國立臺灣大學工學院土木工程學系

博士論文

Department of Civil Engineering

College of Engineering

National Taiwan University

Doctoral Dissertation

基於影像及排程之營建數位孿生自動化進度監控

Automated activity-level progress monitoring from visual data and schedules through digital twin construction

安里達

Aritra Pal

指導教授: 謝尚賢博士, 林之謙 博士

Advisor: Shang-Hsien (Patrick) Hsieh, Ph.D., Jacob Je-Chian Lin, Ph.D.

中華民國 112 年 7 月

July, 2023

doi:10.6342/NTU202302333

國立臺灣大學博士學位論文
口試委員會審定書

National Taiwan University
Doctoral Dissertation Acceptance Certificate

基於影像及排程之營建數位孿生自動化進度監控

Automated activity-level progress monitoring from visual data and schedules through digital twin construction

本論文係 Aritra Pal 君 (D08521024) 在國立臺灣大學土木工程學系完成之博士學位論文，於民國 2023年7月17日 承下列考試委員審查通過及口試及格，特此證明

This is to certify that this Doctoral Dissertation is completed by **Aritra Pal (D08521024)** during his/her studying in Department of Civil Engineering at National Taiwan University, and that the oral defense of this thesis/dissertation is passed on **2023/7/17** in accordance with decision of following committee members:

口試委員 Oral Defense Committee members：

Shang-Hsien (Patrick) HSIEH (**指導教授** Advisor)	Shang-Hsien Hsieh
Jacob Je-Chian LIN (**共同指導** Co-Advisor)	
Chuin-Shan CHEN	
Chien-Cheng CHOU	Chien-Cheng Chou
I-Tung YANG	I-Tung Yang
Shih-Chung KANG	Shih-Chung Kang
Po-Han CHEN	Po-Han Chen

系主任、所長 Louis Ge (**簽名** Signature)

(Department Chair/Program Director)

doi:10.6342/NTU202302333

doi:10.6342/NTU202302333

Acknowledgements

Completing a Ph.D. is no easy feat, and it requires the support and encouragement of many people along the way. As such, I would like to express my profound gratitude to everyone who has played a role in helping me achieve this significant milestone in my academic career.

First and foremost, I would like to extend my deepest thanks to **my advisors, Prof. Shang-Hsien (Patrick) Hsieh and Prof. Jacob J. Lin**, for their unwavering guidance, mentorship, and support throughout my Ph.D. journey. Their dedication to my research and their willingness to provide constructive feedback has been invaluable in shaping my ideas and helping me to grow as a scholar. Their intellectual curiosity, enthusiasm, and kindness have set the standard for excellence, and I feel incredibly fortunate to have had them as my mentor.

In addition to my advisor, I would like to thank the **members of my dissertation committee** for their insightful comments and feedback during the various stages of my research. Their expertise in their respective fields has enriched my work, and their encouragement and support have helped me to persevere through the many challenges of completing my Ph.D.

I am also profoundly grateful to **my colleagues and friends** for their intellectual

doi:10.6342/NTU202302333

companionship, camaraderie, and encouragement. They have provided a stimulating and collaborative environment to conduct my research, and their support and friendship have made the long hours and hard work of a Ph.D. program more bearable.

I am incredibly grateful for **my wife, Paromeeta**, whose unwavering support and love have been the bedrock of my doctoral journey. Her encouragement, understanding, and sacrifices have propelled me forward, and her belief in my abilities has been my guiding light. I am forever grateful for her presence in my life, and I dedicate this dissertation to her with deep love and appreciation.

I would be remiss if I did not acknowledge the love and support of **my family**. Their continuous encouragement has been a constant source of inspiration, and I am grateful for their belief in me even during the most challenging times.

Finally, I would like to express my gratitude to the **funding agencies and institutions: the National Science and Technology Council, Taiwan, and the Ministry of Education, Taiwan**, who have supported my research and studies. Their financial support has enabled me to pursue my passion for knowledge and helped bring my research to fruition. Also, I would like to express my deep appreciation to the **construction companies** that generously provided the data necessary for this research.

In sum, I owe a tremendous debt of gratitude to all those who have contributed to my Ph.D. journey, and I feel honored and privileged to have had such a great network of support. Without their collective efforts, this thesis would not have been possible, and I am grateful beyond words for their kindness and generosity. Thank you from the bottom of my heart!

doi:10.6342/NTU202302333

摘要

營建工程的進度管控是工程成功交付的重要關鍵，而目前影像資料已是瞭解工程進度的重要資訊來源之一。在過去幾十年中，許多研究開發應用電腦視覺於建築施工的自動進度監測的方法，這些方法在監測個別元件（如柱子、樑、牆壁）方面非常有效，但在監測由元件組成之排程工項工程進度（如一樓模板工程、鋼筋綁匝、混凝土澆置）仍有困難。現有方法通常難以推測介於已建或未建之間的未完成進度之工程狀態，因此限制了此技術在分析細節進度資訊方面的應用。本研究旨在透過兩種新的方法解決之研究問題。第一種方法稱為排程工項進度監測系統（ALPMS），旨在監測正在施工的元件之活動級別進度，主要以施工現場影像和四維建築信息模型（BIM）作為輸入，產出數位孿生資訊系統。該系統從影像中生成實境現場的點雲，將其與原排程中的BIM進行比較，並應用基於深度學習的語義分割進行進度推理。因此可估計每個工項的完成百分比，並且為更新專案進度提供有價值的資訊。數位孿生資訊系統同時可將語義資訊整合到實際建造的點雲和BIM中，實現進度狀態的三維可視化。第二種方法研究於在缺乏更新的四維BIM的情況下也能自動比對專案進度與實際模型，該比對方法首先使用三維BIM或控制點將實際模型對應到世界坐標系統，然後應用點雲分割來檢測與特定位置、建築元件和工項相關的進度。使用基於自然語言處理（NLP）的技術從每個工項中提取相關位置、元件和任務的資訊。從實際模型和工項中提取的資料通過基於語意距離的匹配技術進行比對，再將進度資訊與相應的進度活動進行比

doi:10.6342/NTU202302333

對。以上提出的方法已在台灣的工程專案上應用及評估並且有展示相關的有效性和適用性，ALPMS 成功分析工項等級的進度狀態，平均絕對誤差少於 6%, 而缺乏四維 BIM 的情況下仍可準確更新進度資料。這些方法論通過提供有關元件和工項等級進度的分析，為工程進度監測領域進一步的貢獻，也可使人們更容易地理解專案狀態，以實現高效的進度管理和做出明智的決策，進而促進專案成功交付。

關鍵字：深度學習、計算機視覺、活動級進度監控、神經輻射場、自然語言處理、日程更新、數字孿生

doi:10.6342/NTU202302333

Abstract

Monitoring the progress of construction projects is crucial for ensuring successful project delivery. Visual data, such as images and videos, has emerged as a valuable source of information to understand the status of construction operations. Over the past few decades, several vision-based methods have been developed for automated progress monitoring in building construction. These methods have been effective in monitoring individual elements (e.g., columns, beams, walls) but face challenges in monitoring progress at the schedule activity level (e.g., formwork, reinforcement, concrete). Existing methods often struggle to report progress status beyond a binary form of built or not-built, limiting their usefulness in capturing nuanced progress information. This research focuses on addressing these challenges through two novel methodologies. The first methodology, called the Activity Level Progress Monitoring System (ALPMS), aims to monitor progress at the activity level of under-construction building elements. It takes construction site images and a four-dimensional building information model (BIM) as input and creates a Digital

doi:10.6342/NTU202302333

Twin information system. The system generates as-built point clouds from the images, compares them with the as-planned BIM, and applies deep learning-based semantic segmentation for progress reasoning. This enables the estimation of activity-wise completion percentages, providing valuable information for updating project schedules. The DT information system also integrates rich semantic information into the as-built point cloud and BIM, enabling three-dimensional visualization of progress status. The second methodology focuses on automatically aligning project schedules with reality models, even in the absence of an updated 4D BIM. The alignment method starts by aligning reality models to the world coordinate system using a 3D BIM or control points. Point cloud segmentation is then applied to detect progress associated with specific locations, building elements, and tasks. Information about locations, elements, and tasks is extracted from each schedule activity using natural language processing (NLP)-based techniques. Extracted information from the reality models and the schedule activities are matched through a distance-based matching technique, mapping the progress information with the corresponding schedule activities. The proposed methodologies have been applied and evaluated on construction projects in Taiwan, demonstrating their effectiveness and applicability. The ALPMS successfully reports activity level progress status with less than 6% mean absolute error. The automatic alignment method shows promise in accurately updating progress infor-

doi:10.6342/NTU202302333

mation without relying on an updated 4D BIM. These methodologies contribute to the field of construction progress monitoring by providing accurate and detailed insights into progress at both the element and activity levels. They enable a better understanding of project status, efficient schedule management, and informed decision-making, ultimately facilitating successful project delivery.

Keywords: Deep learning, computer vision, activity-level progress monitoring, neural radiance field, natural language processing, schedule update, digital twin

doi:10.6342/NTU202302333

doi:10.6342/NTU202302333

Table of Contents

doi:10.6342/NTU202302333

doi:10.6342/NTU202302333

doi:10.6342/NTU202302333

doi:10.6342/NTU202302333

List of Figures

doi:10.6342/NTU202302333

doi:10.6342/NTU202302333

doi:10.6342/NTU202302333

List of Tables

doi:10.6342/NTU202302333

doi:10.6342/NTU202302333

Abbreviations

ALPMS	Activity Level Progress Monitoring System
BIM	Building Information Modeling
NeRF	Neural Radiance Fields
AI	Artificial Intelligence
DPR	Daily Progress Reports
GPU	Graphics Processing Units
CNN	Convolutional Neural Networks
WBS	Work Breakdown Structures
PMBOK	Project Management Body of Knowledge
ML	Machine Learning
DL	Deep Learning
IFC	Industry Foundation Class

doi:10.6342/NTU202302333

CMU	Concrete Masonry Units
SfM	Structure from Motion
MVS	Multiview Stereo
PM	Progress Monitoring
CAD	Computer-Aided Design
EVM	Earned Value Management
MEP	Mechanical Electrical and Plumbing
TLS	Terrestrial Laser Scanners
HVAC	Heating, Ventilation, and Air Conditioning
LoD	Level of Development
FOV	Field of View
ICP	Iterative Closest Point
DOF	Degrees of Freedom
SOR	Statistical Outlier Removal
WCS	World Coordinate System
PCS	Pixel Coordinate System
ROI	Region of Interest

MLP	Multilayer Perceptron
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity Index Measure
IoU	Intersection over Union
mAP	mean Average Precision
P-R	Precision-Recall
FC	Fully Connected
LR	Learning Rate
SGD	Stochastic Gradient Descent
MAE	Mean Absolute Error
L-E-M	Location - Building Element - Material
CP	Control Point
WBS	Work Breakdown Structure
NER	Named-entity Recognition
NLP	Natural Language Processing
BERT	Bidirectional Encoder Representations from Transformers
GPT3	Generative Pre-trained Transformer-3

doi:10.6342/NTU202302333

LLM	Large Language Model
S3DIS	Stanford 3D Indoor Scene Dataset
S3DIS	Stanford 3D Indoor Scene Dataset

Chapter 1 Introduction

1.1 Motivation and Background

Global demand for building construction is increasing. According to a recent report by Autodesk Redshift (2019), 13,000 buildings need to be built every day up to 2050 to meet the demand of 7 million people living in cities. On the contrary, the construction industry faces challenges like delays in project delivery, cost overruns, disputes, and low productivity. MGI 2017 reported that construction-related spending contributes 13% to the global GDP but the annual productivity growth for this sector has remained at only 1% over the past twenty years. This growth rate is much lower than the global average of 2.8%, as well as the 3.6% growth rate of the manufacturing industry (McKinsey Global Institute 2017). According to a 2021 report by the Office for National Statistics (ONS), UK, the construction industry has one of the lowest productivity growth rates among all major sectors in the United Kingdom (UK). Although the labor productivity rate has grown slightly faster since 2008, it remains below the country average [60]. Automation and digitization in the construction processes could tackle many of the age-old problems of this industry and thus increase the construction businesses' profitability. World economic forum's (2016) report [23] indicates that full-scale digitization in the construction industry could save \$0.7-1.2 trillion in the design, engineering, and construction phase in ten

doi:10.6342/NTU202302333

years. Similarly, Accenture's report highlights that with the incorporation of artificial intelligence (AI) in the construction industry, the profit may increase by 71% by 2035 [75]. Recently, the maturity of building information management techniques and the advancement in real-world data collection, storing, and processing technologies have created a unique opportunity for construction researchers and practitioners to automate various project management tasks such as construction progress monitoring (CPM), quality inspection, productivity assessment, and safety management [68]. Effective progress monitoring at construction sites is necessary to ensure successful project delivery. By promptly monitoring a construction project's progress, construction professionals can identify and address any issues that may arise during the construction process, such as delays or quality issues, which can help improve the overall efficiency of the construction process. Providing the most up-to-date information about the project status to all the stakeholders can also enhance the opportunities for proactive project control and reduce the risks of schedule delays and cost overruns [51]. Conventionally, progress monitoring is done through manual site measurements of as-built constructions and paper-based daily progress reports (DPR). Later the project schedules are updated with the physical progress percentages retrieved from the DPRs. However, this traditional method is time-consuming and needs considerable human effort. Additionally, paper-based reporting cannot instantly verify three-dimensional (3D) progress status. To overcome these challenges, researchers have developed various automated progress monitoring methods. From time to time, researchers have investigated various digital technologies, such as Radio Frequency Identification (RFID), Ultra-Wide Band (UWB), barcodes, digital imaging, laser scanning, and photogrammetry, for automating the CPM processes [67] in building and infrastructure projects. Vision-based automated construction progress monitoring through visual data analytics has gained massive popularity among researchers and practitioners in the last couple of decades because of the following reasons: (1) the increasing availability of low-cost visual data capturing devices, (2) the advancement of graphics processing hardware technologies, and affordable prices of graphics processing units (GPU) in recent times, (3) advent of computer vision and computer graphics technologies, and (4) inception of convolutional neural networks (CNN) for deep-learning-based visual data analytics [68].

The state-of-the-art vision-based progress monitoring methods are categorized into occupancy-based and appearance-based [100]. In occupancy-based methods, as-built reality captured models are compared with as-planned models such as BIMs and their physical occupancy is assessed in a 3D space. For example, [14, 17] introduced a method for detecting 3D CAD model elements from point clouds for checking dimensional compliance. Later [88, 89] used similar methods for schedule monitoring and earned value analysis. On the other hand, in appearance-based methods, the progress status is inferred by analyzing the appearance of BIM elements in 2D images.[38] introduced an operation-level progress monitoring method that relied on appearance-based construction material classification. The technique was further improved and integrated with the occupancy-based methods by [37]. Appearance detection in hyperspectral images generated from laser scans was studied by [5] and [101].[52] used deep-learning-based material classification for appearance-based construction progress monitoring. More recent studies have also leveraged extended reality techniques [3], [45], [72], digital twining methods [4], deep-learning-based 2D [19], [54], and 3D object detection algorithms [43], [99] deep-learning-based 3D point cloud segmentation approaches [80], [58], [2] and automatic scan-to-BIM techniques [21] for enhancing the occupancy and appearance-based progress monitoring methods. Although there have been significant improvements in progress detection

doi:10.6342/NTU202302333

from visual data, some knowledge gaps still hinder its widespread adoption in the construction industry, such as 1) existing methods are more suitable for monitoring element-level progress (beam, column, slab, wall, floor, etc.) than task-level progress (concrete, formwork, rebar, painting, plastering, etc.), 2) Determining the percentage completion of construction activities is still difficult as the state-of-the-art methods report construction progress in binary form, and 3) the schedule update process still relies on manual inputs of detected progress. Therefore an improved version of automated CPM is needed to address these challenges to monitor the progress of construction activities against project baseline schedules. These existing methodological gaps are described in detail in the next sub-sections.

1.2 Problem statement

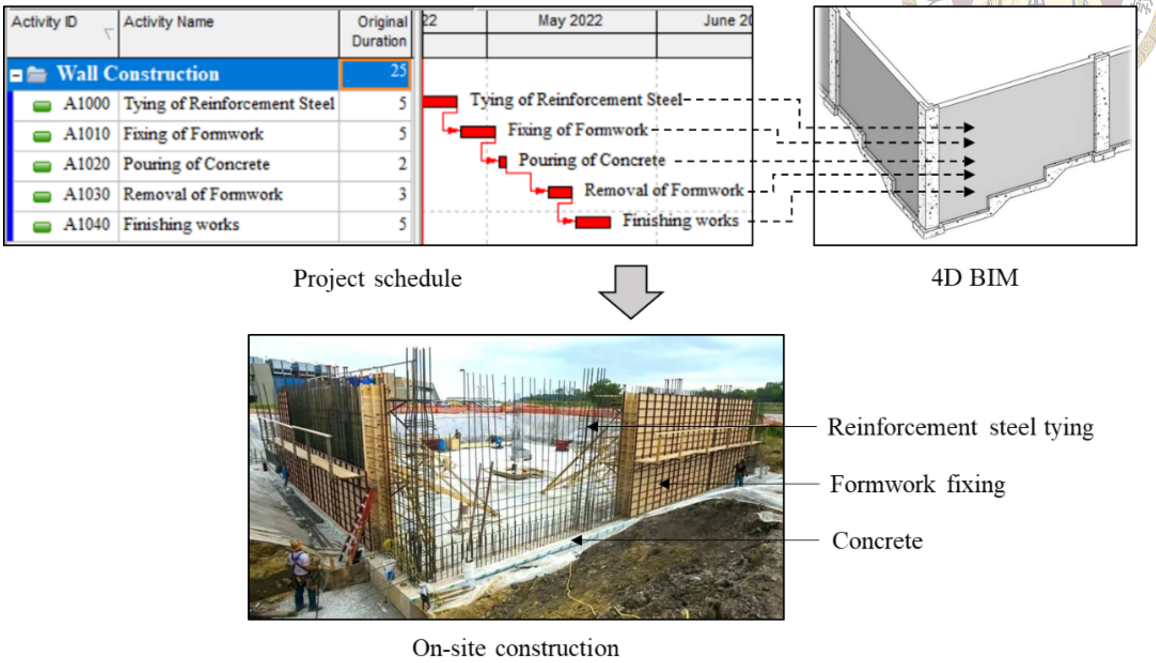
Three major methodological gaps that need immediate attention from the researchers are as follows: 1) Lack of methods for activity-level progress monitoring; 2) Lack of methods for reporting partial completion of progress; 3) Lack of methods for the automatic schedule update. In the subsequent sections, these are discussed in detail.

1.2.1 Lack of methods for activity-level progress monitoring

The project management body of knowledge (PMBOK) guide suggests dividing the projects into work breakdown structures (WBS) for better management of the projects. WBSs are further decomposed into schedule activities for effectively estimating, scheduling, executing, monitoring, and controlling the project work [73]. However, 3D BIMs used during construction are not often modeled to that detailed level [38]. In such cases, while creating the 4D BIM for construction, multiple schedule activities are often linked to a single BIM element. For that reason, inferring the progress status of individual activities becomes difficult, and progress monitoring for that element can only be done after completing all scheduled activities. In Figure 1.1, let's look at the construction schedule of wall construction. It consists of 5 sequential activities starting from the tying of reinforcement steel. The as-planned BIM available for the construction phase only contains one element to represent that wall. So, all sequential activities are linked to that element for creating the as-planned 4D BIM. However, actual construction at the site takes place as per the construction sequences followed in the project schedule. As occupancy-based progress monitoring methods fully rely on geometry modeling, the progress status of the activities that are not modeled in BIM (e.g., formwork, reinforcement steel) can't be inferred using such methods. Few researchers, [38] and [37], tried to overcome such challenges through appearance-based progress detection. However, in such methods, the activity must be 100% completed to be detected during the progress monitoring. Then only the activity details can be inferred from the BIM-registered image patches by conventional machine learning (ML) based construction material classification. Suppose an activity is partially completed, and the test image patch corresponding to a BIM element contains the appearance of more than one construction material. In that case, these methods may not be workable. Figure 1.2 shows one such situation in the example of plastering on a brick wall.

1.2.2 Lack of methods for reporting partial completion of progress

There can be two possible cases while dealing with activity level progress monitoring. Case 1: As-planned BIM is modeled with enough details to link one schedule activity



Activity ID	Activity Name	Original Duration
Wall Construction		25
A1000	Tying of Reinforcement Steel	5
A1010	Fixing of Formwork	5
A1020	Pouring of Concrete	2
A1030	Removal of Formwork	3
A1040	Finishing works	5

Figure 1.1: An example showing many-to-one relationship between project schedule and BIM

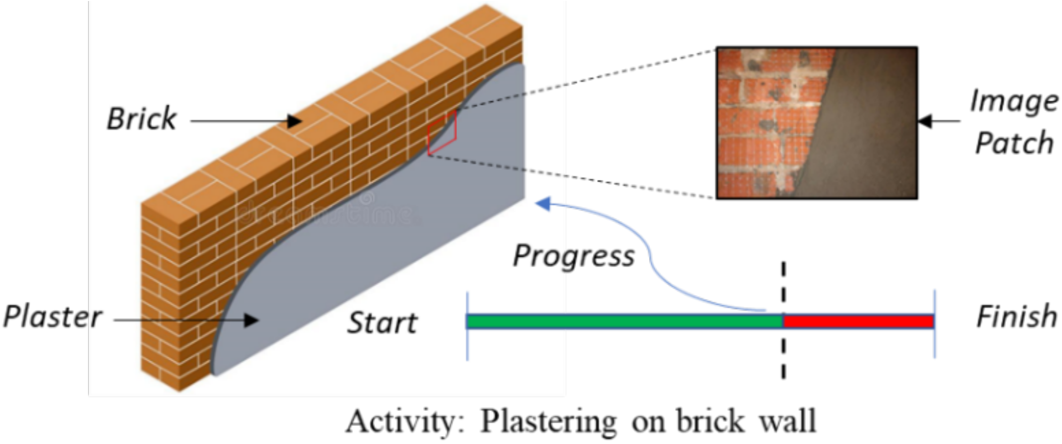


Figure 1.2: An example showing the appearance of multiple materials on a single image patch

doi:10.6342/NTU202302333

with one BIM element. Case 2: As-planned BIM has a low level of development (or detail) (LOD), so multiple schedule activities need to be linked to one BIM element. The requirement for partial completion reporting may arise in both cases because one or more activities may remain in an 'in-progress' state at the time of progress monitoring. The existing vision-based progress monitoring methods mostly report the progress in binary forms such as built/ not built [19] and completed/ not completed [38]. So, the progress status of 'in-progress' activities is often not reported. In Figure 1.3, let's look at the example of the construction of a wall with concrete masonry units (CMU). In this case, the progress reporting through the existing methods is less useful as the construction activity is partially completed on the progress monitoring date. The progress status of the activity can only be reported after the completion of the wall construction. However, recording partial completion of progress and reporting it in terms of completion percentage is necessary for effective project control. The same progress percentages are also required for updating the project schedule. Figure 1.4 shows the screenshot of a popular project scheduling software where progress percentages are being recorded while updating the project schedule. Very few researchers [66], [76] have tried to report progress in the completion percentage. However, their methods entirely relied on occupancy-based progress monitoring, which is unsuitable for inferring the activity level progress status. To the best of the author's knowledge, there is no method to simultaneously identify the activity level progress status and report partial completion of progress in terms of completion percentage.

1.2.3 Lack of methods for automatic schedule update

A construction progress update is a critical process that involves evaluating the progress of a construction project and making necessary updates to the timeline. It is essential to

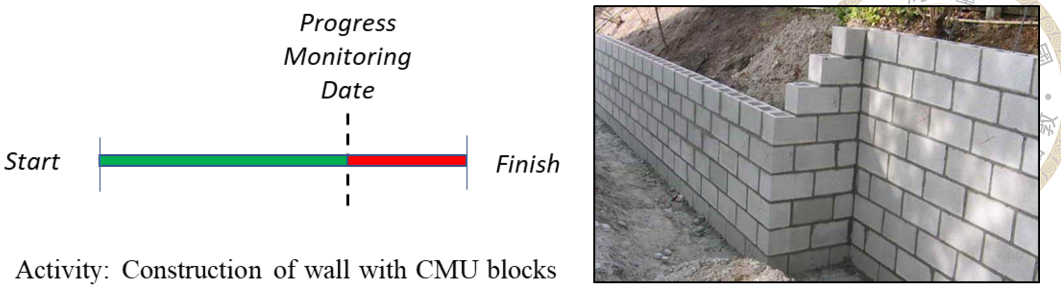
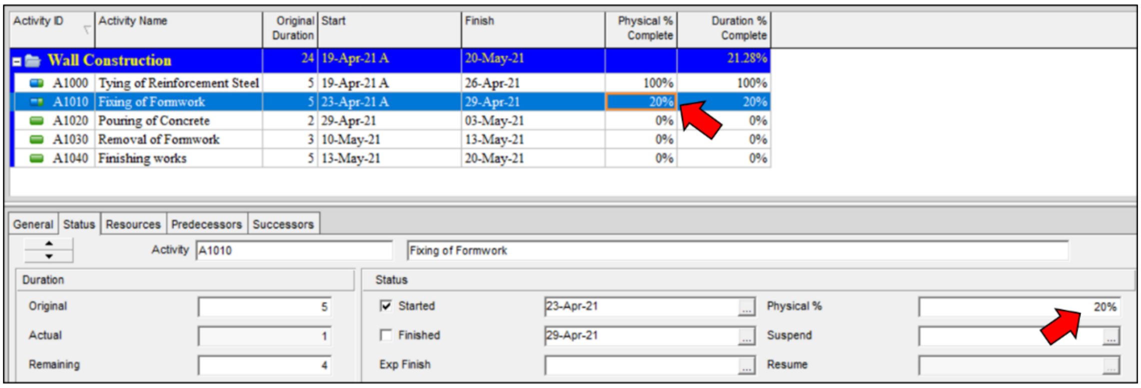


Figure 1.3: Example showing partial completion of construction on the progress reporting date



Activity ID	Activity Name	Original Duration	Start	Finish	Physical % Complete	Duration % Complete
	Wall Construction	24	19-Apr-21 A	20-May-21		21.28%
A1000	Tying of Reinforcement Steel	5	19-Apr-21 A	26-Apr-21	100%	100%
A1010	Fixing of Formwork	5	23-Apr-21 A	29-Apr-21	20%	20%
A1020	Pouring of Concrete	2	29-Apr-21	03-May-21	0%	0%
A1030	Removal of Formwork	3	10-May-21	13-May-21	0%	0%
A1040	Finishing works	5	13-May-21	20-May-21	0%	0%

Activity: A1010 Fixing of Formwork

Duration
Original: 5
Actual: 1
Remaining: 4

Status
☑ Started: 23-Apr-21
☐ Finished: 29-Apr-21
Exp Finish:
Physical %: 20%
Suspend:
Resume:

Figure 1.4: Screenshot showing progress percentage requirement for the schedule update

regularly update the schedule to ensure that the project stays on track and meets its goals within the specified timeframe. Additionally, the update reflects any unforeseen delays or challenges during the construction process. Accurate and up-to-date scheduling information is essential to keep all stakeholders informed, maintain project momentum, and help ensure a successful outcome. Conventionally, project schedules are updated manually, with on-site progress reported through paper-based daily progress reports. With the advancement of visual data analytics techniques, estimating ongoing construction progress from visual data such as images and videos has become very effective for automated construction progress monitoring [68]. Estimated progress is updated in the project schedule with the help of an updated 4D BIM [88]. Although this process is more efficient than conventional schedule updates, the availability of a 4D BIM is necessary. 4D BIM creation requires manually matching BIM elements with construction schedule activities. This process is time-consuming and demands advanced knowledge of construction engineer-

doi:10.6342/NTU202302333

ing. Additionally, the schedule-linked models often become outdated after construction begins. In this situation, project schedules must automatically be linked to the reality models and extract the task-wise physical progress status required for schedule updates. However, no such method is currently available for automatically connecting progress and schedule data.

1.3 Objectives

Digital twin (DT), a high-fidelity digital representation of a physical asset, can monitor, simulate, predict performance, and control the physical system through a bi-directional data-to-information flow throughout its lifecycle [33]. Digital twins are increasingly being used across the industries such as manufacturing [46], aerospace [50], healthcare [90], etc. In construction, DT allows stakeholders to simulate the project's design, construction, and operation, enabling data-driven decision-making, improving the performance and maintenance of the built asset, and monitoring and optimizing the building's systems in real time. According to [81], Digital twin construction (DTC) is a new way of managing design and construction, leveraging the data collected from various site monitoring technologies. DTC uses BIM, sensing technologies, artificially intelligent functions, and lean construction practices to provide accurate status information and proactively analyze and optimize the ongoing design, planning, and production processes. This results in more efficient construction processes and reduces the risk of delays and errors, improving overall productivity. Additionally, DTC technology enables better analysis of construction data and helps to improve the safety, sustainability, and overall performance of buildings and infrastructures, providing greater value to the stakeholders. CPM, an integral part of the construction planning and control process, can be significantly improved through the DTC implementation.

So, this study aims to propose a novel vision-based progress monitoring method based on the DTC concept. The main objective of this study is to propose a methodology for the automatic estimation of activity-wise percentage completion and automatic project schedule update by linking 3D reality models and construction activities. This objective is achieved by solving two important tasks:

1. Progress estimation and visualization
2. Progress update in the project schedule

For solving task 1, an Activity-level progress monitoring system (ALPMS) is developed to report activity-wise completion percentages by taking 4D BIM and site images as input. The system creates an as-built reality-captured model (point cloud) from input images using structure from motion (SfM) and multiview stereo (MVS) algorithm and the model is aligned with the as-planned 4D BIM in the same coordinate system. Then a digital twin of the construction project (DTC) is created by integrating BIM, reality captures, and project schedules to monitor project progress continuously. DTC compares the as-built model with the as-planned BIM through an occupancy-based method to check whether the construction activity has been started for an element. Next, a virtual camera parallel to the BIM element under examination is set up automatically. The camera looks straight at the element's largest surface. An orthographic view of that element's surface is synthesized using projective transformation or the latest computer graphics technique that uses a neural radiance field (NeRF) for novel view synthesis. Parallelly, a deep learning-based semantic segmentation model (Mask RCNN) [39] is trained for activity-level progress reasoning. The synthesized image is then passed through the trained segmentation model,

doi:10.6342/NTU202302333

and activity-wise segmentation masks are created. The progress percentage of a schedule activity is computed by taking the ratio of the mask's area and the appearance of the element's entire face area in the ortho-view image. Finally, the as-planned BIM is color-coded, and the as-built point cloud is semantically segmented for 3D progress visualization by projecting the 2D segmentation masks onto the 3D point cloud. The method is validated in two building construction projects in Taiwan. Four datasets consisting of point clouds created from site images for different scan dates and 4D BIMs are used for validation. The method was tested on four case studies comprised of nearly 500 BIM elements. The method is specifically applicable to building construction projects. There are three significant contributions from this part of the study.

