International Journal for Housing Science and Its Applications
Publish August 5, 2025. Volume 46, Issue 3 Pages 4175-4188

https://doi.org/10.70517/ijhsa463357

Research on three-dimensional target modeling method for multi-view video based on depth image analysis

Yijun Liu[1,2,*] and Junming Zuo[3]
[1] Faculty of Humanities and Arts, Macau University of Science and Technology, Macau, 00853, China
[2] Creative Design Institute, Dongguan City University, Dongguan 523419, China
[3] School of Digital Media and Design, Neusoft Institute Guangdong, Foushan, Guangdong, 528225, China
Corresponding authors: (e-mail: liuyijun@dgcu.edu.cn).

Abstract Machine vision 3D modeling technology has received more and more attention due to its great commercial application value, and has become a fundamental technology in multiple fields. The article proposes a multi-view video 3D target modeling method combining SFM and NeRF network. The growth of seedling plants is taken as the research object, and the multi-view video of seedling plants is obtained by recording video. Python development platform was used to obtain the estimation of image position through COLMAP software, on the basis of which the image position was inputted into NeRF network to realize the 3D target modeling of multi-view video, and the similarity matrix and KNN algorithm were used to cloud the 3D model, and statistical filtering was introduced to remove the outliers of the point cloud. The combination of SFM and NeRF can significantly enhance the accuracy of multi-view video 3D target modeling, and the overall reconstruction efficiency and quality are high. Therefore, actively exploring the application of deep learning techniques in multi-view 3D target modeling can further promote the development of 3D target modeling technology.

Index Terms SFM, NeRF network, Python platform, model point-clouding, statistical filtering, 3D modeling

I. Introduction

In the last decade, the rapid development of computer software and hardware technology has made great progress both in terms of computational efficiency and memory capacity, such as multi-core central processors and image processors with parallel computing capability [1]-[3]. On the other hand, the theories and methods of computer vision have been developed even further, in which one of the important tasks of computer vision is to understand the three-dimensional properties of an object, including three-dimensional geometrical shapes and camera viewpoints [4]-[6]. Solutions to this task are crucial for applications that require interaction between the real and virtual worlds, such as in the fields of autonomous driving and augmented reality [7], [8]. On such a basis, human beings have higher expectations about the ability of machines to perceive objects and process information.

3D target modeling refers to the acquisition of raw information in the environment through sensors such as cameras and radars, and then 3D target modeling algorithms are used to model one or more target objects in the environment in three dimensions to create a data model suitable for computer representation and processing [9]. The three-dimensional model contains a more complete digital information of the object in three-dimensional space, in which the objects represented are in motion or relatively static state, is the key technology of computer perception and expression of the objective environment [10]-[12]. Among them, the information of distance, 3D shape, and viewpoint of an object is a very important one with both academic and application values [13], [14]. Usually, from a single 2D view only the information of a certain angle of the target object can be acquired, and the 3D spatial structure information of the target object cannot be acquired [15], [16]. Therefore, it is necessary to obtain multiple 2D views with different angles to accomplish high-precision 3D reconstruction [17]. Based on this, the question of which equipment should be used to capture images to achieve its demandability, and also how to realize 3D reconstruction based on multiple video image sequences with multiple viewpoints is of great significance in realizing fast and accurate 3D anthropometric measurements.

In order to solve the drawbacks of high cost, long time and low accuracy of traditional methods in 3D modeling in the past, then combined with advanced deep learning technology to form a multi-view video 3D target modeling technology, which is applicable in many fields. In this paper, on the basis of combing the motion recovery structure algorithm and NeRF network, a variety of seedling plants are taken as research objects. Its multi-view video is recorded by cell phone, and COLMAP software in Python platform is used to extract frames from the video, so as to obtain the position estimation results of multi-view images of seedling plants. It was input into NeRF network for

multi-view 3D target reconstruction, and model outliers were removed by model point-clouding and statistical filtering. Comparative ablation experiments were carried out to address the effectiveness of the model, and its reconstruction efficiency and quality performance were explored.

II. Relevant theoretical and technical foundations

Realistic 3D reconstruction of objective environments using computers has always been one of the popular areas studied in computer vision, robotics, and computer graphics. With the tremendous development of the information industry, 3D reconstruction technology is not only applied to traditional places, such as robot navigation, visual surveillance, and construction manufacturing, but also has important application values in such information fields as human-computer interaction, digital entertainment, e-commerce, visual communication, virtual reality, etc. It is conceivable that a 3D reconstruction with both geometric accuracy and texture realism can be realized. It is conceivable that a 3D scene model with both geometric accuracy and texture realism will have a very wide range of applications.

II. A. Camera Model and Calibration Methods

II. A. 1) Camera model

The camera model is a simplification of the optical imaging model, which is divided into two types: linear model and nonlinear model. The more commonly used linear model is the pinhole imaging model, which is an ideal state model where the object and image are in a similar triangular relationship, ignoring the geometric distortion of the camera. The nonlinear model, on the other hand, takes into account that the camera may have problems such as distortion, and is a model of the actual state [18].

The actual camera model due to its optical system in the processing and assembly of the error may occur, then the captured image will appear picture distortion distortion, so that there will be a gap between the actual image formed by the camera and the image formed in the ideal situation. The actual camera model is shown in Figure 1, where $m_i(x_i, y_i)$ denotes the ideal projected point coordinates and $m_j(x_j, y_j)$ denotes the actual projected point coordinates. Then the distortion model of the lens can be expressed as:

$$x_i = x_j + \sigma_x, y_i = y_j + \sigma_y \tag{1}$$

where σ_x and σ_y denote nonlinear distortion values.

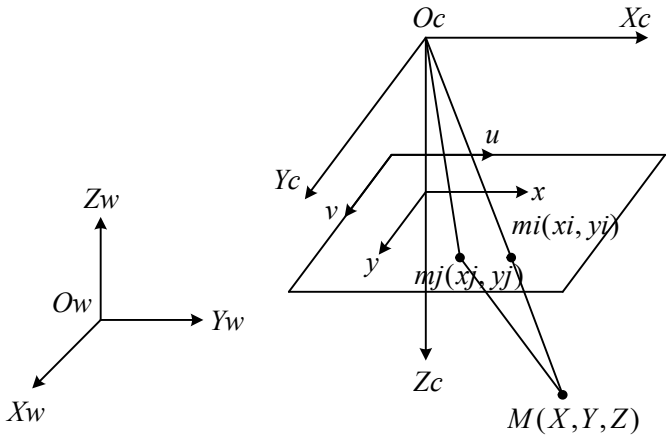

Figure 1: The actual camera model

Camera aberrations generally contain three kinds of radial aberrations, centrifugal aberrations and thin prism aberrations, of which thin prism aberrations are caused by lens design and processing installation errors, usually negligible. Radial aberrations in the lens pixels are centered on the center of the aberration, the deviation of the mirror position, the closer to the edge of the screen, the greater the deviation, is due to the shape of the lens defects.

