each position in a scene across different viewpoints and facilitate targeted style transfers to enhance scene appearance. Additionally, we employ a semi-supervised learning paradigm to achieve the semantic modeling of local regions in 3D scenes, ensuring consistent labeling of text areas across multiple viewpoints. Building upon the semantic modeling of 3D regions, we further model the surface poses of text areas, allowing precise manipulation of text-to-camera positioning and the determination of necessary camera poses for image synthesis. This methodology not only allows for more photo-realistic text editing in potential scene areas but also broadens the diversity of synthetic data.

Our approach surpasses previous methods in the following two aspects: Firstly, it reconstructs actual scenes, offering greater flexibility than 3D graphics engine-based methods. Secondly, it embeds text in genuine 3D space surfaces rather than superimposing it on 2D images, which results in synthetic effects that are more natural and visually pleasing. To verify the efficacy of our approach, we create a dataset of photo-realistic synthetic scene text images. The annotation of this dataset not only includes conventional bounding boxes and transcription attributes but also provides each text instance with its 3D pose relative to the camera, enabling a more nuanced assessment of text detectors' robustness to text perspective effects. To the best of our knowledge, this is the first dataset to offer 3D pose attributes for text. We employ several representative text detectors in extensive experiments from training and test perspectives, illustrating the effectiveness of the proposed method in enhancing scene text detecting performance and revealing their limitations.

The main contributions of our paper are threefold: (1) We introduce a novel scene text synthesis method that harnesses the strengths of NeRF, enabling photo-realistically scene text image synthesis with 3D controllability. (2) Our semi-supervised learning framework for semantic modeling of 3D local regions ensures multi-view consistency of generated text labels and provides the pose information of text instances, thereby facilitating more accurate scene text editing and camera positioning during synthesis. (3) We propose the first scene text image dataset with 3D pose attributes for each text instance, enabling a more detailed evaluation of text detectors concerning perspective effects.

2. RELATED WORK

2.1. Scene Text Detection

Recent advancements in scene text detection are primarily attributed to the integration of deep learning technologies [17]. Generally, the scene text detection methods can be categorized into three main branches: regression-based, part-based, and segmentation-based methods. Regression-based methods, such as those introduced in TextBoxes++ [12] and EAST [39], have shown remarkable efficiency by directly estimating the coordinates of text bounding boxes. Part-based approaches, exemplified by SegLink [26], focus on identifying smaller, manageable segments of text and then heuristically assembling them to form complete text instances. Segmentation-based techniques, like [15, 24, 27, 30, 33, 37], perform pixel-level classification to discern text regions, followed by sophisticated post-processing to delineate text boundaries. These methods benefit from large, annotated datasets, which educate the models on the diverse manifestations of text in natural scenes.

2.2. Scene Text Image Synthesis

The paucity of annotated training images has led researchers to craft synthetic datasets, such as MJSynth [7] and SynthText [5], which generate text images by overlaying textual content onto backgrounds. These datasets, however, often lack contextual realism. Zhan et al. [35] proposed an enhanced approach that utilizes semantic segmentation to ensure text is only placed on contextually appropriate surfaces. The advent of 3D-based synthetic data generation, as seen in SynthText3D [14] and UnrealText [16], offers greater control over environmental variables, producing more lifelike synthetic images that aid in training robust detection models. Nonetheless, the efficacy of such synthetic data is contingent upon the authenticity of the 3D models, the capabilities of the rendering engines, and the diversity of the text fonts used.

2.3. Neural Radiance Fields

The NeRF framework [19] has revolutionized the rendering of 3D scenes, enabling photorealistic image synthesis from novel viewpoints. Its extensions, such as NeRF++ [36] and Mip-NeRF [1, 2], have addressed specific challenges like rendering unbounded scenes and anti-aliasing. To mitigate NeRF's computational intensity, approaches like DVGO [25] and TensorRF [3] have been proposed, significantly accelerating the training and rendering process. Plenoxels [34] and Instant-NGP [20] further advance this by dispensing with neural networks altogether or optimizing data structures, achieving training times on the order of minutes and enabling nearly real-time rendering.

NeRF's applicability extends beyond static scene rendering, applying it to dynamic reconstructions of humans and faces by [11, 31]. Its inherent capabilities for capturing geometric and appearance nuances make it a powerful tool for semantic scene understanding, as demonstrated in [28, 29]. Moreover, NeRF showcases its versatility and transformative impact across diverse fields, including medical imaging[4], satellite imagery[18], and robotic perception[40].

