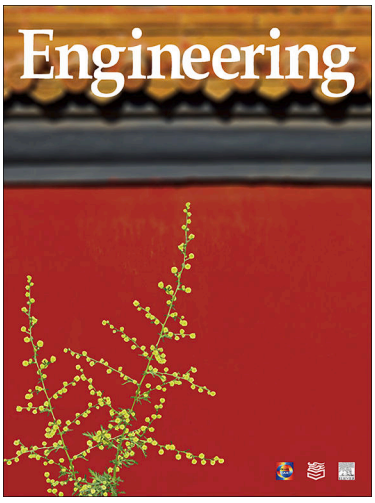
Journal Pre-proofs



Article

ECT-VR: Intuitive Teleoperation via Hybrid Virtual–Reality Immersive Interaction with Multi-Sensor Perception for Agricultural Robots

Tao Liu, Yinling Xie, Yang Li, Jun Zhou, Zhonghua Miao, Baohua Zhang, Chengliang Liu

PII: S2095-8099(26)00563-1
DOI: https://doi.org/10.1016/j.eng.2026.09.013
Reference: ENG 2477

To appear in: *Engineering*

Received Date: 6 August 2025
Revised Date: 18 August 2026
Accepted Date: 2 September 2026

Please cite this article as: T. Liu, Y. Xie, Y. Li, J. Zhou, Z. Miao, B. Zhang, C. Liu, ECT-VR: Intuitive Teleoperation via Hybrid Virtual–Reality Immersive Interaction with Multi-Sensor Perception for Agricultural Robots, *Engineering* (2026), doi: https://doi.org/10.1016/j.eng.2026.09.013

This is a PDF of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability. This version will undergo additional copyediting, typesetting and review before it is published in its final form. As such, this version is no longer the Accepted Manuscript, but it is not yet the definitive Version of Record; we are providing this early version to give early visibility of the article. Please note that Elsevier's sharing policy for the Published Journal Article applies to this version, see: https://www.elsevier.com/about/policies-and-standards/sharing#4-published-journal-article. Please also note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2026 THE AUTHORS. Published by Elsevier LTD on behalf of Chinese Academy of Engineering and Higher Education Press Limited Company

Research

Agricultural Sensors—Article

ECT-VR: Intuitive Teleoperation via Hybrid Virtual–Reality Immersive Interaction with Multi-Sensor Perception for Agricultural Robots

Tao Liu [a], Yinling Xie [b], Yang Li [a], Jun Zhou [c], Zhonghua Miao [e], Baohua Zhang [a,*], Chengliang Liu [d,*]

[a] *School of Smart Agriculture, College of Artificial Intelligence, Nanjing Agricultural University, Nanjing 211800, China*

[b] *Digital Art Academy, Nanjing Vocational College of Information Technology, Nanjing 210023, China*

[c] *College of Engineering, Nanjing Agricultural University, Nanjing 211800, China*

[d] *School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China*

[e] *School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China*

* Corresponding authors.
E-mail addresses: bhzhang@njau.edu.cn (B. Zhang), chlliu@sjtu.edu.cn (C. Liu)

ARTICLE INFO

Article history:

Received

Revised

Accepted

Available online

Keywords:

Teleoperation

Virtual reality

Agricultural mobile robots

3D reconstruction

Human–robot interaction

ABSTRACT

Fully autonomous robots are still difficult to deploy in complex, unstructured, and dynamic agricultural environments, whereas conventional teleoperation systems suffer from insufficient environmental information and poor spatial intuitiveness. To overcome these limitations, this paper presents enhanced conventional teleoperation with virtual reality (ECT-VR), an immersive teleoperation system for agricultural mobile robots based on hybrid three- and two-dimensional (3D+2D) perception and interaction. The system employs a collaborative architecture that combines a preconstructed global 3D map with real-time local color-depth feedback. In the 3D subsystem, aerial images are processed using structure-from-motion (SfM) and multi-view stereo reconstruction, together with multi-stage post-processing methods, to generate large-scale static agricultural maps for global environmental perception. In the 2D subsystem, an onboard color-depth camera provides real-time visual feedback, while depth

2095-8099/© 2026 THE AUTHORS. Published by Elsevier LTD on behalf of Chinese Academy of Engineering and Higher Education Press Limited Company. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Journal Pre-proofs

data are used for robot pose estimation and local dynamic information during field operations. A Unity3D-based virtual-reality interface synchronizes the virtual robot model with the physical robot and incorporates global positioning visualization, collision-warning visualization, and motion-command feedback. Field experiments conducted at four agricultural and field-relevant outdoor sites, together with a user-centered validation involving 20 participants, show that ECT-VR can support environmental information visualization and robot motion control in agricultural teleoperation. The system achieved an overall mean Likert score of 4.75/5.00 across 11 evaluation items, with an average positive-response proportion of 95.0% for scores ≥ 4. These results suggest that ECT-VR provides user-perceived benefits in information completeness, interaction intuitiveness, operational support, and safety assistance for agricultural robot teleoperation.

1. Introduction

With continuous advancements in sensor systems, computational capabilities, and computer vision technologies, agricultural machinery is progressively evolving into a core supporting technology within modern agricultural systems [1]. These robotic solutions can reduce human involvement in labor-intensive tasks (e.g., plowing, seeding, and harvesting) and hazardous agricultural environments involving pesticide, fertilizer application, or toxic crop harvesting [2]. However, the agricultural domain presents unique complexities characterized by unstructured environments, challenging operational requirements, and the need for adaptive intelligent decision-making [3]. Current mobile robotic systems are unable to achieve flawless full autonomy in agricultural settings, nor can they adequately meet the diverse operational demands. Despite technological progress, achieving comprehensive autonomous and intelligent operations remains a long-term objective.

To address these challenges, teleoperated robotic systems, in which a human controls a robot from a distance, have emerged as a promising solution. Teleoperation technology leverages mechanical actuation to reduce strenuous human labor while protecting humans from hazardous conditions in agricultural environments. More importantly, it integrates human intelligence to handle various operational contingencies, thereby overcoming the limitations of full autonomy [4]. This approach provides a practical way to combine operational efficiency with adaptive flexibility, representing an advancement in agricultural automation [5].

Building upon established methodologies in teleoperation research, conventional agricultural teleoperation systems typically employ a leader–follower architecture [6]. In this configuration, devices (followers) capture real-time field conditions and operational status through imaging sensors, transmitting visual data to operator interfaces via network connections. Human operators (leaders) typically issue commands through input devices such as joysticks [7], keyboard and mouse interfaces [8], or gamepad controllers [9] while monitoring real-time visual feedback. This paradigm has also been implemented in adjacent domains, including semi-autonomous mobile rescue robots using Internet Protocol (IP) cameras for disaster-zone surveillance [10], low-cost urban search-and-rescue robots using camera-based visual feedback [11], and military support and rescue robots using wireless communication for remote operation [12].

Conventional video-stream-based control schemes, which rely on single-viewpoint visual feedback, place significant constraints on operational information acquisition, impairing teleoperation effectiveness and performance [13,14]. These limitations become particularly pronounced when operators control robots in complex and dynamic agricultural environments [9,14,15]. To address these constraints, researchers are actively exploring strategies to acquire more comprehensive on-site data. A 160° wide-angle camera with a resolution of 3280 × 2464 pixels was implemented on harvesting machinery to increase field-of-view coverage [16]. The visual system was paired with a display screen composed of three display interfaces, providing operators with panoramic situational awareness. Multi-camera teleoperation configurations have also been developed by deploying static cameras at critical positions, synchronizing and aligning the video feeds, and downscaling their resolution for unified display on a single monitor [17].

Nevertheless, these approaches remain constrained by the two-dimensional (2D) representation of visual data and the physical limitations of planar displays, often requiring multiple monitors, panoramic image stitching, or specialized image processing to improve operator awareness. With the development of virtual reality (VR) head-mounted displays (HMDs), researchers have attempted to overcome these limitations by providing immersive stereoscopic visualization and more natural interaction interfaces. In agricultural applications, Udekwe and Seyyedhasani [18] developed a SteamVR/Unity3D-based teleoperation framework for robot–plant interaction during fruit harvesting, achieving a 90% success rate and demonstrating the feasibility of using VR to improve operator immersion and interaction performance. Wang et al. [5] further explored immersive human–robot interaction for agricultural mobile robots by combining depth-camera-based environmental perception with VR visualization. In broader mobile and field robotic applications, Shen et al. [19] and Nakanishi et al. [20] implemented stereoscopic camera systems on mobile robots to support VR-based telepresence; Park et al. [21] delivered eye-specific binocular video streams to VR displays and combined them with wearable motion controllers; and Wei et al. [22] integrated global and local visual feedback through VR spatial mapping to enhance operational awareness.

These VR-enhanced teleoperation systems can alleviate the depth-perception deficiency of conventional displays by emulating human binocular vision through stereoscopic disparity, thereby providing more intuitive depth cues and a stronger sense of presence than conventional monitor-based interfaces. However, many existing VR teleoperation systems still rely on fixed cameras, robot-mounted cameras, or local stereoscopic video feedback. Although such configurations improve local perception, they remain insufficient for large-scale agricultural environments, where operators must simultaneously understand the global scene layout, local terrain conditions, and dynamic changes around the robot. Therefore, VR

visualization alone cannot fully resolve the information-acquisition problem in agricultural teleoperation; rather, it must be combined with a scene-representation strategy that can provide both global spatial context and local real-time feedback.

To provide operators with the global spatial context beyond local visual feedback, recent studies have integrated three-dimensional (3D) reconstruction, digital mapping, and virtual environments into teleoperation systems. This paradigm allows reconstructed environments to be visualized in VR from arbitrary viewpoints, providing operators with stronger immersion and more comprehensive situational awareness [23,24]. Pace et al. [25] developed a real-time 3D environmental reconstruction system based on color (red–green–blue)-depth (RGB-D) cameras; however, their experimental results showed trade-offs in reconstruction quality, particularly in semantic information fidelity. Wang et al. [5] adopted a simultaneous localization and mapping (SLAM)-based modeling method to support immersive human–robot interaction for agricultural mobile robots, while Naceri et al. [26] reduced computational costs by restricting 3D reconstruction to local regions captured by a wrist-mounted camera on a robotic manipulator. Other studies have attempted to reduce the computational burden through data abstraction or pre-modeling. Zhou et al. [27] used PointNet++ to classify complex point-cloud data and replace it with pre-modeled virtual objects. Holubek et al. [28] improved efficiency through computer-aided design (CAD)-model-based pre-mapping, although such methods require a strict correspondence between physical and virtual elements. Van de Merwe et al. [29] and Pérez et al. [30] also investigated preconstructed environments for VR-based teleoperation and operator training, in which one-to-one virtual–physical mapping was established to enable immersive operation after rendering in VR devices. Nevertheless, these methods are often constrained by local workspaces, predefined scene models, or controlled environments.

Our previous study [4] followed this preconstruction-based direction by combining unmanned aerial vehicle (UAV)-based agricultural digital mapping with VR-assisted control, demonstrating that reconstructed agricultural 3D maps can provide useful global spatial information for immersive teleoperation. However, preconstructed 3D maps are inherently static and cannot independently reflect local dynamic changes during field operation, such as temporary obstacles, terrain disturbances, or the real-time state of the robot's working area. Conversely, in unstructured agricultural environments characterized by vegetation occlusion, uneven illumination, and complex terrain, real-time full-scene 3D reconstruction remains computationally expensive and is often accompanied by latency, reduced resolution, and incomplete reconstruction [31,32]. These limitations indicate that neither purely real-time 3D reconstruction nor purely static preconstructed mapping is sufficient for robust agricultural teleoperation. A more practical solution is to combine a high-quality preconstructed 3D global map with lightweight real-time local perception, thereby providing a feasible trade-off among global scene awareness, local dynamic feedback, and computational feasibility.

