BirdNeRF: Fast Neural Reconstruction of Large-Scale Scenes From Aerial Imagery

Huiqing Zhang, Yifei Xue, Ming Liao, Yizhen Lao*

***Abstract*—In this study, we introduce BirdNeRF, an adaptation of Neural Radiance Fields (NeRF) designed specifically for reconstructing large-scale scenes using aerial imagery. Unlike previous research focused on small-scale and object-centric NeRF reconstruction, our approach addresses multiple challenges, including (1) Addressing the issue of slow training and rendering associated with large models. (2) Meeting the computational demands necessitated by modeling a substantial number of images, requiring extensive resources such as high-performance GPUs. (3) Overcoming significant artifacts and low visual fidelity commonly observed in large-scale reconstruction tasks due to limited model capacity. Specifically, we present a novel bird-view pose-based spatial decomposition algorithm that decomposes a large aerial image set into multiple small sets with appropriately sized overlaps, allowing us to train individual NeRFs of sub-scene. This decomposition approach not only decouples rendering time from the scene size but also enables rendering to scale seamlessly to arbitrarily large environments. Moreover, it allows for per-block updates of the environment, enhancing the flexibility and adaptability of the reconstruction process. Additionally, we propose a projection-guided novel view re-rendering strategy, which aids in effectively utilizing the independently trained sub-scenes to generate superior rendering results. We evaluate our approach on existing datasets as well as against our own drone footage, improving reconstruction speed by 10x over classical photogrammetry software and 50x over state-of-the-art large-scale NeRF solution, on a single GPU with similar rendering quality.**

***Index Terms*—NeRF, large-scale reconstruction, aerial image, spatial decomposition, projection-guided.**

Fig. 1. Illustration of modular scene training, along with performance and time comparisons on the IZAA dataset (comprising 1469 images). We demonstrate an approximately 10x speed improvement over traditional Metashape software. Moreover, when compared to current large-scale reconstruction approaches using deep learning, our method exhibits an approximately 56x enhancement in speed.

I. Introduction

LARGE-SCALE 3D reconstruction on a city-wide level is an intrinsically active and significant task in areas of photogrammetry and remote sensing. This process revolves around constructing detailed and precise 3D models of entire cities utilizing an array of data sources, including, but not limited to, aerial or satellite images, LiDAR data, and street-level imagery. The expeditious advancements in aerial surveying technology have simplified and made cost-effective the procurement of high-resolution images. Consequently, image-based 3D reconstruction has emerged as an affluent and promising domain of study, encompassing manifold applications.

3D urban models find extensive application across various fields, providing a significant impetus to them. For example, urban development reaps the benefits of these models as they facilitate simulations and visual demonstrations of various scenarios, such as the erection of new edifices, the implications of transportation projects, and the layout of public venues, thereby aiding informed decision-making[1]. In the sphere of navigation, 3D urban reconstructions contribute to the creation of accurate and exhaustive maps to augment the precision of GPS devices and mobile applications[2]. Moreover, this technology forms the bedrock of augmented reality applications providing real-time overlaid directions on real world views. Virtual tourism thrives with extensive 3D reconstructions, as these enable individuals to virtually explore cities before actually visiting them and provide access to distant or otherwise difficult-to-reach areas[3]. For the real estate industry, such reconstructions afford potential buyers a lucid and intuitive understanding of a property's environs, facilitating property valuation[4]. 3D reconstructions hasten rescue operations, damage assessments, and post-disaster reconstruction planning during disaster management[5]. Lastly, the domain of historical preservation considerably benefits from 3D reconstructions by aiding research, cultural heritage preservation, and the creation of virtual reconstructions for lost or damaged historical sites[6]. These varied applications underscore the importance of 3D city modeling across diverse sectors.

Existing image-based 3D reconstruction techniques form two broad categories: traditional geometry-based methods and neural network-based methods. Geometry-based methods

entail a two-step process composed primarily of Structure-from-Motion (SfM) and Multiple Views Stereo (MVS) [7]. SfM estimates camera poses and sparse 3D points from input images [8], while MVS refines point clouds and builds a dense 3D model [9]. Contrarily, neural network-based methods, epitomized by Neural Radiance Fields (NeRF) [10], signify a revolutionary progression in 3D reconstruction. NeRF employs neural networks to implicitly represent[11, 12] three-dimensional scene data by training network parameters grounded on input images and corresponding camera poses. It also depicts the capability to generate novel viewpoint images. More recent developments in NeRF research, such as Instant Neural Graphics Primitives (Instant-NGP) [13], accentuate the rapidly evolving terrain of this field. Nonetheless, large-scale urban reconstruction methods currently in use grapple with three main challenges:

1) **Slow Rendering with Large Models:** Large-scale reconstruction processes are inherently time-consuming. As the demand for real-time or near real-time applications, such as navigation and disaster management, intensifies, research is needed into faster and more efficient large-scale 3D reconstruction techniques.
2) **Computational Demands:** Large-scale reconstruction involves handling and processing vast datasets, often exceeding the memory capacity of a single GPU. This can result in slow processing times, out-of-memory errors, and other performance-related concerns, posing challenges to users or researchers with limited memory resources.
3) **Artifacts and Low Visual Fidelity:** Traditional geometry-based 3D reconstruction methodologies often grapple with challenges related to inaccurate camera pose estimation and limitations in model capacity. These issues manifest as artifacts and gaps in the reconstruction, leading to suboptimal visual fidelity.

Propelled by the intricate challenges inherent in large-scale scene reconstruction using aerial imagery, especially employing NeRF, we present BirdNeRF in this study, a dedicated adaptation of NeRF designed for reconstructions of large-scale aerial scenes. The proposed method of BirdNeRF chiefly incorporates spatial decomposition of camera distribution, followed by the modular training of smaller scenes (Fig. 1), and finally generates images from novel viewpoints through our unique projection-guided view re-rendering strategy. BirdNeRF manages not only to enable reconstruction based on extensive aerial survey image inputs but also to guarantee excellent standards of reconstruction quality and speed of modeling. Fig. 1 demonstrates a comparative analysis of the quality of reconstruction between our proposed BirdNeRF and several other large-scale reconstruction methods, along with a comparison of training durations based on a dataset comprising 1469 images. The results depict significant improvements over previous solutions, proof of the efficacy of BirdNeRF in effectively addressing the challenges identified.

A. Related Work and Motivations

Structure-from-Motion and Multi-View-Stereo. Structure-from-Motion (SfM)[14] and Multi-View Stereo (MVS)[9] combine to form a powerful pipeline for 3D reconstruction[7]. SfM aims to recover camera poses and a sparse 3D structure of the scene. Conversely, MVS focuses on creating a dense 3D representation by estimating the depth or disparity of each pixel in the images, culminating in a dense point cloud. Often, MVS goes beyond mere dense reconstruction and incorporates surface reconstruction methods[15, 16], resulting in either a mesh or continuous surface representation derived from the dense point cloud.

Prominent SfM and MVS techniques such as VisualSfM[17], COLMAP[8], and OpenMVG[18] have made significant strides. However, these approaches can be hampered by scalability issues, slow processing speeds, and visual shortcomings, including holes, texture blending, and distortion[19]. Insufficient information and inadequate image coverage in certain regions can lead to the problem of holes. Errors in camera calibration, image noise, and inaccurate feature matching can result in texture blending, visual distortions, and so on. Various post-processing techniques, such as hole filling, texture blending corrections and mesh refinement, are required to mitigate these challenges[20]. Consequently, there's a concerted effort in ongoing research to enhance the accuracy, efficiency, and scalability of SfM and MVS algorithms, particularly for substantial reconstruction tasks.

Neural Radiance Fields. Contrastingly, NeRF [10] employs a deep neural network to model the volumetric scene as a continuous function, bypassing the need for explicit geometrical or point-based representation. NeRF learns from a scene to predict sites appearance and density at any given 3D point, producing high-quality 3D reconstructions and novel views albeit with high computational demands.

Several extensions of naive NeRF have been developed to enhance this method. For instance, Instant-NGP [13] uses a hash encoding to accelerate the process remarkably, rendering it the fastest NeRF method currently. Similarly, NeRF++ [21] and Mip-NeRF 360 [22] have been fashioned specifically for unbounded scenes. DrRF [23] partitions the scene using spatial Voronoi and renders each image part independently, accelerating rendering by three times compared to NeRF [5]. KiloNeRF [24] partitions the scene and assigns it to thousands of small networks for collective training, which speeds up the inference process to a certain extent. However, DeRF and KiloNeRF need an extra expensive initialization [5]. The PixelNeRF [25] optimizes the NeRF model training by leveraging prior information from image features, facilitating rapid model reconstruction from sparse inputs. Nevertheless, none of these methods is suited for city-level reconstruction.

