Geometric, Semantic, and System-Level Scene Understanding for Improved Construction and Operation of the Built Environment

by

Lichao Xu

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Civil Engineering)
in the University of Michigan
2020

Doctoral Committee:

Professor Vineet R. Kamat, Co-Chair
Associate Professor Carol C. Menassa, Co-Chair
Assistant Professor Jia Deng
Professor SangHyun Lee

Lichao Xu

lichaox@umich.edu

ORCID iD: 0000-0001-6654-6274

Dedication

To my parents

and

To my wife

for their love, endless encouragement and support

Acknowledgments

I would like to express my sincere gratitude to my advisors Professor Vineet Kamat and Professor Carol Menassa for guiding and supporting me during my Ph.D. Professor Kamat has been encouraging and supportive, given me the freedom to explore various research directions, provided me with opportunities to exchange ideas and collaborate with researchers from different fields, and continuously helped me improve my research. His experience and wisdom have given me lots of confidence since I know I can always get valuable suggestions from him when I get stuck at some problem. Professor Menassa is always full of passion and has the unique capability of capturing the details of a problem. She has given me many insightful options that allowed me to avoid pitfalls in my projects and helped me continuously improve my work. She has also given me lots of valuable comments and advice on my academic presentation and writing skills. I have learned a great deal from them, and I really appreciate the opportunity to work with them.

I would like to sincerely thank Professor SangHyun Lee for his constructive and critical comments on my projects. These comments were very helpful to organize the research and strengthen my contributions. His thoughtful comments were also highly valuable to improve the logical coherence of the dissertation.

I would also like to sincerely thank Professor Jia Deng for his critical questions, valuable feedback, and constructive comments that all helped me think more about the limitations of different algorithms and eventually find a feasible solution to my research question.

I would also like to sincerely thank Professor Sherif El-Tawil, Professor Atul Prakash, Professor Seymour Spence, Professor Jason McCormick, Professor Benigno Aguirre and other colleagues in the ICoR project, Szu-Yun Lin, Andrew Hlynka, Hao Lu, Omar Sediek, Ahmed Abdelhady, and Eileen Young. Their insightful discussions, constructive advice, and impressive presentations were essential to the success of my work on distributed coupling analysis. Collaboration with them also provided me with a valuable opportunity to know about different fields such as computer science, wind engineering, earthquake engineering, structural engineering, and disaster engineering.

I am also deeply grateful to Dr. Suyang Dong and Dr. Chen Feng, who have given me valuable suggestions on work-life balance, course selection, research projects, communication skills, and collaboration with others. Dr. Feng has also closely collaborated with me on some research projects. His creative ideas and valuable feedback were essential to the completion of the SAFT project.

I would also like to thank my lab mates (LIVE'ers and SICIS'ers) Yong Xiao, Kurt Lundeen, Albert Thomas, Bharadwaj Mantha, Da Li, Ci-Jyun Liang, Xi Wang, Sumukha Udupa and Min Deng, for their thoughtful suggestions, assistance with projects, and the good memories that we had together in the past few years.

Last but not least, I would like to thank all my family members and all my friends whose encouragement and support made this possible.

Table of Contents

Dedication .... **ii**

Acknowledgments .... **iii**

List of Tables .... **viii**

List of Figures .... **ix**

Abstract .... **xii**

Chapter 1 Introduction **1**

1.1 Importance of Research 3

1.2 Background and Literature Review 5

1.2.1 Localization 5

1.2.2 Object Recognition 8

1.2.3 Distributed Coupling Analysis 10

1.3 Objectives 11

1.4 Methodology 12

1.5 Dissertation Outline 13

Chapter 2 An Occupancy Grid Mapping Enhanced Visual SLAM for Real-Time Locating Applications in GNSS-Denied Environments **15**

2.1 Introduction 15

2.2 Review of Related Prior Research 16

2.2.1 Wireless Technology-Based Localization Approaches 17

2.2.2 Vision-Based and Inertial Sensor-Based Localization Approaches 18

2.2.3 SLAM-Based Localization Approaches 19

2.3 Research Objectives, Scope and Contribution 23

2.4 Technical Approach 24

2.4.1 SLAM Mode 27

2.4.2 Localization Mode 38
2.5 Experimental Verification 41
2.5.1 Marker Pose Estimation 43
2.5.2 Experimental Results and Analysis 44
2.6 Example Applications of the Proposed RTLS 50
2.6.1 Path Planning and Real-Time Navigation 51
2.6.2 Geo-Tagged Data Collection 52
2.6.3 Localization-Aware Point Cloud Update 53
2.7 Conclusions and Future Work 54
Chapter 3 A Scene-Adaptive Descriptor for Visual SLAM-Based Locating Applications . 56
3.1 Introduction 56
3.2 Review of Related Prior Research 58
3.2.1 SLAM-Based Locating Solutions 58
3.2.2 Feature-Based vSLAM 60
3.2.3 Feature Descriptors in vSLAM 61
3.3 Research Objective, Scope and Contribution 63
3.4 Deep SAFT Embedded ORB-SLAM2 (RGB-D) 65
3.4.1 Learning-Based Descriptor 68
3.4.2 Tracking with SAFT 73
3.4.3 Online SAFT Learning 75
3.5 Experiments and Evaluation 79
3.5.1 Quantitative Results 80
3.5.2 Qualitative Results 85
3.5.3 Computational Performance 90
3.6 Conclusions and Future Work 91
Chapter 4 Automatic Extraction of 1D Barcodes from Video Scans for Drone-Assisted Inventory Management in Warehousing Applications 94
4.1 Introduction 94
4.2 Technical Approach and Related Work 96
4.3 Technical Approach Details 101
4.3.1 Barcode Direction Estimation 101
4.3.2 Barcode Region Detection 105
4.3.3 Fast Extraction 108

4.4 Experimental Results and Analysis 110
4.5 Conclusions and Future Work 118

Chapter 5 Distributed Coupling Analysis for Modeling and Understanding Built-Environment Processes: Reviews, Limitations, and Recommendations 121

5.1 Introduction 121
5.2 Existing Distributed Analysis Solutions 124
5.2.1 Standards and Standard-Based Solutions 125
5.2.2 Standalone Tools 134
5.3 Limitations 138
5.3.1 Lack of Easy-to-use and Standard Solutions 138
5.3.2 Lack of Scalability and Extensibility for Building Large-Scale Analyses 140
5.3.3 Inability to Rapidly Build and Integrate Application-Specific Analysis Models 141
5.4 Recommendations 142
5.4.1 Proposed Design of a Message Wrapper 144
5.4.2 Hybrid Data Passing Between Message Wrappers 152
5.5 Conclusions and Future Work 153

Chapter 6 Conclusions 155

6.1 Significance of Research 155
6.2 Research Contributions 156
6.3 Future Directions 157
6.3.1 Localization 157
6.3.2 Object Recognition 158
6.3.3 Distributed Coupling Analysis 158

Bibliography 159

List of Tables

Table 2.1 Evaluation results of marker position measurement 48

Table 2.2 Evaluation results of marker distance measurement 50

Table 3.1 Computational performance evaluation 91

List of Figures

Figure 1.1 Research Overview ... 2

Figure 2.1 Overview of the proposed RTLS ... 25

Figure 2.2 SLAM mode of the system ... 27

Figure 2.3 Flowchart of the Modified ORB SLAM algorithm ... 31

Figure 2.4 Flowchart of the Occupancy Grid Mapping algorithm ... 34

Figure 2.5 Visualization of occupancy grid mapping with a single laser scan ... 35

Figure 2.6 Localization mode of the system ... 39

Figure 2.7 Localization results on the maps in a building scale environment ... 41

Figure 2.8 World frame, camera frame and marker frame and the transformations between them ... 44

Figure 2.9 Measurements of a single marker at different locations ... 45

Figure 2.10 Marker deployment in a basement corridor environment and localization results ... 47

Figure 2.11 Visualization of marker position RMSE ... 49

Figure 2.12 Visualization of marker distance RMSE ... 50

Figure 2.13 Testing results of path planning and real-time navigation on two maps ... 52

Figure 2.14 Geo-tagged environmental data collection with a TurtleBot platform ... 53

Figure 2.15 Point cloud maps of two environments created based on the proposed system ... 54

Figure 3.1 System overview of Deep SAFT embedded ORB-SLAM2 (RGB-D) ... 66

Figure 3.2 Visualization of Deep SAFT's weight updates as frame advances ... 67

Figure 3.3 A general Siamese neural network structure for a learning-based descriptor ... 68

Figure 3.4 The training network architecture for SAFT descriptor ... 69

Figure 3.5 Training results and evaluation of different descriptor distance measurements ... 72

Figure 3.6 Visualization of training data preparation in the local map of the current frame 77

Figure 3.7 Some challenging scenes from the evaluation datasets 81

Figure 3.8 Performance evaluation on seven sequences from the TUM RGB-D dataset 83

Figure 3.9 Performance evaluation on the ICL-NUIM dataset 83

Figure 3.10 Evaluation results on the fr1/room sequence from the TUM RGB-D dataset......................... 86

Figure 3.11 Evaluation results on the or3 sequence from the ICL-NUIM dataset..................................... 87

Figure 3.12 Evaluation results on the or1 sequence from the ICL-NUIM dataset..................................... 88

Figure 3.13 Matching inlier ratios in the tracking process on the or3 sequence 90

Figure 4.1 Manual barcode scanning in typical warehouse environments 95

Figure 4.2 Overview of the automatic scan solution.. 97

Figure 4.3 Process of preparing barcode regions for existing decoding algorithms 98

Figure 4.4 Readability of the same barcode in different angular states 102

Figure 4.5 Procedures for barcode direction estimation 104

Figure 4.6 Algorithm for barcode region detection 106

Figure 4.7 Visualization of barcode region detection 107

Figure 4.8 Algorithm for key frame selection based on histogram difference 109

Figure 4.9 Visualization of key frame selection 110

Figure 4.10 Complete algorithm for reading barcode from video scan data.. 111

Figure 4.11 Illustration of processing a specific key frame 112

Figure 4.12 Evaluation of barcode region detection 114

Figure 4.13 Evaluation of barcode direction adjustment 116

Figure 4.14 Evaluation of key frame selection 117

Figure 5.1 Point-to-point message exchange via UDP in DIS 127

Figure 5.2 Mediator-based object exchange via UDP/TCP in HLA.. 130

Figure 5.3 Message exchange via multicast UDP in LCM 137

Figure 5.4 A recommended design for a distributed analysis platform 144

Figure 5.5 An LCM-based distributed coupling analysis framework 145

Figure 5.6 Concept of a message wrapper 147

Figure 5.7 Procedures for developing an analysis model with a message wrapper 148

Figure 5.8 An example implementation of the message wrapper for LCM 149

Figure 5.9 Distributed analysis design of an active control algorithm 150

Figure 5.10 Analysis results without active control 151

Figure 5.11 Analysis results with active control of acceleration 151

Abstract

Recent advances in robotics and enabling fields such as computer vision, deep learning, and low-latency data passing offer significant potential for developing efficient and low-cost solutions for improved construction and operation of the built environment. Examples of such potential solutions include the introduction of automation in environment monitoring, infrastructure inspections, asset management, and building performance analyses. In an effort to advance the fundamental computational building blocks for such applications, this dissertation explored three categories of scene understanding capabilities: 1) Localization and mapping for geometric scene understanding that enables a mobile agent (e.g., robot) to locate itself in an environment, map the geometry of the environment, and navigate through it; 2) Object recognition for semantic scene understanding that allows for automatic asset information extraction for asset tracking and resource management; 3) Distributed coupling analysis for system-level scene understanding that allows for discovery of interdependencies between different built-environment processes for system-level performance analyses and response-planning.

First, this dissertation advanced Simultaneous Localization and Mapping (SLAM) techniques for convenient and low-cost locating capabilities compared with previous work. To provide a versatile Real-Time Location System (RTLS), an occupancy grid mapping enhanced visual SLAM (vSLAM) was developed to support path planning and continuous navigation that cannot be implemented directly on vSLAM's original feature map. The system's localization accuracy was experimentally evaluated with a set of visual landmarks. The achieved marker position measurement accuracy ranges from 0.039m to 0.186m, proving the method's feasibility and applicability in providing real-time localization for a wide range of applications. In addition, a Self-Adaptive Feature Transform (SAFT) was proposed to improve such an RTLS's robustness in challenging environments. As an example implementation, the SAFT descriptor was implemented with a learning-based descriptor and integrated into a vSLAM for experimentation. The evaluation results on two public datasets proved the feasibility and effectiveness of SAFT in improving the matching performance of learning-based descriptors for locating applications.

Second, this dissertation explored vision-based 1D barcode marker extraction for automated object recognition and asset tracking that is more convenient and efficient than the traditional methods of using barcode or asset scanners. As an example application in inventory management, a 1D barcode extraction framework was designed to extract 1D barcodes from video scan of a built environment. The performance of the framework was evaluated with video scan data collected from an active logistics warehouse near Detroit Metropolitan Airport (DTW), demonstrating its applicability in automating inventory tracking and management applications.

Finally, this dissertation explored distributed coupling analysis for understanding interdependencies between processes affecting the built environment and its occupants, allowing for accurate performance and response analyses compared with previous research. In this research, a Lightweight Communications and Marshalling (LCM)-based distributed coupling analysis framework and a message wrapper were designed. This proposed framework and message wrapper were tested with analysis models from wind engineering and structural engineering, where they demonstrated the abilities to link analysis models from different domains and reveal key interdependencies between the involved built-environment processes.

Chapter 1
Introduction

Human activity, safety, and quality of life are closely dependent upon the functionalities of built environments such as homes, workplaces, public infrastructure, and communities. In addition to the construction phase, the utilization and operation of such facilities can also substantially benefit from improved scene understanding methods that are enabled by techniques developed in robotics and enabling fields such as computer vision, deep learning, and inter-process communication. Some of these examples include construction robots [1,2], delivery robots [3], wheelchair robots [4], road crack detection [5], automatic bridge bearing inspection [6], asset and resource tracking [7], building performance modeling [8], and community resilience analysis [9].

With the aim of providing more efficient, accurate, and economical solutions, this dissertation explores and advances three categories of scene understanding capabilities that make up the fundamental building blocks of such applications.

The first category is geometric scene understanding, which, in this dissertation, represents the capability to map the geometry of an environment and determine an agent’s (e.g. robot’s) pose relative to a map of the environment. Specifically, in this research, localization, mapping and navigation are explored to support automatic data collection and task execution by enabling an agent to localize itself, map, and navigate through an environment.

The second category is semantic scene understanding, which, in this dissertation, represents the capability to semantically recognize different objects in an environment for agent-environment interaction. Specifically, in this research, 1D barcode marker-based object recognition is explored to extract 1D barcodes from video scan data for efficient and scalable asset and resource tracking.

The third category is system-level scene understanding, which, in this dissertation, represents the capability to systematically analyze interactions between different time-dependent and coupled built-environment processes that cannot be determined based on immediately perceived information. Specifically, in this research, inter-process communication-based distributed coupling analysis is explored to discover interdependencies between different analysis models for improved understanding of complex built-environment processes.

An overview of the research is depicted in Figure 1.1. The proposed scene understanding algorithms are not application-oriented but are generic and can be applied for a variety of applications. The feasibility and effectiveness of these methods are demonstrated and proved with specific implementations for selected applications in the built environment.

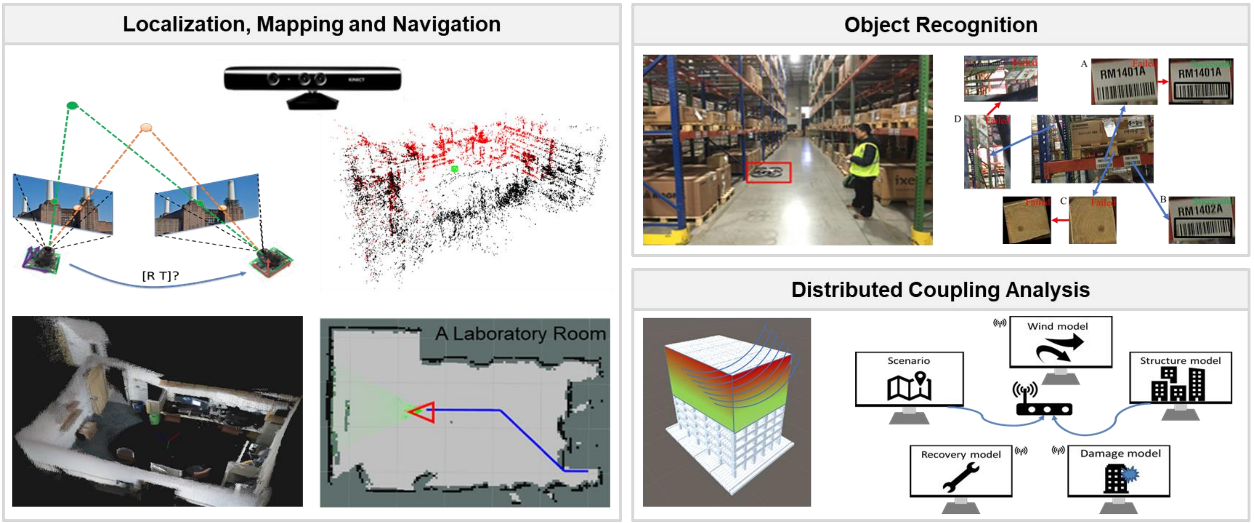


Figure 1.1 Research Overview

1.1 Importance of Research

Autonomy on construction sites and in the built environment has significant potential to liberate people from some repetitive, time-consuming, or dangerous tasks, which will not only benefit human health and safety but also improve efficiency and decision making. As the essential building blocks for such autonomy, scene understanding could be substantially improved by recent advancement in robotics and some enabling fields such as computer vision and inter-process communication.

Localization is an essential part of geometric scene understanding that allows an agent (e.g., person, or robot) to know its current pose (position and orientation) in an environment with reference to a map of that environment. This localization ability further allows an agent to incrementally update the map, navigate between different locations, and collect geo-tagged data for improved understanding of the environment and decision making. In Global Navigation Satellite System (GNSS)-denied environments, this technique is mostly integrated with mobile robots for a variety of indoor and outdoor applications. Such applications include delivery robots [3], wheelchair robots [4], vacuum cleaner robots [10], surveillance robots [11], robots for ambient data collection [12], robots for 3D point cloud modeling [13], tunnel inspection robots [14], bridge bearing inspection robots [6], and robots for inventory data collection [15]. Thus, this dissertation explores and provides a more economical and versatile locating solution that can benefit all the relevant applications, such as those enumerated above.

Besides localization, agents also need the fundamental capability of object recognition to interact with ambient environments to complete specific tasks. This capability is especially beneficial in dealing with high-volume, repeatable tasks such as book finding and picking in a library [16], building components assembly on construction sites [17], parcel sorting in logistics service centers [18], as well as item tracking, picking and delivery in warehouses [15]. Due to substantial appearance similarity, fiducial markers are generally utilized in well-organized environments for recognition purposes. In order to provide an economical and automatic marker reader to improve the above applications, this dissertation explores and designs a vision-based 1D barcode extraction framework and demonstrates its applicability and efficiency for automating inventory management in a warehouse.

With the capabilities of localization and object recognition, an agent is able to move around in an environment, make decisions and interact with the environment adaptively with instant information perception, which is suitable for the applications described above in this section where decisions can be made only by relying on immediate scene perception. However, there is another category of tasks for which decision making is dependent on multiple time-dependent and coupled factors that interact with each other in a complex way and cannot be simply achieved from immediate scene perception. Such tasks include optimizing the energy usage of buildings [8], analyzing deterioration of community buildings [19], estimating damage to buildings under strong winds [20], understanding human evacuation behavior in fire emergencies [21], and evaluating a community's resilience to natural hazards [22].

These processes usually include many elements and influencing factors, and it is too complex to be analyzed with a single analysis model (i.e., closed-from model, empirical model, or simulation-based model). Instead, such problems are generally solved by integration and coupling of different analysis models that can capture the involved determining factors and allow for decision making based on more accurate analyses. In order to support such an understanding of interdependent built-environment processes, this dissertation develops a distributed coupling analysis framework based on inter-process communication, which facilitates the development of complex coupling analyses and enables the discovery of interdependencies between involved factors for an improved understanding of the involved processes.

In all, this research seeks to explore the three categories of scene understanding: localization, object recognition, and distributed coupling analysis that serve as the fundamental blocks needed for a wide range of automation applications on construction sites and in the built environment. By improving relevant building blocks, this work is a critical and significant step toward automation in construction and built environments for improved infrastructure utilization and maintenance, human health, work efficiency, building performance, and community resilience.

1.2 Background and Literature Review

Based on the discussion above, this section provides an overview of relevant studies and their limitations. The literature review is organized into three broad categories. The first category is about localization-related techniques, the second category is object recognition, and the third is distributed coupling analysis. More detailed review and comprehensive analyses of the corresponding studies can be found in the following chapters, where the proposed algorithms in this dissertation are discussed.

1.2.1 Localization

For an agent (e.g., human or robot), localization here is defined as the capability of determining the current location (position and orientation) with respect to a map [23]. With a map, a constraint-based path planning algorithm (such as A* [24] or RRT [25]) can be used to calculate an optimal path from an agent's current location to a destination considering both obstacle avoidance and trajectory cost. With the localization ability and such a planned trajectory, an agent can avail of turn-by-turn instructions to follow the trajectory and navigate to the destination. Therefore, localization, path planning, and navigation are usually utilized together to enable an agent to navigate through an environment.

In most outdoor environments, GNSS (Global Navigation Satellite System) solutions, such as GPS, GLONASS, Galileo, and Beidou, are widely available and used together with web mapping services such as Google Maps and Apple Maps for outdoor localization and navigation tasks. However, GNSS suffers from problems of signal loss and multipath signal and cannot provide accurate localization for robotic applications, especially in urban or indoor environments where GNSS signals are usually blocked or multi-reflected.

Compared with outdoor environments, localization in GNSS-denied environments is much more challenging. The first attempts to achieve such localization are via wireless techniques, including Wireless Local Area Network (WLAN), radio frequency identification device (RFID), Ultra-Wideband (UWB), Bluetooth, and ultrasound. By taking advantage of widespread existing WiFi Networks, WLAN is a relatively economical solution. However, the number, distribution, and quality of existing WiFi access points may not always satisfy the localization requirement [26,27] and still need additional deployment. Even though RFID is cost acceptable, it suffers from large errors in environments with commonly encountered metal or liquid materials that influence absorption and reflection of radio waves [28,29]. Both UWB and Bluetooth are very expensive for large-scale deployment because UWB requires extra hardware on different user devices [30], and Bluetooth requires a large number of pre-installed beacons [31,32]. The ultrasound is less affected by the environment, but its accuracy is seriously impacted by the placement of the emitter sensors [33]. Moreover, all these methods need labor-intensive and time-consuming work to deploy and calibrate transmitters, which prevents them from being rapidly deployed and conveniently utilized in a large-scale environment.

GNSS-denied localization can be also achieved without environment instrumentation. Fixed cameras were used in [34-36] for 3D object localization, but these solutions are limited by the field of view and scene occlusions. Artificial fiducial markers [37] were also used for indoor localization, but this approach needs a dense marker network to guarantee the localization accuracy and cannot scale well to large-scale environments. Inertial sensors provide another alternative solution, but these dead-reckoning methods suffer from drift accumulation over time and distance [38]. By unifying the process of localization and map creation, Simultaneous Localization and Mapping (SLAM) techniques can simultaneously estimate the pose of the sensor and recover the structure of the environment in a map format. Compared with inertial sensors, SLAM allows for recognition of revisited places and correction of the drift introduced into pose estimation and the associated as-built map. Based on the types of sensors, SLAM can be classified into two categories, Lidar-based SLAM and visual SLAM (vSLAM).

Lidar-based SLAM solutions, such as Hector SLAM [39], GMapping [40,41], and Cartographer [42], use Lidar sensors for environment sensing and are usually considered robust in most environments. The main limitation is that they rely on expensive Lidar sensors, which makes them unsuitable for widespread applications in built environments. Moreover, Lidar sensors cannot provide semantic information such as room number, words on a sign, semantic object recognition, or object color that are critical for objection recognition and interaction with the environment.

Compared with Lidar-based SLAM, vSLAM only needs low-cost cameras and can achieve comparable localization performance when a deployment is suitable [43]. In vSLAM, feature-based SLAM methods [44-46] generally achieve higher localization accuracy than direct SLAM methods [47-50]. However, they can only provide a sparse point cloud map that is efficient for localization but does not include enough information to support path planning and navigation tasks. Moreover, vSLAM's robustness varies in different environments and suffers from performance loss in challenging environments with motion blur, low structure, or low texture.

In order to provide a non-intrusive, low-cost, and versatile localization system that can be rapidly deployed, this research seeks to advance vSLAM by addressing its two main limitations.

1.2.2 Object Recognition

In computer vision, object recognition, as a means of semantic scene understanding, is generally referred to as the combined tasks of object classification that decides the class label of a given object and object localization that identifies the location of one or more objects in an image and draws a bounding box around each object.

Object recognition is generally achieved by feature-based methods or marker-based methods. As the state-of-the-art feature-based methods, with the development of neural networks, two deep learning model families (R-CNN family [51-53] and YOLO family [54-56]) have achieved very competitive accuracy on different image datasets. Such feature-based methods could be a promising direction leading to general object recognition. However, for now, most of them are still limited in two aspects. First these methods are good at classifying objects, but they cannot tell the real difference between the objects in one class. For example, it is easy for them to distinguish a book and a desk, but they will not discern the difference between two different books. Sometimes, this information is very important for asset management in environments such as libraries or warehouses. Moreover, it would take a significant amount of effort to allow them to distinguish all the objects, even simply different books, because much more training data needs to be prepared to allow for such tasks.

Object recognition can also be solved with marker-based methods in which the information attached with a marker can be retrieved when a marker is decoded. Many fiducial markers have been developed originally for determining the relative pose between a marker and a camera, and those makers can be also used for object recognition purposes. Among these fiducial markers, AprilTags [57] and ArUco libraries [58] are the most widely used. As such an example, AprilTags were used in [17] for recognizing building components. These markers allow for very accurate object recognition. However, in practical usage, it is very labor-intensive and time-consuming to attach such a marker to each object to be recognized. Compared to these additional markers, the 1D barcodes printed on each product are specially designed for object recognition.

In order to take advantage of such existing information, some vision-based barcode readers have been developed to replace barcode scanners to improve recognition efficiency. Initially, the barcode reading algorithms were mainly implemented based on domain transformation [59] or scanlines [60-62]. In addition, there are also some algorithms developed to read challenging barcodes caused by low resolution and blurring from motion or being out of focus [63,64]. However, most of these algorithms are only applicable to vertical or approximately vertical barcodes. Even though some commercialized algorithms, such as the ClearImage Barcode Reader SDK [65], provide certain abilities to read rotated barcodes from an image, their performance is significantly limited for blurred images.

Instead of focusing on general object recognition, this research seeks to advance 1D barcode extraction methods for improved asset tracking or inventory management in large-scale environments such as libraries, warehouses, and distribution centers.

1.2.3 Distributed Coupling Analysis

As an important tool for understanding complex processes in the built environment, different analysis models have been widely used in different fields. For example, in earthquake engineering, several models have been developed to model various effects of an earthquake on civil infrastructure [66,67]. Similarly, several models exist for power system [68], transportation [69], human response under disasters [21,70,71], evacuation plans [72], emergency response training [73], and post-disaster recovery [74]. However, these are all independent analysis models that cannot reveal complicated interdependencies between the built environment (e.g., buildings and bridges), critical infrastructure systems (e.g., lifelines and telecommunication), social and non-physical systems (e.g., politics and economics).

In order to interpret the deep interdependencies between involved factors and obtain more reliable results, it is necessary to consider the models' coupling effects in an analysis. Due to the complexity of such an analysis, such problems are generally solved by distributed coupling analysis enabled by inter-process communication [75]. The widely used standards and platforms for distributed analysis include Distributed Interactive Simulation (DIS) [76], High Level Architecture (HLA) [77], Test and Training Enabling Architecture (TENA) [78] and DDS [79]. However, both DIS and TENA have to use pre-defined sets of messages, which is not flexible for exchanging messages that are not defined in the standards. Moreover, they can only build real-time analyses that run in wall-clock time. Compared with DIS and TENA, HLA and DDS, are more flexible to use. However, it is still difficult for a novice to rapidly perform a functional distributed analysis and, for experienced users, non-trivial to achieve desired efficiency performance.

In the context of distributed coupling analysis-based system-level scene understanding, this research seeks to design an easy-to-use distributed coupling analysis framework based on inter-process communication that enables users to easily integrate their domain analysis models with models from other domains for constructing more accurate and complex analyses to discover the complex interdependencies between involved processes.

1.3 Objectives

The overall objective of this research includes improving vSLAM for economical and versatile localization in built environments, designing a 1D barcode extraction framework for convenient and efficient asset tracking and inventory management, and designing a distributed coupling analysis framework for convenient and rapid coupling of interdependent process models to understand complex processes in built environments. The specific objectives of this research are organized into three categories as follows.

1. Improved vSLAM algorithms for locating applications
    - Develop and evaluate a vSLAM-based locating system for path planning, navigation, geo-tagged data collection, and 3D point cloud reconstruction in built environments
    - Develop and test a fiducial marker-based algorithm to evaluate the localization accuracy of a vSLAM without using a motion capture system
    - Test and characterize the localization performance of a proposed locating system to assess its applicability in GNSS-denied environments
    - Develop and evaluate a self-adaptive learning-based descriptor for the development of optimized descriptors for vSLAM applications

- Test and characterize a proposed self-adaptive descriptor in a vSLAM for robust feature matching for applications in challenging environments

2. 1D barcode extraction for asset tracking
   - Design a drone-assisted asset scan framework for automatic asset tracking
   - Develop a 1D barcode extraction framework to extract 1D barcodes from video scan data collected in large-scale environments
   - Test and characterize a 1D barcode extraction framework to assess its applicability in a warehouse or a distribution center for inventory management

3. Distributed coupling analysis for deep interdependency discovery
   - Review and investigate existing standards, platforms, and standalone data passing tools that can be used for distributed analyses of built-environment processes
   - Develop and evaluate a distributed coupling analysis framework and a message wrapper for domain users (such as researchers in resilience and disaster engineering) with limited background in distributed coupled analysis to conveniently integrate and couple their domain analysis models for complex built-environment analyses

Together, these objectives contribute to geometric, semantic and system-level scene understanding and improve construction and the utilization, operation, maintenance, and understanding of built environments.

1.4 Methodology

The methodology adopted in this research is mainly based on techniques from robotics and computer vision, and existing methods are extended or improved by addressing the research gaps both in fundamental methods and in application domains.

Even though some specific case studies and domain-specific applications are used to illustrate the technique details and demonstrate the effectiveness, the proposed algorithms and frameworks are generic scene understanding solutions and can be utilized in relevant applications. For example, by using appropriate platforms, the proposed localization system can be used for automatic joint filling on construction sites as discussed in [80], automatic point cloud registration for construction progress monitoring [13], automatic bridge bearing inspection [6], and automatic tunnel inspection [14], and so on. The designed barcode extraction framework can be also easily modified for applications such as book handling in a library [16], building components manipulation on construction site [17], and parcel sorting in a logistics center [18], even though inventory application in a warehouse is used as an example application. This also applies to the developed distributed coupling analysis framework, which can be used to link any domain analysis models even though analysis models in wind engineering and structural engineering are used to explain the detailed design.

1.5 Dissertation Outline

This dissertation is a compilation of peer-reviewed scientific manuscripts that explore three categories of scene understanding: localization, object recognition, and distributed coupling analysis. The remainder of the dissertation is organized as follows.

Chapter 2 demonstrates the development of a locating system based on an occupancy grid mapping enhanced vSLAM that supports path planning and navigation for practical tasks. The proposed locating system is evaluated with a fiducial marker network, and three examples are given to illustrate its applications.

Chapter 3 describes the development of a self-adaptive descriptor to improve the robustness of locating systems based on feature-based vSLAM. The proposed descriptor is tested offline and integrated into a vSLAM by replacing its feature related parts. Two widely used public datasets are used to evaluate the modified vSLAM's localization performance for locating applications.

Chapter 4 describes the development of a barcode extraction framework that supports automatic 1D barcode extraction for asset tracking. Inventory management in a warehouse is used as a case study to demonstrate the accuracy and efficiency of the proposed framework.

Chapter 5 describes the development of a distributed coupling analysis framework and a message wrapper that support convenient integration of analysis models from different domains. Its application in wind engineering and structural engineering is used as a case study to demonstrate its convenience and effectiveness.

Lastly, Chapter 6 provides a summary of this research, including its contributions and future directions.

Chapter 2

An Occupancy Grid Mapping Enhanced Visual SLAM for Real-Time Locating Applications in GNSS-Denied Environments

2.1 Introduction

The burgeoning demand for robotic applications to support key construction and facility management functions is creating a strong need for deployable mobile robots that are capable of performing assigned tasks at specific locations automatically. Examples of such mobile agents include data collection robots [13,15,37,81], infrastructure inspection robots [6,14], indoor service robots [82-84], construction robots [1,17,85,86], or even some robots that can move in complex environments for versatile applications [87,88]. Among all the fundamental technical capabilities that make such autonomous robots possible, Real-Time Locating Systems (RTLS) are indispensable because they allow robots to estimate their own pose (position and orientation) with respect to maps of the environment. RTLS have been extensively utilized to facilitate and improve safety management [89-91], construction resource tracking [92,93], infrastructure inspection [14,94], and progress monitoring [13,95].

Compared to outdoor localization systems that can take advantage of the widely available Global Positioning System (GPS), indoor localization in GPS-denied environments is relatively more challenging. Even though significant research efforts invested in wireless technologies-based indoor RTLS (e.g., Wireless Local Area Network (WLAN), radio frequency identification device (RFID), Ultra-Wideband (UWB), Bluetooth, and ultrasound), their requirements on dedicated hardware and environment instrumentation inevitably prevent them from being widely deployed in large-scale indoor environments [96-98]. As a promising alternative, vision-based RTLS solutions have also been explored [34,99]. However, they either depend on pre-installed fixed cameras which cannot adequately handle inevitable occlusions that occur in typical indoor environments or need computation-intensive structure from motion (SfM) and thus cannot run in real time. More recently, 2D Lidar-based Simultaneous Localization and Mapping (SLAM) has started to receive attention from researchers studying RTLS for unstructured indoor environments such as construction sites, for mapping or navigation applications [13,14]. However, those methods typically need the user input of an initial pose estimation for them to start working correctly. This is inconvenient and often infeasible when such a prior pose estimation is not available.

In order to overcome these limitations and provide a versatile indoor RTLS, this paper proposes a Visual SLAM (vSLAM)-based localization system that is suitable for a wide range of applications in indoor, GPS-denied environments. In this system, an additional OGM is built side by side with the sparse feature map of ORB2 RGBD and enables interaction with users and path planning that cannot be supported by ORB2 RGBD. In addition, the proposed RTLS does not need any environment instrumentation or rely on any existing artificial facilities, which makes its rapid deployment possible. More conveniently, it also provides visualization tools that allow users to monitor the pose of the tracked object and interact with the system intuitively.

2.2 Review of Related Prior Research

This section reviews three types of existing localization methods.

2.2.1 Wireless Technology-Based Localization Approaches

The primary investigated and applied approaches in this category include WLAN, RFID, UWB, Bluetooth, and ultrasound. WLAN is a relatively economical solution since in most cases it can be directly built on widespread existing WiFi networks and WiFi access is readily available on most mobile devices today [100]. However, it is difficult to guarantee that the number, distribution, and quality of existing WiFi access points can always satisfy the localization requirement [26,27] and it cannot be applied in unprepared environments such as construction sites where WiFi is not typically available.

Due to its wide acceptance in industry and its acceptable cost, RFID has gained significant attention for highly dynamic environments such as construction sites and has been adopted in multiple applications to improve the construction process [28,101-103]. However, RFID requires considerable effort to deploy a large number of tags in a large-scale environment. In addition, RFID readings are vulnerable to be influenced by absorption and reflection of radio waves by commonly encountered metal or liquid materials [28,29].

Although UWB is more immune to signal interference compared with RFID, it requires extra hardware on different user devices that is too expensive for large-scale deployment [30]. For Bluetooth, there is a need to pre-install a large number of beacons, and the localization accuracy depends heavily on the number, sizes, and shapes of the localization cells [31,32]. The ultrasound has a negligible penetration of walls and is thus less affected by the environment [104]. However, its localization accuracy is highly dependent on the placement and position calibration of the emitter sensors [33].

Recent research has also attempted to integrate Building Information Modeling (BIM) with one or more of these wireless technologies to obtain better performance [105-107]. However, the limitation is that a prior BIM is not always available, and it is challenging to update a BIM in real-time for localization purposes. In general, all of the reviewed wireless technology-based approaches need varying degrees of instrumentation of the environment and cannot be conveniently and rapidly deployed, especially in large-scale environments.

2.2.2 Vision-Based and Inertial Sensor-Based Localization Approaches

Compared with wireless technology-based approaches, vision-based methods only need to use common and economical cameras as perception sensors and require little instrumentation of the environment. The current 3D tracking solutions in dynamic environments such as construction sites were achieved by 2D tracking of the same objects of interest in two or more cameras and 3D triangulation of the tracked 2D observations [34-36]. However, such solutions depend on fixed cameras deployed in such environments with large baselines, which need complex camera calibration processes, have a fixed field of view, and suffer from inevitable occlusions caused by equipment or temporary structures.