3. METHOD

3.1. Scene Modeling

NeRF for scene modeling. We construct a scene's parametric representation using a Neural Radiance Field (NeRF) [19], which enables photo-realistic rendering from multiple images captured at known camera poses. The scene is encoded in a neural field, where a function $F_\theta(\mathbf{x}, \mathbf{d})$ maps a spatial location $\mathbf{x}$ and a viewing direction $\mathbf{d}$ to the corresponding density $\sigma \in \mathbb{R}$ and RGB color $\mathbf{c} \in \mathbb{R}^3$. To render an image from a specific viewpoint, we cast rays corresponding to the camera's pose and integrate the radiance along these rays to compute the final image:

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i \delta_i))\mathbf{c}_i, \tag{1}$$

where $T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j)$ represents the amount of lights transmitted through the ray r to the sample i, $\mathbf{c}_i$ is the color of the sample i, and δ_i is the distance to the next sample. The differentiable nature of NeRF facilitates its training by minimizing the difference between the predicted and ground-truth colors of the rays $\mathcal{R}$:

$$\mathcal{L}_{color} = \sum_{\mathbf{r} \in \mathcal{R}} \left|\left| \hat{\mathbf{C}}(\mathbf{r}) - \mathbf{C}(\mathbf{r}) \right|\right|_2^2. \tag{2}$$

To enhance the geometric consistency of the reconstructed scenes, we add the distortion loss [2] as a regularizing term to encourage a more focused distribution of weights along the rays:

$$\mathcal{L}_{dist}(\mathbf{s}, \boldsymbol{\omega}) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \omega_i \omega_j \left| \frac{s_i + s_{i+1}}{2} - \frac{s_j + s_{j+1}}{2} \right| + \frac{1}{3} \sum_{i=0}^{N-1} \omega_i^2 (s_{i+1} - s_i), \tag{3}$$

where $\mathbf{s}$ is a vector of the distance from all sample points to the ray origin, and ω is a vector of the weights of all sample points on the ray. Furthermore, we adopt the novel training paradigm, S3IM [32], which extends beyond traditional pixel-wise Mean Squared Error (MSE) by considering the structural coherence within a set of input images:

$$\mathcal{L}_{\mathrm{S3IM}} = 1 - \frac{1}{M} \sum_{m=1}^{M} \mathrm{SSIM}(\mathcal{P}^{(\mathrm{m})}(\hat{\mathbf{C}}), \mathcal{P}^{\mathrm{m}}(\mathbf{C})), \tag{4}$$

where SSIM is the structural similarity, $\mathcal{P}(\mathbf{C})$ is a patch randomly formed from a batch of rays/pixels, and M is the repeat times of SSIM computing. The incorporation of the S3IM loss significantly improves the NeRF model's ability to capture nonlocal structural relationships within the scene. Thus, our NeRF's loss function is a composite of the color loss, the distortion loss, and the S3IM loss:

$$\mathcal{L} = \mathcal{L}_{color} + \lambda_{dist} \mathcal{L}_{dist} + \lambda_{\mathrm{S3IM}} \mathcal{L}_{\mathrm{S3IM}}, \tag{5}$$

where λ_{dist} and λ_{S3IM} denote balance weights for the distortion loss and the S3IM loss, respectively. We leverage instant-NGP [20] for its efficiency and flexibility as our base representation to reconstruct scenes.

Appearance changing. To diversify the synthesized images, we implement an appearance editing operation on the NeRF renderings. Instead of modifying the radiance fields directly which is computationally demanding, we apply style transfer on each rendered image. We observe that many style transfer methods are prone to introducing artifacts that can compromise the realism of the images, particularly in text regions. Therefore, we introduce a novel technique to preserve the scene's geometric structures while performing style transfer. We hypothesize that the importance of a pixel for structural representation is inversely proportional to its frequency of occurrence across multiple views. Hence, we utilize a frequency-based structural importance

