



Investigating low-cost semi-automatic Digital Twin generation for South African deep-level mines

WJ Kilian

 **orcid.org/0000-0002-9722-3627**

Thesis accepted in fulfilment of the requirements for the degree
*Doctor of Philosophy in Engineering with Computer and
Electronic Engineering* at the North-West University

Promoter: Dr JN de Meyer

Co-promoter: Dr JC Vosloo

Graduation: May 2026

Abstract

Title: Investigating low-cost semi-automatic Digital Twin generation for South African deep-level mines
Author: Wilhelm Johannes Kilian
Promoter: Dr JN de Meyer
Co-promoter: Dr JC Vosloo
Degree: PhD in Engineering with Computer and Electronic Engineering
Keywords: Industry 4.0, Digital Twin, localisation and mapping, deep-level mining

The South African deep-level mining industry is shifting towards the Industry 4.0 paradigm, which is the digitisation of complex industrial systems through the implementation and integration of advanced technologies. This paradigm comprises predictive maintenance, real-time remote monitoring and control, and process optimisation, among others. However, financial constraints slow the adoption of the paradigm in the mining industry, restricting benefits.

One of the tools enabled by the Industry 4.0 paradigm is Digital Twins (DTs). However, DTs are time-consuming to create manually and existing semi-automatic generation methods, such as Light Detection and Ranging (LiDAR), are typically expensive and require specialised equipment. In contrast, studies from other industries indicate that low-cost localisation and mapping techniques can be utilised to enable the efficient creation of DTs.

Therefore, there is a need for a low-cost semi-automatic DT generation method to increase the adoption of Industry 4.0 in the South African deep-level mining industry. By investigating the effectiveness of low-cost localisation and mapping techniques, the financially constrained mining sector can take a step closer towards feasible, fully automatic DT generation.

This study proposes a method for low-cost semi-automatic DT generation consisting of three parts. Firstly, a localisation method for offline environments that utilises an inertial measurement unit typically found in mobile devices. Secondly, a method to extract information automatically from images captured inside deep-level mines using image segmentation, trained by utilising a synthetic dataset generator. Lastly, the integration and extension of the previous two parts to create the low-cost semi-automatic DT generation method.

Each method was independently verified to ensure the accuracy of each underlying step and the overall proposed method. The proposed Kalman Filter-based inertial navigation method performed well in its verification. An error of only 2.4 m was obtained for a total walking distance of 80.6 m, with peak signal-to-noise ratio (PSNR) values of 12.84 dB, 35.14 dB, and 19.19 dB for the accelerometer, gyroscope, and magnetometer data.

The verification of the proposed deep-learning image segmentation method, which includes a synthetic dataset generator, also produced promising results. Unseen real-world images were segmented with an Intersection over Union (IoU) score of 34.15 and a Panoptic Quality (PQ) score of 32.41. A recent survey study found an average IoU of 39.3 and an average PQ of 40.31 from the included methods trained using a real-world dataset. The slightly lower performance achieved in this study can be attributed to the reality gap of using a synthetic dataset.

The verification of the overall proposed method showed PSNR results of 12.30 dB, 36.16 dB, and 17.99 dB for the path estimation. The captured images also showed successful segmentation of objects with high confidence scores of classes from the synthetic dataset created. The captured images also showed successful segmentation of objects with high confidence scores of classes from the created synthetic dataset.

The validation of the overall proposed method was done using a case study South African deep-level mine. The results obtained from the five sample sets showed successful path estimation, with PSNR results of 15.31 dB, 39.75 dB, and 16.86 dB, similar to the verification. The image segmentation also produced high-performing results, even in low-light conditions encountered in the deep-level mine.

Further improvements can be made to increase the robustness of each of the proposed methods. However, this research shows that the proposed modular SLAM approach is a feasible method for low-cost semi-automatic DT generation, and it creates a stepping stone towards fully automatic low-cost DT generation.

Acknowledgements

To my wonderful wife, Nerinda, you fill my life with so much love, laughter, and joy. I reflect on the time we have spent together over many years, and I cannot wait to experience the next chapter of our lives together, as our little one will soon join us. My wife, I love you to the stars and back – and then a thousand light-years more.

To my parents, Thomas and Martie, without you none of this would have been possible. Both of you provided Marelé and myself with great opportunities and many valuable life lessons. Dad, your determination remains a force to be reckoned with. During uncertain and difficult times, your example gives me the determination to push on.

Mom, your endless support and love gave me the confidence to pursue my dreams and build a life that I would be proud to call my own. Your example of kindness and humour guides me to cherish and enjoy those important memories that we build over time.

To my sister, Marelé, since I have started pursuing this degree, you have become a mother with little Mieka born not so long ago. Thank you for reminding me to reflect on my journey so far, to see the growth behind me as well as the possibilities ahead. It is heart-warming to see the amazing mother you have become.

To all of my colleagues, some of you who have become wonderful friends, thank you for all of the coffee sessions and words of wisdom. Your insight and support have helped me keep a level head during uncertain times and your friendship gave me a steady hand during the reviews and revisions of this thesis.

To Prof. E. H. Mathews, thank you for providing me the opportunity to complete my master's study and to pursue this doctoral study through IPGIP. Thank you to ETA Operations (Pty) Ltd, for the financial support for both of my postgraduate studies over the years.

Finally, to my grandfather, Oupa Wilhelm, thank you for the person you were in my life during not only those difficult university years, but also the formative years before. Your kindness, humour, and wise words remain fond memories that I will always cherish. P.S. Remember the day you asked me about a girl and I could not stop smiling? Yes, Nerinda was, and remains forever, the one.

Publication resulting from the research

A journal article based on the research in this thesis was published in *Frontiers in Artificial Intelligence* on 13 December 2024 [1]:

W. J. Kilian, J. Prinsloo, J. Vosloo, and S. Taljaard, “A synthetic segmentation dataset generator using a 3D modeling framework and raycaster: a mining industry application,” *Frontiers in Artificial Intelligence*, vol. 7, 2024, doi: 10.3389/frai.2024.1453931.

The published manuscript, inserted as Appendix 1, presents and discusses the first novel contribution of this study from Section 1.9.1.

Contents

Abstract	i
Acknowledgements	iii
Publication resulting from the research	iv
List of Figures	vii
List of Tables	viii
Table of Abbreviations	ix
1 Introduction	1
1.1 Background	1
1.2 Literature analysis strategy	7
1.3 Digital Twins	8
1.4 Localisation and mapping	16
1.5 Kinematic and perceptual methods	22
1.6 Summary of literature analysis	29
1.7 Problem statement and need for the study	31
1.8 Study objectives	32
1.9 Novel contributions	32
1.10 Research design	35
1.11 Thesis overview	36
2 Methodology	37
2.1 Scope definition	37
2.2 Methodology overview	37
2.3 Inertial navigation	38
2.4 Image segmentation	45
2.5 Low-cost semi-automatic DT generation	54
2.6 Summary	55
3 Results	58
3.1 Verification and validation strategy	58
3.2 Inertial navigation	58
3.3 Image segmentation	66
3.4 Low-cost semi-automatic DT generation	75
3.5 Case study	83
3.6 Summary	96
4 Conclusion	98
4.1 Discussion of the research	98
4.2 Key findings, insights, and limitations	102
4.3 Contributions to the field	103

4.4	Recommendations for future work	103
4.5	Closing reflections	104
Appendices		117

List of Figures

1.1	A DT interface.	2
1.2	LiDAR generated geometric data.	3
1.3	INS path estimation.	4
1.4	Geometric data extraction using image segmentation.	5
1.5	DT framework for the mine rescue environment.	11
1.6	LiDAR and IMU combined localisation and mapping.	21
1.7	Velocity estimation using an IMU.	25
2.1	Proposed methodology diagram, adapted and expanded from de Meyer.	38
2.2	IMU reference frame.	39
2.3	UKF algorithm workflow.	42
2.4	IMU reference frame representation within the Earth reference frame.	43
2.5	Proposed INS method.	45
2.6	Proposed synthetic scene implementation method.	47
2.7	Camera perspective example.	48
2.8	Raycasting matrix from the camera's perspective.	49
2.9	Proposed synthetic dataset generation method.	50
2.10	Proposed transfer learning method.	52
2.11	Proposed image segmentation method.	53
2.12	Proposed low-cost semi-automatic DT generation method.	55
3.1	Raw IMU data samples.	59
3.2	Comparison of raw IMU data versus UKF results.	61
3.3	Estimated path from motion equations.	63
3.4	Verification of the proposed INS method.	65
3.5	Real-world images inside a deep-level mine.	66
3.6	Open-source 3D models for objects of interest.	68
3.7	Perspectives of one of the generated synthetic scenes.	68
3.8	Automated labelling results for images in the generated dataset.	69
3.9	Generated dataset analysis.	70
3.10	Transfer learning evaluation result.	73
3.11	Verification of the proposed image segmentation method.	74
3.12	Mobile IMU sampler and image capturing device.	76
3.13	Top-down view of the DT used for verification.	78
3.14	Orthographic view of the DT used for verification.	78
3.15	Top-down view of the verification for the overall proposed method.	79
3.16	Orthographic view of the verification for the overall proposed method.	80
3.17	Segmentation result for the first green sphere.	81
3.18	Segmentation result for the last green sphere.	82

3.19	Orthographic view of the chosen case study.	84
3.20	Orthographic view of the distinct sampling locations in the case study.	85
3.21	The semi-automatically generated DT content for the case study.	86
3.22	Annotated estimated path for area B.	87
3.23	The image segmentations for area B in Figure 3.22 (part 1).	89
3.24	The image segmentations for area B in Figure 3.22 (part 2).	90
3.25	Annotated estimated path for area C.	91
3.26	The images captured for area C in Figure 3.25.	92
3.27	Image segmentation performance for area C in Figure 3.25.	93
3.28	Inertial navigation PSNR results for the case study.	94

List of Tables

1.1	Research topics and criteria for the literature analysis.	7
1.2	Inclusion and exclusion criteria for the study.	8
1.3	Summary of the literature analysis.	30
2.1	Analysis of 3D modelling frameworks for synthetic scene generation.	47
3.1	Quantitative analysis of the raw versus filtered IMU data.	62
3.2	Motion equations measurements.	64
3.3	Quantitative analysis of the UKF for the full verification test.	64
3.4	Verification measurements.	65
3.5	3D modelling framework selection process.	67
3.6	Improvement of performance metrics over each epoch.	72
3.7	Comparison of IoU results with methods from recent studies.	73
3.8	Segmentation performance comparison with state-of-the-art methods.	75
3.9	Quantitative analysis of the INS for DT generation.	81
3.10	Case study requirement checks.	83
3.11	Comparison of all PSNR results achieved.	94
3.12	Actual objects in the case study versus objects segmented.	95
4.1	Objectives achieved and corresponding novel contributions.	101

Table of Abbreviations

AHRS	Attitude and Heading Reference System
BIM	Building Information Management
CAD	Computer Aided Design
CNN	Convolutional Neural Network
CPS	Cyber-Physical System
DT	Digital Twin
EC	Exclusion Criteria
EMI	Electromagnetic Interference
GCP	Google Cloud Platform
GIS	Geographical Information System
GPS	Global Positioning System
IC	Inclusion Criteria
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
IoT	Internet of Things
IoU	Intersection over Union
LiDAR	Light Detection and Ranging
LSTM	Long Short-Term Memory
MEMS	Microelectromechanical Systems
MHSA	Mine Health and Safety Act
NC	Novel Contribution
OPC	Open Plan Communication
PDCA	Plan, Do, Check, Act
PQ	Panoptic Quality
PSNR	Peak Signal-to-Noise Ratio
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
SCADA	Supervisory Control and Data Acquisition
SLAM	Simultaneous Localisation and Mapping
SME	Small to Medium-sized Enterprises
STL	Stereolithography
TLS	Terrestrial Laser Scanning
UAV	Unmanned Aerial Vehicle
UI	User Interface
UKF	Unscented Kalman Filter
VR	Virtual Reality
2D	Two-Dimensional
3D	Three-Dimensional

1 Introduction

1.1 Background

The deep-level mining industry extracts gold and platinum group metals from extreme depths in South Africa [2]. The industry contributed ZAR 444.2 billion to the total gross domestic product of South Africa and employed more than 479,000 people in 2023 [2]. However, the industry faces challenges such as increased operational and labour costs, with their total turnover decreasing by 8.3% from 2022 to 2023 [2, 3]. This has forced the industry to improve profitability by investigating and implementing solutions to increase efficiency.

Many industries, such as robotics and automation, have turned towards the Industry 4.0 paradigm to achieve improved profitability [4–7]. This paradigm was publicly introduced at the Hanover Trade Fair in 2011 as part of Germany’s strategy to reduce production costs and prepare for future growth requirements [4].

Furthermore, this paradigm aims to increase competitiveness in the industrial sector. Since then, Industry 4.0 has been listed as a main topic on the 2016 World Economic Forum agenda, has been the focus of studies in various fields, and is currently being adopted in many industries throughout the world [4–8].

Industry 4.0 focuses on the digitisation of complex systems and the integration of advanced technologies [9–11]. Initiatives such as predictive maintenance, remote real-time monitoring and control, and process optimisation form part of Industry 4.0 [8]. These initiatives require digital technologies, such as the Internet of Things (IoT), Cyber-Physical Systems (CPS), and Digital Twins (DTs), to leverage the Industry 4.0 digitisation paradigm fully.

IoT has been defined as a set of interconnected and identifiable computational devices that can transfer data over a network without human interaction [12]. This enables the utilisation and synchronisation of multiple devices to form a mesh of gathered sensor data, providing a more comprehensive view of a situation compared to a traditional single sensor approach [13]. However, IoT technology is highly dependent on Internet connectivity through cellular networks or WiFi for synchronisation [13].

Another technology introduced by Industry 4.0 is CPS, which builds on the IoT approach by combining the interconnectivity between computational devices and the automation of complex control systems [14, 15]. This enhances the optimisation opportunities available in the control system by providing a synchronised and interconnected view of the overall state of the system. Similarly to IoT, this technology is also highly reliant on bidirectional connectivity to enable the complex control strategies to be executed [14].

Another technology from the Industry 4.0 paradigm that is considered here, the DT, is a digital representation of a physical system with cyber-physical connections to the real

system [16]. Information such as three-dimensional (3D) context, sensor measurement data, and system metadata, such as component maintenance data, can be presented in a DT [17–19].

An example of a DT is shown in Figure 1.1. Figure 1.1.a shows the full interface of the DT system; Figure 1.1.b shows the metadata of the components of the system; and Figure 1.1.c shows the interactive digital 3D model of the physical system. This DT was designed and implemented to assist with manufacturing planning and scheduling optimisation [20].

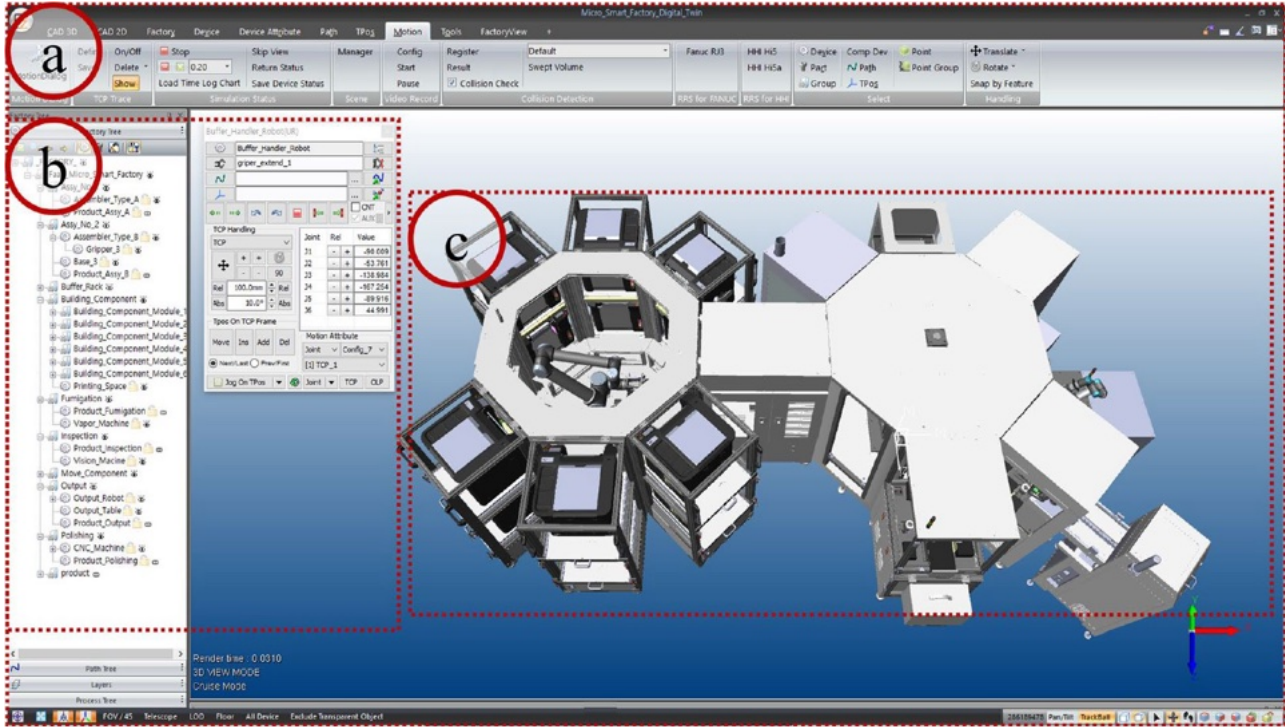


Figure 1.1: A DT interface [20].

DTs are a valuable tool in many different industries. Examples of this include the construction and maintenance industry through Building Information Management (BIM) [21, 22], the manufacturing industry through fault detection and process planning [23], and the urban planning industry through Geographic Information Systems (GIS) for urban transport [24]. The technology has also been utilised in the industrial fields, such as the Norwegian Oseberg-H oil field pilot, where a DT system was used to achieve unmanned production to reduce operational costs by 20%¹.

DT technology is the focus of this study as it contains a 3D digital representation of the physical system, it is not as highly dependent on interconnectivity as IoT and CPS, and DTs are already being used in other industries to decrease operational costs [17, 18, 25]. The DT market was estimated to be USD 8 billion in 2022 and is expected to grow to USD 40 billion in 2026 [26]. However, despite the various advantages of utilising DTs, the adoption rate is slow in the South African mining industry due to the high cost associated with their creation [27–29].

¹”Equinor starts production from Oseberg Vestflanken 2”, Oil & Gas Journal, <https://www.ogj.com/drilling-production/production-operations/field-start-ups/article/17297003/equinor-starts-production-from-oseberg-vestflanken-2> (accessed 10 August 2024).

DTs can be created using one of two approaches, namely, a manual or semi-automatic approach. Manual creation methods are time-consuming and not scalable in the case of many complex systems, such as deep-level mines, causing a large expenditure of working hours to complete [27,30]. This is caused by laborious DT creation from mine planning layouts, if any are available. Without these layouts, underground inspections need to be conducted which are time-consuming and expensive [29].

Semi-automatic creation approaches have utilised Light Detection and Ranging (LiDAR) methods to decrease DT creation time significantly compared to manual creation approaches. However, this is typically expensive as it requires specialised equipment and large data storage capacity [31,32]. This large expenditure requirement reduces the adoption rate of DT technology in the deep-level mining industry [8].

The distinguishing factor between the two methods of creating DTs is that in the semi-automatic approach, the 3D architecture of the system can be generated using various sensors. An example of a 3D architecture generated using LiDAR sensors is shown in Figure 1.2.

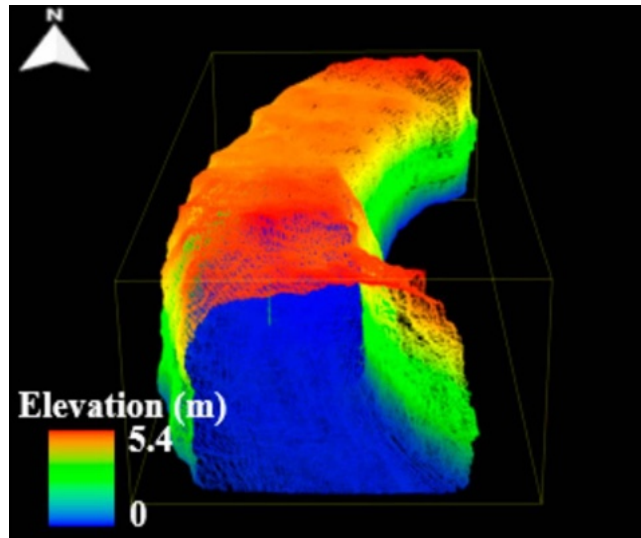


Figure 1.2: LiDAR generated geometric data [33].

This partial geometry of 20 m of an underground tunnel shows all the crevices and alterations of the rock walls and ceiling. Storing point cloud data for this 20 m section, however, requires 540,000 3D points [33]. Underground tunnel networks of typical deep-level mines reach up to 5 km in length [34]. Considering that each coordinate in the LiDAR point cloud represents at least 20 bytes of data [35], sampling the entire tunnel network would require 2.51 GB of storage space. Storing this amount of data with modern technology may seem trivial; however, processing this amount of point cloud data requires significant processing power [36–38].

Another popular method, used in the robotics industry for automatic navigation and digital generation of physical systems, is Simultaneous Localisation and Mapping (SLAM) [39]. SLAM consists of moving through an unknown or partially known environment, during which a digital layout of the environment is created and a variety of data is collected via sensors, such as

images, object locations, and environmental parameters [40–42].

The approach consists of two main processes, namely, localisation and mapping. During localisation, the positions of the entity moving through the system and the 3D architecture of the surrounding system are estimated to create a digital layout of the physical system.

Typical localisation methods, such as Global Positioning Systems (GPS), cannot be used in deep-level mines due to the absence of internet connectivity and satellite communication [8]. Fibre optic and WiFi network installation in deep-level mines has been investigated. However, providing full coverage using these methods is unfeasible at the time of this study due to cost, maintenance issues caused by the nature of the work done underground, and electromagnetic interference (EMI) noise [8, 43–46].

Inertial Navigation Systems (INS) provide another approach to the localisation problem. It is a method based on kinematic equations that describe the motion of an object without reference to the forces acting on it [47]. Recent studies have shown that INS produces promising results when utilising low-cost accelerometer, gyroscope, and magnetometer sensors in Inertial Measurement Units (IMU). These IMUs are typically found in mobile devices such as smartphones for offline localisation in high-noise environments [48–50].

Figure 1.3 shows promising path estimation results obtained using the INS method proposed by Yan et al. The estimated path (blue line) follows the ground truth (red line) in an anti-clockwise direction. The estimated path closely resembles the ground truth during the first half of the total path, most notably, the accurately estimated turning angles over the path. However, the estimation deviates from the ground truth more significantly over time.

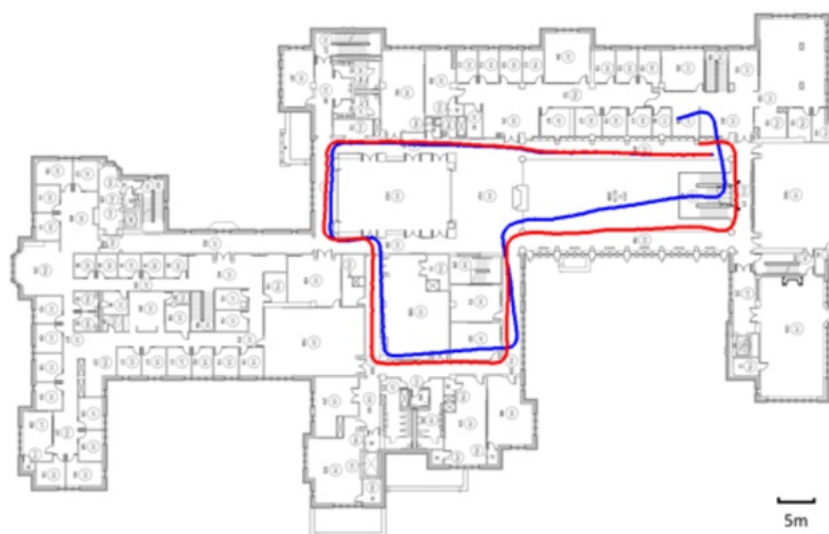


Figure 1.3: INS path estimation (red is ground truth and blue is estimation) [51].

The reduction in estimation accuracy over long periods is due to sensor measurement noise and bias instability of Microelectromechanical Systems (MEMS) IMU devices [51–53]. The other half of SLAM, the mapping, consists of sampling information of the environment around the

entity as it moves through the system and linking it to the relevant 3D coordinate in the digital system being created in the localisation process [54]. The information sampled can range from images to environmental information, such as temperature and humidity [55].

Recent studies have shown that image segmentation, a method that splits an image into smaller parts and categorises each part using a label, such as a person, sidewalk, or traffic light, produces promising results when used to extract information automatically from images [39, 56]. Figure 1.4 shows the geometric information extracted by Marchesi et al. with their proposed SLAM method. They utilised image segmentation to improve the 3D reconstruction of a well-known deep learning SLAM method, ORB-SLAM [57].

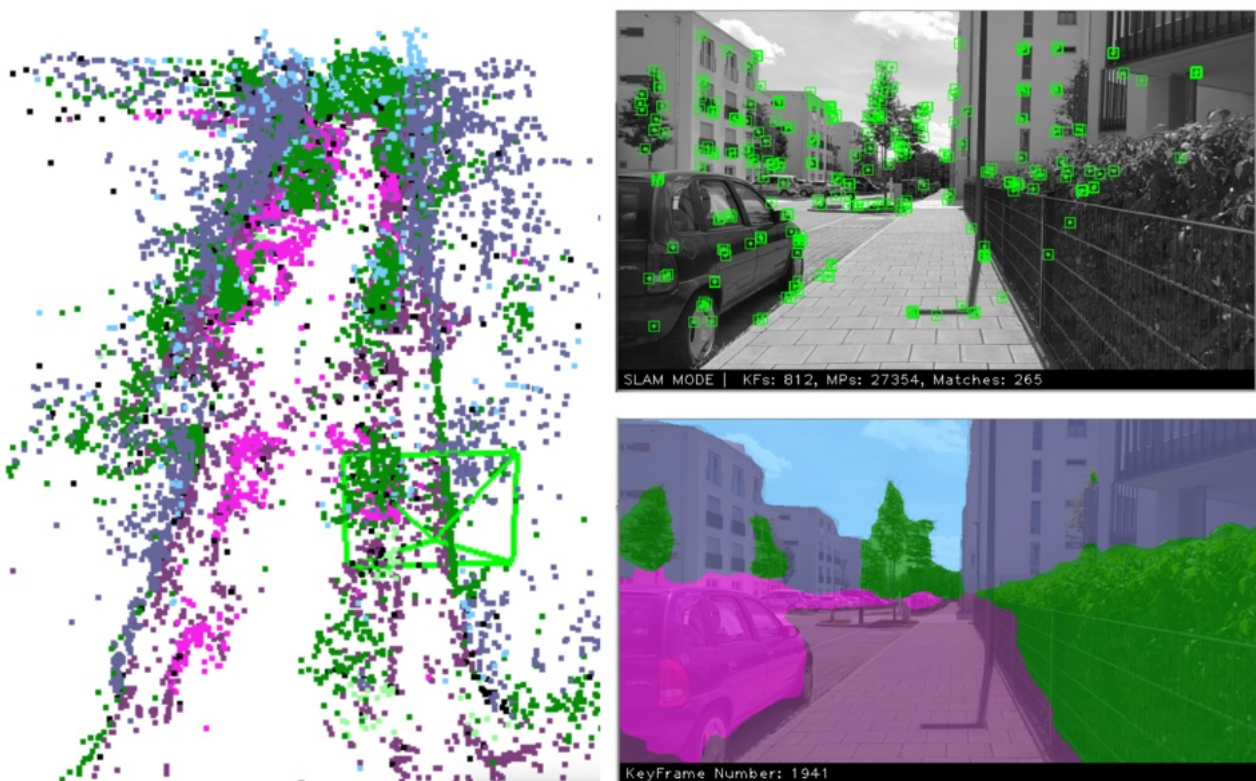


Figure 1.4: Geometric data extraction using image segmentation [54].

The left-side image of Figure 1.4 containing the green pyramid shows the perspective of the two images on the right. The top-right image shows the 3D point tracking of the ORB-SLAM algorithm, and the bottom-right image shows the image segmentation result. The left-side image also shows the results obtained when integrating the 3D geometric data produced by the ORB-SLAM algorithm and the image segmentation result (the coloured segmentation masks applied to the 3D points) [54].

In summary, the cost-effective creation and utilisation of DTs in the South African deep-level mining industry provides optimisation and efficiency improvement opportunities. These opportunities include preventative maintenance, identifying efficiency improvement initiatives, and decreasing operational costs. Through this, the industry can increase profitability and

mitigate increasing operational and labour costs. However, existing DT creation methods are expensive and require specialised equipment.

Therefore, the following research question is presented from the background: How can SLAM be utilised to enable low-cost semi-automatic DT generation and enable increased Industry 4.0 adoption in the South African deep-level mining industry?

1.2 Literature analysis strategy

Three high-level topics and six lower-level criteria were identified from the background and used to create Table 1.1. These topics and criteria will be used as keywords to find related research for the literature analysis.

Table 1.1: Research topics and criteria for the literature analysis.

Topic	Criteria	Description
Digital Twins	Deep-level mining application	DT approaches for deep-level mining applications to improve operational efficiency and decision-making.
	Semi-automatic generation	Methods for creating DTs by leveraging human expertise and computational algorithms to enhance efficiency.
Localisation and mapping	Low-cost method	Accurate localisation and mapping methods using affordable sensors and minimal resources, emphasising simplicity, scalability, and adaptability.
	High-noise environment	Discussions and solutions for localisation and mapping in environments with significant noise or interference.
Kinematic and perceptual methods	Inertial navigation	INSs estimate change in position over time by compensating for sensor errors and drift, therefore improving navigation accuracy and resilience under challenging conditions.
	Image segmentation	Image segmentation partitions images into distinct regions, enabling various applications like object recognition and localisation.

The Google Scholar search engine, specifically its advanced search utility, will be used to identify research related to the topics and criteria, as given in the table above. The inclusion criteria (IC) and exclusion criteria (EC) shown in Table 1.2 will be used to filter the identified research further.

Studies where the scope of the research is engineering will be considered. No studies published before 2017 will be included to ensure that the latest research is analysed, unless it includes fundamental research regarding a core topic of this study. Systemic review papers were also included, as this type of research aggregates many studies into condensed analysis and conclusions.

Table 1.2: Inclusion and exclusion criteria for the study.

Name	Description
IC1	The paper focuses on at least one of the topics and criteria sets of research listed in Table 1.1.
IC2	The scope of the paper is engineering.
EC1	The paper was published before 2017, unless it includes fundamental research regarding a core topic of the current study.
EC2	The paper does not contain at least a theoretical framework proposal or an application of an existing method in a novel application, unless it is a systematic review that aggregates many papers into condensed research.
EC3	The paper is unavailable, even after contacting the authors.

The literature analysis in Sections 1.3 to 1.6 is structured as follows: The topics from Table 1.1 are represented as sections, and the criteria are the subsections. Each research paper considered in the sections will be analysed in a portion consisting of, firstly, a summarised title and reference and background about the research, followed by the specific problem the authors aimed to address. Thereafter, their proposed method and results will be analysed and discussed. Finally, the strengths and possible shortcomings of their proposed method and results will be presented.

1.3 Digital Twins

1.3.1 Preamble

Studies focusing on deep-level mining applications and semi-automatic generation methods are analysed in this section. Deep-level mining applications of DTs are considered as this environment introduces extreme conditions not typically found in places such as urban environments.

1.3.2 Deep-level mining application

DTs for the sustainable mining industry [58]

DTs are one of the technologies enabling Industry 4.0 initiatives, and they have been shown to improve operational efficiency, productivity, and sustainability in many industries, as discussed in the background. However, El Bazi et al. posed the following question in their paper published in 2023: How can this technology be successfully integrated into the mining industry? This is an important question to consider, as more harm than good may be caused when the integration fails.

In their state-of-the-art discussion, El Bazi et al. considered papers focusing on DT case studies

in the mining industry that included conceptual design and practical implementation. They found many papers focused on specific problem areas within the mining industry and concluded that there exists a need for a generic DT implementation framework to increase the chances of successful integration.

El Bazi et al. proposed a generic multi-layered DT asset lifecycle management framework that focuses on sustainability in the mining industry. Their framework includes considerations for the development, production, and maintenance lifecycle phases of assets within the mining industry. The initiatives they focused on addressing with their framework include revenue growth opportunities, operational cost management, quality control, and record retention services.

Their framework includes integrating new or existing Supervisory Control and Data Acquisition (SCADA) systems, Open Plan Communication (OPC) servers, cloud data storage solutions, and simulation software into the DT. They also discussed design considerations when using their proposed framework for a specific application, such as system scalability and visualisation and analysis tools.

Including the integration of SCADA systems into their framework is valuable, as many deep-level mines already have complex operational SCADA systems. Including this, as well as the other main factors they considered in their framework, will increase the likelihood of successfully integrating DTs into the mining industry.

In their paper, they have done much research to identify shortcomings and opportunities to increase the likelihood of successful integration into the mining industry. However, they did not apply their proposed framework to a case study to verify whether they answered their original research question.

DT-based mine hoist temperature monitoring and simulation system [59]

Hoist motors are a critical part of an underground mine's infrastructure. When exposed to prolonged overheating, which can lead to motor winding damage, they pose a significant safety risk due to issues such as slipping. Therefore, Liang et al. proposed a DT temperature simulation and monitoring method for mine hoist systems to improve predictive maintenance and provide proactive online temperature monitoring.

They proposed using a Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) trained using historical data to achieve online monitoring. An LSTM overcomes the classical problems of a standard RNN, such as gradient disappearance or explosion (under- or over-compensation for certain phenomena in the training data). Liang et al. additionally proposed using the Gauss-Kalman joint filtering algorithm to improve the temperature predictions within the DT further.

They applied their proposed method to a case study of a deep-level mine in China. They manually created the 3D model of the DT for the hoist in Unity 3D, a modern cross-platform game development platform. However, a manual approach is not feasible for large-scale DT applications such as an entire underground mining system.

The temperature of the hoist system was accurately monitored, and notifications were sent to relevant contact persons when it exceeded a certain limit. Although the hoist is located on the surface of the underground mine, an advanced filtering method was used to mitigate high noise levels on the temperature sensor from the effects of the large motor with a lifting capacity of 30 tons.

The case study showed that their proposed method does accurately monitor and provide predictive feedback on the status of the critical infrastructure. This demonstrates the advantages that DTs can provide for the critical safety infrastructure found in deep-level mines.

DT monitoring method for hydraulic supports in deep-level mines [60]

The working face in a deep-level mine is defined as the rock surface where mining is progressing [61]. These areas are created to enable miners to drill holes in order to plant explosives which are then later detonated to break the solid rock into smaller, movable pieces ready for ore processing. These working faces are considered some of the most dangerous areas of the mine as the ground loses its support during this process [62].

Hydraulic supports have been widely used to replace the support provided by the removed ground and create a safe working environment for miners to drill for the next blasting round [63]. These hydraulic supports require regular maintenance to function as expected, and therefore Xie et al. proposed a DT monitoring method for hydraulic supports in deep-level mines.

Their proposed method consists of a data acquisition system, a data storage system, and a virtual representation of the hydraulic support. The data acquisition system consists of single-axis and multi-axis tilt angle sensors, a displacement transducer, a proximity switch, and a pressure transducer to collect the hydraulic support state information. SQL Server was chosen as the data storage system, and the hydraulic support's virtual model was imported from available stereolithography (STL) files into Unity 3D.

They applied their proposed method to a hydraulic support system under laboratory conditions, as they could not test their method underground. Xie et al. did not provide a reason for this; however, it may have been due to implementation difficulties or safety concerns in their underground environment. This highlights the importance of the question posed by El Bazi et al. regarding the successful integration of DT technology into the mining industry [58].

The case study results obtained by Xie et al. showed successful monitoring of the physical system through the user interface. However, the test was conducted in a laboratory environment with minimal noise levels. Their method requires relatively low-cost sensors placed on a hydraulic support already purchased by the mine. The data storage solution and visualisation interface can also be considered low-cost.

Although their tests showed that low-cost sensors are feasible in their approach, Xie et al. did not discuss the communication methods that should be used in an underground environment. The installation of Fibre or WiFi networks is currently not feasible for full coverage in a mine. As frequent detonations occur, installation of these types of networks in a specific working face is also not feasible, as the explosions may damage the network equipment.

A framework for DT deep-level mining site rescue environments [64]

Safety is an essential aspect of the deep-level mining industry [65,66]. Emergencies, such as the unexpected fall of ground or rocks in a tunnel, may require rescue efforts if people are hurt or trapped. Due to the increasing number of disasters in Chinese mines, Wen et al. proposed a DT framework to aid in the simulation, monitoring, and coordination of a mine rescue situation.

Their framework, shown in Figure 1.5, consists of six parts, namely, the physical entity of the mine, the underground monitoring terminal, the DT data, the cloud services, the control terminal, and the Virtual Reality (VR) mine rescue environment. Each arrow between the different parts of the framework indicates data and control flow operations.

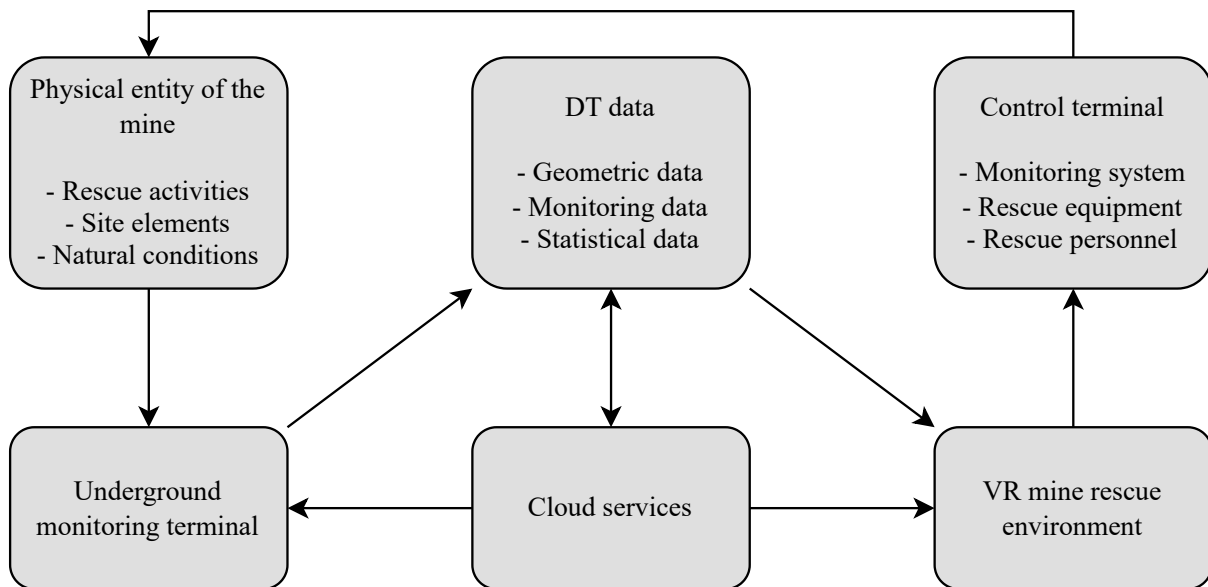


Figure 1.5: DT framework for the mine rescue environment [64].

The DT data and the cloud services form the framework's backbone, as those parts have the most outgoing and incoming information flows. The VR mine rescue environment is designed

to provide the user with as much information about the underground situation as possible to aid coordination during rescue situations.

Wen et al. made a proof-of-concept model to test their proposed framework. The 3D model of the DT was manually constructed using Unity 3D and Unreal Engine 5 [67], and an Azure Kinect DK device was used for the VR interaction [68].

Although their proof-of-concept results look promising, issues such as scalability and cost for a large number of underground mines and implementation costs remain unresolved problems. This is due to the manual creation of the DT used for their virtual model. However, their approach to integrate safety consideration as a core part of the DT framework is valuable.

1.3.3 Semi-automatic generation

The manual creation of DTs is time-consuming and expensive, which hinders the adoption rate of Industry 4.0 in the South African mining industry. Studies focusing on semi-automatic DT generation aim to bridge the gap between manual and fully automatic approaches. Much progress has been made in several different industries; however, certain limitations and problems still need to be addressed before this will be possible.

Urban environment LiDAR point clouds to DTs [69]

Processing unstructured data is typically computationally inefficient. This problem is exacerbated by the large amount of unstructured point cloud data produced by LiDAR sensors. To mitigate the effect of this large amount of data, Xue et al. proposed a method to identify coordinate connectivity, create cross-sections of identified objects, and finally, use this information to create a DT city model semi-automatically.

They applied their proposed method to a car park containing 12 vehicles in Dublin, Ireland. Their results showed that the method accurately clustered 24,126 LiDAR points into 12 separate vehicles, created cross-sections of each, and successfully matched all identified vehicles to their corresponding open-source CAD model.

They state that the proposed method is feasible for large-scale DT generation; however, their method was only applied to a small-scale environment using a relatively powerful desktop PC for processing their algorithm. Increasing the area of the urban environment to something of similar complexity to an underground mine does not seem feasible, as the cross-section of a tunnel network in an underground mine can be around 5 km [34].

Their proposed method also relies on freely available and accurate LiDAR point clouds originally sampled by aerial LiDAR scanning. LiDAR point clouds exist for typical environments. However, freely available point clouds for South African deep-level mines could

not be found at the time of this study. Furthermore, sampling and processing LiDAR data for entire tunnel networks would also not fit into the focus of the current study, which is a low-cost semi-automatic generation approach.

Image-based DT creation for large-scale structures [70]

During the production and maintenance phases of large-scale offshore buildings, such as wind turbines, they are susceptible to defects and structural damage due to human error and extreme weather conditions. Identifying these situations as quickly as possible can result in a large reduction of expenditure to rectify or repair.

Benzon et al. proposed a semi-automatic DT generation and update method for large-scale offshore structures from the production phase to the maintenance phase. Their method utilised colour images captured by a lightweight drone and LiDAR scans to create the virtual structure of the DT. Utilising image segmentation techniques on the images captured by the drone, the paint defects on the structure were identified and located.

Benzon et al. applied their proposed method to an offshore wind turbine case study. The result was an accurate 3D representation of the structure, including the location and severity of paint defects. This virtual representation and localisation of paint defects assisted with preventive maintenance and efficient asset management of the structure.

Their deep learning image segmentation approach utilised the Intersection over Union (IoU) performance metric during the evaluation phase and achieved a score of 0.59. A high-performance model evaluated on the COCO 2017 dataset achieved an IoU of 0.69 [71]. Therefore, Benzon et al. achieved a relatively high segmentation performance. Their method includes semi-automatic DT generation and image segmentation techniques. However, Benzon et al. did not specifically consider deep-level mining applications that introduce further challenges.

Their results show that image segmentation is a feasible method to identify and locate certain phenomena of interest from images, even in a real-world offshore scenario. The results, created from the captured images, combined with LiDAR scans, verified that this approach can successfully generate DTs.

However, their method relies on constant communication with the drone in combination with real-time GPS tracking of the captured images. Line-of-sight communication may be maintained in an underground environment. However, deep-level mines are a GPS-denied environment and therefore this approach would not be feasible. Nevertheless, their photogrammetry approach (creating 3D virtual structures from images) may be feasible for underground mapping if communication could be maintained.

Towards automated DT generation using SLAM [72]

Typical 3D reconstruction tools utilising LiDAR equipment are typically too expensive for Small to Medium-sized Enterprise (SME) factories, hindering the adoption of Industry 4.0 initiatives. Minos-Stensrud et al. proposed a semi-automatic and cost-effective DT generation method that utilises a drone, several inexpensive RGB cameras, and SLAM techniques.

The pose and orientation of the drone were measured using the OptiTrack motion capture system². They applied their method to a case study in a research laboratory where they obtained promising results, displaying the digitally constructed DT using a VR headset as the User Interface (UI).

Research laboratories typically represent near-ideal environments with low noise. However, they did utilise low-cost data-capturing hardware and open-source SLAM software libraries for semi-automatic generation [73]. The SLAM software they utilised was based on a photogrammetry approach, similarly to Benzon et al. [70]. These results reinforce the feasibility of SLAM and image-based approaches for low-cost DT generation.

Integrating 2D and 3D information towards automatic generation of DTs [74]

Standardisation in Industry 4.0 is an important step towards enabling faster adoption of the paradigm. It enables industry specialists to work interchangeably between the many tools developed for Industry 4.0.

Sierla et al. focused on the integration of two-dimensional (2D) and 3D information during the semi-automatic generation of DTs for processing plants that contain various pipe and instrumentation configurations. They proposed a graph generation method to be used in a graph matching approach to identify corresponding elements from piping and instrumentation diagrams to elements within 3D CAD models.

The piping and instrumentation diagrams contain layouts of the control loops of the plant, while the 3D CAD models contain detailed geometric information about the plant. Combining these two data sources to generate the plant DT through graph matching does seem feasible. They applied their proposed graph generation method to a relatively small thermo-hydraulic water process case study that contains ten pipelines.

They obtained promising results, showing that graphs can be generated using 2D and 3D data. However, they did find mismatches between the piping and instrumentation diagrams, the CAD model, and the real-world system. This disparity of input information also caused issues in the generated graphs.

²“Motive - Optical Motion Capture Software,” NaturalPoint Inc., OptiTrack, <https://optitrack.com/software> (accessed 24 February 2025).

In their discussion, they mention that this was likely to occur in the industry over the lifecycle of the plant, and therefore, further work would be required to address this shortcoming. Sierla et al. did not focus on a low-cost approach for their proposed method, as 3D CAD models are not typically available for these types of systems, and generating these full models for complex systems, such as deep-level mines, would be expensive.

Analysis of 3D bridge models generated using Unmanned Aerial Vehicles [75]

Unmanned Aerial Vehicle (UAV) and Terrestrial Laser Scanning (TLS) are some of the most popular methods to generate 3D models of large terrestrial objects, such as bridges, for use in a DT system [76, 77]. This is not a generation method for DTs as 3D models form only one part of a full DT system, according to the definition used in this study [16]. However, the 3D model, and therefore its generation, is an important part of the overall DT system.

Many studies have investigated UAV-based and TLS techniques separately to generate 3D models, but few have evaluated both results with the same testing procedure in terms of quality and geometric accuracy. Therefore, Mohammadi et al. proposed a method to address this gap.

Their method consists of, firstly, a general approach to conduct each scanning technique. The next step is to align both the scale and rotation of the geometries of each 3D model created. Thereafter, the data quality evaluation is performed. The evaluation is based on multiple tests, such as point distribution, outlier noise, visual quality assessment, surface deviations, and geometric accuracy.

Their method was verified using a real-world case study of a bridge in Australia. This case study showed that TLS provides a higher level of point density, significantly less outlier noise, and greater geometric accuracy compared to UAV-based photogrammetry. However, the equipment cost and limited access to the resources required for TLS may make it unfeasible to use in low-cost applications.

Studies focusing on DTs, specifically their applications in deep-level mining and semi-automatic generation methods have been analysed in this section. It was found that none of those focusing on deep-level mining applications included a low-cost semi-automatic generation approach. Further, only one SLAM study used a low-cost approach, however, their verification was done in ideal laboratory conditions.

In the following section, studies focusing on localisation and mapping, specifically low-cost approaches and high-noise environments, will be analysed to gain insight into how SLAM methods may be utilised for the semi-automatic generation of DTs.

1.4 Localisation and mapping

1.4.1 Preamble

SLAM methods are used for navigation and context generation in industries such as robotics and manufacturing. These methods typically require expensive hardware, such as LiDAR sensors [31, 32]. The methods are also sensitive to external noise factors, such as EMI [46]. Studies focusing on low-cost and high-noise environment SLAM methods are analysed in this section.

