Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 12, No. 2, 2024

Augmenting Three-Dimensional Model Annotation System with Enhanced Reality

Yan Luo [1], Tianxiu Lu [2, *], Weihan Zhang [1], Suiqun Li [1] and Xuefeng Wang [1]

[1] School of Automation and Information Engineering, Sichuan University of Science & Engineering, Zigong 643002, China
[2] College of Mathematics and Statistics, Sichuan University of Science & Engineering, Zigong 643002, China

[*] **Corresponding author**: Tianxiu Lu

Abstract: This study proposes an augmented reality-based three-dimensional model annotation system, integrating cloud anchors, three-dimensional reconstruction, and augmented reality technology to achieve explicit three-dimensional annotations on models. Employing an improved ORB algorithm, the annotated model is persistently anchored in three-dimensional space through cloud anchors, presenting accurate spatial information and showcasing the depth of scenes and relationships between elements. The system supports multiple data types for annotations, such as text and images. Through a comparison with traditional two-dimensional annotation in a drone experiment, the system demonstrates higher experimental efficiency, providing more intuitive annotation guidance and enhancing remote guidance efficiency and user understanding of drones.

Keywords: Cloud Anchor; 3D reconstruction; Model Annotation; Augmented Reality.

1. Introduction

With the development of mobile device hardware and computer network technology, video communication has provided an effective solution for remote annotation, breaking geographical limitations and offering real-time audio-visual communication channels for dispersed teams. However, traditional annotation tools have limitations in information transmission. Existing annotation tools primarily rely on video communication, where visual and auditory dimensions fall short in conveying details of the actual scene when dealing with three-dimensional spatial relationships. These tools need spatial awareness of the scene, making it difficult for remote users to comprehend and intervene in on-site situations.

Researchers have explored remote annotation methods to address these limitations. Lai et al. [1] developed an online classroom video annotation system based on the network, allowing students to input answers in pop-up dialogs and teachers to review student answers at corresponding video positions, providing real-time interaction and feedback for both parties. Kuzuoka [2] utilized video streams for remote collaboration, where local users sent work environment videos to remote experts, who annotated the video and provided feedback to local users. However, these annotations would automatically disappear after a change in viewpoint. Although on-site personnel could observe remote annotations in real-time, the annotations were limited to individual points, lacking additional information.

Recognizing the limitations of video communication in remote collaboration, scholars introduced Augmented Reality (AR) technology to guide remote collaboration. Utilizing AR technology, remote users can perform real-time annotations in the actual environment, mapping virtual information into the real scene. Accurate spatial referencing ensures a consistent understanding of objects, scenes, or locations, enhancing collaboration efficiency and precision [3]. Existing solutions annotate on specific frames of the video. Gaugliz et al. [6] employed a method of hand-drawing two-dimensional images on frozen frames of video streams and then projecting these images into the real world to provide more intuitive information. Gaugliz et al. [7] hand-drew two-dimensional images on frozen frames of video streams and projected them into the real world, conveying more intuitive information. Kim et al. [8] addressed the annotation position drift caused by the user's screen movement during real-time video annotation. They proposed a method of freezing the real-time video frame and compared automatic and manual freezing methods. The research results indicated that automatically frozen frames effectively solved the problem of annotation position drift, assisting users in understanding the meaning of annotations more clearly. Gurevich et al. [9] froze frames of video streams and placed text information or 2D images on the screen to provide additional information and assistance. However, in complex three-dimensional spatial environments, factors such as sensor errors, environmental changes, or inaccurate system calibration can lead to inconsistencies between the actual object's position and the annotated position.

Integrating annotations into the real world would enhance user immersion and comprehension. Cho et al. [10] used RGB-D cameras to perceive indoor spaces, separating regions of each object through clustering and replacing them with virtual content. Gaugliz et al. [11] reconstructed the three-dimensional world space, anchored annotations on objects within that space, and visualized them using AR technology to mitigate drift.

