www.itcon.org - Journal of Information Technology in Construction - ISSN 1874-4753

AUTONOMOUS MIXED REALITY FRAMEWORK FOR REAL-TIME CONSTRUCTION INSPECTION

SUBMITTED: November 2024
REVISED: May 2025
PUBLISHED: May 2025
EDITOR: Robert Amor
DOI: 10.36680/j.itcon.2025.035

Tao Boan
Institute for Infrastructure and Environment, School of Engineering, University of Edinburgh, Edinburgh, Scotland, UK
email: boan.tao@ed.ac.uk

Li Jiajun
Institute for Infrastructure and Environment, School of Engineering, University of Edinburgh, Edinburgh, Scotland, UK
email: jiajun.li@ed.ac.uk

Frédéric Bosché
Institute for Infrastructure and Environment, School of Engineering, University of Edinburgh, Edinburgh, Scotland, UK
email: f.bosche@ed.ac.uk

SUMMARY: *The increasing complexity in construction projects necessitates advancements in the precision and efficiency of inspection processes. In response to this challenge, the present study explores the feasibility of a framework for autonomous inspection using Mixed Reality (MR), Building Information Modelling (BIM) and Artificial Intelligence (AI). The proposed framework encompasses techniques for: object detection in images taken through an MR headset; matching to the object instance in the digital twin; and visualisation of detection results in the MR headset to enable real-time human-in-the-loop decision making, thereby optimising the inspection workflow. The framework's efficacy is evaluated with two datasets representing diverse construction settings, including residential and office environments, focusing on the checking of the presence of ubiquitous elements like electrical sockets and switches. These tests illustrate the practical applicability and limitations of the proposed method.*

KEYWORDS: *Mixed Reality, BIM, Digital Twin, Construction Inspection, Progress Monitoring, Quality Control.*

REFERENCE: *Tao Boan, Li Jiajun & Frédéric Bosché (2025). Autonomous Mixed Reality Framework for Real-Time Construction Inspection. Journal of Information Technology in Construction (ITcon), Vol. 30, pg. 852-874, DOI: 10.36680/j.itcon.2025.035*

1. INTRODUCTION

In modern construction projects, ensuring that built or installed components comply with design specifications, safety standards, and functional requirements is critical. Historically, on-site inspections have relied heavily on paper-based drawings and the subjective expertise of inspectors, making the process time-consuming and prone to human error (Kim et al., 2015). Subtle deviations from the design can go unnoticed in the early stages, potentially leading to costly rework or compromised building performance.

To enhance efficiency and reliability, the construction industry has steadily embraced advanced digital approaches. Building Information Modeling (BIM) facilitates the generation of semantically rich and interconnected project data, enabling streamlined design and construction management. Meanwhile, Mixed Reality (MR) supports visualization and interactions by integrating virtual elements into real-world environments, thus enhancing the inspection process (Milgram et al., 1994). MR exists within the Reality-Virtuality continuum, and its application varies depending on the specific use case.

Additionally, recent breakthroughs in computer vision—a branch of artificial intelligence dedicated to enabling machines to interpret and understand visual data—have further propelled the digital transformation of construction. Tasks such as object detection, classification, and anomaly detection can be automated or significantly simplified, offering new opportunities for proactive quality control and issue identification (Chow et al., 2020; De Filippo et al., 2023).

Despite these advancements, many existing inspection approaches utilizing BIM, MR, or computer vision still require significant manual intervention, limiting their potential for holistic and continuous on-site inspections. Typically, inspectors must manually trigger scans or request data analysis, restricting efficiency and increasing the risk of human oversight. The development of fully integrated and autonomous solutions capable of continuously verifying on-site components without manual input could significantly enhance inspection performance.

This paper explores the feasibility of a system that integrates BIM data, MR visualization, and computer vision algorithms for autonomous, in-situ construction inspections. The envisioned system involves an inspector wearing MR glasses who can freely navigate the site while the system autonomously detects and verifies building components in the user's vicinity. It then highlights discrepancies or missing elements in real time, thereby providing immediate visual feedback and facilitating quicker, more informed decision-making. This approach focuses on three core inspection tasks: verifying the existence of components, validating positional accuracy, and ensuring compliance with design criteria.

Building on preliminary results reported by Tao et al. (2024), this study extends the approach by leveraging depth imagery for the simultaneous evaluation of multiple targets. Moreover, a comprehensive performance assessment is conducted in two distinct construction settings—residential and office—to demonstrate the robustness and practicality of the proposed solution.

The rest of this paper is organized as follows. Section 2 discusses relevant work in BIM, MR, and computer vision for construction inspection, elaborating on the specific gaps this paper aims to address. Section 3 details the design of the proposed system and explains how these three components are integrated to achieve proactive site inspection. Section 4 presents results of real-world experiments, and Section 5 offers a discussion of the findings, limitations, and future directions. Finally, Section 6 concludes the paper with a summary of key insights and contributions.

2. LITERATURE REVIEW

2.1 BIM-based Construction Inspection

BIM provides semantically rich and structured digital representations of facilities, encompassing architectural, structural, and MEP components. Its strengths include enabling clash detection (Ma et al., 2021; Chahrour et al., 2021), structural analysis (Tang et al., 2020), performance simulations (Asl et al., 2015), and other preventive quality assurance measures before or during construction. By offering a central, data-rich model, BIM facilitates comparison of on-site conditions with as-planned specifications. For instance, Chen et al. (2014) demonstrate how BIM-based workflows can detect geometric discrepancies early, thereby reducing the need for costly rework.

In addition, Digital Twins (DTs) build on BIM's foundations by dynamically coupling the as-planned model with real-time as-built data. DTs enhance visualization, enable predictive maintenance, and improve decision-making to reduce operational costs (Herrera et al., 2021; Megahed et al., 2022). Despite these potential benefits, the full integration of BIM and DTs in inspection workflows remains underutilized, in large part due to the continued reliance on manual data acquisition and interpretation. Many inspectors still rely on 2D drawings or navigate 3D models on handheld devices, rather than employing a fully integrated, hands-free verification process.

2.2 MR for Construction Inspection

MR merges virtual and physical environments in real time, allowing users to visualize digital models—such as BIM data—overlaid onto actual construction sites (Dunston & Wang, 2005). Wearable MR devices such as Microsoft HoloLens enable inspectors to reduce reliance on paper plans (Chalhoub & Ayer, 2018) and facilitate immediate cross-referencing of the design intent with built conditions. Nguyen et al. (2022) and Feng & Chen (2019) show that MR promotes faster detection of discrepancies, since as-planned and as-built elements can be directly compared in situ.

Beyond these direct comparisons, MR supports training and self-inspection. Workers can follow step-by-step instructions or confirm component installations by visually matching digital references with the physical environment (Riexinger et al., 2018). Moreover, MR can overlay real-time data—such as thermal or acoustic measurements—to identify performance deviations or defects (Riexinger et al., 2018; Holzwarth et al., 2021). This integration can significantly improve both the efficiency and accuracy of construction processes by providing instant feedback on potential problems.

Some researchers have focused on applying MR for bridge inspection and multisource data integration. Riedlinger et al. (2022) demonstrate that the combination of BIM and MR increases the precision of damage location and saves time in damage recording. El Ammari & Hammad (2019) advance this concept by merging multisource facility data, BIM models, and feature-based tracking to improve collaboration between field personnel and managers. Similarly, Jin et al. (2020) develop an MR-based system for bridge inspection and maintenance that superimposes relevant data on the bridge structure itself; inspectors can then view and assess maintenance needs or anomalies on site.

Despite these advances, MR technology in construction inspection often serves as an enhanced viewing platform rather than an autonomous verification system. Existing MR solutions generally require inspectors to select areas for inspection or initiate scans themselves. Other persistent challenges include hardware constraints, safety considerations, interoperability with various data formats, and the high cost of generating suitable virtual content (Dai et al., 2021; Prabhakaran et al., 2022). Addressing these obstacles is essential for fully harnessing MR's potential across construction workflows.

2.3 Computer Vision for Automated Inspection

Computer vision, particularly through machine learning and deep learning, has become a pivotal tool in construction inspection, enabling the automated analysis of visual and spatial data (e.g., images and point clouds). By detecting elements, classifying defects, monitoring progress, and verifying compliance with design plans, computer vision can reduce the reliance on manual checks. For example, De Filippo et al. (2023) combine UAV-acquired visual and thermal images with computer vision algorithms for automatic defect detection, while Chow et al. (2020) develop a vision-based pipeline that performs both anomaly detection and classification in concrete structures.

In some cases, researchers apply computer vision to interpret not just 2D images, but 3D data. Kim & Kim (2020) propose a deep learning approach that inspects and classifies point-cloud representations of bridge components, including abutments, piers, and girders, by comparing these segmented data against design models or historical scans to identify damage or deterioration. Khan et al. (2023) examine the use of deep learning-based computer vision to monitor construction safety compliance, highlighting how scaffolds can be checked for missing guardrails and how waste management can be tracked at regular intervals.

When integrated with MR, computer vision can further automate inspection workflows. Karaaslan et al. (2022) and Zakaria et al. (2023) describe AI systems that operate in real-time, continuously searching for defects in images captured by MR devices, specifically targeting concrete infrastructures. These systems integrate real-time machine

learning models for defect localization and quantification, displaying results dynamically within the MR interface. Through the MR interface, inspectors can fine-tune the defect detection process by adjusting confidence thresholds, verifying and correcting predictions, and modifying bounding boxes to ensure accurate and reliable results. Naticchia et al. (2019) use a YOLO network within an MR setup to identify and locate critical building assets such as fire extinguishers in real time, eliminating the need for manual labeling and data entry. These examples underscore how computer vision improves inspection efficiency, reduces human error, and enables faster resolution of identified issues.

