

PAPER

Digital Twin and Mobile Touch Interaction for High-Rise Construction Visualization and Collaboration

Wenkai Jiang ,
 Yigang You  

Guangxi University of
 Finance and Economics,
 Nanning, China

YouYG@gxufe.edu.cn

ABSTRACT

Mobile digital twin systems for construction progress management are constrained by low interaction efficiency, as touch gestures are typically limited to viewpoint manipulation and are not semantically linked to progress-management operations. To address this limitation, a semantic mapping method from touch gestures to construction progress states was proposed. A bidirectional mapping model between gesture kinematic features and construction progress semantic vectors was established, whereby touch interaction was extended from conventional viewpoint manipulation to semantic querying and collaborative command execution. A seven-dimensional gesture feature vector and a five-dimensional progress semantic vector were constructed, and a mapping matrix was trained through supervised learning. An inverse mapping mechanism was further designed to drive visual feedback in the digital twin model, thereby forming a closed interaction loop. The mapping rules were designed to support key interaction scenarios: two-finger pinch gestures were mapped to switching between detailed and overview representations of construction progress; single-finger long presses were used to trigger progress information cards and initiate collaboration; two-finger rotation gestures were mapped to navigation along the four-dimensional timeline; and three-finger downward swipes were used to activate comparisons between planned and actual progress. Incremental learning was additionally incorporated to enable personalized adaptation. A prototype system was developed using the web graphics library, while Draco compression and level-of-detail rendering were employed to address the performance constraints associated with Building Information Modeling (BIM) models on mobile devices. Experimental results demonstrated that the proposed method effectively improved interaction efficiency and facilitated on-site collaboration in mobile digital twin systems.

KEYWORDS

digital twin, touch interaction, construction progress visualization, semantic mapping, mobile computing

Jiang, W., You, Y. (2026). Digital Twin and Mobile Touch Interaction for High-Rise Construction Visualization and Collaboration. *International Journal of Interactive Mobile Technologies (IJIM)*, 20(19), pp. 62–76. <https://doi.org/10.3991/ijim.v20i19.63335>

Article submitted 2026-06-24. Revision uploaded 2026-08-14. Final acceptance 2026-08-25.

© 2026 by the authors of this article. Published under CC-BY.

1 INTRODUCTION

High-rise building construction is characterized by extensive work areas, interdependent and overlapping construction activities, numerous participating stakeholders, and complex information flows. Consequently, precise construction progress control and efficient on-site collaboration have become critical challenges in project management [1]. Through the establishment of bidirectional mapping between the physical construction site and its virtual model, digital twin technology has provided a new approach to construction process visualization and intelligent management [2]. Meanwhile, the means by which information is accessed on construction sites are being fundamentally transformed by mobile smart devices [3]. Owing to their portability and real-time accessibility, smartphones and tablets have increasingly been adopted as effective terminals for connecting on-site personnel with back-end management systems [4]. However, existing mobile construction management applications remain heavily dependent on menu-based navigation and form-based data entry, requiring users to switch frequently among model visualization, construction progress querying, data reporting, and collaboration initiation [5]. In high-rise construction sites, where work is characterized by fragmented time and highly divided attention, such interaction patterns substantially constrain the practical value of digital twin technology. Multi-touch capabilities provided by mobile devices offer a technological basis for overcoming these limitations [6]. Nevertheless, semantic associations between touch interaction and construction progress management have not yet been adequately established. Touch gestures remain largely restricted to viewpoint manipulation and have not been transformed into intuitive input mechanisms for semantic construction progress queries or collaborative commands.

Research on digital twin-enabled construction management has been conducted from multiple perspectives [7–10]. In the field of digital twin-based construction visualization, dynamic digital representations have been constructed by integrating Building Information Modeling (BIM) with real-time construction data [11]. However, most existing systems have been designed primarily for personal computers or augmented reality head-mounted devices, and their interaction efficiency is substantially reduced when transferred to mobile platforms [12]. For mobile BIM applications, BIM-U, developed by Zaher et al., enables on-site personnel to update construction progress and synchronize the updates with the BIM model [13]. Augmented Reality for Construction (AR4C), developed by Ratajczak et al., supports on-site model overlay visualization and task management; however, interaction remains predominantly dependent on point-and-select operations and menu switching. Pham Van et al. [14] further identified limited three-dimensional graphics processing capabilities and information overload on mobile devices as major barriers to effective mobile BIM applications [14]. In touch gesture interaction, a gesture-driven architecture for digital twin manufacturing cells was proposed by Zhang et al. [15], while vCapTouch was developed to support touch-sensing data synthesis and gesture recognition [16]. Nevertheless, the mapping mechanisms employed in these approaches are generally limited to direct correspondences between gestures and operational commands, with the gestures themselves carrying no domain-specific semantics. In construction progress management, a domain-semantic mapping model between touch gestures and construction progress states has yet to be established. Overall, three critical research gaps can be identified. First, existing interaction paradigms remain predominantly

menu-driven, preventing the interaction capabilities of multi-touch interfaces from being fully exploited. Second, semantic coupling between touch gestures and construction progress management remains insufficient. Third, an integrated technical framework connecting gesture sensing, semantic mapping, and on-site collaboration has yet to be established.

To address these research gaps, a semantic mapping method from touch gestures to construction progress states is proposed. A bidirectional mapping model between gesture kinematic features and construction progress semantics is established, enabling a single touch interaction to simultaneously support viewpoint manipulation, semantic querying, and collaborative command execution. The principal contributions are as follows: a formal model for mapping touch gestures to construction progress semantics is established; a mapping rule library tailored to construction-site scenarios is designed, incorporating key gestures such as two-finger pinch, single-finger long press, two-finger rotation, and three-finger downward swipe; a mobile digital twin prototype system is developed by integrating lightweight model rendering, real-time gesture recognition, semantic mapping inference, and cloud-based collaboration modules; and the effectiveness of the proposed method is experimentally validated.

