

Situated Perception: Spatial AI-Enabled Scene- and Object-level Localisation for Mobile Robotic Assembly

Begüm Saral¹, Hanzhi Chen², Stefan Leutenegger² and Kathrin Dörfler¹

¹Professorship of Digital Fabrication, Technical University of Munich, Germany

²Professorship of Machine Learning for Robotics, Technical University of Munich, Germany

begum.saral@tum.de

Abstract -

Mobile robots are a promising platform for large-scale on-site construction, but unlike fixed-base robots, they lack mechanical reference links between workspaces. On evolving construction sites, this creates a fundamental challenge: maintaining consistent alignment between digital planning models and physical execution across multiple robot positions and operational scales. This paper presents a Spatial-AI-enabled multi-level localisation and alignment framework for mobile robotic construction that continuously relates an evolving physical assembly to a digital twin. It combines scene-level visual-inertial SLAM (OKVIS) for global localisation and navigation with deep-learning-based RGB-D object pose estimation for object-level alignment and manipulation. Trained on synthetic data, the perception system detects individual building elements and estimates their poses in real time, enabling selected assembled elements to serve as spatial anchors for updating the digital twin and fabrication plans during execution. The approach is evaluated in a high-fidelity simulated case study of autonomous dry brick assembly. Experiments assess anchor perception error, scene-to-design re-alignment error, and pick-and-place accuracy. The results demonstrate that coupling global SLAM with object-level as-built anchors enables spatially consistent, adaptive robotic construction across large, changing worksites.

Keywords -

Mobile robotic assembly, Spatial AI, Scene- and object-level localisation, DL-based object perception, SLAM

1 Introduction

Robotics in construction extends beyond robot mobility; it requires continuous self-orientation, motion, observation, and precise manipulation within large, dynamic, and unstructured on-site environments. Unlike industrial settings with fixed layouts and calibrated workspaces, mobile construction robots operate in partially built, continuously changing environments with evolving maps and references. In this context, robots must continuously localise themselves, estimate the spatial state of the construction site and its materials, account for inaccuracies between as-planned and as-built states, and execute construction tasks with centimetre- to millimetre-level accuracy across large-scale workspaces defined by digital design models. The core challenge, therefore, extends beyond any single perception, localisation, or navigation capability and instead lies in how multiple levels of spatial understanding,

scene-level and object-level, can be structured, aligned, and coordinated within a unified method that supports mobile manipulation and task continuity in complex construction environments. In this paper, we therefore propose a Spatial-AI-enabled multi-level localisation and alignment framework that combines global SLAM with object-level as-built anchors to support spatially consistent and adaptive robotic assembly across a large worksite, integrating:

- A **multi-level spatial localisation method** that coordinates scene-level visual-inertial SLAM with object-level perception trained on synthetic data, enabling globally consistent navigation and manipulation-level accuracy.
- A **scene-to-design anchoring strategy** that maintains consistent alignment between the evolving building structure and its digital model through object-level anchors rather than solely scene-level references, enabling continuous re-alignment after robot relocation while preserving task continuity and geometric consistency of the constructed workpiece.
- A **perception-guided assembly approach** that enables flexible pick-and-place operations without reliance on pre-calibrated material stock locations.

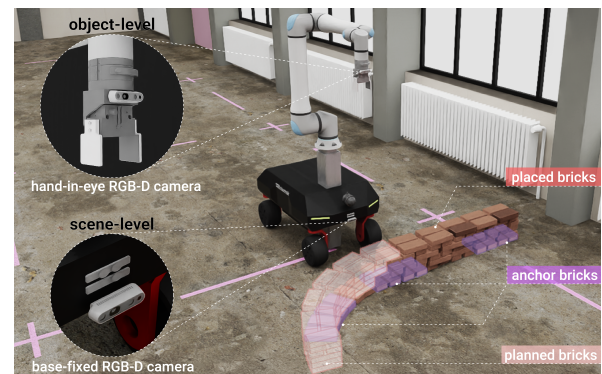


Figure 1. Experimental setup of the mobile robotic brick assembly system with multi-view RGB-D sensing for scene- and object-level localisation and anchor-based alignment.

The approach is evaluated in a high-fidelity simulated case study of autonomous dry brick assembly using an autonomous mobile robot, depicted in Figure 1. The use of simulation enables safe, controlled, and repeatable evaluation with access to ground truth while reflecting realistic sensing and robot behaviour, providing a foundation for future real-world deployment. The evaluation assesses object perception accuracy, alignment quality between the digital model and physical assembly, perception-guided pick-and-place performance, and multi-level localisation robustness enabled by the proposed object-based anchoring strategy.