Nevertheless, several challenges must be addressed before computer vision can be deployed in a comprehensive way in construction. Paneru & Jeelani (2021) point out that data privacy concerns, the need for high-quality and labeled datasets, and the requirement for model retraining or adaptation are among the most pressing issues. Technical and environmental factors, including fluctuating lighting and occlusions, can degrade detection accuracy, while the dynamic nature of construction sites poses additional hurdles for consistent tracking and alignment.

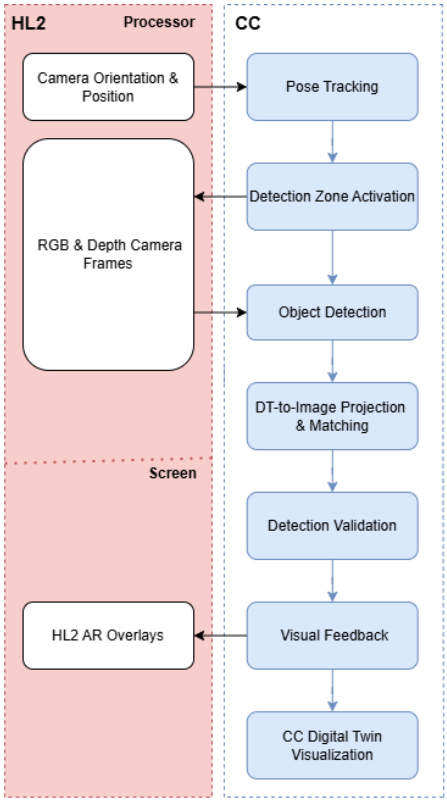

Figure 1: Real time workflow of the system.

2.4 Research Gap and Contribution

Despite the availability of digital tools—from semantically rich BIM models and digital twins to automated computer vision algorithms—construction inspections often still rely on user-driven interactions. Most MR-based inspection solutions act as enhanced viewers, with inspectors required to decide when to collect data, which can invite human error and allow small deviations to accumulate into costly issues. In particular, the inspection of Mechanical, Electrical, and Plumbing (MEP) installations is susceptible to errors due to the large quantity and dispersed nature of components. In such contexts, a solution is needed that integrates BIM data, MR visualization, and continuous computer vision analysis so that inspection can be achieved in an autonomous manner.

To address this research gap, we examine whether MR-based inspection might be made more autonomous by integrating BIM data with computer vision algorithms for real-time continuous identification and verification of MEP elements without the need for user-triggered data acquisition. By performing data acquisition and analysis in real-time without user involvement, the user could then simply navigate the site, and their attention is only

requested to make decisions and act on detected issues. Such a solution would support two main use cases: construction progress and quality control, and facilities management.

To do this, we build a prototype system and explore its capability at three levels. First, we explore the solution's capability to autonomously control whether each element of interest is actually installed, i.e., detecting any missing elements. Second, we explore the solution's capability to assess the positional accuracy of installed items by comparing their actual locations against the spatial arrangements defined in the BIM model. Collectively, these evaluations provide insights into how future systems might automate not only object detection—including the identification of missing elements—but also installation quality, thereby supporting robust and speedy inspections.

3. METHOD

3.1 Method overview

Our proposed system architecture encompasses two primary components: MR device, specifically chosen as the Hololens2 (HL2), and a Computation Centre (CC), which can be either a local computer or a cloud-based platform. The real-time workflow of our proposed framework is then depicted in Figure 1.

In operation, the user equipped with HL2 navigates the construction site. The HL2 (red rectangle) maintains real-time communication with the CC, continuously transmitting spatial data regarding the user's position and orientation. Upon receiving this spatial data, the CC initiates a series of processes. Key stages include:

1. **Real-Time Pose Tracking**. The CC conducts an analysis of real-time data pertaining to the orientation and position of the camera, as transmitted by the HL2. Initially, the CC detects and decodes a QR code, which serves to establish the initial pose (i.e. location and orientation) of the HL2. Following this, the CC continuously updates the camera's pose, from the spatial input stream provided by the HL2. This dynamic adjustment is detailed in Section 3.3.
2. **Detection Zone Activation**. The Central Controller (CC) evaluates whether the user's position intersects with spatially predefined detection zones associated with the objects of interest. The creation of those detection zones, designed to encompass one or more elements in the BIM model, is elaborated upon in Section 3.4. Upon verification of the user's presence within a detection zone, the CC dispatches an activation command to the HL2, initiating the real-time capture of video and depth frames. These frames, along with associated frame information (such as camera intrinsic parameters) are then transmitted to the CC for processing. When the user exits the detection zone, the frame capturing function is deactivated. Upon re-entry into a detection zone, the activation process of the video capturing function is re-initiated.
3. **Object Detection**. The CC loads the AI model and performs object detection on the received RGB camera frame, identifying any of the expected target objects within the detection zone. The outcome is the set of bounding boxes around detected objects. Detailed information on the detector is presented in Section 3.5.
4. **DigitalTwin-to-Image Projection and Object Matching**. This process involves projection of the known 3D positions of target BIM model objects within the detection zone onto 2D image coordinates. Subsequently, these 2D projections are matched with the bounding boxes obtained from the object detection step and the closest matches found. Details of this step are provided in Section 3.6.
5. **Detection Validation**. This step validates the detected objects by comparing their 3D projected positions with the known positions of the target objects in the digital twin. The validation criteria include checking the spatial distance between the detected and target objects' centroids, as well as considering the confidence scores provided by the detection model. This step is detailed in Section 3.7.
6. **Visual Feedback**. The detection results are visually indicated on the HL2 frame using colour-coded bounding boxes. Green bounding boxes denote valid detections that accurately match the known positions of objects, while red bounding boxes highlight detections that deviate from the expected positions. Additionally, new detections, which do not correspond to any known objects, are marked with yellow bounding boxes. Details of this step are provided in Section 3.8.

As indicated, the main steps of the above process are detailed in the following sub-sections: Section 3.2 to Section 3.8.

3.2 BIM-Based Digital Twin

The BIM-based Digital Twin is currently implemented as a Python-based program within the Computing Centre (CC). Its status is continuously updated with the status of "checked" components.

Since this digital twinning only includes a one-way dataflow from reality to virtuality, it may in fact be better described as a Digital Shadow (Sepasgozar, 2021).

3.3 Real-Time Pose Tracking

In the HoloLens 2 (HL2) system, image and video streams undergo distortion correction within the image-processing framework prior to being made accessible to applications. This correction ensures that the transmitted image frames adhere to a perfect pinhole camera model, free from distortion. Consequently, the system satisfies the perspective projection equation (Zhang, 1999):

$$p_i = K[\mathbf{R}|\mathbf{t}]P_i \tag{1}$$

where p_i represents the 2D coordinates of the image point, K is the intrinsic camera matrix, $[\mathbf{R}|\mathbf{t}]$ is the extrinsic matrix, and P_i denotes the 3D coordinates of the world point.

The intrinsic camera matrix K, which encapsulates the camera's focal length and the principal point offset, is computed in real-time by the HL2's autofocus system and communicated to the CC. The extrinsic matrix $[\mathbf{R}|\mathbf{t}]$ combines the rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$. This matrix describes the camera's pose relative to the world coordinates, capturing both its orientation (via $\mathbf{R}$) and its location (via $\mathbf{t}$). The extrinsic parameters are continuously updated in real-time to reflect changes in the camera's position and orientation as the user moves through the environment.

3.3.1 Initialisation

The camera's initial pose $\mathbf{E_0} = [\mathbf{R}_0|\mathbf{t}_0]$ can be established through various methods, such as QR code scanning (Kim et al, 2021), or visual analysis of identifiable structures or features (Sarlin et al, 2021). In this study, the initialisation is accomplished by scanning a QR code strategically affixed to a predetermined location (a wall in the experiments reported below).

The QR code is detected, and the coordinates of its corners are extracted, denoted as q_i in the image coordinates. The corresponding 3D coordinates in a local world coordinate system, designated as Q_i, are known from the pose of the matching twin QR code in the BIM model.

Using the 2D-3D point correspondences between q_i and Q_i, the camera's initial extrinsic parameters ($\mathbf{R_0}$ and $\mathbf{t_0}$) relative to the world coordinates are calculated. This computation is based on the principles outlined in equ:projection, which relates the 3D coordinates of points in the world to their 2D projections in the image, given the camera's intrinsic parameters.

3.3.2 Real-time updating

HL2 transmits real-time orientation ($\Delta\mathbf{R}$) and position ($\Delta\mathbf{T}$) changes relative to the initial pose. This data is used to update the camera's extrinsic matrix.

Rotation update: The new orientation matrix $\mathbf{R}_{\text{new}}$ is computed by multiplying the initial orientation $\mathbf{R}_0$ with the change in orientation $\Delta\mathbf{R}$:

$$\mathbf{R}_{\text{new}} = \mathbf{R}_0 \cdot \Delta\mathbf{R} \tag{2}$$

Location update: The new position vector $\mathbf{P}_{\text{new}}$ is updated by applying the change in position $\Delta\mathbf{T}$ relative to the initial orientation $\mathbf{R}_0$, and adding it to the initial position $\mathbf{P}_0$:

$$\mathbf{P}_{\text{new}} = \mathbf{R}_0 \cdot \Delta\mathbf{T} + \mathbf{P}_0 \tag{3}$$

Extrinsic matrix update: The extrinsic matrix $\mathbf{E}_{\text{new}}$ of the camera, which transforms points from the world coordinates to the camera coordinates, is updated using the new orientation and position:

$$\mathbf{E}_{\text{new}} = [\mathbf{R}_{\text{new}} \quad | \quad -\mathbf{R}_{\text{new}} \cdot \mathbf{P}_{\text{new}}] \tag{4}$$

3.4 Detection Zone Activation

3.4.1 Design of Detection Zone

The detection zones are created to reduce computational demand, by focusing on specific areas that need inspection or monitoring, within which the categories of the objects of interest are known. Each detection zone stores: the precise locations and categories of target elements and relevant geometrical data, such as targets' surface normal lines.