The remainder of the study is organized as follows. Section 2 presents the overall framework and core techniques of the Gesture-Progress Semantic Mapping (GPSM) method. Section 3 describes the implementation of the prototype system. Section 4 reports the experimental evaluation. Finally, Section 5 discusses the limitations and future research directions and summarizes the main findings.

2 SEMANTIC MAPPING FROM TOUCH GESTURES TO CONSTRUCTION PROGRESS STATES

2.1 Overall architecture

The Gesture-Progress Semantic Mapping (GPSM) method is structured into four layers: the perception layer, mapping layer, application layer, and collaboration layer. At the perception layer, raw touch-point sequences $P = \{(x_1, y_1, t_1), (x_2, y_2, t_2), \dots, (x_n, y_n, t_n)\}$ are captured through the native multi-touch interface of the mobile operating system. After being smoothed using Kalman filtering, gesture kinematic features are extracted to construct a seven-dimensional feature vector $h \in R^7$. At the mapping layer, a bidirectional mapping relationship between the gesture feature space $H \subset R^7$ and the construction progress semantic space $S \subset R^5$ is maintained. A forward mapping matrix $W_s \in R^{5 \times 7}$ is employed to transform gesture features into construction progress semantics, whereas an inverse mapping matrix $W_h \in R^{7 \times 5}$ is used to transform semantic changes into visual feedback parameters. At the application layer, the resulting five-dimensional semantic vector $s \in R^5$ is received and used to drive visualization updates in the digital twin model. At the collaboration layer, collaborative commands are encoded as semantic vectors and broadcast to the cloud via WebSocket to enable synchronization across multiple devices. Feature vectors and semantic vectors are used as the principal data carriers across the four layers, thereby forming a closed-loop interaction flow encompassing perception, semantic interpretation, visual feedback, and collaboration. The overall architecture of the GPSM method and the data flow underlying bidirectional semantic mapping are illustrated in Figure 1.

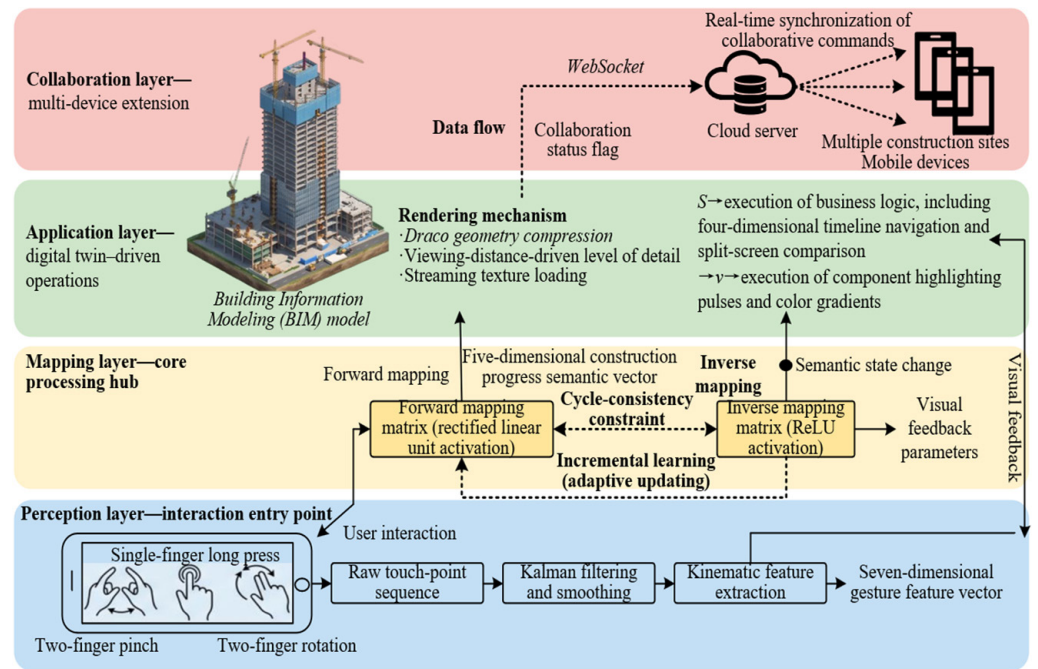


Fig. 1. Overall architecture of the GPSM method and data flow for bidirectional semantic mapping

2.2 Gesture feature extraction and construction progress semantic space modeling

The extraction of gesture kinematic features is a prerequisite for establishing the mapping relationship. For each frame of touch events, a seven-dimensional feature vector $h = [n, \bar{v}, \theta, \kappa, A, d_{\max}, \Delta t]^T$ is extracted, with each component characterizing a distinct kinematic property of the touch interaction. In the formulation, n denotes the number of valid touch points and is used to distinguish between single-finger and multi-finger operations. Different numbers of touch points are associated with different semantic intentions: single-finger gestures are generally associated with selection and querying, two-finger gestures with scaling and rotation, and three-finger gestures with view switching. $\bar{v}$ denotes the average movement of velocity and δ denotes the inter-frame interval. The movement velocity reflects the urgency of the user's operation and can be used as an auxiliary signal for semantic determination.

$$\bar{v} = \frac{1}{n} \sum_{i=1}^n \frac{\sqrt{(x_i^{(t)} - x_i^{(t-\delta)})^2 + (y_i^{(t)} - y_i^{(t-\delta)})^2}}{\delta} \quad (1)$$

where, θ denotes the principal movement direction angle. It is obtained by applying principal component analysis to the displacement vectors of the touch points and determining the direction of the first principal component. This feature is used to identify the direction and magnitude of rotational gestures. κ denotes the trajectory curvature and is calculated as follows:

$$\kappa = \frac{|\ddot{x}\ddot{y} - \dot{x}\dot{y}|}{(\dot{x}^2 + \dot{y}^2)^{3/2}} \quad (2)$$

where, $\dot{x}$, $\dot{y}$ and $\ddot{x}$, $\ddot{y}$ denote the velocity and acceleration of the touch point in the x - and y -directions, respectively. The curvature characterizes the degree of bending

of the gesture trajectory and is used to distinguish linear swipes from curved rotational gestures. A denotes the envelope area of multiple touch points. Its value is set to zero for a single touch point, whereas for multiple touch points, the convex hull area is calculated using the Shoelace formula. Together with the distance between touch points, this feature characterizes the spatial extent of the gesture. $d_{\max}$ denotes the maximum Euclidean distance between touch points.