1.4.2 Low-cost methods

Indoor localisation using IMU and smartphone camera [78]

Visual SLAM methods combine inertial data from an IMU and image or video output from a camera to perform localisation and mapping. These methods have produced promising results, as shown by Poulouse and Han, who proposed a hybrid indoor SLAM method using the IMU and camera of a smartphone.

Their method utilises a Kalman Filter, as samples from an IMU include EMI and sensor bias. The Kalman Filter calculates the state transition matrix of a system over time to reduce the effects of noise on the desired estimation, which in this case is the estimation of the change in position over time [79].

Poulouse and Han applied their proposed method to a relatively small indoor case study with a total walking distance of 18.2 m. It consisted of walking around a table the size of a typical dining room table with a smartphone to collect inertial and visual data, and then transferring the data to an external computer for their localisation and mapping method to estimate the path.

The lowest mean error from their hybrid method's experiments was 0.0617 m over the 18.2 m total distance, while only using the IMU produced a mean error of 0.171 m. Therefore, combining the IMU sensor data and the visual data from the camera produced improved localisation results compared to only using the IMU.

Their experiments, although showing promising path estimation results, were conducted in a relatively small indoor environment with minimal sources of noise and interference. For example, their environment shows consistent and adequate lighting conditions, which enable accurate extraction of all the information present in the images. Their experiments also do not demonstrate the performance of their method over larger distances, as typically found in deep-level mines.

Very low-cost UAV-LiDAR system for forestry applications [80]

UAV-based LiDAR scanning is a popular method for creating and maintaining forestry inventory management information, such as canopy cover and gap fraction (the ratio of openings in the canopy) analysis [81, 82]. These methods were typically expensive due to the highly specialised LiDAR hardware requirements (approximately USD 4000 per unit).

However, in a recent study, using the low-cost LiDAR hardware released by Da-Jiang Innovations (DJI), namely the Livox MID40 (approximately USD 600 per unit), Hu et al. proposed a very low-cost UAV-LiDAR system for forestry applications and evaluated tree-level and plot-level results.

Their proposed system includes a drone, a GPS antenna, a long-range WiFi antenna, and a high-precision IMU. The IMU, GPS, and LiDAR samples were combined to create georeferenced point clouds of the measured forestry data along the drone's flight paths. They performed a comprehensive comparison between their results and those of four other LiDAR systems using high-end expensive hardware.

Their results were comparable to those of systems that use high-end hardware, except for areas with high canopy densities. This shows that more reasonably priced LiDAR hardware may also provide promising localisation results. Their proposed method does, however, require long-range WiFi connectivity for drone control including a GPS signal for point cloud generation.

Long-range WiFi and GPS connectivity are both not currently feasible in deep-level mines. Despite this, the previously unidentified low-cost LiDAR hardware may resolve the issues with the conventional costly LiDAR approach; however, this has not yet been proven.

Adaptive attitude estimation for MEMS IMU [83]

Attitude and Heading Reference Systems (AHRS) utilise the accelerometer, gyroscope, and magnetometer from low-cost MEMS IMU hardware found in modern smartphones to estimate velocity and orientation change over time. A typical issue with these MEMS sensors is that their samples contain high noise levels, are subject to sensor bias over time, and are influenced by external factors, such as EMI induced by nearby materials.

MEMS sensor hardware is improving over time; however, sensor error, bias, and external factors still cause the estimation of velocity and orientation changes over time to deviate significantly as the number of samples increases. The Kalman Filter, and all improved versions of it, aims to track the system's state over time to reduce the influence of errors on the estimations.

Park et al. proposed an adaptive attitude estimation method for MEMS IMU to improve path estimation further. Their method used a Kalman Filter to improve the accuracy of the

orientation calculated from the gyroscope as well as a new ellipsoidal approach to estimate the magnetometer and accelerometer bias. This calculated bias can then be used to update the next estimation of the attitude obtained from the gyroscope.

They created a test-case scenario with random Gaussian noise and sensor bias added to the simulated measurements to test their proposed method. The results showed an improvement in attitude estimation compared to simply using the unfiltered IMU data.

Their proposed method was not verified using a real-world case study, even though they added simulated noise. Therefore, external unknown factors may negatively influence their proposed method as perfectly replicating and simulating a real-world scenario is not feasible.

AHRS through utilising an IMU and Extended Kalman Filter [84]

Traditional Kalman Filters, such as the one utilised by Park et al. [83], require a linearisation step to model the system dynamics for which they will be utilised. Simple numerical functions can be used to model linear systems, while non-linear systems are too complex to be fully described by a numerical model [85].

Often, non-linear systems, such as the position and orientation of a smartphone carried by a person walking, are modelled as linear systems by making certain assumptions. Examples of these are constant velocity or constant acceleration assumptions. The conversion from a non-linear to a linear system often excludes sensor noise, bias, temperature, and alignment error, that creates more discrepancies compared to the actual system [84].

Although still using linearisation, the Extended Kalman Filter (EKF) mitigates the errors by tracking these unknown states using observations over time. Farahan et al. utilised an EKF to estimate system parameters: the gyroscope, accelerometer, and magnetometer's unbiased measurements and their corresponding bias values that change over time. Their proposed method also utilised a quaternion representation for the orientation, which avoids the gimble lock issue when using the Euler representation.

Applying their proposed method to the output of an IMU reduced the observed noise levels. Farahan et al. also utilised fast Fourier transform and probability density function diagrams to investigate the extent of the noise reduction.

Their proposed method was, however, not applied to a real-world deep-level mining case study that introduces further EMI. Despite this, their study showed that an improved Kalman Filter approach removes more noise compared to traditional Kalman Filter approaches and would, therefore, provide more reliable path estimation results.

1.4.3 High-noise environments

In many applications, high levels of noise will typically be found. This may be caused by the operation of heavy machinery, environmental factors, or EMI [46, 86]. Studies focusing on mitigating the effects of high noise levels are analysed in this subsection.

AI-IMU Dead reckoning [87]

In recent years, much progress has been made with self-driving vehicles. However, these technologies often rely on GPS sensor data for localisation. GPS data may not be available in certain scenarios, such as driving through obstructed terrain or in the case of sensor failure. The significant noise levels in inertial sensor data for vehicles also prohibit using typical IMU methods, as their estimations diverge after only a few short seconds.

Brossard et al. therefore proposed a localisation method robust against the effects of noise for self-driving vehicle applications by combining an IMU, an Invariant Extended Kalman Filter (IEKF), and a Convolutional Neural Network (CNN). The IEKF was used to estimate the state of the system over time, and the CNN was used to update the estimated covariance of the noise parameter in the IEKF.

They tested their proposed method using the KITTI inertial dataset which contains sampled IMU data as well as the corresponding ground truth path recorded with high-accuracy GPS sensors [88]. Their final results showed only a translational error of 1.1% from the ground truth recorded in the dataset. Their results rival popular LiDAR and vision localisation methods tested using the KITTI dataset.

Their approach can also be considered low-cost as a LiDAR sensor was not utilised for localisation. However, Brossard et al. did not test their proposed method in a real-world case study, which limits the robustness of the verification of their proposed method.

Localisation and mapping in underground mines using LiDAR [33]

The accuracy of a DT in comparison to the real-world system, known as replication precision, affects the results of various methods enabled by a DT, such as resource planning or maintenance optimisation. The utilisation of a DT is negatively affected when large replication errors exist, especially in localisation applications. Baek et al. therefore proposed a localisation and mapping method using mobile LiDAR technology to reduce this error.

Their proposed method utilised mobile LiDAR equipment to sample 10–20 m portions of an underground mine and then match the sampled portion to an existing global point cloud map. Through this, the high-detail local scan was used to improve the global point cloud's

accuracy while simultaneously locating the user and objects around the user within the underground mine.

Their results showed only a Root Mean Squared Error (RMSE) of 0.13 m between the global point cloud and the local point clouds. However, their localisation method was not tested for sections of underground tunnels spanning greater than 20 m. This is most likely due to the high computational and storage requirements for processing and matching the LiDAR data between global and local point clouds.

LiDAR equipment is expensive not only because of the high-cost sensors but also because of the required computational power and storage to process the collected data. This causes their proposed LiDAR approach to be unfeasible when the aim is to generate a DT semi-automatically for larger portions of a deep-level mine.

Orientation estimation in the presence of magnetic disturbances [89]

A few different IMU studies have been analysed so far. However, Fan et al. proposed a new approach that tracks stationary states and magnetic disturbances over time to reduce the effects of high noise levels on orientation estimation.

They utilised the accelerometer and gyroscope of a low-cost IMU to detect the stationary state, where no magnetic changes were recorded (as these were assumed to be caused by drift). They then used sensor fusion, which is the integration of multiple sensor measurements to reduce noise, combined with a quaternion representation to avoid the gimbal lock issue. Through this, they estimated the magnetic disturbance over time [90].

Their results showed a 51.2% error reduction in orientation estimation. A different method that utilised an EKF achieved a 50–60% error reduction, which shows that the method proposed by Fan et al. also provides promising results. However, their results were obtained in laboratory conditions, which eliminate much of the noise found in real-world case studies.

If their method were extended to include a state-of-the-art Kalman Filter or deep learning model, the accuracy of the orientation estimation may be improved even further. A real-world case study would also improve the robustness of the proposed method.

Autonomous robot SLAM in an underground mine using LiDAR and IMU [91]

In many areas of an underground mine, risk factors, such as toxic gases or the high likelihood of fallen rocks, prevent people from entering the area. Therefore, autonomous vehicle initiatives have been investigated to circumvent this safety concern. Autonomous vehicles, as they do not require direct human intervention to operate, therefore also mitigate typical human error.

Autonomous vehicles do, however, require a method for localisation, and in the case of an underground mining environment, GPS is not an option. Therefore, Kim et al. proposed a method to combine LiDAR technology with an IMU approach to maximise the accuracy of the localisation and mapping while avoiding sensor drift errors.

Their proposed method is summarised in Figure 1.6. They utilised two separate LiDAR sensors to capture horizontal and vertical point clouds. The IMU sensor was used to measure orientation (attitude and heading), and a Kalman Filter was used to estimate roll, pitch, and yaw.

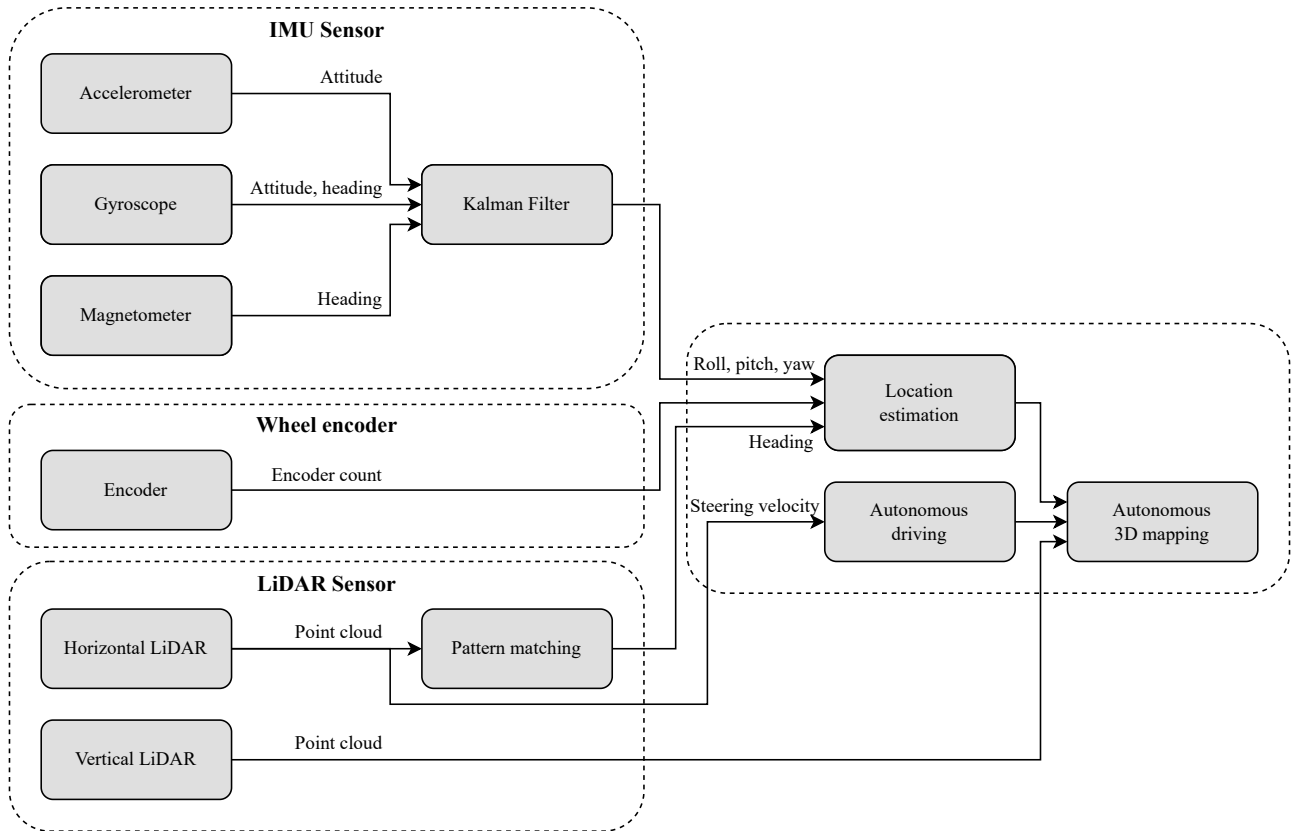


Figure 1.6: LiDAR and IMU combined localisation and mapping [91].

An encoder was used to measure the change in position due to the movement of the wheels of the vehicle. All of these sampled measurements and estimated parameters were combined to achieve autonomous 3D mapping. Their proposed method was tested in an underground mining case study, which verifies the robustness.

The RMSE achieved by their proposed approach was only 0.05 m² over a total distance of 31 m traversed by the vehicle. The localisation error is lower than in other studies, which only utilised the IMU for navigation. However, the LiDAR approach will incur more significant costs than only using an IMU approach, due to the cost of the equipment and data processing capabilities required [35–37].

Extended Kalman Filter for localisation in underground mining environments [92]

The use of LiDAR for underground localisation is expensive when compared to methods that only require an IMU. Brigadnov et al. proposed a Kalman Filter-based localisation method for underground mining environments to address this. Similar to Brossard et al., they used a variation of an EKF, but without a CNN. The Extended Kalman Filter was designed to track the error state over time.

While Brossard et al. tested their proposed localisation method on the KITTI inertial dataset, Brigadnov et al. tested their proposed method using a simulation of an autonomous robot operating system [93]. The simulation included error variance values for all platform measurements, which include all three sensors found in an IMU.

They obtained promising results through the simulation, with the lowest RMSE value between their proposed localisation method and other Kalman Filter approaches available at the time. Although the results are promising, a real-world case study would increase the robustness of the proposed method.

In the following section, studies focusing on kinematic and perceptual methods in SLAM will be analysed to finalise the answer to the research question posed in Section 1.1: How can SLAM be utilised to enable low-cost semi-automatic Digital Twin generation and enable increased Industry 4.0 adoption in the South African deep-level mining industry?

1.5 Kinematic and perceptual methods

1.5.1 Preamble

INS is an approach used to estimate the change in position over time while tracking the error state to increase the accuracy of the estimation [53]. Image segmentation partitions images into distinct regions, enabling object recognition and localisation [56]. Studies focusing on these two areas are analysed in this section.

1.5.2 Inertial navigation

INSs utilise gyroscopes, accelerometers, and magnetometers present in IMU devices to determine the orientation and heading of a person or device such that the change in position over time can be estimated [49]. These sensors measure angular velocity, linear acceleration, and the strength of the surrounding magnetic field, respectively [52].

However, sensor errors and drift exist that significantly decrease the accuracy of the estimation over long periods due to the integration of sensor errors over time [51]. Therefore, INS includes

advanced error estimation methods to track the error over time to increase the accuracy of the estimation [51–53]. Studies focusing on this topic are analysed in this subsection.

Low-cost INS using adaptive Kalman Filter [94]

The Kalman Filter, originally proposed in 1960 [79], and its numerous variations have been widely used in INS since its introduction to reduce state estimation errors over time, as previously discussed [84,92]. However, traditional Kalman Filter approaches are still negatively affected by sensor error and drift over time [87,91].

Navidi et al. proposed a new adaptive Kalman Filter for an IMU-based INS to improve the state-of-the-art estimation accuracy. They modelled the attitude (roll, pitch, and yaw) of the MEMS device using quaternions, similar to the approach of Farahan et al. [84]. They modelled the estimated error of the gyroscope, accelerometer, and magnetometer as 3D vectors. The attitude quaternion and estimated errors formed the state vector of the proposed adaptive Kalman Filter.

They considered three approaches for initial orientation estimation. The first approach is geomagnetic matching, which utilises the accelerometer’s output in a static state, as well as the magnetometer’s output, to determine the orientation relative to the Earth’s magnetic frame.

The second approach is the compass heading technique, which uses the fact that the Earth’s magnetic field has a component in the horizontal plane that points towards magnetic north. The final approach used, which is the approach they proposed, is a combination of the previous two approaches by rearranging the equations to fit the same reference frame (Earth’s magnetic field).

They tested their proposed approach against a high-end reference system and found that the largest improvement of the initial orientation estimation was 45% compared to the first two initialisation approaches. Their approach can be considered low-cost as they utilised a MEMS IMU device. However, their tests were conducted under laboratory conditions without high noise levels typically found in real-world scenarios, such as deep-level mines.

Deep learning inertial navigation [95]

As stated previously, deep learning approaches require a dataset to train an inference model. Deep learning-based INS also requires a dataset, and many different datasets specifically for IMU devices already exist. However, the generalisability of the available datasets for all environments is not clear yet.

Herath et al. proposed a dataset, namely Robust Neural Inertial Navigation (RoNIN), an improvement on the largest IMU motion dataset available at that time, OxIOD [96]. RoNIN

increased the number of hours of motion data for an IMU to 42.7 (an increase of 2.9 times), increased the number of human subjects to 100 (an increase of 20 times), and improved device handling to be a more accurate representation of day-to-day device motion.

Beyond the new dataset, Herath et al. also proposed the use of three new deep learning architectures for deep learning inertial navigation. The first is a residual neural network, a variation of a standard neural network typically used in image segmentation.

The second is an LSTM, a time-series neural network that tracks an error state and updates the inference model as each time step passes. Finally, the third architecture is a Temporal Convolutional Network (TCN), another variation of a time-series neural network that aims to track long-term states in a sequence.

Herath et al. compared the results of using their proposed dataset and three architectures with other datasets and architectures available at that time. All three newly proposed architectures produced the lowest mean trajectory error between the estimate and the ground truth.

However, in each case, the results of training their new architectures on other datasets did not outperform existing methods. Their proposed dataset was also not created in a high-noise environment. They used a low-cost IMU device for validation.

Human velocity estimation from a single IMU [97]

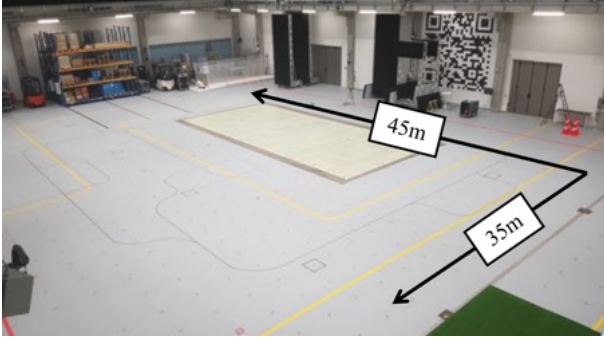
Human inertial navigation aims to optimise the estimation of velocity for specific sensor positions, problem applications, and types of movement. This approach uses a Kalman Filter or one of its many variations, such as an EKF or Unscented Kalman Filter (UKF), to track sensor error and drift over time to increase the accuracy of velocity estimation [92]. However, these methods are not robust or generalisable for most INS applications.

Feigl et al. proposed the combination of a deep learning approach using a CNN and a bidirectional LSTM with an EKF to improve the robustness of INS. In their proposed method, a single uncalibrated IMU was used. They validated their proposed framework using a dataset created using a low-cost IMU device with 20 users and highly dynamic movement. Figure 1.7 shows the area in which they collected data and validated their framework. It also shows the results obtained from their velocity estimation framework.

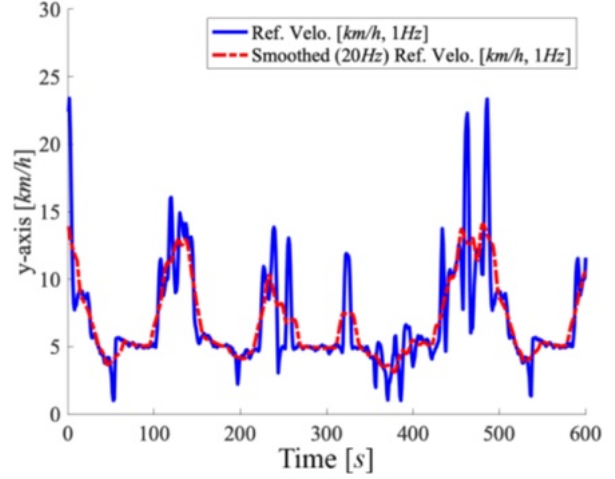
The device used by Feigl et al. was a Samsung Galaxy S7 smartphone (released in 2016) with an STMicroelectronics LSM6DS3 IMU, sampling at a frequency of 100 Hz. The area in which the tests were performed was an indoor warehouse space of 45 m by 35 m, as shown in Figure 1.7a. The estimated velocity has a smoothing effect, as shown in Figure 1.7b, which is expected because the raw peaks contain the largest magnitudes of the sensor noise.

They compared other popular INS and found that their proposed approach of combining deep

learning with a classic Kalman Filter approach produced the best velocity estimation result with an RMSE value of less than 0.4 m/s. This validation shows that a deep learning approach provides an advantage over a classic estimation approach.



(a) Data collection and validation location.



(b) Velocity estimation results.

Figure 1.7: Velocity estimation using an IMU [97].

Although the validation was performed using more dynamic movements, the data was captured in an indoor environment. Therefore, their proposed method was not tested in a high-noise environment such as a deep-level mine.

Adaptive threshold method for pedestrian INS [98]

Gait analysis is a method used in INS to study and quantify human walking patterns using IMU devices [50]. It is a popular method that can model the expected sensor data more accurately and, therefore, reduce estimation errors compared to traditional INS [98]. It is also often combined with advanced filtering techniques, such as EKF, to improve estimation accuracy further [99].

Many studies focused on gait analysis used the zero-velocity update technique to improve their estimations. It is a technique that determines when the IMU device is not moving. In the case of mounted sensors, such as on a person's ankle, when one foot is placed on the ground to move the other, it can be considered stationary.

Traditional threshold zero velocity approaches have limitations, such as a lack of robustness and generalisability. Therefore, Yang et al. proposed an adaptive threshold method for a zero velocity update pedestrian INS. Their method updates the zero velocity threshold based on the gait's data distribution as samples are obtained. Their method also includes an EKF that estimates the sensor bias and error over time.

Yang et al. tested their proposed method for different movement speeds on an athletic training field over a distance of 405 m. Their results showed an improvement of 80% for estimation accuracy when compared to the traditional threshold method, which is a significant improvement.

This study shows that an adaptive threshold zero-velocity update technique is a significant improvement upon traditional INS approaches. Although the method proposed by Yang et al. can be considered low-cost, their verification environment did not introduce high noise levels, such as EMI.

1.5.3 Image segmentation

Image segmentation and other pattern recognition methods have become essential for intelligent industrial systems, such as the systems used in the manufacturing industry, as they enable automation and optimisation of previously manual processes [100]. Image segmentation is a deep learning approach and, therefore, requires training an inference model, typically a neural network, using image and labelled datasets that are expected to be encountered in the specific application [39].

Studies have shown that segmentation performance is significantly affected by the quality of the training dataset and noise levels present during the use of the inference model [101]. Deep learning has been widely used for the segmentation of images of commonly encountered objects, such as people or cars in a city [102]. Therefore, many high-quality training datasets exist for them.

However, for industrial images that typically contain equipment such as compressors or valves, training datasets are sparse, if available at all. Industrial systems also tend to include many sources of high noise that originate from machinery and other objects in the environment [103].

Studies focusing on low-cost image segmentation approaches for high-noise environments will be analysed in this section. It will also include studies where public datasets are not readily available, as is the case for the deep-level mining industry.

Challenges and possible solutions for industrial image segmentation [104]

Liu et al. reviewed methods to improve the feasibility and generalisation ability of deep learning image segmentation approaches for industrial images. The identified and reviewed methods fell into two general categories: improving the utilisation of dataset content and improving the accuracy of the detected segmentations.

The methods identified to improve the utilisation of the content of the dataset include transfer learning, data augmentation, multi-scale learning, cross-validation, and semi-supervised

learning. The methods identified to improve the accuracy of the detected segmentations include image normalisation, multi-task learning, dropout, and early stopping. Liu et al. proposed an improvement framework that contains the methods from both categories.

An automatic welding defect identification case study was used to verify the proposed framework. The case study's image segmentation model was improved using the proposed methods, and it was seen that the identification accuracy significantly improved.

Liu et al. only provided a visual comparison; a more in-depth numerical analysis would provide more insight into how much the performance can be improved using the identified methods. This application also does not include as much variation in the images being segmented as in a deep-level mine. This lack of variation is due to the statically fixed position of the camera used to capture the images.

Synthetic datasets and transfer learning for image segmentation [100]

An approach considered among many recent studies for applications without readily available datasets is the generation and use of synthetic datasets. Synthetic datasets are digital recreations of real-world scenes in combination with relevant segmentation labels, which can both be generated automatically [105].

Synthetic data sets enable deep learning models to be trained without the need to create and label a dataset manually, which is time-consuming and typically expensive [106]. Many recent studies that have utilised synthetic datasets have shown segmentation results similar to those of studies that only used traditional datasets [107, 108].

A widely encountered phenomenon when using synthetic datasets is the reality gap between the digital recreation and the real-world scene [101]. It is not feasible to model a real-world application perfectly. However, transfer learning, another method that has recently gained much attention, has been shown to bridge this reality gap and significantly improve segmentation performance even with a relatively small synthetic dataset [108, 109].

Manettas et al. utilised both of these methods to determine the rotation and orientation of components in a manufacturing process. Transfer learning was used by initialising their inference model using the weights already obtained by pretraining on the ImageNet dataset. The ImageNet dataset consists of 1.2 million labelled real-world images of objects commonly found in natural scenes, such as humans, vehicles, animals, and fruit [110].

Manettas et al. then utilised their synthetic dataset (which consisted of 300 labelled images per class) after the initialisation using the pretrained weights to train the inference model further. This process enables the feature identification performance obtained by pretraining on the large real-world dataset to be fine-tuned for their application using the synthetic dataset.

The key concept of transfer learning is that segmentation performance gained by training on an unrelated dataset can be transferred to a specific application (not represented in the original dataset). This is done by fine-tuning on a relatively small, and in this case, synthetic dataset.

Manettas et al. obtained satisfactory rotation identification performance using their proposed approach. The accuracy of rotation identification varied from 65.9–96.5% in the three different test datasets considered.

A scalable dataset generator using a 3D modelling tool [105]

Synthetic datasets, due to their typically small size and inherent generation methods, tend to suffer from overfitting. This occurs when a dataset lacks sufficient variance (such as different perspectives of the scene, different objects, and orientations of objects in the scene) to produce a segmentation model that performs well on unseen real-world data [109]. This concept also aligns well with the phenomenon of the reality gap previously investigated [101].

Greff et al. proposed utilising a 3D modelling tool to create a scalable dataset generator to mitigate the reality gap and overfitting issue commonly found in applications that use synthetic datasets. The software chosen for their framework was Blender, a well-known 3D modelling tool [111].

Their framework included a rendering engine, a physics simulator, and data export infrastructure to create the scalable dataset generator. Some features in their dataset generator include multi-object video generation, depth maps, surface normals, optical flow, and semi-realistic scenes.

Their framework does produce highly accurate synthetic scenes. However, as multi-object videos are being generated, including physics simulation and other metadata generation, their framework does require significant computational power to produce these datasets.

This required significant computing power does not align with the low-cost requirement of the this study. However, their approach to avoid overfitting through reducing the reality gap may be utilised in a low-cost approach.

An adaptive segmentation method for low light conditions [112]

The lighting conditions throughout deep-level mines are inconsistent, as they vary from fully lit corridors to the only light source present being a headlight [113, 114]. Inconsistent lighting conditions typically have negative effects on segmentation performance, as the difference in the magnitude of the value of a region of pixels, or contrast, is reduced. Therefore, information important for segmentation is lost in noise [112].

Wang et al. proposed an adaptive segmentation method to enable high-performance semantic segmentation at night. Instead of creating a nighttime training dataset, Wang et al. implemented a segmentation model that attempts to mitigate low-light conditions after the training phase is complete. They utilised data augmentation techniques, namely normalisation and edge preservation, and combined them with an updated segmentation network to differentiate between object bodies and edges during inference.

Wang et al. achieved promising results, as their proposed method produced the highest Peak Signal-to-Noise Ratio (PSNR) of 18.10 compared to similar studies varying between 14.78 and 17.48. They also achieved a relatively high IoU score of 45.16 among the other studies considered that scored 25.44–49.9 in IoU. The data augmentation techniques used by Wang et al. show that high-performing image segmentation can be achieved for low-light conditions, such as deep-level mines.

Studies focusing on INS and image segmentation have been analysed in this section. All of the deep learning INS studies produced promising results using low-cost approaches, but none of these studies considered high-noise environments. Two of the image segmentation studies in this section showed that the collection of information about the surrounding environment is feasible in high-noise environments. One of the other studies showed that a low-cost image segmentation approach is also feasible.

1.6 Summary of literature analysis

The literature analysis consisted of three main topics, namely, DTs, SLAM, and kinematic and perceptual methods. Within the DTs topic, further filtering was applied to obtain studies focusing specifically on deep-level mining applications and semi-automatic generation methods. The deep-level mining environment poses certain challenges, such as high noise levels caused by EMI, which are not often found in other applications. Semi-automatic generation methods were considered as manual methods are too time-consuming and expensive.

Within the SLAM topic, studies focusing on low-cost methods and high-noise environments were considered, as these limitations have been identified from the background of this study. Within the kinematic and perceptual methods, studies focusing on inertial navigation and image segmentation were considered. This was due to these methods showing the most promising results in other research studies.

Studies focusing on at least one of the three main topics and other criteria were discussed and analysed to determine their strengths and shortcomings. These shortcomings will be used in the current and following sections to identify and define the research gap in the existing literature.

Table 1.3 shows a tabulated summary of the scope and focus of each study from the literature

analysis to identify a research gap. Only three studies focused on a deep-level mining application in combination with a SLAM approach [33, 91, 92]. Furthermore, none of these three studies considered a semi-automatic DT generation approach.

Table 1.3: Summary of the literature analysis.

	Digital Twins		SLAM		Kinematic and perceptual methods	
	Deep-level mining application	Semi-automatic generation	Low-cost method	High-noise environment	Inertial navigation	Image segmentation
[33]	✓			✓		
[58]	✓					
[59]	✓					
[60]	✓					
[64]	✓					
[69]		✓				
[70]		✓				✓
[72]		✓	✓			
[74]		✓				
[75]		✓		✓		
[78]			✓		✓	
[80]			✓		✓	
[83]			✓		✓	
[84]			✓		✓	
[87]			✓	✓		
[89]			✓	✓		
[91]	✓			✓	✓	
[92]	✓		✓		✓	
[94]			✓		✓	
[95]			✓		✓	
[97]			✓		✓	
[98]			✓		✓	
[104]				✓		✓
[100]			✓			✓
[105]						✓
[112]				✓		✓

Three separate studies, from those that did focus on semi-automatic DT generation, also included the use of image segmentation [70], a low-cost method [72], and high-noise environment considerations [75]. However, none of these studies focused on all the defined criteria.

Studies focusing on SLAM were further filtered to include low-cost methods and high-noise environment mitigations. Nine of these studies considered a low-cost method combined with an INS approach; however, none of the methods proposed from these studies were tested in high-noise environments [78, 80, 83, 84, 92, 95, 97, 115, 116].

One of these SLAM studies, in particular, focused on an INS method for a deep-level mining application. However, the proposed method was only tested using a simulation of a high-noise environment [92]. No real-world case study was used to verify their proposed method.

Two of the studies focusing on SLAM in high-noise environments used an image segmentation approach [104, 112]. However, these studies did not include low-cost considerations. Two other SLAM studies utilised low-cost approaches in high-noise environments. However, neither of these two studies included any other criteria [87, 89].

One SLAM study utilised an INS approach in a high-noise environment using an autonomous robot in a deep-level mining application [91]. However, this study used LiDAR equipment in combination with IMU devices, which does not meet the low-cost criteria.

Finally, studies focusing on kinematic and perceptual methods were analysed. The INS studies focused mainly on low-cost SLAM approaches [95, 97, 115, 116]. One of the INS studies included a deep-level mining application and another included a high-noise environment [91, 92]. Only one study considering image segmentation was used for semi-automatic DT generation [70]. Three of the remaining four image segmentation studies focused on low-cost or high-noise environment approaches [100, 104, 112].

It can be seen from Table 1.3 and the discussion which followed that none of the studies identified using the research topics, as well as the IC and EC, address all the defined criteria. Therefore, a gap in the research has been identified and is defined as follows: No existing study has considered utilising a SLAM approach to achieve low-cost semi-automatic DT generation in high-noise environments.

1.7 Problem statement and need for the study

The problem statement is presented below after a clear gap was identified in the literature analysis.

Digital Twins, one of the technologies of Industry 4.0, are time-consuming to create manually, and existing semi-automatic generation methods are expensive and require specialised equipment. This obstructs the feasible creation and utilisation of DTs, which hinders the adoption of Industry 4.0 initiatives in the South African deep-level mining industry.

Formulated from the problem statement, the need for the study is stated as follows.

There is a need for a low-cost semi-automatic DT generation method to enable the feasible creation and utilisation of DTs in the South African deep-level mining industry, therefore increasing Industry 4.0 adoption potential.

1.8 Study objectives

To address the problem statement and the need for the study, the following objectives for this study were identified:

1. Create a low-cost method for localisation in high-noise environments.
 - (a) Implement an IMU sampler for mobile devices.
 - (b) Design and implement a Kalman Filter-based INS for underground localisation.
2. Create a low-cost method for image segmentation in high-noise environments.
 - (a) Use a low-cost camera to implement an image sampler controlled through a mobile device for underground environments.
 - (b) Design and implement a synthetic dataset generator to train deep-learning image segmentation models for high-noise environments.
 - (c) Apply transfer learning to an image segmentation model using a generated synthetic dataset.
3. Combine (1) and (2) to create a framework for low-cost semi-automatic DT generation for a deep-level mining environment.
 - (a) Design and implement the low-cost semi-automatic DT generation method.
 - (b) Validate the proposed methods in a real-world deep-level mining case study.

1.9 Novel contributions

The novel contributions of this study, which address the need presented in Section 1.7, are discussed in four sections, namely:

- The problem statement,
- Limitations of existing research,
- Research question, and
- Contribution of this study.

1.9.1 Novel contribution 1

A low-cost method to generate synthetic datasets for
deep-learning image segmentation.

Limitations of existing research

Digital Twins are time-consuming to create manually, and existing semi-automatic generation methods, typically using technology such as LiDAR, are expensive and require specialised equipment. SLAM has been identified as a possible solution to this problem, as many studies have shown promising results when utilising INS for localisation and image segmentation to extract information from images of the environment automatically.

Recently, much research has been done to improve the inference performance of deep-learning image segmentation models. Many of these studies aimed to improve performance by proposing new error metrics or deep-learning architectures. These studies considered applications for which many datasets already exist and, therefore, did not need to create or generate a dataset.

However, research on the generation of low-cost synthetic datasets for less popular applications is limited, as shown in Section 1.5. Of the studies that considered synthetic dataset generation, one proposed a framework for a generator specifically for the manufacturing industry, and the other did not focus on a low-cost approach.

Problem definition

Deep-learning image segmentation requires a dataset, consisting of example images and corresponding segmentations, for training. There exist many datasets for typical problem areas, such as an urban environment for self-driving vehicles. However, no image segmentation dataset exists for more unique applications, such as the deep-level mining environment.

Research question

Is it possible to design and implement a low-cost synthetic dataset generator for image segmentation of high-noise environments?

Contribution of this study

In this study, a low-cost synthetic generator is proposed and implemented using a 3D modelling framework and raycaster. An article of this research was published in the *Frontiers in Artificial Intelligence* journal and has been inserted in Appendix 1 [1]. The proposed generator can be applied to many different industries by selecting the relevant 3D models.

1.9.2 Novel contribution 2

A low-cost semi-automatic DT generation method for deep-level mines.

Limitations of existing research

The South African mining industry faces challenges in the form of increased operational and labour costs. Industry 4.0 initiatives have been developed in recent years that aim to increase operational efficiency. The DT, one of the technologies of Industry 4.0, is a digital representation of a physical system with cyber-physical connections to the real system and can be utilised to increase operational efficiency.

The analysed studies that proposed semi-automatic DT generation methods through a LiDAR approach cannot be considered low-cost methods. The studies that did consider a low-cost approach did not design nor test their proposed methods and frameworks in high-noise environments, such as a deep-level mining environment.

SLAM has been identified as a possible solution to create a low-cost semi-automatic method through INS and image segmentation. However, no existing SLAM method has been found that addresses this need, as discussed in Section 1.4 and 1.5.

Problem definition

DTs are time-consuming to create manually and existing semi-automatic generation methods, such as LiDAR, are expensive and require specialised equipment. The deep-level mining environment also presents challenges such as high noise levels and the unavailability of cellular and GPS signals. Furthermore, underground fibre networks providing full coverage are also expensive to install and maintain.

Research question

Is it possible to design and implement a low-cost semi-automatic DT generation method for offline high-noise environments?

Contribution of this study

This study proposes a low-cost semi-automatic DT generation method for deep-level mines. The proposed method was verified using multiple different areas within a case study South African deep-level mine. The method can be applied to many different industries to enable cost-efficient

semi-automatic DT generation and therefore to increase the operational efficiency of complex systems.

1.10 Research design

A pragmatic research philosophy was adopted for this study as it prioritises a practical problem-solving approach and is well suited to addressing real-world challenges [117]. These challenges include GPS-denied and highly noisy underground environments.

Furthermore, a positivistic quantitative research paradigm aligns well for this study as it aims to identify causal relationship from objective measurement in a case study method with an unpredictable environment and sensor noise influencing the measurements [118].

The research approach for this study consists of a hybrid Plan, Do, Check, Act (PDCA) cycle, which forms part of the ISO 9001 standard [119], in combination with a case study approach. The PDCA cycle is an iterative research approach that consists of planning, implementing, monitoring, and improving the methods proposed to achieve the study objectives. It enables continuous improvement of the methods proposed throughout the duration of the study as more information and insight is obtained.

The case study approach is a detailed analysis of a problem within a complex system [120], such as a deep-level mine. One significant advantage of this approach is that it provides contextual insight into the problem statement and proposed methods within the application area. This approach also includes the implementation and testing of the methods proposed within the complex real-world system for verification.

The combination of iterative improvement and contextual insight through the hybrid approach will increase the likelihood that the proposed methods achieve the objectives of the study and therefore resolve the problem statement defined in Section 1.7. A research methodology will be proposed at the beginning of Chapter 2 to discuss the detailed strategy to achieve the objectives of the study.

1.11 Thesis overview

An overview of the thesis has been summarised as follows:

Chapter 1: The background was presented and research areas were identified, investigated, and studies related to them were analysed to identify the need for the study, determine study objectives, and define novel contributions.

Chapter 2: The scope of the study is defined, and a methodology and various methods are proposed to address the need and achieve the objectives stated for this study.

Chapter 3: The proposed methods are verified, and a case study is presented, analysed, and discussed to test and validate the proposed methodology.

Chapter 4: A discussion of the research is presented, key findings are highlighted, and contributions to the field and future research are discussed.

2 Methodology

2.1 Scope definition

This chapter contains the proposed methods that aim to achieve the objectives and address the need for the study identified in Chapter 1. However, before this, certain restrictions will be defined to create an achievable scope for this study.

The first restriction of scope is that the proposed methods must adhere to the low-cost requirement discussed in the previous chapter. The low-cost requirement increases the likelihood of an increased adoption of Industry 4.0 technologies, such as those used in the methods that are proposed in this chapter, in the South African deep-level mining industry.

The second restriction of scope is the high noise mitigation required. Due to EMI, inconsistent lighting conditions, and other environmental factors, significant noise must be reduced or filtered to provide adequate localisation and mapping performance of the proposed methods.

Therefore, the following scope definition is proposed: to adhere to the low-cost restriction, deep learning INS will not be considered, as deep-level mining INS datasets do not already exist, and other advanced noise mitigation methods (such as Kalman Filters) have shown promising results. However, in the relevant literature, only deep learning image segmentation approaches have been found. Although there is no deep-level mining segmentation dataset, it is possible to use a synthetic dataset generation approach.

2.2 Methodology overview

The methodology in Figure 2.1 has been created to achieve the objectives from Section 1.8 and was adapted and expanded from the methodology used by de Meyer [121]. On the left, along the length of the flow diagram, are the four main steps in the methodology, namely investigation, design, verification, and implementation.

At the top of the figure are three titles for portions of the flow diagram; these represent the main topics of the methodology, namely inertial navigation, image segmentation, and low-cost semi-automatic DT generation. All three topics consist of various shaded blocks within each of the four steps along the length of the figure.

The investigation step for each topic consists of gathering information about the characteristics of the data and existing methods from other studies. The existing techniques are then compiled into combined methods with relevant placeholders for the required new methods. The design step for each topic involves obtaining deep-level mining samples of the required data and then designing a new method to achieve the objectives of the study.

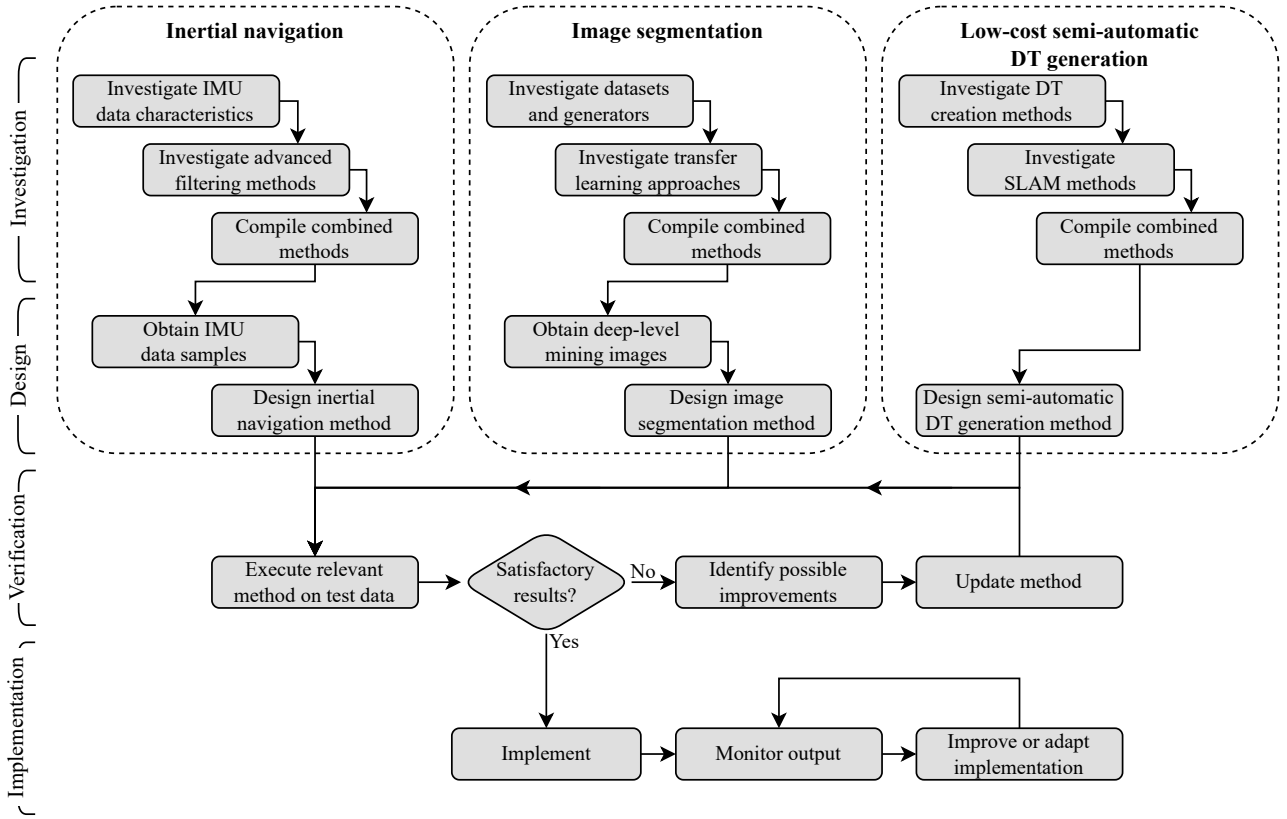


Figure 2.1: Proposed methodology diagram, adapted and expanded from de Meyer [121].

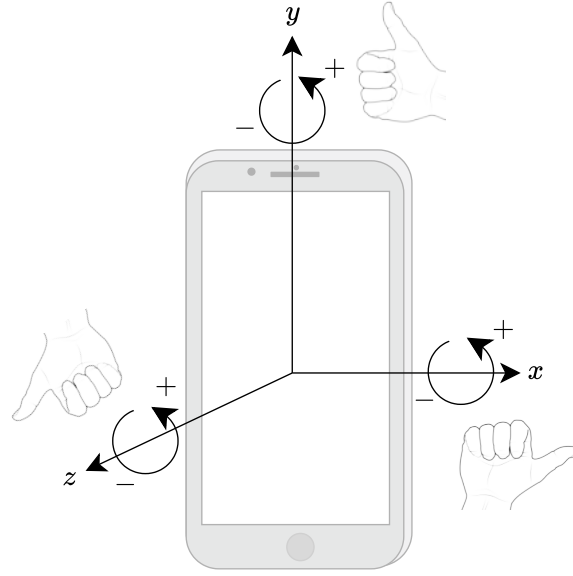
The satisfactory output from the first two designed methods (INS and image segmentation) are used as input for the design of the low-cost semi-automatic DT generation method. The verification step for each topic consists of analysing the result of the designed methods using test data with expected results. During this process, opportunities for improvement are identified if the results are not satisfactory. The designed methods are updated based on the outcome of the relevant verification step.

The final implementation step for each topic consists of a discussion on how the designed methods were implemented. This step also includes monitoring the output, as well as improving and adapting the implementation of the designed methods. The rest of this chapter is divided into three sections representing each topic in the methodology.

2.3 Inertial navigation

2.3.1 IMU data acquisition

MEMS IMU sensors found in mobile devices consist of three parts: an accelerometer (measuring linear acceleration in m/s^2), a gyroscope (measuring rotational rate in rad/s), and a magnetometer (measuring the strength of the Earth's magnetic field in μT). The reference frame for all three measurements is the mobile device itself, as the IMU sensor is attached inside it. Figure 2.2 shows the mobile device reference frame.

Figure 2.2: IMU reference frame³.

The mobile device will not be attached to a fixed position on the user’s body, but rather, it will be held in the user’s hand during sampling. This will allow the user to capture information before, during, and after the IMU sampling process. Once the IMU samples have been obtained from the mobile device, the next step in the method can begin. This consists of using an advanced filtering algorithm to reduce the high levels of noise from the IMU samples.

2.3.2 Unscented Kalman Filter

The UKF algorithm, proposed by Wan et al. [122, 123] and based on the unscented transform earlier proposed by Julier et al. [124], improves upon both the standard KF and EKF approaches. This is caused by calculating the mean and covariance errors from a probability density function and propagating them through all the following calculations without linearisation. Therefore, a UKF approach is chosen for this study to increase the likelihood of high accuracy path estimation.