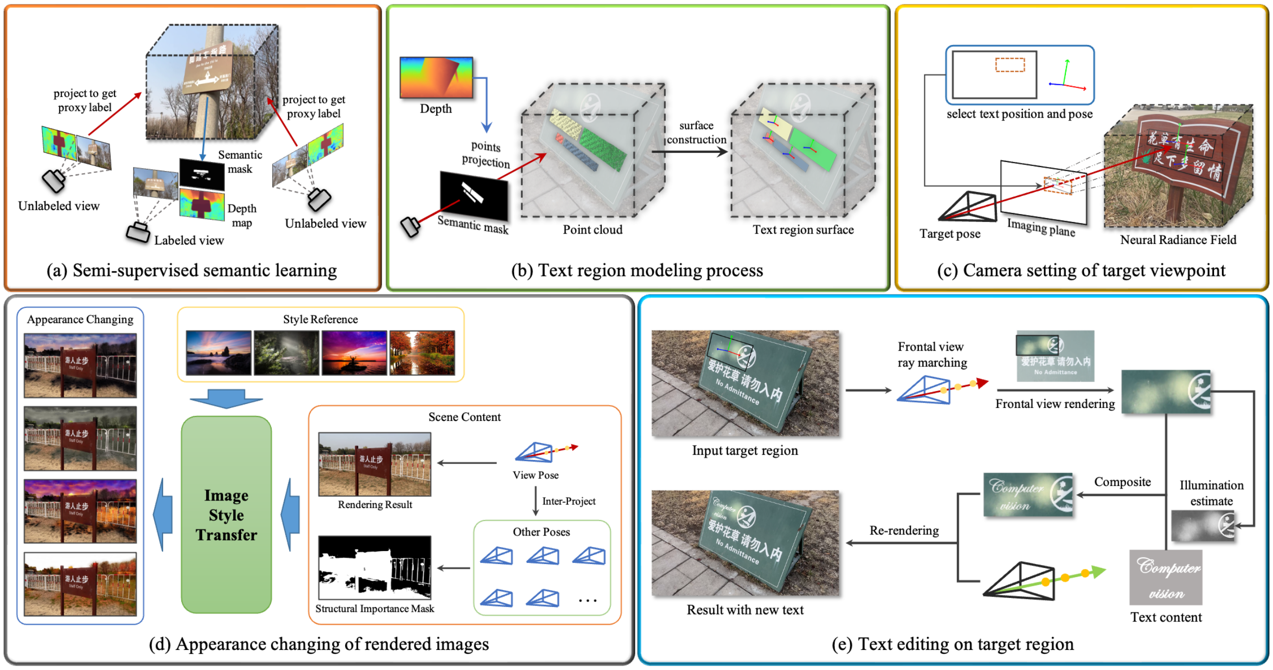


Figure 2. Important components of our method. Based on NeRF's high-quality modeling of the scenes, we respectively implement (a) semi-supervised semantic learning across multi-views, (b) modeling text regions, (c) controllable camera setting for rendering target viewpoint, (d) changing the appearance of rendered images, and (e) photo-realistic text editing on target region.

mask that dictates the degree of participation in style transfer for each pixel. To generate this mask, we adopt the depth information produced by NeRF to assess the frequency of pixel occurrences across different views, thereby creating a mask that identifies structurally important regions within the image. Specifically, we manipulate the projection of pixels between different views to calculate the frequency of each pixel's occurrence across views:

$$\begin{aligned}\mathbf{p}_{c_t} &= F^{\text{proj}}_{c_s \to c_t}(\mathbf{p}_{c_s}) \\ &= d_{c_s}\mathbf{K}_{c_t}\mathbf{P}^{-1}_{c_t}\mathbf{P}_{c_s}\mathbf{K}^{-1}_{c_s}\mathbf{p}_{c_s},\end{aligned} \tag{6}$$

where $\mathbf{p}_{c_s}$ and $\mathbf{p}_{c_t}$ denote the corresponding points in the source and target images, $\mathbf{K}_{c_s}$ and $\mathbf{K}_{c_t}$ are the intrinsic camera matrices, and $\mathbf{P}_{c_s}$ and $\mathbf{P}_{c_t}$ are the camera poses, respectively. Here, the conversion between European coordinates and homogeneous coordinates has been omitted. The depth d_{c_s} of the point $\mathbf{p}_{c_s}$ is obtained via volume rendering along the ray's samples. We map a pixel point in the reference view onto a target view to validate its correspondence. Furthermore, the corresponding point on the target view will be re-projected back onto the original view, fortifying the geometric consistency across multiple viewpoints.

We employ PhotoWCT [10] as the style transfer method for its efficiency, and we carefully select style images that introduce distinct seasonal and climatic attributes. In this manner, we can maintain the semantic integrity of a scene, including legibility in text regions, while transforming the environmental appearance of the images.

3.2. Text Region Modeling

Position-enhanced semantic representation of text regions. To accurately delineate text regions within a scene, we equip our NeRF with a region tailored semantic renderer and formalize the task as a view-invariant function, as in [38]. Unlike physical objects, text regions represent an artificial categorization of the scene surface and lack direct correspondence with physical attributes. This challenges the assumption in [38], where entities of similar shape and appearance are likely to belong to the same class. To address this issue, we add an additional positional encoding of world coordinates with the input NeRF-features to the semantic renderer. This empowers the renderer to discern positional differences within geometrically similar regions, yielding the improved delineation of text regions. With the positionally enhanced semantic renderer integrated, our NeRF's representation of the scene becomes:

$$\begin{aligned}(\sigma, \mathbf{z}) &= \mathcal{F}_{\theta_1}(\gamma_{x_1}(\mathbf{x}), \mathbf{d}), \\ \mathbf{c} &= \mathcal{F}_{\theta_2}(\gamma_d(\mathbf{d}), \mathbf{z}), \\ \mathbf{s} &= \mathcal{F}_{\theta_3}(\gamma_{x_2}(\mathbf{x}), \mathbf{z}).\end{aligned} \tag{7}$$

Here, $\gamma_{x_1}(\cdot)$, $\gamma_d(\cdot)$ and $\gamma_{x_2}(\cdot)$ represent the positional encoding functions for the position coordinates $\mathbf{x}$ and viewing direction $\mathbf{d}$, which in this work are specifically the hash encoding, spherical harmonics and triangular wave function, respectively. Therefore, it is feasible to employ volume rendering to get the semantic label distribution along a ray:

$$\hat{\mathbf{S}}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i \delta_i))\mathbf{s}_i, \tag{8}$$

where $\mathbf{s}_i$ denotes the semantic probability distribution of a sample, and the other terms are as defined in Sec 3.1. This semantic representation is also employed to build custom regions of the scene where we want to add text.