2 State-of-the-art

Recent work in mobile robotics has demonstrated autonomous perception and localisation with minimal site setup and on-board sensing, indicating the feasibility of mobile operations in unstructured construction environments. However, these approaches differ in their sensing strategies and levels of spatial reference. For robot localisation within the scene, [1] utilises environmental markers, whereas systems such as [2, 3] rely on point-cloud- or edge-based scene-to-environment alignment strategies. In parallel, the mobile robotics community explores lightweight and flexible alternatives for scene-level localisation, namely visual-inertial and odometry-based SLAM methods [4], which provide tightly coupled visual-inertial odometry using cameras and IMUs. While effective for drift-reduced localisation without external infrastructure, these approaches operate primarily at the scene-level and remain only loosely coupled with object-level reasoning and manipulation during execution.

At the object level, recent research in the computer vision community has explored object-centric perception using camera-based methods. [5, 6, 7] combine object detection with 6-DoF pose estimation using geometric and photometric optimisation, while others directly regress object poses from RGB(-D) inputs using deep neural networks when CAD models are available [8, 9, 10, 11]. In construction contexts, [12] demonstrates adaptive interaction with irregular on-site elements using LiDAR-based perception, [13] explores edge-based detection for quality assessment, and [14] utilises RGB-D-based 6-DoF pose estimation for brick manipulation. However, these approaches primarily operate at the local object-level and are not tightly integrated into a persistent global spatial framework that maintains consistency across larger workspaces or multiple robot positions.

Overall, while robust scene-level and object-level localisation have been studied extensively, their coordinated use as a multi-level spatial alignment strategy, supporting object-based anchoring, continuous alignment, plan adaptation, and task continuity within a global reference frame,

remains an open challenge in mobile robotic construction.

3 System overview

3.1 System architecture

Our system integrates multiple sensors, multi-level localisation, planning and control modules, coordinated by a high-level task management system (Fig. 2). All modules communicate via ROS 2, providing a unified, time-synchronised data flow between sensing, reasoning, and execution.

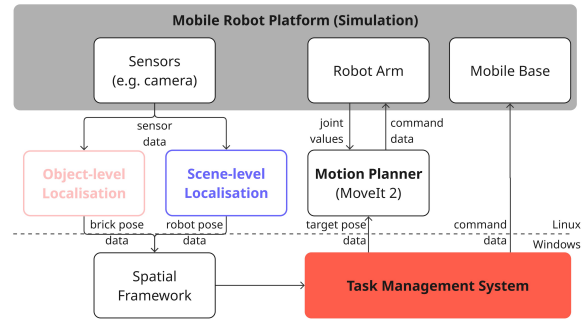


Figure 2. System architecture.

The mobile robot platform consists of a collaborative robot arm mounted on a mobile base. The arm is equipped with a parallel gripper and a hand-in-eye RGB-D camera for object-level localisation, while the base carries a fixed RGB-D camera for scene-level localisation. Sensor data from both cameras, together with the robot arm’s joint values, are continuously streamed via the simulation environment (NVIDIA Isaac Sim), replicating the robot, sensors, and physical site as a high-fidelity digital twin.

The multi-level localisation modules operate on the streamed sensor data on a high-GPU Linux workstation: scene-level localisation (OKVIS) estimates the robot pose over time (Section 4.1), while object-level localisation employs deep learning-based perception to estimate brick poses in real time (Section 4.2).

The high-level task management system, within the Rhino/Grasshopper architectural modelling environment, coordinates all modules to enable autonomous perception-guided assembly (Section 5). It integrates estimated pose data from the localisation modules into the spatial framework (Section 3.2) that stores planned and estimated brick poses and states within graph-based assembly models, along with the robot poses and transformations. Based on this information, the system determines the next bricks to build and sends their planned target poses to the motion planner (MoveIt 2) for execution by the robot arm, while commanding the mobile base directly for robot relocation.

3.2 Spatial framework and relationships

To integrate perception, localisation, and manipulation across large-scale scenes, we introduce a spatial framework that maintains multiple interconnected coordinate frames. It defines a coherent hierarchy linking scene-level (OKVIS) and object-level localisation to the architectural design world (Rhino) representation. Table 1 summarises the spatial framework definitions.