A mobile camera-based solution was proposed in [99], where a point cloud of a site was created by SfM with images collected by a drone and then the objects localized in a drone image could be found in the point cloud by feature matching. However, SfM itself and recovery of camera pose with a whole SfM image set are both time-consuming, and this solution cannot run in real time and provide as-is tracking for a site. Another alternative mobile camera solution that can run in real time is to use artificial fiducial markers [37]. The problem is that this approach needs a dense marker network to guarantee the localization accuracy and is difficult to be applied in large-scale environments. Inertial sensors offer another economical and non-instrumented solution. However, such dead-reckoning methods suffer from drift error accumulation over time and distance [38].

2.2.3 SLAM-Based Localization Approaches

Different from the above methods which need to rely on existing floor plans of a building or environment as a reference map, SLAM allows the incremental construction of a map of an unknown environment while simultaneously inferring an agent's location within the as-built map [108,109]. After a complete map is built, localization-only algorithms can run on the map and the real-time pose of the agent can be determined with respect to the map. Based on the primary perception sensor, SLAM can be mainly divided into two categories, light detection and ranging (Lidar) based SLAM and vSLAM.

2.2.3.1 Lidar-Based SLAM

Due to its high accuracy and an ever-growing number of open-source implementations, Lidar-based SLAM has gained increasing attention from researchers, particularly those focused on localization and mapping in dynamic indoor environments such as construction sites. In [13], an algorithm to determine good scan positions was developed on the Occupancy Grid Map (OGM) built by Hector SLAM [39]. The poses estimated by Hector SLAM at the chosen scan positions were used to align and register the corresponding laser scans. An OGM is a map of the environment represented as an evenly spaced field of binary variables each representing the presence of an obstacle at that location in the environment [110]. In [6], Hector SLAM was used to build an OGM and provide odometry input to the Adaptive Monte Carlo Localization (AMCL) algorithm for further localization and navigation.

Compared to GMapping [40,41] that needs additional odometry input, the advantage of Hector SLAM is that it only requires Lidar measurement and can also support the Inertial Measurement Unit (IMU) based tilt compensation. However, it does not detect loop closure and thus cannot reduce accumulated drift by loop closing. Another disadvantage is that further mapping cannot be performed incrementally on a map saved by a previous mapping process, which is inconvenient to map a large-scale environment.

Cartographer is another Lidar-based SLAM solution that was developed based on [42] and made open source by Google in 2016. This solution supports both loop closure ability and IMU input. However, since it has many parameters that affect each other, it is usually difficult to tune the system to get acceptable performance. Despite its advantages, Lidar-based SLAM also suffers from certain limitations. The key limitation is that laser scanners are still cost-prohibitive today and that makes it infeasible for widespread deployment in near-term applications. Besides, although global localization (the task of estimating an agent's pose without any prior knowledge) is possible with Markov Localization [111] or Monte Carlo Localization (MCL) [112], the localization error is inevitable in geometrically similar environments since such algorithms can only take advantage of unique geometry information in the environment.

2.2.3.2 vSLAM-Based Localization Approaches

vSLAM uses one or multiple cameras as primary perception sensors, which is generally economical compared to laser scanners, while also providing competitive localization performance on several datasets [113]. Another benefit is that vSLAM can run with only frame observations, even though additional odometry or IMU input can help further improve accuracy and robustness.

This implies that all the hardware preparation for using a vSLAM solution is simply to mount one or more ordinary cameras on an existing mobile platform or the objects to be tracked.

However, there are some concerns about the use of vSLAM in complex environments, such as vulnerability to illumination variations, weather conditions and seasons, the difficulty of use in low texture/structure/dynamic environments, and challenges to work under motion blur. Upon comprehensive review, it was found that such issues have already been significantly improved and continue to be better solved with the ongoing development of new algorithms [114-119] and upgrades in new camera hardware [120,121].

vSLAM falls into two categories, direct SLAM that tracks a new frame by directly optimizing over pixel intensities [48-50], and feature-based SLAM that infers the pose of a new frame by extracting sparse features from it, matching them to the features in the last frame or a local map, and optimizing reprojection errors [44-46]. Even though state-of-the-art feature-based methods have better localization performance than direct methods [46,122], the map they create is so sparse that it is only useful for localization but is unusable for path planning or interaction with users.

ORB2 (Red-Green-Blue-Depth) RGB-D SLAM (ORB2 RGBD) [122] is a state-of-the-art feature-based SLAM. By using ORB features that can be much more efficiently extracted and has comparative matching performance compared with SIFT or SURF [123], it can perform frame-rate relocalization and loop detection. With a Kinect camera sensor, it can provide real-time tracking of the camera pose and compute a sparse 3D reconstruction of the environment with true scale. The sparse 3D reconstruction is referred to as the sparse feature map and includes the 3D points corresponding to matched feature points and keyframe poses. Keyframes are the frames that are selected and used as reference frames to represent or localize the pose of other frames. ORB2

RGBD includes three threads, tracking, local mapping, and loop closing. These keyframes are first selected and inserted in the tracking thread, and then are culled in the mapping thread. In the tracking thread, the current frame would be inserted as a keyframe only if enough frames have passed since the last relocalization with the last keyframe, enough frames have passed since the last keyframe insertion, the current frame tracks enough points from the last frame but the tracked feature number is less than a certain percent of the feature number that it can track from its most recent keyframe. The specific thresholds can be found in [46], and this loose keyframe insertion strategy helps to improve tracking robustness. In the local mapping thread, a keyframe would be discarded if 90% of the map points that are observed in the keyframe can be observed by at least 3 other keyframes. This culling strategy further culls redundant keyframes, maintains a sparser keyframe network, and improves the quality of map points and keyframe poses. ORB2 RGBD also supports loop closure (recognition of pre-visited places) that allows correcting the as-built sparse feature map and global re-localization that allows re-localizing the pose of the camera in the sparse feature map without any prior position information. This SLAM algorithm has been extensively tested on different datasets and has demonstrated promising localization results compared with other state-of-the-art direct SLAM and feature-based SLAM algorithms [122].

However, the sparse feature map built by ORB2 RGBD is not intuitive to users, does not support path planning for practical applications, and does not allow user interaction with the SLAM algorithm. To take advantage of its high localization accuracy while overcoming the drawbacks of its sparse feature map, this paper proposes an RTLS based on ORB2 RGBD that is able to build an additional OGM and localize the 2D the camera pose in that OGM. The availability of the built OGM further enables applications such as path planning, geo-tagged data collection, and location-aware 3D point cloud update, making the approach versatile for a variety of indoor RTLS applications.

2.3 Research Objectives, Scope and Contribution

The primary objective of this research is to design a vSLAM-based RTLS that can provide high localization accuracy and can be deployed quickly and conveniently. The designed RTLS should also overcome the main limitations faced by existing methods, such as the requirement of labor-intensive and time-consuming environment instrumentation (wireless technology-based methods), fixed field of view (fixed cameras), trajectory drift over time and distance (inertial sensors), high cost (Lidar-based SLAM), and lack of necessary maps for path planning and interaction with users (feature-based SLAM).

The scope of this research includes improvement of ORB2 RGBD with occupancy grid mapping, 2D pose localization, visualization of real-time 2D camera pose and virtual laser scan on the built OGM for practical applications, and ROS [124]-based communication between different components in the localization system. This research also includes the design of a localization accuracy evaluation method based on fiducial markers. In the paper, only indoor RGB-D SLAM based on Kinect is discussed and evaluated in the paper even though the design of the system can be applied to other RGB-D or stereo cameras for outdoor applications.

The specific contributions of this research are as follows:

- A new RTLS is designed based on an OGM enhanced vSLAM algorithm, which can provide high-accuracy localization on the OGM and enable user interaction with the localization system and practical applications, such as path planning, real-time navigation, geo-tagged data collection, and location-aware point-cloud update.

- The proposed RTLS can work in two modes (SLAM mode and localization mode) and switch between the two modes flexibly as needed. This allows to update an existing map or incrementally build a larger map upon a map built previously.
- The proposed RTLS uses jointly a sparse feature map and an occupancy grid map and capitalizes on their respective advantages.
- A new fiducial landmark-based method is designed to evaluate the localization accuracy of the proposed system.
- Three example applications are described to demonstrate the benefits of the proposed system.

2.4 Technical Approach

The proposed RTLS can work in two modes, SLAM mode and localization mode (Figure 2.1). For an environment in which there does not exist a map or the existing map is not proportionally accurate, or the map requires frequent update due to dynamic changes, the RTLS runs in the SLAM mode to creates a new map of the environment from the beginning or incrementally update the existing map.

The map built in the SLAM mode includes a sparse feature map and an additional OGM. The sparse feature map is composed of the poses of the keyframes and the map points that can be observed by at least 3 keyframes [46]. It is built incrementally by ORB2 RGBD using an RGB frame and its corresponding depth image obtained from a Kinect sensor at the same time. However, the geometric information included in this sparse map is not adequate to be useful for path planning or navigation purposes. In order to address such issues, the OGM is built at the same time and serves as an extension of the sparse map. Without using a laser scanner, this OGM is built with the pose estimated by ORB2 RGBD and the corresponding virtual laser scan created from the point cloud observed by the Kinect. The two maps are saved for localization in the same environment when the SLAM process completes.

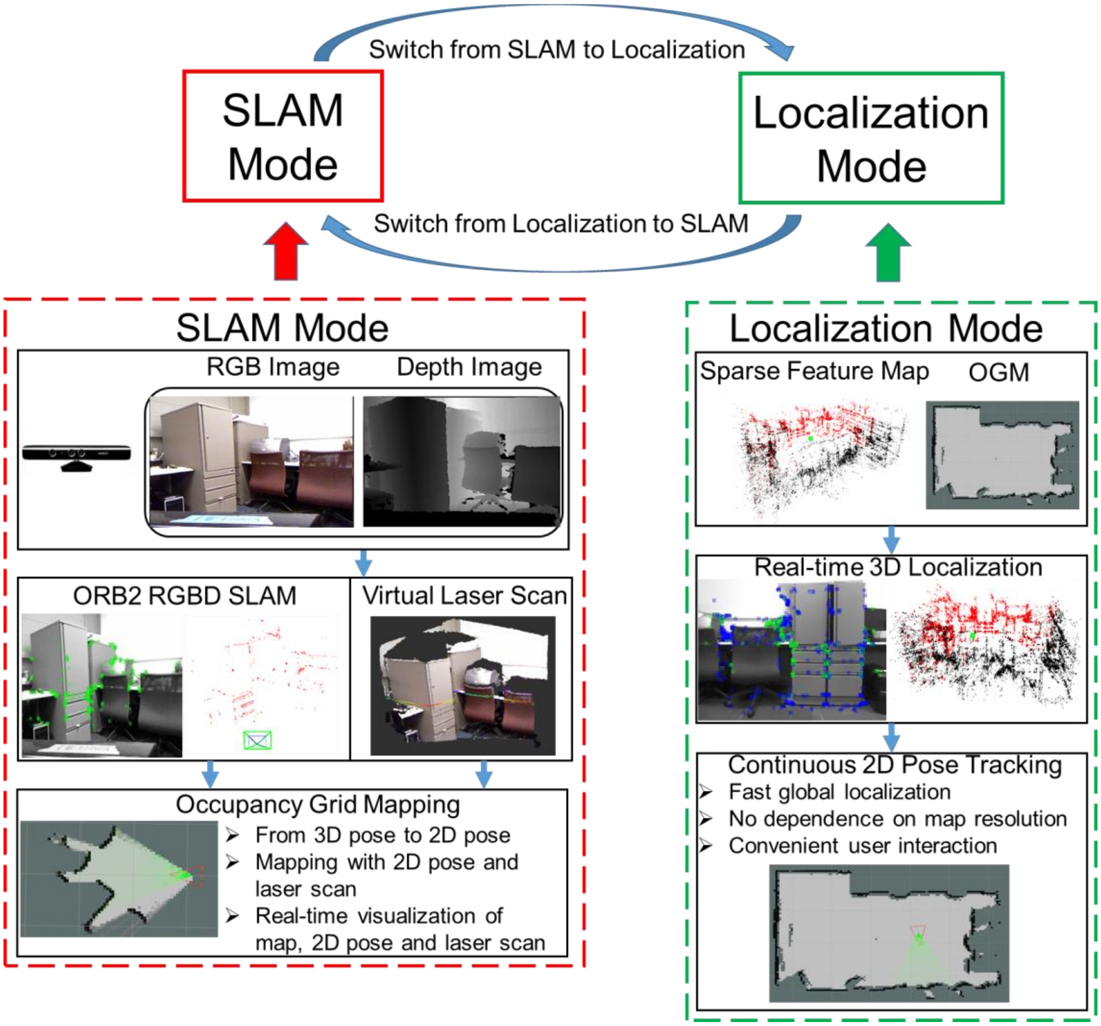

Switch from SLAM to Localization
SLAM Mode
Localization Mode
Switch from Localization to SLAM
SLAM Mode
RGB Image
Depth Image
ORB2 RGBD SLAM
Virtual Laser Scan
Occupancy Grid Mapping
➢ From 3D pose to 2D pose
➢ Mapping with 2D pose and laser scan
➢ Real-time visualization of map, 2D pose and laser scan
Localization Mode
Sparse Feature Map
OGM
Real-time 3D Localization
Continuous 2D Pose Tracking
➢ Fast global localization
➢ No dependence on map resolution
➢ Convenient user interaction

Figure 2.1 Overview of the proposed RTLS

When working in the localization mode, the system loads both the sparse map and the OGM first. Then, ORB2 RGBD converts the descriptors of the key points extracted from the input RGB frame into their bag-of-words (BoW) representations [122,125] and queries the keyframe database for the initial pose estimation of the RGB frame. This process continues until the pose of the current frame is initially estimated and finally optimized and global localization is successful. Subsequently, the 3D pose of a new frame in the sparse map can be tracked by tracking its previous frame (or the most recent reference frame) and its local map. Finally, the 2D pose in the OGM can be found by projecting the 3D pose onto the 2D plane and continuous 2D pose tracking can be achieved.

It is worth noting that the system is designed to be able to switch between the two modes (SLAM mode and localization mode) at any time (Figure 2.1), which provides additional flexibility for a wide range of applications. As mentioned before, one key drawback of hector SLAM is that new map change cannot be updated directly on an old map. This issue can be easily resolved in the proposed system by first using localization mode to localize the current pose and then switching to SLAM mode for further expansion of the existing map.

In addition, the system leverages the complementary advantages of the sparse map and the OGM by using them jointly. On one side, a sparse map is compact but not useful for path planning and user interaction, which can be complemented by an OGM which is well-suited for such tasks. On the other side, OGM based localization depends significantly on the map resolution (dimension of each grid) and therefore imposes a high memory cost for large-scale environments, which can be complemented by localizing with a sparse map that needs lesser memory for map storage. Therefore, the localization accuracy does not depend on the resolution of the OGM and grid size can be relaxed when the map scale goes up. With these features, the proposed system enables a broad range of applications in path planning, geo-tagged data collection, and location-aware 3D point cloud update that are discussed ahead in Section 2.6. The following subsections first describe the technical design of the system in detail.

2.4.1 SLAM Mode

In order to promote maintainability, code re-use and extensibility of the system, ROS [124] is extensively used in the implementation for transferring data among different components where each component itself is designed to be an ROS package. As shown in Figure 2.2, there are mainly four components involved in the SLAM mode (“Kinect Driver”, “Point Cloud to Laser Scan”, “Modified ORB SLAM” and “Occupancy Grid Mapping”) and thus four ROS packages. Among these packages, the first two are existing ROS packages with some custom configuration, the third package is developed upon ORB SLAM [122], and the last package is implemented entirely in this research. After a SLAM process, the system saves both a sparse feature map and an OGM. The two maps are further reused for localization in the localization mode. The two maps shown in Figure 2.2 were built when the system ran in a typical university laboratory room.

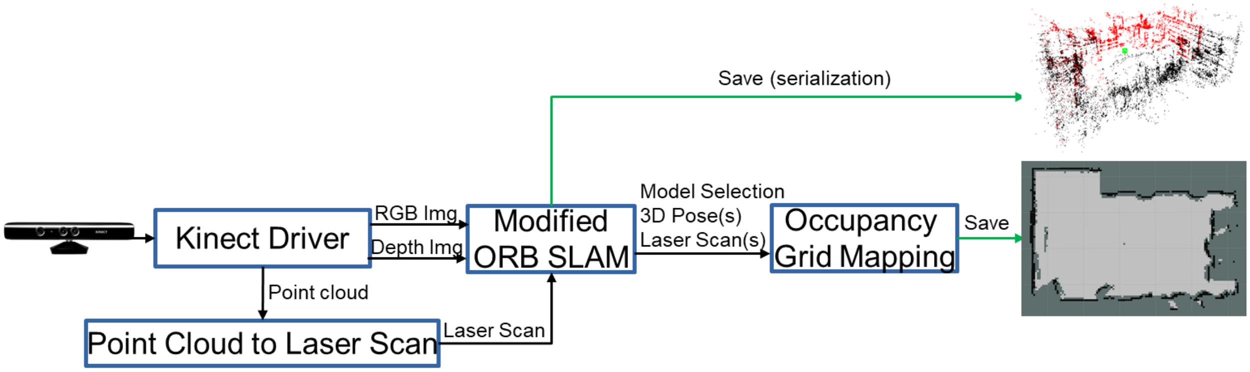


Figure 2.2 SLAM mode of the system

2.4.1.1 *Kinect Driver*

With some configuration changes, the ROS openni_launch package [126] is directly used here as the “Kinect Driver”. As an RGB-D sensor, Kinect can provide RGB images and corresponding depth images simultaneously. However, the RGB images and the depth images generally cannot overlap perfectly due to any existing offset between them. In the developed implementation, the ROS openni_launch package is configured to align the depth image to its RGB image, by setting the depth_registeration argument to true in the command line or enabling it via the ROS rqt_reconfigure tool. After the argument is set correctly, the depth of each pixel in the RGB image can be obtained by finding the value of its counterpart pixel in the registered depth image. Furthermore, a point cloud can be created by looping over all the color pixels to get their depth and unprojecting them from 2D space to 3D space.

In this configuration, the driver automatically publishes different messages to their corresponding topics. Among these messages, only three are further used in the system, of which the RGB image message and the registered depth message are used subsequently by “Modified ORB SLAM” to track the Kinect’s pose in 3D space, and the point cloud message is used by “Point Cloud to Laser Scan” to create virtual laser scans. Instead of using the rectified color images provided by the driver, the unrectified images are input to the “Modified ORB SLAM” since ORB SLAM uses its built-in model to remove distortion from the original image based on camera calibration parameters.

2.4.1.2 Point Cloud to Laser Scan

In most Lidar-based SLAM algorithms, measurements from laser scanners are used alternately to update an OGM based on the tracked pose and to track the pose based on the as-built map [39,41,42]. The difference here is that only a Kinect sensor is adopted in the proposed system and the pose is localized in “Modified ORB SLAM” with color and depth images. There are therefore no direct laser scan measurements available to update the OGM. To address this issue, the pointcloud_to_laserscan ROS package [127] is adopted to convert the point cloud received from

“Kinect Driver” to its corresponding virtual laser scan by cutting out a horizontal slice of the point cloud in a certain height range and selecting the points that have the smallest depth in each column of the slice.

The virtual laser scan created in this way allows the detection of all the obstacles appearing in a height range instead of only being able to detect the obstacles at a fixed height when a real 2D laser scanner is used. This benefit is critical in some situations where an obstacle can only be found by the laser scan at an appropriate height. For example, a desk with legs is very likely to be missed unless a laser scanner is installed at the same height as the desk surface, and this may cause potential collisions between the desk and a moving robot. However, this issue can be readily resolved with the virtual laser scan by setting appropriate min_height and max_height parameters for the node to specify the height range from which obstacles should be detected. There are some other parameters that can be set to control the generation of the virtual laser scan. Such parameter information is also included in the output laser scan message and can be retrieved when these messages are used to update the OGM in “Occupancy Grid Mapping”.

It should also be noted that the accuracy of the virtual laser scan generally has poor quality compared to that from a laser scanner and may not be adequate to use in pure Lidar-based SLAM. However, in the proposed system, the achieved level of quality in the virtual laser scan is sufficient since it is only used for OGM mapping, and localization is achieved by ORB SLAM based on the as-built feature sparse map.

2.4.1.3 Modified ORB SLAM

This component is developed upon ORB2 RGBD [122] when it runs with both its mapping and localization functionalities. An RGB image and its depth map are all that ORB SLAM needs to track the pose of the camera in which the captured scene is observed. Even though generally ORB SLAM can track current camera pose quickly, it still takes some time and the time it takes varies depending on how easily the initial pose can be estimated with feature matching between the current RGB frame and its immediate previous frame (or its keyframe when tracking sequential frames does not work). However, the quality of the OGM relies heavily on the synchronization of the camera pose and the laser scan associated with that pose.

If the 3D pose and laser scan were sent separately to the “Occupancy Grid Mapping”, acceptable synchronization between them would not be guaranteed due to the time latency caused by pose tracking. In order to solve this issue, the RGB image, depth image, and laser scan data are instead synchronized by “Modified ORB SLAM” at an earlier stage and sent together to the “Occupancy Grid Mapping” in a single custom ROS message (Figure 2.2 and Figure 2.3). This message is named “PosesAndLaserScans” as shown in Figure 2.3 and defined to include a one-dimensional array of geometry_msgs/PoseStamped [128] to represent poses, and a one-dimensional array of sensor_msgs/LaserScan [129], with the same number of elements as the pose array, to represent the corresponding laser scans at those poses.

As shown in Figure 2.3, after the current camera pose has been tracked, for occupancy grid mapping, all the necessary information is available about historical keyframes, their corresponding laser scans, as well as the current frame. However, additional strategic steps are necessary in order to use such information usefully in the implementation. Based on the features of the ORB SLAM, two special strategies are adopted to guarantee the quality of the OGM as introduced in [130].

First, instead of each frame pose, only the keyframe pose is used to determine the scan points in the OGM mapping process. It is worth noting that ORB SLAM is a keyframe-based method and only a subset of frames are selected as keyframes to reduce the number of frames used to represent the camera's motion while still being able to cover the whole scene visited by all the original frames. After initially being inserted by the tracking thread, the keyframe poses are further optimized in the mapping thread and in some cases even further, in the loop closing thread when a loop involving them is detected and corrected.

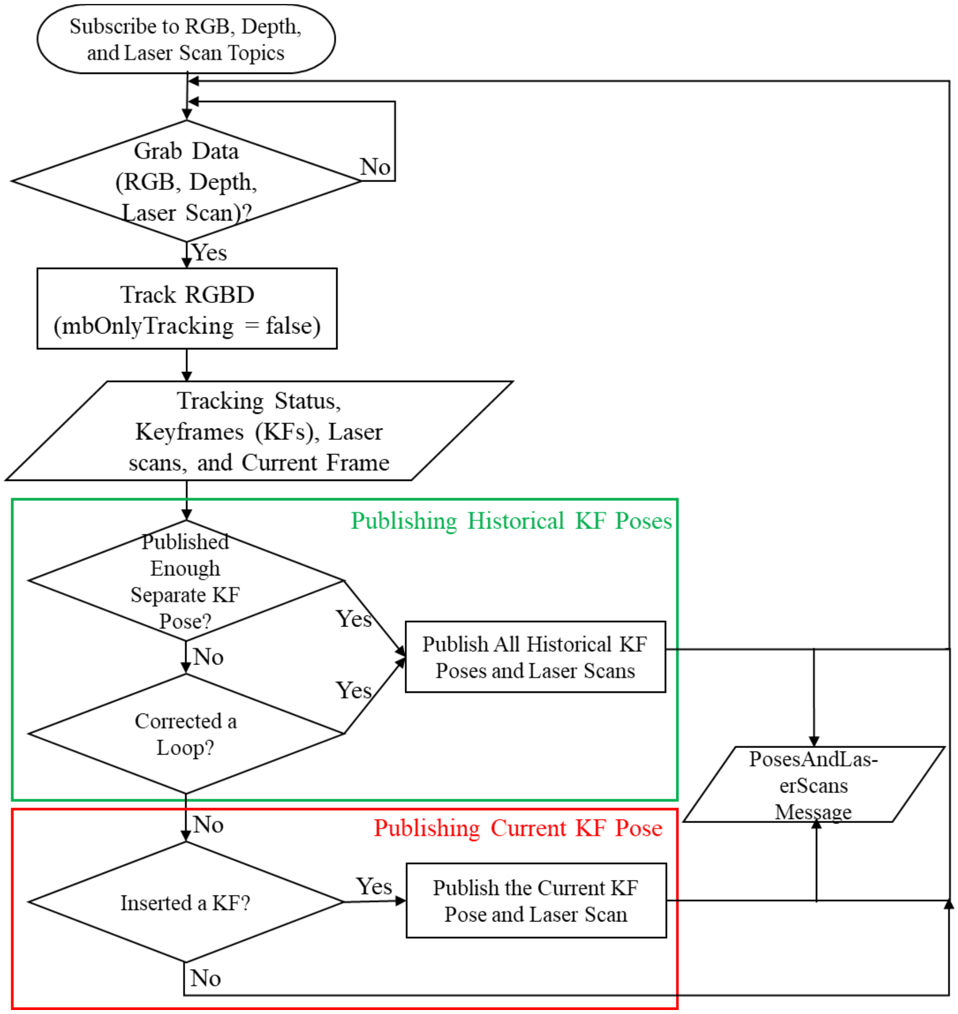

Subscribe to RGB, Depth, and Laser Scan Topics
Grab Data (RGB, Depth, Laser Scan)?
No
Yes
Track RGBD (mbOnlyTracking = false)
Tracking Status, Keyframes (KFs), Laser scans, and Current Frame
Publishing Historical KF Poses
Published Enough Separate KF Pose?
Yes
Publish All Historical KF Poses and Laser Scans
No
Yes
Corrected a Loop?
PosesAndLas-erScans Message
No
Publishing Current KF Pose
Inserted a KF?
Yes
Publish the Current KF Pose and Laser Scan
No

Figure 2.3 Flowchart of the Modified ORB SLAM algorithm

However, for each incoming frame, its pose is finally determined by tracking its local network of keyframes and presented as a relative pose to the keyframe that is nearest to it. Therefore, the estimated frame poses are not as accurate and stable as the keyframe poses. Moreover, compared to the localization process, OGM mapping is much more sensitive to pose estimation. If all the frame poses contributed to building the OGM, then the estimation of every pose would have to be very accurate since even small inconsistencies or errors could potentially corrupt the OGM. Secondly, even though the initial estimation of the frame poses is not highly accurate, they still continue to be optimized with the optimization of the keyframe poses, and better estimation can be used to correct the corrupted map. A critical problem in this scenario is that the number of frames increases unlimitedly over time, so it is impractical, if not impossible, to track back and correct them all with newer results.

Another adopted strategy is that besides publishing separate keyframe poses (and their laser scans) when they are first created (Publishing Current KF Pose in Figure 2.3), all historical keyframe poses (and their laser scans) are also published under certain conditions to help correct the OGM with better keyframe pose estimation that is only available later. The mapping thread keeps optimizing a newly inserted keyframe together with all the keyframes connected to it and culls redundant keyframes. This makes some of the keyframes whose poses have been used in OGM mapping invalid. The adopted solution involves publishing all historic keyframes every time after a fixed number of separate keyframes have been published to ensure that local error in the OGM can, to some extent, be fixed by using the valid keyframes with poses of higher accuracy (Publishing Historical KF Poses in Figure 2.3). Moreover, historical poses and their laser scans are also published after the pose of all the keyframes involved in a loop is corrected by a loop detection and correction step (Publishing Historical KF Poses in Figure 2.3). This allows OGM mapping to correct the drift accumulated in the OGM over a length of elapsed time and distance.

When the SLAM process ends, this component saves the sparse features built by ORB SLAM. It should be noted that the original ORB SLAM does not support such map saving. In the implementation, the feature map is serialized based on the idea described in [131]. The difference is that the proposed approach also serializes the laser scan data in addition to the feature map in order to ensure that “Occupancy Grid Mapping” will have access to all the historical data that has been used to build the OGM and is able to correct the OGM based on loop closure across different SLAM processes.

2.4.1.4 Occupancy Grid Mapping

Once a “PosesAndLaserScans” message arrives, the “Occupancy Grid Mapping” component will extract the pose and laser scan information from the received message and use them to update the OGM it is building and set ROS markers to visualize the camera’s pose in the as-built OGM (Figure 2.4). In the algorithm, the OGM is expressed with an ROS OccupancyGrid message [132], where occupancy probability is in $[0, 100]$ and unknown probability is represented with -1. The log-odds values of all the cells are initialized with -1, and when a cell is explored for the first time, its log odds is first set to 50 for further update.

For a received “PosesAndLaserScans” message, the algorithm first checks if it only includes one pair of pose and laser scan data. If so, the algorithm concludes that the message originates from publishing the current keyframe pose (Figure 2.3) and the pose and laser scan data pair can be directly used to update the OGM based on its current status. However, if the message includes multiple poses (and thus multiple pairs of pose and laser scan data), this indicates that the message is from publishing historical keyframe poses and includes all the further optimized poses of the keyframes that have been used to construct the OGM as of that time. In this case, in order to correct the error in the OGM introduced by the previous inferior estimation of the keyframe poses, the OGM will be completely erased and rebuilt entirely with the received poses. After this, the two situations can be fit into a unified processing step. The primary idea is to process the pose and laser scan pair(s) one by one and only set visualization markers right after the last pair has been processed.

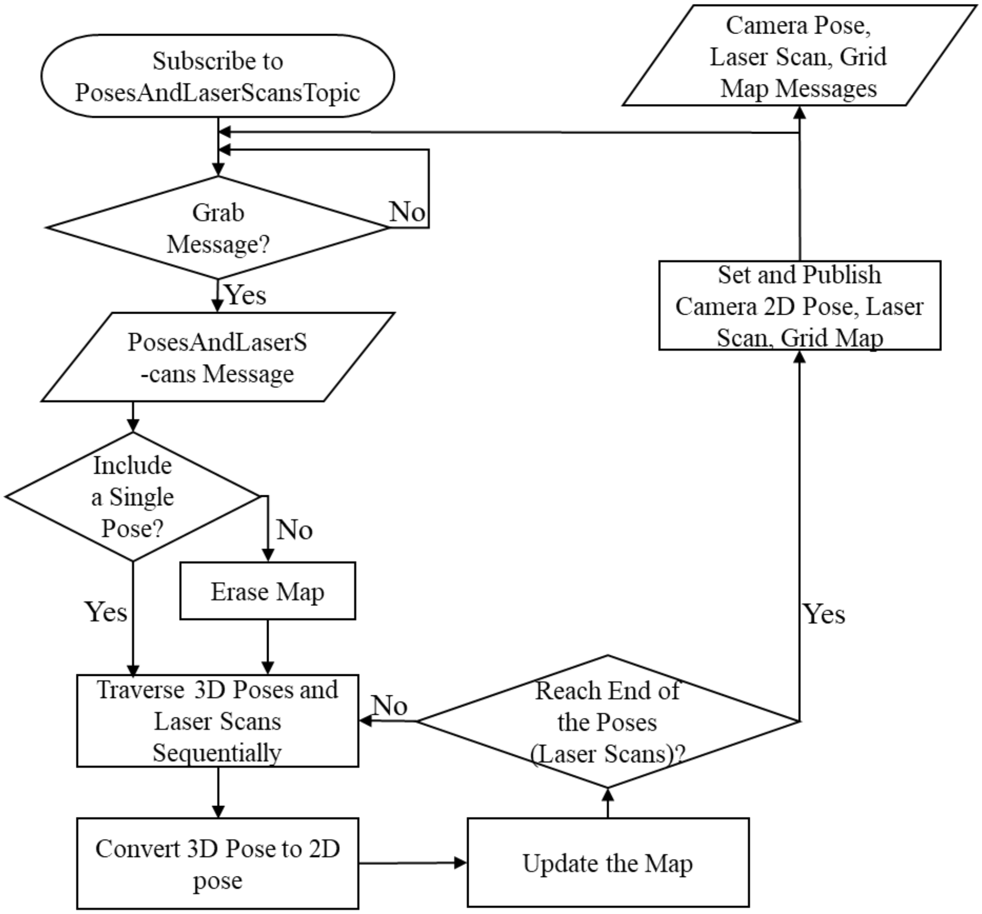

Subscribe to PosesAndLaserScansTopic
Camera Pose, Laser Scan, Grid Map Messages
Grab Message?
No
Yes
Set and Publish Camera 2D Pose, Laser Scan, Grid Map
PosesAndLaserS-cans Message
Include a Single Pose?
No
Erase Map
Yes
Yes
Traverse 3D Poses and Laser Scans Sequentially
No
Reach End of the Poses (Laser Scans)?
Convert 3D Pose to 2D pose
Update the Map

Figure 2.4 Flowchart of the Occupancy Grid Mapping algorithm

For each 3D pose taken out of the pose array in the message, it is first converted to its 2D pose. By default, in ORB SLAM, the coordinate frame attached to the Kinect is defined as shown in Figure 2.8 and the world frame is set with the pose of the camera frame where the tracking process is successfully initialized. In Figure 2.5, the left subfigure shows a visualization of a robot and corresponding laser scan beams in the world frame and in the map frame. The right subfigure shows a visualization of the robot, laser scan and as-built OGM in ROS. For a Kinect camera mounted to a ground robot facing the front (Figure 2.5 Left), its 2D position on the ground plane can be easily obtained, by setting its X position with camera's translation in X, and its Y position with camera's translation in Z. Moreover, considering the inevitable noise in rotation estimation of the camera pose and existence of multiple Euler angle representations for the same quaternion representation, the camera's rotation in quaternion (output format of ORB SLAM) in the message is converted to its axis-angle format and the rotation component is used as the camera's orientation on the ground plane.

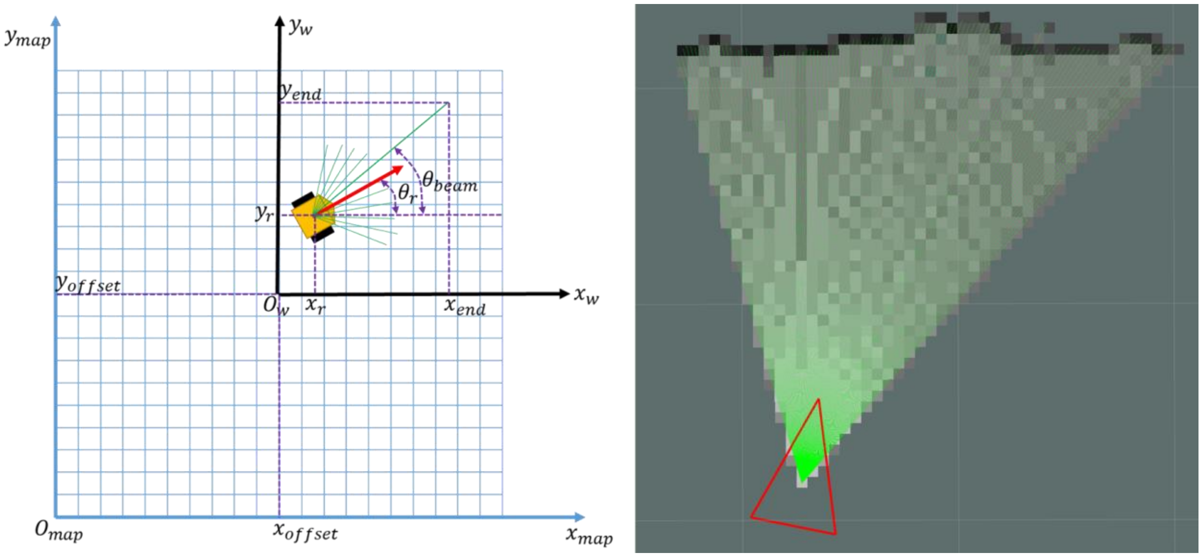


Figure 2.5 Visualization of occupancy grid mapping with a single laser scan

The obtained camera pose in 2D can be denoted as a 3D vector $[x_c, y_c, \theta_c]$, where x_c and y_c represent its position and θ_c represents its counterclockwise rotation with respect to the positive direction of the X axis. As shown in the left of Figure 2.5, the robot pose (with its center at the camera installation point) on the 2D ground plane $[x_r, y_r, \theta_r]$ equals to the camera pose $[x_c, y_c, \theta_c]$. On the other side, the laser scan associated with this pose is extracted as a ROS LaserScan message from the laser scan array. All the beams within the valid beam range in the scan are used to update the OGM and each of these beams is processed sequentially. For each beam, the coordinate of its near end is the same as $[x_r, y_r]$ and the coordinate of its far end can be calculated using the variable values in the LaserScan message (as defined here [129]). Therefore, for the ith beam, the coordinates of the beam beginning point and the beam end can be computed with Equation 2.1, 2.2, and 2.3:

$$[x_{begin}, y_{begin}] = [x_r, y_r] \tag{2.1}$$

$$x_{end} = x_r + ranges[i] * \cos(\theta_r - 0.5(angle\_max - angle\_min) + i * angle\_increment) \tag{2.2}$$

$$y_{end} = y_r + ranges[i] * \sin(\theta_r - 0.5(angle\_max - angle\_min) + i * angle\_increment) \tag{2.3}$$

In the map frame, the corresponding coordinates are calculated with Equation 2.4 and 2.5:

$$[x_{begin}^{map}, y_{begin}^{map}] = [x_{begin} + x_{offset}, y_{begin} + y_{offset}] \tag{2.4}$$

$$[x_{end}^{map}, y_{end}^{map}] = [x_{end} + x_{offset}, y_{end} + y_{offset}] \tag{2.5}$$

Given the cell size k $(m/cell)$, the cell coordinates on the OGM can be further expressed with Equation 2.6 and 2.7:

$$[x_{begin}^{cell}, y_{begin}^{cell}] = \left[\left\lfloor\frac{x_{begin}^{map}}{k}\right\rfloor, \left\lfloor\frac{y_{begin}^{map}}{k}\right\rfloor\right] \tag{2.6}$$

$$[x_{end}^{cell}, y_{end}^{cell}] = \left[\left\lfloor\frac{x_{end}^{map}}{k}\right\rfloor, \left\lfloor\frac{y_{end}^{map}}{k}\right\rfloor\right] \tag{2.7}$$

With the cell coordinates determined, Bresenham's line algorithm [133] is used to find the cells penetrated by the beam and the cell on which the beam end falls. In [110], the algorithm of inverse sensor model for range finders is used to find l_{free}, which is the amount of evidence that a grid is free based on a single beam measurement, and l_{occ}, which is the amount of evidence that a grid is occupied based on a single beam measurement. However, in this research, these values are set to 5 and 15 empirically since the laser scan is converted from a point cloud without using a range finder. In the OGM update, for a single beam, the log odds is decreased by l_{free} for each cell along the beam and increased by l_{occ} for the cell at the beam end. In the same way, the algorithm processes all the beams in the laser scan and finishes updating the OGM with the pair of pose and laser scan data. If there are multiple pose-laser scan pairs in the message, the algorithm will traverse the pairs and process them sequentially as described above.