Semi-supervised learning by geometric consistency. As shown in Fig. 2(a), our training methodology capitalizes on NeRF's geometric modeling strengths by implementing a semi-supervised learning framework that minimizes the need for exhaustive semantic annotations. We annotate a sparse set of regions in a subset of training images and directly supervise the semantic renderer with these labels. For images without explicit annotations, we harness the geometric consistency inherent to NeRF to project semantic labels across views. By establishing pixel-wise correspondences through Eq. 6, we generate proxy labels for unannotated images, allowing for effective learning of text semantics across the scene:

$$\mathbf{S}_i^{proj}(\mathbf{p}_i) = \mathbf{S}_j^{gt}(F_{i \to j}^{\text{proj}}(\mathbf{p}_i)). \tag{9}$$

The semantic loss is defined to encompass both labeled and unlabeled data. For rays intersecting labeled semantic categories, a cross-entropy loss is directly applied, while for rays without explicit semantic labels, proxy labels are obtained through the aforementioned projection to calculate the cross-entropy loss:

$$\mathcal{L}_{sem} = -\sum_{i \in \mathcal{R}_l} \mathbf{S}_i \log(\hat{\mathbf{S}}(r_i)) - \lambda_u \sum_{i \in \mathcal{R}_u} \mathbf{S}_i^{proj} \log(\hat{\mathbf{S}}(r_i)), \tag{10}$$

where λ_u is a balancing weight for unlabeled data.

Text region labeling and editing. Upon completing the semantic modeling of text regions, our approach allows for the precise extraction of text area contours in the target view. Unlike traditional methods, our method delineates the text contours in 3D space by leveraging the geometry information obtained from NeRF, and then projects them onto target 2D images. As shown in Fig. 2(b), we project the pixels within the predicted text semantic regions from each rendered view into 3D space through Eq. 11, generating a point cloud representing the text surface:

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} = K \times d_i \times \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}, \tag{11}$$

where (x_i, y_i) represents a pixel location in the 2D image, d_i is the associated depth value, K is the camera's intrinsic parameter matrix, and (X_i, Y_i, Z_i) corresponds to the 3D coordinates in the scene. To ensure a noise-free and smooth representation, we employ a bilateral filter for point cloud denoising, followed by a moving least squares smoothing technique. For the 3D surface fitting, we implement an iterative least squares method which minimizes the sum of squared distances between the observed points and the surface. The minimum bounding rectangle for each text is derived from the 3D fitted surface, ensuring alignment with the quadrilateral annotations typical in public datasets.

Next, we define the 3D pose of the text regions, which comprises a rotation matrix ($\mathbf{R}$) and a translation vector ($\mathbf{t}$). To compute the rotation matrix, we employ a standard axis-angle representation. We first calculate the 3D normal vector of the text surface, which becomes the z-axis. The long and short sides of the bounding rectangle define the x and y axes respectively. Finally, we compute the centroid of the region and use it as the translation vector, completing the 3D pose representation of the text. We then project the 3D pose and quadrilateral bounding box onto various camera views to obtain the corresponding 2D annotations.

With the 3D geometric representation of a text region, we can readily edit its textual content by rendering its frontal view. Since NeRF does not decouple view-dependent attributes such as shading and shadows from radiance, we cannot modify only the text area in one frontal view. Therefore, we simulate the view-dependent appearance by fixing the camera viewpoint and changing the queried viewing direction. All text editing operations are performed on a series of frontal views with different appearances. For modifications of existing text, we leverage a stable diffusion model [22] to erase the original text content according to its mask, followed by adding new text to the erased area. For adding new text to a custom frontal view region, as shown in Fig. 2(e), we first estimate its illumination and apply it to a new plain text patch image. Then we blend the text patch with the frontal view image to obtain the edited result. We make slight adjustments to the rendering strategy when rendering an image with edited text regions. We treat rays passing through the text regions separately, replacing their color with those of the edited result, while retaining the original volume rendering approach for the remaining rays to achieve a realistic editing effect.

3.3. Controllable Image Synthesis

Camera pose setting. To controllably synthesize images, we develop a systematic approach for camera pose manipulation that ensures accurate perspective effects of text regions. This method primarily addresses two critical aspects: achieving a uniform distribution of text orientations within the synthetic dataset and enabling the generation of specific images to uncover and mitigate model biases, thus contributing to the development of more robust text detection models.

For precise implementation, we present a step-by-step algorithm that circumvents the complexity of directly computing camera poses: **Text Position Selection**: We first define the desired text location on the image plane. **Ray Projection**: A ray is projected from this position through the center of the text region to establish the direction vector relative to the camera. **Target Orientation**: The target orientation for the text region is set by calculating the rotation matrix $\mathbf{R}$ that defines the text's pose relative to the camera. **Area Proportion Adjustment**: We control the text's area proportion on the image by scaling the text region's projection, which in turn determines its depth from the camera. Once the text region's pose is established in camera coordinates, we compute the transformation matrix to convert this pose into the actual scene's text region pose. Applying this transformation to the camera achieves the required camera pose for rendering the scene from the target viewpoint. Fig. 2 (c) shows the detailed camera pose establishment process.