3.4.2 Camera activation and data acquisition

The system monitors the user's location within the environment from the spatial data from the HL2 device. The system compares these location coordinates with the boundaries of the predefined detection zones to identify which zone the user is currently in.

When the user enters a detection zone, CC sends a command to HL2 triggering camera activation and data acquisition. This involves capturing from various data streams, including the personal video (PV) stream and depth stream. The PV stream provides RGB images of the environment, and the depth stream supplies depth information from the environment. The system ensures that the latest data is used for processing by acquiring the most recent frames from these streams, often synchronising the frames to align the timestamps as closely as possible.

Following data acquisition, the captured frames undergo pre-processing to prepare them for further analysis. This pre-processing includes several tasks such as undistorting and normalising the depth data. Undistortion corrects any optical distortions in the depth images caused by the camera lens, ensuring geometric accuracy and consistency. Normalisation adjusts the depth values to a standard scale, making it easier to integrate depth information with other data types. Additionally, the PV frames are aligned with the depth data to create a coherent and integrated dataset. This alignment involves mapping the 2D RGB images from the PV stream onto the 3D depth information, providing a comprehensive view of the environment.

3.4.3 Data transmission

Zaccardi et al (2023) provide insights into using Unity's Barracuda on HoloLens 2 for real-time medical AR systems. They found that simpler models like Lenet5 can achieve over 30 fps. In contrast, more complex models like EfficientNetB0 result in a much lower frame rate, highlighting the difficult balance between model complexity and performance. Therefore, in theory, the computational capabilities of current MR hardware are sufficient to support the execution of deep learning models, including the projection of 3D objects. However, for more effective communication with digital twins and to assess the framework's performance more accurately, we perform both the detection and projection processes in CC.

Dibene and Dunn (2022) propose a HL2 server application to facilitate the real-time streaming of sensor data over TCP (Transmission Control Protocol). This protocol ensures reliable, ordered, and error-checked delivery of a stream of bytes between applications running on hosts communicating via an IP network. In this project, we implement a multiprocessing approach to efficiently direct the streams of front camera, depth camera and spatial input data towards a centralised computational hub. This approach facilitates the concurrent processing of diverse data inputs, enhancing the overall efficiency and throughput of the system.

3.5 Object Detection

In this study, the overall system is illustrated using the inspection of sockets and switches as an example. But, the method is naturally adaptable to other objects (e.g. fire safety equipment (Corneli et al, 2020)). To detect sockets and switches in images captured by the HL2 camera, a deep learning model is developed, based on YOLOv5m (Jocher et al, 2020), noted for its rapid and precise performance. The pre-trained YOLOv5m model is then retrained (with transfer learning) using a dataset comprising 2,026 indoor images featuring sockets and switches, enhanced through various augmentation techniques such as rotation, shearing, and mosaic effects to mimic lens distortion and complex indoor scenarios. The evaluation of the system involved the analysis of 73 images, incorporating 163 instances, and yielded a precision rate of 95% and a recall rate of 86.6%. The system has an inference time of 8.4 milliseconds, and a Non-Maximum Suppression (NMS) time of 2.5 milliseconds per image for an image dimension of (32, 3, 640, 640). This processing speed is particularly advantageous for real-time applications in construction inspection, highlighting the system's capability in both accuracy and efficiency in object detection tasks.

Algorithm 1 Digital Twin to Image plane Projection and Matching

Input:
P_{c_t} : 3D locations of centroid of target objects in the digital twin
User position and orientation
Intrinsic and extrinsic camera parameters
Detected object bounding boxes $bbox_i$
Thresholds: $Incidence_{\max}$, $Align_{\max}$, $Distance_{\max}$, $d_{\max}$ (half the image width)
Output:
Valid bounding boxes
Steps:
1. Calculate camera-to-object measurements
for each object in the detection zone **do**
$incidence$ = calculate_incidence_angle($object, user\_position, user\_orientation$)
$alignment$ = calculate_alignment_angle($object, user\_position, user\_orientation$)
$distance$ = calculate_distance($object, user\_position$)
if $incidence < Incidence_{\max}$ **and** $alignment < Align_{\max}$ **and** $distance < Distance_{\max}$ **then**
Add object to target objects list
end if
end for
2. Project 3D locations of listed target objects onto image plane
for each target P_{c_t} **do**
p_{c_t} = project(P_{c_t}, intrinsic_params, extrinsic_params)
end for
3. Match projected centroids with detected bounding boxes
$valid\_bboxes$ = []
for each projected centroid p_{c_t} **do**
$min\_distance = \infty$
$best\_bbox$ = None
for each $bbox_i$ in detected bounding boxes **do**
if category(p_{c_t}) == category($bbox_i$) **and** confidence($bbox_i$) ≥ 0.8 **then**
$distance_{2d}$ = euclidean_distance(centre($bbox_i$), p_{c_t})
if $distance_{2d} < min\_distance$ **then**
$min\_distance = distance$
$best\_bbox = bbox_i$
end if
end if
end for
if $min\_distance \leq d_{\max}$ **then**
$valid\_bboxes$.append($best\_bbox$)
end if
end for
4. Return valid bounding boxes
return $valid\_bboxes$

3.6 DigitalTwin-to-Image plane Projection and Matching

The system calculates the camera-to-object incidence angle, camera-to-object alignment angle, and camera-to-object distance for each object in the detection zone based on the user's position and orientation. Only if these measurements are less than the predefined thresholds $Incidence_{\max}$, $Align_{\max}$, and $Distance_{\max}$, respectively,

is the object considered a possible target object. These thresholds are necessary to ensure that the object is within an optimal viewing range, minimising errors in projections and improving the system's overall detection accuracy.

Using equ:projection, the system then projects the three-dimensional locations of each of these target objects in the digital twin, denoted as P_i, from the world coordinate system onto the HL2 video image plane, represented as p_i.

Subsequently, for each detection, the system matches the centroid p_{c_i} of each projected target object with the detected object bounding box $bbox_i$. For each p_{c_i}, a bounding box $bbox_i$ of the same category, with a high confidence score (0.8), is assigned based on the smallest Euclidean distance between the centre of $bbox_i$ and p_{c_i}. For each matching pair, if the Euclidean distance is within a predefined threshold $d_{\max}$ (half the image width), the detected bounding box is considered a valid detection.

Considering the estimation errors inherent in the pinhole model, as well as inaccuracies in the external and intrinsic matrices caused by sensor errors, a threshold $d_{\max}$ of half the image width has been found to be appropriate based on empirical evidence. These valid boxes $bbox_{valid_i}$ are then used for further processing. The entire process is depicted in Algorithm 1.

3.7 Detection Validation

3.7.1 Image to Digital Twin Projection

The centres of all bounding boxes $bbox_i$ from the video frames are projected into 3D space. This transformation entails several critical computations. Initially, the centre coordinates (u, v) of bbox_i are converted into normalised camera coordinates (x', y', z') via the intrinsic matrix **K**. This transformation is expressed by the equation:

$$\mathbf{K}^{-1}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

where **K** represents the intrinsic matrix, and (u, v) denote the image coordinates. Subsequently, these normalised coordinates are scaled using the depth values d obtained from the depth frames, as follows:

$$\mathbf{C} = d\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

where d is the depth value.

Finally, these scaled coordinates **C** are transformed into world coordinates (X, Y, Z) utilising the extrinsic matrix **E**, described by the equation:

$$\mathbf{E}^{-1}\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

where **E** encompasses both rotation and translation parameters.

In Section 3.5, the procedure involves projecting target elements onto the image frame and aligning them with the corresponding bounding boxes. Subsequently, these bounding boxes are projected back onto the digital twin to observe deviations within the 3D environment. However, bounding boxes that lack correspondence with any pre-identified elements suggest the presence of either newly detected entities or superfluous data (detection errors). Should these bounding boxes represent elements that are absent from the as-planned digital twin, their inclusion is nevertheless crucial for visualisation within the digital twin framework. These elements provide essential data for inspection to analyse, thereby enriching the accuracy and functionality of the digital twin model. Consequently, these two scenarios will be addressed separately in the subsequent section.

3.7.2 Detection Validation of Matching Bounding Boxes

For those bounding boxes $bbox_{valid_i}$ that correspond to target elements, after transferring the image frame to the world coordinate system, the system compares the centroid of the projected 3D coordinates (X, Y, Z) of bbox_i with

p_{c_i} to ascertain the deviation of the detection by computing the Euclidean distance between them. Given that the frame rate is set at 30 fps, image frames captured over 1 second (yielding 30 distance values) are collected, and the mean distance d_{mean} is calculated as the reliable deviation.

The mean distance d_{mean} is subsequently compared with the threshold distance $d_{\text{threshold}}$ to make a final determination regarding whether the detected position corresponds to the intended design position. If d_{mean} is greater than $d_{\text{threshold}}$, the object is detected but does not correspond to the as-designed plan. Conversely, if d_{mean} is less than or equal to $d_{\text{threshold}}$, the object is detected and corresponds to the as-designed plan. Upon making this determination, the object in the digital twin is marked as "checked". If the object is already marked as "checked," it will not undergo the detection validation process again. This process is summarised in Algorithm 2.

