

# High-fidelity 3D Reconstruction of Plants using Neural Radiance Field

Kewei Hu<sup>1</sup>, Ying Wei<sup>1</sup>, Yaoqiang Pan<sup>1</sup>, Hanwen Kang<sup>1,2</sup>, Chao Chen<sup>2,2</sup>

**Abstract**—Accurate reconstruction of plant phenotypes plays a key role in optimising sustainable farming practices in the field of Precision Agriculture (PA). Currently, optical sensor-based approaches dominate the field, but the need for high-fidelity 3D reconstruction of crops and plants in unstructured agricultural environments remains challenging. Recently, a promising development has emerged in the form of Neural Radiance Field (NeRF), a novel method that utilises neural density fields. This technique has shown impressive performance in various novel vision synthesis tasks, but has remained relatively unexplored in the agricultural context. In our study, we focus on two fundamental tasks within plant phenotyping: (1) the synthesis of 2D novel-view images and (2) the 3D reconstruction of crop and plant models. We explore the world of neural radiance fields, in particular two SOTA methods: Instant-NGP, which excels in generating high-quality images with impressive training and inference speed, and Instant-NSR, which improves the reconstructed geometry by incorporating the Signed Distance Function (SDF) during training. In particular, we present a novel plant phenotype dataset comprising real plant images from production environments. This dataset is a first-of-its-kind initiative aimed at comprehensively exploring the advantages and limitations of NeRF in agricultural contexts. Our experimental results show that NeRF demonstrates commendable performance in the synthesis of novel-view images and is able to achieve reconstruction results that are competitive with Reality Capture, a leading commercial software for 3D Multi-View Stereo (MVS)-based reconstruction. However, our study also highlights certain drawbacks of NeRF, including relatively slow training speeds, performance limitations in cases of insufficient sampling, and challenges in obtaining geometry quality in complex setups. In conclusion, NeRF introduces a new paradigm in plant phenotyping, providing a powerful tool capable of generating multiple representations, such as multi-view images, point cloud and mesh, from a single process.

**Index Terms**—Deep-learning, Robotics, NeRF, Phenotyping

## I. INTRODUCTION

In recent years, integration of emerging sensors and Artificial Intelligence (AI) has revolutionized precision agriculture (PA), significantly enhancing the efficiency, effectiveness, and productivity of breeding and primary production in agriculture industry [1]. Unpredictable threats such as climate, soil characteristics, insect pests, etc. are the main challenges to maintaining and guaranteeing crop yields [2]. This has given rise to the increasing importance of monitoring plant growth through the comprehensive analysis of plant phenotyping [3]. Phenomics studies a variety of phenotypic plant traits, such as growth, tolerance, yield, plant height, leaf area index, etc.

[4]. Traditional methods for manual phenotypic measurement and analysis were labor-intensive, time-consuming, and often destructive[5], [3], [6]. Thus, modern sensor technologies have been widely used to achieve non-invasive and high-throughput plant phenotyping[7]. The most current research shows that optical sensors dominate the detection system [8], [9] and various types of two-dimensional (2D) and three-dimensional (3D) imaging systems can directly measure morphological traits of plants, including colors of seeds, leaves, canopies, fruits, and roots, shapes and sizes of seeds, sizes, numbers, areas, textures, angles, architectures, and total volumes of canopies, leaves, and roots, and volumes sizes, shapes, numbers, and spatial distributions of fruits [10].



(a) Rendered image from NeRF. (b) Mesh from NeRF.

Fig. 1: High-fidelity 2D imaging (a) and 3D imaging (b) plant phenotypes from NeRF.

Despite the fact that 2D imaging systems deploy red, green, blue (RGB) camera to measure morphological traits (color, shape, size, and texture) of plants at affordable prices, these methods are limited by the dimensionality of the data and therefore cannot express the geometric form of the plant[10]. 3D imaging systems enables tracking exact geometry and measurement of plant traits like plant height, plant width, root volume, root surface area, leaf size, leaf width, stem angle and projected canopy area. As a result, the 3D imaging system

<sup>1</sup>K.Hu, Y.Pan, Y.wei, and H.Kang are with College of Engineering, South China Agriculture University, Guangzhou, China

<sup>2</sup> C.Chao is with Department of Mechanical and Aerospace Engineering, Monash University, Melbourne, Australia

can keep track of the actual growth status of the plant at the organ level, which is almost impossible for the 2D imaging system[10], [3]. However, the current phenotyping methods still face the following challenges: 1) Existing phenotyping techniques cannot obtain multiple types of representations in a single collection paradigm, leading to incomplete data acquisition process. 2) In addition to the RGB camera, some other sensors that are used in phenotyping require a stable operational platform for manual collection. This means that it is difficult to collect phenotypic data beyond RGB image data using robots to replace manual collection. As a result, there's a significant risk of human error. 3) Some sensors compromise data quality to increase flexibility. For instance, while inexpensive depth cameras can substitute for Light Detection And Ranging (LiDAR) to capture point clouds, the resulting data often has lower resolution and is severely degenerated by noise and outliers due to the uniqueness of agricultural environments.

Very recently, a deep learning-based method has the potential to achieve geometric information extraction for 3D imaging while using inexpensive RGB cameras to capture 2D image data of a scene and it has gained widespread attention due to its high-fidelity reconstruction of complex objects and scenes: Neural radiance field (NeRF) [11]. NeRF was first proposed to use volume rendering formula to achieve highly photorealistic view synthesis with implicit neural scene representation via Multi-Layer Perceptron (MLP). Essentially, NeRF uses an MLP network  $H_\Theta$  to describe the mapping (1) between density  $\sigma$  and directional emitted color  $c = (r, g, b)$  of each point in a 3D scene with the spatial coordinates  $(x, y, z)$  and corresponding viewing direction vector  $\mathbf{d}$ .

$$H_\Theta(x, y, z, \mathbf{d}) \rightarrow (c, \sigma). \quad (1)$$

A significant advantage of NeRF is the ability to generate high-quality images of new views that are not merely interpolated, but are true inferences of the underlying scene geometry. This capability is valuable for plant phenotyping, where it is neither feasible nor efficient to manually capture all possible views of a plant or crop. Besides, although the main function of NeRF is implicit scene representation and view synthesis, its density information is stored in the MLP, which provides an important insight and basis for the extraction of geometry. Based on these considerations, research on NeRF is likely to be a key bridge between 2D imaging and 3D imaging, two types of plant phenotyping acquisitions, in order to establish a low-cost, high-throughput, non-invasive plant phenotyping system.

Therefore, this study investigates the performance of the latest NeRF model in 2D view synthesis as well as 3D geometry extraction based on the research content of traditional 2D imaging system and 3D imaging system by acquiring images in a variety of plant growth environments. Specifically, the contributions of this paper are as follows:

- A novel technique, NeRF, was exported to agricultural applications in this study, particularly for high-fidelity plant phenotyping.
- A thorough investigation was conducted on the central tasks of extracting high-fidelity multi-view RGB images

and intricate topological geometries using NeRF in actual agricultural scenarios.

- A comparison of several state-of-the-art (SOTA) NeRF models in terms of generating new viewpoints and extracting geometric structures was provided, offering invaluable insights for subsequent research.

The rest of this paper is organised as follows. Section II surveys related work. Section III delineates the fundamental implementation of rendering novel perspective images and extracting essential geometry. Section IV provides a detailed description of the actual implementation of our methodology. The experiment results and discussion are presented in Section V, followed by the conclusion in Section VI.

## II. RELATED WORKS

### A. Plant phenotyping

Measuring and analysing plant phenotypes can be used to establish predictive models to assess plant growth characteristics, which are important for precision agriculture as a decision-making tool[3]. Therefore, it is crucial to investigate non-invasive, affordable and efficient methods for plant phenotyping[12]. In recent years, a large number of scholars have made many attempts to combine novel sensors with computer technology. Among them, 2D imaging, which studies plant traits such as color through multi-view RGB imaging, and 3D imaging[13], [10], which focuses on geometry extraction, have become one of the most important research interests in the field because they serve the most fundamental and widely concerned morphological plant traits.

### B. 2D imaging: Multi-view RGB Imaging

RGB imaging from various perspectives serves distinct purposes in plant phenotyping and growth monitoring[14], [15]. Kang et al. detected the location of fruits by processing RGB images of apple trees through deep learning [16]. Top-view RGB imaging systems are typically employed when examining rosette plants to extract growth rate data. These systems capture top-down RGB images of plants such as *Arabidopsis* (*Arabidopsis thaliana*) and tobacco (*Nicotiana tabacum*) to investigate growth rates under conditions of drought stress, chilling stress, and biotic stress[17], [18]. Plant growth analysis based on top-view images is impacted by challenges such as overlapping leaves and the nastic movement of foliage[19]. These obstacles become particularly pronounced when imaging is limited to a single perspective. Multiview RGB images of cereals, including barley, wheat, rice, sorghum, and various pea field cultivars, are harnessed for the study of growth rates under conditions of drought stress, salt stress, cold stress, and nutrient deficiency[20].