Table 1. Spatial framework definitions	
Frame	Definition
WCFrhino	Global coordinate system of design world.
BCFrhino	Robot base frame expressed in WCFrhino.
OCFrhino	Object (e.g. brick) frame expressed in WCFrhino.
WCFslam	Global coordinate system of OKVIS.
BCFslam	Robot base frame estimated by OKVIS, expressed in WCFslam.
esOCF	Object (e.g. brick) frame estimated by object perception, expressed in BCFslam.

By formalising the rigid transformations between spatial frames, all localisation results are consistently mapped into the design world *WCFrhino*, ensuring unambiguous visualisation, planning, and control across all task phases (Fig. 3).

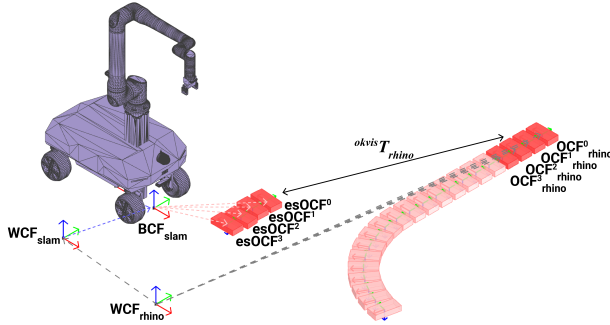


Figure 3. Spatial relationships between frames.

The overall chain of transformations is given by

$$esOCF \rightarrow BCFslam \rightarrow WCFslam \rightarrow WCFrhino. \quad (1)$$

At the object-level, the perception system estimates object (e.g. brick) poses relative to the robot base,

$$T(esOCF \rightarrow BCFslam), \quad (2)$$

while at the scene-level, OKVIS estimates the robot pose within its own OKVIS world frame,

$$T(BCFslam \rightarrow WCFslam). \quad (3)$$

On the design side, the planned object (e.g. brick) poses are fixed within the design world,

$$T(OCFrhino \rightarrow WCFrhino) = \text{fixed}. \quad (4)$$

To bridge the localisation and design domains, a world-to-world transformation situates the OKVIS world within the Rhino design world. This transformation is unknown at initialisation, computed during an explicit alignment phase, and remains fixed until the next realignment.

The alignment is performed by comparing estimated object frames *esOCF*, expressed in *WCFslam*, with their corresponding planned object frames *OCFrhino*, expressed in *WCFrhino*, yielding

$$T(WCFslam \rightarrow WCFrhino) = T_{okvis \rightarrow rhino}, \quad (5)$$

$$T_{okvis \rightarrow rhino} = T(esOCF \rightarrow OCFrhino). \quad (6)$$

The robot base pose, expressed in the design world *WCFrhino*, is then obtained by

$$BCF_{rhino} = T(WCFslam \rightarrow WCFrhino) \times BCF_{slam}. \quad (7)$$

By keeping *WCFslam* fixed between alignments and updating *BCFslam* after robot relocation, the system maintains a stable yet continuously valid geometric relationship between real-world sensing and the design model.

4 Multi-level localisation methods

4.1 Scene-level localisation (OKVIS)

Scene-level localisation provides a globally consistent estimate of the robot base pose for navigation, relocation, and maintaining spatial coherence across the workspace. It is implemented using OKVIS [15, 4], a visual-inertial SLAM system that estimates the robot base pose *BCFslam* in its internal world frame *WCFslam* from stereo RGB and IMU data acquired by a camera mounted on the mobile base (Fig. 4).

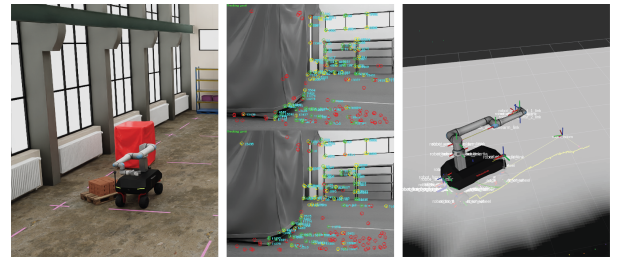


Figure 4. Mobile robot setup in Isaac Sim (left) publishing RGB and IMU data (middle) for the real-time robot pose localisation by OKVIS (right).