(a) Activity-level progress monitoring framework, i.e., ALPMS: This framework detects activity level progress status and reports completion percentage progress. Other methodological novelties include (a) orthographic view synthesis for accurate progress percentage estimation, (b) deep-learning-based semantic segmentation of images for activity-wise appearance detection, and (c) 3D progress visualization through the semantic segmentation of the as-built point cloud.

(b) Semantic segmentation dataset for activity level progress monitoring: Supervised learning-based deep-learning models require a considerable amount of labeled data for training. This paper introduces a dataset for the semantic segmentation of construction images based on major construction activities. The data set can be used for many purposes, including activity-level construction progress monitoring.

(c) Use of Neural Radiance field (NeRF) for novel view synthesis for PM: A realistic orthographic image of the element's surface is essential for inferring activity-wise progress status. This paper introduces a method that uses a NeRF for synthesizing photorealistic orthographic views.

For solving task 2, vision-based progress detection results are integrated with project schedules through Natural Language Processing (NLP). First, semantic information from construction schedules, such as location, object, materials, and task (LOMT) information from each activity, is extracted through similar NLP techniques used in Named Entity Recognition (NER). Parallelly, computer vision algorithms are used to extract semantic information such as location, object, material, and task (LOMT) from reality models. Using common information, project schedules and reality models are linked for construction progress updates. Precisely, given a reality model, it is first positioned in the world coordinate system with the help of 3D BIM (if available) or pre-defined Ground Control Points (GCP). Next, the location information (L) is inferred from BIM or the GCPs. With the availability of 3D BIM, objects (O), i.e., building elements (beams, columns, walls, ceilings, and floors), are detected through occupancy checks. Otherwise, deep learning (DL)-based point cloud segmentation algorithms are employed to detect them. Tasks (T) and materials (M), such as pouring concrete, fixing formwork, etc., are identified through appearance-based segmentation of point clouds or images, and progress percentages of each activity are estimated. LOMTs from schedule activities are then matched with reality models' LOMTs to create a linkage between the schedule and the reality models. Finally, schedule activities are updated with progress percentages derived from the vision-based progress detection results. There are two significant contributions from this part of the study.

(a) Automatic schedule update by linking reality models and construction schedules:

doi:10.6342/NTU202302333

This study proposes a framework for linking progress information excerpted from vision-based methods and schedule activities with or without the availability of a 3D BIM.

(b) Use of Large Language Models (LLM) for schedule information extraction: This study introduces a few-shot learning approach and prompt engineering techniques to extract semantic information for construction schedules needed for automated construction progress monitoring.

1.4 Organization of the thesis

The organization of this thesis is as follows: Chapter 2 reviews previous studies related to vision-based construction progress monitoring and deep learning-based visual data analytics; Section 3 describes the proposed progress estimation and visualization method and its application in real-world construction projects; Section 4 details the Progress update method proposed in this study and its validation through construction case studies; at the end, Section 5 concludes the study with the hints of potential future research.

doi:10.6342/NTU202302333

doi:10.6342/NTU202302333

Chapter 2 Literature Review

Research on automated construction progress monitoring has matured with the advancement and affordable availability of data collection, storage, and processing technologies in the last few decades. Model-assisted progress monitoring with computer vision-based data analytics techniques has received wide acceptance in the construction industry. This section reviews the developments in vision-based construction progress monitoring methods.

2.1 Vision-based construction progress monitoring

Vision-based progress monitoring methods that rely on model-assisted progress reasoning can be divided into two groups based on the level of progress monitoring. 1) methods that can only check the on-site physical completion of the BIM elements, and 2) methods that monitor the progress of the schedule activities associated with the BIM elements.

2.1.1 Element-level Progress Monitoring

In element-level progress monitoring methods, as-built site conditions are compared against as-planned BIMs and the physical presence of the BIM elements at the construction

doi:10.6342/NTU202302333

site is checked. So, these approaches are suitable for monitoring the progress up to the BIM element level, such as column, beam, slab, etc. For detecting the elements in the as-built state, researchers have developed various methods from time to time. Those methods often used time-lapse images and 3D point clouds generated through reality capture techniques such as photogrammetry or laser scanning.

During early research on computer vision-based progress monitoring, [30] detected as-built elements by superimposing a 4D BIM model on time-lapse images.[102] and [98] used image processing techniques and 4D computer-aided design (CAD) based filtering for recognizing elements in construction site images. With the advent of deep learning-based object detection, these image-based element detection methods have improved further [19]. [102][102]) used such techniques for monitoring modular construction progress. DL-based object detection and tracking were used by [93] for monitoring precast wall construction.

Methods that use reality models follow scan-vs-BIM techniques or scan-to-BIM and BIM-to-BIM comparison for object detection [15]. In scan-vs-BIM techniques, as-built point clouds are aligned with as-planned BIM in the same coordinate system, and the planned elements' presence in the as-built scene is detected by geometry-based reasoning. [16] developed a method for detecting 3D CAD elements in laser scan point clouds. [30] and [31] introduced methods for detecting objects in image-based point clouds. An improved object recognition algorithm was proposed by [88], and the same was used for automated earned value management (EVM). To further automate the progress monitoring methods, a machine learning-based voxel coloring and voxel labeling algorithm was introduced by [32]. Improvements in object detection accuracy following BIM element sequencing logic were presented by [18]. Automatic schedule update while progress mon-

doi:10.6342/NTU202302333

itoring was observed in the study by [85]. A progress deviation notification system was proposed by [66]. [51] proposed a visual and virtual production management system for proactive project controls.

Scan-to-BIM techniques first segment the as-built point clouds into meaningful regions, detect geometric features and their relationships, and finally create BIM objects leveraging them. Methods that follow scan-to-BIM techniques in progress monitoring compare as-planned BIM and as-built BIM for progress assessments. [83] developed a method for recognizing and modeling structural elements from the 3D data collected through a stereo vision system. [15] introduced an integrated scan-vs-BIM and scan-to-BIM approach for the progress monitoring of building mechanical, electrical, and plumbing (MEP) components. [74] proposed an exciting method for progress monitoring through real-time 3D scans from the camera fitted on workers' hard hats. A typical scan-to-BIM approach was adopted by [59] and [44] to detect the progress of reinforced concrete elements. More recent studies have included deep-learning-based semantic segmentation of the point clouds in the scan-to-BIM workflow [21]. Deep-learning based 3D object detection methodology was presented by [99]

Beyond image-based or laser scan-based data collection and processing, few recent studies investigated extended reality options for real-time progress monitoring. For example, Kopsida and Brilakis (2020) [45] used a Microsoft HoloLens device to detect the existence of volumetric objects such as beams, columns, and walls in physical space.

Although these element-level progress monitoring methods were convenient in various instances, they were unsuitable for effective schedule control because of activity-level progress inference limitations. Previous studies have discussed these and highlighted the

doi:10.6342/NTU202302333

need for activity-level progress monitoring.

2.1.2 Activity-level Progress Monitoring

The progress status of schedule activities associated with BIM elements is inferred in activity-level progress monitoring methods. Material recognition and classification corresponding to the schedule activities and BIM elements in the as-built scene are essential in the activity-level progress monitoring methods. Initial studies have only focused on material recognition in construction site images. For example, [104] proposed an ML-based concrete region identification method. Similarly,[84] proposed another method for concrete detection by comparing three ML algorithms. Later [27] introduced the construction material library (CML) with 20 typical construction materials and proposed a novel ML-based method for detecting construction materials in 200x200 pixel image patches.

With the maturity of material recognition techniques, researchers have focused on incorporating them in model-assisted progress monitoring workflows. [38] developed an appearance-based progress monitoring method that checked BIM elements' physical appearance from 2D image patches. [38] presented a BIM-integrated construction sequence ontology for improving activity-level progress monitoring. Later [37] introduced two more appearance-based procedures: Texture-based reasoning for image-based 3D point clouds and color-based reasoning for laser-scanned point clouds. Few studies investigated activity-level progress monitoring in indoor construction scenarios. [36] took a conventional computer vision-based approach to detect the activities of an indoor partition wall construction. Another indoor construction progress monitoring method was presented by [47] . BIM-assisted floor tiling activity monitoring was studied by [24]. Researchers also tried to classify common building materials in hyperspectral images collected from

doi:10.6342/NTU202302333

terrestrial laser scanners (TLS) [5], [101].

Recent studies have implemented deep learning-based image classification. [52]. [3] ‘ring column finishing activities and Heating, Ventilation, and Air Conditioning (HVAC) installation activities.

Although existing methods have overcome the challenges of element-level progress monitoring, they still have some inherent limitations. Completion percentages of activities are essential information for effective schedule control. However, prevailing methods can only report the progress status in binary form. In some methods [38], [37], selective image patches of the BIM elements are analyzed for appearance checking. However, those patches may not always represent the appearance of the whole element. In a real construction scenario, multiple activities may take place simultaneously on the same element, and the appearance of multiple materials may be observed in a single image patch. Existing methods have limited usability in such cases. Some earlier approaches have been developed specifically for certain construction activities [36], [24]. Generalizing them for different construction scenarios is difficult. The authors highlighted the need for partial progress monitoring in their previous studies to overcome these challenges [69]

2.1.3 Progress monitoring of partially completed elements and activities

Few past studies tried to estimate and report the progress of partially completed building elements or activities. However, unfortunately, none of them are well suited for activity-level progress monitoring and reporting of partially completed building elements. Early studies by [88, 89], and [85] focused on updating construction schedules by comparing 4D BIM and point clouds. These studies took an occupancy-based approach to recognize and count completed building elements and later computed the progress percentage of a schedule activity by taking a ratio between the number of recognized objects and the expected objects. This method is workable if, in 4D linking, multiple elements are linked to one schedule item. However, it's ineffective in the two cases mentioned in Section 1.2. Later [66] and [76] tried to estimate the partial completion of each BIM element by comparing the number of points and point density in the as-planned point cloud and as-built point cloud. However, this method can get affected by the completeness of the as-built point cloud data. As all these previous methods followed an occupancy-based approach, it faces the limitations of occupancy-based progress monitoring.

A few research focused on detecting task-level completion through a conventional ML-based image processing approach. For example, [36] studied under-construction indoor partitions, [24] studied floor tiling tasks to report progress in percentage. One of the significant drawbacks of these methods is their application is particular to a specific construction task and not easily generalizable for other tasks. Recently, [69] and [97] have tried to implement an image-segmentation-based approach for operation-level progress monitoring. However, the underlying assumption of these studies is element face for which the progress is being detected will always be within a single camera view. In other words, the face appearance can be represented in a single image. That may not be the case for all elements in a construction project.

doi:10.6342/NTU202302333

2.2 Deep-learning-based data analytics in construction management

Although researchers discovered the importance of big data, especially visual data and text data, in construction management applications almost two decades ago, utilization of those data was limited to some specific tasks because of the limitations of data analytics techniques. Widespread applications of big visual or text data in construction management were constrained until the adoption of deep learning [68]. Deep learning (DL) is a subset of machine learning that is largely based on artificial neural networks (ANN) [25]. They use neural networks with multiple layers to progressively extract higher-level features from raw input data [49]. The end-to-end learning approach significantly differs from traditional data analytics methods, which rely heavily on handcrafted features. Extraction and selection of important features for initiating the traditional data analytics methods require expert feature engineers and a long trial-and-error process [65]. Figure 2.1 shows the difference in workflows between traditional methods and DL methods. Deep-learning applications for solving computer vision (CV) tasks and natural language processing (NLP) tasks related to construction management are discussed in the subsequent sections.

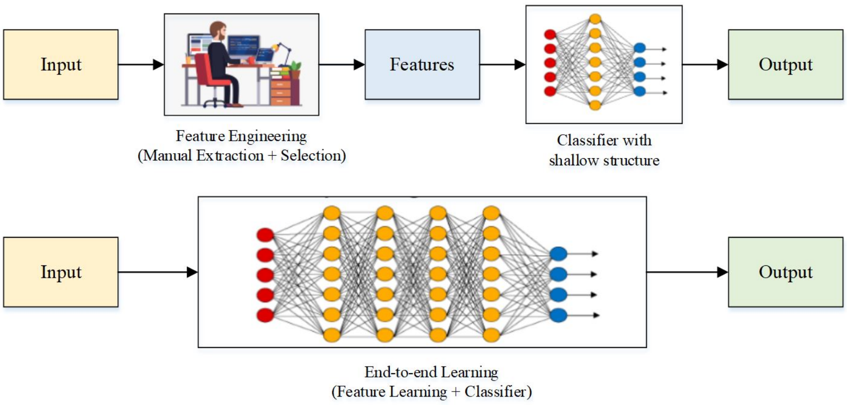


Figure 2.1: Workflows of the traditional methods (top) and DL-based methods (bottom)

doi:10.6342/NTU202302333

2.2.1 Knowledge extraction from visual data

In recent years, researchers and practitioners have extensively used DL for analyzing visual data and inferring valuable information for effective construction management purposes. The visual data collected from construction sites are utilized for various operations-level management purposes, such as monitoring construction safety [29], monitoring equipment and worker performance [53, 79], monitoring the progress of construction activities [19], in situ and post-construction quality assessment [55], construction waste management [96], facilities management [28], and dynamic worksite management [57]. Various CV tasks such as image classification, object detection, object tracking, pose estimation, and activity recognition were solved through DL algorithms as an integral part of these applications. A study by Pal and Hsieh (2021) [68] identified six major fields and 52 subfields of construction management applications through an in-depth review of 142 selected papers. The major application fields included safety management (44%), productivity management (24%), facilities management (19%), progress monitoring (5%), quality management (3%), construction waste management (2%), and other applications (3%). The percentages in parentheses for each application indicate the proportions of the entire collection of papers. Figure 2.2 shows a tree-diagram visualization of those applications.

2.2.2 Information extraction from construction schedules

NLP algorithms help extract information from text data. Construction schedules prepared by experienced construction schedulers are written in natural languages. Applications of NLP in semantic information extraction from construction schedules are found

doi:10.6342/NTU202302333

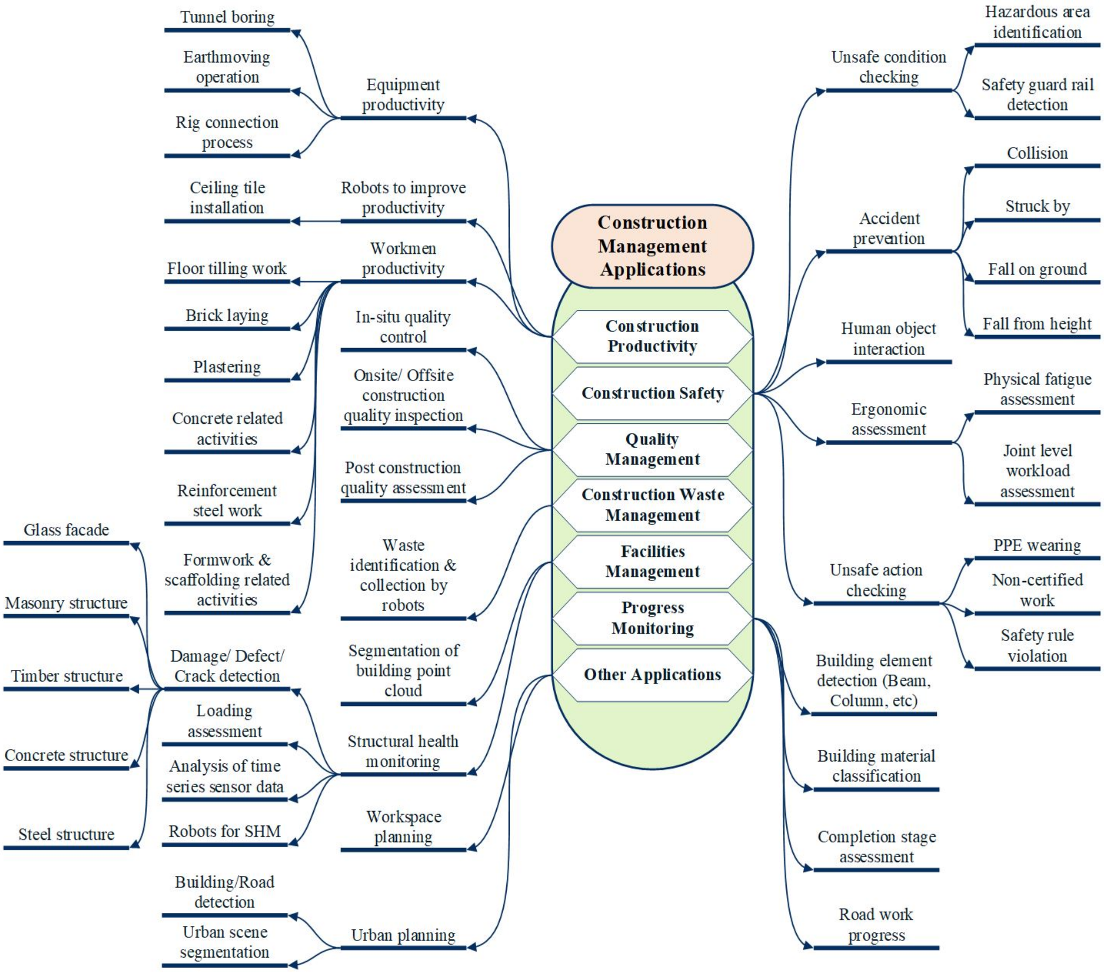

Construction Management Applications
Construction Productivity
Construction Safety
Quality Management
Construction Waste Management
Facilities Management
Progress Monitoring
Other Applications
Equipment productivity
Tunnel boring
Earthmoving operation
Rig connection process
Robots to improve productivity
Ceiling tile installation
Workmen productivity
Floor tilling work
Brick laying
Plastering
Concrete related activities
Reinforcement steel work
Formwork & scaffolding related activities
In-situ quality control
Onsite/ Offsite construction quality inspection
Post construction quality assessment
Waste identification & collection by robots
Segmentation of building point cloud
Structural health monitoring
Damage/ Defect/ Crack detection
Loading assessment
Analysis of time series sensor data
Robots for SHM
Glass facade
Masonry structure
Timber structure
Concrete structure
Steel structure
Workspace planning
Urban planning
Building/Road detection
Urban scene segmentation
Unsafe condition checking
Hazardous area identification
Safety guard rail detection
Accident prevention
Collision
Struck by
Fall on ground
Fall from height
Human object interaction
Ergonomic assessment
Physical fatigue assessment
Joint level workload assessment
Unsafe action checking
PPE wearing
Non-certified work
Safety rule violation
Building element detection (Beam, Column, etc)
Building material classification
Completion stage assessment
Road work progress

Figure 2.2: DL-based visual data analytics applications in construction management

doi:10.6342/NTU202302333

promising. For example, Zhao et al. (2019) [103] tried to extract construction knowledge from project schedules through parts of speech (POS) tagging and named entity recognition (NER). Amer and Golparvar-Fard (2019) [6] proposed Part-of-Activity (POA) tagging to identify action, object, location, and responsible party from the activity descriptions. Information retrieved from project schedules was further utilized for linking long-term and short-term project schedules using transformer machine learning models [7]. Logical linking between activities is essential for schedule quality control. Amer et al. (2022) [8] used the bidirectional long short-term memory (LSTM) model to learn pairwise activity sequences. Hong et al. (2023) [40] used a Graph Convolutional Network (GCN) to identify and understand different activity types and sequences. To achieve this, they used various input features such as numerical data, text descriptions of activities and WBS, graphical features, and logic links between the activities. Although schedule information extraction through NLP has started gaining importance, very few studies have tried to use identified schedule information for automated construction monitoring [71, 78].

2.3 Summary

The literature review focuses on vision-based construction progress monitoring and deep-learning-based data analytics in construction management. Vision-based progress monitoring methods are categorized into element-level monitoring and activity-level monitoring. Element-level methods focus on comparing as-built site conditions with as-planned BIM models to detect the presence of BIM elements. Activity-level monitoring involves inferring the progress of schedule activities associated with BIM elements. For activity-level monitoring, material recognition in construction site images has been a key area of research, and it has been integrated into model-assisted progress monitoring workflows.

However, existing methods have limitations in reporting partial progress and are often specific to certain construction tasks.

Deep learning has revolutionized data analytics in construction management, particularly in analyzing visual and text data and extracting valuable information. Applications include construction safety monitoring, equipment and worker performance monitoring, activity progress monitoring, quality assessment, waste management, and facilities management. Various computer vision tasks, such as image classification, object detection, tracking, pose estimation, and activity recognition, have been successfully addressed using deep learning algorithms. Similarly, deep learning is also applied to natural language processing (NLP) tasks. In construction, an exciting application of DL-based NLP is observed in information extraction from construction schedules, such as semantic information and activity sequencing. However, few studies have utilized this extracted schedule information for automated construction monitoring.

Overall, the review highlights the advancements in vision-based progress monitoring and the transformative impact of deep learning in construction management data analytics. However, there is still a need for further research to address the limitations in partial progress reporting and generalize activity-level monitoring methods to different construction scenarios. The utilization of NLP-extracted schedule information for automated monitoring also holds promise for future developments.

doi:10.6342/NTU202302333

Chapter 3 Progress estimation and visualization

3.1 Methodology: Overview of the activity-level progress monitoring system (ALPMS)

Based on the task involved, the proposed ALPMS is divided into eight modules: (1) input, (2) 3D reconstruction, camera pose estimation, and BIM+point cloud registration, (3) detection of elements under construction, (4) orthographic view synthesis, (5) dataset preparation for semantic segmentation, (6) progress status detection, (7) completion percentage estimation, and (8) progress visualization. The methodology adopted in each phase is described in detail in this section. The overview of the ALPMS is shown in Figure 3.1.

3.1.1 Input

The ALPMS takes 4D production-level BIM and construction site images as input. The 4D production level BIM is used to know the as-planned project status at any given time. It is created by linking schedule activities with 3D model elements. For a project, 4D

doi:10.6342/NTU202302333

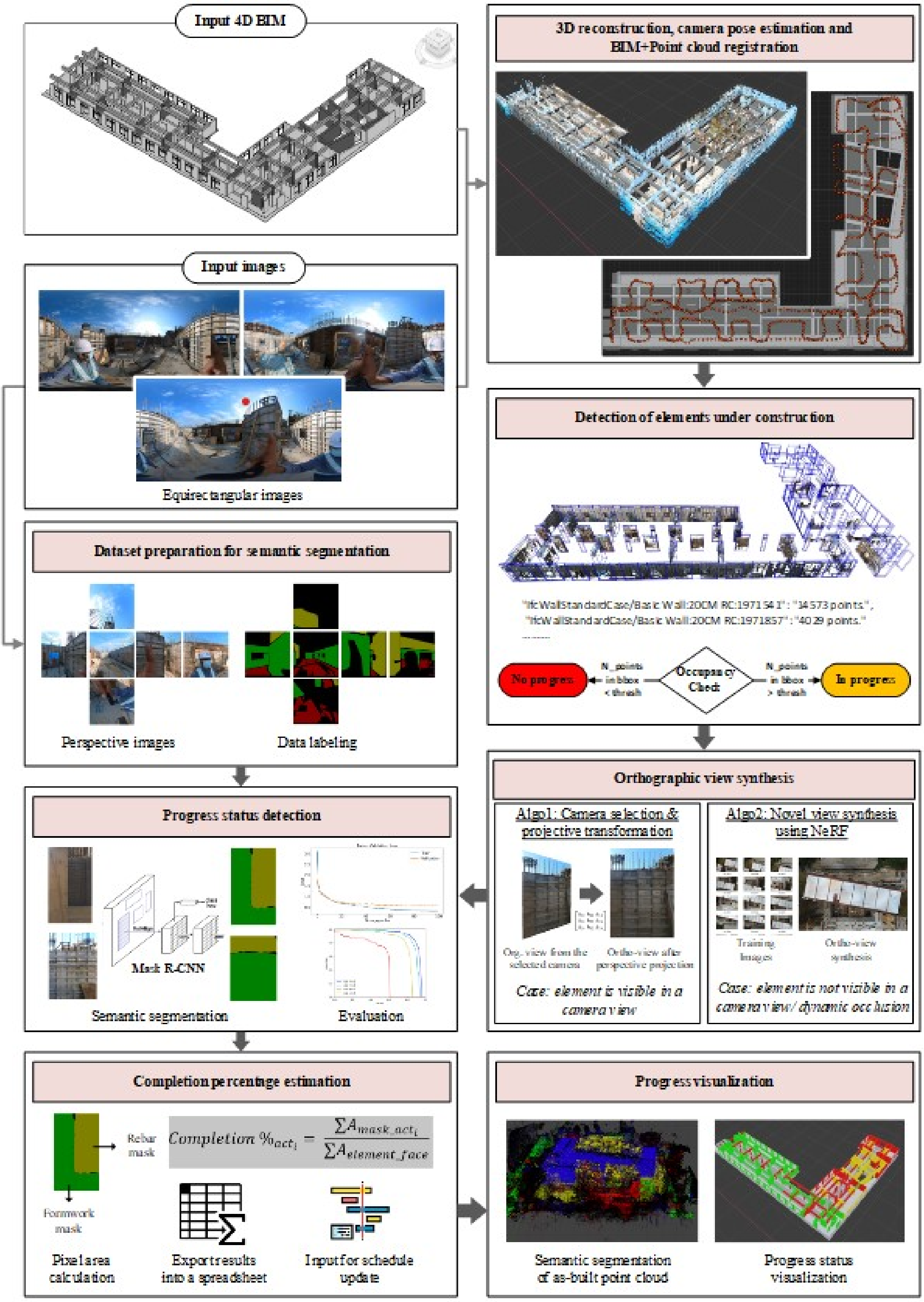

Input 4D BIM
Input images
Equirectangular images
Dataset preparation for semantic segmentation
Perspective images
Data labeling
Progress status detection
Mask R-CNN
Semantic segmentation
Evaluation
Completion percentage estimation
Rebar mask
$Completion\ \%_{act_i} = \frac{\sum A_{mask\_act_i}}{\sum A_{element\_face}}$
Formwork mask
Pixel area calculation
Export results into a spreadsheet
Input for schedule update
3D reconstruction, camera pose estimation and BIM+Point cloud registration
Detection of elements under construction
"IfcWallStandardCase/Basic Wall:20CM RC:1971541" : "14573 points.",
"IfcWallStandardCase/Basic Wall:20CM RC:1971857" : "4029 points."
No progress
N_points in bbox < thresh
Occupancy Check
N_points in bbox > thresh
In progress
Orthographic view synthesis
Algo1: Camera selection & projective transformation
Org. view from the selected camera
Ortho-view after perspective projection
Case: element is visible in a camera view
Algo2: Novel view synthesis using NeRF
Training images
Ortho-view synthesis
Case: element is not visible in a camera view/ dynamic occlusion
Progress visualization
Semantic segmentation of as-built point cloud
Progress status visualization

Figure 3.1: Overview of the ALPMS

BIM creation is generally a one-time effort and is done during the project's initial phase. Project schedules prepared in commercial project scheduling software are exported in the supported file format of the BIM coordination tool, and the coordination tool links the schedule activities and the 3D model elements by querying their task ids and element ids, respectively. Based on the model element's LoD and the project schedule's WBS level, linkages can be of three types, as shown in Table 3.1. In a 4D BIM, the activities associated with each element can be represented through a Gantt chart. As-planned model for a specific date is derived by filtering the model elements by their start or finish dates.

Table 3.1: Linkage scenarios for various LoD and WBS level

Sl. No.	Linkage type	Situation
1.	One-to-one linkage between the activity and the element	LoD of the element and WBS level matches with each other
2.	Multiple activities linked to one model element	High WBS level but low LoD of element
3.	Multiple elements linked to one activity	Low WBS level

The ALPMS can take construction site images or videos captured by any device, including smartphone cameras, commercial digital cameras, drones, cameras with spherical 360-degree lenses, or fisheye lenses as input. Given a video input, the system detects the keyframes in the one-second interval and extracts them as image sequences. The 360-degree spherical images are first converted into equirectangular photos through the internal image processing pipeline. Further, each equirectangular image is mapped as a 3D cube, and six perspective images of top, bottom, right, left, front and back faces are extracted. Similarly, fisheye camera images are also preprocessed to remove the distortion before using them in the next module. Commonly 360-degree cameras and fisheye cameras are used for interior construction monitoring for their higher field of view (FOV); alternatively, drones are used for capturing exterior construction data.

3.1.2 3D reconstruction, camera pose estimation, and BIM+point cloud registration

This module uses an SfM-MVS-based 3D reconstruction pipeline to create a reality model. The SfM algorithm uses overlapping images from different viewpoints of the same construction site as input. It outputs a sparse as-built point cloud model of the construction site and the intrinsic (focal length, optical center, image sensor format) and extrinsic (rotation and translation) camera parameters for all input images. The SfM pipeline detects and matches features in the corresponding images and uses image registration, triangulation, and bundle adjustment to incrementally reconstruct the sparse 3D scene [1]. Later MVS pipeline computes depth maps and normal maps from the SfM output and produces a dense point cloud of the 3D scene by fusing them from multiple images [82]. Figure 3.2 shows the typical 3D reconstruction pipeline.

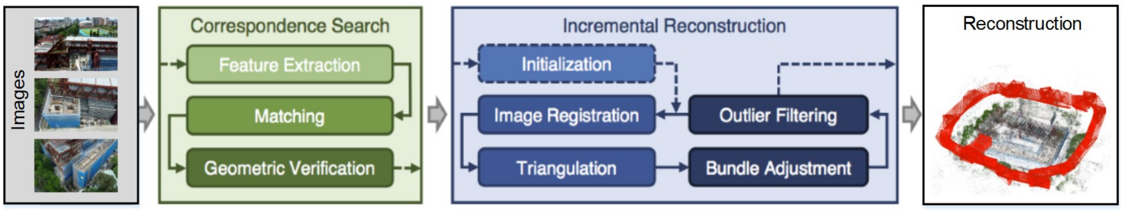


Figure 3.2: A typical 3D reconstruction pipeline (Schönberger and Frahm 2016)

The reality model and camera poses generated from the 3D reconstruction are up to scale and generally located in an arbitrary 3D coordinate system. First, the as-planned 4D BIM in industry foundation class (IFC 2x3) format and the as-built point cloud model with corresponding camera poses are imported into an open-source 3D computer graphics software. A two-step registration process is followed to align them in the same coordinate system. In the coarse registration step, three or more corresponding points (vertices) are selected manually between the as-planned BIM and the as-built point cloud. The point cloud is then roughly scaled and transformed into BIM coordinate system by solving the seven degrees of freedom (DOF) similarity transformation. The iterative closest point (ICP) algorithm [61] is used in the fine registration step to register the BIM and point cloud closely. Before fine registration, the BIM was converted into an equivalent point cloud by sampling points around its surfaces. The point cloud was cleaned up by applying a statistical outlier removal. In statistical outlier removal, the average distance d_i from point P_i to its k nearest neighbors in the 3D point cloud is computed. The point is eliminated if the average distance d to its k neighbors exceeds the threshold T. The threshold is calculated using Equation 1, where α is a standard deviation multiplier, μ_d and σ_d are the mean and standard deviation of distances d_i.

$$T = \mu_d + \alpha * \sigma_d \tag{1}$$

At the end of the registration process, a 4x4 fine transformation matrix, a combination of rotation, scale, and translation, is obtained. Initial camera positions are then transformed into the BIM coordinate by applying the fine transformation matrix.