Commercial Reconstruction Software. Pix4D Mapper [26] and Agisoft Metashape [27] are both well-known and widely used commercial software packages providing comprehensive solutions for photogrammetry and 3D reconstruction. They offer robust and reliable ways to process aerial and terrestrial images, generating accurate and detailed 3D models, point clouds, orthomosaics, and digital surface models. Benefiting from well-established algorithms for camera calibration, feature matching, dense point cloud generation, and mesh

reconstruction, they can deliver an exceptional reconstruction.

Despite their capabilities, these software tools have inherent limitations and challenges. Photogrammetry and 3D reconstruction processes are computationally intensive, especially when the task involves large datasets or highly complex scenes. As such, Pix4D Mapper and Agisoft Metashape both demand a significant amount of computational resources, including processing power, memory, and ample storage. Users attempt to run this software on lower-end machines or systems with limited resources may encounter sluggish processing times or reduced performance.

Large Scale Reconstruction. Over the past few decades, wide-ranging efforts have been made towards achieving large-scale reconstruction. Studies such as [28, 29] focus on employing parallelism in the reconstruction process. A substantial breakthrough in large-scale reconstruction tasks has been accomplished by [30–32] through the application of SfM and MVS methods. For instance, [30] introduces a divide-and-conquer framework for handling large scale global SfM, while [31] proposes a distributed method to address global bundle adjustment for colossal scale SfM computations. On the other hand, [32] employs surface-segmentation-based camera clustering to achieve the decomposition of large-scale MVS. These novel approaches provid significant inspiration for our work.

Several examples of large-scale reconstruction works, such as [5, 19, 33], are based on NeRF. Despite Mega-NeRF [5] achieving large-scale reconstruction on a single GPU, it suffers from extremely long training times due to its dependency on the original NeRF implementation. Block-NeRF [19] primarily uses images captured by vehicle-mounted cameras and decomposes the scene into distinct spatial units, each corresponding to a fixed city block. However, this method demands significant training resources. Furthermore, BungeeNeRF [33] models diverse multi-scale scenes using multiple data sources. Although these methods have achieved their goal of large-scale reconstruction, they have done so at the cost of extensive training resources, while also not effectively addressing the challenge of limited GPU memory in practical applications.

Motivations. From the above discussion, it becomes evident that a fast, high-quality, large-scale 3D reconstruction methodology that operates within limited resource constraints is yet to be developed. The prevailing methods, especially those grounded in NeRF for 3D modelling, impose considerable demands on the training resources. This high computational burden, along with elongated training durations, often results in constrained model capacity, leading to visual distortions such as artifacts, voids, and blurring. A quick and high-quality modelling process is imperative, especially in vital fields such as urban planning and disaster response, making exploration of expedited, high-quality, large-scale 3D reconstruction methodologies that work within limited memory constraints both pertinent and necessary. These efforts hold significant potential for catering to the growing and changing needs of such crucial applications.

B. Contributions

In response to the substantial challenge of enabling swift, high-quality, large-scale 3D reconstruction within the bounds of limited memory resources, we present BirdNeRF. The process begins with a spatial decomposition, using the spatial distribution of cameras to discern separate clusters, thereby segmenting the training scenes. Each resulting sub-scene is then trained independently. In the final stage, our unique projection-guided novel view re-rendering strategy is utilized to register and align query cameras, facilitating the rendering of the requested query viewpoints. This methodical approach, integrating spatial decomposition, independent training, and a customized re-rendering strategy, optimizes large-scale 3D reconstruction in resource-limited settings. We conducted extensive experiments to evaluate our approach, providing both qualitative and quantitative insights into its effectiveness. The outcomes highlight its distinct advantages in terms of both modeling time and the quality of the rendered images. The primary contributions of this study can be summarized as follows:

1) **Unprecedented speed in large-scale reconstruction:** We introduce a pipeline for large-scale reconstruction that is unparalleled in its speed. Our method is capable of reconstructing scenes of up to 1 square kilometer in approximately half an hour, eclipsing the speeds of commercial software like Metashape by a factor of ten or more, and outperforming current deep learning approaches by more than fifty times. This speed advantage only increases for larger datasets.
2) **Adaptability to GPU memory constraints:** Our methodology is marked by its adaptability to a variety of GPU memory resources. Through the use of a spatial decomposition strategy based on camera distribution, we are able to manage large-scale reconstructions effectively within the constraints of limited GPU memory. This adaptability makes our method versatile and scalable, demonstrating its applicability across a range of hardware setups.
3) **Innovative re-rendering strategy for high-quality results:** We introduce an innovative projection-guided novel view re-rendering strategy that ensures precise registration and query of cameras during the rendering process. This strategy meticulously brings the relevant sub-models into play for rendering output, guaranteeing an accurate and efficient combination of rendered images from various viewpoints across scenes of all sizes.

II. METHODOLOGY

Representing and reconstructing large-scale scenes, such as those present in aerial imagery, poses considerable challenges due to the inherent scalability limitations of training a single NeRF. In addressing these challenges, we propose BirdNeRF, a method characterized by decomposing the environment into a series of NeRFs, each trained individually based on the bird view field of view (FOV). During the inference phase, the novel view is rendered by aggregating the outputs from these disparate NeRFs. This strategy, which we term the ”split-unite

paradigm", successfully circumvents the limitations of model capacity that have stymied previous NeRF research, enabling efficient reconstruction of expansive scenes even with limited computational resources.

As depicted in Fig. 2, BirdNeRF encompasses two principal phases: (1) spatial decomposition, which involves dividing the scene into manageable, cluster-based segments, and (2) projection-guided novel view re-rendering, which reunites the independently processed segments to render the desired viewpoint.

A. Background

BirdNeRF is fundamentally grounded in the original NeRF [10] methodology and its subsequent advancement, Instant-NGP [13]. Herein, we provide a synopsis of these foundational methods, while detailed expositions can be found in the respective source papers.

• **NeRF overview.** NeRF is a transformative model that introduces a fresh paradigm to scene representation and view synthesis, enabling the creation of photorealistic 3D scenes from a set of 2D images. NeRF operates by associating each pixel in an image with a corresponding ray in 3D space, estimating color and density along these rays, and adjusting network parameters to minimize the difference between observed and predicted colors. Once trained, NeRF can be leveraged to synthesize novel views by projecting rays from new camera positions and integrating the color and density estimates to produce lifelike images from viewpoints not captured in the original dataset.

To elucidate, let's consider an image pixel at the camera center with specific pixel coordinates. For this pixel, NeRF constructs a ray $\boldsymbol{r}(t) = \boldsymbol{o} + t\boldsymbol{d}$ in 3D space. The points along this ray are represented by their position $\boldsymbol{x}^i = [x, y, z]$ and direction $\boldsymbol{d}^i = [d1, d2, d3]$. Positional encoding (Eq. (1)) is then applied to these coordinates, enabling NeRF to capture higher frequency details in the scene. These encoded coordinates are then fed to the NeRF model, a Multilayer Perceptron (MLP), which outputs the color $c^i = [r, g, b]$ and density σ^i for each point.

$$\begin{aligned}\gamma(p) = (&\sin(2^0\pi p), \cos(2^0\pi p), ..., \\ &\sin(2^{L-1}\pi p), \cos(2^{L-1}\pi p))\end{aligned} \tag{1}$$

where L is the number of levels of positional encoding, NeRF sets L = 10 for $\gamma(\boldsymbol{x}^i)$ and L = 4 for $\gamma(\boldsymbol{d}^i)$.

The volume rendering process (Eq. (2)) integrates these color and density estimates to calculate the final pixel color $\hat{C}(\boldsymbol{r})$ for the ray $\boldsymbol{r}$.

$$\begin{aligned}\hat{C}(\boldsymbol{r}) &= \sum_{i=1}^{N} T_i(1 - exp(-\sigma_i\delta_i))c_i, \\ where\ T_i &= exp(-\sum_{j=1}^{i-1}\sigma_j\delta_j)\end{aligned} \tag{2}$$

where δ_i is the distance between samples p_i and p_{i+1}.