OKVIS jointly optimises camera poses, robot velocities, IMU biases, and sparse 3D landmarks within a factor graph, where visual reprojection errors and IMU preintegration constraints are tightly coupled and optimised in a sliding-window manner. To ensure real-time performance, older states are marginalised while their information is retained through linearised prior factors, enabling consistent

localisation over long navigation trajectories and repeated repositioning of the mobile platform.

However, as a visual-inertial SLAM system, OKVIS is subject to drift over time and distance. While sufficient for coarse localisation around the building structure, its accuracy does not meet the requirements for manipulation. It therefore serves as the scene-level backbone of the proposed multi-level localisation framework, complemented by object-level localisation and scene-to-design alignment to recover global consistency.

4.2 Object-level localisation (DL-based perception)

Object-level localisation provides high-precision pose estimates of individual building elements (e.g., bricks) for manipulation and scene-to-design alignment. For this, we employ a deep learning (DL)-based real-time detection and pose estimation pipeline for predefined geometries (Fig. 5), trained using a custom synthetic data generation process. The system estimates brick poses *esOCF*, initially expressed in the camera frame, and propagates them through the robot’s calibrated kinematic chain to the robot base frame *BCFslam*, where they are registered within the system’s spatial framework and serve as object-level references.



Figure 5. Mobile robot setup in Isaac Sim (left) publishing RGB-D image data for the real-time brick pose estimation (middle), and registration of the estimated poses in the modelling environment (right).

Given an input RGB-D frame $\{I, D\}$ captured by the hand-in-eye camera mounted on the robot arm, a deep neural network-based object detector first generates bounding box proposals for individual bricks. The detector follows the PointRend architecture [16] with a ResNet-101 backbone [17]. For all detected instances, the corresponding RGB-D crops are extracted and forwarded to a dedicated pose estimation network, inspired by [18], which recovers a set of 6-DoF object poses $\{T_i\}_{i=1}^{N_o}$, where N_o denotes the number of detected objects.

While the pipeline produces accurate per-frame pose estimates, temporal noise arises from prediction variability. To improve consistency, detected brick poses are associated with existing tracks and fused over time using an Extended Kalman Filter (EKF); no association initialises

a new track. The network-predicted pose serves as the measurement update, while the prediction step propagates the state forward in time. This filtering reduces high-frequency noise and suppresses outliers, yielding stable pose estimates suitable for manipulation and alignment.

4.3 Scene-to-design alignment via as-built objects

Scene-to-design alignment spatially anchors the OKVIS localisation world within the Rhino design world. During initialisation, it establishes a consistent mapping between scene-level localisation and the digital assembly model using object-level estimations, thereby fixing *WCFslam* in the design domain. After robot base relocation, the procedure is re-triggered to correct accumulated drift without resetting OKVIS, thereby preserving local consistency while restoring global accuracy.

To robustly compute $T_{okvis \rightarrow rhino}$, we employ a multi-algorithmic alignment strategy comparing estimated poses with their corresponding planned poses. First, an initial estimate is obtained using a Kabsch–Wahba formulation. Given N corresponding brick frames with centres;

$$\{\mathbf{p}_i^{\text{src}}\}_{i=1}^N \subset BCF_{\text{slam}}, \quad \{\mathbf{p}_i^{\text{tgt}}\}_{i=1}^N \subset WCF_{\text{rhino}}, \quad (8)$$

the optimal rotation, minimising the squared positional error, is given by

$$\mathbf{R}_{\text{pos}} = \arg \min_{\mathbf{R} \in SO(3)} \sum_{i=1}^N \left\| \mathbf{R}(\mathbf{p}_i^{\text{src}} - \bar{\mathbf{p}}^{\text{src}}) - (\mathbf{p}_i^{\text{tgt}} - \bar{\mathbf{p}}^{\text{tgt}}) \right\|^2, \quad (9)$$

solved via singular value decomposition of the cross-covariance matrix (Kabsch algorithm). Orientation ambiguities from symmetric object geometry are mitigated with a Wahba-style formulation over the brick frame axes, yielding a second rotation estimate $\mathbf{R}_{\text{orient}}$. The final rotation is obtained via spherical linear interpolation (SLERP), with $w \in [0, 1]$ balancing positional and orientational alignment, and the translation is computed as

$$\mathbf{R} = \text{SLERP}(\mathbf{R}_{\text{pos}}, \mathbf{R}_{\text{orient}}, w), \quad \mathbf{t} = \bar{\mathbf{p}}^{\text{tgt}} - \mathbf{R} \bar{\mathbf{p}}^{\text{src}}, \quad (10)$$