3.1.3 Detection of elements under construction

This module checks whether construction has started for a BIM element through a simple occupancy-based method. Finely registered BIM and point cloud models are fed into algorithm 1 to get the element e wise construction status and the number of points present in the element's bounding box n_{pc^e}. If n_{pc^e} exceeds a threshold number n per square meter surface area, the element is considered under construction and labeled as "in progress"; otherwise labeled as "no progress.", The value of n is set experimentally.

To accommodate the registration error and the presence of the temporary structure such as formwork around the element, the element's bounding box dimensions are increased by a threshold value ϵreg. Points inside the elements' bounding boxes are retained and counted. Finally, a filtered point cloud within the BIM's boundary is obtained, as shown in (the example in the experiment section). This filtering removes the unnecessary points beyond the region of interest and enhances the system's efficiency. Elements marked as "in progress" are only passed to the subsequent modules for checking the activity level progress status.

```
Algorithm 1: Detect elements under construction
Input:  A BIM and a point cloud $pc$
Output: number of points inside a BIM element's bbox $n_{pc^e}$
        Construction status of the element
foreach $e$ in BIM do
    $co \leftarrow$ coordinates the bounding box $(B)$
    Compute cog of $B$: $1/8 \sum co$
    Compute extent of $B$ : dimensions of $e$
    Threshold: registration error $(\epsilon_{reg})$
    $B$ with threshold $(B')$: dimensions of $e_i + \epsilon_{reg}$
    Point cloud inside $B'$: $pc^e = pc < B'$
    Count points inside $B'$: $n_{pc^e}$
    if $n_{pc^e} >$ n $points/m^2$ then
        $e \leftarrow$ "in progress"
        return $pc^e$
        Select $e$ for activity-level progress detection
    else
        $e \leftarrow$ : "no progress"
    end
end
```

3.1.4 Orthographic view synthesis

In practice, activity level progress status is determined by looking at the appearance of the elements' surfaces. Viewing elements' surfaces from an orthographic position is necessary for accurate progress percentage measurement [97]. Oblique views may often cause measurement errors [47]. However, it is impractical and often impossible to collect orthographic views of all surfaces of all the elements present in the as-planned BIM during reality capture. So, in this study, we propose two approaches for synthesizing orthographic views of element faces. These two approaches address two scenarios that may occur while using ALPMS, as shown in Table 3.2.

Table 3.2: Scenarios and proposed approaches for orthographic view synthesis

Scenario.	Details of the scenarios	Proposed approach
1.	Element's face is entirely visible in at least one camera view without dynamic occlusion.	Projective transformation of a selected camera view
2.	Element's face is not entirely visible in one camera view because a. reality capture path passes very close to that face; b. the face area is too big to be accommodated in a single camera view; c. the face is occluded by dynamic occlusions in k camera views.	Novel orthographic view synthesis using Neural radiance field (NeRF)

3.1.4.1 Projective transformation of a selected camera view

In this approach, the system checks whether the face f of an element e in the BIM is entirely visible from any camera c that belongs to the reality capture C. Algorithm 2 helps to find the best camera θ_f that captures the element's face entirely within their FOV. For this purpose, first, each camera cast a ray towards the center of the face O_f, cameras whose ray hits the target are considered for the next check. Next, vertices V_f of face f are back-projected from the world coordinate system (WCS) to the camera's pixel coordinates (PCS) using Equation 2.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K^{\theta}[R^{\theta}T^{\theta}] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} [R^{\theta}T^{\theta}] = \begin{bmatrix} R_{3*3} & T_{3*1} \\ 0_{3*3} & 1 \end{bmatrix} \tag{2}$$

doi:10.6342/NTU202302333

This equation transforms any point (X, Y, Z) in WCS to its equivalent pixel coordinate (u, v) by applying an 11 DOF camera projection matrix $K^\theta[R^\theta T^\theta]$.

Algorithm 2: Camera selection

```
Input: Elements $e \in BIM$ ; Cameras $c \in C$
Output: $\theta^f$ : Selected camera corresponding to each face of each element in
        BIM
foreach element $e$ in BIM do
    foreach face $f$ in $e$ do
        $O^f \leftarrow$ center of $f$
        $V^f \leftarrow$ vertices of $f$
        $c^f$ : list of selected cameras corresponding to $f$
        foreach camera $c$ in $C$ do
            $\eta \leftarrow 0$ : number of vertices within camera view
            Cast a ray from $c$ to $O^f$
            if ray hits $O^f$ then
                /* Check if the vertex is within camera FOV */
                foreach vertex $v$ in $V^f$ do
                    $v(u, v) \leftarrow$ back project $v$ to image plane of $c$
                    if $v(u, v) < c(u, v)$ then
                        $\eta + +$
                    end
                end
                if $\eta =$ count of $V^f$ then
                    $c^f \leftarrow +c$ // append c in $c^f$
                    $area(f)_c \leftarrow$ Compute area of $f$ in $c$ camera view
                else
                    Apply NeRF-based solution
                end
            end
        end
        /* Choose top k cameras and sort based on $area(f)_c$ */
        foreach selected camera $c_k$ in $c^f$ do
            Check dynamic occlusion
            if $f$ is not occluded in $c_k$ view then
                $area(f)_{c_k} \leftarrow$ Compute area of $f$ in $c_k$ camera view
            end
            $\theta^f \leftarrow$ arg max all $area(f)_{c_k}$
            Apply projective transformation
        end
        return $\theta^f$
    end
end
```

Here 3 DOF rotation R^θ and 3 DOF translation T^θ are represented by an extrinsic camera matrix $[R^{\theta}T^{\theta}]$, and the 5 DOF intrinsic camera matrix is defined as K^{θ} [86]. The back-projection process is illustrated graphically in Figure 3.3.

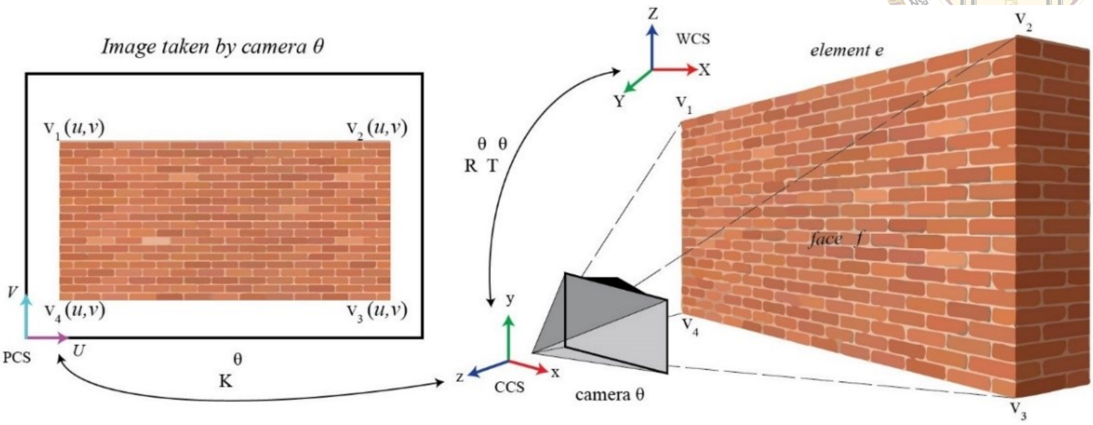


Figure 3.3: 3D object's back projection to the pixel coordinates of a camera

The real image taken by the camera is checked for dynamic occlusion if all the back-projected vertices $\forall v(u,v)$ are seen within the image boundary $c(u,v)$. Here dynamic occlusion refers to the occlusion caused by moving objects on the construction site, such as workers and machines. It is assumed that the presence of this type of occlusion will differ from camera to camera because of their moving nature. The reality model's depth map and as-planned BIM's depth map are compared for the dynamic occlusion checking. Region of interest (ROI) areas are cropped from the depth maps for pixel-by-pixel depth analysis. ROI is the image area covered by the back-projected vertices in PCS. In case of difference between depths exceeds a threshold, the respective camera image is considered occluded; hence the camera is rejected. The face area in each camera of k number of top selected cameras is computed. The camera θf with maximum area coverage is chosen for projective transformation. In case no suitable camera is found, the system prompts for NeRF based solution.

A virtual camera Θ is added to the 3D scene for orthographic view synthesis through projective transformation. Algorithm 3 transforms the image taken from the selected camera θ_f into an ROI cropped ortho view Π_f of face f. Virtual camera is placed programmatically at the center of the face f at a distance of d_f in the normal direction of f to look at the face orthographically. The camera-to-face distance d_f is computed to fit the diagonal length of the face into the vertical sensor height of the camera. Next, the face vertices V_f are back-projected to the pixel coordinates of camera θ_f and θ, and an 8 DOF homography matrix H_f is computed by mapping the corresponding vertex coordinates in PCS. Finally, the actual image taken by θ_f is transformed into an orthographic view by applying the homography matrix H_f and the area within the ROI is cropped to get Π_f. The projective transformation mechanism can be seen in Figure 3.4.

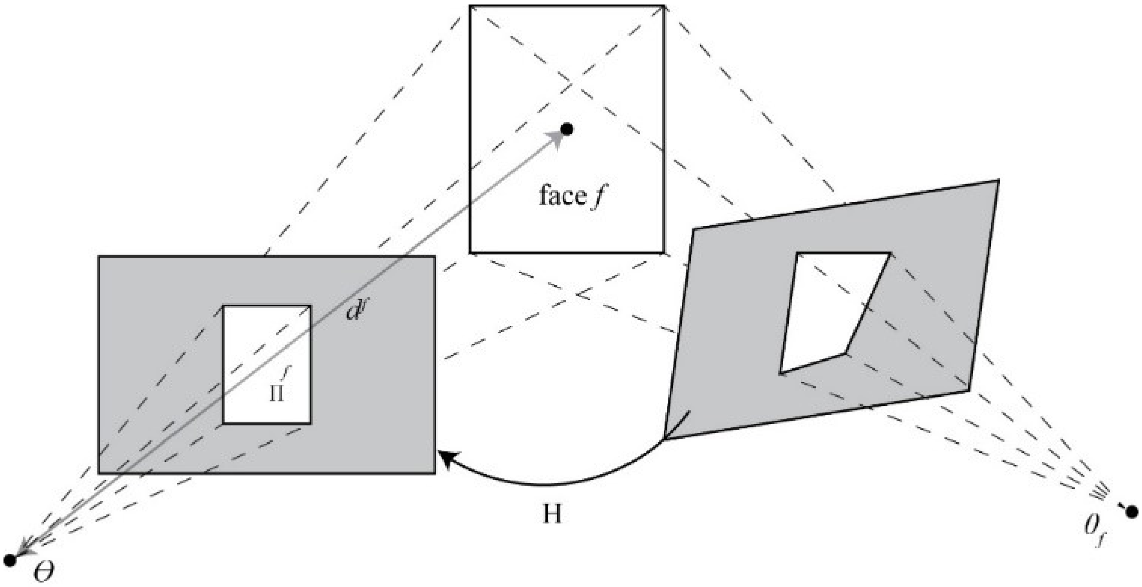


Figure 3.4: Projective transformation of an image

3.1.4.2 Novel orthographic view synthesis using Neural radiance field (NeRF)

The ALPMS adopts a NeRF-based approach when the above process fails to find a suitable camera in the reality capture plan for projective transformation. This situation is evident when the reality capture path passes very close to the element's face, if the face is too large to be accommodated in a single camera view in the reality capture, or if the face

```
Algorithm 3: Orthoview synthesis by perspective projection
  Input:  Face $f \in$ element $e$
          Selected camera $\theta^f$
          Virtual camera $\Theta$
  Output: $\Pi^f$ : ROI cropped orthoview of face $f$
  foreach face $f$ in $e$ do
      $O^f \leftarrow$ center of $f$
      $V^f \leftarrow$ vertices of $f$
      $\hat{N}^f \leftarrow$ Normal of $f$
      /* place $\Theta$ at the center of $f$ at a distance $d^f$ */
      $d^f = \frac{D^f \times F^\Theta}{h^\Theta}$ // $D^f$ : diagonal length of $f$; $F^\Theta$ & $h^\Theta$ : focal
        length and sensor height of $\Theta$
      Move $\Theta$ to $O^f$ to $d^f \times \hat{N}^f$
      $V^f_\Theta(u, v) \leftarrow$ back-projected $V^f$ on image plane of $\Theta$
      $V^f_\theta(u, v) \leftarrow$ back-projected $V^f$ on image plane of $\theta^f$
      homography matrix $H^f \leftarrow$ Projective transformation between $V^f_\Theta(u, v)$ and
        $V^f_\theta(u, v)$
      Orthoview image $\pi^f \leftarrow H^f$. image taken from $\theta^f$ // Transform the
        image taken from $\theta^f$ by applying $H^f$
      $\Pi^f \leftarrow$ Crop $\pi^f$ inside the boundary of $V^f_\Theta(u, v)$
      return $\Pi^f$
  end
```

is occluded in a k number of selected camera images.

NeRF is a fully-connected neural network that can generate novel views of complex 3D scenes given a sparse set of 2D input images. A novel view indicates a photorealistic rendered image of the 3D scene generated from any arbitrary camera location. In NeRF method, a 3D scene is represented as a 5D vector-valued function. It takes 3D spatial location (x, y, z) and 2D viewing direction (θ, ϕ) as input and outputs the emitted color (r, g, b) and volume density (α) while passing through a multilayer perceptron (MLP). The MLP comprises nine fully connected (FC) layers of 256 channels. A non-linear ReLU activation layer connects two consecutive FC layers. The network architecture of a NeRF model is shown in Figure 3.5.

View synthesis from a given camera follows four steps to generate a neural radiance

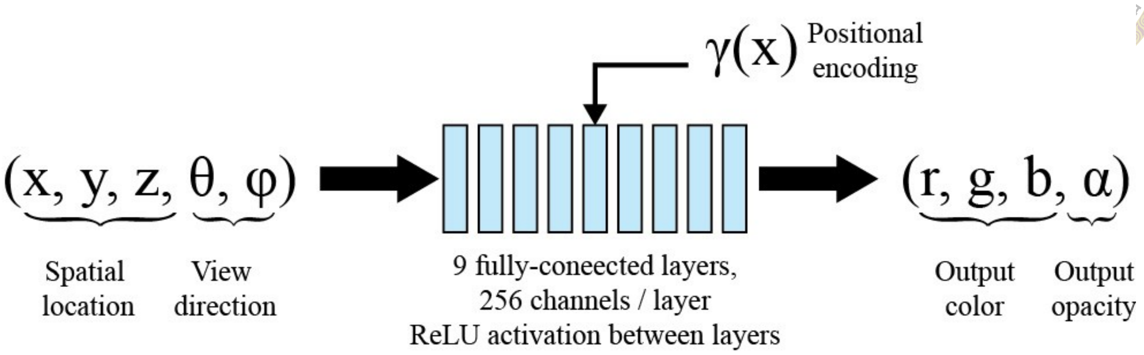


Figure 3.5: Network Architecture of the NeRF model [62]

field, as shown in Figure 3.6. (a) A ray from the camera marched through the pixel's position and the 3D scene to generate a sampled set of 3D points. (b)Those points' location and viewing directions are inputted to the MLP to produce colors and density corresponding to those points. (c) A classical volume rendering technique is adopted to accumulate the color and densities into a 2D image. (d) Finally, the scene representations are optimized by minimizing the loss between the synthetically generated image and the original image. Further, the quality of the output image is improved by applying positional encoding $\gamma(x)$ and hierarchical volume sampling [62]. Although initial NeRF implementation [62] was very costly to train and evaluate, a recent one [63] has reduced the time significantly from 1-2 days to a few minutes by introducing multiresolution hash encoding. NeRF implementation in this study follows the method suggested by [63]. Finally, The NeRF model's performance is evaluated by image quality assessment metrics named peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM). Higher PSNR and SSIM values indicate better quality of image synthesis.

For training a NeRF model, a collection of input images and their corresponding camera intrinsic and extrinsic parameters are needed. Once the face f of an element e_B

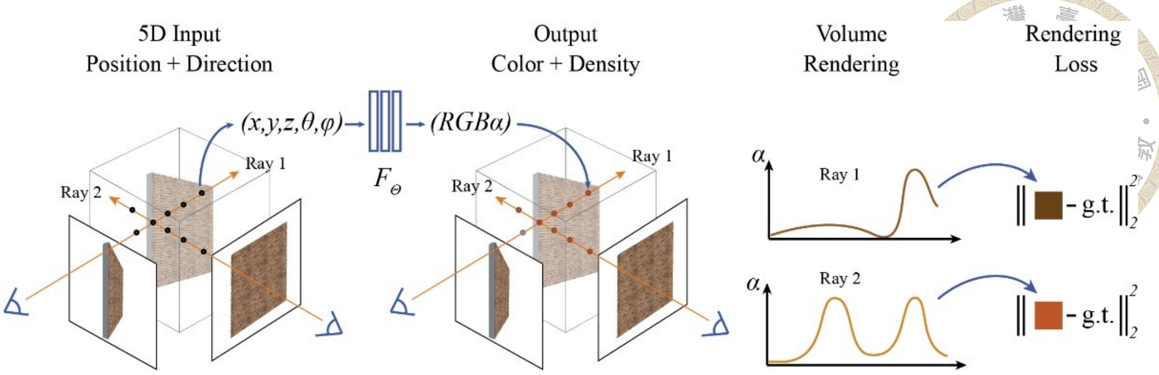


Figure 3.6: Overview of the NeRF-based view synthesis process [62]

that requires a NeRF-based solution is identified, algorithm 4 finds cameras that capture f partially or fully in their FOV. Each camera in the reality capture casts a ray towards their respective view direction; if the ray hits the target face, the camera is selected and included in the NeRF training camera set. A NeRF model M^f is trained using the real images taken by those selected cameras and their camera parameters. A collection of orthographic views π^f_{NeRF} along the face f is synthesized from the trained NeRF model M^f through volumetric rendering. The virtual camera θ is programmatically placed at a distance of 1.5 m from the face's lower left vertex to look at the face orthographically. The camera-to-face distance of 1.5 m is determined experimentally. The camera moves horizontally and vertically, generating $(n^f_v * n^f_h)$ number of orthographic views π^f_{NeRF}. Algorithm 4 computes the camera's step size in both directions considering a 30% overlap between consecutive views. A case of a large element whose face cannot be fitted entirely in a camera view is illustrated in Figure 3.7. The cameras in orange color represent the cameras being used for NeRF training. And the camera in the lower left corner is the ortho view camera that traverses along the face. The rendered ortho views π^f_{NeRF} may also contain images of objects in the 3D scene behind the target face. Before progress status detection, the target face region is cropped out from these views through binary ROI masks generated from the as-planned BIM.

doi:10.6342/NTU202302333

Algorithm 4: Orthoview synthesis by NeRF

```
Input: Face $f \in$ element $e_B$ that requires NeRF-based solution
       Virtual camera $\Theta$
       Cameras $c \in C$
Output: $\pi^f_{NeRF}$ : A set of orthoview images of face $f$
foreach face $f$ in $e_B$ do
    $c^f$ : list of selected cameras corresponding to $f$
    foreach camera $c$ in $C$ do
        Cast a ray from $c$ towards its view direction
        if ray hits $f$ then
            $c^f \leftarrow +c$ // append c in selected camera list
            get camera pose of $c$
        end
    end
    Train a NeRF Model $M^f$ using images and poses of $c^f$
    /* Move $\Theta$ along $f$ and synthesize a set of orthographic
       views                                                        */
    $Lv^f \leftarrow$ lower left vertex of $f$
    $\hat{N^f} \leftarrow$ Normal of $f$
    $d^f \leftarrow$ average dist. of training cams // distance between $f$ and $\Theta$
    base $\leftarrow$ Move $\Theta$ to $Lv^f$ to $d^f \times \hat{N^f}$ // place $\Theta$ at lower left
     corner at a distance $d^f$ m from $f$
    Camera coverage $CC^\Theta = \frac{d^f \times h^\Theta}{F^\Theta}$ // $h^\Theta$ & $F^\Theta$ : sensor height and
     focal length of $\Theta$
    $Overlap \leftarrow$ overlap between two consecutive images
    $Step \leftarrow 1 - Overlap$
    /* Compute number of frames in horizontal $n^f_h$ and vertical
       $n^f_v$ direction                                              */
    $n^f_h = L^f / Step + 1$ ; $n^f_v = H^f / Step + 1$ // $L^f$ & $H^f$ : length and
    height of $f$
    while $i < n^f_v$ do
        while $j < n^f_h$ do
            Synthesize orthographic view image $F^f_{ij}$ from NeRF model $M^f$
            $\pi^f_{NeRF} \leftarrow +F^f_{ij}$ // append $F^f_{ij}$ in $\pi^f_{NeRF}$
            Move $\Theta$ one $Step$ in horizontal direction
        end
        base + Move $\Theta$  $i$  $Step$ in vertical direction
    end
end
```

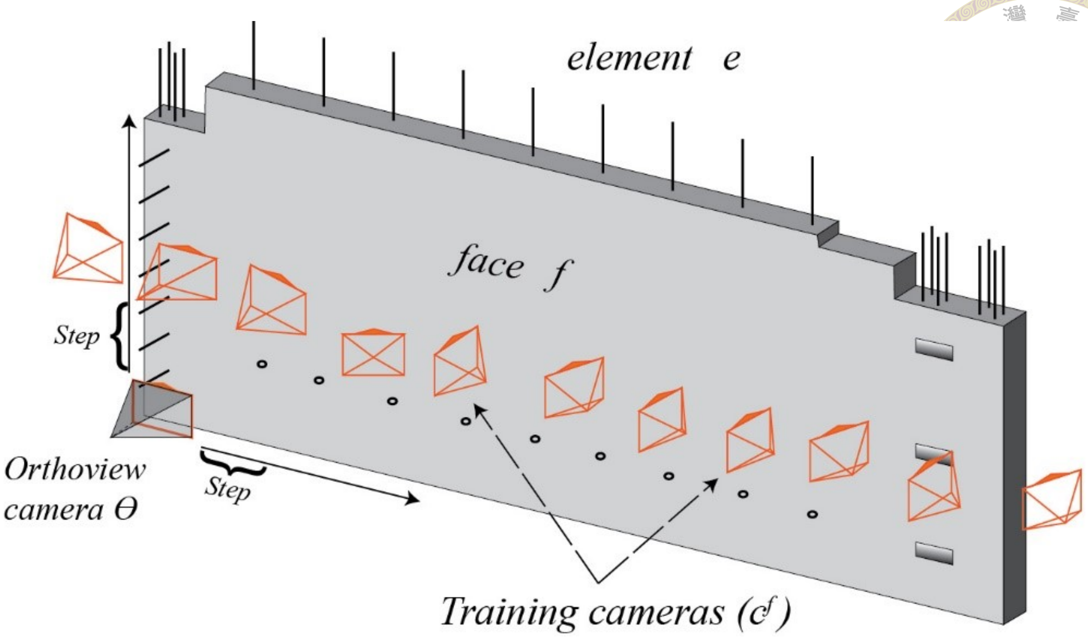


Figure 3.7: Example of camera setup for NeRF-based ortho view synthesis

3.1.5 Dataset preparation for semantic segmentation

For detecting the progress status from the ortho-view images of an element's face, ALPMS leverages the semantic image segmentation method. This method clusters image pixels that belong to the same class. Achieving this task through supervised learning requires a set of labeled training images. In such a labeling process, parts of an image that belong to a class are masked with a polygon. In this case, these classes are construction materials that represent a specific construction activity. So far, no dataset has been publicly available for construction activity-based semantic segmentation. So, during this study, we have prepared a construction image segmentation dataset by labeling construction activities with polygon masks. Open source application LabelMe [42] is used for image annotation with polygons. The dataset contains 2,458 images captured from multiple building construction sites through UAVs, 360-degree-cameras, and fisheye cameras. Six perspective images extracted from each 360-degree camera image are annotated, as shown in Figure 3.8. Fisheye camera images are undistorted first and then annotated

doi:10.6342/NTU202302333

for dataset preparation. These images are annotated with six construction material classes representing six different construction activities. Four of these activities, such as concrete, formwork (wooded and aluminum), reinforcement steel, and steel structure, are related to structural work. Two other activities, wall and floor tiles are related to finishing work. Sample annotations are shown in Figure 3.8.

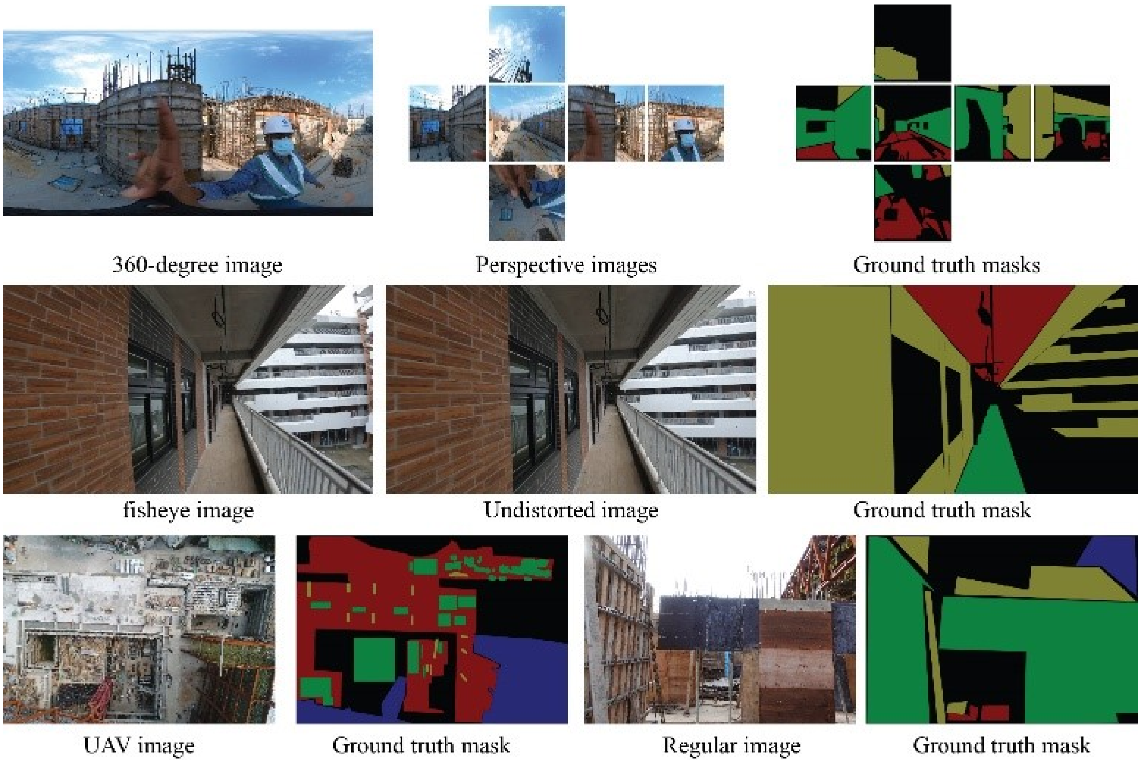

360-degree image
Perspective images
Ground truth masks
fisheye image
Undistorted image
Ground truth mask
UAV image
Ground truth mask
Regular image
Ground truth mask

Figure 3.8: Sample annotations of dataset images

3.1.6 Progress status detection

For progress status detection in ortho view images, ALPMS uses a mask region-based convolutional neural network (Mask R-CNN). Mask R-CNN is chosen because of its high accuracy and efficiency than other image segmentation models [97]. The Mask R-CNN model consists of 5 parts: a backbone, a region proposal network (RPN), a region of interest alignment layer (ROI Align), a class prediction and bounding box regression module, and a mask generation module. The network architecture is shown in Figure 3.9.

The backbone is the primary feature extractor. In this study, ResNet101 is chosen as the backbone. Once an image is inputted, it passes through the different layers of ResNet101 and turns into a feature map. RPN proposes the regions where a class object may exist in the feature map. The ROI Align layer extracts feature vectors from the RPN proposed feature map regions and turns them into a fixed-size tensor for further processing. The class prediction and bounding box regression module is a fully connected (FC) layer that predicts the class ID and the bounding box from the ROI-aligned feature vector. The mask generation module consists of a series of transpose convolutional and convolutional layers. This module generates predicted binary masks for each class detected in the class prediction module.

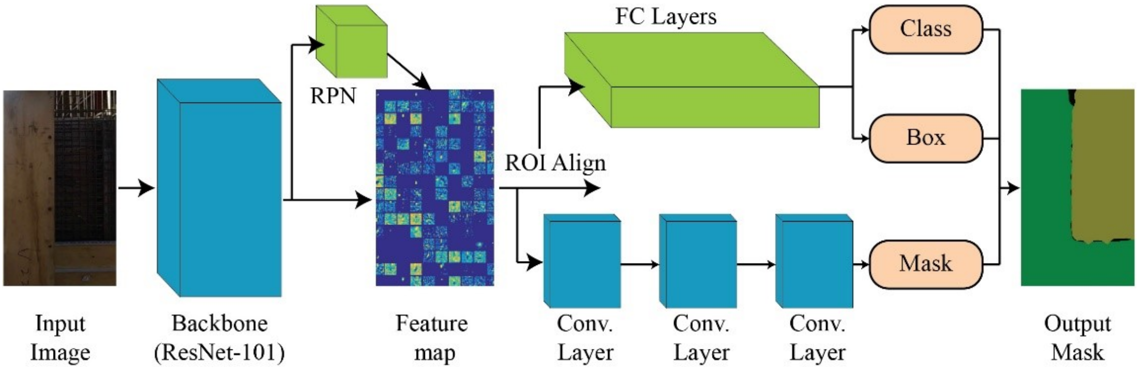


Figure 3.9: Network architecture of Mask-R-CNN

The transfer learning method is adopted to finetune and train a pre-trained Mask R-CNN model for detecting five activity classes and a background class. The finetuning process changes different hyperparameters, such as backbone architecture, optimizers, learning rates (LR), LR schedules, bach size, and epoch size. The dataset is divided into training, validation, and testing with a 70:20:10 ratio. A set of evaluation metrics evaluates the semantic image segmentation model's performance on the test dataset. These metrics are mean intersection over union (mIoU), mean average precision (mAP), and precision-recall curve (P-R curves). One may refer to [68] for further details on DL model training

doi:10.6342/NTU202302333

and evaluation. Ortho view images synthesized from the previous module are fed into the trained Mask R-CNN model to extract color-coded masks representing different construction activities in the image.