Algorithm 2 Detection Validation of Matching Bounding Boxes

Input:
- P_{c_i} : 3D locations of centroids of target objects in the digital twin
- $bbox_{valid_i}$: Valid bounding boxes from video frames
- Camera Intrinsic matrix **K**
- Camera Extrinsic matrix **E**
- Distance threshold $d_{\text{threshold}}$

Output:
- Marked objects in digital twin as "checked"

Procedure:

```
/* Step 1: Project centres of bounding boxes into 3D space */
for each bbox_valid_i do
    (u, v) = centre(bbox_valid_i)
    (x', y', z') = K^-1 × (u, v, 1)
    (X, Y, Z) = E^-1 × (x', y', z')
end for
/* Step 2: Check if the object is already marked as "checked" */
for each object in target_objects do
    if object is marked as "checked" then
        /* Skip to the next object if already checked */
        continue
    else
        /* Step 3: Calculate Euclidean distance between projected 3D coordinates and
centroids */
        for each bbox_valid_i do
            distances_3d = [ ]
            for 30 frames do
                distance_3d = Euclidean_distance((X_c, Y_c, Z_c), P_c_i)
                Append distance_3d to distances_3d
            end for
        end for
        /* Step 4: Calculate the mean distance */
        d_mean = mean(distances_3d)
        /* Step 5: Compare d_mean with the threshold to validate correspondence */
        if d_mean > d_threshold then
            Mark object as detected but not corresponding to the given target object
        else
            Mark object as detected and corresponding to the given target object
            Mark the target object in the digital twin as "checked"
        end if
    end if
end for
```

3.7.3 Detection Validation of Non-Matching Bounding Boxes

In scenarios where bounding boxes derived from 2D object detections do not find corresponding matches within the as-planned digital twin, these bounding boxes are hypothesised to represent either a new object or a false positive. To assess this, the centroids of the unmatched bounding boxes are projected into the 3D space using the camera's intrinsic and extrinsic parameters, alongside depth data. Then the system tracks the projected points over successive video frames to validate the hypothesis. For each subsequent frame, the bounding box detection algorithm attempts to detect the same object by projecting the centroids of newly detected bounding boxes into 3D space. If the projected 3D point of a bounding box in a new frame lies within a **0.1m tolerance radius** of a previously hypothesised 3D position, the system regards this as a re-detection of the same object.

Algorithm 3 Detection Validation of Non-Matching Bounding Boxes

Input:
  $bbox_{valid_i}$: Bounding boxes from video frames
  Camera Intrinsic matrix $\mathbf{K}$
  Camera Extrinsic matrix $\mathbf{E}$
  Depth map $depth$
  Centroid dictionary $centroid\_dict$ (key: 3D position, value: [count, inactivity counter])
  Set of removed keys $removed\_set$
  Threshold: $d_{\text{tolerance}}$, $d_{\text{detections}}$, $inactive\_threshold$
Output: Positions of newly detected objects $new\_objects$
Procedure:
for each $bbox$ in $bbox_{valid_i}$ **do**
  Compute centroid (u, v) from $bbox$
  Retrieve $centroid\_depth$ from $depth(u, v)$
  if $centroid\_depth$ is valid **then**
    $projected\_3D\_point$ = `compute_3D_coordinates`$(\mathbf{K}, \mathbf{E}, (u, v), centroid\_depth)$
    Add $projected\_3D\_point$ to $detected\_centroids$ list
  end if
end for
if $centroid\_dict$ is empty **then**
  for each $point$ in $detected\_centroids$ **do**
    if $point \notin removed\_set$ **then**
      $centroid\_dict[point] \leftarrow [1, 0]$ ▷ Count = 1, Inactivity Counter = 0
    end if
  end for
end if
for each $detected\_point$ in $detected\_centroids$ **do**
  $matched \leftarrow False$
  for each key in $centroid\_dict.keys()$ **do**
    if $|key - detected\_point| \leq d_{\text{tolerance}}$ **then**
      $centroid\_dict[key][0] \leftarrow centroid\_dict[key][0] + 1$ ▷ Increment count
      $centroid\_dict[key][1] \leftarrow 0$ ▷ Reset inactivity counter
      $matched \leftarrow True$
      if $centroid\_dict[key][0] \geq d_{\text{detections}}$ **then**
        Mark key as newly verified and update digital twin
      end if
      Break ▷ Stop further checks for this point once matched
    end if
  end for
  if not $matched$ and $detected\_point \notin removed\_set$ **then**
    $centroid\_dict[detected\_point] \leftarrow [1, 0]$ ▷ New detection, add to dictionary if not previously removed
  end if
end for
for each key in $centroid\_dict.keys()$ **do**
  $centroid\_dict[key][1] \leftarrow centroid\_dict[key][1] + 1$ ▷ Increment inactivity counter
  if $centroid\_dict[key][1] \geq inactive\_threshold$ **then**
    $removed\_set \leftarrow removed\_set \cup \{key\}$ ▷ Add to removed set
    Remove key from $centroid\_dict$ ▷ Remove inactive point
  end if
end for

Each successful re-detection increments a detection count for the hypothesised object. If the accumulated count for a given 3D position reaches a **threshold of 30 successful re-detections** over time, the object is confirmed as a valid, newly detected entity. The threshold is designed to ensure robustness, minimising the likelihood of falsely confirming objects that might be generated due to noise or transient detection errors. Upon confirmation, the object is formally incorporated into the digital twin representation of the environment and is visually presented to the user.

In instances where a projected 3D point fails to meet the detection threshold within a specified **inactive threshold**, the system treats the projection as a false positive, likely caused by noise or transient errors in detection. To prevent such erroneous data from accumulating within the system, an **adaptive removal mechanism** is employed. This mechanism assigns an inactivity counter to each hypothesised 3D point. The inactivity counter increments with each frame in which the point is not re-detected. If the inactivity counter for a given point exceeds a predefined **inactivity threshold**, the system removes the point from the tracking process and adds it to a set of discarded points, preventing its reintroduction.

The full implementation of this process is detailed in Algorithm 3.

3.8 Visual Feedback

Visual feedback is provided in two ways: within the HL2 image frame and in the digital twin representation.

3.8.1 Visual Feedback in HL2 Image Frame

In the HL2 image frame, bounding boxes are drawn around each detected object within the video frame. These bounding boxes are colour-coded according as follows:

• **Green**: This category is assigned when a bounding box corresponds to a targeted element ($bbox_{valid_i}$) and the deviation of its projection into the 3D environment is within the specified threshold distance ($d_{\text{threshold}}$). This indicates that the detected object's position and characteristics are consistent with those of a corresponding target element in the digital twin.

• **Red**: This category is applied when a bounding box corresponds to a targeted element ($bbox_{valid_i}$) but the deviation from the 3D projection exceeds the threshold distance ($d_{\text{threshold}}$). This discrepancy between the detected object's position and the pre-defined object location highlights potential errors in detection or inconsistencies in the data.

• **Yellow**: This category is used for bounding boxes that do not correspond to any targeted elements ($bbox_{n_i}$) but are detected within the same location range ($d_{\text{tolerance}}$) across multiple frames. These are considered newly identified objects, requiring further validation to confirm their existence and relevance.

4. EXPERIMENTAL RESULT

4.1 Performance Analysis of Pose Initialisation

Using scanning QR codes for determining camera position and orientation is a cost-effective and accessible method. However, this approach has its limitations. The accuracy can be significantly affected by factors such as poor lighting, low camera resolution, and environmental interference. To enhance the accuracy of the initialisation of the camera's position and orientation, the QR code is continuously scanned for a duration of 5 seconds while the user remains stationary. The mean values of the position and orientation collected during this period are then calculated, so that transient errors caused by sudden changes in the environment or by the initial positioning of the camera are averaged out.

In our experiment, a comparative evaluation is conducted between the computed camera position derived from the pin hole model and the position obtained through manual measurements. This comparison revealed that the average position deviation in this initialisation step is approximately 3.49. This discrepancy can be attributed to two significant factors. Firstly, lens distortion, particularly in the form of radial and tangential distortions, can alter the perceived geometry of the scanned QR code, leading to inaccuracies in the calculation of the camera's position and orientation. Secondly, due to the user's breathing, subtle body movements occur that generate slight but impactful shift in the camera's position.

4.2 Parameter Optimisation on Real-time projection

In Section 3.5 and 3.6, the projection steps involve transforming coordinates between the image frame and the 3D world system, and vice versa. The accuracy of this projection process is influenced by several factors, including: [noitemsep]

- Camera-to-object incidence angle θ_i: the angle between the camera's optical axis and the normal to the object's surface.
- Camera-to-object alignment angle θ_a: the angle between the camera's optical axis and the line connecting the camera to the object.
- Camera-to-object distance d_{co}: the direct line distance between the camera and the object.

These factors illustrated in Figure 2.