### C. 3D imaging: Geometry Extraction

Geometry Extraction of complex unstructured agricultural scenes is the key prerequisite for quantitative extraction of plant metrics. A number of papers applied technologies, which can be divided into two main categories, to obtain the geometric representation of a scene [21], [22]. The first is the

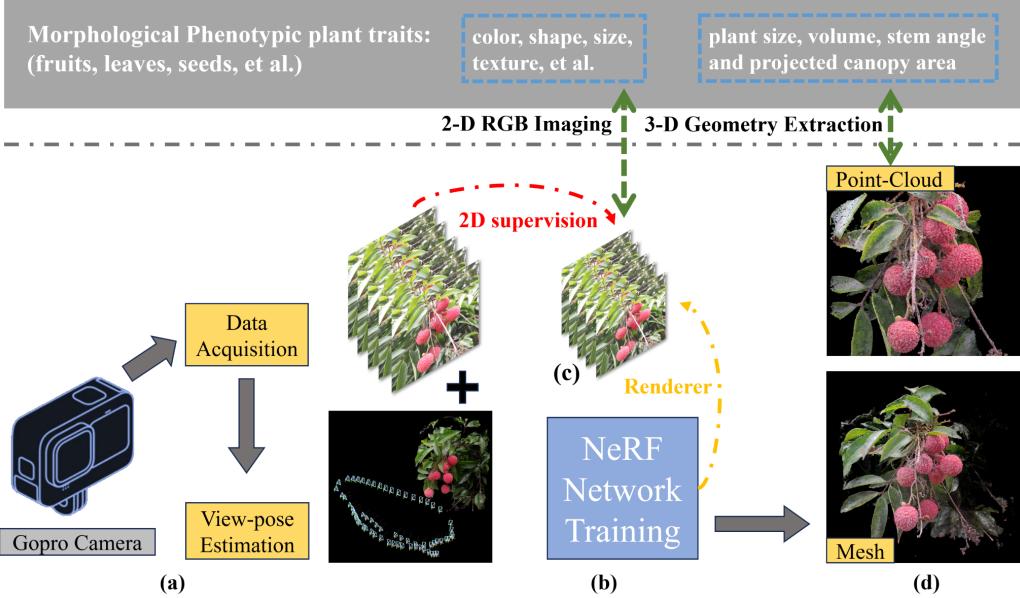


Fig. 2: Framework of phenotyping system via NeRF: (a) Data Preparation, (b) Network Training, (c) Images Rendering, (d) Geometry Extraction.

explicit method, represented by LiDAR, while the other is the implicit method, with Signed Distance Function (SDF) as its representative.

**1) Explicit methods :** Guo et al. utilized the Realsense D435i to capture continuous multi-view images of the cabbage and input the images into professional 3D reconstruction software called RealityCapture to create 3D point cloud data for calculating the target cabbage dimensions [23]. Wu et al. developed a detachable and adjustable according to the size of the target shoot to acquire multiview stereo (MVS) images and reconstructed 3D point clouds using MVS-based commercial software [24]. The aforementioned studies calculate the internal parameters of the images, along with the external parameters between them, using feature matching in a series of unordered images. They then proceed to sequentially perform sparse point cloud reconstruction and dense point cloud generation. The quality of the results obtained through these methods is heavily reliant on the resolution and volume of image data, making the process time-consuming. For instance, Guo et al. required 5-8 minutes to capture 150 photographs and an additional 20 minutes or more to complete the reconstruction of a single cabbage. Kang et al. proposed a LiDAR-color fusion-based visual sensing and perception strategy for achieving precise scene comprehension and fruit localization in orchards [25], [26]. While this method enhances the density of point cloud data and depth sensor accuracy, it remains costly and time-consuming to accumulate a sufficient number of point clouds. Eugene Kok et al. processed RGB data and depth information from a depth camera using a semantic segmentation network and deep learning skeletonization method to reconstruct spatial information of both visible and hidden branches from a single-view image [27]. However, the algorithm can only reconstruct trees from a single viewpoint and is not suitable for trees with more

complex geometries. Yang et al. developed a system for rapid 3D model reconstruction using RGB-D cameras and the point cloud self-registration method [28]. Although they introduced a rotating table to obtain a complete point cloud, this method is only applicable to potted plants and not suitable for field conditions.

**3) Implicit methods :** Since the conventional reconstruction method represented a 3D scene explicitly using grids of voxels, point clouds, or meshes, the reconstructed 3D shapes were discrete and at a limited resolution. The novel implicit methods parameterize different kinds of features from the scene (for instance, density, color, occupancy probability, SDF value) as a continuous function approximated via an MLP network. Due to the high accuracy of MLP's function fitting, implicit representations of the scene are often accurate at arbitrary resolutions[29]. IM-NET (Chen et al. 2019) trains the network through deep learning using VAE+GAN, which replaced the traditional reconstruction method with a new implicit surface function decoder during the input single view implementation of 3D modeling, resulting in improved reconstruction effectiveness and efficiency[30]. Occupancy Networks (Mescheder et al. 2019) predict binary occupancy rates by acquiring feature vectors and points in space, so that the Occupancy Networks were able to implicitly represent 3D surfaces as continuous decision boundaries for deep neural network classifiers, enabling efficient 3D structural coding through the use of continuous functions to model objects in space[31]. Unlike the principle of Occupancy Networks, DeepSDF (Park et al.) implicitly represents continuous 3D spatial surfaces by directly regressing the Signed Distance Function (SDF). DeepSDF can represent more complex shapes without discrete errors and requires significantly less memory. This concept offers a promising avenue for further research on using neural networks to define implicit scene representations[32]. Nevertheless, these

implicit representations necessitate supervised learning with the pre-existing knowledge of a 3D object's shape, rendering it challenging to directly employ these techniques in real-world environments.

### III. PROBLEM STATEMENT

Scene representation on volume rendering is the essential for NeRF to generate novel viewpoint images and geometry. This section gives the formulations of the NeRF working mechanism in both novel-view rendering and geometry reconstruction.

#### A. Rendering Novel Viewpoints with NeRF

The volume rendering technique based on ray tracing[33] is used to render novel viewpoint images from an invisible MLP. During the rendering process, a ray  $r(t) = o + t\mathbf{d}$  is emitted into the 3D scene from a given camera position  $o = (x_o, y_o, z_o)$ , where  $t$  is the straight-line distance from the point on the ray to the camera's origin  $o$  and  $\mathbf{d}$  is the 3D Cartesian unit vector representing viewing direction. As the ray travels, a sufficient number of spatial points of this ray are sampled, and the  $\sigma$  and  $c$  of each point are queried to the  $H_\Theta$ . Finally, the equation (2) from classical volume rendering [33] is used to accumulate all the sampled points and obtain the color value of the corresponding pixel on the image plane.

$$C(\mathbf{r}) = \int_{t_1}^{t_2} T(t) \cdot \sigma(\mathbf{r}(t)) \cdot \mathbf{c}(\mathbf{r}(t), \mathbf{d}) \cdot dt, \quad (2)$$

where  $T(t)$  denotes the transmittance function, alternatively referred to as the accumulated density. This function quantifies the possibility that a ray traverses the distance from  $t_1$  to  $t_2$  without running into an obstruction, as described by the following equation:

$$T(t) = \exp(- \int_{t_1}^t \sigma(\mathbf{r}(u)) \cdot du), \quad (3)$$

For every pixel, a squared error photometric loss is employed for the optimization of the weight  $\Theta$  of MLP . When applied across the entire image, this loss is represented as follows:

$$\mathcal{L}_{\text{NeRF}} = \sum_{r \in R} \|C(\mathbf{r}) - C_{gt}(\mathbf{r})\|^2. \quad (4)$$

where  $C_{gt}(\mathbf{r})$  is the ground truth color of the training image pixels associated with the ray  $\mathbf{r}$ , and  $R$  is a batch of rays associated with the image to be synthesised.

#### B. Extraction Geometry From NeRF

1) *Point-Cloud Extraction from NeRF*: While NeRF primarily focuses on the reconstruction and rendering of scenes from novel viewpoints, it inherently contains a wealth of 3D structural information, making it possible to extract point-cloud data. The continuous feature of NeRF's representation allows the inference of spatial geometries by observing changes in radiance and density along camera rays.