The UKF algorithm was chosen above other filtering methods, such as deep-learning INS, as no publicly available inertial dataset could be obtained specifically for the deep-level mining industry, as discussed in Section 1.5.2. Creating this type of inertial dataset is not feasible for this study, and attempting to use a dataset for another application would have provided decreased performance.

The UKF approach estimates the state matrix, x , of a system at time t by selecting prior sigma points, χ^- , around the current state and applying an unscented transform based on weighted means and covariances to obtain a posterior set of sigma points, χ^+ . These sigma points are

³Adapted from “Core Motion”, Apple Inc., <https://developer.apple.com/documentation/coremotion/getting-processed-device-motion-data> (accessed 2 May 2025).

then used to calculate the cross-covariance and Kalman gain. Lastly, the latest state matrix and covariance estimates are calculated. This process is done for every data point sampled.

In this study, the state matrix was chosen as the true value of each sensor within the IMU without drift or additive noise, similar to Navidi et al. [94]. In other words, an IMU dynamics-based model was selected to estimate and remove sensor error and bias from the measurements. Therefore, the state matrix consists of measurements directly from the IMU as

$$x = \begin{bmatrix} a_x & a_y & a_z & m_x & m_y & m_z & \omega_x & \omega_y & \omega_z \end{bmatrix}^T \quad (2.1)$$

where a denotes the true linear acceleration of the IMU, m denotes the true strength of the Earth's magnetic field at the IMU, and ω denotes the true angular velocity of the IMU. Therefore, L , the size of the chosen state matrix, is nine.

The UKF requires three covariance matrices: the covariance of the state P , the covariance of the process Q , and the covariance of the measurements R . The covariance matrices P and Q are initialised as diagonal matrices of size $L \times L$. The initial covariance of the measurements, R , is defined as

$$R = \begin{bmatrix} \sigma_{a_x}^2 & \sigma_{a_y}^2 & \sigma_{a_z}^2 & \sigma_{m_x}^2 & \sigma_{m_y}^2 & \sigma_{m_z}^2 & \sigma_{\omega_x}^2 & \sigma_{\omega_y}^2 & \sigma_{\omega_z}^2 \end{bmatrix}^T \quad (2.2)$$

where σ_a^2 denotes the variance of the accelerometer, σ_m^2 denotes the variance of the magnetometer, and σ_ω^2 denotes the variance of the gyroscope.

The state estimation $\hat{x}$ and covariance of the state P are used to calculate the set of sigma points, χ , of size $2L + 1$, as

$$\chi_{t,i}^- = \begin{cases} \hat{x}_t^- & , i = 0 \\ \hat{x}_t^- + \sqrt{(L + \lambda)P_t^-} & , i = 1, \dots, L \\ \hat{x}_t^- - \sqrt{(L + \lambda)P_t^-} & , i = L + 1, \dots, 2L \end{cases} \quad (2.3)$$

and λ , which is a scaling factor representing the distance between the sample point and the prior mean, is calculated as

$$\lambda = \alpha^2(L + k) - L \quad (2.4)$$

where the constant α determines the spread of the sigma points around the prior mean $\hat{x}_t^-$ and the constant k is a secondary scaling parameter [124]. α is typically a small number around 1×10^{-3} and k is typically set to zero for state estimation [125].

The weights of the mean and covariance are also required, which are calculated once after the

initialisation step using λ and L as

$$W_m = \begin{cases} \lambda(L + \lambda) & , i = 0 \\ \frac{\lambda}{2(L+\lambda)} & , i = elsewhere \end{cases} \quad (2.5)$$

and

$$W_c = \begin{cases} \frac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta) & , i = 0 \\ \frac{\lambda}{2(L+\lambda)} & , i = elsewhere \end{cases} \quad (2.6)$$

where β is equal to 2 for Gaussian random variables [126].

Once the initialisation has been completed, the iteration is started through each time step t . The first step is to calculate the sigma points $\chi_{t,i}^-$ using Equation 2.3. Following this, the unscented transform of the process is calculated as

$$\hat{x}_t^- = \sum_{i=0}^{2L} W_m \chi_{t,i}^- \quad \hat{P}_{x,t}^- = \sum_{i=0}^{2L} W_c (\chi_{t,i}^- - \hat{x}_t^-)(\chi_{t,i}^- - \hat{x}_t^-)^T + Q \quad (2.7)$$

and then the unscented transform of the measurement is calculated as

$$\hat{y}_t^- = \sum_{i=0}^{2L} W_m Y_t \quad \hat{P}_{y,t}^- = \sum_{i=0}^{2L} W_c (Y_t - \hat{y}_t^-)(Y_t - \hat{y}_t^-)^T + R \quad Y_t = W_m \hat{x}_t^- \quad (2.8)$$

where Y_t is calculated using the weights of the mean, W_m , and the prior estimation of the state, $\hat{x}_t^-$. The cross-covariance can then be calculated as

$$\hat{P}_{x,y,t}^- = \sum_{i=0}^{2L} W_c (\chi_{t,i}^- - \hat{x}_t^-)(Y_t - \hat{y}_t^-)^T \quad (2.9)$$

with W_c representing the weights of the covariance. The Kalman gain is then calculated as

$$K_t = \hat{P}_{x,y,t}^- (\hat{P}_{y,t}^-)^{-1} \quad (2.10)$$

using the cross-covariance and the transformation of the measurement. The Kalman gain, including the transformed process and measurement, is then used to calculate the estimated state of the system as

$$\hat{x}_t = \hat{x}_t^- + K_t (Y_t - \hat{y}_t^-) \quad (2.11)$$

and the updated covariance of the state is calculated as

$$\hat{P}_{x,t} = \hat{P}_{x,t}^- - K_t \hat{P}_{y,t}^- K_t^T \quad (2.12)$$

where $\hat{x}_t$ contains the estimated true IMU measurements at time t .

The summary of the UKF approach is provided in Algorithm 1. The estimated true IMU

samples are calculated for each time t in step six, where the state matrix is updated from the sigma points, unscented transforms, cross-covariance updates, and Kalman gain calculations.

Algorithm 1: Summary of the UKF estimation algorithm [125, 126].

Data: $x \in L \times t$: Noisy IMU measurements

Result: $\hat{x} \in L \times t$: Estimated true IMU measurements

Estimate initial state vector $\hat{x}_0$ and initial covariance matrix $\hat{P}_0$

Calculate the weights of the mean and covariance (Eq. 2.5 and 2.6)

for t **in** $\{1, 2, \dots, n\}$ **do**

 Step 1: Get sigma points $\chi_{t,i}^-$ (Eq. 2.3)

 Step 2: Unscented transform of process to get $\hat{x}_t^-$ and $\hat{P}_{x,t}^-$ (Eq. 2.7)

 Step 3: Unscented transform of measurement to get $\hat{y}_t^-$ and $\hat{P}_{y,t}^-$ (Eq. 2.8)

 Step 4: Calculate cross-covariance $\hat{P}_{x,y,t}^-$ (Eq. 2.9)

 Step 5: Calculate Kalman gain K_t (Eq. 2.10)

 Step 6: Update state matrix $\hat{x}_t$ (Eq. 2.11)

 Step 7: Update covariance $\hat{P}_{x,t}$ (Eq. 2.12)

This UKF algorithm will be used to filter and reduce the noise and sensor bias of the IMU samples. A visual representation of the workflow for the UKF algorithm is shown in Figure 2.3.

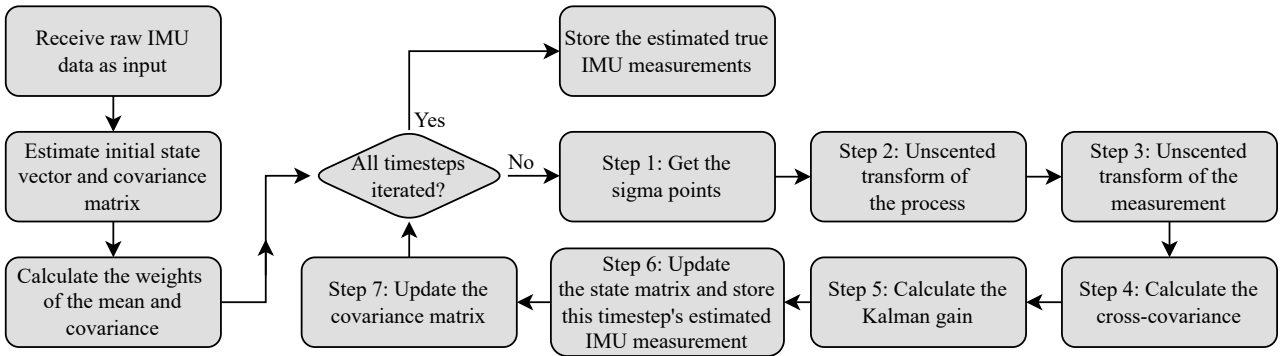


Figure 2.3: UKF algorithm workflow.

RMSE and PSNR will be used to quantify the filtering performance achieved as

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2} \quad (2.13)$$

and

$$PSNR = 10 \log_{10} \left(\frac{MAX^2}{MSE} \right) \quad (2.14)$$

where n represents the number of samples, x_i is the raw data, and $\hat{x}_i$ is the filtered data. In the following subsection, motion equations are used to convert the IMU data from the mobile reference frame to the Earth reference frame, allowing the calculation of the change in position over time from these filtered samples.

2.3.3 Motion equations

Samples from a mobile IMU are represented in its own reference frame, as already shown in Figure 2.2. Once these raw IMU samples have been obtained and filtered using the UKF algorithm discussed in the previous subsection, they can then be used to estimate the path traversed by the user [92, 98].

However, these filtered samples must first be converted from the mobile reference frame to the Earth reference frame before path estimation can be done [127]. The mobile reference frame in relation to the Earth reference frame is shown in Figure 2.4. In order to perform this conversion, the actual orientation of the mobile device within the Earth reference frame must be determined.

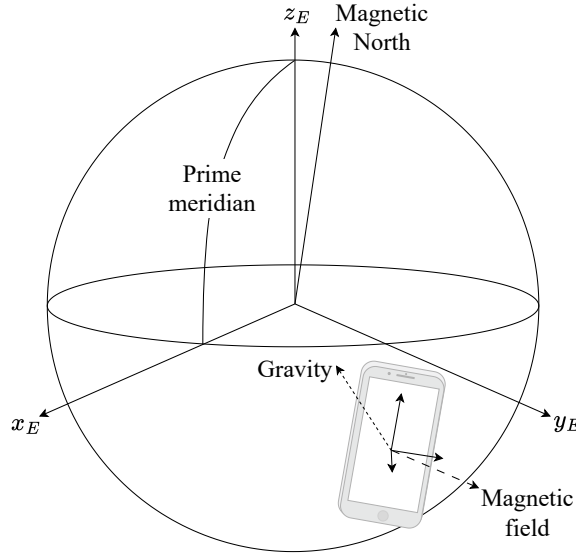


Figure 2.4: IMU reference frame representation within the Earth reference frame.

Orientation can be represented using three parameters, namely roll, pitch, and yaw. These three parameters will be used to calculate a rotation matrix to transform the IMU measurements to the Earth reference frame for path estimation [127].

The first step to obtain the rotation matrix is to calculate the roll, ϕ , and pitch, θ , as

$$\phi = \text{atan}(a_y, a_z) \quad \theta = \text{atan}(-a_x, \sqrt{a_y^2 + a_z^2}) \quad (2.15)$$

where a denotes the linear acceleration measurement in the three relevant axes. The world magnetic field vector is then calculated as

$$m_w = \begin{bmatrix} m_x \cos(\theta) + m_x \sin(\theta) \sin(\phi) + m_x \sin(\theta) \cos(\phi) \\ m_y \cos(\phi) - m_y \sin(\phi) \\ -m_z \sin(\theta) + m_y \cos(\theta) \sin(\phi) + m_y \cos(\theta) \cos(\phi) \end{bmatrix} \quad (2.16)$$

where m denotes the magnetometer measurement in the three relevant axes. The yaw, ψ , is

then calculated as

$$\psi = \text{atan}(-m_{w,y}, m_{w,x}) \quad (2.17)$$

from the world magnetic field vector m_w . Finally, the rotation matrix from the IMU frame to the Earth navigation frame, also known as the direction cosine matrix, is calculated as

$$R_b^n = \begin{bmatrix} \cos(\psi)\cos(\theta) & \sin(\psi)\cos(\theta) & -\sin(\theta) \\ -\sin(\psi)\cos(\phi) + \cos(\psi)\sin(\theta)\sin(\phi) & \cos(\psi)\cos(\phi) + \sin(\psi)\sin(\theta)\sin(\phi) & \cos(\theta)\sin(\phi) \\ \sin(\psi)\sin(\phi) + \cos(\psi)\sin(\theta)\cos(\phi) & -\cos(\psi)\sin(\phi) + \sin(\psi)\sin(\theta)\cos(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix} \quad (2.18)$$

and the IMU samples relative to the Earth navigation reference frame can then be calculated as

$$a_n = R_b^n \begin{bmatrix} a_{b,x} \\ a_{b,y} \\ a_{b,z} \end{bmatrix} \quad m_n = R_b^n \begin{bmatrix} m_{b,x} \\ m_{b,y} \\ m_{b,z} \end{bmatrix} \quad \omega_n = R_b^n \begin{bmatrix} \omega_{b,x} \\ \omega_{b,y} \\ \omega_{b,z} \end{bmatrix} \quad (2.19)$$

where acceleration can be used to calculate the change in position over time in the Earth navigation reference frame, in other words, the estimated path from the filtered IMU samples.

2.3.4 Proposed method

The acquisition of IMU data, the filtering using a UKF, and the conversion of the filtered samples to an estimated path have been discussed in the previous three subsections. This was done using the methodology presented in Figure 2.1. The proposed INS method is therefore presented and discussed in this subsection. This method, proposed to achieve objective 1 in Section 1.8, will form the localisation part of the overall semi-automatic DT generation method. The proposed INS method is shown in Figure 2.5.

The proposed method is split into two parts, where the former utilises a mobile device and the latter utilises a server to execute the steps in the method. This design choice was made because the calculations required for the UKF algorithm and the motion equations require more computational power than is available in a mobile device.

The method starts at the top left of the figure, where the mobile device is used to collect the IMU samples in an offline environment. Once the sampling is done and a server connection is established, the samples are uploaded to the server. The server is then informed that a new set of samples is ready for processing.

The first step for the server is to obtain the IMU samples for processing. The IMU samples are then filtered using the UKF presented in Algorithm 1 to reduce the negative effects of sensor noise and bias. The last steps of the method require iteration over the filtered IMU samples.

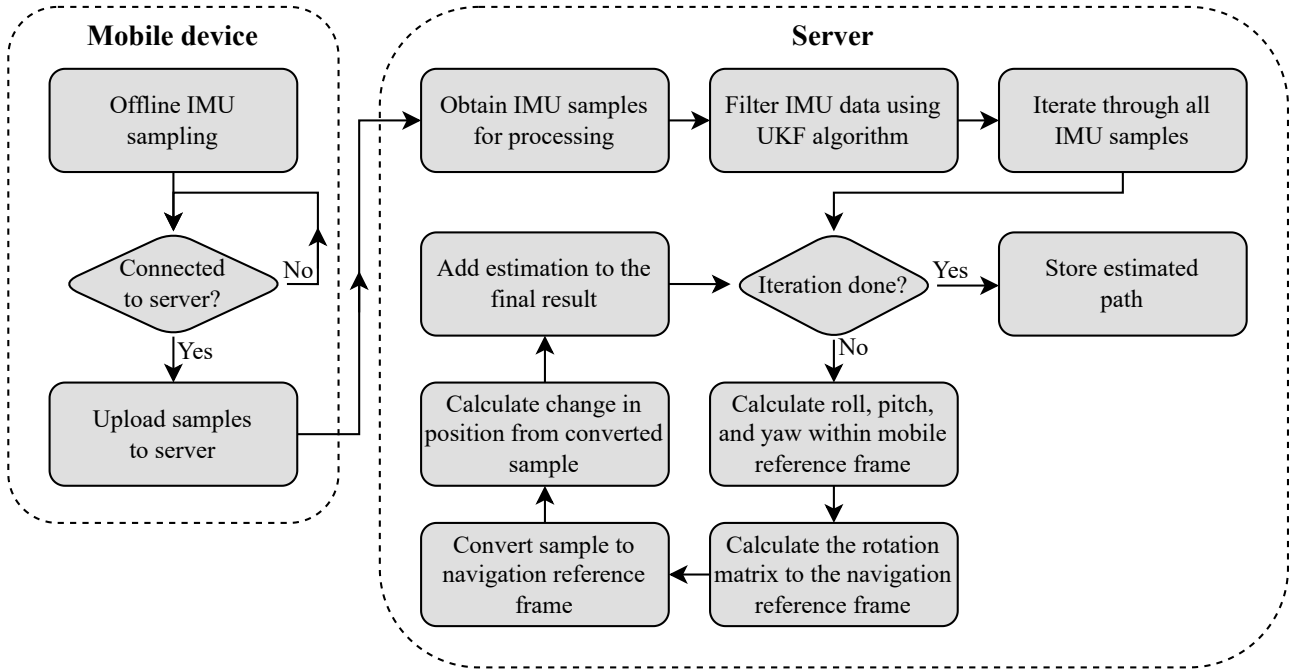


Figure 2.5: Proposed INS method.

In every one of these iterations, the roll, pitch, and yaw within the mobile reference frame are calculated from the current filtered IMU sample using Equations 2.15, 2.16, and 2.17. These three values are then used to calculate the rotation matrix from the mobile reference frame to the Earth navigation reference frame using Equation 2.18.

The navigation reference frame describes the movement of the body holding the mobile device in the world navigation frame. Once the rotation matrix is calculated, all the filtered samples are converted from the mobile reference frame to the navigation reference frame using Equation 2.19.

Finally, the change in position from the converted sample is calculated and added to the estimation of the current iteration to the final result. Once all iterations have been completed, the estimated path is stored for use in the DT generation method.

2.4 Image segmentation

2.4.1 Synthetic dataset generation

Deep learning image segmentation requires a dataset for training, and, to the best of the author's knowledge, there exists no readily available deep-level mining image segmentation dataset, as discussed in the literature analysis. Therefore, to avoid the labour-intensive and time-consuming process of manual creation, a synthetic dataset generator will be proposed in this subsection to create the dataset required for training.

The dataset to be created should consist of multiple variations of perspectives within

the generated synthetic scene, including varying object placements. This will increase the probability of generating a high-quality dataset [100, 108, 109].

A dataset can be considered high quality when it has enough examples of segmentations so that the deep-learning training produces a high-performing image segmentation model [101]. Evaluation metrics to measure segmentation performance are discussed in Section 2.4.2.

Synthetic scene generation

The first step of synthetic scene generation is to analyse the real-world scene and the elements, such as objects and phenomena, within it. A perfect digital copy of any real-world scene is unfeasible due to the reality gap [101]. Therefore, depending on the application, the most relevant objects and phenomena should be digitally modelled to increase the likelihood of high-performing image segmentation.

The classes of objects to be identified by the segmentation model are of the highest importance; thereafter, the context surrounding them should be considered. This context refers to the placement of objects within the scene to be segmented, including the other objects around them, such as walls or ceilings.

Phenomena, such as lighting conditions, must also be considered, as this significantly affects the images to be segmented, especially in low-light conditions. Analysing real-world scenes requires example images from the relevant application. Once the scene analysis is completed, the process to generate the synthetic scenes can begin.

Synthetic scenes are typically generated using a 3D modelling framework and pre-existing or newly created 3D models of the objects that exist within the real-world scenes [105]. Many different 3D modelling frameworks exist, and each has been designed and built to achieve different outcomes.

Some aim to provide the most accurate object interaction physics, while others focus on the visual fidelity of the final scene. Therefore, Table 2.1 presents categories with which a 3D modelling framework can be chosen based on the application and constraints.

The categories include functionality such as advanced physics and simulation, usability and flexibility, raycasting, scalability and hardware requirements, 3D model imports and exports, and finally, licensing and cost. Raycasting is an important mechanism that will be used in the rest of the data generation method, as it can be used to locate objects within a 3D modelling framework.

Table 2.1: Analysis of 3D modelling frameworks for synthetic scene generation [1].

Category	Description
Advanced physics and simulation	Does the framework support object interactions that aim to mimic real-world phenomena, such as water flow?
Usability and flexibility	Does the framework have comprehensive documentation, an active community, and can it generate various scenes?
Raycasting	When a straight line is cast from an arbitrary point in a scene, can the first encountered object along the straight line be identified?
Scalability and hardware requirements	How complex can the scene become while balancing the hardware requirements to be feasible for the application and restrictions?
3D model imports and exports	Can 3D models in various formats be imported and exported from the framework to increase generation productivity?
Licensing and cost	Is the cost of purchasing the licence feasible for the application?

Once the 3D modelling framework has been selected, the scene implementation process can begin. Figure 2.6 shows the proposed synthetic scene implementation method, which is based on the PDCA cycle [119].

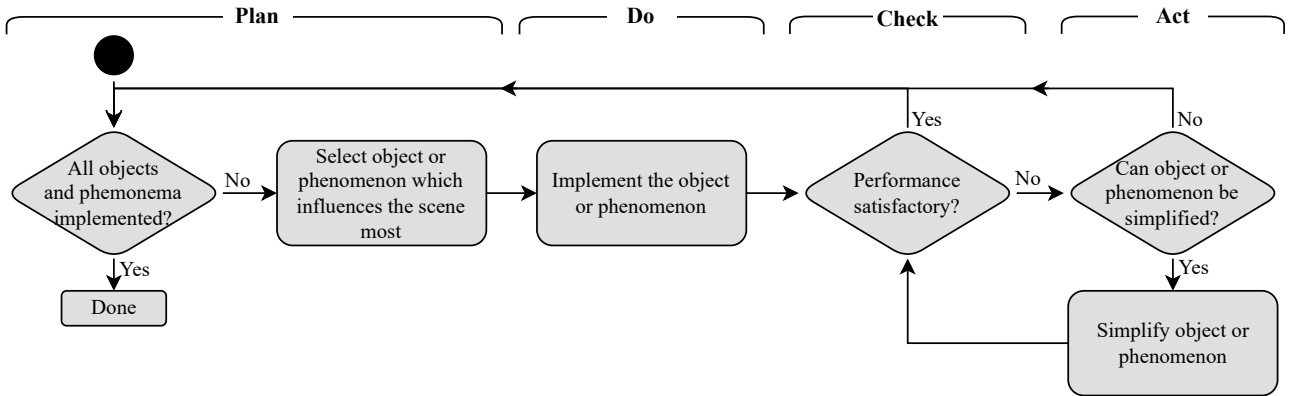


Figure 2.6: Proposed synthetic scene implementation method [1].

Firstly, the object or phenomenon that has the greatest influence on the scene is selected from the remaining items identified from the real-world scene analysis. This object or phenomenon is then implemented in the scene with a high level of accuracy, such as using a high-fidelity 3D model of a compressor pump, by importing it into the modelling framework.

Suppose that the new high-fidelity item causes the performance of the scene to degrade significantly. In that case, the item must be simplified. This simplification is done by lowering the level of detail of the object or phenomenon. All objects and phenomena are iterated through this way until all have been implemented.

Once the initial implementation is completed, the scene variation process is applied. This consists of creating multiple versions of the synthetic scene that will be used in the automatic labelling step. A scene variation consists of moving objects to pseudo-random positions (within

reasonable constraints) in the scene and rotating them arbitrarily within the rules identified in the analysis phase.

These variations will increase the performance of the image segmentation model by avoiding overfitting during the training phase, therefore making it more generalisable and reducing the reality gap. This, as well as other techniques such as data augmentation (discussed in Section 2.4.2), will be used to increase the quality of the generated dataset and therefore increase the probability of high-performing image segmentation. The automatic labelling process to create the synthetic dataset from the generated scenes is discussed next.

Automatic labelling using a raycaster

3D modelling frameworks utilise camera objects to visualise digital scenes. Cameras can be defined and customised as required for the application, such as a specific field of view and aspect ratio. Multiple different perspectives of a synthetic scene can be created by moving a camera throughout a scene and placing it in different positions and orientations.

Figure 2.7 shows a simplified example of such a perspective of a camera in a digital scene. The viewing frustum (the area that is visible from the camera's perspective) is delimited by (x_1, y_1, z_1) to (x_4, y_4, z_4) . The pyramid, cylinder, and cuboid that fall within the camera's viewing frustum have been added as example objects that will be automatically labelled in the method to be proposed.

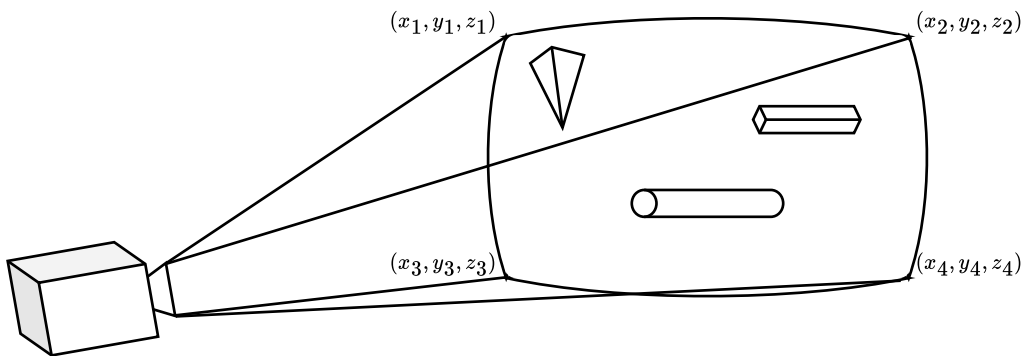


Figure 2.7: Camera perspective example [1].

Figure 2.8 shows the raycasting matrix of size $M \times N$, using the same perspective as in Figure 2.7, to identify all objects within the current camera perspective. Raycasting is the process of casting a straight line from a point of origin, such as the origin coordinate of a camera's perspective, and identifying the first object found along that path.

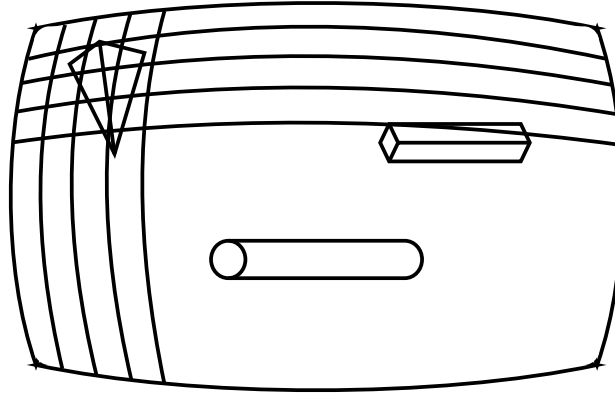


Figure 2.8: Raycasting matrix from the camera's perspective [1].

A perspective of a scene is then automatically labelled by raycasting all points within a camera's perspective. The raycasting process produces a populated matrix of identified objects that is converted to a set of polygons (segmentation labels for irregularly shaped objects) that is used to create a synthetic dataset.

Algorithm 2 has been proposed to convert the $M \times N$ matrix to a set of polygons. The algorithm iterates through the matrix from left to right and top to bottom. During this iteration, if an object of interest (an object relevant to the dataset being generated) is found in a cell, the polygon conversion begins.

Algorithm 2: Proposed $M \times N$ raycast matrix result to polygon labels conversion [1].

Data: $\phi \in M \times N$: Raycast matrix result

$y \in P$: Object classes of interest

Result: $\hat{x} \in S \times T$: Instance segmentation polygons identified

Initialise η as all unique objects in ϕ for all relevant y

Initialise σ as edges of objects to be found

for row **in** M **&** col **in** N **do**

if object **at** $\phi[\text{row}][\text{col}]$ **in** η **then**

 Step 1: Set *origin* to $\phi[\text{row}][\text{col}]$

 Step 2: Set *edgesOfObject* to [*origin*]

 Step 3: Set *edge* to next edge of *object* found in a clockwise search from *origin*

while *edge* **not equal to** *origin* **do**

 Step 4.1: Set *edge* to next edge of *object* found in a clockwise search from *edge*

 Step 4.2: Add *edge* to *edgesOfObject*

 Step 5: Add *edgesOfObject* to σ

 Step 6: Clear elements in ϕ encapsulated by *edgesOfObject*

Step 7: Convert all edges in σ to the desired dataset format

The conversion begins by considering all cells around the current cell in a clockwise direction to determine where the next edge of the current object can be found. The process continues if

the next identified edge is not equal to the first edge where the object was found.

Each time a new edge of the current object is identified, the coordinate is added to the current polygon. When the original cell is identified again, the polygon is completed and added to the final dataset. Through this algorithm, a 2D matrix of raycast results is converted to polygons of identified objects from a generated synthetic image that can be used for image segmentation training.

The synthetic scene generation method, as well as the automatic labelling method using a raycaster, form the two halves of the proposed synthetic dataset generation method shown in Figure 2.9. Scene generation starts with real-world scene analysis, selection of the 3D modelling framework, and collection and creation of 3D models for the relevant objects. The scene is then implemented as shown in Figure 2.6, and the scene variations are generated.

The automatic labelling iterates over all scene variations and all camera perspectives. Within each iteration, the 2D image of the perspective is captured and the raycasting process is performed as shown in Figure 2.8. The raycast matrix is then converted to polygons as described in Algorithm 2 and, lastly, the captured image and segmented polygons are added to the dataset. The generated dataset is then stored after all the iterations are completed.

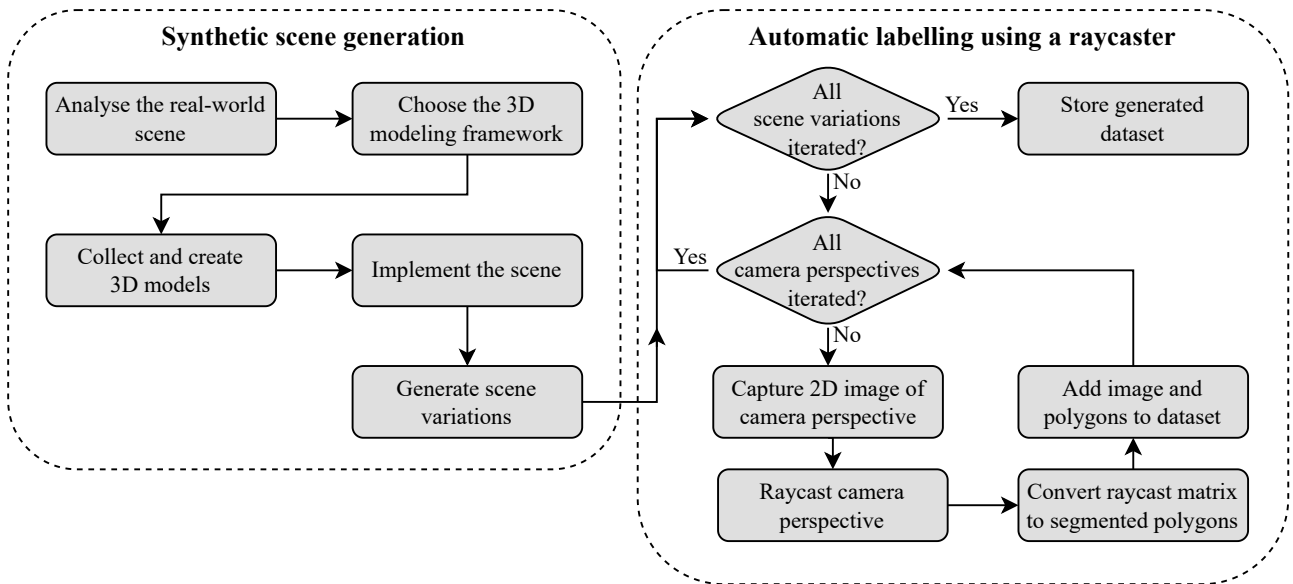


Figure 2.9: Proposed synthetic dataset generation method.

Refer to Appendix 1 for the published article for a more detailed discussion of the proposed synthetic dataset generation method. The dataset generated from this proposed method is used in the next step of the overall image segmentation method, which consists of training an image segmentation model using transfer learning.

2.4.2 Transfer learning

Transfer learning is the process of applying fine-tuning training to an image segmentation model that has already been trained on a large dataset that is not related to the current application [108]. Transfer learning has the advantage of reducing the reality gap for applications that do not have real-world datasets.

This advantage is gained from training on common application datasets that consist of hundreds of thousands of segmented images, with millions of segmented object instances, such as the Microsoft COCO dataset [71]. Thereafter, the inner layers of the model are frozen to retain the learned features, and the final layer is fine-tuned using a dataset designed specifically for the required application. Another advantage of transfer learning is that the dataset used for fine-tuning does not need to be as large as the dataset used during the initial training [107,109].

Beyond datasets, performance metrics are also required to train and validate an image segmentation model. Metrics are a quantitative measure of segmentation performance used during the training and validation phases to calculate the segmentation error, i.e. the difference between the inference and ground-truth segmentation. The first metric that is used in this method is IoU, also known as the Jaccard index, and is calculated using

$$IoU = \frac{|A \cap B|}{|A \cup B|} \quad (2.20)$$

where A is the ground truth and B is the prediction. It is calculated as the area of intersection between the ground truth and the prediction, divided by the total area of the union of the ground truth and the prediction [101].

The second metric that is used in this method is Panoptic Quality (PQ), originally introduced by Kirillov et al. [128], and is calculated using

$$PQ = \frac{\sum IoU}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|} \quad (2.21)$$

where the sum of IoU for all object instances is divided by the sum of the true positive, TP , the false positive, FP , and lastly, the false negative matches, FN .

The aim of this study is not to propose a new segmentation model architecture; therefore, existing open-source image segmentation models and their pre-trained weights are utilised. The transfer learning method shown in Figure 2.10 utilises these open-source models as well as the synthetic dataset generated using the method proposed in Section 2.4.1.

The method starts at the top left of the figure, where data augmentation methods are implemented. Data augmentation is a technique in which a set of translations is applied to a dataset to increase the effective number of labelled samples [108]. Some of these translations

are rotations and inversions of the segmented image, while other translations update the colour-space of the segmented image, such as auto-contrasting.

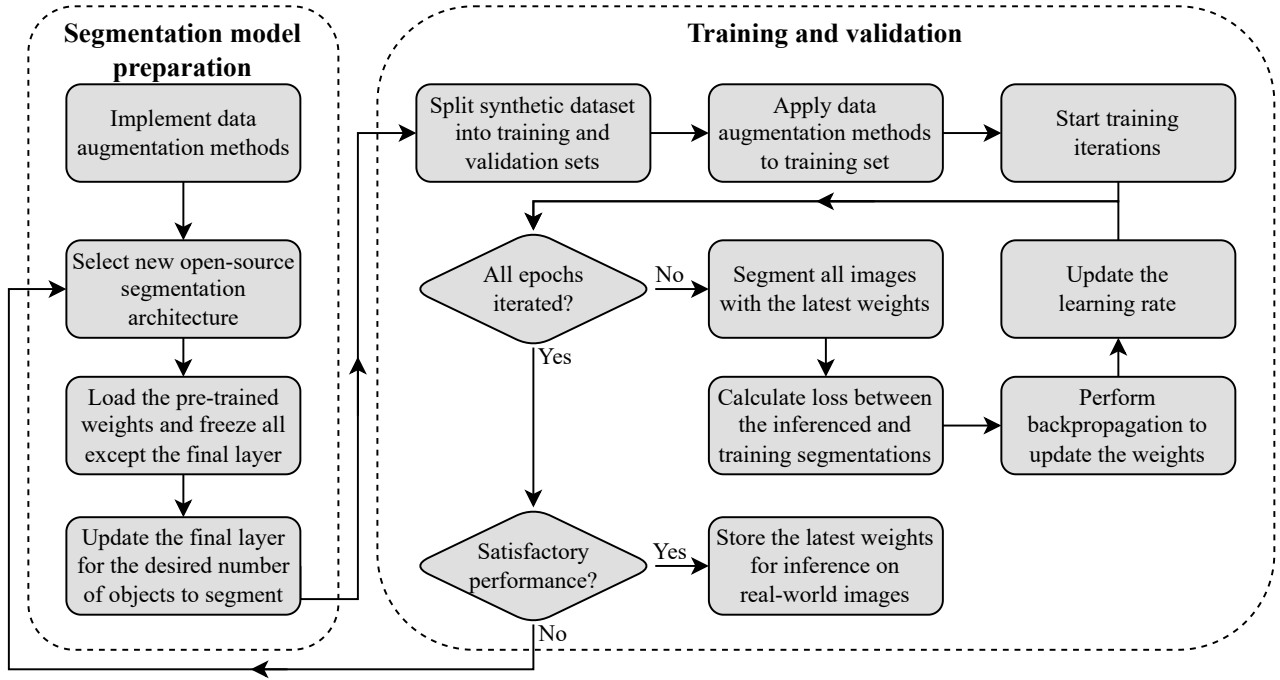


Figure 2.10: Proposed transfer learning method.

Following this, a new open-source segmentation architecture is chosen from those available, its pre-trained weights are loaded, and the backbone weights are frozen. All weights except the final layer are frozen to retain the learned features from pre-training. The final layer of the architecture is then configured for the number of object categories that need to be segmented. The training and validation portion of the method then begins.

The synthetic dataset is first divided into training and validation sets, and the data augmentation methods are applied to the training set. The training then starts by iterating over the chosen number of epochs (the number of times that the training will be repeated using the training dataset).

Each time a new epoch is started, the learning rate is updated to reduce the likelihood of overfitting [108]. Inside each epoch iteration, all training images are segmented using the latest model, and the metrics from Equations 2.20 and 2.21 are used to calculate the loss, i.e. the difference between the inference results of the model and the expected training segmentations.

This loss is then used to perform backpropagation to update the final layer's weights. Following this, the learning rate is updated, and the next epoch begins. Once all epoch iterations have been completed, the performance of the latest model is tested against the validation dataset using Equations 2.20 and 2.21.

If performance is unsatisfactory, a new open-source segmentation architecture is chosen, and the rest of the method starts again. However, if the performance is satisfactory, then the

segmentation model and latest weights are stored for later use on real-world images.

2.4.3 Proposed method

Synthetic dataset generation and transfer learning have been discussed in the previous subsections. This was done using the methodology presented in Figure 2.1. Therefore, in this subsection, the proposed image segmentation method is presented and discussed. This method, proposed to achieve objective 2 in Section 1.8, forms the mapping part of the overall semi-automatic DT generation method. The proposed image segmentation method is presented in Figure 2.11.

The method starts at the top left, where a mobile device is used to capture images in an offline environment. Multiple sets of images can be captured as these images are stored on the mobile device's local storage. Once a server connection is established, the images stored on the mobile device are uploaded to the server. The server is then informed that a new set of samples is ready for processing.

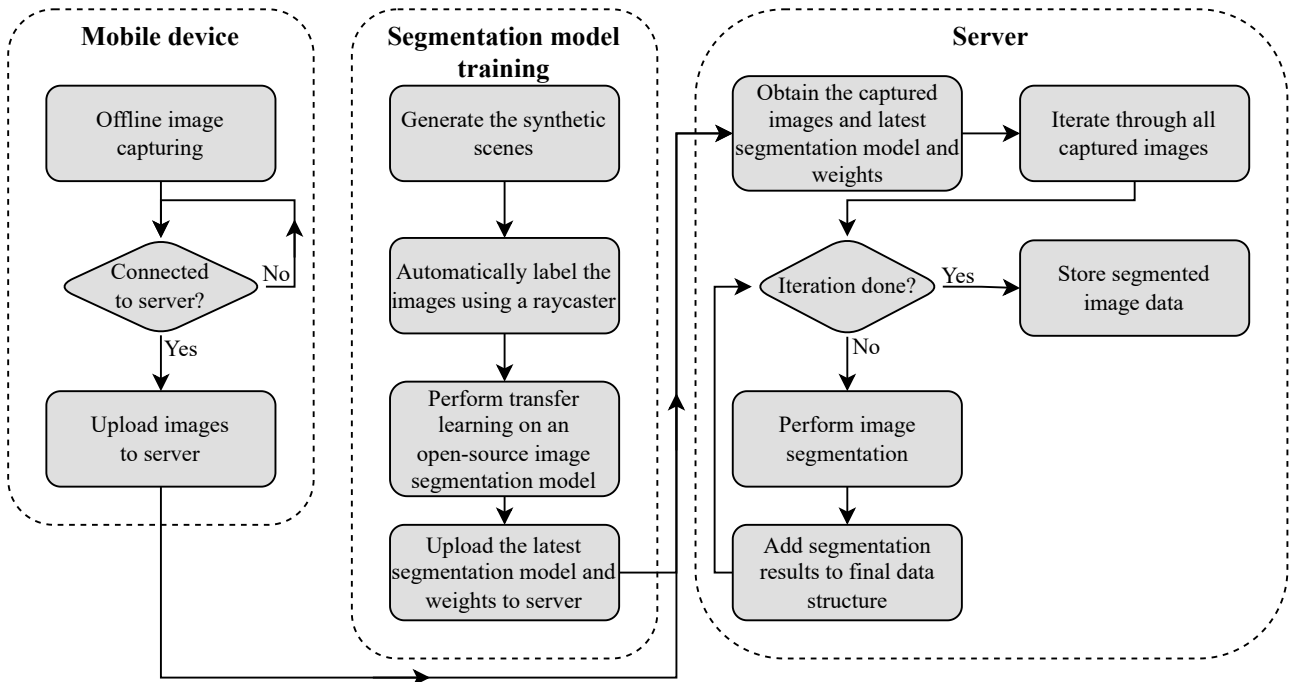


Figure 2.11: Proposed image segmentation method.

In a separate step, in the middle of the figure, the synthetic scenes are generated using the proposed method in Figure 2.9, which includes Algorithm 2, to label the generated images automatically. Transfer learning is then performed using the proposed method in Figure 2.10 to train an image segmentation model. The latest segmentation model and weights are then uploaded to the server.

The first step for the server is to obtain the captured images, the latest segmentation model, and the weights for processing. All photos are then iterated to perform the last steps of the

method. Each image is segmented, and the results are added to the final data structure. Once all images have been segmented, the results are stored for use in the DT generation method.

2.5 Low-cost semi-automatic DT generation

2.5.1 Proposed method

In this chapter, two methods have already been proposed, the INS method in Section 2.3 and the image segmentation method in Section 2.4. These two proposed methods provide the localisation (the estimated path from the IMU samples) and mapping (the segmentations of the captured images) inputs for the proposed low-cost semi-automatic DT generation method shown in Figure 2.12. This overall method, proposed to achieve objective 3 in Section 1.8, was designed, tested, and improved using the methodology presented in Figure 2.1.

The method starts at the top left of the figure, where a mobile device is used to sample an IMU and capture images. The user is required to input certain metadata for each sample set, which will be used later in the proposed method. The input data consists of an initial orientation and a start position. Once a connection between the mobile device and the server is established, the sample sets are uploaded.

The server then converts the IMU samples to estimated paths and segments the images using the methods proposed in Sections 2.3 and 2.4. The methods have been proposed separately to create a generalisable DT generation method, especially during the implementation phase, which will be discussed in Chapter 3. However, as the methods are separate, synchronisation is required to align the outputs of the two methods.

Mobile devices used for IMU sampling and image capturing can also store the time when the this data was sampled. However, IMUs and image capturing devices do not necessarily produce samples in the same intervals. Therefore, each segmented image should be assigned a position based on the two coordinates in the estimated path closest to the image sample time. After the synchronisation step is completed and each segmented image has been assigned an estimated coordinate, the DT generation process begins.

The first step in the DT-specific portion of the proposed method is to check whether the samples were taken for a new DT or to enhance an existing DT. When no existing DT content is present, a new DT is initialised from the metadata captured by the user during sampling as well as other contextual data. When an existing DT is present, its content is loaded in order to perform improvements from the samples.

Once the DT has been initialised or loaded, all uploaded sample sets are iterated in the final loop of the method. Firstly, the initial orientation and start position of the sample are obtained

from the stored user input. Following this, an existing tunnel in the DT is searched for that matches the current sample. The specific matching requirements will vary depending on the different applications of the proposed method. The matching requirements specific to this study are discussed in Chapter 3.

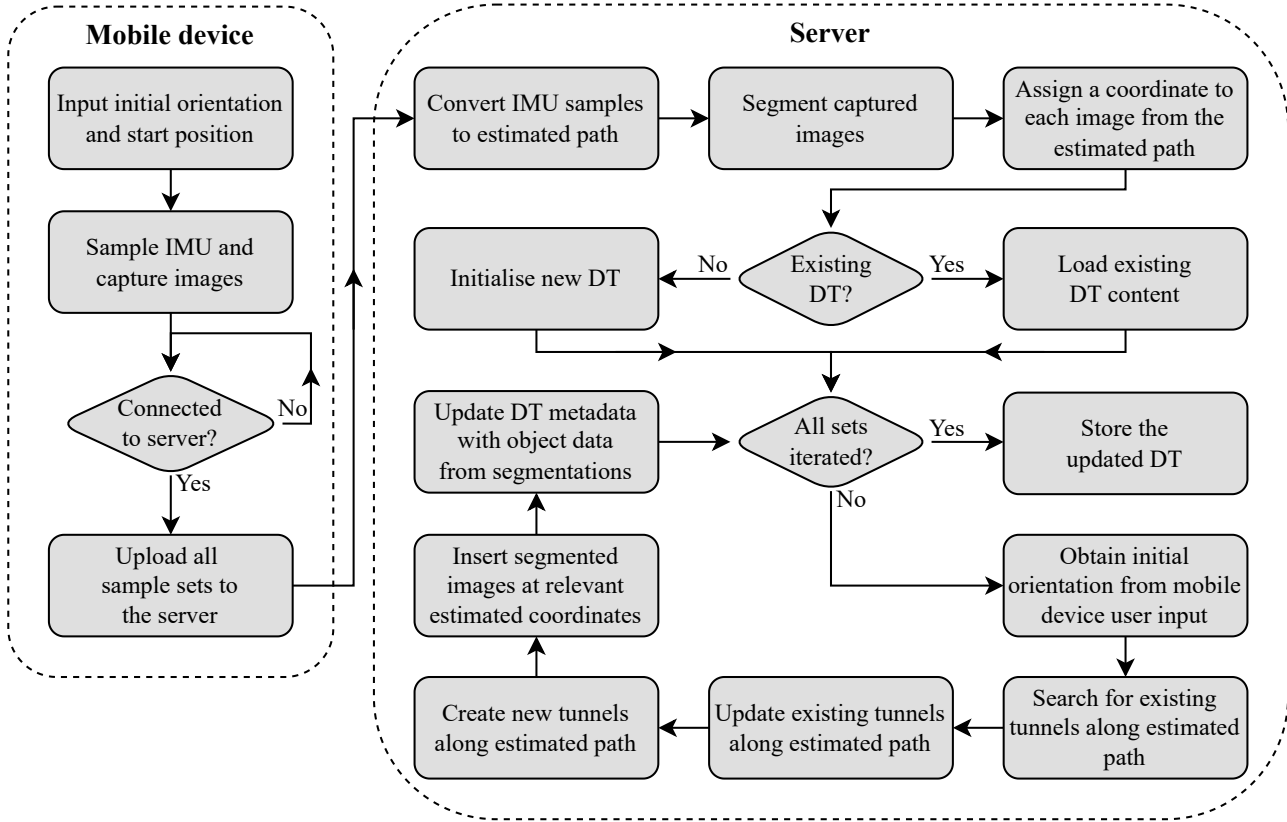


Figure 2.12: Proposed low-cost semi-automatic DT generation method.

If a matching tunnel is found, then the current sample set is used to improve the estimated path of the tunnel. Part of this improvement is adding the segmented images to the tunnel metadata at the estimated coordinates of each image. The segmentation data of the tunnel section, such as the number of certain objects found in this tunnel, is also updated.

A new tunnel is created along the estimated path from the IMU samples when no matching tunnel is identified in the DT. The same process of adding segmented images and metadata is followed again for the new tunnel. After all sample sets have been iterated, the new or updated DT content is stored.