In response to the shortcomings of existing annotation methods, this paper proposes a study on a three-dimensional model annotation system based on cloud anchors. Annotation of three-dimensional models can accurately present spatial relationships, and cloud anchor-enhanced annotations prevent drift due to changes in viewpoint, ensuring real-time and stable annotations.

2. Overall System Design

The overall framework of the three-dimensional model annotation system based on cloud anchors is illustrated in Figure 1, comprising four main components: three-dimensional model construction, model annotation drawing,

annotation module, and anchor module.

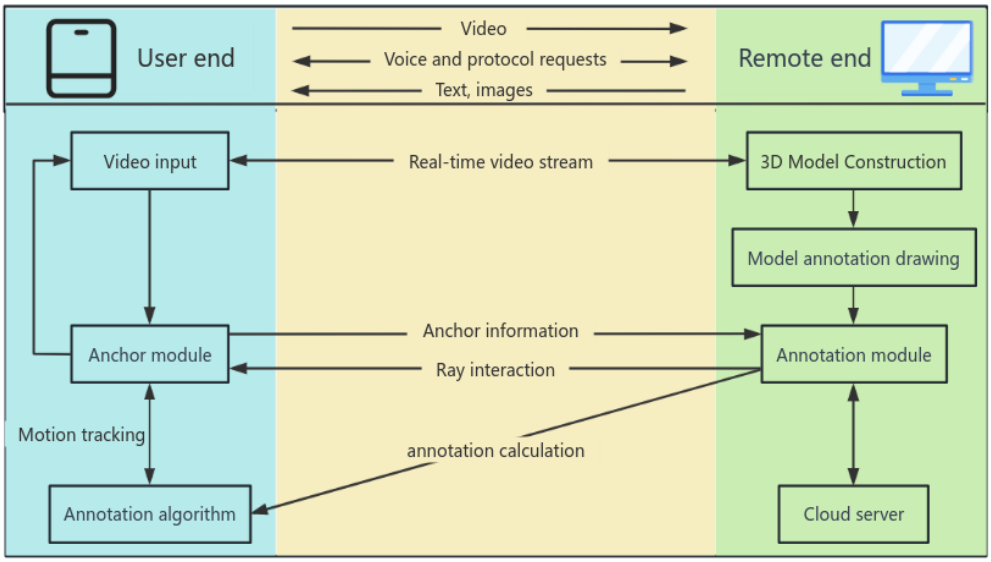

Figure 1. System Framework

2.1. 3D Model Construction

Field users capture surrounding objects with a mobile device, while ARCore collects sparse point clouds of the real-world environment. Frames are extracted from the real-time video stream and imported into Reality Capture. The foundational model is created by aligning images, determining relative positions and poses, generating sparse and dense point clouds, establishing surface mesh models, and creating textures in a series of steps. Ultimately, the utilization of the sparse point cloud detects and fills voids within the model. The final step involves exporting the three-dimensional model of the annotated object. Figure 2 illustrates the process.