The crucial step in point-cloud extraction is depth estimation. By analyzing the transmittance function  $T(t)$  in (3) along

a ray  $\mathbf{r}(t)$ , one can observe the depth where the function experiences significant change, indicating the presence of a surface. The depth at which  $T(t)$  sees a sharp decline is usually aligned with the surface of an object within the scene. Mathematically, this depth  $t_{surface}$  for a ray  $\mathbf{r}(t)$  can be pinpointed as the position where the transmittance's rate of change is most abrupt by minimizing the first order derivative of  $T(t)$  relative to  $t$ :

$$t_{surface} \approx \arg \min_t \left( \frac{dT(t)}{dt} \right), \quad (5)$$

With the depth approximated, the next step is to calculate the corresponding 3D point on  $r(t) = o + t\mathbf{d}$  :

$$\mathbf{p} = \mathbf{o} + t_{surface}\mathbf{d}, \quad (6)$$

Where  $\mathbf{p}$  represents the 3D point,  $\mathbf{o}$  is the camera's origin,  $\mathbf{d}$  signifies the ray direction and  $t_{surface}$  represents the distance travelled by the ray as it crosses the surface.

Upon determining the 3D position, the color value at this position can be directly queried from NeRF (1):

$$\text{color} = \mathbf{c}(\mathbf{p}, \mathbf{d}). \quad (7)$$

Repeating these steps for every pixel across one or multiple images generates a dense point-cloud. Each point within this cloud corresponds to a surface in the original scene, carrying a color that mirrors the appearance of that surface under the sampled viewing direction.

2) *Mesh Extraction from NeRF*: Given a predefined 3D region of interest, a set of spatial points  $P = \{p_1, p_2, \dots, p_n\}$  is generated via dense volumetric sampling. For each point  $p_i \in P$ , it's evaluated through the NeRF model to obtain The density values,  $\sigma(p_i) = \text{NeRF}_\sigma(p_i)$ , form the basis for surface extraction. The Marching Cubes [34] algorithm identifies the isosurface by thresholding the density values:

$$M = \text{MarchingCubes}(P, \sigma_{\text{threshold}}), \quad (8)$$

Where  $M$  is the resultant mesh and  $\sigma_{\text{threshold}}$  is an optimal density value demarcating the object's boundary.

For every vertex  $v_j$  in mesh  $M$  , a viewing ray  $r_j$  is constructed and queried  $\mathbf{c}(v_j) = \text{NeRF}_\mathbf{c}(v_j, r_j)$  in NeRF. Those radiance values derived from NeRF are mapped onto mesh  $M$ , assigning color to each vertex:

$$\text{Color}(v_j) = \mathbf{c}(v_j), \quad (9)$$

For a 2D texture representation, the vertex-colored mesh undergoes UV unwrapping [35]. To minimize distortion, algorithms like Least Squares Conformal Mapping (LSCM) [36] can be employed. The objective of LSCM is to minimize the conformal energy:

$$E(u, v) = \int_{\Omega} (|\nabla u|^2 + |\nabla v|^2) dA. \quad (10)$$

Where  $(u, v)$  are the 2D texture coordinates for each vertex in  $M$ ,  $\Omega$  represents the object's surface, and  $dA$  is a differential area element on the mesh's surface, indicating that the energy is computed by integrating over the entire surface of the mesh.

#### IV. METHODS

To adapt NeRF effectively for agricultural scenarios, we made specific enhancements to address its initial limitations. Recognizing that the original NeRF training was slow, we introduced hash encoding of Instant-NGP[37] to speed up the training process. To strengthen the geometric constraints within the model, we brought in a rendering method based on SDF[38], [39]. In this chapter, we will walk through the entire process, from image data collection and preparation for training to the architecture and training of the network model. Finally, we will give an overview of the experimental setup used in our study.

##### A. Data Acquisition

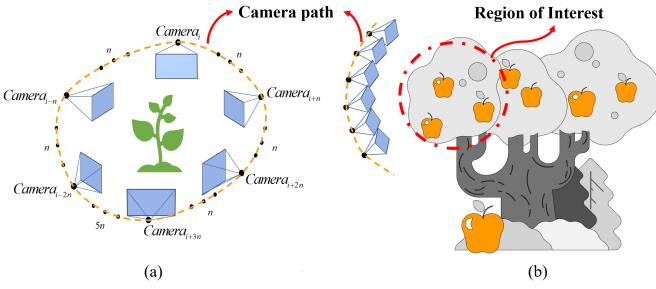


Fig. 3: (a) 360° image capturing, (b) front views capturing.

We captured high-resolution plant images from various agricultural scenes using the GoPro Hero 11 action camera in Tab. I. To reduce motion blur and graphic quality issues, the camera was set to work at 120 Hz in a 4K resolution linear imaging mode. This setting allowed us to collect image data at a rate of 120 frames per second with a resolution of 3840x2160 pixels. For single plants in Fig. 3(a), we aimed to capture 360° all-around images to cover all details. For larger, more complex scenes in Fig. 3(b), we chose front views of regions of interest and took images from multiple angles.

Parameter	Value
Name	GoProHERO11
Weight	470.00g
Resolution	4K+
Lens Stabilization	Electronic
Battery Life	90 minutes

TABLE I: Camera Parameters for GoProHERO11

##### B. View-pose estimation

COLMAP[40], a SOTA Structure-from-Motion (SfM) and Multi-View Stereo (MVS) pipeline, reconstructs 3D models from unordered image sets. The pipeline of SfM was used to estimate images' poses as a prior to supervise the training of the network.

For every image in the datasets, COLMAP detects and describes local features. A pairwise matching algorithm then associates keypoints based on their descriptors. Formally, if  $p_i$

and  $p_j$  are keypoints in images  $I_i$  and  $I_j$  respectively, their match is established based on:

$$D(p_i, p_j) = \|\text{desc}(p_i) - \text{desc}(p_j)\|_2, \quad (11)$$

Where  $\text{desc}(p)$  returns the descriptor for keypoint  $p$ . After feature matching, geometrically consistent matches are pinpointed by employing a fundamental or essential matrix. The essential matrix, denoted as  $E$ , captures the geometric relationship between two calibrated images. It is a matrix that relates corresponding points in one image to epipolar lines in the other image. Formally:

$$p_j^T E p_i = 0, \quad (12)$$

Where  $p_i$  and  $p_j$  are corresponding points in homogeneous coordinates.

COLMAP utilizes an incremental approach to SfM. Starting with a pair of images with the largest number of geometrically consistent matches, the scene is incrementally expanded by registering additional images based on shared keypoints.

After initial camera pose estimation, COLMAP refines the camera parameters, 3D structure, and even image keypoints simultaneously using bundle adjustment. The objective function  $J$  being minimized is:

$$J = \sum_{i,j} w_{ij} \|p_j - \pi(P_i, X_j)\|^2. \quad (13)$$

Where  $w_{ij}$  is a visibility term, which is 1 if point  $X_i$  is visible in image  $I_i$  and 0 otherwise.  $\pi$  is the projection function.  $P_i$  is the projection matrix of the  $i$ -th image.  $X_j$  is the 3D position of the  $j$ -th point.

##### C. Learning From NeRF

**Multi-resolution hash encoding.** To address the drawback of slow nerf model training, Instant-NGP [37] makes use of a multi-resolution hashed positional encoding as additional learned features, the model could represent scenes accurately with tiny and efficient MLPs. In detail, Instant-NGP operates on the premise that the object to be reconstructed is enclosed within multi-resolution voxel grids. Each of these voxel grids at different resolutions is then correspondingly linked to a hash table, featuring a fixed-size array of adaptable feature vectors.

For any spatial point  $\mathbf{x} \in \mathbb{R}^3$  within various resolution grids, it obtains the hash encoding  $h^i(\mathbf{x}) \in \mathbb{R}^d$  ( $d$  is the dimension of a feature vector,  $i = 1, \dots, L$ ) corresponding to the respective level by trilinear interpolation. The hash encodings at all  $L$  levels are subsequently concatenated to form the multi-resolution hash encoding  $h(\mathbf{x}) = \{h^i(\mathbf{x})\}_{i=1}^L \in \mathbb{R}^{L \times d}$ .

