applied sciences

MDPI

Article

See-Then-Grasp: Object Full 3D Reconstruction via Two-Stage Active Robotic Reconstruction Using Single Manipulator

Youngtaek Hong [1], Jonghyeon Kim [1], Geonho Cha [2], Eunwoo Kim [3,*] and Kyungjae Lee [4,*]

[1] Department of Artificial Intelligence, Chung-Ang University, Seoul 06974, Republic of Korea; volunt4s@cau.ac.kr (Y.H.); kjh980610@cau.ac.kr (J.K.)
[2] NAVER Cloud, Seongnam 13561, Republic of Korea; geonho.cha@navercorp.com
[3] Department of Computer Science and Engineering, Chung-Ang University, Seoul 06974, Republic of Korea
[4] Department of Statistics, Korea University, Seoul 02841, Republic of Korea
* Correspondence: eunwoo@cau.ac.kr (E.K.); kyungjae_lee@korea.ac.kr (K.L.)

Abstract: In this paper, we propose an active robotic 3D reconstruction methodology for achieving full object 3D reconstruction. Existing robotic 3D reconstruction approaches often struggle to cover the entire view space of the object or reconstruct occluded regions, such as the bottom or back side. To address these limitations, we introduce a two-stage robotic active 3D reconstruction pipeline, named See-Then-Grasp (STG), that employs a robot manipulator for direct interaction with the object. The manipulator moves toward the points with the highest uncertainty, ensuring efficient data acquisition and rapid reconstruction. Our method expands the view space of the object to include the entire perspective, including occluded areas, making the previous fixed view candidate approach time-consuming for identifying uncertain regions. To overcome this, we propose a gradient-based next best view pose optimization method that efficiently identifies uncertain regions, enabling faster and more effective reconstruction. Our method optimizes the camera pose based on an uncertainty function, allowing it to identify the most uncertain regions in a short time. Through experiments with synthetic objects, we demonstrate that our approach effectively addresses the next best view selection problem, achieving significant improvements in computational efficiency while maintaining high-quality 3D reconstruction. Furthermore, we validate our method on a real robot, showing that it enables full 3D reconstruction of real-world objects.

Keywords: active 3D reconstruction; object full 3D object reconstruction; robot vision

Academic Editors: Zhufeng Shao and Fumin Zhang

Received: 28 November 2024
Revised: 22 December 2024
Accepted: 27 December 2024
Published: 30 December 2024

Citation: Hong, Y.; Kim, J.; Cha, G.; Kim, E.; Lee, K. See-Then-Grasp: Object Full 3D Reconstruction via Two-Stage Active Robotic Reconstruction Using Single Manipulator. *Appl. Sci.* **2025**, *15*, 272. https://doi.org/10.3390/app15010272

1. Introduction

Robotic manipulation has emerged as a critical area of research, both in industrial applications [1–3] and in everyday home environments [4–6], with significant attention focused on enabling robots to interact effectively with objects in their surroundings. A fundamental challenge in this domain is the need for comprehensive 3D information about objects, such as geometric shapes and structural details. Traditional methods in object manipulation, which rely on pre-existing 3D models, perform well with familiar objects but falter when faced with novel ones. Recent learning-based approaches show promise for simpler tasks with abundant training data but often struggle with more complex tasks, such as novel object assembly and rearrangement. In these cases, the ability to leverage 3D models becomes essential, underscoring the importance of active 3D reconstruction in enhancing robotic capabilities in diverse and unstructured environments.

In 3D modeling, traditional methods based on sensors, such as LiDAR [7,8] or structured light sensors [9,10], have been used for 3D reconstruction of novel objects. These high-cost sensors capture detailed depth information and are capable of reconstructing fine geometry. However, these methods often necessitate changes in various settings and configurations during the reconstruction process, leading to inevitable human intervention and increased complexity.

To overcome the limitations of these high-cost sensors, recent advancements have focused on using RGB cameras for 3D reconstruction. One such method is structure-from-motion (SfM), which also uses RGB images to reconstruct 3D models. However, SfM often requires multiple viewpoints and precise alignment to generate accurate geometry and can be computationally expensive for complex objects. More recently, neural implicit representations, such as neural radiance fields (NeRF) [11], have emerged as an alternative, enabling 3D reconstruction of unknown objects using only RGB images. NeRF has garnered significant attention in the robot vision field due to its ability to produce high-quality 3D reconstructions without requiring high-cost sensors. NeRF's scene-agnostic 3D reconstruction capabilities have facilitated a range of complex robotic downstream tasks [12–16].

Building on these advances, recent studies in active robotic 3D reconstruction have increasingly incorporated neural implicit representations [17–20]. However, most research in this area assumes rooted objects, which prevents the reconstruction of hidden surfaces such as the bottom of an object, the back of a partially occluded object, or intricate details in cavities and recesses. This issue is particularly pronounced in single fixed-based robotic manipulator systems, where the reachable view space is limited to the manipulator's workspace, restricting object 3D reconstruction to these limited viewpoints [21,22]. As a result, critical parts of the object, such as its bottom side or occluded regions, may not be reconstructed accurately, leading to incomplete models.

To address these challenges, we propose a two-stage robotic active 3D reconstruction pipeline, named See-Then-Grasp (STG), using two cameras within a single robot manipulator system. In the first stage, a hand-eye camera captures the object's approximate geometry. In the second stage, the robot grasps the object and presents it to a fixed external camera, achieving full 3D coverage. To expedite viewpoint selection, we introduce a gradient-based pose optimization technique, enabling the robot to efficiently identify and observe uncertain regions, ensuring a complete 3D reconstruction. Our proposed pipeline is illustrated in Figure 1.

Our contributions can be summarized as follows:

- We propose a novel two-stage active robotic 3D reconstruction pipeline to address the limitations of fixed-base single manipulator systems, which are constrained to partial object reconstruction due to restricted view space. Our pipeline leverages two cameras to achieve full 3D reconstruction of objects by combining hand-eye and external fixed camera perspectives.
- We develop a gradient-based next best view selection method, designed to efficiently explore the full view space of an object by identifying uncertain regions. This method leverages uncertainty in neural implicit representations to optimize view poses for uncovering unseen regions.
- Through experiments on synthetic objects, we demonstrate that our method finds uncertain regions faster and more accurately compared to other approaches. Additionally, by applying our method to a robotic system, we successfully enable full 3D reconstruction of novel real-world objects.

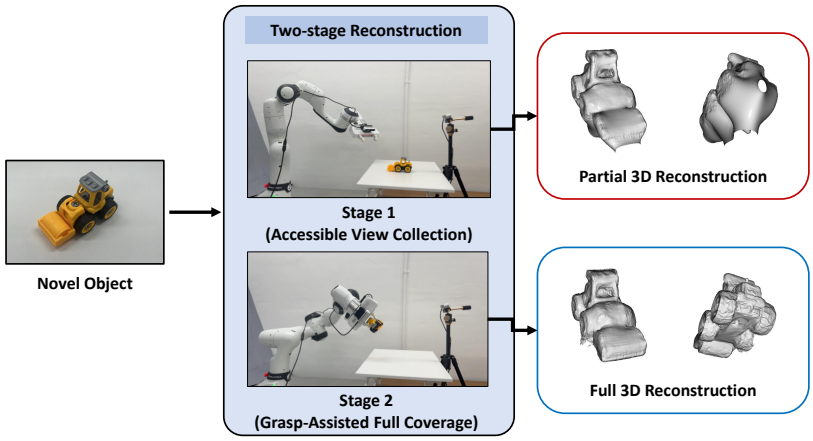

Figure 1. Overview. We propose an active learning framework that automates the full 3D reconstruction of novel objects using a single robot manipulator system. As shown in the red box, a fixed-base single manipulator system can only partially reconstruct the front part of an object due to the constraints of the robot's view space. By utilizing our proposed See-Then-Grasp method, the robot can directly interact with the object, granting access to the entire view space of the object. As illustrated in the blue box, this allows for full 3D reconstruction.

2. Related Work

2.1. Autonomous 3D Reconstruction in Robotic System

Autonomous 3D reconstruction of objects using robots is a long-standing problem. Previous works have employed various robotic platforms for this task, including Unmanned Aerial Vehicles (UAVs) [19,20], wheeled mobile robots [17,23], and fixed-base robotic arms [9,21,22,24].

To model the entire geometry of an object, a robot must have access to the entire view space that allows for the observation of all sides of the object. In this regard, UAVs and wheeled mobile robots are frequently used robotic platforms as they can cover a 360° view of the object without space restriction. Specifically, Lee et al. [17] generate finite-view candidates that cover the whole-object geometry, allowing the mobile robot to access view candidates. Yan et al. [19] use UAVs to freely access the view space within a continuous SE(3) manifold.

In the fixed-based robotic arm setup, the workspace of the robot is limited, resulting in the inability to access the view space of static objects, such as the back or bottom side of the objects. To address this limitation, Wu et al. [9] utilize a multi-robot system to expand the view space of static objects. Lee et al. [24] employ an additional rotary table to enable a single robot system to access the 360° view space of the object.

Most 3D reconstruction approaches in robotic applications often rely on RGB-D data captured by sensors such as Microsoft Kinect, leveraging both color and depth information for high-accuracy reconstructions [25–28]. For instance, KinectFusion [26] uses RGB-D data to perform real-time 3D reconstruction by aligning and merging depth frames. Similarly, Isler et al. [25] formulate volumetric information gain using RGB-D sensors, enabling active reconstruction with mobile robots. However, these methods depend on specialized depth sensors, which can be costly, less scalable, and sensitive to factors like lighting, calibration, and object reflectivity.

