machines

MDPI

Article

Virtual and Real Occlusion Processing Method of Monocular Visual Assembly Scene Based on ORB-SLAM3

Hanzhong Xu [1], Chunping Chen [1], Qingqing Yin [1], Chao Ma [1] and Feiyan Guo [2,*]

[1] Shanghai Institute of Aerospace Technical Foundation, Shanghai 201109, China
[2] School of Mechanical Engineering, University of Science and Technology Beijing, Beijing 100083, China
* Correspondence: guofy@ustb.edu.cn

Abstract: Addressing the challenge of acquiring depth information in aero-engine assembly scenes using monocular vision, which complicates mixed reality (MR) virtual and real occlusion processing, we propose an ORB-SLAM3-based monocular vision assembly scene virtual and real occlusion processing method. The method proposes optimizing ORB-SLAM3 for matching and depth point reconstruction using the MNSTF algorithm. MNSTF can solve the problems of feature point extraction and matching in weakly textured and texture-less scenes by expressing the structure and texture information of the local images. It is then proposed to densify the sparse depth map using the double-three interpolation method, and the complete depth map of the real scene is created by combining the 3D model depth information in the process model. Finally, by comparing the depth values of each pixel point in the real and virtual scene depth maps, the virtual occlusion relationship of the assembly scene is correctly displayed. Experimental validation was performed with an aero-engine piping connector assembly scenario and by comparing it with Holynski's and Kinect's methods. The results showed that in terms of virtual and real occlusion accuracy, the average improvement was 2.2 and 3.4 pixel points, respectively. In terms of real-time performance, the real-time frame rate of this paper's method can reach 42.4 FPS, an improvement of 77.4% and 87.6%, respectively. This shows that the method in this paper has good performance in terms of the accuracy and timeliness of virtual and real occlusion. This study further demonstrates that the proposed method can effectively address the challenges of virtual and real occlusion processing in monocular vision within the context of mixed reality-assisted assembly processes.

Keywords: occlusion handling; mixed reality; ORB-SLAM3; assisted assembly

Academic Editor: Panagiotis Kyratsis

Received: 20 January 2025
Revised: 3 March 2025
Accepted: 4 March 2025
Published: 6 March 2025

Citation: Xu, H.; Chen, C.; Yin, Q.; Ma, C.; Guo, F. Virtual and Real Occlusion Processing Method of Monocular Visual Assembly Scene Based on ORB-SLAM3. *Machines* **2025**, *13*, 212. https://doi.org/10.3390/machines13030212

1. Introduction

The background of an aero-engine assembly site is highly complex, featuring numerous overlapping components and intricate spatial relationships. This complexity renders it challenging to accurately capture and process depth information. Existing methods often fail to handle these complexities due to their reliance on sparse depth maps or limited computational capabilities, resulting in inaccurate occlusion relationships and poor virtual–real fusion effects [1,2]. Mixed reality (MR)-assisted assembly guidance methods are widely used in the assisted assembly process of major equipment scenarios such as aero-engines [3–5]. After the virtual and real objects have finished tracking and registering, the virtual model or other guidance information can be correctly superimposed on the real scene position in MR glasses, but with the movement of the operator, the viewing angle of the MR glasses and the positional relationship between the virtual and real models

will change, and the occlusion relationship between them will also change. Assembly sites for other complex manufacturing tasks such as aero-engines, satellite manufacturing, automobile manufacturing, etc. have complex backgrounds and complex occlusion relationships between assembly objects [6–9]. The virtual–real occlusion processing method is crucial for the virtual–real fusion display of the assembly scene, and the correct virtual–real occlusion relationship can help the operator understand the guided information more easily when using the MR operation guidance assistance system [10]. Therefore, in the process of virtual–reality fusion for MR-assisted assembly of an aero-engine, the correct occlusion processing between virtual objects and objects in the real environment is a problem that needs to be solved urgently.

At present, there are three commonly used methods for real-image occlusion processing: real-image occlusion processing based on 3D model reconstruction [11], image depth information computation [12], and image features [13]. However, the virtual occlusion processing method based on 3D model reconstruction requires a large amount of preliminary 3D modeling of the assembly scene, which is a large amount of work and requires too much accuracy in model construction for such major and complex equipment as aero-engines. The advantage of the image feature-based virtual masking processing method is that it requires less equipment, but the disadvantage is also obvious: it needs the real assembly scene to have enough features to identify and detect, but the external accessories of an aero-engine, such major equipment, has the characteristics of a metal surface lacking textural features. Therefore, the method based on image depth information computation for virtual masking processing is more suitable for complex assembly scenes such as aero-engines. However, the traditional virtual masking processing method has poor robustness, low accuracy, and poor timeliness for this kind of assembly scenario.

The aero-engine assembly scene is characterized by a complex background, intricate relationships between assembly models, and an inability to use binocular vision cameras. To address the virtual and real occlusion problem in the MR-assisted assembly process of aero-engine external accessories, we propose a monocular vision-based virtual and real occlusion processing method using an improved ORB-SLAM3 framework. The proposed method reconstructs the depth points of the assembly scene using the enhanced ORB-SLAM3 algorithm, densifies the sparse depth map through bicubic interpolation, and integrates the depth information from the 3D model in the digitized process model to generate a complete depth map of the real scene. By comparing the depth values of each pixel in the real and virtual scene depth maps, our method accurately determines the spatial relationship between virtual and real models, correctly handles virtual–real occlusion, and optimizes the virtual–real fusion display effect for MR-assisted assembly of aero-engine external attachments.

- Propose a novel method based on ORB-SLAM3 for handling virtual–real occlusion in MR environments, specifically tailored for the complex assembly scenes of aero-engines.
- Propose the use of the MNSTF algorithm for matching optimization of ORB-SLAM3 and depth point reconstruction of assembly scenes, which is suitable for feature point extraction and matching in weakly textured or untextured regions.
- Propose a bicubic interpolation-based method to densify sparse depth maps and integrate them with the depth information from the 3D model in the digitized process model, generating a complete and accurate depth map of the real scene.

The rest of the paper is organized as follows. Section 2 presents an overview of the current state of domestic and international research on virtual occlusion processing. Section 3 describes the monocular vision virtual occlusion processing flow based on ORB-SLAM3. Section 4 describes the virtual occlusion rendering flow based on depth images. Section 5 uses an assembly scene of space engine piping connectors for experimental validation and analyzes the results.

2. Related Work

The virtual and real occlusion method based on depth image information calculation first acquires the depth information of the real object, and through the to-be-superimposed region of the virtual object, the depth information of the virtual object and the real object are compared and judged to obtain the correct occlusion relationship [14]. The current way of depth image information acquisition is mainly accomplished by binocular vision and monocular vision [15,16].

In terms of binocular vision to obtain depth image information, Kim [17] used the depth information of a real assembly scene to obtain the occlusion relationship between real and imaginary objects, searched for smooth and accurate parallax vector fields with clear object boundaries in binocular image pairs, and reconstructed the three-dimensional surface of the real scene using the vector fields to obtain the occlusion relationship between the real and imaginary models. Zhen [18] proposed a new algorithm based on binocular stereo vision (BSV) for recognition and depth estimation of inland vessels and proposed a sub-pixel-level feature point detection and matching algorithm based on the ORB algorithm, which further improved the density of image feature points and detection accuracy and was more conducive to the computation of the parallax value of the image. Yang [19] proposed a three-dimensional reconstruction system combining binoculars and depth cameras that effectively improved the accuracy of the three-dimensional reconstruction and accurately recognized the distance from the camera. Luo [20] proposed a method to improve the accurate acquisition of depth information by combining the multi-channel information of RGB-D, which can determine the depth relationship between the virtual object and the real object pixel point by pixel point. Zhang [21] realized the virtual masking processing of virtual and real scenes through the layering of the depth information and the optimization of the mapping.