**Volume rendering of SDF.** To precisely extract the geometric surface from NeRF's implicit representation, Neus [38] proposes to represent 3D scene as a signed distance function (SDF)  $f(\mathbf{x}) : \mathbb{R}^3 \rightarrow \mathbb{R}$  instead of NeRF's density field and introduce a  $S$ -density  $\phi_b(f(\mathbf{x})) = be^{-bf(\mathbf{x})}/(1 + e^{-bf(\mathbf{x})})^2$ , where  $b$  is a trainable hyper parameter and gradually increases to a large number as the network training converges. And the surface  $\mathcal{S}$  can be extracted by the zero-level set of its SDF:

$$\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^3 | f(\mathbf{x}) = 0\}, \quad (14)$$

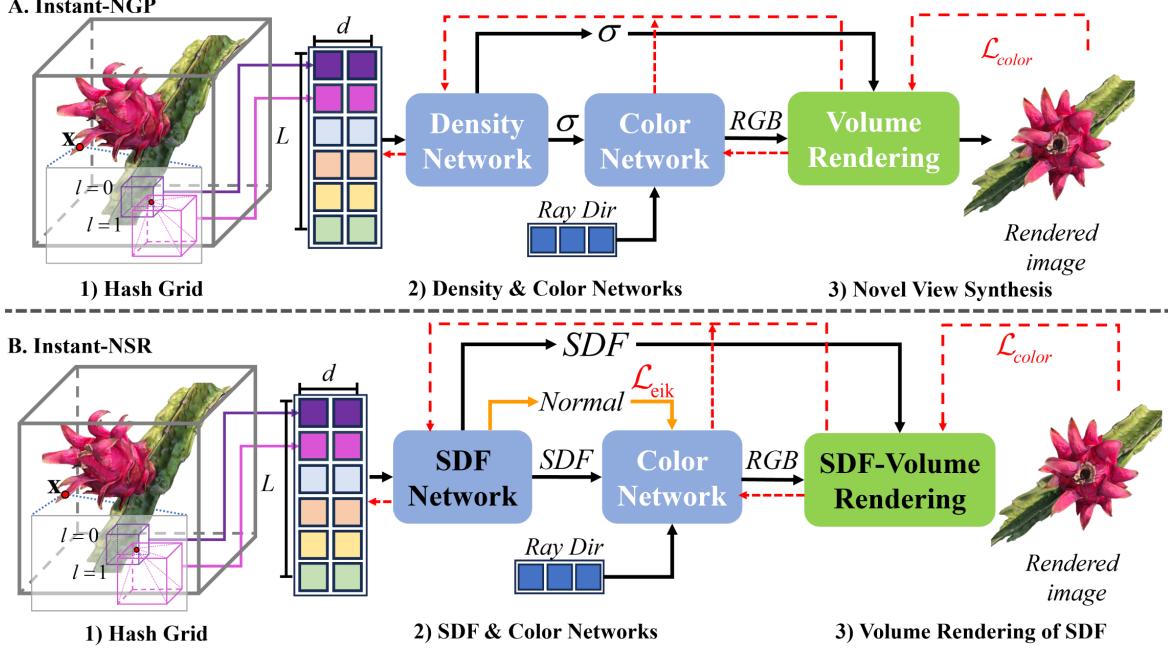


Fig. 4: Pipeline of Instant-NGP and Instant-NSR. **A). Instant-NGP:** Given a 3D point  $\mathbf{x}$ , The 1)hash grid corresponding to each level  $l$  in the voxel grid is interpolated to hash encoding, then the density and color values are predicted by the 2)MLPs of density and color, and the color of the pixel is calculated by 3)volumetric rendering. **B). Instant-NSR:** Compared to the previous NGP, both the 2) MLPs and the 3)volume rendering are based on SDF and employ an extra normal regularisation to strengthen the geometrical constraints in the network training.

To train the neural SDF representation, Neus followed NeRF’s volume rendering equation 2. Given a pixel, the renderer emitted a ray from this pixel as  $\{\mathbf{p}(t) = \mathbf{o} + t\mathbf{v}|t \geq 0\}$ , where  $\mathbf{o}$  is origin of the ray and  $\mathbf{v}$  is the ray direction and accumulate the colors along the ray by:

$$C(\mathbf{o}, \mathbf{v}) = \int_0^{+\infty} w(t)c(\mathbf{p}(t), \mathbf{v})dt, \quad (15)$$

where  $C(\mathbf{o}, \mathbf{v})$  is the rendered color for this pixel, and  $c(\mathbf{p}(t), \mathbf{v})$  the sampled colors along the ray. Especially, the weight  $w(t)$  for point  $\mathbf{p}(t)$  is rebuilt by unbiased and occlusion-aware properties to guarantee that the surface of an actual object contributes the most to the rendering result, that is:

$$w(t) = \frac{\phi_s(f(\mathbf{p}(t)))}{\int_0^{+\infty} \phi_s(f(\mathbf{p}(u)))du}. \quad (16)$$

**Truncated SDF Hash Grids.** To increase the stability of network training, here, we introduce a neural surface reconstruction method that accelerates training with Hash encoding, Instant-NSR[39]. as in Fig. 4, this method is similar to Instant-NGP [37] in that it Hash encodes points in spatial at the front-end of the neural network, but employs Neus’s SDF architecture [38] in the neural network instead of the NGP’s density architecture, and in the image renderer, also SDF-based volume rendering formulation (15) is used.

In addition, Instant-NSR uses Truncated SDF (TSDF) to skilfully solve the convergence problem caused by applying SDF representations to hash coding frameworks. Since original SDF-based methods utilize cumulative density distribution  $\phi_b(f(\mathbf{x})) = be^{-bf(\mathbf{x})}/(1 + e^{-bf(\mathbf{x})})^2$  to the compute  $w(t)$  in

equation (16), the term  $-bf(\mathbf{x})$  will be a large positive number when  $b$  is increased, resulting in  $e^{-bf(\mathbf{x})}$  closing to infinity. This numerical instability will cause the network to converge hardly during the training process. The characteristic of TSDF value between -1 to 1 can effectively prevent the occurrence of network divergence caused by numerical overflow. Therefore, we utilize the sigmoid function  $\pi(\cdot)$  after the SDF output of the network to achieve the truncation effect of the TSDF, as below:

$$\pi(f(\mathbf{x})) = \frac{1 - e^{-bf(\mathbf{x})}}{1 + e^{-bf(\mathbf{x})}}. \quad (17)$$

Thus, we can now replace the formula  $\phi_b(f(\mathbf{x})) = be^{-bf(\mathbf{x})}/(1 + e^{-bf(\mathbf{x})})^2$  in Neus with  $\phi_b(f(\mathbf{x})) = be^{-b\pi(f(\mathbf{x}))}/(1 + e^{-b\pi(f(\mathbf{x}))})^2$ .

#### D. Network Training

In order to obtain an optimal representation of the scene via our neural network model, we employ a compound loss function. This loss function is constructed using two primary components: the rendering loss and the eikonal loss.

*1) Rendering Loss:* The primary goal of our method is to produce high-quality renderings that closely match the ground truth images. Therefore, the rendering loss is crucial as it quantifies the discrepancy between the rendered images from the network and the actual images, and we generally apply this  $\mathcal{L}_{color}$  in both Instant-NGP and Instant-NSR.

Given a set of ground truth images  $I_{gt}$  and the corresponding set of images rendered by the network  $I_{pred}$ , the rendering loss  $\mathcal{L}_{color}$  is defined as:

$$\mathcal{L}_{color} = \frac{1}{N} \sum_{i=1}^N \|I_{gt}^{(i)} - I_{pred}^{(i)}\|_2^2, \quad (18)$$

where  $N$  is the total number of images in the datasets.

2) *Eikonal Loss*: While the rendering loss  $\mathcal{L}_{color}$  ensures the visual accuracy of the rendered images, the eikonal loss  $\mathcal{L}_{eik}$  is used in Instant-NSR to ensure that the estimated SDF values conform to the properties of a true SDF following one of the primary properties that the gradient  $\nabla f(\mathbf{x})$  of the SDF should have a magnitude of 1 everywhere.

Given an SDF represented by  $f(\mathbf{x})$ , the eikonal loss  $\mathcal{L}_{eik}$  is given by:

$$\mathcal{L}_{eik} = \frac{1}{M} \sum_{j=1}^M (\|\nabla f(\mathbf{x}_j)\|_2 - 1)^2, \quad (19)$$

where  $M$  is the total number of sampled points from the scene, and  $\nabla f(\mathbf{x}_j)$  is the gradient of the SDF for the  $j$ -th point.

3) *Total Loss*: Combining both losses, the total loss function  $L_{total}$  used to train the Instant-NSR network is:

$$L_{total} = \alpha L_{render} + \beta L_{eik}. \quad (20)$$

where  $\alpha$  and  $\beta$  are weighting factors that balance the contribution of the two losses. These weights are hyperparameters and are chosen based on cross-validation to achieve the best performance on a validation set.

#### E. Implementation details

1) *Image Processing and Data Preparation*: After the acquisition, the COLMAP toolbox was utilized to compute the camera parameters and relative poses for the captured images. An essential step in data preparation involved the conversion of the computed image parameters into the Local Light Field Fusion (LLFF) [41] format. This transformation was achieved through a TensorFlow implementation of the LLFF toolbox. The selection of the LLFF format was based on its robust representation, which is indispensable for the neural network models employed in this study.

2) *Computing Infrastructure*: All neural network training and computational experiments were conducted on an Ubuntu-based platform. The hardware specifications of the system included an Intel Core i9-13900 CPU. For graphic processing and deep learning computations, the system was equipped with two Colorful GeForce RTX 3090 graphics cards, providing a combined video memory of 48GB. This high-end setup ensured the efficient execution of data-intensive processes and neural network operations.