In our work, we propose a method that overcomes these limitations by utilizing only RGB cameras combined with neural implicit representation. By directly manipulating objects within a single robotic arm setup, our approach enables complete 3D reconstruction, including occluded regions, without the need for depth sensors, making it more cost-effective and robust.

2.2. Neural Implicit Representation

Neural implicit representations, such as NeRF [11], have gained significant attention in the fields of novel view synthesis [29] and object surface reconstruction [30,31] due to their geometry-agnostic nature and flexible training pipelines. Notably, neural implicit representations enable high-quality 3D scene modeling using only 2D images, prompting numerous efforts to integrate these representations into robotic tasks such as object grasping [15,16], SLAM [14], and visual navigation [12,13].

To apply neural implicit representations in robotic systems, fast and efficient training is crucial. Recent advancements have utilized various techniques to enhance training speeds. Instant-NGP [32] uses hash encoding to provide efficient data encoding while reducing memory usage, allowing for rapid scene modeling. DVGO [33] combines explicit voxel grids with neural networks to enable fast reconstruction of high-resolution 3D scenes. TensoRF [34] leverages high-dimensional tensor decomposition techniques to enhance the expressiveness of NeRF models and improve training efficiency, achieving superior performance even in complex scenes.

Our primary goal is to quickly reconstruct the 3D surface of objects within a robotic framework using sparse object views. To this end, we adopt Voxurf [35], which uses neural SDFs for accurate object surface prediction and voxel grid representation to enable a fast training speed.

2.3. Uncertainty Measurement for Next Best View Selection

A next best view (NBV) selection is a robot vision problem in which the robot chooses the optimal viewpoint to minimize uncertainty in an unseen environment or object model. To solve the NBV selection problem efficiently, a high-quality uncertainty measurement is crucial.

Recently, using neural implicit representation for uncertainty quantification in NBV has been attracting attention. Refs. [36–38] have proposed a stochastic NeRF incorporating Variational Inference [39] into the NeRF learning pipeline. Refs. [40,41] have designed additional MLPs within the NeRF model to directly quantify uncertainty. Ref. [42] utilizes a deep ensemble approach [43] to quantify NeRF uncertainty through the variance of ensemble members.

Although prior works have shown successful results, these approaches require modifications to the NeRF model architecture or involve an increase in the number of network parameters, which extends rendering times. This can lead to increased decision-making times, making them potentially unsuitable for robotic frameworks.

Consequently, we adopt volumetric uncertainty by calculating the entropy of ray weight distribution, as in [17,44]. Using entropy-based uncertainty allows for a focused analysis of an object's 3D geometry and enables rapid uncertainty calculation without the need for additional training or NeRF model modification.

3. Background

Three-dimensional representation with NeRF [11] generally utilizes multilayer perceptrons (MLPs) to model complex 3D scene geometry. Specifically, the MLPs take 3D coordinates $\mathbf{x} = (x, y, z)$ and 2D viewing directions $\mathbf{d} = (\theta, \phi)$ to predict the RGB colors $\mathbf{c}$ and volumetric density σ as

$$\text{MLP}(\mathbf{x}, \mathbf{d}) = (\mathbf{c}, \sigma) : \mathbb{R}^3 \times \mathbb{R}^2 \mapsto \mathbb{R}^4. \tag{1}$$

Then, once the NeRF MLP is trained, $\mathbf{c}$ and σ can be used to synthesize images from novel viewpoints through the volume-rendering method [45]. In the field of robotics, the volumetric density σ can further be used to extract the geometric information of objects, which can then be utilized for manipulation tasks.

In volume rendering, the color of each pixel is rendered as the expected color computed along with the ray from the camera's origin to the pixel location. Since the NeRF dataset consists of pairs of ground truth RGB images $\mathcal{I}$ and camera poses $\mathcal{T} \in \mathrm{SE}(3)$, denoted as $\{\mathcal{I}_i, \mathcal{T}_i\}_{i=1}$, the origin and viewing direction of the camera can be obtained from the corresponding camera pose $\mathcal{T}$. Given the camera's origin $\mathbf{o}$ and viewing direction $\mathbf{d}$ based on the pixel location, the ray is defined as a half-line starting from the origin, i.e., $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ for $t \geq 0$. The expected color $C(\mathbf{r})$ along with the ray $\mathbf{r}$ is computed as

$$C(\mathbf{r}) := \int_0^\infty T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt, \tag{2}$$

$$T(t) := \exp\left(-\int_0^t \sigma(\mathbf{r}(s))ds\right) \tag{3}$$

where $T(t)$ is the accumulated transmittance along the ray $\mathbf{r}$. While Equation (2) provides the theoretically grounded manner to calculate the expected pixel color, the integration in Equation (2) is often intractable when $\mathbf{c}$ and σ are modeled as the MLP. Hence, numerical estimation [46] of the expected color is widely employed. The integration in Equation (2) is approximated using a set of N discrete points $\{p_i\}_{i=1}^N$ sampled along the ray $\mathbf{r}$, i.e., $p_i := \mathbf{o} + t_i\mathbf{d}$ for some $t_i \in (0, \infty)$. Then, the estimated color can be computed as follows,

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} w_i \mathbf{c}_i, \; w_i = T_i\alpha_i, \; T_i = \prod_{j=1}^{i-1}(1 - \alpha_j), \tag{4}$$

where $\mathbf{c}_i$ and σ_i are a color and density at p_i, i.e., $(\mathbf{c}_i, \sigma_i) = f_\theta(p_i, \mathbf{d})$; w_i is a ray weight at p_i; T_i represents the accumulated transmittance; $\alpha_i = 1 - \exp(-\sigma_i\delta_i)$ represents the ray termination probability at p_i; and $\delta_i = t_{i+1} - t_i$ is the distance between adjacent samples. $\hat{C}(\mathbf{r})$ is computed as the weighted sum of color samples $\mathbf{c}_i$ with ray weight w_i.

Despite NeRF being highly effective in 3D scene representation, its dependency on volume density fields for novel view synthesis poses significant challenges for extracting high-quality surfaces. To mitigate this limitation, an existing approach combines neural implicit representation with signed distance functions (SDFs), leading to the development of neural SDF representation. In this framework, the SDF value $f(x)$ at a given 3D point x is modeled by the neural network. In [30], the employment of SDF values introduces a distinct formulation of the opacity term, α_i, different from the approach utilized in NeRF. Specifically, in [30], α_i is defined as

$$\alpha_i = \max\left(\frac{\Phi_s(f(p_i)) - \Phi_s(f(p_{i+1}))}{\Phi_s(f(p_i))}, 0\right), \tag{5}$$

where $f(p_i)$ represents the SDF value at 3D point p_i, and $\Phi_s(x) = (1 + e^{-sx})^{-1}$ denotes the sigmoid function. The parameter s is either automatically learned during the training process or manually set.

Due to the neural SDF representation, NeuS demonstrates strong capabilities in reconstructing the surfaces of novel objects. However, its application in real-time robotic systems is constrained by the time-consuming nature of the learning process and the significant computational costs involved. To address this limitation, we adopt Voxurf [35], a model that leverages voxel-based representations for accelerating the surface reconstruction process. Voxurf enhances the training speed by combining an explicit voxel grid with a neural SDF representation. Specifically, Voxurf directly learns the SDF values of the geometry through an SDF voxel grid $V^{(\mathrm{SDF})} \in \mathbb{R}^{1 \times N_x \times N_y \times N_z}$ and predicts the color using a feature voxel grid

$V^{(\text{feat})} \in \mathbb{R}^{C \times N_x \times N_y \times N_z}$ with a lightweight MLP. For a specific $(\mathbf{x}, \mathbf{d})$, the SDF value d' and the color $\mathbf{c}$ can be obtained using trilinear interpolation as follows:

$$d' = \text{interp}(\mathbf{x}, \mathcal{G}(V^{(\text{SDF})}, k_g, \sigma_g)), \tag{6}$$

$$\mathbf{c} = \text{MLP}(\text{interp}(\mathbf{x}, V^{(\text{feat})}), \mathbf{x}, \mathbf{d}, n). \tag{7}$$

For d', $\mathcal{G}$ represents a 3D convolution on the voxel grid $V^{(\text{SDF})}$ using a Gaussian kernel with size k_g and standard deviation σ_g to smooth the voxel SDF values rather than using the raw data. For $\mathbf{c}$, the normal vector n is derived from the $V^{(\text{SDF})}$. Lastly, d' is substituted into Equation (5) to calculate α_i and then volume rendering is performed as in Equation (4) to obtain the final pixel color.

4. Proposed Method

We propose the See-Then-Grasp (STG) method for active 3D modeling using a single robotic manipulator system. This method allows the robot to interact with the object, efficiently exploring and addressing uncertain regions for comprehensive 3D reconstruction. The pipeline consists of two main stages. The first stage is an accessible view collection. In this stage, the robot's hand-eye camera captures initial images from reachable viewpoints, building a preliminary understanding of the object's geometry. The second stage involves grasp-assisted full coverage. In the second stage, the robot grasps the object and uses a fixed external camera to capture additional images from previously inaccessible angles, ensuring complete 3D coverage. The entire pipeline is summarized in Figure 2.