After processing all the pose-laser scan pairs in the message, the algorithm sets the ROS visualization markers that can be displayed in the ROS rviz tool [134]. In the implementation, the robot pose is expressed with a red isosceles triangle created by a Line-List Marker in ROS, with its apex representing the head of the robot (Figure 2.5 Right). In addition, the laser scan beams are shown as separate green lines with another Line-List Marker (Figure 2.5 Right). The setting of the laser scan marker is very straightforward since the coordinates of the two ends of each beam are already known and can be used directly. For the robot pose marker, the transformation matrix derived from $[x_r, y_r, \theta_r]$ can be used to transform the marker to the right position in the world coordinate. When the SLAM process ends, this component saves the built OGM. The OGM can be readily saved with the ROS map save tool since the OGM is represented with an OccupancyGrid message in the proposed implementation.

2.4.2 Localization Mode

The localization mode (Figure 2.6) is used to localize the camera in the maps built in the SLAM mode, and this process has several similarities to the way the SLAM mode works. Therefore, instead of describing the localization mode in detail, only its differences from the SLAM mode are discussed in this section. In the localization mode, “Modified ORB SLAM” works only with its localization functionality. It loads (serializes) the saved sparse feature map and localizes the 3D pose of the camera in the feature map. In order to enable real-time visualization of the camera pose on the OGM, it publishes the pose of each frame and corresponding laser scan instead of keyframe pose and laser scan in the SLAM mode as shown in Figure 2.3. The process flowchart is simply replacing the “publishing historical keyframe poses” and the “publishing the current keyframe pose” in Figure 2.3 with publishing the pose of the current frame (and its laser scan). Even though it has all the historical keyframes available from the loaded feature map, ORB SLAM does not create new keyframes when it works for localization.

The “OGM Localization” loads the OGM in the localization mode, and subsequently receives a single 3D pose-laser scan pair at a time representing the 3D pose and laser scan of the real-time frame, and then converts the 3D pose to its 2D pose and visualizes the 2D pose and laser scan on the OGM. The difference of “OGM Localization” from its counterpart “Occupancy Grid Mapping” in the SLAM mode is that the OGM is not updated when the system works only for localization.

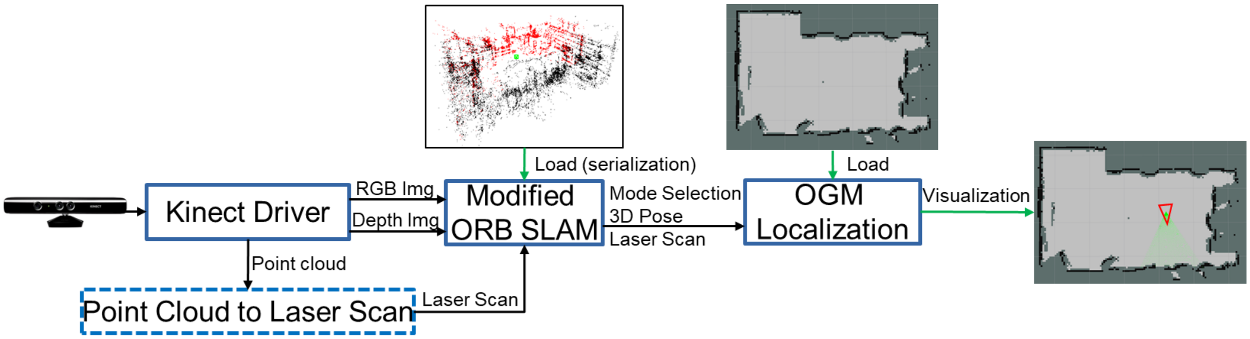


Figure 2.6 Localization mode of the system

As can be observed from this description, the "Point Cloud to Laser Scan" is conceptually not necessary since laser scan information is not used to update the OGM in the localization mode and is rather only used for sensing visualization. However, there are some key reasons for creating a laser scan. One reason is that visualization of laser scan data can help users recognize the scene that the camera is facing. This is very helpful to allow users to switch the system to its SLAM mode at the camera position to update only part of the OGM that is outdated due to changes in the physical world (e.g., obstructions added or removed). Another reason is that in some extreme environments for vSLAM, such as featureless locations or environments with highly repetitive features, the virtual laser scan can be potentially used by Lidar-based SLAM to help localize the camera in the OGM when vSLAM does not work well. This algorithm is planned to be integrated into this system later.

Although the real-time requirement is not a problem for the original ORB SLAM when it is tested with image sequences, it is still important to evaluate the RTLS's practical localization speed after improvement with OGM mapping and ROS-based integration. Similar to Figure 2.3, when the RTLS runs in the localization mode, it always attempts to grab the most current RGB-depth image pair from the ROS topics and process it to localize the camera pose. This means that the RTLS automatically downsamples the input streams of RGB image and depth image to the rate that it can process instead of trying to process every RGB-depth pair. With the benefit of ROS, it is convenient to use the rostopic tool [135] to inspect the update frequency of the 2D pose shown on the occupancy map (as shown in the right-most subfigure in Figure 2.6). The frequency is equivalent to the frames that the system can process in one second. The localization speed of the system was evaluated on a laptop with an Intel Core i7-4940MX CPU@ 3.1GHz. As reported by ROS, with the image size of 640 x 480 for both the RGB image and the depth image (registered to the RGB image), the average speed is ~17.1 FPS (the maximum speed is ~24.4 FPS and the minimum speed is ~12.8 FPS). Although a decrease is observed compared with ORB SLAM, the average speed is still over 15 FPS and is considered as a real-time visual SLAM algorithm [136]. In [137], for construction equipment localization, tracking speed that is equivalent to or greater than 1 Hz is defined as real-time tracking. Therefore, the proposed RTLS achieves much faster tracking speed than that and should meet the real-time requirement for most applications in construction and other civil infrastructure systems.

The maps and localization results shown in Figure 2.6 were from a laboratory room-scale environment. The proposed system was also tested in an entire building scale environment and the corresponding results are shown below in Figure 2.7. The left side of Figure 2.7 shows the sparse feature map built by ORB SLAM and 3D localization within it. Its subfigure on the right bottom corner shows the feature matching between the features in the current frame and the features in the sparse feature map. The right side of Figure 2.7 shows the built OGM and the corresponding 2D localization results within it. The next section characterizes the localization accuracy based on tests and analyses conducted with both these two maps.

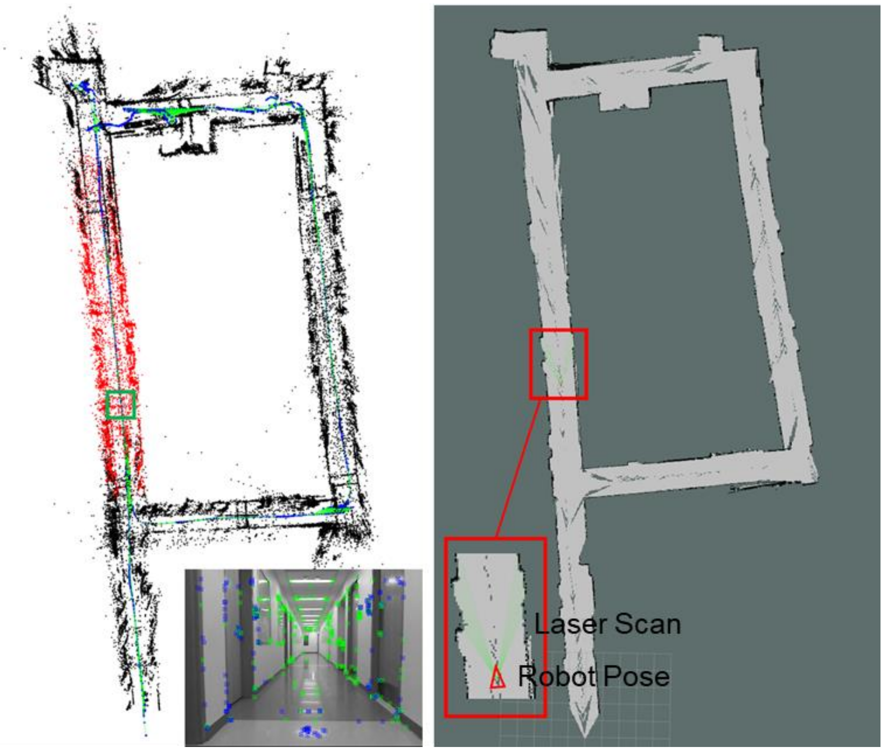


Figure 2.7 Localization results on the maps in a building scale environment

2.5 Experimental Verification

The objective of the proposed RTLS is to obtain the 2D localization on the OGM built side by side with the ORB2 RGBD and enable practical applications that cannot be done only with ORB SLAM. In its implementation (Section 2. 4), the RTLS directly takes advantage of ORB2 RGBD's localization accuracy instead of improving it, and thus it follows that the RTLS would achieve the same localization accuracy as ORB2 RGBD when it is tested on the datasets in [122]. However, since the proposed RTLS is aimed for practical applications on indoor construction sites, its location accuracy is tested additionally in two typical indoor construction environments in this section. The proposed evaluation method proposed in this section could also be used to evaluate other types of locating systems.

The evaluation of localization accuracy of a localization system in a large-scale environment is challenging to characterize [138]. In vSLAM, the localization accuracy is typically evaluated by the error between the estimated 3D pose of the camera and its true 3D pose that is generally provided by a motion capture system. Therefore, there are only a limited number of public datasets that can be used to evaluate the localization performance of vSLAM algorithms, and it is thus difficult to evaluate them in a custom environment.

In order to overcome the inability of deploying a motion capture system in a large-scale custom environment, a marker-based evaluation was developed to evaluate the system's localization accuracy. In this evaluation, the accuracy is evaluated by comparing the 2D positions of the markers deployed in the environment as estimated by the system, with their measured 2D true positions (i.e., ground truth). Since it is difficult to obtain the ground-truth pose of the camera in the 3D space, this method attempts to recast the challenging problem of measuring the 3D pose of the camera to the achievable problem of measuring the distance between markers that are randomly installed in the environment.

This evaluation in effect measures the system's 2D localization accuracy of the markers in the environment instead of measuring the 3D tracking accuracy of the camera, even though the 3D tracking accuracy can be implicitly characterized by the evaluated 2D accuracy. In fact, this is also the likely configuration for how the system can be used in practice to track static or dynamic construction assets or workers that can be recognized via attached markers or through deep learning methods. For example, a human or robot inspector can carry a camera with themselves. When they perform inspection, their global position can be located automatically in the OGM via the proposed RTLS, therefore the construction assets and workers observed by the camera can be also located in the OGM when they are recognized by the camera using some visual markers [86] or objection recognition algorithms [139]. In practice, the human or the robot inspector usually need to make an additional movement to obtain a clear line of sight of the objects to be tracked.

In the next subsections, the marker-based evaluation algorithm will first be introduced in detail and then the localization results will be presented and discussed for both the laboratory scale environment and the entire building scale environment (as shown in Section 2.4.2).

2.5.1 Marker Pose Estimation

The purpose of this algorithm is to estimate the marker's 2D positions in the world frame with the proposed localization system, so the estimated 2D positions can be evaluated with the measured true 2D positions. As shown in Figure 2.8, the transformation from the world coordinate system (WCS) to the marker frame, T_{wg} can be calculated from the transformation chain T_{wc} and T_{cg}. T_{wc} is the transformation from the world frame to the camera frame and is the output of the localization system. T_{cg} is the transformation from the camera frame to the marker frame. AprilTag markers are used in the experiment and the transformation T_{cg} can be calculated by the marker detection algorithm [57]. Therefore, $T_{wg} = T_{wc} * T_{cg}$ and the 3D positon of a marker comprises only the translation part of the transformation matrix T_{wg}.

In the next section, the localization accuracy of the RTLS is tested when it runs in its localization mode by reusing the maps (the sparse feature map and the OGM map). In the localization mode, vSLAM only runs with its localization functionality. However, since the localization accuracy is impacted by both the map quality built in the SLAM mode and the localization performance of the localization mode, the localization accuracy results actually evaluate both of these two modes.

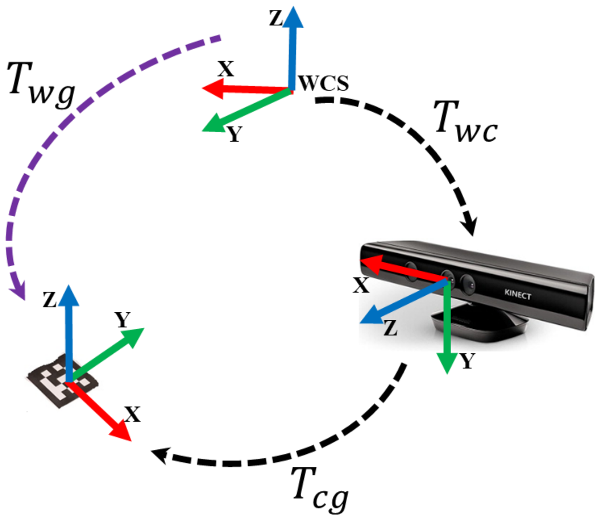


Figure 2.8 World frame, camera frame and marker frame and the transformations between them

2.5.2 Experimental Results and Analysis

2.5.2.1 Measurement of a Single Marker in a Laboratory Scale Environment

This experiment was conducted to verify the robustness of the measurement of the system with the laboratory maps as shown in Figure 2.6. In this test, a marker was fixed on the ground and the 3D position of the marker in the world frame was measured 100 times by the system at each of the six locations as shown in Figure 2.9. The difference in the camera's position relative to the marker can be observed by the position of the door, the wood shelf and the chair in different frames. More specifically, each measurement was obtained with the algorithm explained in Section 2.5.1, and the result of each measurement is T_{wg}, which is the transformation from ORB2 RGBD's world frame to the frame attached at the center of the marker, as shown in Figure 2.8 and Figure 2.9. The detailed format of T_{wg} is as below,

$$T_{wg} = \begin{bmatrix} R_{wg} & t_{wg} \\ 0 & 1 \end{bmatrix} \tag{2. 8}$$

The vector t_{wg} in Equation 2.8 represents the marker's position (x, y, z) in the world frame. At each of the six camera positions, the marker's position was measured 100 times, therefore there are totally 600 measurements of the marker's position (x, y, z), and thus 600 measurements of x, y, z, respectively.

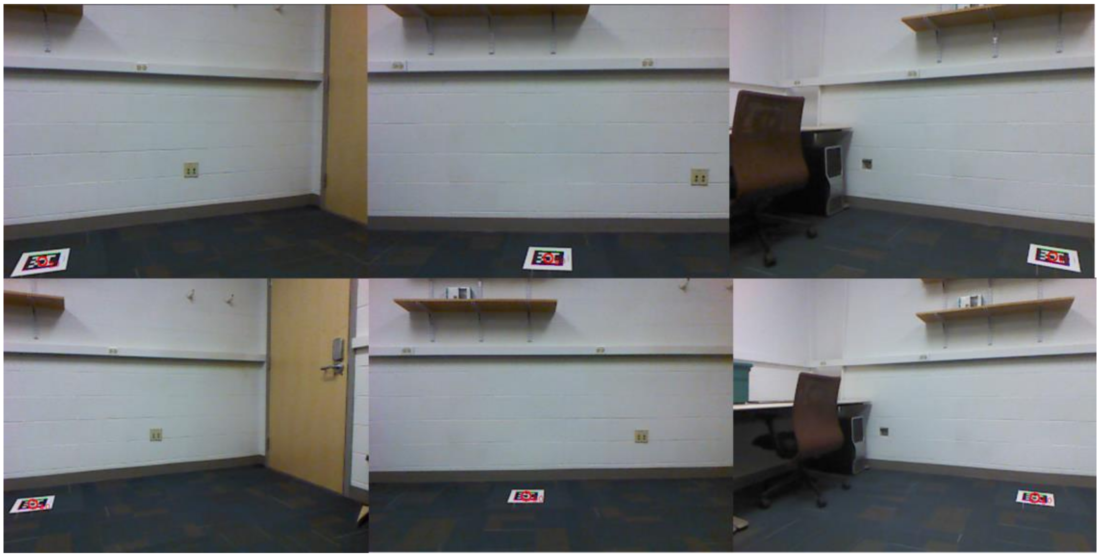


Figure 2.9 Measurements of a single marker at different locations

However, since the marker is also fixed in the environment, the position measurement of the marker should ideally be the same even if the position is measured from different camera locations. In this regard, the standard deviation of these measurements can be used to measure the robustness of the estimated results. In the test, the standard deviation of the x, y, z position of the marker is calculated from the 600 measurements for each of them, and the final results are 0.002m, 0.002m, and 0.003m respectively. More specifically, the standard deviation of x , $\sigma_x = \sqrt{\frac{1}{600} * \sum_{i=1}^{600}(x_i - \mu_x)^2}$, where $\mu_x = \frac{1}{600} * \sum_{i=1}^{600} x_i$. It is the same for the calculation of σ_y and σ_z. The standard deviation results indicate that the repeated measurement accuracy for a single marker is very high but does not necessarily mean that the measurement results are accurate.

This test demonstrates how robustly the position of the maker can be estimated from different observation locations. A similar situation generally arises when the same mobile asset has to been observed and localized from different locations. The marker pose estimation algorithm was implemented in ORB SLAM, so that the robot carrying the camera would not need to stop to collect data. Therefore, the measurement location and the measurement times of each marker were not explicitly controlled in the following tests.

2.5.2.2 Measurement of Multiple Pre-Deployed Markers in a Building Scale Environment

In this section, the system's localization accuracy is evaluated by measuring the position of multiple markers in a building scale environment. As shown in Figure 2.10, fifteen markers were pre-deployed on the ground along the corridor of a basement and formed a loop whose length was about 80m. The maps shown in Figure 2.7 were built in this environment and were used in this experiment for evaluating localization accuracy.

In the experiment, a robotic wheelchair equipped with a Kinect was used as a mobile robotic platform to collect experimental data. The marker pose estimation algorithm can run side by side with the system, estimate the 3D position of the marker appearing in the camera color frame, and write the marker ID and corresponding estimated position to a text file that can be analyzed after the experiment is completed. In order to collect sufficient data to evaluate the localization accuracy, the wheelchair platform was moved along the corridor for five complete loops.

The position of the marker estimated by the localization system is in 3D space, but the pose of the system's world frame was not associated with the physical environment and was thus hard to find. However, it was possible to measure the distance between different markers and build a marker network frame to describe the marker position. In order to make it structured, this marker network frame was set at the center of the #1 marker, with the x axis pointing to the right, y axis point upwards and z axis pointing out of the plane of the screen, as shown in the right part of Figure 2.10.

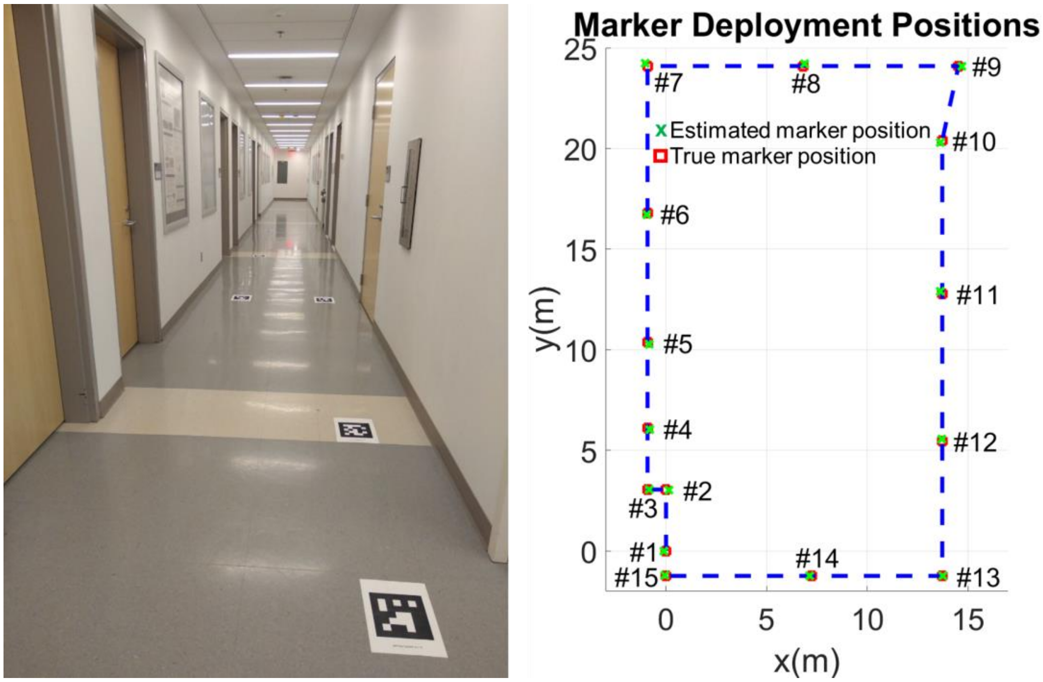


Figure 2.10 Marker deployment in a basement corridor environment and localization results

In order to evaluate the distance between a marker's estimated position and its true position, the system' world frame was first aligned to the marker network frame using a 3D transformation found by the 15 pairs of marker coordinates. Then, the estimated positions of the 15 markers could be projected to the marker network frame and compared directly with the true positions (Figure 2.10 right side). Since the markers were attached to the ground, the z coordinates were always zero. Therefore, only the 2D position of the marker was used for evaluation.

After the coordinate frame alignment, the accuracy of the localization system was evaluated using two metrics, the marker position root-mean-square error (RMSE) and marker distance RMSE. The marker position RMSE represents the RMS of the distance between a marker's estimated position and its true position over multiple measurements. The corresponding results are shown in Table 2.1 and Figure 2.11. It can be seen that the range of RMSE is 0.039m to 0.186m, which is very competitive compared with other indoor localization systems as shown in [140]. It can also be observed that the maximum measurement errors occurred at the #7 marker and the #9 marker, which are 0.186m and 0.185m respectively. The key reason for this observation is that the wheelchair went very close to the wall when the system observed and localized these two markers. Since only a small number of features could be extracted from the surface of the wall, the localization accuracy of the system was impacted in this situation.

Table 2.1 Evaluation results of marker position measurement

Marker ID	**Measured times**	**Estimated Marker Position**		**True Marker Position**		**Position Error**
		X (m)	Y (m)	X (m)	Y (m)	RMSE (m)
#1	76	-0.095	0.004	0.000	0.000	0.098
#2	54	0.147	3.031	0.000	3.050	0.151
#3	63	-0.830	3.057	-0.917	3.050	0.088
#4	54	-0.799	6.042	-0.917	6.100	0.135
#5	42	-0.844	10.277	-0.917	10.370	0.121
#6	50	-0.969	16.725	-0.917	16.778	0.075
#7	50	-1.035	24.240	-0.917	24.097	0.186
#8	64	6.887	24.207	6.816	24.097	0.133
#9	80	14.720	24.090	14.539	24.097	0.185
#10	68	13.640	20.296	13.729	20.412	0.149
#11	53	13.630	12.907	13.729	12.788	0.155
#12	57	13.695	5.547	13.729	5.470	0.088
#13	30	13.764	-1.242	13.729	-1.239	0.039
#14	93	7.146	-1.235	7.238	-1.239	0.094
#15	41	-0.058	-1.216	0.000	-1.239	0.065

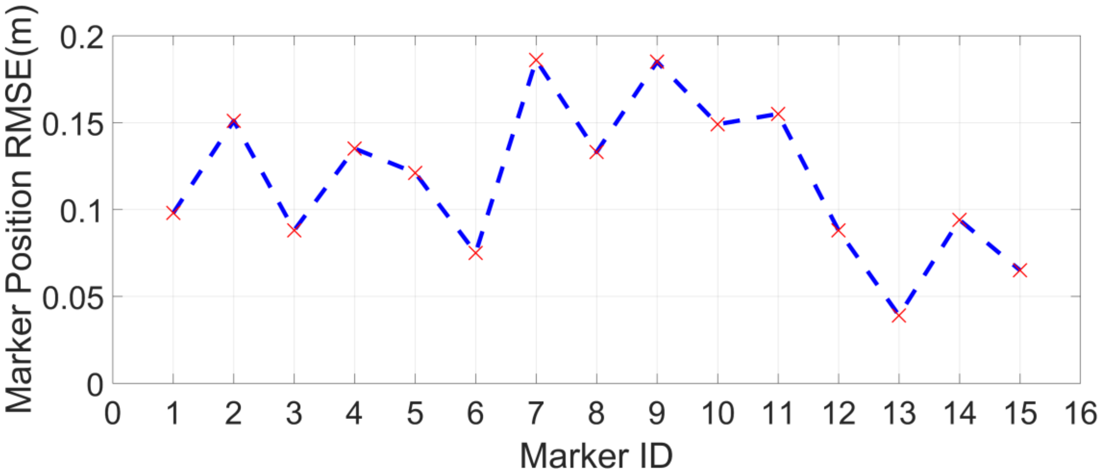


Figure 2.11 Visualization of marker position RMSE

Another metric is the marker distance RMSE, which represents the RMS of the difference between the estimated value and the true value of the distance between markers. As shown in Table 2.2 and Figure 2.12, the distance between every two adjacent markers was estimated with the system and compared with the true values. The range of RMSE is 0.018m to 0.235m, which is apparently wider than the position RMSE. The reason is that the position measurement error can occur in any direction. The distance measurement error will be increased when the position errors in two adjacent markers occur in distinctly different directions, and it will be decreased when the position errors occur in a similar direction and counteract (i.e., cancel) each other. The described experiment is a very stringent evaluation of the system since it is a corridor environment devoid of many distinct features and is generally considered as one of the typical environments where vSLAM cannot work well [119,141]. Even though the proposed system can still achieve satisfactory performance in such a challenging environment, its performance was still impacted by the limited availability of features. In practice, such performance loss can be partially addressed by performing Lidar-based SLAM with the virtual laser scan.

Table 2.2 Evaluation results of marker distance measurement

	Estimated Distance (m)	**True Distance (m)**	**RMSE (m)**
#1-#2	3.037	3.050	0.018
#2-#3	0.978	0.917	0.061
#3-#4	2.984	3.050	0.067
#4-#5	4.235	4.270	0.035
#5-#6	6.449	6.409	0.041
#6-#7	7.516	7.319	0.202
#7-#8	7.922	7.733	0.193
#8-#9	7.834	7.723	0.113
#9-#10	3.945	3.773	0.177
#10-#11	7.389	7.624	0.235
#11-#12	7.360	7.318	0.045
#12-#13	6.790	6.709	0.083
#13-#14	6.618	6.491	0.132
#14-#15	7.204	7.238	0.039
#15-#1	1.221	1.239	0.019

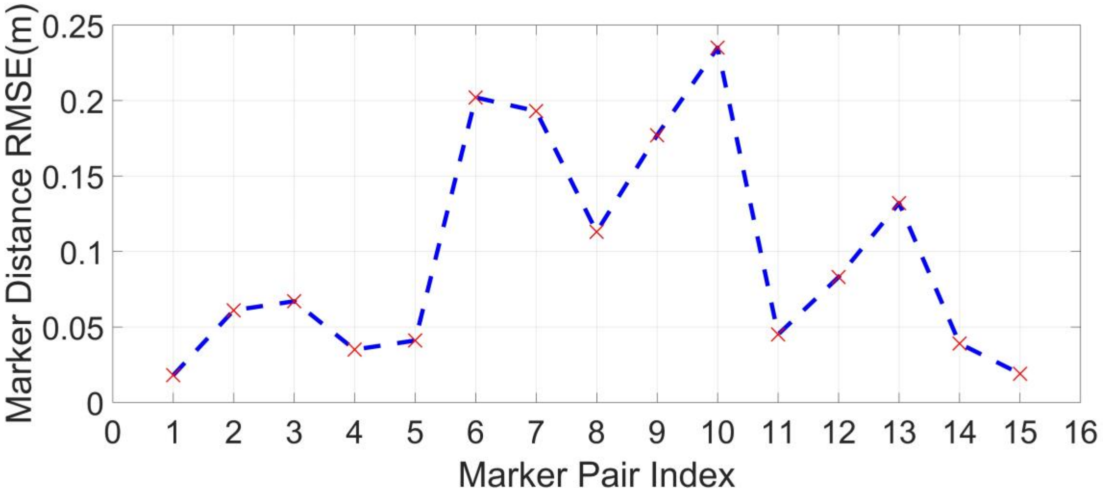


Figure 2.12 Visualization of marker distance RMSE

2.6 Example Applications of the Proposed RTLS

After discussing the implementation of the proposed system and testing its localization accuracy, this section provides three example applications to demonstrate its potential. Describing the implementation of these applications in detail is beyond the scope of this paper, and the examples introduced here are only intended to highlight the capabilities of the developed RTLS in the specific deployments.

2.6.1 Path Planning and Real-Time Navigation

As noted before, it is significantly challenging and often impossible to perform path planning on the sparse feature map built by ORB2 RGBD. However, with the benefits of the proposed localization system, it is straightforward to implement path planning and real-time navigation algorithms with the OGM. An A* [24] based path planning algorithm was implemented as an example and the idea was demonstrated with a robotic powerchair, as shown in Figure 2.13. In the system's localization mode, after it loads the sparse feature map and the OGM of the environment where the powerchair operates, the system localizes the pose of the powerchair in real time and visualizes the pose on the OGM as a triangle in ROS rviz [134].

In ROS rviz, a destination pose can be set with the "2D Navi Goal" and the corresponding ROS message can be captured by the implemented path planning algorithm (an independent ROS package). In the path planning algorithm, the shortest path is calculated from the powerchair's current pose to the set goal position and visualized on the same OGM as shown in Figure 2.13. When the pose of the powerchair changes, the planned path is automatically recalculated and updated, such that the powerchair is able to navigate to the destination step by step in real time. By minimally changing the implementation, users can be allowed to choose their preferred paths by selecting a few navigation goals at a time. This is the basis of autonomous robots and has significant potential to be used for indoor navigation for individuals or to be deployed on indoor robots to enable them to automatically navigate to destinations to perform specific tasks. In the construction domain, this algorithm is also potentially used to extend many existing algorithms in construction research by providing a compact and economic indoor localization solution. For example, the limitation of [1] is that the robot manipulator needs to be within the proximity of the scene for it to work properly. With this algorithm and an appropriate motion control algorithm, the robot manipulator could be navigated to any point of interest in a mapped environment to perform the scan and the following tasks. This could be also used to replace the localization system in [37] for the evaluation of building retrofit performance and the localization system in [13] for convenient registration of 3D point clouds for multiple construction applications.

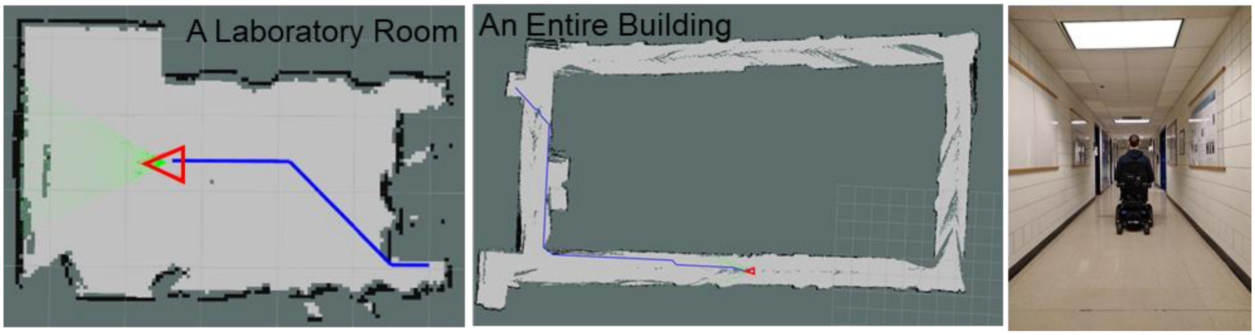


Figure 2.13 Testing results of path planning and real-time navigation on two maps

2.6.2 Geo-Tagged Data Collection

With the benefits of the proposed system, indoor geo-tagged data collection can become more convenient compared with previous algorithms [12,37,142]. Based on the motion planner implemented above, users only need to manually map the physical world to the OGM once (Figure 2.14). Then they can set points of interest where data needs to be collected on the OGM. When the data collection platform (manual or automatic) is close enough to one of those data collection positions, sensors on the platform can be triggered automatically to collect the desired data. This process does not require the deployment of markers in the environment [12,37] nor the use of GPS that has limited performance in indoor environments [142].

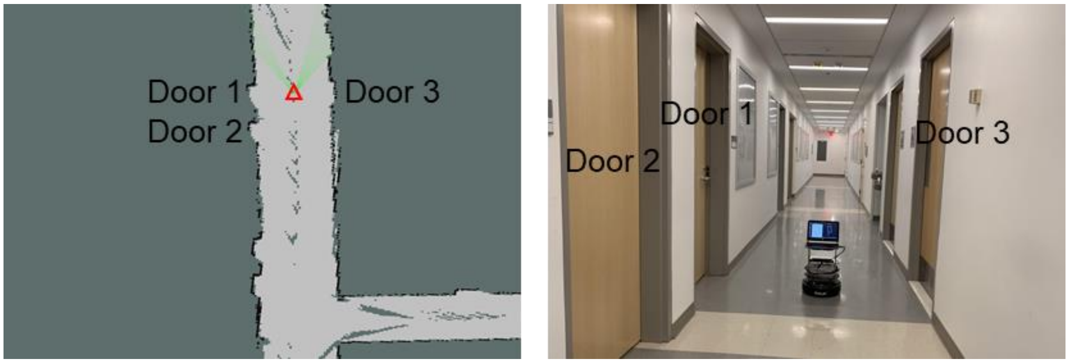


Figure 2.14 Geo-tagged environmental data collection with a TurtleBot platform

2.6.3 Localization-Aware Point Cloud Update

Point cloud technique has been widely used in construction for construction management, construction project scheduling, infrastructure condition evaluation, and so on [13]. This example application provides a possible way to update the point cloud more efficiently for construction infrastructure geometric modeling. The key point is that the localization accuracy is sufficient to initially register point clouds and further update them, and the sensors used to construct the point clouds could be replaced with any other high precision sensors such as a laser scanner or a Velodyne.

With the RGB images, depth images and corresponding estimated poses from ORB2 RGBD, colored point clouds of the environment can be built incrementally in real time (as shown in Figure 2.15). However, it is difficult to simply update a specific part of the point cloud map with reference to the sparse feature map since it includes little semantic information. This issue is appropriately addressed by the proposed localization system. After labeling the real-world locations on the OGM (as done in Section 2.6.2), the area, of which the corresponding point cloud needs to be updated, can be selected on the OGM and the system can be programmed to run in localization mode when the robot is approaching this area and run in SLAM mode when the robot is in this area. Therefore, with the proposed system, there is no need to update the whole point cloud if some changes only occur in a limited area of the environment.