3.1.7 Completion percentage estimation

A rule based-approach is adopted to estimate the activity-level completion percentage. The progress status detection module generates color-coded masks for each activity appearing on the element's face. This module finds the latest ongoing activity on the face through precedence relationships. Finally, the completion percentage δeact of the activity is estimated by Equation 3.

$$\delta_{act}^{e} = \frac{\sum Area(mask)^{act}}{Area(face)^{e}} \times 100\% \tag{3}$$

This equation calculates the percentage of the element's face covered by the total area of masks corresponding to that activity by taking a ratio between the summation of mask area in pixels to that of total face area in pixels.

For different progress monitoring scenarios, a completion percentage estimation rule is set, as shown in Figure 3.10. For outdoor monitoring, the progress percentage of the latest activity associated with an element is only estimated from the appearance on the outer face of the element. In the case of indoor construction monitoring, elements can be categorized into two types: Inner and outer elements. Activity progress is estimated on the inner largest of the outer elements on the two largest faces of an inner element. If both faces of an inner element have the same ongoing latest activity, the progress percentage is calculated as the average progress on both faces. Alternatively, the latest and preceding

doi:10.6342/NTU202302333

activity progress is recorded separately if different activities are in progress on two faces.

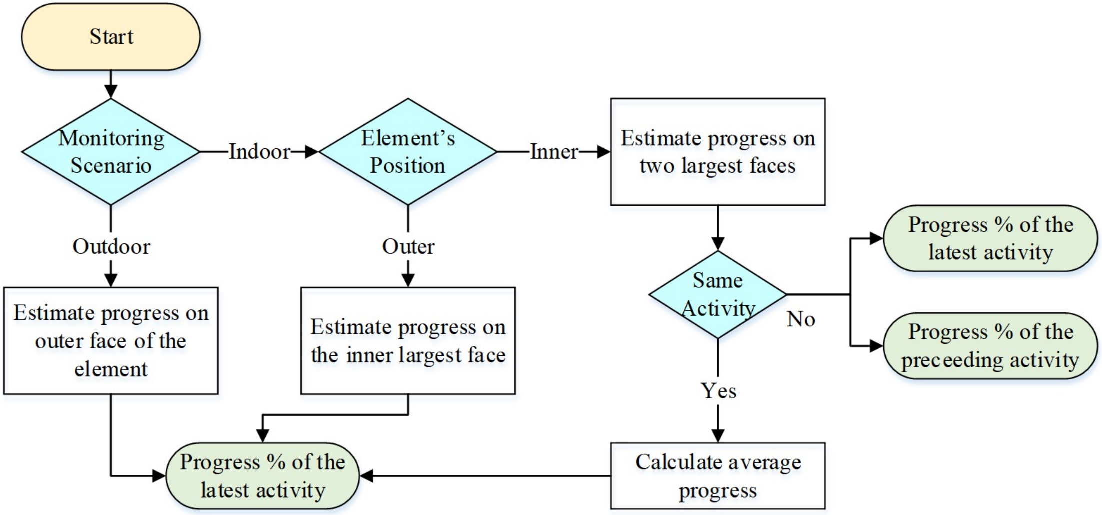

Start
Monitoring Scenario
Indoor
Element's Position
Inner
Estimate progress on two largest faces
Outdoor
Outer
Progress % of the latest activity
Same Activity
No
Progress % of the preceeding activity
Estimate progress on outer face of the element
Estimate progress on the inner largest face
Yes
Progress % of the latest activity
Calculate average progress

Figure 3.10: Rule-based progress estimation

ALPMS can easily calculate the activity level progress percentage for different linkage types used in a 4D BIM. Table 3.3 shows the formulae for the same. While type A and type B progress can be calculated directly, progress for type C can be derived by taking the mean progress of the activity across all elements present in the as-planned BIM. Progress percentages of activities associated with the building elements are then exported into a spreadsheet. These percentages can be used for schedule updates, look ahead planning and delay analysis. However, schedule update is beyond the scope of the present study.

Table 3.3: Completion percentage estimation for different 4D linkage types

Type	Linkage	Progress percentage calculation
A	One-to-one linkage between the activity and the element	Activity progress = δ^{e}_{act}
B	Multiple activities linked to one model element	Latest activity progress = $\delta^{e}_{act_{latest}}$
C	Multiple elements linked to one activity	Activity progress = $1/n \sum \delta^{e_i}_{act}$

doi:10.6342/NTU202302333

3.1.8 Progress visualization

Progress visualization on a 3D model helps to understand the project status instantly. Visualization results help project managers identify and focus on areas that need attention. Managers can resolve constraints or deploy additional resources to speed up the progress in slow-progress regions. The last module of ALPMS helps visualize the activity-wise progress status on the 3D as-planned model and the as-built point cloud. The as-planned model's elements are color coded with the traffic signal metaphor to show the progress status. Elements with ongoing latest activities similar to the planned schedule are colored with green, elements with preceding activities are colored with amber, and the 'no progress' status is highlighted with red. An example is shown in Figure 3.28, in the experiment section.

Additionally, the as-built point cloud is classified with activity level progress information. All images of reality capture are passed through the image segmentation network, and activity-wise color-coded masks are generated. Pre-computed depth maps and the camera poses from the 3D reconstruction module are used to project the 2D masks onto the 3D point cloud; thus, a semantically segmented as-built point cloud is generated. The process for semantic segmentation of point cloud is shown in Figure 3.11. An example of a case study can be seen in Figure 3.29. This study uses the segmented point cloud for progress visualization only.

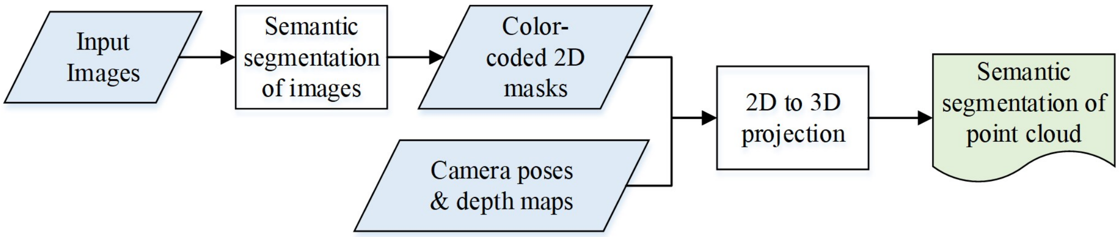

Input Images
Semantic segmentation of images
Color-coded 2D masks
Camera poses & depth maps
2D to 3D projection
Semantic segmentation of point cloud

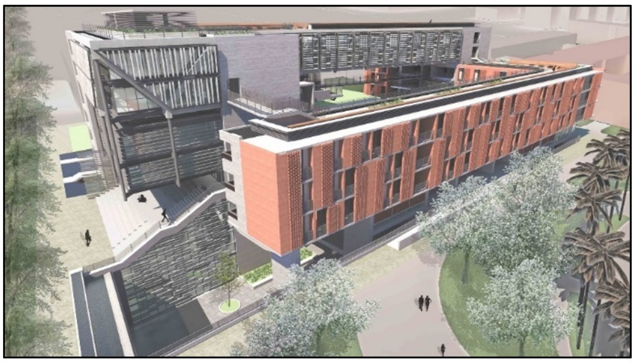
Figure 3.11: Semantic segmentation of the point cloud

doi:10.6342/NTU202302333

3.2 Experiments

3.2.1 Input data

The ALPMS was tested in two construction projects in Taiwan to validate the methodology. Project A is an educational building construction project within National Taiwan University, and Project B is an office building construction project in Taipei. Rendered 3D views of these buildings are shown in Figure 3.12. These two projects were chosen based on their project completion status. While structural work was in progress in Project A, Project B was in the finishing-work stage. Four case studies were conducted to test the performance of the ALPMS system for different progress monitoring scenarios. These monitoring scenarios include outdoor, indoor without the roof, complete indoor, and indoor at the balcony. The illumination condition for the first two cases was brighter than the others, whereas the complete indoor condition was relatively the darkest. Three different types of cameras: UAV, 360-degree cameras, and fisheye cameras, were used for reality capture. Video data were collected from the construction sites and used as input to the ALPMS system. The system further processed the video and extracted overlapping image frames per second and six perspective images of 600 x 600 size from each 360-degree image. The details of four case studies are given in Table 3.4.



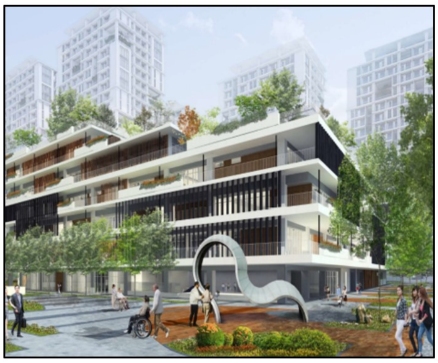


Figure 3.12: Rendered 3D views of the projects: Left- project A and Right- project B

doi:10.6342/NTU202302333

Table 3.4: Details of the case studies

Case Study	Construction Status	Monitoring scenario	Camera Type	Image Frames	Perspective Images
A_1F	Structural Work	Indoor – w/ roof	360-degree	280	1680
A_3F	Structural Work	Indoor –w/o roof	360-degree	1155	6930
A_4F	Structural Work	Outdoor	UAV	665	—
B_3F	Finishing Work	Indoor - Balcony	fisheye	483	—

The BIM used in the construction phase and the project schedules prepared in commercial scheduling software were received from the contractors. The process described in Section 3.1.1 was used to create 4D BIMs. Project A's 4D BIM is shown as an example in Figure 3.13. The 4D BIMs in the IFC format were exported and used for model-assisted progress monitoring in further steps.

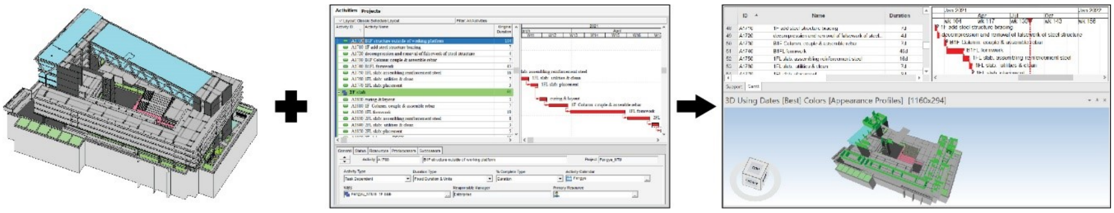

3D Using Dates [Best] Colors [Appearance Profiles] [1160x294]

Figure 3.13: An example of 4D BIM creation

3.2.2 3D reconstruction, registration, occupancy detection

Module 2 of ALPMS created dense point clouds and estimated camera poses from the input images using the SfM-MVS-based 3D reconstruction method described in Section 3.1.2. Further, 4D BIMs and as-built point clouds were registered in the same coordinate system using the coarse-to-fine registration method. During fine registration, BIMs were converted into equivalent point clouds by sampling 500 points/ mm2 of surface area. Also, as-built point clouds were cleaned using statistical outlier filtering. Six nearest neighbor points were considered for estimating the threshold by Equation 1, and the standard deviation multiplier α was set as 1. These values were tuned to get the point cloud with optimum point density. The registration process generated a 4x4 transformation matrix that helped to superimpose the BIMs and point clouds and transformed the cameras into as-planned models' coordinates. The BIM+Point cloud Registration registration errors for four case studies are shown in Table 3.5. BIM in IFC format, the registered point clouds, cameras, and images corresponding to each camera were hosted, visualized, and processed in an open-source 3D computer graphics software [13].

Table 3.5: BIM+Point cloud registration error

Case Study	BIM+Point cloud Registration error
A_1F	1.89 cm
A_3F	2.99 cm
A_4F	1.61 cm
B_3F	1.08 cm

The outcome of the processes described above is illustrated graphically in Figure 3.14. Each column in this figure represents a case study. The first row shows the sample of input images for image-based 3D reconstruction. In the second row, reconstructed point clouds are shown. 4D BIMs are shown in the third row. The registered BIM and point cloud models can be seen in the 4th row of the image. Camera positions in the BIM coordinate are displayed in the fifth row. And the last row shows the samples of one camera view from each case study where the BIM, point cloud, and the corresponding real images can be seen together. The cameras and their related real images were further used for orthographic view synthesis.

Elements in the as-planned 4D BIMs were checked through an occupancy-based method to confirm that construction had started for a particular element. The outcome of this occupancy check guided the ALMPS to only focus on those elements that are under construction. The relevant method is explained in Section 3.1.3. Each element's bound-

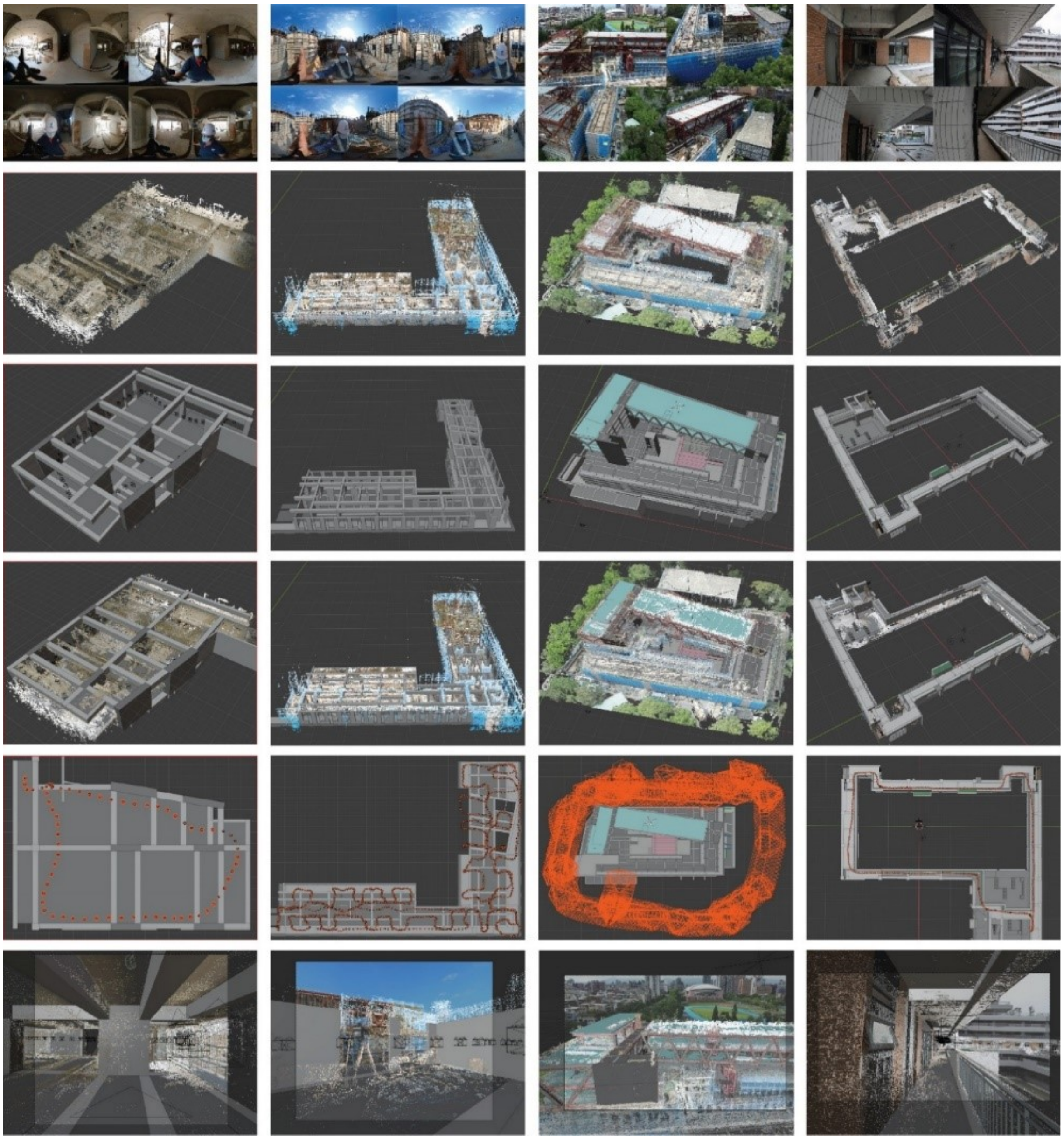


Figure 3.14: 3D reconstruction and BIM+point cloud registration: First row- input images for reconstruction; second row- reconstructed point clouds; third row- as-planned 4D BIMs; fourth row −registered BIM+point clouds; fifth row- camera positions in BIM coordinates; and sixth row- registered image+BIM+point clouds

doi:10.6342/NTU202302333

ing boxes were increased with a threshold to accommodate the registration errors and the presence of temporary structures. For case studies, the threshold value varied between 0 and 10 percent higher than the registration errors, and an appropriate value of ϵ_{reg} was set. The in-progress status of an element was determined by counting the number of points inside the element's bounding box. The element was considered under construction if the count exceeded 1000 $points/m^2$ [19]. Finally, a filtered point cloud was generated for each case. The filtered point cloud of the A_3F case for different ϵreg is shown as an example in Figure 3.15. The element-wise bounding boxes are shown in blue color. In the A_3F case study, with ϵ_{reg} = 3cm, all elements' presence in the reality model could be checked accurately.

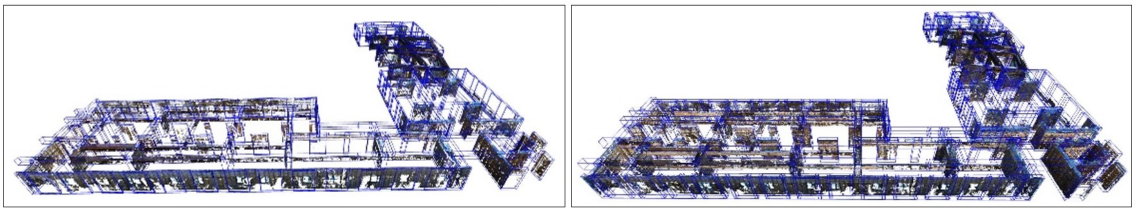


Figure 3.15: Filtering for occupancy check with different ϵreg values: left: ϵreg. = 0 cm ; and right: ϵreg. = 3 cm

3.2.3 Orthographic view synthesis

3.2.3.1 Projective transformation of selected camera view

For elements that fit the criteria for orthographic view synthesis by projective transformation, the method described in the first part of Section 3.1.4 was used for processing. An equivalent orthographic image of the element's face was generated by applying projective transformation on a selected camera image. In case the element's face was visible from n number of camera views, only the first five images where maximum face coverage was observed were selected for dynamic occlusion checking. Manually captured

doi:10.6342/NTU202302333

360-degree images also capture the person who holds the camera. The target element's face gets occluded by the person's appearance in a few camera images. The dynamic occlusion checking algorithm eliminated such camera views. Finally, an image with no dynamic occlusion was selected for projective transformation. By fixing the number five, cameras close to the elements were only considered. The appearance of the element's face gets distorted in far-away cameras. Also, chances of occlusion increase with higher element-to-camera distance. Some examples of orthographic view synthesis by the projective transformation method are shown in Figure 3.16. In this figure, each row represents one example. The first and second column of the figure shows the element's face position in the selected camera view and the ROI cropped image of the element's face in the selected image. The view from the virtually placed orthographic camera is shown in the third column. And the last column shows the equivalent orthographic view of the element's face.

3.2.3.2 NeRF-based orthographic view synthesis

In case orthographic view synthesis through projective transformation was not feasible, a NeRF-based approach was taken for novel view synthesis, as explained in the second part of Section 3.1.4. For training a NeRF model, two different methods were tested. In the first case, a NeRF model was trained with all the images and camera poses belonging to a reality capture. In the other case, individual models for individual elements were trained with images and camera poses corresponding to that element. In the case of a particular element-wise model, the number of input images varied from 10 to 200 based on the element's size, shape, and position in the 3D scene. The total number of rays marched through an image was equal to the image's height multiplied by the width of the image. For

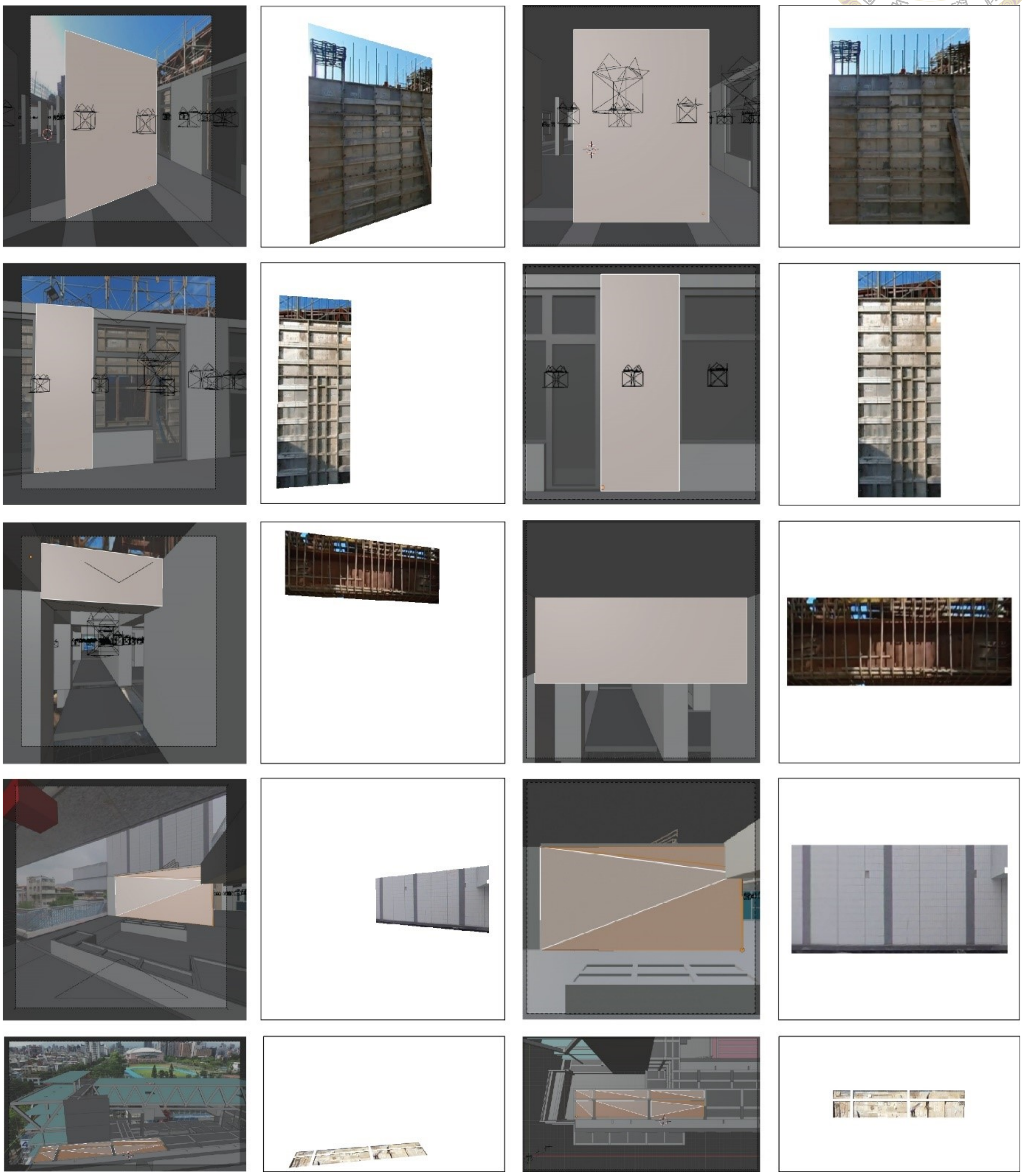
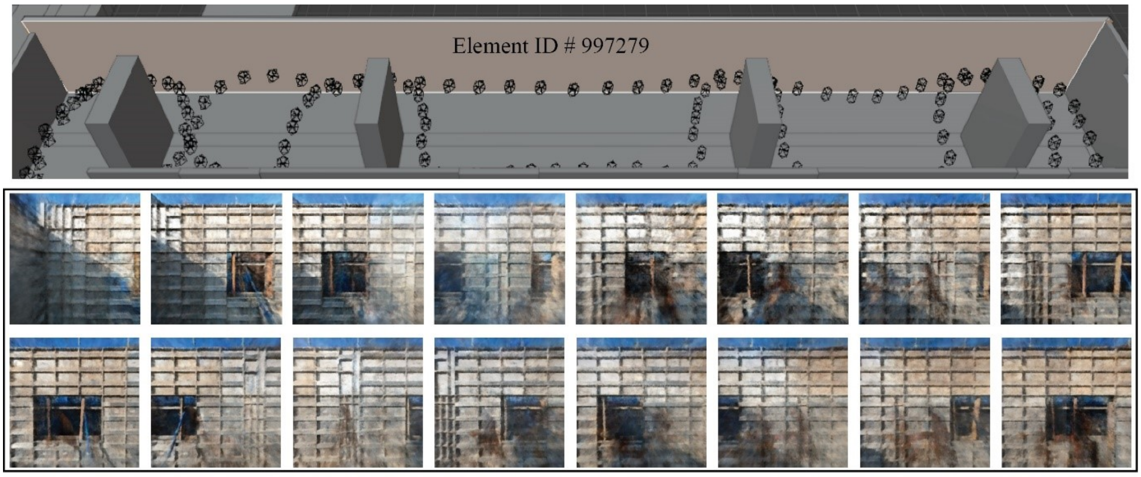


Figure 3.16: Orthographic view synthesis through perspective projection: First column- element's face in the selected camera view; second column- ROI cropped selected camera image; third column- element's face in the virtual camera view; fourth column －ROI cropped orthographic view of the element's face

doi:10.6342/NTU202302333

example, in the case of a 600x600 image number of rays marched was 3,60,000. During training and validation, 1024 rays were processed in a batch, and 64 points were sampled along each ray. All models' outcomes were checked by visually comparing a ground truth image and a synthesized image and by image reconstruction assessment metrics such as PSNR and SSIM. Each model was trained for 20,000 iterations. The time required for training using the method proposed by [63] was between 2 to 5 minutes on a server computer with a Linux operating system, 128-gigabyte memory, and 24-gigabyte NVIDIA RTX 3090 graphics processing unit (GPU). For generating the orthographic views from the NeRF model, the virtual camera was placed at different distances from the element's face, and the quality of the synthesized images was checked. The distances varied from 0.5m to 3m were tested. An example of NeRF-generated orthographic images is shown in Figure 3.17. The top row in the figure shows the element's position in the 3D view, and the bottom row shows the orthographic views of the element's inner face.



Othographic views in 16 frames

Figure 3.17: Orthographic view synthesis using NeRF: Top- 3D view of the element, bottom –orthographic views in 16 frames

3.2.4 Deep-learning-based semantic segmentation

For progress status detection through semantic segmentation, a deep-learning model Mask RCNN was chosen because of its higher accuracy than its peers. The transfer learning approach was adopted to train, validate and test the model with our custom dataset. The mask-RCNN model's last layer was modified, and the backbone architecture was altered to customize it for detecting construction activity classes: concrete, formwork, reinforcement steel, steel structure, wall tiles, and floor tiles. The dataset was split into the train, validation, and test data, with 1723, 490, and 245 images. The model was fine-tuned with six sets of hyperparameters, as shown in Table 3.6. The hyperparameter tuning included changing the backbone architecture, optimizers, learning rate (LR), and LR schedules. Four different backbone architectures with different depths such as ResNet-18, ResNet-34, ResNet-50, and ResNet-101, were tested for feature detection. Different neural network depths are tested because the deeper network often may cause overfitting. The ResNet models were pre-trained on Microsoft's common object in context (COCO) dataset and finetuned during segmentation model training. The model was trained for 100 epochs for each hyperparameter setting. During the training, one epoch's training followed one validation round, and ten images were processed in a batch. Data augmentation was applied to the training data to bring variability and avoid overfitting. Augmentation methods included horizontal and vertical flip, random rotation, random cropping, and random photogrammetric distortions (change of brightness, contrast, hue, and saturation).

Optimizer plays a vital role in deep-learning model training. It changes network weights and learning rates to reduce the losses. In this study, three popularly used optimizers: Adam, SGD with momentum, and Adam with weight decay (AdamW), are tested

doi:10.6342/NTU202302333

Table 3.6: Hyperparameter settings

Hyper-parameters	R18 _adamw	R34 _adamw	R50 _adamw	R101 _SDG	R101 _adam	R101 _adamw
Backbone	ResNet-18	ResNet-34	ResNet-50	ResNet-101	ResNet-101	ResNet-101
Optimizer	AdamW	AdamW	AdamW	SDG with Momentum 0.9	Adam	AdamW
Learning Rate (LR)	0.00005	0.00005	0.00005	0.0001	0.0001	0.00005
Weight Decay	0.005	0.005	0.005	0.0005	-	0.005
LR Scheduler	Reduce on plateau: factor 0.75 Patience: 3 epochs	Reduce on plateau: factor 0.75 Patience: 3 epochs	Reduce on plateau: factor 0.75 Patience: 3 epochs	Reduce 10x per 25 epoch	Reduce 10x per 25 epoch	Reduce on plateau: factor 0.75 Patience: 3 epochs
Epochs	100	100	100	100	100	100
Batch Size	10	10	10	10	10	10

in combination with various initial learning rates and learning rate schedules. Previous studies have mainly used the Adam optimizer for its faster convergence rate than Stochastic gradient descent (SGD). However, the solutions obtained from Adam are not always generalizable [34]. That means the model performs less accurately when applied to the test data. On the other hand, the improved version of Adam, called AdamW, which includes a weight decay in its algorithm, performs much better in generalizability and convergence [56]. Weight decay is a form of regularization that helps to reduce the chance of overfitting. For the cases where the AdamW optimizer was used, the initial learning rate was set as 0.00005. The learning rate scheduler was introduced so that when the gradient reaches a plateau and the validation loss does not reduce for three consecutive epochs, the LR will be reduced by a factor of 0.75. Similarly, for Adam and SDG with momentum, the initial LR value was 0.0001, and the LR scheduler reduced it ten times after every 25 epochs. During training, how learning rate values were changed in every epoch is shown in Figure 3.18. The deep-learning models were developed using the PyTorch framework

doi:10.6342/NTU202302333

and trained on a server computer with a Linux operating system, 128-gigabyte memory, and 24-gigabyte NVIDIA RTX 3090 grapics processing unit (GPU).