The above research was based on a binocular vision sensor to obtain depth image information, which has certain performance requirements. With the current focus on lightweight devices and convenience [22], many scholars have begun to study monocular vision to obtain scene depth information of [23–25]. Simon [26] presented MonoNav, a fast 3D reconstruction and navigation stack that leverages recent advances in deep predictive neural networks to enable accurate 3D scene reconstruction from monocular images and pose streams. MonoNav uses off-the-shelf pretrained monocular depth estimation and fusion techniques to build maps. Chang [27] proposed a combination of monocular visual depth estimation and multiview depth estimation to complement the strengths of the two methods. Multiview depth is very accurate, but only for highly textured regions and high parallax values. Single-view depth captures the local structure of mesoscopic regions, including untextured regions, but the estimated depth lacks global consistency. Luo [28] proposed a deep neural network that can simultaneously estimate camera pose and reconstruct full-resolution depth information about the environment using only monocular continuous images. Fink [29] proposed taking a monocular depth map as input, scaling that depth map to absolute distances based on structure from motion, and converting the depth to a triangular surface mesh. This depth mesh is then refined in a local optimization that enforces photometric and geometric consistency. Zhang [30] presented a new unsupervised

learning framework for estimating scene depth and camera pose from video sequences that can be used as a basis for 3D reconstruction and augmented reality (AR). Li [31] used monocular SLAM to reconstruct an assembly scene, converted the reconstructed sparse 3D points into depth points in the depth map, and thickened the sparse map processing. Finally, the depth relationships between the real assembly scene, assembly objects, and the virtual model were compared. Holynski [32] proposed a new monocular depth estimation algorithm that propagates the sparse depth to each pixel, obtains the sparse depth of the key points using the SLAM method, and optimizes the propagation of the sparse depth with the image edges. The method can compute the depth map in near-real time.

In summary, when addressing the MR virtual–real occlusion problem in complex assembly scenarios, such as those of aero-engines, both binocular and monocular vision methods exhibit certain limitations in computational efficiency, occlusion accuracy, and adaptability to complex scenes. To reduce overreliance on hardware performance, a monocular vision approach is selected herein to acquire depth image information of an assembly site during MR virtual–real occlusion processing for a complex aero-engine assembly site.

3. Monocular Vision-Based Occlusion Handling Method

3.1. Virtual and Real Occlusion Handling Framework

Owing to the suboptimal performance of contemporary MR glasses, with many lacking binocular vision capabilities, the application of binocular vision algorithms is constrained. Consequently, in this study, we adopt a ORB-SLAM3 monocular vision-based virtual–reality occlusion processing method. In the domain of MR-assisted assembly, monocular vision technology offers notable advantages over depth-camera technology. Monocular vision technology obviates the need for specialized depth cameras, thus alleviating the weight burden of the overall hardware configuration. In this paper, considering the characteristics of MR-assisted assembly scenarios for aero-engine external attachments, a real-time virtual–reality occlusion processing method grounded in monocular vision is proposed. The processing flow framework is depicted in Figure 1.

The framework mainly describes how to fuse the virtual scene and the real scene using the improved ORB-SLAM3 monocular vision technique to realize the processing of virtual and real occlusion. First, the virtual camera data and virtual scene model are constructed from 3D digitized process model information, and then the virtual scene depth map is generated from these data. In the real scene, the real scene model and depth map are obtained by the improved ORB-SLAM3 sparse depth point reconstruction technique, and the depth map is densified. Next, the virtual scene depth map is matched with the real scene depth map to determine the occluded objects in the virtual scene. According to the depth map information of the virtual and real scenes, the depth value images of the virtual and real scenes are obtained by RGB pixel value filling, and then the values of the scenes are calculated and judged. The judgment rule is that if the depth value of the real scene is greater than the depth value of the virtual scene, then render the depth value of the virtual scene, and vice versa render the depth value of the real scene. Thus, the corresponding depth images are generated to realize the accurate rendering of dynamic assembly objects in the MR environment. Finally, in the MR-assisted assembly system for aero-engine external accessories, the depth information relationship between the real scene and the virtual scene is judged, and the final virtual–reality fusion scene is rendered and processed to obtain the final accurate virtual–reality fusion assembly scene.

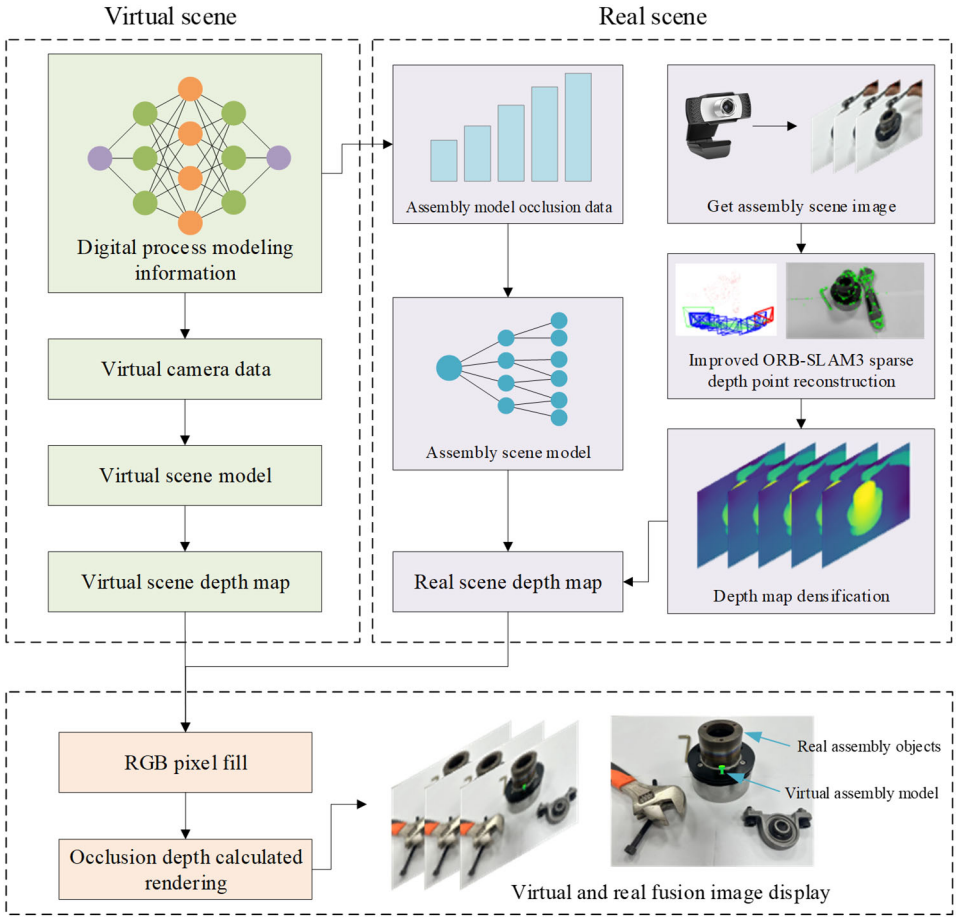


Figure 1. Monocular vision-based occlusion processing flow framework.

3.2. The Relationship Between Virtual and Real Object Occlusion

In the MR-assisted assembly system for aero-engine external accessories, the occlusion relationship between virtual and real objects represents a complex and crucial factor that directly impacts the efficiency of assisted assembly. As illustrated in Figure 2, the virtual–real object occlusion relationship is present in the aero-engine external attachment assembly scene. In the MR-assisted assembly system for aero-engine external accessories, virtual objects pertain to the visualization elements and virtual models derived from the 3D digital process model. These virtual objects are grounded in pre-established model data and projected into the real assembly environment via MR technology to furnish functions such as assembly guidance, information display, and process simulation. Virtual objects, relying on computer-generated 3D models, possess no physical entity in the real world. However, in the MR-assisted assembly scenario, they interact with real objects to form a virtual–real integrated working environment. Conversely, real objects denote the elements existing in the assembly scene, encompassing real assembly components (e.g., aero-engine parts, assemblies, and connectors) and real-world scenes (e.g., work platforms, workshop backgrounds, etc.). These real objects are physical entities with actual shapes, sizes, and positions during the assembly process, and their states change as the assembly operation progresses.

In MR-assisted assembly systems, the occlusion relationship between virtual and real objects is critical. This relationship must accurately reflect the physical occlusion in the real world to ensure that the operator receives correct visual guidance. For example, if a virtual object should be occluded by a real object in the real world, the virtual object must also appear behind the real object in the MR-assisted assembly scenario. By establishing precise

occlusion relationships, operators can better understand the assembly process and avoid errors caused by visual misdirection.