Building on the above considerations, this study proposes an enhanced teleoperation system for agricultural mobile robots, named enhanced conventional teleoperation with virtual reality (ECT-VR). ECT-VR is designed as a hybrid information-representation framework for human-in-the-loop agricultural teleoperation, rather than merely a combination of sensing and visualization modules. Instead of relying solely on real-time video streams or real-time full-scene 3D reconstruction, the system adopts a hybrid "3D+2D" architecture to integrate virtual and real-world information during robot operation. In the 3D subsystem, UAVs are used to periodically acquire agricultural scene data, and static 3D global maps are reconstructed and refined to provide operators with global environmental perception, route-level understanding, and pre-operation observation. In the 2D subsystem, a depth camera mounted on the robot provides real-time RGB-D video feedback and supports pose estimation based on real-time appearance-based mapping (RTAB-MAP), thereby supplying local dynamic information around the robot during field operation. The virtual robot model in Unity is synchronized with the physical robot through real-time pose data, allowing the operator to observe the robot state within the preconstructed global map while simultaneously receiving local onsite feedback. Based on the Unity3D engine and VR interaction devices, the system further integrates intuitive control, global positioning visualization, collision-risk highlighting, and motion-command feedback. Through this integrated representation, ECT-VR is designed to support operator situational awareness and decision-making in unstructured agricultural environments, while improving the completeness and intuitiveness of teleoperation information for agricultural mobile robots.

The main contributions of this study are summarized as follows:

(1) An integrated hybrid VR teleoperation framework is developed for agricultural mobile robots in dynamic and unstructured field environments. By providing immersive spatial information and real-time local feedback, the proposed framework combines human decision-making with robot motion control, thereby alleviating the limitations of both full autonomy and conventional video-based teleoperation.

(2) A hybrid 3D+2D perception architecture is proposed to provide more comprehensive environmental awareness. Periodically updated UAV-based 3D global maps offer large-scale scene perception, while real-time RGB-D video feedback from the robot provides local dynamic information during field operation.

(3) A large-scale agricultural scene reconstruction and processing workflow is established by integrating COLMAP, a structure-from-motion (SfM) reconstruction tool, fast multi-view stereo network (Fast-MVSNet), Poisson reconstruction, Taubin smoothing, and other multi-stage post-processing techniques.

(4) A multi-source perception-based VR interaction interface is developed by integrating RTAB-MAP-based pose estimation and virtual-physical pose synchronization. This interface supports information visualization and robot motion control through immersion, intuitive control, global positioning, and collision-warning assistance. The proposed system is further validated through field experiments and user-centered evaluation.

2. Materials and methods

2.1 Overall framework

To address the inherent limitations in information acquisition during conventional agricultural teleoperation, this study developed the ECT-VR mobile robot system for dynamic agricultural environments. Fig. 1 presents the overall architecture of the proposed ECT-VR system, which integrates a 3D global mapping subsystem, a 2D real-time feedback subsystem, and a VR-based interaction module. Unlike traditional video streaming or reconstruction methods, the system employs a 3D+2D approach. The 3D subsystem constructs a virtual space based on periodically updated unstructured static maps, providing operators with global scene awareness through large-scale environmental modeling. Although the maps themselves are static, the system achieves pose synchronization between the virtual robot model and the physical robot through operation-time pose feedback. The 2D subsystem, through real-time video stream transmission, focuses on the working area in front of the robot, providing real-time feedback on motion trajectories and operational status.

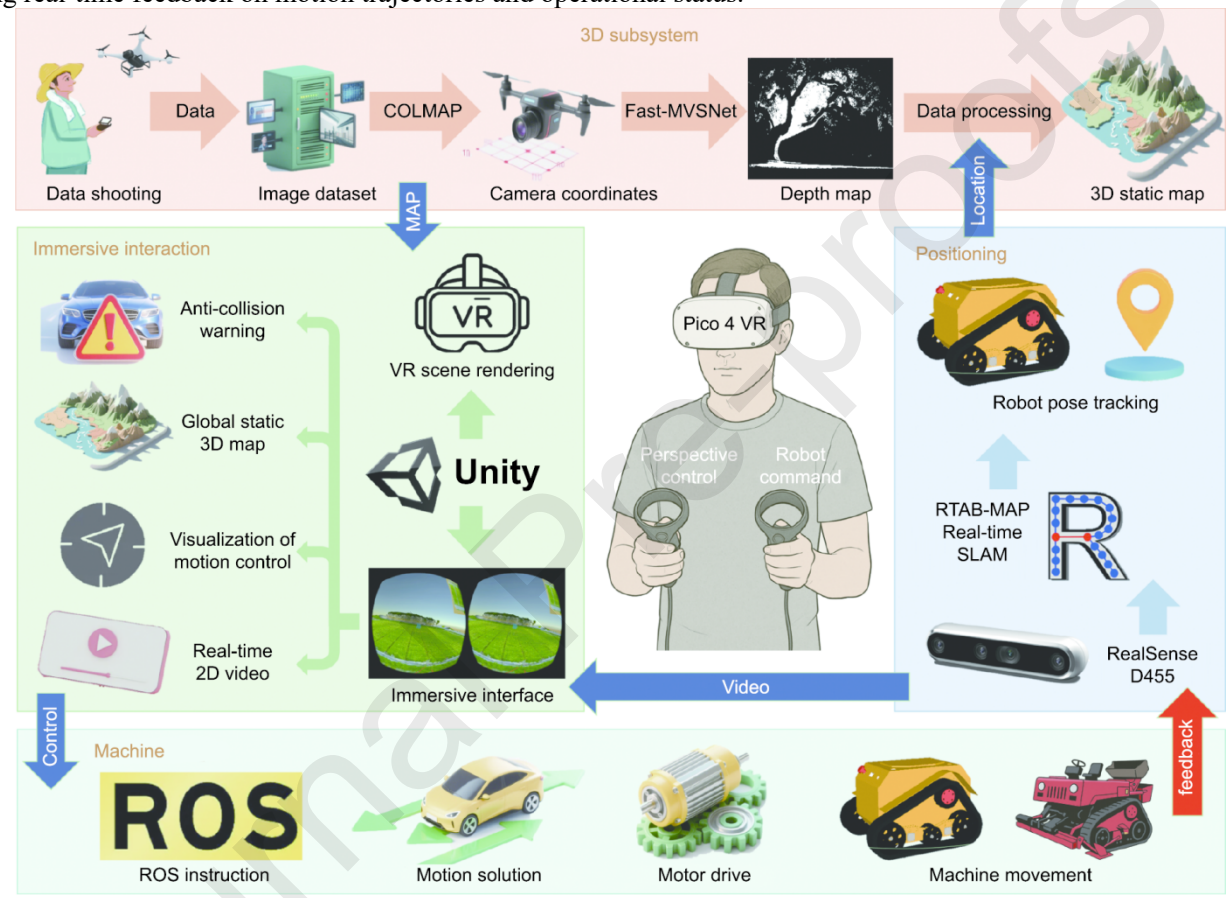


Fig. 1. Overall framework of the proposed system.

In the VR module, the system constructs an immersive interactive interface that integrates a stereoscopic information display with natural interaction control functions. At the control level, intuitive operations with spatial pose mapping are achieved through VR input devices, supporting head pose detection and controller command input to enhance control precision and operational naturalness. In terms of data visualization, drawing on concepts from gaming and autonomous driving domains, a stereoscopic information processing system was developed: The global map presents the geographic coordinates, the collision-warning module highlights risk areas, and control commands provide visual feedback. Multi-sensor perception is provided through the system-level integration of UAV image data, robot-mounted RGB-D sensing, RTAB-MAP-based pose estimation, and VR-side interaction sensing.

2.2 The 3D subsystem

2.2.1 Data collection and processing

UAV image data were acquired using a DJI Mini 3 Pro equipped with a built-in RGB camera. Video frames with a resolution of 1280 × 960 pixels were extracted as the input for subsequent SfM and multi-view stereo processing. The UAV acquisition strategy was adjusted according to the spatial structure and occlusion characteristics of each experimental site. Rotational scans were used for the site comprising a small garden and wooded area to improve image coverage under surrounding-building or vegetation occlusion, whereas raster scans were used for the larger park-like scene and the field-

validation environment to cover elongated and open field terrain. The UAV had a maximum wind-speed resistance of 10.7 $m \cdot s^{-1}$, and data collection was conducted under outdoor conditions within this operational limit. The corresponding camera-pose distributions estimated by COLMAP were used to check image coverage before dense reconstruction. After image acquisition, ECT-VR employs COLMAP, an SfM reconstruction tool, to estimate the camera pose for each image in preparation for subsequent 3D reconstruction [33]. First, COLMAP performs scale-invariant feature transform (SIFT) feature extraction on all images [34]. SIFT identifies key points in the images and generates 128-dimensional feature vector descriptors, which are robust to illumination, rotation, and scale variations. Feature matching is then used to find corresponding feature points across images, with the random sample consensus (RANSAC) algorithm employed to filter out incorrect matches [35]. Based on the matched points, triangulation generates an initial sparse point cloud. Feature points in this sparse point cloud are continuously matched with those in new images. As new images are integrated, the system performs triangulation, pose estimation, and local bundle adjustment (BA) optimization [36]. Finally, global BA optimization is applied to all camera poses and the sparse point cloud to obtain accurate camera poses.

The processing workflow of the 3D subsystem is summarized in Fig. 2. In the current prototype, the UAV-based 3D map update is intended for the periodic or event-triggered refresh of relatively stable global scene structures, rather than real-time reconstruction of rapidly changing objects. In practical deployment, the update interval should depend on field changes, crop growth stage, task schedule, and the occurrence of large structural changes, whereas short-term dynamic elements are handled by the 2D real-time feedback subsystem through RGB-D video, depth sensing, and collision-warning visualization during teleoperation.

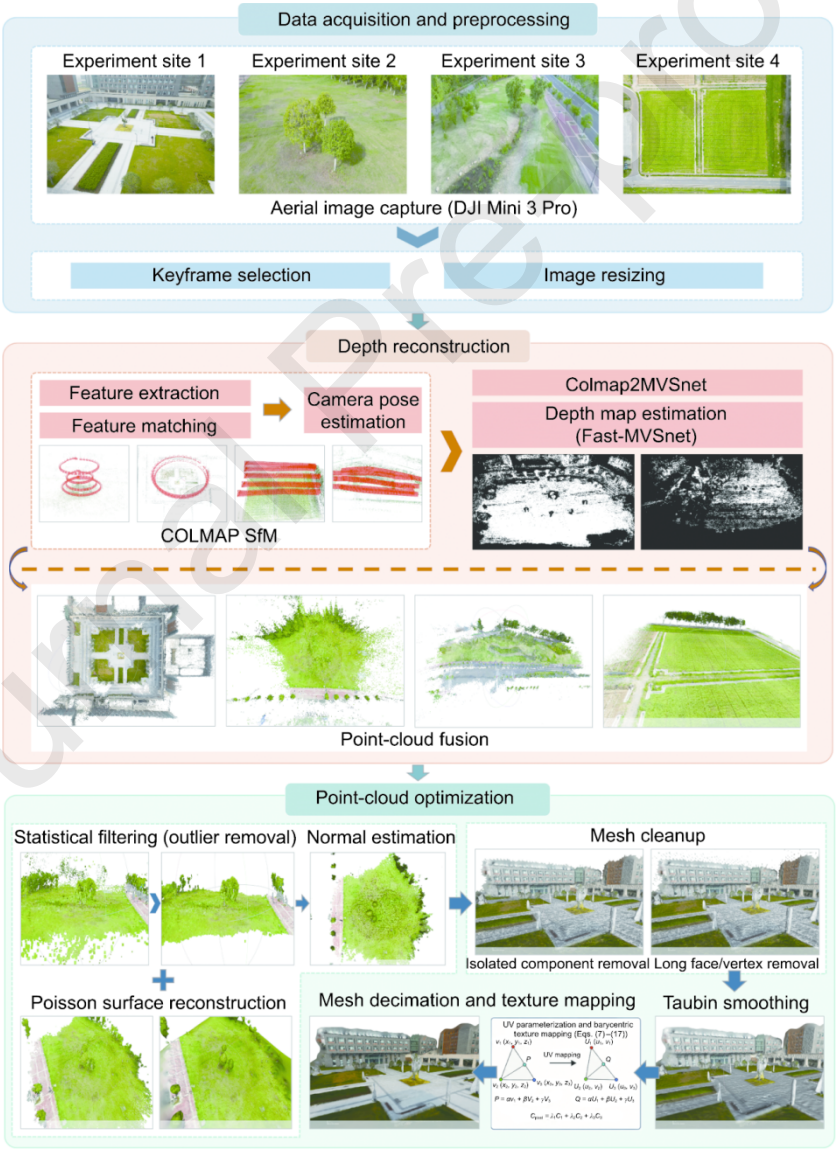


Fig. 2. The 3D subsystem processing framework.