The model of radial distortion can be expressed as:

$$\begin{cases} \sigma_x = x_j(k_1 r^2 + k_1 r^4 + \cdots) \\ \sigma_y = y_j(k_1 r^2 + k_1 r^4 + \cdots) \end{cases} \tag{2}$$

where k_1, k_2 etc. denote the radial aberration coefficients, $r = \sqrt{x_j^2 + y_j^2}$.

Centrifugal aberrations, which are caused by the optical center of the lens not coinciding with the geometric center of the image plane, contain radial aberrations and can be expressed as:

$$\begin{cases} \sigma_x = p_1 x_j (x_j^2 + 3y_j^2) + 2p_2 x_j x_j + o[(x_j, y_j)^4] \\ \sigma_y = p_2 x_j (3x_j^2 + y_j^2) + 2p_1 x_j x_j + o[(x_j, y_j)^4] \end{cases} \tag{3}$$

where p_1, p_2 denote the tangential distortion coefficients.

II. A. 2) Camera calibration

The basic assumption of multi-view 3D reconstruction is that the camera calibration matrix is known, and there are usually two ways to get the calibration matrix. One is to use the EXIF information in the image to generate the calibration matrix, this method is simple but cannot guarantee the accuracy. The other is to calibrate the camera first, and then use the camera to capture the image, this method can get higher accuracy. Therefore, the calibration of the camera is often performed before 3D reconstruction with non-incremental methods [19]. From the camera model, the mapping relationship between the 3D points in the world coordinate system and the pixel points in the image plane is:

$$x = K[R \mid t]X \tag{4}$$

If we denote the coordinates of x by $(u, v, 1)$ and the coordinates of X by (X, Y, Z, I), and without loss of generality, we assume that the Z coordinate of X is 0, then the above equation can be written as:

$$\begin{bmatrix} s \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = K \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{5}$$

where s is the scale factor, r_i is the i th column of the rotation matrix R , K is the calibration matrix, and t is the translation vector. Then the relationship between the spatial point X and the image point x can be expressed by a single should, i.e.:

$$sx = HX, H = K[r_1 \quad r_2 \quad t] \tag{6}$$

Given a number of sets of points corresponding to X and x, the single response matrix H can be found using the DLT algorithm. It can be obtained from the above equation:

$$[h_1 \quad h_2 \quad h_3] = \lambda K [r_1 \quad r_2 \quad t] \tag{7}$$

where h_i is the i th column of H and λ is the scale factor. Using the orthogonality of r_i with r_2 can be obtained:

$$\begin{aligned} h_1^T K^{-T} K^{-1} h_2 &= 0 \\ h_1^T K^{-T} K^{-1} h_1 &= h_2^T K^{-T} K^{-1} h_2 \end{aligned} \tag{8}$$

For a general finite camera, the calibration matrix K has six degrees of freedom, and two constraints can be obtained by photographing the calibration plate from one direction, so acquiring the calibration image from three different directions theoretically solves for the calibration matrix K .

II. B.Motion Recovery Structure Algorithm

The main idea of the Structure for Motion Recovery (SFM) algorithm is the process of taking a sequence of multiview images of a target object in all different viewpoints using changes in camera position, and then solving for changes in camera pose to generate a 3D point cloud of the object. When made complementary to the multi-view stereo vision matching algorithm, it is possible to construct detailed 3D models from these photo collections in a fully automated way. The implementation of the Motion Recovery Structure algorithm consists of two main steps, i.e., two-view point cloud generation and multi-view point cloud fusion. The final structure of the target, i.e., the fused 3D point cloud, is recovered by the multi-view point cloud fusion, and it is crucial to optimize the resulting 3D point cloud by beam method leveling.

II. B. 1) Two View Point Cloud Generation

(1) Image feature extraction and matching. In the image-based 3D reconstruction process, the first and most important step is the extraction and matching of 2D image features, because the subsequent reconstruction process is based on the obtained image matching pairs, if the feature point matching pairs are inaccurate on the subsequent reconstruction will have a great impact on the reconstruction, which may lead to the reconstruction results and the actual error is larger.The SIFT features are mainly used in the Gaussian difference scale space to extract the extreme points, and then filter the extreme points and calculate the feature point descriptors. SIFT features mainly use the Gaussian difference scale space to extract the extreme points, and then filter the extreme points and calculate the feature point descriptors.SIFT features are invariant to image scale scaling, rotation and lighting changes, and also have a certain degree of stability to noise.

(2) Camera pose estimation. After ending the feature matching, the basis matrix and essence matrix can be estimated according to the image feature matching relationship. Then the random sampling consistent algorithm removes the wrong matching to obtain the high-precision initial matching homonymous points, and then use these high-precision homonymous points for the estimation of camera pose.

(3) 3D point cloud generation. After the camera attitude estimation, the outer parameter matrix of the camera is obtained. Using the triangulation measurement method, all the 3D point coordinates of the target can be obtained by estimating the known camera parameters and the tracking sequence of feature points.

II. B. 2) Multi-view point cloud fusion

Since the matching points of the two images are usually the result of the projection of only a part of the reconstructed object, only a part of the target can be reconstructed at best, so new views need to be added to gradually enrich the rest of the reconstructed object. On top of the

already reconstructed initial scene, a third view is added, and let the newly added projection point $m_3 = (u_3, v_3, 1)$ and the already reconstructed m_1, m_2 be the projection points at the same point in the space, then we have:

$$X_{c3} m_3 = PM_3 = P\begin{pmatrix} R_{13} & T_{13} \\ 0 & 1 \end{pmatrix} M_1 = \begin{bmatrix} P_{11}^n & P_{12}^n & P_{13}^n & P_{14}^n \\ P_{21}^n & P_{22}^n & P_{23}^n & P_{24}^n \\ P_{31}^n & P_{32}^n & P_{33}^n & P_{34}^n \end{bmatrix} \begin{pmatrix} X_{t1} \\ Y_{t1} \\ Z_{t1} \\ 1 \end{pmatrix} \tag{9}$$