yielding the initial rigid transformation estimation

$$T_0 = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}. \quad (11)$$

The final transformation embedding the OKVIS world within the Rhino design world is computed using Iterative Closest Point (ICP), initialised with T_0

$$T_{okvis \rightarrow rhino} = \arg \min_{T \in SE(3)} \sum_j \left\| T \mathbf{x}_j^{\text{src}} - \mathbf{x}_{\pi(j)}^{\text{tgt}} \right\|^2, \quad (12)$$

where $\mathbf{x}_j^{\text{src}}$ and $\mathbf{x}_j^{\text{tgt}}$ are corner points sampled from estimated and planned bricks, and $\pi(j)$ denotes closest-point correspondences.

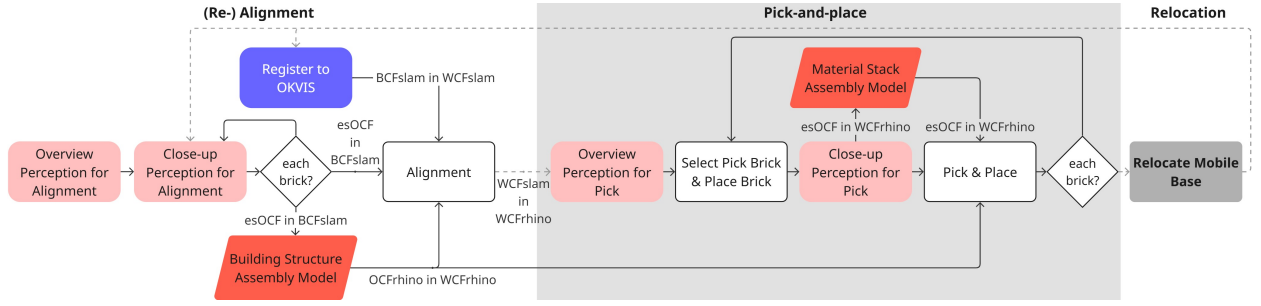


Figure 6. Task flow of the perception-guided assembly routines.

5 Perception-guided assembly

This section formulates the proposed perception-guided assembly routine as a closed-loop method for mobile robotic construction, in which localisation, scene-to-design alignment, model adaptation, motion planning, and robot control are continuously coordinated across successive assembly stages and robot relocations (Fig. 6).

5.1 Initial localisation and scene alignment

At the start of an assembly session, a robot model is initialised, and the building structure design is imported into the design world, where it remains fixed. The design is represented as a graph-based assembly model, referred to as the building structure assembly, in which a subset of bricks is designated as initial anchor bricks and is explicitly marked as such in the assembly. In the physical scene, these bricks are also placed with sufficient accuracy and serve as spatial references for scene-to-design alignment.

The task sequence begins with an overview perception phase, where the robot arm moves through a set of predefined snapshot poses, triggering the object localisation system to detect and register any visible bricks. This is followed by a diagonal close-up snapshot for each detected brick to improve pose estimation accuracy. Coupled with the robot base pose $BCFslam$ provided by OKVIS, the estimated and planned brick poses of the anchor bricks are passed to the alignment procedure to compute the $T_{okvis \rightarrow rhino}$ transformation. This aligns $WCFslam$ within the design world and completes the transformation chain for assembly execution.

5.2 Perception-guided pick-and-place

The pick-and-place routine begins once the alignment is established. The robot executes a set of predefined snapshot poses (to the side of the robot) to detect any visible brick within a material stack, which are registered into a separate assembly model. Then, a pick brick is selected from the topmost accessible layer of the stack, and a relative close-up snapshot is executed to refine the

brick's pose estimate. During this process, other visible bricks within the stack are also registered in the assembly model, ensuring a continuous pick-and-place loop.

A target placement brick is also selected from the planned building structure based on dependency checks, ensuring that all prerequisite bricks are already in place. To support a stable, stack-like construction strategy, the leftmost eligible brick is prioritised.

Based on the refined pick pose and the planned placement pose, motions for pick and place are planned and executed. In simulation, this process is performed under physically realistic conditions, including high-force gripper control, accurate masses, and frictional surfaces between the gripper and the bricks. The routine is repeated until reliable placement is no longer possible from the robot's current base position.