2.2.2 Point-cloud reconstruction

Inferring 3D information from 2D images has long been a core challenge in computer vision and a crucial technique for achieving the shift from planar perception to 3D information acquisition. ECT-VR employs the Fast-MVSNet model [37], an

end-to-end multi-view stereo algorithm based on deep learning. Fast-MVSNet is an efficient improvement over the classic MVSNet [38], with advantages in terms of speed: It is multiple times faster than traditional models such as MVSNet and Point-MVSNet [39]. This computational advantage is beneficial for updating static 3D maps in the proposed workflow. It enables the estimation of pixel-level depth in multi-view images without prior geometric assumptions. ECT-VR takes the multi-view images captured by the UAV as input to a convolutional neural network to extract multi-scale semantic features. Then, based on the internal and external parameters of the camera (i.e., focal length, pose matrix), it warps the features of each view to the coordinate system of the reference view, constructs a sparse 3D cost volume, and generates a high-resolution sparse depth map through 3D U-Net regularization and Softmax–Argmax operations. Since the sparse depth map does not cover all pixels, a joint bilateral upsampling module is needed. Using the original image as a guiding signal, it learns the spatial distance weights and feature similarity weights between pixels to densify the depth map. Finally, a Gauss–Newton layer is introduced to transform the depth estimation into a nonlinear least-squares problem. By optimizing the multi-view reprojection error, the dense depth map is refined to sub-pixel accuracy. Herein, ECT-VR utilized the Fast-MVSNet model trained on the DTU training dataset [40].

After obtaining the depth information of the images, multi-view depth fusion is necessary. Here, we adopt the point-cloud fusion algorithm from D²HC-RMVSNet [41]. As described in that work, the transformation from a depth pixel to a world-coordinate 3D point follows the standard back-projection geometry using the camera's intrinsic and extrinsic parameters, as expressed in Eqs. (1) and (2):

$$\begin{bmatrix} X_{\mathrm{w}} \\ Y_{\mathrm{w}} \\ Z_{\mathrm{w}} \\ 1 \end{bmatrix} = T_{\mathrm{w}}^{\mathrm{c}} \begin{bmatrix} Z_{\mathrm{c}} \cdot \frac{u-c_x}{f_x} \\ Z_{\mathrm{c}} \cdot \frac{v-c_y}{f_y} \\ Z_{\mathrm{c}} \\ 1 \end{bmatrix} \quad (1)$$

where (u,v) are the image pixel coordinates, (c_x,c_y) are the principal point coordinates, f_x and f_y are the focal lengths in the x- and y-directions, Z_{c} is the depth value in the camera coordinate system, $T_{\mathrm{w}}^{\mathrm{c}}$ is the extrinsic matrix of the camera, and $(X_{\mathrm{w}},Y_{\mathrm{w}},Z_{\mathrm{w}})$ are the 3D point coordinates in the world coordinate system.

$$T_{\mathrm{w}}^{\mathrm{c}} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \quad (2)$$

where R is a 3 × 3 rotation matrix, and t is a 3 × 1 translation vector.

After obtaining the point cloud, to improve its quality, unreliable depth values must be filtered out based on the photometric confidence and geometric consistency of the point cloud in multiple views.

2.2.3 Post-processing and meshing

Although the point cloud has already undergone filtering during the depth fusion process, it is still necessary to further remove outliers. Here, statistical filtering was employed as a standard point-cloud outlier-removal strategy [42]. Statistical filtering traverses all the points and calculates the average distance of the points within the neighborhood of each point. For each point P_i in the point cloud, its k nearest neighbor points are found, and the average distance d_i from P_i to these neighborhood points is calculated:

$$d_i = \frac{1}{k}\sum_{j=1}^{k} \| p_i - p_j \| \quad (3)$$

where p_j is the jth nearest neighbor point of p_i. Then, the average distance μ and the standard deviation σ_{d} of all points are calculated:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} d_i \quad (4)$$

$$\sigma_{\mathrm{d}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (d_i - \mu)^2} \quad (5)$$

After that, different thresholds must be set for different scenarios to filter out outliers accordingly. The following are the criteria for identifying outliers:

$$d_i > \mu + \alpha\sigma_{\mathrm{d}} \quad (6)$$

where α is a scene-dependent statistical filtering coefficient that controls the strictness of outlier removal. After filtering, meshing is required. First, the normals of the point-cloud data are calculated to obtain the orientation of the local surface at each point, which is perpendicular to the local tangent plane. To ensure the quality of the normals, we employ multiple smoothing iterations to prevent excessive differences in normal directions between adjacent points caused by the significant unevenness of various plants in agricultural scenes. Subsequently, Poisson surface reconstruction is used to mesh the point cloud [43]. In Poisson reconstruction, the point-cloud space is divided using a multi-resolution octree to support efficient computation. Here, to ensure the accuracy of the model, the maximum depth of the octree is set to 12. Adaptive sampling is applied to the depth. For areas with higher point-cloud density, deeper voxels are used; for low-density areas, shallower voxels are adopted. The parameters used for statistical filtering, normal estimation, and Poisson reconstruction at the four experimental sites are summarized in Table 1. The parameters in Table 1 were selected empirically through preliminary reconstruction trials, considering the point-cloud noise, density distribution, terrain-detail preservation, and model complexity.

Table 1
Parameters for statistical filtering, normal estimation, and Poisson reconstruction across four experimental sites.

	Statistical filtering		Normals		Poisson reconstruction depth
	k	α	Neighbor number	Smoothing iterations	
Experiment site 1	5	5	200	5	12
Experiment site 2	50	1	60	3	12
Experiment site 3	6	1	50	20	12
Experiment site 4	6	1	50	20	12

After obtaining the mesh, due to the inherent shortcomings of Poisson reconstruction or issues with outliers, redundant Poisson faces may emerge. Herein, the system performs Poisson cleanup—specifically, "Select faces with edges longer than..."—on each mesh model. For the model after Poisson reconstruction, all faces in the mesh are traversed, and the length of each edge of every face is checked. If a face contains at least one edge whose length exceeds a set threshold, that face is selected and removed. Subsequently, the model undergoes "remove isolated pieces" to further delete isolated faces that are far from the main model. These operations reduce invalid information and redundant abnormal faces in the model to a certain extent.

Since COLMAP and Fast-MVSNet were not originally designed for processing large-scale agricultural information, which is inherently complex, the generated large-scale models inevitably contain numerous minor errors. These errors not only compromise the subsequent semantic analysis of the models but also exacerbate computational complexity. To address this issue, the system applies Taubin smoothing to each model, which adjusts the positions of mesh vertices to reduce surface noise and make the mesh smoother [44]. This algorithm differs from traditional smoothing algorithms: Conventional Laplacian smoothing achieves smoothing by moving vertices to the average position of their neighboring vertices, which, after multiple iterations, often leads to gradual shrinking and volume loss of the model. However, the Taubin smoothing algorithm, developed from Laplacian smoothing, can resolve this problem. Each iteration of Taubin smoothing involves one shrinking step and one expanding step; the synergistic effect of these two operations counteracts the shrinking issue, ensuring that the model is smoothed while retaining more original information.

Even after undergoing Poisson cleanup and the removal of isolated pieces, the volume and size of the model may still be excessively large, requiring processing while ensuring that accuracy and semantic information are preserved. Here, the system employs the simplification: quadric edge collapse decimation algorithm [45], which reduces the number of vertices and faces by merging the two endpoints of an edge into a new vertex. The algorithm's most critical feature is the use of a quadric error metric, which enables the reduction of vertices and faces while preserving the key geometric features of the model as much as possible, achieving low-distortion simplification. The parameters used for edge-length filtering, isolated-piece removal, Taubin smoothing, and quadric edge collapse decimation at the four experimental sites are listed in Table 2. The parameters in Tables 1 and 2 were selected empirically through preliminary reconstruction trials by balancing outlier removal, terrain-detail preservation, mesh continuity, and Unity rendering efficiency.

Table 2
Parameters for edge length filtering, isolated piece removal, Taubin smoothing, and quadric edge collapse across four experimental sites.

Edge-length filtering threshold (model units)	Remove isolated pieces		Taubin smoothing			Quadric edge-collapse target face ratio (0–1)
	Absolute threshold	Percentage threshold	Lambda (λ)	Mu (μ)	Iterations	

	0.10	0.3297	1.5	0.9	0.2	10	0.4
Experiment site 1							
	0.10	3.3979	11.0	0.9	−0.1	10	0.2
Experiment site 2							
	0.15	2.8672	10.0	0.9	−0.1	10	0.2
Experiment site 3							
	0.80	1.7211	10.0	0.9	−0.1	10	0.2
Experiment site 4							

After obtaining the mesh, the 3D surface is unwrapped into a 2D UV space using MeshLab. For texture generation, the mesh surface is parameterized into a 2D UV domain, and barycentric coordinates are used to map triangle vertices and interpolate vertex colors within each texture triangle [46,47]. As illustrated in the inset of Fig. 2, each triangle on the 3D surface was mapped to a corresponding triangle in the UV domain. Suppose that the three vertices of a triangle are denoted as $V_1(x_1,y_1,z_1)$, $V_2(x_2,y_2,z_2)$, and $V_3(x_3,y_3,z_3)$. Then, any point in this triangle can be represented by the following formula, and each point is transformed into a $(\alpha,\ \beta,\ \gamma)$ coordinate:

$$P = \alpha V_1 + \beta V_2 + \gamma V_3 \ (\alpha + \beta + \gamma = 1, \alpha \geq 0, \beta \geq 0, \gamma \geq 0) \quad (7)$$

where $(\alpha,\ \beta,\ \gamma)$ are barycentric-coordinate weights that describe the relative contribution of the three triangle vertices to point P. To ensure accurate area mapping, each 3D triangle will be mapped to the corresponding 2D triangle according to the relative size of its area. The formula is as follows:

$$\vec{a} = V_2 - V_1 \quad (8)$$

$$\vec{b} = V_3 - V_1 \quad (9)$$

$$A_{3D} = \frac{1}{2} \| \vec{a} \times \vec{b} \| \quad (10)$$

$$A_{2D} = \frac{A_{3D}}{A_{total}} \times S \quad (11)$$

$$S = \text{textdim} \times \text{textdim} \quad (12)$$

If we assume that the total area of all triangles is A_{total} and the area of the current triangle is A_{3D}, then the area A_{2D} allocated to this triangle in the 2D parameterized domain can be expressed as follows. After determining the area occupied by the triangle in the 2D texture, they will be arranged sequentially in the 2D plane, with a reserved boundary (border = 2) between each other. After selection, an attempt can be made to correspond the three 3D triangle vertices $V_1(x_1,y_1,z_1)$, $V_2(x_2,y_2,z_2)$, and $V_3(x_3,y_3,z_3)$ to $U_1(u_1,v_1)$, $U_2(u_2,v_2)$, and $U_3(u_3,v_3)$.