The equations M_1 and m_3 are known, and the unknowns are X_{c3} and the 12 elements in the coefficient matrix, which can be obtained by substituting the equation obtained from X_{c3} into the first two lines according to the third line:

$$\begin{cases} u_3 X_1 P''_{31} + u_3 Y_1 P''_{32} + u_3 Z_1 P''_{33} + u_3 P''_{34} - X_1 P''_{11} - Y_1 P''_{12} - Z_1 P''_{13} - P''_{14} = 0 \\ v_3 X_1 P''_{31} + v_3 Y_1 P''_{32} + v_3 Z_1 P''_{33} + v_3 P''_{34} - X_1 P''_{11} - Y_1 P''_{12} - Z_1 P''_{13} - P''_{14} = 0 \end{cases} \tag{10}$$

Writing the equation in matrix form gives:

$$AP = 0$$

$$A = \begin{bmatrix} -X_1 & -Y_1 & -Z_1 & 1 & 0 & 0 & 0 & 0 & u_3 X_1 & u_3 Y_1 & u_3 Z_1 & u_3 \\ 0 & 0 & 0 & 0 & -X_1 & -Y_1 & -Z_1 & 1 & v_3 X_1 & v_3 Y_1 & v_3 Z_1 & v_3 \end{bmatrix} \tag{11}$$

$$P = \begin{bmatrix} P_{11}^n & P_{12}^n & P_{13}^n & P_{14}^n & P_{21}^n & P_{22}^n & P_{23}^n & P_{24}^n & P_{31}^n & P_{32}^n & P_{33}^n & P_{34}^n \end{bmatrix}$$

From the system of equations, there are 12 unknowns for P and 11 free variables, which means that at least 11 equations are needed to solve, then 5.5 pairs of matching points are needed to solve for the vector P, and in the experiments, 6 pairs of matching points were chosen. Once the projection matrix of the third view has been solved, the world coordinates of the matching points in views two and three that have not yet been reconstructed can be computed based on the two-view 3D reconstruction method.

II. B. 3) Beam method leveling optimization

In the process of recovering 3D coordinates, the results of using the derived projection matrix to reconstruct the reconstructed spatial points in back-projection will be biased due to the large amount of data and the interference of noise points. According to the principle of least squares, it is desired to find such a set of estimates of the projection matrix and the spatial points that minimize the sum of squares of the distances between the backprojected points and the feature points [20]. Let the counter-projection point of the space point M be $\tilde{m}$ and the

corresponding feature point of M be m, then find the projection matrix $\tilde{P}$ and the space point $\tilde{M}$ such that the distance between the two points is minimized as:

$$\min \sum d(\tilde{m}, m)^2 \tag{12}$$

The above process is the principle of beam leveling at a single point. This method of adjusting camera parameters and three-dimensional structure to achieve uniform error distribution and "overall optimization" is called beam leveling (BA), which is a nonlinear least squares problem. The beam leveling is essentially a nonlinear least squares optimal solution problem, and this paper adopts the gradient descent method to optimize the solution.

The gradient is a vector that describes the direction in which the function value at a point on the function f grows fastest, and conversely, the negative direction of the gradient is the direction in which the function value decreases fastest. Along the positive and negative directions of the gradient, one gets the extreme values and the extreme values, respectively. Taking $f(X)-Y$ to denote the error value η, we have $\eta = f(X)-Y$, then there must exist a solution such that $s(X) = \sum \eta^2$, and the value of the gradient of $S(X)$ can be expressed as:

$$\begin{aligned} \frac{\partial}{\partial X} S(X) &= \frac{\partial}{\partial X}((f(X)-Y)^T (f(X)-Y)) \\ &= 2(\frac{\partial f(X)}{\partial X})^T (f(X)-Y) \\ &= 2J^T \eta \end{aligned} \tag{13}$$

where J is the Jacobi determinant of the function expressed as $J = \frac{\partial f(X)}{\partial X}$, in practice, the threshold is often used to determine whether the requirements are met, so the condition can be modified to:

$$x \leftarrow x - \lambda J^T \eta \tag{14}$$

Choosing the value of λ depending on the situation is called gradient descent.

II. C.Neural Radiation Field Basics

II. C. 1) Neural radiation fields

Neural Radiation Field (NeRF) is a continuous implicit representation of a 3D scene, NeRF uses the absorption and emission model commonly used in body rendering [21]. That is, each point in the scene is set as a light source that not only absorbs light, but also emits light itself, incorporating all the information about the scene's geometry, materials, and lighting, etc. NeRF expresses the entire scene as an implicit function using a fully-connected multilayer perceptron (MLP), whose inputs are the position p in 3D space with the viewpoint direction d, and the output is the point's volume density σ with RGB color c, i.e., $F(p,d) = (\sigma, c)$.

Where $p = (x, y, z)$ is the positional coordinates of the point in 3D space, $d = (\theta, \varphi)$ is the 2D camera direction, $c = (R, G, B)$ is the color of the point, and since the color may not be the same in different viewpoints, the value of the color is related to the viewing direction d and spatial position p, σ is the point's volume density of the point, which can also be interpreted as the probability that a ray of light stops at this point, and this item is only related to the spatial position p.

For a trained model. The NeRF employs a body-rendering approach to synthesize the image in the new viewpoint. Specifically. A ray of light is emitted from each pixel in the target picture, this ray will pass through the whole scene and integrate the color and density along the path to finally get the color of the pixel point, i.e.:

$$C(r) = \int_{t_n}^{t_f} T(t)\sigma(r(t))c(r(t), d)dt \tag{15}$$

where $T(t)$ is the cumulative transmittance, which represents the probability that light is not absorbed from t_n to t, and is calculated by the equation $T(t) = e^{\int_{t_n}^{t} -\sigma(s)ds}$

Since continuous integrals are difficult to compute, the NeRF, and the vast majority of subsequent work, is approximated by discrete sampling.

First, $[t_n, t_f]$ is divided into N equal-length intervals, and then a sample is randomly drawn within each interval, and the distance between neighboring samples is denoted by $\delta_i = t_{i+1} - t_i$. The rendering equation can be expressed as:

$$\hat{C}(r)=\sum_{i=1}^{N}T_i(1-e^{-\sigma_i\delta_i})c_i \tag{16}$$

where $T_i=e^{-\sum_{j=1}^{i-1}\sigma_i\delta_i}$ is the approximate cumulative transmittance. For each pixel, the light emitted by the pixel is summed up by the discrete sampling method in the above equation to get the desired color of the pixel, and finally all the pixels are put together to get the final rendered image. The difference between the predicted pixel color $\hat{c}$ and the real pixel color $C_{g,t}$, $L=\sum_{r\in R}\|\hat{C}(r)-C_{g,t}(r)\|_2^2$ is used as the loss function to train the network.