2.6 Summary

The background and literature analysis in Chapter 1 were used to identify the problem statement and the need for the study, as described in Section 1.7. In short, Industry 4.0 initiatives, specifically DTs, can be used to increase the operational efficiency of South African deep-level mines. Manual DT creation methods are time-consuming, and existing

semi-automatic generation methods are expensive.

Therefore, the methodology of the study was presented at the start of this chapter, and following this, various methods were proposed to achieve the three objectives of this study from Section 1.8. The creation of a low-cost method for localisation in high-noise environments was the first objective. The second objective was the creation of a low-cost method for image segmentation in high-noise environments. The final objective was the creation of a low-cost semi-automatic DT generation method.

The methodology presented in Figure 2.1 describes the research approach taken to investigate existing approaches, design newly proposed methods, verify these proposed methods, and implement them. The verification and implementation content are presented in Chapter 3.

In the first of the remaining three sections of this chapter, Section 2.3, the proposed INS method was discussed and presented. IMU data acquisition was discussed, which provided insight into the mobile versus Earth navigation reference frames for the inertial samples. An existing UKF algorithm was adapted for the proposed method and discussed. The UKF will be used to reduce sensor noise and bias to increase the accuracy of the estimated path from the IMU samples.

Thereafter, motion equations were discussed to convert the filtered IMU samples in the mobile reference frame to changes in position over time in the Earth reference frame. The full proposed INS method was presented in Figure 2.5 and discussed. The proposed INS method forms the localisation part of the overall semi-automatic DT generation method.

The proposed image segmentation method was discussed and presented in Section 2.4. The synthetic dataset generation method was proposed and discussed, which will be used to create a deep learning dataset for deep-level mining image segmentation. A 3D framework and raycaster form part of this method. Thereafter, the transfer learning approach was presented and discussed.

This approach enables the use of image segmentation models pretrained on large datasets of common applications. The pretrained backbone weights are frozen to retain learning information, and the last layer is configured for the relevant number of objects of interest. Following this, the final layer is fine-tuned using a smaller synthetic dataset.

The full proposed image segmentation method was presented in Figure 2.11 and discussed. This proposed method forms the mapping part of the overall semi-automatic DT generation method. Finally, in Section 2.5, the proposed localisation and mapping methods were combined and extended to create the proposed low-cost semi-automatic DT generation method, presented in Figure 2.12. The proposed method discusses how the IMU samples and captured images are converted to estimated paths and segmented images to update or create new DT content.

Deep-level underground environments represent a unique challenge as they are offline, unless

extremely expensive fibre networks are installed, and even if they are installed, they do not provide coverage throughout the entire tunnel system. Another characteristic is that these environments have increased EMI due to large exposed ferromagnetic materials underground.

The proposed methods aim to overcome these challenges and provide a stepping stone towards fully automatic low-cost DT generation. The verification and implementation of the proposed methods will be presented and discussed in the following chapter.

3 Results

3.1 Verification and validation strategy

Experiments with known paths and segmentation outputs were utilised in the verification of the methods proposed in the previous chapter. The first two methods, INS and image segmentation, were verified separately to ensure that they address the correct objectives as stated in Section 1.8, namely objectives 1 and 2, independently and show generalisability.

The verification of the proposed low-cost semi-automatic DT generation method was then done to ensure that the integration and extension of the previous two methods address objective 3. These verification steps were outlined in the research methodology in Figure 2.1.

Each module of the INS method, namely, IMU data acquisition, UKF, and motion equations, was individually verified to ensure that it achieves its intended outcomes. The verification of the complete INS method was then quantified by calculating the difference between the estimated path and the expected path from the test set, and the RMSE and PSNR values.

Each module of the image segmentation method, namely, synthetic dataset generation and transfer learning, was verified individually to ensure that it also achieves its intended outcomes. Following this, the verification of the complete image segmentation method was quantified by calculating the error metrics from Equations 2.20 and 2.21 between the ground truth and inference segmentations.

The verification of the final proposed method, the low-cost semi-automatic DT generation, was quantified using both the previous method's quantifications as well as a manual inspection of the final DT content. Lastly, validation was performed using a case study of a real-world scenario to ensure that the proposed methods successfully achieve the objectives outlined in Section 1.8; thereby resolving the need for the study and addressing the problem statement.

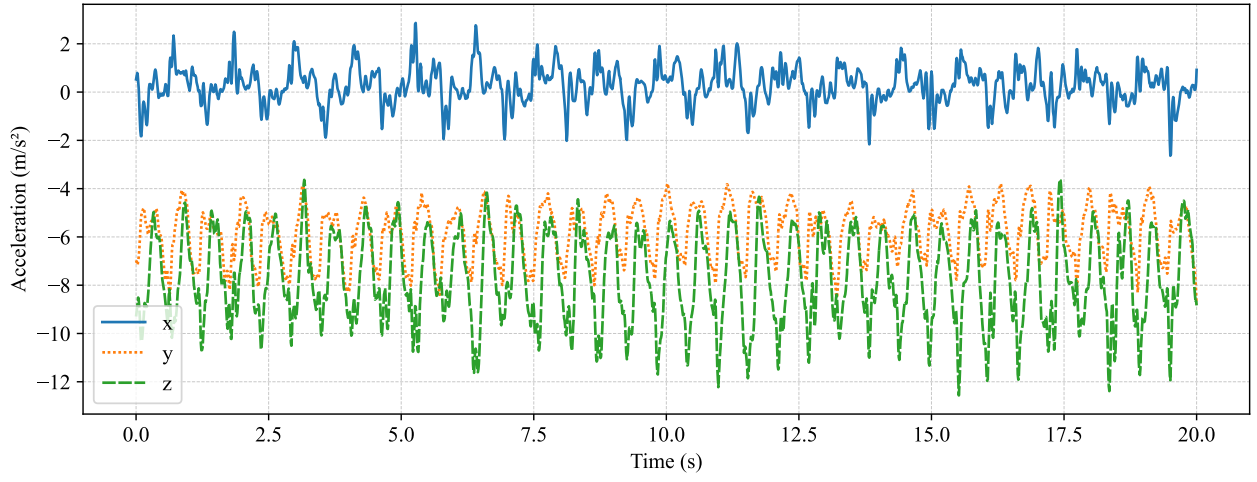
3.2 Inertial navigation

3.2.1 IMU data acquisition

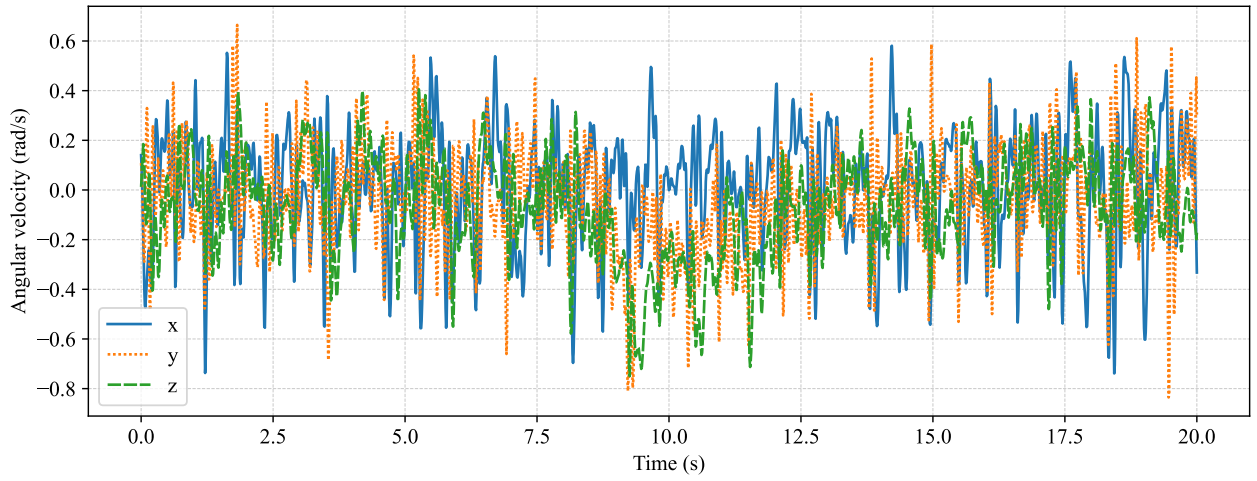
An approach was discussed in Section 2.3.1 to obtain and define the context of raw IMU data from a mobile device. This raw data was used as input during the implementation of the UKF algorithm, which aims to reduce sensor noise and bias.

In this subsection, the results of the implemented IMU sampler for mobile devices are discussed and analysed. A native implementation of the IMU data sampler was done for iOS-based mobile devices, utilising the corresponding API [129]. Figure 3.1a–c shows the outputs of the IMU data

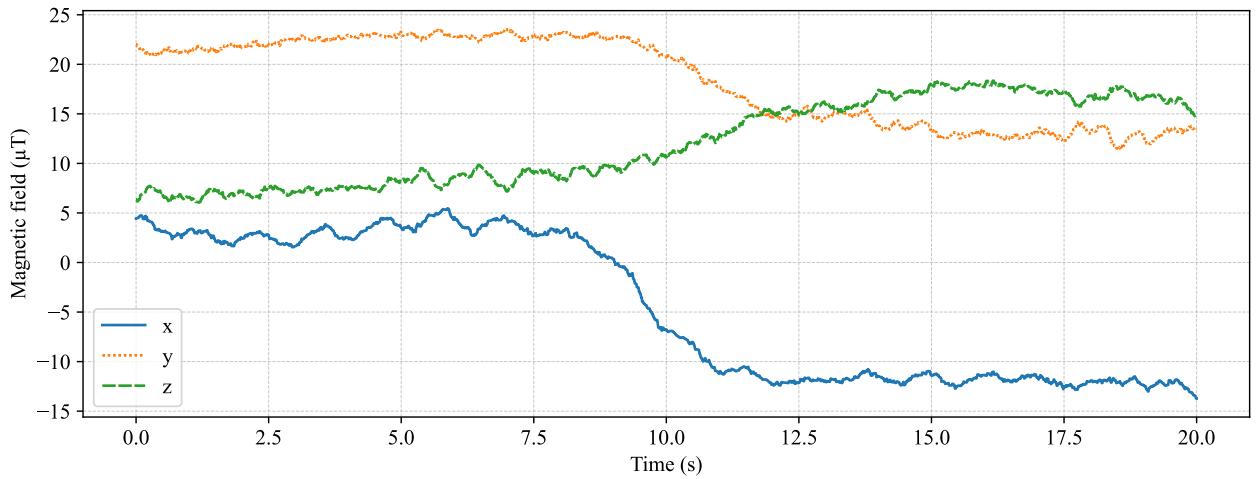
over 20 seconds, using a sampling frequency of 100 Hz, of regular walking while the device was held in front of the test subject in a neutral position.



(a) Accelerometer output.



(b) Gyroscope output.



(c) Magnetometer output.

Figure 3.1: Raw IMU data samples.

This raw output of the accelerometer, magnetometer, and especially the gyroscope shows high-frequency noise, as expected. However, a clear step pattern can still be seen in the accelerometer and magnetometer outputs. At approximately time step 7.5, a gradual 90° right turn was made. This is most apparent from the magnetometer output and the z-axis of the angular velocity. The subject walked as straight a line as possible, before turning right and continuing straight.

3.2.2 Unscented Kalman Filter

Algorithm 1 was discussed in Section 2.3.2 to filter sensor error and drift from the raw IMU samples. These filtered samples were utilised in the motion equations to convert from the mobile reference frame to the Earth reference frame, and the results are discussed in Section 3.2.3.

In this subsection, the results obtained from implementing the UKF algorithm are discussed and analysed. C# was chosen as the programming language to implement the filter, as well as the rest of the proposed INS method, as it is a high-performing, versatile, and scalable language used in server applications [130].

The first step in the algorithm is to estimate the initial state vector $\hat{x}_0$ and the initial covariance matrix $\hat{P}_0$. The state vector has already been chosen to be the true acceleration, magnetic field, and angular velocity. Therefore, L , the size of the state matrix, is 9. No other heading or orientation information is available; therefore, the first ten measurements (equivalent to 100 ms of data) from the raw IMU samples are averaged to initialise the state vector $\hat{x}_0$. The covariance matrix is initialised as a 9×9 diagonal matrix, populated along the diagonal with 1.

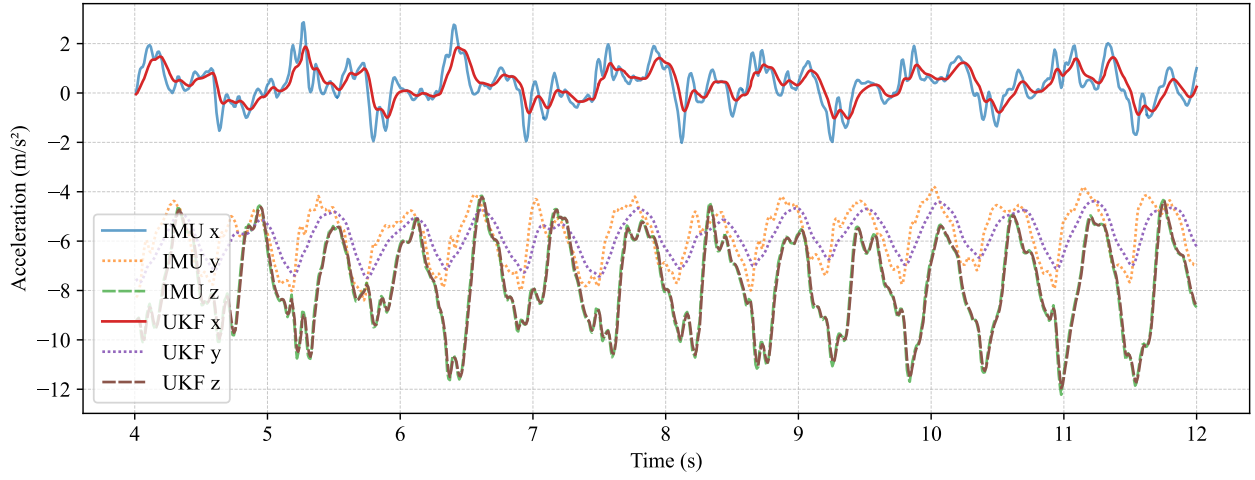
The next matrices to initialise are the covariance of the process, Q , and the covariance of the measurements, R . Q was also initialised as a 9×9 diagonal matrix, populated along the diagonal with 0.1^2 . The covariance of the measurement was initialised as

$$R = \begin{bmatrix} 0.6^2 & 0.83358^2 & 0.09843^2 & \sigma_m^2 & \sigma_m^2 & \sigma_m^2 & (\frac{0.35439\pi}{180})^2 & (\frac{0.44803\pi}{180})^2 & (\frac{0.32338\pi}{180})^2 \end{bmatrix}^T \quad (3.1)$$

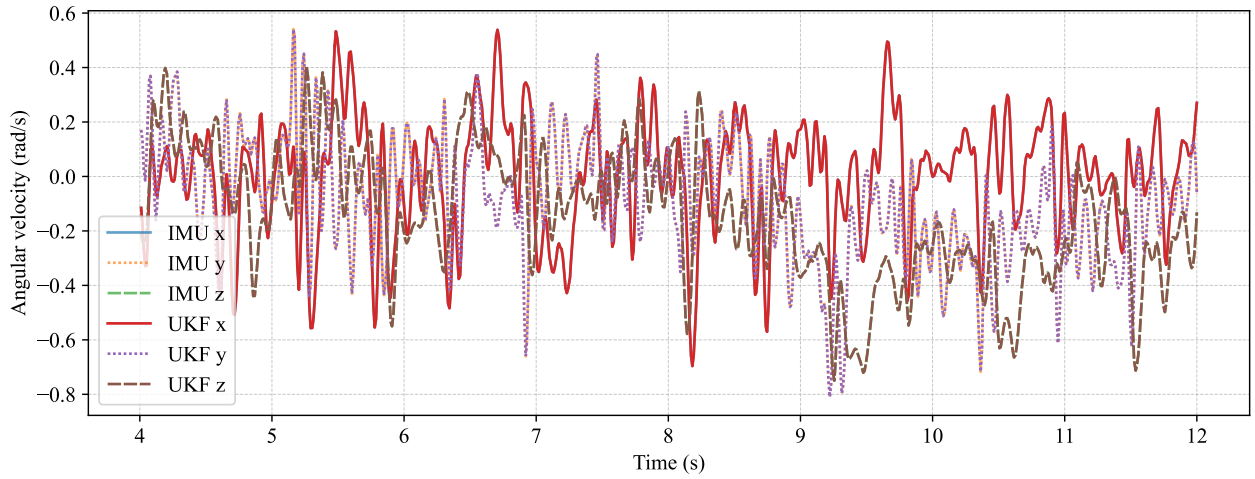
as proposed by Araghi et al. in their mean and standard deviation estimations for accelerometers and gyroscopes [131]. The $\pi/180$ multiplication is required to convert the gyroscope covariance estimations from degrees to radians to match the units in the state matrix. The standard deviation of the magnetometer σ_m^2 was estimated as 5% of the magnitude of the magnetic field in the first ten samples.

The next step was to calculate the weights of the mean and covariance using Equations 2.5 and 2.6. The inputs required for these equations are L , which is 9, and λ , which was calculated using Equation 2.4 as -8.99999 . As all the required initialisations have been calculated, the iterations over each time step and corresponding sample can be done. Figure 3.2a–c shows the output of the UKF algorithm compared to the raw IMU samples in Figures 3.1a–c for the

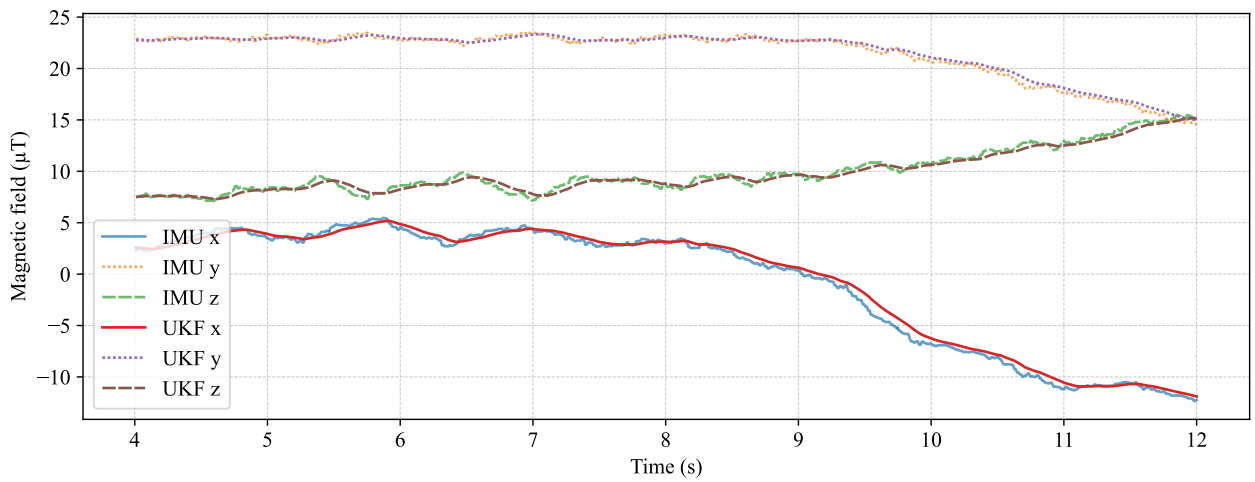
period of 4 to 12 seconds from the full 20-second range.



(a) Accelerometer comparison.



(b) Gyroscope comparison.



(c) Magnetometer comparison.

Figure 3.2: Comparison of raw IMU data versus UKF results.

These results clearly show less rapid variations as well as a slight offset for the filtered versus

the raw IMU data in Figures 3.2a and 3.2c, namely the accelerometer and magnetometer outputs. The gyroscope output in Figure 3.2b does not seem to reflect the same significant filtering.

The quantitative analysis for each sensor in Table 3.1 through the RMSE and PSNR calculations using Equations 2.13 and 2.14 shows that significant filtering was achieved for all IMU data. Lower RMSE values indicate less distortion and greater alignment between the raw and filtered values. Higher PSNR values represent lower error levels and, therefore, better filtering performance.

Table 3.1: Quantitative analysis of the raw versus filtered IMU data.

Sensor		RMSE	PSNR (dB)
Accelerometer	x	0.5789 m/s ²	8.02
	y	0.6157 m/s ²	10.83
	z	1.9653 m/s ²	18.28
	Mean	1.0533 m/s ²	12.38
Gyroscope	x	0.2851 rad/s	33.99
	y	0.2144 rad/s	31.54
	z	0.1911 rad/s	36.81
	Mean	0.2302 rad/s	34.11
Magnetometer	x	0.1662 μ T	15.81
	y	0.1697 μ T	19.36
	z	0.1905 μ T	17.86
	Mean	0.1754 μ T	17.68

The RMSE values cannot be compared between these three sensors, as they have different units. However, from all the RMSE results, one outlier is interesting, namely, the z-axis accelerometer result of 1.9653 m/s². This is more than three times the result of the other two axes. This may be due to the significant influence of gravity on the z-axis. PSNR standardises the units through a logarithmic power calculation; therefore, the best filtering results were obtained for the gyroscope, followed by the magnetometer and finally the accelerometer.

Compared to a recent study by Farahan et al. [84], where PSNR results for the sensors of 45 dB, 25 dB, and 5 dB respectively were obtained, the results from the current study of 12.38 dB, 34.11 dB, and 17.68 dB are within the expected range. Farahan et al. utilised an EKF [132], which is typically a lower performing filter than the UKF utilised in this study, which explains why improved PSNR values for the gyroscope and magnetometer were obtained.

These filtered results more accurately represent the actual linear acceleration, angular velocity, and magnetic field encountered during the sampling process. Therefore, using these more accurate IMU samples will translate into more accurate path estimation in the INS. The outputs of the UKF will be used to convert the IMU samples from the mobile device's reference frame

to the Earth's reference frame for path estimation. The combination of the filtering with the motion equations aims to achieve objective 1b from Section 1.8.

3.2.3 Motion equations

An approach was discussed in Section 2.3.3 to convert filtered IMU data from the mobile device's reference frame to the Earth's reference frame. These motion equations constitute the last steps in obtaining the estimated path from the raw IMU samples. The conversion was done by calculating the direction cosine matrix, as in Equation 2.18. The yaw, pitch, and roll values are required for this calculation and were obtained using Equations 2.15, 2.16, and 2.17.

After the direction cosine matrix was obtained, each filtered output was matrix multiplied, as in Equation 2.19. This converts the sensor values from the IMU reference frame to the Earth reference frame. These values were then used to calculate the changes in position over time. The result of this process is shown in Figure 3.3, where the change in position over time is given for the 20-second test data of Figure 3.1, following the UKF filtering process.

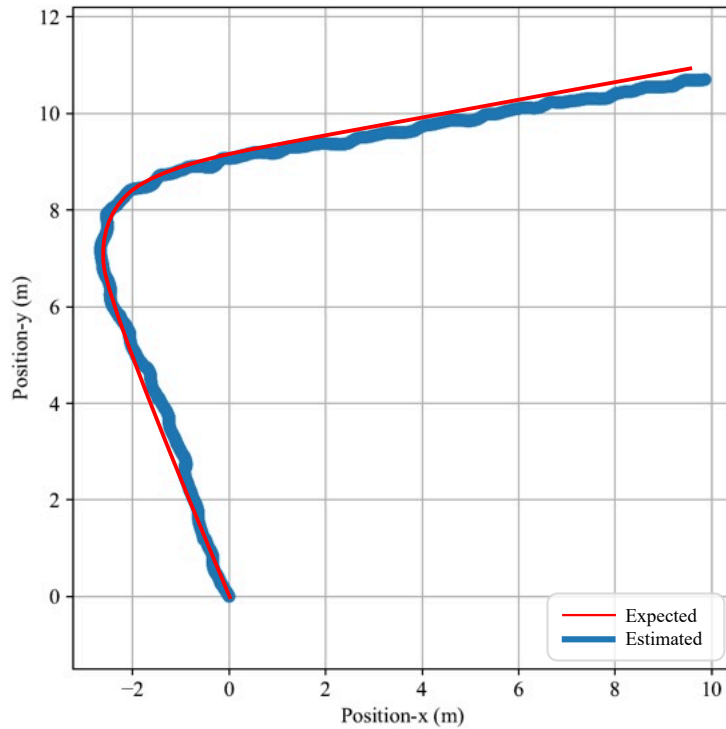


Figure 3.3: Estimated path from motion equations.

The line starts at position (0,0), where the test subject walked in a straight line towards position $(-2.75, 7)$, where the gradual 90° turn to the right was made, and the test subject continued to walk in a straight line again to position $(10, 10.75)$. Each step caused a ripple on the estimated path line, as expected due to the sudden change in sensor values. However, the two expected straight portions, as well as the expected turn to the right, can be seen.

Table 3.2 provides further results from the motion equations, where the average walking speed is given as well as the total distance. The speed of 1.37 m/s is slightly less than the average walking speed of 1.434 m/s determined in the study by Bohannon et al. [133]. The estimated total distance of 27.1 m was checked through manual measurement during the test procedure, and an error of 0.3 m was found.

Table 3.2: Motion equations measurements.

Measurement	Estimation
Average walking speed	1.37 m/s
Total distance	27.1 m

3.2.4 Path estimation

A method was proposed in Section 2.3.4 to perform localisation using an INS approach. This was used for path estimation in the overall DT generation method, the results of which are discussed in Section 3.4. In this subsection, the results obtained from the implemented proposed INS method shown in Figure 2.5 are discussed and analysed.

The complete IMU test set, consisting of 60 seconds of IMU data, of which a partial subset was used throughout this section, was used as input for verification of the proposed INS method. Table 3.3 shows the UKF performance for the verification test data through the RMSE and PSNR results.

Table 3.3: Quantitative analysis of the UKF for the full verification test.

Sensor		RMSE	PSNR (dB)
Accelerometer	x	0.5490 m/s ²	8.63
	y	0.5944 m/s ²	11.15
	z	1.9147 m/s ²	18.75
	Mean	1.0193 m/s ²	12.84
Gyroscope	x	0.2566 rad/s	34.86
	y	0.2075 rad/s	32.46
	z	0.1895 rad/s	38.10
	Mean	0.2179 rad/s	35.14
Magnetometer	x	0.1619 μ T	18.14
	y	0.1790 μ T	19.23
	z	0.1556 μ T	20.20
	Mean	0.1655 μ T	19.19

The UKF implementation performed the best in terms of mean PSNR for the gyroscope, with 35.14 dB, followed by the magnetometer, with 19.19 dB, and lastly, the accelerometer, with

12.84 dB. All PSNR values showed that the UKF implementation performed better for the entire test set compared to the performance of the 20-second subset in Table 3.1, except for the y-axis of the magnetometer. The complete estimated path for the test data is shown in Figure 3.4. The path starts at $(0, 0)$ and ends at $(23, 33.5)$.

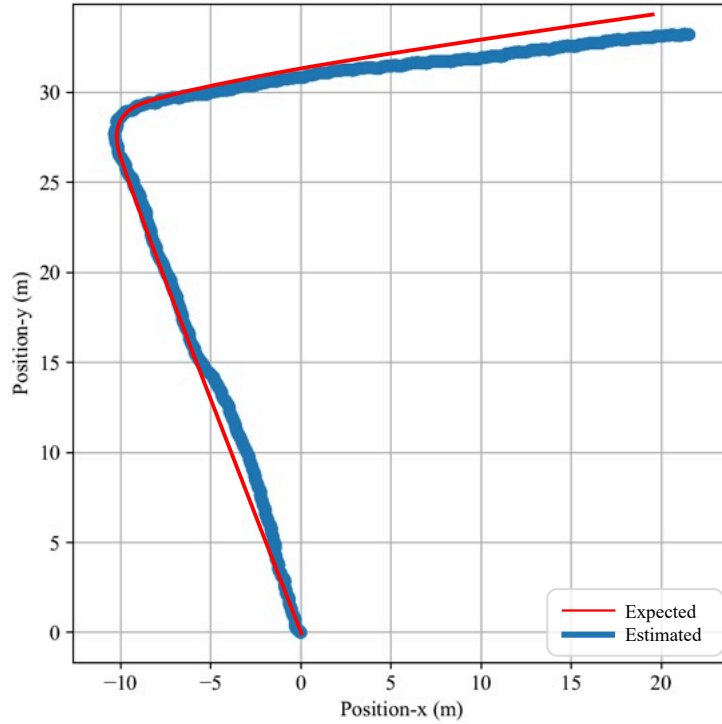


Figure 3.4: Verification of the proposed INS method.

The estimated path aligns with the expected results, as the full test set consisted of a test subject walking in a straight line, turning gradually to the right, and walking in another straight line. Some deviation from the first straight line section, at approximately $(-5, 14.5)$, may be due to external EMI interference or an uneven walking pattern at that moment. Increased accuracy during path estimation by the proposed INS system will directly translate to more spatially correct DT generation results.

Table 3.4 shows an average walking speed of 1.34 m/s, which is similar to the expected approximate speed of 1.434 m/s [133]. The estimated total distance of 80.6 m was checked through manual measurement during the test procedure, and an error of 2.4 m was found, which is approximately a 3.3% error.

Table 3.4: Verification measurements.

Measurement	Estimation
Average walking speed	1.34 m/s
Total distance	80.6 m

In this section, it has been verified that the methods proposed in Section 2.3 address the correct

objectives of Section 1.8, that is, objectives 1a and 1b for low-cost offline path estimation. In the following section, the results from the implementation of the proposed image segmentation method are presented and discussed.

3.3 Image segmentation

3.3.1 Synthetic dataset generation

Synthetic scenes creation

A method was proposed in Section 2.4.1 to generate synthetic datasets for image segmentation in high-noise environments using a 3D modelling framework and raycaster. A dataset generated using this proposed method was used for training in the implementation of the transfer learning method, and the results are discussed in Section 3.3.2.

In this subsection, the results obtained from implementing the synthetic dataset generation method are discussed and analysed. The first step in designing and creating the synthetic scenes of the dataset generator was to analyse real-world scenes of the application. Figure 3.5 contains images of an area inside a deep-level mine.

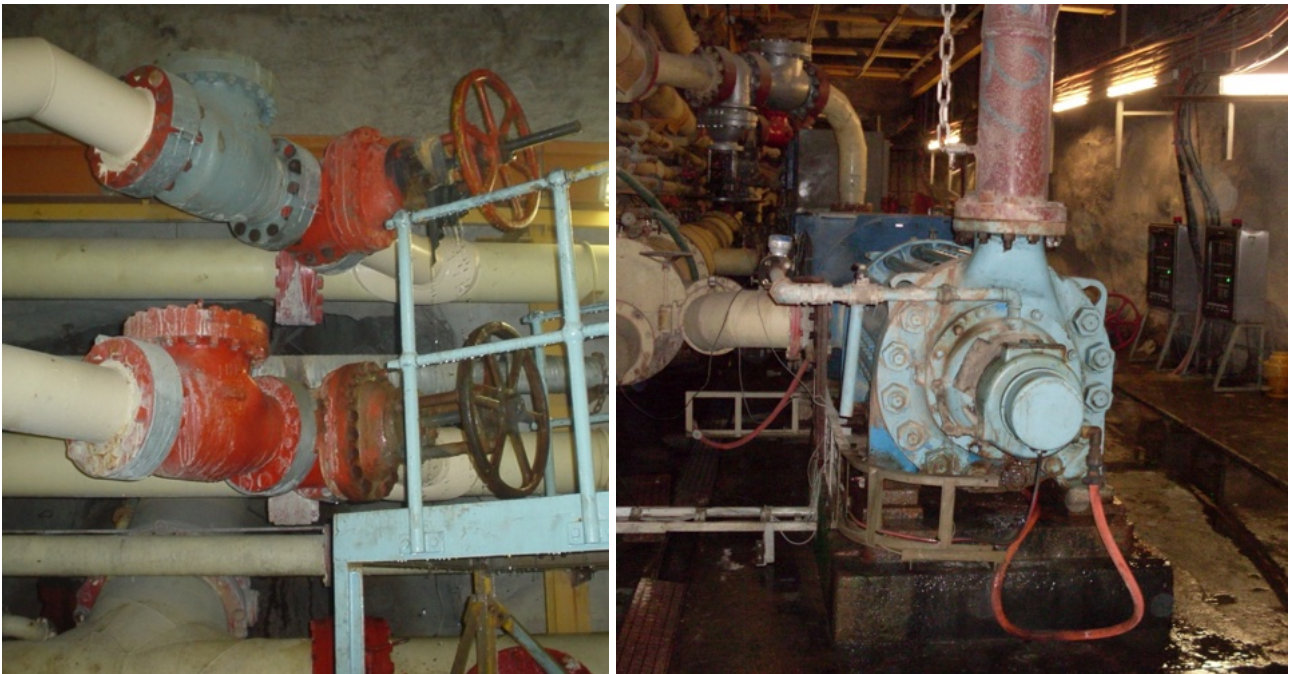


Figure 3.5: Real-world images inside a deep-level mine.

The analysis of these and other images from inside deep-level mines showed that the objects typically found are valves, pipes, ladders, induction motors, heat exchangers, and centrifugal pumps. Therefore, these six classes were identified as the objects of interest in the dataset to be generated. Some images also showed inconsistent lighting conditions, as not all areas are well-lit.

The next step to create the synthetic scenes was to select a 3D modelling framework based on the questions discussed in Table 2.1. Three.js (referred to hereafter as ‘the framework’) was chosen based on the analysis done in Table 3.5 below within the context of this study.

Popular alternatives to Three.js include Blender and Unity. These and other 3D modelling frameworks offer advanced simulation and physics features. However, this is not required in the context of this study and, therefore, would unnecessarily increase the computational requirements. Some frameworks, such as Unity, also require the purchase of costly licences. Following the analysis, the next step in creating synthetic scenes was to obtain open-source 3D models of the objects identified in the real-world scenes.

Table 3.5: 3D modelling framework selection process.

Category	Discussion
Advanced physics and simulation	These are supported by the framework but are not relevant to the current application and, therefore, are not considered.
Usability and flexibility	Comprehensive documentation and an active community ensure high usability. The framework is an abstraction of the WebGL API (used to render interactive 3D graphics in most web browsers), which decreases the implementation time of the synthetic scene generation while still ensuring flexible customisation.
Raycasting	The framework includes a camera and raycasting class to perform the required perspective changes and raycast analysis efficiently.
Scalability and hardware requirements	As the framework is an abstraction of the WebGL API, the scalability and hardware requirements are mostly affected by implementation efficacy. The choice in this category is not influenced by the costly computations of advanced physics and simulations, as they are not required.
3D model imports and exports	The framework supports many different import formats.
Licensing and cost	The open-source framework aligns with the low-cost requirement of this study.

Figure 3.6 shows three open-source models obtained, namely, a valve, a pipe, and a centrifugal pump. These objects will be placed in various pseudo-random positions in the synthetic scenes to be generated. This refers to the positions in which these objects would typically be found in a real-world image. These pseudo-random positions also refer to the way that a perspective should not intersect a solid object, as this would not accurately represent the real-world perspectives.

As an example, pipes are typically located towards the top portion of a horizontal perspective in a walkway of a deep-level mine. These and other informed assumptions were required to increase the likelihood that the final synthetic dataset produces performant segmentation results by

avoiding unnecessarily high levels of variance [105].

Datasets used for transfer learning should be created specifically for the application required as this increases the likelihood of performant segmentation [108]. Although the dataset will be generated for a specific application, the proposed method is generalisable for many applications due to the first step of the method, where the relevant real-world scenes are analysed.

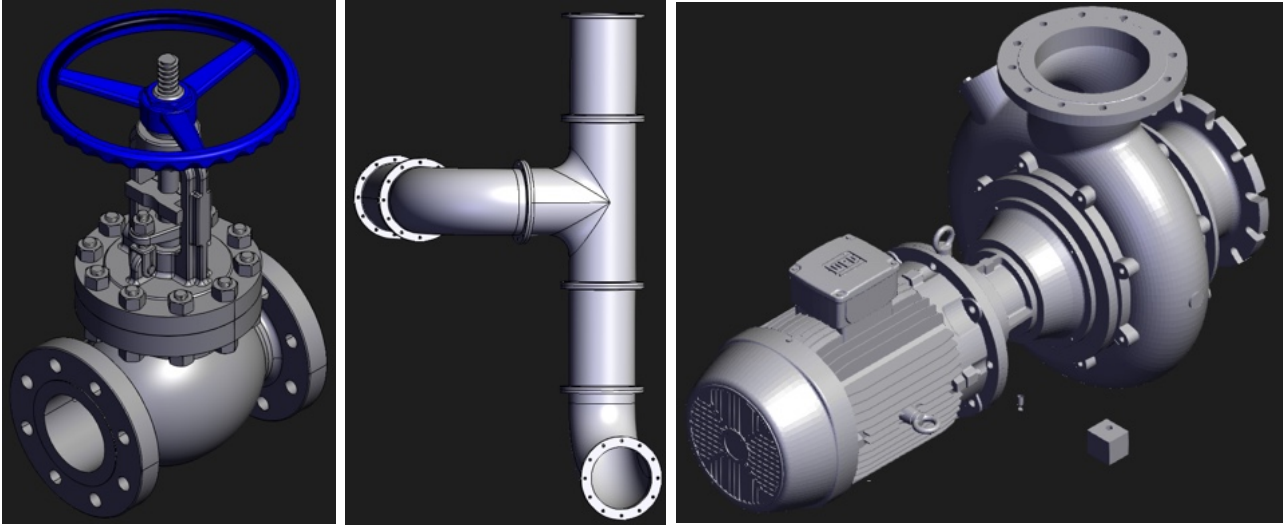


Figure 3.6: Open-source 3D models for objects of interest.

As the 3D modelling framework has been selected and relevant open-source 3D models have been obtained, the scene implementation process can be done using the method proposed in Figure 2.6. This involves iterating over all the objects and phenomena identified in the scene analysis, implementing them in the synthetic scene, analysing the performance, and simplifying the objects based on the achieved performance. Figure 3.7 shows two different perspectives of a synthetic scene that was created using the proposed method.

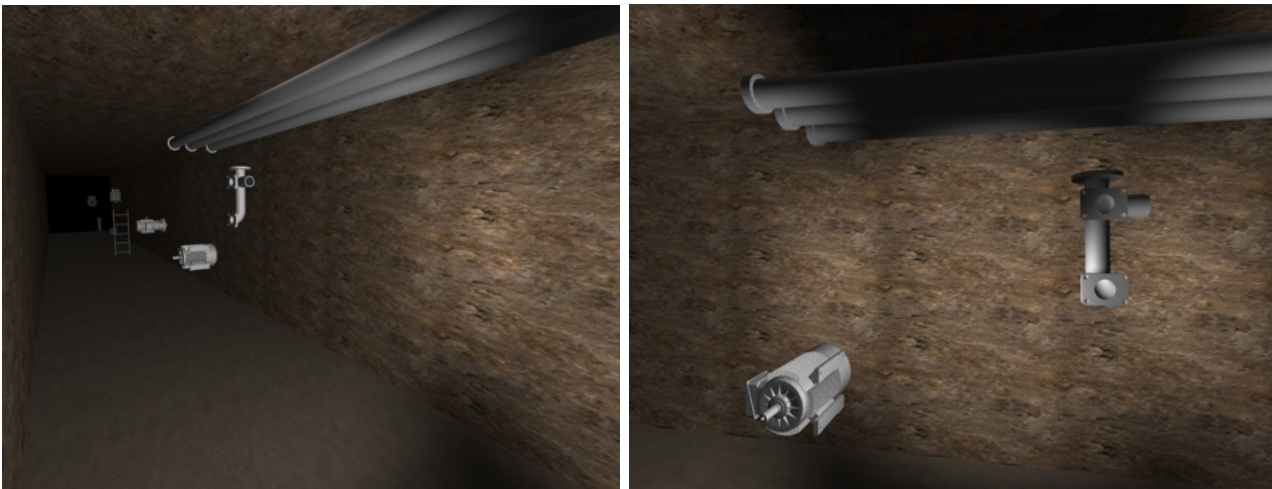


Figure 3.7: Perspectives of one of the generated synthetic scenes.

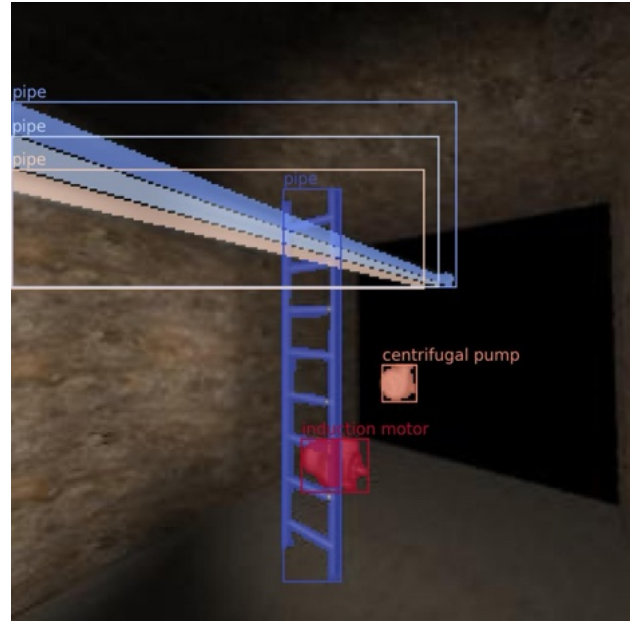
The majority of the pixels within the images are the walls, floor and ceiling of the tunnel. Various pipes, motors, and valves that have been identified as objects of interest can also be seen.

Automated labelling using raycasting

The automated labelling process was implemented using the 3D modelling framework's camera and raycasting functionality. Various perspectives of all the synthetic scenes generated were captured using the camera's perspective, as shown in Figure 2.7. Each perspective was then divided into a 256×256 matrix as shown in Figure 2.8 and raycast to produce a matrix of raycast results for each perspective. Following this, each raycast matrix was converted to polygon labels using Algorithm 2. The labelling results of two different generated scenes are shown in Figure 3.8.



(a) An image from the generated dataset.



(b) Automated labelling result.



(c) An image from the generated dataset.



(d) Automated labelling result.

Figure 3.8: Automated labelling results for images in the generated dataset.

Figure 3.8a contains three pipes, a ladder, an induction motor, and a centrifugal pump. Figure 3.8b shows that the raycasting process successfully identified and segmented the individual objects of interest within the scene. Figure 3.8d shows similar successful automatic identification and segmentation from the input image in Figure 3.8c.

These and many other perspectives and their corresponding polygon raycast results were combined to create the final synthetic dataset, as shown in the proposed method in Figure 2.9, which will be used for segmentation training in the following subsection. The generated dataset contains a total of 329 automatically labelled images. Figures 3.9a–c show the analysis of the final created dataset in terms of the total segmentation count per class, the total segmentation area per class, and lastly the average segmentation area per class.

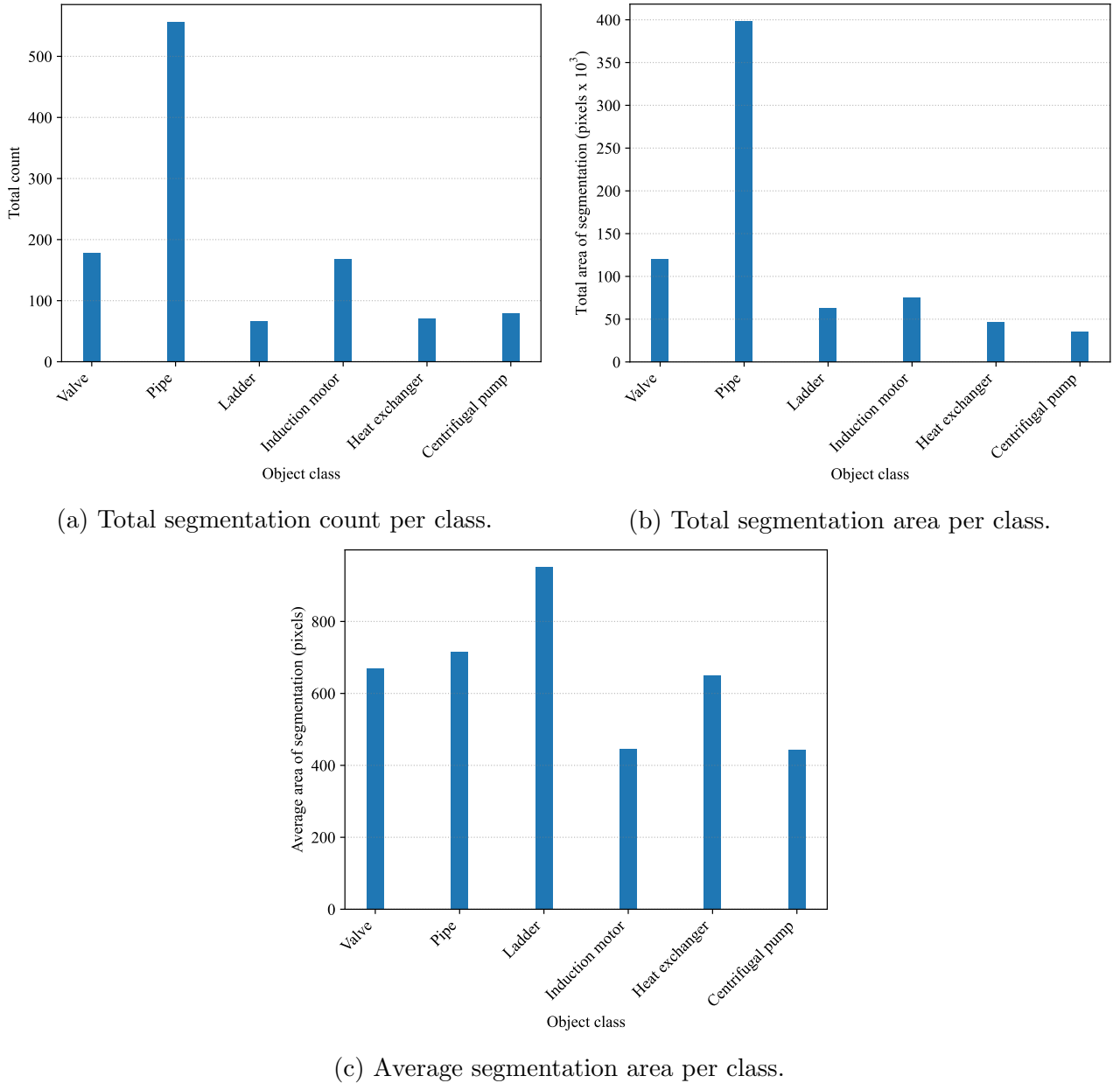


Figure 3.9: Generated dataset analysis.

The total segmentation count in Figure 3.9a indicates that the dataset contains significantly

more pipes, by a margin of 444, compared to the average of the other classes. The total segmentation area in Figure 3.9b also reflects this, where the pipe class has the largest total area by a margin of 67226.3 pixels compared to the average of the other classes.

An unbalanced dataset may cause decreased segmentation performance for the classes that are less prevalent in the dataset. This means additional false positive matches or even false negatives on these affected classes. However, Figure 3.9c shows that the average segmentation area per class is more balanced. The unbalanced dataset was caused by the analysis of the real-world scenes, where it was found that pipes are the most commonly encountered objects.

3.3.2 Transfer learning

Segmentation model preparation

A method was proposed in Section 2.4.2 to use a generated synthetic dataset and existing image segmentation architectures in a transfer learning approach to train a segmentation model. A model trained using this proposed method was utilised for image segmentation, and the results are discussed in Section 3.3.3.

In this subsection, the results obtained from implementing the proposed transfer learning method are discussed and analysed. There exist many deep-learning frameworks; however, PyTorch was chosen for its customisability, the number of available segmentation architectures and pre-trained weights, the fact that the framework is open-source, and lastly, its performance on relatively low-powered machines [134]. Consequently, Python was utilised as the programming language, as PyTorch is a Python framework.

As the framework and language have now been selected, the proposed method shown in Figure 2.10 can be implemented. The first step was to implement the data augmentation methods. The chosen data augmentation techniques were auto-contrast, random rotation, and grayscale. Auto-contrast maximises the difference in magnitude between all pixel values in an image. Therefore, the darkest pixel in an image would be converted to black (representing the lowest possible value), and the lightest pixel to white (representing the highest possible value).