The loss function L (Eq. (3)) is computed by comparing the rendered color value to the original image color value to supervise the model training. By training the model, an implicit representation of the 3D scene is obtained, allowing for rendering arbitrary camera views from different viewpoints.

$$L = \sum_{r\in\Re} \left\| C(\boldsymbol{r}) - \hat{C}(\boldsymbol{r}) \right\|^2 \tag{3}$$

where $\Re$ is the set of rays in each batch, $C(\boldsymbol{r})$ is the ground truth and $\hat{C}(\boldsymbol{r})$ is the predicted RGB colors for ray $\boldsymbol{r}$ respectively.

• **Instant-NGP.** Instant-NGP [13] is a high-performance neural radiance field method developed by NVIDIA. It distinguishes itself by implementing a multi-resolution hash encoding approach, which accelerates the training process. The method also incorporates an enhanced multi-resolution hash table, designed to accommodate trainable feature vectors and reduce the overall model size. Instant-NGP is a fully integrated system, offering a seamless and efficient deployment for rapid training and real-time rendering of detailed scenes.

B. Initialization

Our methodology begins with the utilization of the classical Multi-View Stereo (MVS) software COLMAP [8] on a dataset of aerial images. This step involves computing camera poses and generating a sparse point cloud for the scene. The camera model is set as a pinhole camera model, using uniform internal parameters for feature extraction and matching. An incremental reconstruction approach combined with camera pose estimation is applied to obtain sparse point clouds representing the scene and pose information for all cameras in the dataset.

C. Spatial decomposition

• **Spatial partitioning.** Once the camera poses and sparse point clouds are obtained from the initialization phase in Section II-B, an initial partitioning step is performed based on the spatial coordinates of the cameras. This is done using the K-Means clustering algorithm [34]. The optimal number of clusters, denoted as K, is determined based on the available GPU memory size. Each cluster represents a sub-scene in the dataset.

• **Sub-scenes extension.** We extend the sub-scenes to ensure a defined degree of overlap, enhancing the efficacy of image registration, as depicted in Fig. 3. Concretely, we introduce an expanded threshold denoted as σ for the number of cameras, guiding the augmentation of each scene to guarantee a predetermined level of overlap within the partitioned sub-scenes. The computation of intra-cluster distances, represented by d_k for each cluster, facilitates the identification of the maximum intra-cluster distance. This maximum distance is then multiplied by the designated threshold to determine the new cluster diameter. Subsequently, utilizing the preceding cluster centroids as centers, we search for cameras within the range of the recalibrated diameter and incorporate them into the updated clusters. It is noteworthy

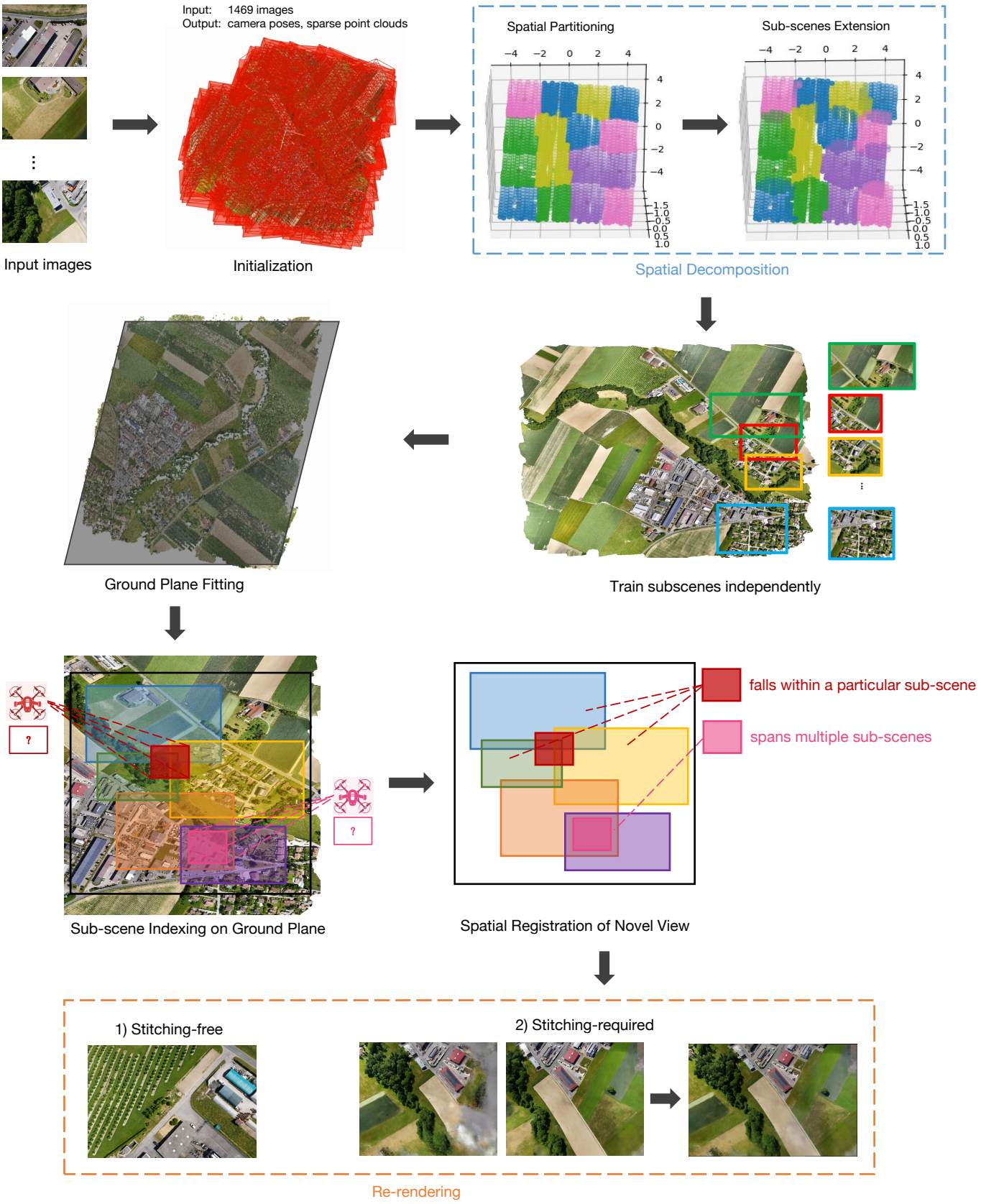

Fig. 2. The BirdNeRF pipeline is initiated by the preprocessing phase II-B, where input images are processed to obtain camera positions and coefficient point clouds. Spatial decomposition follows in section II-C, categorizing cameras into clusters. For each cluster, associated images facilitate independent training mentioned in section II-D, creating multiple sub-scenes. The novel projection-guided view re-rendering strategy described in section II-E synthesizes the final rendering images.

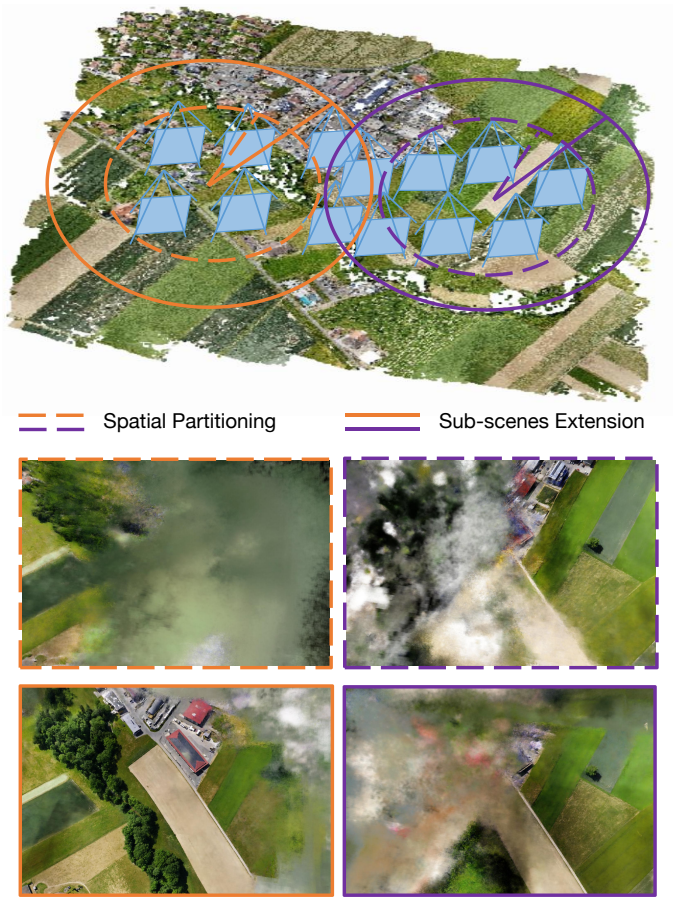

Fig. 3. Sub-scenes extension. The strategic expansion of sub-scenes enhances scene overlap, thereby elevating the success rate of post-image registration in our proposed approach.

that a maximum limit on the number of images per sub-scene is imposed to ensure seamless training within the allocated GPU memory resources. This stage signifies the completion of camera division, denoting the subdivision of the scene into distinct sub-scenes. A comprehensive algorithmic depiction is presented in Alg. 1.