$$d_{\max} = \max_{i,j} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (3)$$

where, Δt denotes the gesture duration and is used to distinguish rapid swipes from long-press gestures. For two-finger pinch and spread gestures, the distance $d(t)$ between the two touch points and its rate of change $\dot{d}(t)$ are further calculated. A spread gesture is identified when the rate is greater than 30 pixels/frame, whereas a pinch gesture is identified when the rate is less than -30 pixels/frame.

$$d(t) = \sqrt{(x_1(t) - x_2(t))^2 + (y_1(t) - y_2(t))^2} \quad (4)$$

$$\dot{d}(t) = \frac{d(t) - d(t - \delta)}{\delta} \quad (5)$$

To prevent unintended triggering caused by jitter during the initiation and termination phases of a gesture, the mean squared error between touch heatmaps I_t of consecutive frames is also calculated as follows:

$$MSE_t = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n [I_{t-1}(i, j) - I_t(i, j)]^2 \quad (6)$$

A gesture is confirmed to be in a stable, sustained phase only when $MSE_t < 0.01$.

The construction progress semantic space is used to encode state information from the construction management domain into a structured mathematical representation, thereby assigning explicit operational semantics to the outputs of touch interactions. A five-dimensional semantic vector $s = [c, \rho, \delta, \phi, \sigma]^T$ is defined, with each component corresponding to a core dimension of on-site construction progress control. c denotes the unique identifier of the building component currently in focus. A ray-picking algorithm in the three-dimensional scene is used to identify the building component corresponding to the user's touch position. If the gesture does not target a specific component, this component of the semantic vector is left empty, indicating that the current operation is applied to the global view. $\rho \in [0, 1]$ denotes the percentage of completed work. For an individual component, the actual completion percentage of that component is used. In the global overview state, a weighted average of the completion percentages of all visible components within the viewport is calculated. The weights are determined according to the planned quantities of the corresponding components, allowing the construction progress overview to reflect the actual distribution of work quantities. $\delta \in [-1, 1]$ denotes the construction progress deviation index and is calculated as follows:

$$\delta = (\rho_{\text{actual}} - \rho_{\text{planned}}) / \max(\rho_{\text{planned}}, \epsilon) \quad (7)$$

where, ρ_{actual} denotes the actual completion percentage, ρ_{planned} denotes the planned completion percentage, and $\epsilon = 10^{-6}$ is a small positive constant introduced to prevent division by zero. A positive value of δ indicates that construction progress is ahead of schedule, whereas a negative value indicates a delay. The absolute

magnitude of δ quantifies the severity of the deviation. ϕ denotes the timeline position, expressed as the number of days elapsed since the project start date, and is used to determine the construction stage currently being visualized. This representation enables construction progress to be reviewed retrospectively or explored prospectively along the temporal dimension. $\sigma \in \{0, 1, 2\}$ denotes the collaboration status flag, with its values corresponding to three modes: view only, issue marking, and collaboration initiation. The semantic vector is therefore used not only to represent query information but also to encode collaborative commands.

2.3 Bidirectional semantic mapping model

The forward mapping establishes a transformation mechanism from gesture features to construction progress semantics and is mathematically expressed as $s = f(W_s \cdot h + b_s)$, where W_s denotes the mapping weight matrix, b_s denotes the bias term, and $f()$ denotes the rectified linear unit activation function. In essence, this mapping projects continuously varying kinematic features in the seven-dimensional gesture feature space onto a hybrid discrete–continuous state representation in the five-dimensional construction progress semantic space. The mapping matrix is obtained through offline supervised learning using a training dataset $D = \{(h_i, s_i^{label})\}_{i=1}^N$ labeled by domain experts according to the operational logic of construction management. The training objective is defined as the minimization of the mean squared error between the predicted and labeled semantic vectors:

$$L_{forward} = \frac{1}{N} \sum_{i=1}^N \|s_i^{label} - f(W_s \cdot h_i + b_s)\|^2 \quad (8)$$

Optimization is performed using the Adam algorithm with a learning rate of 10^{-3} and a batch size of 32. Taking the two-finger spread gesture described above as an example, the feature vector h_{pinch} is transformed through mapping to obtain $s_{pinch} = [c_{focused}, \rho_{detail}, \delta_{current}, \phi_{current}, 0]^T$. The central building component within the enlarged region is thereby automatically identified, and the granularity of construction progress visualization is switched from a floor-level overview to component-level details of individual construction activities, while the current construction progress deviation state and timeline position are retained. A single-finger long press is mapped to $s_{longpress} = [c_{picked}, \rho_{detail}, \delta_{current}, \phi_{current}, 1]^T$. While the construction progress of the selected component is displayed, the collaboration status is simultaneously set to issue marking, allowing a single gesture to support both information querying and collaboration initiation.