Random rotation, as its name suggests, applies a random rotation to both the image content and the labelled segmentations. Grayscale removes colour from the images. These data augmentations will be applied to some of the training samples from the generated dataset. This will increase the generalisability of the segmentation model after transfer learning.

The next step in the proposed method was to select a segmentation architecture. The architecture chosen for instance segmentation was the improved Mask-RCNN with a ResNet-50-FPN backbone [135]. It was selected due to its performance in transfer learning approaches. The model was initialised with available pre-trained weights that were calculated

using the freely available COCO dataset [71]. All of the layers in the architecture were frozen except for the last to enable transfer learning.

The final step in preparing the segmentation model was to update the final layer to accommodate the desired number of object classes to segment. The dataset contains six objects of interest; therefore, the final layer was configured to have six outputs, each representing a distinct class. Each of these outputs consists of an array representing different inference instances of the class with a confidence score assigned to each.

Training and evaluation

Following the preparation of the segmentation model, the second part of the proposed method, namely, training and evaluation, was started. This was done by splitting the generated synthetic dataset into training and evaluation sets. The split was random and consisted of a ratio of training to evaluation image and segmentation sets of 300:29. The implemented data augmentation methods were then applied to the training set, and the training iterations were started.

A stochastic gradient descent (SGD) optimiser was combined with a decreasing learning rate per training epoch (one round of training on all the training images), as this has shown promising results in transfer learning approaches [136]. The learning rate was initialised to 0.004, and the weight decay was set to 0.0002. The training was performed using a computer equipped with an NVIDIA GeForce GTX 1650 GPU and 4 GB of dedicated memory.

The training loop was then performed, where the losses between the inference and training segmentations were calculated using Equations 2.20 and 2.21. Table 3.6 shows the improvement of the performance metrics obtained over each of the five training epochs. Epochs higher than five did not show a significant increase in performance. Therefore, the number of epochs was kept at five to avoid overfitting during transfer learning. The final training time was 16 minutes.

Table 3.6: Improvement of performance metrics over each epoch.

Metric	Epoch 1	Epoch 2	Epoch 3	Epoch 4	Epoch 5
IoU	18.98	31.26	37.04	38.85	39.35
PQ	21.23	39.24	44.33	47.14	49.16

The performance metrics both increased significantly over the first few epochs. This increase slowed after the third epoch. This is to be expected as the learning rate is at its highest in the first epoch, and the training sets have not yet been seen. Both metrics reached their maximum in the final epoch.

Table 3.7 shows a comparison between the final IoU results obtained after training using the method proposed in this study and existing methods discussed in a recent survey

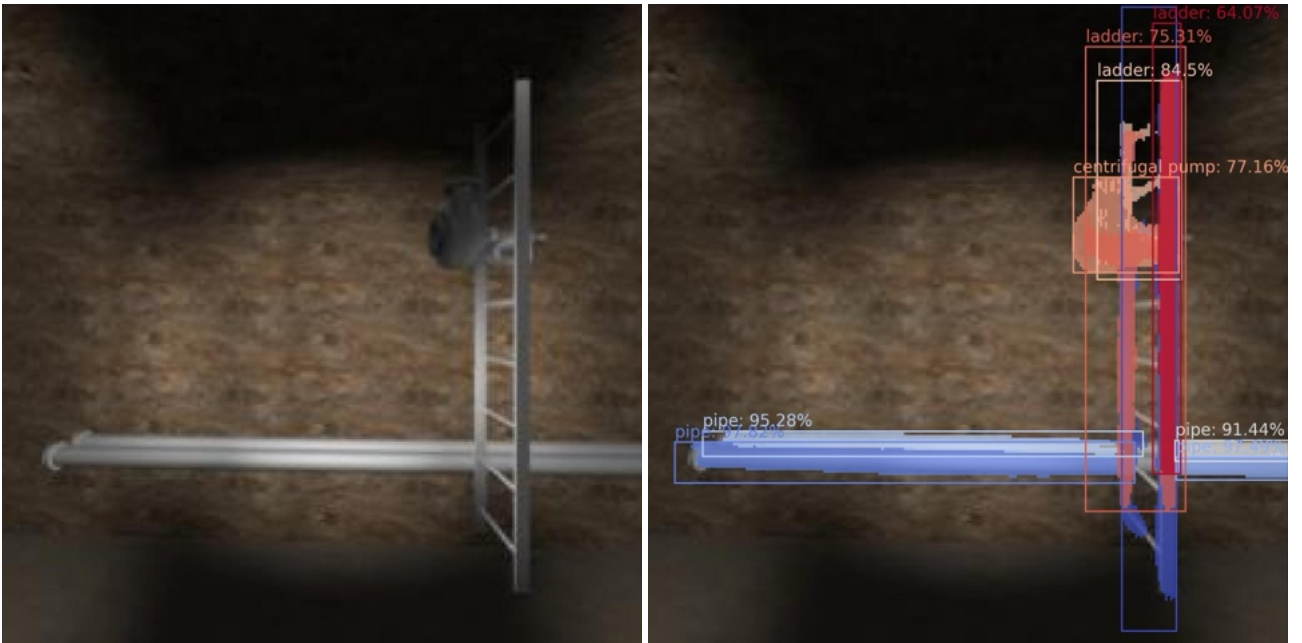
paper [108]. These methods from the survey paper were also trained using the COCO dataset, with the difference being the transfer learning applied in this study.

Table 3.7: Comparison of IoU results with methods from recent studies.

Method	Backbone	IoU
DA-Net	ResNet-50	37.9
The proposed method	ResNet-50	39.35
EMA-Net	ResNet-50	39.9
AC-Net	ResNet-101	40.1

In comparing the IoU results from the proposed method with those from the methods of other studies, it would not be accurate to say that the proposed model outperforms any other model, as transfer learning was applied to the proposed model on a custom synthetic dataset. This comparison aims to show that the performance obtained from implementing the proposed method produced results within a reasonable range.

The last step of the proposed transfer learning method was the evaluation, where the segmentation performance on the evaluation set was analysed. Figure 3.10 shows one of these evaluation images as well as the corresponding inference result from the trained segmentation model.



(a) Input synthetic image for evaluation.

(b) Segmentation inference result.

Figure 3.10: Transfer learning evaluation result.

All of the object classes were identified correctly with confidence scores ranging from 64.07% up to 97.49%. The segmentation masks were also highly accurate, excluding a small portion of the centrifugal pump behind the ladder, as it was incorrectly included as part of the ladder. The boundary detection for each object instance was also quite accurate, excluding some horizontal

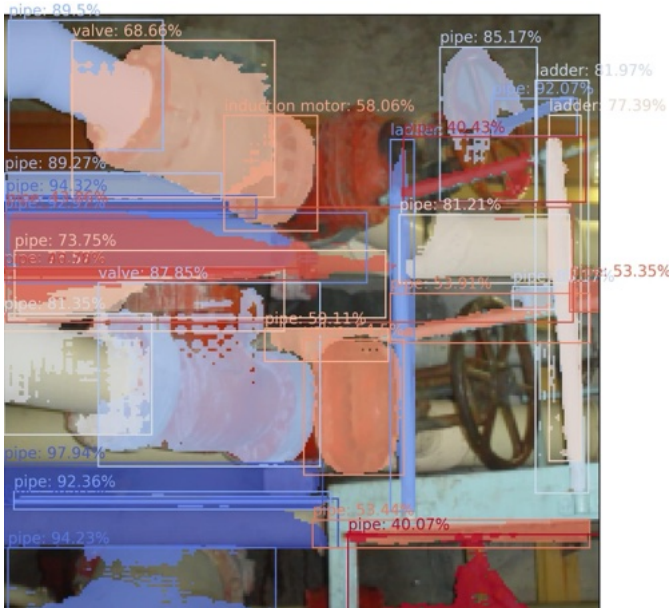
steps of the ladder, which may be due to the size of the ladder steps versus how fine the raycasting matrix was created.

3.3.3 Segmentation

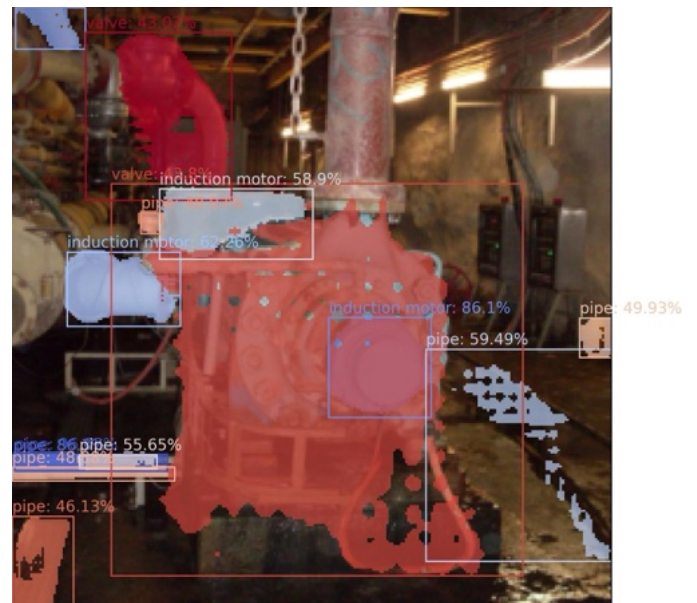
A method was proposed in Section 2.4.3 to perform mapping using image segmentation. This was used to extract information from captured images in the overall DT generation method, for which the results are discussed in Section 3.4. In this subsection, the results obtained from implementing the proposed image segmentation method shown in Figure 2.11 are discussed and analysed.

The images from Figure 3.5 were used as inputs for verification of the proposed image segmentation method. The first results of verification are shown in Figure 3.11, which contains the segmentation inference results. All confidence scores lower than 40% were ignored to reduce the number of false positive matches. Most of the pipes in Figure 3.11a were accurately labelled and segmented with confidence scores varying from 40.07% up to 97.94%.

Three of the four valves in this image were labelled and segmented mostly correctly, with some parts of the valve to the top right of the image being ignored or mistaken as pipes. The shape, incorrectly segmented as a pipe on the handle of this valve, does have a similar cylindrical form to that of a pipe in terms of the handle's shape. Figure 3.11b shows a non-drive end bearing housing in the centre of the image that was segmented but incorrectly labelled with a confidence score of 86.1% as an induction motor, as their shapes are similar.



(a) Segmentation inference result 1.



(b) Segmentation inference result 2.

Figure 3.11: Verification of the proposed image segmentation method.

The profile of the large pump in the centre of the image was segmented well but mislabelled as a

valve due to its similar shape. A portion of the valve to the top-left of the image was segmented and labelled correctly with a confidence score of 43.02%. Various other objects, such as the railway to the bottom right and the grate to the bottom left of the image, were mislabelled as they resemble pipes in the dataset.

Table 3.8 shows the corresponding verification results in terms of the performance metrics IoU and PQ. These results were obtained by manually labelling ground-truth segmentation masks for the two images and calculating the performance metrics achieved. The performance achieved after the final epoch on the evaluation set, 39.35 and 49.16, was higher than the 34.15 and 32.41 achieved for the verification images in Figure 3.11, as expected.

Table 3.8: Segmentation performance comparison with state-of-the-art methods.

Segmentation method	Test dataset	IoU	PQ
Proposed method	Evaluation synthetic training images	39.35	49.16
Proposed method	Real-world verification images	34.15	32.41
State-of-the art methods [71,108]	Real-world COCO dataset	39.3	40.31

The use of transfer learning to retain learned information from the real-world dataset training and fine-tune it to the specific application was successful, as only the synthetic dataset contained segmentations relevant to the application. The performance during verification of the proposed method was slightly lower than the evaluation set. However, verification showed usable performance on real-world images compared to an average IoU of 39.3 and PQ of 40.31 from methods considered in a recent survey study trained using the COCO dataset [71,108].

In this section, it has been verified that the methods proposed in Section 2.4 address the correct objectives of Section 1.8, that is, objectives 2a, 2b, and 2c for low-cost image segmentation. In the following section, the results from the implementation of the low-cost semi-automatic DT generation will be presented and discussed.

3.4 Low-cost semi-automatic DT generation

3.4.1 Preamble

A method was proposed in Section 2.5.1 for low-cost semi-automatic DT generation. It consists of two main parts, namely, localisation through the proposed INS method, and mapping through the proposed image segmentation method. These two methods were integrated and expanded to create the overall proposed method in Figure 2.12.

These two methods were independently verified in Sections 3.2 and 3.3 to show generalisability for different applications. The implementation details and results from the verification and validation of the overall method are presented, discussed, and analysed in this section.

3.4.2 Implementation

As discussed in Section 3.2.1, a native IMU sampler was implemented for iOS-based mobile devices. The specific device used to collect the samples was the iPhone 7 Plus (iOS version 15.8.4), as shown in Figure 3.12. The implementation includes input capturing from a user to store certain metadata, such as the title of the captured set. In terms of image capturing, an Insta360 ONE X2⁴ camera was utilised. This camera supports connection via WiFi; therefore, it was automatically controlled through the mobile application through the wireless connection.

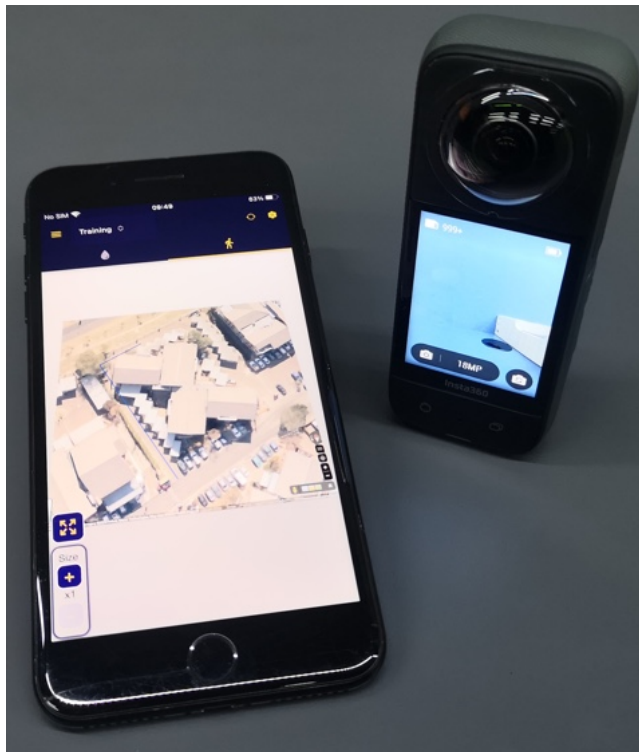


Figure 3.12: Mobile IMU sampler and image capturing device.

The mobile application was integrated with the DT framework created in the study by Kilian [137]. This enables the user to select a starting position, input the desired name for the capture set, and, when ready, start the IMU and camera sampling process. This consisted of sampling the mobile device’s IMU at 100 Hz, and capturing an image in five-second intervals to allow for the image to be processed and cached in the camera’s storage.

Once the sampling process was completed, all images were transferred to the mobile device from the camera via WiFi. These images, as well as all collected IMU data, were saved to the long-term storage of the mobile device for upload once a server connection was established.

The server implementation for the proposed INS method was done using a .NET API hosted using Google Cloud Platform’s (GCP) Cloud Run⁵ functionality. The specifications for the

⁴”Insta360 ONE X2”, Arashi Vision Inc., Insta360, <https://www.insta360.com/product/insta360-onex2> (accessed 4 July 2025).

⁵”Cloud Run”, Alphabet Inc., Google, <https://cloud.google.com/run> (accessed 4 July 2025).

Cloud Run API were set to four virtual CPU units with 14 GB allocated memory⁶. The server implementation for the proposed image segmentation was done using a Python application hosted on a computer. The specifications were discussed in Section 3.3.2. The Python application was not hosted on the cloud platform due to the time constraints of the study.

The DT framework was also utilised for the visualisation of the results from the path estimation and the image segmentation. This was done to integrate into existing DTs and also to enable the semi-automatic generation of new DTs. The results are presented and discussed in the following subsection.

3.4.3 Localisation and mapping

Experiment setup

The verification of the overall proposed low-cost semi-automatic DT generation method was performed using an experimental setup around an office building, for which a basic DT was created manually and is shown in Figures 3.13 and 3.14. The DT consists of a 2D image⁷ and two straight blue pipe sections along the left side of the office building in the centre of the image.

The experiment started on label 1 at the top point of the blue pipes, where the test subject initialised the sampling process. The test subject then walked in an anticlockwise direction around the office block (along the path of the pipes) towards label 2, turning left at the corner of the building, and continued walking in a straight line until the road was encountered, where the lower blue pipe terminates at label 3.

The test subject turned left again at label 3, where no blue pipe is present in the DT, and continued walking along the road towards and past label 4 in a straight line. After some time, the capture process was stopped, and all samples and images were stored for upload. The output from this experiment will show the results of updating and creating DT content from the estimated path and segmented images.

The portions where pipes already exist show how DT content will be updated, and the portions where no pipes exist show how new DT content will be created without reference points. Although not a deep-level mining environment, this verification will provide insights into the results of the overall proposed method.

⁶“Configure CPU limits for services”, Alphabet Inc., Google, <https://cloud.google.com/run/docs/configuring/services/cpu> (accessed 4 July 2025).

⁷“Anonymised location in South Africa”, Alphabet Inc., Google Imagery, <https://www.google.com/maps> (accessed 16 July 2025).



Figure 3.13: Top-down view of the DT used for verification.

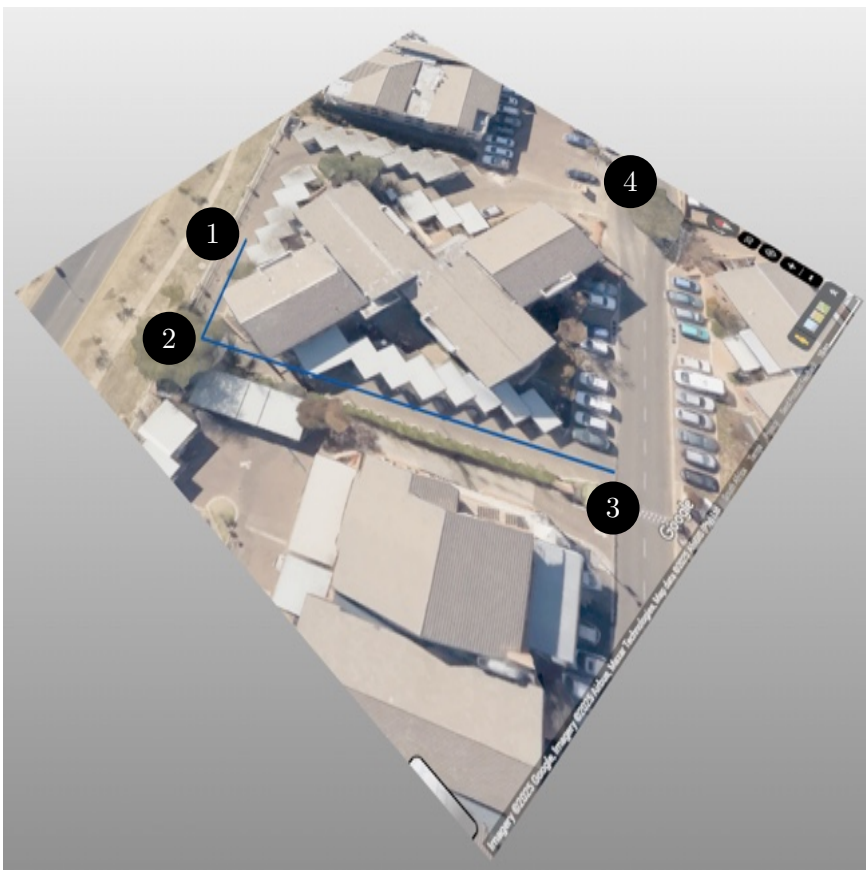


Figure 3.14: Orthographic view of the DT used for verification.

Estimated paths and segmentations

Following the automatic IMU sampling and image capturing using the mobile devices in Figure 3.12, the samples were uploaded to the server for processing. The processing consisted of utilising the implemented INS and image segmentation methods to convert the samples to an estimated path and segmented images, respectively. Following this step, each segmented image was assigned an estimated coordinate based on its capture time.

The server then started the DT creation and update steps, where each set of samples was iterated and processed. This involved obtaining and estimating the initial orientation based on the user input provided during the capturing process, such as the start position, as well as the existing DT content. The corresponding estimated coordinate of each segmented image was then used to determine whether the sample point correlates with a coordinate within an existing DT component, such as a pipe.

In the case of an existing component, the segmentation metadata was added to the component's known context, and the estimated coordinate was plotted in the DT. In the case of a component to be created, the estimated coordinate was plotted and added to an array of new DT content to be created. Figures 3.15 and 3.16 show the results obtained from the verification of the overall proposed method.

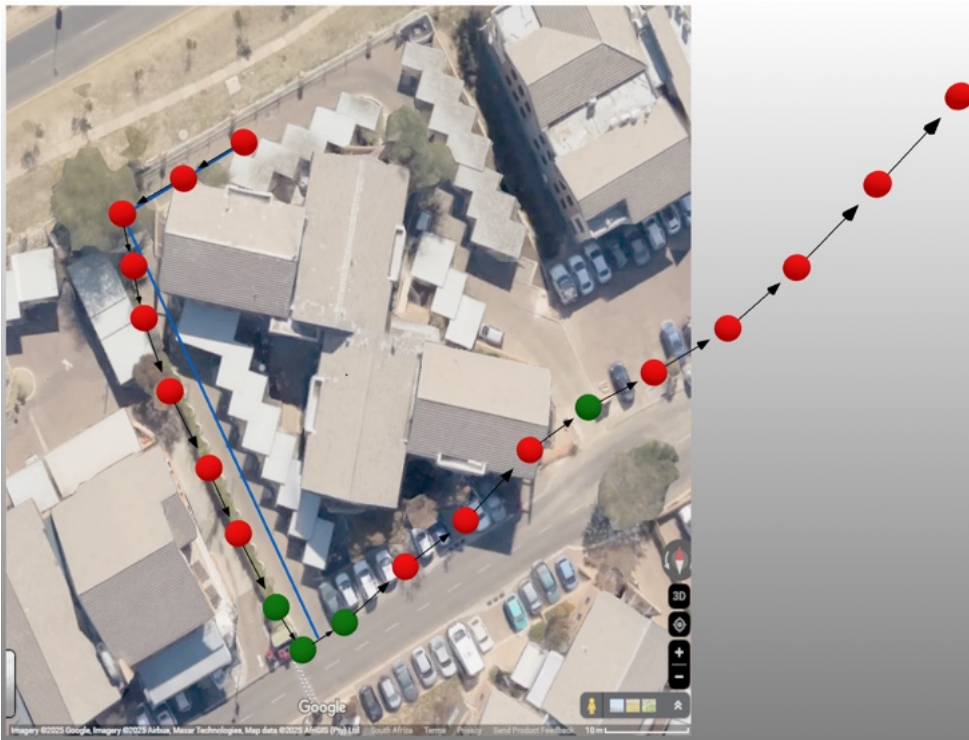


Figure 3.15: Top-down view of the verification for the overall proposed method.

Each sphere indicates the estimated positions where every image was captured using the proposed method. The black arrows between each sphere show the direction of the movement between them. A red sphere indicates that no object classes of interest were segmented, whereas

a green sphere indicates that one or more object classes were successfully segmented from the captured image at that estimated coordinate.

The first three estimated positions show a straight estimated path, as described in the experiment setup. Thereafter, when the test subject turned approximately 90° to the left, the fourth estimated position also shows an approximate 90° turn to the left. The angle of the first turn was slightly underestimated; however, the following estimated positions along the length of the second straight section alone correlated well with the actual path walked.

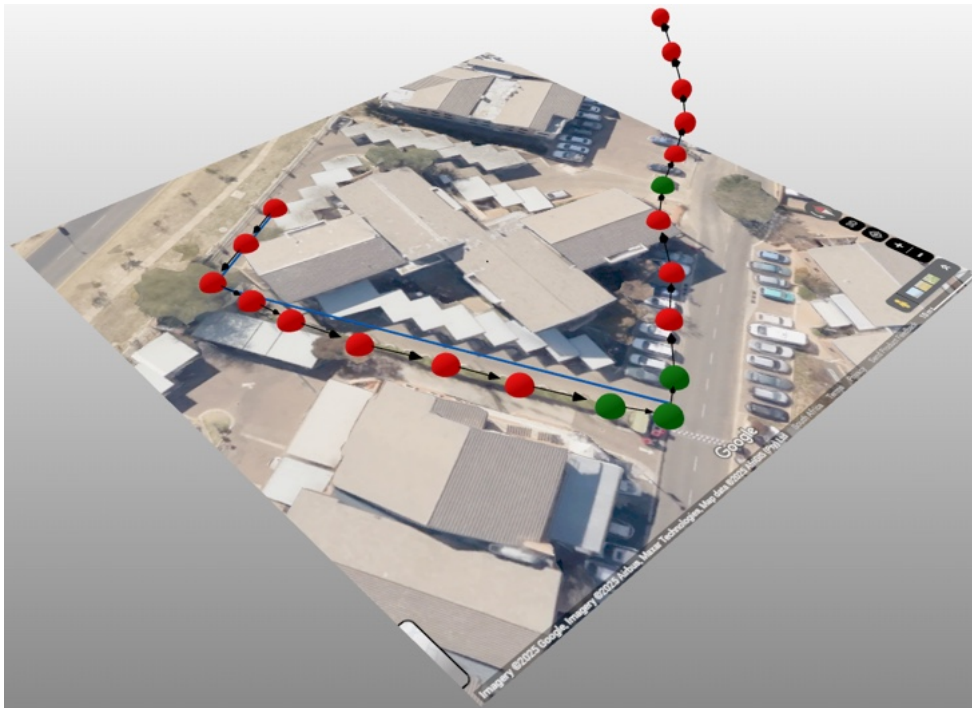


Figure 3.16: Orthographic view of the verification for the overall proposed method.

At the intersection with the road (label 3 in Figure 3.13), the approximately 90° turn to the left was estimated quite well. However, a slight overestimation of the second turn angle can be observed from the rest of the estimated positions along the road. This is because the actual path is along the edge of the road, not to the left of the road.

Although the estimated turn angles were slightly inaccurate, the full estimated path closely resembles the actual path. The last set of samples from Figures 3.15 and 3.16 verify that without blue pipes as references for the path, or even the 2D image, the proposed method is capable of generating new DT content.

The first two estimated sections reflect the actual straight path, and the third section along the road even shows the slight indentation of the road around the tree that grows in the median of the road (close to label 4 from Figure 3.13). Table 3.9 shows the path estimation performance.

Table 3.9: Quantitative analysis of the INS for DT generation.

Sensor		RMSE	PSNR (dB)
Accelerometer	x	0.4730 m/s ²	8.41
	y	0.3875 m/s ²	9.56
	z	2.0932 m/s ²	18.94
	Mean	0.9846 m/s ²	12.30
Gyroscope	x	0.2583 rad/s	35.60
	y	0.2283 rad/s	33.19
	z	0.1574 rad/s	39.69
	Mean	0.2147 rad/s	36.16
Magnetometer	x	0.1691 μ T	17.59
	y	0.2178 μ T	14.88
	z	0.1321 μ T	21.50
	Mean	0.1730 μ T	17.99

These results are similar to the results from the verification of the INS method in Section 3.2.4, with the accelerometer showing a slightly lower mean PSNR of 12.30 dB compared to 12.84 dB from Table 3.3. The magnetometer also showed a slightly lower mean PSNR of 17.99 dB compared to 19.19 dB. The filtering performed, in this case, better for the gyroscope, with a mean PSNR of 36.16 dB compared to 35.14 dB.

Figure 3.17 shows the segmentation result that caused the first sphere to be green on the path when approaching label 3 of Figure 3.13. Figure 3.18 shows the segmentation result that caused the last sphere to be green on the same path when approaching label 4. Figure 3.17b shows the segmentation result obtained, where the mini-substation's heat exchanger was successfully segmented and labelled with a confidence score of 90.68% and 94.66%.



(a) Relevant portion of the captured image.

(b) Segmentation result.

Figure 3.17: Segmentation result for the first green sphere.

Although the single heat exchanger was identified as two separate objects, the segmentation shapes cover the profile of the partially shaded object well. This is a promising result considering the inconsistent lighting conditions identified for deep-level mining. Segmentation performance will be analysed for low-light conditions in the case study in Section 3.5.

Two incorrect inferences were made in Figure 3.17b, where a part of a building's roof and a drain cover were incorrectly segmented as pipes. The second and third green spheres in Figures 3.15 and 3.16 were caused by the roof of the building being incorrectly labelled as a pipe and still visible from these locations. Figure 3.18b shows the segmentation result obtained from the input in Figure 3.18a for the final green sphere in Figure 3.16.

The water meter and the pipes connected to it in the middle of the image were segmented as a single valve, with a confidence score of 90.52%. It was segmented as a valve, as its shape bears a significant resemblance to the valves from the training dataset.



(a) Relevant portion of the captured image.

(b) Segmentation result.

Figure 3.18: Segmentation result for the last green sphere.

The profile of the water meter and the pipes was segmented with high precision, with only a small portion below it that can be considered incorrect. No other objects were segmented above the required confidence score for this captured image.

In this section, it has been verified that the method proposed in Section 2.5 addresses the correct objective of Section 1.8, that is, objective 3a. This is due to the performance achieved for the path estimation, image segmentations, as well as the semi-automatic DT generation. In the following section, a case study will be presented to validate the proposed methods and demonstrate that they successfully achieve the study objectives.

3.5 Case study

3.5.1 Requirements

The case study was chosen based on a set of requirements to ensure that it is relevant and provides an accurate representation of a typical real-world situation where the proposed methods could be used. Therefore, the first requirement was that it must be a deep-level mine in South Africa that reflects the typical operations and real-world challenges faced in the industry. Examples of this include GPS denial, inconsistent lighting conditions that cause poor visibility, elevated temperatures, and high humidity.

The second requirement was that the areas within the deep-level mine where the samples will be captured must be accessible and safe, as specified in the South African Mine Health and Safety Act (MHSA). The final requirement was that a basic DT layout of the deep-level mine should already exist, which will be used to validate the proposed methods. This will be achieved by improving existing and creating additional DT content. Table 3.10 summarises the requirement checks for the chosen case study, shown in Figure 3.19.

Table 3.10: Case study requirement checks.

Requirement	Description	Compliant
RSA deep-level mine	The case study must be a South African deep-level mine with extensive infrastructure and be in an operational condition that reflects the real-world challenges of this industry. This includes GPS denial, inconsistent lighting conditions, and harsh conditions.	✓
Accessible and safe	The areas of the mine where samples will be captured must be safe as defined in the MHSA of South Africa.	✓
Existing DT content	There must already exist a basic DT layout of the deep-level mine to validate the proposed methods by improving and updating the existing DT. This requirement also enables the test subject to capture information such as the starting position on the DT.	✓

3.5.2 Chosen case study

The chosen anonymised case study is an operational deep-level mine located in South Africa. The case study mine is approximately 2 km deep to provide some context. The areas where samples are to be taken have also been deemed accessible and safe according to the South African MHSA. Finally, existing DT content for this mine has already been created and is shown in Figure 3.19.

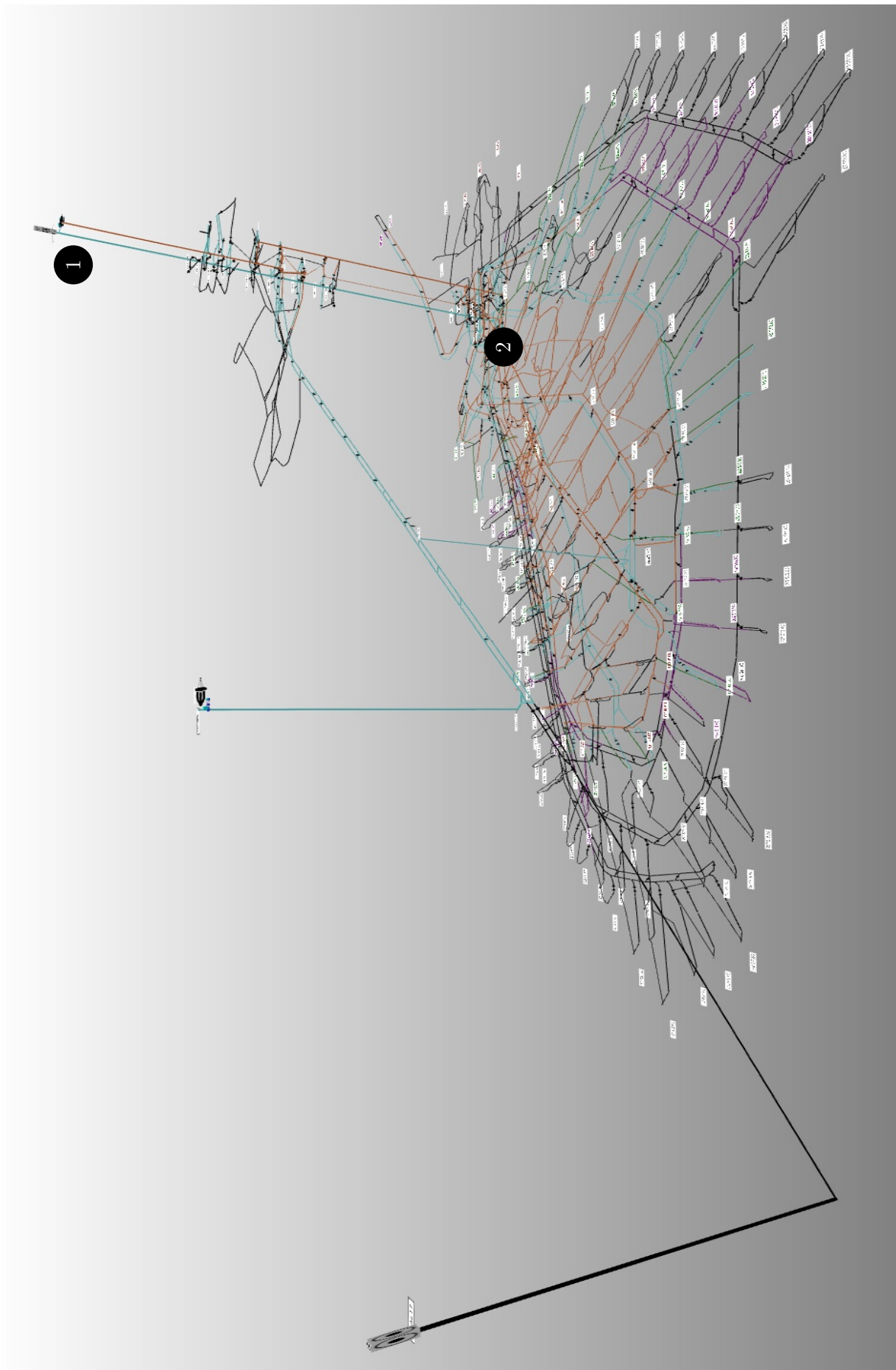
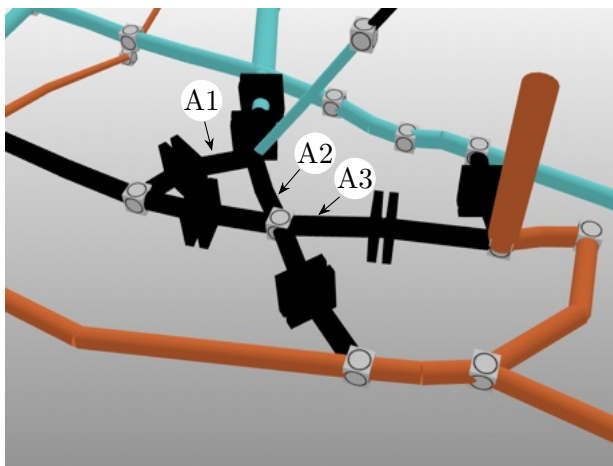


Figure 3.19: Orthographic view of the chosen case study.

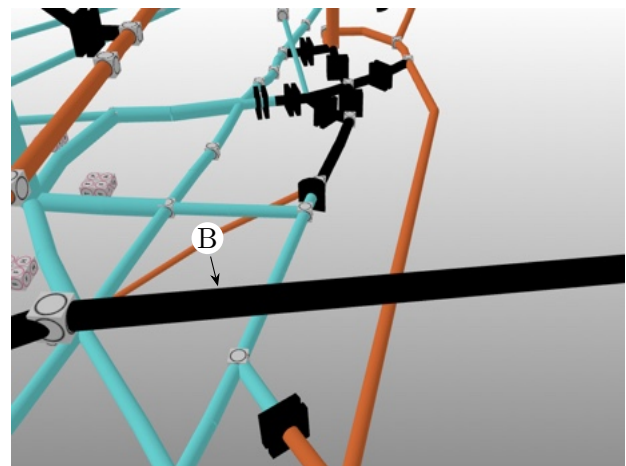
This DT content was manually created over several months through underground audits and the collection of planning schematics. Therefore, this case study supports the need for this study. If the study objectives are achieved, this will allow the semi-automatic generation of DT content for other deep-level mines in South Africa, as well as other parts of the world. This will also enable the improvement and addition of the DT content over time for the case study.

Label 1 in Figure 3.19 shows the location within the chosen case study of the main shaft, a vertical tunnel used to move skips and lift cages between the ground level and lower levels of the mine, typically using hoist motors and high tensile strength cables. Skips are used to transport ore for processing. In contrast, lift cages are used to transport personnel and small equipment. Label 2 shows the area within the deep-level mine where the samples were captured.

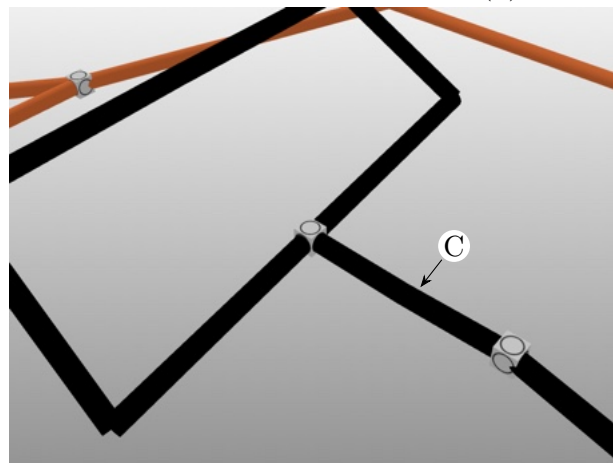
Figures 3.20a–c show the orthographic views of the three distinct areas within the case study mine where the samples were captured. Multiple sample sets were taken in the black tunnels that create the cross structure in the centre of Figure 3.20a, namely A1, A2, and A3. Sample set B was taken in the black tunnel in Figure 3.20b. Area A can be seen to the top of the image above label B. Figure 3.20c shows the final location where the last sample set C was captured.



(a) Data capture area A.



(b) Data capture area B.



(c) Data capture area C.

Figure 3.20: Orthographic view of the distinct sampling locations in the case study.

3.5.3 Generated DT content

The DT was used underground during the sampling process to select the appropriate tunnel and starting position from which the samples were taken. Following the IMU sampling and image capture process, as in the proposed method, the samples were uploaded to the server for processing once an Internet connection was established above ground.

Figures 3.21a–f show the estimated paths for the case study from the proposed semi-automatic DT generation method. The paths are represented by red and green spheres with black arrows indicating the movement between each sphere, as in the previous section. Figures 3.21a and b show the three estimated paths for the three sample sets taken in area A. These results show relatively straight estimated paths for Area A, as expected from the existing DT content as the spheres overlap the black tunnels.

Therefore, Area A shows promising results for the rest of the case study areas. Figures 3.21c and d show the estimated path from the sample set in area B, including the estimated paths from area A in the background. Figures 3.21e and f show the final estimated path from area C.

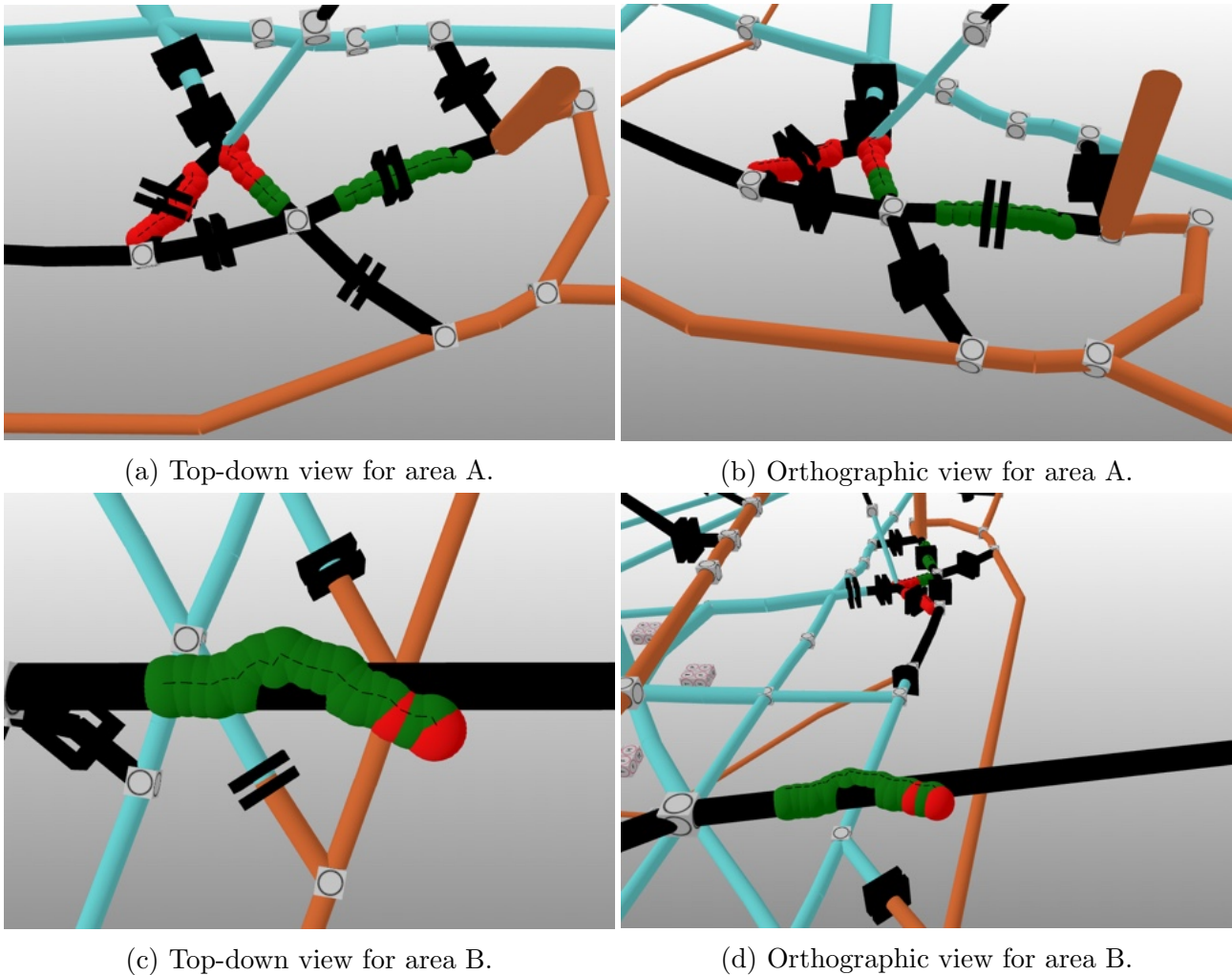


Figure 3.21: The semi-automatically generated DT content for the case study.

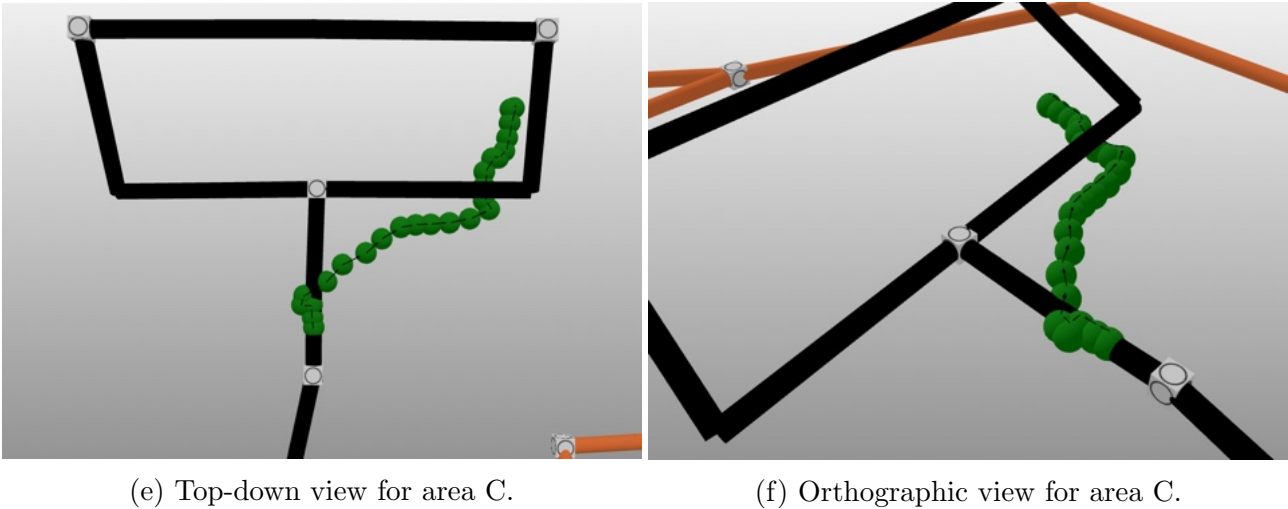


Figure 3.21: The semi-automatically generated DT content for the case study continued.

The estimated path for area B indicates that a straight line was not walked during the sampling process. Figure 3.22 shows this well, with an initial straight line from the first green sphere on the left side of the image towards label 1. A slight left turn is then made towards label 2. A turn to the right is made at label 3 and then again just before label 4. Thereafter, a relatively straight path is walked towards and past label 5.

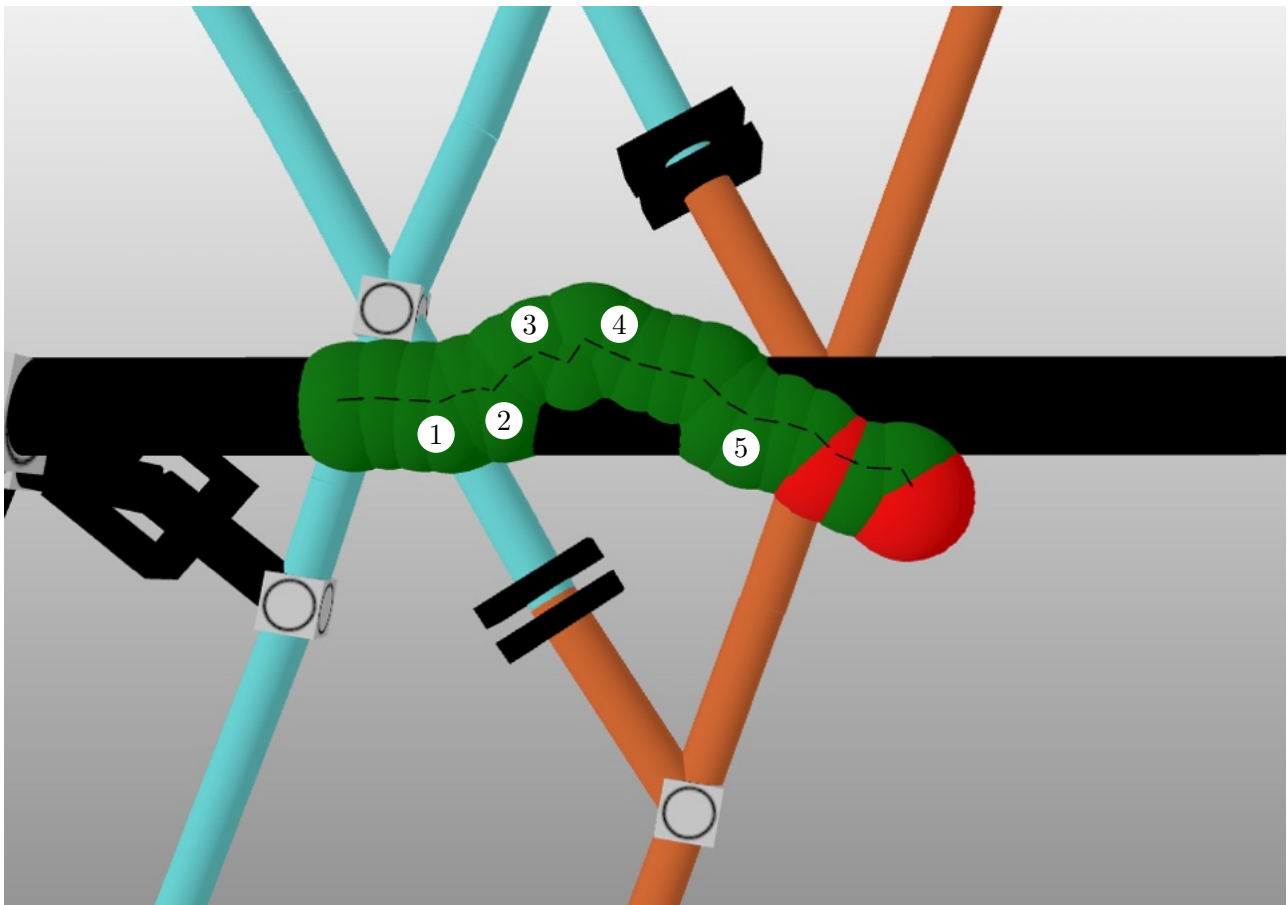


Figure 3.22: Annotated estimated path for area B.