D. Individual training

After the spatial decomposition process, we obtain separate camera clusters where the camera parameters and corresponding images within each cluster form the training data for the respective sub-scenes. Independent training is performed for each sub-scene using Instant-NGP as the base training model. Once the training is completed, the resulting model parameters are stored offline on disk for future use in the re-rendering process. It is important to note that the models are stored as network parameters, which occupies much less disk space compared to alternative representations such as point clouds or grids. This efficient storage of the model parameters allows for easier access and retrieval when needed.

Algorithm 1 Spatial decomposition algorithm.

Input:
Maximum number of cameras per subscene, **maxN**.
Total number of cameras in a dataset, **N**.
The expanded threshold, $\boldsymbol{\sigma}$.
Process:
K ← N / maxN
Labels, Centers ← KMeans(K)
for $k \in \{1, ..., K\}$ **do**
  for $i \in \{1, ..., maxN\}$ **do**
   $d_k \leftarrow$ MAX (EuclideanDistances($T_i, Center_k$))
  endfor
endfor
for $k \in \{1, ..., K\}$ **do**
$d'_k \leftarrow \sigma * d_k$
endfor
for $k \in \{1, ..., K\}$ **do**
  for $i \in \{1, ..., N\}$ **do**
  if EuclideanDistances($T_i, Center_k$))$< d'_k$:
    $Scene_k \leftarrow T_i$
  endfor
endfor
Output:
The subscene partition obtained through expanded clustering, $\boldsymbol{Scene_k(k = 1, ..., K)}$.

E. Projection-guided novel view re-rendering

In order to effectively query target scenes, we employ a set of methods to optimize the novel view fusion stage, as shown in Fig. 4.

• **Ground plane fitting.** Prior to the subsequent procedure, the ground plane parameters are initially determined using the Least Squares [35] approach.

• **Sub-scene indexing on ground plane.** We employ a methodology involving the projection of the four corner points of an image onto the ground plane, a plane fitted through the sparse point cloud. This technique facilitates the determination of the scene extent corresponding to the image captured by the camera.

In this study, the impact of camera type on our results is negligible. Consequently, we default to assuming our method is based on the pinhole camera model. For a pinhole camera, the coordinates of the four corner points of the $i-th$ image in the camera coordinate system are denoted as

$$\begin{aligned} p_1^i &= [0 - cx, 0 - cy, f_i] \\ p_2^i &= [w - cx, 0 - cy, f_i] \\ p_3^i &= [w - cx, h - cy, f_i] \\ p_4^i &= [0 - cx, h - cy, f_i] \end{aligned} \tag{4}$$

The spatial coordinates of the optical center associated with the $i-th$ camera, corresponding to a given image, are denoted as

$$o^i = [0, 0, 0] \tag{5}$$

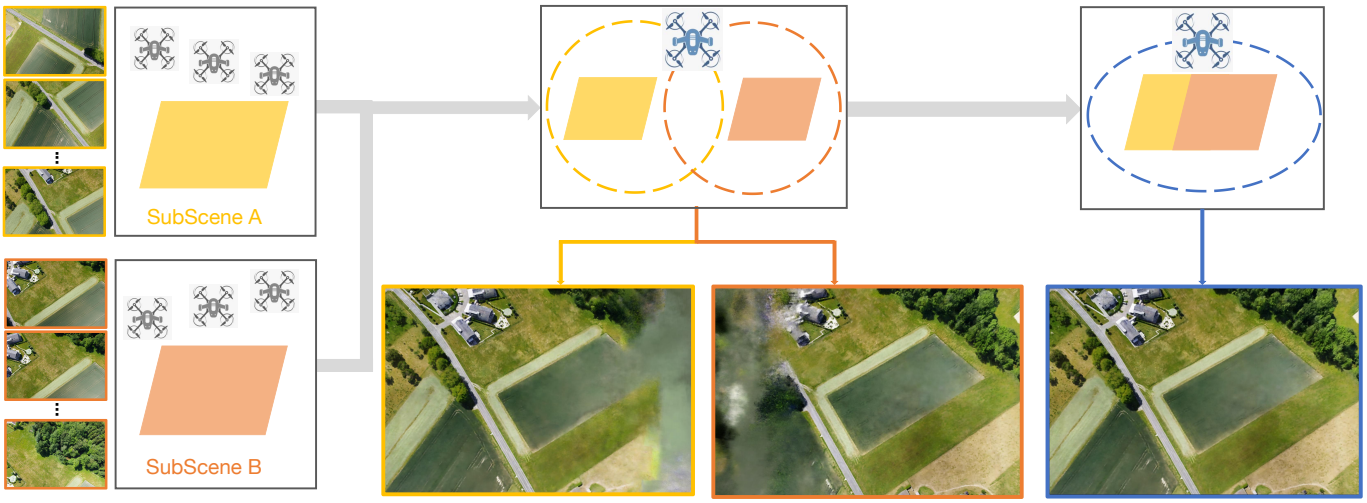

Fig. 4. Projection-guided novel view re-rendering. Beginning with independently constructed input NeRFs, namely Sub-scene A and Sub-scene B, we perform image rendering from novel viewpoints. Then, employing a sequence of image stitching and fusion techniques, we achieve higher-quality re-rendering results.

Utilizing the pinhole camera model, we transform the coordinates of the four corner points and optical center from the camera coordinate system to the world coordinate system, as indicated by Eq. (6).

$$P_k^i = R^i \cdot p_k^i + T^i \tag{6}$$

Here, P_k^i signifies the coordinates of the $k-th$ corner point in the world coordinate system of the $i-th$ camera, while p_k^i denotes the coordinates of the same point in the camera coordinate system of the $i-th$ camera. The parameters R^i and T^i characterize the camera pose with respect to the world coordinate system.

The conversion of the coordinates for the camera's optical center is likewise conducted using the aforementioned equation, as indicated by Eq. (7).

$$O^i = R^i \cdot o^i + T^i \tag{7}$$

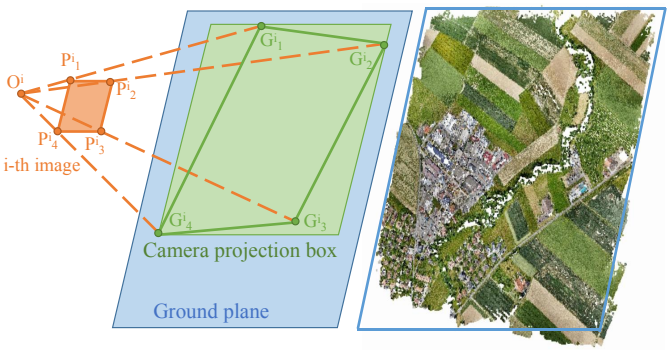

Fig. 5. Ground plane fitting and pixel projection.

Following this, the intersection of line segments, formed by connecting the optical center and the corner points, with the ground plane is computed to derive the intersection points, as depicted in Fig. 5. The minimum rectangle that encompasses all such points is determined upon obtaining these intersection points. This rectangle effectively represents the extent of the scene captured by the $i-th$ camera, known as the camera projection box.

As detailed in Section II-C, we employ a spatial decomposition methodology to allocate each camera to multiple sub-scenes. Each sub-scene comprises several cameras, with its bounding box defined as the amalgamation of the individual bounding boxes of all cameras assigned to that specific sub-scene. Furthermore, a minimum bounding rectangle is computed to encapsulate the collective scene range captured by all cameras within the sub-scene. This resulting bounding rectangle serves as the scene bounding box for the sub-scene, as visually represented in Fig. 6.