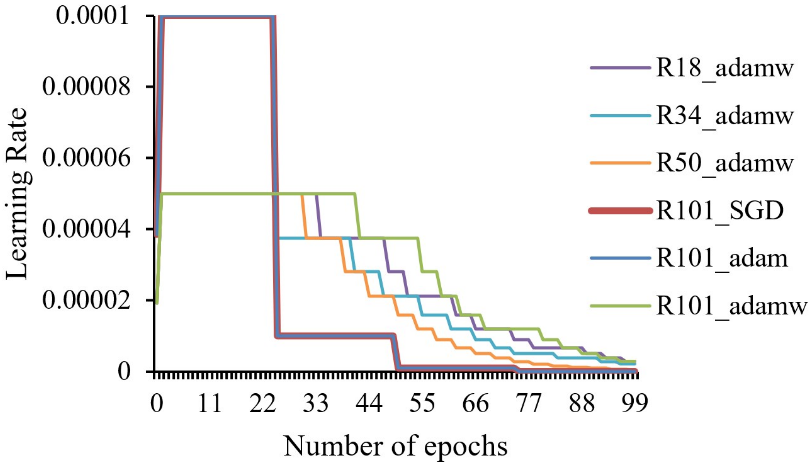


Figure 3.18: Changes in learning rates per epoch

3.3 Results and discussions

The following section discusses the performance of NeRF models, the semantic segmentation model, and the overall activity-level progress monitoring system.

3.3.1 Performance of the NeRF

3.3.1.1 Factors affecting synthetic image quality and training time

Two different methods were tested while training the NeRF models. In one case, all images of a reality capture were used to prepare a NeRF model for the entire scanned area. In the other case, for each element that required NeRF based solution, a separate NeRF model was trained, using the method introduced in the second part of Section 3.2.3. The results and the synthetic image output from the two case studies are shown in Table 7. It is observed that with proper training cameras and image sets, the similarity between a ground

doi:10.6342/NTU202302333

truth image and a synthetic image can reach more than 90%. On the contrary, an improper setting outputs a blurred and nosy image. The higher PSNR and SSIM values and the output image quality indicate that element-wise training ensures better image synthesis. The phenomenon is normal with NeRF implementation proposed by [62]. The NeRF model focuses on the center of the 3D scene reconstructed from the available camera poses. While using all cameras in reality capture, the focus point doesn't remain on the target element if it is not located at the scene's center. On the other hand, element-wise training with corresponding cameras confirms the model's focus on the target element only. The view direction of all cameras used for training must be toward the target element.

The experiment shown in table 7 was extended to check the effect of input image numbers and resolution on the output image quality and training time. It is found that the output image quality from a NeRF model largely depends on the resolution of the input images, camera positions, and view directions but not on the number of images used for training. Results in Table 3.7 show that training loss is reduced in higher resolution images. Still, a higher number of high-resolution input images has no significant impact on training time while using the method proposed by [63]. Although the perspective images derived from a 360-degree image have a lower resolution, it has no significant impact on the performance of the ALPMS. A NeRF model trained with proper camera positions and view directions outputs synthetic images with the sufficient quality required for activity-level progress monitoring. The performance of ALPMS is discussed in the third part of this section.

doi:10.6342/NTU202302333

Table 3.7: Factors affecting NeRF models output

Training cameras	No. of images	Image size	Training Loss	Training Time (min)
All cameras in reality capture A_3F	1155	600 x 600	0.00912	3.43
Cameras associated with the element ID# 997279	149	600 x 600	0.00728	3.52
All cameras in reality capture B_3F	483	3840 x 2160	0.00275	2.42
Cameras associated with the element ID# 840815	13	3840 x 2160	0.00152	3.58

3.3.1.2 Effect of element-to-camera distance on orthographic view synthesis

Following Algorithm 4, a virtual camera was placed and traversed along the element to generate images containing orthographic views of the element's surface. Element-to-camera distance controls the number of images generated per face of the element and the quality of the orthographic views. Suppose the camera is placed closer to the element's face. In that case, more images will be generated to get the appearance of the entire face. Although a distant camera placement will reduce the number of generated images, the quality of an output image will be affected. In the case of a higher element-to-camera distance, the camera view may get occluded by the appearance of other objects between the camera and the element's face. Additionally, the synthetic view generated from a distant camera observes blurring effects and visual artifacts. Two examples of synthesized images from 0.5 to 3-meter camera distances are shown in Figure 3.19. The presence of visual artifacts can be seen in photos with higher element-to-camera distances. These artifacts may affect the performance of the image segmentation task significantly. Table 3.8 shows examples of elements 997643 and 432802 with dimensions (32.0mx3.08mx0.2) and (4.1mx3.35mx0.15m), respectively. This table shows the number of images generated for each camera distance and the average segmentation accuracy. 1.5m element-to-camera

doi:10.6342/NTU202302333

distance was optimal for ALPMS as it balances the number of orthographic view images generated per face and the performance of the segmentation on the generated images.

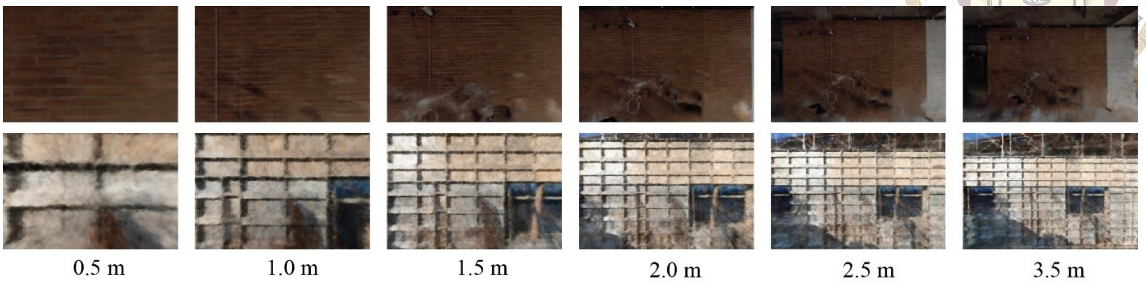


Figure 3.19: Ortho-views with changing element to camera distances

Table 3.8: Effects of element-to-camera distance

Distance	Element # 997643 No. of frames	Element # 432802 No. of frames
0.5	46	8
1.0	32	6
1.5	16	4
2.0	15	2
2.5	12	1
3.0	8	1

3.3.2 Performance of the semantic segmentation

3.3.2.1 Hyperparameter tuning and model selection

The performance of the semantic segmentation models depends on the appropriate hyperparameter selection. In this study, the Mask-RCNN model with six combinations of hyperparameters were tested, and the model, trained with ResNet-101 backbone and AdamW optimizer (R101_adamw), was found to be the best-performing model. The train and validation loss curves are shown in Figures 3.20 and 3.21, respectively. The R101_adamw model displays the lowest loss in training and validation, whereas the model trained with SGD with momentum exhibited the highest. The mAP evaluation metrics evaluated the model performance on the test data. The R101_adamw model performed with the highest accuracy of 0.900 for IoU > 0.5. The model performed significantly

doi:10.6342/NTU202302333

better than other models because of the weight decay regularization effect offered by the AdamW optimizer. Also, using the LR scheduler "reduce on plateau" helped maintain appropriate LRs throughout the training and thus avoided settling the model at the local minima. The comparison of mAP values for six models is shown in Table 3.9. Following the test results, the best-performing segmentation model was chosen for implementation within ALPMS.

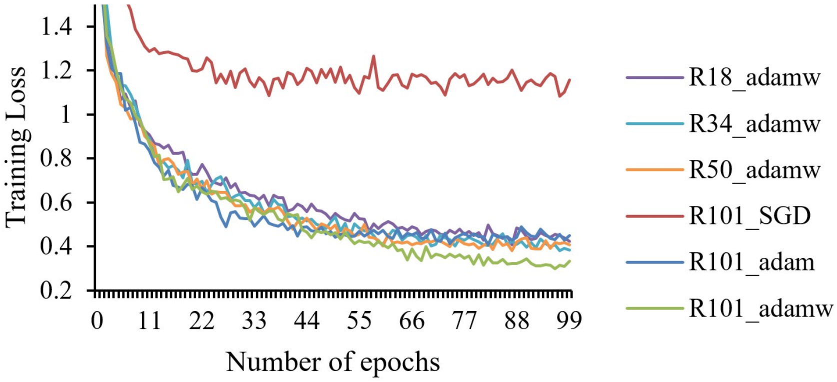


Figure 3.20: Training loss curves

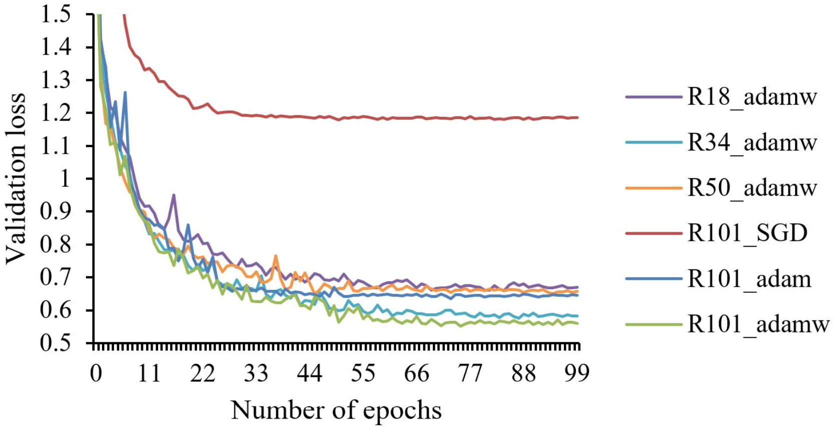


Figure 3.21: Validation loss curves

Table 3.9: Performance comparison of different models

Evaluaton Metric	R18 _adamw	R34 _adamw	R50 _adamw	R101 _SDG	R101 _adam	R101 _adamw
Overall mAP	0.774	0.798	0.781	0.203	0.743	0.900

doi:10.6342/NTU202302333

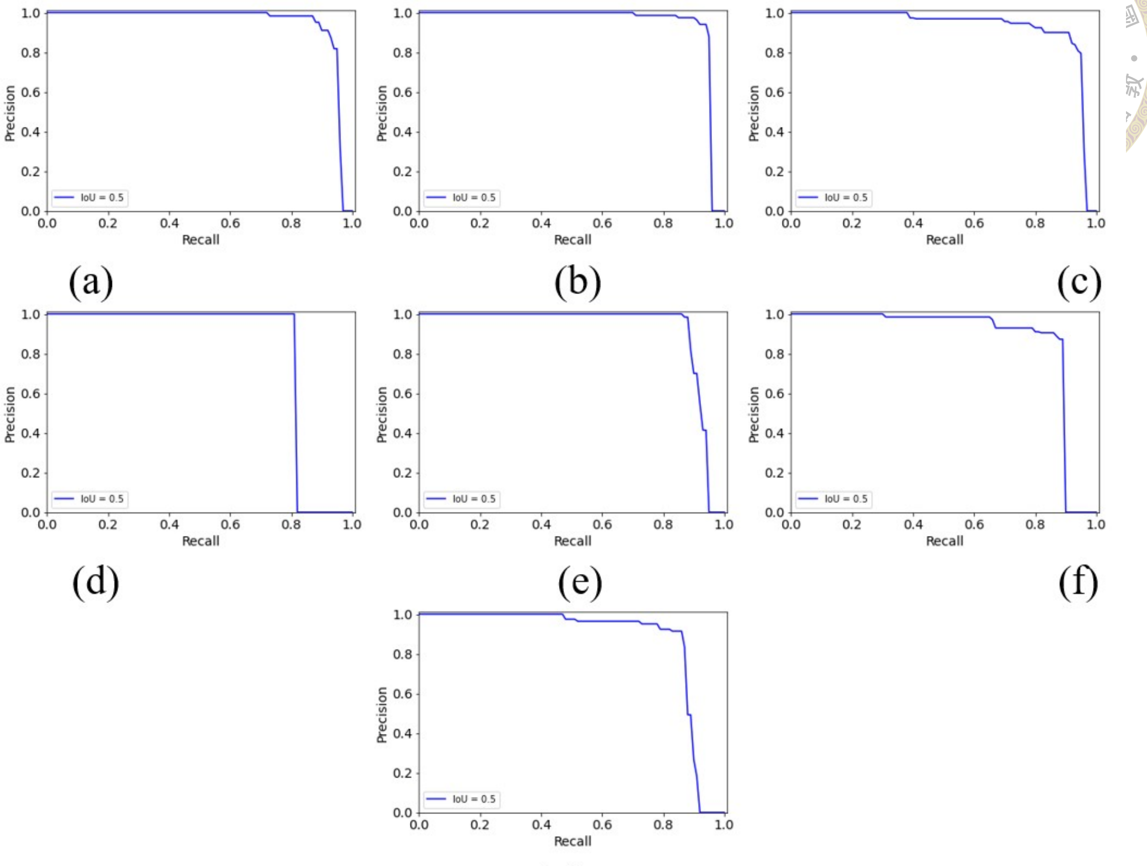
3.3.2.2 Performance of the selected model on original images

The selected Mask-RCNN model was applied to the test dataset containing original site images, and class-wise mAP values for IoU thresholds 0.5 and 0.75 were checked. The results are shown in Table 3.10. The model performed the segmentation task with more than 90% accuracy for IoU threshold 0.5 in most classes. For IoU > 0.75, the model achieved more than 75% overall accuracy. Among all the classes, the detection accuracy for the formwork class is the highest. The distinct and widespread appearance of formwork material on an image helped the model learn the features of that class better. The area under the Precision-recall curves represents the mAP for each category. Figure 3.22 shows the P-R curves for IoU>0.5 for six construction activity classes (a) concrete, (b) formwork, (c) reinforcement steel, (d) steel structure, (e) floor tile, (f) wall tile, and (g) all classes together. Figure 3.23 shows some sample detection results. The left part of this figure shows the detection results on as-is images, where the first column shows the original image and the second and third columns show the ground truth and detected masks. The right part shows sample detection results on ortho-view images created by projective transformation. In this figure, for each element, the left is the ortho view generated by projective transformation, and the right is the detected masks. The overall segmentation model's accuracy does not directly relate to the overall accuracy of the ALPMS because, for any given case study, all six classes have not appeared simultaneously.

3.3.2.3 Performance on synthetic data

The Mask-RCNN model trained with original site images was tested on a dataset containing synthetic photos generated from the NeRF model. However, the model's perfor-

doi:10.6342/NTU202302333

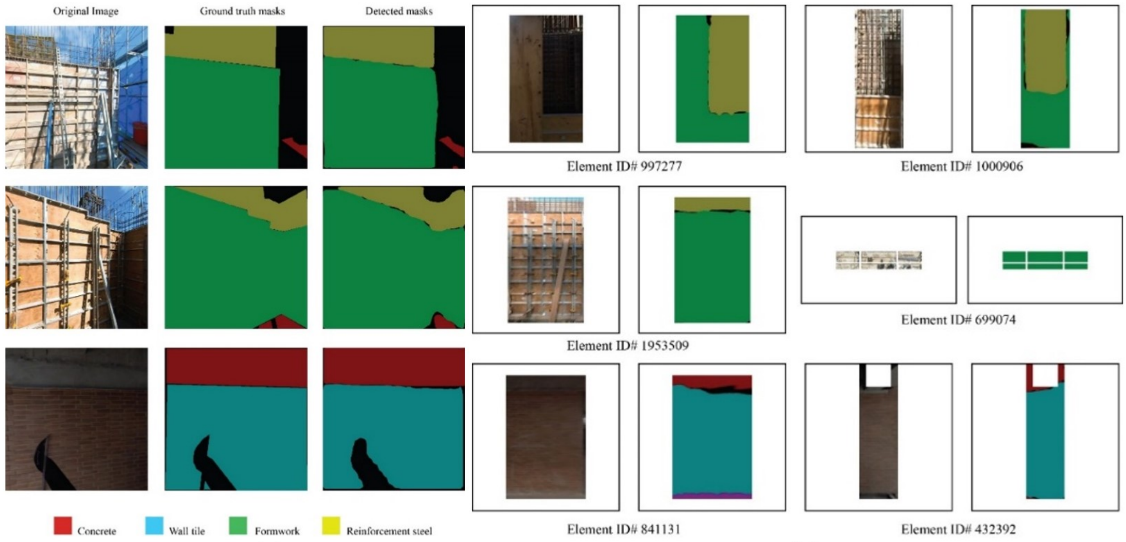


(a) (b) (c)

(d) (e) (f)

(g)

Figure 3.22: Precision-Recall curves: (a) concrete, (b) formwork, (c) reinforcement steel, (d) steel structure, (e) floor tile, (f) wall tile, and (g) all classes



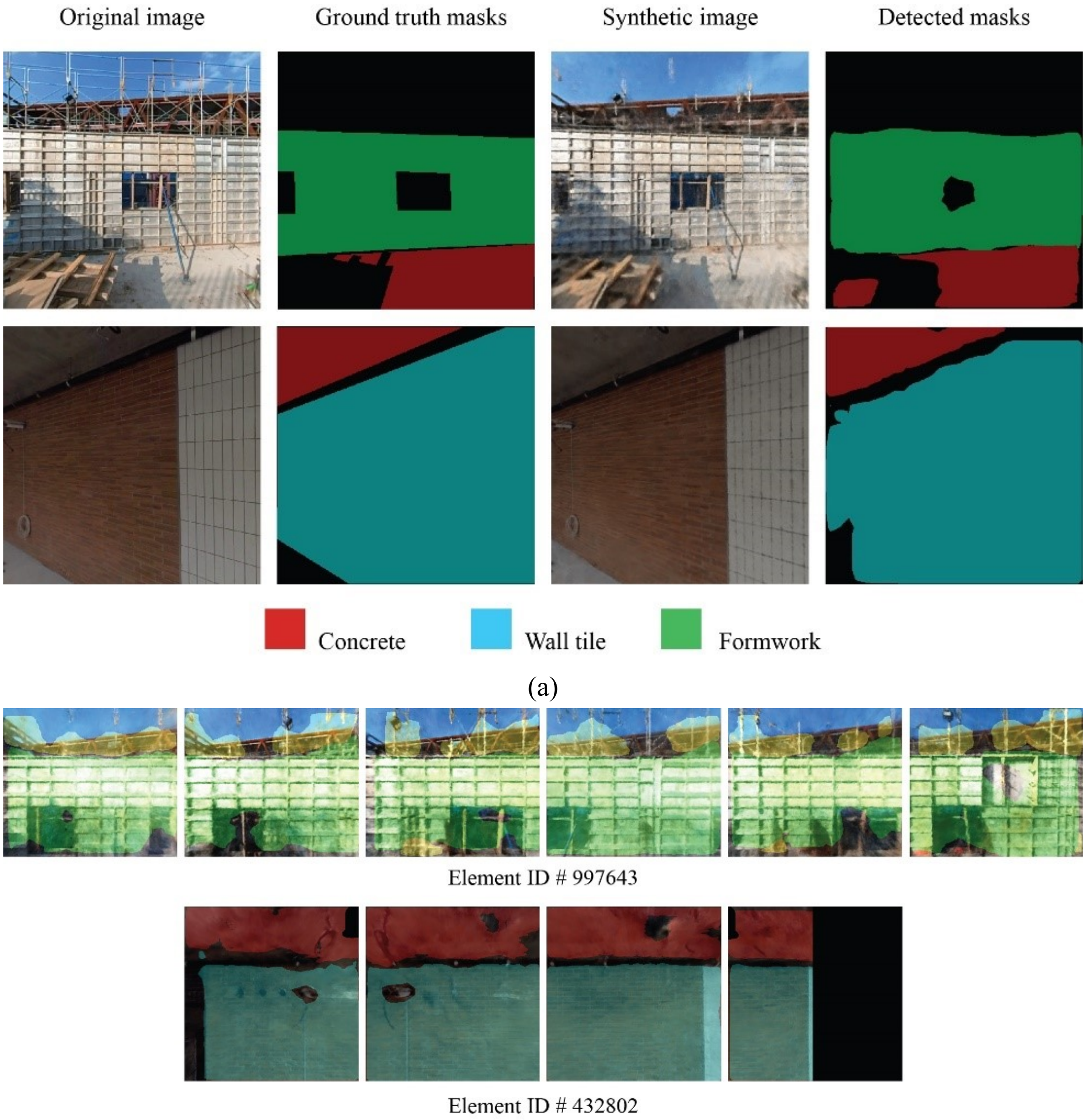
(a) (b)

Figure 3.23: Visualization of the segmentation results: (a) as-is images (b) ortho view images created by a projective transformation

Table 3.10: Class-wise performance of the Mask-RCNN model

Class	AP (IoU >0.5)	AP (IoU >0.75)
Concrete	0.943	0.636
Formwork	0.944	0.915
Reinforcement steel	0.920	0.900
Steel structure	0.812	0.723
Floor tiles	0.916	0.721
Wall tile	0.866	0.750
Overall(mAP)	0.900	0.774

mance was not satisfactory. The segmentation model was then re-trained with a synthetic image dataset to improve its performance on the synthetic data. Although the re-training improved the accuracy up to a certain level, it was not helpful for activity level progress monitoring. Later, original and synthetic data were blended to train the model. The blending could significantly improve the model's performance. The model trained with a blend of 70% original and 30% synthetic images achieved a segmentation mAP of 0.713 and bounding box mAp of 0.885 for more than the 0.5 IoU threshold. However, the addition of more percentage of original data further reduced the model's performance on synthetic image segmentation. The results of different tests are shown in Table 3.11. The top part of Figure 3.24 shows some sample detection results on the synthetic data. This figure's first and third columns are the original and NeRF-generated synthetic images, and the second and fourth are ground truth masks and detected masks. The bottom part of Figure 3.24 shows the image segmentation results on orthographic views synthesized from a NeRF model. This figure shows the segmentation masks overlaid on the synthetically generated orthographic image frames.



(a)

(b)

Figure 3.24: Visualization of the segmentation on synthetic images

doi:10.6342/NTU202302333

Table 3.11: Segmentation model's performance on a synthetic image dataset

Training data	Test Data	Overall mAP (box)	Overall mAP (seg)
Original	Synthetic	0.035	0.000
Synthetic	Synthetic	0.721	0.428
Original+Synthetic (50:50)	Synthetic	0.837	0.650
Original+Synthetic (70:30)	Synthetic	0.885	0.713

3.3.3 Performance of the overall activity-level progress monitoring system

3.3.3.1 Percentage completion estimation

In the end, ALPMS estimated and reported the progress of construction activities in percentage. The method explained in Section 3.1.7 was implemented to calculate the percentage completion of the latest activities per element. The output was exported into a spreadsheet, and the estimated progress percentage per element was compared with the actual progress percentages. The estimated progress percentage progress can be used to update the project schedule. However, the schedule update is beyond the scope of the present study and will be researched in the future. Absolute estimation error was calculated by taking the difference between the predicted and actual progress percentages. The overall performance of activity-level progress monitoring was expressed by the mean absolute error (MAE). MAE is calculated by taking the average of all elements' absolute error. MAE is represented by Equation 4.

$$MAE = \frac{1}{n} \sum y_{pred} - y_{act} \tag{4}$$

The orthographic images were manually annotated to create ground truth masks of

doi:10.6342/NTU202302333

the related activities. Later, the actual progress of the latest activity was measured by Equation 3 using the ground truth masks. The element-wise absolute error estimation sample is shown in Table 3.12.

Table 3.12: Sample of element-wise absolute error estimation.

Element ID	Type	Latest activity	Actual progress	Detected progress	Absolute error
1971166	Wall	Formwork	85	83.25	1.75
1971541	Wall	Formwork	35	34.28	0.72
615274	Wall	Formwork	95	98.12	3.12
673854	Beam	Formwork	100	97.77	2.23
673856	Beam	Formwork	100	89.76	10.24
673858	Beam	Formwork	100	99.11	0.89
………					

The MAE values are calculated for four case studies and shown in Table 3.13. This table shows the overall performance of ALPMS. Initially calculated MAE values indicated in the MAE1 column are affected by three main factors: artifacts or blurring effects in the orthographic views, static occlusion effects caused by material stacking in the construction sites, and dark lighting conditions.

Table 3.13: Overall performance of ALPMS

Case Study	Number of elements checked	MAE1	MAE2	Factors affecting
A_1F	47	5.60	3.98	Lighting condition
A_3F	283	9.54	4.48	Artifacts, static occlusion
A_4F	15	6.73	5.69	Static occlusion
B_3F	152	14.21	8.36	Lighting condition, artifacts, static occlusion
Average		9.02	5.63	

The accuracy of the progress estimation depends on the performance of the segmentation model, and the performance of the segmentation models on the ortho-view images largely depends on the quality of the ortho-view images. The quality of the equivalent orthographic image generated from the projective transformation depends on the appropriate

doi:10.6342/NTU202302333

camera selection from the reality capture. If the selected camera's position is too oblique or distant from the element's face, the equivalent ortho-view image would be blurry. An example of this effect is shown in Figure 3.25. This makes the segmentation tasks difficult. The camera selection algorithm (algorithm 2) in ALPMS is designed to prevent this issue from arising while conducting activity level progress monitoring. Also, instead of segmenting the ortho-view image, the original image was segmented, and the projective transformation was applied to the segmentation masks.

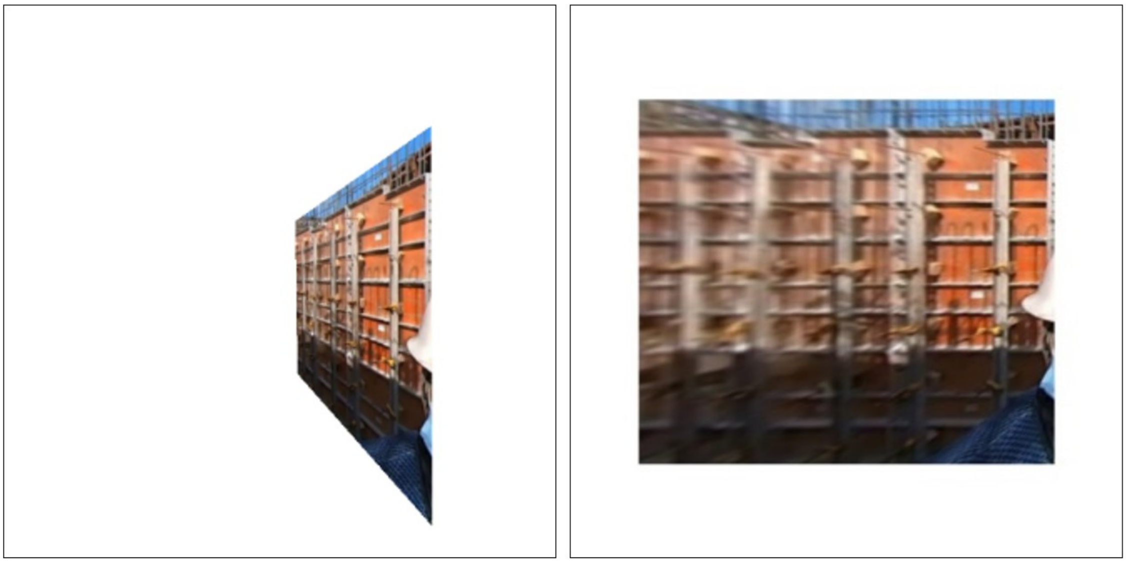


Figure 3.25: Blurring effect caused by the projective transformation of oblique images

Similarly, the output image quality of the NeRF-generated image depends on several factors discussed in Section 3.3.1. The performance of the segmentation model on the synthetic image is also discussed in the second part of Section 3.3.2. Even after considering those factors in ALPMS, in a few cases, artifacts in the bottom half of the NeRF-generated synthetic images have caused a performance reduction in the overall system. Figure 3.26 shows the effect of the artifact in a NeRF-generated ortho view image. For case studies A_1F, A_3F, and B_3F, the construction site images are captured manually. So, the reality capture path's height equals the average human height. And it passes through the upper half of the vertical elements. For this reason, the NeRF model trained with these input images contains fewer details of the bottom part of the element and causes artifacts in the bottom half. In this study, to neglect the effect of the artifacts on the ortho view images, only the top half of the NeRF-generated images are analyzed, and revised MAE2 is estimated. However, automatic data capturing in two levels along the elements could solve this issue in the future.

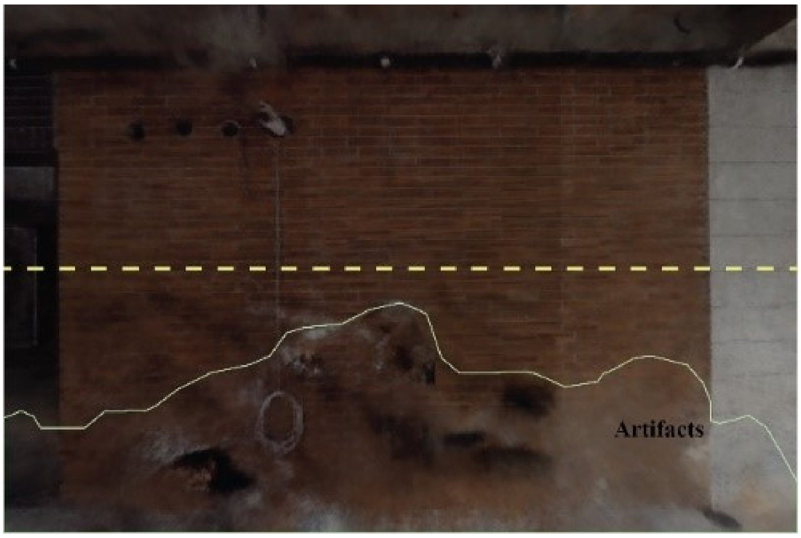


Figure 3.26: Artifacts in the bottom half of the NeRF-generated orthographic image.

The performance of the overall system was also affected by the static occlusion caused by the stacking of different construction objects near the element's face. The dynamic occlusions, whose appearance will not remain in all camera images, and the static occlusion caused by other building elements was handled by algorithm 2 in ALPMS. However, in a few cases, static occlusion caused by material stacking, which blocks the surface appearance of the target face in all camera views in the reality capture, caused a problem for activity-level progress detection. An example of static and dynamic occlusion is shown in Figure 3.27. Methods like occlusion removal and image inpainting [10] can be tried in the future to overcome this static occlusion problem.