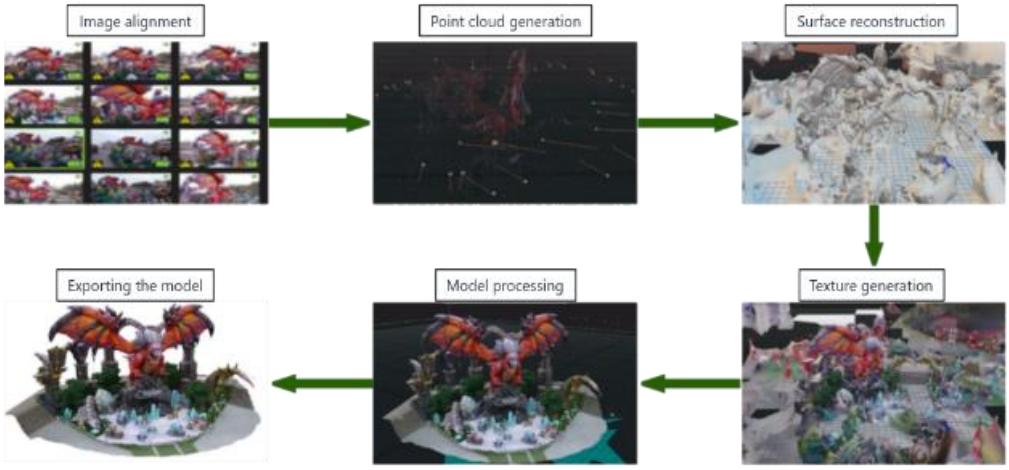

Figure 2. 3D Model Generation Process

2.2. Model Annotation Drawing

Model annotation drawing involves the three-dimensional reconstruction of the annotated object to obtain its model. Subsequently, the Blender software is employed to modify the model by adding elements such as arrows, text, and images, creating a new model that incorporates various user-added annotation information. This process aims to provide users with more spatially aware and information-rich three-dimensional annotations, enabling them to understand and communicate information about objects or scenes accurately. In contrast to two-dimensional plane annotations, three-dimensional model annotation involves objects or scenes in space, requiring consideration of multiple perspectives and observation points. By annotating on a three-dimensional model, users can finely control the position, direction, and attributes of annotations, enhancing accuracy and usability. Based on the nature and requirements of the task, one can add various annotation elements to convey spatial information carried by annotations. Ultimately, the annotated model can be converted to GLTF format and uploaded to a cloud server, supporting remote collaboration, sharing, and visualization.

2.3. Annotation Module

The annotation process of the system involves mapping two-dimensional annotations to three-dimensional space and establishing anchors using the three-dimensional point cloud, binding them to annotations to achieve persistent three-dimensional annotations. Processing the real-time video stream into video frames is illustrated in Figure 3. The improved ORB algorithm is employed to identify feature points and calculate descriptor information for each image, saving them as a set of dictionary data. This data set is serialized and uploaded to a cloud server. The annotation module utilizes the tracked feature points to solve the transformation matrix. Applying this transformation matrix

and utilizing the deserialized feature point data, one can display the annotated three-dimensional model in the AR annotation area alongside hand-drawn two-dimensional annotations.

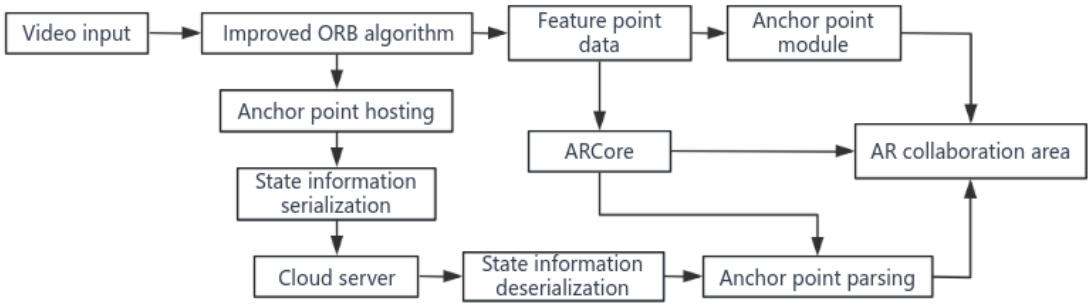

Figure 3. Annotation Process

2.4. Anchor Module

During the initial collaboration, the device scans the environment and saves the data. Local users can use AR drawing tools to create semi-transparent annotation areas in real space. Using ray detection, they anchor the annotation position at the collision point, uploading anchor information and real-world environmental features to the cloud server. The cloud server processes the uploaded data into a sparse point cloud map and stores the annotated GLTF format model. Other devices can load the current scene's cloud anchors and annotation content by resolving the cloud anchor ID. Ultimately, users can use AR interactive tools to manipulate the loaded model flexibly, allowing for a more precise analysis of the annotation content. Figure 4 illustrates the process of the anchor module.