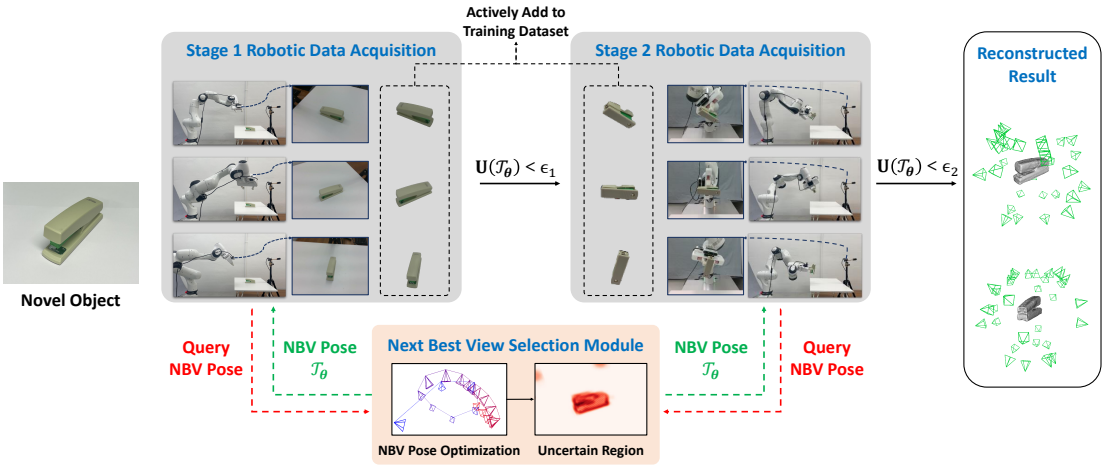

Figure 2. Overall pipeline. We propose the See-Then-Grasp method for learning the full 3D geometry of an object in a two-stage reconstruction process. In each stage, the robot actively acquires data by receiving the optimal next view pose $\mathcal{T}_{\boldsymbol{\Theta}}$ from the next best view selection module, as represented by the red and green dashed lines, moving to that position, and capturing images. The captured images undergo post-processing and are actively added to the training dataset as represented by the black dashed lines, ultimately enabling the model to learn the full 3D geometry of the object.

4.1. Object Uncertainty Measurement

Unlike other uncertainty measurement methods [36,40,42] that require model modifications or extra parameter learning, we measure object geometry uncertainty using the entropy of the ray weight distribution. This entropy-based approach is efficient and fast, with no need for model changes.

$$H(\mathbf{r}) = -\sum_{i=1}^{N} [w_i \log(w_i) + (1 - w_i) \log(1 - w_i)], \tag{8}$$

where w_i is the ray weight defined in (4). Noisy ray weight distributions indicate inaccurate geometry, leading to high entropy. As geometry accuracy improves, the distribution converges to a single peak, reducing the entropy. Finally, the 2D pixel-wise uncertainty map $\mathbf{U}$, corresponding to the single camera pose $\mathcal{T}_{\text{cam}} \in \text{SE}(3)$, is calculated as the mean value of the entropy of all the rays:

$$\mathbf{U}(\mathcal{T}_{\text{cam}}) = \frac{1}{|\mathcal{R}_{\text{cam}}|} \sum_{\mathbf{r} \in \mathcal{R}_{\text{cam}}} H(\mathbf{r}), \tag{9}$$

where $\mathcal{R}_{\text{cam}}$ represents the batch of rays originating from single camera pose $\mathcal{T}_{cam}$.

While entropy-based uncertainty effectively represents object geometry uncertainty, applying it directly to the original NeRF model can cause overfitting, missing uncertainty in unseen views. To address this, we introduced an additional uncertain alpha voxel grid $V^{(unc,\alpha)}$, inspired by [44]. Initialized to maximum uncertainty, this grid is updated during training with α values from $V^{(\text{SDF})}$, maintaining high uncertainty in unseen viewpoints and enabling the NBV agent to capture diverse views. As illustrated in Figure 3, our uncertainty measurement can capture both seen-view and unseen-view uncertainties.

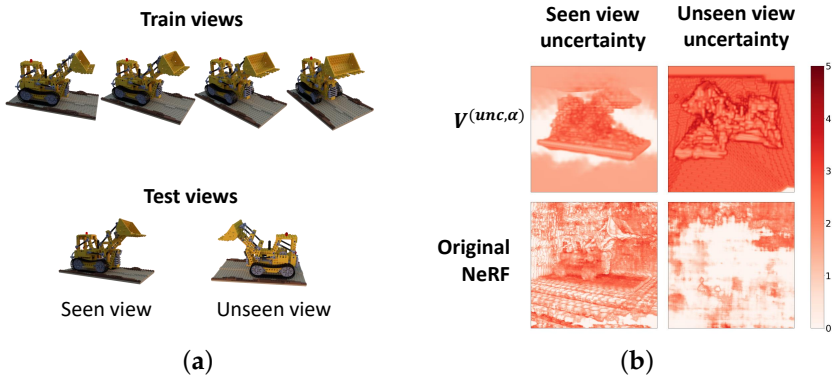

Figure 3. Comparison of uncertainty in the original NeRF and uncertain alpha voxel grid for the unseen view. We divide the views into (**a**) a seen view, which is similar to the train view, and an unseen view, which represents an out-of-distribution view, to investigate the overfitting issue of entropy-based uncertainty. As illustrated in (**b**), the original NeRF model fails to accurately capture the uncertainty in the unseen view due to the overfitting issue when trained using only one side of an object (**bottom right**). However, by applying $V^{(\text{unc},\alpha)}$ to a voxel-based model, it can effectively capture the uncertainty in the unseen view (**top right**).

4.2. Next Best View Pose Optimization

In robotic active reconstruction, quickly identifying the most uncertain view for NBV pose selection is crucial. The existing methods rely on fixed candidate views, requiring time-consuming rendering from all predefined viewpoints.

We propose a gradient-based NBV pose optimization method that reduces the rendering time by selecting NBV poses with fewer renderings. By using gradient information to pinpoint where uncertainty increases most rapidly, our approach achieves an efficient NBV with minimal renderings.

To generate an object-centric differentiable pose, we define the transformation matrix $\mathcal{T}_{\boldsymbol{\Theta}}$ using spherical coordinates (ϕ, θ, r) where $\boldsymbol{\Theta} := (\phi, \theta)$ indicates the learnable parameters. First, we define the translation vector $\mathbf{t}$ by converting the spherical coordinates to Cartesian coordinates:

$$\mathbf{t}_{\boldsymbol{\Theta}} = [x, y, z]^{\intercal} = [r \sin\theta \cos\phi, r \sin\theta \sin\phi, r \cos\theta]^{\intercal}. \tag{10}$$

Next, we define the rotation axes to ensure an object-centric view:

- The z-axis of the camera frame denoted as $\vec{w}$ is oriented toward the object origin:

$$\vec{w} = -\frac{(x, y, z)}{\sqrt{x^2 + y^2 + z^2}}. \tag{11}$$

- The x-axis of the camera frame denoted as $\vec{u}$ is aligned parallel to the xy-plane of the global object frame:

$$\vec{u} = \frac{(-y, x, 0)}{\sqrt{x^2 + y^2}}. \tag{12}$$

- The y-axis of the camera frame denoted as $\vec{v}$ is derived as the cross product of the rotation z-axis and the rotation x-axis:

$$\vec{v} = \vec{w} \times \vec{u}. \tag{13}$$

The rotation matrix $\mathbf{R}$ is formed by these orthogonal vectors, $\vec{u}$, $\vec{v}$, and $\vec{w}$:

$$\mathbf{R}_{\boldsymbol{\Theta}} = \begin{pmatrix} \vec{u} & \vec{v} & \vec{w} \end{pmatrix}^{\mathsf{T}}. \tag{14}$$

Finally, the 4×4 transformation matrix $\mathcal{T}_{\boldsymbol{\Theta}}$ is defined as follows:

$$\mathcal{T}_{\boldsymbol{\Theta}} = \begin{pmatrix} \mathbf{R}_{\boldsymbol{\Theta}} & \mathbf{t}_{\boldsymbol{\Theta}} \\ 0 & 1 \end{pmatrix}. \tag{15}$$

By this construction, the transformation matrix $\mathcal{T}_{\boldsymbol{\Theta}}$ is differentiable since all the pose construction steps are tractable.

To ensure diverse exploration during the NBV pose optimization process and avoid overfitting to the existing training set poses, we introduce a simple regularizer loss $\mathcal{L}_{\text{reg}}$ as follows:

$$\mathcal{L}_{\text{reg}} = \frac{1}{N_t} \sum_{i=1}^{N_t} \text{err}(\mathcal{T}_{\boldsymbol{\Theta}}, \mathcal{T}_i), \tag{16}$$

where N_t is the number of poses in the existing training set, and $\text{err}(\cdot, \cdot)$ refers to the absolute value of the pose error, including both the euclidean distance between the translations and the rotational error between the orientations of the two poses.

The final NBV loss $\mathcal{L}_{\text{NBV}}$ is calculated as the negative sum of the uncertainty map as in Equation (9) observed from $\mathcal{T}_{\boldsymbol{\Theta}}$ and $\mathcal{L}_{\text{reg}}$:

$$\mathcal{L}_{\text{NBV}} = -\lambda_1 \mathbf{U}(\mathcal{T}_{\boldsymbol{\Theta}}) - \lambda_2 \mathcal{L}_{\text{reg}}, \tag{17}$$

where λ_1 and λ_2 are scaling parameters. By optimizing $\mathcal{L}_{\text{NBV}}$ with a gradient descent optimizer such as Adam [47], $\mathcal{T}_{\boldsymbol{\Theta}}$ gradually moves toward viewpoints with increasing uncertainty within the current scene. As shown in Figure 4, our proposed NBV pose optimization enables direct optimization for uncertain viewpoints, allowing for 60% faster exploration compared to the fixed view candidate methodology.

After performing the optimization iterations, $\mathcal{T}_{\boldsymbol{\Theta}}$ corresponds to the camera pose with the highest uncertainty. In other words, by moving the manipulator to satisfy this camera pose with respect to the novel object and actively updating the dataset, the 3D model can be reconstructed more quickly. A robotic pipeline capable of performing full 3D reconstruction of the object will be introduced in Section 4.3.