Lighting condition is a crucial factor to be considered while conducting progress monitoring in the indoor environment. In the case of indoor construction monitoring,

doi:10.6342/NTU202302333

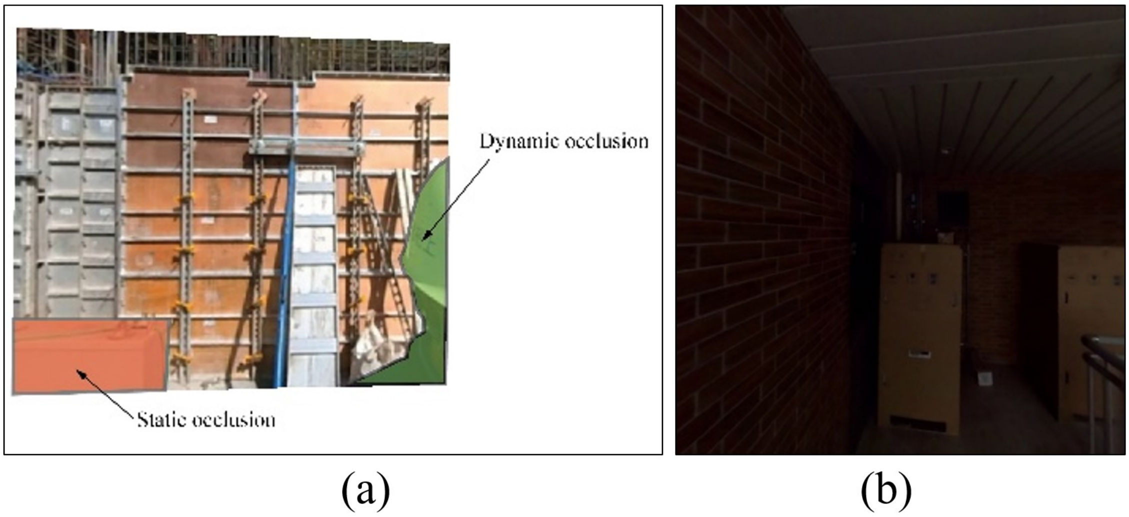
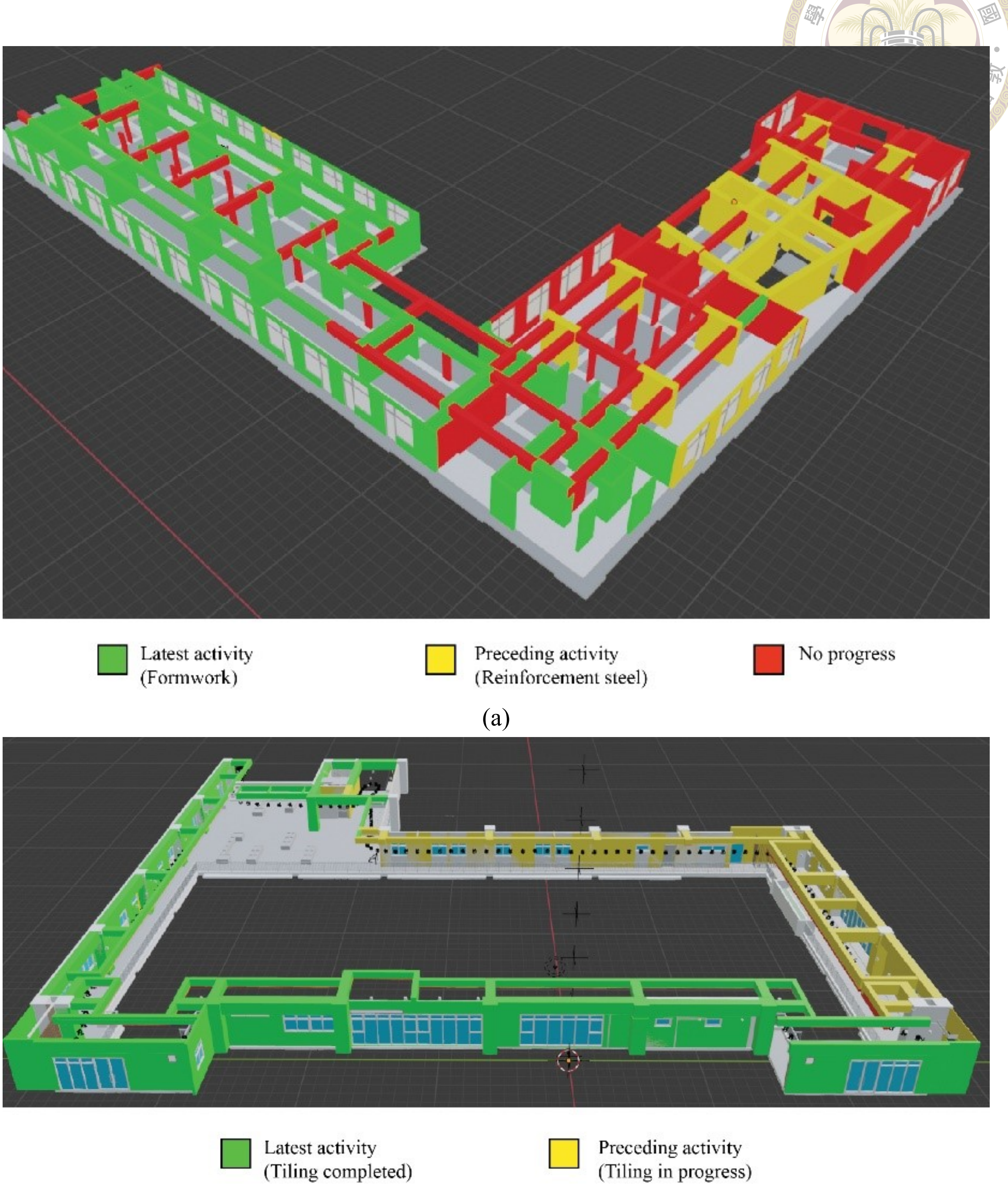


Figure 3.27: (a) Static and dynamic occlusion blocking the appearance of an element's face (b) dark lighting condition

some locations are very dark due to extremely low light. Progress detection results in such areas are often erroneous. A few elements in case studies A_1F and B_3F have faced such situations. Image preprocessing techniques, such as brightening dark photos [48], could be used in the future to solve this issue. For improved MAE2 calculations, such scenarios were neglected.

3.3.3.2 Progress visualization

The progress status of each element was visualized by color coding the as-planned BIM as mentioned in Section 3.1.8. This visualization can help project managers to identify project status in 3D. They can take immediate action in low-progress areas to avoid or mitigate further delays. In the color coding approach, however, one can only understand the latest activity name currently in progress but can not get the intuition on how much that activity is completed. Understanding the completion percentage of the activity is essential to taking appropriate action. For example, in the 3D color-coded display of cases A_3F and B_3F shown in Figure 3.28, seeing the green color, we can understand where the latest activities are in progress. However, it isn't easy to judge their completion status.


🟩 Latest activity (Formwork)
🟨 Preceding activity (Reinforcement steel)
🟥 No progress

(a)

🟩 Latest activity (Tiling completed)
🟨 Preceding activity (Tiling in progress)

(b)

Figure 3.28: Progress visualization through color-coded as-planned BIM

Further, the as-built point cloud model was semantically segmented to enhance the 3D visualization with activity-level progress completion status. By looking at the semantically segmented point cloud, one can easily perceive the activity-wise progress status. Figure 3.29 shows two examples of cases A_1F and B_3F. The green patches on the A_1F point cloud indicate the quantity and location of the formwork removal activity are yet to complete. Similarly, the exact spot where the tiling work is yet to be finished can be determined in B_3F. In this study, the segmented point clouds are only used for progress visualization. However, future research in this direction would explore the opportunity for activity-level progress monitoring through the digital twin construction.

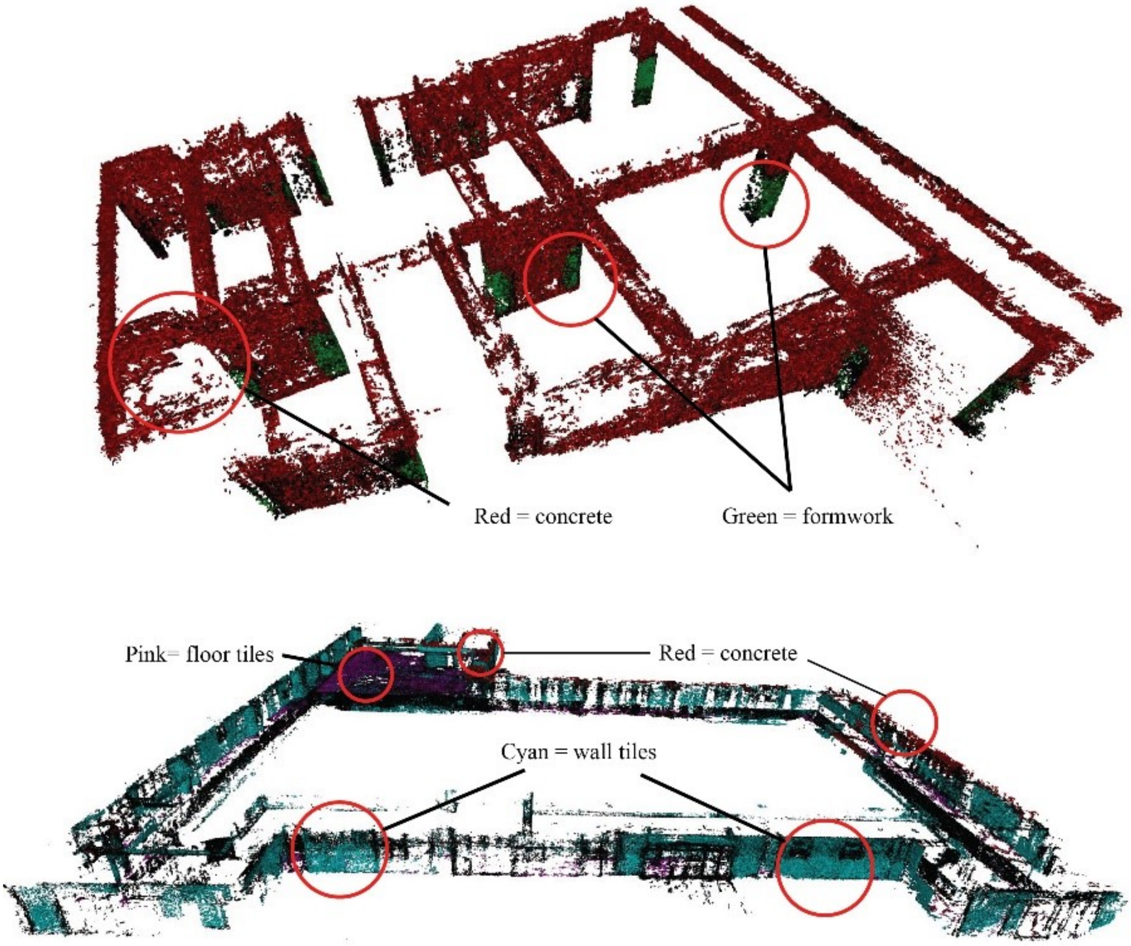


Figure 3.29: Progress visualization through semantic segmentation of as-built point cloud

3.3.4 Implementing ALPMS in the construction management process

In construction management, progress monitoring is a crucial aspect. It helps project managers track construction activity progress, identify potential issues, and take corrective actions before they become critical. The ALPMS framework proposed in this research can automate the progress monitoring process and provide more accurate and objective information about the progress of construction activities. By using 3D-informed orthographic views and deep learning-based semantic segmentation, the ALPMS framework can detect and report the completion percentage of construction activities. It can help construction managers to track the progress of individual activities, such as formwork, reinforcement, and concrete placement, and identify potential delays or issues earlier in the construction process. The progress of each activity can be visualized through BIM color coding and point cloud segmentation, which can help project managers to make better-informed decisions and improve resource allocation.

Implementing the ALPMS framework in the construction management process would require some effort. First, a monitoring mechanism needs to be established, which includes selecting devices and methods for capturing images of the construction site, planning the reality capture, and executing the steps of ALPMS. Processes involved in implementing ALPMS and an approximate estimation of time spent for each process are shown in Table 3.14. Time data is estimated from the four case studies presented in this research. Processes that are performed once during the project are colored in peach. Sky-blue color is used to represent the processes that are conducted every time after data collection. And the green-colored processes are applied to each element that is under construction. The remarks column highlights the factors affecting the average time taken for each process. It is observed that 3D reconstruction through SfM-MVS is the most time-consuming process that needs to be performed each time. Post-3D reconstruction workflow of ALPMS can be completed within an hour. Future research should investigate reducing the time and frequency required for 3D reconstruction through Simultaneous Localization and Mapping

(SLAM) [87] and CNN-based 2D to 3D image localization techniques [95] (Wang et al. 2023).

Table 3.14: Details of approximate time spent for implementing ALPMS

Processes	Algorithms	Approx. time spent	Remarks
Data collection (video)	NA	15-20 mins/ round	Time depends on the area covered
Data download and upload to the cloud server	NA	5 mins/ 10GB	Time depends on file size
3D reconstruction and camera pose estimation	SfM-MVS	10-15 hrs/ model	Time depends on the number and resolution of images
Point cloud cleaning	SOR	5 mins/ model	Time depends on the size of the as-built point clouds
4D BIM and point cloud registration	Manual point picking	10-15 mins/ model	Coarse registration
	ICP	10-12 secs/ model	Fine registration
Detection of elements under construction	Algo. 1	2-5 mins/ model	Time depends on the number of elements in the model
Dataset preparation*	Manual labeling (polygon mask)	3-5 mins/ image	Time depends on the number of images in the dataset
Deep-learning model training for semantic segmentation*	Transfer learning, hyperparameter tuning	12-15 hrs/ project	Time depends on several factors: Hardware capacity, dataset size, etc.
Camera selection, ortho-view synthesis, progress status detection, percent-complete estimation	Algo. 2, Projective Transformation	10 secs/ element	Applicable for most of the elements
	Algo. 2, NeRF	2-5 mins/ element	Only a few elements require NeRF
Progress visualization	BIM coloring	10 secs/ model	Time depends on the number of elements in the model
	Point cloud Segmentation	5-10 mins/ model	Time depends on the size of the as-built point clouds

* This is a one-time effort, not required for every project

Other factors that must be considered for the practical implementation of ALPMS are image quality, the requirement of images for representing an element’ s appearance, and the frequency of image data collection. Image sizes used in four case studies and the overall performances of ALPMS are shown in Tables 3 and 12, respectively. It is observed that images with a minimum size of 600 x 600 confirmed the satisfactory performance of ALPMS. For activity-level progress monitoring, the appearance of the target element must be captured fully in at least five or more images in the reality capture. While data collection, the same was ensured by recording videos near and around each building element. However, automatic data collection following optimized reality capture plans [41] shall be tried in the future. The frequency of image data collection depends on the project requirements. Project managers can decide the frequency considering the pace of the project’ s progress. During the initial project phase, when the progress pace is relatively slow, fortnightly data collection can be conducted. However, during peak construction time, reality capture shall be performed weekly.

3.4 Summary

The proposed ALPMS (Activity-Level Progress Monitoring System) is a comprehensive solution for estimating and visualizing construction progress at the schedule activity level. By leveraging construction images and 4D BIM data, ALPMS accurately determines the progress percentages of construction activities associated with BIM elements. It employs as-built point clouds reconstructed from site images, compares them with the as-planned BIM, and measures the progress of activities related to the in-progress elements through the segmentation of orthographic views. The system utilizes two innovative approaches, projective transformation and NeRF-based novel view synthesis, to generate

doi:10.6342/NTU202302333

equivalent ortho-views, and employs a deep-learning model for semantic segmentation to detect construction activities. Through validation in multiple case studies, ALPMS demonstrates promising performance with an average error rate of 5.63%. While some challenges, such as artifacts in NeRF-generated images and occlusion caused by material stacking, need to be addressed, the system sets the stage for future research directions, including volume-based progress estimation and predictive scheduling, to further enhance construction progress monitoring and decision-making processes.

Chapter 4 Progress update in the project schedule

4.1 Methodology

The proposed methodology is shown in Figure 4.1. It displays the steps required for progress updates by automatically linking project schedules and reality models. Two main components of this methodology are to recognize the location (L), building elements (E), and materials (M) from reality models and project schedules and to map them for progress updates.

4.1.1 Progress estimation from reality models with L-E-M information

Two separate cases are considered: with the availability of 3D BIM (right side of Figure 4.1) and without having a 3D BIM (left side of Figure 4.1) for the construction project.

In case the 3D BIM is available for a construction project, the reality models (as-built point clouds) are registered with that manually by choosing three or more distinct corre-

doi:10.6342/NTU202302333

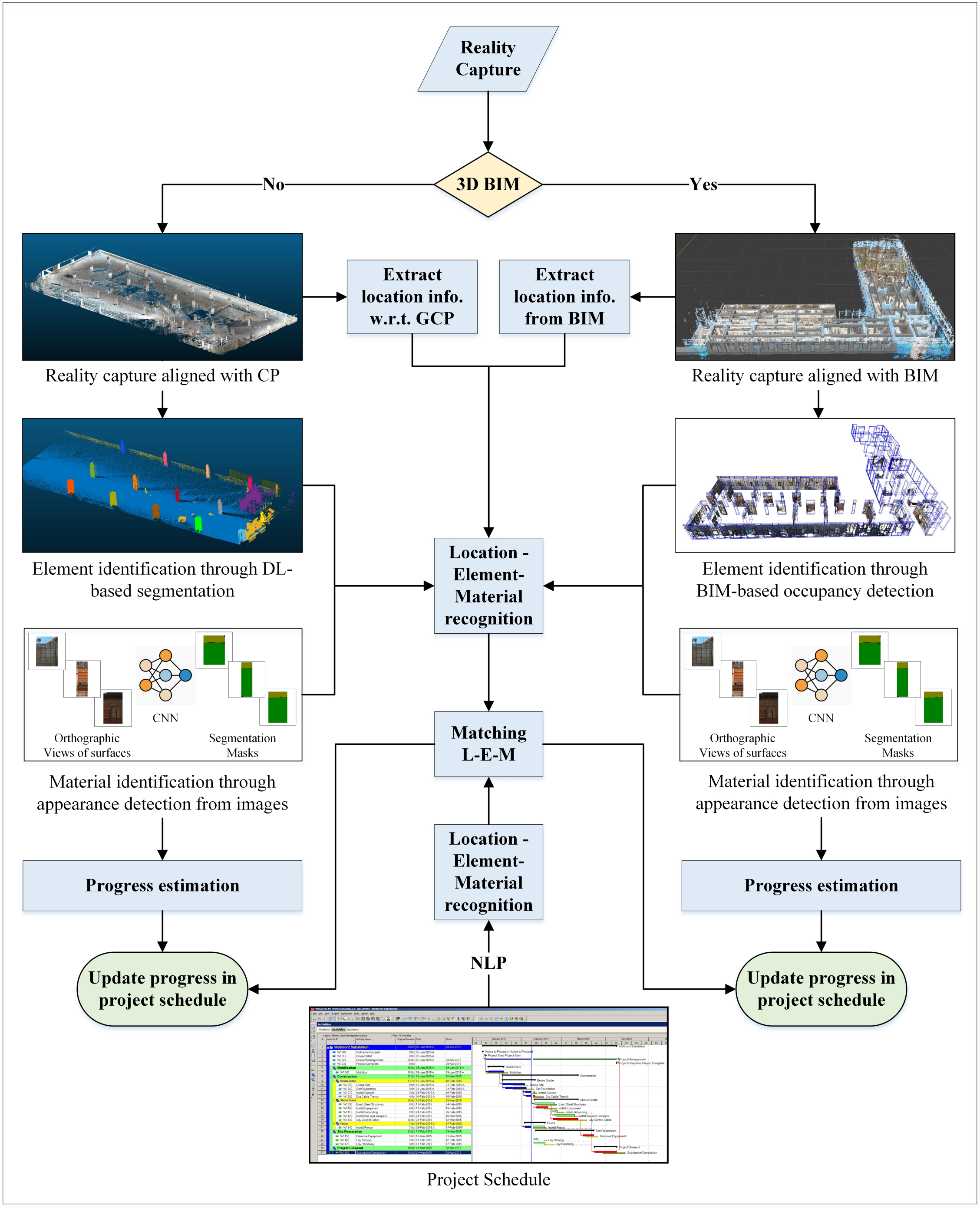

Reality Capture
3D BIM
No
Yes
Reality capture aligned with CP
Extract location info. w.r.t. GCP
Extract location info. from BIM
Reality capture aligned with BIM
Element identification through DL-based segmentation
Location - Element- Material recognition
Element identification through BIM-based occupancy detection
Orthographic Views of surfaces
CNN
Segmentation Masks
Material identification through appearance detection from images
Matching L-E-M
Orthographic Views of surfaces
CNN
Segmentation Masks
Material identification through appearance detection from images
Progress estimation
Location - Element- Material recognition
Progress estimation
NLP
Update progress in project schedule
Update progress in project schedule
Project Schedule

Figure 4.1: Proposed methodology for linking progress data with schedule activities

sponding points between the models and by solving seven degrees of freedom similarity transformation [69]. This process helps position the reality model in the BIM's world coordinate system. Hence, location information for any part of the reality model can be extracted from the corresponding BIM's location. Next, building elements are detected through a BIM-based occupancy check. While comparing the registered BIM and point cloud model, if the number of points available in a BIM element's bounding box exceeds a certain threshold, the presence of that element in the as-built model is confirmed. For identifying the ongoing tasks associated with a building element, on-site images are collected and registered with the 3D model. A deep-learning model for image segmentation is trained to detect construction tasks based on the appearance of the materials. Later, progress percentages of the tasks are determined by taking the ratio of the area of masks representing a construction task and the total surface area of the building element using Equation 3 and Table 3.3. The details of this method can be found in the previous chapter.

In the case of the non-availability of a BIM, the reality models are aligned with the pre-surveyed control points (CP). Any part of the reality model's location can be identified with a reference from the CPs. Reality models, in their raw form, lack semantic information. So, a point cloud segmentation approach is adopted to extract building element information from the reality model. First, a DL model for point cloud segmentation is trained to detect building elements such as beams, columns, walls, floors, and ceilings [59, 70]. Similar to the BIM-based approach, for detecting construction tasks, onsite images are collected and registered with the 3D reality model in the same coordinate system. Later, an image segmentation model is employed to detect the appearance of the construction materials to identify ongoing construction activity. This process helps in identifying the construction tasks associated with the material appearance. Finally, the progress sta-

doi:10.6342/NTU202302333

tus is reported by taking a ratio of the detected elements and the total number of elements expected for completing that activity using Equation 5.

$$Prog_{detect,i} = \frac{|\,\{Ele_{detect}\} \cap \{Ele_{expect}\}\,|_i}{|\,\{Ele_{expect}\}\,|_i} \times 100 \tag{5}$$

Here, $\{Ele_{expect}\}$ is the set of expected building elements for activity i, and $\{Ele_{detect}\}$ is the set of recognized elements for that activity.

For progress estimation of both cases, it is assumed that if the appearance of the material associated with the latest activity is detected, proceeding activities are deemed fully completed. Finally, the estimated progress is saved in a spreadsheet file with the location, building element, and material information.

4.1.2 Location, building element, and material recognition from schedule activity

A construction schedule is a detailed plan that outlines the timeline and milestones for completing a construction project. A well-designed construction schedule includes a breakdown of each project component (WBS) and lists activities to be completed in each component. These activity descriptions, along with their WBS name, generally contain basic information, such as the name of the material related to the construction task, building elements associated with that task, and the task's location. For example, in the activity description: "Erection of steel columns at floor 2," erection is the task, steel is the material name, columns are the building elements, and the location is floor 2. Named entity recognition (NER) is a task in NLP that identify and extract meaningful information (entities) from the text. Named entities are words or phrases that refer to specific information. In

doi:10.6342/NTU202302333

the case of construction schedule analysis, these entities are locations, building elements, and materials. In this research, two powerful transformer models developed for natural language processing are used, and their performance on schedule information extraction is compared. They are pre-trained Bidirectional Encoder Representations from Transformers (BERT) [26] and Generative Pre-trained Transformer-3 (GPT3)[20].

BERT is a language model developed by Google in 2018. It captures contextual word representations by considering both preceding and succeeding words. Pre-trained on vast amounts of text data, BERT learns rich word and sentence representations through masked language modeling. It can be fine-tuned for specific tasks, adapting its parameters to achieve high performance. BERT's bidirectional approach and Transformer architecture enable it to understand language nuances and capture contextual relationships effectively. It has revolutionized natural language processing and achieved state-of-the-art results in various tasks like text classification, named entity recognition, and sentiment analysis. BERT has become a widely used model, powering numerous language-related applications. In this study, a pre-trained BERT model is fine-tuned to detect construction schedule entities. The mechanism of the BERT model for schedule information extraction is shown in Figure 4.2. While preparing the training data, IOB tagging is used to ensure the BERT model recognizes named entities that may consist of single or multiple words. The IOB tagging scheme represents each word in a sentence as either Inside, Outside, or Beginning of an entity. An example is shown in Figure 4.3. Next WordPiece tokenization and post-tokenization label adjustment are made. Finally, the trained BERT model is used for identifying and extracting L-E-Ms from a given schedule activity by classifying embedding vector output from all of the tokens with three classes: location, building element, and material.

doi:10.6342/NTU202302333

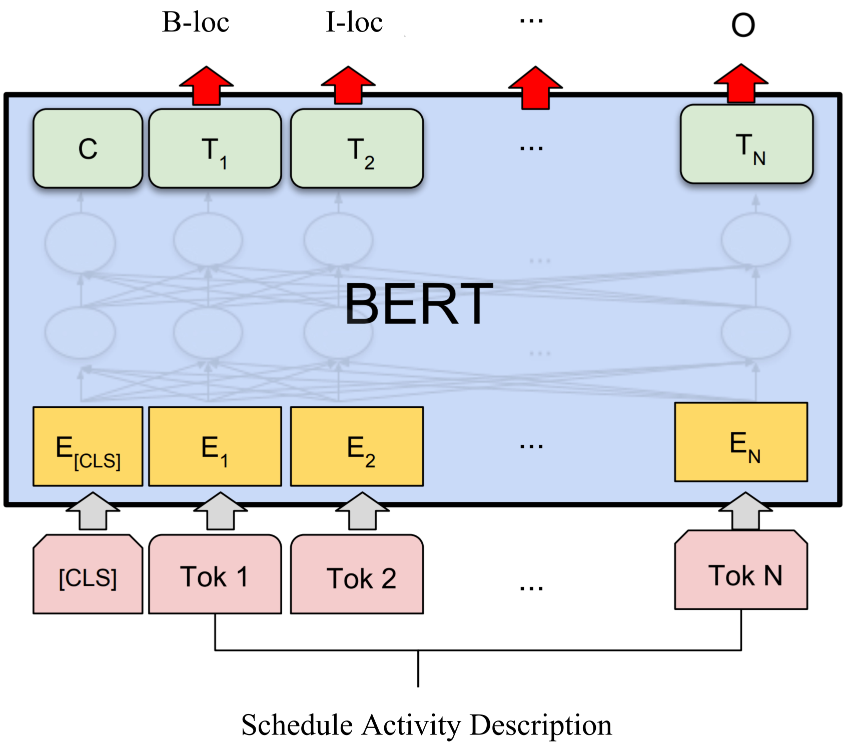


Figure 4.2: Information extraction from activity descriptions using BERT

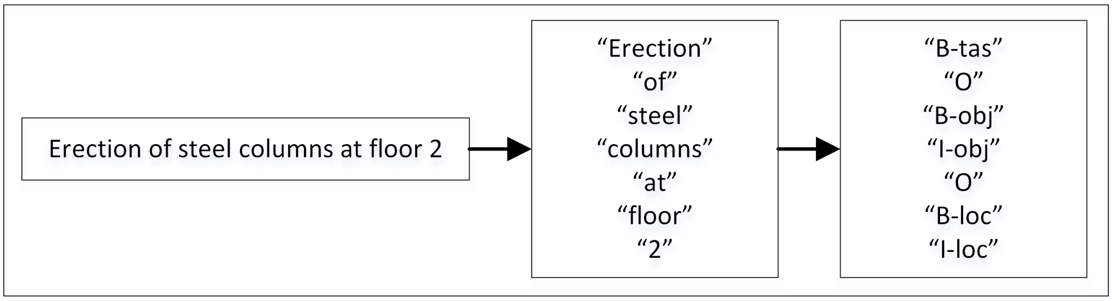


Figure 4.3: Example of data labeling for fine-tuning BERT model

doi:10.6342/NTU202302333

GPT-3 is an advanced language model developed by OpenAI. It belongs to the family of Transformer models and has gained significant attention for its impressive text generation capabilities. GPT-3 is a generative model, meaning it can generate coherent and contextually relevant text based on a given prompt. Unlike traditional language models that process text sequentially, GPT-3 utilizes a Transformer architecture that allows it to capture long-range dependencies and understand the context of words. It is pre-trained on an enormous amount of internet text data, enabling it to learn patterns, grammar, and semantic relationships. With 175 billion parameters, GPT-3 is one of the largest language models available. Its size enables it to generate highly coherent and contextually appropriate responses, making it capable of producing human-like text in a wide range of applications. In this study, GPT-3 model is used for extracting L-E-M information from construction schedule activities using a few-shot learning approach. To achieve this, along with target activity descriptions, a few examples (maximum three) of construction activity descriptions with L-E-M tags are promoted to the GPT-3 model. OpenAI's API key is used to request access for the pre-trained GPT-3 model named "text-davinci-003". The information extraction process using GPT-3 is shown in Figure 4.4

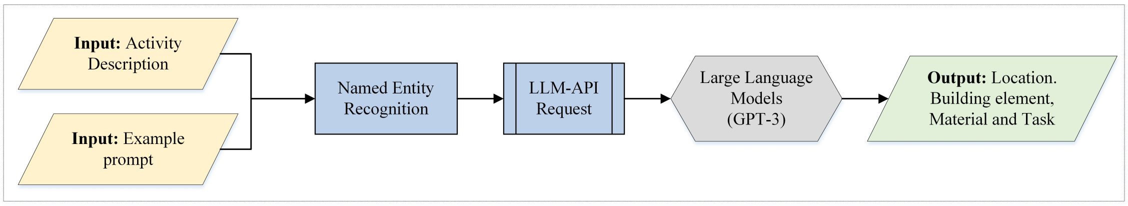

Input: Activity Description
Input: Example prompt
Named Entity Recognition
LLM-API Request
Large Language Models (GPT-3)
Output: Location. Building element, Material and Task

Figure 4.4: Information extraction from activity descriptions using GPT-3

4.1.3 Matching activity and progress data using L-E-M strings

Once progress data is stored in text format with location, building element, and material information, and the L-E-M strings are extracted from the scheduled activity, string

doi:10.6342/NTU202302333

matching techniques are used to map the progress data and the schedule activity. Algorithm 5 calculates the best matching score between activities and progress data. A sequential matching approach is adopted to match the location, building element, and materials strings sequentially from schedule and progress data. Matching scores are stored in matrices $[L], [E], and[M]$. Finally, an $argmax$ operation on the element-wise multiplication results of $[L], [E], and[M]$ matrices maps activities with their most appropriate progress counterpart.