## V. EXPERIMENT AND DISCUSSION

### A. Experimental Method

In this study, we utilised a high-speed motion camera, GoPro Hero 11, to acquire the image datasets for our experiments, which were divided into three levels,  $L_1$ ,  $L_2$ , and  $L_3$ , based on the geometrical distribution of important structures such as leaves and fruits of the plants in them. Firstly, we tested the

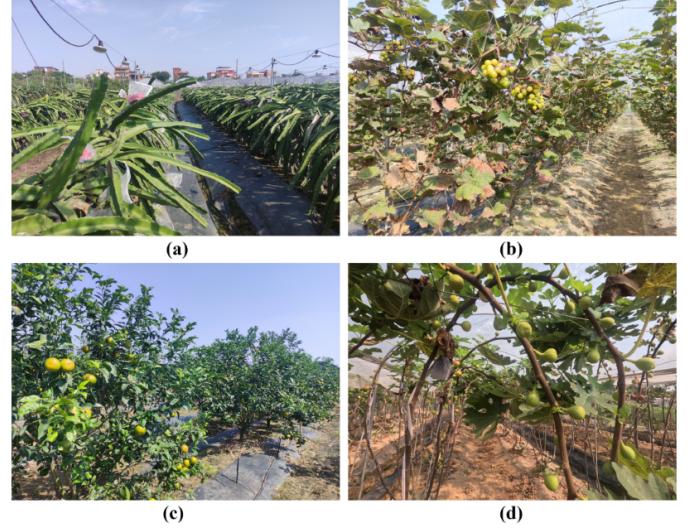


Fig. 5: Photographs of indoor and outdoor orchards: (a) pitahaya orchard, (b) grape orchard, (c) orange orchard, (d) fig orchard.

performance of Instant-NGP in rendering the images in these datasets, and secondly, we tested two different geometrically expressed NeRF models, Instant-NGP based on the density field and Instant-NSR based on the SDF, for their ability to extract the geometrical models of the plants in these datasets.

### B. Data Preparation

In this section, we collected image datasets of litchi at the Litchi Expo Park in Zengcheng District, Guangzhou, image datasets of bell peppers, tomatoes, and watermelons planted in greenhouses at the Baiyun Experimental Base of the Guangzhou Academy of Agricultural Sciences, and image datasets of grapes, pitahaya, pitahaya flowers, oranges, and figs at the Shangguo Ecological Picking Garden in Panyu District, Guangzhou in Fig.6.

1) *Levelling of datasets*: Furthermore, this study classified the scenes based on the interplay and occlusion among these key plant constituents. Three distinct levels were defined to represent these datasets:  $L_1$ ,  $L_2$ , and  $L_3$ .

- $L_1$  represents scenes where the fruits, leaves and branches are clearly visible with minimal to no occlusion between them. In such scenarios, each component is clearly visible, making it an ideal representation of less dense plant geometry.
- $L_2$  represents scenes with a slightly denser configuration. Here, several fruits overlap each other and there is slight occlusion by the leaves. This level is moderately challenging and represents environments where components begin to intertwine.
- $L_3$  is indicative of the most complex and confused scenarios. In these scenes, fruit, leaves and branches are chaotically distributed and the geometric topology is highly complex. Such environments resemble dense plant canopies and thickets, where distinguishing individual components becomes particularly challenging.

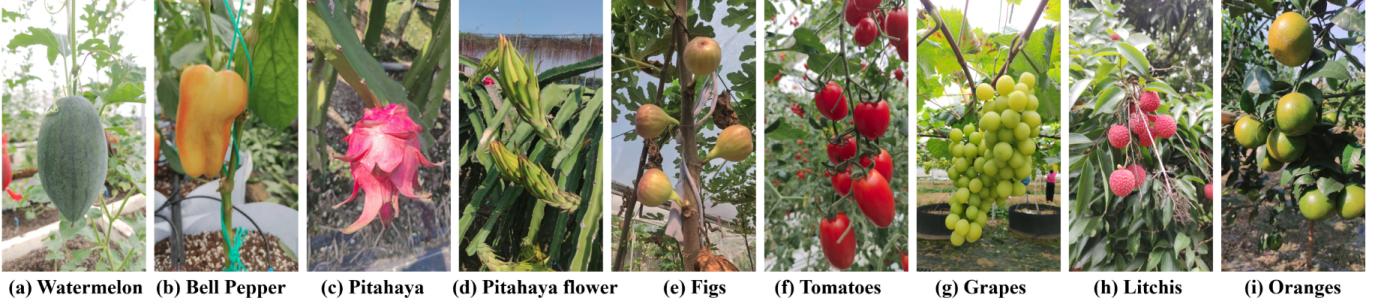


Fig. 6: Demonstration of our datasets with three progressive levels:  $L_1$ : (a),(b),(c);  $L_2$ : (d),(e),(f);  $L_3$ : (g),(h),(i).

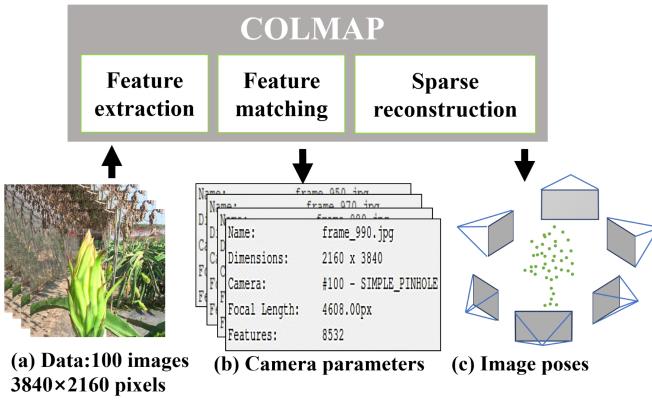


Fig. 7: Framework of data processing.

2) *Data processing and format conversion:* Before training the NeRF models, first of all, we apply COLMAP to compute the camera parameters and the poses between images according to the method in Sec. IV-B. To assist NeRF in processing datasets from different sources, these images, along with their correlated camera poses, camera parameters, are required to be converted into a certain format.

The Local Light Field Fusion (LLFF) format [41] was designed for capturing real-world scenes using a series of photographs taken from various viewpoints. Distinct from traditional light field cameras, LLFF does not require specialized hardware but leverages conventional cameras. By moving the camera throughout the scene and capturing multiple images, a scene representation is generated. Given NeRF’s (Neural Radiance Fields) objective to learn 3D representations of a scene from a series of images, the LLFF datasets format naturally becomes an optimal choice for representing input data from real scenes.

A LLFF datasets typically comprises the following crucial components: (1) A set of images captured from different perspectives, (2) Camera intrinsic parameters, (3) Camera extrinsic parameters. Within the previous steps, we have obtained the (1) image sequences and (2) intrinsic parameters of cameras. To represent the output camera poses of COLMAP in the LLFF format, it is needed to invert the transformation from a world-to-camera format (COLMAP) to camera-to-world format for LLFF.

Specifically, COLMAP outputs a rotation matrix  $R$  and a

translation vector  $t$  for each camera’s pose  $C = -R^T \cdot t$ . For a rotation matrix, which is orthogonal, the inverse  $R^{-1}$  is equal to the transpose  $R^T$ . The translation in the world-to-camera format can be found by transforming the camera’s position in the world coordinates using the inverse rotation:

$$t' = -R^T \cdot t \quad (21)$$

Therefore, the LLFF camera-to-world transformation matrix is constructed as:

$$M = \begin{bmatrix} R^T & t' \\ 0 & 1 \end{bmatrix} \quad (22)$$

### C. Demonstration in 2D imaging of NeRF

This section demonstrates that the results of real-time rendering of our datasets in the quickest training NeRF model Instant-NGP [37]. To explore the effect of the complexity of the scene on the NeRF rendering quality, we set the number of images in all datasets to 100 images, all of which were captured by GoPro cameras in 120 HZ, linear imaging mode, with a resolution of 3840x2160 pixels, according to the settings in Sec. IV-A.