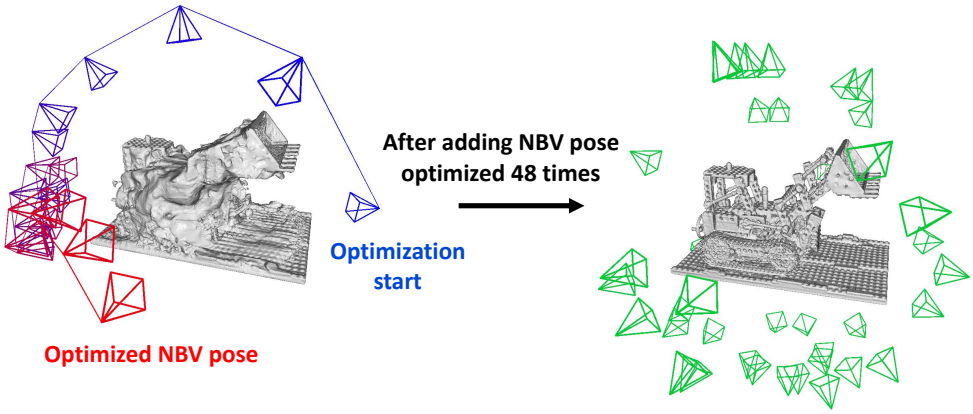

Figure 4. Visualization of our proposed gradient-based NBV pose optimization. Starting from the blue-colored pose added in the previous optimization step, we obtain a camera trajectory that moves to the uncertain red-colored pose while optimizing $\mathcal{L}_{\text{NBV}}$ (**left side**). Repeating this optimization step 50 times and adding it to the training set yields refined 3D reconstruction results (**right side**).

4.3. See-Then-Grasp: Two-Stage Robotic Full 3D Reconstruction Pipeline

Our proposed NBV pose optimization enables access to all viewpoints within a spherical view space for complete 3D reconstruction of an object. Previous studies, which used a fixed-base single manipulator system and a static object, often failed to achieve full coverage, especially for hidden surfaces. This system can only observe specific parts of the object, limiting the view space and preventing full 3D reconstruction of hidden surfaces like the bottom or back side. To overcome this limitation, we propose a novel two-stage reconstruction method utilizing a single robot manipulator and two cameras that directly interact with the object. The summarized method is represented in Algorithm 1.

Algorithm 1 See-Then-Grasp: Two-stage Robotic Active 3D Reconstruction

Require: Voxurf model, Initial training set $\{\mathcal{I}_i, \mathcal{T}_i\}_{i=1}^{1}$
Initialize Voxurf model and start training with $\{\mathcal{I}_i, \mathcal{T}_i\}_{i=1}^{1}$
Stage 1: Accessible View Collection
  Constrain NBV view space θ, ϕ
  while $\mathbf{U}(\mathcal{T}_{\boldsymbol{\Theta}}) > \epsilon_1$ **do**
    if Training iteration mod $t_{\text{NBV}} = 0$ **then**
      Optimize $\mathcal{L}_{\text{NBV}}$ to find NBV pose $\mathcal{T}_{\boldsymbol{\Theta}}$
      Move robot hand-eye camera to $\mathcal{T}_{\boldsymbol{\Theta}}$
      Capture image $\mathcal{I}_t$ and apply mask $\mathcal{M}$
      Update training set $\{\mathcal{I}_i, \mathcal{T}_i\}_{i=1}^{t+1} \leftarrow \mathcal{I}_t, \mathcal{T}_{\boldsymbol{\Theta}}$ and retrain model
    end if
    Optimize $\mathcal{L}_{\text{recon}}$
  end while
  Generate $\mathcal{T}_{\text{grasp\_eef}}^{\text{base}}$ and grasp the object
Stage 2: Grasp-Assisted Full Coverage
  Release the constraints θ, ϕ
  while $\mathbf{U}(\mathcal{T}_{\boldsymbol{\Theta}}) > \epsilon_2$ **do**
    if Training iteration mod $t_{\text{NBV}} = 0$ **then**
      Optimize $\mathcal{L}_{\text{NBV}}$ to find NBV pose $\mathcal{T}_{\boldsymbol{\Theta}}$
      Move robot end-effector to $\mathcal{T}_{\text{move}}$
      Capture image $\mathcal{I}_t$ and apply mask $\mathcal{M}$
      Update training set $\{\mathcal{I}_i, \mathcal{T}_i\}_{i=1}^{t+1+\text{stage 1}} \leftarrow \mathcal{I}_t, \mathcal{T}_{\boldsymbol{\Theta}}$ and retrain model
    end if
    Optimize $\mathcal{L}_{\text{recon}}$
  end while

4.3.1. Stage 1 (Accessible View Collection)

Stage 1 involves utilizing the robot's hand-eye camera to perform the 3D reconstruction of objects within the robot's reachable view space. Specifically, the view space is constrained within the angular limits of $\theta_{\text{lim.1}} \leq \theta \leq \theta_{\text{lim.2}}$ and $\phi_{\text{lim.1}} \leq \phi \leq \phi_{\text{lim.2}}$, where $\theta_{\text{lim.1}}$, $\theta_{\text{lim.2}}$, $\phi_{\text{lim.1}}$, and $\phi_{\text{lim.2}}$ represent the maximum range the hand-eye camera can achieve. The primary objective of stage 1 is to preserve the areas that will be occluded by the gripper when the view space is expanded in stage 2.

At the beginning of stage 1, the Voxurf model starts training using the initial training dataset $\{\mathcal{I}_i, \mathcal{T}_i\}_{i=1}^{1}$, where $\mathcal{I}_i$ and $\mathcal{T}_i$ are augmented by selecting any view within the constrained view space. When the overall training iteration reaches every active learning iteration t_{NBV}, the NBV pose optimization begins. As a result of optimizing $\mathcal{L}_{\text{NBV}}$, the NBV pose $\mathcal{T}_{\boldsymbol{\Theta}}$ with the highest uncertainty is obtained. The robot hand-eye camera is then maneuvered to align with the object center frame according to the pose $\mathcal{T}_{\boldsymbol{\Theta}}$. Afterward, a new image is captured, and a segmentation mask is applied to acquire the NBV image data $\mathcal{I}_t$. The new image and pose pair are added to the current training dataset, updating it to $\{\mathcal{I}_i, \mathcal{T}_i\}_{i=1}^{t+1} \leftarrow \mathcal{I}_t, \mathcal{T}_{\boldsymbol{\Theta}}$. Because the training dataset is updated through active learning, the entire training ray batch $\mathcal{R}_{\text{train}}$ is also augmented with rays from the updated pose set.

Finally, the reconstruction loss is calculated using the updated training ray batch $\mathcal{R}_{\text{train}}$ as follows:

$$\mathcal{L}_{\text{recon}} = \frac{1}{|\mathcal{R}_{\text{train}}|} \sum_{r \in \mathcal{R}_{\text{train}}} (||\tilde{\mathbf{C}}(\mathbf{r}), \hat{\mathbf{C}}(\mathbf{r})||_2^2), \tag{18}$$

where the expected pixel color $\hat{\mathbf{C}}(\mathbf{r})$ is computed as in Equation (4), while $\tilde{\mathbf{C}}(\mathbf{r})$ represents the ground truth pixel color for ray $\mathbf{r}$ and can be obtained from the training image dataset $\mathcal{I}_i$. In our active learning setting, data are gradually added for the uncertain regions of the object, resulting in the progressive improvement in the object's shape as the reconstruction loss $\mathcal{L}_{\text{recon}}$ is optimized. The active learning process continues until the object uncertainty $\mathbf{U}(\mathcal{T}_{\boldsymbol{\Theta}})$ falls below a predefined threshold ϵ_1, at which point the system transitions to stage 2.

4.3.2. Stage 2 (Grasp-Assisted Full Coverage)

In stage 2, the manipulator grasps the object and utilizes another fixed external camera to cover all areas of the object by showing the uncertain regions. Specifically, to obtain data regarding the NBV pose $\mathcal{T}_{\boldsymbol{\Theta}}$ that allows for capturing the complete view of the object, the robot grasps the object and moves its end effector, satisfying $\mathcal{T}_{\boldsymbol{\Theta}}$ relative to the external camera. The robot first employs a grasp pose generator such as Contact-GraspNet [48] to generate the grasp pose $\mathcal{T}_{\text{grasp\_eef}}^{\text{base}}$ for grasping the object. Subsequently, we release the view space constraints from stage 1, expanding the view space around the object to a full spherical range. Then, the robot end-effector pose $\mathcal{T}_{\text{move}}^{\text{base}}$ is generated following the transformation order.

$$\mathcal{T}_{\text{move}}^{\text{base}} = \mathcal{T}_{\text{cam2}}^{\text{base}} (\mathcal{T}_{\boldsymbol{\Theta}})^{-1} (\mathcal{T}_{\text{obj}}^{\text{base}})^{-1} \mathcal{T}_{\text{grasp\_eef}}^{\text{base}}, \tag{19}$$

where $\mathcal{T}_{\text{cam2}}^{\text{base}}$ and $\mathcal{T}_{\text{obj}}^{\text{base}}$ represent the pose of the external camera and the object canonical pose placed on the table, both relative to the robot's base frame. Note that $(\mathcal{T}_{\text{obj}}^{\text{base}})^{-1} \mathcal{T}_{\text{grasp\_eef}}^{\text{base}}$ represents the transformation between the object pose and the end-effector frame after the object has been grasped and attached. By planning the path of the end effector according to $\mathcal{T}_{\text{move}}^{\text{base}}$ while the robot grasps the object, the object pose as seen from the external camera can satisfy $\mathcal{T}_{\boldsymbol{\Theta}}$.