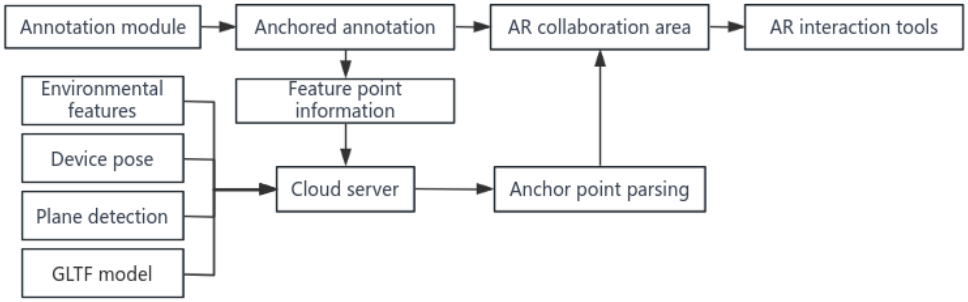

Figure 4. Anchor Module Process

3. System Algorithms for Annotation

Lighting conditions, surface texture, and device angles influenced the performance and feature point detection of ARCore, as revealed in the study. To better adapt to mobile devices, this paper makes partial improvements to the ORB algorithm [12]. The enhancements include using the CBRIEF operator's feature descriptor scheme to describe the extracted feature points and validating the feature matching results with the GMS verification algorithm. This improvement aims to enhance the algorithm's performance and efficiency on mobile devices by providing a more accurate description of feature points using the CBRIEF operator's feature descriptor scheme. Simultaneously, applying the GMS verification algorithm can effectively validate the accuracy of feature matching, further enhancing the overall stability and reliability of the algorithm. These improvement measures contribute to optimizing the application of the ORB algorithm on mobile devices, making it more suitable for specific requirements in real-world scenarios. Figure 5 illustrates the implementation process of the improved algorithm.

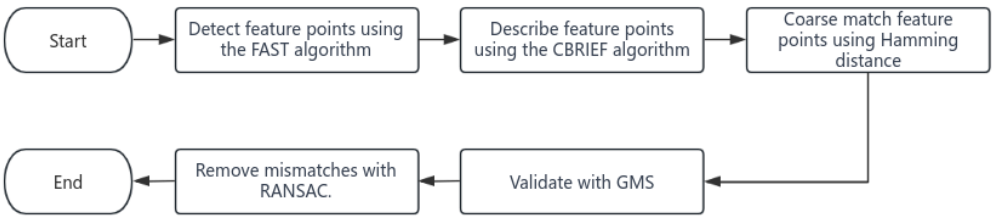

Figure 5. Improved Algorithm Implementation Process.

3.1. Descriptor Scheme Based on CBRIEF Operator

In order to address the issue of rotational invariance, the ORB algorithm utilizes the grayscale centroid method to determine the centroid of the image in the vicinity of the feature point. It then uses the vector direction from the feature point to the centroid as the primary orientation of the feature point. It applies a rotation operation to the descriptor using this orientation. This paper introduces a descriptor scheme based on the CBRIEF operator, which inherently possesses rotational invariance, eliminating the need to calculate the feature point's centroid and resulting in faster processing speeds. The specific implementation process of this scheme is detailed below.

Initially, a Gaussian difference pyramid is established for image processing with a standard deviation 1.2. FAST feature point detection is applied to each layer of the pyramid,

utilizing the descriptor method for feature point description. A patch with a radius of 5 pixels centered on the feature point is selected, along with four concentric circles centered on the feature point with radii of 2, 3, 4, and 5 pixels. The calculation of grayscale difference values between adjacent pixels on each circular ring is conducted, followed by a modulus operation on the obtained results. Equation (1) represents the modulus of grayscale differences in the vertical direction.

$$\mathrm{mod}(x,y)=\left|(f(x+1,y)-f(x,y))/f(x,y)\right| \quad (1)$$

Here, $f(x,y)$ represents the pixel grayscale value at the corresponding point of (x,y), and $\mathrm{mod}(x,y)$ is the modulus of the grayscale difference. The modulus of the grayscale difference values for adjacent pixels is calculated clockwise on the same circular ring, resulting in matrices P_2, P_3, P_4, and P_5. Equation (2) represents the resulting overall matrix P.