The inverse mapping transforms changes in semantic states into visual interaction feedback parameters and is formulated as $h_{feedback} = g(W_h \cdot \Delta s + b_h)$, where $\Delta s = s_{current} - s_{previous}$ denotes the difference between the semantic vectors of the current and previous frames, W_h denotes the inverse mapping matrix, b_h denotes the bias term, and g denotes the sigmoid activation function, through which the output is constrained to the interval $[0, 1]$. The output of the inverse mapping is not used to execute specific gesture commands; instead, it is used to drive the visual feedback system of the digital twin model, including the amplitude and frequency of component highlighting pulses, the transition rate of construction progress colors, and the timing of prompt presentation. To ensure consistency between the forward and inverse mappings, a cycle-consistency constraint is introduced:

$$L_{cycle} = \frac{1}{N} \sum_{i=1}^N \|h_i - g(W_h \cdot (f(W_s \cdot h_i + b_s) - s_{ref})) + b_h\|^2 \quad (9)$$

where, s_{ref} denotes the semantic vector at the previous time step. This constraint ensures that, after a gesture is transformed through the forward mapping and subsequently reconstructed through the inverse mapping, the reconstructed representation remains close to the original input, thereby maintaining alignment between interaction feedback and user expectations. Based on the bidirectional mapping model described above, mapping rules for five core gesture types are defined for construction management scenarios. A two-finger spread gesture triggers component-level construction progress details, whereas a two-finger pinch gesture switches the visualization to a global overview. A single-finger long press displays a construction progress information card and enables collaboration initiation. A two-finger rotation gesture drives navigation along the timeline, enabling retrospective exploration of four-dimensional construction progress. A three-finger downward swipe activates a split-screen comparison between planned and actual construction progress. Importantly, these mappings are not limited to discrete gesture-to-command correspondences. Continuous changes in pinch/spread magnitude are mapped to ρ , enabling a smooth transition from overview to detailed visualization, while the magnitude of the rotation angle determines the step size by which ϕ moves forward or backward along the timeline. This continuous mapping mechanism fundamentally distinguishes the proposed interaction approach from existing approaches in which gestures are used solely to trigger discrete commands.

2.4 System deployment and adaptive mechanism

Deployment in real-world construction environments imposes two principal requirements on system performance: the limited computational resources of mobile devices must support real-time rendering of the digital twin model, and adequate generalization of the mapping model must be maintained across different users and projects. To address the former requirement, the original BIM model is subjected to a three-stage lightweight processing procedure. During the geometric compression stage, the industry foundation classes model is converted to the graphics language transmission format, and the positions, normals, and indices of mesh vertices are quantized and compressed using the Draco algorithm, reducing the file size to approximately 5–10% of its original size. Level-of-detail rendering is then employed to dynamically adjust model fidelity according to the camera viewing distance d . When $d > 50$ m, the polygon count is limited to 5×10^4 ; when $10 < d \leq 50$ m, it is limited to 2×10^5 ; and when $d \leq 10$ m, the upper limit is set to 1×10^6 . Octree-based spatial partitioning is further integrated with frustum culling, such that only visible objects within the viewing frustum are submitted to the rendering pipeline. Texture data are compressed using the adaptive scalable texture compression 6 × 6 format and streamed in blocks according to the viewport region. Texture blocks within the current field of view are loaded preferentially, while those in adjacent regions are loaded asynchronously as the viewpoint changes. Following these optimizations, the per-frame rendering time is maintained below 16 ms, thereby ensuring a stable interactive frame rate of 60 frames per second.

To enable personalized adaptation of the mapping model, an online incremental learning mechanism is introduced. When the mapping result of a gesture is explicitly corrected by the user through a menu or voice input, the current gesture feature vector and the user-specified target semantic vector are combined to form a new paired sample $(h_{new}, s_{new}^{label})$, which is then added to the online training buffer. Once 10 correction samples have been accumulated in the buffer, an incremental update

$W_s \leftarrow W_s - \eta \cdot \nabla_{W_s} L_{forward}^{new}$ is performed in the background, where $\eta = 10^{-4}$ denotes the online learning rate.

The empirical loss calculated exclusively from the newly added samples is defined as follows:

$$L_{forward}^{new} = \frac{1}{K} \sum_{j=1}^K \|s_j^{label} - f(W_s \cdot h_j + b_s)\|^2 \quad (10)$$

where, K denotes the number of newly accumulated samples in the buffer. Through this mechanism, the mapping matrix is progressively adapted to individual user interaction patterns and project-specific management requirements while its baseline mapping capability is retained.

3 PROTOTYPE SYSTEM IMPLEMENTATION

The prototype system is implemented using a browser-server architecture. On the mobile side, a three-dimensional rendering engine is developed using Web Graphics Library and Three.js, while the back end is implemented using Node.js. WebSocket is employed for real-time message delivery, and MongoDB is used to store model metadata and construction progress data. The system is divided into three components: the mobile client, cloud platform, and collaboration client. Touch capture, feature extraction, mapping inference, and model rendering are performed on the mobile client; model distribution, construction progress storage, and message routing are managed by the cloud platform; and view synchronization is achieved on the collaboration client by subscribing to state changes through WebSocket. During operation, touch gestures are transformed into semantic vectors by the on-device mapping model. These vectors are used to drive local visualization, while construction progress updates and collaborative commands are simultaneously broadcast through the cloud platform to other collaboration clients.

The core modules comprise a gesture recognition engine, a model processing pipeline, and a data synchronization module. In the gesture recognition engine, multi-touch sequences are captured through the native touch interface, and filtering and smoothing, feature extraction, and forward mapping inference are performed at the JavaScript layer. Model weights are loaded using TensorFlow.js, with inference latency maintained below 16 ms. In the model processing pipeline, Revit source files are used as input, exported to the industry foundation classes format, converted to the graphics language transmission format, and subsequently compressed using Draco. The compressed files are uploaded to a cloud-based content delivery network, and model blocks are dynamically requested by the mobile client according to the level-of-detail strategy. During initial loading, only the low-detail global model is retrieved, whereas high-detail local geometry is progressively loaded as interaction proceeds. In the construction progress data synchronization module, once construction progress reporting or issue marking has been completed by the user, the construction progress information and collaboration status encoded in the semantic vector are encapsulated into a data message and transmitted to the cloud platform via WebSocket. After the model state has been updated in the cloud, the state change is broadcast through a message queue to all online collaboration clients. Upon receipt of the incremental state update, only the affected regions are locally redrawn on each client, thereby avoiding the bandwidth and computational overhead associated with full-scene refreshing.