Figures 3.23 and 3.24 show the images captured at labels 1 to 5 in area B, as well as the

corresponding segmentation results, in the correct order of capture. These images show a relatively straight tunnel, with a rail track in the centre of it. The track splits into two separate sections, as shown in Figures 3.23a and c, just after the hopper cart used to transport the mined materials. It can be seen from these images that the actual path walked starts to the right of the hopper cart in Figure 3.23a.

A straight line was walked along the cart until the end of it was reached, where the sampler turned to walk across the splitting tracks. This is evident in Figure 3.22, where the initial path is straight along the cart, and then a slight left turn toward label 2 occurs after label 1. The images for label 2 are shown in Figures 3.23c and e where the split track is being crossed from right to left. Figure 3.23e shows the same hopper cart in the centre of the image as in Figure 3.23a. Figure 3.24a shows the image for label 3 approaching the yellow barricades.

The next section of the actual path was to the left of the barricades. Afterwards, the sampler turned right and approached the next hopper cart in Figure 3.24c. This can be seen in the estimated path in Figure 3.22 between labels 3 and 4. Up to this point, sufficient lighting conditions were present. However, Figures 3.24c and e (representing labels 4 and 5 in Figure 3.22) show images captured where the present light source no longer fully illuminated the surrounding area, which frequently occurs in deep-level mines.

Therefore, considering the estimated path in Figure 3.22 and the images captured for it, the path estimation for area B in the case study does seem reasonable. Further analysis of the estimated paths for the case study will follow after the discussion of the segmentation results.

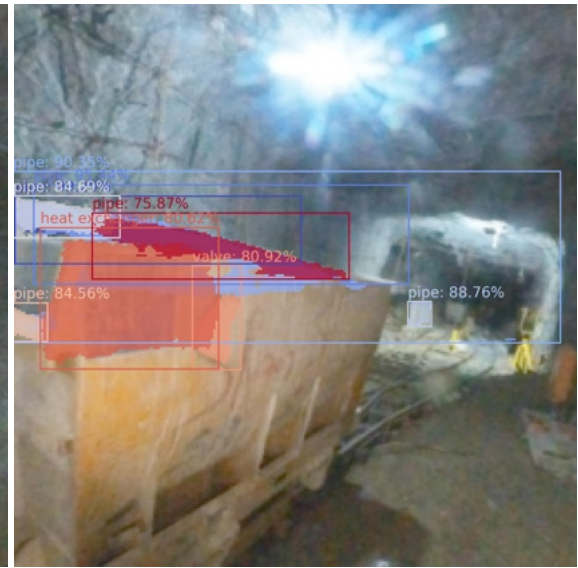
The image segmentation results in Figures 3.23 and 3.24 show promising results. Figure 3.23b shows that the pipes visible above the hopper cart were segmented with confidence scores of 84.69% and 75.87%. However, some false positive matches involved a portion of the hopper cart that was segmented as a heat exchanger.

Figure 3.23d, the front view for label 2, shows even better results, with both pipes on the left wall segmented with confidence scores of 98.9%, 97.74%, 93.7%, and 70.06%. The two large pipes were incorrectly segmented as four separate pipes; however, their overall profiles were accurately covered. The orange hoses below the two large pipes were also segmented as pipes with a confidence score of 90.76%, due to their similarity to the shape of the pipes in the dataset. The two yellow barricades were incorrectly segmented as pipes in this image.

Figure 3.23f, the rear view for label 2, again shows the two pipes on the right wall being segmented well with confidence scores of 89.22%, 88.89%, 85.04%, and 84.21%. There were three false positive matches for this image: the rail track, a portion of the concrete slab, and a portion of the hopper cart. Figure 3.24b, the front view for label 3, also shows the expected segmentation of the two pipes on the left side of the image. Three false positive matches were made for a portion of orange pipes, a barricade, and a portion between the split tracks.



(a) Image for label 1 (front view).



(b) Corresponding segmentation result.



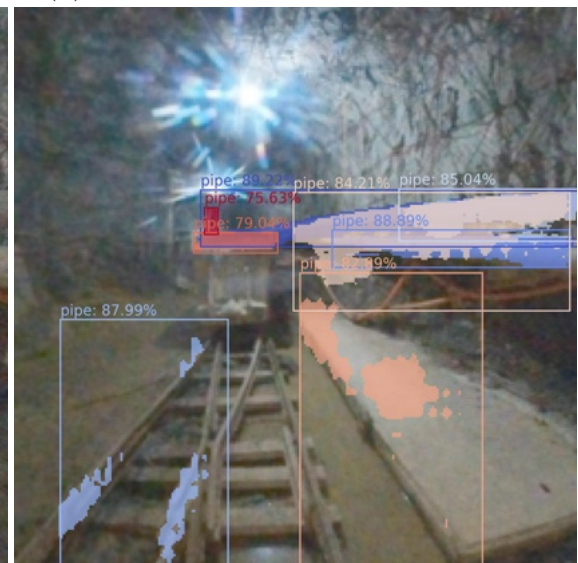
(c) Image for label 2 (front view).



(d) Corresponding segmentation result.



(e) Image for label 2 (rear view).



(f) Corresponding segmentation result.

Figure 3.23: The image segmentations for area B in Figure 3.22 (part 1).



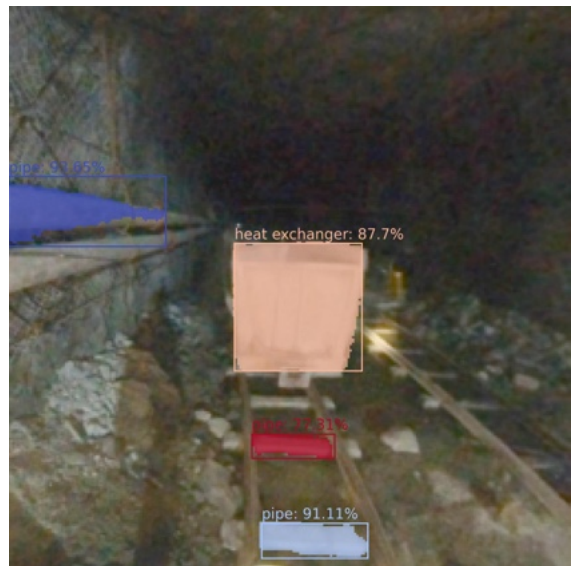
(a) Image for label 3 (front view).



(b) Corresponding segmentation result.



(c) Image for label 4 (front view).



(d) Corresponding segmentation result.



(e) Image for label 5 (rear view).



(f) Corresponding segmentation result.

Figure 3.24: The image segmentations for area B in Figure 3.22 (part 2).

Figure 3.24d, the front view of label 4, shows that only one of the pipes on the left wall was segmented with a confidence score of 93.65%. The hopper cart was incorrectly segmented as a heat exchanger due to its similarity to the heat exchangers in the dataset. Two rail track sleepers were also incorrectly segmented as pipes.

Figure 3.24e shows the rear view for label 5, where the light source was too far away to provide adequate illumination. However, the corresponding segmentation result in Figure 3.24f shows that the pipes above the hopper carts were successfully identified as a pipe with a confidence score of 71.12% despite poor illumination.

Figure 3.25 shows the annotated estimated path from area C. From the start of the sample set at label 1, it appears that an initial straight line was walked until an obstacle was encountered just before arriving at label 2, which required the sampler to walk around it. Following this, from label 3, another relatively straight line was estimated until just after label 4, where a second obstacle appears to have been encountered due to another sharp turn. It is likely that the rotation angles were under- or overestimated. Figures 3.26a–f show the images for area C.

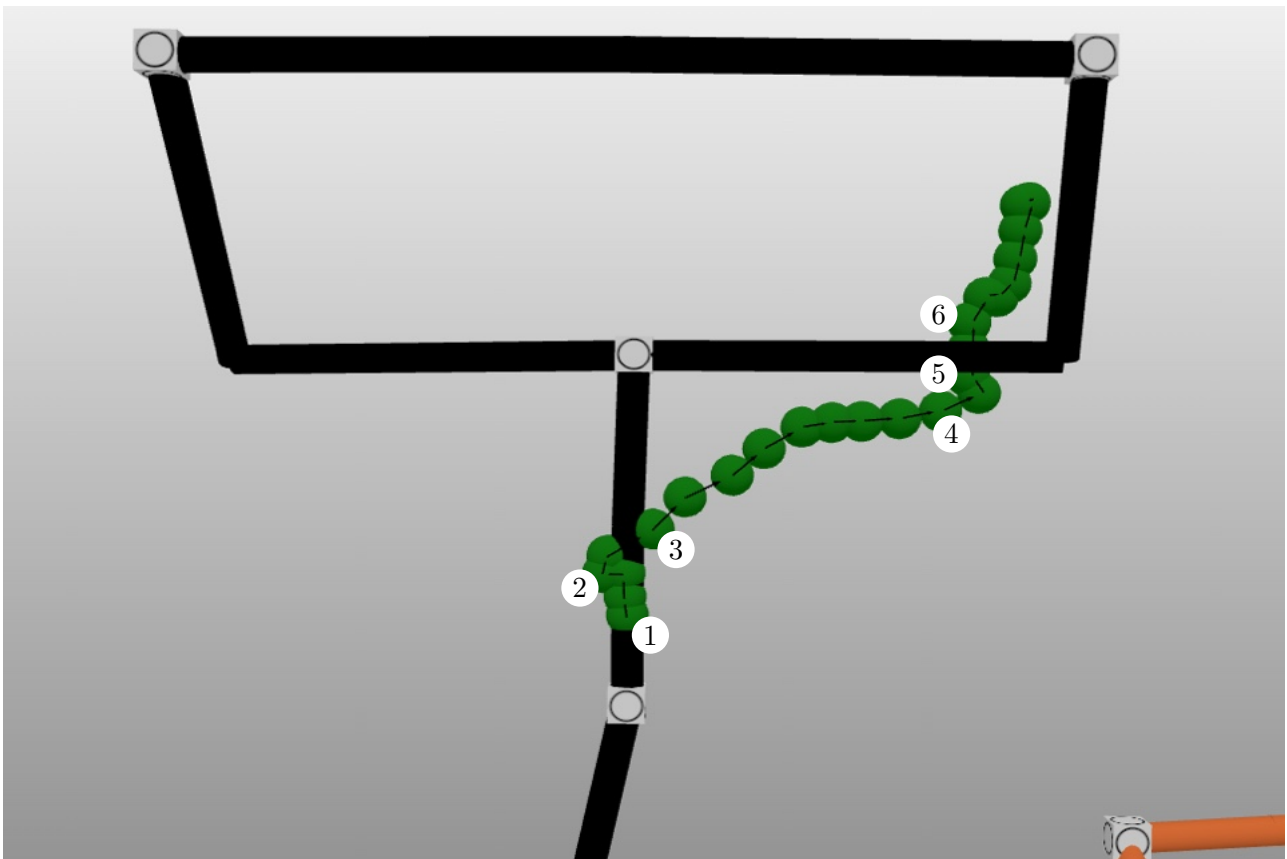


Figure 3.25: Annotated estimated path for area C.

Figures 3.26a–f show that the samples were taken in a brightly lit straight tunnel containing various pumps and pipes. This type of area is typically referred to as a pumping station in a deep-level mine. Interestingly, it can be seen from all these images that the floor of area C is flooded with water that is approximately shin deep. This flooding is not supposed to occur under standard mining operations, as dewatering should avoid flooding.

Figures 3.26a, the front view for label 1, and 3.26b, the front view for label 2, reflect the same straight section as the estimated path towards the orange centrifugal pump motor in the middle of the walkway. This was the obstruction encountered that caused the sharp left turn in the estimated path just before label 2. Figure 3.26c, the rear view for label 3, confirms that the sampler was required to walk around the centrifugal pump motor to continue down the walkway towards label 4.



(a) Image for label 1 (front view).



(b) Image for label 2 (front view).



(c) Image for label 3 (rear view).



(d) Image for label 4 (front view).



(e) Image for label 5 (front view).



(f) Image for label 6 (rear view).

Figure 3.26: The images captured for area C in Figure 3.25.

From the first three images, it can be confirmed that the estimated path does reflect the left turn around the obstacle. However, the following section of the estimated path does not reflect that the user continued in the same direction after label 3, as the initial straight line from label 1. This was caused by an underestimation of the turn angle after the obstacle. This may be due to the unexpected influence that walking through water has on the performance of the proposed path estimation method.

Figures 3.26d–f show the second obstacle on the path, which caused the sharp turn to the left between labels 4 and 5. Again, the sampler walked around the obstacle, as seen in the images. However, the estimated turn angle after the obstacle was overestimated, causing a discrepancy between the actual path and the estimated path.

The flooded walkway negatively influenced the path estimation for area C to be the worst among all areas in the case study, as already shown in Figure 3.28. Despite this, the obstacles in the walkway were still identified quite well. Figures 3.27a and b show an image segmentation result for area C between labels 3 and 4.

Two of the large pipes were well segmented with confidence scores of 83.71% and 76.5%. The profile of the centrifugal pump motor in the middle of the image was mostly segmented; however, it was misidentified as a valve. The centrifugal pump motor to the left of the image, which represented the first obstacle on the estimated path, was segmented, although it was mislabelled as a valve with a confidence score of 94.34%.



(a) Image between label 3 and 4 (rear view).

(b) Corresponding segmentation result.

Figure 3.27: Image segmentation performance for area C in Figure 3.25.

Other objects within the image were also incorrectly segmented and labelled. This is likely due to the brightly lit environment, which may have caused overexposure in the captured images, leading to degraded performance compared to area B in Figures 3.23 and 3.24.

3.5.4 Summarised performance

The overall inertial navigation performance for the case study, in terms of the averaged PSNR for each sensor over each set of samples, is shown in Figure 3.28. The best PSNR values for the accelerometer (17.82 dB) and gyroscope (40.72 dB) were obtained for area A3. The best PSNR value for the magnetometer (18.78 dB) was obtained for area B. The path estimation and segmentation results for area B were discussed and are shown in Figures 3.22, 3.23, and 3.24.

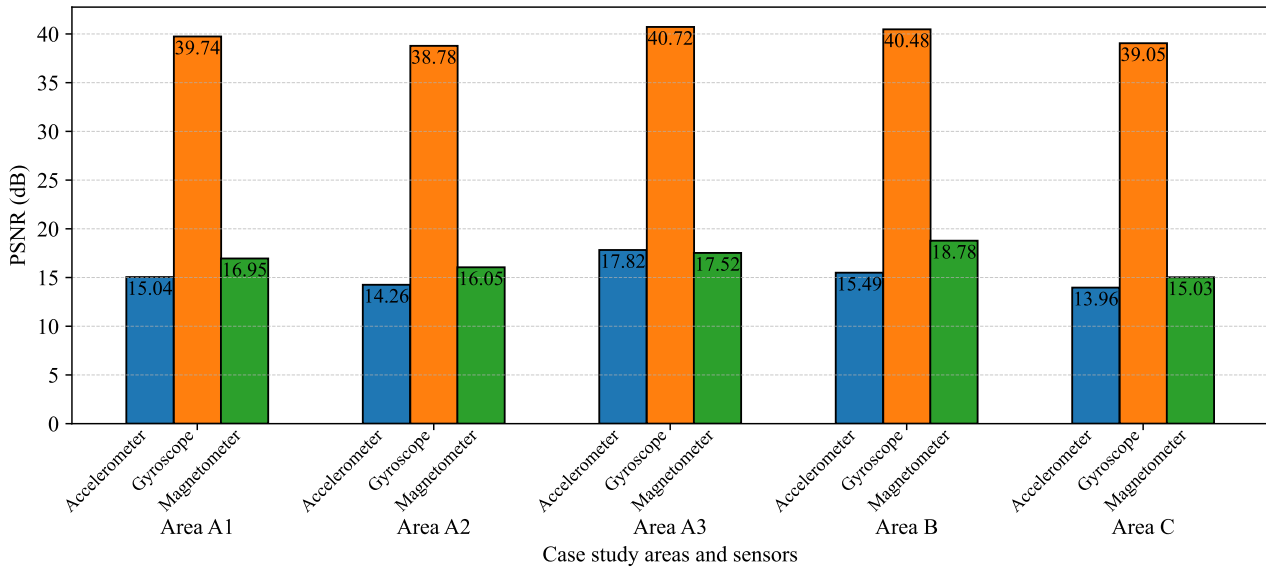


Figure 3.28: Inertial navigation PSNR results for the case study.

The worst PSNR results were obtained for area C, except for the gyroscope, which yielded the second-worst result. Area C does not, as the other path estimations, align well with the existing DT content. This is most likely due to the poor PSNR results obtained for the path estimation process.

Table 3.11 shows a comparison between the PSNR results achieved in Section 3.2.4, the verification of the INS method, as well as Section 3.4.3, the verification of the DT generation method and, finally, the results achieved for the case study. It is worth noting that the PSNR for the accelerometer and gyroscope was the highest for the case study. However, the PSNR for the gyroscope in the case study showed the worst result.

Table 3.11: Comparison of all PSNR results achieved.

Sensor	INS	DT generation	Case study
Accelerometer	12.84 dB	12.30 dB	15.31 dB
Gyroscope	35.14 dB	36.16 dB	39.75 dB
Magnetometer	19.19 dB	17.99 dB	16.86 dB

The magnetometer appears to have the most significant effect on the result, as indicated by the estimated path for area C and its corresponding PSNR of 15.03 dB. This can also be seen from the highest PSNR obtained for the magnetometer for area B, where the estimated path was highly representative of the actual path. The image segmentation for area B provided promising results, even in low-light conditions, as shown in Figures 3.24d and f.

The PSNR results achieved using the proposed UKF algorithm are in the range of what can be expected compared to a recent study by Farahan et al. [84], where 45 dB, 25 dB, and 5 dB for each respective sensor in Table 3.11 was achieved. The proposed UKF algorithm achieved higher PSNR results for the gyroscope and magnetometer, as expected [132].

Table 3.12 provides a summary of the actual objects present in the case study figures versus the objects segmented by the proposed method for the case study. Pipes are present in all the figures, whereas four valves, a centrifugal pump, as well as an induction motor are present in the final figure.

Table 3.12: Actual objects in the case study versus objects segmented.

Figure	Actual objects	Segmented objects
3.23a label 1 (front)	Pipes (2)	Pipes (6) Heat exchangers (1) Valves (1)
3.23c label 2 (front)	Pipes (2)	Pipes (8)
3.23e label 2 (rear)	Pipes (2)	Pipes (8)
3.24a label 3 (front)	Pipes (2)	Pipes (9)
3.24c label 4 (front)	Pipes (2)	Pipes (3) Heat exchangers (1)
3.24e label 5 (rear)	Pipes (2)	Pipes (1)
3.27a label 3–4 (rear)	Pipes (9) Valves (4) Centrifugal pumps (1) Induction motors (1)	Pipes (6) Valves (3)

The segmentation results for all the figures identified at least one and at most nine pipes. The number of pipes segmented for Figures 3.24e and 3.27a aligned the closest to the actual number of pipes in the figure. Multiple false-positive matches were made for the pipe class in the first five figures, as many objects resemble a pipe’s structure. Two hopper carts were mislabelled as heat exchangers in Figures 3.23a and 3.24c. A false-positive match was also made for a valve in Figure 3.23a.

Although not perfectly accurate, these automatic segmentations provide a good indication as to the different object classes present in images captured within a deep-level mine. The segmented object profiles within these images, as shown in Figures 3.23, 3.24, and 3.27, are highly promising even in low-light conditions.

Therefore, after verification of each proposed method individually, as well as the validation of the overall performance for all five areas in the case study, it has been shown that the proposed method does enable low-cost semi-automatic DT generation in South African deep-level mines.

3.6 Summary

The results of this study were presented and analysed in this chapter. The first section discussed the verification and validation strategy, which consists of the individual verification of each proposed method using experiments with known paths and segmentation outputs. The strategy also included the validation of the overall proposed method using a case study that accurately represents the real-world situation.

In Section 3.2, the proposed INS method was verified. This was done, firstly, by analysing the data collected over a known test path by the implemented IMU sampler. This data reflected similar sensor noise as discussed in the literature analysis of Section 1.5. The results from the UKF algorithm for the sampled IMU data were then presented. The filtered data showed a significant reduction in sensor noise with PSNR values of 12.38 dB, 34.11 dB, and 17.68 dB for the accelerometer, gyroscope, and magnetometer, respectively.

Thereafter, the motion equations were used to convert the filtered IMU data from the mobile reference frame to the Earth reference frame. This yielded an average walking speed of 1.37 m/s over a total distance of 27.1 m. This walking speed was similar to that determined in the study by Bohannon et al. [133]. The error between the estimated and actual path walked was 0.3 m.

The complete proposed INS method was then verified using 60 seconds of IMU data from a known test path. The final position error was 2.4 m, with PSNR values similar to the previous test. An average walking speed of 1.34 m/s was calculated, also similar to the previous test. Therefore, the proposed INS method was verified using these results.

In Section 3.3, the proposed image segmentation method was verified. This was done, firstly, by analysing the results of the synthetic dataset generation method, which consists of creating multiple different digital scenes representing the real-world application and then automatically labelling multiple perspectives of each scene using a raycaster. These results are shown in Figures 3.7 and 3.8, which confirm that a synthetic dataset was successfully generated for image segmentation.

Subsequently, the proposed transfer learning method was implemented using an existing image segmentation architecture and fine-tuning it using the synthetic dataset generated for the deep-level mining application. The implementation of the transfer learning method resulted in a PQ score of 49.16 and an IoU score of 39.35 in the final verification dataset, which is within a similar range achieved by recent studies, as shown in Table 3.7.

Following this, real-world images from deep-level mines were segmented using the final trained image segmentation model. Promising results were achieved, showing that the profiles of various pipes, valves, and centrifugal pumps were successfully segmented and identified. An IoU score of 34.15 and a PQ score of 32.41 were calculated using manually labelled ground truth

segmentations of the real-world images. Performance was slightly lower for real-world images compared to the verification dataset, which was expected due to the reality gap phenomenon.

Next, the verification of the overall proposed low-cost semi-automatic DT generation method was performed. This was done using an experimental setup around an office block with known expected results. The DT content generated in Figures 3.15 and 3.16 and the PSNR results (similar to those achieved in Section 3.2) showed a high-accuracy estimate of the path.

The generated DT content also included the estimated locations of successful image segmentation matches, as shown in Figures 3.17 and 3.18. The first image shows the successful segmentation of a heat exchanger from a mini-substation encountered on the path. The second shows a water meter and pipes, which were misidentified as valves (since they closely resemble the valves in the dataset); however, its profile was segmented with high precision.

The final section of this chapter contains the validation of the overall proposed method using a case study of a South African deep-level mine. The case study was chosen based on the requirements in Table 3.10. A total of five sets of samples were collected in three different areas of the mine.

The path estimation results for area B were validated using the images captured for this sample set. The path traversed, as seen in Figures 3.23 and 3.24, showed that the path estimation does reflect the actual path quite accurately. The segmentation results for this sample set also showed promising results, with various objects correctly segmented, even under low light conditions.

The path estimation results for area C were not as accurate as for area B. This may be due to the magnetic interference of the first obstacle, the centrifugal pump motor windings, as reflected by the PSNR for the magnetometer of 15.03 dB. There were also unforeseen influences on the IMU data of the shin-deep water through which the sampler walked.

However, inspection of the captured images for area C shows that the two obstacles encountered in the actual path were identified quite well in the estimated path, although the direction of the estimated straight sections was not highly accurate. The segmentation performance for these images was also degraded compared to the results for area B. This is possibly due to the high number of bright lights, which oversaturated the captured images.

The following chapter presents a discussion of the research, key findings and insights, as well as limitations to the proposed methods. The contributions to the field and recommendations for future work will be discussed, and finally, the study will be concluded.

4 Conclusion

4.1 Discussion of the research

The South African deep-level mining industry has faced increasing labour and operational costs in recent years. The industry has been searching for and implementing potential solutions to this problem to remain profitable, continue supporting hundreds of thousands of households, and continue contributing to South Africa's GDP. Therefore, the Industry 4.0 paradigm has come to the attention of the industry, following its introduction at the Hanover Trade Fair in 2011 as part of Germany's strategy to reduce production costs and increase profitability.

The paradigm consists of digitising complex real-world systems and integrating advanced technologies. One of these technologies, the DT, and specifically the current expensive manual and semi-automatic creation methods, forms part of the problem statement for this study. These existing high-cost methods, due to their manual nature and the equipment required, respectively, restrict the implementation rate of DTs, which therefore limits the adoption rate of the Industry 4.0 paradigm.

This led to the literature analysis of SLAM methods, which are widely used in the robotics industry for simultaneous navigation and generation of digital content. Typical navigation methods, such as GPS, are not available in deep-level mines. The alternatives, such as fibre optic and WiFi networks that provide full coverage throughout complex tunnel networks, are not feasible at the time of this study. Furthermore, existing LiDAR approaches are expensive and require specialised equipment.

Inertial navigation is one of the methods in SLAM that has shown promising results in popular problems such as pedestrian localisation in urban environments. Inertial navigation is a method that uses accelerometer, gyroscope, and magnetometer data from low-cost devices to perform path estimation. However, this inertial data contains sensor error and bias. The deep-level mining environment also introduces additional noise, namely EMI, from large equipment, such as powerful motors for centrifugal pumps.

Deep learning image segmentation, a popular method used to extract information automatically from images, has also shown promising results in recent studies. However, no existing deep-level mining image segmentation dataset was found at the time of this study. Additionally, manual creation of such a dataset would be time-consuming and expensive. Despite this, synthetic datasets, which can be generated automatically, have produced well-performing segmentation models when utilising transfer learning to bridge the reality gap.

Therefore, a research gap for this study was identified from the literature analysis for a low-cost semi-automatic DT generation method, utilising inertial navigation and image segmentation, for high-noise environments. This research gap led to the objectives defined for this study, which

consists of, first, the creation of a low-cost method for localisation in high-noise environments.

The second objective is the creation of a low-cost method for image segmentation in high-noise environments, which included a synthetic dataset generator for high-noise environments. The third and final objective is the creation of an overall method for low-cost semi-automatic DT generation for a deep-level mining environment.

In Chapter 2, the methodology and the proposed methods for this study were presented and discussed. The methodology was divided into three topics: inertial navigation, image segmentation, and low-cost semi-automatic DT generation. Within each of these topics were four stages: investigation, design, verification, and implementation. This methodology was created to achieve the objectives of this study.

The initial step of the proposed inertial navigation method consists of defining the IMU data in the mobile reference frame in relation to the Earth reference frame. The second and arguably the most impactful step is the advanced filtering algorithm used to reduce sensor noise and bias, as well as to reduce EMI from the environment. The last step is the conversion of the filtered IMU data from the mobile reference frame to the Earth reference frame to determine the estimated path using the motion equations.

The first half of the proposed image segmentation method consists of the synthetic dataset generation. This was divided into the creation of digital scenes, the automatic labelling using a raycaster, and conversion from raycast results to polygons to populate the dataset. The second half of the method consists of the transfer learning approach, which uses the synthetic dataset created to fine-tune a segmentation model that is initialised using pre-trained parameters.

The overall method for low-cost semi-automatic DT generation consists of integrating and extending the previous two methods to generate new DT content. Chapter 3 opens with the proposed verification and validation strategy, which was used to determine the effectiveness of each proposed method and whether the objectives of the study were achieved. Further, the verification of each proposed method was done independently to show generalisability.

The verification of the proposed inertial navigation method produced promising results. It showed a significant PSNR filtering output of 22.39 dB, a feasible estimated walking speed of 1.34 m/s, and positional error of 2.4 m over a total distance of 80.6 m.

The verification of the proposed image segmentation method also produced positive results. It showed successful generation of a synthetic dataset and well-performing transfer learning results with a final PQ and IoU score of 32.41 and 34.15 on unseen real-world images of the application, which is reasonable compared to recent studies.

The implementation of the proposed low-cost semi-automatic DT generation method was done using a low-cost mobile device, a separate image-capturing device, as well as a hybrid server

implementation to process the captured samples using cloud-based computational resources and a laptop. Verification of the overall method was done by capturing samples around an office block for which a basic DT was constructed and showed promising results.

The path estimation reflected the actual path quite accurately, with a PSNR of 22.15 dB, similar to that achieved in the previous verification. The image segmentation results also performed well, showing the identification and profile matching of a heat exchanger, as well as a water meter and pipes. These results were reflected in the generated DT content.

A case study of a South African deep-level mine, chosen using a defined set of requirements, was used for the validation of the proposed methods. Five sample sets were captured in different areas throughout the mine. The first three sample sets in area A showed straight path estimations, as expected from the existing DT content. The fourth sample set, which was captured in area B, did not reflect the same straight line as shown by the existing DT content. However, the path estimation proved accurate from inspection of the captured images.

This inspection consisted of analysing the movement of the sampler by referencing static objects in each captured 360° image, such as rail tracks and pipes attached to the wall of the tunnel. The actual path, where the sampler initially walked straight along the hopper carts, then turned left across the rail tracks, turned right after some time and finally continued in a straight line, was well reflected by the path estimation results.

The analysis of the image segmentation results for this sample set showed high performance. The most prevalent object class identified was pipes, as expected. The implemented segmentation model also performed well under low light conditions, correctly identifying and profiling pipes against the tunnel wall above the hopper carts in the last few images of this sample set.

The path estimation of the final sample set appeared to be inaccurate. However, after inspection of the segmented images, the two obstacles encountered on the actual path were reflected well in the estimated path. Two of the rotation angles on the path were not estimated well, which caused the unexpected result. This may have been due to the flooded walkway encountered in this sample set, combined with the close proximity of the centrifugal pump motor.

The PSNR for all sample sets in the case study showed similar results to those achieved for the inertial navigation as well as the DT generation. The proposed methods performed the best for area B in terms of PSNR. In contrast, the worst results were obtained for area C. This was reflected in the path estimations achieved.

Table 4.1 shows a summary of the novel contributions and objectives presented in Chapter 1. It also includes the specific sections within this thesis where these objectives were completed and the novel contributions were achieved.

Table 4.1: Objectives achieved and corresponding novel contributions.

Novel contribution	Objective	Sections	Achieved
2. A low-cost semi-automatic DT generation method for deep-level mines.	1a. Implement an IMU sampler for mobile devices.	2.3.1, 3.2.1	✓
	1b. Design and implement a KF-based inertial navigation for underground localisation.	2.3.2, 3.2.2	✓
1. A low-cost method to generate synthetic datasets for deep-learning image segmentation.	2a. Use a low-cost camera to implement an image sampler controlled through a mobile device for underground environments.	3.4	✓
	2b. Design and implement a synthetic dataset generator to train deep-learning image segmentation models for high-noise environments.	2.4.1, 3.3.1	✓
	2c. Apply transfer learning to an image segmentation model using a generated synthetic dataset.	2.4.2, 3.3.2	✓
2. A low-cost semi-automatic DT generation method for deep-level mines.	3a. Design and implement the low-cost semi-automatic DT generation method.	2.5, 3.4	✓
	3b. Validate the proposed methods in a real-world deep-level mining case study.	3.5	✓

The verification, using known experiments, and the validation, through a case study, of the proposed methods show that the objectives of the study were completed successfully. This, in turn, demonstrates that the novel contributions were achieved.

Therefore, the need for the study has been addressed and the problem statement has been resolved. However, future research should be done to further improve the robustness and accuracy of the proposed inertial navigation, image segmentation, and semi-automatic DT generation methods.

4.2 Key findings, insights, and limitations

4.2.1 Inertial navigation

The proposed UKF-based inertial navigation method performed well within certain constraints, such as distances up to 80 m and moderate walking speeds and rotations. Unexpected walking conditions, such as flooded walkways, as well as extreme EMI when walking next to the motor of a large centrifugal pump, are a challenge for the proposed inertial navigation method without external correction.

However, the results obtained for the verification of the proposed method and the case study answer a portion of the research question for novel contribution (NC) 2: It is possible to design and implement a low-cost path estimation method for high-noise offline environments, such as deep-level mines.

4.2.2 Image segmentation

The proposed method that utilises a generated synthetic dataset and a transfer learning approach produced well-performing image segmentation for the case study deep-level mine. Object classes from the dataset were successfully identified and segmented, even in low-light conditions.

False positive segmentations were found in these results. This may be avoided in the future by adding unrelated objects to the synthetic scenes of the dataset generator. This will increase the effectiveness of the training against false-positives by affecting the IoU and PQ metrics achieved each epoch.

The results obtained for the proposed method, as well as the published journal article in Appendix 1, including the case study, answer the research question for NC 1: It is possible to design and implement a low-cost synthetic dataset generation method for image segmentation of high-noise environments, such as deep-level mines.

4.2.3 Low-cost semi-automatic DT generation

The proposed method that utilises SLAM-style inertial navigation and image segmentation to generate DT content semi-automatically performed well, as shown in the case study validation results. Multiple sample sets were captured in the high-noise offline environment, and after processing, DT content was generated from the results. This generated content consisted of segmented images linked to estimated coordinates within the DT of the case study.

Images were sampled in intervals of approximately 5 seconds, which may cause some information

loss between each sample. Higher image sampling rates will reduce information loss, but will need to be balanced against the available storage capacity and transfer latencies of the mobile device. The inertial data was sampled at 100 Hz, which is the typical upper limit of low-cost mobile devices. Higher sampling rates may reduce path estimation error.

The overall results obtained for the the case study answer the research question for NC 2: It is feasible to design and implement a low-cost semi-automatic DT generation method for high-noise offline environments, such as deep-level mines.

4.3 Contributions to the field

This research contributes to the field by proposing a low-cost method to generate synthetic datasets for deep-learning image segmentation in high-noise environments, as well as a low-cost semi-automatic DT generation method for deep-level mines. These two novel contributions were validated through a South African deep-level mining case study.

The verification of the proposed methods and the validation using the case study also demonstrated that a modular SLAM framework is a feasible approach to generate DT content for high-noise offline environments. The case study also confirmed that the use of commodity low-cost mobile devices and scalable cloud computing for the implementation of the proposed method is feasible.

The conclusion of this thesis has shown that semi-automatic low-cost DT generation is possible for the deep-level mining environment. Therefore, the practical implication for the deep-level mining industry is that the proposed method can be used to create DTs for the underground environment in a more feasible manner and, in particular, the industry is a step closer to fully automatic DT generation.

4.4 Recommendations for future work

The proposed inertial navigation method can be extended with a deep-learning algorithm to update the state equation over time, possibly reducing sensor error and EMI influence further for enhanced path estimation. However, this will require the creation of a new inertial odometry dataset specifically for deep-level mining conditions.

Path estimation on the mobile device during sampling can be investigated to enable real-time localisation. The performance of the inertial navigation algorithm on mobile devices will need to be investigated to achieve this. Additionally, more than one mobile device may be utilised during the sampling process to cross-reference independent IMU samples for improved path estimation.

More extensive domain adaptation methods can be applied to the segmentation dataset generator, such as adversarial learning. This is a method where, during training, a segmentation is made and an adversarial model is used to detect whether or not the segmentation is unrealistic or can be considered a false-negative result. This will further improve the performance of the image segmentation model.

The effect of a well balanced dataset should also be investigated, as this has a high likelihood of increasing segmentation performance across all classes in the synthetic dataset. Additionally, new deep-learning architectures not available at the time of this study can also be tested using the proposed segmentation method to investigate possible performance improvements.

Photogrammetry methods can be used to enhance the content created by the overall semi-automatic DT generation method. Photogrammetry is a method used to extract reliable 3D measurements of real-world objects from static images. This method can also be applied to video footage, further enhancing the possible results that can be obtained.

Finally, fully automatic DT generation is the main recommendation for future research, as this will increase DT utilisation further compared to the proposed semi-automatic generation method. This fully automated generation method will enable DTs to be kept aligned with the corresponding real-world systems more efficiently as they change over time. These DTs will therefore be a more accurate reflection of the real-world system.

4.5 Closing reflections

An increased adoption rate of the Industry 4.0 paradigm, including technologies such as DTs, is critical for the South African deep-level mining industry to remain profitable and continue supporting hundreds of thousands of households and contribute to the GDP. Efficient DT creation is vital for the feasible implementation of the paradigm in the industry.

This research has shown that a modular SLAM approach is a feasible method for low-cost semi-automatic DT generation. Further enhancement of the proposed methods has the potential to lead to fully automatic low-cost DT generation, which will result in increased DT utilisation and Industry 4.0 adoption. The potential for practical applications of SLAM and deep-learning methods combined with cloud computing and commodity devices continues to increase, leading to exciting new opportunities on the horizon.

References

- [1] W. J. Kilian, J. Prinsloo, J. Vosloo, and S. Taljaard, “A synthetic segmentation dataset generator using a 3D modeling framework and raycaster: a mining industry application,” *Frontiers in Artificial Intelligence*, vol. 7, 2024, doi: 10.3389/frai.2024.1453931.
- [2] “Comprehensive facts and figures,” Minerals Council South Africa, Johannesburg, South Africa, Technical Report, 2023, [Online] Available: <https://www.mineralscouncil.org.za/component/jdownloads/?task=download.send&id=2310&catid=17&m=0>.
- [3] J. Swanepoel, J. Vosloo, J. van Laar, and W. Pelser, “Prioritisation of environmental improvement projects in deep-level mine ventilation systems,” *Mining, Metallurgy & Exploration*, vol. 40, no. 2, pp. 599–616, 2023, doi: 10.1007/s42461-023-00738-w.
- [4] E. Hofmann and M. Rüsch, “Industry 4.0 and the current status as well as future prospects on logistics,” *Computers in Industry*, vol. 89, pp. 23–34, 2017, doi: 10.1016/j.compind.2017.04.002.
- [5] F. Chromjakova, “Process stabilisation - key assumption for implementation of Industry 4.0 concept in industrial company,” *Journal of Systems Integration*, vol. 8, no. 1, pp. 3–12, 2017, doi: 10.20470/jsi.v8i1.295.
- [6] S. Aheleroff, X. Xu, R. Y. Zhong, and Y. Lu, “Digital Twin as a service in Industry 4.0: an architecture reference model,” *Advanced Engineering Informatics*, vol. 47, Jan. 2021, doi: 10.1016/j.aei.2020.101225.
- [7] Q. Qi and F. Tao, “Digital Twin and big data towards smart manufacturing and Industry 4.0: 360 degree comparison,” *IEEE Access*, vol. 6, pp. 3585–3593, 2018, doi: 10.1109/ACCESS.2018.2793265.
- [8] J. Prinsloo, J. C. Vosloo, and E. H. Mathews, “Towards Industry 4.0: a roadmap for the South African heavy industry sector,” *South African Journal of Industrial Engineering*, vol. 30, no. 3, pp. 174–186, 2019, doi: 10.7166/30-3-2237.
- [9] S. Mihai, M. Yaqoob, D. V. Hung, W. Davis, P. Towakel, M. Raza, M. Karamanoglu, B. Barn, D. Shetve, R. V. Prasad, H. Venkataraman, R. Trestian, and H. X. Nguyen, “Digital Twins: a survey on enabling technologies, challenges, trends and future prospects,” *IEEE Communications Surveys & Tutorials*, vol. 24, no. 4, pp. 2255–2291, 2022, doi: 10.1109/COMST.2022.3208773.
- [10] L. Chen, D. Cao, Y. Ai, J. Wang, B. Tian, L. Li, G. Wang, and F.-Y. Wang, “Parallel mining operating systems: from Digital Twins to mining intelligence,” *Journal of Command and Control*, vol. 1, no. 1, pp. 42–46, 2022, doi: 10.1109/DTPI52967.2021.9540195.
- [11] C. T. Farrelly and J. Davies, “Interoperability, integration, and Digital Twins for mining - part 2: pathways to the network-centric mine,” *IEEE Industrial Electronics Magazine*, vol. 15, no. 3, pp. 22–31, 2021, doi: 10.1109/MIE.2020.3029388.

- [12] M. Lombardi, F. Pascale, and D. Santaniello, “Internet of Things: a general overview between architectures, protocols and applications,” *Information*, vol. 12, p. 87, Feb. 2021, doi: 10.3390/info12020087.
- [13] U. Satija, B. Ramkumar, and M. Sabarimalai Manikandan, “Real-time signal quality-aware ECG telemetry system for IoT-based health care monitoring,” *IEEE Internet of Things Journal*, vol. 4, no. 3, pp. 815–823, 2017, doi: 10.1109/JIOT.2017.2670022.
- [14] D. G. Pivoto, L. F. de Almeida, R. da Rosa Righi, J. J. Rodrigues, A. B. Lugli, and A. M. Alberti, “Cyber-Physical Systems architectures for industrial Internet of Things applications in Industry 4.0: a literature review,” *Journal of Manufacturing Systems*, vol. 58, pp. 176–192, 2021, doi: 10.1016/j.jmsy.2020.11.017.
- [15] U. Bodkhe, D. Mehta, S. Tanwar, P. Bhattacharya, P. K. Singh, and W.-C. Hong, “A survey on decentralised consensus mechanisms for Cyber-Physical Systems,” *IEEE Access*, vol. 8, pp. 54 371–54 401, 2020, doi: 10.1109/ACCESS.2020.2981415.
- [16] R. Stark, C. Freseman, and K. Lindow, “Development and operation of Digital Twins for technical systems and services,” *CIRP Annals*, vol. 68, no. 1, pp. 129–132, 2019, doi: 10.1016/j.cirp.2019.04.024.
- [17] J. Plavšić and I. Mišković, “Industrial applications of Digital Twin technology in the mining sector: an overview,” *CIM Journal*, vol. 14, no. 2, pp. 97–106, 2023, doi: 10.1080/19236026.2022.2145431.
- [18] P. Radanliev, D. De Roure, R. Nicolescu, M. Huth, and O. Santos, “Digital Twins: Artificial Intelligence and the IoT Cyber-Physical Systems in Industry 4.0,” *International Journal of Intelligent Robotics and Applications*, vol. 6, no. 1, pp. 171–185, 2022, doi: 10.1007/s41315-021-00180-5.
- [19] M. Ahmad, S. Sadiq, A. S. Alluhaidan, M. Umer, S. Ullah, M. Nappi *et al.*, “Industry 4.0 technologies and their applications in fighting COVID-19 pandemic using deep learning techniques,” *Computers in Biology and Medicine*, vol. 145, p. 105418, 2022, doi: 10.1016/j.combiomed.2022.105418.
- [20] K. T. Park, J. Yang, and S. D. Noh, “VREDI: virtual representation for a Digital Twin application in a work-center-level asset administration shell,” *Journal of Intelligent Manufacturing*, vol. 32, no. 2, pp. 501–544, 2021, doi: 10.1007/s10845-020-01586-x.
- [21] L. Ortega, J. Jurado, J. Ruiz, and F. Feito, “Topological data models for virtual management of hidden facilities through digital reality,” *IEEE Access*, vol. 8, pp. 62 584–62 600, 2020, doi: 10.1109/ACCESS.2020.2984035.
- [22] F. Jiang, L. Ma, T. Broyd, and K. Chen, “Digital Twin and its implementations in the civil engineering sector,” *Automation in Construction*, vol. 130, p. 103838, 2021, doi: 10.1016/j.autcon.2021.103838.
- [23] P. Zhao, J. Liu, X. Jing, M. Tang, S. Sheng, H. Zhou, and X. Liu, “The modeling and using strategy for the Digital Twin in process planning,” *IEEE Access*, vol. 8, pp. 41 229–41 245, 2020, doi: 10.1109/ACCESS.2020.2974241.