The Peak Signal-to-Noise Ratio (PSNR) has been widely recognized as an essential metric for the quantitative evaluation of image quality. Predominantly used in the domain of image compression and reconstruction, its application has further expanded into the emerging realms of computer vision and neural graphics[11]. Higher PSNR values imply superior image fidelity, indicating a closer match to the reference. The foundation of PSNR lies in the Mean Squared Error (MSE), which quantifies the average squared discrepancies between the pixel values of the reference and the examined images. Formally, for an image of size  $MN$ , the MSE is defined as:

$$\text{MSE} = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N [I_{\text{reference}}(i, j) - I_{\text{examined}}(i, j)]^2 \quad (23)$$

Where  $I_{\text{reference}}$  and  $I_{\text{examined}}$  represent the pixel intensities of the reference and examined images, respectively. With the MSE in hand, the PSNR is calculated using:

$$\text{PSNR} = 10 \times \log_{10} \left( \frac{\text{MAX}_I^2}{\text{MSE}} \right) \quad (24)$$

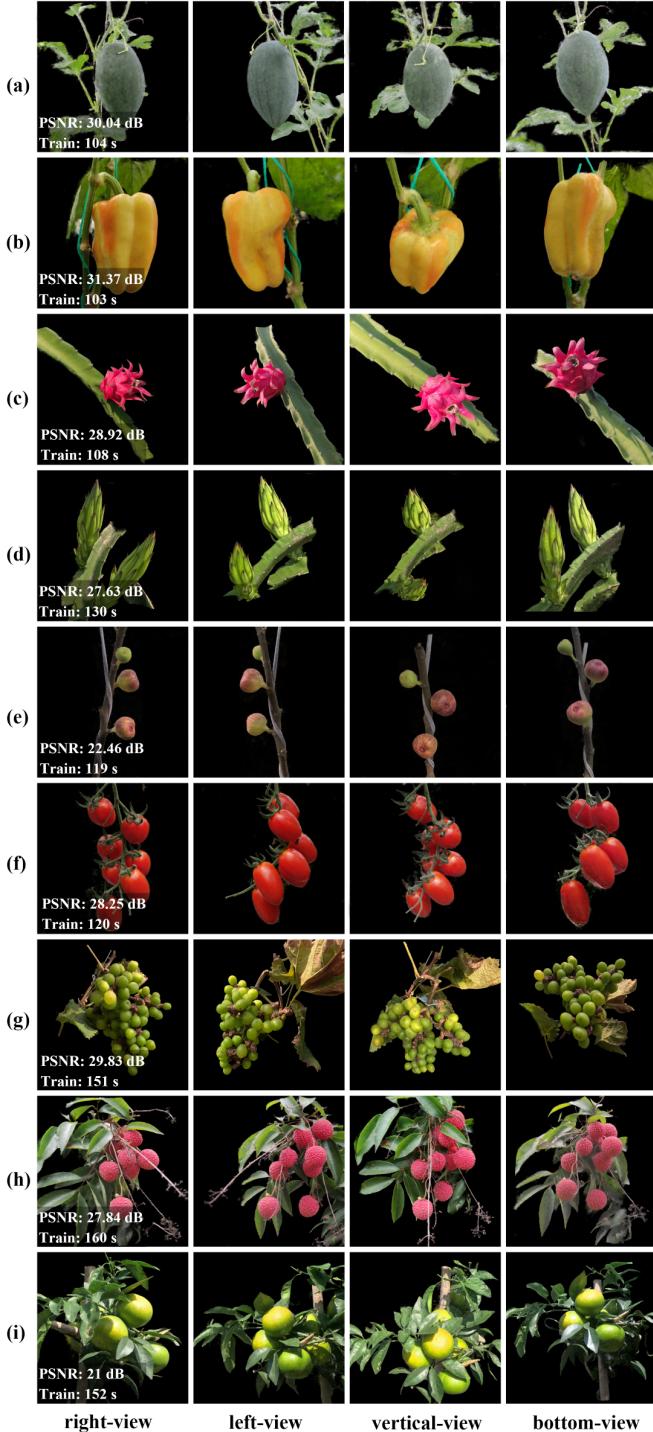


Fig. 8: Novel-view image synthesised results (from different views in sequence.) by using Instant-NGP that trained on  $L_1, L_2, L_3$  datasets.

where  $MAX_I$  signifies the maximum feasible pixel intensity for the image. ( For instance, for a typical 8-bit grayscale image,  $MAX_I$  equals 255. )

To quantitatively evaluate these experimental results, we referenced the original NeRF paper's real-world dataset benchmark which recorded a PSNR of 26.50 dB in their Real Forward-Facing dataset. On this baseline, a dataset with a PSNR close to or higher than 26.50 dB indicates high reconstruction quality, and the opposite indicates that the dataset struggles to converge well.

Fig. 8 shows the training time of our full datasets in Instant-NGP and the PSNR for each scene. To demonstrate the power of NeRF to synthesise new views, we have selected four views other than the training data used for the model, namely right, left, vertical and bottom views.

#### D. Demonstration in 3D imaging of NeRF

This section details the experimental results of the geometry extraction from NeRF based on our plant datasets. First of all, we extract the point clouds and meshes of the plants from Instant-NGP according to the method introduced in Sec.III-B, and demonstrate these results comprehensively in Fig.9. And note that in this experiment, the mesh models generated by Reality capture is used as a reference for the geometric extraction results, because NeRF is supervised by 2D image data without the ground truth of 3D information. (Reality capture is a commercial MVS-based modelling software, which integrates the core methods of MVS as well as comprehensive steps, including photo alignment, feature extraction, feature matching, camera viewpoint calculation, and 3D point cloud reconstruction[24].)

Furthermore, Fig. 10 shows the comparison between the mesh of the plants extracted from Instant-NGP and Instant-NSR using the Marching cubes algorithm in Sec. III-B. The goal of this controlled experiment is to explore the differences in geometric representation between the NeRF model based on the density architecture and the NeRF model based on the SDF architecture.

Dataset Method	Metric	RC Time	Instant-NGP		Instant-NSR	
			Time	PSNR	Time	PSNR
$L_1$	Watermelon	12 min	1.73 min	30.04	12.69 min	29.5
	Bell Pepper	11 min	1.71 min	31.37	12.59 min	24.5
	Pitahaya	12 min	1.80 min	28.92	12.37 min	29.2
$L_2$	Pitahaya flower	15 min	2.16 min	27.63	12.88 min	26.8
	Figs	14 min	1.98 min	22.46	11.39 min	23.5
	Tomatoes	16 min	2 min	28.25	12.28 min	28.4
$L_3$	Grapes	22 min	2.51 min	29.83	13.46 min	25.5
	Litchis	25 min	2.66 min	27.84	13.58 min	27.5
	Oranges	21 min	2.53 min	29.89	13.47 min	20.5

TABLE II: Comparison of time and PSNR metrics for different datasets using Reality Capture, Instant-NGP, and Instant-NSR.

#### E. Discussion

First of all, agricultural scenes, unique in their natural design, present a myriad of challenges when it comes to accurate 3D modeling and analysis.

- **Natural Diversity and Variances:** Unlike manufactured objects or indoor scenes, nature doesn't adhere to a

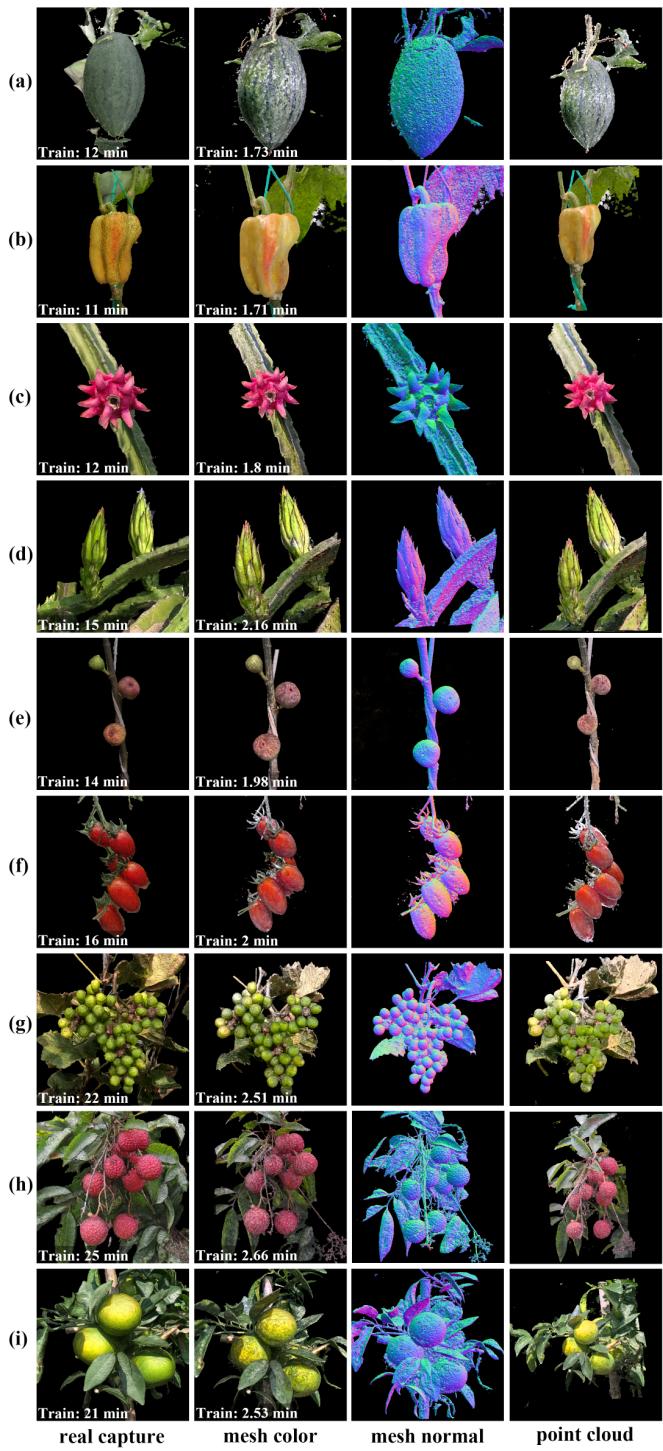


Fig. 9: 3D models extracted from  $L_1, L_2, L_3$  datasets. The leftmost line of data is the real reference views of the plants, followed by the mesh models extracted from Reality-Capture, the normal mapping models, the textured mesh models, and the point clouds extracted from Instant-NGP in sequence.

standardized pattern. This unpredictable variance in terms of plant growth patterns, fruit sizes, leaf orientations, and the positioning of branches makes generalization extremely challenging.