4 EXPERIMENTAL EVALUATION

To comprehensively evaluate the effectiveness of the GPSM method, five sets of experiments were designed and conducted, with performance assessed across five dimensions: recognition accuracy, task efficiency, cognitive load, learning curve, and on-site collaboration. The experiments were conducted at the construction site of a 30-story residential building under construction. All participants had experience in high-rise building construction management.

4.1 Gesture recognition accuracy and response latency

A total of 12 participants were recruited, with a mean age of 34.2 years and an average of 8.5 years of construction-site work experience. Under real-world construction-site conditions, the participants performed the five core gesture types using an iPhone 14 Pro and a Samsung Galaxy S23. Each gesture type was repeated 30 times, resulting in a total of 3,600 gesture operations. Recognition accuracy, false-trigger rate, and end-to-end response latency were recorded for each gesture type on both devices, as presented in Table 1.

Table 1. Comparison of gesture recognition performance across devices and gesture types

Gesture Type	Device	Recognition Accuracy (%)	False-Trigger Rate (%)	Mean Response Latency (ms)	Latency Standard Deviation (ms)
Two-finger spread	iPhone 14 Pro	97.8	1.7	72.4	8.3
	Galaxy S23	96.3	2.2	79.1	9.7
Two-finger pinch	iPhone 14 Pro	97.2	1.9	70.8	7.9
	Galaxy S23	95.8	2.5	77.6	9.2
Single-finger long press	iPhone 14 Pro	96.7	2.3	68.5	7.1
	Galaxy S23	95.1	2.9	74.2	8.8
Two-finger rotation	iPhone 14 Pro	95.9	2.8	79.6	9.4
	Galaxy S23	94.3	3.4	85.3	10.5
Three-finger downward swipe	iPhone 14 Pro	96.1	2.5	73.7	8.6
	Galaxy S23	94.8	3.1	80.4	9.9
Overall average	iPhone 14 Pro	96.7	2.2	73.0	8.3
	Galaxy S23	95.3	2.8	79.3	9.6

In terms of recognition accuracy, an overall accuracy of 96.7% was achieved on the iPhone 14 Pro, compared with 95.3% on the Galaxy S23, with both devices exceeding the predefined threshold of 95%. The two-finger rotation gesture exhibited lower recognition accuracy than the other gesture types on both devices, with an accuracy of only 94.3% on the Galaxy S23. This result may be attributed to the precise angular threshold determination required for rotation gesture recognition, which is more susceptible to differences in touch sampling rates across devices. Regarding false triggering, the overall false-trigger rate ranged from 2.2% to 2.8%, remaining below the predefined threshold of 3%. For response latency, mean values

of 73.0 ms and 79.3 ms were recorded for the iPhone 14 Pro and Galaxy S23, respectively, both below the expected threshold of 80 ms. Among the five gesture types, the shortest response latency was observed for the single-finger long press, whereas the longest was observed for the two-finger rotation gesture, which was directly associated with the length of temporal information required for gesture determination.

4.2 Comparison of task completion efficiency

A within-subject design was adopted, in which the 12 participants completed four typical construction progress management tasks using both the GPSM interaction mode and the conventional menu-based interaction mode. Each task was performed three times under each interaction mode, and the mean completion time and number of operational steps were calculated. The task order was counterbalanced using a Latin square design to minimize potential learning effects. The task completion efficiency metrics for the two interaction modes are summarized in Figure 2.

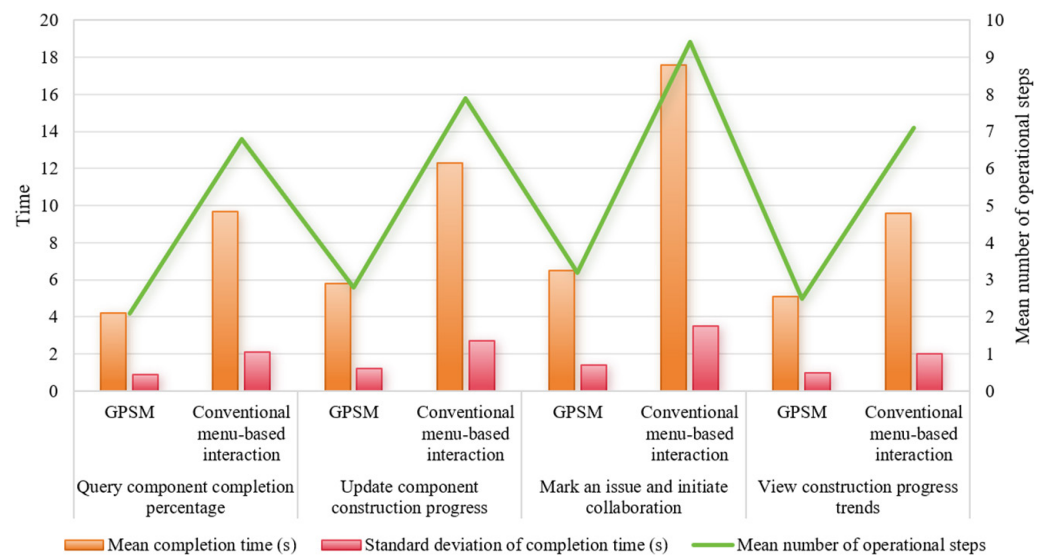


Fig. 2. Comparison of task completion efficiency between the two interaction modes

Note: GPSM denotes gesture-progress semantic mapping.

Significantly shorter completion times were achieved with the GPSM mode across all four task types than with the conventional menu-based mode. The overall mean completion time was reduced from 12.3 s to 5.4 s, corresponding to a reduction of 56.1%. The greatest improvement was observed for the task of marking an issue and initiating collaboration, for which completion time was reduced by 63.1%. This improvement can be attributed to the direct activation of the construction progress information card and collaboration interface through a single-finger long press in the GPSM mode, whereas multiple independent steps—including component selection, function switching, and collaborator selection—were required in the conventional menu-based mode. The overall number of operational steps was reduced by 65.4%, indicating that operations previously distributed across multiple menu levels were consolidated into continuous touch gestures through GPSM, thereby substantially shortening the interaction path. Paired t-tests indicated that the differences in completion time between the two interaction modes were statistically significant for all task types ($p < 0.01$).