- [24] W. Zeng, X.-J. Chang, and J. Lv, “Design of data model for urban transport GIS,” *Journal of Geographic Information System*, vol. 2, pp. 106–112, 2010, doi: 10.4236/jgis.2010.22016.
- [25] M. G. Juarez, V. J. Botti, and A. S. Giret, “Digital Twins: review and challenges,” *Journal of Computing and Information Science in Engineering*, vol. 21, no. 3, p. 030802, Apr. 2021, doi: 10.1115/1.4050244.
- [26] M. Attaran and B. G. Celik, “Digital Twin: benefits, use cases, challenges, and opportunities,” *Decision Analytics Journal*, vol. 6, p. 100165, 2023, doi: 10.1016/j.dajour.2023.100165.
- [27] J. Verster, P. Roux, F. Magweregwe, W. de Ronde, G. Crafford, M. Mashaba, S. Turundu, M. Mpofu, J. Prinsloo, P. Ferreira, and H. Brodner, “A Digital Twin framework to support vehicle interaction risk management in the mining industry,” *MATEC Web Conference*, vol. 388, p. 11002, 2023, doi: 10.1051/mateconf/202338811002.
- [28] F. J. Naude, “The utilisation of a Digital Twin in the asset management of continuous miners,” Master’s thesis, University of Johannesburg, South Africa, 2023, available: <https://hdl.handle.net/10210/504638>.
- [29] N. Mthimkhulu, “Digital transformation opportunities and challenges in the South African mining sector,” Master’s thesis, University of Witwatersrand, South Africa, 2023, available: <https://hdl.handle.net/10539/39878>.
- [30] R. Lu and I. Brilakis, “Digital Twinning of existing reinforced concrete bridges from labelled point clusters,” *Automation in Construction*, vol. 105, 2019, doi: 10.1016/j.autcon.2019.102837.
- [31] Q. Lu, L. Chen, S. Li, and M. Pitt, “Semi-automatic geometric Digital Twinning for existing buildings based on images and CAD drawings,” *Automation in Construction*, vol. 115, p. 103183, 2020, doi: 10.1016/j.autcon.2020.103183.
- [32] D. Conde, J. Balado, M. Soilán, J. Martínez, and P. Arias, “LiDAR data processing for digitisation of the Castro of Santa Trega and integration in Unreal Engine 5,” *International Journal of Architectural Heritage*, pp. 1–21, 2023, doi: 10.1080/15583058.2023.2271877.
- [33] J. Baek, J. Park, S. Cho, and C. Lee, “3D global localisation in the underground mine environment using mobile LiDAR mapping and point cloud registration,” *Sensors*, vol. 22, pp. 1–23, Apr. 2022, doi: 10.3390/s22082873.
- [34] D. Jacobs, J. van Laar, and C. Schutte, “Strategy to identify and mitigate hazards in deep-level mine ventilation systems using a calibrated Digital Twin,” *South African Journal of Industrial Engineering*, vol. 33, no. 3, pp. 204–217, 2022, doi: 10.7166/33-3-2795.
- [35] J. Boehm and K. Liu, “NoSQL for storage and retrieval of large LiDAR data collections,” *The International Archives of the Photogrammetry, Remote Sensing and*

- Spatial Information Sciences*, vol. 40, pp. 577–582, 2015, doi: 10.5194/isprsarchives-XL-3-W3-577-2015.
- [36] D. Deibe, M. Amor, and R. Doallo, “Big data storage technologies: a case study for web-based LiDAR visualisation,” in *2018 IEEE International Conference on Big Data*, 2018, pp. 3831–3840, doi: 10.1109/BigData.2018.8622589.
- [37] C. N. Lokugam Hewage, D. F. Laefer, A.-V. Vo, N.-A. Le-Khac, and M. Bertolotto, “Scalability and performance of LiDAR point cloud data management systems: a state-of-the-art review,” *Remote Sensing*, vol. 14, no. 20, 2022, doi: 10.3390/rs14205277.
- [38] J. A. Béjar-Martos, A. J. Rueda-Ruiz, C. J. Ogayar-Anguita, R. J. Segura-Sánchez, and A. López-Ruiz, “Strategies for the storage of large LiDAR datasets—a performance comparison,” *Remote Sensing*, vol. 14, no. 11, 2022, doi: 10.3390/rs14112623.
- [39] Y. Wang, M. Mikawa, and M. Fujisawa, “FCH-SLAM: a SLAM method for dynamic environments using semantic segmentation,” *2nd International Conference on Image Processing and Robotics*, pp. 1–6, Mar. 2022, doi: 10.1109/ICIPRob54042.2022.9798717.
- [40] D. Kiss-Illés, C. Barrado, and E. Salami, “GPS-SLAM: an augmentation of the ORB-SLAM algorithm,” *Sensors*, vol. 19, no. 22, 2019, doi: 10.3390/s19224973.
- [41] R. Lin, J. Xu, and J. Zhang, “GLO-SLAM: a SLAM system optimally combining GPS and LiDAR odometry,” *Industrial Robot: the International Journal of Robotics Research and Application*, vol. 48, no. 5, pp. 726–736, June 2021, doi: 10.1108/IR-12-2020-0272.
- [42] M. Giulia, E. Christian, P. David A., I. Yuta, and K. Gudrun, “EnvSLAM: combining SLAM systems and neural networks to improve the environment fusion in AR applications,” *ISPRS International Journal of Geo-Information*, vol. 10, no. 772, p. 772, 2021, doi: 10.3390/ijgi10110772.
- [43] H. Ikeda, O. Kolade, M. A. Mahboob, F. T. Cawood, and Y. Kawamura, “Communication of sensor data in underground mining environments: an evaluation of wireless signal quality over distance,” *Mining*, vol. 1, no. 2, pp. 211–223, 2021, doi: 10.3390/mining1020014.
- [44] T. Gaokgorwe, “The application of robotics in deep level mining in South Africa and its influence on health and safety,” Master’s thesis, University of Witwatersrand, South Africa, 2023, available: <https://hdl.handle.net/10539/38954>.
- [45] T. Zvarivadza, M. Onifade, O. Dayo-Olupona, K. O. Said, J. M. Githiria, B. Genc, and T. Celik, “On the impact of Industrial Internet of Things - mining sector perspectives,” *International Journal of Mining, Reclamation and Environment*, pp. 1–39, 2024, doi: 10.1080/17480930.2024.2347131.
- [46] C. Zhou, M. Reyes, and M. Girman, “Electromagnetic interference In underground coal mines: a literature review and practical considerations,” *Mining, Metallurgy & Exploration*, vol. 39, no. 2, pp. 421–431, 2022, doi: 10.1007/s42461-021-00535-3.
- [47] A. Goel, A. Ul Islam, A. Ansari, O. Kouba, and D. S. Bernstein, “An introduction to Inertial Navigation from the perspective of state estimation,” *IEEE Control Systems*

- Magazine*, vol. 41, no. 5, pp. 104–128, 2021, doi: 10.1109/MCS.2021.3092855.
- [48] R. V. Garcia, H. K. Kuga, and M. C. F. Zanardi, “Unscented Kalman Filter for determination of spacecraft attitude using different attitude parameterisations and real data,” *Journal of Aerospace Technology and Management*, vol. 8, no. 1, pp. 82–90, 2016, doi: 10.5028/jatm.v8i1.509.
- [49] M. Kok, J. D. Hol, and T. B. Schön, “Using inertial sensors for position and orientation estimation,” *Foundations and Trends in Signal Processing*, vol. 11, no. 1–2, p. 1–153, 2017, doi: 10.1561/20000000094.
- [50] N. A. Abiad, Y. Kone, V. Renaudin, and T. Robert, “Smartstep: a robust step detection method based on smartphone inertial signals driven by gait learning,” *IEEE Sensors Journal*, vol. 22, no. 12, pp. 12 288–12 297, 2022, doi: 10.1109/JSEN.2022.3169621.
- [51] H. Yan, Q. Shan, and Y. Furukawa, “RIDI: Robust IMU Double Integration,” in *Computer Vision (ECCV)*. Munich, Germany: Springer International Publishing, 2017, pp. 641–656, doi: 10.48550/arXiv.1712.09004.
- [52] N. El-Sheimy, H. Hou, and X. Niu, “Analysis and modeling of inertial sensors using Allan Variance,” *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 1, pp. 140–149, 2008, doi: 10.1109/TIM.2007.908635.
- [53] P. Ren, F. Elyasi, and R. Manduchi, “Smartphone-based Inertial Odometry for blind walkers,” *Sensors*, vol. 21, no. 12, p. 4033, June 2021, doi: 10.3390/s21124033.
- [54] G. Marchesi, C. Eichhorn, D. A. Plecher, Y. Itoh, and G. Klinker, “EnvSLAM: Combining SLAM systems and neural networks to improve the environment fusion in AR applications,” *ISPRS International Journal of Geo-Information*, vol. 10, no. 11, 2021, doi: 10.3390/ijgi10110772.
- [55] Z. Wang, H. Chen, S. Zhang, and Y. Lou, “Active view planning for visual SLAM in outdoor environments vased on continuous information modeling,” *IEEE/ASME Transactions on Mechatronics*, vol. 29, no. 1, pp. 237–248, 2024, doi: 10.1109/TMECH.2023.3272910.
- [56] Y. Wei and B. Akinici, “A vision and learning-based indoor localisation and semantic mapping framework for facility operations and management,” *Automation in Construction*, vol. 107, p. 102915, 2019, doi: 10.1016/j.autcon.2019.102915.
- [57] R. Mur-Artal and J. D. Tardós, “ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras,” *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017, doi: 10.1109/TRO.2017.2705103.
- [58] N. El Bazi, M. Mabrouki, O. Laayati, N. Ouhabi, H. El Hadraoui, F.-E. Hammouch, and A. Chebak, “Generic multi-layered Digital-Twin-framework-enabled asset lifecycle management for the sustainable mining industry,” *Sustainability*, vol. 15, no. 4, 2023, doi: 10.3390/su15043470.
- [59] X. Liang, J. Wu, and K. Ruan, “Simulation modelling and temperature over-advance perception of mine hoist system based on Digital Twin technology,” *Machines*, vol. 11,

- p. 966, Oct. 2023, doi: 10.3390/machines11100966.
- [60] J. Xie, X. Wang, Z. Yang, and S. Hao, “Virtual monitoring method for hydraulic supports based on Digital Twin theory,” *Mining Technology*, vol. 128, no. 2, pp. 77–87, 2019, doi: 10.1080/25726668.2019.1569367.
 - [61] X.-X. Gao, G.-J. Chai, and G.-A. Zhu, “Layout of the first-mining working face in the protected seam for pressure relief by peak stress shift,” *Shock and Vibration*, vol. 2021, pp. 1–11, 2021, doi: 10.1155/2021/5555584.
 - [62] A. Verma and D. Deb, “Numerical analysis of an interaction between hydraulic-powered support and surrounding rock strata,” *International Journal of Geomechanics*, vol. 13, no. 2, pp. 181–192, 2013, doi: 10.1061/(ASCE)GM.1943-5622.0000190.
 - [63] D. Chen and S. Li, “Measurement of coal mining height based on hydraulic support structural angle,” *Journal of the China Coal Society*, vol. 41, no. 3, pp. 788–793, Mar. 2016, doi: 10.13225/j.cnki.jccs.2015.0685.
 - [64] H. Wen, S. Liu, X. Zheng, G. Cai, B. Zhou, W. Ding, and Y. Ma, “The Digital Twins for mine site rescue environment: application framework and key technologies,” *Process Safety and Environmental Protection*, vol. 186, pp. 176–188, 2024, doi: 10.1016/j.psep.2024.04.007.
 - [65] J. Wang, Y. Huang, W. Zhai, J. Li, S. Ouyang, H. Gao, Y. Liu, and G. Wang, “Research on coal mine safety management based on Digital Twin,” *Heliyon*, vol. 9, no. 3, 2023, doi: 10.1016/j.heliyon.2023.e13608.
 - [66] A. Kychkin and A. Nikolaev, “IoT-based mine ventilation control system architecture with Digital Twin,” *International Conference on Industrial Engineering, Applications and Manufacturing*, pp. 1–5, 2020, doi: 10.1109/ICIEAM48468.2020.9111995.
 - [67] Y. A. El-Wajeh, P. V. Hatton, and N. J. Lee, “Unreal Engine 5 and immersive surgical training: translating advances in gaming technology into extended-reality surgical simulation training programmes,” *British Journal of Surgery*, vol. 109, no. 5, pp. 470–471, 2022, doi: 10.1093/bjs/znac015.
 - [68] M. Antico, N. Balletti, G. Laudato, A. Lazich, M. Notarantonio, R. Oliveto, S. Ricciardi, S. Scalabrino, and J. Simeone, “Postural control assessment via Microsoft Azure Kinect DK: an evaluation study,” *Computer Methods and Programs in Biomedicine*, vol. 209, p. 106324, 2021, doi: 10.1016/j.cmpb.2021.106324.
 - [69] F. Xue, W. Lu, Z. Chen, and C. J. Webster, “From LiDAR point cloud towards Digital Twin city: clustering city objects based on Gestalt principles,” *Journal of Photogrammetry and Remote Sensing*, vol. 167, pp. 418–431, 2020, doi: 10.1016/j.isprsjprs.2020.07.020.
 - [70] H.-H. Benzon, X. Chen, L. Belcher, O. Castro, K. Branner, and J. Smit, “An operational image-based Digital Twin for large-scale structures,” *Applied Sciences*, vol. 12, no. 7, 2022, doi: 10.3390/app12073216.

- [71] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: common objects in context,” in *Computer Vision (ECCV)*. Zurich, Switzerland: Springer, Sept. 2014, pp. 740–755, doi: 10.48550/arXiv.1405.0312.
- [72] M. Minos-Stensrud, O. H. Haakstad, O. Sakseid, B. Westby, and A. Alcocer, “Towards automated 3D reconstruction in SME factories and Digital Twin model generation,” in *18th International Conference on Control, Automation and Systems (ICCAS)*, Pyeongchang, South Korea, Oct. 2018, pp. 1777–1781.
- [73] M. Labbe and F. Michaud, “Appearance-based loop closure detection for online large-scale and long-term operation,” *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 734–745, 2013, doi: 10.1109/TRO.2013.2242375.
- [74] S. Sierla, M. Azangoo, A. Fay, V. Vyatkin, and N. Papakonstantinou, “Integrating 2D and 3D digital plant information towards automatic generation of Digital Twins,” in *IEEE 29th International Symposium on Industrial Electronics (ISIE)*, Delft, Netherlands, June 2020, pp. 460–467, doi: 10.1109/ISIE45063.2020.9152371.
- [75] M. Mohammadi, M. Rashidi, V. Mousavi, A. Karami, Y. Yu, and B. Samali, “Quality evaluation of Digital Twins generated based on UAV photogrammetry and TLS: bridge case study,” *Remote Sensing*, vol. 13, no. 17, p. 3499, 2021, doi: 10.3390/rs13173499.
- [76] S. Feroz and S. Abu Dabous, “UAV-based remote sensing applications for bridge condition assessment,” *Remote Sensing*, vol. 13, no. 9, p. 1809, 2021, doi: 10.3390/rs13091809.
- [77] M. Rashidi, M. Mohammadi, S. Sadeghlou Kivi, M. M. Abdolvand, L. Truong-Hong, and B. Samali, “A decade of modern bridge monitoring using terrestrial laser scanning: review and future directions,” *Remote Sensing*, vol. 12, no. 22, p. 3796, 2020, doi: 10.3390/rs12223796.
- [78] A. Poulou and D. Han, “Hybrid indoor localisation using IMU sensors and smartphone camera,” *Sensors*, vol. 19, no. 23, 2019, doi: 10.3390/s19235084.
- [79] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Journal of Basic Engineering*, no. 3, pp. 35–45, Mar. 1960, doi: 10.1115/1.3662552.
- [80] T. Hu, X. Sun, Y. Su, H. Guan, Q. Sun, M. Kelly, and Q. Guo, “Development and performance evaluation of a very low-cost UAV-LiDAR system for forestry applications,” *Remote Sensing*, vol. 13, p. 77, Dec. 2020, doi: 10.3390/rs13010077.
- [81] L. Wallace, A. Lucieer, C. Watson, and D. Turner, “Development of a UAV-LiDAR system with application to forest inventory,” *Remote sensing*, vol. 4, no. 6, pp. 1519–1543, 2012, doi: 10.3390/rs4061519.
- [82] M. M. Bustamante, I. Roitman, T. M. Aide, A. Alencar, L. O. Anderson, L. Aragão, G. P. Asner, J. Barlow, E. Berenguer, J. Chambers *et al.*, “Toward an integrated monitoring framework to assess the effects of tropical forest degradation and recovery on carbon stocks and biodiversity,” *Global change biology*, vol. 22, no. 1, pp. 92–109, 2016, doi: 10.1111/gcb.13087.

- [83] S. Park, J. Park, and C. G. Park, “Adaptive attitude estimation for low-cost MEMS IMU using ellipsoidal method,” *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 9, pp. 7082–7091, 2020, doi: 10.1109/TIM.2020.2974135.
- [84] S. B. Farahan, J. J. M. Machado, F. G. de Almeida, and J. M. R. S. Tavares, “9-DOF IMU-based attitude and heading estimation using an Extended Kalman Filter with bias consideration,” *Sensors*, vol. 22, no. 9, Apr. 2022, doi: 10.3390/s22093416.
- [85] E. W. Grafarend and J. Awange, *Applications of linear and nonlinear models*. Switzerland: Springer, 2022, doi: 10.1007/978-3-030-94598-5.
- [86] I. U. Hassan, K. Panduru, and J. Walsh, “Review of data processing methods used in predictive maintenance for next generation heavy machinery,” *Data*, vol. 9, no. 5, p. 69, 2024, doi: 10.3390/data9050069.
- [87] M. Brossard, A. Barrau, and S. Bonnabel, “AI-IMU dead-reckoning,” *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 4, pp. 585–595, Mar. 2020, doi: 10.1109/TIV.2020.2980758.
- [88] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: the KITTI dataset,” *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, Aug. 2013, doi: 10.1177/0278364913491297.
- [89] B. Fan, Q. Li, C. Wang, and T. Liu, “An adaptive orientation estimation method for magnetic and inertial sensors in the presence of magnetic disturbances,” *Sensors*, vol. 17, p. 1161, May 2017, doi: 10.3390/s17051161.
- [90] J. Diebel *et al.*, “Representing attitude: Euler angles, unit quaternions, and rotation vectors,” *Matrix*, vol. 58, no. 15-16, pp. 1–35, 2006.
- [91] H. Kim and Y. Choi, “Location estimation of autonomous driving robot and 3D tunnel mapping in underground mines using pattern matched LiDAR sequential images,” *International Journal of Mining Science and Technology*, vol. 31, no. 5, pp. 779–788, 2021, doi: 10.1016/j.ijmst.2021.07.007.
- [92] I. Brigadnov, A. Lutonin, and K. Bogdanova, “Error state extended Kalman Filter localisation for underground mining environments,” *Symmetry*, vol. 15, no. 2, 2023, doi: 10.3390/sym15020344.
- [93] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng *et al.*, “ROS: an open-source robot operating system,” in *IEEE ICRA workshop on open source software*, vol. 3. Kobe, 2009, p. 5.
- [94] N. Navidi and R. Landry, “A new perspective on low-cost MEMS-based AHRS determination,” *Sensors*, vol. 21, no. 4, 2021, doi: 10.3390/s21041383.
- [95] S. Herath, H. Yan, and Y. Furukawa, “RoNIN: Robust Neural Inertial Navigation in the wild: benchmark, evaluations, & new methods,” *IEEE International Conference on Robotics and Automation*, pp. 3146–3152, 2020, doi: 10.1109/ICRA40945.2020.9196860.
- [96] C. Chen, P. Zhao, C. Xiaoxuan Lu, W. Wang, A. Markham, and N. Trigoni, “OxIOD: the dataset for deep inertial odometry,” *arXiv*, Sept. 2018, doi: 10.48550/arXiv.1809.07491.

- [97] T. Feigl, S. Kram, P. Woller, R. H. Siddiqui, M. Philippsen, and C. Mutschler, “RNN-aided human velocity estimation from a single IMU,” *Sensors*, vol. 20, no. 13, 2020, doi: 10.3390/s20133656.
- [98] B. Yang, Y. Ma, X. Liang, C. Tian, and D. Su, “An adaptive threshold method with error correction for pedestrian inertial navigation system,” *Measurement Science and Technology*, vol. 34, no. 7, pp. 75 109–75 120, 2023, doi: 10.1088/1361-6501/acc5ff.
- [99] J. Kuang, D. Xia, Y. Wang, X. Meng, and X. Niu, “A shin-mounted inertial navigation system for pedestrian walking and running gait,” *IEEE Sensors Journal*, vol. 25, no. 6, pp. 9449–9458, 2025, doi: 10.1109/JSEN.2025.3533138.
- [100] C. Manettas, N. Nikolakis, and K. Alexopoulos, “Synthetic datasets for deep learning in computer-vision assisted tasks in manufacturing,” *Procedia CIRP*, vol. 103, pp. 237–242, Oct. 2021, doi: 10.1016/j.procir.2021.10.038.
- [101] Y. Wang, U. Ahsan, H. Li, and M. Hagen, “A comprehensive review of modern object segmentation approaches,” *Foundations and Trends in Computer Graphics and Vision*, vol. 13, no. 2–3, pp. 111–283, 2022, doi: 10.1561/06000000097.
- [102] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, “Segment Anything,” *IEEE International Conference on Computer Vision*, 2023, doi: 10.1109/ICCV51070.2023.00371.
- [103] T. Golovina, A. Polyanin, A. Adamenko, E. Khegay, and V. Schepinin, “Digital Twins as a new paradigm of an industrial enterprise,” *International Journal of Technology*, vol. 11, no. 6, pp. 1115–1124, 2020, doi: 10.14716/ijtech.v11i6.4427.
- [104] J. W. Tianyuan Liu, Jinsong Bao and J. Wang, “Deep learning for industrial image: challenges, methods for enriching the sample space and restricting the hypothesis space, and possible issue,” *International Journal of Computer Integrated Manufacturing*, vol. 35, no. 10-11, pp. 1077–1106, 2022, doi: 10.1080/0951192X.2021.1901319.
- [105] K. Greff, F. Belletti, L. Beyer, C. Doersch, Y. Du, D. Duckworth, D. J. Fleet, D. Gnanapragasam, F. Golemo, C. Herrmann, T. Kipf, A. Kundu, D. Lagun, I. Laradji, H.-T. D. Liu, H. Meyer, Y. Miao, D. Nowrouzezahrai, C. Oztireli, E. Pot, N. Radwan, D. Rebain, S. Sabour, M. S. M. Sajjadi, M. Sela, V. Sitzmann, A. Stone, D. Sun, S. Vora, Z. Wang, T. Wu, K. M. Yi, F. Zhong, and A. Tagliasacchi, “Kubric: a scalable dataset generator,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 3749–3761, doi: 10.48550/arXiv.2203.03570.
- [106] J. G. Lee, J. Hwang, S. Chi, and J. Seo, “Synthetic image dataset development for vision-based construction equipment detection,” *Journal of Computing in Civil Engineering*, vol. 36, no. 5, 2022, doi: 10.1061/(ASCE)CP.1943-5487.0001035.
- [107] Z. Wu, L. Zhao, and H. Zhang, “MR-UNet commodity semantic segmentation based on transfer learning,” *IEEE Access*, vol. 9, pp. 159 447–159 456, Nov. 2021, doi: 10.1109/ACCESS.2021.3130578.

- [108] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, “Image segmentation using deep learning: a survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3523–3542, 2022, doi: 10.1109/TPAMI.2021.3059968.
- [109] A. Hosna, E. Merry, J. Gyalmo, Z. Alom, Z. Aung, and M. A. Azim, “Transfer learning: a friendly introduction,” *Journal of Big Data*, vol. 9, no. 1, p. 102, 2022, doi: 10.1186/s40537-022-00652-w.
- [110] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, “ImageNet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115, pp. 211–252, 2015, doi: 10.1007/s11263-015-0816-y.
- [111] L. Soni, A. Kaur, and A. Sharma, “A review on different versions and interfaces of Blender software,” in *7th International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India, 2023, pp. 882–887, doi: 10.1109/ICOEI56765.2023.10125672.
- [112] M. Wang, Z. Zhang, and H. Liu, “An adaptive semantic segmentation network for adversarial learning domain based on low-light enhancement and decoupled generation,” *Applied Sciences*, vol. 14, no. 8, 2024, doi: 10.3390/app14083295.
- [113] M. J. Martell and J. J. Sammarco, “Luminance measurement for underground mine lighting,” in *IEEE Industry Applications Society Annual Meeting*, Cincinnati, OH, USA, 2017, pp. 1–9, doi: 10.1109/IAS.2017.8101836.
- [114] G. Nagel, “The influence of the physical underground workplace conditions on the production at a deep level gold mine,” Ph.D. dissertation, North-West University, South Africa, 2020.
- [115] C. Chen, C. Lu, J. Wahlstrom, A. Markham, and N. Trigoni, “Deep neural network based Inertial Odometry using low-cost Inertial Measurement Units,” *IEEE Transactions on Mobile Computing*, vol. 20, no. 4, pp. 1351–1364, 2021, doi: 10.1109/TMC.2019.2960780.
- [116] B. Chen, R. Zhang, S. Wang, L. Zhang, and Y. Liu, “Deep-learning-based Inertial Odometry for pedestrian tracking using attention mechanism and Res2Net module,” *IEEE Sensors Letters*, vol. 6, no. 11, pp. 1–4, 2022, doi: 10.1109/LSSENS.2022.3212472.
- [117] D. L. Morgan, “Paradigms lost and pragmatism regained: Methodological implications of combining qualitative and quantitative methods,” *Journal of mixed methods research*, vol. 1, no. 1, pp. 48–76, 2007, doi: 10.1177/2345678906292462.
- [118] I. M. Ali, “A guide for positivist research paradigm: From philosophy to methodology,” *Ideology Journal*, vol. 9, no. 2, 2024, doi: 10.24191/idealogy.v9i2.596.
- [119] M. P. Dentch, *The ISO 9001:2015 Implementation Handbook*. Milwaukee, Wisconsin, USA: ASQ Quality Press, 2016, isbn: 978-0-8738-9938-3.
- [120] N. Denzin and Y. Lincoln, *The SAGE Handbook of Qualitative Research*. London, UK: SAGE Publications, 2017, isbn: 978-1-4833-4980-0.

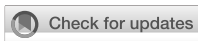
- [121] J. N. De Meyer, “Determining the integrity of single-source condition-based maintenance data,” Ph.D. dissertation, North-West University, South Africa, 2022.
- [122] E. Wan, R. Van Der Merwe, and A. Nelson, “Dual estimation and the Unscented Transformation,” *Advances in neural information processing systems*, vol. 12, 1999.
- [123] E. A. Wan and R. Van Der Merwe, “The Unscented Kalman Filter for nonlinear estimation,” in *IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium*, Lake Louise, AB, Canada, Oct. 2000, pp. 153–158, doi: 10.1109/ASSPCC.2000.882463.
- [124] S. Julier and J. Uhlmann, “Unscented filtering and nonlinear estimation,” *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, Mar. 2004, doi: 10.1109/JPROC.2003.823141.
- [125] K. György, A. Kelemen, and L. Dávid, “Unscented Kalman Filters and particle filter methods for nonlinear state estimation,” *Procedia Technology*, vol. 12, pp. 65–74, Oct. 2014, doi: 10.1016/j.protcy.2013.12.457.
- [126] P. Shrivastava, T. K. Soon, M. Y. I. B. Idris, and S. Mekhilef, “Overview of model-based online state-of-charge estimation using Kalman Filter family for lithium-ion batteries,” *Renewable and Sustainable Energy Reviews*, vol. 113, p. 109233, 2019, doi: 10.1016/j.rser.2019.06.040.
- [127] J. Farrell, *Aided navigation: GPS with high rate sensors*. McGraw-Hill, Inc., 2008, doi: 10.1036/0071493298.
- [128] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, “Panoptic Segmentation,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9404–9413, doi: 10.1109/CVPR.2019.00963.
- [129] J. Huai, Y. Zhang, and A. Yilmaz, “The mobile AR sensor logger for Android and iOS devices,” in *2019 IEEE Sensors*. IEEE, 2019, pp. 1–4, doi: 10.1109/SENSORS4301.1.2019.8956816.
- [130] K. Kronis and M. Uhanova, “Performance comparison of Java EE and ASP.NET Core technologies for Web API development,” *Applied Computer Systems*, vol. 23, no. 1, pp. 37–44, 2018, doi: 10.2478/acss-2018-0005.
- [131] G. Araghi and R. J. Landry, “Temperature compensation model of MEMS inertial sensors based on neural network,” in *IEEE/ION Position, Location and Navigation Symposium (PLANS)*, Monterey, CA, USA, Apr. 2018, pp. 301–309, doi: 10.1109/PLANS.2018.8373395.
- [132] J. F. Guerra, R. Garcia-Hernandez, M. A. Llama, and V. Santibañez, “UKF-based neural training for nonlinear systems identification and control improvement,” *IEEE Access*, vol. 10, pp. 114 501–114 513, 2022, doi: 10.1109/ACCESS.2022.3217816.
- [133] R. W. Bohannon and A. W. Andrews, “Normal walking speed: a descriptive meta-analysis,” *Physiotherapy*, vol. 97, no. 3, pp. 182–189, 2011.
- [134] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison,

- A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “PyTorch: an imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.
- [135] Y. Li, S. Xie, X. Chen, P. Dollar, K. He, and R. Girshick, “Benchmarking detection transfer learning with vision transformers,” *arXiv*, 2021, doi: 10.48550/arXiv.2111.11429.
- [136] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826, doi: 10.48550/arXiv.1512.00567.
- [137] W. J. Kilian, “A framework to implement Digital Twin functionalities into web applications,” Master’s thesis, North-West University, South Africa, 2022.

Appendices

Appendix 1: Publication

A journal article was published by the Frontiers in Artificial Intelligence journal on 13 December 2024. The published article has been appended in the following pages and is also available online at <https://doi.org/10.3389/frai.2024.1453931>.



OPEN ACCESS

EDITED BY

Alaa Eleyan,
American University of the Middle East, Kuwait

REVIEWED BY

Soydan Redif,
American University of the Middle East, Kuwait
Gulden Eleyan,
American College of the Middle East, Kuwait

*CORRESPONDENCE

Wilhelm Johannes Kilian
✉ winkilian@gmail.com

RECEIVED 25 June 2024

ACCEPTED 07 November 2024

PUBLISHED 13 December 2024

CITATION

Kilian WJ, Prinsloo J, Vosloo J and Taljaard S
(2024) A synthetic segmentation dataset
generator using a 3D modeling framework
and raycaster: a mining industry application.
Front. Artif. Intell. 7:1453931.
doi: 10.3389/frai.2024.1453931

COPYRIGHT

© 2024 Kilian, Prinsloo, Vosloo and Taljaard.
This is an open-access article distributed
under the terms of the [Creative Commons
Attribution License \(CC BY\)](#). The use,
distribution or reproduction in other forums is
permitted, provided the original author(s) and
the copyright owner(s) are credited and that
the original publication in this journal is cited,
in accordance with accepted academic
practice. No use, distribution or reproduction
is permitted which does not comply with
these terms.

A synthetic segmentation dataset generator using a 3D modeling framework and raycaster: a mining industry application

Wilhelm Johannes Kilian*, Jaco Prinsloo, Jan Vosloo and
Stéphan Taljaard

Faculty of Engineering, North-West University, Potchefstroom, South Africa

Many industries utilize deep learning methods to increase efficiency and reduce costs. One of these methods, image segmentation, is used for object detection and recognition in localization and mapping. Segmentation models are trained using labeled datasets; however, manually creating datasets for every application, including deep-level mining, is time-consuming and typically expensive. Recently, many papers have shown that using synthetic datasets (digital recreations of real-world scenes) for training produces highly-accurate segmentation models. This paper proposes a synthetic segmentation dataset generator using a 3D modeling framework and raycaster. The generator was applied to a deep-level mining case study and produced a dataset containing labeled images of scenes typically found in this environment, therefore removing the requirement to create the dataset manually. Validation showed high accuracy segmentation after model training using the generated dataset (compared to other applications that use real-world datasets). Furthermore, the generator can be customized to produce datasets for many other applications.

KEYWORDS

deep learning, computer vision, image segmentation, deep-level mining, real applications in engineering

1 Introduction

Many economies are in various stages of Industry 4.0 adoption. The Industry 4.0 paradigm can be described as the digitization of complex industrial systems through the implementation and integration of advanced technologies (Prinsloo et al., 2019). This enables initiatives such as predictive maintenance, remote real-time monitoring and control, and process optimization, which are the value drivers for the adoption of Industry 4.0 (Stark et al., 2019).

The South African deep-level mining industry is in the process of shifting toward Industry 4.0 to utilize these value drivers (Prinsloo et al., 2019; Stark et al., 2019; Dhamija, 2022). Digital twins (DTs), one of the technologies enabling these value drivers, are digital representations of physical systems with cyber-physical connections (Aheleroff et al., 2021).

DTs enable access to sensor measurement data and system metadata within the system's three-dimensional (3D) context (Aheleroff et al., 2021). However, DTs are time-consuming to create manually and existing semi-automatic generation methods, e.g., light detection and ranging (LiDAR), which are typically expensive and require specialized equipment (Baek et al., 2022).

Localization and mapping is the process of capturing information about the surrounding area such that a digital map is created (Wei and Akinci, 2019). This technique has been used to generate DTs instead of creating them manually. However, simply using a 360° camera to capture all the tunnels in a deep-level mine is not feasible, as even a short 10-minute recording using this type of camera can result in large data storage requirements (Bidgoli et al., 2020).

In recent years, deep learning methods have shown promising results in the shift toward the Industry 4.0 paradigm (Radanliev et al., 2022; Ahmad et al., 2022; Khayyam et al., 2020). One of these deep learning methods, namely image segmentation, has been used for localization and mapping problems (Manettas et al., 2021). Image segmentation is a method used to divide images into smaller regions to locate and identify certain objects of interest (Minaee et al., 2022). Utilizing image segmentation, the required data storage size can be reduced for localization and mapping. Segmentation can identify whether an image contains valuable information, determining whether an image needs to be stored.

Deep learning image segmentation models require labeled datasets to be trained. The higher the quality of the dataset, the better the chance of high-accuracy segmentation. Dataset quality is typically determined by factors such as size, label accuracy, and variance (to avoid overfitting) (Duan et al., 2022). However, these labeled datasets are almost always created manually (Lee et al., 2022).

Manual creation of these labeled datasets is an extremely time-consuming process and, typically, expensive as many people are often employed to parallelize its creation (Lee et al., 2022). At the time of this research, no labeled dataset for deep-level mining applications could be found, and manually creating it would be inefficient in terms of time and cost. Recently, the creation and use of synthetic datasets (digital recreations of real-world scenes which are automatically labeled) for the training of deep learning models have shown promising results (Manettas et al., 2021; Greff et al., 2022).

Much research has been invested into image segmentation, as can be seen in the paper by Minaee et al., where they surveyed over 100 recent papers (Minaee et al., 2022). Most of these papers focused on applications containing bicycles, people, vehicles, fire hydrants, and other objects found in typical street scenes. Many of these papers applied the technology for object identification and localization in self-driving vehicles, which require the identification of common objects, for which many datasets already exist. It is more challenging to find datasets for uncommon scenarios, and in some cases, they do not even exist, as was the case for deep-level mines at the time of this study.

Minaee et al. (2022) also summarized the most well-known metrics used for image segmentation models, in order of proven resulting performance, such as Pixel Accuracy (PA), Intersection over Union (IoU), and the Dice coefficient (also known as the F1 score).

Pixel Accuracy (PA) is simply a measurement of the proportion of correctly- to incorrectly-labeled pixels. This rather simple metric, unfortunately, suffers from the class imbalance problem. Intersection over Union (IoU) considers the overlapped segmentation area versus the total area of the predicted and ground

truth labels. It circumvents the class imbalance issue and provides better entity distinction in instance segmentation, making it widely used in various papers (Wang et al., 2023; He et al., 2021). The Dice coefficient, or F1 score, assesses twice the overlap of predicted and ground truth labels against their total area (Minaee et al., 2022). Valued for its balanced measure and insensitivity to global differences, it does not skew based on object class frequencies or object sizes (Wu et al., 2021; Milletari et al., 2016).

Cheng et al. (2020) considered another metric, Panoptic Quality (PQ), for their segmentation model named Panoptic-Deeplab. They used the Cityscapes (Cordts et al., 2016), Mapillary Vistas (Neuhoff et al., 2017), and COCO (Lin et al., 2014) datasets for training and validation. All of these are real-world datasets of objects typically found in traffic images and other common scenarios.

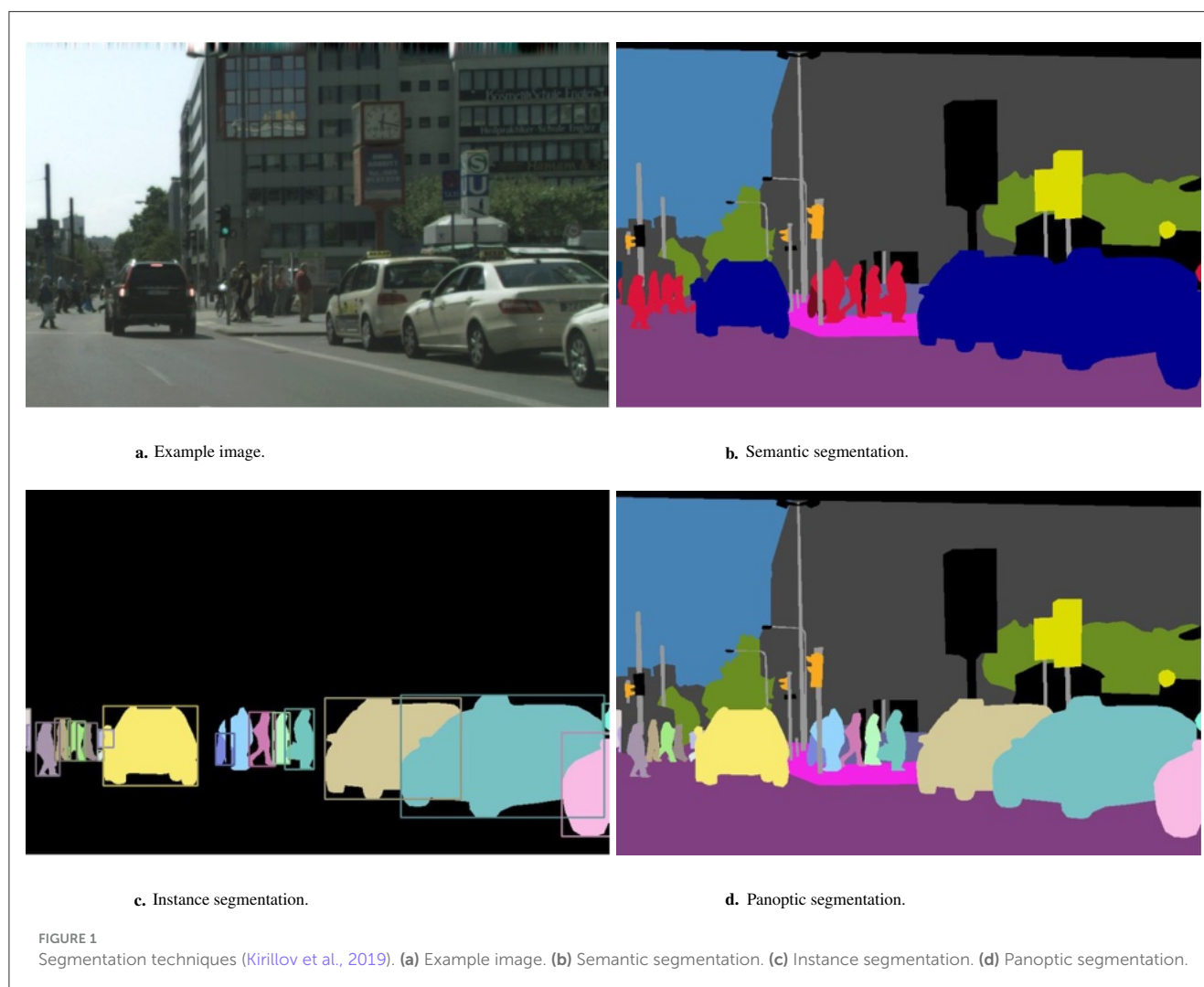
PQ, originally proposed by Kirillov et al. (2019), is the average IoU over each class, divided by the sum of the true positive and false positive matches. The class imbalance problem is avoided by averaging the IoU and giving each class equal weight, while incorrect labels are penalized by dividing by the false matches. Kirillov et al. proposed this metric to combine semantic and instance segmentation approaches to create panoptic segmentation, as shown in Figure 1. Semantic segmentation (Figure 1b) groups all pixels in an image together in classes represented by one color per class, instance segmentation (Figure 1c) identifies unique instances of objects and assigns each a different color, and panoptic segmentation (Figure 1d) is a combination of the two.

Kirillov et al. (2023) introduced the Segment Anything Model (SAM) in a more recent study. Their study aimed to create a model capable of segmenting any image. Their model has been shown to segment a wide variety of images. However, object classes not present in their dataset are only labeled as generic objects. Therefore, it is not suitable for segmenting and labeling objects of interest when they are uncommon and not present in the training dataset they used.

Zanella et al. (2021) proposed a synthetic dataset generator for the semantic segmentation of cable-like objects. They utilized domain randomization to create datasets that are generalisable across different environments. Domain randomization consists of creating images of the objects of interest with many different backgrounds. However, this generalisability has been shown to reduce segmentation performance for specific problems not well represented in the generated dataset (Choi et al., 2021). They utilized the Dice coefficient for the evaluation of their proposed generator.

Kong et al. (2020) produced a synthetic dataset for urban environments. However, their study focused on high-resolution satellite imagery for building footprint segmentation. They combined their synthetic labeled dataset with the limited real-world datasets available to train their models. This combination resulted in high-accuracy segmentation performance. Kong et al. used IoU to evaluate the performance of their segmentation model, which is a sensitive evaluation metric compared to PQ (Kirillov et al., 2019).

Greff et al. (2022) recently introduced a dataset generator using Blender (a 3D computer graphics tool), which can be used to generate synthetic scenes of many different types of real-world scenarios. Their framework, however, includes features such as depth mapping, optical flow, segmentation, and surface normals.



This causes the generator to require significant computational power to generate the synthetic dataset as it computes many other properties of the synthetic scenes.

Many papers have discussed the reality gap when using synthetic data to train deep learning models (Wang et al., 2022). The reality gap is the real-world effects that are not represented in the synthetic data because it would be so complex to model that creating the dataset becomes unfeasible. Some examples of these real-world effects are object deformation, foggy weather, and complex lighting conditions (Wu et al., 2021).

Synthetic datasets also tend to suffer from overfitting, which occurs when the dataset does not have enough variance to produce a segmentation model that performs well on unseen real-world data (Hosna et al., 2022). Recently, authors have utilized transfer learning during the training phases to overcome this reality gap.

Transfer learning is the process of training a deep learning model using real-world datasets to gain knowledge about the real domain and then fine-tuning the model to the specific application by training it in a second phase using the synthetic dataset (Manettas et al., 2021). Transfer learning has been shown to produce highly-accurate segmentation models (Wu et al., 2021; Wang et al., 2022).

In their paper, Manettas et al. (2021) utilized transfer learning in computer-vision tasks in manufacturing. They initialized their model's weights using a model pre-trained on the ImageNet dataset. The ImageNet dataset, at the time of their study, contained 1.2 million real-world labeled images. After initialization, they trained the model on their relatively small synthetic dataset (300 labeled images per class) and achieved segmentation performance comparable to models trained using a real-world dataset created specifically for the application and even outperformed some.

Wu et al. (2021) improved commodity segmentation using transfer learning. They used the weights of the ResNet model to initialize their model and then used a dataset they created manually, consisting of 2,800 labeled commodity images, to fine-tune for their application. A significant improvement was found when they compared the segmentation accuracy of the model, which was trained using transfer learning, to one trained without transfer learning. Wu et al. (2021) utilized IoU and the Dice coefficient to analyze performance.

For this paper, instance segmentation will be used, as only the predefined classes of interest are regarded as regions of interest. The IoU and PQ metrics will be used during the validation of the dataset generator proposed in the following section. The IoU metric was

chosen as it circumvents the class imbalance problem prevalent in metrics such as PA (Wang et al., 2023). The Dice coefficient was not chosen, as it is very similar to the IoU metric. The PQ metric was chosen as it improves upon the IoU metric by considering the true positive and false positive matches (Kirillov et al., 2019). However, both will still be used for comparison purposes during the validation process.

Digital twins, a technology enabling the value drivers of Industry 4.0, are time-consuming to create manually, and existing semi-automatic generation methods are typically expensive and require specialized equipment (Baek et al., 2022). Localization and mapping applications can be used to generate DTs. One of the methods used in localization and mapping is image segmentation. However, the manual creation of the datasets used for the training of image segmentation deep learning models is time-consuming and typically expensive (Lee et al., 2022).

Existing papers have explored the use of synthetic datasets for image segmentation. However, none of them focused on the cost-efficient automatic generation of instance segmentation datasets (Greff et al., 2022; Manettas et al., 2021; Zanella et al., 2021). Therefore, there is a need for a method to automatically generate synthetic, labeled datasets in a cost-efficient manner for many different applications. This paper proposes a synthetic dataset generator using a 3D-modeling framework and raycaster. A South African deep-level mining case study is then used to verify the proposed generator.

2 Synthetic segmentation dataset generator

This section is split into three parts. First, the synthetic scene creation process will be proposed. Then, the automatic labeling process using a raycaster will be proposed. Finally, the strategy that will be used to validate the generator will be discussed.

2.1 Synthetic scene generation

2.1.1 Elements in a scene

A real-world scene and the elements (objects and phenomena) within it must be analyzed before the scene can be synthetically recreated. It is unfeasible in most applications to precisely model every aspect of a real-world scene due to the reality gap (Wang et al., 2022). Therefore, elements of interest must be chosen to synthesize the real-world scene accurately enough for the relevant application (Greff et al., 2022).

The elements of interest and the required accuracy with which real-world elements must be modeled depend on the application and constraints. The object classes which should be recognized in the trained segmentation model are the highest priority. Other fundamental parts of the real-world scene, such as the walls and floors in a corridor when considering an indoor environment, should also be included as elements of interest.

The object classes (the first category of elements previously mentioned) must be defined before creating the segmentation dataset. Some examples of object classes are ladders, vehicles, windows, and doors. More detailed object classes may be chosen,

such as different vehicle types or door shapes. The level of detail for the chosen classes must be based on the specific application.

The number of chosen object classes should be restricted. Segmentation performance has been shown to degrade when the number of object classes to identify passes a certain threshold (Wang et al., 2022). This threshold is determined by the available computational resources and training time available.

Phenomena (the second category of elements) should also be analyzed, such as object interaction and ambient lighting in real-world scenes. This can be done by posing questions such as (1) is the scene well-lit, (2) are there objects moving throughout the scene, and (3) is it expected that solid objects cross each other's boundaries?

An answer to the last question could be that solid objects solely occupy their own space, but there are situations where, for example, a pipe has been placed such that it passes through a wall from one room to another. Another phenomenon to consider is the perspective of one moving through the real-world scenes, as the recreation of it should replicate the real scenario as accurately as possible.

These are only a few examples of phenomena that should be analyzed as they introduce some of the nuances of the real-world scene into the synthetic scene, therefore increasing the possibility of high-performance segmentation. Phenomena that do not significantly increase the similarity of the synthetic scene to the real-world scene should be ignored. The generation of the synthetic scenes can begin once the analysis of the real-world scenes has been completed.

2.1.2 3D modeling framework

Many open-source 3D models exist for various object classes (pipes, compressors, doors, stairways, etc.). These 3D models can be utilized to digitally represent small parts of real-world scenes, which removes the requirement to create them manually and may increase modeling accuracy (if the open-source model is a more realistic representation of the real-world object in comparison to a 3D model that can be created manually in a reasonable amount of time).

These open-source models are typically used in 3D modeling frameworks and computer graphics engines to create virtual scenes, such as the visual aspect of DTs and augmented reality experiences (Lv et al., 2022). These 3D models are stored in many different data formats, such as the stereolithography (STL) and Wavefront OBJ formats. The modeling framework selection method where these 3D models will be utilized is discussed next.

A 3D modeling framework should be chosen based on the combination of outcomes from the analysis of each relevant category described in Table 1. A category is considered irrelevant when it does not significantly affect the application.

Once the modeling framework has been chosen, the synthetic scenes can be designed and generated. Creating the synthetic scenes comprises implementing the phenomena and placing the objects previously identified in real-world scenes. The full implementation method has been abstracted into the flow diagram shown in Figure 2 to be relevant for many applications and modeling frameworks.

The implementation method is based on the Plan, Do, Check, Act (PDCA) cycle, which forms part of the ISO 9001 standard

(Dentch, 2016). The first step is to select the unimplemented object or phenomenon that would change the scene most significantly. After it has been selected, the object or phenomenon is implemented in the previously chosen 3D modeling framework.

Implementing the object consists of importing the object's geometry into the virtual scene and applying a texture to represent the object's surface. The object's geometry is imported from the open-source file obtained in the chosen format, such as STL or OBJ. The choice of the texture to show on the surface of the object depends on what type of object is being loaded. However, many open-source textures (typically stored as 2D images) are available.

The loaded object is then rendered into the scene, and its scale is updated to ensure a realistic representation of the real-world

TABLE 1 Categories for analyzing 3D modeling frameworks for an application.

Category	Description
3D model imports and exports	The import and export of 3D models in various formats to increase generation productivity.
Raycasting	The outline of visible objects in each perspective should be identifiable to create the labeled dataset.
Usability and flexibility	Does the framework have comprehensive documentation, an active community, and can generate various scenes?
Advanced physics and simulation	Does the framework support object interactions that aim to mimic real-world phenomena, such as water flow?
Scalability and hardware requirements	How complex can the scene become while balancing the hardware requirements to be feasible for the application?
Licensing and cost	Is the framework free to use?

scene. Following this, the performance of the resulting scene is then analyzed to determine whether the newly implemented object or phenomenon should be simplified. Once the performance is satisfactory, the next most significant object or phenomenon should be implemented.