Each point corresponding to a 2D triangle can be mapped using the following formula:

$$Q = \alpha U_1 + \beta U_2 + \gamma U_3 \quad (13)$$

After obtaining the texture mapping, the colors of the corresponding vertices are interpolated onto the texture map. Let $q = (u,v)$ denote a texture pixel inside a UV triangle, and let $U_1 = (u_0,v_0)$, $U_2 = (u_1,v_1)$, and $U_3 = (u_2,v_2)$ denote the three vertices of this triangle in the UV domain. The corresponding vertex colors are denoted as C_1, C_2, and C_3. The barycentric-coordinate weights λ_1, λ_2, and λ_3 represent the relative contributions of the three vertices to the color of q and satisfy $\lambda_1 + \lambda_2 + \lambda_3 = 1$. To avoid using undefined variables, λ_2 and λ_3 are first calculated as follows:

$$\lambda_2 = \frac{(u-u_0)(v_2-v_0)-(v-v_0)(u_2-u_0)}{(u_1-u_0)(v_2-v_0)-(u_2-u_0)(v_1-v_0)} \quad (14)$$

$$\lambda_3 = \frac{(v-v_0)(u_1-u_0)-(u-u_0)(v_1-v_0)}{(u_1-u_0)(v_2-v_0)-(u_2-u_0)(v_1-v_0)} \quad (15)$$

Journal Pre-proofs

$$\lambda_1 = 1 - \lambda_2 - \lambda_3 \quad (16)$$

After that, color proportion mixing is carried out according to the barycentric coordinates:

$$C_{\text{pixel}} = \lambda_1 C_1 + \lambda_2 C_2 + \lambda_3 C_3 \quad (17)$$

This triangle-wise parameterization and barycentric interpolation strategy helps allocate texture space according to the relative geometric area of each triangle, thereby improving texture utilization while preserving the visual consistency of the reconstructed agricultural scene.

2.3 2D video feedback and SLAM-based pose estimation

To achieve real-time environmental feedback and SLAM-based pose estimation for robotic operations, this study employs a depth camera as the core sensor. Unlike the RGB cameras used in conventional teleoperation technologies, the ECT-VR system utilizes a depth camera for three key reasons: First, it provides critical distance-sensing data for the design of the collision-warning module, which will be detailed in Section 2.5; second, it supports the operation of SLAM real-time positioning algorithms, accelerating the acquisition of environmental depth information through the camera's built-in active stereo vision hardware framework to obtain richer 3D spatial information, thereby improving the speed and quality of positioning; and third, it leverages hardware resources to their fullest extent by offloading 3D distance calculations to the depth camera's internal hardware and software. This approach reduces both system construction and computational costs while eliminating the need for additional hardware across multiple modules.

The Intel RealSense D455, mounted at the front of the robot, is used to continuously collect depth and color data of the forward scene. The Intel RealSense D455 is a depth camera based on active stereo vision technology, equipped with dual infrared stereo cameras and operating within a range of 0.6 to 6 m. In the robot operating system (ROS) environment, both the RGB and depth camera resolutions are set to 1280 × 720, with a frame rate configured at 30 frames per second (FPS). By launching the realsense2_camera driver package, the camera's RGB and depth information are published on the /camera/color/image_raw and /camera/depth/image_rect_raw topic, respectively. To facilitate transmission, the acquired images are compressed into JPEG format. While processing data, the system monitors all network interfaces and transfers data via the WebSocket protocol. The WebSocket supports full-duplex communication and was used for bidirectional data exchange between Unity and the robot-side ROS system, making it suitable for scenarios with high real-time requirements. The data receiver is integrated into the Unity3D engine on the server, and the received video data is visualized in the center of the screen to assist operators in understanding onsite conditions.

Beyond these functional requirements, the RealSense D455 was selected as the primary local exteroceptive sensing device on the robot side because the proposed ECT-VR system is designed as a human-in-the-loop teleoperation framework rather than a fully autonomous navigation system. The RGB-D camera simultaneously provides the color images, depth measurements, and geometric information required for real-time video feedback, RTAB-MAP-based pose estimation, and collision-risk visualization. Compared with a multi-sensor configuration involving lidar or ultrasonic sensors on the robot platform, the RGB-D-centered configuration reduces the hardware cost, calibration complexity, and data-synchronization burden. Since the robot operates at a relatively low speed under human supervision, the forward RGB-D sensing region provides adequate local perception for the main working area during teleoperation.

For operation-time robot pose estimation, the ECT-VR system also relies on the ROS framework and uses the RTAB-MAP algorithm [48], which subscribes to the depth camera's RGB image, depth image, and camera intrinsic parameter topics through ROS. The camera intrinsic parameters are used to correct image distortion and convert depth pixel values into 3D point-cloud coordinates (*X*, *Y*, *Z*). First, the algorithm extracts robot operating system (ORB) (oriented FAST and rotated BRIEF) keypoints [49] and binary robust independent elementary features (BRIEF) descriptors [50] from the corrected RGB images. Then, it maps the 2D coordinates of the key points to the 3D space through the camera intrinsic parameters, generating feature points with *XYZ* coordinates. During this process, the feature points corresponding to invalid values in the depth map are filtered out to ensure that only valid 3D features participate in subsequent processing. The BRIEF descriptors are used to match the ORB features of adjacent frames. After removing mismatched points using RANSAC, the camera pose transformation matrix is solved by the Perspective-n-Point (PnP) algorithm based on the correspondences between 3D spatial points and 2D image points, thereby obtaining the motion state of the robot between consecutive frames [51]. RTAB-MAP continuously optimizes the pose estimation results and reduces cumulative errors by maintaining key frames and using a loop-closure detection mechanism.

RTAB-MAP outputs the motion and orientation of the machine's current position in real time. These data are collected by the system via the nodes of the ROS system and then transmitted in real time to Unity's virtual environment through the WebSocket protocol, supporting pose synchronization between the physical robot and the virtual robot model in the Unity environment. In the proposed ECT-VR system, RTAB-MAP is used to provide operation-time relative pose feedback for virtual–physical synchronization and robot-position visualization in the VR interface. The pose-estimation subsystem serves as an enabling module for teleoperation interface synchronization rather than as a metric-grade localization solution for fully autonomous navigation. The above reconstruction, mesh-processing, and pose-estimation algorithms were adopted as

standard enabling methods in the ECT-VR workflow, and no modification was made to their core formulations.

2.4 Joystick control and body pose acquisition

The Pico 4 headset was used as the six-degree-of-freedom (6DoF) VR interaction terminal for immersive visualization and controller-based command input. Its hardware configuration includes dual 2.56-inch Fast-LCD displays with a combined resolution of 4320 × 2160 for both eyes and a 105° field of view (FOV), supporting refresh rates of up to 90 Hz. The Pico 4 was selected because its relatively lightweight headset body can reduce the burden of prolonged VR operation.

During operation, the Snapdragon XR2 chip built into the Pico 4 VR headset lacks sufficient computing power to handle large-scale 3D map rendering. Therefore, the system employs a wired streaming solution via a USB 3.2 Gen 2 interface, offloading rendering tasks to a high-performance server (configuration: Intel i9-14900KF, NVIDIA RTX 3090, 128 GB RAM). The server establishes connections through the Pico Developer Platform, implementing virtual scene rendering, sensor data fusion, and interaction logic processing based on the Unity engine.

To acquire the operator's control signals, ECT-VR primarily relies on the VR controllers and headset. The left and right VR controllers perform distinct control functions: The left-hand controller manages virtual viewpoint movement, enabling operators to navigate freely within large-scale 3D static maps, while the right-hand controller handles mobile robot control. Acquired commands are transmitted directly to the mobile robot via the WebSocket protocol to regulate its movement. Additionally, the inertial measurement unit (IMU) and inside-out tracking cameras integrated into the Pico 4 headset are used to estimate the operator's head position and orientation in real time, providing 6DoF pose information for immersive viewpoint control and interaction within the virtual environment. The controller tracking data and button states are also acquired as interaction inputs and transmitted to the Unity-based control interface. Fig. 3 illustrates the VR-driven joystick-control and pose-acquisition framework.

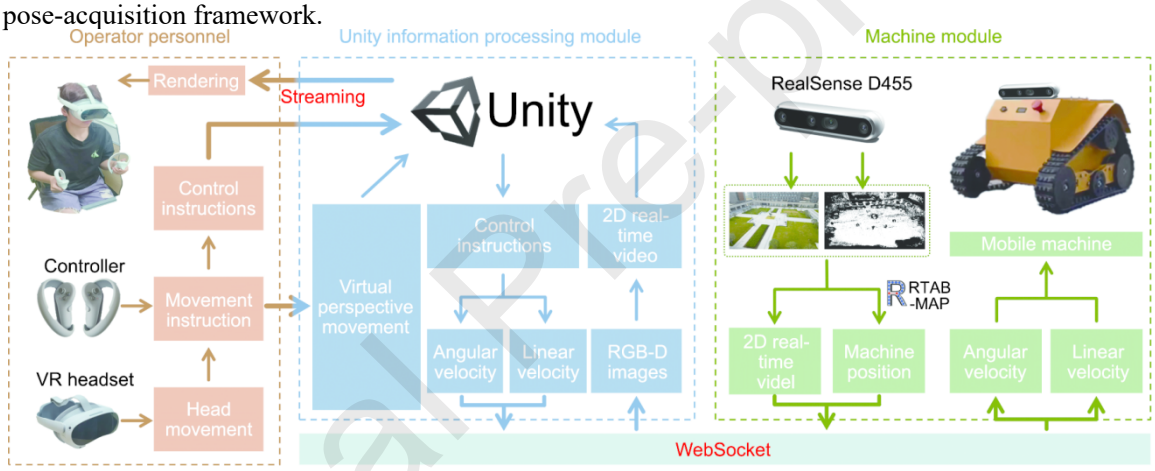


Fig. 3. VR-driven joystick-control and pose-acquisition framework for mobile machines.

2.5 Data-processing system on Unity3D

The deployment of the information-processing system primarily relies on the Unity3D engine. Due to its excellent image rendering, interactive development, and physical engine optimization, Unity3D is used for large-scale 3D scene rendering and real-time data processing. Unlike conventional teleoperation, this system primarily utilizes Unity's asynchronous rendering pipeline and data parallel processing framework. In terms of VR connection, the system employs the Unity XR Interaction Toolkit to construct the VR interaction framework. The communication link between the VR end and the Unity end is established using the PICO Developer Center software and Unity Live Preview Plugin, enabling bidirectional data synchronization between the Unity engine and the PICO 4 headset. Interactive data such as the headset's IMU pose (6DoF) and controller button states are transmitted in real time to the server-side Unity scene, while the virtual scene rendered by Unity is encoded into an H.264 stream and transmitted to the headset display via a USB 3.2 Gen 2 interface. The PICO Unity Integration SDK resource package provides controller model data and precise interaction mapping parameters for VR development.

When communicating with the mobile robot side, the system implements a Unity-side communication script based on the WebSocket protocol for bidirectional data exchange with the robot-side ROS system. This script receives robot-side feedback data and transmits control commands from Unity to the mobile robot during operation.

The system integrates a collision-avoidance warning module that leverages the depth camera on the robot side for information acquisition. Specifically, it relies on the depth map from the depth camera to calculate the distances between the robot and objects in the working area ahead. Subsequently, threshold segmentation is applied to generate masks and process the depth data. Objects that are too close to the robot are identified by their positions in the depth map, which are then mapped to corresponding locations in the RGB image. Such objects trigger an alarm, highlighted in red on the RGB display.