When stage 2 begins, the Voxurf model initiates the active learning process, incorporating the training dataset obtained from stage 1. By optimizing $\mathcal{L}_{\text{NBV}}$, the full-coverage NBV pose $\mathcal{T}_{\boldsymbol{\Theta}}$ is determined, and the robot's end effector is moved to $\mathcal{T}_{\text{move}}^{\text{base}}$ to align with

$\mathcal{T}_{\Theta}$ as seen from the external camera. Subsequently, an image is captured with the external camera and a segmentation mask is applied, adding the new image and pose pair to the current training dataset: $\{\mathcal{I}_i, \mathcal{T}_i\}_{i=1}^{t+1+\text{stage 1}} \leftarrow \mathcal{I}_t, \mathcal{T}_{\Theta}$. Similar to stage 1, the training process continues by optimizing the reconstruction loss $\mathcal{L}_{\text{recon}}$ in Equation (18). The process is concluded when the object uncertainty $\mathbf{U}(\mathcal{T}_{\Theta})$ drops below the predefined threshold ϵ_2.

Due to the use of two cameras, the backgrounds of the views from each camera differ, making it challenging to satisfy the view consistency of the object. To ensure the view consistency for each camera, we applied HQ-SAM [49] to generate object segmentation masks from the captured images. Furthermore, to retain the geometric information obtained in stage 1 during stage 2, we prevent gradients from flowing in areas outside the mask, thus preserving the information in regions occluded by the gripper. Specifically, a binary segmentation mask $\mathcal{M}$, which has the same size as the raw image $\mathcal{I}_{\text{raw}}$, is defined as follows:

$$\mathcal{M}(\mathcal{I}_{\text{raw}}) = \begin{cases} 1 & \text{if the pixel in } \mathcal{I}_{\text{raw}} \in \text{object,} \\ 0 & \text{if the pixel in } \mathcal{I}_{\text{raw}} \in \text{background.} \end{cases} \tag{20}$$

At each stage, after generating a mask that excludes the background areas, including non-object regions such as the robot gripper and the surrounding environment, the mask is applied to the original image to create the image used for training, $\mathcal{I}$, as follows:

$$\mathcal{I} = \mathcal{M}(\mathcal{I}_{\text{raw}}) \cdot \mathcal{I}_{\text{raw}}. \tag{21}$$

By using the masked images for training, the object images generated at each stage maintain view consistency, allowing the Voxurf model to focus on learning the objects. As a result, the data acquired from each stage enable the complete 3D reconstruction of the object.

5. Experiment

We conduct experiments on synthetic and real-world objects to evaluate our proposed method for active 3D reconstruction through NBV pose optimization. In Section 5.1, we validate the proposed uncertainty measurement method and NBV pose optimization on synthetic objects. In Section 5.2, we describe the experimental results of applying the proposed STG method to a real robot for novel real-world objects.

5.1. Experiment on Synthetic Objects

5.1.1. Implementation Details

To implement the proposed NBV pose optimization, we set the pose parameters ϕ and θ as learnable parameters using PyTorch (version 2.0) [50] while fixing the radius r at 4.0. We use the Adam [47] optimizer with a learning rate of 1.0 to optimize the NBV loss function $\mathcal{L}_{\text{NBV}}$. Additionally, the scaling parameters λ_1 and λ_2 are set to 1.0 and 0.5, respectively. For each optimization step, we perform 20 optimization iterations every 2000 training iterations to obtain the NBV pose. The experiments utilize the code provided by the authors of Voxurf [35]. Since Voxurf consists of a coarse-to-fine stage training procedure, the data obtained from NBV optimization in the coarse stage are used for training in the fine stage to achieve the final 3D reconstruction results. All the synthetic dataset experiments are conducted on a system equipped with an Intel Xeon Gold 6226R CPU at 2900 MHz, 125 GB of RAM, and an NVIDIA GeForce RTX A5000 GPU. The CPU is manufactured by Intel Corporation (Santa Clara, CA, USA), and the GPU is manufactured by NVIDIA Corporation (Santa Clara, CA, USA).

5.1.2. Dataset

We select four objects—Lego, Ficus, Chair, and Mic—from the NeRF-Synthetic [11] dataset. For each NBV pose optimization step, we use Blender [51] to render the image of the optimized pose to generate the training image data. Additionally, we generate 150 images and corresponding pose data that cover the entire view of each object, including the bottom side, for use as the test dataset.

5.1.3. Metrics

We evaluate the results of the synthetic experiments in terms of rendering quality and geometric quality. For rendering quality, we use the peak signal-to-noise ratio (PSNR) and the learned perceptual image patch similarity (LPIPS) metric. The PSNR measures the fidelity of the rendered 2D images by calculating the pixel-wise difference between the images rendered from the trained model and the ground truth images, with higher PSNR values indicating better image quality. LPIPS, on the other hand, evaluates perceptual similarity using a deep learning-based model, where lower LPIPS values indicate greater visual similarity to the ground truth images.

For geometric quality, we use the F-score and Chamfer distance metrics to assess the difference between the reconstructed 3D mesh and the ground truth mesh. The F-score is calculated by sampling points from each mesh and setting a threshold of 1 cm to evaluate completeness and accuracy, where higher F-score values indicate better alignment between the reconstructed and ground truth meshes. The Chamfer distance measures the average distance between points on the reconstructed mesh and their closest counterparts on the ground truth mesh, with smaller values indicating a closer match and thus higher geometric fidelity.

5.1.4. Results

The quantitative results of our experiments are presented in Table 1, while the qualitative results of our baseline are shown in Figure 5. According to Table 1, our baseline outperforms other baselines in terms of rendering and geometric quality when 24 NBV images are added. This performance difference is particularly significant for geometrically complex datasets, such as Lego and Ficus. Even with the addition of 48 NBV images, our baseline demonstrates superior results, although the other baselines tend to converge to a similar level.

Figure 5. Qualitative 3D reconstruction results of our baseline. We present the 3D reconstruction of four synthetic objects, improved by adding 48 NBV images to our baseline. The ground truth images are on the left, with the 3D reconstruction meshes on the right, showcasing our method's ability to accurately capture fine details.

To further evaluate the NBV selection performance of each baseline, we present the qualitative comparison results in Figure 6 and the F-score progression for the Ficus dataset in Figure 7. These results indicate that our baseline reaches the oracle line more rapidly than other methods and effectively captures both the geometric details and the overall shape.

Table 1. Quantitative results of synthetic object reconstruction. We quantitatively evaluated the performance of NBV selection using four synthetic objects. The experiments were conducted under two scenarios: using 24 NBV images and using 48 NBV images. Each experiment was repeated five times with different random seeds, and the average results are reported. Grad denotes our NBV pose optimization method, whereas Fixed refers to a fixed pose candidate method. Note that CD represents the Chamfer distance, with the results being scaled by a factor of 100. Higher values are better for PSNR and F-Score, while lower values are better for LPIPS and CD.

	Lego				Ficus				Chair				Mic			
Method	**PSNR ↑**	**LPIPS ↓**	**F-Score ↑**	**CD ↓**	**PSNR ↑**	**LPIPS ↓**	**F-Score ↑**	**CD ↓**	**PSNR ↑**	**LPIPS ↓**	**F-Score ↑**	**CD ↓**	**PSNR ↑**	**LPIPS ↓**	**F-Score ↑**	**CD ↓**
Quality after 24 NBV images added																
Grad + Entropy (Ours)	31.69	0.0508	0.4022	0.2613	26.96	0.0564	0.6403	0.1071	31.30	0.0319	0.3810	0.3345	30.26	0.0312	0.4010	0.3045
Grad + Ensemble [42]	30.48	0.0637	0.3908	0.2708	24.24	0.0780	0.6129	0.1479	31.46	0.0312	0.3891	0.3168	28.58	0.0319	0.4091	0.2868
Fixed [17,18] + Entropy	29.29	0.0682	0.3651	0.3665	25.41	0.0544	0.5308	0.1899	30.13	0.0394	0.3887	0.3214	29.11	0.0394	0.4098	0.2848
Fixed [17,18] + Ensemble [42]	30.26	0.0522	0.3819	0.6322	24.11	0.0798	0.6184	0.1381	31.85	0.0329	0.3898	0.3148	27.73	0.0389	0.4087	0.2914
FVS	28.63	0.0611	0.3509	0.4427	24.52	0.0539	0.5003	0.2215	29.97	0.0464	0.3688	0.4269	26.15	0.0464	0.3888	0.4169
Random	27.46	0.0789	0.2649	0.7026	23.26	0.0671	0.4834	0.3083	28.56	0.0641	0.3464	0.5938	26.30	0.0641	0.3664	0.5938
Quality after 48 NBV images added																
Grad + Entropy (Ours)	34.73	0.0319	0.4886	0.1408	34.49	0.0204	0.8482	0.0504	35.86	0.0275	0.4168	0.2844	34.15	0.0230	0.7241	0.0785
Grad + Ensemble [42]	34.04	0.0321	0.4711	0.1425	34.13	0.0244	0.8149	0.1048	36.10	0.0248	0.4216	0.2805	34.79	0.0202	0.7211	0.0736
Fixed [17,18] + Entropy	34.16	0.0324	0.4295	0.2857	34.51	0.0203	0.8405	0.0514	35.60	0.0294	0.4144	0.2869	33.73	0.0229	0.7232	0.0787
Fixed [17,18] + Ensemble [42]	34.10	0.0363	0.4603	0.1616	30.08	0.0715	0.8198	0.0945	36.04	0.0253	0.4128	0.2833	34.41	0.0212	0.7181	0.0791
FVS	32.79	0.0384	0.4413	0.2134	32.85	0.0289	0.8356	0.1101	35.27	0.0271	0.4135	0.2845	34.02	0.0234	0.7178	0.0814
Random	33.41	0.0390	0.4118	0.2772	33.62	0.0237	0.8301	0.1148	35.59	0.0263	0.4121	0.2883	33.89	0.0241	0.7072	0.0836
All Images	-	-	0.4992	0.095	-	-	0.8560	0.0476	-	-	0.4244	0.2615	-	-	0.7279	0.0715