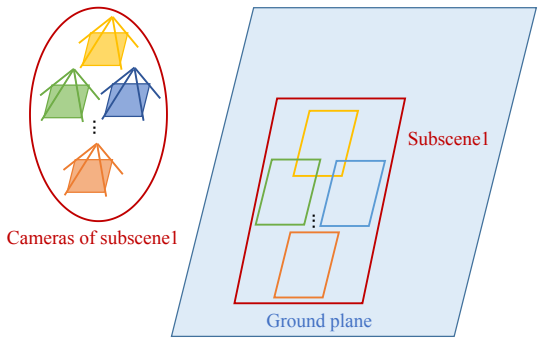

Fig. 6. Sub-scene bounding Box. The projection boxes of all cameras within each divided sub-scene collectively form the projection box of the sub-scene range.

• **Spatial registration of novel view.** As per the methodology employed in the preceding section, we compute the projection box of the queried camera onto the ground plane. Subsequently, we systematically iterate through all the sub-scene boxes to identify those that intersect with the projection box of the queried camera. These identified sub-scene bounding boxes are preserved as rendering schemes, guiding the subsequent rendering process for the queried

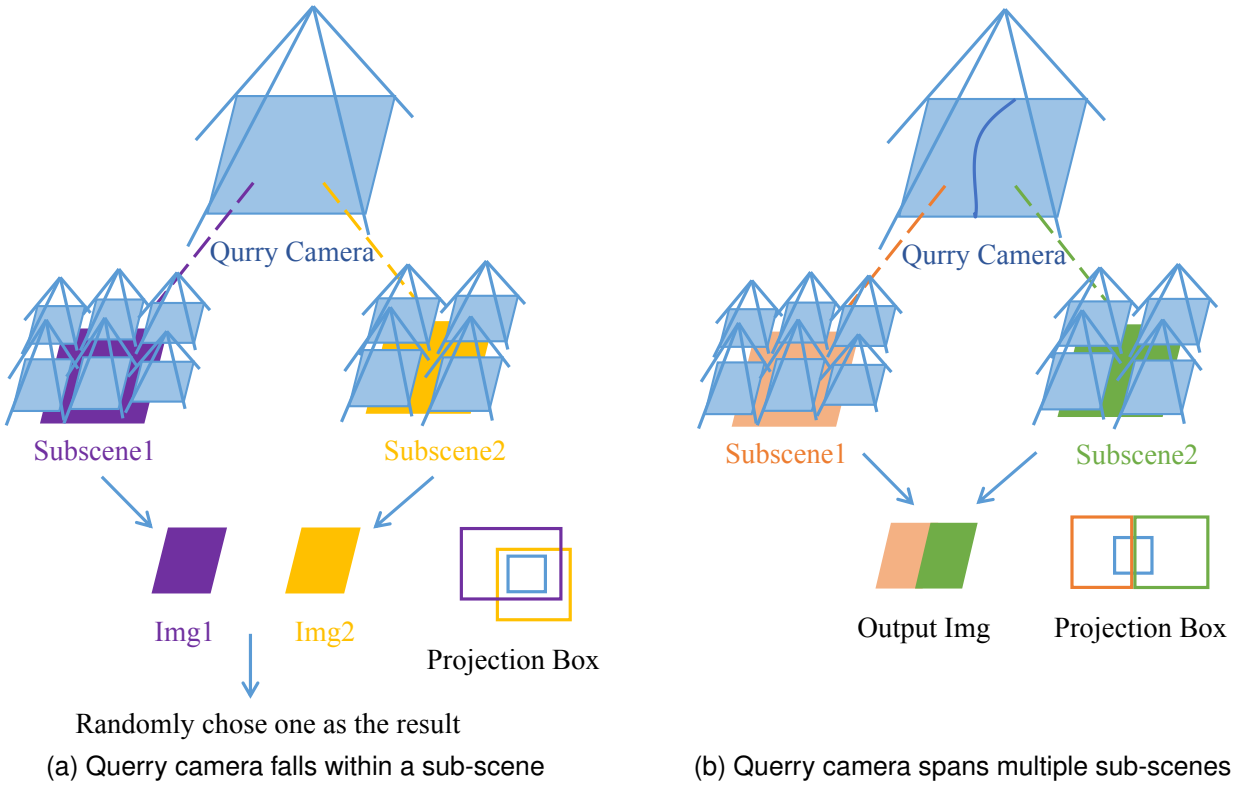

Fig. 7. Two types of query camera position distributions. When the query camera projection box is entirely contained within the bounding box of a specific sub-scene, the rendered image is directly output from that sub-scene. In cases where the query camera is positioned at the boundary of multiple sub-scenes, registration, and fusion of outputs from these sub-scenes are necessary to obtain the final result.

image. This rendering process is executed by leveraging the trained models associated with each respective sub-scene.

• **Re-rendering.** Once we obtain the bounding boxes for each independently trained sub-scene, we can automatically generate a rendering strategy for the query camera that needs to be rendered. Generally speaking, there are two situations when querying scenes (Fig. 7):

1) Stitching-free: In situations where the retrieved scene lies entirely within a specific sub-scene, rendering can be performed using the network model of that sub-scene alone. Fig. 7a illustrates an example where the identified scene matches the requirements for stitching-free rendering. In this case, the intersection of projection boxes is applied to the sub-scenes that encompass the queried camera's placement. Since the camera resides within a single sub-scene, a complete rendering can be achieved by using that sub-scene alone. Therefore, the resulting image can be generated from any of these relevant sub-scenes.

2) Stitching-required: In this case, the retrieved scene range does not fall entirely within a certain sub-scene range. We need to render multiple sub-scene network models containing the retrieved scene in order to render and stitch together an image containing the entire target scene range. As illustrated in Fig. 7b, when the queried camera is situated at the boundaries of multiple scenes, the resulting rendered image requires collaborative composition from various sub-scenes, where each scene contributes to rendering a distinct portion

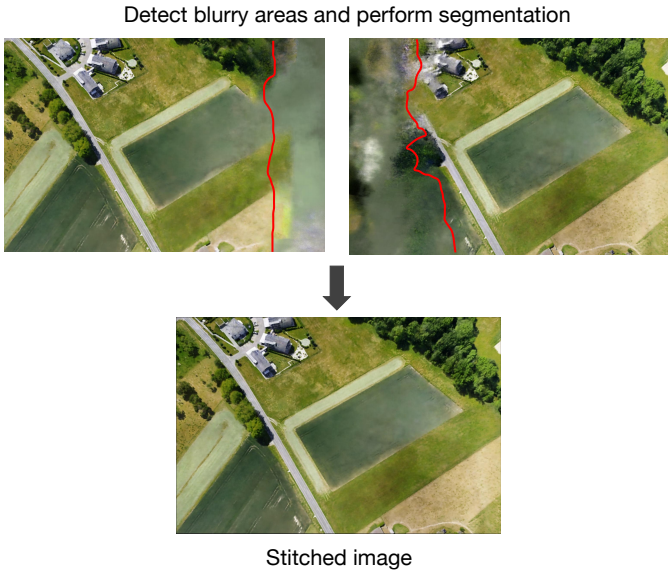

Fig. 8. Blur area detection and image stitching.

of the image. Subsequent to invoking these sub-models for rendering and generating their respective images, we undertake the detection and segmentation of the blurred areas within each rendered image[39], as delineated in Fig. 8. Following the removal of blurred portions, we employ an image stitching algorithm[36, 37], which integrates multiple panoramas, incor-

porates gain compensation, employs simple blending, and implements multi-band blending. This comprehensive approach amalgamates the partial images into a coherent new rendering result.

III. Experiments

A. Implementation

We run our method on a 12th Gen Intel(R) Core(TM) i9-12900KF with 32GB RAM. All the experiments in this paper are conducted on a single NVIDIA GeForce RTX 3090 GPU(24GB). We set the default number of cameras in a sub-scene to 90 in order to determine the initial clustering clusters (K value). Additionally, we set the maximum number of cameras in each partitioned sub-scene to 115 to ensure that our method runs without encountering GPU memory overflow. This value can be adjusted according to the GPU specifications of the experimental environment. We set the expanded threshold σ to 1.1 when scaling the maximum intra-cluster distance for camera cluster augmentation. For the model training for each sub-scene, we set the number of training iterations to 5×10^3, while keeping the other parameters at their default values.