$$P=\begin{bmatrix}P_2 & P_3 & P_4 & P_5\end{bmatrix}=\begin{pmatrix} n_{21} & \cdots & n_{212} \\ n_{31} & \cdots & n_{316} \\ n_{41} & \cdots & n_{420} \\ n_{51} & \cdots & n_{528}\end{pmatrix} \quad (2)$$

To ensure the rotational invariance of the descriptor, rearrange P_2, P_3, P_4, and P_5. Place the maximum value, m, at the far left of the matrix, and pad the digits in front of the original maximum value to the end of the matrix. Then, normalize the rearranged vector, as shown in Equation (3).

$$\overline{P}=\frac{p_i}{\sqrt{\sum_1^n p_i^2}} \quad (3)$$

Here, $\overline{P}$ represents the row matrix in the P matrix after normalization. The descriptors obtained through the above calculations possess scale invariance, illumination invariance, and rotational invariance. Matching between descriptors from adjacent frames is performed using Euclidean distance.

Consider two sets of adjacent frame feature points, C_1 and C_2, with corresponding descriptor sets V_1 and V_2. Let d be the target feature point to be matched in C_1, and P be the descriptor matrix for any feature point. The closest feature point to the target one satisfies Equation (4).

$$\forall \overline{P}\in V_2, \left|d\leftrightarrow P\right|\le\left|d\leftrightarrow\overline{P}\right| \quad (4)$$

Here, the matrix distance $|d\leftrightarrow P|=\sqrt{\sum_i^k(d_i-P_i)^2}$, d_i represents the values of the descriptor matrices.

3.2. Validation Algorithm Based on GMS

The GMS (Grid-based Motion Statistics) algorithm rapidly eliminates incorrect matches through grid partitioning and motion statistical characteristics, enhancing matching stability [13]. Given the pronounced image distortions produced by the camera during the motion of mobile devices, the motion smoothness constraint of GMS is well-suited for mobile device scenarios.

The image region is gridified, and each grid contains relatively independent matching point pairs, each with a specific feature point. The number of matching point pairs in a grid neighborhood satisfies Equation (5).

$$\partial_i\sim\begin{cases}B(K_n,q_t) & w_i=true\\ B(K_n,q_f) & w_i=false\end{cases} \quad (5)$$

Here, q_t represents the correct matching rate, q_f represents the incorrect matching rate, w_i represents the i-th feature in the grid point pair set, and K_n represents the number of matching point pairs corresponding to the w_i grid. Grid partitioning enhances the efficiency of matching point pair division. Substituting the variance and mean of the calculation of incorrect and correct matches into Equation (6) calculates the difference value A6 between correct and incorrect matches.

$$P=\frac{K_{nqt}-K_{nqf}}{\sqrt{K_{nqt}(1-q_t)}-\sqrt{K_{nqf}(1-q_f)}} \quad (6)$$

By increasing the number of matching points in the image, one can deduce an elevation in the distinguishability between correct and incorrect matches, along with a larger support in the domain, as indicated by Equations (5) and (6). Consequently, a higher count of correct matches is achieved within the grid region, retaining the information of positive matching point pairs in the grid area and enhancing the algorithm's capacity for accurate matching. As a result, there are more correct matches in the grid region. The algorithm retains the information of positively matched point pairs in the grid region, enabling it to have more correct matches.

3.3. Analysis of Algorithm Performance

Various factors influence the real-time performance of the AR system on mobile devices. To assess the performance of our proposed algorithm in terms of tracking stability, we compared the test results of the ORB algorithm and the proposed method for feature point extraction in the same experimental environment. Tables 1 and 2 present the specific data.

By comparing the results presented in Tables 1 and 2, it is evident that the improved ORB algorithm outperforms the original ORB algorithm regarding feature matching. This observation underscores the effectiveness of utilizing CBRIEF operator-generated descriptors and employing a grid-based positive matching verification method to significantly increase the number of matches, thereby enhancing the accuracy of image feature detection and matching.