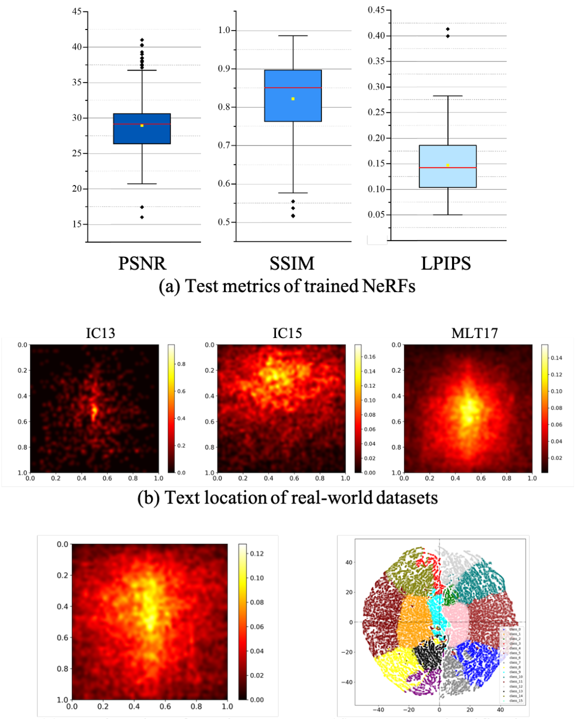
4. EXPERIMENTS

In this section, we conducted extensive experiments on both synthetic and real-world scene text datasets. Our method was implemented to create a diverse set of images derived from real scene text scenarios. The effectiveness of each synthesis method was evaluated by training text detectors on these synthesized images and testing them on real-world datasets. A distinctive feature of our approach is the capability to generate images from multiple viewpoints within a single scene, thereby offering a more detailed analysis of detector performance to perspective transformations. Furthermore, we conducted ablation studies to assess the impact of the structural importance mask on appearance changing and the role of positional encoding in text region delineation.

4.1. Datasets.

Synthetic Datasets: We selected three synthetic datasets for comparison. (1) The Verisimilar Image Synthesis Dataset (VISD) [35] comprises 10,000 images synthesized using background images from the COCO dataset. (2) The Synthetic Image from 3D Virtual World (SynthText3D) [14] includes 10,000 images generated from 30 virtual scenes. (3) The UnrealText (UT) [16] dataset initially provided 728,000 English/Latin images, yet our quality control measures excluded images lacking text or predominantly black. Consequently, we sampled 10,000 images to align with the aforementioned datasets in quantity.

Real-world Datasets: The evaluation utilized three popular benchmark datasets: (1) ICDAR 2013 [8] (IC13), featuring horizontally-oriented text; (2) ICDAR 2015 [9] (IC15), consisting of incidental scene text with diverse conditions; and (3) MLT 2017 [21], focused on multilingual scene text detection across nine languages and six scripts.



(a) Test metrics of trained NeRFs

(b) Text location of real-world datasets

(c) Text location of our dataset

(d) Text pose classification

Figure 3. (a): The test metrics of trained NeRFs. (b): The spatial text occurrence probability across the three real-world datasets. (c): Text location distribution used during synthesis. (d): The t-SNE reduced dimensionality visualization of clustered text poses.

Our Synthetic Dataset: A collection of 300 real-world scene text videos was collected as the foundation for our dataset construction. Frames compromised by blurriness were excluded, yielding between 50 to 80 sharp images for each scene. Among them, 85% were allocated for training our NeRFs, with the remaining 15% reserved for testing, aligning with practices commonly adopted in NeRF-related literature [2, 19, 36]. Camera parameters for each frame were estimated utilizing COLMAP [23]. A subset of images, ranging from 1 to 3 per scene were annotated to define text regions. Individual scenes underwent training for 15 to 30 epochs, averaging between 10 to 30 minutes per scene. During the scene modeling phase, we configured the loss weight parameters as follows: λ_{dist} and λ_{S3IM} were set to 0.001 and 1.0, respectively, while λ_u was adjusted to 0.3 in the text modeling phase. To assess the quality of training across various scenes, we computed the PSNR, SSIM, and LPIPS metrics for all NeRF models, obtaining average values of 28.96, 0.822, and 0.147, respectively. A box plot distribution of these metrics across all test images from different scenes, as illustrated in Fig. 3(a), confirms the consistency of NeRF training with our expectations.

Train dataset	DB									EAST								
	ICI3			ICI5			MLT17			ICI3			ICI5			MLT17		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
VISD-10k	71.41	66.39	68.81	70.89	55.22	62.08	56.49	36.70	44.49	69.13	**68.44**	68.78	65.96	63.99	64.96	46.73	40.53	43.41
ST3D-10k	71.31	64.47	67.72	75.84	57.73	65.56	59.87	41.71	49.17	69.40	54.50	61.05	**74.01**	60.04	66.30	54.63	43.40	48.37
UT-10k	**79.69**	51.60	62.64	**76.40**	48.77	59.54	**69.08**	39.70	50.42	**76.58**	64.20	69.85	64.98	52.96	58.36	54.99	47.93	51.22
Our-10k	73.22	**67.44**	**70.21**	74.91	**59.64**	**66.41**	64.13	**46.22**	**53.72**	74.15	66.52	**70.13**	71.44	**66.21**	**68.73**	**60.93**	**47.24**	**53.22**
Real	72.66	46.12	56.42	84.11	76.94	80.37	70.34	56.09	62.41	65.48	46.75	54.55	76.38	82.38	79.27	68.50	54.32	60.59
VISD-10k + Real	84.67	69.59	76.39	85.98	81.22	83.53	**75.50**	54.44	63.26	77.53	63.37	69.74	84.81	85.22	85.01	**73.84**	54.13	62.47
ST3D-10k + Real	83.19	72.33	77.38	87.27	80.21	83.59	70.70	56.67	62.91	70.06	68.69	69.37	74.49	80.26	77.27	70.24	55.44	61.97
UT-10k + Real	85.52	70.68	77.40	89.51	80.50	84.77	71.98	58.51	64.55	76.07	70.33	73.09	**88.77**	83.00	85.79	71.36	57.62	63.76
Ours-10k + Real	**86.28**	**73.09**	**79.14**	**89.75**	**81.87**	**85.63**	70.15	**60.43**	**64.93**	**81.67**	**71.89**	**76.47**	86.30	**85.46**	**85.88**	70.20	**59.28**	**64.28**