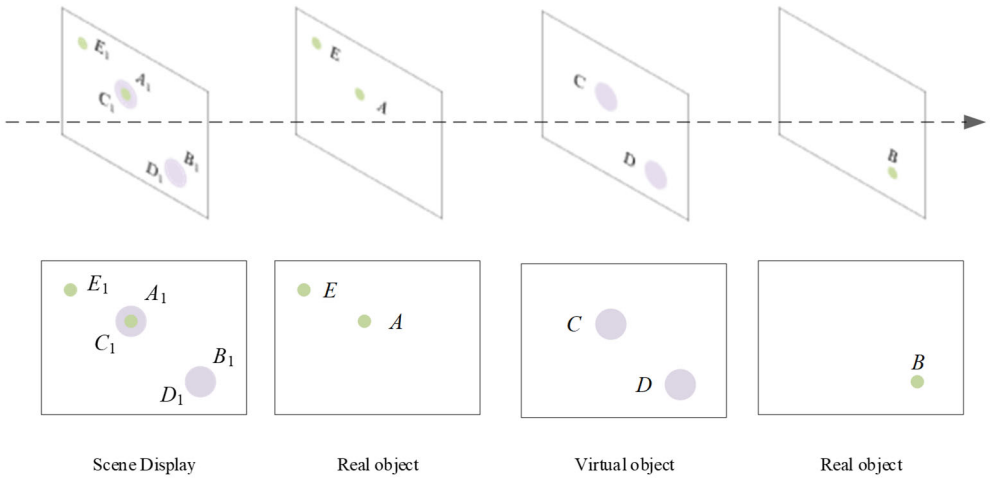


Figure 2. Schematic diagram of occlusion relationships.

4. Depth Image-Based Voxel Occlusion Rendering

4.1. Assembly Scene Sparse Depth Point Reconstruction

In the visual guidance process of aero-engine external accessory assembly, images captured by monocular vision systems lack depth information, which restricts their application in virtual occlusion processing. Nevertheless, the acquisition of depth information from monocular vision images can be achieved through the utilization of SLAM technology. Initially, the monocular camera mounted on the MR glasses must be calibrated to determine the camera's internal and external reference matrices K and M. This calibration involves the determination of parameters such as focal length, principal point position, and aberration coefficient. Subsequently, a sequence of images of the assembly scene is captured using the calibrated camera. These images will be employed for subsequent depth point reconstruction and voxel-masking processing.

The acquired images undergo feature extraction using the fast and rotated oriented brief—simultaneous localization and mapping (ORB-SLAM3) algorithm. ORB is an algorithm for image feature detection and description capable of rapidly and accurately extracting key points in an image. Feature points of neighboring images are matched, and feature trajectories are constructed from the matching results. This process facilitates the establishment of spatial relationships between consecutive image frames. Based on the matched feature points and feature trajectories, sparse depth points of the assembly scene are reconstructed using the triangulation principle. These depth points offer information regarding the approximate position and shape of objects within the scene. ORB-SLAM3 is a monocular vision-based SLAM algorithm that can estimate the camera pose and reconstruct the 3D structure of the environment from the video information captured by the monocular camera. As the monocular camera cannot directly measure depth, ORB-SLAM3 employs the structure from motion (SFM) method to estimate the depth information of points in the scene.

In SFM, when the same assembly scene is captured by the camera at different locations, the camera motion as well as the 3D position of the point P in the scene can be estimated by tracking the movement of the feature point between consecutive frames. The polar geometry constraint is the correspondence between two rays formed by the assembly scene space point P and the camera optical centers O_1 and O_2 in both views. Point P in the assembly scene space has projection points p_1 and p_2 in the camera views at two different

locations, and then the lines connecting these two points to the respective camera optical centers O_1 and O_2 intersect at one point in space. This is shown in Figure 3.

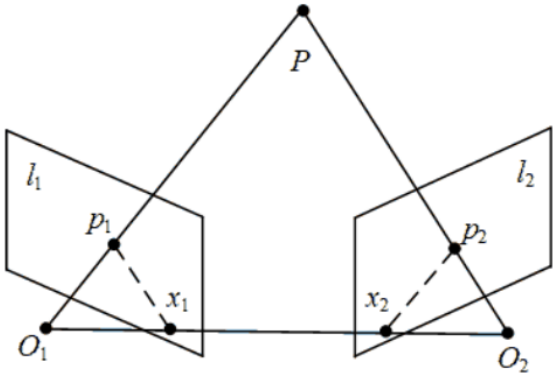


Figure 3. Feature point pair-pole geometry constraints.

The positions of p_1 and p_2 are calculated based on the camera's small-hole imaging model:

$$p_1 = K \cdot [R|t] \cdot P \tag{1}$$

$$p_2 = K \cdot [R'|t'] \cdot P \tag{2}$$

where K is the internal reference matrix of the camera containing the focal length and the position of the optical center. r, t, and R', t' denote the forward and backward rotation and translation matrices from the first camera coordinate system to the second camera coordinate system in the world coordinate system, respectively. The 3D coordinates of the spatial point P in the world coordinate system are the corresponding 2D-pixel coordinates.

For each pair of matching points, the following equations can be established based on their homogeneous coordinates and the projection relationship of the camera:

$$p_2^T F p_1 = 0 \tag{3}$$

where p_1 and p_2 are homogeneous coordinate vectors of the matching points in the first and second views, respectively, and F is the basis matrix.

The basis matrix F and the essence matrix E are of the form:

$$F = K^{-T} E K^{-1} \tag{4}$$

$$E = t \times R \tag{5}$$

The internal reference K is obtained by camera calibration, and the rotation matrix R and translation matrix t of the camera in the SLAM coordinate system can be obtained according to the above equations.

Considering only the translation along the baseline direction and ignoring the motion perpendicular to the baseline direction, the problem can be reduced to a two-dimensional problem. Therefore, the above system of equations can be simplified as:

$$x_1 = f\frac{X}{Z} \tag{6}$$

$$x_2 = f\frac{X + \Delta X}{Z + \Delta Z} \tag{7}$$

where x_1 and x_2 are the transverse coordinates of the projections of the point in the two views, ΔX is the amount of translation of the camera between the two positions along the

direction of the baseline, ΔZ is the amount of depth change of point P between the two views, and f is the focal length of the camera.

With the above system of equations, it is possible to solve for the depth value Z. First, the difference between the transverse coordinates of the two projected points is utilized to express ΔX:

$$\Delta x = x_2 - x_1 = f\left(\frac{X + \Delta X}{Z + \Delta Z} - \frac{X}{Z}\right) \tag{8}$$

This can be transformed into a linear equation about ΔZ:

$$f\frac{\Delta X}{Z + \Delta Z} - f\frac{\Delta X}{\Delta Z} = \Delta x - f\frac{X}{Z} + f\frac{X}{Z + \Delta Z} \tag{9}$$

Further simplification leads to:

$$\Delta Z = \frac{fZ^2}{f\Delta X - Z\Delta x} \tag{10}$$

where ΔX is the actual travel distance between the two camera positions, provided by the inertial measurement unit (IMU). Above, the depth value Z can be solved by solving for Δx according to Equation (10) and the focal length f obtained from the camera calibration.

4.2. Improved ORB-SLAM3 Feature Point Matching

The aero-engine and external accessories themselves have smooth surfaces, similar colors, and a lack of texture. When using the traditional ORB feature point matching, it shows some limitations: the correct point pairs to be matched become fewer, and it faces the problem of fewer matching points and increased false-matching rate. This will lead to incorrect or non-generation of depth maps. Therefore, to obtain enough matched point pairs in the aero-engine external accessory assembly scenario, in this paper, the ORB matching link in the ORB-SLAM3 algorithm is improved. Using the multi-neighborhood structure tensor feature (MNSTF) algorithm, which is an optimization method for feature point matching [33], the matching accuracy is improved by considering the local structure information around the feature points, and it is suitable for feature point extraction in weakly textured or untextured regions and matching.

The core of the MNSTF algorithm is to extract structural tensor features in multiple neighborhoods of the feature points and combine these features for matching. The following are the steps in the computational process of the MNSTF algorithm.

For each feature point pi after each matching, define K neighborhoods of different scales, each containing a certain number of neighboring points. These neighborhoods can be circular, square, or other shapes. For each neighborhood, the structure tensor T_j is computed, which is a tensor reflecting the distribution of points within the neighborhood of second or higher order. The structure tensor can be computed from the positional information of the points within the neighborhood.

$$T_j(k) = \sum_{p \in N_j} w(p, p_i) \cdot (p - p_i) \otimes (p - p_i) \tag{11}$$

Here, N_j is the set of points in the j neighborhood, p is a point in the neighborhood, w (p, p_i) is a weight function to adjust the contribution of each point to the structure tensor, and $\otimes$ denotes the tensor product.