Table 1. Image Feature Extraction Results

Feature Point Extraction Method	Extraction Time (ms)	Number of Feature Points
ORB	**28**	**790**
Improved ORB	**22**	**765**

Table 2. Feature Point Detection and Matching Test Results

Feature Point Extraction Method	Feature Points to be Matched	Matching Time (ms)	Number of Matches
ORB	790/783	27	345
Improved ORB	765/761	21	389

4. Experiments and Analysis

4.1. Experimental Development Environment

The mobile configuration of the system is a Samsung Galaxy S10. In contrast, the remote configuration includes a laptop with an i7-11800H processor running at 2.30GHz, a GTX 3060 graphics card, and 16GB of RAM. The remote operating system is Windows 10, and the development environment involves Visual Studio 2019 and Unity3D. The cloud anchor database utilizes Firebase, and three-dimensional annotation tools encompass Reality Capture and Blender. The configuration combination aims to guarantee efficient system operation on both mobile and remote ends, delivering users a seamless and stable experience.

4.2. Experimental Verification Environment

We validated the effectiveness of the system through drone operation experiments. The experimental group utilized the model annotation system to guide local users in basic drone operations, while the control group received guidance using traditional two-dimensional annotations. The experiment aimed to compare the performance of the two groups in drone operation tasks and assess the effectiveness of the model annotation system in this scenario. The experiment included the following tasks: (A) Unfolding the drone, (B) Installing the remote controller, (C) Debugging the drone, (D) Takeoff, and (E) Guided landing.

A: Unfolding the drone - Participants took out the drone, opened the gimbal protective cover, and installed the battery. Sequentially unfold the drone arms.

B: Installing the remote controller - Participants installed the remote controller joysticks, unfolded the antenna, connected the phone to the controller, updated the drone firmware as per the prompts, and familiarized themselves with the various buttons on the drone.

C: Debugging the drone - Participants turned on the drone power and heard a beep after successful frequency matching. The mobile device displayed the drone gimbal shooting screen, used the joystick to move the gimbal and set drone parameters.

D: Takeoff - Observing the surrounding environment and confirming there were no obstacles, participants pressed the corresponding key to lift the drone, visually controlling the drone to navigate obstacles.

E: Guided landing - After the drone reached the destination, participants manually guided the drone to land at a fixed location and, finally, turned off the drone power.