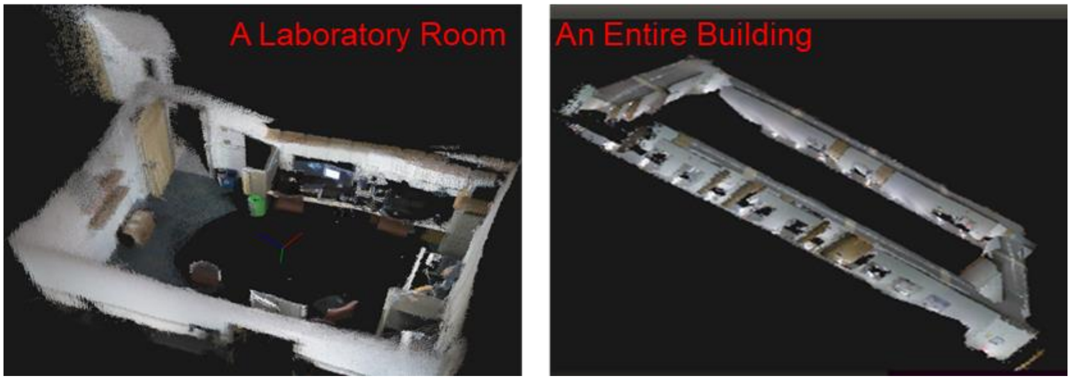


Figure 2.15 Point cloud maps of two environments created based on the proposed system

2.7 Conclusions and Future Work

This paper proposed a vSLAM-based localization system for indoor, GPS-denied environments by building an OGM alongside a sparse feature map. The system can work in SLAM mode to create the maps (sparse feature map and OGM) of the environment and in localization mode to localize the position of the camera in the built maps. The accuracy of the system was evaluated with a landmark-based evaluation method and the evaluation results showed its high localization accuracy and applicability for a broad range of indoor applications. Three examples were also demonstrated to show the potential applications of the system.

However, since the RTLS was developed based on ORB2 RGBD that is a feature-based vSLAM algorithm, it is difficult to run in complex environments including severe illumination variations or highly repetitive features. In addition, ORB descriptors cannot recognize the feature points observed from viewpoints with large differences and it needs smooth rotation in the corridor environments. Therefore, a better feature detector and descriptor need to be developed to improve the system's robustness and accuracy in such difficult environments. Considering that it is difficult to obtain satisfactory feature performance for hand-crafted features, in our future work, learning based methods will be explored to extract robust features from the environment and describe them with improved descriptors that allow better feature matching performance. Besides since in featureless environments feature-based methods would never work, Lidar-based SLAM will be integrated into the system in the future. Finally, the designed system will also be extended and tested in outdoor environments with an RGB-D camera that can work outdoors, and a stereo or monocular camera.

Chapter 3
A Scene-Adaptive Descriptor for Visual SLAM-Based Locating Applications

3.1 Introduction

As a crucial aspect of context, location (including both position and orientation) is one of the most fundamental and valuable pieces of information that bridges local ambient sensing and global digital information in the cloud and allows for optimal decisions based on real-time location-aware information. Benefiting from the continuous advancement of location sensing techniques, location-aware computing has been extensively studied and applied in the built environment. For example, by referring to a BIM model for a robot's current location and the workpieces the robot should be looking at, [80] proposed a framework to automatically make motion plans that can adapt to workpiece geometry and execute the planned tasks. Another example is [37], in which a designed data collection robotic platform can localize itself with a fiducial marker network pre-deployed in the environment and automatically collect geo-tagged data that can be used for evaluating building retrofit performance. Besides these indoor robots, locating techniques can be also integrated into some multifunctional robotic platforms [87,88] for more complex tasks such as inspection and teleoperation in dangerous environments. For a more different application, in [143], localization was achieved by a particle filter that combines BLE beacon fingerprinting and pedestrian dead reckoning, and the localization information was used to give blind people turn-by-turn navigation instructions. It is natural to extend such an application to wheelchair users with physical disabilities to improve their independent mobility in unfamiliar environments [144]. In addition to the above applications, localization systems have also be extensively applied in construction for automatic data collection [15,145], construction management [146,147], construction site safety [148,149], and infrastructure inspection [6,150].

Among the locating techniques that are widely used in such environments, Simultaneous Localization and Mapping (SLAM) based methods can be rapidly and conveniently deployed due to their independence from environment instrumentation, and are thus ideal alternatives to infrastructure-based locating methods such as radio frequency identification device (RFID) [102,151], ultra-wideband (UWB) [90,152], wireless local area network (WLAN) [100], Bluetooth [153], and ultrasound [154] and marker-based methods that primarily use fiducial markers [57,155]. In the SLAM category, visual SLAM (vSLAM) using only camera sensors are significantly more economical compared with Lidar-based SLAM that needs to use 2D or 3D laser scanners. Therefore, vSLAM is generally preferable when the deployment environment is suitable [43].

Based on the method of tracking frames, most of the current vSLAM solutions fall into categories of feature-based method and direct method. The feature-based methods [44-46] track a new frame by matching features in the frame to created map points and minimizing reprojection error of the observed map points in the frame. Instead of tracking with hand-crafted features, direct methods [47-50] track a new frame by direct frame alignment and the camera pose is optimized over pixel intensities. Even though the state-of-the-art feature-based methods achieve better tracking and reconstruction performance than the state-of-the-art direct methods, they suffer from robustness issues in environments with motion blur, low structure or low texture. The main reason is that it is difficult to extract enough pre-designed key points and match them robustly in these situations using either fixed hand-crafted descriptors [122,156,157] or fixed learning-based descriptors (this is proved in this paper).

In order to improve feature-based vSLAM's applicability under various conditions in practical applications, we propose and explore Deep SAFT, an online learning-based scene adaptive feature transform that is self-adaptive towards recently observed scenes. By taking advantage of the strong representation power of convolutional neural network (CNN), Deep SAFT can be used to replace its fixed counterpart in a feature-based vSLAM system (e.g., ORB-SLAM2 in this research) and fine-tune itself online using true feature correspondences obtained after a geometric verification step, and run in parallel to tracking and mapping threads. Moreover, the fine-tuning process is adjustable to balance the feature representation between general representation from offline trained weights and local presentation from updated weights.

3.2 Review of Related Prior Research

3.2.1 SLAM-Based Locating Solutions

SLAM algorithms allow to track agent pose and recover observed environment simultaneously only using mobile sensors. As a promising alternative to traditional localization solutions (such as RFID, UWB, WLAN, Bluetooth, and ultrasound), it has gained increasing attention for a variety of applications in built environments [158]. In [159], a robotic system equipped with a monocular camera and a 2D laser scanner was developed to apply spray foam insulation to underfloor voids. The platform pose was initially estimated by feature matching using the input image and then finally determined by optimizing point cloud alignment using iterative closest point (ICP) [160]. In [13], Kim et al. demonstrated a robotic system that facilitated point cloud registration using the location information estimated by Hector SLAM [39]. In [6], Peel et al. proposed to combine

Adaptive Monte-Carlo Localization (AMCL) [161] and Hector SLAM [39] to improve localization performance for increasing autonomy in bridge bearing inspection. For autonomous tunnel inspection, [14] proposed a navigation strategy based on SLAM using a set of pre-deployed markers in a tunnel. SLAM-based localization was also utilized in [162] for collaborative 3D printing performed by a team of robots that could map the environment with GMapping [41] and then localize themselves in the map with AMCL. All the above solutions depend on Lidar-based SLAM. However, besides distance, Lidar sensors do not provide enough semantic information (room number, words on a sign, semantic object recognition, or object color). Moreover, cost-prohibitive Lidar sensors limit their widespread deployment for applications in built environments or on construction sites, at least in the near term.

Compared with Lidar-based SLAM, vSLAM only needs low-cost camera sensors and are more economical and favorable in the environments where it can meet the requirements for localization accuracy. In order to facilitate data collection on construction sites, Asadi et al. [163] proposed a mobile robotic platform that was equipped with a camera sensor and NVIDIA GPUs and suitable for monocular SLAM and deep learning-based scene understanding tasks. This work qualitatively proved vSLAM's effectiveness for applications on construction sites but did not evaluate vSLAM's accuracy quantitatively in custom environments. More recently, we proposed a real-time locating system [158] built upon a state-of-the-art feature-based SLAM (ORB-SLAM2 [122]). This locating system is suitable for construction-related applications such as path planning and real-time navigation, geo-tagged data collection and location-aware point cloud update. While taking advantage of high localization accuracy of ORB SLAM, like other feature-based vSLAM algorithms, this system also suffers from robustness issues in challenging environments (motion blur, low texture, low structure and so on).

3.2.2 Feature-Based vSLAM

As an important refinement method in structure from motion (SFM) [164] solutions, real-time bundle adjustment (BA) [165] had long been considered computationally prohibitive and unachievable until the work of [166] solved this by optimizing over only selected keyframes (a subset of frames with enough overlapped observation that are selected to reduce optimization cost). Following this work, parallel tracking and mapping (PTAM) [45] proposed separate parallel tracking (initially estimating the pose of the current frame) and mapping (optimizing keyframe poses and updating world points in the map) threads to take advantage of the difference in their update frequency, which has been adopted by most modern vSLAM solutions. [167] and [168] were two early representative solutions of stereo SLAM, which integrated visual odometry, constant-time mapping, appearance-based loop closure detection and global map correction and can be used for real-time large-scale SLAM. Different from stereo SLAM that is easy to triangulate depth from one observation of a stereo camera, the lack of depth information makes monocular SLAM difficult to do map initialization and introduces the extra scale drift in the built map. The current common ways of map initialization are inverse depth parameterization [169] and camera pose recovery from a homography matrix [45] or a fundamental matrix [170] or an automatic selection from one of them [46]. To correct scale drift at loop closure, [171] proposed a pose-graph optimization [172] technique based on Lie group [173] that optimized 7 degrees of freedom (DoF) pose. With the advent of low-cost RGB-D sensors such as Microsoft Kinect and Asus Xtion PRO LIVE, depth information became easier to obtain and many RGB-D SLAM solutions have been proposed [174-176]. [177] proposed a method to synthesize the depth of a feature point into an additional pixel coordinate as if it was observed in another camera, and thus a SLAM system can treat input from a stereo camera or an RGB-D camera in the same way [122]. Besides, some other recent efforts include appearance-based place recognition for real-time loop closure [123,178], and optimization with covisibility information [179] for efficient large-scale operation [122,177].

In continued efforts to improve vSLAM’s robustness and accuracy, the studies above mainly focused on vSLAM framework design and optimization strategies that are built on top of fundamental tracking quality. However, this research focuses its efforts on improving the tracking process itself from the aspect of robust and accurate feature representation and matching. The proposed SAFT descriptor in this work needs to learn from sequential frames. It is specially designed for vSLAM-based locating applications and must work together in vSLAM. It can be integrated into any of the above algorithms by replacing its feature-related part and take advantage of the corresponding optimization framework. As one of the most successful vSLAM algorithms, ORB-SLAM2 not only achieves state-of-the-art accuracy but also provides an open-source solution for monocular, stereo and RGB-D SLAM. Therefore, it is an ideal choice as a baseline algorithm. For the convenience of validating the proposed SAFT and providing an example integration implementation, we use the RGB-D part of ORB-SLAM2 for evaluation purposes.

3.2.3 Feature Descriptors in vSLAM

Distance between descriptors has been extensively used to measure the similarity between image patches. In vSLAM, traditional hand-crafted fixed feature descriptors such as SIFT [180], SURF [181], BRIEF [182], and ORB [123], are generally used for matching purpose and achieve satisfactory performance in most applications. However, their robustness and accuracy exhibit considerable degradation in applications involving especially motion blur [156], low structure [122] or low texture [157] that are quite common in practical applications. In order to address these limitations, some researchers have also dedicatedly designed some descriptors for specific challenging situations. [156] developed a blur-invariant descriptor by combing integral projection from four directions. Besides, [157,183] developed more robust descriptors to recognize textureless objects. However, for hand-crafted descriptors, it is difficult and requires significant effort to improve their matching robustness by characterizing diverse challenging conditions. As such, these dedicated designed descriptors were only focused on a specific challenging situation and it is difficult to continuously improve them. Moreover, improvement becomes even more difficult when it comes to balancing speed and accuracy requirements for vSLAM applications.

Due to the above reasons, some researchers shifted their attention to learning-based methods. Compared with hand-crafted descriptors, without the need for explicit models, learning-based descriptors are much easier to design and improve by training learnable parameters. They can also learn from multiple hand-crafted descriptors to get better performance. Therefore, some learning-based descriptors have already achieved better matching performance than the traditional hand-crafted feature descriptors on different datasets [184-186]. MatchNet [184] and [185] trained a unified Siamese CNN for both feature presentation and feature comparison, and in [185] more different network structure variances were designed and tested. In [186], random sparse connections between network layers and hard negative mining [187] were introduced to increase speed and prediction accuracy. Following these works, [188] proposed to train a CNN with triplets of patches and significantly reduced training and execution time. After optimizing matching performance, computational efficiency can be further improved with network acceleration and compression techniques [189,190]. The key point to note is that with learning-based design, matching performance and efficiency can be optimized quite separately, which is much easier than optimizing them together in hand-crafted design. This ability thus offers a more feasible way to develop better descriptors for vSLAM applications.

However, all these learning-based descriptors were only trained and tested with some image patch datasets [191]. These datasets were prepared with images collected in a limited number of scenes and generally lack challenging samples. Even though some trained descriptors outperform the traditional hand-crafted descriptors on these datasets, as fixed descriptors, for applications in vSLAM they still suffer from generalization issues and performance degradation in blurred/low-structured/low-textured scenes (as proved in evaluation results shown in Figure 3.8 and Figure 3.9).

With respect to generalization issues, given any learning-based descriptor, theoretically we can always improve it by training its network separately with images collected in different scenes and when we use it in a specific scene, we select to use the model previously trained in the same scene. This can be approximately achieved even though the process is time-consuming. However, it is very difficult to include in a dataset enough training samples characterizing feature matching under challenging conditions.

In this study, we propose to solve the two categories of issues uniformly by designing a dynamic descriptor based on a CNN, which allows online training and prediction to obtain a better description of the current scenes. To the best of our knowledge, this paper is the first attempt to use an online training CNN descriptor for solving the vSLAM problem.

3.3 Research Objective, Scope and Contribution

With the goal of improving the robustness and accuracy of feature-based vSLAM algorithms in challenging environments for practical applications, the primary objective of this research is to explore the performance of a learning-based descriptor (Deep SAFT) in vSLAM and the potential benefits of online training that allows the descriptor to learn from the scenes that are recently visited.

The scope of this research includes implementation of Deep SAFT by improving an existing descriptor network for online training and testing applications, integration of the designed Deep SAFT into ORB-SLAM2 for a demonstration implementation, and performance evaluation of the Deep SAFT embedded ORB-SLAM2 with two widely used public datasets. Even though SAFT is implemented with a descriptor network in the paper, its learning architecture does not necessarily have to be a neural network - it can be any descriptor model whose parameters can be tuned. Better computational and accuracy performance of SAFT embedded feature-based vSLAM can be achieved by implementing SAFT with some neural network acceleration techniques [189,190,192], or an optimized neural network, or a different learning architecture and integrating it into a vSLAM with an optimized design. It is worth noting that with the objective of exploring the feasibility and benefits of Deep SAFT, this work does not focus on optimal implementation of Deep SAFT and its integration with ORB-SLAM2, even though computational performance and some improvement strategies are discussed in Section 3.5.3.

The specific contributions of this research are as follows:

- A scene-adaptive feature transform (SAFT), which is self-adaptive to a better description of the currently observed scenes, is proposed to improve learning-based descriptors' matching robustness for vSLAM applications.
- An integration framework is proposed to integrate the SAFT into a state-of-the-art feature-based SLAM (ORB-SLAM2) and train it online.

- Better localization performance of SAFT compared with its offline model is demonstrated to verify the feasibility and effectiveness of SAFT for matching robustness improvement.
- Localization performance of SAFT and a corresponding baseline algorithm is demonstrated to be comparable to prove that a vSLAM can be improved by SAFT implemented with a well pre-trained learning-based descriptor that is appropriately integrated into the vSLAM.

3.4 Deep SAFT Embedded ORB-SLAM2 (RGB-D)

Our SAFT embedded ORB-SLAM2 (RGB-D) system, as shown in Figure 3.1, includes a modified tracking thread, a new online learning thread and other threads (local mapping and loop closure from ORB-SLAM2 with minor modification) that run in parallel to do pose tracking with SAFT and train SAFT online with image patches obtained from observations of recently visited scenes. The proposed algorithm replaces the rotated BRIEF descriptor of ORB feature in the original ORB-SLAM2 with Deep SAFT descriptor, such that corresponding modifications are also made in the subroutine functions involved in the local mapping and loop closure threads. These changes are relatively straightforward compared with the implementation of the tracking and the online learning threads, such that they are not reflected in Figure 3.1.

The tracking thread extracts SAFT features (FAST keypoints [193,194] + SAFT descriptor) from the input RGB frame with the updated prediction net. Then, it initially tracks the current frame by matching its SAFT features to the map points in the last frame or its reference frame [158]. If this is not successful, it would attempt to relocalize the current frame using a BoW vocabulary created with SAFT descriptors extracted from the dataset Bovisa 2008-09-01 [195] and SAFT descriptor matching. After initial pose estimation, this tracking thread also tracks the current pose in the local map by finding the map points in the local map corresponding to the key points in the current frame and further refining the estimated pose, as done in ORB-SLAM2. In addition, it is also in charge of determining whether to prepare training data for the online learning thread.

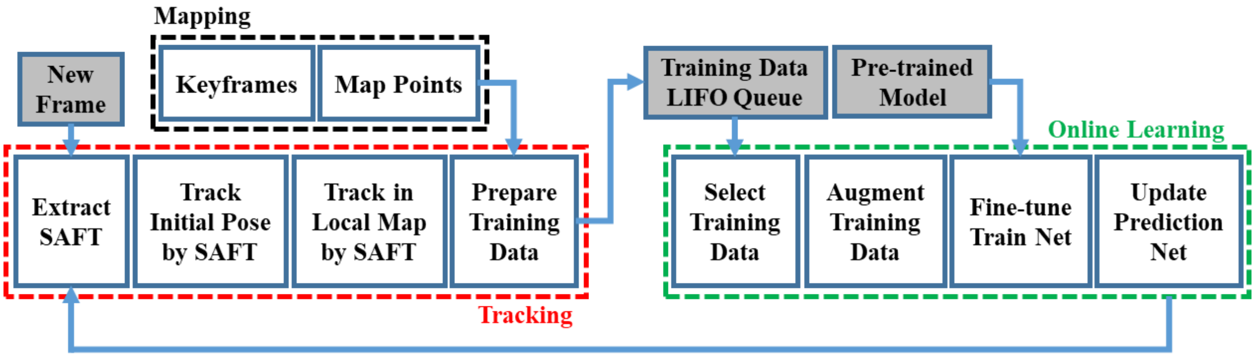


Figure 3.1 System overview of Deep SAFT embedded ORB-SLAM2 (RGB-D)

The online learning thread keeps checking the training data queue, selects training data, augments the data and fine-tunes the train net to update its weights to reflect the recent observations. The weights of the prediction net are always updated with the newest weights of the train net to allow the tracking thread to use the most recently learned knowledge. More details about the design of the whole system will be discussed in the following subsections.

Figure 3.2 shows the observation of weight updates for some selected convolutional filters during an online training process in our experiment. The first row shows the frames (frame number increases from left to right) from the officeroom3 (or3) sequence in the ICL-NUIM dataset [196]. These frames are the first frames that adopt newly updated CNN weights. Row 2/3/4 shows the weight differences (except for the first column that shows the initial weights) between successive CNN updates of a randomly selected kernel in each of the three convolution layers (Conv_1/Conv_7/Conv_13 in Figure 3.4). For example, the first image in Row 2 shows the original values of the 60th filter in the Conv_1 layer in Figure 3.4, and the 5th image in Row 2 shows the value difference between the filter weights for frames 333~417 and the filter weights for frames 267~332. For visualization, the values are normalized between [0,1] and darker colors represent smaller values. The weight updates of one filter depend on not only viewing angles but also updates of many other filters in the same layer and in other layers, change in observed scenes, scene scale, even motion blur, and many other factors. Considering such a complicated relationship, it is difficult to even qualitatively explain how weights are updated according to observed scenes.

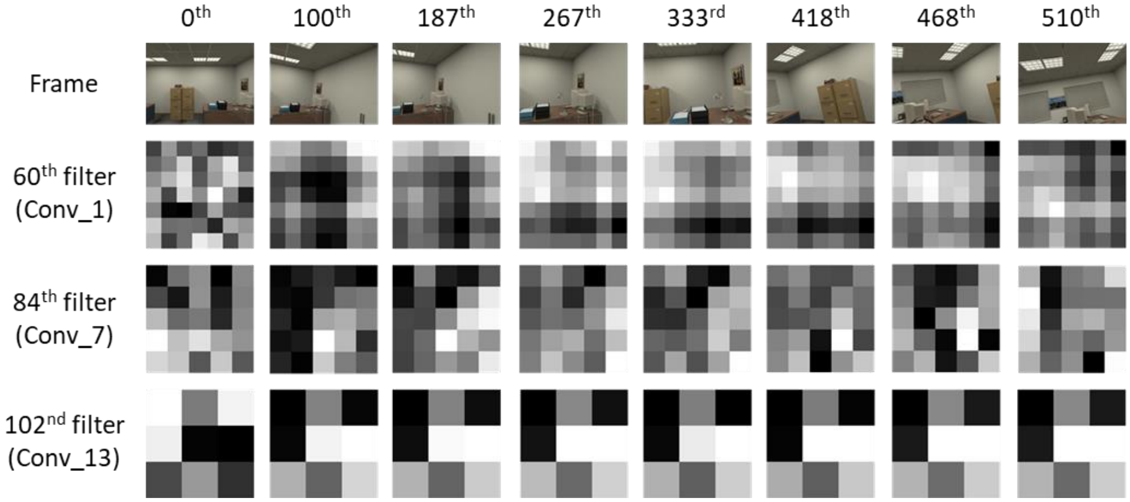


Figure 3.2 Visualization of Deep SAFT's weight updates as frame advances

From another perspective, this is the main benefit of CNN representation that allows updating weights to fit this complex nonlinear relationship between weights and observed scenes without explicitly knowing the model. Therefore, instead of trying to show how Deep SAFT adapted to an environment, Figure 3.2 is mainly meant to show that a fixed descriptor is not always optimal for various scenes, and Deep SAFT can keep updating the weights to a customized representation of the current observation. Moreover, as proved by the evaluation results shown in Figure 3.8, this customized representation is a better presentation.

3.4.1 Learning-Based Descriptor

Different from the traditional way of measuring descriptor distance, learning-based patch comparison can be done with or without a direct notion of patch descriptor. For feature matching in vSLAM, in order to avoid testing all combinations of image patches in a brute-force manner, it is preferred to provide similarity comparison based on explicit descriptor extraction. The most popular way of achieving this is by stacking a top Siamese network used for descriptor extraction and a bottom decision network used for similarity evaluation (Figure 3.3), as done in [184-186].

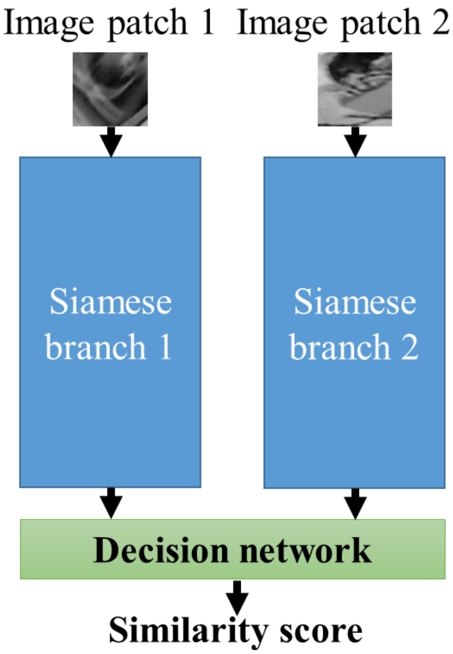


Figure 3.3 A general Siamese neural network structure for a learning-based descriptor

To demonstrate the feasibility and benefits of online descriptor learning, the Siamese architecture proposed in [185] is correspondingly implemented and customized in Caffe [197] for learning descriptor representation since this network has a simple structure and is easy to train for online training purpose. Figure 3.4 shows the training network architecture for the SAFT descriptor. For a convolutional layer, the three parameters in parentheses represent (filter number, filter size, stride size). For a pooling layer, the two parameters in parentheses represent (filter size, stride size). For an inner product layer, the parameter in parentheses represents the output dimension. As shown in Figure 3.4, the network takes as input two 64 × 64 grayscale patches and a label that is +1 when the two patches are matched (describing the same point in the world) and -1 when they are not matched.

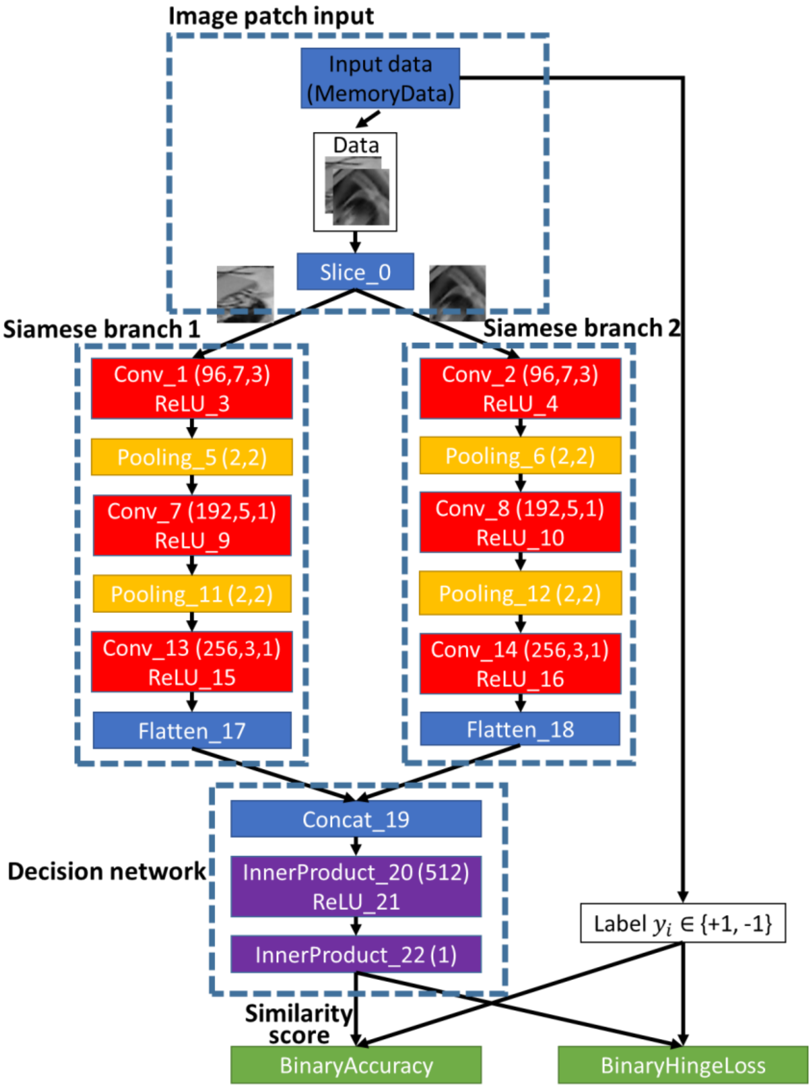


Figure 3.4 The training network architecture for SAFT descriptor

Following a slice layer, the two image patches are separated and input into the two Siamese branches where two 256-dimensional descriptors are extracted and output at layers Flatten_17 and Flatten_18 (Figure 3.4). Using these two descriptors, the following decision network computes the similarity score of the two original image patches. The similarity score and the input label of the image patch pair are finally used to calculate the binary hinge loss at the custom BinaryHingeLoss layer and current learning accuracy at the custom BinaryAccuracy layer. While the loss is used to train parameters of the network, the accuracy provides a clue of training status and helps control the learning process.

As with most other neural networks that are designed for offline applications where accuracy is much more important than efficiency, the original network proposed in [185] only supports reading input data from saved image data on the hard disk. Considering this is too time-consuming for SLAM applications, a MemoryData layer is utilized as the input layer to read image and label data directly from memory instead of from previously saved data (Figure 3.4). Moreover, the following binary hinge loss function is used to train the neural network.

$$L(\omega, I_i, y_i) = \sum_{i=1}^{N} max\ (0, 1 - y_i o_i^{IP\_22}) \tag{3.1}$$

where ω is the network parameter, I_i represents the i^{th} input image patch pair, y_i is the label of I_i. The term on the right-hand side is a binary hinge loss, in which $o_i^{FC\_22}$ depends on ω and I_i and represents the output at InnerProduct_22 in Figure 3.4. This is also the loss function used in [185]. The difference is that we implemented our custom BinaryLoss layer in Caffe to accommodate the use of the labels of -1 and +1. In the custom BinaryLoss Layer, for forward computation, Equation 3.1 is directly used to update the loss. For back propagation, Equation 3.2, the derivative of Equation 3.1 is used to update the differential values.

$$\frac{\partial L}{\partial o_i^{IP\_22}} = \begin{cases} -y_i & 1 - y_i o_i^{IP\_22} \geq 0 \\ 0 & otherwise \end{cases} \tag{3.2}$$

Equation 3.1 does not include the regularization term for the actual training process, since when regularization parameters are set properly in the network definition protobuf and solver protobuf file, Caffe processes the regularization term automatically and incorporates the results into the above forward computation and back propagation. In our implementation, we set regularization type to L2 and weight decay λ set to 0.0005. Correspondingly, a custom BinaryAccuracy layer was implemented to monitor the accuracy change while the training process is going on, which helps avoid overfitting in offline training. It is also a good indicator to decide when to stop training and share learned weights with the prediction net in an online training process.

The prediction net (Figure 3.1) only includes one modified input layer from the training network on the top and one Siamese branch from it at the bottom. The modified input layer only takes in a single image patch from the memory, and the following Siamese branch computes its 256-dimensional descriptor with shared weights from the training network. Considering that the decision network in the training network (Figure 3.4) is not efficient for online applications such as vSLAM, instead, L2 norm is used to measure descriptor distance. However, direct L2 norm matching suffered from lots of performance loss compared with the matching performance of the trained decision network (quantitative results will be shown later). This is solved by normalizing the learning-based descriptor first before they are used for feature matching. This normalized learning-based descriptor can be equivalently viewed as a hand-crafted descriptor and can be used in the same way for feature matching and bag-of-words (BoW) vocabulary creation in vSLAM.

Different from the training parameters reported in [185], we trained our re-implemented network offline from scratch on the Liberty subset from [191] with the following parameters: SGD with step-size learning rate, base learning rate 0.0001, gamma 0.1, step size 6000, momentum 0.9 and weight decay λ = 0.0005. Figure 3.5 (left) shows how the loss and accuracy changed during a training process in which the network was trained on Liberty subset from [191] and tested on Notre Dame subset from the same dataset. It can be observed that its performance became steady after about 10000 iterations. The right subfigure of Figure 3.5 shows the performance of the three above descriptor distance measurements (decision network, L2 norm with normalization and L2 norm without normalization) evaluated with receiver operating characteristic (ROC) curve and false positive rate at 95% recall (FPR95). In the right subfigure, the number at the end of each legend is the corresponding FPR95 value.

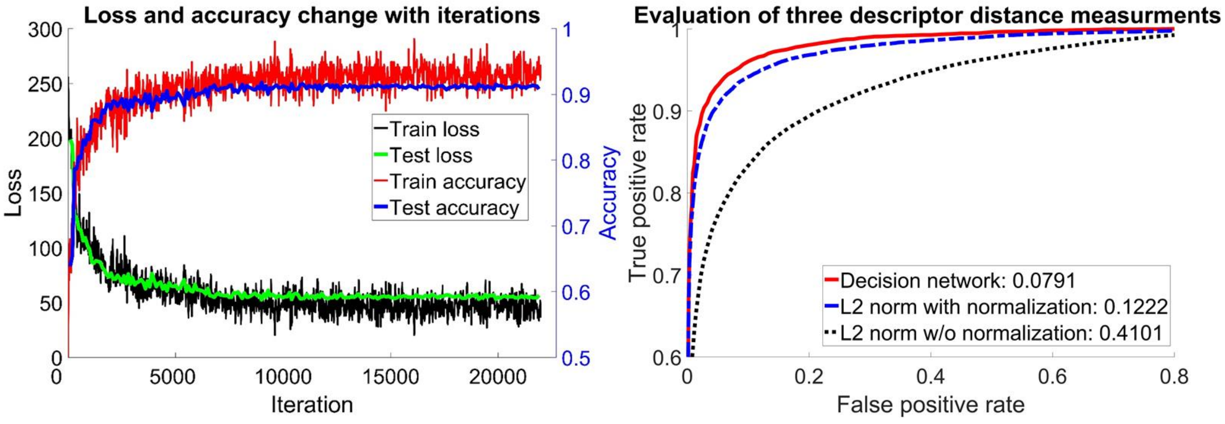


Figure 3.5 Training results and evaluation of different descriptor distance measurements

The results show that the performance of the decision network is quite comparable to the corresponding results in [67]. Without using exactly the same distance measurement (the decision network) that was used to train the network, it is no surprise that L2 norm suffered from some performance loss. For L2 norm without normalization, the performance becomes a lot worse (with FPR95 value 0.4101) than using the decision network. By normalizing the descriptor before taking L2 norm, the FPR95 value can be improved back up to about 0.1222 and is much closer to the performance of the trained decision network. Considering the significant difference in performance loss, a reasonable conjecture is that the nonlinear model expressed by the decision network does some normalization implicitly. Even though L2 norm with normalization still suffers from some performance loss compared with the decision network, its performance is still better than SIFT (FPR95 0.2809 [80]) and ORB (FPR95 0.4803 [81]). Considering the tradeoff between matching efficiency and matching accuracy for vSLAM, L2 norm with normalization is used as the SAFT descriptor distance measurement in our modified vSLAM system.

3.4.2 Tracking with SAFT

In the baseline algorithm, a new incoming frame is first tracked with a constant motion model assumption. It assumes constant motion between sequential frames and uses the relative motion between the last two frames to infer a guess of the pose for the current frame. With this pose guess, each map point in the last frame is projected into the current frame and its matching key point in the current frame is searched for within a region around its projected point by measuring the Hamming distance between the ORB descriptors (rotated BRIEF) of the map point and all the key points in the region. Then the initial pose of the current frame can be estimated if enough correspondences are found. If tracking with motion model fails due to insufficient correspondences, the tracking thread then would attempt to estimate the current pose by tracking with the most recent reference keyframe or performing relocalization when the tracking is lost, both of which involve searching for correspondences with bags of words (BoW) vocabulary [122,125] created from ORB descriptors.

In order to fully test the SAFT descriptor's performance in a vSLAM system, the ORB descriptor in the baseline algorithm is completely replaced with the learning descriptor computed with the prediction network (Figure 3.1). In the detailed implementation, for each frame, FAST detector

[193,194] is first utilized to extract near-uniformly distributed key points at different pyramid levels. Then, in each pyramid level, 64 × 64 grayscale patches centered at the key points extracted from the same pyramid level are cropped and used as input into the prediction network. As discussed in Section 3.4.1, the prediction network with shared weights from the training network (Figure 3.4) computes the 256-dimensional float descriptors of the input single image patches. Correspondingly, Hamming distance used to measure the distance between ORB descriptors is replaced with L2 norm with normalization to measure the new learning descriptor's distance.

In ORB descriptor, the centroid of a circle patch centered at each key point needs to be computed to achieve good rotation invariance. However, it is found that this procedure can be just omitted since the learning descriptor is already invariant to rotation to some extent by being trained with image patches observed from variant viewpoints. However, the image pyramid is still necessary to improve robustness to motion blur and invariance to scale for the learning-based descriptor. Even though an improved network architecture and training data consisting of multi-scale image patches could further improve the learning descriptor and help remove the limitation, this is beyond the scope of this paper.

As in ORB-SLAM2, after SAFT features are extracted, the distance between map point descriptor and keypoint descriptor is utilized both in tracking the initial pose of the current frame and in further optimizing its pose by tracking it in the local map. For a static descriptor model such as ORB descriptor, a map point descriptor can be set to the keypoint descriptor that has the least median distance to all the others among all the keypoint descriptors associated with the map point.

However, for a dynamic descriptor model as we are using, changes in a landmark's description result from not only new viewing angles or environment variations, but also model updates, which further makes it less intuitive to define such a map point descriptor. On the other side, noticing that the pre-trained model (Figure 3.1) is a training result on a large offline dataset (Liberty subset [191]), online SAFT learning with recent neighboring observations can be viewed as a process to fine-tune the weights of the pre-trained model (Figure 3.1) and such a map point descriptor might still work. Indeed, as proved by the evaluation results in Section 3.5, this map point descriptor works reasonably well with the dynamic feature transform adopted in the learning framework.

In addition, the BoW vocabulary utilized in the original algorithm is replaced with a new BoW vocabulary created with the prediction network and DBoW2 library [125]. With the weights from the pre-trained model, the prediction network computes the learning descriptors at every key point detected by FAST detector in the images from the dataset Bovisa 2008-09-01 [195]. Then, all these descriptors are used to create the new BoW vocabulary. In order to be compatible with other parts of ORB-SLAM2, the new BoW vocabulary in text file format is created with 10-medians clustering, 6 vocabulary tree levels, term frequency-inverse document frequency (IF-IDF) weighting and L1 norm scoring [125]. Furthermore, the binary-based vocabulary storage algorithm in [198] is modified to deal with the float descriptor and the created BoW vocabulary is converted to its binary format for efficient vocabulary loading when the modified vSLAM algorithm starts running.