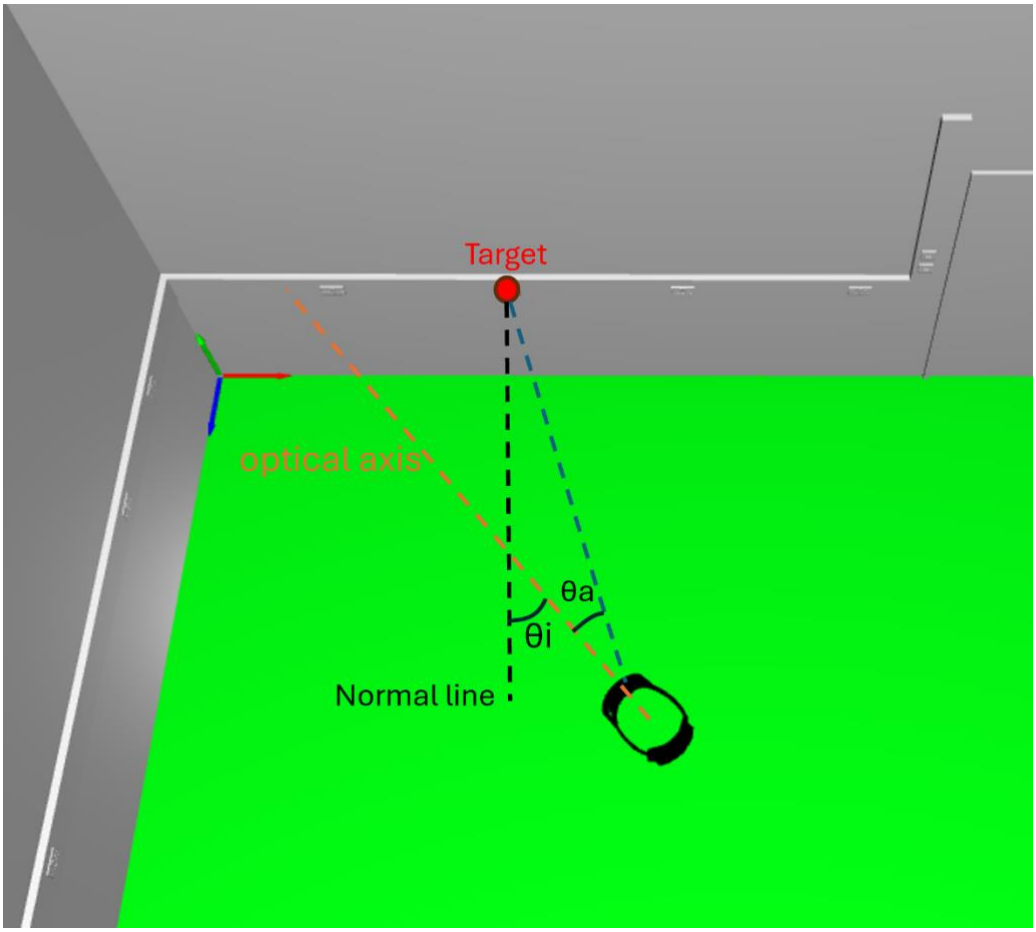

Figure 2: Visulisation of parameters. (The black item represents HL2, and the red dot indicates the target socket in a digital twin environment).

To explore the correlation between these factors and projection errors, we conducted an experimental study using a single socket target. The experiment is initialised by scanning the QR code and then detection and projection are performed at varying angles and distances with a socket located right next to the QR code (to reduce the impact of localisation drift).

We define deviation as the spatial distance calculated from the centre point of the 'as-designed' socket to the centroid of the 3D projected bounding box. In total 2,488 data points are acquired for analysis. In the analysis, the controlled variable method is utilised to ensure rigour and accuracy in the interpretation of the data.

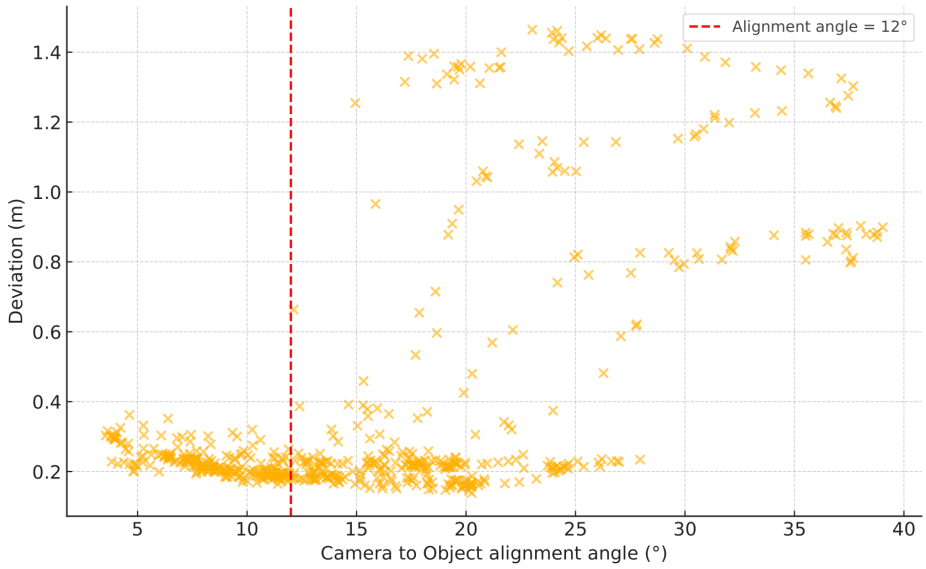

Figure 3: Relationship between the camera-to-object alignment angle θ_a and 3D projection deviation.

Initially, we fix the θ_i less than 10, given that the majority of the data falls within this range. This angle is also chosen due to its minimal distortion impact on the projection, ensuring it did not significantly affect the analysis of other parameters. Subsequently, we set the d_{co} less than 1.5. This moderate distance is selected to avoid any undue influence on the results. These constraints on θ_i and d_{co} results in a data subset (607 data) that is analysed to evaluate any relationship between the θ_a and the observed deviations. The results are summarised in the 3D scatter plot shown in Figure 3. We can see that once θ_a is larger than 12, the deviations become unstable and peak at higher values. This can be attributed to two possible reasons: (1) the image distortion becomes more pronounced at larger angles; and (2) the IMU sensor measurement inside HL2 is not accurate and stable, accumulating errors over time.

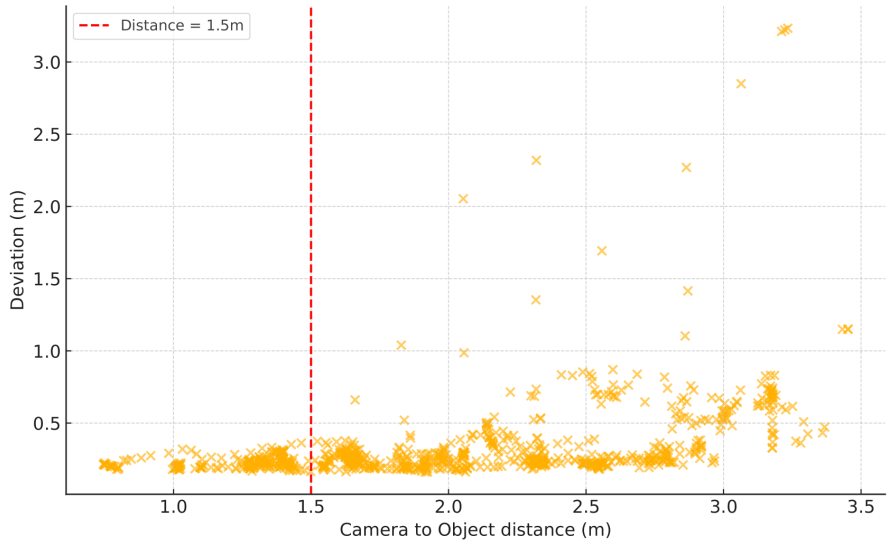

Figure 4: Relationship between the camera-to-object distance d_{co} and 3D projection deviation.

Subsequently, we set θ_i less than 10 and θ_a less than 12 to construct another data subset comprising 901 data points and that is utilised to examine the relationship between d_{co} and the 3D projection deviation. The results are reported in Figure 4. Our findings indicate that within a range of less than 1.5, the deviation remains consistently low and stable (< 0.4). Beyond that, the deviation increases significantly and becomes more erratic. This phenomenon can be attributed primarily to two factors: (1) the amplification of errors in preceding stages, such as sensor measurement or QR code initialisation, due to longer distances; and (2) the inherent limitations of the camera's capabilities adversely affecting detection at extended ranges.

Setting θ_a less than 0-12 and d_{co} less than 1.5 results in minimal and stable deviation, as evidenced by prior findings. Using these settings, a third data subset comprising 414 data points is created, that is used to investigate the relationship between camera-to-object incidence angle θ_i and the 3D projection deviation. The results are reported in Figure 5. The analysis reveals an increase in deviation corresponding to an increase in θ_i, particularly when the angle exceeds 10. This trend is attributed to factors similar to those affecting θ_a, such as image distortion at larger angles and sensor measurement inconsistencies.

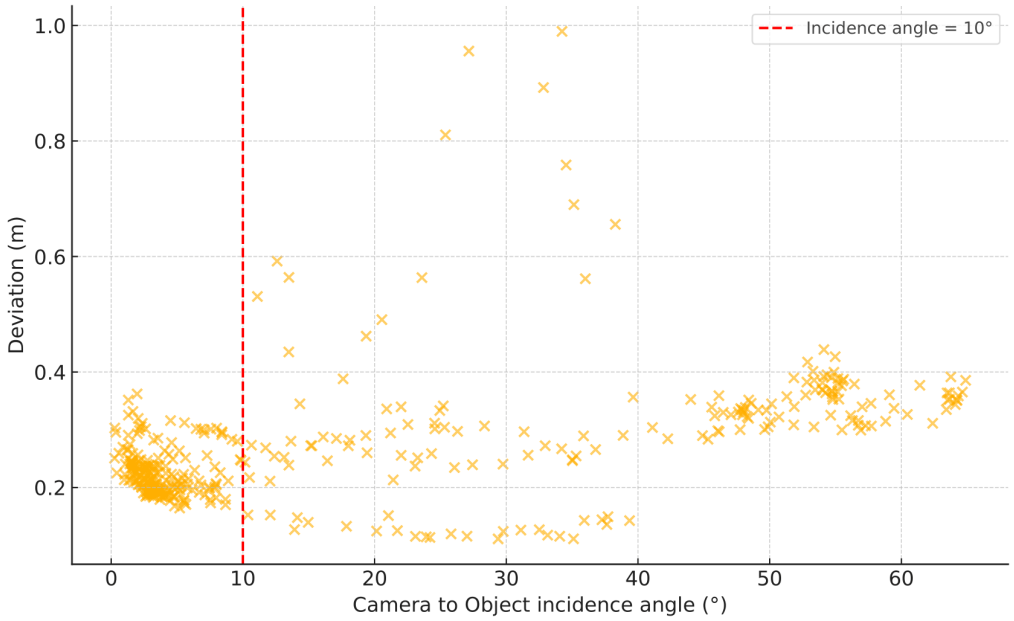

Figure 5: Relationship between the camera-to-object incidence angle θ_i and 3D projection deviation.