The structural tensor features of each neighborhood are combined to construct a comprehensive feature descriptor D_i for representing the local structural information of the feature point p_i:

$$D_i = \{T_1(p_i), T_2(p_i), \ldots, T_K(p_i)\} \tag{12}$$

For all feature point pairs (p_i, p_j) in both images, their descriptors D_i and D_j are used to compute the similarity score S_{ij}:

$$S_{ij} = \text{Similarity}(D_i, D_j) \tag{13}$$

where similarity is the similarity measure based on the similarity score S_{ij}. The best-matching pairs are selected and the matching results are further optimized by the random sample consensus (RANSAC) robust method to eliminate outliers and improve the accuracy of matching.

The modified ORB-SLAM3 algorithm is used to construct a sparse depth point cloud in the aero-engine external accessory MR-assisted assembly system for virtual occlusion processing. The specific process is as follows. First, images of the assembly scene from multiple angles are captured by a moving camera. Next, feature points in neighboring images are matched to determine their correspondence in space. The result of the matching forms a feature trajectory, which describes the motion trajectory of the feature points in consecutive image frames. Using the principle of triangulation, the 3D positions of the objects in the scene can be calculated from the feature points and the feature trajectory. Finally, all the calculated 3D position points are combined to form a sparse depth point cloud, and the corresponding sparse depth image is generated. The detected feature points and camera positions of the assembled scene are shown in Figure 4, where the data of the 10th, 20th, 30th, 50th, 70th, 100th, 125th, and 150th frames are shown.

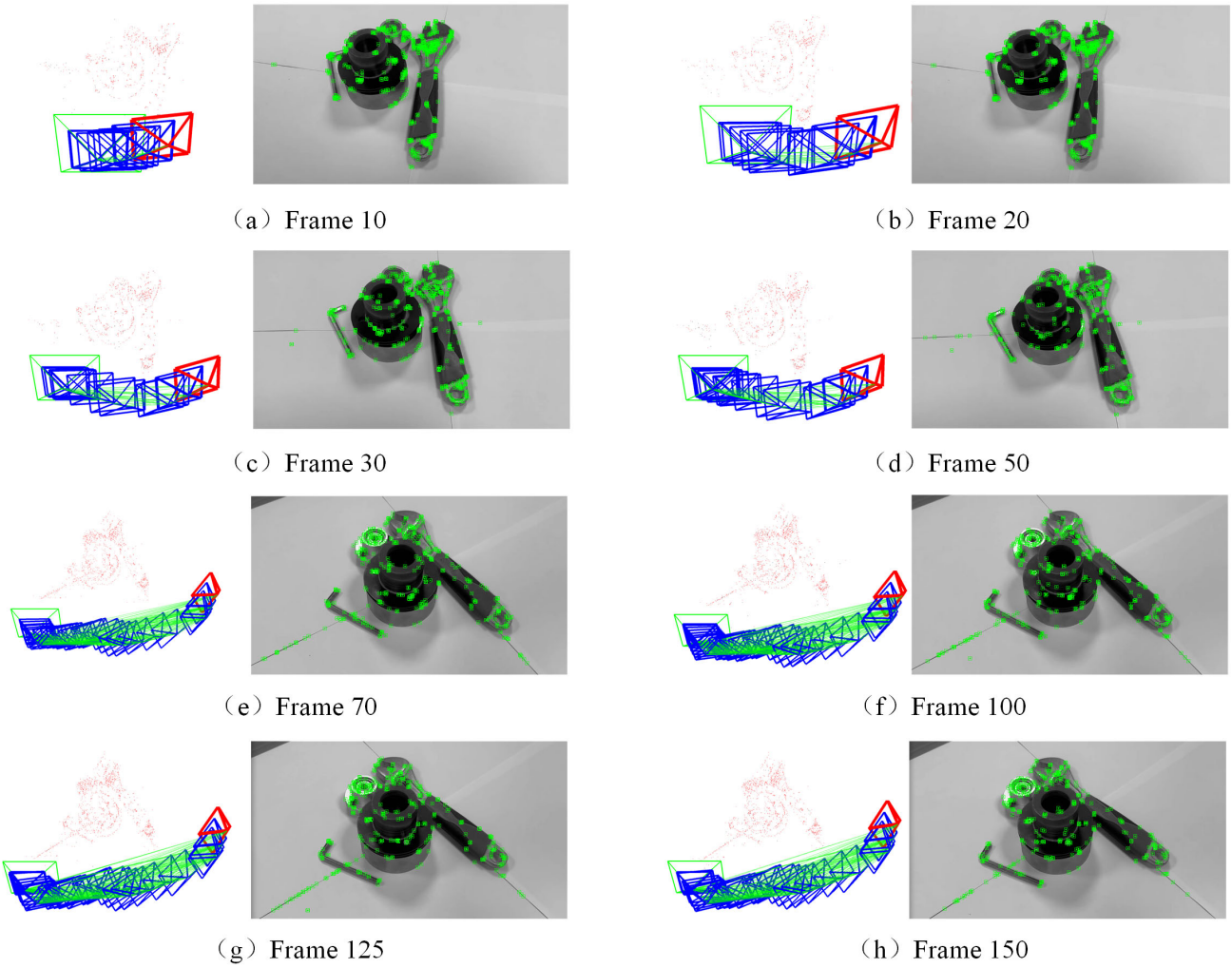
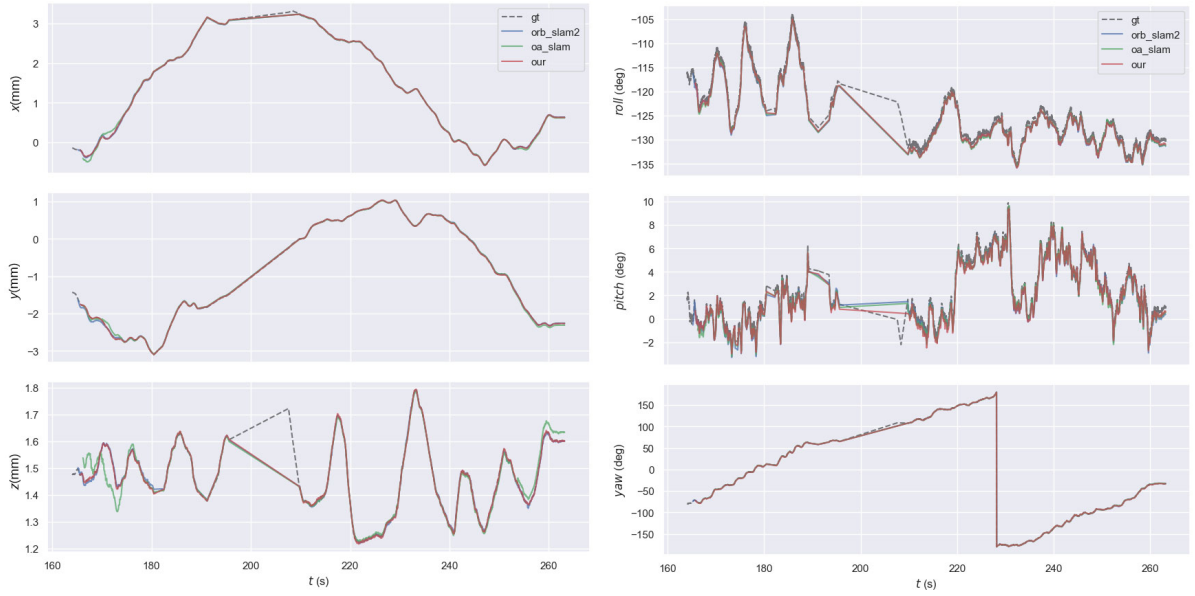


Figure 4. Key points and camera positions.

Figure 5 shows the comparison of the localization effect of different algorithms for camera tracking, and the 3D trajectories are extracted by quantitative analysis of the camera position data. This is compared and analyzed with the 3D trajectories of other algorithms. As can be seen in Figure 5a, b, in the comparison of the trajectory translations and rotations, the highest error of the improved ORB-SLAM3 algorithm proposed in this paper is 3.2 mm and the smallest is only 0.42 mm, with an average accuracy error of 1.6 mm, while the average errors of the other two algorithms, ORB-SLAM and ORB-SLAM2, are 2.3 mm and 2.1 mm. It is proved that the proposed algorithm in this paper has higher localization accuracy compared with other algorithms. This is also mainly due to the improvement in the matching algorithm in this paper, which makes the number of matched pairs of points increase and the correct rate of matching increase such that it obtains higher accuracy.