- **Occlusions and Overlapping:** The growth habit of many plants leads to considerable overlapping and occlusion. Fruits hidden behind leaves, branches intertwining, and dense foliage create visual blockages. These complexities are particularly pronounced in agricultural landscapes, where maximized plant growth is often desired for better yields.
- **Surface Complexities:** The intricate surface structures, such as the ruggedness of a bark or the delicate vein patterns on leaves, add another layer of complexity. Modeling these minutiae demands high-resolution data capture and advanced algorithms.

*1) Analysis of rendering results:* In terms of efficiency in processing data, one of the key strengths of the Instant-NGP model lies in its efficiency. Across our datasets  $L_1$ ,  $L_2$ , and  $L_3$ , the model consistently converges in under three minutes. Furthermore, once converged, the model is capable of real-time rendering from any given viewpoint, demonstrating its prowess in generating novel views. This efficiency exceeds that of traditional RGB camera sampling techniques for phenotype collection, demonstrating the potential of NeRF to enable more versatile and efficient ways of observing plant traits from different perspectives.

In terms of the quantitative metric PSNR for image rendering, instant-NGP reaches the highest PSNR of 31.37 dB within two minutes for the three scenes in  $L_1$ . In  $L_2$ , both datasets (d) and (f) can achieve a PSNR of higher than 26.5 dB. In the scenes of Litchis and grapes in  $L_3$ , the training time is on average 50 seconds longer than that in  $L_1$ , but the PSNR also exceeds the baseline of 26.5 dB for all of them. Taking into account the class of the dataset and the quality of the image data, we can see that with simple geometry, instant-NGP can converge very quickly and get a high quality of new view rendering, while complex geometry will cause the network to converge slower.

However, the dataset (e) of figs and (i) of oranges show a very low level of PSNRs: 22.46 dB and 21 dB. Since the PSNR is an average representation of the difference between the rendered image and the ground truth image, we displayed the difference between the full rendered result containing the background and the plant subject and the actual image in Fig.11. In this case **(1) and (2)**, the background in the rendering result is severely defocused, additionally, the background inside differs a lot from the real data, resulting in a low PSNR. On the contrary of **(3)**, the background and the subject are clearly distinguishable and not far from each other, thus giving a high PSNR. From this result, it can be observed that Instant-NGP works well for reconstructing plants in the centre of the scene, but it cannot recover background objects in the distance, which may limit the application of NeRF in large-scale phenotype acquisition.

*2) Analysis of geometry extraction results:* In the comparison of Instant-NGP and Reality Capture, it is not hard to see that Instant-NGP has a faster modelling speed and at the

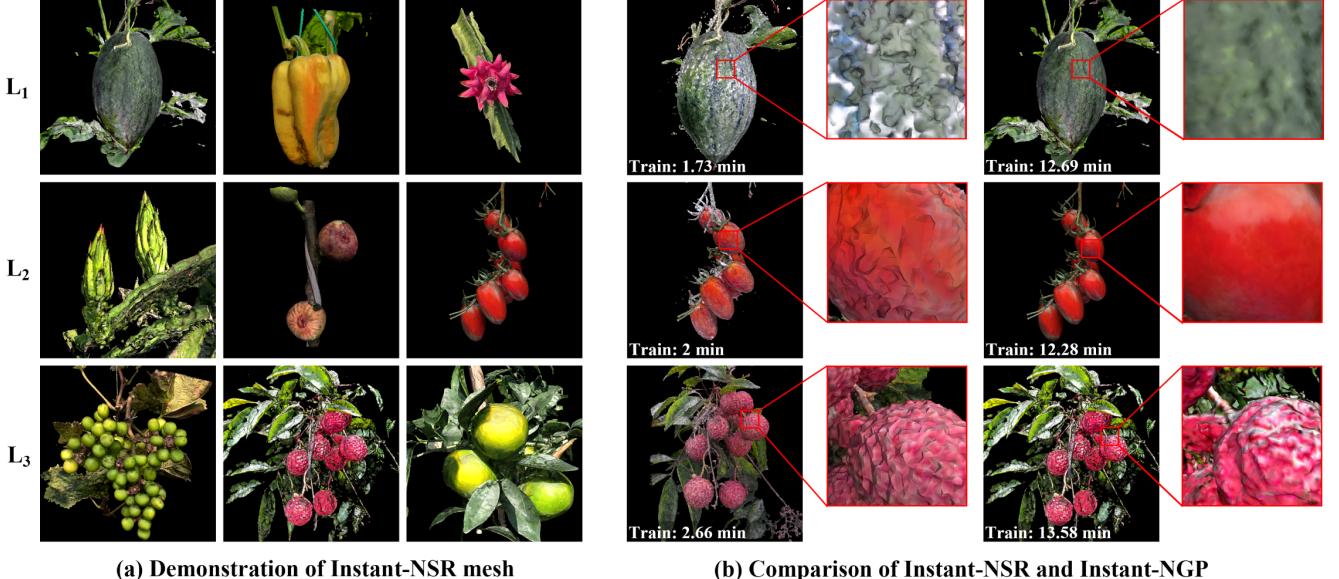


Fig. 10: (a) Demonstration of the meshes obtained via Instant-NSR, (b) Comparison of Instant NGP and Instant NSR in details on the model surface.

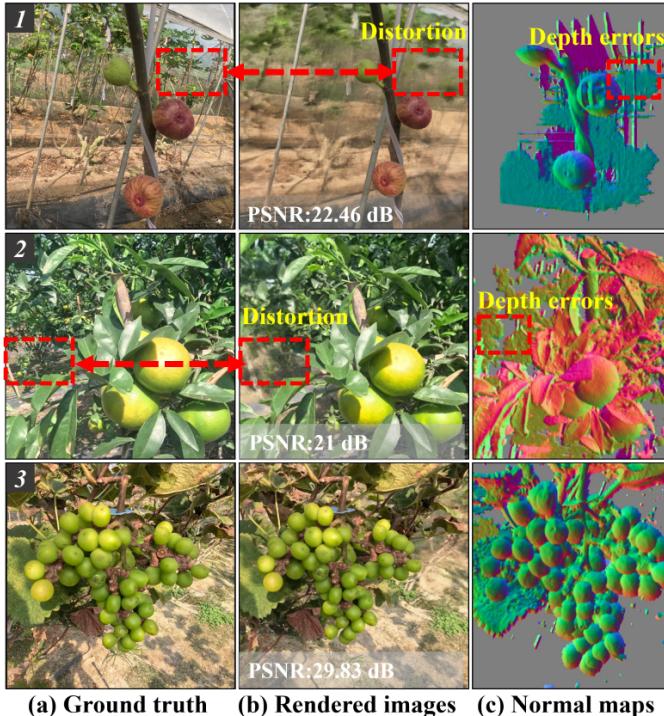


Fig. 11: Analysis of quantitative metric PSNR

meantime, it provides a variety of geometrical representations (point cloud, mesh, texture), and even capable of generating rendered images with arbitrary viewpoints. Compared with the traditional Imaging system, NeRF provides a new pattern of information acquisition, which is not to model the plants directly, but to store volumetric and colour information in the neural network, and then convert the parameters of the

neural network into the multi-source information needed for phenotyping by methods such as volume rendering.

However, the NeRF model based on volume density representation lacks geometric constraints to accurately represent the surface of an object, so when modelling plants with smooth surfaces such as watermelons and bell peppers, their meshes are distorted in surface details. The SDF-based NeRF models are excellent at representing the surface of an object with a smooth and continuous iso-surface, but in scenarios where the bump mapping is extremely complex and the SDF gradient varies drastically, the NeRF models converge very slowly.

## VI. CONCLUSION AND FUTURE WORK

In this study, we investigated a novel neural radiance field-based approach for multi-source phenotypic information acquisition to achieve high-fidelity and high-throughput phenotypic reconstruction of a wide range of plants. In our experiments, Instant-NGP is the key method to accelerate the training of NeRF networks, which can model and infer multiple geometric representations faster than the traditional MVS reconstruction method. Moreover, NeRF can generate realistic viewpoint maps through volume rendering. Further, we comprehensively evaluate the performance of NeRF models with two architectures (volume density and SDF) for extracting object surfaces. The experimental results indicate that the volume density-based NeRF model (e.g., Instant-NGP) is suitable for representing plants with uneven surfaces such as litchi, however, the SDF-based one (e.g., Instant-NRS) is more suitable for reconstructing smooth and continuous plant surfaces such as watermelon and grape.