4.3 Cognitive load evaluation

Immediately after completion of Experiment 2, the participants completed the NASA Task Load Index questionnaire, in which the two interaction modes were subjectively evaluated across six dimensions on a scale ranging from 0 to 100. The scores for each dimension and the weighted overall workload scores are reported in Table 2.

Table 2. Results of the NASA Task Load Index cognitive load evaluation

Evaluation Dimension	Gesture-Progress Semantic Mapping (GPSM) Mode	Conventional Mode	Reduction (%)	Paired <i>t</i> -Test (<i>p</i> -Value)
Mental demand	32.4 ± 8.7	51.7 ± 10.2	37.3	< 0.001
Physical demand	28.6 ± 7.5	35.8 ± 9.1	20.1	0.018
Temporal demand	33.1 ± 9.2	42.3 ± 11.4	21.7	0.009
Performance	25.8 ± 7.9	33.6 ± 10.5	23.2	0.014
Effort	34.6 ± 9.8	56.2 ± 12.7	38.4	< 0.001
Frustration	27.3 ± 10.1	50.4 ± 14.6	45.8	< 0.001
Weighted overall workload	30.7 ± 7.6	43.6 ± 10.3	29.6	< 0.001

Lower scores were obtained with the GPSM mode than with the conventional mode across all six dimensions. The weighted overall workload score was reduced from 43.6 to 30.7, corresponding to a reduction of 29.6%. Particularly pronounced reductions were observed for mental demand and effort, which decreased by 37.3% and 38.4%, respectively. These results indicate that, by directly encoding construction progress semantics into touch gestures, the GPSM mode substantially reduced the cognitive burden associated with memorizing menu structures and locating functions during interaction. The largest reduction, 45.8%, was observed for frustration, suggesting that fewer operational barriers and task interruptions were experienced during GPSM-based interaction. Paired *t*-tests showed statistically significant differences between the two interaction modes across all six dimensions ($p < 0.05$), with $p < 0.001$ for mental demand, effort, and frustration.

4.4 Learning curve analysis

To evaluate the feasibility of on-site deployment of the GPSM system, 24 participants with no prior experience using GPSM were recruited and randomly assigned to either the GPSM group or the control group, with 12 participants in each group. Participants in both groups used their respective systems to perform routine construction progress management tasks for five consecutive days. Mean task completion time and the number of operational errors were recorded daily. The learning curve data collected over the five-day period are presented in Table 3.

For the GPSM group, mean task completion time decreased from 11.8 s on Day 1 to 5.4 s on Day 5, representing an overall reduction of 54.2%. In comparison, the completion time for the control group decreased from 13.1 s to 11.8 s, corresponding to a reduction of only 9.9%. The GPSM group achieved a completion time of 6.2 s on Day 3, and no statistically significant difference was observed between the values

recorded on Day 3 and Day 5 ($p = 0.082$), indicating that a stable level of operational performance had been reached by the third day. In contrast, no significant performance inflection point was observed for the control group throughout the five-day period. Regarding operational errors, the mean number of errors in the GPSM group decreased from 4.2 on Day 1 to 1.0 on Day 5, representing a reduction of 76.2%. By comparison, the number of errors in the control group decreased only from 3.8 to 3.0, corresponding to a reduction of 21.1%. The steeper learning curve observed for the GPSM group indicates that its interaction logic can be learned intuitively and that proficiency can be achieved after a relatively short period of practice, providing empirical support for rapid on-site deployment.

Table 3. Comparison of task completion time and number of errors between the two groups over five consecutive days

Day	Gesture-Progress Semantic Mapping (GPSM) Group Completion Time (s)	Control Group Completion Time (s)	GPSM Group Errors	Control Group Errors	Time Standard Deviation of the GPSM Group (s)
Day 1	11.8 ± 2.6	13.1 ± 2.4	4.2 ± 1.1	3.8 ± 1.0	2.6
Day 2	8.3 ± 1.9	12.5 ± 2.2	2.8 ± 0.9	3.5 ± 0.9	1.9
Day 3	6.2 ± 1.4	12.2 ± 2.1	1.6 ± 0.6	3.3 ± 0.8	1.4
Day 4	5.7 ± 1.2	12.0 ± 2.0	1.2 ± 0.5	3.1 ± 0.8	1.2
Day 5	5.4 ± 1.1	11.8 ± 1.9	1.0 ± 0.4	3.0 ± 0.7	1.1

4.5 On-site collaboration efficiency evaluation

The GPSM prototype system was deployed in two structurally similar construction sections of an actual project. Eight management personnel in Section A, designated as the experimental group, used the GPSM system, whereas eight management personnel in Section B, designated as the control group, used conventional paper-based drawings in combination with a mobile menu-based system. Both groups were observed continuously for four weeks. The comparative results for the collaboration efficiency metrics are presented in Figures 3 and 4.

The average daily frequency of construction progress updates reached 18.6 updates/day in the experimental group, compared with 8.8 updates/day in the control group, representing an increase of 111.4%. This improvement may be attributed to the GPSM mode, in which construction progress reporting was simplified from a multistep menu-based procedure to a two-step process involving a long press on the target component followed by direct entry of its completion percentage. Consequently, the time cost and operational effort associated with construction progress data entry were substantially reduced. The collaboration response time was reduced from 6.1 h in the control group to 2.8 h in the experimental group, corresponding to a reduction of 54.1%. The construction progress deviation correction cycle was shortened from 4.9 days to 3.2 days, representing a reduction of 34.7%. The average daily number of issues marked was 4.2 in the experimental group, compared with 2.5 in the control group, corresponding to an increase of 68.0%. This result indicates that the more convenient issue-marking mechanism facilitated more timely recording of construction anomalies by on-site personnel. The time required for each construction progress update was reduced from 18.4 s to 8.6 s, which was broadly consistent with

the task completion time observed for component construction progress updates in Experiment 2. This consistency further demonstrates that the efficiency advantages of GPSM were maintained under real-world construction-site conditions.