It should be noted that the depth camera is used to monitor the forward working area rather than to provide complete 360°

environmental coverage. To reduce the influence of the near-field blind region, the warning threshold was set greater than the minimum reliable sensing distance of the D455, so that potential obstacles can be highlighted before they enter the near-field region where depth measurements become less reliable. In addition, the camera was mounted at the front of the robot to cover the main driving direction. The preconstructed 3D global map and the real-time video stream further allow the operator to inspect static terrain risks and local dynamic conditions before issuing motion commands. In this way, blind-spot risks are mitigated through the combination of forward depth sensing, conservative warning design, global map observation, real-time visual feedback, and human supervision. In the current prototype, the red-highlight collision warning was triggered using a fixed safety-distance threshold of 0.7 m. When the measured depth value of an object or local region in the forward depth image was smaller than 0.7 m, the corresponding region was mapped onto the RGB image and highlighted in red in the VR interface. This threshold was empirically selected according to the reliable sensing range of the RealSense D455 and the low-speed human-supervised teleoperation setting, and it was fixed rather than adaptive or scenario-dependent during the experiments.

Although the preconstructed 3D map provides useful global spatial context, it should be regarded as a static environmental prior rather than a complete real-time representation of all field elements. In practical agricultural environments, moving machinery, personnel, animals, and temporarily placed tools may cause discrepancies between the preconstructed map and the current field state. In the current ECT-VR framework, this risk is mitigated by combining the static global map with real-time RGB-D video feedback, depth-based collision-warning visualization, and human-in-the-loop supervision. The operator can use the global map for route-level spatial understanding while relying on the real-time local video and depth information to identify temporary or moving obstacles near the robot. When a clear mismatch between the static map and the live RGB-D feedback is observed, the operator should rely primarily on the live feedback for immediate decision-making and trigger a map refresh before subsequent operation if the field structure has changed substantially.

Additionally, to enhance the operator's spatial awareness, the system implements a global positioning function. By processing the robot's positional data, it provides real-time updates of the vehicle's location in the upper-right corner of the interface. The vehicle's position is displayed on a local mini-map from a top-down global perspective, with the robot represented as a prominent red dot to facilitate intuitive spatial orientation.

For the 3D visualization of control commands, the system translates virtual viewpoint movements controlled by the left controller and the motion commands of the mobile robot issued via the right controller into arrow-shaped indicators of distinct colors. These arrows appear at the center of the interface only when commands are issued, remaining hidden otherwise to avoid obstructing normal observation. This visualization method directly and intuitively represents the operator's control intentions, clarifying the spatial characteristics of operational commands.

2.6 Mobile machinery control

To establish communication between the control terminal and the robotic platform, the system employs the WebSocket protocol. The joystick actions on the VR controller are detected in real time by Unity, with forward/backward movements corresponding to the robot's linear velocity and left/right movements to its angular velocity. Unity parses these joystick inputs into control commands, converts them into ROS-compliant velocity messages encompassing linear velocity and angular velocity, and publishes them via the cmd_vel topic.

The robot terminal subscribes to the cmd_vel topic through an ROS node to receive velocity commands in real time. Based on the physical parameters of the device, such as the distance between the left and right tracks, gear-reduction ratio, and wheel radius, this node calculates the target rotational speeds of the left and right wheels from the linear and angular velocities, converts them into control signals in hexadecimal format, and sends them to the motor driver via a serial port, ultimately driving the robot to move.

To provide feedback on the actual robot motion state, the system does not use the theoretical values of control commands as feedback for motion states; instead, it relies on the RTAB-MAP algorithm to estimate the actual motion of the robot in real time. This approach helps the interface present the estimated actual motion state of the robot rather than only the commanded motion. In the current prototype, this pose feedback was mainly used for visualization and virtual–physical synchronization.

3. Results and discussion

3.1 Front-end processing

To evaluate the adaptability of the proposed reconstruction and visualization workflow, tests were conducted at four agricultural and field-relevant outdoor sites with different scene structures. These experiments were designed as a preliminary feasibility-oriented validation of the ECT-VR workflow rather than as a comprehensive agronomic field trial across different crop types, seasons, and production operations. Experiment site 1 was a small garden between buildings, experiment site 2 was a wooded area, experiment site 3 was a park-like open scene, and experiment site 4 was a field environment containing terrain variations and ditch-like obstacles. These sites were selected to represent different levels of scene scale, vegetation density, terrain complexity, and reconstruction difficulty. The reconstruction-related evaluation focused on UAV image acquisition, camera-pose estimation, point-cloud reconstruction, mesh post-processing, and Unity-based visualization across these representative environments.

The dataset for this study was collected using a DJI MINI 3 Pro drone, and the acquired data were comprehensively processed by the system. The processing workflow relied primarily on an RTX 3090 GPU for computations, with 3D visualization implemented via Unity 3D. For the videos captured by the drone, images with a resolution of 1280 × 960 pixels were extracted to serve as input for subsequent algorithmic computations.

Fig. 4 illustrates the sparse point clouds and camera poses computed using COLMAP. Different columns in the figure correspond to the reconstruction results from different experimental sites, where points of varying colors denote key feature points involved in camera-pose estimation, and the red camera coordinate frames represent the relative poses of the cameras. Different imaging methods were employed according to the site characteristics, including rotational scans and raster scans. The UAV acquisition strategy was adjusted according to the spatial structure and occlusion characteristics of each site. For experiment site 1, a rotational scan was adopted around the central garden area, and images were acquired from two circular trajectories at different viewing heights to improve coverage of both the target area and the surrounding vertical structures. For experiment site 2, a rotational scan was also used around the wooded area to reduce occlusion among tree trunks and canopies caused by single-view acquisition. For experiment site 3, a raster scan was adopted because the scene was larger and more elongated, containing mixed terrain elements such as trees, depressions, reed-like vegetation, and wetland-like terrain. For experiment site 4, which represented the field-validation environment, a raster scan was used to cover the field terrain and ditch-like risk areas. Therefore, the red camera-pose symbols in Fig. 4 not only indicate the estimated camera poses from COLMAP but also provide a visual record of the actual image-acquisition coverage and scan pattern for each site.

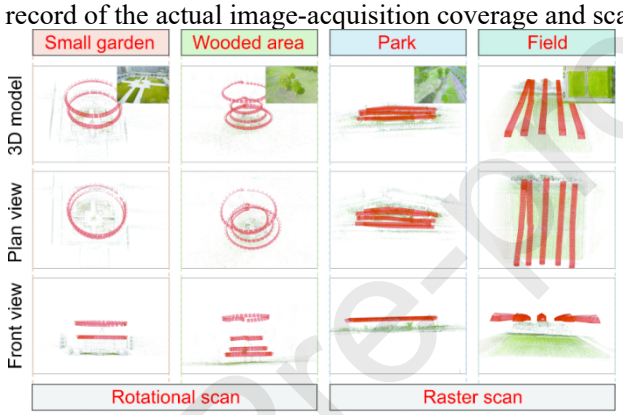


Fig. 4. COLMAP-derived camera pose visualization across scales, scan modes, and views.

3.2 Point-cloud map and post-processing

Following processing by Fast-MVSNet, corresponding depth maps were generated for each input image. These depth maps were spatially aligned with the input images on a pixel-by-pixel basis, such that each pixel's RGB value uniquely corresponded to a depth value. Simultaneously, a pixel-wise confidence map was generated for each depth map to characterize the reliability of the depth values. The information for each pixel could thus be represented as a triple—RGB, D, and C —where D denotes the depth value and C denotes the confidence value. During the subsequent point-cloud fusion stage, the depth data were processed based on the confidence maps to mitigate the influence of low-reliability measurements.

The first part of Fig. 5 demonstrates the fusion results. For large-scale scenes, the algorithm generated initial dense point-cloud data, clearly preserving geometric features such as road morphology and terrain undulations in the scene. However, the initial point cloud produced by the algorithm contained certain outliers, primarily originating from sensor measurement noise, stereo matching errors, and pseudo-reconstruction in occluded areas caused by the drone's viewing angle limitations. As shown in Fig. 5, discrete noise points at the boundaries of distant buildings and pseudo-points in tree regions were effectively removed after filtering, improving the purity of the point cloud. To improve the traceability of the visual comparison, zoomed-in detail insets are included in Fig. 5 to highlight representative regions affected by boundary noise, vegetation-related pseudo-reconstruction, and outlier removal after statistical filtering. The target scenarios are unstructured agricultural scenes, where the point-cloud density is highly non-uniform, and noise and valid local areas are constantly mixed. Therefore, using statistical filtering is reasonable and robust. Since the Poisson reconstruction algorithm relies on point-cloud data with normal information, outliers can lead to normal estimation bias, causing errors such as local protrusions and non-manifold connections in the reconstructed surface. The above filtering process improved the quality of the input point cloud and provided a cleaner basis for subsequent mesh reconstruction, helping the Poisson reconstruction preserve the main geometric structure of the scene.

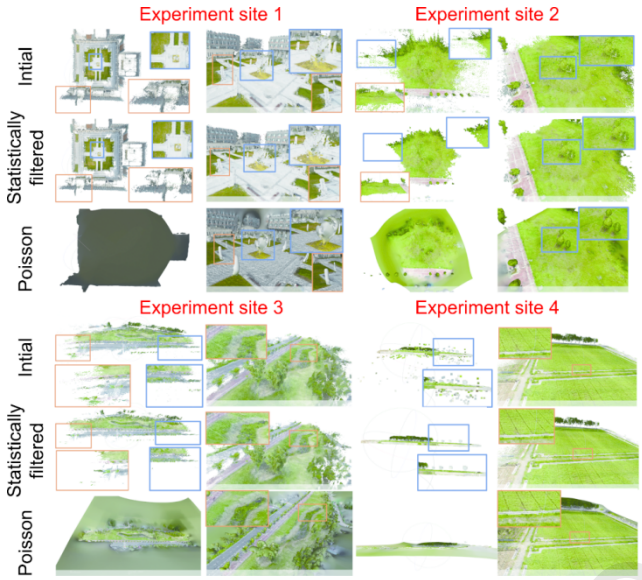


Fig. 5. Point-cloud reconstruction and filtering results across four experimental sites, including (clockwise starting top left for each set of images) initial point clouds, statistically filtered point clouds, Poisson-reconstructed meshes, and zoomed-in detail insets. The enlarged regions highlight representative boundary noise, vegetation-related pseudo-reconstruction, and local improvements after filtering.

Fig. 5 demonstrates the model results after Poisson reconstruction. Poisson surface reconstruction estimates a continuous implicit function from oriented point clouds and therefore tends to generate continuous or partially closed surfaces when the input data contain incomplete observations, sparse boundaries, or surrounding vertical structures. Thus, the "capping" phenomenon observed in experiment site 1 is mainly a general algorithmic tendency of Poisson reconstruction under bounded or incomplete point-cloud conditions, rather than a problem unique to agricultural scenes. In this experiment, the surrounding high-rise buildings and the target scene were jointly involved in the reconstruction, causing a closed surface to be generated above the model. This effect is less critical in typical flat and open farmland, where tall enclosing structures are absent and the ground surface can be more continuously captured by UAV images. Possible artifacts are more likely to appear near sparse boundaries rather than over the main robot operating area. It can be further mitigated by region-of-interest cropping, increasing oblique-view UAV images, removing long faces and isolated components, and applying mesh cleanup before Unity rendering.

Fig. 6 shows the mesh optimization workflow after Poisson reconstruction, where the first two rows correspond to the results of Poisson cleanup and isolated faces removal, respectively. These two operations aim to address the issues of excessive meaningless faces and model closure caused by Poisson reconstruction; that is, in Unity 3D rendering, closed surfaces can form optical closed shells, leading to incorrect light irradiation inside the scene and degraded global illumination.