Algorithm 5: Linking progress data with schedule activity

```
Input: Location ($A^l$), building element ($A^e$), and material ($A^m$)
       information from schedule activity
       Progress data with location ($P^l$), building element ($P^e$),
       and material ($P^m$) information
Output: Matching between activity and progress data
$[L]$ ← A matrix containing location matching score
$[E]$ ← A matrix containing building element matching score
$[M]$ ← A matrix containing material matching score
foreach activity $A$ in $Schedule$ do
    foreach progress $P$ in $Progress\ data$ do
        Calculate location matching score between $A^l$ and $P^l$ and update $[L]$
        Calculate element matching score between $A^e$ and $P^e$ and update $[E]$
        Calculate material matching score between $A^m$ and $P^m$ and update $[M]$
    end
end
$[F]$ ← A matrix containing final matching score
$[F] = [L] * [E] * [M]$
Best match = $argmax[F]$
return Best match
```

A fuzzy string matching using the Levenshtein Distance algorithm [9] and Cosine similarity using text embedding models [94] are tested in this study. Fuzzy string matching using the Levenshtein Distance algorithm is a technique used to compare and measure the similarity between two strings. The Levenshtein Distance, also known as the Edit Distance, calculates the minimum number of insertions, deletions, and substitutions required

doi:10.6342/NTU202302333

to transform one string into another. It is calculated using Equation 6:

$$\operatorname{lev}(A^s, P^s) = \begin{cases} \max(|A^s|, |P^s|) & \text{if } \min(|A^s|, |P^s|) = 0 \\ \min \begin{cases} \operatorname{lev}(\operatorname{tail}(A^s), \operatorname{tail}(P^s)) + 1, \\ \operatorname{lev}(A^s, \operatorname{tail}(P^s)) + 1, \\ \operatorname{lev}(\operatorname{tail}(A^s), P^s) + 1 \end{cases} & \text{otherwise} \end{cases} \tag{6}$$

where A^s and P^s are the strings from activity and progress data, respectively, and $tail(expression)$ is the expression without its first character. For example, $tail($“$floor3$”$)$ is “$loor3$”.

Another string-matching approach that uses text embedding and cosine similarity calculation is used in this study. Text embedding provides condensed representations of text that capture semantic information, facilitating efficient analysis and processing in natural language processing tasks, and cosine similarity calculates the cosine distance between a text pair in the n-dimensional vector space. Higher similarity values indicate better matching. In this approach, first, the extracted texts from activities and progress data are tokenized. Then the tokens and, subsequently, the whole text are vectorized using a large text embedding model named “e5-large-v2.” . $E5$ is a general-purpose text embedding model proposed by Microsoft’s research team for tasks requiring a single-vector representation. This model has 24 layers, and it creates an embedding with a size of 1024. The cosine distance between the text pairs is calculated using Equation 7.

$$Similarity = \cos(A, P) = \frac{A.P}{\|A\|\|P\|} = \frac{\sum_{i=1}^{n} A_i P_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2}\sqrt{\sum_{i=1}^{n} (P_i)^2}} \tag{7}$$

Steps for this string-matching approach are shown in Figure 4.5.

doi:10.6342/NTU202302333

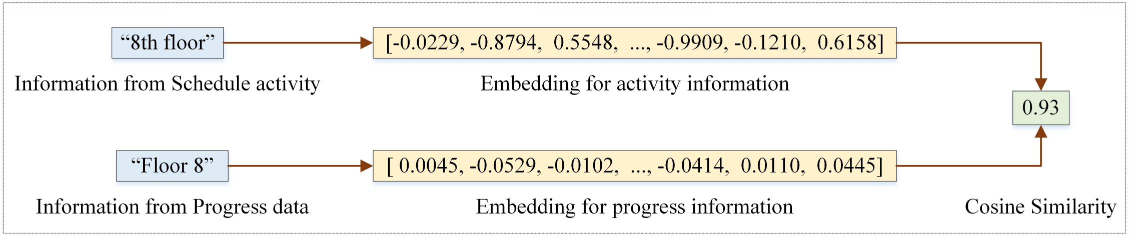


Figure 4.5: Distance-based matching between activity and progress information

Finally, based on the results of the matching algorithm, the progress data is mapped with the schedule activities. This mapping is further utilized for updating the project schedule with activity completion percentages derived from reality models.

4.2 Experiments and Results

The proposed methodology was tested and validated in two construction projects in Taiwan. Project A was an educational building construction project within National Taiwan University's main campus, and Project B was a hospital building construction project at the National Taiwan University Hospital campus. Project A was chosen to demonstrate the with-BIM workflow, and Project B was for the without-BIM workflow. Structural work was in progress during the validation period in both Projects A and B. Two different types of construction methodologies were adopted in these projects. While Project A followed in-situ concrete construction, Project B was constructed using pre-fabricated steel structures. The structural element construction on two levels (3rd and 4th floor) of Project A and three levels (12th, 13th, and 14th) of Project B was selected as case studies to test the performance of the proposed methodology. Each floor's 360-degree videos were recorded for reality capture at Project A. Later, as-built point clouds per floor were reconstructed using the photogrammetry technique explained in the previous chapter. For Project B, the floor-wise reality models were captured using a LiDAR and depth camera

doi:10.6342/NTU202302333

assembly, which could provide a clean point cloud with 3D coordinates and color values. 360-degree videos of each floor were also recorded for image-based point cloud reconstruction. For project A, reality models (point clouds) were registered with the 3D BIM in the same world coordinate system, and the location information was extracted from the BIM's location. Similarly, Project B's as-built point cloud models were registered with three to four nearest control points. The location information in the case of Project B was extracted by taking the reference from the nearest CP's location. As the BIM-based workflow used in project A followed the same occupancy-based approach described in the previous chapter for building element detection, those results are not discussed here. However, the performance of the point cloud segmentation model proposed for building element detection in the without-BIM workflow is discussed in the following subsection. As both workflows use the same image segmentation approach for appearance detection presented in the previous chapter, the results of appearance detection are also not discussed here exclusively. But the overall performance of the progress estimation for both with-BIM and without-BIM workflows is highlighted. Additionally, the performance of the language models for schedule information extraction and the matching results are presented in the following subsections.

4.2.1 Semantic segmentation of point clouds

An instance segmentation model named SoftGroup [92] and its upgraded version SoftGroup++ [91] were used to detect building elements from the point clouds. Unlike previous point cloud segmentation methods that use hard predictions and suffer from low overlaps and false positives, SoftGroup performs bottom-up soft grouping and top-down refinement. It allows each point to be associated with multiple classes, reducing the im-

doi:10.6342/NTU202302333

pact of semantic prediction errors and effectively suppresses false positive instances by categorizing them as background. Experimental results on various datasets demonstrate that SoftGroup outperforms the strongest prior method by a significant margin, achieving higher average precision (AP_{50}) scores. Considering these benefits, SoftGroup++ was chosen for this study.

The segmentation model was initially trained with Stanford 3D Indoor Scene Dataset (S3DIS) [11]; later, it was fine-tuned with a small amount of custom data from the precast building project. The S3DIS dataset consists of 3D point clouds captured from six large-scale indoor areas, including office spaces, conference rooms, and hallways. The point clouds are collected using a 3D laser scanner and represent the geometry and spatial layout of the scenes. Each point in the point cloud is associated with semantic labels, indicating the class or category of the corresponding surface or object in the indoor scene. The dataset provides labels for various objects and surfaces, such as walls, floors, ceilings, columns, beams, windows, doors, and furniture types commonly found in indoor environments.

The model's performance for building element detection (mainly structural components) was tested on a dataset created from Project B. The average segmentation results (average precision, recall, and mean intersection over union) of six classes (beam, column, ceiling, floor, wall, and clutter) are shown in Table 4.1. Also, Figure 4.6 shows the visualization of building element detection results and compares them with the ground truth annotations. The results confirm the satisfactory performance of the model for building element detection from raw point cloud data. Semantic and instance segmentation of building elements for a typical floor of Project B is shown in Figure 4.7.

doi:10.6342/NTU202302333

Table 4.1: Evaluation of the segmentation model for building element detection

Class	AP_50%	AP_25%	RC_50%	RC_25%	mIoU
Ceiling	1.000	1.000	1.000	1.000	52.2
Floor	1.000	1.000	1.000	1.000	97.8
Wall	0.576	0.576	0.875	0.875	80.1
Beam	0.483	0.673	0.577	0.712	79.1
Column	1.000	1.000	1.000	1.000	96.3
Clutter	0.111	0.111	0.111	0.111	48.5
Average	**0.695**	**0.755**	**0.813**	**0.883**	**75.7**

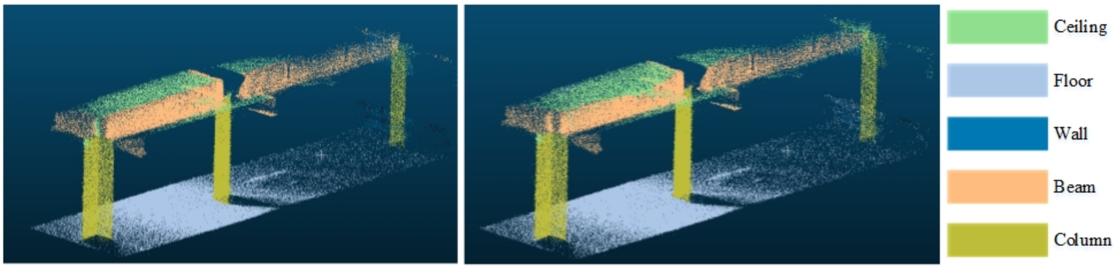
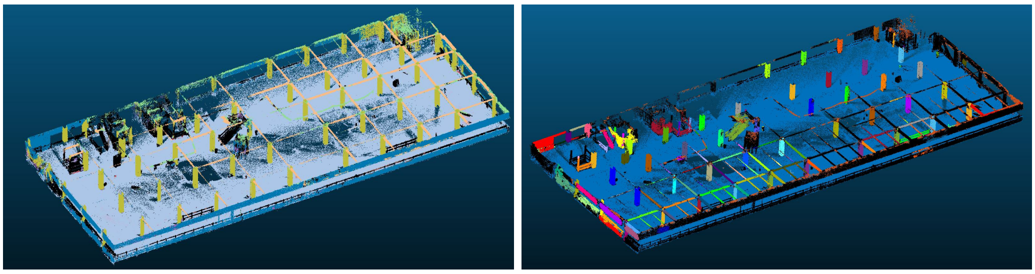


Figure 4.6: Visualization of semantic segmentation results: ground truth (left) and prediction (right)



(a) Semantic segmentation

(b) Instance segmentation

Figure 4.7: Point cloud segmentation of a typical floor (excluding ceiling)

4.2.2 Progress percentage estimation

For without-BIM workflow, the detected progress percentage per activity was calculated using Equation 5. $\{Ele_{expect}\}$ for each activity was estimated by taking the reference from the previously completed floor, and The actual progress percentage was estimated using Equation 8.

$$Prog_{actual,i} = \frac{|\ \{Ele_{actual}\} \cap \{Ele_{expect}\}\ |_i}{|\ \{Ele_{expect}\}\ |_i} \times 100 \tag{8}$$

$\{Ele_{actual}\}$ was determined by manual inspection at the construction sites. The appearance of the building elements associated with the activity was determined by the image segmentation model, and the estimated progress percentages were recorded against the task/material whose appearance was detected during the progress monitoring process. However, previous tasks' progress was recorded as 100% once the latest task's appearance was present. Progress estimation results of the case studies selected from Project B are shown in Table 4.2

For estimating the progress of with-BIM workflow, the method described in the previous chapter was used. Formulae shown in Table 3.3 were leveraged to estimate the progress of each construction activity. Similar to without-BIM workflow, the preceding activities' progress was recorded as 100% once the latest task's appearance was detected. Progress estimation results of the case studies selected from Project A are shown in Table 4.2

Finally, the progress data were stored in spreadsheet files along with location, building element, and material information.

4.2.3 Information extraction from schedule activities

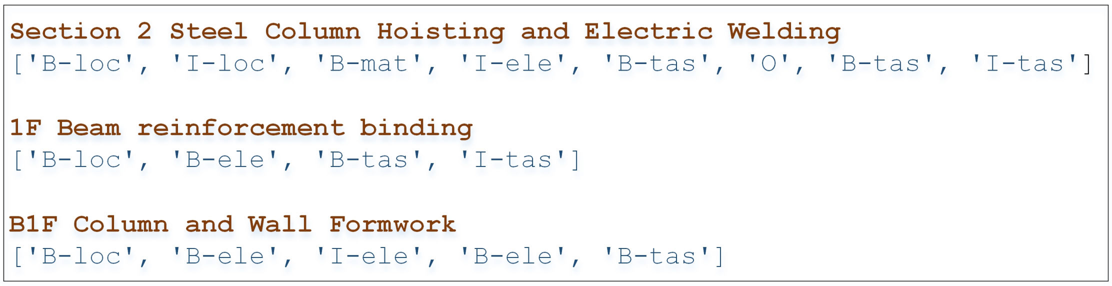
The BERT model was trained with 1351 schedule activities from three construction schedules to recognize L-E-Ms from the schedule activities. The model achieved 95% training and 92% validation accuracy. However, when the trained model was tested on a test dataset created by taking 100 sample activities from Project A and B's schedule, it could detect the location, building element, and materials from the activities with only 57% average accuracy. A few examples of correctly predicted results are shown in Figure 4.8.

doi:10.6342/NTU202302333

Table 4.2: Progress estimation of Project B (without-BIM)

Location	Building element	Material	Actual progress	Detected progress	Absolute error
12 Floor	Column	Steel	100%	100%	0.0%
12 Floor	Beam	Steel	100%	100%	0.0%
12 Floor	Roof	Steel	100%	100%	0.0%
12 Floor	Column	Fireproofing	100%	100%	0.0%
12 Floor	Beam	Fireproofing	100%	95%	5.0%
12 Floor	Roof	Fireproofing	100%	100%	0.0%
12 Floor	Wall	Concrete	100%	93%	7.0%
13 Floor	Column	Steel	100%	100%	0.0%
13 Floor	Beam	Steel	100%	100%	0.0%
13 Floor	Roof	Steel	100%	100%	0.0%
13 Floor	Column	Fireproofing	100%	100%	0.0%
13 Floor	Beam	Fireproofing	100%	97%	3.0%
13 Floor	Roof	Fireproofing	100%	100%	0.0%
13 Floor	Wall	Concrete	90%	84%	6.0%
14 Floor	Column	Steel	100%	100%	0.0%
14 Floor	Beam	Steel	100%	94%	6.0%
14 Floor	Roof	Steel	100%	100%	0.0%
14 Floor	Column	Fireproofing	100%	100%	0.0%
14 Floor	Beam	Fireproofing	100%	95%	5.0%
14 Floor	Roof	Fireproofing	100%	100%	0.0%
14 Floor	Wall	Concrete	65%	60%	5.0%

Possible causes for the BERT model's poor performance could be the small size training dataset and a considerable variation between the training and test data. Project schedules used for training were mostly from in-situ building construction projects, where activity descriptions differed somewhat from precast construction projects.



```
Section 2 Steel Column Hoisting and Electric Welding
['B-loc', 'I-loc', 'B-mat', 'I-ele', 'B-tas', 'O', 'B-tas', 'I-tas']

1F Beam reinforcement binding
['B-loc', 'B-ele', 'B-tas', 'I-tas']

B1F Column and Wall Formwork
['B-loc', 'B-ele', 'I-ele', 'B-ele', 'B-tas']
```

Figure 4.8: Examples of correct predictions by BERT

However, the GPT-3 model with few shot learning performed significantly better than the BERT model. It could achieve around 98% average testing accuracy while detecting

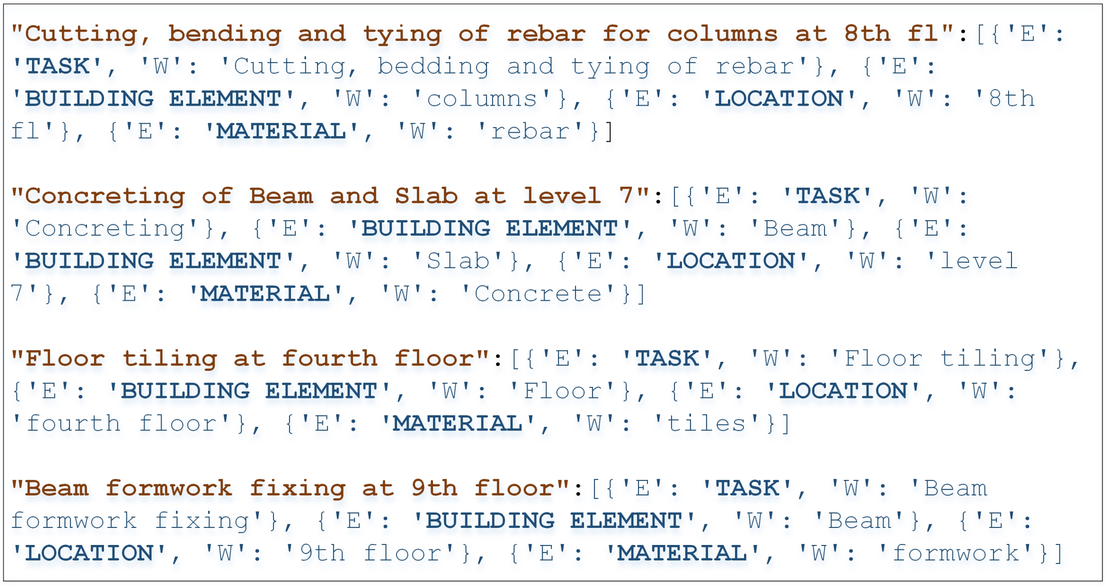
Table 4.3: Progress estimation of Project A (with-BIM)

Location	Building element	Material	Actual progress	Detected progress	Absolute error
3 Floor	Column	Reinforcement steel	100%	100%	0%
3 Floor	Wall	Reinforcement steel	100%	100%	0%
3 Floor	Column	Formwork	100%	100%	0%
3 Floor	Wall	Formwork	100%	100%	0%
3 Floor	Column	Concrete	100%	100%	0%
3 Floor	Wall	Concrete	100%	100%	0%
3 Floor	Beam	Reinforcement steel	100%	100%	0%
3 Floor	Beam	Formwork	100%	100%	0%
3 Floor	Slab	Reinforcement steel	100%	100%	0%
3 Floor	Slab	Formwork	100%	100%	0%
3 Floor	Slab	Concrete	100%	100%	0%
4 Floor	Column	Reinforcement steel	100%	100%	0%
4 Floor	Wall	Reinforcement steel	100%	100%	0%
4 Floor	Column	Formwork	85%	83%	2%
4 Floor	Wall	Formwork	85%	89%	4%
4 Floor	Column	Concrete	0%	0%	0%
4 Floor	Wall	Concrete	0%	0%	0%
4 Floor	Beam	Reinforcement steel	100%	100%	0%
4 Floor	Beam	Formwork	60%	52%	8%
4 Floor	Slab	Reinforcement steel	0%	0%	0%
4 Floor	Slab	Formwork	35%	38%	3%
4 Floor	Slab	Concrete	0%	0%	0%

location, building elements, and materials from the schedule activity descriptions of the test dataset. A few examples of GPT-3 predictions are shown in Figure 4.9. A comparison of testing accuracy between the BERT and GPT-3 models is shown in Table 4.4. Because of the significantly better performance of the GPT-3 model for all classes, it was selected for further extraction of L-E-M information from projects A and B's schedule activities.

Table 4.4: Comparison of testing accuracy between BERT and GPT-3 model

Entity	BERT	GPT-3
Location	0.75	0.98
Building Element	0.60	1.00
Material	0.35	0.95
Average	**0.57**	**0.98**



```
"Cutting, bending and tying of rebar for columns at 8th fl":[{'E':
'TASK', 'W': 'Cutting, bedding and tying of rebar'}, {'E':
'BUILDING ELEMENT', 'W': 'columns'}, {'E': 'LOCATION', 'W': '8th
fl'}, {'E': 'MATERIAL', 'W': 'rebar'}]

"Concreting of Beam and Slab at level 7":[{'E': 'TASK', 'W':
'Concreting'}, {'E': 'BUILDING ELEMENT', 'W': 'Beam'}, {'E':
'BUILDING ELEMENT', 'W': 'Slab'}, {'E': 'LOCATION', 'W': 'level
7'}, {'E': 'MATERIAL', 'W': 'Concrete'}]

"Floor tiling at fourth floor":[{'E': 'TASK', 'W': 'Floor tiling'},
{'E': 'BUILDING ELEMENT', 'W': 'Floor'}, {'E': 'LOCATION', 'W':
'fourth floor'}, {'E': 'MATERIAL', 'W': 'tiles'}]

"Beam formwork fixing at 9th floor":[{'E': 'TASK', 'W': 'Beam
formwork fixing'}, {'E': 'BUILDING ELEMENT', 'W': 'Beam'}, {'E':
'LOCATION', 'W': '9th floor'}, {'E': 'MATERIAL', 'W': 'formwork'}]
```

Figure 4.9: Examples of predictions by GPT-3

4.2.4 Mapping schedule activities and estimated progress

Schedule activities and progress data were mapped using the string-matching method described in Section 4.1.3. Location, building element, and material information from schedule activities were matched with progress data sequentially. The matching performance of the fuzzy matching technique and the text embedding-based matching approach is compared in Table 4.5. It is observed that the fuzzy matching score is purely based on text appearance, not on their semantic meanings. That's why, although the location of the "8th floor" and "7th floor" is semantically different, a high fuzzy matching score is calculated because of their similarity in appearance. On the other hand, text embedding captures high-semantic information, showing a high cosine similarity between "8th level" and "floor 8" despite their significant difference in text appearance. Because of the significantly better performance of the text embedding models, the same was applied for L-E-M matching required for mapping schedule activities and estimated progress. Mapping results for the case studies with or without BIM are shown in Figure 4.10 and 4.11, respectively.

Table 4.5: Comparison of string matching results

Text pair	Fuzzy Matching Score	Cosine Similarity Score
“8th level” & “floor 8”	15	**93**
“8th level” & “7th floor”	47	87
“8th floor” & “7th floor”	**89**	91

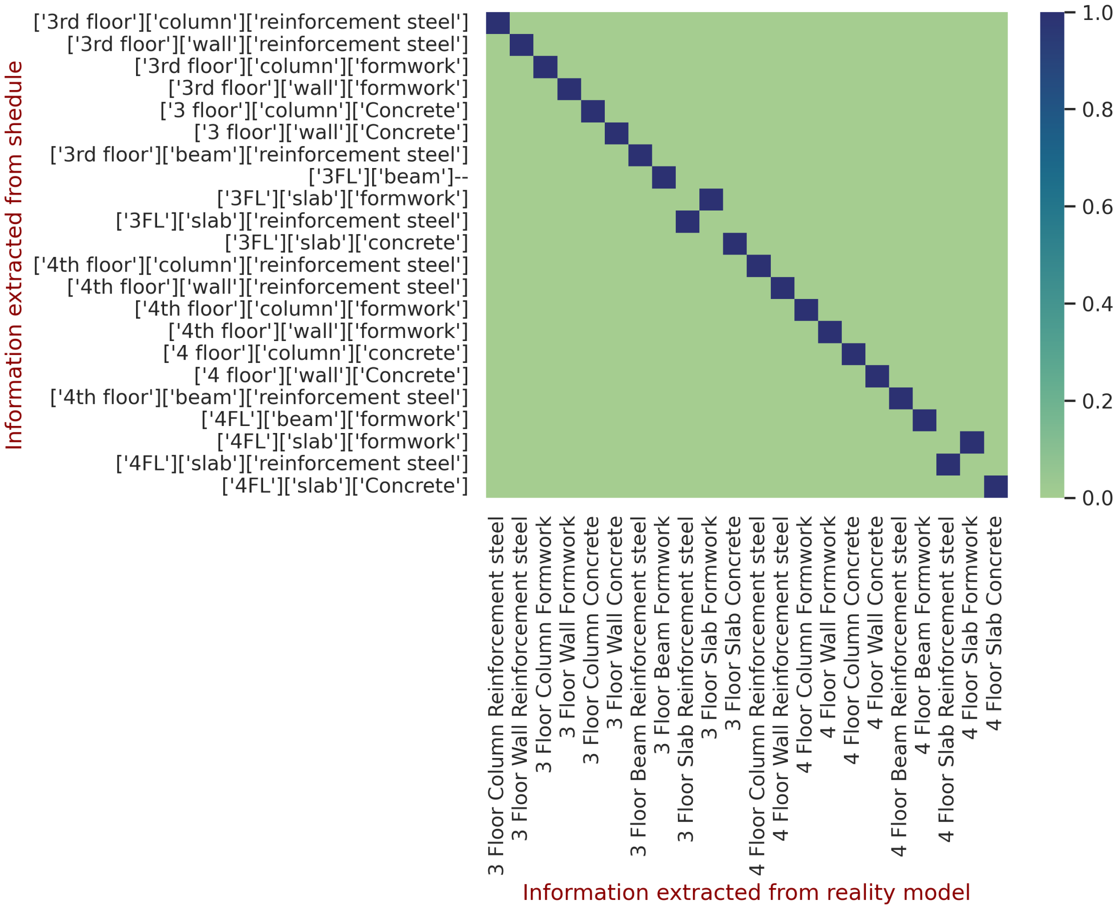


Figure 4.10: Matching results: with BIM case

4.2.5 Comparison between with and without BIM progress estimation workflows

The method with BIM relies on the availability of a complete and up-to-date BIM model, which allows for precise alignment of the reality model and the BIM, enabling accurate location information for each part of the construction project. The inherent semantic information in the BIM makes it easier to detect and identify various building elements,

doi:10.6342/NTU202302333

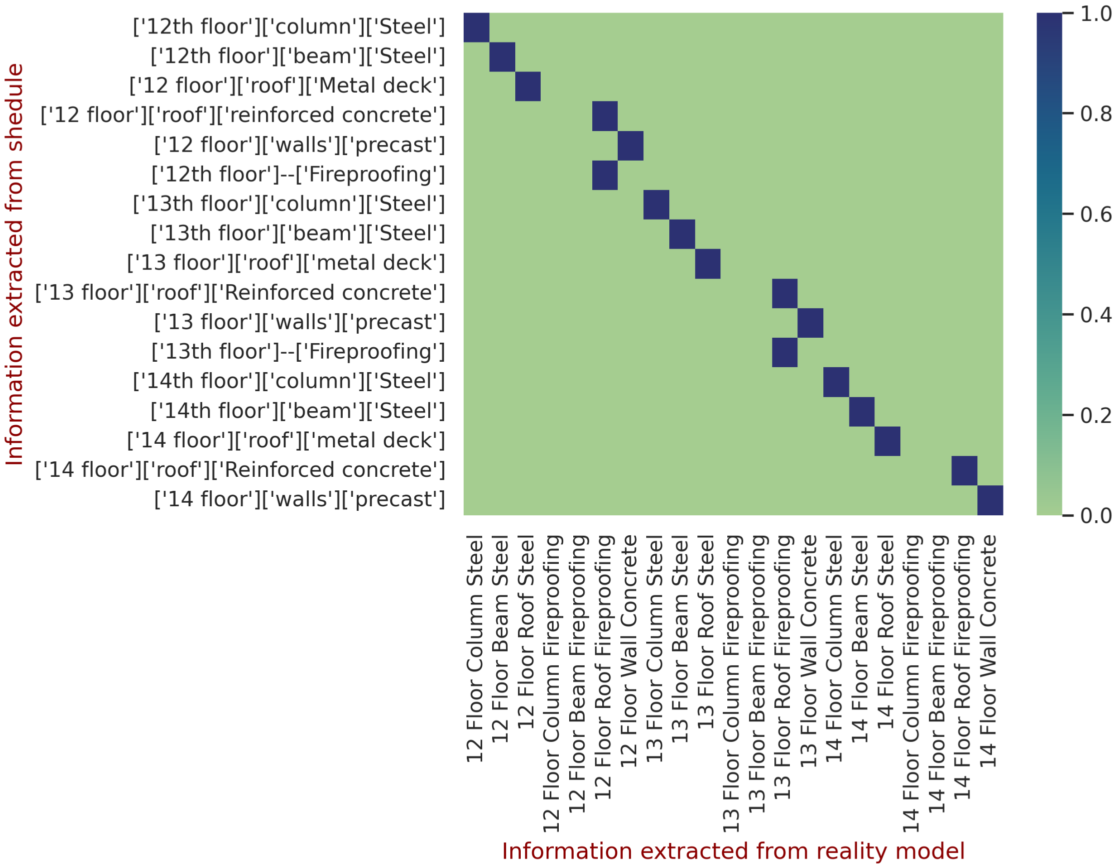


Figure 4.11: Matching results: without BIM case

doi:10.6342/NTU202302333

improving the accuracy of element detection through BIM-based occupancy checks. Additionally, the comprehensive nature of the BIM model facilitates better progress tracking, providing a holistic view of the entire project scope, which aids in efficient task monitoring and management. However, this approach has limitations, including the cost and effort involved in creating and maintaining the BIM model, which might not be feasible for all construction projects, particularly smaller ones or those with limited resources. Moreover, the manual registration process required for aligning the reality model with the BIM can be time-consuming and prone to human errors.

On the other hand, the method without BIM offers more versatility as it can be applied to construction projects where BIM is not available or practical. By aligning the reality model with pre-surveyed control points, this approach enables location identification without BIM dependency. This reduces project costs by eliminating the need for BIM creation and maintenance. However, without the inherent semantic information in a BIM model, this method relies on point cloud segmentation to extract building element information, which might result in less accurate element detection compared to the BIM-based approach. The scope of this method may also be limited to specific elements, such as beams, columns, walls, etc., rather than providing a comprehensive model of the entire project. Additionally, as this method completely relies on a learning-based approach, additional processing and training time would be required for element detection.

The choice between the two methods depends on the specific requirements of the construction project and the available resources. The BIM-based approach offers higher accuracy, semantic information, and comprehensive progress tracking but requires a functional BIM model. On the other hand, the method without BIM provides flexibility and reduced cost but may have limitations in semantic information, processing time require-

doi:10.6342/NTU202302333

ments and overall accuracy, making it suitable for projects where BIM is not available or cost-effective.

4.3 Summary

This part of the research introduces an innovative methodology for the automatic alignment of project schedules and reality models in construction progress updates. It focuses on detecting and matching locations, building elements, and materials (L-E-M) information from reality models and schedule activities. Two distinct workflows are presented, one leveraging 3D BIM and the other without. For detecting L-E-Ms from construction activity descriptions, NLP algorithms for Named Entity Recognition (NER) are employed, with GPT-3 exhibiting superior performance. The methodology incorporates text embedding-based models for accurate mapping of schedule information and progress estimation. Through on-site case studies, the methodology demonstrates its practical application for the automatic update of construction schedules with real-time progress information. While the current implementation focuses on structural elements, further research is needed to explore its potential application to other construction activities and processes, ensuring its scalability and effectiveness across diverse projects.

doi:10.6342/NTU202302333

doi:10.6342/NTU202302333

Chapter 5 Conclusion and future research directions

5.1 Conclusion

This research presents a method for activity-level automated construction progress monitoring integrating on-site visual data and construction schedules through digital twin construction. It is divided into two main parts. The first part proposes an activity-level progress monitoring system called ALPMS for schedule activity-level construction progress estimation and visualization, and the second part focuses on a project schedule update by linking activity information and estimated progress.