5.3 Robot relocation and scene realignment

The mobile base is autonomously relocated at a predefined distance and rotation by issuing joint-level motion commands. After relocation, the robot base pose $BCFslam$ is updated via OKVIS, enabling the robot base to be re-expressed in the design world as $BCFrhino$, allowing continued motion planning and execution relative to the building structure.

As part of the proposed realignment strategy, a new set of anchor bricks is manually placed along the structure at the ground level to serve as fresh spatial references. Given that their approximate positions in the design world are known, close-up perception is performed directly on the anchor bricks without requiring a global overview perception. The estimated brick poses are then used in the scene-to-design realignment procedure to compute an updated $T_{okvis \rightarrow rhino}$ and re-establish $WCFslam$. This ensures a new, consistent geometric relationship, allowing the pick-and-place process to resume reliably after relocation.

After relocation, an optional correction of neighbouring bricks is performed by estimating the pose of the last placed brick from the previous assembly stage and averaging it with its planned pose using SLERP. The brick is

then robotically adjusted to maintain geometric continuity across successive assembly stages.

6 Results

We conducted several sets of experiments designed to isolate and independently evaluate individual sources of pose deviation. Pose errors are computed as a relative transformation between the estimated poses stored in the graph-based assembly model and corresponding ground-truth poses extracted from the Isaac Sim runtime.

$$\mathbf{T}_{\text{err}} = \begin{bmatrix} \mathbf{R}_{\text{err}} & \mathbf{t}_{\text{err}} \\ \mathbf{0}^T & 1 \end{bmatrix}, e_{\text{trans}} = \|\mathbf{t}_{\text{err}}\|_2, e_{\text{yaw}} = |\text{atan2}(r_{21}, r_{11})|. \quad (13)$$

The yaw error corresponds to rotation about the vertical (Z) axis, assuming a Z-up coordinate system.

6.1 Initial localisation and scene alignment

We conducted repeated experiments to first evaluate the average pose deviation of the anchor bricks, computed by comparing estimated brick poses with ground-truth poses (Table 2). Using these estimates, the alignment procedure computes the robot base frame *BCFrhino* expressed in the design world, with the deviations reported in Table 3.

Table 2. Estimated pose error of anchor bricks after close-up perception snapshot.

Metric	Mean	Std	Max
Translation (m)	0.014	0.002	0.018
Yaw (deg)	2.586	1.267	4.208

Table 3. Robot base frame *BCFrhino* error after scene-to-design alignment.

Metric	Mean	Std	Max
Translation (m)	0.019	0.005	0.025
Yaw (deg)	0.641	0.314	0.955

This residual alignment error propagates to all subsequent planned object frames, effectively bounding the precision of downstream pick-and-place operations. Despite residual object-level orientation noise, the alignment constrains global rotational error, achieving sub-degree base-frame accuracy sufficient for assembly planning.

6.2 Perception-guided pick-and-place

The perception-guided pick-and-place routine was evaluated using a single scene-to-design alignment, which showed an initial base frame error of 0.022 m in translation and 0.96° in yaw. Object-level localisation to pick bricks from the material stack demonstrated high accuracy,

as summarised in Table 4, averaged over 30 attempts. The pick success rate was primarily determined by perception errors in the lateral y direction, as the gripper opening allowed a tolerance of approximately 1.1 cm on either side; errors exceeding this threshold led to grasp failure. Only a single failure was observed out of 30 attempts, and all remaining pick motions were successfully planned and executed (Fig. 7).

Table 4. Estimated pose error of pick bricks within material stack after close-up perception.

Metric	Mean	Std	Max
X error (m)	0.015	0.008	0.030
Y error (m)	0.005	0.004	0.017
Z error (m)	0.005	0.004	0.016
Translation (m)	0.017	0.007	0.030
Yaw (deg)	2.273	2.095	7.278

Placement accuracy results are reported in Table 5, defined by comparison to their planned poses. Deviations in final brick placement arise from a combination of residual scene alignment error, perception error of the picked bricks (particularly along the x direction), and minor effects introduced by simulated grasping and contact dynamics. Despite these factors, the results demonstrate reliable placement performance.

Table 5. Final placement error after perception-guided pick-and-place.