Table 1. Comparison between previous synthetic datasets and our dataset on the ICDAR2013, ICDAR2015, ICDAR2017MLT datasets. **R**: Recall, **P**: Precision, **F**: F-score, **Real**: the corresponding training set of the evaluation dataset.

After training, the NeRFs were harnessed to render images, where camera poses were established as described in Sec 3.3. To bridge the gap with real-world data, we analyzed text location distributions across three real datasets, providing a prior for text placement in our synthetic dataset. To streamline camera pose configuration for synthetic images, a clustering analysis was performed on the text poses by employing k-means. This analysis categorized rotation matrices (converted to quaternions) into 16 distinct clusters for subsequent sampling during text image synthesis. Fig. 3(b)(c)(d) show the visualization of analysis results. The final synthetic dataset comprised 12,520 images, with annotations including bounding boxes, transcriptions, and 3D pose attributes for text regions.

4.2. Scene Text Detection

Comparison with state-of-the-art methods. To assess the quality of our dataset against current state-of-the-art counterparts [14, 16, 35], we employed EAST[39] and DB[13] as baseline text detectors by directly using their official source codes. To ensure an equitable experimental setup, the total number of images in each synthetic dataset was controlled to 10,000. All models featured a ResNet-50 backbone and were trained on 8 NVIDIA GeForce RTX 3080 GPUs with a batch size of 56. Initially, models were exclusively trained on each synthetic dataset to establish their generalizability to real-world scenarios. Both EAST and DB detectors underwent a training phase of 1200 epochs on the four synthetic datasets, followed by evaluation on three real-world datasets. Subsequent fine-tuning on real-world data was conducted to scrutinize the impact of synthetic data as a pre-training resource. The results are succinctly summarized in Table 1. Across all benchmarks, it was observed that the overall performance of DB slightly surpassed that of EAST. When focusing on models trained solely on synthetic data, those trained on UT-10K exhibited relatively higher precision. However, detectors trained on our dataset consistently achieved superior recall and F-scores. Notably, when comparing the fine-tuning effects on real data following pre-training on synthetic datasets, the advantages of our dataset became even more pronounced. DB and EAST demonstrated remarkable F-score improvements of 22.72%, 5.26%, 2.52%, and 21.92%, 6.61%, 3.69% on the IC13, IC15, and MLT17 datasets over models trained exclusively on real data. We attribute this success partly to the high fidelity of synthetic data from real-world textual scenes. As for the superior recall rates, we believe it to be a consequence of incorporating prior text distribution knowledge from real data during synthesis. This underscores the strengths of our methodology in offering enhanced control over data generation.

Multi-view robustness evaluation. To conduct a more granular assessment, we exploited our synthesis method's provision of text pose attributes across diverse viewpoints to examine detector biases and evaluate robustness against text perspective effects. The synthesized scenes were partitioned in a 9:1 ratio for training and testing, preserving the quantity and pose distribution of text instances per scene. Sixteen images from each scene were chosen to ensure comprehensive coverage of all pose categories and to maximize the diversity of viewpoints. In this manner, a dataset of 4,320 training images and 480 test images was constructed. To validate our balanced viewpoint distribution strategy for setting camera poses during rendering, a control dataset was sampled from 270 training scenes. For this dataset, 16 images per scene were randomly selected without considering viewpoint diversity. The previously mentioned text detectors, configured identically, were trained on both two datasets and evaluated against the same test set. Additionally, we computed detector performance on a scene-by-scene basis by averaging the precision and recall rates for all viewpoint images within a scene, thereby providing a scene-level performance metric that reflects model robustness to varying perspectives within the same context. The results are compiled in Table 2. It is evident from the tabulated data that models trained on datasets utilizing a balanced viewpoint sampling strategy consistently outperform those trained on randomly sampled data. Moreover, on a scene-level basis, detectors trained with a balanced viewpoint approach exhibit a lesser degree of performance degradation compared to those trained on randomly sampled datasets, attesting to the robustness of our viewpoint distribution methodology.

Model	Train data	Instance level			Scene level		
		P	R	F	P	R	F
DB	Random	73.25	57.22	64.25	71.57	48.62	57.90
	Balance	**89.89**	**75.53**	**82.09**	**85.70**	**70.36**	**77.28**
EAST	Random	68.13	58.14	62.74	61.51	50.08	55.21
	Balance	**87.39**	**74.90**	**80.66**	**83.66**	**68.22**	**75.16**

Table 2. Detection performance on the random sampled dataset and view balanced dataset. **R**: Recall, **P**: Precision, **F**: F-score.