3.4.3 Online SAFT Learning

For online SAFT learning, the training network in Figure 3.4 is trained online with fixed learning rate 0.0001 and the same other training hyper-parameters used in offline training (Section 3.4.1).

3.4.3.1 Training Data Preparation

Considering different ranges of local scenes to learn from, training image patches can be prepared in two different ways in the tracking thread. One way is to crop the patches only from the last frame and the current frame, which generally produces fewer training data and allows the training network weights to be updated quickly. However, since the matching patches prepared in this way are too close to each other in neighboring frames, it would actually take lots of effort and time for the training net to learn meaningful information and improve upon the pre-trained model. In order to address this learning issue, a better strategy is adopted in which learning patches are cropped from all the keyframes that share at least one map point with the current frame. Compared with image patches from sequential frames, these learning patches are much farther away from each other in terms of viewpoint and include more representative information of the observed environment.

As discussed in Section 3.4.2, during the tracking process, matched feature points are initially found using SAFT's offline trained model at the beginning of the tracking process or the most recently updated online trained model after online SAFT starts working. False matches are removed by optimizing reprojection error in "Track Initial Pose by SAFT" and "Track in Local Map by SAFT" as shown in Figure 3.1 and also in the mapping thread (not shown in Figure 3.1). The ground-truth matching and non-matching image patch pairs are found by only cropping around the inlier observations after these geometric verification steps. For ease of expression, we just use "observation" to represent "inlier observation" in the following part of this section.

Figure 3.6 gives a visualization of training data preparation in the local map of the current frame, where MP is short for map point and P_i^j represents the image patch associated with the observation of the i^{th} map point in the j^{th} frame. Specifically, as shown in Figure 3.6, for each valid map point (MP #1, #2, #3, #4) observed in the current frame, first its observations in all the keyframes (KF #1 and #2) are searched for. With the benefit that in ORB-SLAM2 each map point stores the information about its observations from the keyframes in the keyframe database, this search can be done conveniently and efficiently. Then, for each map point that can be observed from at least one keyframe (such as MP #1, #2, #3 in Figure 3.6), the 64 × 64 image patches centered at its observations in the keyframes and the current frame are cropped and stored in a list associated to the map point. For example, the lists associated with MP #1 and MP #3 both include 3 image patches ($[P_1^0, P_1^1, P_1^2]$ and $[P_3^0, P_3^1, P_3^2]$ respectively), the list of MP #2 only has two image patches ($[P_2^0, P_2^1]$), and MP #4's observation list is empty since it cannot be observed from any keyframes.

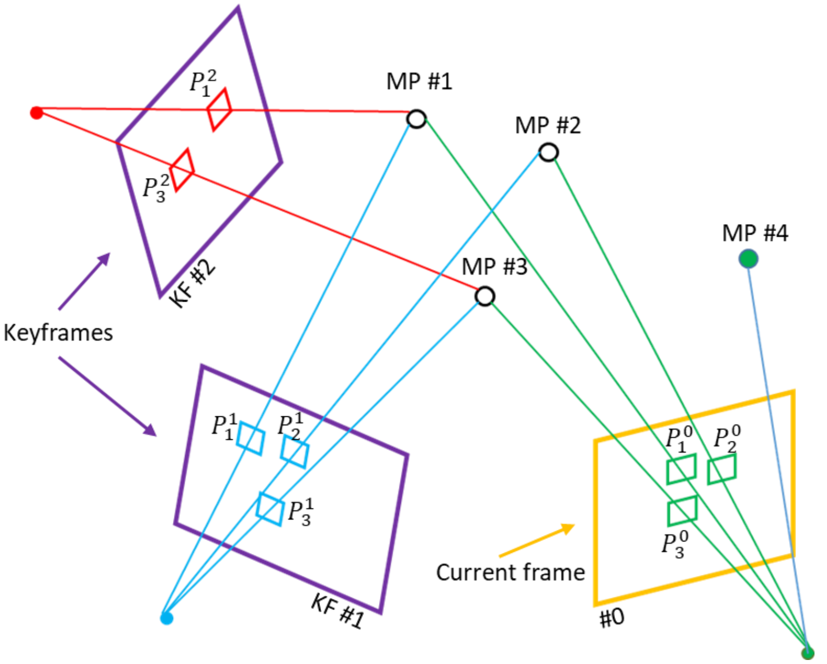


Figure 3.6 Visualization of training data preparation in the local map of the current frame

With the observation lists, a matching image pair with label 1 is created by randomly putting together two image patches in the same observation list. Similarly, a non-matching image patch pair with label -1 is created by randomly sampling two image patches from two different observation lists. Moreover, in order to balance the number of positive and negative pairs for the training purpose, equal number of matching and non-matching patch pairs are prepared each time. When such a patch preparation process finishes, the prepared patch pairs are inserted into the training data queue (Figure 3.1) for further processing. Note that this step happens inside the tracking thread.

3.4.3.2 Online SAFT Training

In the online learning thread, when new learning patches are detected in the queue, it would always choose to use the most recent patch pair sets for training and clear the remaining data in the queue, i.e., the queue is last-in-first-out (LIFO). This is because the core idea of online training is to intentionally use the network's strong representation power, or its “over-fitting" ability, to obtain optimized descriptions of the recent scenes. Thus, the training data observed from the most recent scenes are always preferred. However, in order to make full use of the limited data, the training data is randomly shuffled, and each patch pair is randomly flipped horizontally or vertically or rotated 90, 180, or 270 degrees on the fly just before the data is fed into the training net.

Compared with typical offline training, there are two key issues for online learning: 1) how to balance between the global, generic feature transform modeled as the pre-trained model and the more local, scene-specific ones modeled as the fine-tuned models; and 2) when to stop training and update the weights of the prediction net. For the former issue, since online learning is a fine-tune process with a small amount of data, it does not affect the learned local transform to deviate significantly from the global one. In particular, it is found that the starting training accuracy rate for each separate training process is almost always above 0.95 except for the first training process in our experiments. Therefore, there is no need to reset train net weights back to the pre-trained weights periodically to maintain a certain global feature transform. For the latter, if the training process can be done in real time, in order to obtain the best local descriptor, the ideal condition to stop ongoing training and share weights is when the training accuracy reaches 1.00 (intentionally utilizing overfitting). However, in practice, since the learning speed is relatively slow compared with the tracking speed, it results in discrete learning where big differences may exist between the scene to be described and the scene used for updating weights, and it is named discrete SAFT. A strategy to combat this delayed overfitting is to continuously update weights whenever the current learning data fed into the training net is completely consumed regardless of the current training accuracy, and this is named Continuous SAFT. We include a detailed comparison of the two methods (Discrete SAFT vs. Continuous SAFT) in the experiment section.

3.5 Experiments and Evaluation

It is challenging and labor-intensive to evaluate the performance of a vSLAM system in custom environments. On one side, this generally needs to deploy a motion capture system in a large-scale environment to get the ground-true pose of a camera. A high-accuracy motion capture system is usually quite expensive and not available. Moreover, it is also difficult to calibrate a motion capture system in a large-scale environment. On the other side, in order to fully evaluate a vSLAM's performance, it needs to be tested in various environments including different patterns of motion, texture, and structure. This makes an evaluation in custom environments even more difficult and time-consuming. Therefore, with the aim to facilitate vSLAM's evaluation and performance comparison, most researchers evaluate their vSLAM algorithms on public datasets that were collected in different types of practical or rendered environments. Even though we have proposed a marker-based vSLAM evaluation method in [158], we evaluated our algorithm on public datasets so that other results evaluated on the same datasets can be conveniently compared with those presented in this paper.

The proposed Deep SAFT is aimed to improve feature matching robustness and accuracy in feature-based vSLAM, and it works with monocular, stereo and RGB-D SLAM in ORB-SLAM2. Without loss of generality and for the sake of evaluation convenience, Deep SAFT is evaluated with RGB-D SLAM quantitatively and qualitatively against two baselines (handcrafted ORB descriptor, and offline trained static learning descriptor) on two popular public datasets (the TUM RGB-D dataset [138] and the ICL-NUIM dataset [196]) that are widely used to compare all well-known RGB-D vSLAM algorithms.

3.5.1 Quantitative Results

In order to analyze performance difference due to different modifications, ORB-SLAM2 RGBD (ORB2 RGBD), SAFT with pre-trained model without updating weights (Offline SAFT), SAFT with continuous learning strategy (Continuous SAFT), and SAFT with discrete learning strategy (Discrete SAFT) were evaluated quantitively on the same RGB-D SLAM benchmark datasets that provide trajectory ground truth, the TUM RGB-D dataset and the ICL-NUIM dataset. The TUM RGB-D dataset is one of the most widely used datasets for RGB-D SLAM evaluation. It includes various sequences collected by a handheld camera and a camera mounted on a robotic platform in practical environments such as office room, industrial hall, and some environments with dynamic objects or different structure and texture backgrounds. The sequences contain RGB images and depth images that were collected when a Kinect moved with different length of trajectories, average translational velocities and average angular velocities in different environments. The ground truth trajectory was obtained by a high-accuracy motion capture system that was deployed in the environments where the sequences were collected. However, for the ICL-NUIM dataset, an estimated trajectory in a real-world living room obtained by running Kintinuous [85] was used as the ground trajectory and corresponding RGB and depth image sequences were obtained from rendered scenes in POVRay [86] based on this trajectory. This dataset only contains eight sequences, four were collected in a rendered living room and the other four were collected in a rendered office room. Figure 3.7 shows some challenging scenes from the two datasets. Figure 3.7(A) is a motion-blurred scene caused by fast camera motion from fr1/room in the TUM RGB-D dataset. Figure 3.7(B) is a low-structured scene with several posters attached to a plane from fr3/nst in the TUM RGB-D dataset. Figure 3.7(C) is a low-textured scene including clustered key points, and Figure 3.7(D) is a low-textured scene with very few key points from or3 and or1 in the ICL-NUIM dataset respectively. The environment where the last two sequences were collected can be viewed as an indoor construction site.

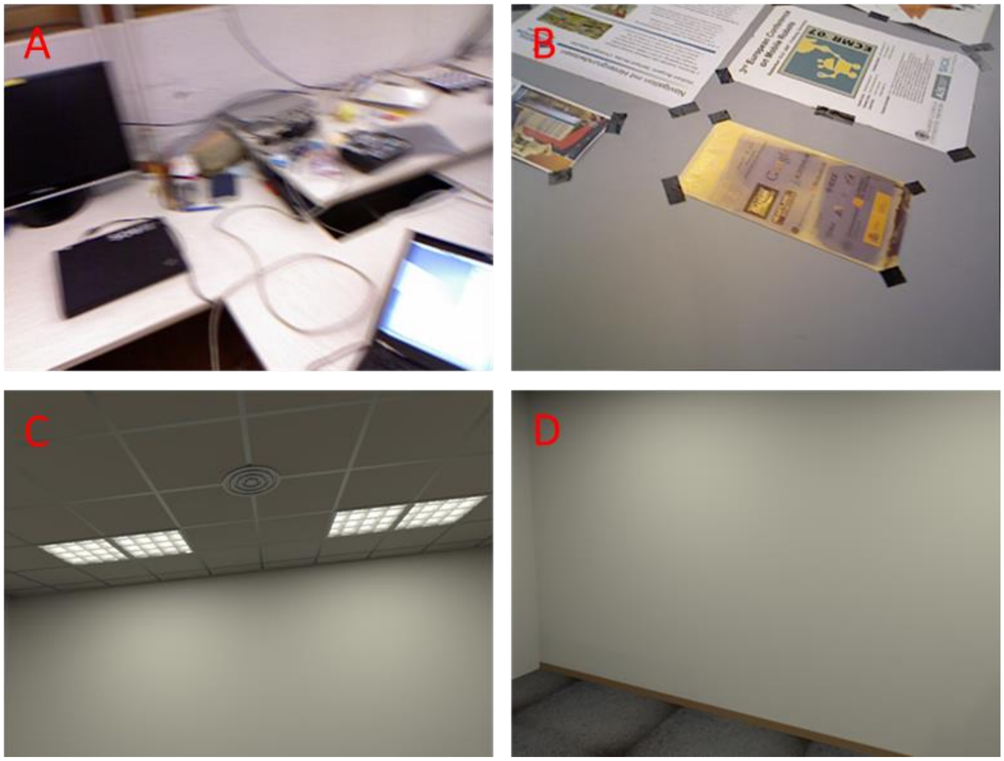
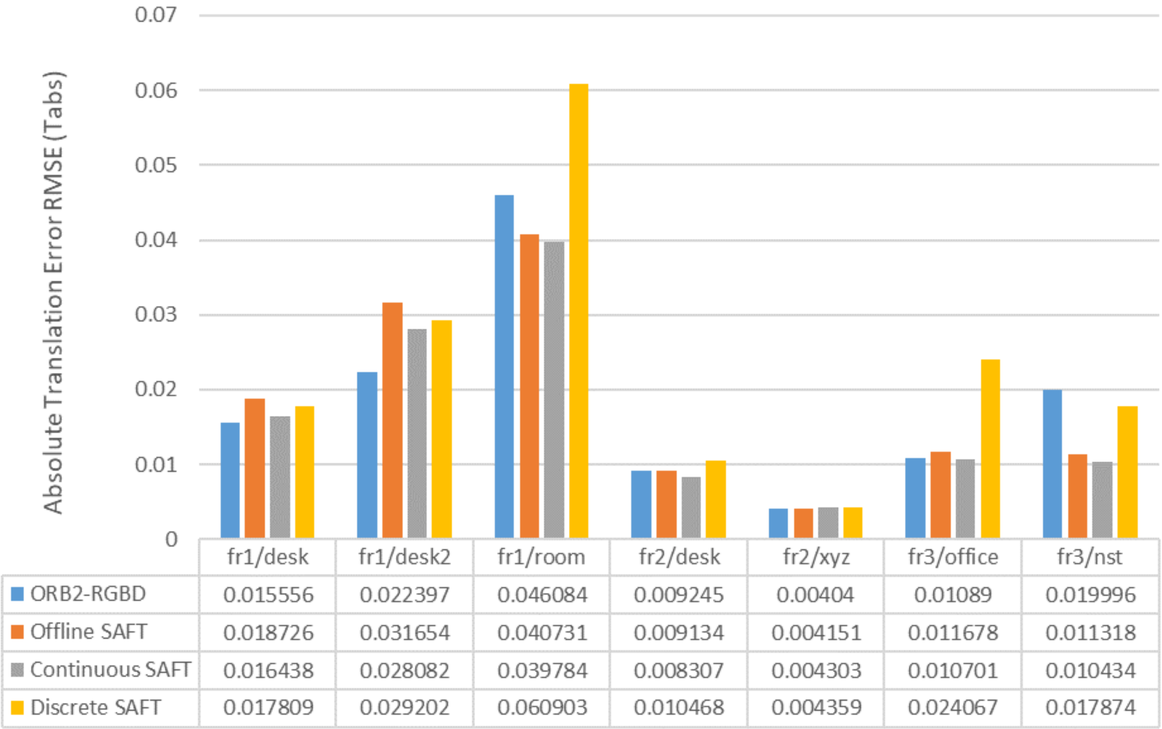


Figure 3.7 Some challenging scenes from the evaluation datasets

For evaluation purposes, different algorithms were evaluated on seven typical sequences from the TUM RGB-D dataset where most RGB-D SLAM methods were evaluated and all the eight sequences in the ICL-NUIM dataset. In addition, the absolute translation error RMSE (T_{abs}), which measures the distance difference between an estimated trajectory and its ground true trajectory [138], was used as the evaluation metric in the evaluations. Moreover, in order to conquer the randomness introduced by the multiple threads in the framework, for any algorithm, it ran five times on a sequence from the datasets and the median performance of the five runs was used as its performance on the evaluation sequence [122].

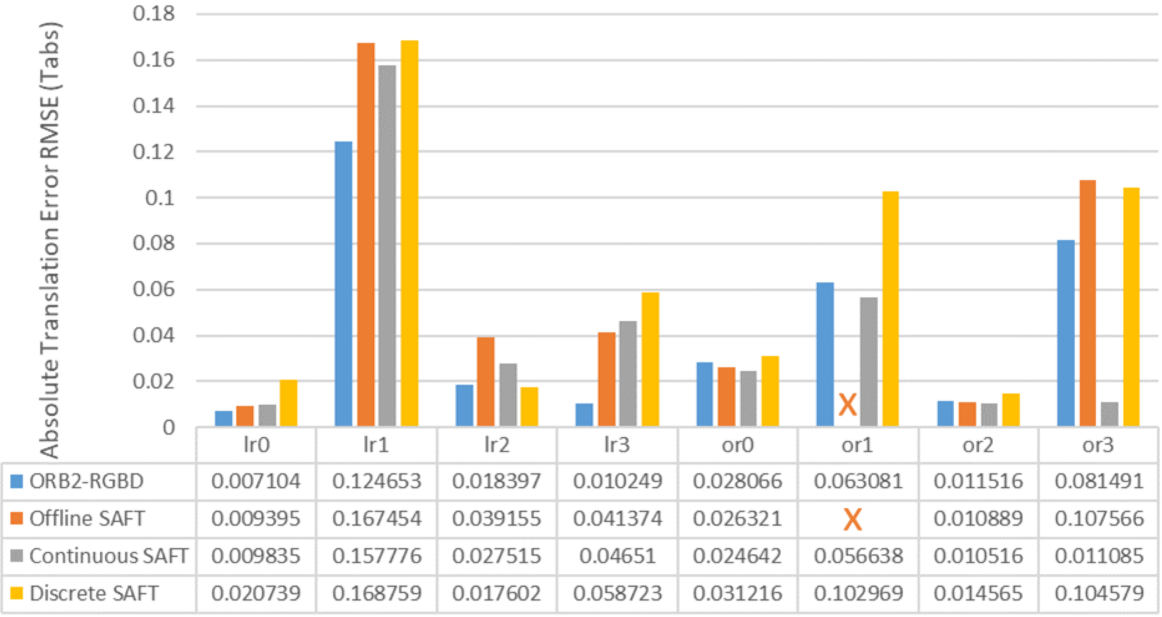
The corresponding evaluation results on the two datasets are shown in Figure 3.8 and Figure 3.9, respectively. It can be observed that Offline SAFT that directly uses the pre-trained model for feature matching generally performs worse than the baseline algorithm ORB2-RGBD. It is slightly worse than ORB2 RGBD on the TUM RGB-D dataset since it only wins on three sequences (fr1/room, fr2/desk, and fr3/nst) but loses on four sequences (fr1/desk, fr1/desk2, fr2/xyz, and fr3/office). On the ICL-NUIM dataset, it only wins on two sequences (or0 and or2) but loses on the other six sequences.

In addition, Offline SAFT loses too much on some sequences and thus cannot give robust and consistent performance in different situations. In fact, it failed to robustly track the camera pose for all its five runs on sequence or1 (office room 1). This indicates that the fixed pre-trained model cannot generalize well to the scenes in some sequences, even if the trained weights were tested with L2 norm with normalization and proved to outperform SIFT descriptor and ORB descriptor on the Notre Dame subset of [191]. The reason is that the image patches in the dataset [191] that were used to train the deep descriptor were almost always cropped around salient key points.



	fr1/desk	fr1/desk2	fr1/room	fr2/desk	fr2/xyz	fr3/office	fr3/nst
🟦 ORB2-RGBD	0.015556	0.022397	0.046084	0.009245	0.00404	0.01089	0.019996
🟧 Offline SAFT	0.018726	0.031654	0.040731	0.009134	0.004151	0.011678	0.011318
◼ Continuous SAFT	0.016438	0.028082	0.039784	0.008307	0.004303	0.010701	0.010434
🟨 Discrete SAFT	0.017809	0.029202	0.060903	0.010468	0.004359	0.024067	0.017874

Figure 3.8 Performance evaluation on seven sequences from the TUM RGB-D dataset



	lr0	lr1	lr2	lr3	or0	or1	or2	or3
🟦 ORB2-RGBD	0.007104	0.124653	0.018397	0.010249	0.028066	0.063081	0.011516	0.081491
🟧 Offline SAFT	0.009395	0.167454	0.039155	0.041374	0.026321	X	0.010889	0.107566
◼ Continuous SAFT	0.009835	0.157776	0.027515	0.04651	0.024642	0.056638	0.010516	0.011085
🟨 Discrete SAFT	0.020739	0.168759	0.017602	0.058723	0.031216	0.102969	0.014565	0.104579

Figure 3.9 Performance evaluation on the ICL-NUIM dataset

However, in the Offline SAFT, the FAST key points extracted at different pyramid levels were not always very salient ones due to lack of features or motion blur caused by rapid motion. This deviation from the training set caused the descriptor performance loss on the vSLAM test sequences. In order to address this issue, a natural solution is to include in the training set a certain percent of image patch pairs obtained under these challenging conditions. However, the problem is that it is very difficult since under such conditions no known features can achieve satisfactory matching performance and be used to prepare training patch pairs. Moreover, such a solution would become more difficult when it comes to challenging situations such as varying illumination or nontextured environments.

The online SAFT algorithms attempt to mitigate this problem in a different way. Considering that when a situation becomes challenging, it can be always viewed as a continuous process if the observed scene is perceived at a high frame rate. Therefore, if an algorithm can improve its own tracking performance with the perceived information before the situation becomes too challenging to deal with, such improvement should be able to help improve the algorithm's performance when the situation becomes really challenging. This idea is proved by the evaluation results of Continuous SAFT, whose performance is better than Offline SAFT on most of the test sequences. Especially, while Offline SAFT cannot work properly on sequence or1, Continuous SAFT achieves T_{abs} as low as 57 mm that is even better than ORB2-RGBD. However, it is not guaranteed that Continuous SAFT always achieves better performance than ORB2-RGBD. For example, if Offline SAFT loses too much to ORB2-RGBD on a sequence, with additional performance improvement, Continuous may still be worse than ORB2-RGBD. Considering the winning and losing times and the extent to which it wins or loses, Continuous SAFT's overall accuracy is quite comparable with ORB2-RGBD on the test sequences.

It is also worth investigating Discrete SAFT's performance. As observed in Figure 3.8 and Figure 3.9, sometimes it outperforms Offline SAFT and sometimes performs worse, but its performance is almost always worse than Continuous SAFT. The underlying reason is that the weights used by the prediction network (Figure 3.1) to describe a scene are always trained with previous scenes that are a certain different, this delay makes the inconsistency in Discrete SAFT's performance. It only works well when the scenes that are used to update the weights are close in appearance to the future scenes the updated model is used to describe. However, this is not the case most of the time, and it is difficult to determine whether a learned descriptor becomes obsolete or not. This concludes that training SAFT online with the most recent images observed from the scene is critical to the performance of the SAFT descriptor.

3.5.2 Qualitative Results

Figure 3.10 shows the ground-truth trajectory (blue) in fr1/room sequence from the TUM RGB-D dataset and the estimated trajectories (red) from ORB2-RGBD (top left), Offline SAFT (top right), Continuous SAFT (bottom left) and Discrete SAFT (bottom right). On this sequence, both Offline SAFT and Continuous SAFT outperform the baseline algorithm. As shown in the figure, since the pre-trained model generalizes well in this environment, Offline SAFT outperforms ORB2-RGBD at Place 1 and achieves better performance than ORB2-RGBD.

In addition, with continuous weight update, Continuous SAFT further boosts the performance by enabling more accurate loop closure and improving the alignment at Place 2. However, for Discrete SAFT, compared to Offline SAFT, using delayed weights causes a larger deviation from the ground-truth trajectory at Place 3 and results in worse performance.

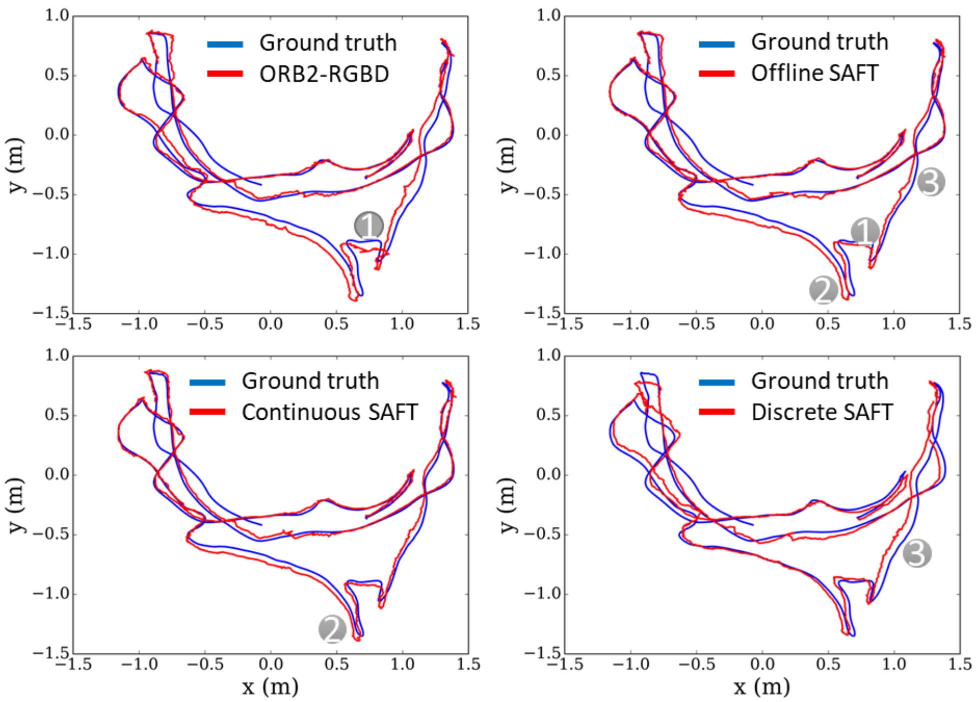


Figure 3.10 Evaluation results on the fr1/room sequence from the TUM RGB-D dataset

Figure 3.11 shows the ground-truth trajectory (blue) in or3 sequence from the ICL-NUIM dataset and the estimated trajectories (red) from ORB2-RGBD (top left), Offline SAFT (top right), Continuous SAFT (bottom left) and Discrete SAFT (bottom right). As shown in Figure 3.11, it is particularly apparent that the online learning process can help improve the robustness and accuracy of feature tracking. Sequence or3 typically includes similar low-textured scenes to Figure 3.7(C). For such low-textured scenes, the algorithms can still extract tens of key points in each frame. However, these key points are mainly concentrated in several small areas in a frame (the two areas showing the lights on the ceiling for Figure 3.7(C)) and not well distributed. This clustered key point distribution decreased the algorithms’ accuracy [122]. Moreover, low texture with repeated patterns further increase feature matching difficulty and increase the rate of false matches. Due to the performance difference between ORB descriptor and Offline SAFT descriptor, ORB2-RGBD cannot track features robustly at the beginning and the end of the sequence, while the pre-trained weights used by Offline SAFT works well most of the time but the tracking gets lost around the middle of the trajectory. By leveraging the perceived information, Continuous SAFT significantly improves the feature tracking performance and avoids this tracking loss.

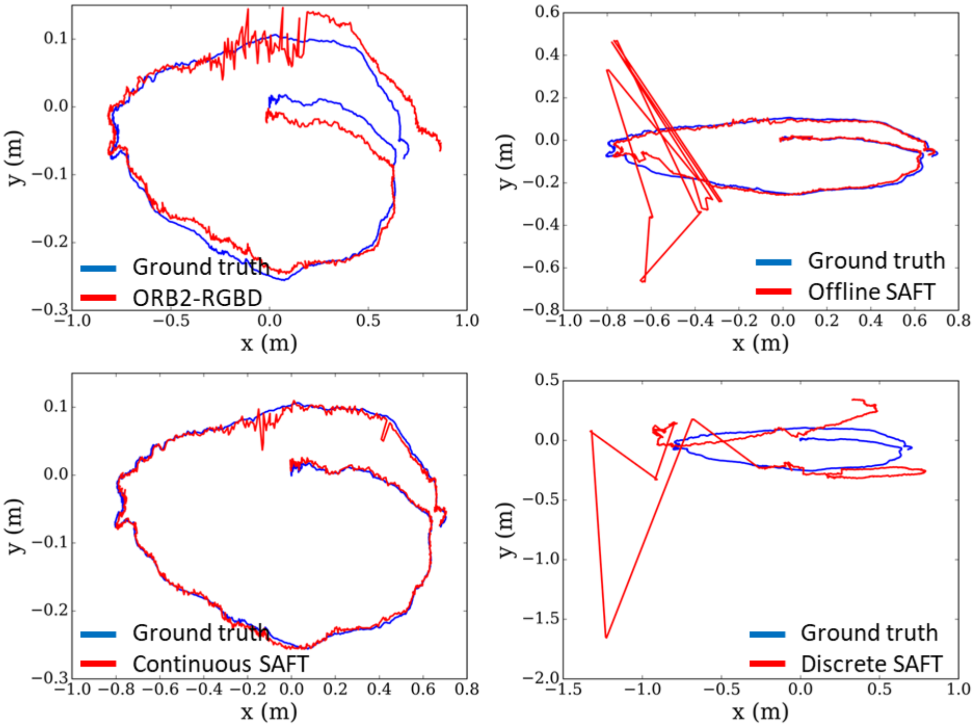


Figure 3.11 Evaluation results on the or3 sequence from the ICL-NUIM dataset

However, due to its limitation by low weight update frequency (this will be analyzed in detail in Section 3.5.3), Continuous SAFT cannot achieve satisfactory performance in some sequences where challenging scenes appear fast. The sequence or1 is a good example of this situation. Figure 3.12 shows the ground-truth trajectory (blue) in or1 sequence from the ICL-NUIM dataset and the estimated trajectories (red) from ORB2-RGBD (top left), Offline SAFT (top right), Continuous SAFT (bottom left) and Discrete SAFT (bottom right). As shown in Figure 3.12, even though Continuous SAFT achieves better tracking performance at the end of the sequence and thus gives a lower T_{abs} compared to ORB2-RGBD, it suffers from some tracking lost at the bottom-left corner of the trajectory (Figure 3.12 bottom left). Around this place, the scene changes from a ceiling+wall+ground scene to a wall+ground scene (as shown in Figure 3.7(D)) in about twenty frames. From these frames, algorithms can only extract a limited number of key points and low texture makes it very difficult to match these key points, and the large error in feature matching corrupts the estimated trajectory for Offline SAFT. For continuous SAFT, since it lacks features on the wall, most key points were extracted from the ceiling at the beginning of this change and the Continuous SAFT weights were updated with these feature matching. At the end of this change, all the key points were from the ground plane, and the updated weights were used to match these key points. Notably, the key points from the ground plane are different from the key points from the ceiling and need a different descriptor. This further caused continuous SAFT's performance loss at this place.

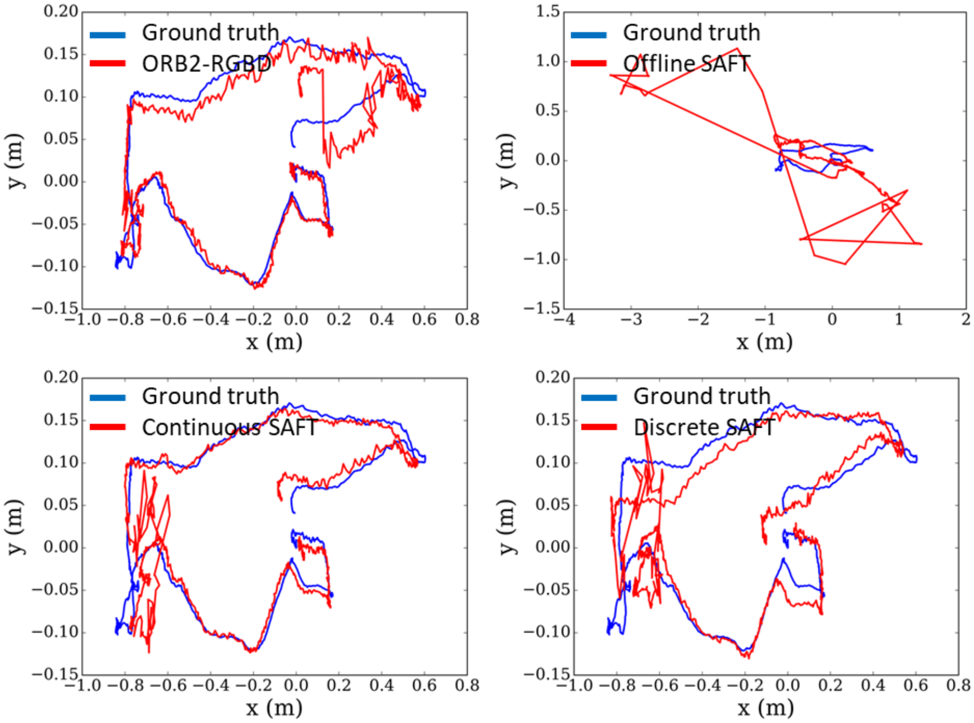


Figure 3.12 Evaluation results on the or1 sequence from the ICL-NUIM dataset

Besides the overall trajectories, it is also interesting to think about how the online learning process improves the quality of feature matching when it works against its offline version. Due to the limitations on the scale, rotation and deformation invariance, the offline trained descriptors can be only used to match scenes that are close enough to give close descriptors. However, for SAFT, even if a pre-trained model may not work well for a specific scene, as mentioned in the method section, it actually learns from feature correspondences in a local map representing a local scene. These correspondences are more accurate and reliable since they are selected after being filtered by multiple steps of geometric verification (as discussed in Section 3.4.3.1). Moreover, besides close correspondences, they include some correspondences that are somewhat far away and cannot be recognized as correspondences directly by the offline descriptor. By learning from a local map, compared to Offline SAFT, Continuous SAFT should be able to recognize more correspondences in challenging situations where the features to be matched are further away from each other.

To intuitively prove this, we tracked the inlier ratios after estimating initial pose and after tracking in local map in the tracking process for both Continuous SAFT and Offline SAFT and show the results in Figure 3.13. Figure 3.13 shows matching inlier ratios in the tracking process for Continuous SAFT and Offline SAFT on or3 sequence (first 1000 frames). It can be observed that for non-challenging scenes the inlier ratios of the two algorithms are very close. However, in some challenging frames that are marked with blue circles, Continuous SAFT achieves much higher inlier ratios for both initial pose estimation and further pose estimation after tracking in local map. This adds robustness and accuracy especially when the scene is challenging.

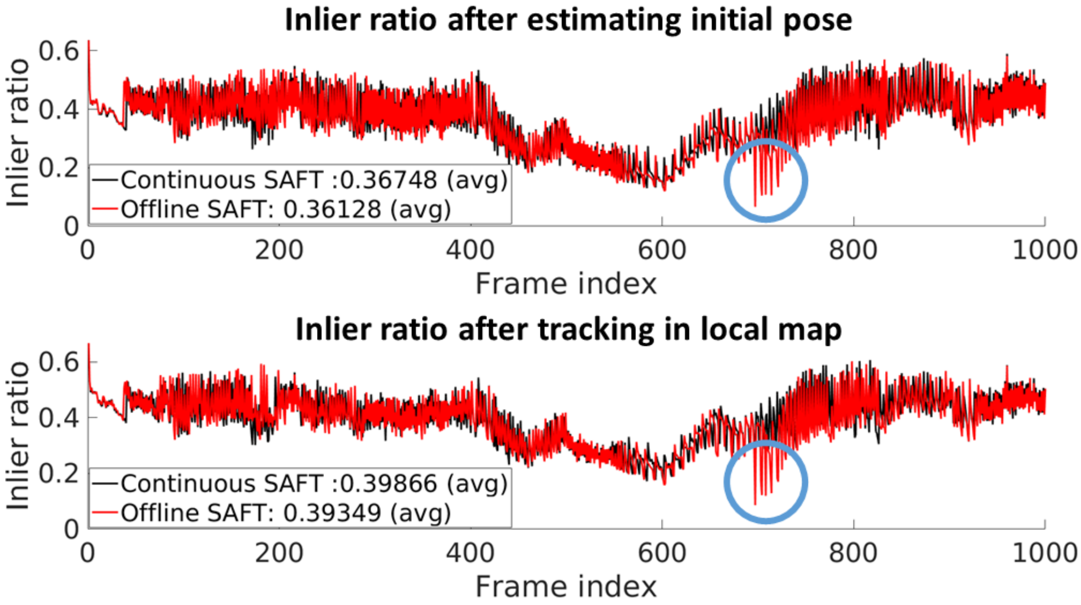


Figure 3.13 Matching inlier ratios in the tracking process on the or3 sequence

3.5.3 Computational Performance

One limitation of this implementation of deep SAFT is its reduced computational performance. Table 3.1 shows average time cost for descriptor computation (not including image patch preparation) and tracking process, which were evaluated on an Intel Core i7-4790K CPU@ 4.00GHz and a GeForce GTX970 GPU. For the SAFT related algorithms, network training and descriptor computation were processed on GPU. In the training process, in order to use the training data efficiently, we always let the network forward a certain number of steps first to consume the data fed into the MemoryData input layer and then check the trained accuracy. In this process, the corresponding time cost is not well controlled, and it is difficult to measure the exact value, so we use weight update frequency instead to give a qualitative evaluation of the training part.