Metric	Mean	Std	Max
Translation (m)	0.023	0.007	0.035
Yaw (deg)	2.853	1.654	7.041

6.3 Robot relocation and scene realignment

The final and most substantial evaluation focuses on robot base relocation and the realignment strategy. To establish a baseline, we first evaluate SLAM-only localisation over different relocation distances using OKVIS without initial scene alignment (Table 6), and compare it to results with initial scene alignment (Table 7). The SLAM-only baseline shows that mean localisation accuracy degrades minimally with increasing travel distance, reflecting the bounded drift inherent to visual-inertial SLAM. However, when initial scene alignment is introduced, even residual *WCFslam* alignment error of 0.022 m in translation and 0.96° in yaw leads to noticeably increased *BCFrhino* deviations over larger distances.

To mitigate this effect, we evaluate the proposed realignment strategy by comparing robot base frame deviation errors after relocation between OKVIS with initial alignment only and with realignment. Realignment is performed by placing new anchor bricks at 1 m spacing and

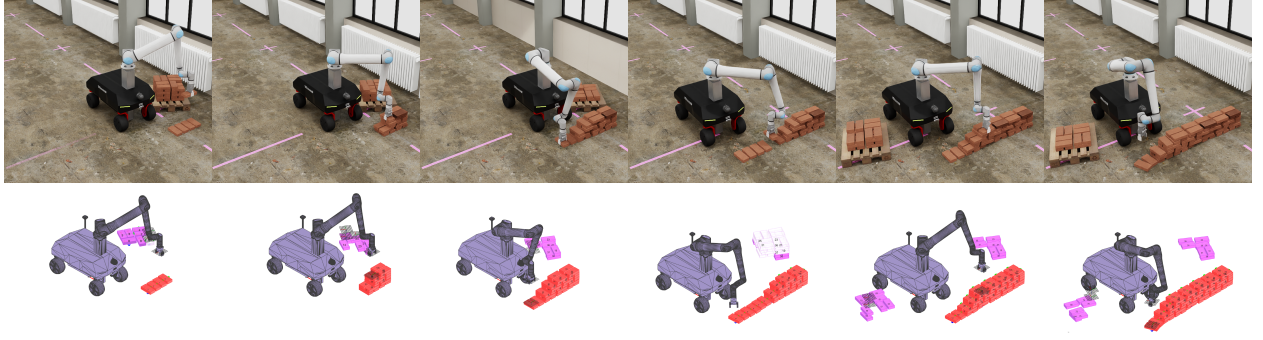


Figure 7. Perception-guided pick-and-place from two robot base positions, shown in simulation (top) and assembly models of the material stack and the growing structure in the modelling environment (bottom).

Table 6. Robot base frame *BCFrhino* mean error, OKVIS only (with gt initial state).

Metric	0.5 m	1.0 m	3.0 m	6.0 m
Translation (m)	0.023	0.026	0.027	0.092
Yaw (deg)	0.284	0.049	0.164	0.618

Table 7. Robot base frame *BCFrhino* mean error, OKVIS with initial alignment.

Metric	0.5 m	1.0 m	3.0 m	6.0 m
Translation (m)	0.028	0.039	0.082	0.188
Yaw (deg)	0.932	0.975	1.133	1.587

recomputing *WCFslam*. Following the initial pick-and-place experiment, the robot was relocated approximately 0.7 m, realigned, and executed a second batch of pick-and-place operations. It was then relocated along an arc of approximately 1 m radius, after which the realignment was repeated. The results show that the realigned *BCFrhino* exhibits significantly lower error than the baseline and preserves stronger local structural consistency immediately after relocation (Table 8).

Table 8. Robot base frame *BCFrhino* mean error after relocations, OKVIS with only initial alignment vs realignment.

Metric	1st Relocation		2nd Relocation	
	Al.	Real.	Al.	Real.
Translation (m)	0.032	0.012	0.220	0.016
Yaw (deg)	0.932	0.706	12.771	0.264

7 Discussion

The experimental results validate a multi-level localisation framework in which scene- and object-level localisation operate within a unified coordinate system without resetting one another, yet complement each other when needed. Scene-level SLAM serves as the global backbone,

providing continuous localisation across the workspace. The key contribution lies in the scene-to-design alignment strategy, in which selected elements of the building structure are manually placed with global precision, serving as stable anchors in the world coordinate frame. These anchors allow the robot to repeatedly realign object-level localisation to globally consistent references, preventing error accumulation from purely local, sequential alignment to the as-built structure. In this way, the robot can locally adapt execution to deviations in the as-built state while maintaining global geometric consistency and local flexibility, achieving manipulation-level precision. This is particularly important in mobile robotic construction, where repeated relocations and incremental assembly demand accurate registration not only to the local construction state but also to the overall building geometry. Furthermore, perception-guided assembly enables the robot to autonomously identify and retrieve elements directly from material stock, avoiding mechanical referencing or predefined fixture-based positioning.