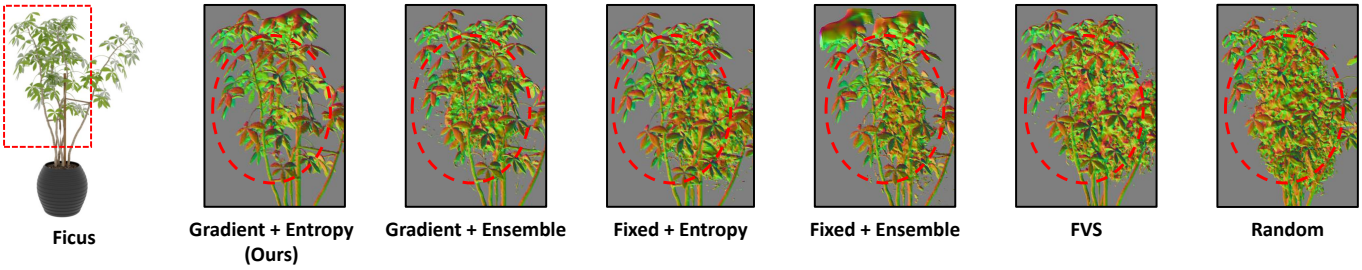

Figure 6. Qualitative comparison of the baselines on the Ficus dataset. We present the surface normal plots of the 3D models using 24 NBV images from the Ficus dataset. The red dashed line shows that our baseline effectively captures geometric details, while others produce noisy geometries.

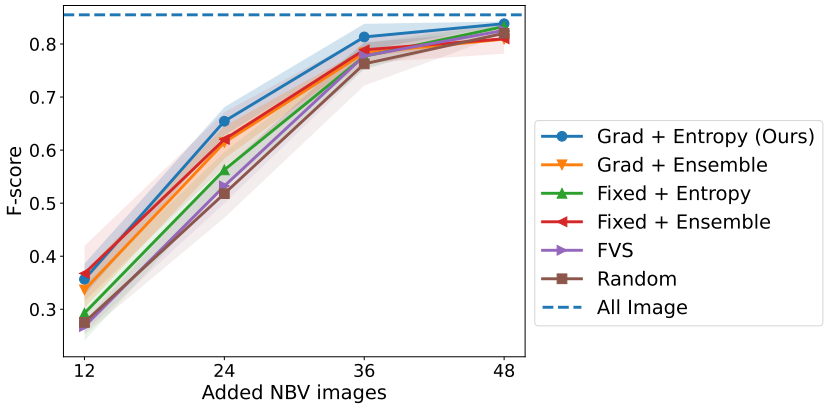

Figure 7. F-score improvement plot for the baselines on the Ficus dataset. We analyze the F-score improvement on the Ficus dataset using 12, 24, 36, and 48 NBV images, with the results averaged over five random seeds. The mean-variance plots show that our baseline quickly identifies regions that significantly boost the F-score, especially in the early stages of NBV selection.

Finally, the time cost measurements for each baseline are illustrated in Figure 8. Our NBV pose optimization efficiently identifies uncertain regions without exploring all view

candidates through a single NBV pose optimization. As a result, our method solves the NBV selection problem 60% faster compared to the fixed view candidate method.

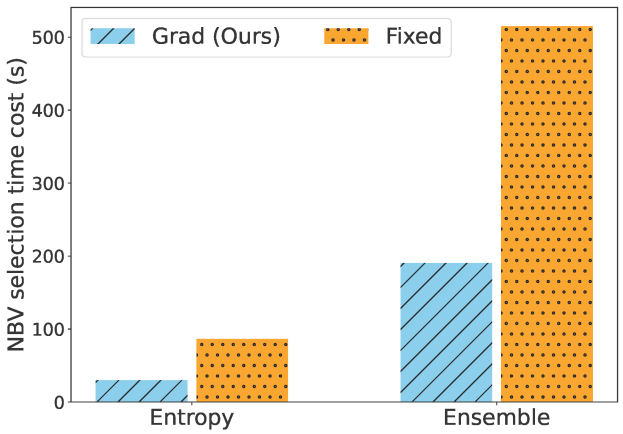

Figure 8. Comparison of NBV selection time costs. We measured the time for each NBV selection across different uncertainty techniques and methods. The sky blue textured bar represents the time taken when performing our NBV pose optimization, while the orange textured bar represents the time taken when selecting the NBV pose using fixed view candidates. Our method consistently incurs lower time costs compared to the fixed candidate method.

5.2. Experiment on Real-World Objects

5.2.1. Implementation Details

We apply the proposed two-stage reconstruction method to a real robot to achieve full reconstruction of real-world objects. The overall NBV pose optimization approach follows the same settings as defined in Section 5.1. In stage 1, to generate $\mathcal{T}_{\Theta}$ within the robot's reachable range, we set $\theta_{\text{lim.1}}$ and $\theta_{\text{lim.2}}$ to $\pi/5$ and $\pi/3$, respectively, and $\phi_{\text{lim.1}}$ and $\phi_{\text{lim.2}}$ to $7\pi/6$ and $11\pi/6$, respectively. An HQ-SAM [49] model with pretrained ViT-L [52] is employed to create segmentation masks for the objects at each stage. The stop criteria for each stage, ϵ_1 and ϵ_2, are set to 0.5 and 1.0, respectively.

We utilize the Franka Panda manipulator (Franka Emika GmbH, München, Germany) as the robotic hardware platform and employ two RealSense D435i cameras (Intel Corporation, Santa Clara, CA, USA), with one configured as a robot hand-eye camera and the other as an external fixed camera. For collision-free motion planning of the robot, the RRT-Connect [53] algorithm is employed. All the real-world experiments are conducted on a system equipped with a 12th Gen Intel Core i7 CPU at 3600 MHz, 32 GB of RAM, and an NVIDIA GeForce RTX 3080 GPU.

5.2.2. Datasets

We select four real-world objects: Mouse, Toy car, Stapler, and Toy screw. During stages 1 and 2, images of the objects are captured using cameras set to a 640 $\times$ 480 resolution. Conducting experiments on novel real-world objects without ground truth 3D models presents significant challenges in obtaining accurate object poses. Therefore, we predefine $\mathcal{T}_{\text{move}}$ with the help of an AprilTag [54] cube, ensuring that during training the obtained $\mathcal{T}_{\text{move}}$ aligns with the nearest predefined poses. We pregenerate 300 $\mathcal{T}_{\text{move}}$ to cover all the viewpoints of the object, ensuring that the maximum error between the poses generated during training and the pregenerated poses remains below 0.1.

5.2.3. Results

We present the qualitative results using our proposed STG method for each real object in Figure 9. Our method allows the robot to interact directly with the object, capturing

all viewpoints, including the bottom side, which are not visible from a robot hand-eye camera. As illustrated in Figure 9, our method effectively reconstructs the back and bottom of objects, even in a limited view space with a fixed-based manipulator setup.

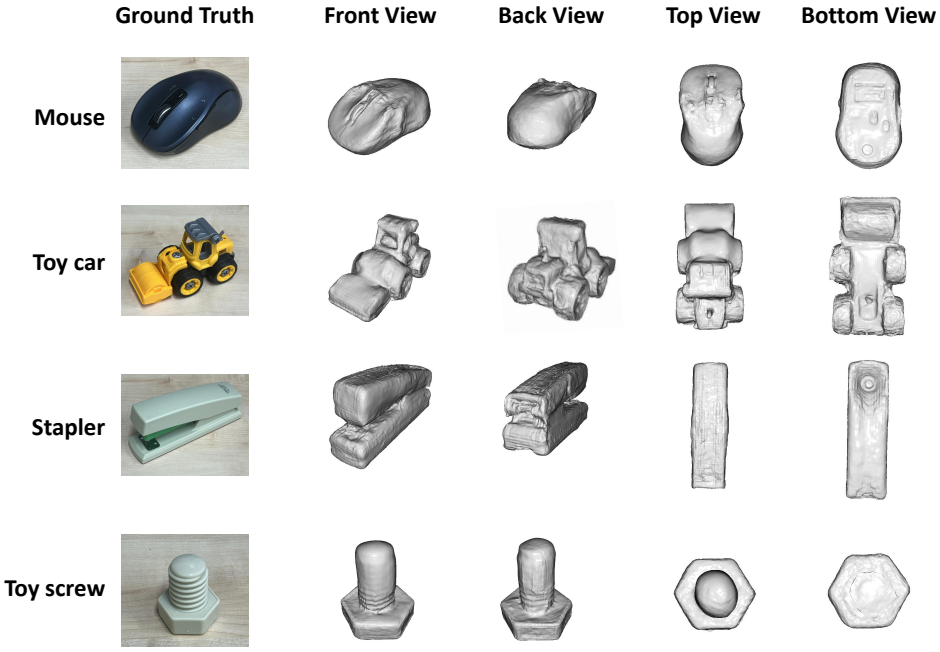

Figure 9. Qualitative results of 3D models for four real-world objects. We present 3D reconstructions of novel objects using our STG method, capturing all viewpoints, including the bottom side.