As shown in Table 3.1, without an online learning process, Offline SAFT can achieve the speed of $4\frac{1}{6}$ fps, and the performance reduces to 4 fps for Continuous SAFT and $3\frac{1}{3}$ fps for Discrete

SAFT, which are both much slower than the baseline algorithm. For the SAFT, most time was spent on computing SAFT descriptors. Possible solutions to reduce this cost includes using lower-dimensional descriptors by replacing the network in Figure 3.4 with a smaller one or with another learning architecture that can be updated more efficiently. Another way to increase the processing speed of SAFT embedded vSLAM is to separate "training data preparation" from the tracking thread and to put it in a separate thread. These are possible implementation strategies, and the main point highlighted here is that SAFT can bring accuracy and robust benefits to feature-based vSLAM.

Table 3.1 Computational performance evaluation

Algorithm	**Descriptor computation** (ms)	**Tracking** (ms)	**Total** (ms and fps)		**Weight update frequency** (frames/1 update)
ORB2-RGBD	9.5×10^{-1}	1.6×10^{1}	2.7×10^{1}	$37\frac{1}{27}$	--
Offline SAFT	2.0×10^{2}	2.6×10^{1}	2.4×10^{2}	$4\frac{1}{6}$	--
Continuous SAFT	2.0×10^{2}	2.6×10^{1}	2.5×10^{2}	4	28
Discrete SAFT	2.0×10^{2}	2.6×10^{1}	3.0×10^{2}	$3\frac{1}{3}$	24

3.6 Conclusions and Future Work

This paper proposed a learning-based dynamic descriptor SAFT, which is a dynamic feature transform that can learn to achieve a better description of recently observed scenes. As a proof of concept, we chose to implement it using a deep neural network and integrated this deep SAFT into a feature-based vSLAM (ORB SLAM2) by completely replacing its ORB features. The evaluation results demonstrated the feasibility of Deep SAFT, performance improvement compared with its offline trained version, and the comparable performance of Deep SAFT embedded ORB-SLAM2 with the original ORB-SLAM2, which enables a further step towards improved vSLAM applications in challenging environments for real missions.

The proposed SAFT can be conveniently implemented to work with existing feature-based vSLAM algorithms using monocular, stereo, or RGB-D cameras and allow for more economical and robust localization for applications in the built environment or on construction sites. Such applications include automatic joint filling on construction sites as discussed in [80], automatic point cloud registration for construction progress monitoring [13], automatic bridge bearing inspection [6], and automatic tunnel inspection [14].

Even though SAFT can be used to improve any learning-based descriptors, its practical performance depends on custom implementation. For the example implementation of SAFT in the paper, the computational efficiency of the descriptor network resulted in delayed scene description and is the main factor that limited its performance. This can be further improved by compressing and accelerating the present network [189,190] or replacing it with a more lightweight network that can be updated more efficiently. This will also help reduce the computational cost of the implemented SAFT embedded vSLAM. Its tracking cost can be also improved by separating the operation of preparing training data from tracking thread and making it a separate thread.

Besides detailed implementation, another limitation is that the current version of SAFT is only trying to learn a better descriptor without putting efforts on dynamically adapting the feature detector. Thus, for the environments where it is difficult to extract key points, it can only make better description of the limited key points. Our next step is to design a pipeline network to learn to adapt both detector and descriptor to further improve vSLAM performances.

Chapter 4

Automatic Extraction of 1D Barcodes from Video Scans for Drone-Assisted Inventory Management in Warehousing Applications

4.1 Introduction

One-dimensional (1D) barcodes are widely used for product identification and inventory management in supply chains and retail transactions. Compared to two-dimensional (2D) barcodes (e.g., Quick Response (QR) codes), even though 1D barcodes can only contain basic information, their redundant design provides improved readability in situations of partial tear or abrasion, making them robust and reliable in harsh industrial environments [199]. The utilization of 1D barcodes has thus represented a significant milestone in automated stock and inventory management. Notwithstanding, barcode scanning is largely a human effort-intensive process since a worker typically has to manually focus a barcode scanner (handheld or equipped on a forklift, Figure 4.1) on all codes to be read one by one, and from close proximity. This makes their application suitable to situations where relatively small numbers of barcodes must be scanned such as store checkout lanes, but not in situations where large numbers of laterally-distributed barcodes have to be regularly scanned for inventory management or stock-keeping in warehouses or distribution centers.

Long-range barcode scanners offer a potential solution in such industrial environments. However, their applicability is limited due to several practical issues that include small viewing angles (i.e., closely spaced racks result in too small viewing angles for reading barcodes at high places), and sight occlusion (i.e., product barcodes are occluded by other products or shelves and rack components). Thus, even with long-range barcode scanners, a barcode scanner has to get within close vicinity of all codes that need to be scanned, resulting in the more practical use of standard-range barcode scanners having a range of 6 to 24 inches [200]. In addition to significant scanning workloads, workers in warehouse-like environments face several other challenges. For instance, for all products stored above ground level on racks or shelves, workers have to use ladders, lifts, or forklifts to visually access and scan barcodes (Figure 4.1), significantly increasing risks of falls or other injuries and causing general waste of energy in operating forklifts or other lift platforms.

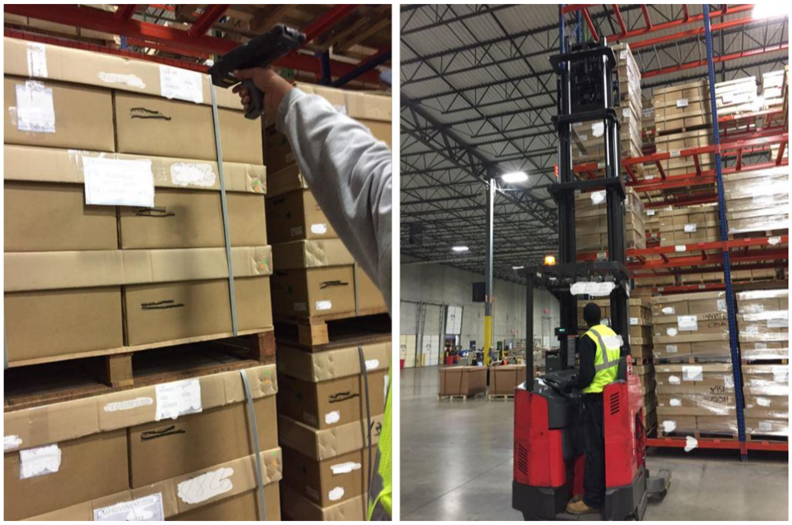


Figure 4.1 Manual barcode scanning in typical warehouse environments

Besides such issues, the large scale of effort involved in barcode scanning in warehouses also presents a strong case for automation. For example, a typical warehouse supporting a manufacturing supply chain has hundreds of sections and thousands of racks, most of which hold high turnover products (i.e., products come in and go out quickly over a matter of hours or days). In this situation, inventory has to be scanned multiple times in a week or sometimes at least once a day, which is a very laborious and time-consuming job demanding a team of employees. A promising idea towards automation of such inventory management is to mount a barcode scanner on a drone and manually fly the drone to scan barcodes.

As is estimated in [201], in a warehouse environment, a drone operator can scan 119 times faster than a person using a handheld barcode scanner. This solution can not only greatly improve operation efficiency, but can also liberate workers from this laborious and dangerous work while also conserving energy (the energy consumed by a flying drone carrying a barcode scanner is much less than that needed for lifting a heavy forklift platform). However, the idea to scan barcodes with a drone-mounted barcode scanner is in essence still a line-of-sight scan, which requires the drone to pause momentarily in front of each barcode for reading [201]. On the one hand, this stop-and-go scan pattern dictates that the drone has to fly at a very low speed making the scan process very time-consuming. In addition, the high positioning accuracy requirement for drone hovering presents a major challenge for current self-navigation algorithms and further limits its application in complete automatic scans.

4.2 Technical Approach and Related Work

To mitigate these issues, the proposed method scans barcodes with a video camera that can both enable area-of-sight scan and reduce the high requirement for positioning accuracy, making it suitable for a completely automatic scan at a relatively high speed. With the help of vision-based barcode reading and drone navigation algorithms, our overall solution is to automatically scan a warehouse with a drone-mounted camera and extract barcode information from the obtained video, while requiring little human assistance for monitoring, verification, and maintenance. Figure 4.2 presents an overview of the whole system. In this overall automatic scan solution, to automate the whole process, the barcode scanning task is divided into two low-level tasks of automatic video data collection and automatic barcode extraction that make up the task layer. These two tasks are further implemented and supported by the underlying algorithms listed in the algorithm layer. In addition, right above the task layer, humans are only responsible for high-level tasks of monitoring, evaluating, and maintaining the two low-level sub-tasks, such as drone state monitoring, barcode verification, and system maintenance that make up the human layer.

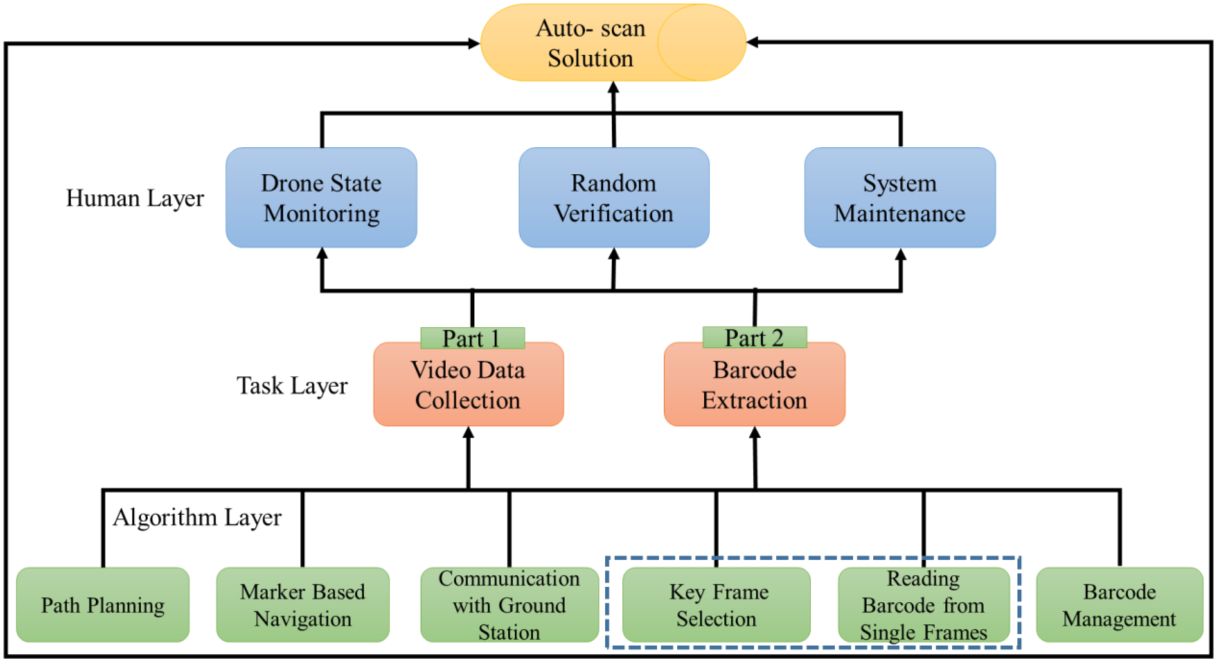


Figure 4.2 Overview of the automatic scan solution

This paper primarily focuses on techniques for extracting barcodes from arbitrary sequences of scanned video data (enclosed by the dashed box in Figure 4.2), and this is a key component of our overall solution. By building on existing well-developed barcode decoding methods, our algorithms focus on improving recognition rate and efficiency by developing methods for preparing easy-to-decode barcode regions. In particular, our method efficiently processes video sequences containing an unspecified number of barcodes oriented in arbitrary directions and located in any part of the frames.

The steps followed to obtain such ideal barcode regions from a video scan are shown in Figure 4.3. To efficiently process multiple frames with overlapping scenes, in the first step, fewer frames (called key frames here) that do not miss any barcode information need to be selected for further processing. Then the problem that remains is how to read multiple barcodes from a single key frame. This can be further solved by the following two steps: recognizing potential barcode regions in a frame and adjusting the direction of each of these barcode regions for decoding. In an effort to provide a clear description of this paper's contributions, these three steps will be discussed in reverse order (also the order in which they were developed) compared to the sequence shown in Figure 4.3.

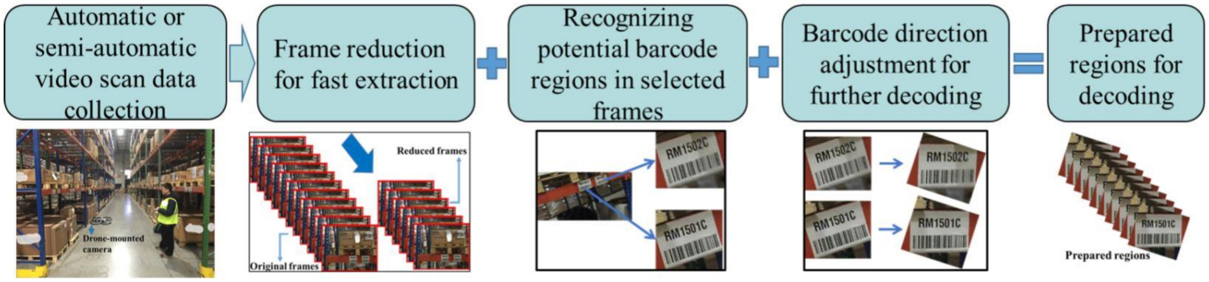

Automatic or semi-automatic video scan data collection
Frame reduction for fast extraction
Recognizing potential barcode regions in selected frames
Barcode direction adjustment for further decoding
Prepared regions for decoding
Drone-mounted camera
Reduced frames
Original frames
RM1502C
RM1501C
Prepared regions

Figure 4.3 Process of preparing barcode regions for existing decoding algorithms

With the popularity of barcodes as a tagging system, significant prior work has been done on reading barcodes using computer vision-based methods. Initially, barcode reading algorithms were mainly implemented on desktop computers based on domain transformation, such as the Fourier transformation or the Hough transformation as proposed in [59]. Compared with domain transformation, reading algorithms using scanlines need less computational resources and can effectively run on mobile devices, which has resulted in their rapid development recently [60-62].

In addition, there also exist some algorithms for reading challenging barcodes caused by low resolution, motion blur or out of focus [63,64].

However, most of these algorithms are only applicable to vertical or approximately vertical barcodes (Figure 4.4A), which greatly limits their wide application in practice. In addition, even though some commercialized algorithms such as the ClearImage Barcode Reader SDK (referred to as ClearImage hereinafter) [65] already provide certain abilities to read rotated barcodes (Figure 4.4B and Figure 4.4C) from an image, their performance is significantly limited for blurred images. Instead of focusing on decoding a barcode itself, this component of our proposed solution focuses on estimating barcode orientation in an image in an effort to make existing decoding algorithms more effective.

Many methods have been developed to solve this problem. In [61,202], barcode direction was determined by the intersections of scan lines and bars. In [203], the main direction was estimated by using an orientation filter in four directions. Besides, Hough transformation has also been used [204,205]. However, these methods are either not robust enough to detect arbitrarily rotated barcodes, or are not time-efficient, or are too complicated to be implemented. Taking into consideration that Hough transformation alone does not work well in situations of complex spatial context or high image noise, we propose to use corner detection and Hough transform together to implement a robust, efficient, and easier solution.

With the development of barcode reading techniques, barcode localization algorithms have also experienced significant progress. Compared with finding a single barcode in an image [206-209], we are more interested in the ability to simultaneously recognize multiple barcodes of any size and orientation, which is more suitable for the motivated application in warehouse settings. Based on morphological operations, Lin et al. [210] implemented their barcode detection algorithm by background small cluster reduction. Other work such as Bodnar et al. [211] used image primitive operations and detected barcodes relying on distance transformation. Such algorithms rely on basic image operation, and their performance is sensitive to threshold parameters that are not easy to find. Besides, methods using machine learning [212] or Maximal Stable Extremal Region (MSER) [213] detection have also been proposed for this problem.

All of these methods have either been tested with non-public image datasets or public image datasets where the barcodes take up a large portion of the whole image in each frame. In addition, the images used typically have a simple background and appear in specific patterns, thereby providing few insights about these methods' performance in complex practical environments. In order to address this, we propose a barcode region detection algorithm based on connectivity and geometry properties of barcode areas, which can work effectively and efficiently on real warehouse videos as well as find potential barcode regions beyond the reading ability of subsequent decoding algorithms such as ClearImage that is chosen in this work for decoding barcodes. It should be noted that some primitive image operations are used in our method, but in our case, finding appropriate thresholds is easier for extracting barcodes from consecutive frames under similar illumination conditions. The difficulty is how to get rid of a large number of redundant frames to improve efficiency. This problem is addressed by the last technique introduced in this paper.

Selecting fewer key frames that can represent the content of a video can not only help improve barcode reading efficiency but can also assist human verification. Such techniques are usually used for movie abstraction [214,215]. The difference and difficulty of our case are that there are lots of similar and repetitive scenes in a warehouse that makes a selection using features very difficult, thereby rendering feature-based algorithms ineffective even if they work well for traditional movie abstraction purposes [216,217]. For applications in this challenging environment, we propose to choose key frames based on histogram difference. This algorithm enables the use of color information from the whole region of a frame, which makes it more robust compared with extracted features. In the next section, each of these three algorithms that help improve barcode extraction from video frames is discussed in detail.

4.3 Technical Approach Details

In this section, three algorithms are proposed to improve the process of extracting barcodes from arbitrary video frames corresponding to the three steps in Figure 4.3, i.e., barcode direction estimation, barcode region detection, and key frame selection.

4.3.1 Barcode Direction Estimation

General methods of decoding barcodes from images are based on the encoding rule to find the best representation of the binary patterns sampled along scanlines that move from top to bottom of barcode areas. To be able to read out barcode information, there has to exist at least a readable region in which a horizontal scanline intersects with all bars. In addition, since the scanline usually moves down with a fixed distance at each step, the larger the readable region is, the more chance that the barcode can be successfully read. For this reason, algorithms proposed in [200,218] are only limited to processing the situations where the bars of barcodes are close to the vertical direction. However, similar algorithms would become more valuable, and their applications would significantly broaden if, prior to decoding barcodes, the images could be preprocessed by an angle-aware rotation through which they can be adjusted to bring them to the ideal state shown in Figure 4.4A from prior states such as Figure 4.4B or Figure 4.4C.

In Figure 4.4, where solid blue lines represent the margin of readable regions, solid red lines represent valid scanlines, and red dash lines represent invalid scanlines. Figure 4.4A is the ideal state where the barcode can be read from any scanline between the top and the bottom of the barcode. Figure 4.4B is a suboptimal state where the readable region still exists but is really small. Figure 4.4C represents the worst situation where there is no readable region anymore and the barcode cannot be read out with any horizontal scanline.

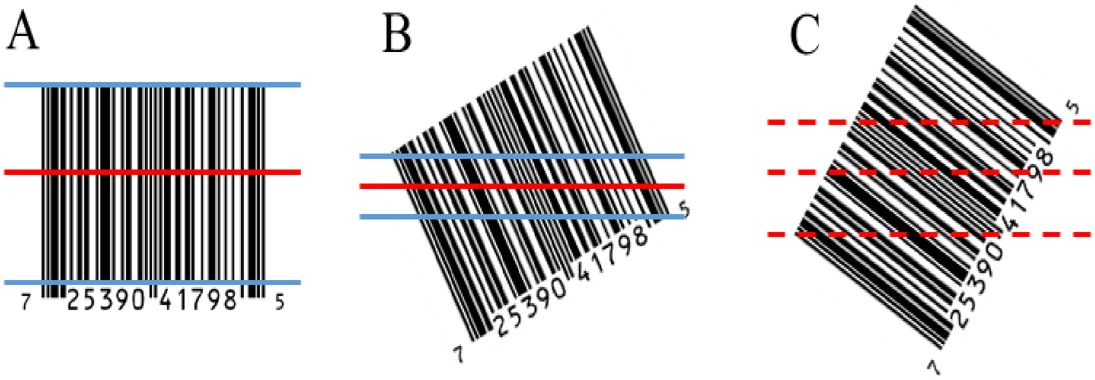


Figure 4.4 Readability of the same barcode in different angular states

To estimate barcode direction, Hough transform is generally used to recognize bar features (straight lines) in an image [59,204]. Instead of traditional representation of straight lines, it uses the Hesse normal form $r = xcos\theta + ysin\theta$ [219] and thus associates each straight line with a parameter pair (r, θ), where r is the distance from origin to the straight line to be represented and θ is the angle between x axis and the line passing through the origin as well as perpendicular to the line. It follows that in the (r, θ) space, representation of all straight lines passing through point (x, y) forms a sinusoidal curve and the intersection of such curves gives the (r, θ) parameter of the straight line connecting the points corresponding to the intersected curves. Intersection multiplicity values at different (r, θ) parameters form a parameter space matrix (also called Hough space) whose rows and columns correspond to r and θ values, and this matrix describes the voting scores for all (r, θ) values in the space [219].

With this benefit, straight lines can be found by selecting the parameter points in Hough space with big intersection multiplicity values. Since such intersection multiplicity values are found using a voting strategy, Hough transformation enables discontinuous lines (due to noise, reflection, etc.) to be recognized. However, it was found in our experimentation that Hough transform alone cannot work robustly if image noise is relatively large or repetitive patterns appear in a barcode's background. In such situations, the barcode direction is usually drowned by noisy directions, making it hard to be distinguished. Some researchers recently proposed to identify a characteristic pattern in (r, θ) space using machine learning [205,212], but such solutions need significant training data preparation effort.

Noticing that a large number of corners exist at the bar ends, the idea here is to first recognize these corners and use Hough transformation on such corner features instead of on the original image. The main reason why this works is that the corners extracted at bar ends in most cases are arranged perfectly in straight lines with high density, which makes the straight lines passing through these points have the largest votes in Hough transform and can be easily and robustly found. From computer vision perspective, a corner point should be easily recognized by looking at intensity values within a small window, and a small shift of the window in any direction should yield a large change in appearance. To find those corners at the end of the bars, Harris corner detector is applied, which finds corner points by evaluating the weighted squared sum of intensity change in a small window and approximating the intensity change in the first order [220].

The detailed results of estimating barcode direction are shown in Figure 4.5. It first converts an original RGB image in [221] (Figure 4.5A) to a grayscale image and finds corners with Harris corner detector (Figure 4.5B). In Figure 4.5B, it is clear that a large number of corner points at bar ends are detected as shown in the visualization in Figure 4.5C. Then Hough transform is applied on these corner points, and Hough peaks (limited to at most 20) are found in Hough space, i.e., (r, θ) space (ρ represents r)(Figure 4.5D). After that, the peaks are put into ten even spaced bins between the minimum and maximum value of the θ coordinate of the peaks, and the center of bins containing the maximum number of peak points is considered as the direction perpendicular to bar direction (Figure 4.5E). In addition, the barcode is also rotated to the ideal state by the corresponding angle (Figure 4.5F).

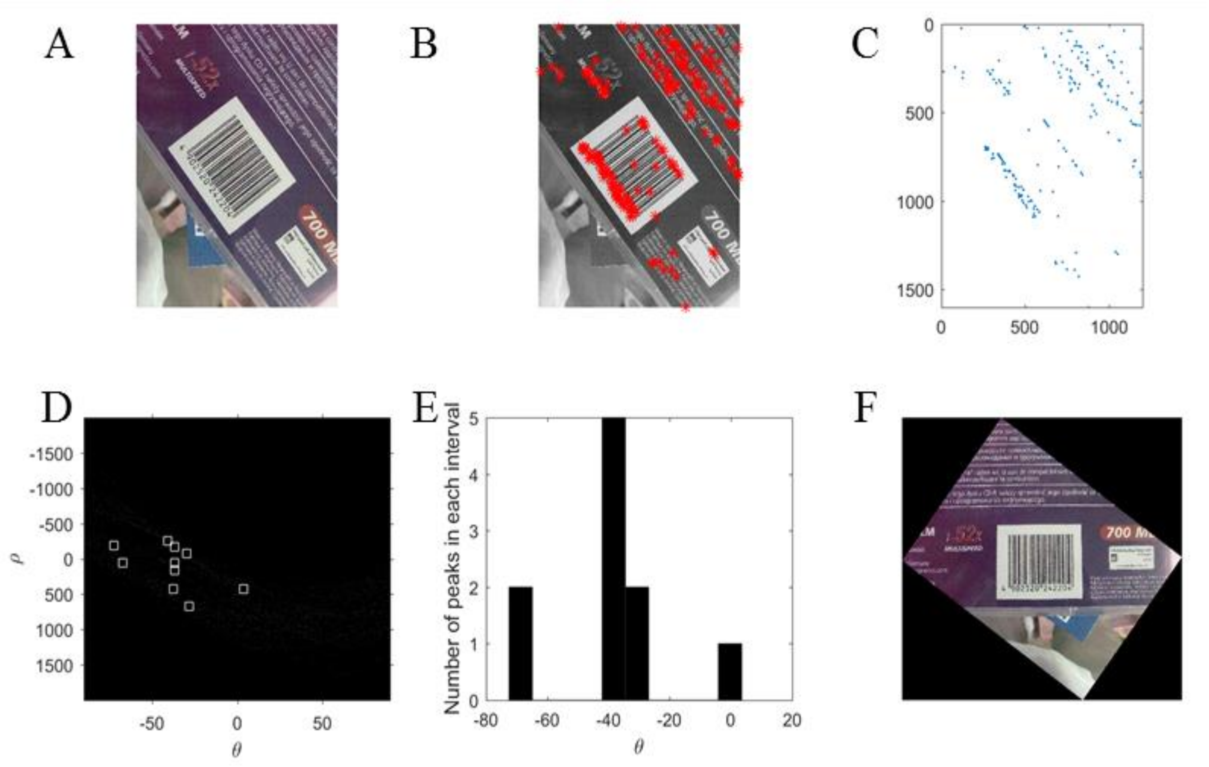


Figure 4.5 Procedures for barcode direction estimation

This algorithm is straightforward to implement and works robustly with one single barcode. For images that include multiple barcodes, potential barcode regions have to be identified and selected first before this direction adjustment algorithm can be applied. This aspect of our proposed method is discussed next.

4.3.2 Barcode Region Detection

In the present time, there is little difficulty in recognizing a barcode with a mobile phone camera or reading barcodes from most public barcode datasets when barcodes are usually intentionally focused on and occupy a relatively large part of a whole image. Different from such situations, the difficulty in our situation arises mainly from multiple barcodes with unexpected direction existing in one frame and a much smaller portion of separate barcode regions. The fallout of this situation is that in the decoding phase, significant time has to be spent on searching recognizable barcodes in the whole image. Our proposed idea is to help find potential barcode regions for the decoding algorithm and thus save time by avoiding the processing of non-value-adding regions.

To identify barcode regions in an image, the most intuitive idea is to see whether a certain number of parallel straight lines come together in a local region. However, detecting bars is very sensitive to image noise and similar line structures in the background, which makes it unreliable to use in practice. Instead of detecting straight lines, we propose to recognize barcode regions through their following properties: connectivity, quadrilateral contour as well as least area to be decoded, which is more robust, scale-invariant and applicable to find multiple barcodes. In order to better articulate how this process works, the flowchart of this barcode region detection algorithm and its results after key steps on a given image stitched by four different images from [221] are shown in Figure 4.6 and Figure 4.7 respectively.

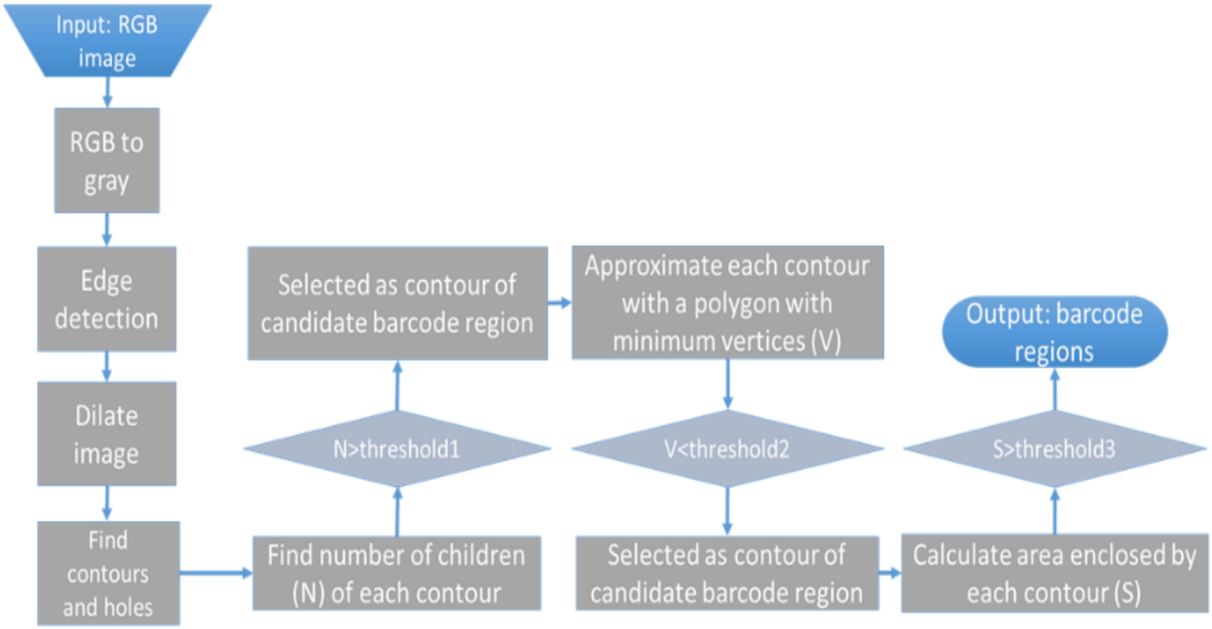

Input: RGB image
RGB to gray
Edge detection
Dilate image
Find contours and holes
Find number of children (N) of each contour
N>threshold1
Selected as contour of candidate barcode region
Approximate each contour with a polygon with minimum vertices (V)
V<threshold2
Selected as contour of candidate barcode region
Calculate area enclosed by each contour (S)
S>threshold3
Output: barcode regions

Figure 4.6 Algorithm for barcode region detection

Figure 4.7A shows an original image including barcodes with different backgrounds. As is shown in the flowchart (Figure 4.6), the RGB image is first converted to a gray image and then edge detection (makes barcode regions convenient to be detected by highlighting their edges and bars included) and dilation (helps close some discontinuous parts in edges of barcode regions) are performed. This result is shown in Figure 4.7B, from which it can be observed that edges of barcode regions approximately emerge because of gray change from the background to barcode regions. Based on Figure 4.7B, all the contours and holes can be searched out (Figure 4.7C).

Before discussing the core of the algorithm, some terms have to be explained first. If one region is entirely inside another region, this region is another region’s child, and another region is this region’s parent. According to this definition, one region can have multiple children or/and multiple parents. In order to select the most possible barcode regions from these contours, three steps are needed. The first step is to eliminate contours with no or a small number of children by setting a threshold of children's number of each contour (Figure 4.7D). The primary reason is that barcode regions usually contain more children due to the multiple bars contained within. Then, considering that a barcode region is usually quadrilateral, if a polygon is used to approximate it with a certain accuracy, the polygon should not have many vertices, and this vertice number is limited by threshold2. After this step, only the contours that have relatively regular shapes or very small areas are left as shown in Figure 4.7E. At last, the final result (Figure 4.7F) is given by eliminating invalid barcode regions with the area less than threshold3 that makes it difficult to read by a decoding program.

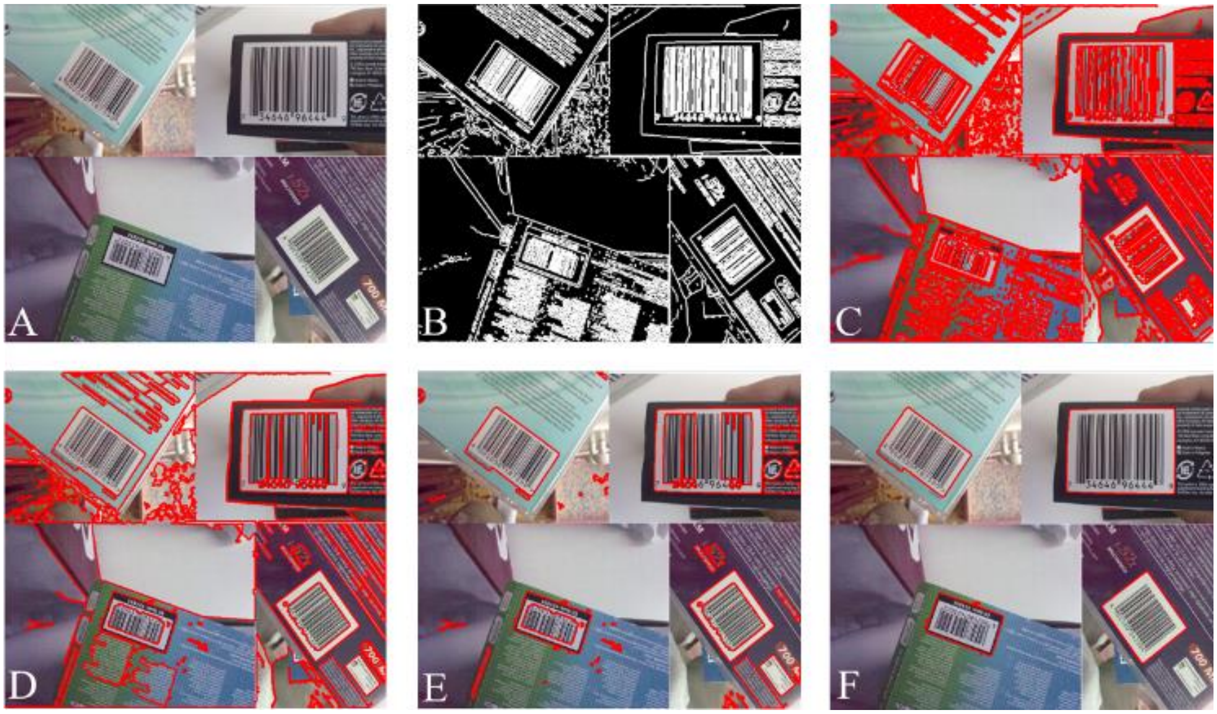


Figure 4.7 Visualization of barcode region detection

For the specific example above, this algorithm works well to recognize all the barcode regions, but several points still need to be emphasized. One observation is that edge detection has to be applied here because backgrounds of different barcodes in the given image result from the combination of 4 separate images. However, for real warehouse environments (e.g., Figure 4.11) where the background of barcode areas is relatively uniform, this operation can simply be replaced by a binarization operation that uses less time. Another observation is that the processing above does not use any special features of barcodes and just identifies the regions meeting the three restrictions. As a result, the final regions may also include some redundant ones besides the real barcode regions (Figure 4.11).

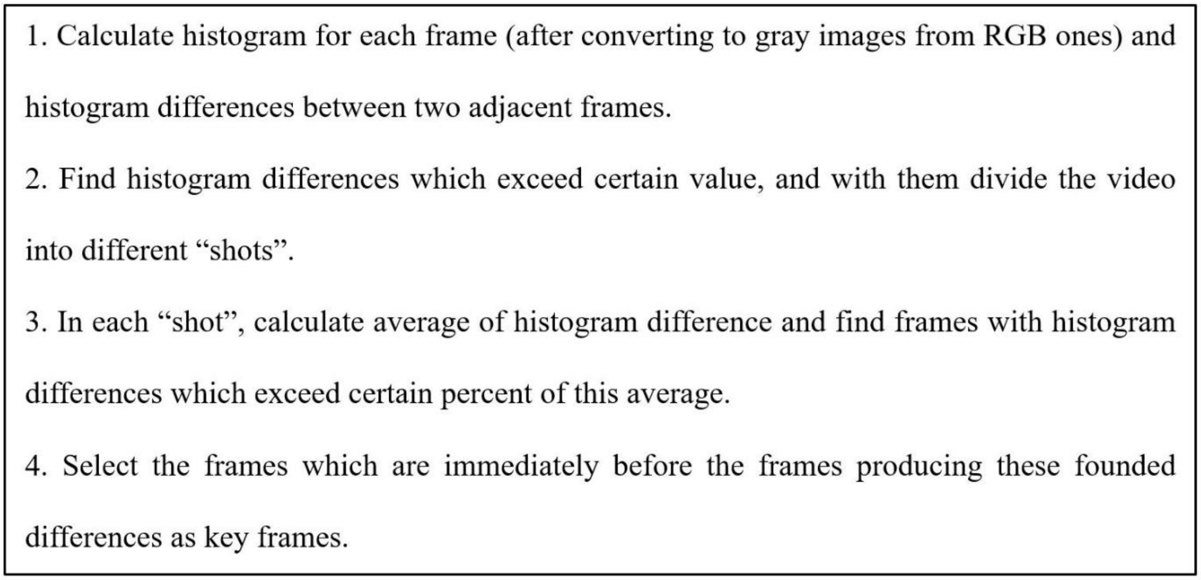
4.3.3 Fast Extraction

The two parts introduced above together are sufficient to find multiple barcode areas in an image, adjust their direction, and read them one by one. The problem left here is that when they are utilized to process large volumes of video data containing thousands of frames, it will take a long time to extract all the barcodes since each frame has to be processed separately. However, it is clear that not all frames can provide new barcode information, especially when two or more sequential frames generally have a big overlap that contains redundant information. The motivation of our fast extraction algorithm is to use fewer frames (key frames) to identify and extract all barcodes of interest in a shorter time.