4.3. Ablation Study

Effect of structural importance mask on appearance changing. In order to examine the influence of structural information on appearance control in the style transfer process, we conducted a comparative analysis between two settings: one with the application of style transfer on synthesized images using structural importance masks, and the other without such masks. Fig. 4 visually demonstrates the differences in content authenticity and semantic completeness between the two settings. It is evident that without the inclusion of structural importance masks, the content of text regions undergoes uncontrollable modifications. This uncontrolled modification poses a significant challenge to the generation of high-quality synthetic text image data. Hence, the incorporation of structural importance masks is crucial in preserving the original content and semantic meaning in the style transfer process.

Effect of positional encoding on text region modeling. In this experiment, we investigated the impact of positional encoding on the rendering accuracy of text semantic labels. We compared the performance of two distinct text semantic renderers: the one incorporates positional encoding and the other one that does not. Fig. 5(a) compares the semantic loss of the two renderers on labeled and unlabeled images during training on a same scene. The results show a noticeable disparity, with the renderer without positional encoding registering higher losses on both labeled and unlabeled images. Fig. 5(b) visualizes the learning outcomes and further substantiates the comparison. This discrepancy underscores the insufficiency of relying solely on NeRF's geometry features to precisely delineate text regions within a scene. The integration of positional encoding injects additional high-frequency details that enhance the semantic renderer to accurately identify these highly customized regions.

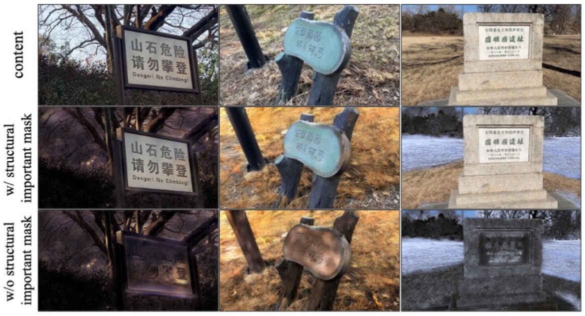
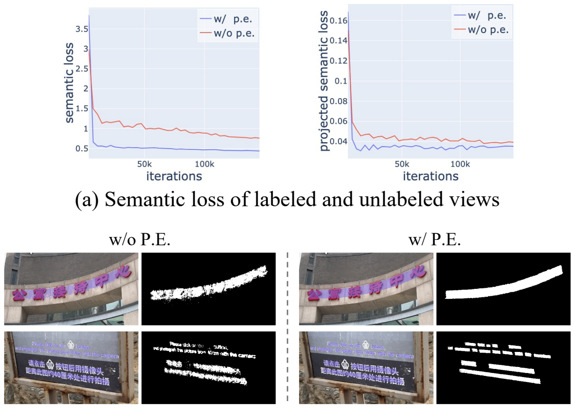


Figure 4. **The difference of the style transfer results with and without employing structural importance masks**.



(a) Semantic loss of labeled and unlabeled views

(b) Results of semantic renderer w/o and w/ positional encoding

Figure 5. **The training losses and rendering results of two different semantic renderers**.

5. Conclusion

This paper proposed a novel NeRF-based method for synthesizing scene text images with 3D controllability. Our approach leverages the geometric modeling capabilities of NeRF to semantically model text regions in 3D space and produces photo-realistic and diverse synthetic data. Both qualitative and quantitative experimental results demonstrated that our method significantly enhances the robustness and performance of scene text detectors. Moreover, the introduction of a dataset with 3D pose annotations is conducive to more in-depth evaluation of text detection models.

Limitations. While our approach has shown promising results, there still exists some limitations. The current method needs to train a specific NeRF for each scene, which limits scalability. There may be some color inconsistencies caused by style transfer, and it is difficult to capture text under extreme conditions, such as highly stylized fonts or severe occlusions. In future work, we aim to further optimize our synthesis pipeline to reduce computational demands and explore the extension of our method to other domains requiring fine-grained control over synthetic data generation.

References

[1] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. *arXiv: Computer Vision and Pattern Recognition*, 2021. 3

[2] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *arXiv: Computer Vision and Pattern Recognition*, 2021. 3, 6

[3] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*, pages 333–350. Springer, 2022. 3

[4] Abril Corona-Figueroa, Jonathan Frawley, Sam Bond-Taylor, Sarath Bethapudi, Hubert P. H. Shum, and Chris G. Willcocks. Mednerf: Medical neural radiance fields for reconstructing 3d-aware ct-projections from a single x-ray, 2022. 3

[5] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2315–2324, 2016. 2

[6] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv: Computer Vision and Pattern Recognition*, 2014. 2

[7] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. *International journal of computer vision*, 116:1–20, 2016. 2

[8] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere De Las Heras. Icdar 2013 robust reading competition. In *2013 12th international conference on document analysis and recognition*, pages 1484–1493. IEEE, 2013. 2, 6

[9] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *2015 13th international conference on document analysis and recognition (ICDAR)*, pages 1156–1160. IEEE, 2015. 2, 6

[10] Yijun Li, Ming-Yu Liu, Xueting Li, Ming-Hsuan Yang, and Jan Kautz. A closed-form solution to photorealistic image stylization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 453–468, 2018. 4