4.3 Result Analysis and visualisation

Based on the results presented in Section 4.2, the optimal parameter for detection with the lowest error is found to be when θ_a is less than 12, d_{co} is less than 1.5, and θ_i is less than 10. Under these conditions (which will be discussed further in Section 5), an experiment was conducted on-site to detect elements in two different scenes. The videos documenting these experimental tests are accessible online: the first experiment, conducted in a residential living room, can be viewed at [https://youtu.be/WAE5rRlHDfk], while the second experiment, conducted in an office environment, is divided into two parts: Part 1, focusing on optimised seated inspection, is available at [https://youtu.be/GjM4ImMmjME], and Part 2, which addresses natural standing inspection, can be found at [https://youtu.be/vINaBHCHjmk].

4.4 First Experiment in a Residential Living Room

The first scene involves a large living space (see in Figure 6). A 1:1 BIM model was created to serve as a Digital Twin of the physical space. The user's movements are updated in real-time within CC. In Figure 6a, the Digital Twin of the room is shown in grey, while four differently coloured squares indicate the designated detection zones.

The current pose of the HL2 device, updated in real-time, is shown in black.The HL2 screen interface, illustrated in Figure 6c, displays the detection bounding box to the user during the inspection process. They are shown in green indicating that those two detections correspond to two matching objects in the Digital Twin. This result is also shown in the Digital Twin in Figure 6a, where two green spheres can be seen on the wall adjacent to the blue detection zone, indicating the two detected target objects.

Figure 6a: Screenshot of the Digital Twin.

Figure 6b: User wearing the HL2 in operation.

Figure 6c: HL2 video frame.

Figure 6: First Experiment in a Residential Living Room.

To evaluate the detection accuracy, the threshold distance ($d_{\text{threshold}}$) was set at 1. Errors from each detection were recorded for analysis, with the results reported in Figure 10. The results show a mean deviation was 0.37, with a standard deviation of 0.156.

The sockets (Socket1 to Socket6) were scanned sequentially, following the order depicted in Figure 7, with traversal distances increasing progressively. The proximity of the first two sockets, located around 2 from the initial scanning position, resulted in relatively low deviations, with an average error of approximately 0.185. In contrast, a significant increase in deviation was observed beyond Socket4, where traversal distances ranged between 10 and 15. This deviation can be attributed to the lower positioning of Sockets 4, 5, and 6, which necessitated adjustments in the user's stance to maintain a camera-to-object incidence angle (θ_i) below 10. These positional adjustments introduced additional sensor drift in the HoloLens 2 (HL2) system, adversely affecting detection accuracy.

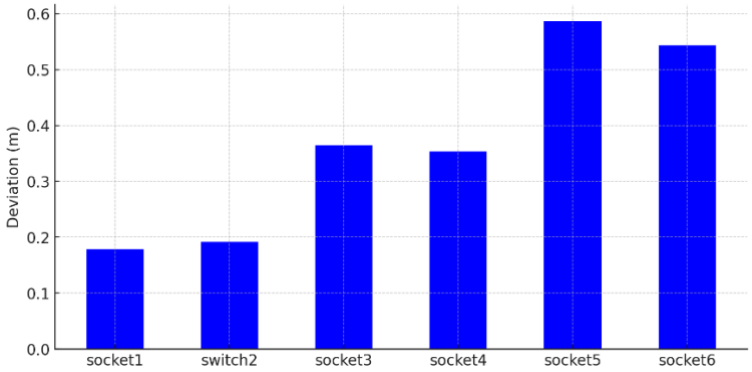

Figure 7: Assessment of Detection Deviations Across Elements in Experiment 1.

4.5 Second Experiment in an Office Environment

The second experiment is conducted in an office environment (see Figure 8a). The configuration comprises two office rooms and one corridor. In this experiment, each room designated as a detection zone: Zone A and Zone B. Initially, the user's position is in Zone A. Zone A contains ten objects (Switches 1-2 and Sockets 3-10) located within a relatively compact area. The total distance required to detect objects in this zone ranges from approximately 1 to 4.8. After completing the detection in Zone A, the user transitions to Zone B via the corridor, which contains the remaining ten objects (Switches 11-12 and Sockets 13-20). Zone B covers a larger area, with the user needing to travel between 10.6 meters and 15.3 meters to detect the objects in this zone.

Given that all the elements within this environment are positioned on the lower part of the walls (as indicated in Figure 8b), the user conducted the experiment on a moving seat to ensure that the angle θ_i remained below 10.

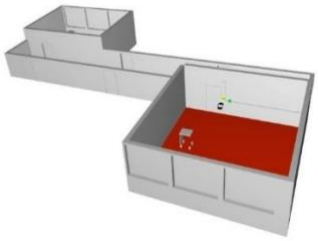

Figure 8a: Screenshot of the Digital Twin.

Figure 8b: User wearing the HL2 in operation.

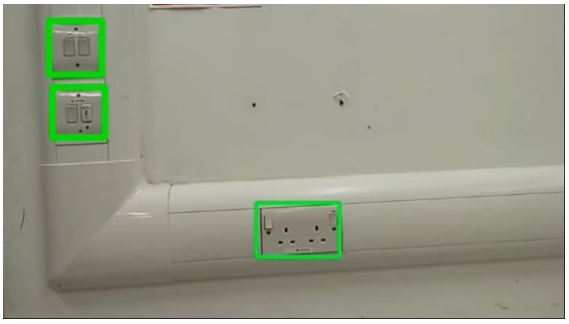

Figure 8c: HL2 video frame.

Figure 8: Second Experiment in an Office Environment.

The detections and deviations for a total of 20 elements were recorded. The results are presented in Figure 9. The first 10 elements, marked in sky blue, are located in Zone A, while those in dark blue are in Zone B. The mean deviation was 0.161, with a standard deviation of 0.073. Notably, all deviations are stable and maintained below 0.3, which was likely helped by the lack of vertical motion of the user on the moving seat, which minimised errors from the HL2 sensor.

The detection process was repeated under the condition that the user stands and walks normally, disregarding the requirement that camera-to-object incidence angle (θ_i) be less than 10. This adjustment was made to assess the significance of the incidence angle in the detection process. The deviation measurements were updated for the same 20 elements, as shown in Figure 10. The first 10 elements, marked in light blue, correspond to Zone A, while the remaining 10 elements, shown in dark blue, belong to Zone B. Unlike the first set of results, the deviations (Figure 11) show greater variance across both zones, with several elements exceeding the initial 0.3 threshold. The mean deviation increased significantly to 0.515, with a standard deviation of 0.216. The elements in Zone B exhibit notably higher deviations. This suggests that the observed errors can be attributed to the accumulation of errors resulting from user movement, as well as variations in head orientation as the user adjusts their gaze to inspect different elements.

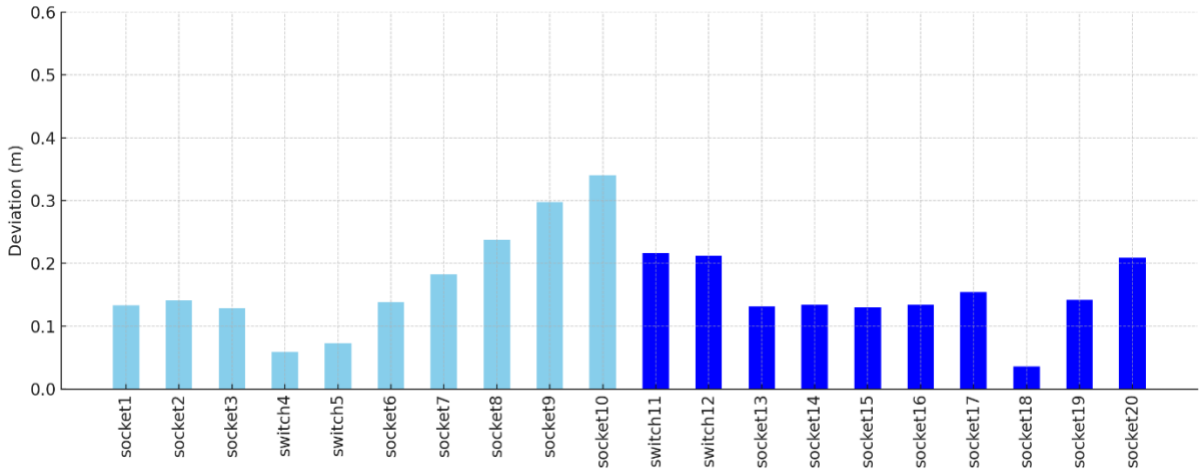

Figure 9: Assessment of Detection Deviations Across Elements in Experiment 2.

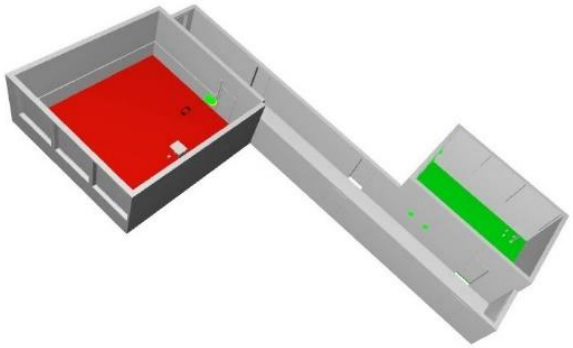

Figure 10a: Screenshot of the Digital Twin.