Although from a human perspective, a warehouse is a simple repetitive environment that is well-organized for management operations, its repetitive pattern of shelves, boxes, labels, and barcodes renders it difficult for algorithms to measure the difference between different or subsequent video frames. Therefore, instead of measuring overlap with commonly used features such as SIFT [216] and MOPs [217], histogram difference can utilize color information from a whole image and is used in our approach for measuring frame change. The procedure followed by our algorithm and the corresponding results on a video (the same one from Section 4.4 but only some front frames are used to explain frame selection results) are shown in Figure 4.8 and Figure 4.9, respectively. This algorithm works effectively mainly depending on two strategies.

First considering different levels of histogram difference between sequential frames due to scene change or/and camera moving speed change, the concept of the virtual shot is introduced here to reflect this kind of frame change (even though the video is a one-take shot). Since generally the frames with larger histogram differences have less chance of being readable (due to more likelihood of being blurred), these different shots are considered to be divided by the frames with larger histogram differences. The threshold set here is usually determined by the camera moving patterns that can be easily measured using some consecutive frames.

Another strategy used in this approach is that the final frames selected are not exactly the same as those found in step 3 (Figure 4.8) but rather are the frames that are immediately before them. The direct effect of this is that frames with smaller, medium or larger histogram differences all have a likelihood of being selected albeit with different probabilities, which makes the final frames manifest enough frame change while keeping a certain number of clearer images to ensure recognition rate (Figure 4.9).



1. Calculate histogram for each frame (after converting to gray images from RGB ones) and histogram differences between two adjacent frames.

2. Find histogram differences which exceed certain value, and with them divide the video into different “shots”.

3. In each “shot”, calculate average of histogram difference and find frames with histogram differences which exceed certain percent of this average.

4. Select the frames which are immediately before the frames producing these founded differences as key frames.

Figure 4.8 Algorithm for key frame selection based on histogram difference

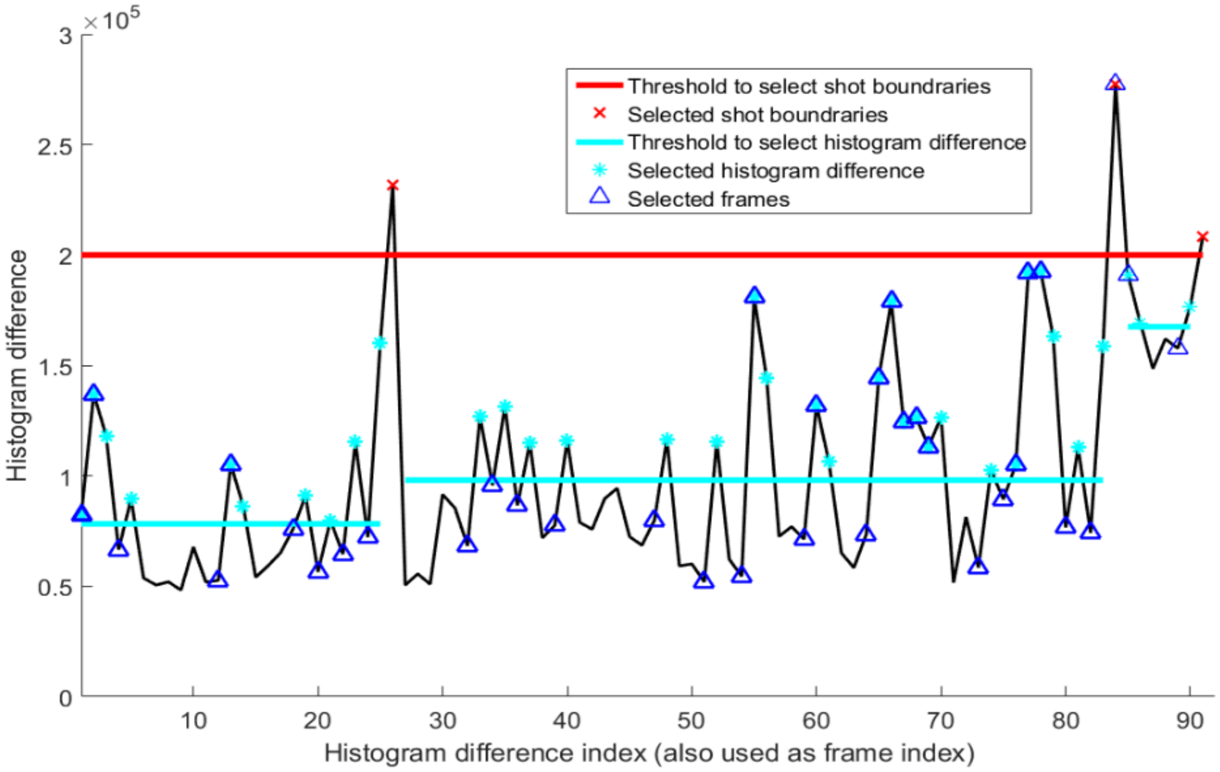


Figure 4.9 Visualization of key frame selection

4.4 Experimental Results and Analysis

The previous sections have discussed all proposed techniques - barcode direction estimation, barcode region detection, and fast extraction – that together work effectively to extract barcodes from an arbitrary video scan. In this section, we test our algorithm using video scan data obtained from an active logistics warehouse supporting an automobile manufacturing supply chain located in the metro Detroit area. It should be noted that for testing the algorithm's effectiveness and robustness, the video is taken by a handheld camera under normal illumination conditions (under which the warehouse is normally operated), and intentionally includes continuous left and right shaking of the camera, various shot angles, rapid change of camera moving speed as well as some re-visiting frames. All of these intentional artifacts help simulate the difficulties for barcode extraction that are likely to be bigger than that can be expected when a drone-mounted camera conducts automatic scans across the entire expanse of a warehouse (current commercial camera-equipped drones, such as DJI Phantom 4 [222], can easily record video with much better frame stability).

The entire technical approach including all the components (the three techniques proposed above as well as a chosen barcode decoding algorithm, ClearImage) is shown in Figure 4.10, and an example of processing a key frame is given in Figure 4.11. Figure 4.11 shows how a specific key frame is processed in practice. The blue arrow represents the process of barcode region detection. The red arrow represents the process of barcode direction adjustment. Failed/Successful represents whether a barcode is read out from the current state.

In the complete solution, with video frame input, key frame selection first helps select fewer number of frames necessary to process (the main parameter is key frame selection threshold). Then in each selected frame, potential barcode regions are picked out by the barcode region detection algorithm (the main parameter is the threshold of binarization), as shown in Figure 4.11 (regions A, B, C, and D). In the following decoding procedure, ClearImage is selected for use due to its partial ability to read multiple rotated barcodes from an image.

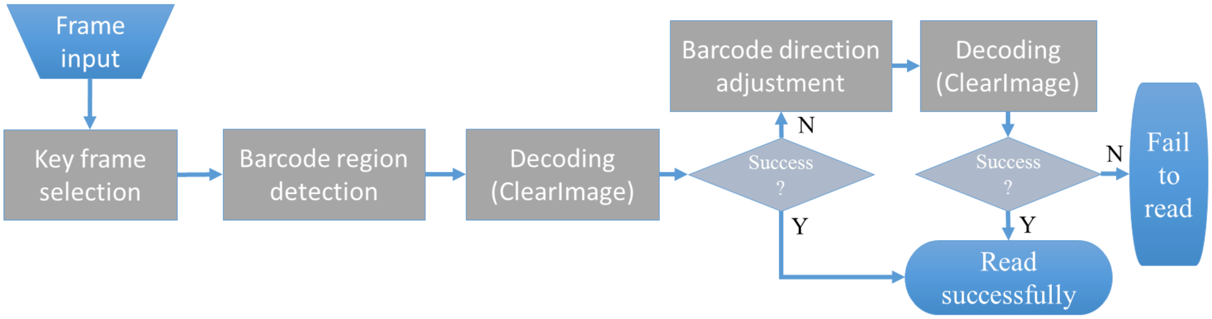


Figure 4.10 Complete algorithm for reading barcode from video scan data

Generally, most of the barcodes are already in the relatively ideal angular state for decoding and considering that an algorithm such as ClearImage can process some rotated barcodes, in order to save time by not rotating unnecessary barcodes, it is first directly used to decode barcodes from the identified regions (Figure 4.11, region B is successfully read). If this step fails (Figure 4.11, region A, C, and D), the direction adjustment algorithm is then applied to rotate the failed region to let ClearImage attempt the decode step again to determine if it can be successfully recognized (Figure 4.9, region A is finally successfully read, but regions C and D still fail).

In practice application, it is usually not necessary to use all the components indicated by the greyed boxes in Figure 4.10. With the benefits of modular design, it is easy to just plug in different combinations of the components and they would be ready to work with other parts of the solution. The users would be expected to choose the best specific solution by testing different combinations of these components and different threshold parameters using some sample frames of the video scan data to be processed.

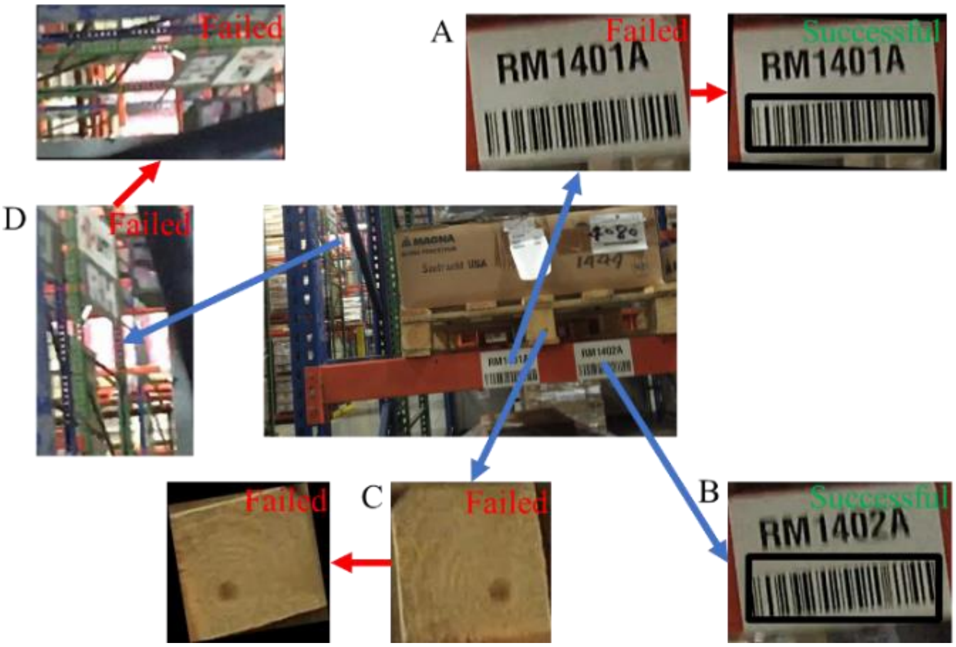
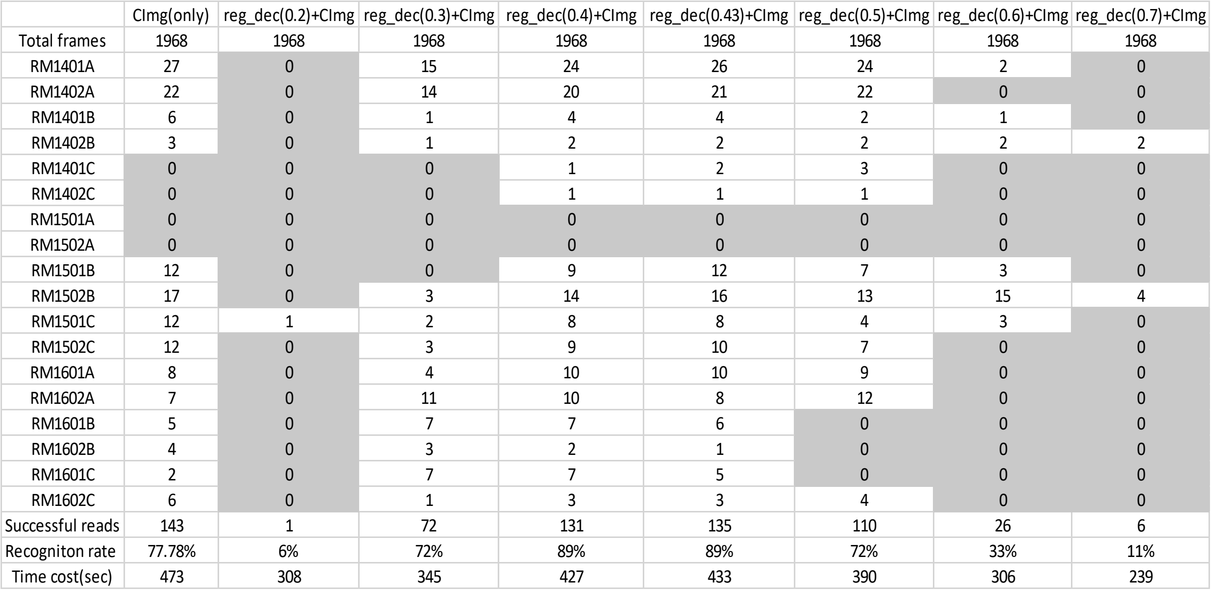


Figure 4.11 Illustration of processing a specific key frame

In this experiment, we tested the performance of different combinations of the proposed techniques and analyzed how well each technique discussed in Section 4.3 performs to contribute to better performance of a complete solution.

Different from the previous order used to describe the various components of the proposed algorithms, in this section, it is more convenient to test the barcode region detection algorithm first. For this purpose, only barcode region detection and barcode decoding algorithms are used (without key frame selection and barcode direction adjustment). This implies that in this special case, all the input frames will be used for barcode region detection and ClearImage only reads each detected region once without direction adjustment for a second attempt.

The given video totally has 18 different location-identifying barcodes recorded in 1968 frames. Such barcodes are usually attached to storage racks to identify the location of goods stored in each cell of the rack (Figure 4.11). The corresponding experimental result is shown in Figure 4.12, CImg represents ClearImage, reg_dec represents our barcode region detection algorithm, and the number in the parentheses behind is the threshold used for binarization. The number on the right of each barcode is how many times the barcode is successfully read from all frames. Successful reads are calculated by adding up all the successful reading numbers in the corresponding column. The recognition rate is the percentage of the barcodes that are read successfully at least once (the total number is 18), which is equivalent to calculating the percentage of storage cell positions that can be successfully located out of 18 different cell positions. Such position information is very important to automatically navigate a drone in a warehouse and provide location information for stored goods.



	CImg(only)	reg_dec(0.2)+CImg	reg_dec(0.3)+CImg	reg_dec(0.4)+CImg	reg_dec(0.43)+CImg	reg_dec(0.5)+CImg	reg_dec(0.6)+CImg	reg_dec(0.7)+CImg
Total frames	1968	1968	1968	1968	1968	1968	1968	1968
RM1401A	27	0	15	24	26	24	2	0
RM1402A	22	0	14	20	21	22	0	0
RM1401B	6	0	1	4	4	2	1	0
RM1402B	3	0	1	2	2	2	2	2
RM1401C	0	0	0	1	2	3	0	0
RM1402C	0	0	0	1	1	1	0	0
RM1501A	0	0	0	0	0	0	0	0
RM1502A	0	0	0	0	0	0	0	0
RM1501B	12	0	0	9	12	7	3	0
RM1502B	17	0	3	14	16	13	15	4
RM1501C	12	1	2	8	8	4	3	0
RM1502C	12	0	3	9	10	7	0	0
RM1601A	8	0	4	10	10	9	0	0
RM1602A	7	0	11	10	8	12	0	0
RM1601B	5	0	7	7	6	0	0	0
RM1602B	4	0	3	2	1	0	0	0
RM1601C	2	0	7	7	5	0	0	0
RM1602C	6	0	1	3	3	4	0	0
Successful reads	143	1	72	131	135	110	26	6
Recogniton rate	77.78%	6%	72%	89%	89%	72%	33%	11%
Time cost(sec)	473	308	345	427	433	390	306	239

Figure 4.12 Evaluation of barcode region detection

From Figure 4.12, it can be observed that in this illumination condition, region detection works best at the binarization threshold from 0.4 to 0.43, when it can help recognize two more barcodes than ClearImage alone and increase recognition rate from 77.78% to 89%. It also helps save time since ClearImage only needs to directly process useful areas of images instead of the whole images, which saves it about 40 seconds while processing this video. Besides, as the binarization threshold either decreases or increases, the recognition rate would always decrease even though it uses lesser time. The reason behind this is that for the specific illumination condition of the given video, the optimal binarization threshold is around the range from 0.4 to 0.43, which can give the richest counters. As the threshold goes up or down away from the optimal value, more and more counter details would be lost. Correspondingly, it takes more time to process richer counters and produces better recognition results, and vice versa.

Another observation is that in most frames barcode regions can be identified correctly, but some of them would likely be omitted when the barcode labels do not have approximately uniform intensity especially due to shadow from surrounding objects (like the square wood beam in Figure 4.11). However, ClearImage searches for all the valid barcodes in the whole image, such that using ClearImage alone recognized one barcode more times in the frames it appeared compared to other methods that did barcode region detection first (Figure 4.12).

In order to evaluate the performance of barcode direction adjustment, Figure 4.13 shows barcode recognition results when the binarization threshold is set to 0.43. The left side of Figure 4.13 shows recognition results of all 18 barcodes in the video scan including 1968 frames, with CImg(only), reg_dec(0.43)+CImg and reg_dec+CImg+rot. rot represents additional rotation as shown in Figure 4.10. All other abbreviations represent the same meaning as those in Figure 4.12. The right side shows several examples of barcodes whose directions have to be adjusted before they can be read. That is, they cannot be read out directly using ClearImage.

For most clear images, ClearImage can work well to recognize barcodes in different directions. However, it suffers from performance loss when reading rotated blurred barcodes that frequently exist in a video scan (Figure 4.13, right side). The results in Figure 4.12 are thus expected to be improved by adding an extra barcode direction adjustment step to rotate the region to the near-ideal state for another read (as shown in Figure 4.10, the only difference is that all the frames are used here). The left side of Figure 4.13 shows that the direction adjustment operation can enable 14 more successful reads and makes its total number higher than using ClearImage alone. However, it needs significantly more time and does not further help increase the recognition rate compared to reg_dec(0.43)+CImg. In theory, there is a trade-off between how thoroughly barcodes need to be read and how much time can be afforded.

	CImg(only)	reg_dec(0.43)+CImg	reg_dec(0.43)+Cimg+rot
Total frames	1968	1968	1968
RM1401A	27	26	29
RM1402A	22	21	21
RM1401B	6	4	4
RM1402B	3	2	2
RM1401C	0	2	2
RM1402C	0	1	1
RM1501A	0	0	0
RM1502A	0	0	0
RM1501B	12	12	12
RM1502B	17	16	17
RM1501C	12	8	8
RM1502C	12	10	12
RM1601A	8	10	13
RM1602A	7	8	9
RM1601B	5	6	7
RM1602B	4	1	1
RM1601C	2	5	7
RM1602C	6	3	4
Successful reads	143	135	149
Recogniton rate	78%	89%	89%
Total time(sec)	473	433	666

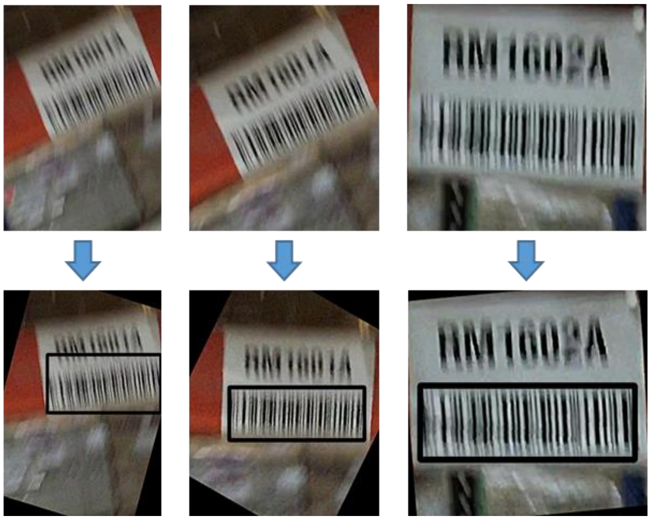
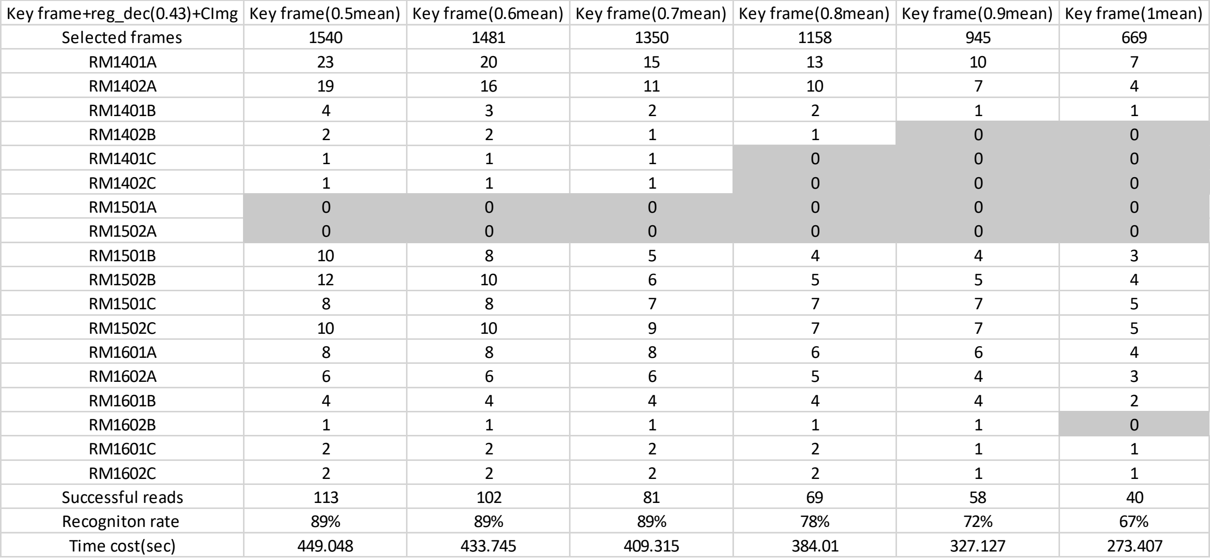


Figure 4.13 Evaluation of barcode direction adjustment

Finally, the key frame selection component is evaluated with different parameter settings. Figure 4.14 shows recognition results of all 18 barcodes in the video scan including1968 frames, with key frame selection+reg_dec(0.43)+CImg. Key frame represents key frame selection, and the number in the parentheses is the threshold to select histogram difference in procedure 3 (Figure 4.8). All other abbreviations represent the same meaning as those in Figure 4.12.

As shown in Figure 4.14, as the selection threshold parameter increases, the number of frames selected and time cost both keep decreasing. Initially, for parameter 0.5mean, even though fewer frames (1540 out of 1968) are used for further processing, the recognition rate is maintained but the time cost is even higher (449 s > 433 s) than the case without using key frame selection (Figure 4.12), since in this case time spent on selecting frames is more than the time saving it provides.

Subsequently, when the parameter increases to 0.6mean, the algorithm can obtain almost the same performance in recognition rate and time cost as the case when key frame selection is not used. As the parameter further goes up to 0.7mean, much fewer frames (1350<1968) and time cost (409 s < 433 s) are achieved while still maintaining the original recognition rate (89%). With this video, the recognition rate starts to decrease as the parameter increases to 0.8mean, which means that accuracy has to be sacrificed if more time is desired to be saved. This is however unique to this specific video.



Key frame+reg_dec(0.43)+CImg	Key frame(0.5mean)	Key frame(0.6mean)	Key frame(0.7mean)	Key frame(0.8mean)	Key frame(0.9mean)	Key frame(1mean)
Selected frames	1540	1481	1350	1158	945	669
RM1401A	23	20	15	13	10	7
RM1402A	19	16	11	10	7	4
RM1401B	4	3	2	2	1	1
RM1402B	2	2	1	1	0	0
RM1401C	1	1	1	0	0	0
RM1402C	1	1	1	0	0	0
RM1501A	0	0	0	0	0	0
RM1502A	0	0	0	0	0	0
RM1501B	10	8	5	4	4	3
RM1502B	12	10	6	5	5	4
RM1501C	8	8	7	7	7	5
RM1502C	10	10	9	7	7	5
RM1601A	8	8	8	6	6	4
RM1602A	6	6	6	5	4	3
RM1601B	4	4	4	4	4	2
RM1602B	1	1	1	1	1	0
RM1601C	2	2	2	2	1	1
RM1602C	2	2	2	2	1	1
Successful reads	113	102	81	69	58	40
Recogniton rate	89%	89%	89%	78%	72%	67%
Time cost(sec)	449.048	433.745	409.315	384.01	327.127	273.407

Figure 4.14 Evaluation of key frame selection

In fact, compared with the case of not using key frame selection, the new four barcodes that cannot be read after the parameter goes up to 1mean, RM1402B, RM1401C, RM1402C and RM1602B only appear a few times in the video and have been poorly recognized (successful reads are 2, 2, 1 and 1 in Figure 4.12 even if all the frames are used. This observation suggests that these barcodes are very sensitive to key frame selection. In a real application, performance can be further improved if video scan data collection is carefully controlled to ensure that each barcode is captured a sufficient number of times in the video frames.

In the experiment above, the optimized solution, keyframe(0.7mean)+barcode region detection(0.43)+ClearImage, can process video data including 18 barcodes in about 400 seconds, with the efficiency of about 22 seconds for each barcode, which is still relatively lower than manual scan. However, this comparison is based on the situation of scanning barcodes at a lower position of storage racks within human reach. For those barcodes at higher places, this reading efficiency would be very competitive compared to manual scans, not to mention other benefits of automation, energy efficiency, and worker safety. In addition, the efficiency of the optimized solution can be further greatly improved by breaking down an original scan video into shorter pieces and processing the shorter videos in parallel. From this perspective, the method is very promising for deployment in practice, even if we have not yet integrated a drone platform and performed a scan test for a whole storage rack or a whole warehouse in the paper.

4.5 Conclusions and Future Work

Even though many algorithms have been developed to extract barcode information from images (as those listed in Section 4.2), they have primarily been tested only on non-public or public image datasets that were well prepared (with the barcodes being in the center area and taking up a large portion of each image). These tests do not adequately reflect their effectiveness or robustness for video data collected under more challenging conditions with drones. In addition, none of them have any intentional design features to reduce redundant information in a video to improve efficiency.

In contrast, in an effort to enable drone-assisted inventory management in warehousing applications, we proposed three algorithms to correspondingly address the three key issues involved in the automatic extraction of 1D barcodes from arbitrary video scan data. In barcode direction adjustment, Harris corner detector and Hough transform work together to enable a fast and robust estimation of the direction of one single barcode. In addition, based on connectivity and geometry properties, barcode region detection helps to find all the potential barcode regions in one frame. Finally, to deal with a large number of frames in a video, a fast extraction algorithm using histogram difference to select key frames is discussed to exploit effective information efficiently.

Experiments conducted using video footage collected in an active warehouse show that the proposed algorithm components work effectively to read out and extract the majority of the location (i.e., cell) identifying barcodes robustly, given that the video was intentionally shot in challenging conditions. Another significance of this work is that each of the three techniques discussed above does not use specific information from other steps, which makes it easy to combine with other algorithms or computational sequences.

These characteristics increase the prospects of their wide application, even though some technical challenges still remain for future work before their practical feasibility. The main limitation is that some thresholds, such as the threshold in binarization and selecting histogram difference, have to be chosen by analyzing a small part of a complete video first and needs human assistance. This step can benefit from automatically comparing the performance of different parameter settings and choosing the best combination of the threshold parameters. In order to further eliminate the step of choosing the binarization threshold, we plan to use deep learning methods to recognize barcode regions automatically, in which the labor-intensive task of preparing labeled data can be significantly alleviated by using the processing results of our current solution.

Furthermore, the selection of the keyframe selection threshold can be conducted more effectively by integrating the pose estimation of the camera when it is available. Another limitation is that, besides location (i.e., cell) identifying barcodes, various other barcodes (e.g., manufacturer’s barcode, shipper’s barcode, recipient’s barcode) present on stored inventory products must also be simultaneously extracted and sorted for overall warehouse management and inventory control. Our current algorithm has no difficulty in reading such barcodes if their size in the video is large enough to be readable. Since such barcode labels are usually significantly smaller compared to the location-identifying barcodes, to guarantee their size, a drone has to go closer when capturing them and the drone's trajectory has to be carefully designed.

The proposed method is scalable to video scans collected by any manual or automated means. Even though the overall methodology is proposed around video scans collected using drone-mounted cameras, the algorithms themselves work effectively with other sources of video data such as hard hat cameras, or forklift mounted cameras that are also easy to deploy in warehouse environments. The research presented in this paper is complementary to the authors' ongoing work on drone localization and control in GPS-denied environments. Ongoing work is also focused on integrating the presented research results with warehouse inventory management systems.

Chapter 5

Distributed Coupling Analysis for Modeling and Understanding Built-Environment Processes: Reviews, Limitations, and Recommendations

5.1 Introduction

As in many other fields, analysis models used in studying built-environment processes have primarily evolved along with separate disciplines. For example, in earthquake engineering, several models have been developed to analyze various effects of an earthquake on civil infrastructure [66,67]. Another example in the fire propagation area is NIST's Fire Dynamics Simulator (FDS) and Smokeview [223]. Similarly, several models exist such as wind [9,20], tsunami [224], flood [225], power system [68], transportation [69], human response under disasters [21,70,71], and to a lesser extent, evacuation plans [72], emergency response training [73], and post-disaster recovery [74].

However, extreme complex processes, such as earthquakes, tornadoes, floods, and hurricanes, often induce complicated interdependencies between the built environment (e.g., buildings and bridges), critical infrastructure systems (e.g., lifelines and telecommunication), social and non-physical systems (e.g., politics and economics). As such, the analysis for studying these complex processes, more broadly, is a highly multi-disciplinary research topic. Many US government documents [226-229] and researchers [230,231] have called for the development of comprehensive frameworks that can integrate the efforts from different sub-fields and enhance interdisciplinary collaborations between researchers from different fields. In order to deal with this lack of compatibility, one promising and practical strategy is to modularize each discipline-specific computational model and then integrate them for coupling analysis with standards or standard-based platforms such as Distributed Interactive Simulation (DIS) [76], High Level Architecture (HLA) [77], Test and Training Enabling Architecture (TENA) [78] and Distributed Data Services (DDS) [79], or data passing tools such as Robot Operation System (ROS) [232] and Lightweight Communication and Marshalling (LCM) [233]. An integrated analysis achieved with such an approach is referred to as a distributed coupling analysis in this paper.

The current state of affairs in this field is that since each domain has been evolving separately, most of the existing integrated analyses are developed upon and limited to domain-specific development environments and lack the benefits of interoperability, reusability, and scalability provided by the generic distributed analysis platforms listed above. For example, in earthquake engineering, Integrated Earthquake Simulator (IES) [234,235] was originally developed to seamlessly integrate analysis models and simultaneously analyze almost all processes involved in earthquake events in Japan. However, even for a similar analysis, a new version of IES had to be developed separately for the Istanbul, Turkey earthquake due to differences in numerical analysis methods and available urban information [236]. Moreover, IES is sequential and thus inconvenient to be integrated with other models with different analysis resolutions such as dynamic debris and transportation systems for coupling analysis. Similarly, Miles and Chang studied the interactions that occur between various entities during emergent events [237,238]. However, the proposed approaches do not support model coupling, so they cannot analyze multiple events or even the same event that happens sequentially. In addition, without using distributed analysis, these analysis models need to run on a single host device with limited processing power, which usually limits the scale of the problem that can be analyzed.

Some researchers have realized the necessities and benefits of standards, platforms, or tools that can be used for distributed coupling analysis and started to use them in their own fields even though such efforts have been limited in scope and application. For example, Mandiak et al. developed a disaster monitoring interface and integrated it into an HLA-based earthquake model for post-disaster data fusion [239]. Fiedrich proposed a distributed analysis system based on HLA that focused on resource management issues during emergent events [74], e.g. allocation of scarce resources. To improve people's emergency response, Liu et al. demonstrated an emergency training analysis achieved by HLA [73]. Nan and Eusgeld developed an HLA-compliant analysis testbed and demonstrated that HLA is a viable option to analyze and capture interdependencies among different analysis models [240]. More recently, Lin et al. proposed to analyze interdependent effects with LCM and implemented an example application in wind engineering [9,20].

Due to the limitations in available tools for distributed analysis, the nontrivial gaps between data passing and domain knowledge, as well as the difficulty of handling multiple disciplines, most existing integrated analyses, as outlined above, have focused on the interactions that occur between two or, at most, three related process models. In practical processes, there are usually more factors that interact with each other and this coupling effect further impacts the final analysis results. In order to facilitate the development of compatible domain analysis models and the integrated analysis incorporating deep interdependencies between multiple analysis models, this paper surveys the main available solutions for interdependent study and complex process analyses in the built environment. These tools include standards and standard-based platforms (DIS, HLA, TENA, and DDS) and standalone data passing tools (ROS and LCM).

These solutions can also benefit various studies in civil engineering [241-245] by improving the scale and resolution of the analyses. The strengths and weaknesses of each representative distributed analysis solution are identified to guide researchers or users to choose the appropriate tools for their specific applications while being aware of the limitations. After the systematic review of the distributed analysis solutions, the key limitations to the existing solutions are summarized to highlight the specific needs for studying complex processes in the built environment. Finally, based on a synthesis of the gathered information, two platform design recommendations are provided, namely message exchange wrapper and hybrid communication, to help further improve distributed analysis capabilities in existing solutions and provide some guidance for the design of an improved distributed analysis platform.

5.2 Existing Distributed Analysis Solutions

Distributed computing emerged about forty years ago when the US Department of Defense (DoD) started developing communication protocols to enable interactive models involving various types of weapon systems. Among the platforms developed were Distributed Interactive Simulation (DIS) [76], High Level Architecture (HLA) [77], and Test and Training Enabling Architecture (TENA) [78]. Besides military training and simulation, they have also been utilized in marine analyses [246], space projects [247], infrastructure system analyses [248], and virtual tests [249].

Independently driven by the challenges of conducting real-time sensing, information fusion, and control in robots, researchers in robotics engineering developed low-latency data passing solutions. For example, ROS [232] and LCM [233] have been developed and widely used in real-time robotics applications. Even though they were not originally developed as distributed analysis tools, due to their ease of use and high efficiency, researchers have started exploring their applications in distributed analyses for modeling coupling interactions between building energy consumption and human comfort [8] and interdependent effects in wind-building interaction [9,20].

In recent years, due to the rising interest in the extension of internet connectivity, many solutions have been proposed to address the emerging need for Internet-of-Things (IoT) applications. Among such work, IoTivity [250], which uses a constrained application protocol (CoAP) as its software protocol, is mainly focused on device-to-device connection. Distributed Data Services (DDS) [79] is a more general data communication protocol and standard developed by the Object Management Group (OMG), which is suitable for all kinds of connections in IoT applications. Even though DDS was developed for real-time operations, it provides competitive features (such as API Standard, Data Modeling Standard, Quality of Service, and Time Management), as compared to HLA, and is also suitable for distributed analysis.

The remainder of this section reviews the two categories of solutions for distributed analyses: standards and standards-based solutions and standalone tools.

5.2.1 Standards and Standard-Based Solutions

5.2.1.1 Distributed Interactive Simulation (DIS)

The early efforts of the US Defense Community to address the need for networked multi-user simulation led to the SIMNET (Simulation Networking) project [251]. For about a decade, SIMNET formed the technological foundation for many of its descendants and was the origin of a sequence of IEEE standards. One of SIMNET’s derivatives, the DIS protocol, was published as an industry standard from 1993 to 1998 by IEEE [252]. The standard was considered dominant until a new standard (IEEE std 1278.1-2012 [76]) was released. It was planned to be used in a future version of evacuation analysis in fire emergencies [21]. For legacy reasons, DIS is still used today in some modern studies.

The DIS protocol is designed to be a message passing standard (not an existing software or package) that specifies message types and the procedures to transmit the messages across a network of different analysis models. If it is followed correctly, compliant analysis models are capable of sending and receiving messages to and from any other compliant model, even if the local DIS implementations that run on different hosts are diverse. More specifically, DIS adopts a communication pattern for message exchange with point-to-point communication via User Datagram Protocol (UDP) as shown in Figure 5.1. The message format is well specified and referred to as protocol data unit (PDU), which consists of an entity ID, entity type, and any expected values an analysis model requires to function, represented in binary format. The standard defines exactly what variables can be present. Values like “position,” “orientation” and “collision” all take a certain number of bits and have pre-defined limits to the range of values they can contain.