B. Datasets

Our experiments utilize aerial datasets from 8 distinct geographical regions, each varying in size and characteristics. The datasets are selected from both publicly available sources and those captured using the DJI Mavic Air 2 drone in Hexi University Town, Changsha. These datasets collectively cover urban, suburban, industrial, agricultural, and university campus environments, providing diverse scenes for comprehensive evaluation.

We select three real photogrammetry datasets from Pix4D's example projects[38]:

- **Urban area(UA).** UA represents a city dataset covering 0.0214 km^2, featuring 100 images with a resolution of 6000 x 4000 pixels.
- **Suburban area(SA).** SA) comprises 188 images at a resolution of 5472 x 3648 pixels, covering an area of 0.041 km^2.
- **Industrial zone and agriculture area(IZAA).** IZAA dataset consist of 1469 images at a resolution of 6000 x 4000 pixels, encompassing an extensive area of 1.154 km^2. This dataset includes diverse regions such as an industrial zone, suburban residential areas, and surrounding agricultural landscapes.

Utilizing the DJI Mavic Air 2 drone, we conduct aerial surveys capturing five distinct regions within Hexi University Town, situated in Changsha, Hunan Province. These surveys result in the creation five datasets, varying in size from hundreds to thousands of images. Notably, the flight trajectories of our drone missions introduce additional complexities compared to the publicly available datasets previously mentioned.

- **CSU1.** The CSU1 dataset contains 408 images with a resolution of 4000 x 3000 pixels, covering an area of approximately 0.2 km^2, including the sports stadium and gymnasium areas of the new campus at Central South University.
- **CSU2.** The CSU2 dataset comprises 713 images with a resolution of 4000 x 3000 pixels, covering approximately 1/3 of the southwest area of the new campus at Central South University. The total area covered is approximately 0.2 km^2.
- **HNU.** The HNU dataset consists of 391 images with a resolution of 4000 x 3000 pixels, covering an area of approximately 0.1 km^2 around the Tianma Student Dormitory at Hunan University.
- **CSUS.** The CSUS dataset consists of 777 captured photos with a resolution of 4000 x 3000 pixels, covering an on-site area of approximately 0.7 km^2 in the Southern Campus of Central South University.
- **CSUHU.** The CSUHU dataset comprises 1706 photos covering an approximate area of 1 km^2. The main shooting locations include parts of the New Campus of Central South University and portions of the Houhu International Art Park buildings. The images in the dataset have a resolution of 4000 x 3000 pixels.

C. Results

1) Comparison methods: In our experiments, the proposed method BirdNeRF is compared to four large-scale reconstruction solutions that can model with a single GPU:

- Metashape[27]: Agisoft Metashape is a commercial software designed to process digital images using photogrammetry methods. We conducted our control experiments using Agisoft Metashape Pro 1.6.5.
- Mega-NeRF[5]: Mega-NeRF is a scene segmentation method, but it performs segmentation at the pixel level.
- Instant-NGP[13]: Instant-NGP is currently the fastest neural radiance field training method, and it serves as a benchmark for our approach BirdNeRF.

2) Evaluation metrics: We quantify the performance of our method using two established quantitative metrics: peak signal-to-noise ratio (PSNR) and structural similarity (SSIM). These metrics accurately assess the quality and similarity between the rendered images and their corresponding ground truth images. Additionally, we assess the efficiency of different methods by comparing the training time required after aligning the cameras in a consistent training environment. This dual evaluation framework provides a comprehensive analysis, addressing our proposed method's quality and efficiency dimensions.

3) Qualitative results: In Fig. 9, we exhibit the post-modeling rendering outputs of the assessed methods. It is evident that Metashape's rendering output often contains undesirable holes and can lead to visually unrealistic objects. Mega-NeRF, on the other hand, exhibits subpar modeling results overall. In contrast, our proposed method demonstrates comparatively superior visual output results. Furthermore, we present additional results featuring randomly generated camera poses from our method, which may require stitching, as illustrated in Fig. 10. This additional visualization provides

Fig. 9. Rendering outputs of Metashape, Mega-NeRF, and our method BirdNeRF.

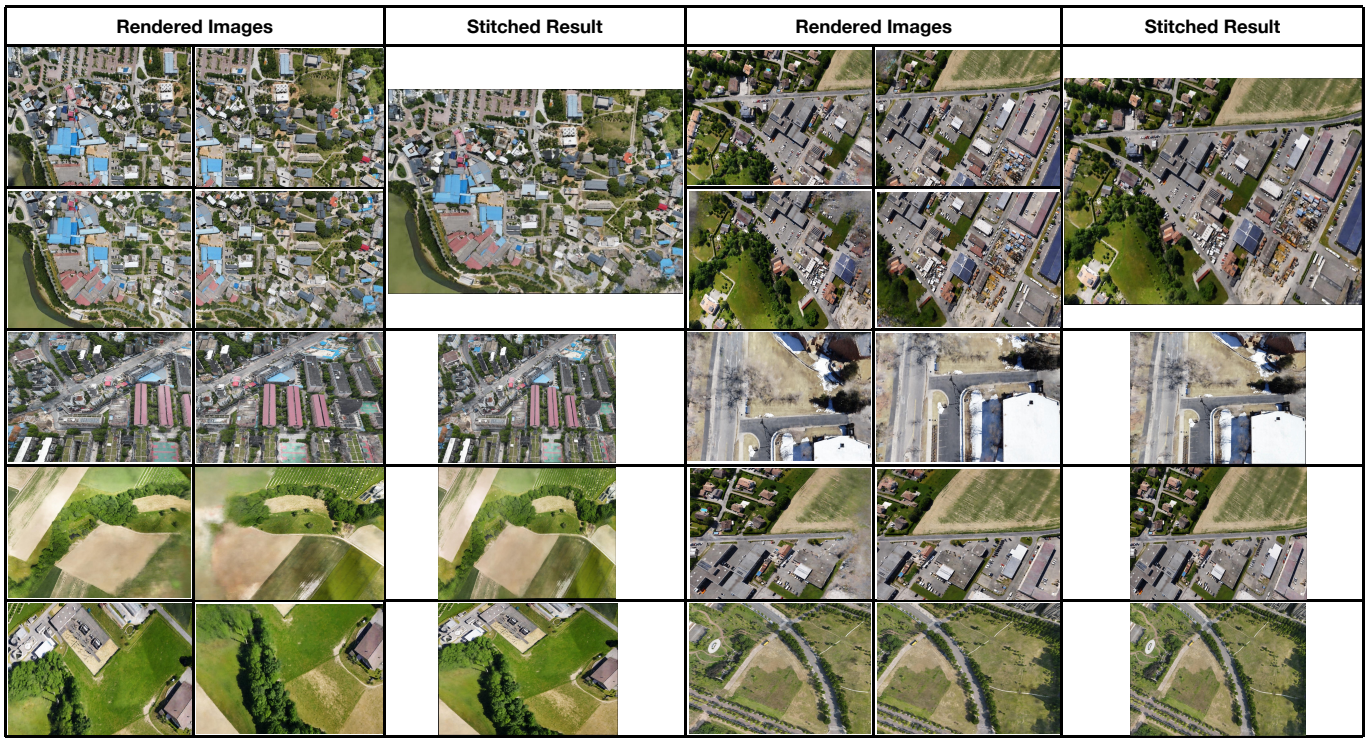

Fig. 10. More results of randomly generated camera pose which may need a stitch.

insight into our approach's versatility and potential challenges in handling various camera poses.

4) Quantitative analysis: We conduct a quantitative analysis of the rendering results and training speed across different methods. Our findings reveal that our method excels in rapidly attaining high-quality results in large-scale 3D reconstruction, even under constraints of limited GPU memory resources.

PSNR and SSIM scores. Tab. I depicts the PSNR and SSIM scores, offering a comparative analysis of the rendering results from all evaluated methods in this study against the ground truth images. Our method consistently achieves the highest PSNR scores across all datasets, securing SSIM superiority on nearly half of them, demonstrating greater stability in both metrics. Metascape displays a broader range of PSNR fluctuations. In contrast, our method maintains a consistently narrow range of fluctuations in scores, indicating its robustness and ability to deliver consistent and reliable results.