Figure 10b: User wearing the HL2 in operation.

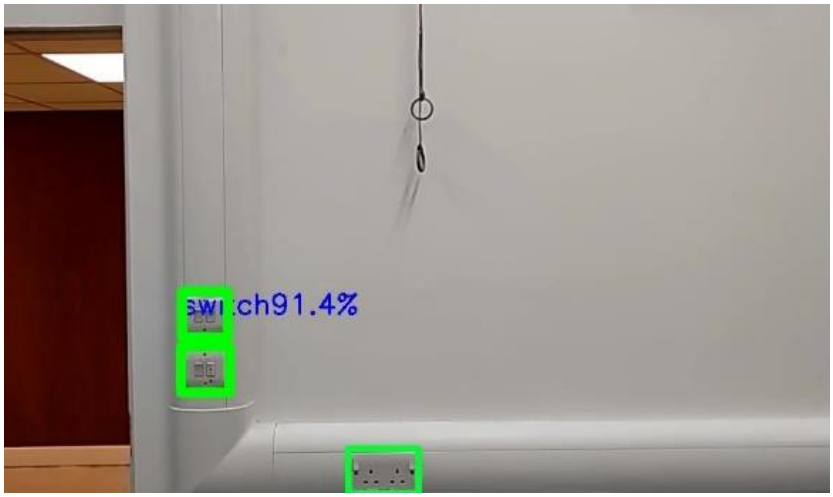

Figure 10c: HL2 video frame.

Figure 10: Third Experiment in an Office Environment with the user in a standing position.

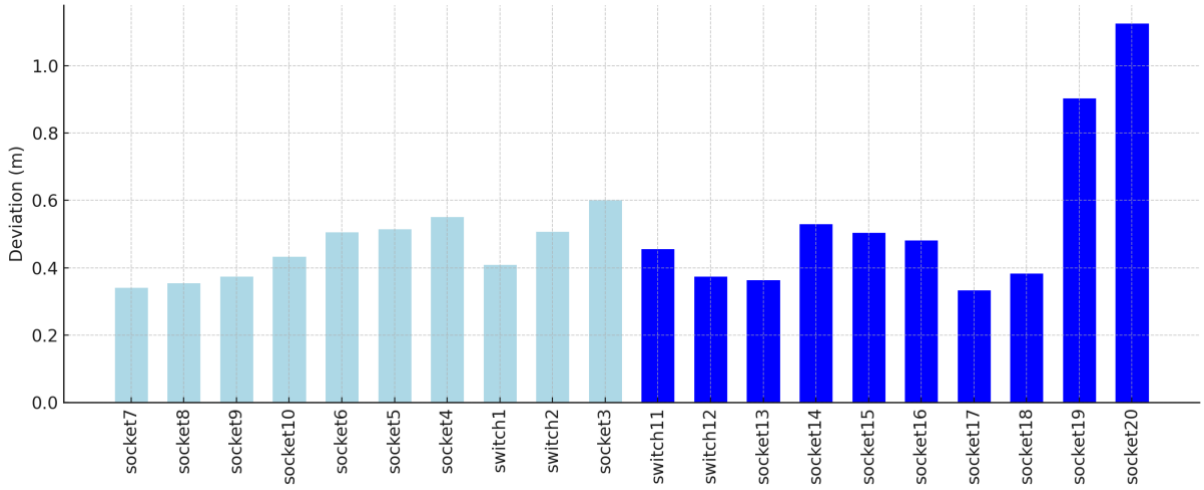

Figure 11: Assessment of Detection Deviations Across Elements in Experiment 3.

5. DISCUSSION

5.1 Performance Analysis

In light of the experimental findings, it can be deduced that optimal system performance is attained when θ_a is less than 12, d_{co} is less than 1.5, and θ_i is less than 10. Under these specific conditions, the system demonstrates enhanced efficacy, as evidenced by a mean deviation of approximately 16.1. This conclusion aligns with findings reported in (Hubner et al, 2020), where optimal performance is also shown to depend on maintaining low angles of incidence and short distances. Specifically, the HoloLens depth sensor, as evaluated in their work, exhibits minimal noise when the angle of incidence is kept below 20, similar to the our study's recommendation of θ_i less than 10. Furthermore, both studies highlight the importance of proximity, with performance degradation occurring beyond 2.5 in the HoloLens study, echoing the results of this experiment, which found d_{co} should remain below 1.5 for optimal accuracy.

As a result, it appears that using the current prototype system based on state-of-the-art MR technology (Hololens), construction positioning conformance can only be confirmed within 16 cm, under very constraining conditions (short distance and low incidence angle). In fact, to further evaluate the impact of the incidence angle, an additional experiment was conducted with the user standing and tilting their head downward to observe the elements, which would likely occur in practice. In this scenario, the observed deviation further increased to 50 cm. Unfortunately, this significantly limits the range of possible use cases. While the current prototype may still be useful to detect and match some elements, it cannot be reliably used for positioning conformance control.

The reported rapid degradations in performance can be ascribed to the following factors:

- Pose tracking inaccuracies result from two types of errors. There is an inherent error in the process of initializing the camera's location and orientation using QR code scanning. As discussed earlier, this error results in a positional deviation of approximately 3.5 cm. More critically, errors from devices like inertial measurement units (IMUs) introduce further pose tracking errors, especially during quick motions.
- Errors in camera intrinsic parameter estimation (e.g., perspective and lens distortions) influence the perceived sizes and shapes within an image, potentially introducing errors during the projection process between two-dimensional and three-dimensional environments.
- Environmental conditions also impact the results.The way lighting and shadows appear in the image affects how accurately objects are detected and represented.
- For a system like the one proposed here to achieve a level of performance that is sufficient for all envisioned use cases, the following enhancements should be pursued:

- To improve the accuracy of camera pose tracking, more robust IMUs could be considered. Alternatively, additional sensors (e.g., beacon-based location tracking) can be integrated to enhance both pose initialization and pose tracking. In fact, this is the strategy already favored by current commercial AR solutions for construction applications. For indoor use, vGIS (vGIS Incorporated, 2024) or XYZ Reality (World Intellectual Property Organization, 2021) employ beacons for assistance, while for outdoor use, high-accuracy AR systems integrate GNSS and RTK to achieve precision levels up to 1 cm.
- Accurately identifying complex elements like multi-functional media sockets is challenging due to their diverse designs and the need to distinguish their specific types and orientations. Sophisticated object detection technologies, trained on an extensive array of socket designs and configurations, must be developed to enable broad and robust applicability in practice. Notwithstanding, the proposed detection zone-based approach, and more generally utilizing any prior information contained in the BIM model (or Digital Twin), are important strategies to reduce the need for solutions that must be overly general (and thus difficult to train).

5.2 Role of Users and System Implications

The proposed system is designed to augment human inspectors by automating routine construction inspection tasks while preserving the critical role of human expertise in complex scenarios. By leveraging Mixed Reality (MR), the system minimizes user intervention during routine inspections. Users are alerted only when non-compliance is detected, as indicated by red-colored marks on the display, allowing them to focus on other tasks until their attention is needed. This hands-free approach enhances task efficiency and minimizes disruptions to the user's workflow.

The system operates in real-time or near-real-time, ensuring seamless interaction during inspections. While this study focuses on the system's design and functionality, future work should include comprehensive user experience testing to evaluate usability, task efficiency, and overall performance. These studies will guide further refinements to ensure the system aligns with the practical needs of construction professionals.

While robotic systems equipped with cameras and sensors could theoretically replicate some inspection tasks, they lack the adaptability and contextual reasoning required in dynamic construction environments. Construction sites often present unique and unforeseen challenges, such as temporary obstructions, changing environmental conditions, or project-specific nuances, which are still best handled by human judgment. Human inspectors, supported by the proposed system, can interpret site-specific complexities, prioritize tasks, and interact with stakeholders to make informed decisions. This hybrid, light human-in-the-loop approach balances the precision of automated detection with the flexibility and problem-solving abilities unique to humans, ensuring the system remains practical and effective for real-world applications.

6. CONCLUSION

This paper explored the feasibility of employing a novel MR-based construction inspection framework that integrates AI-based object detection with 2D-to-3D projection techniques and matching against the facility's Digital Twin (DT) information to enable more autonomous inspection workflows than current approaches. The system is facilitated by a robust communication mechanism between the MR device and the Computation Centre. Inspection results are stored in the DT and reported to the MR user on-site in real-time, enabling prompt decision-making.

The practicality and effectiveness of the framework were evaluated in two real-life indoor environments, demonstrating the system's feasibility in real-world inspection processes. However, the results also highlighted limitations in tracking accuracy and detection quality, pointing to areas requiring further research. In particular, future work could focus on enhancing the accuracy of camera pose tracking through the integration of external sensors, which would improve both initialization and continuous pose tracking. Additionally, object detection systems could be expanded to recognize more complex elements through training on diverse datasets. These technical enhancements would not only refine the system's performance but also unlock broader applications, such as monitoring compliance with safety regulations on construction sites.

ACKNOWLEDGEMENT

For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) license to any Author Accepted Manuscript version arising from this submission.