(**a**) (**b**)

Figure 5. Comparison of tracking and positioning effects of different algorithms. (**a**) Comparison of trajectory translations; (**b**) Comparison of trajectory rotations.

4.3. Depth Map Densification

The matching method of ORB-SLAM3 is improved by the MNSTF algorithm, which makes the matched correct point pairs more uniform and accurate. Next, the obtained sparse depth map needs to be densified. The densification process used in this paper utilizes the bilinear interpolation (BI) method. Using the weighted average of the 16 surrounding pixel points to calculate the value of the new pixel provides smoother and more accurate results when the image is zoomed in or out.

It is necessary to compute the pixel value d of the point (x, y) in the new image I_{new}, which has the floating-point coordinates (x', y') in the original image and can be found surrounded by 16-pixel points with coordinates (x_i, y_j), where i and j range from 0 to 3, respectively, and each of the pixel points has a corresponding depth value $d_{i,j}$.

Next, the weighted value of each pixel point in this 4×4 grid is calculated:

$$d(x', y') = (1-u)(1-v)d_{0,0} + u(1-v)d_{1,0} + (1-u)vd_{0,1} + uvd_{1,1} + \ldots + (1-u)u)(1-v)d_{3,3} \quad (14)$$

where $u = x' - \lfloor x' \rfloor$ and $v = y' - \lfloor y' \rfloor$ are the positions of the new pixel points (x, y) concerning the nearest integer pixel point. $f(x, y)$ is the pixel value of point (x, y) in the original image.

Two interpolations were performed on the formula. Interpolating each y-value in the x-direction yields a one-dimensional interpolation result:

$$g(y') = \sum_{3}^{i=0} f(i, y0) B_{i,1}(u) + \sum_{3}^{i=0} f(i, y1) B_{i,2}(u) + \sum_{3}^{i=0} f(i, y2) B_{i,3}(u) + \sum_{3}^{i=0} f(i, y3) B_{i,4}(u) \quad (15)$$

Next, the one-dimensional interpolation results obtained above are interpolated in the y-direction to obtain the final pixel values:

$$h(x') = \sum_{3}^{j=0} g(y0) B_{j,1}(v) + \sum_{3}^{j=0} g(y1) B_{j,2}(v) + \sum_{3}^{j=0} g(y2) B_{j,3}(v) + \sum_{3}^{j=0} g(y3) B_{j,4}(v) \quad (16)$$

where $B_{i,k}(u)$ and $B_{j,k}(v)$ are one-dimensional bicubic interpolation basis functions corresponding to the u and v directions, respectively. Through the above process, the pixel value $h(x')$ of the new pixel point (x, y) can be obtained. This process needs to be repeated for each new pixel point until the entire image is interpolated. Bicubic interpolation [34] provides smoother and more natural results than bilinear interpolation [35] and trilinear interpolation [36].

The densified depth image can be obtained by using the method based on double-trilinear interpolation, as shown in Figure 6. Figure 6a shows the original RGB images of different scenes and Figure 6b shows the obtained densified depth image. The depth image provides the depth information of each pixel point in the scene, which reflects the distance and positional relationship between the objects in the scene and expresses the process of depth from near too far in yellow and blue. Figure 6c–f shows the results of the densified depth maps for different assembly scenes, and the use of the method based on bi-trilinear interpolation has the characteristics of high resolution, smooth transition, local detail preservation, adaptability, computational efficiency, and accuracy in generating the densified depth images. These features enable the method to provide accurate depth information in MR-assisted assembly systems for aero-engine external accessories.

4.4. Assembly Scene Occlusion Rendering

To realize accurate virtual and real occlusion effects in MR-assisted assembly of external attachments of an aero-engine, real scene depth maps and virtual scene depth maps are obtained according to the above method. These maps are used to represent the depth information at different locations in the scene. A uniform coordinate system is used to process these two scene depth maps. This helps to convert the different depth maps to the same reference frame for subsequent processing, as shown in Figure 7. During the virtual occlusion rendering process, the depth maps of the real scene are computed using an improved algorithm based on ORB-SLAM3. It is ensured that these depth maps are consistent with the depth map coordinate system of the virtual scene model in the same viewpoint. Next, for each pixel point, the depth values of the corresponding positions in the two depth maps are compared to determine the front–back relationship between the real and virtual objects. If the depth value of the real pixel point is greater than that of the virtual point, it indicates that the virtual object is closer to the camera, and therefore the pixel value of the virtual pixel point should be rendered and displayed. Conversely, if the depth value of the real pixel point is less than or equal to that of the virtual point, the pixel value of the real pixel point should be rendered. Finally, the final rendered image is presented by traversing all the pixel points on the image and merging them for display.

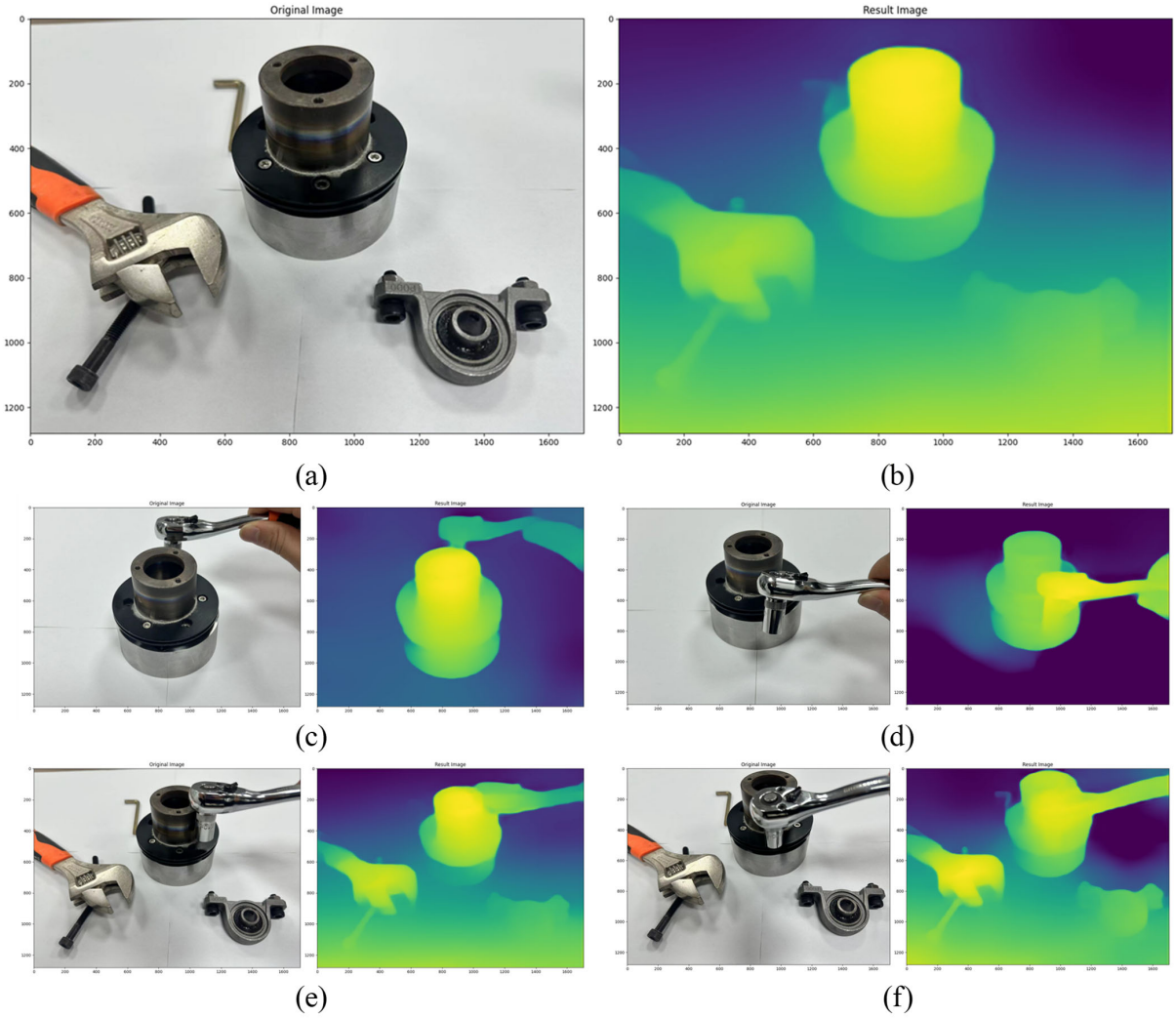


Figure 6. RGB image and depth diagram of the assembly scene. (**a**) Original RGB images; (**b**) Densified depth image; (**c**–**f**) Results of the densified depth maps for different assembly scenes.