Journal Pre-proofs

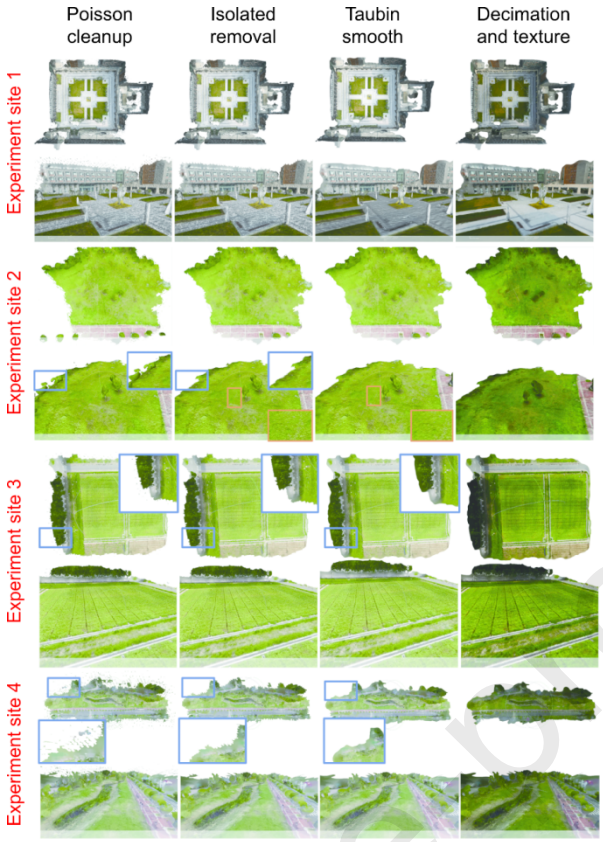


Fig. 6. Mesh post-processing results at four experimental sites, including Poisson cleanup, isolated-face removal, Taubin smoothing, decimation, and texture mapping.

As shown in Fig. 6, due to the large scene scale and irregular sampling of point-cloud data, local geometric errors were inevitably introduced, leading to high-frequency minor undulations on the model surface. To address this issue, the Taubin smooth non-uniform mesh-smoothing algorithm was applied to process the model. As illustrated in Fig. 6, after processing, noise on the ground surface was effectively suppressed, and surface continuity was significantly improved, better approximating the smooth characteristics of real terrain. During the mesh optimization stage, decimation was performed on the model. Although the number of polygons was reduced, the decimation operation had minimal impact on the overall scene quality. The processed model still retained abundant structural details and key semantic information.

After this processing and mesh optimization, the model more accurately reflects the true characteristics of the original data, with redundant noise and non-representative geometric information effectively removed. This treatment not only reduces the computational and storage overheads caused by high-frequency geometric fluctuations but also improves the efficiency of real-time rendering in Unity, avoiding unnecessary resource consumption during the rendering process.

To display the processed 3D scene model, it was imported into the Unity Engine for visualization rendering. Considering the natural characteristics of agricultural scenes, directional light illumination was not adopted; instead, ambient light based on the universal render pipeline was employed for global illumination calculations. As shown in Fig. 7, the reconstructed model reproduced the main geometric features and terrain undulations in Unity, supporting its use as a VR-renderable global scene representation for teleoperation.

3.3 System stereoscopic information

Fig. 7 presents the main stereoscopic information-display functions of the ECT-VR interface. Figs. 7(a) and (b) show Unity-rendered scene visualizations from two representative viewpoints, demonstrating the reconstructed agricultural scene presented to the operator. Figs. 7(c) and (d) show VR-view screenshots after the reconstructed scene are streamed to the headset. Each VR screenshot contains the left- and right-eye rendered views, thereby providing binocular parallax and enhancing the operator's sense of presence in the reconstructed environment.

To balance environmental information integrity with interaction-prompt clarity, the motion-indication arrows are rendered with 50% transparency. The arrows are displayed only when the corresponding commands are activated and remain hidden when neither the virtual viewpoint nor the physical robot is being controlled. Figs. 7(e) and (f) show magnified regions cropped from Figs. 7(c) and (d), respectively, to provide a clearer view of the motion-indication arrows. Blue arrows indicate pan or rotation commands of the virtual viewpoint, whereas green arrows indicate motion commands issued to the physical robot. This design allows the operator to understand the command direction without obstructing important environmental information.

Figs. 7(g) and (h) illustrate the global map module designed in this study, which draws inspiration from the minimap mechanism in game interactions. The global map is fixed in the upper-right corner of the interface as a 2D thumbnail by default and presents a bird's-eye view of the scene. Operators can enlarge the map by pressing a controller button when necessary. The red circular marker represents the real-time robot position, and its coordinates are dynamically synchronized with the 3D model in the scene. This design helps operators quickly establish the spatial correspondence between the virtual map and the physical world.

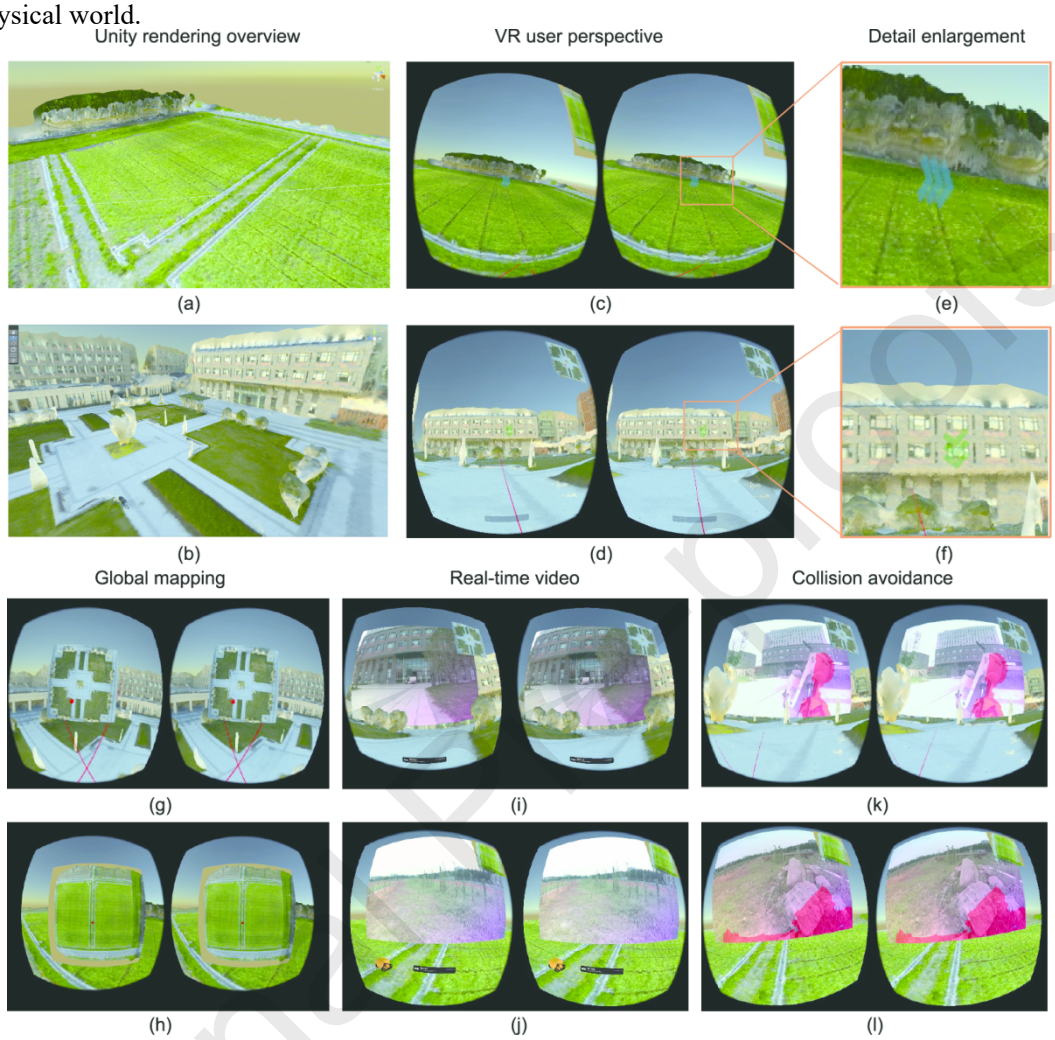


Fig. 7. Stereoscopic information display of the ECT-VR system: (a, b) Unity-rendered scene visualizations; (c, d) VR-view screenshots; (e, f) magnified views of motion-indication arrows; (g, h) global map display; (i, j) real-time 2D video display; (k, l) collision-warning visualization.

Figs. 7(i) and (j) show the real-time 2D video display integrated into the ECT-VR interface. Operators can obtain local environmental feedback from the robot's working area through the video stream while maintaining global position awareness through the reconstructed 3D map. This hybrid 3D+2D display mode supports both global scene understanding and local dynamic observation, thereby providing the operator with complementary spatial and real-time information in complex agricultural scenarios.

In the collision-warning module, the system uses depth information to identify objects within the predefined safety threshold and maps the corresponding warning regions onto the RGB display. As shown in Figs. 7(k) and (l), potential risk areas are highlighted in red to improve the operator's recognition of nearby obstacles. Through this stereoscopic information-display design, risk-related cues are presented directly in the operator interface, which can assist operators in environmental monitoring and risk judgment during teleoperation.

The hybrid 3D+2D representation also changes the way environmental information is delivered to the operator. The 3D map supports global comprehension of the scene, including terrain layout, relative robot position, and possible route choices, while the 2D RGB-D video provides time-sensitive local information from the robot's working area. The pose-synchronized virtual robot and motion-command indicators further connect the operator's control intention with the robot's physical state. Therefore, the interface is intended to support not only visual immersion but also operator situational awareness and decision preparation during agricultural teleoperation.

During teleoperation, the runtime behavior of the ECT-VR interface was recorded to support the description of operation-time feedback. The robot-side RGB-D camera was configured at 1280 × 720 pixels and 30 FPS, and the Pico 4 headset was operated at 72 Hz through a USB 3.2 Gen 2 wired streaming link. The Unity server was equipped with an Intel i9-14900KF CPU, an NVIDIA RTX 3090 GPU, and 128 GB RAM, while robot–server communication was implemented using the

WebSocket protocol. Based on timestamp logs recorded at ROS transmission, Unity reception, and Unity display update, the WebSocket transmission latency was (15.3 ± 6.0) ms, and the ROS-to-Unity interface display latency was (25.5 ± 5.2) ms. The 2D RGB-D stream was therefore used as an operation-time feedback stream, whereas the UAV-based 3D map was generated outside the real-time control loop and updated periodically or when substantial field-structure changes occurred.

3.4 System validation

To further evaluate the practical feasibility and user-centered experience of the proposed system, a user-centered validation was conducted at experimental site 4. This site was selected because it contained representative field terrain, route-selection requirements, and ditch-like risk areas, making it suitable for evaluating spatial observation, risk perception, and robot-control assistance during agricultural teleoperation. The system-validation scenarios and representative route-selection cases are shown in Fig. 8. In Fig. 8(c), the dot denotes the start point and the star denotes the target point. Figs. 8(d)–(f) show representative suboptimal routes observed during the user experiment, including an incorrect route and excessively long routes.