ALPMS takes construction images and 4D BIM as input and outputs progress percentages of construction activities associated with BIM elements. As-built point clouds reconstructed from site images are compared with as-planned BIM to detect components under construction. The percentage progress of construction activities related to the in-progress elements is then measured by segmenting orthographic views of the elements' faces. Two approaches: projective transformation, and NeRF-based novel view synthesis, are proposed to generate an equivalent ortho-view of element surfaces. A deep-learning model for semantic segmentation is implemented to detect polygon masks representing

doi:10.6342/NTU202302333

construction activities. The model is trained with a custom dataset comprised of six construction material classes. Finally, the activity-level progress status is visualized through color-coded BIM and semantically segmented as-built point clouds.

The framework is validated in 4 case studies prepared from two construction sites. Activity-level progress monitoring was conducted on these case studies with an average error of 5.63%. Factors that affect the performance of ALPMS are studied in detail. A series of tests are conducted to improve the synthetic image quality and the performance of the segmentation models. The NeRF model trained for individual elements, and the virtual camera placed at a distance equal to the average distance of the training cameras from the element's face synthesized the best quality ortho-view image. The segmentation model trained with 70% original and 30% synthetic data performed the best on synthetic image segmentation.

The system faces performance deficiency in the current implementation due to three open challenges: artifacts in the NeRF-generated ortho view image caused due manual data collection, static occlusion caused by the material stacking at the construction sites, and low-light conditions in the indoor environment. Some other research challenges that may interest future researchers are as follows. The proposed method is suitable for activities whose progress can be measured through area-based measurements. Activity-wise segmented as-built point cloud generated as an end product of this method can be used for volume-based partial progress estimation. Updating project schedules with activity-level partial progress details and predicting project completion dates could be tried in the future. As this method entirely relies on appearance-based progress detection, significant changes in the elements' surface appearance are needed for effective progress estimation. The construction image segmentation dataset can be expanded with various construction materials representing several other construction activities and different project settings, such as variable lighting and weather conditions, such as rainy, cloudy, and sunny. Synthetic image rendering with different backgrounds can be tried to augment the existing dataset.

The second part of this research proposes a novel methodology for automatically aligning project schedules and reality models for construction progress updates. It recommends detecting and matching locations, building elements, and materials (L-E-M) information from reality models and schedule activities. Two separate workflows are presented for detecting such information: with or without a 3D BIM. NLP algorithms for NER are proposed for detecting L-E-Ms from construction activity descriptions. Two transformer-based NLP models are tested and compared, and the Large Language Model: GPT-3 shows significantly higher performance in schedule information extraction. Two sting matching algorithms are tested for mapping schedule information with estimated progress. The text-embedding-based model shows high semantic understanding and higher matching accuracy. Later, progress schedules are updated with progress data. Finally, the on-site application of this methodology is demonstrated through case studies from two construction projects. The results confirm its applicability for an automatic update of the construction schedule with up-to-date progress information.

However, future studies need to address a few limitations for the full-scale application of this method. As of now, the method is tested on the construction of structural elements only. However, the applicability of this method in different construction activities and processes will be investigated in the future.

doi:10.6342/NTU202302333

5.2 Future research directions

This study can be further expanded and applied to improve several construction management practices. Three such applications are explained here: (a) real-time monitoring of construction projects, (b) productivity assessment by linking progress and resource data, and (c) predictive monitoring of construction projects.

5.2.1 Real-time monitoring of construction projects

Real-time monitoring of construction progress can give instant information about project status, which can help project managers make timely decisions and take corrective actions to avoid delays and cost overruns. Fig.5.1 suggests a workflow for real-time progress monitoring that leverages SLAM-based autonomous navigation, AI-powered progress detection, and AR-based progress visualization. The workflow proposes using a camera-equipped robotic system with an attached AR device, similar to the one used by [35]. The SLAM technology is used to create an initial map of the construction site by capturing images or videos of the environment and processing them to detect key points or features. The map is then used as a reference frame to estimate the position of the AR device within the environment. The initial map of the construction site is overlaid on the BIM to create a combined model of the as-built and as-planned environment. Alignment with BIM can improve localization accuracy and be used for progress monitoring [12]. As the AR device moves within the construction site, SLAM algorithms track its position and orientation, updating the combined model accordingly. The model can be updated in real-time as construction progresses. Then AI-powered image segmentation algorithm proposed for activity percent complete measurement can be applied to the BIM-registered

doi:10.6342/NTU202302333

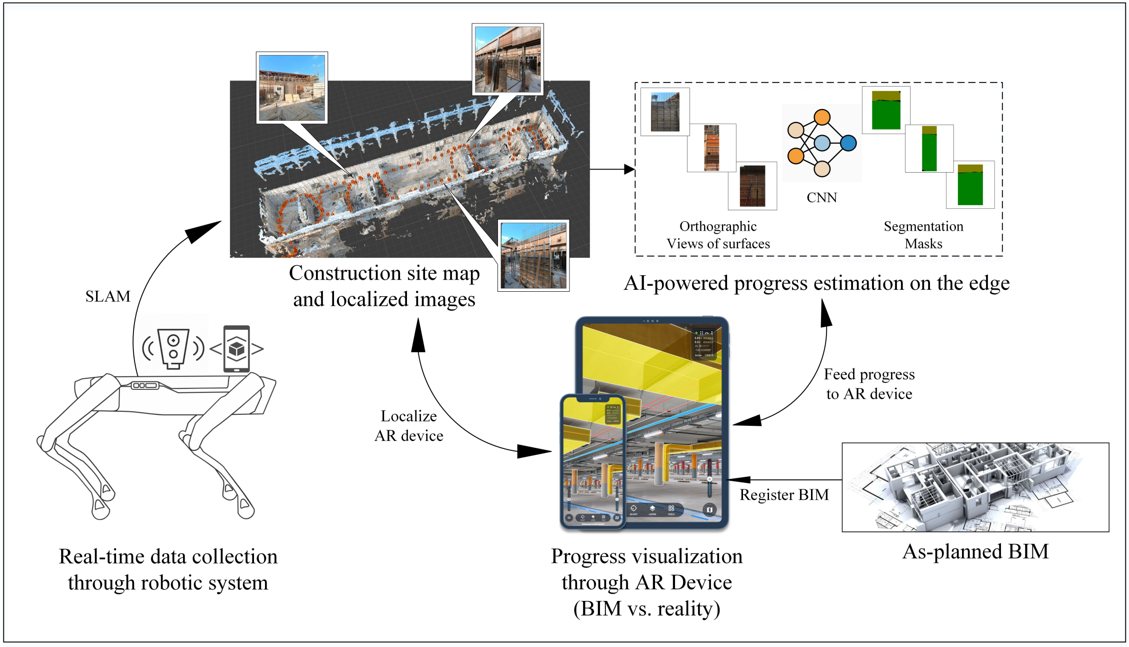


Figure 5.1: Real-time construction progress monitoring workflow

images to detect progress percentages. AI models can be deployed on-site through local-server, cloud computing, or edge computing devices such as NVIDIA Jetson Nano and Raspberry Pi 3B+ with Intel NCS [68]. AR-device can be used to display the progress, and the responsible project team member can annotate areas of the construction site that need attention. Multiple team members can access the same virtual model and view construction progress from different perspectives. This visualization enables real-time collaboration and decision-making, even when team members work remotely or on different shifts.

5.2.2 Productivity assessment by linking progress and resource data

Productivity, which measures resource utilization efficiency, is a key indicator for measuring project success at an intermediate time. By measuring the productivity of ongoing progress and comparing it against the plan, project controllers can get a sense of whether there will be a delay or cost overrun. Subsequently, corrective actions can be

doi:10.6342/NTU202302333

taken. Some previous studies investigated automated equipment productivity assessment from visual data for earthwork activities [22]. However, current automated progress monitoring methods cannot link the progress data (output) and resource data (input) to estimate the productivity of structural, finishing, and MEP works. Also, there are some limitations in AI-powered vision-based resource data analysis. These are limited datasets for activity recognition of construction resources, difficulties in workgroup identification, and group activity recognition. With the ability to integrate data from different sensors, DTC has the potential to address this open challenge. Future research should focus on solving these issues and developing a data integration platform for integrating progress data and resource data to facilitate productivity estimation in terms of progress per unit of resource utilized in unit time. Additionally, the combination of visual, audio, and kinematic sensors can be studied to complement vision-based methods [77].

The productivity data needs to be reviewed by the management team at specific intervals. It can be compared within and between projects. This comparison helps benchmark the current status and identify the causes of low and high performance. Low productivity not only indicates the use of excessive resources but also signals the need for improved collaboration and logistics between trade contractors. This understanding enables the management team to make decisions that reduce waste and increase value [64].

5.2.3 Predictive monitoring of construction projects

Predictive monitoring of construction projects is a strategy that uses data analysis and machine learning algorithms to predict potential issues and risks in a construction project before they occur. This approach can help project managers identify and address potential problems early on, reducing the risk of delays, cost overruns, and other issues arising

doi:10.6342/NTU202302333

during construction. Predictive monitoring involves collecting data from various sources, such as project schedules, resource utilization and work productivity metrics, progress monitoring metrics, and external data sources, like weather forecasts and economic indicators. This data is then analyzed using machine learning algorithms to identify patterns and predict future performance based on the project's current state.

For example, predictive monitoring can be used to analyze project schedules and identify potential delays based on historical data and other factors such as resource availability, weather conditions, and productivity levels. It can also be used to monitor progress in real-time and identify areas where work is falling behind schedule, or progress deviations are likely to occur.

The benefits of predictive monitoring in construction projects include improved project performance, reduced risk of delays and cost overruns, improved safety, and better decision-making. By identifying potential issues before they occur, project managers can take proactive measures to mitigate risks and keep the project on track, ultimately leading to the successful completion of the project. The DTC framework for closed-loop project control proposed in this study is expected to make the process of predictive monitoring in construction smoother. In the near future, with the availability of extensive data from closed-loop project operations, the DTC platforms will not only predict the risks in the construction process but also assist managers with potential mitigation actions proactively.

5.2.4 Quality control and quality assurance

As part of future work, the integration of quality control and quality assurance processes within the ALPMS framework presents a promising avenue for enhancing construc-

doi:10.6342/NTU202302333

tion management practices. By incorporating quality control mechanisms, construction managers can proactively monitor and inspect construction activities to ensure compliance with specified standards, reducing the likelihood of defects and rework. Concurrently, integrating quality assurance measures enables continuous evaluation of the ALPMS's performance, ensuring accurate completion percentage estimations and enhancing the reliability of the monitoring system. The synergistic integration of both quality control and quality assurance within ALPMS creates a closed-loop feedback system, where detected issues during quality control inspections can inform iterative improvements to the deep learning model and the overall progress monitoring methodology. Ultimately, this holistic approach not only enhances the accuracy and efficiency of construction progress monitoring but also fosters continuous improvement and greater project success.

doi:10.6342/NTU202302333

References

[1] M. Adorjan. OpenSFM - A collaborative Structure-from-Motion System. PhD thesis, Vienna University of Technology, 2016.

[2] E. Agapaki and I. Brilakis. Instance Segmentation of Industrial Point Cloud Data. Journal of Computing in Civil Engineering, 35(6):04021022, 2021.

[3] A. K. Ali, O. J. Lee, D. Lee, and C. Park. Remote indoor construction progress monitoring using extended reality. Sustainability (Switzerland), 13(4):1–24, 2021.

[4] S. Alizadehsalehi and I. Yitmen. Digital twin-based progress monitoring management model through reality capture to extended reality technologies (DRX). Smart and Sustainable Built Environment, 12(1):200–236, 2021.

[5] K. Amano, E. C. Lou, and R. Edwards. Integration of point cloud data and hyperspectral imaging as a data gathering methodology for refurbishment projects using building information modelling (BIM). Journal of Facilities Management, 17(1):57–75, 2019.

[6] F. Amer and M. Golparvar-Fard. Automatic Understanding of Construction Schedules: Part-of-Activity Tagging. Proceedings of the 2019 European Conference on Computing in Construction, 1:190–197, 2019.

doi:10.6342/NTU202302333

[7] F. Amer and M. Golparvar-Fard. Modeling dynamic construction work template from existing scheduling records via sequential machine learning. Advanced Engineering Informatics, 47(October 2020):101198, 2021.

[8] F. Amer, J. Hockenmaier, and M. Golparvar-Fard. Learning and critiquing pairwise activity relationships for schedule quality control via deep learning-based natural language processing. Automation in Construction, 134(October 2021):104036, 2022.

[9] F. Amer, Y. Jung, and M. Golparvar-Fard. Transformer machine learning language model for auto-alignment of long-term and short-term plans in construction. Automation in Construction, 132(March):103929, 2021.

[10] O. Angah and A. Y. Chen. Removal of occluding construction workers in job site image data using U-Net based context encoders. Automation in Construction, 119(July):103332, 2020.

[11] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese. 3d semantic parsing of large-scale indoor spaces. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, 2016.

[12] K. Asadi, H. Ramshankar, M. Noghabaei, and K. Han. Real-Time Image Localization and Registration with BIM Using Perspective Alignment for Indoor Monitoring of Construction. Journal of Computing in Civil Engineering, 33(5):04019031, 2019.

[13] O. C. Blender. Blender - a 3D modelling and rendering package. 2022.

[14] F. Bosché. Automated recognition of 3D CAD model objects in laser scans and cal-

culation of as-built dimensions for dimensional compliance control in construction. Advanced Engineering Informatics, 24(1):107–118, 2010.

[15] F. Bosché, M. Ahmed, Y. Turkan, C. T. Haas, and R. Haas. The value of integrating Scan-to-BIM and Scan-vs-BIM techniques for construction monitoring using laser scanning and BIM: The case of cylindrical MEP components. Automation in Construction, 49:201–213, 2015.

[16] F. Bosche and C. T. Haas. Automated retrieval of 3D CAD model objects in construction range images. Automation in Construction, 17(4):499–512, 2008.

[17] F. Bosche, C. T. Haas, and B. Akinci. Automated Recognition of 3D CAD Objects in Site Laser Scans for Project 3D Status Visualization and Performance Control. Journal of Computing in Civil Engineering, 23(6):311–318, 2009.

[18] A. Braun, S. Tuttas, A. Borrmann, and U. Stilla. A concept for automated construction progress monitoring using BIM-based geometric constraints and photogrammetric point clouds. Journal of Information Technology in Construction, 20(January):68–79, 2015.

[19] A. Braun, S. Tuttas, A. Borrmann, and U. Stilla. Improving progress monitoring by fusing point clouds, semantic data and computer vision. Automation in Construction, 116(May):103210, 2020.

[20] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCan-

doi:10.6342/NTU202302333

dlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners, 2020.

[21] J. Chen, Z. Kira, and Y. K. Cho. Deep Learning Approach to Point Cloud Scene Understanding for Automated Scan to 3D Reconstruction. Journal of Computing in Civil Engineering, 33(4):04019027, 2019.

[22] M. Y. Cheng, M. T. Cao, and C. K. Nuralim. Computer vision-based deep learning for supervising excavator operations and measuring real-time earthwork productivity. Journal of Supercomputing, (0123456789), 2022.

[23] P. R. de Almeida, M. Z. Solas, A. Renz, M. Bühler, P. Gerbert, S. Castagnino, and C. Rothballer. Shaping the Future of Construction: A Breakthrough in Mindset and Technology, may 2016.

[24] H. Deng, H. Hong, D. Luo, Y. Deng, and C. Su. Automatic Indoor Construction Process Monitoring for Tiles Based on BIM and Computer Vision. Journal of Construction Engineering and Management, 146(1):04019095, 2020.

[25] L. Deng and D. Yu. Deep learning: Methods and applications. Foundations and Trends® in Signal Processing, 7(3–4):197–387, 2014.

[26] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[27] A. Dimitrov and M. Golparvar-Fard. Vision-based material recognition for automated monitoring of construction progress and generating building information modeling from unordered site image collections. Advanced Engineering Informatics, 28(1):37–49, 2014.

doi:10.6342/NTU202302333

[28] Z. Dong, J. Wang, B. Cui, D. Wang, and X. Wang. Patch-based weakly supervised semantic segmentation network for crack detection. Construction and Building Materials, 258:120291, 2020.

[29] Q. Fang, H. Li, X. Luo, C. Li, and W. An. A sematic and prior-knowledge-aided monocular localization method for construction-related entities. Computer-Aided Civil and Infrastructure Engineering, 35(9):979–996, 2020.

[30] M. Golparvar-Fard, F. Peña-Mora, C. A. Arboleda, and S. Lee. Visualization of Construction Progress Monitoring with 4D Simulation Model Overlaid on Time-Lapsed Photographs. Journal of Computing in Civil Engineering, 23(6):391–404, 2009.

[31] M. Golparvar-Fard, F. Peña-Mora, and S. Savarese. Integrated Sequential As-Built and As-Planned Representation with D4AR Tools in Support of Decision-Making Tasks in the AEC/FM Industry. Journal of Construction Engineering and Management, 137(12):1099–1116, 2011.

[32] M. Golparvar-Fard, F. Peña-Mora, and S. Savarese. Automated Progress Monitoring Using Unordered Daily Construction Photographs and IFC-Based Building Information Models. Journal of Computing in Civil Engineering, 29(1):04014025, 2015.

[33] M. Grieves and J. Vickers. Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior

pages 85–113. Springer International Publishing, Cham, 2017.

[34] A. Gupta, R. Ramanath, J. Shi, and S. S. Keerthi. Adam vs. SGD: Closing the generalization gap on image classification. In OPT2021: 13th Annual Workshop on Optimization for Machine Learning, pages 1–7, 2021.

doi:10.6342/NTU202302333

[35] S. Halder, K. Afsari, J. Serdakowski, S. Devito, M. Ensafi, and W. Thabet. Real-Time and Remote Construction Progress Monitoring with a Quadruped Robot Using Augmented Reality. 2022.

[36] H. Hamledari, B. McCabe, and S. Davari. Automated computer vision-based detection of components of under-construction indoor partitions. Automation in Construction, 74:78–94, 2017.

[37] K. Han, J. Degol, and M. Golparvar-Fard. Geometry- and Appearance-Based Reasoning of Construction Progress Monitoring. Journal of Construction Engineering and Management, 144(2):04017110, 2018.

[38] K. K. Han and M. Golparvar-Fard. Appearance-based material classification for monitoring of operation-level construction progress using 4D BIM and site photologs. Automation in Construction, 53:44–57, 2015.

[39] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN, 2017.

[40] Y. Hong, H. Xie, E. Agapaki, and I. Brilakis. Graph-Based Automated Construction Scheduling without the Use of BIM. Journal of Construction Engineering and Management, 149(2):1–15, 2023.

[41] A. Ibrahim, M. Golparvar-Fard, and K. El-Rayes. Metrics and methods for evaluating model-driven reality capture plans. Computer-Aided Civil and Infrastructure Engineering, 37(1):55–72, 2022.

[42] W. Kentaro. Labelme: Image Polygonal Annotation with Python, 2016.

[43] H. Kim and C. Kim. 3D as-built modeling from incomplete point clouds using connectivity relations. Automation in Construction, 130(January):103855, 2021.

[44] S. Kim, S. Kim, and D. E. Lee. Sustainable application of hybrid point cloud and BIM method for tracking construction progress. Sustainability (Switzerland), 12(10):1–16, 2020.

[45] M. Kopsida and I. Brilakis. Real-Time Volume-to-Plane Comparison for Mixed Reality‑Based Progress Monitoring. Journal of Computing in Civil Engineering, 34(4):04020016, 2020.

[46] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn. Digital Twin in manufacturing: A categorical literature review and classification. IFAC-PapersOnLine, 51(11):1016–1022, 2018.

[47] C. Kropp, C. Koch, and M. König. Interior construction state recognition with 4D BIM registered image sequences. Automation in Construction, 86(October 2017):11–32, 2018.

[48] M. Lamba and K. Mitra. Restoring Extremely Dark Images in Real Time. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 3486–3496, 2021.

[49] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. Nature, 521(7553):436–444, 2015.

[50] L. Li, S. Aslam, A. Wileman, and S. Perinpanayagam. Digital Twin in Aerospace Industry: A Gentle Introduction. IEEE Access, 10:9543–9562, 2022.

[51] J. J. Lin and M. Golparvar-Fard. Visual and Virtual Production Management System for Proactive Project Controls. Journal of Construction Engineering and Management, 147(7):04021058, 2021.

doi:10.6342/NTU202302333

[52] J. J. Lin, . J. Y. Lee, and M. Golparvar-Fard. Exploring the Potential of Image-Based 3D Geometry and Appearance Reasoning for Automated Construction Progress Monitoring. In Computing in Civil Engineering 2019, pages 162–170, 2019.

[53] Z.-H. Lin, A. Y. Chen, and S.-H. Hsieh. Temporal image analytics for abnormal construction activity identification. Automation in Construction, 124:103572, 2021.

[54] C. Liu, S. M.E. Sepasgozar, S. Shirowzhan, and G. Mohammadi. Applications of object detection in modular construction based on a comparative evaluation of deep learning algorithms. Construction Innovation, 22(1):141–159, 2022.

[55] L. Liu, R.-J. Yan, V. Maruvanchery, E. Kayacan, I.-M. Chen, and L. K. Tiong. Transfer learning on convolutional activation feature as applied to a building quality assessment robot. International Journal of Advanced Robotic Systems, 14(3):1729881417712620, 2017.

[56] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. 7th International Conference on Learning Representations, ICLR 2019, 2019.

[57] X. Luo, H. Li, H. Wang, Z. Wu, F. Dai, and D. Cao. Vision-based detection and visualization of dynamic workspaces. Automation in Construction, 104:1–13, 2019.

[58] J. W. Ma, T. Czerniawski, and F. Leite. Semantic segmentation of point clouds of building interiors with deep learning: Augmenting training datasets with synthetic BIM-based point clouds. Automation in Construction, 113(February):103144, 2020.

[59] R. Maalek, D. D. Lichti, and J. Y. Ruwanpura. Automatic recognition of common structural elements from point clouds for automated progress monitoring and

dimensional quality control in reinforced concrete construction. Remote Sensing, 11(9), 2019.

[60] J. Martin. Productivity in the construction industry, UK: 2021, An investigation into productivity growth and its drivers for the UK construction industry. Office for National Statistics, pages 1–20, 2021.

[61] N. D. McKay and P. J. Besl. A method for registration of 3-d shapes. IEEE Transactions of Pattern Analysis and Machine Intelligence, 14(2):239–256, 1992.

[62] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 12346 LNCS:405–421, 2020.

[63] T. Müller, A. Evans, C. Schied, and A. Keller. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. ACM Transactions on Graphics, 41(4), 2022.

[64] D. Murguia, Q. Chen, A. Rathnayake, T. Jansen Van Vuuren, V. Vilde, C. Middleton, and L. O'rourke. Digital Measurement of Construction Performance: Data-to-dashboard Strategy. In World Building Congress 2022, pages 1–10, 2022.

[65] N. O'Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. V. Hernandez, L. Krpalkova, D. Riordan, and J. Walsh. Deep learning vs. traditional computer vision. In K. Arai and S. Kapoor, editors, Advances in Computer Vision, pages 128–144, Cham, 2020. Springer International Publishing.

[66] H. Omar, L. Mahdjoubi, and G. Kheder. Towards an automated photogrammetry-

based approach for monitoring and controlling construction site activities. Computers in Industry, 98:172–182, 2018.

[67] T. Omar and M. L. Nehdi. Data acquisition technologies for construction progress tracking. Automation in Construction, 70:143–155, 2016.

[68] A. Pal and S. H. Hsieh. Deep-learning-based visual data analytics for smart construction management. Automation in Construction, 131(August):103892, 2021.

[69] A. Pal, J. J. Lin, and S.-H. Hsieh. A Framework for Automated Daily Construction Progress Monitoring Leveraging Unordered Site Photographs. In Proc. of ASCE International Conference on Computing in Civil Engineering 2021, 2022.

[70] J. Park and Y. K. Cho. Point Cloud Information Modeling: Deep Learning−Based Automated Information Modeling Framework for Point Cloud Data. Journal of Construction Engineering and Management, 148(2):1–14, 2022.

[71] V. H. Pham, P. H. Chen, A. Pal, Christian, and S. H. Hsieh. Automatic extraction of daily concrete requirements from 3D BIM and project schedules. AIP Conference Proceedings, 2428(November), 2021.

[72] F. Pour Rahimian, S. Seyedzadeh, S. Oliver, S. Rodriguez, and N. Dawood. On-demand monitoring of construction projects through a game-like hybrid application of BIM and machine learning. Automation in Construction, 110(August 2019):103012, 2020.

[73] Project Management Institute. A Guide to the project management body of knowledge (PMBOK® guide). 6th ed. 6th edition, 2017.

[74] Z. Pučko, N. Šuman, and D. Rebolj. Automated continuous construction progress

doi:10.6342/NTU202302333

monitoring using multiple workplace real time 3D scans. Advanced Engineering Informatics, 38(October 2017):27–40, 2018.

[75] M. Purdy and P. Daugherty. How AI Boosts Industry Profits and Innovation, 2017.

[76] N. Puri and Y. Turkan. Bridge construction progress monitoring using lidar and 4D design models. Automation in Construction, 109(September 2019):102961, 2020.

[77] A. Rasul, J. Seo, and A. Khajepour. Development of integrative methodologies for effective excavation progress monitoring. Sensors (Switzerland), 21(2):1–25, 2021.

[78] R. Ren and J. Zhang. An Integrated Framework to Support Construction Monitoring Automation Using Natural Language Processing and Sensing Technologies. Computing in Civil Engineering 2021 - Selected Papers from the ASCE International Conference on Computing in Civil Engineering 2021, pages 1101–1109, 2021.

[79] D. Roberts, W. T. Calderon, S. Tang, and M. Golparvar-Fard. Vision-based construction worker activity analysis informed by body posture. Journal of Computing in Civil Engineering, 34(4):04020017, 2020.

[80] R. Romero-Jarén and J. J. Arranz. Automatic segmentation and classification of BIM elements from point clouds. Automation in Construction, 124(January 2020), 2021.

[81] R. Sacks, I. Brilakis, E. Pikas, H. S. Xie, and M. Girolami. Construction with digital twin information systems. Data-Centric Engineering, 1(6), 2020.

[82] J. L. Schonberger and J. M. Frahm. Structure-from-Motion Revisited. Proceedings

of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-Decem:4104–4113, 2016.

[83] H. Son and C. Kim. 3D structural component recognition and modeling method using color and 3D data for construction progress monitoring. Automation in Construction, 19(7):844–854, 2010.

[84] H. Son, C. Kim, and C. Kim. Automated Color Model–Based Concrete Detection in Construction-Site Images by Using Machine Learning Algorithms. Journal of Computing in Civil Engineering, 26(3):421–433, 2012.

[85] H. Son, C. Kim, and Y. Kwon Cho. Automated Schedule Updates Using As-Built Data and a 4D Building Information Model. Journal of Management in Engineering, 33(4):04017012, 2017.

[86] R. Szeliski. Computer Vision : Algorithms and Applications 2nd Edition. Springer, pages XXII, 925, 2021.

[87] M. Trzeciak, K. Pluta, Y. Fathy, L. Alcalde, S. Chee, A. Bromley, I. Brilakis, and P. Alliez. ConSLAM: Periodically Collected Real-World Construction Dataset for SLAM and Progress Monitoring. In European Conference on Computer Vision Workshops, pages 317–331, Tel Aviv, Israel, 2023. Springer.

[88] Y. Turkan, F. Bosche, C. T. Haas, and R. Haas. Automated progress tracking using 4D schedule and 3D sensing technologies. Automation in Construction, 22:414–421, 2012.

[89] Y. Turkan, F. Bosché, C. T. Haas, and R. Haas. Toward Automated Earned Value Tracking Using 3D Imaging Tools. Journal of Construction Engineering and Management, 139(4):423–433, 2013.

[90] K. P. Venkatesh, M. M. Raza, and J. C. Kvedar. Health digital twins as tools for precision medicine: Considerations for computation, implementation, and regulation. npj Digital Medicine, 5(1):150, 2022.

[91] T. Vu, K. Kim, T. M. Luu, T. Nguyen, J. Kim, and C. D. Yoo. Scalable softgroup for 3d instance segmentation on point clouds, 2022.

[92] T. Vu, K. Kim, T. M. Luu, X. T. Nguyen, and C. D. Yoo. SoftGroup for 3D Instance Segmentation on 3D Point Clouds. In CVPR, 2022.

[93] B. Wang, C. Yin, H. Luo, J. C. Cheng, and Q. Wang. Fully automated generation of parametric BIM for MEP scenes based on terrestrial laser scanning data. Automation in Construction, 125(May 2020):103615, 2021.

[94] L. Wang, N. Yang, X. Huang, B. Jiao, L. Yang, D. Jiang, R. Majumder, and F. Wei. Text embeddings by weakly-supervised contrastive pre-training, 2022.

[95] T. H. Wang, A. Pal, J. J. Lin, and S.-H. Hsieh. Construction photo localization in 3D reality models for vision-based automated daily project monitoring. Journal of Computing in Civil Engineering, (Under review), 2023.

[96] Z. Wang, H. Li, and X. Yang. Vision-based robotic system for on-site construction and demolition waste sorting and recycling. Journal of Building Engineering, 32:101769, 2020.

[97] W. Wei, Y. Lu, T. Zhong, P. Li, and B. Liu. Integrated vision-based automated progress monitoring of indoor construction using mask region-based convolutional neural networks and BIM. Automation in Construction, 140(1239):104327, 2022.

[98] Y. Wu, H. Kim, C. Kim, and S. H. Han. Object Recognition in Construction-Site Images Using 3D CAD-Based Filtering. Journal of Computing in Civil Engineering, 24(1):56–64, 2010.

[99] Y. Xu, X. Shen, and S. Lim. CorDet: Corner-Aware 3D Object Detection Networks for Automated Scan-to-BIM. Journal of Computing in Civil Engineering, 35(3):04021002, 2021.

[100] J. Yang, M.-W. Park, P. A. Vela, and M. Golparvar-Fard. Construction performance monitoring via still images, time-lapse photos, and video streams: Now, tomorrow, and the future. Advanced Engineering Informatics, 29(2):211–224, 2015. Infrastructure Computer Vision.

[101] L. Yuan, J. Guo, and Q. Wang. Automatic classification of common building materials from 3D terrestrial laser scan data. Automation in Construction, 110(December 2019):103017, 2020.

[102] X. Zhang, N. Bakis, T. C. Lukins, Y. M. Ibrahim, S. Wu, M. Kagioglou, G. Aouad, A. P. Kaka, and E. Trucco. Automating progress measurement of construction projects. Automation in Construction, 18(3):294–301, 2009.

[103] X. Zhao, K.-W. Yeoh, and D. K. H. Chua. Extracting Construction Knowledge from Project Schedules Using Natural Language Processing. In The 10th International Conference on Engineering, Project, and Production Management, pages 197–211, 2019.

[104] Z. Zhu and I. Brilakis. Parameter optimization for automated concrete detection in image data. Automation in Construction, 19(7):944–953, 2010.

doi:10.6342/NTU202302333