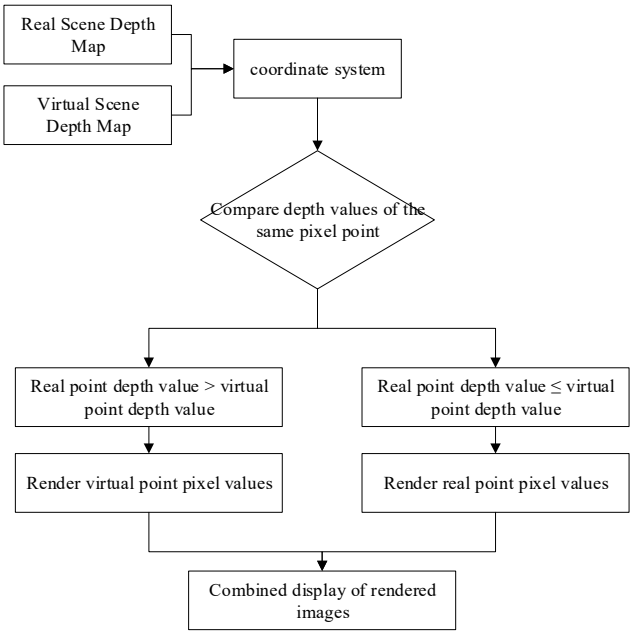


Figure 7. Rendering process for virtual–real occlusion of assembly objects.

5. Case Study

To test the effectiveness of the false-real occlusion processing method based on improved ORB-SLAM3 monocular vision proposed in this paper, this section compares and analyzes it with a variety of algorithms in terms of false and real occlusion effect, accuracy, and timeliness. The computer configuration for this experiment was a Windows 11 operating system, i5-13500H, 32 GB RAM, Nvidia RTX4050 (8 GB graphics memory), and the image transmission used a USB mobile camera with the resolution set to 680 × 480. Taking the assembly of the external accessory piping connectors of the aero-engine as an example, the MR-assisted assembly process of the external accessory piping using mutual virtual and real occlusion is studied. The pipeline attachment assembly mainly includes the pipeline base body, assembly wrench, small bolt, small nut, and other parts. According to the test requirements, a 3D digitized model is designed to obtain the depth map information of the external accessory piping model. The test process involves the digitized process manual of MR-assisted assembly of the external accessory pipeline and associates the involved model information with the corresponding nodes to obtain the virtual and real mutual occlusion data about the pipeline-assisted assembly information.

5.1. Virtual and Real Occlusion Effects

In the MR-assisted assembly process of the external accessories of the aero-engine, the effect of virtual–reality fusion for the static virtual model and the real scene is shown in Figure 8. In the case of no virtual–real masking, the masking relationship between the assembly object and the virtual model is not obvious. This may lead to misunderstandings or errors in the assembly process. After the virtual–real occlusion process, the occlusion relationship between the assembly objects and the virtual model is more obvious. This means that in MR-assisted assembly systems, when a real object is in the way of a virtual object, the virtual object will appear behind the real object and vice versa. This processing accurately reflects the physical occlusion situation in the real world, provides accurate visual guidance to the operator, and helps to avoid assembly errors.

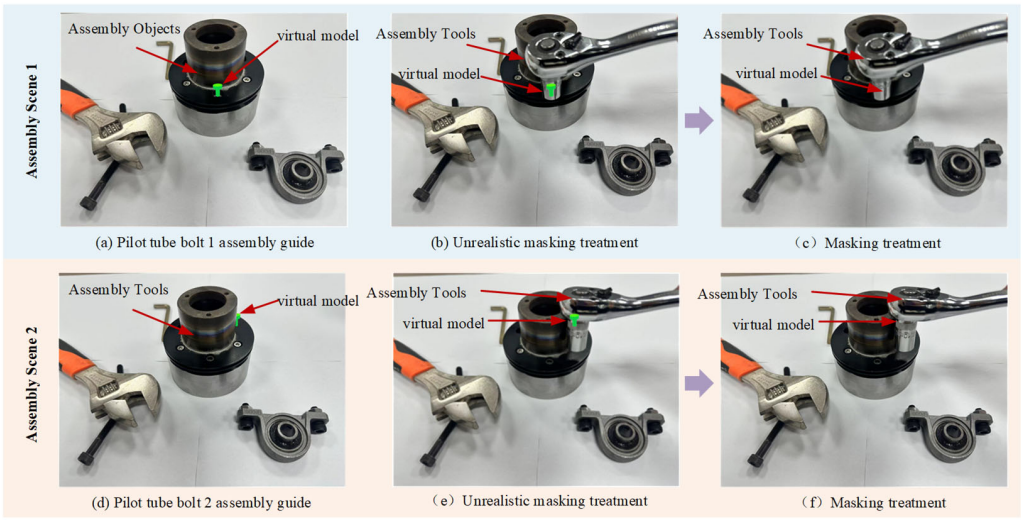


Figure 8. Aero-engine external accessories exhaust pipe bolt virtual model and scene fusion.

Figure 8 shows the guided assembly process of the pilot tube bolts, and Figure 8a and 8b show the guided assembly process of pilot tube bolts 1 and 2, respectively. These two figures are in the assembly process of the typical occlusion relationship intercepted for the virtual model and the real scene superimposed analysis. When the bolts are overlaid with the virtual model and the real scene, it can be seen in Figure 8b that the bolts are located at

the very front of the scene. The normal situation should be the location of the bolt hole, but when the assembly tool wrench is positioned over it, the bolt is still located in the front of the scene, which is not processed by the virtual reality masking. As shown in Figure 8c, when the bolt is superimposed on the virtual model and the real scene, firstly the wrench can be seen blocking the virtual model, and secondly the real scene is also blocking part of the virtual model of the bolt. Figure 8e,f presents the relationships before and after the virtual and real occlusion treatments during the assembly of pilot tube bolt 2. From the results, these occlusion relationships are consistent with the real occlusion relationships, which proves the effectiveness of the virtual model and real scene-based virtual occlusion processing algorithm proposed in this paper, and further verifies the reliability of the method in this paper.

5.2. Accuracy Analysis of Virtual and Real Occlusion

To analyze the accuracy of the proposed algorithms in aero-engine external attachment assembly scenarios in terms of virtual and real occlusion, the depth image contours computed by different algorithms in aero-engine external attachments were analyzed by comparing them with the real contour values and comparing with Holynski's [32] and Kinect's [37] depth camera methods. Both methods could generate depth maps of assembly scenes in previous studies and have achieved good results. To compare with the depth images in this paper, the contour line of the connecting tube of the external attachment assembly of the aero-engine in Figure 9a is taken as an example. Each time, the contour line is sampled by uniformly selecting 30 sampling points. Each of the three methods is tested five times, and the average value is taken as the value of the sampling points for comparative analysis.