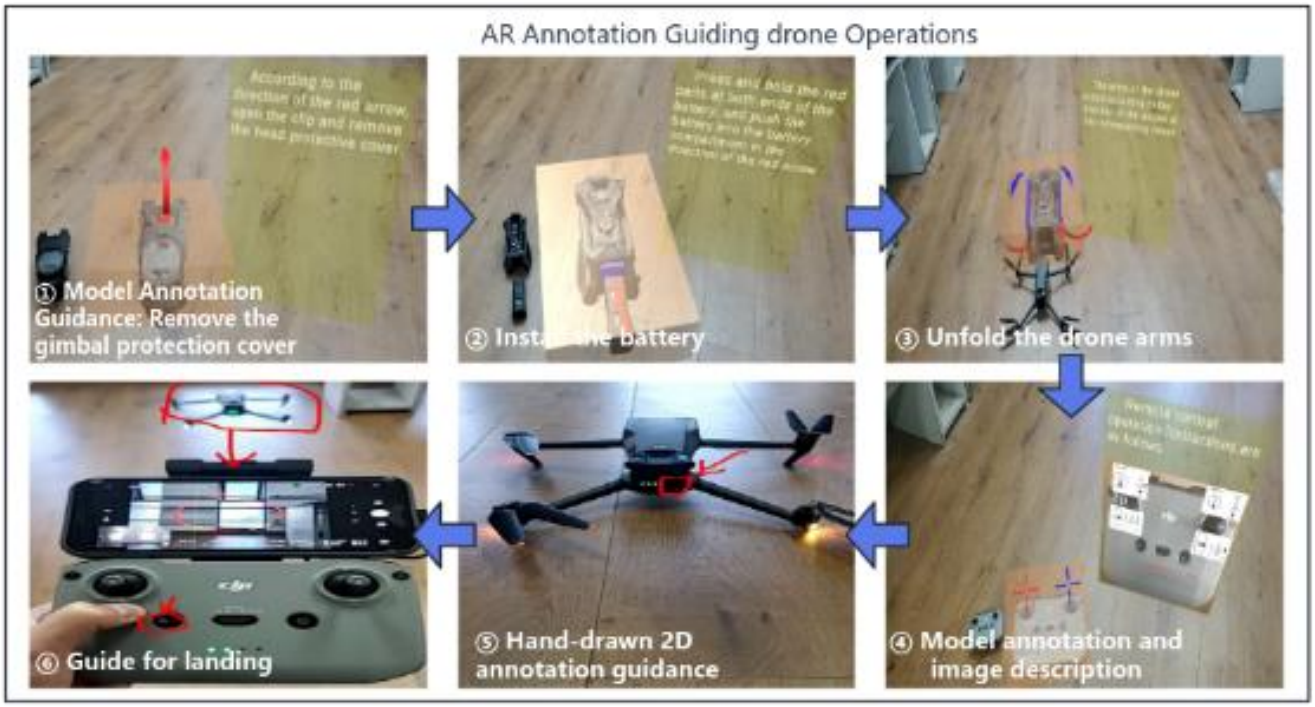

Figure 6. AR Annotation Guidance

Illustrated in Figure 6 is the workflow of the system. The mobile and remote users establish the connection, followed by the mobile device scanning the surrounding environment to create an AR collaboration area. The remote user constructs a three-dimensional model annotation based on the on-site environmental data and anchors the model annotations and textual comments to the AR collaboration area using cloud anchors. Mobile users can view the three-dimensional model annotations from any perspective. Subsequently, remote users guide mobile users in basic drone operations through three-dimensional model annotations, hand-drawn two-dimensional annotations, textual comments, and voice instructions. All annotation information is stored in the cloud, enabling other users to load annotations based on cloud anchor IDs for similar instructional operations in different environments. This system's workflow aims to achieve real-time guidance and information sharing in remote collaboration, and the application of cloud anchors ensures the persistence and sharing of annotation information.

4.3. Experimental Evaluation

In this study, we selected a user who frequently uses the drone as the remote guidance personnel and recruited 18 participants to act as operators. The participants were undergraduate or graduate students aged between 18 and 25, with an average age of 21.56 and a standard deviation of 1.83. These participants came from various academic disciplines, including communication engineering, computer science,

electronic information, and mathematics. Overall, participants indicated no prior experience with augmented reality applications, with an average score of 0.33 and a standard deviation of 0.58 (0 indicating 'I have never used an augmented reality application,' 1 indicating 'I occasionally use augmented reality applications,' and two indicating 'I frequently use augmented reality applications'). Simultaneously, all of the 18 participants had yet to experience with drones. An experimental group and a control group were randomly assigned, with the experimental group testing the augmented reality-based model annotation system and the control group utilizing the traditional two-dimensional annotation system.

After the experiment, all 18 participants completed the basic drone operation experiment. Following the experiment, the nine users in the experimental group evaluated aspects of the system, such as intuitiveness, accuracy, and interactivity, on a scale of 0 to 10. Table 3 presents the results of the user satisfaction assessment. Table 3 shows that the system effectively guides participants in operations, enhancing their understanding of drones and accessories. However, due to limitations in mobile interaction, participants gave relatively lower ratings to the system.

Table 3. User Satisfaction Evaluation

Scoring	Intuitiveness	Accuracy	Interactivity
8-10 points	8	7	2
6-8 points	1	2	2
4-6 points	0	0	5
2-4 points	0	0	0
0-2 points	0	0	0

According to the data presented in Table 4, the average completion time for the control group was 222.56 seconds, significantly longer than the experimental group's 202.33 seconds. Statistical analysis results indicate a significant difference in the experiment completion time between the experimental and control groups, with a p-value less than 0.01. Regarding accuracy, the p-value between the experimental and control groups was also significantly less than 0.01, with the experimental group consistently demonstrating higher accuracy than the control group.