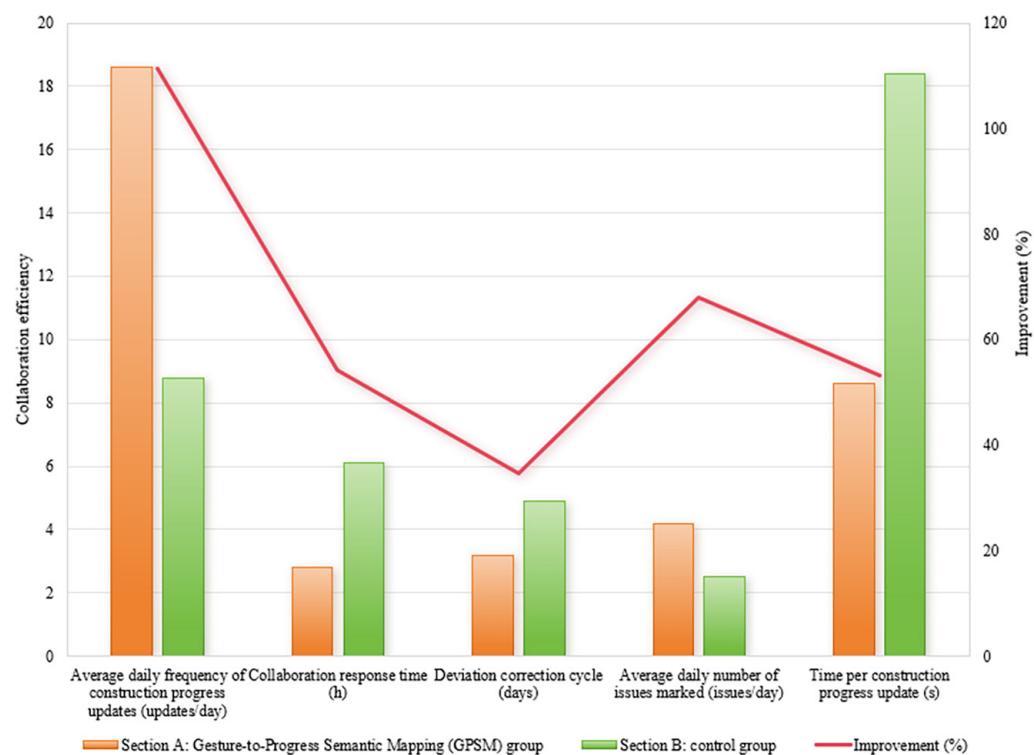


Fig. 3. Comparison of on-site collaboration efficiency metrics between the two construction sections

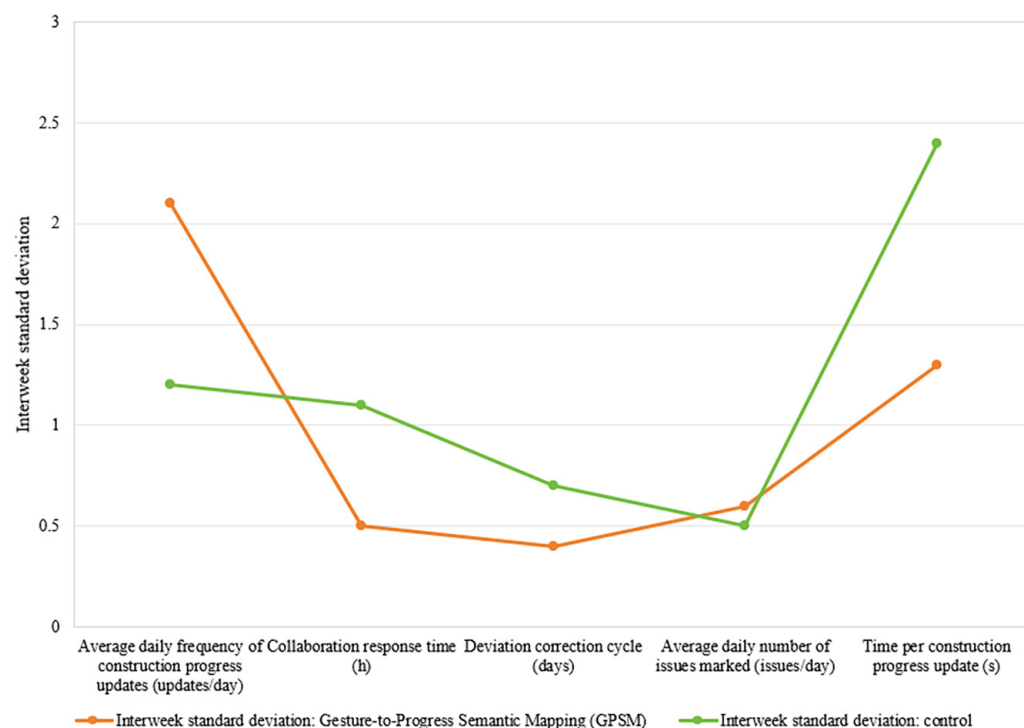


Fig. 4. Comparison of inter-week standard deviations between the two construction sections

5 CONCLUSION

To address the low efficiency of mobile interaction in construction progress visualization and on-site collaboration for high-rise buildings, a semantic mapping method from touch gestures to construction progress states was proposed. Through this method, touch interaction was extended from conventional viewpoint manipulation to semantic querying and collaborative command execution. At its core, a bidirectional mapping model between gesture kinematic features and construction progress semantics was established, enabling a single touch interaction to simultaneously support multiple functions, including viewpoint adjustment, information retrieval, and collaboration initiation. Based on this framework, methods for gesture feature extraction and construction progress semantic space modeling were systematically established, together with mathematical formulations and training strategies for the forward and inverse mapping functions. A set of representative mapping rules tailored to construction-site scenarios was developed, and an incremental learning mechanism was incorporated to enable personalized adaptation. These components collectively formed an integrated technical workflow spanning gesture sensing, semantic interpretation, and collaborative response. A web graphics library-based mobile prototype system was implemented, in which geometric compression, level-of-detail rendering, and streaming strategies were employed to address the rendering performance constraints imposed by high-rise BIM models on mobile devices. The experimental results consistently demonstrated significant advantages of the GPSM method in interaction efficiency, cognitive usability, and practical applicability under real-world construction-site conditions. The proposed approach provides a new paradigm for human-computer interaction design in mobile digital twin systems and offers a practical technical pathway for the digital management of construction sites.