8 Limitations and outlook

The current method is subject to several limitations. It was evaluated on dry-stack brickwork tasks, representing discrete placement operations, and does not capture construction processes involving material bonding or continuous interactions, such as 3D printing or plaster spraying. At the same time, this setup enabled a discrete, material-based anchoring strategy that can be further explored through sparse anchor placement and averaging to mitigate rotational errors. However, such a strategy still remains dependent on the availability and correctness of anchor materials: missing or misplaced components may reduce anchor reliability and necessitate more redundant anchoring mechanisms. Evaluation across a broader range of tasks is therefore required to assess generality, while achieving material-independent applicability will require

extending anchoring to additional fixed building elements (e.g., walls, beams, columns) by expanding object perception to more diverse geometries and semantic classes, supported by synthetic data generation and model training.

In addition to localisation deviations, placement errors were influenced by simulation artefacts such as brick sliding and oscillation. While simulation enables controlled evaluation with ground-truth access, it does not fully capture real-world conditions, and a sim-to-real gap remains due to modelled environmental factors. Therefore, real-world experiments are a necessary next step to validate the approach under physical constraints. More broadly, this work highlights the importance of integrated localisation frameworks: spatial consistency across scales, rather than isolated accuracy, is key to mobile robotic construction.

References

- [1] P. Ruttico, M. Pacini, and C. Beltracchi. Brix: an autonomous system for brick wall construction. *Construction Robotics*, 8, 2024.
- [2] T. Sandy, M. Giftthaler, K. Dörfler, M. Kohler, and J. Buchli. Autonomous repositioning and localization of an in situ fabricator. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2852–2858, 2016.
- [3] S. Ercan Jenny, H. Blum, A. Gawel, R. Siegwart, F. Gramazio, and M. Kohler. Online synchronization of building model for on-site mobile robotic construction. In *37th International Symposium on Automation and Robotics in Construction (ISARC)*, pages 1508–1514, 2020.
- [4] S. Leutenegger. Okvis2: Realtime scalable visual-inertial slam with loop closure. *arXiv preprint arXiv:2202.09199*, 2022.
- [5] M. Rünz and L. Agapito. Co-fusion: Real-time segmentation, tracking and fusion of multiple objects. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4471–4478, 2017.
- [6] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, A. Davison, and S. Leutenegger. Mid-fusion: Octree-based object-level multi-instance dynamic slam. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5231–5237, 2019.
- [7] M. Rünz, M. Buffier, and L. Agapito. Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 10–20, 2018.
- [8] W. Kehl, F. Manhardt, F. Tombari, S. Ilıc, and N. Navab. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In *IEEE International Conference on Computer Vision*, pages 1521–1529, 2017.
- [9] G. Wang, F. Manhardt, F. Tombari, and X. Ji. Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 16611–16621, 2021.
- [10] M. Rad and V. Lepetit. Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In *IEEE International Conference on Computer Vision*, pages 3828–3836, 2017.
- [11] H. Chen, F. Manhardt, N. Navab, and B. Busam. Tex-pose: Neural texture learning for self-supervised 6d object pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4841–4852, 2023.
- [12] R. L. Johns, M. Wermelinger, R. Mascaro, D. Jud, F. Gramazio, M. Kohler, M. Chli, and M. Hutter. Autonomous dry stone: On-site planning and assembly of stone walls with a robotic excavator. *Construction Robotics*, 4, 2020.
- [13] D. Apostolopoulos, H. Schempf, and J. West. Mobile robot for automatic installation of floor tiles. In *IEEE International Conference on Robotics and Automation*, volume 4, pages 3652–3657, 1996.
- [14] P. Ruttico and F. Bordoni. Bric-a-brick: Toward a reframing of clay block wall construction. *Preprints*, 6 2025. doi:10.20944/preprints202506.1657.v1.
- [15] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale. Keyframe-based visual-inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3):314–334, 2015.
- [16] A. Kirillov, Y. Wu, K. He, and R. Girshick. Pointrend: Image segmentation as rendering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9799–9808, 2020.
- [17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
- [18] Y. Hu, J. Hugonot, P. Fua, and M. Salzmann. Segmentation-driven 6d object pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3385–3394, 2019.