Since neural networks are more inclined to learn low-frequency signals in space, while real natural scenes often have complex geometries and a lot of color jumps, it is difficult to fit a large number of high-frequency signals in space by using the position coordinates directly as inputs. Therefore, NeRF adopts a position coding approach. Mapping the low-frequency position information into a higher dimensional space, i.e:

$$\begin{aligned}\gamma(p)=&(\sin(2^0\pi p)),(\cos(2^0\pi p)),(\sin(2^1\pi p)),\\&(\cos(2^1\pi p)),\cdots,(\sin(2^{N-1}\pi p)),(\cos(2^{N-1}\pi p)).\end{aligned} \tag{17}$$

where N is a hyperparameter set to N=10 in the encoding of position p and $N=4$ in the encoding of direction d in the original NeRF.

II. C. 2) NeRF Scene Representation

NeRF is based on the principle of light reversibility and camera imaging, and uses 2D images to fit the rendering of mapping functions of 3D spatial point positions, orientations, colors, and body densities within the camera's field of view through a multi-layer perceptron neural network.NeRF training is carried out by minimizing the error between the rendering result and the real image, which makes the rendered image closer and closer to the real image.

Figure 2 shows the NeRF scene representation. The 5D coordinates are first sampled along the camera rays (a), the relevant positions are fed into the multilayer perceptron to generate the volume density and color (b), and finally the image combination is achieved by the volume rendering technique (c). Where the rendering function can be optimized and scene representation by minimizing the residual values between the synthesized and real images (d).

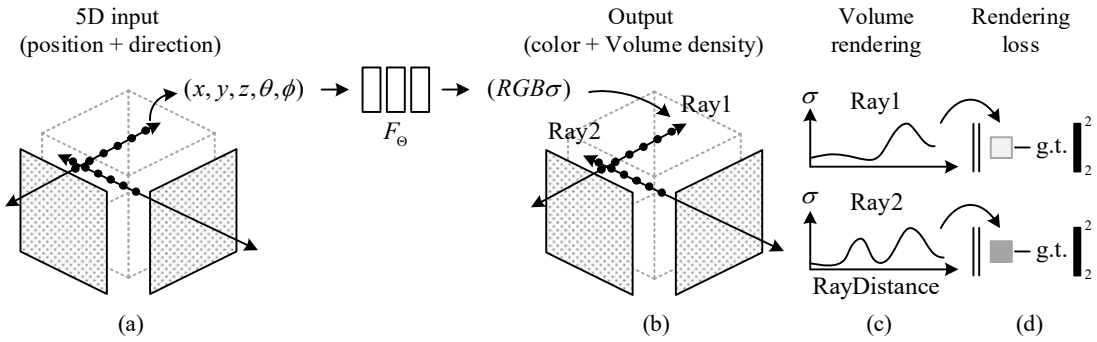

Figure 2: NeRF scene representation method

The NeRF network structure is a fully connected neural network, which centers on its input, processing and output processes. First of all, the input data of NeRF includes spatial position (x,y,z) and observation direction (θ,φ), which are positionally encoded before they are fed into the network in order to improve the resolution and expressiveness of the rendering. Positional encoding typically uses the sin and cos functions to encode the input coordinates so that high-frequency details can be captured. In the processing stage, the input data is propagated forward through a network. The network structure contains multiple fully connected layers that are connected together by nonlinear activation functions such as the linear rectifier function (ReLU). During processing, the network learns how to map from the input 5D coordinates to color and body density. To enhance the representation of coordinate information in the network, spatial coordinate information is sometimes re-input at a point in the network structure. In the output phase, the network outputs color and body density information for each sampling point. The color information is represented in RGB format, while the body density represents the opacity or presence probability of the point in the scene.

II. D.Python Development Platform

PyCharm, as an IDE tool based on the Python programming language, aims to provide Python developers with an efficient and convenient solution for code writing and management. In terms of functional architecture, PyCharm's interface is well-designed and consists of four core modules, namely, tool menu bar, code editor, project directory structure and information display area, which bring comprehensive programming support to users. This includes, but is not limited to, code completion and optimization suggestions, dynamic code analysis, convenient project management and advanced debugging, making users significantly more efficient in the process of Python code compilation and maintenance.

(1) Coding assistance environment: PyCharm provides a code editor with automatic code completion and code fragment management, it also supports code block folding and window splitting operations, these features can significantly improve the efficiency and convenience of the user programming tasks.

(2) Code Analysis. The code analysis function enables users to perform code editing, error highlighting and fast code completion, aiming to provide an optimized coding experience.

(3) Project Code Navigation. In terms of project code navigation, PyCharm can easily enable users to jump from one project's documents to another project, with its unique declaration or method to penetrate the hierarchy of class structure. In addition, the IDE provides a series of shortcut keys, which greatly optimizes users' coding performance.

(4) Python refactoring: PyCharm, as a highly efficient IDE, provides users with the convenience of editing, renaming, moving files, and backtracking within the project file, which greatly simplifies the process of code refactoring.

(5) Code debugging. PyCharm integrated a powerful code debugging capabilities, developers can set multiple breakpoints in the program code to achieve step-by-step debugging. In this process, developers can also use the information display area to monitor and check variable properties and other relevant information in real time.

In this paper, based on Python development platform, SFM and NeRF are combined to construct a 3D target reconstruction model for multiview video, and realize 3D target reconstruction for multiview video.

III. NeRF-based three-dimensional target modeling

In recent years, with the wide application of deep learning techniques, high-quality and low-cost 3D reconstruction techniques have been rapidly developed, and researchers have pushed forward the field of 3D reconstruction from various directions. Among them, implicit neural representation, which implicitly represents object shapes as decision boundaries in 3D space, has attracted the attention of more and more researchers by virtue of its ability to represent continuous shapes and generate 3D objects at arbitrary resolution.

III. A. Data Acquisition and Preprocessing

III. A. 1) Experimental design and data acquisition

This experiment was conducted from March to August 2024 in a key laboratory of N University. Seedling potted plants of pepper, tomato, strawberry and greens were selected as experimental subjects, and the plants were photographed in a phenotyping platform with no wind and stable ambient light. The main body of the phenotyping platform consisted of a rotating stage, a background plate and a tripod, the background plate was placed on the opposite side of the tripod, the target plant was placed on the rotating stage, and the tripod was 50 cm away from the center of the rotating stage, the height of the tripod and the angle of the camera were adjusted according to the height of the plant, and the lens of the camera was pointed at the center of the target plant, and the rotating stage rotated to drive the plant to rotate slowly and uniformly, and the rotating stage rotated at a speed of 7.0 °/s, in order to obtain continuous and stable video streams. The rotary table rotated at a speed of 7.0°/s to obtain a continuous and stable video stream. Fix the cell phone camera and connect it to the computer, and use the written Python program to call the camera through the Android SDK to capture the video of the seedling potted plant. After the shooting was completed, the acquired video was accurately processed using a video frame extraction program based on the Python language to extract image sequences with multiple viewpoints, providing high-quality image information for subsequent data analysis and processing.