Table 4. Analysis of Experiment Completion Time and Operation Accuracy

Variable	M	SD	t	p
Completion Time			-2.84	0.01
Experimental Group	202.33	14.01		
Control Group	222.56	8.93		
Operational Accuracy			4.91	0.0005
Experimental Group Control Group	86.42% 78.33%	2.66 2.62		

A comprehensive analysis of the results indicates that, overall, the experimental group demonstrated superior average experimental performance. Compared to traditional two-dimensional annotation systems, this system provides more intuitive annotation guidance, effectively enhancing the efficiency of remote guidance and improving users' understanding of drones and their accessories.

5. Conclusion

In order to enhance the quality and efficiency of annotations while illustrating the spatial relationships of annotated scenes, this study integrates cloud anchors, 3D reconstruction, and augmented reality technology into the annotation system. Firstly, we elaborate in detail on the overall framework of an augmented reality-based 3D model annotation system. Secondly, model annotations are constructed by combining cloud anchors and 3D reconstruction technology, focusing on investigating an improved AR tracking and registration algorithm to display annotations with spatial information in an AR environment. Finally, a drone basic operation experiment validates the system's effectiveness. However, the system exhibits certain limitations regarding flexibility on the mobile end, and future improvements may include adding more interactive means for mobile users. Additionally, cloud anchors' response time may impact annotations' timeliness, prompting potential enhancements in the anchor module.

References

[1] Qiu C, Zhou S, Liu Z, et al. Digital assembly technology based on augmented reality and digital twins: a review[J]. Virtual Reality & Intelligent Hardware, 2019, 1(6): 597-610.

[2] Oda O, Sukan M, Feiner S, et al. Poster: 3D referencing for remote task assistance in augmented reality[C]//2013 IEEE Symposium on 3D User Interfaces (3DUI). IEEE, 2013: 179-180.

[3] Lai A F, Li W H, Lai H Y. A Study of Developing a Web-based Video Annotation System and Evaluating Its Suitability on Learning[C]//Proceedings of the 2nd International Conference on Education and Multimedia Technology. 2018: 44–48.

[4] Kuzuoka H. Spatial workspace collaboration: a SharedView video support system for remote collaboration capability[C]//Proceedings of the SIGCHI conference on Human factors in computing systems. 1992: 533–540.

[5] Gauglitz S, Nuernberger B, Turk M, et al. In touch with the remote world: Remote collaboration with augmented reality drawings and virtual navigation[C]//Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology. 2014: 197-205.

[6] Kim S, Lee G A, Sakata N. Comparing pointing and drawing for remote collaboration[C]//2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). IEEE, 2013: 1–6.

[7] Gurevich P, Lanir J, Cohen B, et al. TeleAdvisor: a versatile augmented reality tool for remote assistance[C]//Proceedings of the SIGCHI conference on human factors in computing systems. 2012: 619–622.

[8] Venerella J, Sherpa L, Tang H, et al. A Lightweight Mobile Remote Collaboration Using Mixed Reality[C]//Proceedings of Computer Vision and Pattern Recognition. 2019: 1–4.

[9] Cho H, Jung S U, Jee H K. Real-time interactive AR system for broadcasting[C]//2017 IEEE Virtual Reality (VR). IEEE, 2017: 353–354.

[10] Gauglitz S, Nuernberger B, Turk M, et al. World-stabilized annotations and virtual scene navigation for remote collaboration[C]//Proceedings of the 27th annual ACM

symposium on User interface software and technology. 2014: 449-459.

[11] Nuernberger B, Lien K C, Höllerer T, et al. Interpreting 2d gesture annotations in 3d augmented reality[C]//2016 IEEE symposium on 3D user interfaces (3DUI). IEEE, 2016: 149-158.

[12] Rublee E, Rabaud V, Konolige K, et al. ORB: An efficient alternative to SIFT or SURF[C]//2011 International conference on computer vision. Ieee, 2011: 2564-2571.