As demonstrated in Section 5.1, our gradient-based NBV pose optimization significantly reduces the number of rendering iterations required to identify uncertain regions compared to the fixed view candidate method. Under the same experimental conditions, the real-world experiments showed consistent results, achieving a 60% improvement in efficiency for identifying uncertain regions. This efficiency translates into a total average time of approximately 15 min for NBV data collection and full 3D reconstruction, including the robot's operation time. These results demonstrate that our proposed STG method enables robots to rapidly acquire the necessary training data for the comprehensive 3D reconstruction of objects, maintaining both speed and reconstruction quality across synthetic and real-world scenarios.

6. Ablation Study

6.1. Performance Comparison Based on NBV Pose Optimizer Parameters

Since we use the Adam [47] optimizer for NBV pose optimization, we report the comparative results in Table 2, obtained by varying the learning rate and optimization steps of the optimizer. Our NBV pose optimization explores the entire sphere; so with a low learning rate, the exploration range is reduced, requiring many optimization steps to reach uncertain areas. Therefore, we increase the learning rate to 1.0 to quickly reach uncertain areas with fewer optimization steps. Intuitively, more optimization steps lead to a more accurate approach to uncertain areas, but this comes with the trade-off of increased time costs. Thus, we utilize the optimizer configuration that achieves acceptable results with only 20 optimization steps, enabling an efficient approach to uncertain regions with a relatively low time cost.

Table 2. Quantitative results of the NBV pose optimizer with adjusted parameters for the Ficus dataset. We report the quantitative results from varying the learning rate and optimization steps of the NBV pose optimizer on the Ficus dataset. Our configuration achieves acceptable results with a reduced time cost compared to other settings. Higher values are better for PSNR and F-Score, while lower values are better for LPIPS, CD and time costs.

Learning Rate/Optimization Step	PSNR ↑	LPIPS ↓	F-Score ↑	CD ↓	Time Cost (s) ↓
0.1/10	23.13	0.0771	0.5185	0.2498	13.41
0.1/20	23.19	0.0702	0.5548	0.1848	30.04
0.1/40	23.42	0.0689	0.5741	0.1754	60.85
1.0/10	25.49	0.0607	0.5713	0.1248	14.85
1.0/20 (Ours)	26.96	0.0564	0.6403	0.1071	29.69
1.0/40	27.89	0.0479	0.6387	0.1046	60.31

6.2. Effectiveness of Our Two-Stage Reconstruction Pipeline

To validate the effectiveness of our two-stage method, we provide the qualitative results of iterative 3D reconstructions at each stage in Figure 10. As shown in Figure 10, during stage 1, where 4 and 12 NBV images are added, the approximate geometry of the Toy car can be captured from the upper view. However, the geometry from the lower view, which is inaccessible to the view space, cannot be adequately captured. Subsequently, when 24 NBV images are added, the data from stage 2 begin to improve the geometry of the bottom view. When 36 NBV images are added, the geometric details from all views can be fully captured. Thus, our STG method effectively achieves a comprehensive 3D reconstruction of the object.

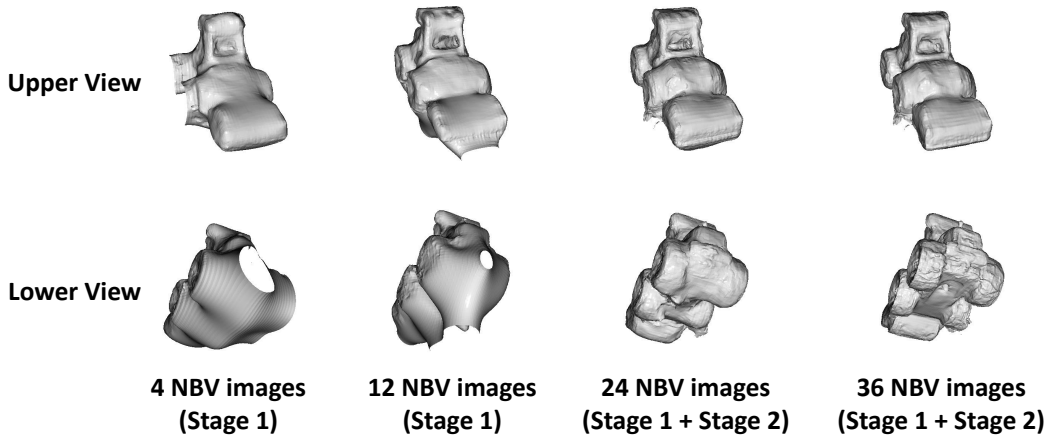

Figure 10. Iterative 3D reconstruction results for the Toy car dataset. We present the qualitative results of the iterative 3D reconstruction on the Toy car dataset, showing that our method progressively captures both upper and lower views.

7. Conclusions

This paper tackles the problem of active robotic 3D reconstruction for novel objects using a single robot manipulator system. We proposed a NBV pose optimization method that efficiently identifies uncertain regions of the object using a gradient-based approach. Our pose optimization method, which directly moves to uncertain regions, offers a notable improvement in time efficiency compared to the fixed candidate approach that evaluates all pose candidates. To address the limited viewpoint issue of a single robot manipulator system, we introduced the See-Then-Grasp (STG) method, a two-stage robotic full 3D reconstruction pipeline. Initially, the robot learns the approximate geometry from reachable viewpoints, then grasps the object and expands the viewpoint by showing the uncertain

region to an external fixed camera. We conducted experiments on both synthetic and real-world objects, demonstrating the effectiveness of our approach.

Our results confirm that the proposed NBV pose optimization directs the robot to uncertain regions more quickly and accurately than other baseline methods, achieving a 60% improvement in computational efficiency. Furthermore, applying our method to real-world objects validates its capability to achieve full 3D reconstruction, including challenging occluded areas such as the bottom and back sides of objects. These findings underscore the potential of our method to significantly enhance the efficiency and quality of robotic 3D reconstruction in diverse environments.

While this work successfully addresses several challenges in active robotic 3D reconstruction, it is limited by its reliance on a single manipulator and the inherent constraints of fixed-base systems. Future work will focus on extending this approach to multi-robot systems and exploring real-time 6D pose estimation [55,56] to further improve reconstruction performance. By providing a robust framework for autonomous and complete object modeling, our method opens new opportunities for applications in manufacturing, logistics, and robotics research.

Author Contributions: Conceptualization, Y.H., J.K. and K.L.; methodology, Y.H. and K.L.; software, Y.H. and J.K.; validation, Y.H.; writing—original draft preparation, Y.H., J.K. and K.L.; writing—review and editing, Y.H., G.C., E.K. and K.L.; supervision, G.C., E.K. and K.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2023-00211357) and the Chung-Ang University Graduate Research Scholarship in 2023.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all the subjects involved in this study.

Data Availability Statement: The data supporting the findings of this study are available upon request from the corresponding author. Public access is restricted to protect privacy.

Conflicts of Interest: Author Geonho Cha was employed by the company NAVER Cloud. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Ota, K.; Jha, D.K.; Oiki, T.; Miura, M.; Nammoto, T.; Nikovski, D.; Mariyama, T. Trajectory optimization for unknown constrained systems using reinforcement learning. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 3487–3494.
2. Luo, J.; Solowjow, E.; Wen, C.; Ojea, J.A.; Agogino, A.M.; Tamar, A.; Abbeel, P. Reinforcement learning on variable impedance controller for high-precision robotic assembly. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 3080–3087.
3. Sun, Y.; Wang, W.; Chen, Y.; Jia, Y. Learn how to assist humans through human teaching and robot learning in human–robot collaborative assembly. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *52*, 728–738. [CrossRef]
4. Liang, J.; Huang, W.; Xia, F.; Xu, P.; Hausman, K.; Ichter, B.; Florence, P.; Zeng, A. Code as policies: Language model programs for embodied control. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 9493–9500.
5. Zitkovich, B.; Yu, T.; Xu, S.; Xu, P.; Xiao, T.; Xia, F.; Wu, J.; Wohlhart, P.; Welker, S.; Wahid, A.; et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In Proceedings of the 7th Conference on Robot Learning, PMLR, Atlanta, GA, USA, 6 November 2023; pp. 2165–2183.
6. Liu, Z.; Bahety, A.; Song, S. REFLECT: Summarizing Robot Experiences for Failure Explanation and Correction. In Proceedings of the 7th Conference on Robot Learning, PMLR, Atlanta, GA, USA, 6 November 2023; pp. 3468–3484.