III. A. 2) Corresponding Position Acquisition for Multi-view Images

The camera position of the image needs to be input in 3D modeling, so it is necessary to use the motion recovery structure algorithm to obtain the relevant information. At present, the main software based on the motion recovery structure algorithm for 3D reconstruction to obtain the relevant position information includes VisualSFM, COLMAP, AgiSoft PhotoScan and so on. In this paper, we use COLMAP software for the estimation of image position acquisition, which is an open-source computer vision tool for tasks such as 3D reconstruction, camera localization

and SLAM. The general process of utilizing COLMAP to acquire the bit position corresponding to a multi-view image is as follows:

(1) Image import. Images of seedling plants taken from multiple viewpoints are imported into COLMAP.

(2) Feature extraction and matching.COLMAP uses the feature points in the image to calculate the camera position. First, it will extract feature points by obtaining each image through algorithms such as SIFT, SURF, etc., and then determine the correspondence between images by matching these feature points, and the process will establish feature point matching between each image pair. During this period, the total number of feature matching points obtained by the software for the relevant seedling plant pulling stage is 85,724.

(3) Initialization of position. Based on the matched feature points, COLMAP tries to initialize the camera's bit position. Here the estimated initial camera position, orientation and internal and external parameters are obtained.

(4) Incremental SFM: COLMAP uses an incremental motion recovery structure method to optimize the camera pose and 3D point cloud step by step. In this process, the internal and external camera parameters and 3D point locations are gradually optimized to maximally fit the observed image.

(5) BA optimization. A global beam leveling method is performed to improve the overall accuracy by optimizing the camera parameters and the position of the 3D point cloud. This step is a global optimization to ensure that the 3D model is consistent with all images.

(6) Post-processing as well as exporting the results. Finally, COLMAP may perform some post-processing steps, such as removing outliers, performing color consistency correction, etc., to further improve the quality of the reconstruction. And based on this, the result data such as three-point cloud and camera parameters are exported.

After getting the COLMAP position matching data, the position information of each image should be formatted and converted to LLFF format for easy reading by NeRF model.

III. B. NeRF-based 3D reconstruction

III. B. 1) Three-dimensional reconstruction of the network structure

In order to realize the 3D target reconstruction of multi-view video, this paper adopts the NeRF-based Instant-NGP algorithm for 3D reconstruction, and the specific flow is shown in Fig. 3. Firstly, COLMAP software in Python development platform is utilized to process the image sequences to estimate the camera pose, and then the 5D input information is encoded using multi-resolution hash coding and spherical harmonic function joint coding method, which is fitted using MLP to generate the neural radiation field. Then the 3D model is obtained and the model is point-clouded based on the depth information generated from the near point of the light as well as the far point of the light, and the similarity matrix and KNN algorithm are utilized for point-cloud filtering, and finally combined with statistical filtering to remove the noisy point cloud.

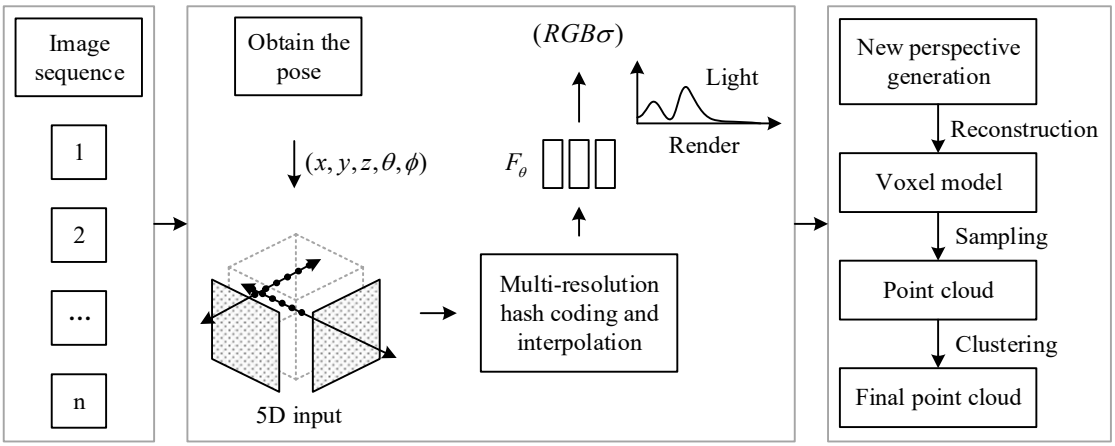

Figure 3: Three-Dimensional Reconstruction Based on NeRF

In order to maintain a low computational cost while preserving details during training, 3D multi-resolution voxel hash coding is used to divide the 3D space into stereo grids of different resolutions. Instead of storing the density value of each voxel, this algorithm utilizes the surrounding 8 grids for interpolation, which increases the speed of network training. In addition, L2 loss is used in order to present the color of seedling plants more realistically. This multi-resolution voxel hash coding method effectively improves the efficiency of model training and the accuracy of results. To wit:

$$L_2 = \sum_{r \in R} \left[\| \hat{C}_c(t) - C(r) \|_2^2 + \| \hat{C}_f(t) - C(r) \|_2^2 \right] \tag{18}$$

where $\hat{C}_c(t)$ is the coarse resolution grid output and $\hat{C}_f(t)$ is the fine resolution grid output.

III. B. 2) Model Point Clouding and Filtering

(1) Model point cloud reconstruction

Spectral methods generally utilize a priori knowledge of model geometry for unsupervised learning when dealing with 3D model correspondences constructed with triangular meshes; however, point clouds lack direct connectivity information, and their geometric properties cannot be directly applied for unsupervised training. In this paper, we address this issue by introducing a point cloud reconstruction module that allows learning the features of the point cloud in the absence of predefined real correspondences, and unsupervised acquisition of correspondences computed through the similarity matrix and using the K-nearest neighbor algorithm.

First, the similarity matrix $\hat{P}$ with the original input model M, N is inputted into the K nearest neighbor feature search module, and the K nearest neighbor algorithm calculates the k points on the model N, which have the highest probability of corresponding to the i th point on M, and will be denoted as $\{N_j\} j \in [1,k]$.