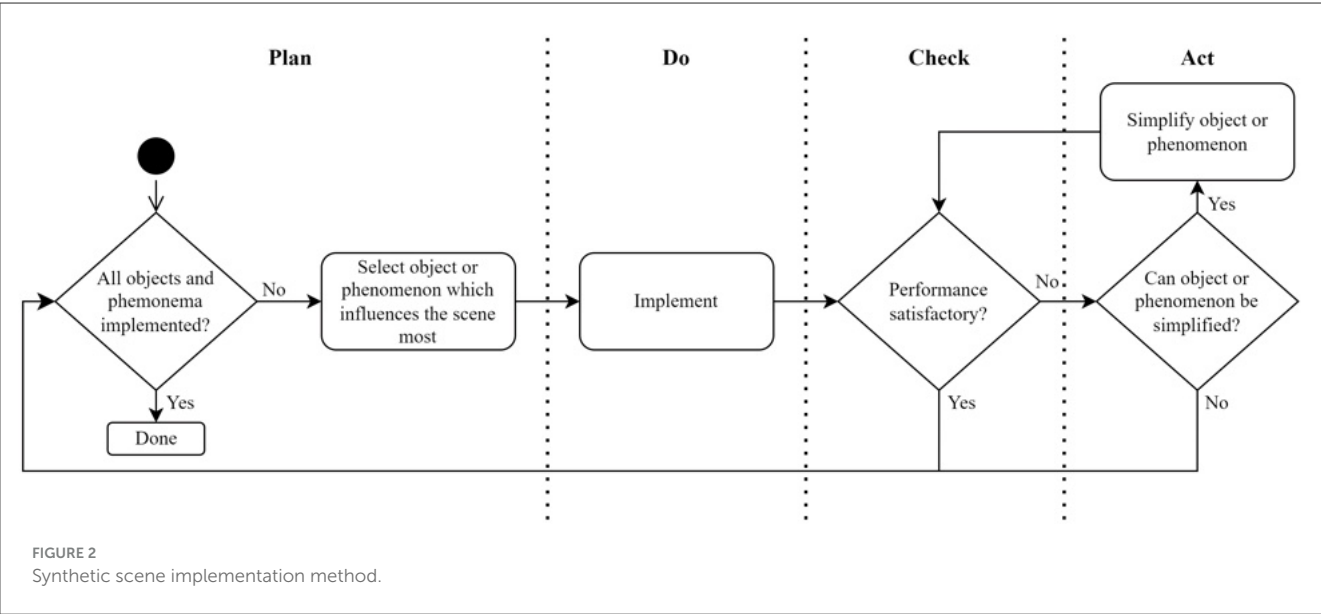
This process continues until the synthetic scene is populated with as many objects and phenomena present in the real-world scene as possible, such that the performance enables feasible dataset creation times. The objects in the synthetic scenes should not be placed exactly as seen in the real world; otherwise, the training would be prone to overfitting. Placement should rather be pseudo-random (within certain constraints) to increase the dataset variance and different perspectives of the objects.

After all feasible information has been added to the synthetic scene, a randomization process is used to create new unique scenes within the rules defined during the analysis phase. The automatic labeling process of these pseudo-random variations of the synthetically created scene is proposed in the following section.

2.2 Automatic labeling using a raycaster

All 3D modeling frameworks have a method of visualizing the scenes generated (Greff et al., 2022). This visualization is typically implemented using a camera object with a predefined location within and perspective of the scene. Now, consider a person walking down a street, their perspective can be mimicked by placing a camera in a synthetic scene at the height of an average person. Altering the camera's location would then mimic the person walking through the scene while altering the camera's pitch, roll, and yaw would mimic head rotation.

This imitation of the change of perspective will be used to generate multiple different images from one scene variation. Consequently, multiple scene variations will be produced, increasing the potential size of the generated dataset. Deep learning datasets are required to contain many different variations of inputs



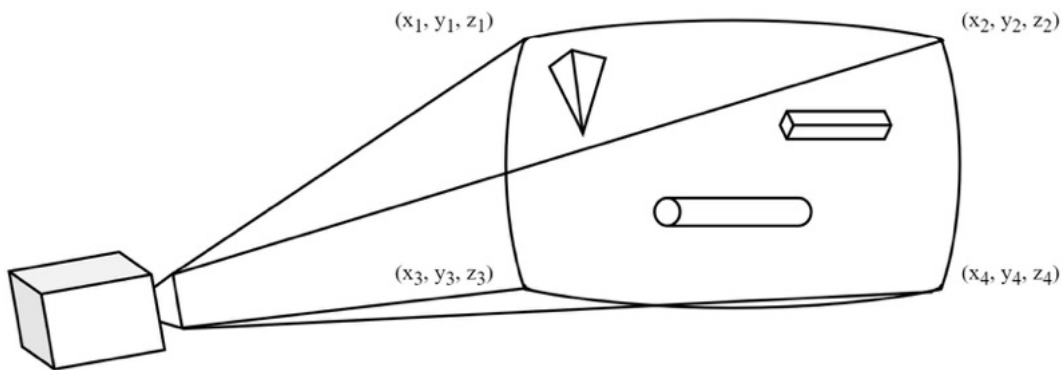


FIGURE 3
Camera perspective example.

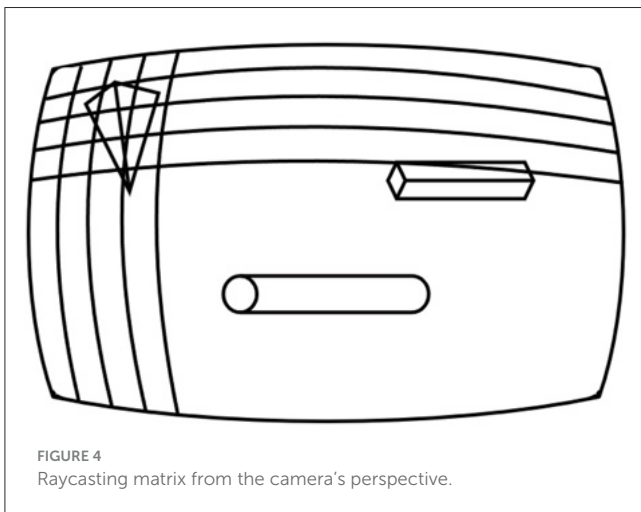


FIGURE 4
Raycasting matrix from the camera's perspective.

and expected outputs to the problem that needs to be solved. A dataset that is too small may result in overfitting, while a dataset that is too large may result in over-generalization during training.

The automatic labeling will be applied to each perspective sampled in the synthetic scenes. This process will determine the outlines (polygons) of all the objects visible within the current perspective. The perspective of a camera object is shown with lines indicating the extremes of the perspective in Figure 3. Objects (a pyramid, cylinder, and cuboid) have also been added to the scene for illustration purposes.

The automatic labeling process starts by defining the limits of the camera perspective (each corner represented by an (x_i, y_i, z_i) value in Figure 3) and then capturing the current camera's perspective as a 2D image. Therefore, the generated 3D scene is converted to a 2D image of the current perspective of the scene.

After the 2D image has been captured, the current camera perspective is divided into an $M \times N$ matrix for raycasting. Raycasting is the process of tracing a straight line sent from the origin of the camera's perspective toward a chosen coordinate to find the first object in its path. Figure 4 represents a simplified example of the raycasting matrix for visualization.

Each element in this matrix, represented by a vector with its origin at the camera's location, is raycast iteratively to identify what,

if any, object is present. This $M \times N$ matrix is then converted to sets of polygons (segmentation labels for irregularly shaped objects, i.e., not squares or circles) using Algorithm 1.

The raycast result matrix is iterated from top-left to bottom-right to identify all the individual instances of each predefined object class. When an element in the matrix matches an object class, the elements around it are tested in a clockwise approach (starting from the element directly to the right of it) to determine if their raycast result matches the same object class and ID (to differentiate between instances of object classes).

This process continues until the full outline of the current object is completed. Once all matrix elements are identified as part of an outline or do not contain the desired object classes, we convert these outlines into sets of (x, y) coordinates.

Data: $\phi \in M \times N$: Raycast matrix result
 $y \in P$: Object classes of interest
Result: $\hat{x} \in S \times T$: Instance segmentation polygons identified

initialize η = all unique objects in ϕ for all y
 initialize σ = edges of objects to be found

for $rowIndex$ in M & $colIndex$ in N **do**

if object at $\phi[rowIndex][colIndex]$ in η **then**

 set $origin = \phi[rowIndex][colIndex]$

 set $edgesOfObject = [origin]$

 set $edge =$ next edge of object found in a clockwise search from $origin$

while $edge \neq origin$ **do**

$edge =$ next edge of object found in a clockwise search from $edge$

 add $edge$ to $edgesOfObject$

 add $edgesOfObject$ to σ

 clear elements in ϕ encapsulated by $edgesOfObject$

convert all edges in σ to desired dataset format

Algorithm 1. $M \times N$ raycast matrix result to polygon labels.

These coordinate sets and the 2D perspective image are then added to the dataset. The labeling format should integrate well with the selected 3D framework and model training software, as described in Section 2.1.2. Repeating this process for all perspectives and scene variations will yield a synthetic instance segmentation dataset for the desired application. Increasing the scene variations and perspectives will expand the final dataset.

The proposed method of synthetic scene generation and automatic labeling comprises the overarching methodology for creating synthetic segmentation datasets designed to be broadly applicable. The validation strategy is discussed in the following section.

2.3 Validation strategy: image segmentation using transfer learning

A methodology has been proposed for synthetic dataset generation. However, its effectiveness in segmentation performance must be evaluated using a validation strategy. This strategy involves applying a transfer learning approach to train a segmentation model on the generated dataset and then using evaluation metrics to assess performance. Transfer learning leverages pre-trained models, freezing inner layers to retain learned features while modifying the final layer's weights using the new dataset. It circumvents the time-consuming and typically costly process of training a model from scratch.

The segmentation performance of the trained models will be analyzed using the evaluation metrics discussed in the introduction, namely, Intersection over Union (IoU) and Panoptic Quality (PQ). IoU is calculated as the area of overlap of the ground truth labels (A) and the predicted labels (B) divided by the area of the union of the ground truth labels and the predicted labels for each object instance, as shown in Equation 1.

$$IoU = \frac{|A \cap B|}{|A \cup B|} \quad (1)$$

PQ, shown to be a more stable metric than IoU (Mao et al., 2023), is calculated as the sum of the IoU for each object instance divided by the sum of the true positive (TP), false positive (FP), and false negative (FN) matches as shown in Equation 2.

$$PQ = \frac{\sum IoU}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|} \quad (2)$$

This validation strategy will be used to determine whether applying the proposed methodology creates datasets that, when used in a transfer learning approach, show increasing IoU and PQ performance metrics over each validation epoch. The results of applying the methodology and validation strategy are discussed in the following section.

3 Results and discussion

A methodology was proposed in the previous section for a synthetic segmentation dataset generator using a 3D modeling framework and raycaster. This section will present and discuss

each major result of the methodology's application to a deep-level mining case study. The results from the methodology's validation will be presented at the end of this section.

3.1 Synthetic scenes

The first part of the proposed synthetic scene creation is the analysis of the case study's real-world scenes. Figure 5 shows two images captured in the case study deep-level mine during the analysis phase. The analysis of these and other sample images showed that the objects typically found are pipes, valves, and various pumps and compressors. It was also found that inconsistent lighting conditions are present.

The 3D modeling framework chosen to be used in the generation of the synthetic scenes was Three.js (referred to hereafter as "the framework"). The framework was chosen due to its characteristics in each category described in Table 1, with the results listed below.

- 3D model imports and exports: the framework supports many different import formats.
- Raycasting: the framework includes a camera and raycasting class to efficiently perform the required changing of perspectives and raycast analysis.
- Usability and flexibility: comprehensive documentation and an active community ensure high usability. The framework is an abstraction of the WebGL API (used to render interactive 3D graphics in most web browsers), which decreases the implementation time of the synthetic scene generation while still ensuring flexible customization.
- Advanced physics and simulation: these are supported by the framework but are not relevant to the current application and, therefore, are not considered.
- Scalability and hardware requirements: as the framework is an abstraction of the WebGL API, the scalability and hardware are mostly affected by implementation efficacy. The choice in this category is not affected by costly computations of advanced physics and simulations as they are not required.
- Licensing and cost: The open-source framework aligns with the low-cost requirement of the application.

Popular alternatives to the framework are Unity and Blender. These include highly-advanced physics and simulation features. However, these features (amongst others) are not required by the current application and would only increase the computational requirements. Unity also requires the purchase of costly licenses.

The next step in the method is to find 3D models that can represent the real-world objects in the scene. As discussed in the analysis of the scenes in Section 2.1, some of the objects of interest include pipes, valves, and centrifugal pumps. Open-source 3D models for each of these object types are shown in Figure 6.

The final stage of scene generation uses the flow diagram in Figure 2, the chosen 3D modeling framework, and open-source 3D models. During the "Act" stage, an oversimplification that could compromise valuable training information should be avoided. The fidelity of a 3D model, typically defined by the count of triangular faces representing its surface, impacts the complexity of the raycast



FIGURE 5
Real-world images of the case study deep-level mine.

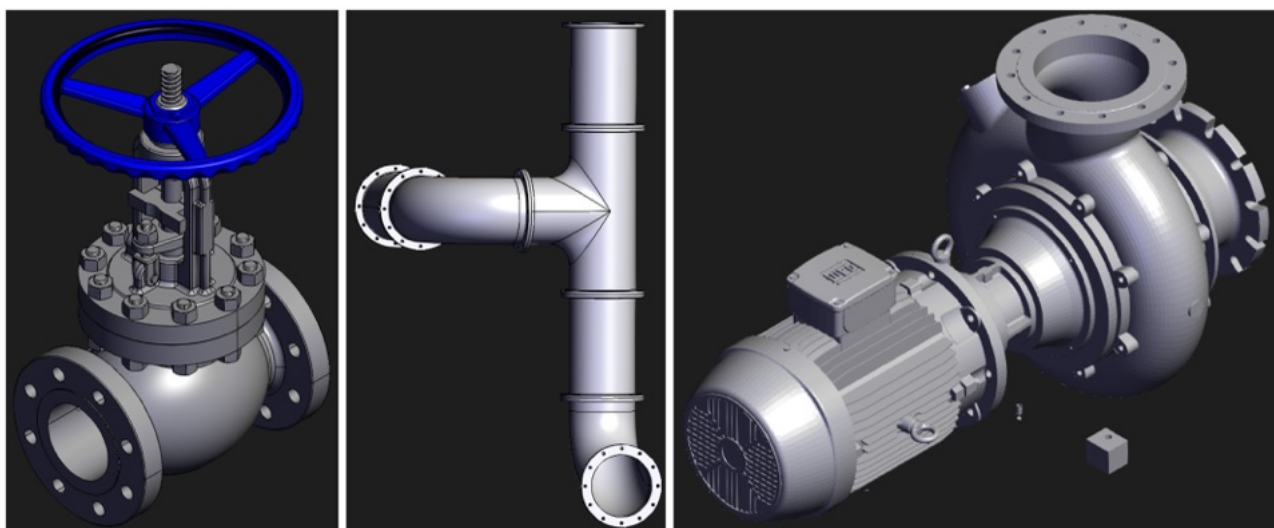


FIGURE 6
Open-source 3D models for objects of interest.

computation. Therefore, it is important to find a balance between model fidelity and raycast performance.

The images in Figure 7 show the result of applying the flow diagram using the chosen 3D modeling framework and open-source 3D models. The first image in Figure 7 shows the tunnel length generated (with textured walls) and the 3D models placed at pseudo-random positions. The second image shows a different perspective of the same generated synthetic scene.

Certain informed assumptions were required to increase the likelihood that the generated synthetic dataset produces

performant segmentation results by avoiding unnecessarily high levels of variance (Greff et al., 2022). Datasets are typically created for a specific application, as this ensures that they contain the required information for efficient segmentation model training (Minaee et al., 2022). Although this limits the generalizability of the specific dataset created, the proposed generation method is generalizable for many applications. This can be seen in the first steps of the proposed method, where the specific application's real-world scenes are analyzed. Therefore, the chosen scene can be varied as required.

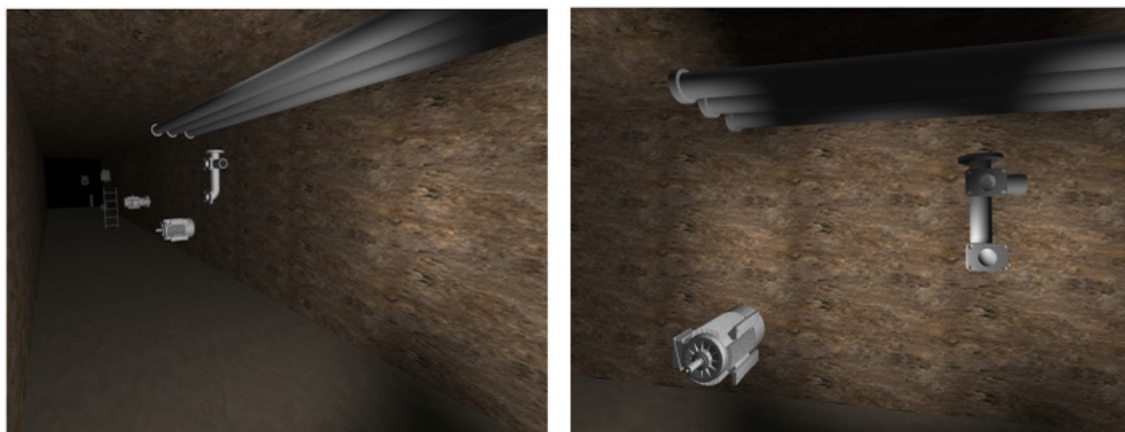


FIGURE 7
Two different perspectives of a generated synthetic scene.

3.2 Automatic labeling

This process was implemented using the framework's camera and raycast functionality. Once each perspective was generated, it was divided into a 256×256 matrix as shown in Figure 4, and each element was raycast to obtain the object present. Once all the elements in the matrix were raycast, the polygon edges were determined using Algorithm 1. Figures 8a, c show images from the generated dataset and Figures 8b, d the labeled images from the automated raycasting process.

Three pipes, a ladder, an induction motor, and a centrifugal pump have been automatically segmented and labeled in Figure 8b. A centrifugal pump, induction motors, pipes, and a ladder were automatically segmented in Figure 8d. This and other similar generated and labeled perspectives will be used in the transfer learning process to train the deep learning segmentation model in the following section.

3.3 Validation results

The dataset created contains six object classes: valves, pipes, ladders, induction motors, heat exchangers, and centrifugal pumps. The dataset consists of 329 automatically-labeled images. Two of the labeled images in the dataset are shown in Figure 8.

Figures 9a–c shows the analysis of the content of the dataset in terms of the total count per class, the total area per class, and finally, the average area per class. The dataset is unbalanced regarding the total segmentation count and total segmentation area. The pipe class has the highest count in the dataset by a margin of 444 against the average of the other classes. The pipe class has the largest total area by a margin of 67226.3 pixels against the average of the other classes. Figure 9c does, however, show a more balanced distribution for the average area of the segmentations per class. The unbalanced dataset was caused by the conclusion drawn from the analysis of

the real-world scenes, shown in Figure 5. The analysis showed that pipes are the most common objects in the images.

The deep-learning model chosen to use for instance segmentation is the improved Mask-RCNN with a ResNet-50-FPN backbone (Li et al., 2021) due to its instance segmentation performance in transfer learning approaches. The model was initialized from pre-trained weights using the COCO dataset. The weights for the backbone layers of the model were frozen to leverage the features learned from pre-training on the real-world dataset. A stochastic gradient descent (SGD) optimizer was combined with a decreasing learning rate each epoch as it yields competitive results in transfer learning approaches (Szegedy et al., 2016).

The performance metrics chosen for this study were the IoU and PQ metrics. IoU, as shown in Equation 1, was chosen as it circumvents the class imbalance problem of other metrics. PQ was chosen as it extends IoU to include true positives, false positives, and false negatives in the calculation, as shown in Equation 2. These reasons increase the likelihood of training a high-performing image segmentation model. The higher the IoU and PQ metric values, the better the segmentation performance.

The training was done for five epochs on 300 of the synthetic dataset images (the remaining 29 images were used during evaluation) with a learning rate of 0.004 and a weight decay of 0.0002. The training was done on a computer utilizing an NVIDIA GeForce GTX1650 GPU with 4GB dedicated memory. The average loss value steadily decreased until the final epoch. Additional training did not show a significant decrease in loss; therefore, the number of epochs was not increased beyond five to avoid overfitting, resulting in a final training time of 16 minutes.

Table 2 shows the improvement of performance metrics over each training epoch. Both metrics increased quickly in the first few epochs (as all the training data is new and the learning rate is at its highest). The metrics increased less after the third epoch. The final epoch produces the best result for both metrics.

A comparison between the produced segmentation results and existing methods discussed in a recent survey paper (Minaee et al., 2022) is shown in Table 3. The methods in the survey paper were

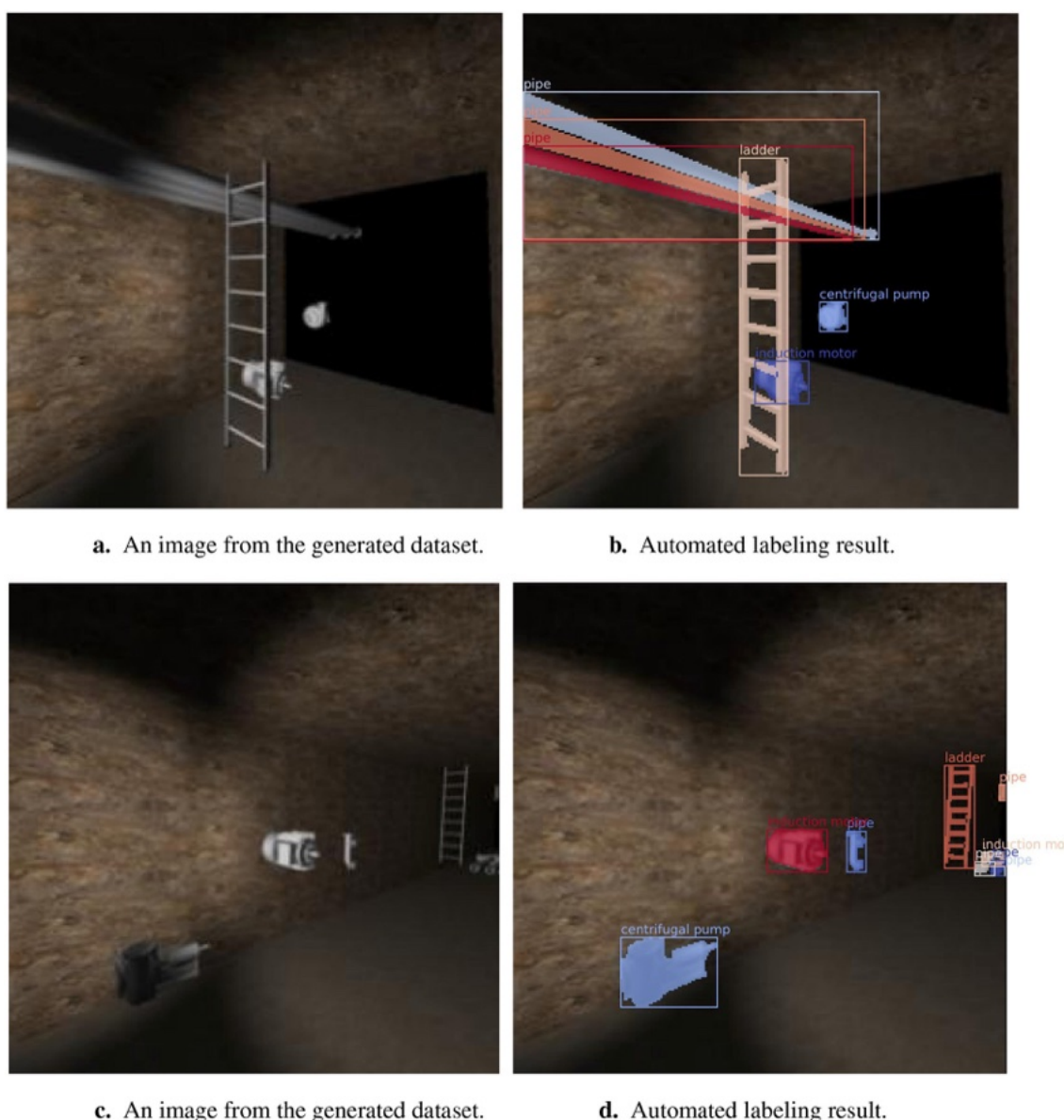


FIGURE 8

Automated labeling results for images in the generated dataset. (a) An image from the generated dataset. (b) Automated labeling result. (c) An image from the generated dataset. (d) Automated labeling result.

trained on the real-world COCO dataset. The model trained with the synthetic dataset created from our proposed generator reached similar performance metric results.

It would not be accurate to claim that the segmentation model trained with synthetic data outperformed any other model, as results from the survey paper were obtained using a different real-world dataset. However, from this comparison, it can be concluded that training a model using a dataset generated through our proposed method produces similar performance results.

The model using a ResNet-101 backbone obtained the best results among the models from the survey paper, as expected, due to the increased number of layers. However, the increased number of layers does lengthen the training and final inference time and the required computational power.

Figure 10a presents an automatically-generated input image, and Figure 10b shows the segmentation outcome post-final training epoch. All objects in the test image were segmented with confidence scores of at least 64.07%. The two pipes were segmented with high confidence scores (above 91%). However, they were split due to the ladder being in front of the pipes.

Although also located behind the ladder, the centrifugal pump was segmented with a 77.16% confidence score, and the bounding box encompasses almost the entire object. The ladder, although split into multiple segments, was identified with high confidence scores (ranging from 64% to 84%). The performance of the segmentation model in the final round of evaluation was an IoU and PQ score of 39.35 and 49.16, respectively, as shown in Table 2.

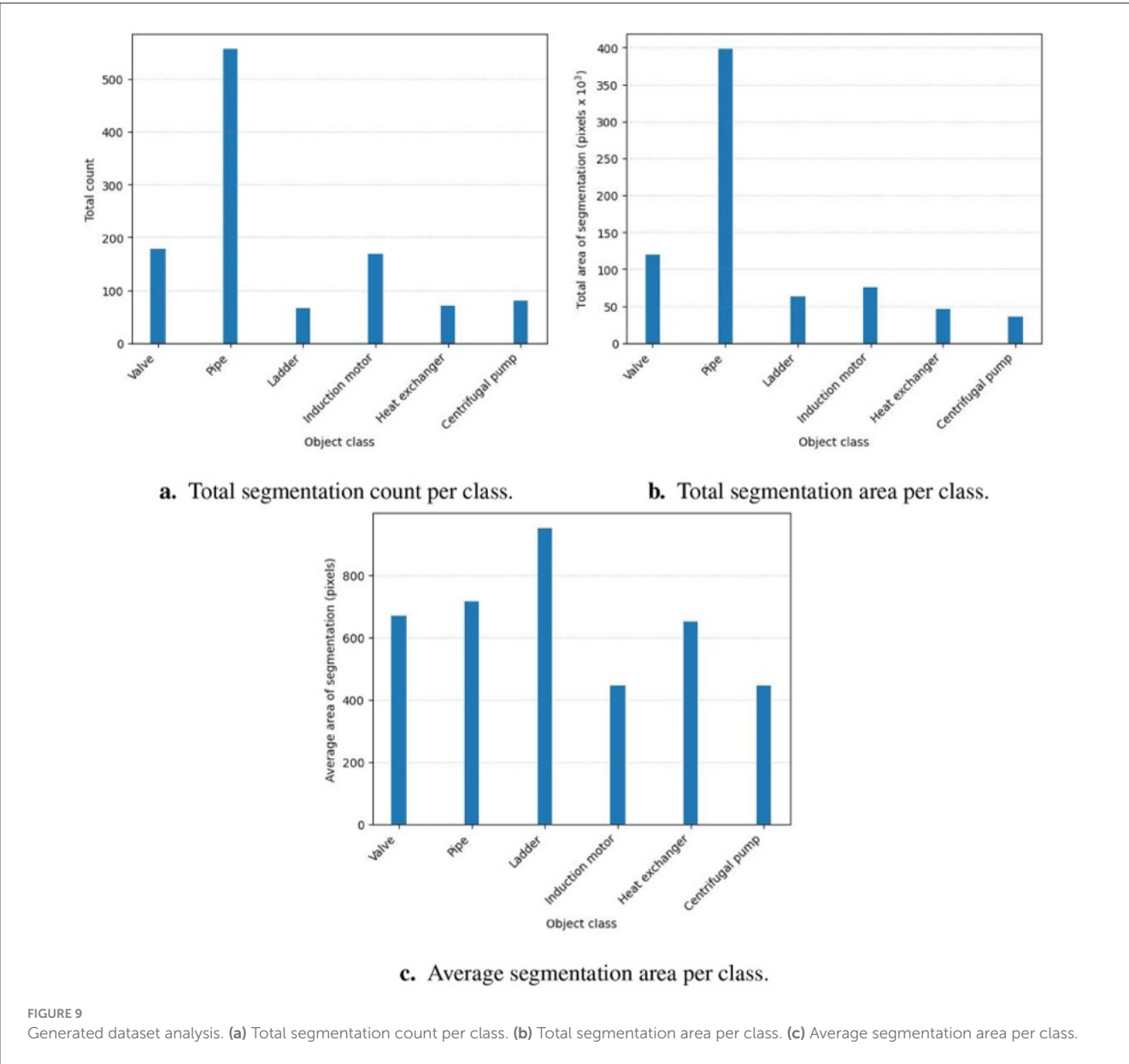


TABLE 2 Improvement of performance metrics over epochs.

Metric	Epoch 1	Epoch 2	Epoch 3	Epoch 4	Epoch 5
IoU	18.98	31.26	37.04	38.85	39.35
PQ	21.23	39.24	44.33	47.14	49.16

Figure 11 shows the segmentation results of the real-world case study images from Figure 5. All segmentations with a confidence score lower than 40% were ignored. In Figure 11a, almost all the pipes were correctly segmented and labeled. The top-left valve was segmented with a confidence score of 68.66% (with a segmentation discrepancy at the top and the right of the valve).

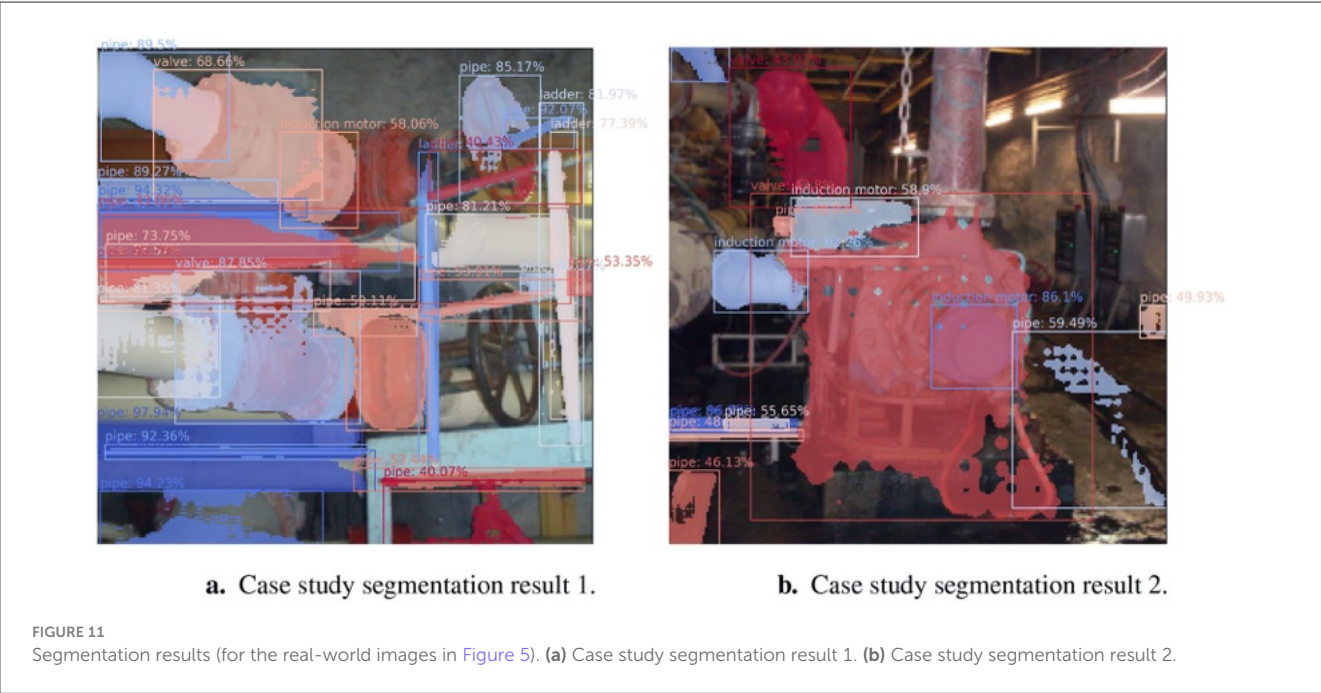
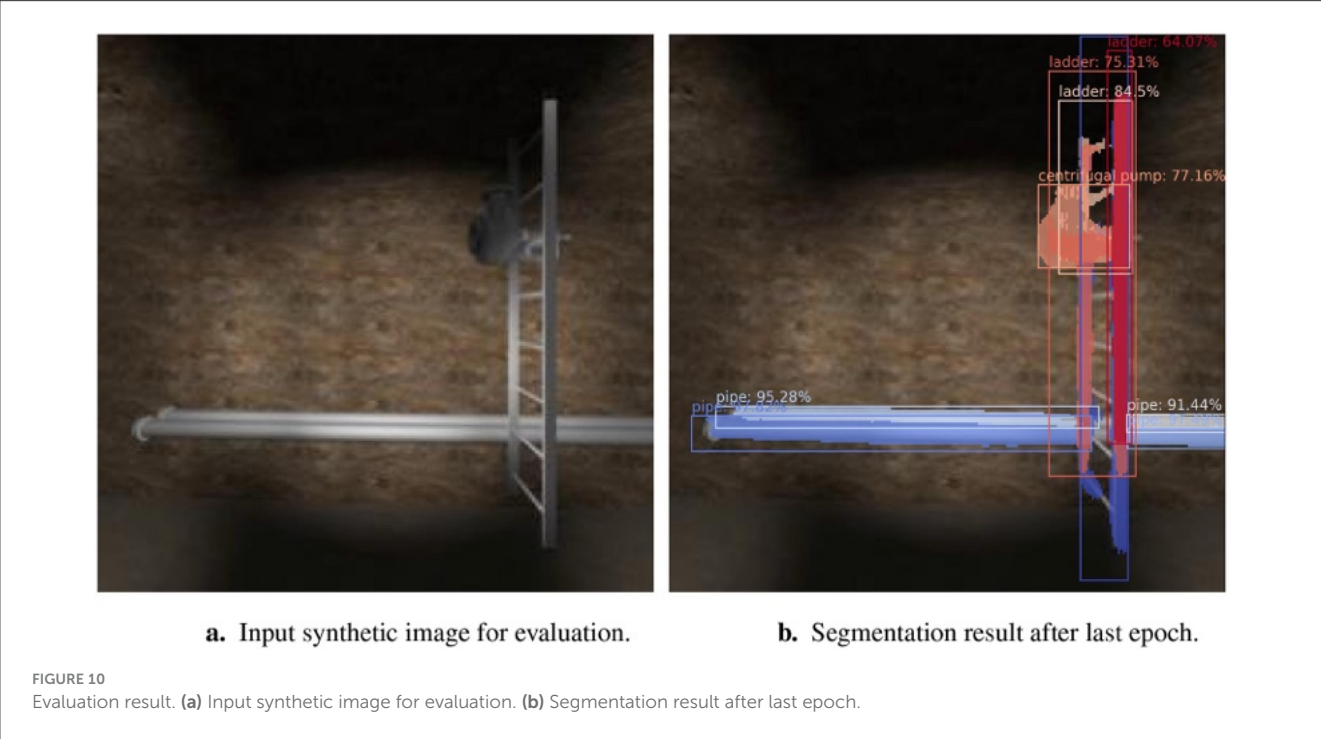
In Figure 11a, the bottom-left valve’s profile was mostly segmented with a confidence score of 87.85% (with a discrepancy at the top of the profile). The railing was identified as a ladder, as it

TABLE 3 IoU results obtained in comparison with recent studies.

Method	Backbone	IoU
DA-Net	ResNet-50	37.9
Our proposed generator	ResNet-50	39.35
EMA-Net	ResNet-50	39.9
AC-Net	ResNet-101	40.1

resembles the shape of a ladder in the training data. Two valves and three of the pipes were not segmented.

The non-drive end bearing housing in the center of Figure 11b was segmented but incorrectly labeled as an induction motor with a confidence score of 86.1% as it resembles the profile of an induction motor in the synthetic dataset. The profile of the large pump in the middle of the image was segmented well but was mislabeled as it resembles a valve in the training dataset. The valve to the top-left of



the image was partially segmented with a score of 43.02%. Various objects (such as the grate to the bottom left and the railway to the bottom right) were mislabeled as they resemble the pipes in the training dataset.

The segmentation performance on the case study images was obtained by manually creating ground truth segmentations and calculating the IoU and PQ metrics for the automatically segmented images. The average IoU and PQ scores achieved on the case study images were 34.15 and 32.41, respectively, as shown in Table 4.

TABLE 4 Final segmentation performance.

Segmented content	IoU	PQ
Evaluation subset of synthetic dataset	39.35	49.16
Case study images	34.15	32.41

The PQ result for the case study images was lower than the IoU result, which is not the case for the synthetic dataset. This is possibly due to the PQ metric being influenced by the false positives, false

negatives, and true positives, as some of the objects in the case study images were segmented well in terms of the object profile but were mislabeled as other classes.

The segmentation model's performance on the case study images was slightly lower than that achieved in the final evaluation step during the testing phase with the synthetic dataset. The use of transfer learning to retain the real-world learned features was successful, as no real-world images of the chosen case study were used during model training, and only a slight loss of performance was seen.

All the validation results have shown that the proposed method to generate synthetic data produces datasets usable for training instance segmentation models and that the performance is comparable to those of recent studies using real-world datasets. The validation also showed that the method is cost-efficient, as a relatively low-powered computer was used to train the segmentation model, and the final training time was only 16 minutes.

4 Conclusion

The proposed methodology for creating a synthetic dataset of segmented and labeled scenes was applied to a deep-level mining case study. The real-world images contained objects such as pipes, valves, induction motors, and ladders. These images were analyzed, and the synthetic scenes were designed and constructed. 3D models, such as pipes and valves, were utilized to increase the similarity between synthetic and real-world scenes. A camera and raycaster were used to create and label different perspectives within the generated scenes to produce the final synthetic dataset.

The validation comprised utilizing the created dataset in a transfer learning approach on an instance segmentation model while calculating the IoU and PQ metrics after each epoch. The performance metrics increased (initially in large increments) over each epoch and plateaued after five epochs. The inference on the validation dataset showed high confidence instance segmentation and labeling on each object of interest present.

The segmentation performance achieved with the proposed generator was compared to the results reported in a recent survey paper. It showed that the IoU metric results obtained are similar to those reported in the survey paper.

The segmentations of the case study images showed that most of the real-world object's profiles could be segmented and labeled with high confidence and accuracy. Some objects of interest defined for the synthetic dataset were not segmented or mislabeled. However, the results on the real-world data are promising when considering the relatively inexpensive hardware used to train the segmentation model, the final training time, and the fact that a synthetic dataset was used for training.

References

- Ahleroff, S., Xu, X., Zhong, R. Y., and Lu, Y. (2021). Digital Twin as a Service (DTaaS) in Industry 4.0: an architecture reference model. *Adv. Eng. Inform.* 47:101225. doi: 10.1016/j.aei.2020.101225
- Ahmad, M., Sadiq, S., Alluhaidan, A. S., Umer, M., Ullah, S., Nappi, M., et al. (2022). Industry 4.0 technologies and their applications in fighting COVID-19 pandemic using deep learning techniques. *Comput. Biol. Med.* 145:105418. doi: 10.1016/j.combiomed.2022.105418

Future work includes experiments with the proposed generator in various case studies, such as image segmentation for indoor manufacturing or construction sites, to test its generalizability. Investigations into the effects of creating a balanced versus biased dataset for various case studies' real-world scenes can also be conducted to determine a balance between generalizability and segmentation performance. Lastly, testing the dataset generator with various segmentation algorithms is also recommended for future work.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

WJK: Conceptualization, Formal analysis, Methodology, Software, Writing – original draft. JP: Validation, Writing – review & editing. JV: Conceptualization, Supervision, Validation, Writing – review & editing. ST: Formal analysis, Writing – review & editing.

Funding

The authors declare that this study received funding from ETA Operations (Pty) Ltd. The funder supplied the case study images for the validation of the proposed dataset generator. The funder was not involved in the study design, collection, analysis, interpretation of data, the writing of this article, or the decision to submit it for publication.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

- Baek, J., Park, J., Cho, S., and Lee, C. (2022). 3D global localisation in the underground mine environment using mobile LiDAR mapping and point cloud registration. *Sensors* 22, 1–23. doi: 10.3390/s22082873
- Bidgoli, N. M., Maugey, T., and Roumy, A. (2020). Fine granularity access in interactive compression of 360° images based on rate-adaptive channel codes. *IEEE Trans. Multim.* 23, 2868–2882. doi: 10.1109/TMM.2020.3017890
- Cheng, B., Collins, M. D., Zhu, Y., Liu, T., Huang, T. S., Adam, H., et al. (2020). “Panoptic-DeepLab: a simple, strong, and fast baseline for bottom-up panoptic segmentation,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12475–12485. doi: 10.1109/CVPR42600.2020.01249
- Choi, S., Jung, S., Yun, H., Kim, J. T., Kim, S., and Choo, J. (2021). “RobustNet: improving domain generalisation in urban-scene segmentation via instance selective whitening,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11580–11590. doi: 10.1109/CVPR46437.2021.01141
- Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., et al. (2016). “The Cityscapes dataset for semantic urban scene understanding,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 3213–3223. doi: 10.1109/CVPR.2016.350
- Dentch, M. P. (2016). *The ISO 9001, 2015 Implementation Handbook*. Milwaukee, Wisconsin: ASQ Quality Press.
- Dhamija, P. (2022). South Africa in the era of Industry 4.0: an insightful investigation. *Scientometrics* 127, 5083–5110. doi: 10.1007/s11192-022-04461-z
- Duan, R., Deng, H., Tian, M., Deng, Y., and Lin, J. (2022). SODA: a large-scale open site object detection dataset for deep learning in construction. *Automat. Constr.* 142:104499. doi: 10.1016/j.autcon.2022.104499
- Greff, K., Belletti, F., Beyer, L., Doersch, C., Du, Y., Duckworth, D., et al. (2022). “Kubric: a scalable dataset generator,” in *IEEE Conference on Computer Vision and Pattern Recognition*. doi: 10.1109/CVPR52688.2022.00373
- He, J., Erfani, S., Ma, X., Bailey, J., Chi, Y., and Hua, X.-S. (2021). “Alpha-IoU: a family of power Intersection over Union losses for bounding box regression,” in *Advances in Neural Information Processing Systems*, 20230–20242.
- Hosna, A., Merry, E., Gyalmo, J., Alom, Z., Aung, Z., and Azim, M. A. (2022). Transfer learning: a friendly introduction. *J. Big Data* 9:102. doi: 10.1186/s40537-022-00652-w
- Khayyam, H., Jamali, A., Bab-Hadiashar, A., Esch, T., Ramakrishna, S., Jalili, M., et al. (2020). A novel hybrid machine learning algorithm for limited and big data modeling with application in Industry 4.0. *IEEE Access* 8, 111381–111393. doi: 10.1109/ACCESS.2020.2999898
- Kirillov, A., He, K., Girshick, R., Rother, C., and Dollár, P. (2019). “Panoptic segmentation,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9404–9413. doi: 10.1109/CVPR.2019.00963
- Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., et al. (2023). “Segment anything,” in *IEEE/CVF International Conference on Computer Vision*. doi: 10.1109/ICCV51070.2023.00371
- Kong, F., Huang, B., Bradbury, K., and Malof, J. (2020). “The Synthinel-1 dataset: a collection of high resolution synthetic overhead imagery for building segmentation,” in *IEEE/CVF Winter Conference on applications of Computer Vision*, 1814–1823. doi: 10.1109/WACV45572.2020.9093339
- Lee, J. G., Hwang, J., Chi, S., and Seo, J. (2022). Synthetic image dataset development for vision-based construction equipment detection. *J. Comput. Civil Eng.* 36:04022020. doi: 10.1061/(ASCE)CP.1943-5487.0001035
- Li, Y., Xie, S., Chen, X., Dollar, P., He, K., and Girshick, R. (2021). Benchmarking detection transfer learning with vision transformers. *arXiv preprint arXiv:2111.11429*.
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., et al. (2014). “Microsoft COCO: common objects in context,” in *Computer Vision* (Springer), 740–755. doi: 10.1007/978-3-319-10602-1_48
- Lv, Z., Wang, J.-Y., Kumar, N., and Lloret, J. (2022). *Augmented Reality, Virtual Reality semantic 3D reconstruction*. Basel: Multidisciplinary Digital Publishing Institute. doi: 10.3390/books978-3-0365-6062-5
- Manettas, C., Nikolakis, N., and Alexopoulos, K. (2021). Synthetic datasets for deep learning in computer-vision assisted tasks in manufacturing. *Proc. CIRP* 103, 237–242. doi: 10.1016/j.procir.2021.10.038
- Mao, L., Ren, F., Yang, D., and Zhang, R. (2023). ChainNet: deep chain instance segmentation network for panoptic segmentation. *Neural Proc. Lett.* 55, 615–630. doi: 10.1007/s11063-022-10899-2
- Millietari, F., Navab, N., and Ahmadi, S.-A. (2016). “V-Net: fully convolutional neural networks for volumetric medical image segmentation,” in *Fourth International Conference on 3D Vision* (IEEE), 565–571. doi: 10.1109/3DV.2016.79
- Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., and Terzopoulos, D. (2022). Image segmentation using deep learning: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 44, 3523–3542. doi: 10.1109/TPAMI.2021.3059968
- Neuhold, G., Ollmann, T., Rota Bulò, S., and Kotschieder, P. (2017). “The Mapillary Vistas dataset for semantic understanding of street scenes,” in *IEEE International Conference on Computer Vision*, 4990–4999. doi: 10.1109/ICCV.2017.534
- Prinsloo, J., Vosloo, J. C., and Mathews, E. H. (2019). Towards Industry 4.0: a roadmap for the South African heavy industry sector. *South African J. Ind. Eng.* 30, 174–186. doi: 10.7166/30-3-2237
- Radanliev, P., De Roure, D., Nicolescu, R., Huth, M., and Santos, O. (2022). Digital Twins: Artificial Intelligence and the IoT Cyber-Physical Systems in Industry 4.0. *Int. J. Intell. Robot. Applic.* 6, 171–185. doi: 10.1007/s41315-021-00180-5
- Stark, R., Freseemann, C., and Lindow, K. (2019). Development and operation of Digital Twins for technical systems and services. *CIRP Ann.* 68, 129–132. doi: 10.1016/j.cirp.2019.04.024
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). “Rethinking the inception architecture for computer vision,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2818–2826. doi: 10.1109/CVPR.2016.308
- Wang, J., Huang, Y., Zhai, W., Li, J., Ouyang, S., Gao, H., et al. (2023). Research on coal mine safety management based on Digital Twin. *Heliyon* 9:e13608. doi: 10.1016/j.heliyon.2023.e13608
- Wang, Y., Ahsan, U., Li, H., and Hagen, M. (2022). A comprehensive review of modern object segmentation approaches. *Found. Trends Comput. Graph. Vision* 13, 111–283. doi: 10.1561/06000000097
- Wei, Y., and Akinci, B. (2019). A vision and learning-based indoor localisation and semantic mapping framework for facility operations and management. *Autom. Constr.* 107:102915. doi: 10.1016/j.autcon.2019.102915
- Wu, Z., Zhao, L., and Zhang, H. (2021). MR-UNet commodity semantic segmentation based on transfer learning. *IEEE Access* 9, 159447–159456. doi: 10.1109/ACCESS.2021.3130578
- Zanella, R., Caporali, A., Tadaka, K., De Gregorio, D., and Palli, G. (2021). “Auto-generated wires dataset for semantic segmentation with domain-independence,” in *International Conference on Computer, Control and Robotics*, 292–298. doi: 10.1109/ICCCCR49711.2021.9349395