[11] Zhe Li, Zerong Zheng, Hongwen Zhang, Chaonan Ji, and Yebin Liu. Avatarcap: Animatable avatar conditioned monocular human volumetric capture. In *European Conference on Computer Vision (ECCV)*, 2022. 3

[12] Minghui Liao, Baoguang Shi, and Xiang Bai. Textboxes++: A single-shot oriented scene text detector. *IEEE transactions on image processing*, 2018. 2

[13] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. Real-time scene text detection with differentiable binarization. *Cornell University - arXiv*, 2019. 7

[14] Minghui Liao, Boyu Song, Shangbang Long, Minghang He, Cong Yao, and Xiang Bai. Synthtext3d: synthesizing scene text images from 3d virtual worlds. *Science China Information Sciences*, 63:1–14, 2020. 2, 6, 7

[15] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. Real-time scene text detection with differentiable binarization. In *Proceedings of the AAAI conference on artificial intelligence*, pages 11474–11481, 2020. 2

[16] Shangbang Long and Cong Yao. Unrealtext: Synthesizing realistic scene text images from the unreal world. *arXiv preprint arXiv:2003.10608*, 2020. 2, 6, 7

[17] Shangbang Long, Xin He, and Cong Yao. Scene text detection and recognition: The deep learning era. *International Journal of Computer Vision*, 2021. 2

[18] Roger Marí, Gabriele Facciolo, and Thibaud Ehret. Sat-NeRF: Learning multi-view satellite photogrammetry with transient objects and shadow modeling using RPC cameras. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1310–1320, 2022. 3

[19] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. 2020. 2, 3, 6

[20] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. 2023. 3

[21] Nibal Nayef, Fei Yin, Imen Bizid, Hyunsoo Choi, Yuan Feng, Dimosthenis Karatzas, Zhenbo Luo, Umapada Pal, Christophe Rigaud, Joseph Chazalon, et al. Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, pages 1454–1459. IEEE, 2017. 2, 6

[22] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 5

[23] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. *Computer Vision and Pattern Recognition*, 2016. 6

[24] Tao Sheng, Jie Chen, and Zhouhui Lian. Centripetaltext: An efficient text instance representation for scene text detection. *Advances in Neural Information Processing Systems*, 34:335–346, 2021. 2

[25] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[26] Jun Tang, Zhibo Yang, Yongpan Wang, Qi Zheng, Yongchao Xu, and Xiang Bai. Seglink++: Detecting dense and arbitrary-shaped scene text by instance-aware component grouping. *Pattern Recognition*, 2019. 2

[27] Zhuotao Tian, Michelle Shu, Pengyuan Lyu, Ruiyu Li, Chao Zhou, Xiaoyong Shen, and Jiaya Jia. Learning shape-aware

embedding for scene text detection. *Computer Vision and Pattern Recognition*, 2019. 2

[28] Wei-Cheng Tseng, Hung-Ju Liao, Yen-Chen Lin, and Min Sun. Cla-nerf: Category-level articulated neural radiance field. 3

[29] Suhani Vora, Noha Radwan, Klaus Greff, Henning Meyer, Kyle Genova, Mehdi S. M. Sajjadi, Etienne Pot, Andrea Tagliasacchi, and Daniel Duckworth. Nesf: Neural semantic fields for generalizable semantic segmentation of 3d scenes, 2021. 3

[30] Wenhai Wang, Enze Xie, Xiang Li, Hou Wenbo, Tong Lu, Gang Yu, and Shuai Shao. Shape robust text detection with progressive scale expansion network. *Cornell University - arXiv*, 2019. 2

[31] Chung-Yi Weng, Brian Curless, Pratul P. Srinivasan, Jonathan T. Barron, and Ira Kemelmacher-Shlizerman. HumanNeRF: Free-viewpoint rendering of moving people from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16210–16220, 2022. 3

[32] Zeke Xie, Xindi Yang, Yujie Yang, Qi Sun, Yixiang Jiang, Haoran Wang, Yunfeng Cai, and Mingming Sun. S3im: Stochastic structural similarity and its unreasonable effectiveness for neural fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18024–18034, 2023. 3

[33] Chuhui Xue, Shijian Lu, and Fangneng Zhan. Accurate scene text detection through border semantics awareness and bootstrapping. *Cornell University - arXiv*, 2018. 2

[34] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. 2023. 3

[35] Fangneng Zhan, Shijian Lu, and Chuhui Xue. Verisimilar image synthesis for accurate detection and recognition of texts in scenes. *arXiv: Computer Vision and Pattern Recognition*, 2018. 2, 6, 7

[36] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv: Computer Vision and Pattern Recognition*, 2020. 3, 6

[37] Zheng Zhang, Chengquan Zhang, Wei Shen, Cong Yao, Wenyu Liu, and Xiang Bai. Multi-oriented text detection with fully convolutional networks. *Cornell University - arXiv*, 2016. 2

[38] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J. Davison. In-place scene labelling and understanding with implicit scene representation. 2021. 4

[39] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, He Weiran, and Jiajun Liang. East: An efficient and accurate scene text detector. *arXiv: Computer Vision and Pattern Recognition*, 2017. 2, 7

[40] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3