Then, based on the model point similarity matrix $\hat{P}$ that we have already computed, and at the same time, combining with the previously obtained point set $\{N_j\} j \in [1,k]$ to compute the corresponding matrix $\{S_{ij}\}$ for the first i point on M, the Softmax function will normalize the matrix $\{S_{ij}\}$ by normalizing it and then summing it to get the set w_{ij}, which is computed as follows:

$$w_{ij} = e^{S_{ij}} / \sum_{b \in H(M_i)} e^{S_{ib}} \tag{19}$$

where the ordinal number of the point in $\{N_j\} j \in (1,k]$ is denoted as $H(M_i)$, S_{ij} is the probability that the i th point on M corresponds to the j th point in the set of points $\{N_j\}$, and b denotes an ordinal number in the set $H(M_i)$. Finally, the reconstructed point $\hat{N}_{M_i}$ is computed from the obtained weight matrix $\{w_{ij}\}$ and the set of points $\{N_j\}$ as:

$$\hat{N}_{M_i} = \sum_{b \in H(M_i)} w_{ib} N_b \tag{20}$$

Similarly, the reconstructed point $\hat{M}_{N_j}$ can be calculated. At this point, $H(N_j)$ in Eq. (20) and Eq. (21) needs to be modified to be the ordinal number of the k points in the model M that have the largest probability corresponding to the j th point on the model N.

(2) Outlier denoising based on statistical filtering

After the model point-clouding, although the clutter background noise of the 3D point cloud data is significantly suppressed, there are still some outlier points, and these noises will directly affect the spatial position accuracy, surface detail reduction and morphological consistency of the reconstructed model. In order to ensure the reliability of the information, statistical filtering algorithm is used in this study to eliminate the outlier point cloud noise in the point cloud.

Statistical filtering is a point cloud preprocessing technique based on probabilistic statistical theory, and its principle is to analyze the distribution characteristics of the neighborhood of each point k in the point cloud data, and calculate the average distance between the point and the points in the neighborhood, and then, calculate the mean value of all average distances μ and the standard cha σ, and based on the equation (22), combined with the scaling coefficient α, determine the distance threshold dmax, and finally, traversing the point cloud again, points with average distance greater than dmax are identified as outlier noisy points for elimination. Namely:

$$d\max = \mu + \alpha\sigma \tag{21}$$

The steps of the statistical filtering algorithm are:

Step1 Calculate the total number of points in the point cloud n, set the neighborhood to k, and calculate the average distance d from each point to all points within the distance k. Let there be any two non-overlapping points

$p_1(x_i, y_i, z_i)$ and $p_2(x_j, y_j, z_j)$ in the point cloud, and the formula for the average distance of each point to a point in the neighborhood of k in the iterative point cloud is:

$$d_i = \sum_{j=1}^{k} \frac{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}}{k} \tag{22}$$

Step2 Take the obtained set of n distances $\{d_1, d_2, d_3, ..., d_n\}$, and find its mean μ and standard deviation σ based on the following equations, i.e.:

$$\mu = \frac{1}{n} \sum_{i=1}^{n} d_i \tag{23}$$

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (d_i - \mu)^2} \tag{24}$$

Step3 Based on the calculated mean value μ and standard deviation σ, use the Gaussian distribution model to determine the maximum threshold value of dmax, where α as a coefficient to regulate the degree of influence of the standard deviation, and the size of its value directly determines the range of threshold values.

Step4 Compare the average distance μ obtained from each calculation with dmax, and identify and exclude the points whose average distance μ exceeds the threshold value dmax, which are regarded as abnormal noise points.

III. C. Experimental Methods and Analysis of Results

III. C. 1) Point cloud filtering effect

In order to better ensure the effect of 3D target modeling, this paper uses statistical filtering algorithm to remove the outlier points. In the statistical filtering algorithm, the number of neighborhood points and the standard deviation multiplier take the value of α have a direct impact on the size of the threshold. Fixing the number of neighborhood points to 30, when α takes different values, Table 1 shows the statistical results after filtering of a single seedling stage plant.

From the table, it can be seen that the smaller the value of α is, the more outlier points are filtered out, and the larger the value of α is, the fewer outlier points are filtered out, but the more points are filtered out, it cannot ensure that the characteristics of the original point cloud are retained, and it will also result in the phenomenon of point cloud nulling. Therefore, it is necessary to choose a suitable parameter for statistical filtering, and when the value of parameter α is 1.6, the best effect of the statistically filtered point cloud is 95.16%. In order to further verify the effect of statistical filtering when the value of parameter α is 1.6, the point cloud data of multiple seedling plants are intercepted for statistical filtering experiments, and the number of neighborhood points is also fixed at 30, and the value of α parameter is the same as that of a single fruit tree, and it is found that the statistical filtering effect of point cloud is the best after the parameter α is taken as 1.6 after filtering of multiple seedling plants. Therefore, in this paper, the number of neighborhood points is 30 and the value of α is 1.6 for the statistical filtering of the model point cloud, and the point cloud data after statistical filtering is used to realize the 3D target reconstruction of multi-view video.

Table 1: Outlier removal parameter settings

Original point number	α	Points after filtering	Retain percentage
85724	0.4	68124	79.47%
85724	0.8	73268	85.47%
85724	1.2	77919	90.90%
85724	1.6	81572	95.16%
85724	2.0	83996	97.98%
85724	2.4	84731	98.84%
85724	2.8	85613	99.87%

III. C. 2) Model comparison and ablation experiments

(1) Model comparison experiment

NVIDIA RTX A6000 graphics card was used for network training, the grid was trained from 644 resolution, by pruning and gradually upsampling to 1284, 2564, and finally 5123, 1000 steps were trained before each upsampling,

and the total number of optimization iterations was set to 106 steps, using a threshold of 1.25 for the optimization of the coarse-to-fine pruning and the batch size set to The batch size is set to 6000.The evaluation metrics are MSE and Peak Signal to Noise Ratio (PSNR).MSE is the square of the average difference between the modeled image and the input multiview map, the smaller the MSE, the higher the quality of the modeling. PSNR is the Peak Signal to Noise Ratio (PSNR) between the two images, the higher the PSNR, the smaller the difference between the two. When the PSNR value is greater than 35 dB, the modeling quality is good.

In the experimental design process of this paper, four kinds of seedling plants, namely, pepper, tomato, strawberry and green plant, are mainly selected to carry out the acquisition of multi-view video data, based on the acquired data, combined with the three-dimensional reconstruction steps given in the previous section, the quantitative experimental results of the three-dimensional target modeling of multi-view video of seedling plants are obtained as shown in Table 2. As can be seen from the table, the average value of PSNR for multiview video 3D modeling of seedling plants using the method of this paper is 37.04dB, and the average value of MSE for 3D target reconstruction is 19.17*10-5, and the overall training time is 21.61min. On the basis of ensuring the accuracy and quality of 3D target modeling of multiview video, the training time is relatively acceptable, which also indicates that the method of this paper effectiveness on multi-view video 3D target modeling.