REFERENCES

Chahrour, R., Hafeez, M. A., Ahmad, A. M., Sulieman, H. I., Dawood, H., Rodriguez-Trejo, S., Kassem, M., Naji, K. K., Dawood, N. (2021). Cost-benefit analysis of bim-enabled design clash detection and resolution. Construction Management and Economics, 39, 55-72. doi:10.1080/01446193.2020.1802768

Chalhoub, J., Ayer, S. K. (2018). Using mixed reality for electrical construction design communication. Automation in Construction, 86, 1-10. doi:10.1016/j.autcon.2017.10.028

Chen, L., Luo, H. (2014). A bim-based construction quality management model and its applications. Automation in Construction, 46, 64-73. doi:10.1016/j.autcon.2014.05.009

Chow, J. K., Su, Z., Wu, J., Li, Z., Tan, P. S., Liu, K. F., Mao, X., Wang, Y.-H. (2020). Artificial intelligence-empowered pipeline for image-based inspection of concrete structures. Automation in Construction, 120, 103372. doi:10.1016/j.autcon.2020.103372

Corneli, A., Naticchia, B., Vaccarini, M., Bosché, F., Carbonari, A. (2020). Training of Yolo Neural Network for the detection of fire emergency assets. In ISARC Proceedings of the International Symposium on Automation and Robotics in Construction, 37, 836-843. doi:10.22260/ISARC2020/0115

Dai, F., Olorunfemi, A., Peng, W., Cao, D., Luo, X. (2021). Can mixed reality enhance safety communication on construction sites? An industry perspective. Safety Science, 133, 105009. doi:10.1016/j.ssci.2020.105009

De Filippo, M., Asadiabadi, S., Kuang, J., Mishra, D., Sun, H. (2023). AI-powered inspections of facades in reinforced concrete buildings. Transactions Hong Kong Institution of Engineers, 30, 1-14. doi:10.33430/V30N1THIE-2020-0023

Dibene, J. C., Dunn, E. (2022). HoloLens 2 Sensor Streaming. arXiv preprint arXiv:2211.02648. doi:10.48550/arXiv.2211.02648

Dunston, P. S., Wang, X. (2005). Mixed reality-based visualization interfaces for architecture, engineering, and construction industry. Journal of Construction Engineering and Management, 131, 1301-1309. doi:10.1061/(ASCE)0733-9364(2005)131:12(1301)

El Ammari, K., Hammad, A. (2019). Remote interactive collaboration in facilities management using BIM-based mixed reality. Automation in Construction, 107, 102940. doi:10.1016/j.autcon.2019.102940

Feng, C.-W., Chen, C.-W. (2019). Using BIM and MR to improve the process of job site construction and inspection. WIT Transactions on the Built Environment, 192, 21-32. doi:10.2495/BIM190031

Herrera, R. F., Mourgues, C., Alarcón, L. F., Pellicer, E. (2021). Comparing team interactions in traditional and bim-lean design management. Buildings, 11. doi:10.3390/buildings11100447

Holzwarth, V., Steiner, S., Schneider, J., vom Brocke, J., Kunz, A. (2021). BIM-enabled issue and progress tracking services using mixed reality. In Smart Services Summit: Digital as an Enabler for Smart Service Business Development, Springer, 49-58. doi:10.1007/978-3-030-72090-2_5

Hübner, P., Clintworth, K., Liu, Q., Weinmann, M., Wursthorn, S. (2020). Evaluation of hololens tracking and depth sensing for indoor mapping applications. Sensors, 20. doi:10.3390/s20041021

Jocher, G., Stoken, A., Borovec, J., Changyu, L., Hogan, A., Diaconu, L., Poznanski, J., Yu, L., Rai, P., Ferriday, R., et al. (2020). ultralytics/yolov5: v3. 0. Zenodo. doi:10.5281/zenodo.3983579

Karaaslan, E., Zakaria, M., Catbas, F. N. (2022). Mixed reality-assisted smart bridge inspection for future smart cities. In The Rise of Smart Cities, Elsevier, 261-280. doi:10.1016/B978-0-12-817784-6.00002-3

Khan, N., Zaidi, S. F. A., Yang, J., Park, C., Lee, D. (2023). Construction work-stage-based rule compliance monitoring framework using computer vision (cv) technology. Buildings, 13. doi:10.3390/buildings13082093

Kim, H., Kim, C. (2020). Deep-learning-based classification of point clouds for bridge inspection. Remote Sensing, 12. doi:10.3390/rs12223757

Kim, J.-I., Gang, H.-S., Pyun, J.-Y., Kwon, G.-R. (2021). Implementation of qr code recognition technology using smartphone camera for indoor positioning. Energies, 14, 2759. doi:10.3390/en14102759

Kim, M.-K., Cheng, J. C., Sohn, H., Chang, C.-C. (2015). A framework for dimensional and surface quality assessment of precast concrete elements using bim and 3d laser scanning. Automation in Construction, 49, 225-238. doi:10.1016/j.autcon.2014.07.010

Ma, G., Wu, M., Wu, Z., Yang, W. (2021). Single-shot multibox detector- and building information modeling-based quality inspection model for construction projects. Journal of Building Engineering, 38, 102216. doi:10.1016/j.jobe.2021.102216

Megahed, N. A., Hassan, A. M. (2022). Evolution of bim to dts: A paradigm shift for the post-pandemic aeco industry. Urban Science, 6. doi:10.3390/urbansci6040067

Microsoft. (2023). Locatable camera in mixed reality. Retrieved from https://learn.microsoft.com/en-us/windows/mixed-reality/develop/advanced-concepts/locatable-camera-overview

Milgram, P., Takemura, H., Utsumi, A., Kishino, F. (1995). Augmented reality: a class of displays on the reality-virtuality continuum. In H. Das (Ed.), Telemanipulator and Telepresence Technologies, 2351, 282-292. doi:10.1117/12.197321

Naticchia, B., Corneli, A., Carbonari, A., Bosché, F., Principi, L. (2019). Augmented reality application supporting on-site secondary building assets management. In Proceedings of the Creative Construction Conference (CCC), 806-811. doi:10.3311/CCC2019-110

Nguyen, D. C., Jin, R., Jeon, C. H. (2020). Developing a mixed-reality based application for bridge inspection and maintenance. In The 20th International Conference on Construction Applications of Virtual Reality (CONVR 2020), Teeside University. doi:10.1108/CI-04-2021-0069

Nguyen, D.-C., Nguyen, T.-Q., Jin, R., Jeon, C.-H., Shim, C.-S. (2022). BIM-based mixed-reality application for bridge inspection and maintenance. Construction Innovation, 22, 487-503. doi:10.1108/CI-04-2021-0069

Paneru, S., Jeelani, I. (2021). Computer vision applications in construction: Current state, opportunities & challenges. Automation in Construction, 132, 103940. doi:10.1016/j.autcon.2021.103940

Prabhakaran, A., Mahamadu, A.-M., Mahdjoubi, L. (2022). Understanding the challenges of immersive technology use in the architecture and construction industry: A systematic review. Automation in Construction, 137, 104228. doi:10.1016/j.autcon.2022.104228

Rahmani Asl, M., Zarrinmehr, S., Bergin, M., Yan, W. (2015). BPOpt: A framework for BIM-based performance optimization. Energy and Buildings, 108, 401-412. doi:10.1016/j.enbuild.2015.09.011

Riedlinger, U., Klein, F., Hill, M., Lambracht, C., Nieborowski, S., Holst, R., Bahlau, S., Oppermann, L. (2022). Evaluation of Mixed Reality Support for Bridge Inspectors Using BIM Data: Digital Prototype for a Manual Task with a Long-Lasting Tradition. i-com, 21, 253-267. doi:10.1515/icom-2022-0019

Riexinger, G., Kluth, A., Olbrich, M., Braun, J.-D., Bauernhansl, T. (2018). Mixed reality for on-site self-instruction and self-inspection with building information models. Procedia CIRP, 72, 1124-1129. doi:10.1016/j.procir.2018.03.160

Sarlin, P.-E., Unagar, A., Larsson, M., Germain, H., Toft, C., Larsson, V., Pollefeys, M., Lepetit, V., Hammarstrand, L., Kahl, F., Sattler, T. (2021). Back to the Feature: Learning Robust Camera Localization from Pixels to Pose. In 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 3246-3256. doi:10.1109/CVPR46437.2021.00326

Sepasgozar, S. M. E. (2021). Differentiating digital twin from digital shadow: Elucidating a paradigm shift to expedite a smart, sustainable built environment. Buildings, 11. doi:10.3390/buildings11040151

Tang, F., Ma, T., Zhang, J., Guan, Y., Chen, L. (2020). Integrating three-dimensional road design and pavement structure analysis based on bim. Automation in Construction, 113, 103152. doi:10.1016/j.autcon.2020.103152

Tao, B., Li, J., Bosché, F. (2024). Smart passive mixed reality-based construction inspection framework. In Proceedings of the 41st International Symposium on Automation and Robotics in Construction, International Association for Automation and Robotics in Construction (IAARC), Lille, France, 776-783. doi:10.22260/ISARC2024/0101

vGIS Incorporated. (2024). High-accuracy Augmented Reality (AR) for BIM, GIS, Reality Capture, GNSS, RTK, and Total Station. Retrieved from https://www.vgis.io/high-accuracy-augmented-reality-ar-for-bim-gis-reality-capture-gnss-rtk-total-station-survey-grade-leica-trimb

World Intellectual Property Organization. (2021). XYZ Reality Brings Precision and Technology to Construction Sites. Retrieved from https://www.wipo.int/wipo_magazine/en/ip-at-work/2021/xyz.html

Zaccardi, S., Frantz, T., Beckwée, D., Swinnen, E., Jansen, B. (2023). On-device execution of deep learning models on hololens2 for real-time augmented reality medical applications. Sensors, 23. doi:10.3390/s23218698

Zakaria, M., Karaaslan, E., Catbas, F. N. (2023). Real-Time AI-based Bridge Inspection Using Mixed Reality Platform. In Structures Congress 2023, 120-131. doi:10.1061/9780784484777.012

Zhang, Z. (1999). Flexible camera calibration by viewing a plane from unknown orientations. In Proceedings of the Seventh IEEE International Conference on Computer Vision, 1, 666-673. doi:10.1109/ICCV.1999.791289