Runtime comparison. Instant-NGP demonstrates rapid convergence after 3×10^3 training iterations, whereas Mega-NeRF, an enhanced version of the initial NeRF, requires approximately 1×10^5 iterations to reach convergence. To ensure sufficient training for both Mega-NeRF and our method BirdNeRF, we set the training iterations for our method to 5×10^3 and for Mega-NeRF to 1×10^5 iterations. In Mega-NeRF's clustering mask partitioning stage, we set the grid dimension to 2x4, dividing each scene into 8 sub-scenes. Utilizing a single NVIDIA GeForce RTX 3090 GPU (24GB), the training environment allows for parallel training of two sub-scenes in Mega-NeRF. Additionally, the time Metashape consumes from sparse point cloud to mesh generation is considered the training time.

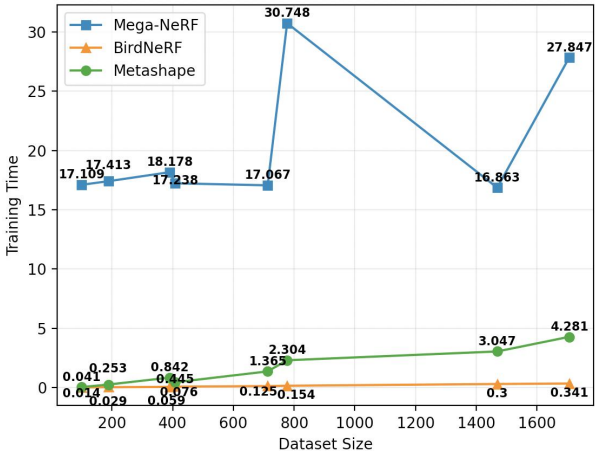

Fig. 11. The figure above illustrates the trend of training time across all datasets. Our method shows a very gradual increase with the growth of data volume, and the larger the dataset, the more pronounced the advantage of our approach becomes.

Tab. II provides a comprehensive comparison of training times for various methods, highlighting the significant speed advantage of our approach. Fig. 11 visually represents this result, clearly illustrating the overall lengthy training times of Mega-NeRF, based on the naive NeRF implementation. Metashape's training time escalates quickly with the dataset size, whereas our method exhibits superior adaptability to large datasets with a slow increase in training time as the dataset grows.

The training time of our method remains independent of image resolution but is contingent on the number of training

TABLE I
PSNR AND SSIM SCORES OF RENDERED RESULTS FROM METASHAPE[27], MEGA-NERF[5],INSTANT-NGP[13] AND BIRDNERF.

Dataset	Method	PSNR ↑			SSIM ↑		
		Min	Max	Average	Min	Max	Average
UA	Metashape	13.619	24.713	19.891	0.890	0.963	**0.930**
	Mega-NeRF	9.860	13.106	11.382	0.244	0.355	0.282
	Instant-NGP	19.787	21.35	20.154	0.514	0.617	0.562
	BirdNeRF	19.603	20.768	**20.167**	0.563	0.6560	0.595
SA	Metashape	5.135	34.819	18.517	0.686	0.995	**0.935**
	Mega-NeRF	7.105	12.514	9.808	0.146	0.469	0.296
	Instant-NGP	\	\	\	\	\	\
	BirdNeRF	16.508	23.120	**20.002**	0.448	0.688	0.555
IZAA	Metashape	6.353	35.99	21.523	0.654	0.994	**0.887**
	Mega-NeRF	8.757	19.116	12.845	0.124	0.548	0.276
	Instant-NGP	\	\	\	\	\	\
	BirdNeRF	17.223	28.847	**21.829**	0.399	0.775	0.544
CSU1	Metashape	2.492	30.568	9.787	0.427	0.987	0.686
	Mega-NeRF	16.813	23.428	20.101	0.401	0.696	0.540
	Instant-NGP	\	\	\	\	\	\
	BirdNeRF	18.226	26.255	**22.579**	0.602	0.772	**0.689**
CSU2	Metashape	5.974	31.373	19.167	0.725	0.985	**0.911**
	Mega-NeRF	8.887	15.577	12.308	0.125	0.395	0.257
	Instant-NGP	\	\	\	\	\	\
	BirdNeRF	11.013	25.290	**19.868**	0.286	0.701	0.539
HNU	Metashape	5.806	28.251	21.127	0.723	0.988	**0.929**
	Mega-NeRF	10.236	13.852	11.937	0.188	0.319	0.252
	Instant-NGP	\	\	\	\	\	\
	BirdNeRF	17.664	22.954	**21.158**	0.427	0.618	0.552
CSUS	Metashape	0.000	39.632	12.885	0.000	0.998	0.541
	Mega-NeRF	10.102	13.325	12.023	0.227	0.423	0.314
	Instant-NGP	\	\	\	\	\	\
	BirdNeRF	19.287	23.907	**21.580**	0.448	0.659	**0.564**
CSUHU	Metashape	0.957	38.553	8.776	0.190	0.996	0.658
	Mega-NeRF	10.748	20.995	14.135	0.191	0.921	0.468
	Instant-NGP	\	\	\	\	\	\
	BirdNeRF	17.459	26.787	**21.505**	0.442	0.889	**0.659**

iterations. To showcase the effectiveness of our method, we compare training speed and rendering results on two datasets at different iteration counts. Tab. III presents the PSNR and SSIM scores for varying training iterations, revealing negligible differences in modeling performance. This flexibility allows us to adjust training iterations based on our time requirements.

IV. CONCLUSIONS

This paper introduces BirdNeRF, a fast neural reconstruction method designed for processing a large number of aerial images. It stands out as the fastest large-scale reconstruction method to date, emphasizing efficient memory resource utilization and high rendering quality. The spatial decomposition strategy, grounded in camera distribution, empowers BirdNeRF to decompose and train scenes within specified memory constraints, underscoring its robust scalability. The projection-guided novel view Re-rendering strategy ensures accurate indexing of sub-scene bounding boxes and precise querying of related sub-models, ensuring high-quality rendering for diverse camera poses. Evaluation results indicate substantial advancements over classical photogrammetric software and the state-of-the-art large-scale NeRF solutions. BirdNeRF achieves a

TABLE II
A COMPARISON OF TRAINING TIME FOR VARIOUS METHODS.

Method	Training Time(h) ↓		
	Metashape	Mega-NeRF	BirdNeRF
UA	0.041	17.109	0.014
SA	0.253	17.413	0.029
IZAA	3.047	16.863	0.300
CSU1	0.445	17.238	0.076
CSU2	1.365	17.067	0.125
HNU	0.842	18.178	0.059
CSUS	2.304	30.748	0.154
CSUHU	4.281	27.847	0.341

TABLE III
THE PSNR AND SSIM SCORES FOR DIFFERENT TRAINING ITERATIONS OF OUR METHOD BIRDNERF.

	Training iterations	Training time(h) ↓	PSNR ↑	SSIM ↑
UA	5×10^3	0.014	20.167	0.595
	1×10^4	0.029	20.432	0.610
	3×10^4	0.093	20.830	0.636
SA	5×10^3	0.029	20.002	0.555
	1×10^4	0.063	20.236	0.568
	3×10^4	11.667	20.725	0.590
IZAA	5×10^3	0.300	21.829	0.544
	1×10^4	0.599	22.231	0.554
	3×10^4	1.730	22.725	0.569
CSU1	5×10^3	0.076	22.579	0.690
	1×10^4	0.149	23.306	0.713
	3×10^4	0.461	23.766	0.738
CSU2	5×10^3	0.125	19.868	0.539
	1×10^4	0.253	20.107	0.557
	3×10^4	0.753	20.329	0.582
HNU	5×10^3	0.059	21.158	0.552
	1×10^4	0.119	21.745	0.587
	3×10^4	0.361	22.451	0.633
CSUS	5×10^3	0.154	21.580	0.564
	1×10^4	0.310	21.987	0.582
	3×10^4	0.890	22.832	0.627
CSUHU	5×10^3	0.341	21.504	0.659
	1×10^4	0.671	21.763	0.670
	3×10^4	1.927	22.132	0.687

speed increase of more than ten times on a single GPU while maintaining commendable rendering quality. Positioned as a noteworthy contribution to the field, BirdNeRF offers practical solutions for critical challenges, significantly enhancing the speed, scalability, and visual realism of large-scale aerial scene reconstruction. This improvement is particularly vital for real-time applications such as disaster response.