6 REFERENCES

- [1] S. Su, R. Y. Zhong, Y. Jiang, J. Song, Y. Fu, and H. Cao, "Digital twin and its potential applications in construction industry: State-of-art review and a conceptual framework," *Advanced Engineering Informatics*, vol. 57, p. 102030, 2023. <https://doi.org/10.1016/j.aei.2023.102030>
- [2] Y. Pan and L. Zhang, "A BIM-data mining integrated digital twin framework for advanced project management," *Automation in Construction*, vol. 124, p. 103564, 2021. <https://doi.org/10.1016/j.autcon.2021.103564>
- [3] A. Sidani *et al.*, "Recent tools and techniques of BIM-based augmented reality: A systematic review," *Journal of Building Engineering*, vol. 42, p. 102500, 2021. <https://doi.org/10.1016/j.jobbe.2021.102500>
- [4] J. M. D. Delgado, L. Oyedele, P. Demian, and T. Beach, "A research agenda for augmented and virtual reality in architecture, engineering and construction," *Advanced Engineering Informatics*, vol. 45, p. 101122, 2020. <https://doi.org/10.1016/j.aei.2020.101122>
- [5] K. Amin, G. Mills, and D. Wilson, "Key functions in BIM-based AR platforms," *Automation in Construction*, vol. 150, p. 104816, 2023. <https://doi.org/10.1016/j.autcon.2023.104816>
- [6] J. Garbett, T. Hartley, and D. Heesom, "A multi-user collaborative BIM-AR system to support design and construction," *Automation in Construction*, vol. 122, p. 103487, 2021. <https://doi.org/10.1016/j.autcon.2020.103487>

- [7] D. J. Opoku, S. Perera, R. Osei-Kyei, and M. Rashidi, "Digital twin application in the construction industry: A literature review," *Journal of Building Engineering*, vol. 40, p. 102726, 2021. <https://doi.org/10.1016/j.jobe.2021.102726>
- [8] Y. Liao and H. Luo, "Real-time defect detection and carbon footprint visualization in green construction using mobile augmented reality and building information modeling," *International Journal of Interactive Mobile Technologies*, vol. 19, no. 9, pp. 92–106, 2025. <https://doi.org/10.3991/ijim.v19i09.55579>
- [9] M. N. M. Nawli, M. Yaakob, A. Q. Adeleke, M. N. Baharuddin, R. Omar, and R. Ghani, "Real-time prediction and prevention of worker accidents and safety hazards on construction sites using mobile machine learning framework," *International Journal of Interactive Mobile Technologies*, vol. 20, no. 5, pp. 144–157, 2026. <https://doi.org/10.3991/ijim.v20i05.60117>
- [10] F. Y. Ou, X. N. Zhu, Q. X. Li, and R. Yan, "Risk assessment of underground utility tunnel projects in Q City using the analytic hierarchy process," *Journal of Engineering Management and Systems Engineering*, vol. 4, no. 4, pp. 284–303, 2025. <https://doi.org/10.56578/jemse040405>
- [11] M. Ghanbari, D. Zolfaghari, and Z. Yadegari, "Mitigating construction delays in Iran: An empirical evaluation of building information modeling and integrated project delivery," *Journal of Engineering Management and Systems Engineering*, vol. 2, no. 3, pp. 170–179, 2023. <https://doi.org/10.56578/jemse020304>
- [12] S. Alizadehsalehi, A. Hadavi, and J. C. Huang, "From BIM to extended reality in AEC industry," *Automation in Construction*, vol. 116, p. 103254, 2020. <https://doi.org/10.1016/j.autcon.2020.103254>
- [13] S. Halder, K. Afsari, J. Serdakowski, S. DeVito, M. Ensafi, and W. Thabet, "Real-time and remote construction progress monitoring with a quadruped robot using augmented reality," *Buildings*, vol. 12, no. 11, p. 2027, 2022. <https://doi.org/10.3390/buildings12112027>
- [14] B. P. Van, P. Wong, and B. Abbasnejad, "A systematic review of criteria influencing the integration of BIM and immersive technology in building projects," *Journal of Information Technology in Construction*, vol. 30, no. 11, pp. 243–297, 2025. <https://doi.org/10.36680/j.itcon.2025.011>
- [15] F. Zhang, B. Xu, X. Zeng, and K. Ding, "Gesture-driven interaction service system for complex operations in digital twin manufacturing cells," *Journal of Engineering Design*, vol. 36, no. 11, pp. 2028–2054, 2024. <https://doi.org/10.1080/09544828.2024.2360852>
- [16] C. Xia, Q. Peng, Z. Fan, T. Min, D. Zhang, and C. Wang, "VCapTouch: Interactive touch sensing data synthesis for hand gesture recognition based on digital twin," *IEEE Internet of Things Journal*, vol. 12, no. 13, pp. 23823–23834, 2025. <https://doi.org/10.1109/JIOT.2025.3553561>

7 AUTHORS

Wenkai Jiang is affiliated with the Business School, Guangxi University of Finance and Economics, Nanning, China (E-mail: jiangwenkai@gxufe.edu.cn).

Yigang You is affiliated with the Business School, Guangxi University of Finance and Economics, Nanning, China (E-mail: YouYG@gxufe.edu.cn).