Table 2: Quantitative results of seedling plant modeling

Plant	PSNR/dB	MSE/10-5	Training time/min
Peppers	37.42	24.16	21.28
Tomatoes	38.46	13.28	22.07
Strawberries	35.71	17.64	21.35
Greens	36.58	21.59	21.73
Means	37.04	19.17	21.61

In order to further illustrate the effectiveness of the multi-view video 3D target modeling method combining SFM and NeRF in this paper, based on the four types of seedling plant data acquired in this paper, SRN and NV are selected as the comparison models, and structural similarity (SSIM) is additionally selected as the evaluation index, SSIM is in the range of [0,1], and the larger the value is, it means the better the quality of the image, when the two images are are identical, at which time SSIM is 1. The comparison results of different methods are shown in Table 3.

The experimental results on the experimental dataset show that the PSNR and SSIM values of the 3D model built by the proposed method SRN and NV models for the four scenarios of chili peppers, tomatoes, strawberries, and greens, and the mean values of PSNR are improved by 8.43 dB and 5.85 dB, and the mean values of SSIM are increased by 23.55% and 17.67% compared to those of SRN and NV, respectively. In the four scenarios of pepper, tomato, strawberry and green, the network model proposed in this paper outperforms the comparison methods in all the evaluation metrics, which fully verifies its superiority in the task of 3D target modeling for multi-view videos. Combined with the actual 3D modeling diagrams, it can be seen that the images generated by SRN and NV have different levels of blurring and missing. In SRN, although the integrity can be better restored, the texture details are more blurred in the lower half and tip of the seedling plant. In NV, although the clarity can be better restored, there is more noise on the surface of the seedling plant, and the effect is still poor on the surface of the seedling plant with rich texture. In SFM-NeRF, the contour and edge details of the object can be preserved, and combined with the statistical filtering algorithm, a lot of noise effects are eliminated, and the characterization ability of the detailed features is improved to some extent. Meanwhile, in terms of 3D model integrity, the method effectively reduces the surface missing rate, especially in the complex structure region of the continuity has been significantly improved.

Table 3: Comparative experiments

Plant	Index	SRN	NV	Ours
Peppers	PSNR/dB	28.43	31.26	37.43
	SSIM	0.628	0.675	0.824
Tomatoes	PSNR/dB	27.15	30.38	38.06
	SSIM	0.593	0.604	0.715
Strawberries	PSNR/dB	30.24	32.85	36.18
	SSIM	0.615	0.638	0.769
Greens	PSNR/dB	28.86	30.51	36.72
	SSIM	0.627	0.669	0.735

(2) Model ablation experiment

In this paper, when carrying out 3D target reconstruction of multi-view video, multi-resolution hash grid + ball harmonic function is applied to carry out joint coding before carrying out 3D reconstruction using NeRF, so as to ensure the 3D target reconstruction effect. In order to verify the effectiveness of the joint coding method, this paper chooses four coding methods, namely Dense Grid (DG), Multi-Resolution Hash Grid (MHG), Spherical Harmonic Function (SHF) and Oneblob, for the ablation experiment. Depth L1 error (Depth L1 (cm)), accuracy (ACC (cm)), completeness (COMP (cm)), and completeness ratio (COMP Ratio (%)) with a threshold of 2 cm, GPU memory (G), and runtime (RT (min)) are selected as evaluation metrics. Table 4 shows the comparison results of reconstruction quality under different encoding methods.

From the table, it can be seen that joint coding is better than single coding, and the overall best results are obtained from the joint coding of multi-resolution hash grid and spherical harmonic function. Coordinate coding methods (frequency coding, ball-harmonic function) can accomplish void filling to some extent, but at the cost of a significant increase in training time. Achieving the filling through a finer spatial representation, although reasonably effective, the long training process limits the efficiency of its practical application. In contrast, the localized nature of parameter-based coding methods (dense grids, hash grids) makes them perform poorly in dealing with a large range of voids, and they are unable to fill large areas effectively, leading to insufficient accuracy and stability in complex scenes.

The joint coding strategy (multi-resolution hash grid + spherical harmonic function) proposed in this paper has significant advantages in filling voids. By combining coordinate coding and parametric coding, the strategy not only achieves smooth void filling, but also preserves detailed structural features. Specifically, the joint coding of multi-resolution hash mesh and ball harmonic function effectively overcomes the shortcomings of the traditional methods in filling accuracy and efficiency, and at the same time is more accurate in the detail part compared with other joint coding methods (dense mesh + ball harmonic function, multi-resolution hash mesh + Oneblob), which makes its void filling in complex scenes more natural and complete, and the detail part is more accurate and efficient.

Table 4: The comparison results of reconstruction quality

DG	MHG	SHF	One	ACC	COMP	COMP ratio
√				2.12	1.85	94.63
		√		15.38	19.43	38.51
	√			2.03	1.98	94.65
√		√		1.63	1.89	94.58
	√		√	18.27	2.46	92.43
	√	√		2.04	1.71	95.69
DG	MHG	SHF	One	Depth L1	GPU	RT
√				1.15	6.35	7.58
		√		28.24	0.64	3.34
	√			1.46	1.28	4.18
√		√		0.95	2.56	4.46
	√		√	4.47	1.07	3.15
	√	√		0.69	0.83	3.02

III. C. 3) Evaluation of reconstruction efficiency and quality

This paper further explores the relationship between reconstruction time and quality of modeled multi-view video 3D targets by controlling the number of iterations of the parameters in the neural radiation field. The experiment sets the number of iterations to 2000, 4000, 8000, 12000, 16000, 20000, 24000, 28000, 32000, 36000 times, and then evaluates the model quality. The dataset is randomly divided into a training set and a test set, and the test set is not involved in training, aiming to calculate the evaluation index as a real scenario when evaluating. Table 5 shows the results of the comparison between the number of iterations and the evaluation indexes, where ↑ and ↓ indicate that larger and smaller values are better, respectively.

As can be seen from the table, in the process of 0~28000 iterations on the model quality enhancement is more significant, 28000~32000 iterations in the process of image quality is improved, but not significant, 32000~36000 iterations in the process of PSNR there is a slight decrease in the SSIM, LPIPS and basically remain unchanged. The training time increases with the number of iterations, in addition to 0~8000 iterations process only cost 4.73min, the rest of the time spent per 4000 iterations about 3min, the time required to train a high-quality model is 20.25min.

Due to the optimization process adopts the stochastic gradient descent method, the Loss floating is more obvious, but the overall trend is decreasing. In order to better explore the relationship between the quality and time of the neural radiation field, the number of experimental iterations is expanded to 100000 times, and the Loss floating trend is investigated. During the first 30000 times of training there were many times when the Loss value plummeted and the model quality was unstable. During the 30000~60000 iterations, the Loss value shows a stable decreasing trend, and the decreasing amplitude is gradually reduced. 60000~100000 iterations, the Loss basically tends to stabilize.