Future work will focus on enhancing the utility of NeRF by enabling fast and accurate NeRF modelling in the presence of sparse views. Further, we will also investigate how to enhance

the background processing while ensuring the reconstruction effect of the subject, which will facilitate the application of NeRF to large-size plant phenotype acquisition.

## REFERENCES

- [1] R. P. Sishodia, R. L. Ray, and S. K. Singh, "Applications of remote sensing in precision agriculture: A review," *Remote Sensing*, vol. 12, no. 19, p. 3136, 2020.
- [2] L. Fu, F. Gao, J. Wu, R. Li, M. Karkee, and Q. Zhang, "Application of consumer rgb-d cameras for fruit detection and localization in field: A critical review," *Computers and Electronics in Agriculture*, vol. 177, p. 105687, 2020.
- [3] L. Feng, S. Chen, C. Zhang, Y. Zhang, and Y. He, "A comprehensive review on recent applications of unmanned aerial vehicle remote sensing with various sensors for high-throughput plant phenotyping," *Computers and electronics in agriculture*, vol. 182, p. 106033, 2021.
- [4] M. S. M. Asaari, S. Mertens, S. Dhondt, D. Inzé, N. Wuyts, and P. Scheunders, "Analysis of hyperspectral images for detection of drought stress and recovery in maize plants in a high-throughput phenotyping platform," *Computers and Electronics in Agriculture*, vol. 162, pp. 749–758, 2019.
- [5] Z. Li, R. Guo, M. Li, Y. Chen, and G. Li, "A review of computer vision technologies for plant phenotyping," *Computers and Electronics in Agriculture*, vol. 176, p. 105672, 2020.
- [6] R. T. Furbank and M. Tester, "Phenomics—technologies to relieve the phenotyping bottleneck," *Trends in plant science*, vol. 16, no. 12, pp. 635–644, 2011.
- [7] G. Rebetzke, J. Jimenez-Berni, R. Fischer, D. Deery, and D. Smith, "High-throughput phenotyping to enhance the use of crop genetic resources," *Plant Science*, vol. 282, pp. 40–48, 2019.
- [8] Y. Wang, J. Fan, S. Yu, S. Cai, X. Guo, and C. Zhao, "Research advance in phenotype detection robots for agriculture and forestry," *International Journal of Agricultural and Biological Engineering*, vol. 16, no. 1, pp. 14–25, 2023.
- [9] H. Zhou, X. Wang, W. Au, H. Kang, and C. Chen, "Intelligent robots for fruit harvesting: Recent developments and future challenges," *Precision Agriculture*, vol. 23, no. 5, pp. 1856–1907, 2022.
- [10] Y. Zhang and N. Zhang, "Imaging technologies for plant high-throughput phenotyping: a review," *Frontiers of Agricultural Science and Engineering*, vol. 5, no. 4, pp. 406–419, 2018.
- [11] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.
- [12] G. Zhao, R. Yang, X. Jing, H. Zhang, Z. Wu, X. Sun, H. Jiang, R. Li, X. Wei, S. Fountas *et al.*, "Phenotyping of individual apple tree in modern orchard with novel smartphone-based heterogeneous binocular vision and yolov5s," *Computers and Electronics in Agriculture*, vol. 209, p. 107814, 2023.
- [13] S. Kolhar and J. Jagtap, "Plant trait estimation and classification studies in plant phenotyping using machine vision—a review," *Information Processing in Agriculture*, vol. 10, no. 1, pp. 114–135, 2023.
- [14] J. P. Kumar and S. Dominic, "Image based leaf segmentation and counting in rosette plants," *Information processing in agriculture*, vol. 6, no. 2, pp. 233–246, 2019.
- [15] J. Ubbens, M. Cieslak, P. Prusinkiewicz, and I. Stavness, "The use of plant models in deep learning: an application to leaf counting in rosette plants," *Plant methods*, vol. 14, pp. 1–10, 2018.
- [16] H. Kang and C. Chen, "Fast implementation of real-time fruit detection in apple orchards using deep learning," *Computers and Electronics in Agriculture*, vol. 168, p. 105108, 2020.
- [17] M. Jansen, F. Gilmer, B. Biskup, K. A. Nagel, U. Rascher, A. Fischbach, S. Briem, G. Dreissen, S. Tittmann, S. Braun *et al.*, "Simultaneous phenotyping of leaf growth and chlorophyll fluorescence via growscreen fluoro allows detection of stress tolerance in arabidopsis thaliana and other rosette plants," *Functional Plant Biology*, vol. 36, no. 11, pp. 902–914, 2009.
- [18] P. Clauw, F. Coppens, K. De Beuf, S. Dhondt, T. Van Daele, K. Maleux, V. Storme, L. Clement, N. Gonzalez, and D. Inzé, "Leaf responses to mild drought stress in natural variants of arabidopsis," *Plant physiology*, vol. 167, no. 3, pp. 800–816, 2015.
- [19] B. Dellen, H. Scharr, and C. Torras, "Growth signatures of rosette plants from time-lapse video," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 12, no. 6, pp. 1470–1478, 2015.
- [20] J. F. Humplík, D. Lazár, A. Husičková, and L. Spíchal, "Automated phenotyping of plant shoots using imaging methods for analysis of plant stress responses—a review," *Plant methods*, vol. 11, no. 1, pp. 1–10, 2015.
- [21] S. Paulus, "Measuring crops in 3d: using geometry for plant phenotyping," *Plant methods*, vol. 15, no. 1, pp. 1–13, 2019.
- [22] H. Kang, Y. Zang, X. Wang, and Y. Chen, "Uncertainty-driven spiral trajectory for robotic peg-in-hole assembly," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6661–6668, 2022.
- [23] R. Guo, J. Xie, J. Zhu, R. Cheng, Y. Zhang, X. Zhang, X. Gong, R. Zhang, H. Wang, and F. Meng, "Improved 3d point cloud segmentation for accurate phenotypic analysis of cabbage plants using deep learning and clustering algorithms," *Computers and Electronics in Agriculture*, vol. 211, p. 108014, 2023.
- [24] S. Wu, W. Wen, Y. Wang, J. Fan, C. Wang, W. Gou, and X. Guo, "Mvspheno: a portable and low-cost phenotyping platform for maize shoots using multiview stereo 3d reconstruction," *Plant Phenomics*, vol. 2020, 2020.
- [25] H. Kang, X. Wang, and C. Chen, "Accurate fruit localisation for robotic harvesting using high resolution lidar-camera fusion," *arXiv preprint arXiv:2205.00404*, 2022.
- [26] H. Kang and X. Wang, "Semantic segmentation of fruits on multi-sensor fused data in natural orchards," *Computers and Electronics in Agriculture*, vol. 204, p. 107569, 2023.
- [27] E. Kok, X. Wang, and C. Chen, "Obscured tree branches segmentation and 3d reconstruction using deep learning and geometrical constraints," *Computers and Electronics in Agriculture*, vol. 210, p. 107884, 2023.
- [28] T. Yang, J. Ye, S. Zhou, A. Xu, and J. Yin, "3d reconstruction method for tree seedlings based on point cloud self-registration," *Computers and Electronics in Agriculture*, vol. 200, p. 107210, 2022.
- [29] T. Samavati and M. Soryani, "Deep learning-based 3d reconstruction: A survey," *Artificial Intelligence Review*, pp. 1–45, 2023.
- [30] Z. Chen and H. Zhang, "Learning implicit fields for generative shape modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5939–5948.
- [31] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4460–4470.
- [32] J. J. Park, P. Florene, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 165–174.
- [33] K. J. V. H. BP, "Ray tracing volume densities acm siggraph comput," *Graph*, vol. 18, no. 3, p. 165, 1984.
- [34] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," in *Seminal graphics: pioneering efforts that shaped the field*, 1998, pp. 347–353.
- [35] P. V. Sander, J. Snyder, S. J. Gortler, and H. Hoppe, "Texture mapping progressive meshes," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001, pp. 409–416.
- [36] B. Lévy, S. Petitjean, N. Ray, and J. Maillot, "Least squares conformal maps for automatic texture atlas generation," in *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, 2023, pp. 193–202.
- [37] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics (ToG)*, vol. 41, no. 4, pp. 1–15, 2022.
- [38] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," *arXiv preprint arXiv:2106.10689*, 2021.
- [39] F. Zhao, Y. Jiang, K. Yao, J. Zhang, L. Wang, H. Dai, Y. Zhong, Y. Zhang, M. Wu, L. Xu *et al.*, "Human performance modeling and rendering via neural animated mesh," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 6, pp. 1–17, 2022.
- [40] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.
- [41] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar, "Local light field fusion: Practical view synthesis with prescriptive sampling guidelines," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–14, 2019.