7. Tachella, J.; Altmann, Y.; Mellado, N.; McCarthy, A.; Tobin, R.; Buller, G.S.; Tourneret, J.Y.; McLaughlin, S. Real-time 3D reconstruction from single-photon lidar data using plug-and-play point cloud denoisers. *Nat. Commun.* **2019**, *10*, 4984. [CrossRef] [PubMed]
8. Wu, Q.; Yang, H.; Wei, M.; Remil, O.; Wang, B.; Wang, J. Automatic 3D reconstruction of electrical substation scene from LiDAR point cloud. *ISPRS J. Photogramm. Remote Sens.* **2018**, *143*, 57–71. [CrossRef]
9. Wu, C.; Zeng, R.; Pan, J.; Wang, C.C.; Liu, Y.J. Plant phenotyping by deep-learning-based planner for multi-robots. *IEEE Robot. Autom. Lett.* **2019**, *4*, 3113–3120. [CrossRef]
10. Wu, S.; Sun, W.; Long, P.; Huang, H.; Cohen-Or, D.; Gong, M.; Deussen, O.; Chen, B. Quality-driven poisson-guided autoscanning. *ACM Trans. Graph. (TOG)* **2014**, *33*, 1–12. [CrossRef]
11. Mildenhall, B.; Srinivasan, P.P.; Tancik, M.; Barron, J.T.; Ramamoorthi, R.; Ng, R. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* **2021**, *65*, 99–106. [CrossRef]
12. Adamkiewicz, M.; Chen, T.; Caccavale, A.; Gardner, R.; Culbertson, P.; Bohg, J.; Schwager, M. Vision-only robot navigation in a neural radiance world. *IEEE Robot. Autom. Lett.* **2022**, *7*, 4606–4613. [CrossRef]
13. Kwon, O.; Park, J.; Oh, S. Renderable neural radiance map for visual navigation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 9099–9108.
14. Zhu, Z.; Peng, S.; Larsson, V.; Xu, W.; Bao, H.; Cui, Z.; Oswald, M.R.; Pollefeys, M. Nice-slam: Neural implicit scalable encoding for slam. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 12786–12796.
15. Dai, Q.; Zhu, Y.; Geng, Y.; Ruan, C.; Zhang, J.; Wang, H. Graspnerf: Multiview-based 6-dof grasp detection for transparent and specular objects using generalizable nerf. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 1757–1763.
16. Blukis, V.; Lee, T.; Tremblay, J.; Wen, B.; Kweon, I.S.; Yoon, K.J.; Fox, D.; Birchfield, S. One-Shot Neural Fields for 3D Object Understanding. In Proceedings of the CVPR Workshop on Advances in NeRF for the Metaverse (XRNeRF), Vancouver, BC, Canada, 18 June 2023.
17. Lee, S.; Chen, L.; Wang, J.; Liniger, A.; Kumar, S.; Yu, F. Uncertainty guided policy for active robotic 3d reconstruction using neural radiance fields. *IEEE Robot. Autom. Lett.* **2022**, *7*, 12070–12077. [CrossRef]
18. Ran, Y.; Zeng, J.; He, S.; Chen, J.; Li, L.; Chen, Y.; Lee, G.; Ye, Q. Neurar: Neural uncertainty for autonomous 3d reconstruction with implicit neural representations. *IEEE Robot. Autom. Lett.* **2023**, *8*, 1125–1132. [CrossRef]
19. Yan, D.; Liu, J.; Quan, F.; Chen, H.; Fu, M. Active Implicit Object Reconstruction using Uncertainty-guided Next-Best-View Optimization. *IEEE Robot. Autom. Lett.* **2023**, *8*, 6395–6402. [CrossRef]
20. Zeng, J.; Li, Y.; Ran, Y.; Li, S.; Gao, F.; Li, L.; He, S.; Chen, J.; Ye, Q. Efficient view path planning for autonomous implicit reconstruction. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 4063–4069.
21. Monica, R.; Aleotti, J. A probabilistic next best view planner for depth cameras based on deep learning. *IEEE Robot. Autom. Lett.* **2021**, *6*, 3529–3536. [CrossRef]
22. Pan, S.; Jin, L.; Hu, H.; Popović, M.; Bennewitz, M. How many views are needed to reconstruct an unknown object using nerf? In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024; pp. 12470–12476.
23. Hardouin, G.; Moras, J.; Morbidi, F.; Marzat, J.; Mouaddib, E.M. A multirobot system for 3-D surface reconstruction with centralized and distributed architectures. *IEEE Trans. Robot.* **2023**, *39*, 2623–2638. [CrossRef]
24. Lee, I.D.; Seo, J.H.; Kim, Y.M.; Choi, J.; Han, S.; Yoo, B. Automatic pose generation for robotic 3-D scanning of mechanical parts. *IEEE Trans. Robot.* **2020**, *36*, 1219–1238. [CrossRef]
25. Delmerico, J.; Isler, S.; Sabzevari, R.; Scaramuzza, D. A comparison of volumetric information gain metrics for active 3D object reconstruction. *Auton. Robot.* **2018**, *42*, 197–208. [CrossRef]
26. Newcombe, R.A.; Izadi, S.; Hilliges, O.; Molyneaux, D.; Kim, D.; Davison, A.J.; Kohi, P.; Shotton, J.; Hodges, S.; Fitzgibbon, A. Kinectfusion: Real-time dense surface mapping and tracking. In Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality, Basel, Switzerland, 26–29 October 2011; pp. 127–136.
27. Tychola, K.A.; Tsimperidis, I.; Papakostas, G.A. On 3D reconstruction using rgb-d cameras. *Digital* **2022**, *2*, 401–421. [CrossRef]
28. Cai, Z.; Han, J.; Liu, L.; Shao, L. RGB-D datasets using microsoft kinect or similar sensors: A survey. *Multimed. Tools Appl.* **2017**, *76*, 4313–4355. [CrossRef]
29. Barron, J.T.; Mildenhall, B.; Verbin, D.; Srinivasan, P.P.; Hedman, P. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 5470–5479.

30. Wang, P.; Liu, L.; Liu, Y.; Theobalt, C.; Komura, T.; Wang, W. NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 27171–27183.
31. Li, Z.; Müller, T.; Evans, A.; Taylor, R.H.; Unberath, M.; Liu, M.Y.; Lin, C.H. Neuralangelo: High-fidelity neural surface reconstruction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 8456–8465.
32. Müller, T.; Evans, A.; Schied, C.; Keller, A. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph. (TOG)* **2022**, *41*, 1–15. [CrossRef]
33. Sun, C.; Sun, M.; Chen, H.T. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 5459–5469.
34. Chen, A.; Xu, Z.; Geiger, A.; Yu, J.; Su, H. Tensorf: Tensorial radiance fields. In Proceedings of the 17th European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Springer: Cham, Switzerland, 2022; pp. 333–350.
35. Wu, T.; Wang, J.; Pan, X.; Xu, X.; Theobalt, C.; Liu, Z.; Lin, D. Voxurf: Voxel-based Efficient and Accurate Neural Surface Reconstruction. In Proceedings of the International Conference on Learning Representations (ICLR), Kigali, Rwanda, 1–5 May 2023.
36. Shen, J.; Ruiz, A.; Agudo, A.; Moreno-Noguer, F. Stochastic neural radiance fields: Quantifying uncertainty in implicit 3D representations. In Proceedings of the 2021 International Conference on 3D Vision (3DV), Virtual, 1–3 December 2021; pp. 972–981.
37. Shen, J.; Agudo, A.; Moreno-Noguer, F.; Ruiz, A. Conditional-flow nerf: Accurate 3d modelling with reliable uncertainty quantification. In Proceedings of the 17th European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Springer: Cham, Switzerland, 2022; pp. 540–557.
38. Shen, J.; Ren, R.; Ruiz, A.; Moreno-Noguer, F. Estimating 3D uncertainty field: Quantifying uncertainty for neural radiance fields. In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024; pp. 2375–2381.
39. Blei, D.M.; Kucukelbir, A.; McAuliffe, J.D. Variational inference: A review for statisticians. *J. Am. Stat. Assoc.* **2017**, *112*, 859–877. [CrossRef]
40. Pan, X.; Lai, Z.; Song, S.; Huang, G. Activenerf: Learning where to see with uncertainty estimation. In Proceedings of the 17th European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Springer: Cham, Switzerland, 2022; pp. 230–246.
41. Martin-Brualla, R.; Radwan, N.; Sajjadi, M.S.; Barron, J.T.; Dosovitskiy, A.; Duckworth, D. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 7210–7219.
42. Sünderhauf, N.; Abou-Chakra, J.; Miller, D. Density-aware nerf ensembles: Quantifying predictive uncertainty in neural radiance fields. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 9370–9376.
43. Lakshminarayanan, B.; Pritzel, A.; Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6402–6413.
44. Zhan, H.; Zheng, J.; Xu, Y.; Reid, I.; Rezatofighi, H. Activermap: Radiance field for active mapping and planning. *arXiv* **2022**, arXiv:2211.12656.
45. Kajiya, J.T.; Von Herzen, B.P. Ray tracing volume densities. *ACM SIGGRAPH Comput. Graph.* **1984**, *18*, 165–174. [CrossRef]
46. Max, N. Optical models for direct volume rendering. *IEEE Trans. Vis. Comput. Graph.* **1995**, *1*, 99–108. [CrossRef]
47. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
48. Sundermeyer, M.; Mousavian, A.; Triebel, R.; Fox, D. Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 13438–13444.
49. Ke, L.; Ye, M.; Danelljan, M.; Liu, Y.; Tai, Y.W.; Tang, C.K.; Yu, F. Segment anything in high quality. In Proceedings of the 37th International Conference on Neural Information Processing Systems, New Orleans, LA, USA, 10–16 December 2023; pp. 29914–29934.
50. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An imperative style, high-performance deep learning library. In Proceedings of the 33rd International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 8026–8037.
51. Hess, R. *Blender Foundations: The Essential Guide to Learning Blender 2.5*; Routledge: London, UK, 2013.
52. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. In Proceedings of the 9th International Conference on Learning Representations (ICLR), Virtual, 3–7 May 2021.
53. Kuffner, J.J.; LaValle, S.M. RRT-connect: An efficient approach to single-query path planning. In Proceedings of the 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation, San Francisco, CA, USA, 24–28 April 2000; Volume 2, pp. 995–1001.
54. Olson, E. AprilTag: A robust and flexible visual fiducial system. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 3400–3407.

55. Wen, B.; Yang, W.; Kautz, J.; Birchfield, S. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–21 June 2024; pp. 17868–17879.
56. You, Y.; Xiong, K.; Yang, Z.; Huang, Z.; Zhou, J.; Shi, R.; Fang, Z.; Harley, A.W.; Guibas, L.; Lu, C. PACE: A Large-Scale Dataset with Pose Annotations in Cluttered Environments. In Proceedings of the 18th European Conference on Computer Vision, Milan, Italy, 29 September–4 October 2024; Springer: Cham, Switzerland, 2025; pp. 473–489.