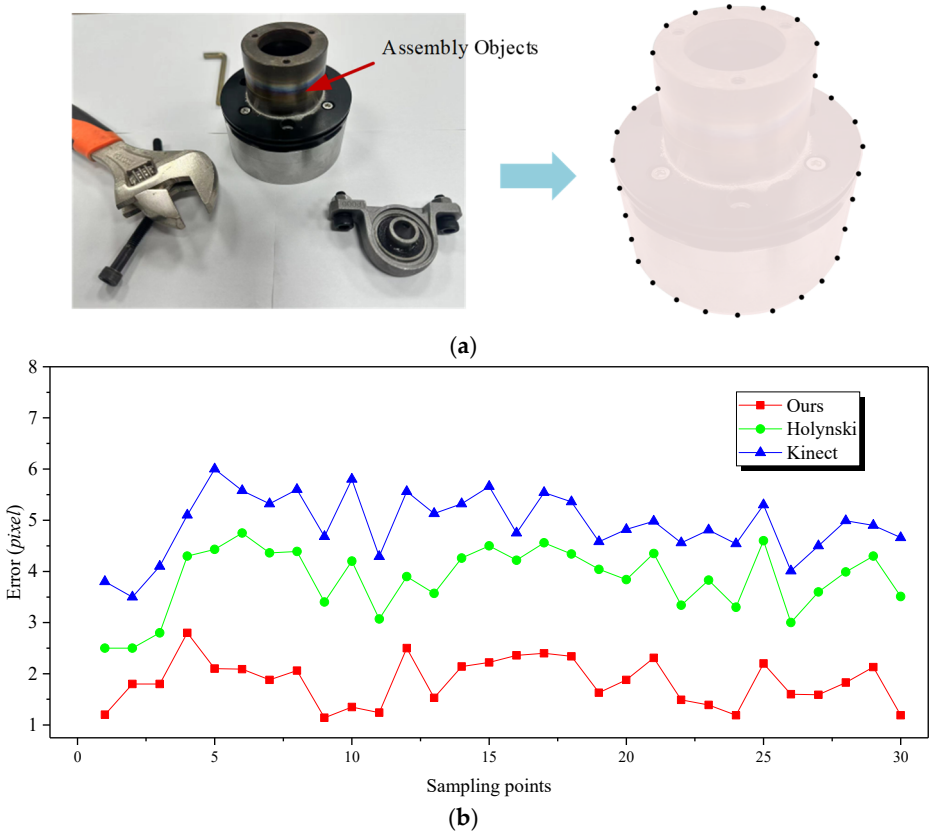


Figure 9. Comparison of edge errors in depth images. (**a**) Schematic diagram of the accessory connecting tube and its contour extraction. (**b**) Depth map sampling point error.

Figure 9b shows the error curves of different algorithms for the depth image sampling points, from which the error values of Holynski [32] and Kinect [37] methods are relatively high, with an average error of 3.9 pixels and 5.1 pixels, respectively. In contrast, the algorithm based on the improved ORB-SLAM3 method proposed in this paper has an average error of 1.7 pixels. Mainly, the matching algorithm with improved ORB is used in the depth image processing process, which can identify and extract the contour feature points of the connecting tube more accurately. By optimizing the descriptors and matching strategies of the feature points, the accuracy of feature point matching is improved and false matching is reduced. Moreover, it has better robustness in dealing with environmental factors such as light change and perspective change and can work stably under different conditions to obtain lower error values.

5.3. Timeliness Analysis of Occlusion

Timeliness of occlusion refers to the speed and efficiency of the algorithm to process the real-void occlusion depth image, which is crucial for the MR-assisted assembly guidance system of aero-engine external accessories because it is directly related to the timeliness of the whole MR guidance system in rendering the before-and-after relationship of the real-void model, which affects the real-time performance and experience of the operator on the MR guidance system. To verify the real-time performance of the virtual–reality occlusion processing algorithm proposed in this paper, the virtual–reality occlusion processing algorithms of Li [31], Holynski [32], and Kinect [37] are compared and analyzed in terms of the processing time and running real-time frame rate. We performed 30 repetitions of the three methods in our experiments and calculated the mean, standard deviation, and 95% confidence intervals for processing time and frame rate.

Table 1 shows the processing time and real-time frame rate of different methods. From the table, the processing time of the virtual and real masking algorithms proposed by Holynski [32] and Kinect [37] are 41.8 ms $\pm$ 1.2 ms and 44.2 $\pm$ 1.5 ms, respectively, and the frame rates are 23.9 $\pm$ FPS and 22.6 $\pm$ FPS. These two methods need to perform preprocessing of the acquired image, which affects the efficiency of the subsequent processing, thus increasing the computation time. Li [31] proposed a method that improves ORB-SLAM2 to achieve better real-time results, with a processing time of 28.9 $\pm$ 0.9 ms and a real-time frame rate of 34.6 $\pm$ 1.1 FPS, while the method proposed in this chapter has an operation time of 23.6 $\pm$ 0.7 ms and a real-time frame rate of greater than 42 $\pm$ 1.3 FPS, which improves the real-time frame rate over the methods of Holynski and Kinect by 77.4% and 87.6%, respectively. The algorithm can effectively improve the efficiency of depth image processing, mainly since the method in this paper utilizes the MNSTF algorithm for optimization in feature point extraction matching. This method can quickly eliminate the wrong matching points, reduce the computation amount for feature point matching, and improve the processing efficiency of the algorithm for the assembly scene image. Thus, it can provide a higher real-time frame rate for the occlusion rendering of the depth map of the assembly scene and meet the real-time requirements of the MR-assisted assembly guidance system for the external accessories of the aero-engine.

Table 1. Processing time and real-time frame rate of different methods.

Methods	Holynski [32]	Azure Kinect [37]	Li [31]	Ours
Time (ms)	41.8 $\pm$ 1.2	44.2 $\pm$ 1.5	28.9 $\pm$ 0.9	23.6 $\pm$ 0.7
Frame (FPS)	23.9 $\pm$ 0.8	22.6 $\pm$ 0.7	34.6 $\pm$ 1.1	42.4 $\pm$ 1.3

6. Conclusions

To address the challenge of obtaining depth information for the assembly scene of aero-engine external accessories via monocular vision, which in turn hinders virtual–real occlusion processing, this paper presents an MR-assisted assembly virtual–real occlusion processing method grounded in ORB-SLAM3 monocular vision technology. This method aims to resolve the virtual–real occlusion issue in the MR-assisted assembly of aero-engine external accessories. The main conclusions of this paper are as follows.

- By incorporating the MNSTF algorithm, we enhanced the feature matching and optimization capabilities of ORB-SLAM3, enabling more accurate reconstruction of the assembly scene with sparse depth points. This improvement significantly reduces computational overhead while maintaining high precision in depth estimation.
- Our method compares the depth values of each pixel in the real and virtual scene depth maps to determine the spatial relationship between virtual and real objects. This ensures accurate occlusion handling and optimizes the visual fusion effect in MR-assisted assembly scenarios.
- The MR-assisted assembly guidance process of aero-engine piping connectors was used as experimental validation and compared with the methods of Holynski and Kinect, and the results show that the method in this paper can effectively solve the problem of dealing with the real–virtual occlusion in the MR-assisted assembly process. The proposed method of false-reality occlusion processing based on improved ORB-SLAM3 monocular vision performs well in terms of false-reality occlusion effect, accuracy, and timeliness.

The work in this paper still has some limitations, such as inaccuracies in capturing depth information for parts smaller than 20 mm. With the rapid development of deep learning models, future work will consider leveraging deep learning techniques to recognize and supplement depth information for small target parts. Deep learning is particularly well suited for this task due to its ability to learn complex patterns and features from large datasets, which can enhance the accuracy of depth estimation for small and intricate structures. Additionally, deep learning can be integrated with the existing ORB-SLAM3 framework to refine sparse depth maps and improve the overall robustness of the system. This hybrid approach has the potential to significantly enhance the accuracy and reliability of depth information capture for small parts, addressing the current limitations of the proposed method.

Author Contributions: Conceptualization, H.X. and F.G.; methodology, Q.Y.; software, C.M.; validation, C.C. and H.X.; writing—original draft preparation, H.X.; writing—review and editing, H.X. and F.G.; funding acquisition, F.G. and C.M All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Shanghai Specialized Program for Promoting High-Quality Development (2023-GZL-RGZN-01024), National Defense Industrial Technology Development Program of China (JCKY2023205B006), and National Natural Science Foundation of China (52175450).

Data Availability Statement: The data are contained within the article.