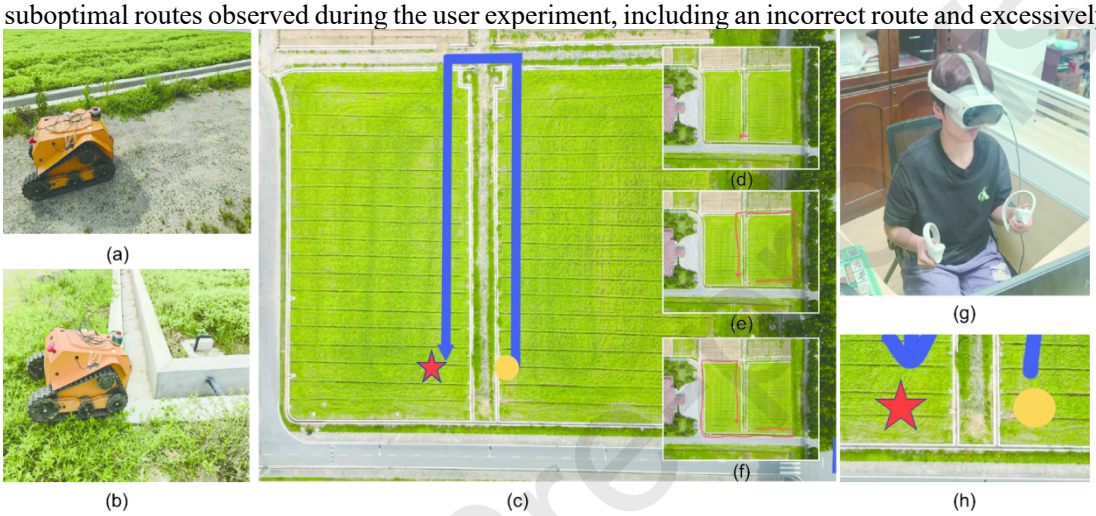


Fig. 8. System-validation scenarios: (a) normal operation; (b) ditch encounter; (c) planned route; (d) incorrect route; (e, f) excessively long routes; (g) operator control; (h) detailed view of the start–target region.

A total of 20 participants were recruited from Nanjing Agricultural University and the surrounding student population. All participants had normal or corrected-to-normal vision, and their ages ranged from 19 to 24 years. Fourteen participants reported prior experience with VR devices, nine had experience with robot teleoperation or game-controller-based control, and five had agricultural fieldwork familiarity. All participants were informed of the experimental procedure before the test, provided written consent, and participated voluntarily. The participants were required to control an all-terrain vehicle from the position marked by a circle to the position marked by a star in Fig. 8. In addition to using the ECT-VR interface, participants were exposed to a conventional video-streaming teleoperation mode on the same start–target task so that they could provide a perceived comparison between the two interaction modes. Since this study focused on initial user-centered validation, the questionnaire results are interpreted as perceived usefulness and interface-support evidence rather than as a task-level statistical comparison. Subsequently, a structured questionnaire was used to evaluate the main user-centered aspects of the ECT-VR system, including comfort, immersion, map fidelity, risk perception, virtual-scene exploration, perceived operational support, interface comprehensibility, global positioning, collision-warning visualization, and motion-command indication. The questions were as follows:

(1) Did you experience any discomfort during or after the VR experience?
(2) How strongly did you feel the “presence” and the sense of reality of events in the VR scenario?
(3) How similar were the colors and textures of the virtual map to the real scene?
(4) Did you notice any potential risks while using the system?
(5) Was the function of virtually entering the experimental scenario and moving freely useful for your operations?
(6) Was operating the robot through the virtual interface more effective compared with the traditional video-streaming interface?
(7) Do you think this system can save time during training, simulation, and control processes?
(8) Were the mini-map and 2D displays in the system easy to understand?
(9) Could the top-down map quickly locate the robot’s position in the global environment?
(10) Was the collision-warning function natural and non-intrusive?
(11) Did the directional arrows clearly indicate the machine’s motion state?

Responses were provided on a five-point Likert scale: very dissatisfied (1 point), dissatisfied (2 points), neutral (3 points), satisfied (4 points), and very satisfied (5 points). Although Item 1 was a discomfort-related item, it is reported as a comfort-related evaluation in the statistical summary, so that the scoring direction is consistent with those of the other items; therefore, a higher score indicates lower perceived discomfort and better comfort during and after VR use. The mean score, standard

deviation, and proportion of positive responses (scores ≥ 4.00) were calculated for each item to quantify the participants' evaluations. Table 3 presents the statistical results of the responses to each question.

Table 3
User-centered evaluation results of the ECT-VR system.

Item	Very satisfied (5 points)	Satisfied (4 points)	Neutral (3 points)	Dissatisfied (2 points)	Very dissatisfied (1 point)	Mean score	Standard deviation	Proportion of scores ≥ 4
(1)	13	3	4	0	0	4.45	0.82	80.0%
(2)	15	2	3	0	0	4.60	0.75	85.0%
(3)	14	3	3	0	0	4.55	0.75	85.0%
(4)	16	3	1	0	0	4.75	0.55	95.0%
(5)	18	2	0	0	0	4.90	0.31	100.0%
(6)	17	3	0	0	0	4.85	0.37	100.0%
(7)	20	0	0	0	0	5.00	0.00	100.0%
(8)	16	4	0	0	0	4.80	0.41	100.0%
(9)	18	2	0	0	0	4.90	0.31	100.0%
(10)	14	6	0	0	0	4.70	0.47	100.0%
(11)	16	4	0	0	0	4.80	0.41	100.0%

The questionnaire results indicate that the proposed ECT-VR system was positively evaluated by most participants. Across the 11 evaluation items, the overall mean score was 4.75, and the average proportion of positive responses (scores ≥ 4.00) reached 95.0%. Among all items, one item achieved a mean score of 5.00, and eight items achieved mean scores of ≥ 4.70. These results suggest that the system provided a favorable user experience in terms of immersion, functional practicality, interface comprehensibility, and perceived safety support.

The items related to virtual-scene exploration, perceived operational support relative to conventional video-streaming operation, and perceived time saving in training, simulation, and control obtained mean scores of 4.90, 4.85, and 5.00, respectively. These results indicate that participants perceived the VR-based interface and the hybrid 3D+2D information display as useful for supporting spatial understanding, route planning, and operation preparation. In addition, the items related to risk perception, global positioning, collision-warning visualization, and motion-command indication achieved mean scores of 4.75, 4.90, 4.70, and 4.80, respectively, indicating that the integrated assistance functions were generally considered clear, intuitive, and non-intrusive.

From the perspective of agricultural human–robot interaction, these results provide preliminary user-centered evidence that the hybrid representation was perceived as useful for supporting operator situational awareness and human-in-the-loop decision-making. The reconstructed 3D map helped participants understand the global spatial relationships between the starting point, target point, terrain, and potential risk areas, while the real-time 2D video, global positioning display, collision-warning visualization, and motion-command indicators supported local observation and control-state interpretation. During the experiment, operators could observe the reconstructed static scene from multiple viewpoints before controlling the machine. In several cases, this pre-observation process helped them avoid drainage ditches or correct inappropriate routes, which aligns with the high scores for virtual-scene exploration, global positioning, and collision-warning visualization.

It should be noted that pre-observation introduces an additional preparation-time cost before active robot control. Therefore, its benefit should be interpreted as a trade-off between initial observation time and the reduction of operational uncertainty, route correction, and potential error-recovery actions. The proposed pre-observation function is expected to be most beneficial in unfamiliar, obstacle-rich, or risk-prone agricultural tasks, such as operations involving drainage ditches, narrow traversable regions, irregular terrain, occluded areas, or high penalties for incorrect maneuvers. In contrast, for simple open-field tasks with clear routes and few obstacles, the additional pre-observation time may not always yield a substantial efficiency advantage. Thus, the advantage of ECT-VR is expected to be more evident in complex teleoperation scenarios where improved spatial understanding and safer decision-making can compensate for the extra observation time.

The current validation mainly provides preliminary user-centered evidence of perceived usefulness, interface clarity, and decision-support potential. A few participants also experienced dizziness and visual fatigue during VR operation. In addition to headset-related factors, these symptoms may be associated with virtual-environment factors such as vergence-accommodation conflict, visually induced motion sickness, and rendering or streaming latency. Future interface optimization will focus on reducing latency, smoothing viewpoint transitions, and improving stereo-depth presentation. In addition, obstacle-recognition and classification capabilities should be further enhanced. In future work, a counterbalanced within-

subject comparison between ECT-VR and conventional video-streaming teleoperation will be introduced. Standardized task-level indicators, such as task completion time, path efficiency or trajectory optimality, collision or near-collision events, operation errors, and operator workload metrics, will be recorded. Appropriate paired statistical tests will also be used to quantify performance differences under different teleoperation modes.

4. Conclusions

To address the issues of limited information acquisition, insufficient perception, and low operational efficiency in agricultural teleoperation, this study designed and implemented an enhanced teleoperation system called the ECT-VR. Through an integrated architecture of static 3D maps + dynamic 2D real-time video streams combined with VR immersive interaction, this system attempts to overcome the drawbacks of single-source data and the inability to display 3D information. It achieves large-scale scene modeling by periodically reconstructing static 3D maps via UAVs, and integrates real-time 2D video streams from depth cameras with real-time SLAM-based localization of the machine. This approach not only avoids the high computational load associated with real-time full-scene 3D reconstruction but also ensures timely feedback of local dynamic information.

ECT-VR integrates a 3D reconstruction and post-processing workflow adapted for agricultural outdoor scenes, leveraging COLMAP and Fast-MVSNet to acquire onsite depth data. A multi-stage refined post-processing pipeline has been designed that combines statistical filtering and geometric optimization to address challenges such as vegetation pseudo-reconstruction and terrain noise in agricultural environments, thereby providing a high-fidelity environmental foundation for VR interaction. The system employs decimation and texture mapping to reduce polygon complexity while preserving key terrain features and visual details of the reconstructed scene, improving the suitability of the model for Unity3D-based VR rendering. The VR interface enhances operators' spatial perception and operational intuitiveness by streamlining the display of situational and control information through features such as stereoscopic visualization, global positioning indicators, and collision-warning visualization. This immersive display solution, incorporating Unity engine rendering technology, overcomes the information-carrying limitations of traditional planar interfaces and offers operators a multidimensional information-interaction experience.

Field tests and user-centered evaluation indicate that ECT-VR can support environmental information visualization and robot motion control in the tested agricultural-related scenarios. User questionnaire data further indicate that the system's immersive interface and 3D information presentation can assist operators during teleoperation. Moreover, participants reported that pre-scene observation helped reduce operational uncertainty and avoid inappropriate routes. Although this process increased the initial observation time, it may support operation preparation and contribute to operational efficiency in complex or risk-prone tasks by reducing operational uncertainty, route corrections, error-recovery actions, and potential safety risks.

However, this system still has certain limitations, such as the real-time updating of 3D maps in dynamic environments and the stability of sensor data under extreme weather conditions, which have not been fully resolved. Future work will further optimize static map iteration strategies and improve the applicability of ECT-VR in highly dynamic agricultural scenarios by incorporating dynamic object detection and local map-updating mechanisms. For example, RGB-D-based semantic detection or tracking could be used to identify moving machinery, personnel, and temporary obstacles in the robot's working area. These dynamic elements could then be projected onto a short-term local occupancy map or displayed as an additional dynamic-risk layer in the VR interface. Periodic UAV-based map updates may also be introduced when large structural changes occur in the field environment, allowing the system to retain the global advantages of preconstructed 3D maps while improving its responsiveness to dynamic environmental changes.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (32472010), the Natural Science Foundation of Jiangsu Province (BK20231478), and the Jiangsu Agricultural Science and Technology Innovation Fund (JASTIF) (CX(24) 3027).

References

[1] Ghazal S, Munir A, Qureshi WS. Computer vision in smart agriculture and precision farming: techniques and applications. Artif Intell Agric 2024;13:64–83.

[2] Juwono FH, Wong WK, Verma S, Shekhawat N, Lease BA, Apriono C. Machine learning for weed-plant discrimination in agriculture 5.0: an in-depth review. Artif Intell Agric 2023;10:13–25.

[3] Pandey DK, Mishra R. Towards sustainable agriculture: harnessing AI for global food security. Artif Intell Agric 2024;12:72–84.