ACKNOWLEDGMENTS

This work is supported by the National Key R&D Program of China (No. 2022ZD0119003), Nature Science Foundation of China (No. 62102145), and Jiangxi Provincial 03 Special Foundation and 5G Program (Grant No. 20224ABC03A05).

REFERENCES

[1] Chauve, Anne-Laure and Labatut, Patrick and Pons, Jean-Philippe, "Robust piecewise-planar 3D reconstruction and completion from large-scale unstructured point data," in *CVPR*, 2010.

[2] Gay-Bellile, Vincent and Lothe, Pierre and Bourgeois, Steve and Royer, Eric and Collette, S Naudet, "Augmented reality in large environments: Application to aided navigation in urban context," *IEEE International Symposium on Mixed and Augmented Reality*, 2010.

[3] Bastanlar, Y and Grammalidis, N and Zabulis, X and Yilmaz, E and Yardimci, Y and Triantafyllidis, G,"3D reconstruction for a cultural heritage virtual tour system," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 37-B5*, 2008.

[4] Sequeira, Vitor and Ng, Kia C and Wolfart, Erik and Goncalves, Joao GM and Hogg, David C, "Automated 3D reconstruction of interiors with multiple scan views," *Videometrics VI*, 1998.

[5] Turki, Haithem and Ramanan, Deva and Satyanarayanan, Mahadev, "Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs," in *CVPR*, 2022.

[6] Gomes, Leonardo and Bellon, Olga Regina Pereira and Silva, Luciano, "3D reconstruction methods for digital preservation of cultural heritage: A survey," *Pattern Recognition Letters*, 2014.

[7] Zhang, Runze, "Towards large scale 3D reconstruction from images," *PHD thesis, https://hdl.handle.net/1783.1/93125*, 2018.

[8] Schonberger, Johannes L and Frahm, Jan-Michael, "Structure-from-motion revisited," in *CVPR*, 2016.

[9] Furukawa, Yasutaka and Hernández, Carlos and others, "Multi-view stereo: A tutorial," *Foundations and Trends® in Computer Graphics and Vision*, 2015.

[10] Mildenhall, Ben and Srinivasan, Pratul P. and Tancik, Matthew and Barron, Jonathan T. and Ramamoorthi, Ravi and Ng, Ren, "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis," in *ECCV*, 2020.

[11] Niemeyer, Michael and Mescheder, Lars and Oechsle, Michael and Geiger, Andreas, "Differentiable Volumetric Rendering: Learning Implicit 3D Representations Without 3D Supervision," in *CVPR*, 2020.

[12] Sitzmann, V. and Zollhfer, M. and Wetzstein, G., "Scene representation networks: Continuous 3d-structure-aware neural scene representations," in *NeurIPS* , 2019.

[13] Müller, Thomas and Evans, Alex and Schied, Christoph and Keller, Alexander, "Instant neural graphics primitives

with a multiresolution hash encoding," *ACM Transactions on Graphics (ToG)*, 2022.

[14] Özyeşil, Onur and Voroninski, Vladislav and Basri, Ronen and Singer, Amit, "A survey of structure from motion*," *Acta Numerica*, 2017.

[15] Hoppe, Hugues and DeRose, Tony and Duchamp, Tom and McDonald, John and Stuetzle, Werner, "Surface reconstruction from unorganized points," in *Proceedings of the 19th annual conference on computer graphics and interactive techniques*, 1992.

[16] Kazhdan, Michael and Bolitho, Matthew and Hoppe, Hugues, "Poisson surface reconstruction," in *Proceedings of the fourth Eurographics symposium on Geometry processing*, 2006.

[17] Wu, Changchang, "VisualSFM: A visual structure from motion system" *http://www. cs. washington. edu/homes/ccwu/vsfm*, 2011.

[18] Moulon, Pierre and Monasse, Pascal and Perrot, Romuald and Marlet, Renaud, "Openmvg: Open multiple view geometry" in *Reproducible Research in Pattern Recognition: First International Workshop*, 2017.

[19] Tancik, Matthew and Casser, Vincent and Yan, Xinchen and Pradhan, Sabeek and Mildenhall, Ben and Srinivasan, Pratul P and Barron, Jonathan T and Kretzschmar, Henrik, "Block-nerf: Scalable large scene neural view synthesis," in *CVPR*, 2022.

[20] Shan, Qi and Adams, Riley and Curless, Brian and Furukawa, Yasutaka and Seitz, Steven M, "The visual turing test for scene reconstruction" in *2013 International Conference on 3D Vision*, 2013.

[21] Zhang, Kai and Riegler, Gernot and Snavely, Noah and Koltun, Vladlen, "Nerf++: Analyzing and improving neural radiance fields," *arXiv preprint arXiv:2010.07492*, 2020.

[22] Barron, Jonathan T and Mildenhall, Ben and Verbin, Dor and Srinivasan, Pratul P and Hedman, Peter, "Mip-nerf 360: Unbounded anti-aliased neural radiance fields" in *CVPR*, 2022.

[23] Rebain, Daniel and Jiang, Wei and Yazdani, Soroosh and Li, Ke and Yi, Kwang Moo and Tagliasacchi, Andrea, "Derf: Decomposed radiance fields," in *CVPR*, 2021.

[24] Reiser, Christian and Peng, Songyou and Liao, Yiyi and Geiger, Andreas, "Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps" in *ICCV*, 2021.

[25] Yu, Alex and Ye, Vickie and Tancik, Matthew and Kanazawa, Angjoo, "pixelnerf: Neural radiance fields from one or few images," *CVPR*, 2021.

[26] "PIX4Dmapper," *https://www.pix4d.com/product/pix4dmapper-photogrammetry-software/*.

[27] "Agisoft Metashape," *https://www.agisoft.com/*.

[28] Snavely, Noah and Seitz, Steven M and Szeliski, Richard, "Photo tourism: exploring photo collections in 3D," in *SIGGRAPH*, 2006.

[29] Agarwal, Sameer and Furukawa, Yasutaka and Snavely, Noah and Simon, Ian and Curless, Brian and Seitz, Steven M and Szeliski, Richard, "Building rome in a day," *Communications of the ACM*, 2011.

[30] Zhu, Siyu and Zhang, Runze and Zhou, Lei and Shen, Tianwei and Fang, Tian and Tan, Ping and Quan, Long, "Very Large-Scale Global SfM by Distributed Motion Averaging," in *CVPR*, 2018.

[31] Zhang, Runze and Zhu, Siyu and Fang, Tian and Quan, Long, "Distributed Very Large Scale Bundle Adjustment by Global Camera Consensus," in *ICCV*, 2017.

[32] Zhang, Runze and Li, Shiwei and Fang, Tian and Zhu, Siyu and Quan, Long, "Joint camera clustering and surface segmentation for large-scale multi-view stereo," in *ICCV*, 2015.

[33] Xiangli, Yuanbo and Xu, Linning and Pan, Xingang and Zhao, Nanxuan and Rao, Anyi and Theobalt, Christian and Dai, Bo and Lin, Dahua, "Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering," in *ECCV*, 2022.

[34] Hartigan, John A and Wong, Manchek A, "Algorithm AS 136: A k-means clustering algorithm," *Journal of the royal statistical society*, 1979.

[35] Miller, Steven J, "The method of least squares," *Mathematics Department Brown University*, 2006.

[36] "Image Stitching(Part of a project for the Computer Vision course of CentraleSupélec)," *https://github.com/CorentinBrtx/image-stitching*.

[37] Brown, Matthew and Lowe, David G, "Automatic panoramic image stitching using invariant features," *International journal of computer vision*, 2007.

[38] "Industrial zone and agriculture area, Urban area and Suburban area," *Courtesy of Pix4D, https://support.pix4d.com/hc/en-us/articles/360000235126-Example-projects-real-photogrammetry-data*.

[39] Su, Bolan and Lu, Shijian and Tan, Chew Lim, "Blurred image region detection and classification," *ACM International Conference on Multimedia*, 2011.

Huiqing Zhang Huiqing Zhang is a master's candidate at Hunan University, advised by Prof. Yizhen Lao, where she works in Computational Photography and 3D Computer Vision. She focuses on 3D Reconstruction and Neural Rendering. Before that, She got Bachelor's Degree in Computer Science and Technology from Jiangxi Normal University. She previously led the Association of Computer Science at JXNU, and her undergraduate experience mainly focused on the Programming Contest.