Acknowledgments: The authors gratefully acknowledge the support of the Intelligent Interactive Innovation Studio of Shanghai Aerospace Technology Foundation for the experimental conditions.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Mal, F.; Karaman, S. Sparse-to-Dense: Depth Prediction from Sparse Depth Samples and a Single Image. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018.
2. Fu, H.; Gong, M.; Wang, C.; Batmanghelich, K.; Tao, D. Deep Ordinal Regression Network for Monocular Depth Estimation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 18–23 June 2018.
3. Raj, S.; Murthy, L.R.; Shanmugam, T.A.; Kumar, G.; Chakrabarti, A.; Biswas, P. Augmented reality and deep learning based system for assisting assembly process. *J. Multimodal User Interfaces* **2024**, *18*, 119–133. [CrossRef]
4. Yuan, M.L.; Ong, S.K.; Nee, A.Y.C. Augmented reality for assembly guidance using a virtual interactive tool. *Int. J. Prod. Res.* **2008**, *46*, 1745–1767. [CrossRef]
5. Subramanian, K.; Thomas, L.; Sahin, M.; Sahin, F. Supporting Human-Robot Interaction in Manufacturing with Augmented Reality and Effective Human–Computer Interaction: A Review and Framework. *Machines* **2024**, *12*, 706. [CrossRef]
6. Mu, X.; Wang, Y.; Yuan, B.; Sun, W.; Liu, C.; Sun, Q. A New assembly precision prediction method of aeroengine high-pressure rotor system considering manufacturing error and deformation of parts. *J. Manuf. Syst.* **2021**, *61*, 112–124. [CrossRef]
7. Li, J.; Wang, S.; Wang, G.; Zhang, J.; Feng, S.; Xiao, Y.; Wu, S. The effects of complex assembly task type and assembly experience on users' demands for augmented reality instructions. *Int. J. Adv. Manuf. Technol.* **2024**, *131*, 1479–1496. [CrossRef]
8. Patricio, A.; Valente, J.; Dehban, A.; Cadilha, I.; Reis, D.; Ventura, R. AI-Powered Augmented Reality for Satellite Assembly, Integration and Test. *arXiv* **2024**, arXiv:2409.18101.
9. Wolfartsberger, J.; Hallewell Haslwanter, J.D.; Lindorfer, R. Perspectives on Assistive Systems for Manual Assembly Tasks in Industry. *Technologies* **2019**, *7*, 12. [CrossRef]
10. Li, W.; Wang, J.; Liu, M.; Zhao, S.; Ding, X. Integrated registration and occlusion handling based on deep learning for augmented-reality-assisted assembly instruction. *IEEE Trans. Ind. Inform.* **2022**, *19*, 6825–6835. [CrossRef]
11. Tian, Y.; Long, Y.; Xia, D.; Yao, H.; Zhang, J. Handling occlusions in augmented reality based on 3D reconstruction method. *Neurocomputing* **2015**, *156*, 96–104. [CrossRef]
12. Zhu, J.; Pan, Z.; Sun, C.; Chen, W. Handling occlusions in video-based augmented reality using depth information. Computer Animation and Virtual Worlds. *Comput. Animat. Virtual Worlds* **2010**, *21*, 509–521. [CrossRef]
13. Hayashi, K.; Hirokazu, K.; Shogo, N. Occlusion Detection of Real Objects Using Contour Based Stereo Matching. In Proceedings of the ICAT05: The International Conference on Augmented Tele-Existence, Christchurch, New Zealand, 5–8 December 2005; pp. 180–186.
14. Walton, D.R.; Steed, A. Accurate Real-Time Occlusion for Mixed Reality. In Proceedings of the VRST' 17: 23rd ACM Symposium on Virtual Reality Software and Technology, Gothenburg, Sweden, 8–10 November 2017; pp. 1–10.
15. Lee, J.H.; Kim, C.S. Monocular depth estimation using relative depth maps. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 9729–9738.
16. Liu, F.; Zhou, S.; Wang, Y.; Hou, G.; Sun, Z.; Tan, T. Binocular light-field: Imaging theory and occlusion-robust depth perception application. *IEEE Trans. Image Process.* **2019**, *29*, 1628–1640. [CrossRef] [PubMed]
17. Kim, H.; Yang, S.J.; Sohn, K. 3D reconstruction of stereo images for interaction between real and virtual worlds. In Proceedings of the Second IEEE and ACM International Symposium on Mixed and Augmented Reality, Tokyo, Japan, 10 October 2003.
18. Zheng, Y.; Liu, P.; Qian, L.; Qin, S.; Liu, X.; Ma, Y.; Cheng, G. Recognition and depth estimation of ships based on binocular stereo vision. *J. Mar. Sci. Eng.* **2022**, *10*, 1153. [CrossRef]
19. Yang, Y.; Meng, X.; Gao, M. Vision system of mobile robot combining binocular and depth cameras. *J. Sens.* **2017**, *2017*, 4562934. [CrossRef]
20. Luo, T.; Liu, Z.; Pan, Z.; Zhang, M. A virtual-real occlusion method based on GPU acceleration for MR. In Proceedings of the 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Osaka, Japan, 23–27 March 2019; pp. 1068–1069.
21. Zhang, C.; Xu, R.C.; Han, C.; Zhai, H.Y. An Occlusion Consistency Processing Method Based on Virtual-Real Fusion. In *Frontier Research and Innovation in Optoelectronics Technology and Industry*; CRC Press: Parkway, NW, USA, 2018; pp. 47–56.
22. Ibrahim, M.M.; Liu, Q.; Khan, R.; Yang, J.; Adeli, E.; Yang, Y. Depth map artefacts reduction: A review. *IET Image Process.* **2020**, *14*, 2630–2644. [CrossRef]
23. Simon, N.; Majumdar, A. Mononav: Mav navigation via monocular depth estimation and reconstruction. In Proceedings of the 18th International Symposium on Experimental Robotics (ISER 2023), Chiang Mai, Thailand, 26–30 November 2023; Springer Nature: Cham, Switzerland, 2023; pp. 415–426.
24. Chaplot, D.S.; Gandhi, D.; Gupta, S.; Gupta, A.; Salakhutdinov, R. Learning to explore using active neural slam. *arXiv* **2022**, arXiv:2004.05155.
25. Chaplot, D.S.; Salakhutdinov, R.; Gupta, A.; Gupta, S. Neural topological slam for visual navigation. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 12875–12884.

26. Muravyev, K.; Bokovoy, A.; Yakovlev, K. tx2_fcnn_node: An open-source ROS compatible tool for monocular depth reconstruction. *SoftwareX* **2022**, *17*, 100956. [CrossRef]
27. Chang, M.F. Monocular Depth Reconstruction Using Geometry and Deep Convolutional Networks. Master's Thesis, Carnegie Mellon University, Pittsburgh, PA, USA, May 2018.
28. Luo, Y.; Liu, G.; Liu, H.; Liu, T.; Tian, G.; Ji, Z. Simultaneous Monocular Visual Odometry and Depth Reconstruction with Scale Recover. In Proceedings of the 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), Dali, China, 6–8 December 2019; pp. 682–687.
29. Fink, L.; Franke, L.; Keinert, J.; Stamminger, M. Refinement of Monocular Depth Maps via Multi-View Differentiable Rendering. *arXiv* **2024**, arXiv:2410.03861.
30. Zhang, X.; Zhao, B.; Yao, J.; Wu, G. Unsupervised monocular depth and camera pose estimation with multiple masks and geometric consistency constraints. *Sensors* **2023**, *23*, 5329. [CrossRef]
31. Li, W.; Wang, J.; Liu, M.; Zhao, S. Real-time occlusion handling for augmented reality assistance assembly systems with monocular images. *J. Manuf. Syst.* **2022**, *62*, 561–574. [CrossRef]
32. Holynski, A.; Kopf, J. Fast depth densification for occlusion-aware augmented reality. *Trans. Graph.* **2018**, *37*, 1–11. [CrossRef]
33. Han, X.; Chen, X.; Deng, H.; Wan, P.; Li, J. Point Cloud Deep Learning Network Based on Local Domain Multi-Level Feature. *Appl. Sci.* **2023**, *13*, 10804. [CrossRef]
34. Dengwen, Z. An Edge-Directed Bicubic Interpolation Algorithm. In Proceedings of the 2010 3rd International Congress on Image and Signal Processing, Yantai, China, 16–18 October 2010; Volume 3, pp. 1186–1189.
35. Kirkland, E.J.; Kirkland, E.J. Bilinear Interpolation. In *Advanced Computing in Electron Microscopy*; Springer: Boston, MA, USA, 2010; pp. 261–263.
36. Bai, Y.; Wang, D. On the comparison of trilinear, cubic spline, and fuzzy interpolation methods in the high-accuracy measurements. *IEEE Trans. Fuzzy Syst.* **2010**, *18*, 1016–1022.
37. Azure Kinect DK. Available online: https://learn.microsoft.com/zh-cn/azure/kinect-dk/depth-camera (accessed on 15 May 2024).