[4] Liu T, Zhang B, Tan Q, Zhou J, Yu S, Zhu Q, et al. Immersive human-machine teleoperation framework for precision agriculture: integrating UAV-based digital mapping and virtual reality control. Comput Electron Agric 2024;226:109444.

[5] Wang D, Zhang B, Zhou J, Xiong Y, Liu L, Tan Q. Three-dimensional mapping and immersive human-robot interfacing utilize Kinect-style depth cameras and virtual reality for agricultural mobile robots. J Field Robot 2024;41(7):2413–26.

[6] Li G, Li Q, Yang C, Su Y, Yuan Z, Wu X. The classification and new trends of shared control strategies in telerobotic systems: a survey. IEEE Trans Haptics 2023;16(2):118–33.

[7] Cho SK, Jin HZ, Lee JM, Yao B. Teleoperation of a mobile robot using a force-reflection joystick with sensing mechanism of rotating magnetic field. IEEE/ASME Trans Mechatron 2010;15(1):17–26.

[8] Labonte D, Boissy P, Michaud F. Comparative analysis of 3-D robot teleoperation interfaces with novice users. IEEE Trans Syst Man Cybern B Cybern 2010;40(5):1331–42.

[9] Yanco HA, Drury J. "Where am I?" acquiring situation awareness using a remote robot platform. In: Proceedings of the 2004 IEEE International Conference on Systems, Man and Cybernetics; 2004 Oct 10–13; The Hague, Netherlands. Piscataway: IEEE; 2004. p. 2835–40.

[10] Uddin Z, Islam M. Search and rescue system for alive human detection by semi-autonomous mobile rescue robot. In: Proceedings of the 2016 International Conference on Innovations in Science, Engineering and Technology; 2016 Dec 15–17; Dhaka, Bangladesh. Piscataway: IEEE; 2016. p. 1–5.

[11] Sharmin S, Salim SI, Sanim KRI. A low-cost urban search and rescue robot for developing countries. In: Proceedings of the 2019 IEEE International Conference on Robotics, Automation, Artificial-intelligence and Internet-of-Things; 2019 Dec 27–29; Dhaka, Bangladesh. Piscataway: IEEE; 2019. p. 60–4.

[12] Ismail RM, Muthukumaraswamy S, Sasikala A. Military support and rescue robot. In: Proceedings of the 2020 4th International Conference on Intelligent Computing and Control Systems; 2020 Jun 11–13; Madurai, India. Piscataway: IEEE; 2020. p. 156–62.

[13] Scholtz JC. Human-robot interactions: creating synergistic cyber forces. In: Schultz AC, Parker LE, editors. Multi-robot systems: from swarms to intelligent automata; 2001; Washington, DC, USA. Dordrecht: Springer Netherlands; 2002. p. 177–84.

[14] Drury JL, Scholtz J, Yanco HA. Awareness in human–robot interactions. In: Proceedings of the 2003 IEEE International Conference on Systems, Man and Cybernetics; 2003 Oct 5–8; Washington, DC, USA. Piscataway: IEEE; 2003. p. 912–8.

[15] Jacoff A, Messina E, Evans J. Reference test courses for autonomous mobile robots. In: Unmanned ground vehicle technology III; vol 4364; 2001 Aug 1–3; Orlando, FL, USA. Bellingham: SPIE; 2001. p. 341.

[16] Kim GH, Kim KD, Lee HS, Choi YS, Mun HS, Oh JH, et al. Development of Wi-Fi-based teleoperation system for forest harvester. J Biosyst Eng 2021;46(3):206–16.

Journal Pre-proofs

[17] Praveena P, Molina L, Wang Y, Senft E, Mutlu B, Gleicher M. Understanding control frames in multi-camera robot telemanipulation. In: Proceedings of the 2022 17th ACM/IEEE International Conference on Human-Robot Interaction; 2022 Mar 7–10; Tokyo, Japan. New York City: ACM; 2022. p. 432–40.

[18] Udekwe D, Seyyedhasani H. Human robot interaction for agricultural tele-operation, using virtual reality: a feasibility study. Comput Electron Agric 2025;228:109702.

[19] Shen X, Chong ZJ, Pendleton S, James Fu GM, Qin B, Frazzoli E, et al. Teleoperation of on-road vehicles via immersive telepresence using off-the-shelf components. In: Menegatti E, Michael N, Berns K, Yamaguchi H, editors. Intelligent autonomous systems 13; 2015 Jun 24–26; Padua, Italy. Cham: Springer International Publishing; 2015. p. 1419–33.

[20] Nakanishi J, Itadera S, Aoyama T, Hasegawa Y. Towards the development of an intuitive teleoperation system for human support robot using a VR device. Adv Robot 2020;34(19):1239–53.

[21] Park S, Jung Y, Bae J. An interactive and intuitive control interface for a tele-operated robot (AVATAR) system. Mechatronics 2018;55:54–62.

[22] Wei D, Huang B, Li Q. Multi-view merging for robot teleoperation with virtual reality. IEEE Robot Autom Lett 2021;6(4):8537–44.

[23] Cai Z, Han J, Liu L, Shao L. RGB-D datasets using microsoft kinect or similar sensors: a survey. Multimed Tools Appl 2017;76(3):4313–55.

[24] Yu S, Liu X, Tan Q, Wang Z, Zhang B. Sensors, systems and algorithms of 3D reconstruction for smart agriculture and precision farming: a review. Comput Electron Agric 2024;224:109229.

[25] Pace FD, Gorjup G, Bai H, Sanna A, Liarokapis M, Billinghurst M. Leveraging enhanced virtual reality methods and environments for efficient, intuitive, and immersive teleoperation of robots. In: Proceedings of the 2021 IEEE International Conference on Robotics and Automation; 2021 May 30–Jun 5; Xi'an, China. Piscataway: IEEE; 2021. p. 12967–73.

[26] Naceri A, Mazzanti D, Bimbo J, Tefera YT, Prattichizzo D, Caldwell DG, et al. The Vicarios virtual reality interface for remote robotic teleoperation. J Intell Robot Syst 2021;101(4):80.

[27] Zhou T, Zhu Q, Du J. Intuitive robot teleoperation for civil engineering operations with virtual reality and deep learning scene reconstruction. Adv Eng Inform 2020;46:101170.

[28] Holubek R, Ružarovský R, Delgado Sobrino DR. Using virtual reality as a support tool for the offline robot programming. Res Pap Fac Mater Sci Technol Slovak Univ Technol 2018;26(42):85–91.

[29] Van de Merwe DB, Van Maanen L, Ter Haar FB, Van Dijk RJE, Hoeba N, Van der Stap N. Human-robot interaction during virtual reality mediated teleoperation: how environment information affects spatial task performance and operator situation awareness. In: Chen JYC, Fragomeni G, editors. Virtual, augmented and mixed reality. applications and case studies; 2019 Jul 17–21; Orlando, FL, USA. Cham: Springer International Publishing; 2019. p. 163–77.

Journal Pre-proofs

[30] Pérez L, Diez E, Usamentiaga R, García DF. Industrial robot control and operator training using virtual reality interfaces. Comput Ind 2019;109:114–20.

[31] Lesniak K, Tucker CS. Dynamic rendering of remote indoor environments using real-time point cloud data. J Comput Inf Sci Eng 2018;18(3):031006.

[32] Wang C, Cho YK. Smart scanning and near real-time 3D surface modeling of dynamic construction equipment from a point cloud. Autom Constr 2015;49:239–49.

[33] Schönberger JL, Frahm JM. Structure-from-motion revisited. In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition; 2016 Jun 27–30; Las Vegas, NV, USA. Piscataway: IEEE; 2016. p. 4104–13.

[34] Lowe DG. Distinctive image features from scale-invariant keypoints. Int J Comput Vis 2004;60(2):91–110.

[35] Fischler MA, Bolles RC. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM 1981;24(6):381–95.

[36] Triggs B, McLauchlan PF, Hartley RI, Fitzgibbon AW. Bundle adjustment—a modern synthesis. In: Triggs B, Zisserman A, Szeliski R, editors. Vision algorithms: theory and practice; 1999 Sep 13–17; Corfu, Greece. Berlin: Springer; 2000. p. 298–372.

[37] Yu Z, Gao S. Fast-MVSNet: sparse-to-dense multi-view stereo with learned propagation and Gauss–Newton refinement. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020 Jun 14–19; Seattle, WA, USA. Piscataway: IEEE; 2020. p. 1946–55.

[38] Yao Y, Luo Z, Li S, Fang T, Quan L. MVSNet: depth inference for unstructured multi-view stereo. In: Ferrari V, Hebert M, Sminchisescu C, Weiss Y, editors. Computer vision—ECCV 2018; vol 11212; 2018 Sep 8–14; Munich, Germany. Cham: Springer International Publishing; 2018. p. 785–801.

[39] Chen R, Han S, Xu J, Su H. Point-based multi-view stereo network. In: Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision; 2019 Oct 27–Nov 2; Seoul, Republic of Korea. Piscataway: IEEE; 2019. p. 1538–47.

[40] Jensen R, Dahl A, Vogiatzis G, Tola E, Aanæs H. Large scale multi-view stereopsis evaluation. In: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition; 2014 Jun 23–28; Columbus, OH, USA. Piscataway: IEEE; 2014. p. 406–13.

[41] Yan J, Wei Z, Yi H, Ding M, Zhang R, Chen Y, et al. Dense hybrid recurrent multi-view stereo net with dynamic consistency checking. 2020. arXiv:2007.09728.

[42] Rusu RB, Cousins S. 3D is here: point cloud library (PCL). In: Proceedings of the 2011 IEEE International Conference on Robotics and Automation; 2011 May 9–13; Shanghai, China. Piscataway: IEEE; 2011. p. 1–4.

Journal Pre-proofs

[43] Kazhdan M, Bolitho M, Hoppe H. Poisson surface reconstruction. In: Proceedings of the Fourth Eurographics Symposium on Geometry Processing; 2006 Jun 26–28; Goslar, Germany. Goslar: Eurographics Association; 2006. p. 61–70.

[44] Taubin G. Curve and surface smoothing without shrinkage. In: Proceedings of the 1995 IEEE International Conference on Computer Vision; 1995 Jun 20–23; Cambridge, MA, USA. Piscataway: IEEE; 1995. p. 852–857.

[45] Garland M, Heckbert PS. Surface simplification using quadric error metrics. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques; 1997 Aug 4–8; Los Angeles, CA, USA. New York City: ACM Press/Addison-Wesley Publishing Co.; 1997. p. 209–16.

[46] Shirley P, Ashikhmin M, Marschner S. Fundamentals of computer graphics. 3rd ed. New York City: A K Peters/CRC Press; 2009.

[47] Heckbert PS. Fundamentals of texture mapping and image warping. Report. Berkeley: University of California, EECS Department; 1989 Jun.

[48] Labbé M, Michaud F. RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. J Field Robot 2019;36(2):416–46.

[49] Rublee E, Rabaud V, Konolige K, Bradski G. ORB: an efficient alternative to SIFT or SURF. In: Proceedings of the 2011 International Conference on Computer Vision; 2011 Nov 6–13; Barcelona, Spain. Piscataway: IEEE; 2011. p. 2564–71.

[50] Calonder M, Lepetit V, Strecha C, Fua P. BRIEF: binary robust independent elementary features. In: Daniilidis K, Maragos P, Paragios N, editors. Computer vision—ECCV 2010; vol 6311; 2010 Sep 5–11; Heraklion, Greece. Berlin: Springer; 2010. p. 778–92.

[51] Lepetit V, Moreno-Noguer F, Fua P. EPnP: an accurate O(n) solution to the PnP problem. Int J Comput Vis 2009;81(2):155–66.

Declaration of Interest Statement

The authors declare that they have no know competing financial interests or personal relationships that could have appeared of influence the reported in this paper.

Journal Pre-proofs