Table 5: Iteration count and evaluation metrics

Iterations	PSNR/dB ↑	SSIM ↑	LPIPS ↓	Training time	Loss
2000	31.75	0.712	0.461	3.57	0.088
4000	33.26	0.733	0.458	4.05	0.076
8000	35.18	0.756	0.432	4.73	0.063
12000	35.64	0.789	0.405	7.51	0.045
16000	35.79	0.814	0.381	12.58	0.046
20000	36.08	0.823	0.362	15.29	0.037
24000	36.29	0.828	0.343	18.37	0.038
28000	36.18	0.831	0.328	20.25	0.035
32000	36.22	0.831	0.328	23.14	0.036
36000	36.19	0.831	0.328	29.06	0.034

IV. Conclusion

In this paper, a multi-view video 3D target modeling method based on SFM and NeRF is proposed, taking the seedling plant as a research example, by recording the video combined with Python for video frame extraction, and then inputting it into the NeRF network to realize the 3D reconstruction, as well as point-clouding and filtering to improve the accuracy of the 3D modeling. The results found that the average PSNR value of this paper's method for multiview video 3D modeling of seedling plants is 37.04 dB, the average MSE value of 3D target reconstruction is 19.17*10-5, and the overall 3D reconstruction efficiency is faster. Carrying out 3D target modeling of multi-view video based on deep learning technology can further optimize the 3D modeling effect and introduce 3D modeling technology into more and deeper application areas.

Although this study has achieved certain research results, the overall research sample lacks universality due to the selection of 3D reconstruction targets as seedling plants. In subsequent studies, the research sample will be further expanded to provide more possibilities for the wide application of 3D target modeling technology.

References

[1] Touloupaki, E., & Theodosiou, T. (2017). Performance simulation integrated in parametric 3D modeling as a method for early stage design optimization—A review. Energies, 10(5), 637.

[2] Abu Alhaija, H., Mustikovela, S. K., Mescheder, L., Geiger, A., & Rother, C. (2018). Augmented reality meets computer vision: Efficient data generation for urban driving scenes. International Journal of Computer Vision, 126, 961-972.

[3] Munkberg, J., Hasselgren, J., Shen, T., Gao, J., Chen, W., Evans, A., ... & Fidler, S. (2022). Extracting triangular 3d models, materials, and lighting from images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 8280-8290).

[4] Ioannidou, A., Chatzilari, E., Nikolopoulos, S., & Kompatsiaris, I. (2017). Deep learning advances in computer vision with 3d data: A survey. ACM computing surveys (CSUR), 50(2), 1-38.

[5] Dino, I. G., Sari, A. E., Iseri, O. K., Akin, S., Kalfaoglu, E., Erdogan, B., ... & Alatan, A. A. (2020). Image-based construction of building energy models using computer vision. Automation in Construction, 116, 103231.

[6] Zheng, J., Zhang, J., Li, J., Tang, R., Gao, S., & Zhou, Z. (2020). Structured3d: A large photo-realistic dataset for structured 3d modeling. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16 (pp. 519-535). Springer International Publishing.

[7] Izard, S. G., Torres, R. S., Plaza, Ó. A., Méndez, J. A. J., & García-Peñalvo, F. J. (2020). Nextmed: automatic imaging segmentation, 3D reconstruction, and 3D model visualization platform using augmented and virtual reality. Sensors (Basel, Switzerland), 20(10), 2962.

[8] Fernández-Palacios, B. J., Morabito, D., & Remondino, F. (2017). Access to complex reality-based 3D models using virtual reality solutions. Journal of cultural heritage, 23, 40-48.

[9] Bikmullina, I., & Garaeva, E. (2020, October). The development of 3D object modeling techniques for use in the unity environmen. In 2020 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon) (pp. 1-6). IEEE.

[10] Guo, M., Wang, B., Ma, P., Zhang, T., Owens, C., Gan, C., ... & Matusik, W. (2024). Physically compatible 3d object modeling from a single image. Advances in Neural Information Processing Systems, 37, 119260-119282.

[11] Murgitroyd, E., Madurska, M., Gonzalez, J., & Watson, A. (2015). 3D digital anatomy modelling–practical or pretty?. The Surgeon, 13(3), 177-180.

[12] Shah, S. A. A., Bennamoun, M., & Boussaid, F. (2017). Keypoints-based surface representation for 3D modeling and 3D object recognition. Pattern Recognition, 64, 29-38.
[13] Furrer, F., Novkovic, T., Fehr, M., Grinvald, M., Cadena, C., Nieto, J., & Siegwart, R. (2023). Modelify: An approach to incrementally build 3D object models for map completion. The International Journal of Robotics Research, 42(3), 45-65.
[14] Han, X. F., Laga, H., & Bennamoun, M. (2019). Image-based 3D object reconstruction: State-of-the-art and trends in the deep learning era. IEEE transactions on pattern analysis and machine intelligence, 43(5), 1578-1604.
[15] Pepik, B., Stark, M., Gehler, P., & Schiele, B. (2015). Multi-view and 3d deformable part models. IEEE transactions on pattern analysis and machine intelligence, 37(11), 2232-2245.
[16] Gadelha, M., Maji, S., & Wang, R. (2017, October). 3d shape induction from 2d views of multiple objects. In 2017 international conference on 3d vision (3DV) (pp. 402-411). IEEE.
[17] Delanoy, J., Aubry, M., Isola, P., Efros, A. A., & Bousseau, A. (2018). 3d sketching using multi-view deep volumetric prediction. Proceedings of the ACM on Computer Graphics and Interactive Techniques, 1(1), 1-22.
[18] Hao Lian,Meian Li,Ting Li,Yongan Zhang,Yanyu Shi,Yikun Fan... & Haibo Wu. (2025). Vehicle speed measurement method using monocular cameras. Scientific Reports,15(1),2755-2755.
[19] Nghia T Vo. (2025). Discorpy: algorithms and software for camera calibration and correction. Journal of synchrotron radiation.
[20] Yuquan Zhang & Guosheng Feng. (2025). Neural Radiance Field Dynamic Scene SLAM Based on Ray Segmentation and Bundle Adjustment. Sensors,25(6),1679-1679.
[21] Chibuike Onuoha,Shihao Luo,Jean Flaherty,Truong Thu Huong,Pham Ngoc Nam & Truong Cong Thang. (2025). A benchmark dataset for objective quality assessment of view synthesis for neural radiance field (NeRF). Data in brief,60,111484.