It is presumed that each analysis model is capable of encoding and translating these values to binary format, knowing in advance the exact location and number of bits in which each value exists (as defined by the standard). Therefore, a set of PDUs used in different fields for different purposes are predefined in the standard and only these PDUs are available to model developers. Such an approach is very inflexible. If a custom type of PDU is required, it has to be included in the standard first and only then can it be used in analysis models. For example, in order to analyze wind’s effect on multiple buildings, a scenario model needs to be set up first to send scenario information to other models such as wind generator model, structure analysis model, and damage model. The scenario information needs to include building location and geometry and even their material types and thus need a more complex data structure than what DIS provides in its PDUs. Therefore, it is very inconvenient to make such an integrated analysis with DIS.

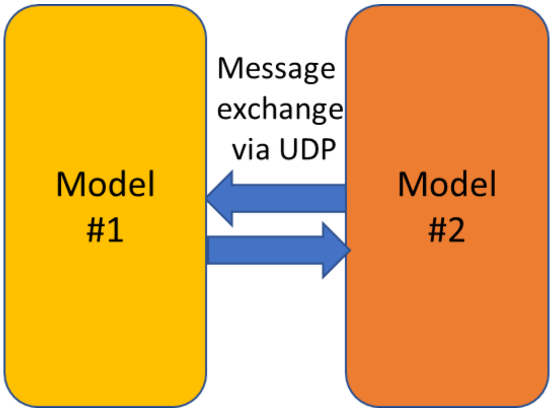


Figure 5.1 Point-to-point message exchange via UDP in DIS

In point-to-point communication (Figure 5.1), there is no middleware or center server maintaining message exchange. Instead, each message sender would manually connect to its receivers using their network Internet Protocol (IP) addresses. This makes it only suitable for its original focus, i.e. individual weapon modeling for military training and real-time wargaming but not scale well for the aggregate level modeling of a battlefield. Another problem is that although the standard describes in great detail the format of data being sent over a network, it does not specify how exactly network communication should be implemented and is open to any implementation (typically hidden to an end user). This further leads to the following two disadvantages: 1) it is up to users to create their own communication tools by following the standard, and 2) users must be capable of creating the tool themselves or must be able to obtain a premade solution (open-source or commercial) such as Open DIS [253] and VR-Link [254].

Knowledge of the data format in advance is the most straightforward manner to maintain consistency across different analysis models. However, DIS puts full responsibility on the user to correctly implement its standard. Inflexibility in the data format and consequences for peer-to-peer network connections make it scale poorly for different use cases and challenging to implement in the case scenarios involving multiple simultaneous analysis models that are quite common in an analysis in the built environment. DIS also does not encompass other important features, such as time management and network management that would be desirable when multiple analysis models are involved in an integrated analysis. Newer standards attempted to address these drawbacks. Among them are "Common Training Instrumentation Architecture" (CTIA), "Aggregate Level Simulation Protocol" (ALSP), "High Level Architecture" (HLA) and "Test and Training Enabling Architecture" (TENA). Among these, HLA and TENA are the most widely known.

5.2.1.2 High Level Architecture (HLA)

HLA was developed by the US Department of Defense (DoD) and the Defense Modeling and Simulation Office (DMSO) in 1995 [255] based on experience with DIS and the desire to develop a high-level architecture that would facilitate interoperability and reusability of components in an integrated analysis. It became a DoD standard in 1998 (The U.S. DoD HLA 1.3 specification) [256] and an IEEE standard in 2000 (IEEE 1516-2000 standard) [257], and then evolved again to its latest version in 2010 (IEEE 1516-2010 standard) [77], and continues to be an active standard as of 2019. HLA was once widely used in distributed analyses, where it was utilized to model interdependencies between critical. infrastructure systems [74,258,259] and interdependencies involved in disaster responses [73,260].

HLA has some advantages over DIS. First, HLA-compliant software uses an application programming interface (API), which in turn can be used by an analysis model, called a federate application in HLA. This design facilitates connections between federates. The API includes functionality to control time management and to sync data exchange between different analysis models. Second, unlike DIS where the data structure has to be predefined in the standard, by invoking the HLA Object Model Template (OMT), a user is allowed to model data as an object instance or an interaction (also called HLA objects) which includes the data (attributes and parameters, separately) to be exchanged among the federates in a federation execution at design time. This is clearly more flexible than the DIS alternative for a complex analysis where an interaction instance can be conveniently used to model a fire event, an evacuation event, or a recovery event, and its effect can be reflected in some object instances used to model buildings or lifelines. Third, in terms of how data is exchanged, instead of using the point-to-point communication in DIS (Figure 5.1), HLA routes data as HLA objects via a middleware (called runtime infrastructure [RTI]) using a Publish/Subscribe (P/S) pattern (Figure 5.2). In this way, the sender and receiver federates just need to declare what data they need and what data they provide, without the requirement of knowing about other federates. This feature further improves the reuse of each federate by decreasing the coupling among the implementation of different federates in a federation, and makes it scale better for systems with a large number of analysis models. In other words, the federates each connects to a single point, rather than to each other.

The process to use HLA is as follows: 1) Prepare or obtain an HLA-compliant solution, consisting of an RTI and a local-client library or API; 2) Prepare the expected data definition, in accordance with the HLA specification and according to what each federate in the system requires; 3) Compile the HLA software with the data format, such that the HLA solution now can expect the data in low-level bytecode; 4) Run the RTI on a local or remote machine; 5) Create or modify a federate to use the HLA solution’s client API to be able to connect to the RTI; 6) Run the integrated analysis to successfully connect to the RTI, and to send and receive data.

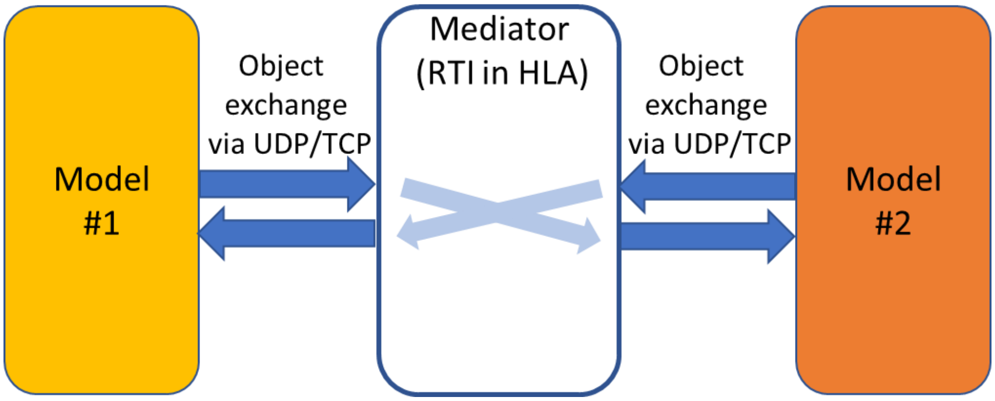


Figure 5.2 Mediator-based object exchange via UDP/TCP in HLA

Although HLA is more general than DIS, it still suffers from multiple flaws. It acts as an architecture framework for distributed analysis [77], not a software or an implementation. Therefore, HLA software must be able to "connect" to the RTI. However, it does not specify how the connection works, leaving the implementation up to the creator of the HLA compliant software. Similarly, time-management is described as a function that must exist, but how it functions can be unique in many implementations. Indeed, its very existence is all that is needed to be compliant. For example, if the wind-building analysis mentioned in Section 5.2.1.1 is created in HLA with different RTIs, the analysis's efficiency can be quite different depending on detailed RTI implementations. In practice, users usually have to try different RTIs to get satisfactory efficiency performance. Moreover, having to compile a data format allows efficiency to be maintained as to the number of bytes in each message but requiring compilation on a user's local machine every time that the data format and content changes is onerous.

Commercial HLA software packages available for use are CERTI [261], Portico [262], MAK [263], and Pitch [264]. While there are multiple open-source solutions, as of 2019, many of them have been discontinued or are unobtainable, or they are not 100% compliant with HLA standards. It is not trivial to program HLA software from scratch, as there are six separate management systems (federation, declaration, object, ownership, time, data distribution) [77] that have their own specification section, each lengthy, but lacking in the detail required for systematic implementation. The benefit of the standard is that a prepared analysis model can be compatible with any compliant HLA software. However, generally, only one vendor's HLA software can be used in a federation to allow compatibility with the RTI and local federate's API. In order to help ensure compliance and encourage adoption, the US DoD offered a public service to check if a new implementation met the HLA standard, but this service was later abandoned when the original website shifted domains [265].

HLA is still in use, but interest in it has decreased since its inception. The standard's ambiguity and a lack of easily available implementations have made it difficult for newcomers to utilize HLA in practice.

5.2.1.3 Test and Training Enabling Architecture (TENA)

TENA was introduced by the U.S. Department of Defense (DoD). Designed after HLA, TENA's development traces as far back as 1998 [266] and continues to be maintained as of 2019. Similar to HLA, TENA allows for the development of individual analysis models interoperable with each other for distributed systems, saving time and money in the development process. As a tool, its functionality is revised based on early user feedback, but its core intentions drive its development.

The architecture of TENA consists of TENA-compliant applications, TENA Middleware, and TENA Utilities, including a gateway accessible by non-TENA systems. The middleware acts as a communication channel, where data must be formatted according to a TENA Object Model. The Object Model is capable of evolving, which is different from HLA.

Different from HLA and other standards, TENA is not intended to act as a professional standard document [78], but as a standard tool. Monitored closely by its development group, access to TENA and its documentation requires applying for a free account through an online portal. While this process is open to all applicants, the existence of a screening process that asks for contact information, project intentions and grant usage makes it difficult for researchers to test TENA's functionality to confirm it meets their requirements. Therefore, even though TENA was made explicitly to overcome the limitations of HLA, its inaccessibility, together with how its functionality and design are still in development, make TENA a difficult choice for practical use cases in the near future.

5.2.1.4 Data Distribution Service (DDS)

The DDS is a standard for data communication between distributed machines and software for real-time systems [79]. Unlike the previous solutions in this section that are oriented for distributed modeling and created by government entities, DDS was originally designed for real-time distributed operational systems and developed by a professional non-profit collective called Object Management Group since 2004 [267]. DDS is not a single tool or software solution, so users must utilize accessible documentation on the standard to prepare their own, or must use existing solutions, including RTI Connext [268] or OpenDDS [269]. The intention is that all DDS solutions follow the specification carefully, such that each user can use any vendor's DDS solution and be interoperable with each other on the network. Professional demonstrations have been given to show this interoperability to be possible with DDS software from different vendors. It has been used for crisis management caused by natural hazards [270,271].

Best suited for Internet-of-Things (IoT) applications, DDS is flexible for use across a variety of domains jointly with HLA or as a replacement. In DDS, data is pre-defined as a message format in a struct-like file suffixed with .idl and compiled with the DDS software to make it recognizable when it is written to or retrieved from the DDS global data space with a DataWriter or a DataReader. Using a specially designed topic-based P/S mechanism to share data within a domain participant, DDS does not depend on any global knowledge and supports fully dynamic discovery and matching of different DataWriters and Data Readers, which is more flexible than HLA that still requires static declaration in FOM even though different publishers and subscribers can be matched dynamically. It also provides richer (22 versus 2 in HLA) quality of service (QoS) policies that help to control local and end-to-end properties of DDS entities. Conversely, since it was originally designed for real-time application in distributed operational systems, the main disadvantage of DDS is that it does not explicitly provide time management mechanisms for different types of time advancement controls as HLA does.

While DDS is not as feature-complete as HLA for distributed analysis, its simplified standard makes it more accessible for users. Not requiring a single access point (like HLA's RTI) makes it less prone to slow-down from the RTI's perspective when adding more analysis models, and the complexity of connections is handled internally without the user's concern. Compiling a data format ahead of time is still a limitation that TENA sought to overcome but doing so enables DDS to maintain optimal speed in data communication.

With each of these solutions, the need to study lengthy standards and rules together with the need to deal with the compatibility of legacy standards, often make them difficult to use. However, these solutions provide instructions that others can follow, putting the responsibility on individual users to ensure their own local analysis models are built correctly to be compliant without worrying about how other analysis models might function.

5.2.2 Standalone Tools

5.2.2.1 Robot Operation System (ROS)

Unlike the approaches listed above that are either designed for distributed analyses or distributed operational systems, ROS is a robotics middleware that provides services including hardware abstraction, low-level device control, package management, and message passing [232]. Such a design makes it possible for robotic engineers to quickly and conveniently build up a robot by taking advantage of many existing hardware drivers and implemented algorithms distributed as ROS packages [15,158,272].

While ROS is best suited for applications in robotics, its message-passing design based on a Publish/Subscribe communication can be applied to distributed analysis applications in the built environment with some benefits. More specifically, the three different patterns of data exchange supported by ROS all have their corresponding applications in distributed analyses. In most cases, the output of one node needs to take as input the runtime outputs of some other nodes, and in turn, its output can be used as part of the input for other nodes. Such input and output information generally needs to be exchanged continuously with a small time step and can be modeled as messages in ROS.

A message in ROS is a data structure that can be defined flexibly in a .msg file by following a syntax similar to C structs. Most messages have a header field that is filled with a timestamp by ROS and is used for time management. Once a message is compiled with ROS, by importing or including its bindings, a node can encode and retrieve information into and from the corresponding message automatically with ROS. Another type of data exchange is conducted in the Request/Response way where the data structures in a request and a response are formatted together as service in ROS and are defined in a .srv file by following a similar syntax to ROS messages. In this pattern, by agreeing upon the same srv, a client provides the required input for the request and requests a server to give a response based on the request. The returned response depends on the implementation on the server side.

For a distributed analysis, this is suitable for acquiring the global configuration (such as the scenario information in the wind-building example in Section 5.2.1.1) from a server or for commanding some other nodes to behave in a specific way. However, for the latter use, if the server takes a significant amount of time to perform the requested action or does not respond to the request, the client would not receive any feedback and thus know nothing about the status of the server. This lack of knowledge of the server statusr can be solved by using the actionlib pattern. This pattern specifies the formats of the goal (the result and the feedback message) in an action file in a similar way to ROS msg and ROS srv. In this way, after a client sends out an action request to a server, it can keep listening to the feedback from the server and make further decisions based on the feedback. This approach is beneficial for a distributed analysis whose nodes are modeling reality at different time scales (such as an earthquake node and a recovery node). Those nodes that run faster can request the others to catch up via an action request.

For its wide use in the robotics community, ROS is well documented, and it is easy to access help from different technological forums. Despite the above advantages, it also has some drawbacks. Since it was not specially designed for distributed analysis purposes, it lacks implementation of time management and quality of service (QoS) policies compared to other distributed analysis-oriented approaches. Besides, it does not provide a convenient way to set up connections among different nodes. Each node needs to explicitly specify the topics the node subscribes to and the way the node wants to receive the messages on these topics. Therefore, when an analysis model is developed as a ROS node, the code for message communication is usually interspersed with the code for the analysis model function. This lack of convenient communication interface makes it scale poorly as the number of nodes increases, limiting its suitability for large-scale analysis.

5.2.2.2 Lightweight Communication and Marshaling (LCM)

LCM is another data passing tool oriented for real-time robotics applications [233]. It has been applied to distributed analyses [9,20] recently owing to its beneficial features including low-latency, platform and language independence, and publish-subscribe data transmitting scheme. As a lightweight solution, it is mainly comprised of three functionalities, message type specification, message marshaling, message communication, and despite what its name suggests, some data analysis tools. In LCM, the data to be transmitted over a network need to be first structured as a message type, by following its specific type specification language whose syntax is very similar to C structs.

After the message type is well defined, the provided lcm-gen tool is invoked to generate its language-specific bindings that can be further included or imported in a custom analysis model to use the corresponding message. Such bindings can be generated to support multiple languages (C, C++, C#/.NET, Java, Lua, and Python) on different platforms (Linux, OS X, Windows, and any POSIX-1.2001 system), which is very convenient for developers with different preferences. In the actual communication, a message is marshaled by attaching to it a fingerprint derived from its channel name and message type and routed from its sender to its receivers with a Publish/Subscribe pattern. LCM uses multicast UDP based peer-to-peer communication, in which there is no mediator and each analysis model can be both a sender and a receiver.

In LCM, messages are routed to all the LCM subscribers that are in the same multicast group and each subscriber further selects the messages it is expecting based on the channels to which it has subscribed. As shown in Figure 5.3, for any analysis model #i in the multicast UDP group, its LCM subscribers receive all the messages published within the same group. After receiving these messages, its subscribers automatically select the messages published to the channels that they have subscribed to by dropping all the other messages, such that the analysis model #i can work with the messages it is interested in by just subscribing to the appropriate channels.

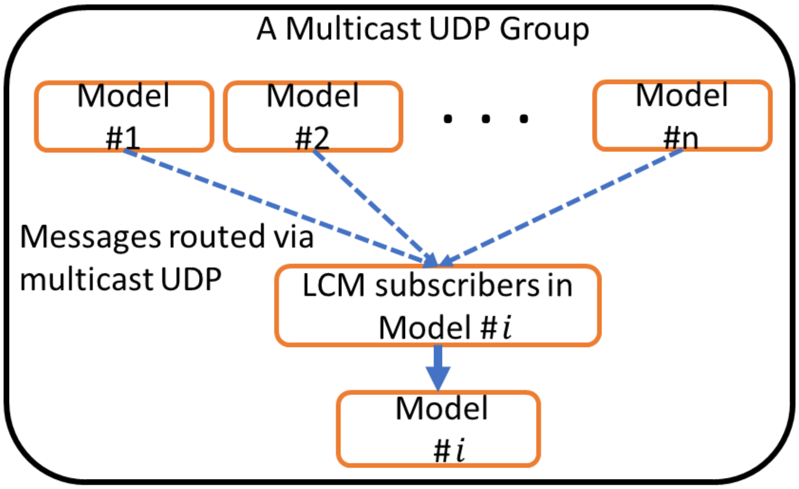

A Multicast UDP Group
Model #1
Model #2
. . .
Model #n
Messages routed via multicast UDP
LCM subscribers in Model #i
Model #i

Figure 5.3 Message exchange via multicast UDP in LCM

Besides, LCM also provides some useful tools (logging, replaying and inspecting traffic) to help with debugging during development as well as help inspect and analyze efficiency performance during testing.

As a pure data passing tool, LCM provides great flexibility for further development of different features by users. However, as a robotics tool, it inevitably lacks the specific features dedicated to distributed analysis, such as time management and QoS policies. Moreover, due to reasons similar to ROS, it does not scale well.

Compared to the approaches in the last section, approaches from the robotics community include ready-to-use libraries and provide well-documented instructions, which make them easier to use for skilled programmers. The main problem with these methods is the lack of a systemic way to deal with scalability issues. It is the users' responsibility to make sure that the connections among different analysis models and time management for each model are set up correctly by adding corresponding code to the models. This mixture of code for connection and model functionality makes it hard to manage the models when their numbers greatly increase and thus limits these approaches to small or medium-scale problems.

5.3 Limitations

5.3.1 Lack of Easy-to-use and Standard Solutions

Among the standards and standard-based solutions, both DIS and TENA have to use pre-defined sets of messages, which is not flexible for information exchange between different analysis models (as explained in the wind-building example in Section 5.2.1.1). Moreover, they can only build real-time distributed analyses that run in wall-clock time. This makes the analysis of a recovery process, a typical process after natural hazards involved in the built environment, very prolonged and inefficient.

Compared with DIS and TENA, HLA and DDS, are more suitable for modeling complex processes. As standards, they levy many requirements on the design of API, and some implementations have been designed by following such specifications. However, it is still difficult for a novice to rapidly build a functional distributed analysis and, for experienced users, non-trivial to achieve desired efficiency performance. On one side, with the aim of allowing an interoperability level of integration across areas in distributed analyses by defining common data types and specifying APIs, they have become formidably long standards that are quite hard to follow and adhere to. Therefore, it is common that some implementations just follow and support part of the API specifications and it is necessary for users to be aware of the deviations from the standards in addition to a basic understanding of HLA or DDS concepts.

While HLA and DDS include the detailed requirement for API, they do not specify precisely what algorithms need to be used and how the API function should be implemented, which leaves the flexibility to API implementers. This flexibility for the implementers leads to diverse API implementations with different vendor-specific features and advantages, and it is important for users to be able to choose the appropriate implementations to achieve their custom efficiency performance goals. In practice, achieving efficiency performance goals requires the users to know about different implementations and the differences between them since these differences are generally non-trivial, and experience from one implementation cannot be directly applied to another.

Unlike tools built on standards, ROS (and LCM) can be viewed of as a standalone tool providing much less, but necessary, APIs for data sharing, which is particularly well suited to users who need to quickly build up a small-scale application-specific analysis and distribute it over a network. While this approach provides a flexible and convenient way of constructing distributed analysis, the issue for this category of tools is that different analysis models have to agree on the structure of the shared message due to lack of standardization, even though it is not difficult to come up with simple specifications on the data structure for application-specific problems.

5.3.2 Lack of Scalability and Extensibility for Building Large-Scale Analyses

Generally, standards do not specify the scale of distributed analysis that an API needs to and should support, and in theory, users can try to connect as many analysis models as they want in one distributed analysis. However, in practice, the scalability of the standard-based methods is significantly impacted by the detailed API implementations, and the practical performance can vary greatly as the size of the distributed analysis changes. When the analysis scale is small, such as an analysis of interdependencies between wind and several buildings, peer-to-peer communication is preferred since mediator-based communication would need one extra message copy for each subscriber of a message and thus need more bandwidth and result in more latency. As the analysis scale increases to the middle scale, such as a city-scale analysis of wind-building interaction, mediator-based communication becomes preferable. The reason is that the overhead resulting from additional message copies becomes less critical compared to the total message routing time, and mediator-based communication also provides other benefits such as monitoring of individual analysis models and more flexible central time management.

However, when the scale increases to a large scale, such as the same analysis as above in a country scale, the performance bottleneck of the analysis is usually the power of the device where the mediator is hosed since the mediator has to route a great number of different types of messages and conduct corresponding time management for a large number of analysis models. Therefore, it generally needs additional algorithms to distribute the work of the mediator over multiple host devices, which increases the complexity of the distributed analysis. Since both the standard-based tools and standalone tools reviewed above use a fixed message delivery method, it is difficult for them always to obtain the best performance for different analysis scales.

For extensibility, DIS and TENA are seriously limited since they can only use fixed sets of messages. Other standard-based tools such as HLA and DDS support custom messages, which make them convenient for extending the information shared between different analysis models. For standalone tools, new information to be shared has to be defined as new messages or added to the old message definitions, and the created or modified message definitions have to be recompiled to make sure different analysis models can recognize them. This process almost always includes modification of the relevant analysis models to make sure they can send and receive the pre-compiled messages. This process is not convenient and sometimes even difficult for experienced users.

5.3.3 Inability to Rapidly Build and Integrate Application-Specific Analysis Models

The most important goals of standards and standards-based tools are to improve reusability and interoperability, and these make analysis models usable across different fields. The benefits are significant when users have easy access to many choices of models that have been developed by people from different fields for different purposes. However, in practice, these benefits are limited for two reasons. First, it is still challenging to integrate models developed by others without any knowledge of them, even if they are compatible with the same standard. Such knowledge includes model time resolution, model mode (time-driven, event-driven or hybrid) and time management option, which users may have to modify to make the models work correctly. Therefore, models' reusability and interoperability are mainly achieved in some relevant distributed analyses that are developed by the same group of people who developed the models.

The second reason is that the complexity of utilizing the distributed analysis tools to develop reusable and interoperable models limits the number of available compliant models. Skilled users of distributed analysis tools are good at achieving the reusability and interoperability of models when they are given functional models from different domains. However, it is usually difficult for them to develop analysis models from scratch without enough domain knowledge. Instead, it is the people with good domain knowledge that are more suitable to develop domain-specific models for specific applications. However, the complexity of standards-based distributed analysis solutions creates a non-trivial gap between domain knowledge and an analysis model compatible with the same distributed analysis solutions.

It is also difficult to rapidly get started with building a functional distributed analysis for domain expertise with limited background of distributed analysis tools. Users need to at least have some knowledge of the standard, the usage of the API implementation they have selected, and some programming skills to configure and compile the standard-based tool on their custom computers, which entails a steep learning curve. In this regard, standalone tools are also inappropriate. For these tools, time management has to be implemented additionally and it is difficult to separate message exchanging code and analysis model function code for scalability (domain users may care more about scalability than analysis efficiency). These are all challenging to achieve for users without much programming experience.

5.4 Recommendations

In a distributed analysis involving multiple analysis models, it is usually natural and straightforward to implement each analysis model as a separate model that interacts with other models and implement a sub-analysis model as a separate sub-model that interacts with other sub-models within the same model. For example, in an analysis of interactions between sequential earthquakes and corresponding recovery processes, it is natural to define a seismic model separately to model the earthquake and its impact on the infrastructure in the environment, and a recovery model to model the recovery effort and how infrastructure functionalities are recovered. The seismic model can include several sub-models that work together to complete its tasks, such as a sub-model to model the effect of the earthquake on the ground surface and other sub-models to model how such effects further interact with and damage buildings, transportation systems, and other infrastructure. Similarly, the recovery model can include a group of sub-models to model how the recovery process evolves with the interaction among resources such as first responders, equipment and material, recovery strategy and as-is recovery status. There can be any number of sub-models, and interactions between them depend on the analysis resolution. Correspondingly a varying number of messages need to be delivered and exchanged.

In order to address the limitations discussed in the last section, a recommended distributed analysis platform is proposed for modeling complex processes in the built environment, which is depicted in Figure 5.4. The system design is proposed to take respective advantages of a standard-based method and a standalone tool (such as HLA and LCM). In the design, two main improvements are made to ensure its benefits.

First, in order to make it easy to develop and convenient to extend a distributed analysis, a message wrapper is developed to receive and send out information for model functions. In this way, analysis models can be developed with only domain knowledge and, if necessary, some knowledge of the settings controlling the resolution of the analysis. The implementation of a message wrapper can be viewed as an improvement upon a standalone tool.

Second, in order to improve scalability, an improvement is made in which mediator-based communication and peer-to-peer communication are jointly used to exchange messages between models and sub-models via message wrappers. A distributed analysis platform based on a single communication approach does not adapt well with the scaling of the analysis in terms of efficiency and time management as discussed previously. The mediator-based communication between models allows for convenient time management and error recovery, and the peer-to-peer communication between sub-models can help reduce the load of the mediator and make the solution adapt well with analysis scale. This improvement can be viewed as an improvement on a standard-based tool such as the RTI of HLA. The following sections will discuss the design of the message wrapper and the data passing between such message wrappers in detail.

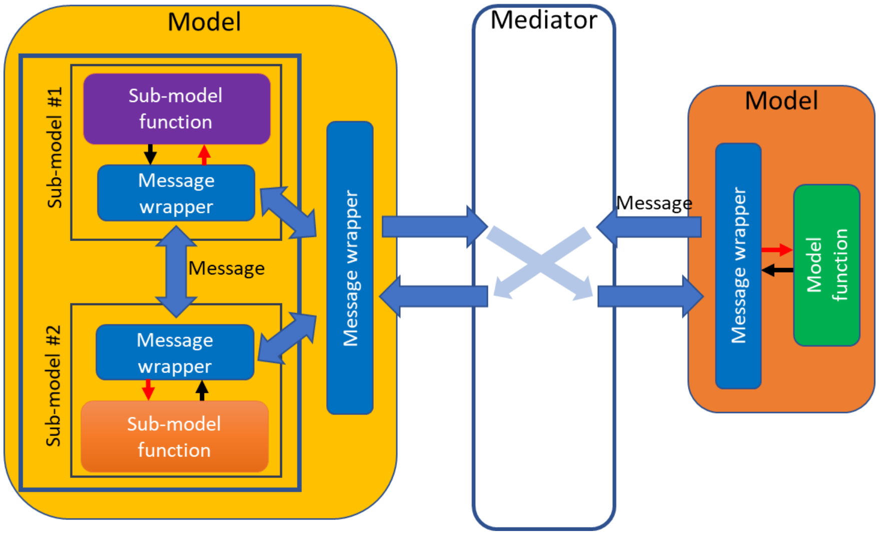


Figure 5.4 A recommended design for a distributed analysis platform

5.4.1 Proposed Design of a Message Wrapper

LCM was previously used as a data passing platform in our previous modeling of wind-building interaction [9,20]. Here we standardize a distributed coupling analysis for general purposes and propose an LCM-based disturbed coupling analysis framework for distributed analyses. As shown in Figure 5.5, analysis model developers only need to follow a couple of fixed steps to create a complex coupling analysis involving multiple analysis models. With the benefit of LCM, different models can be developed with different languages and run on different operating systems listed in Section 5.2.2.2. In this framework, different models can be developed separately and connected with LCM-based message passing. In each model, it first initializes LCM and subscribes to the message channels from which it can get the messages that the model depends on. LCM can help receive the available messages from the subscribed channels, and the current model needs to decide if a received message is one that is currently expected.

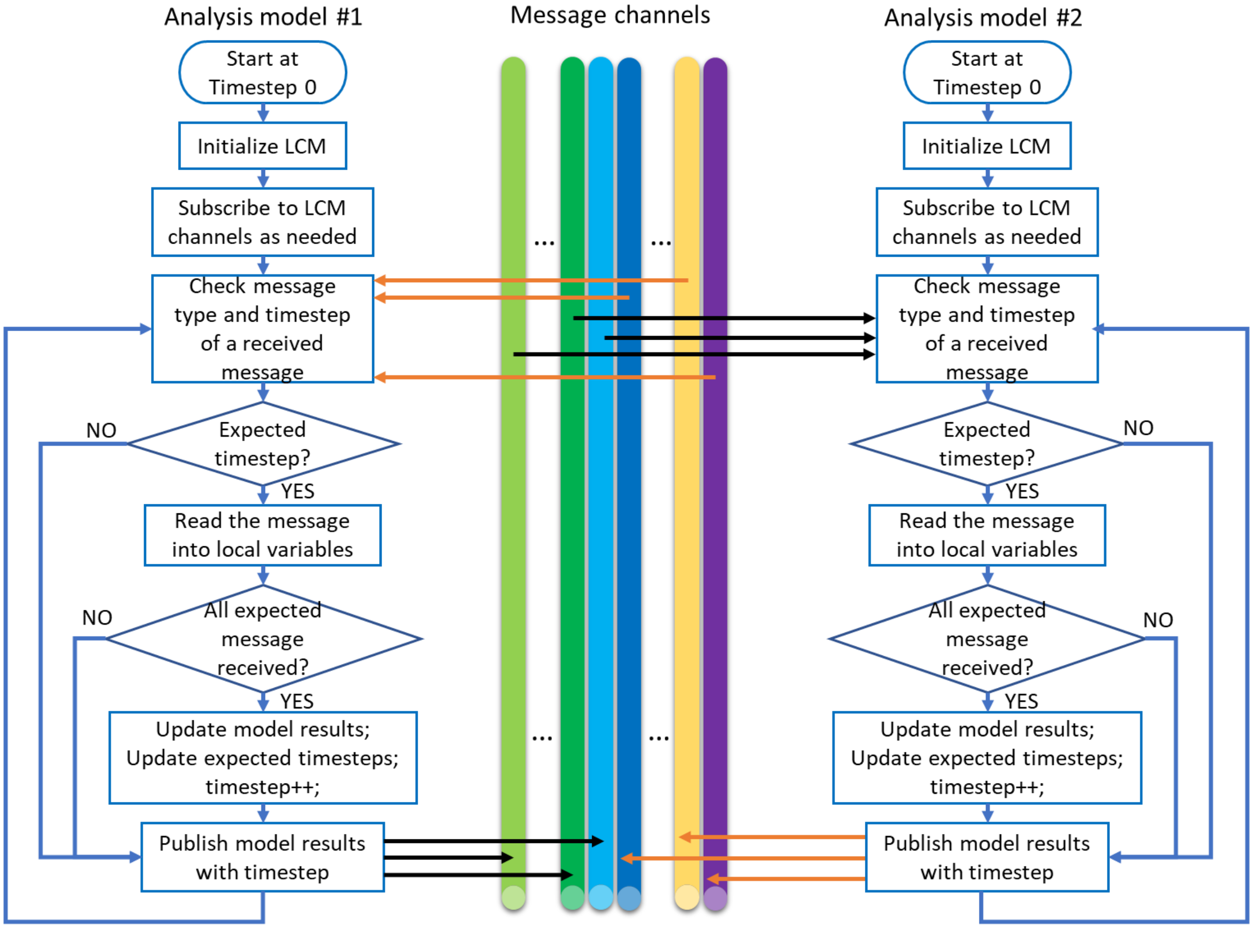

Analysis model #1
Message channels
Analysis model #2
Start at Timestep 0
Initialize LCM
Subscribe to LCM channels as needed
Check message type and timestep of a received message
Expected timestep?
NO
YES
Read the message into local variables
All expected message received?
NO
YES
Update model results; Update expected timesteps; timestep++;
Publish model results with timestep
...
Start at Timestep 0
Initialize LCM
Subscribe to LCM channels as needed
Check message type and timestep of a received message
Expected timestep?
NO
YES
Read the message into local variables
All expected message received?
NO
YES
Update model results; Update expected timesteps; timestep++;
Publish model results with timestep

Figure 5.5 An LCM-based distributed coupling analysis framework

There are two things to check, message type and expected timestep for this type of message. After all the expected messages are received, this model will continue for one timestep, update model results, update expected timestep for each expected type of message, and update the current timestep in the model. After getting new model results, the model will immediately publish it with the current timestep value. It should be noted that a model still needs to keep publishing model results even though when checking if a message is an expected message or if all the expected messages are received, results in failure. The reason is that the messages are live data on the channels, and the same message needs to be sent repeatedly in order to make sure the message can be received by the models that need it to proceed. Different models can be developed separately by following the same steps and then they will automatically work together to make up a complete distributed analysis. Compared with standards and standard-based solutions, this framework is more flexible and more convenient to quickly create a small-scale analysis with domain knowledge.

In addition, as shown in our work in [9,20], even though models and sub-models were not differentiated from each other and all the separate components were implemented as separate models, LCM still worked efficiently to pass messages between different models benefiting from the fact that it uses UDP multicast as its transport and does not use a mediator to route the messages or broker connections between models. This LCM-based model communication scales well with the number of the involved models and is also extensible. However, the code dealing with receiving and sending messages was implemented together with the model functions, and thus it requires model developers to know basic usage of LCM. Moreover, this coding work becomes more complex and error-prone as the number of models increases and the interaction between models becomes complex. This drawback further limits scalability and extensibility in practice and makes it only suitable for relatively small-scale analyses.

Ideally, analysis model developers should not be required to have deep knowledge about the distributed analysis platform being used. Instead, they should be able to focus their attention on developing models in their domains and specifying how they want their models to communicate with each other. In order to achieve this benefit, a message wrapper design is proposed to work together with the model functions and receive and send messages from and to the channels for them. The term “channel” is inherited from LCM and is used to illustrate the new concept design. As shown in Figure 5.6, the proposed message wrapper acts as a bridge connecting message channels and a model function. It subscribes to the channels from which the model function gets input data, decodes messages when required messages are received, calls the model function to update the outputs of the model, encodes output messages and publishes them to the specified channels.

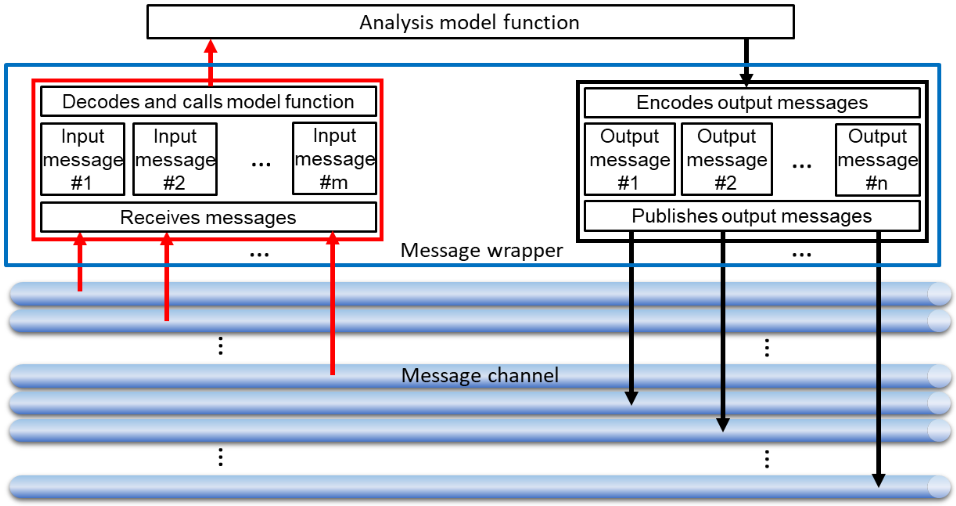


Figure 5.6 Concept of a message wrapper

Figure 5.7 shows the process of developing an analysis model with a message wrapper. This general design can work with any standalone data passing tools such as LCM, ROS, or other custom data passing platforms. For convenience, LCM is used as an example here to show the detailed implementation of the files in Figure 5.7. Model developers first need to prepare two files: a model configuration file and a message definition file. The model configuration file includes all the settings about the model, including model name, the channels this model needs to subscribe to, the channels it needs to publish on, time step relationship between the current model and the messages it depends on, the model's dependence on historical data, and whether the model needs to publish initial data for other models to start working.

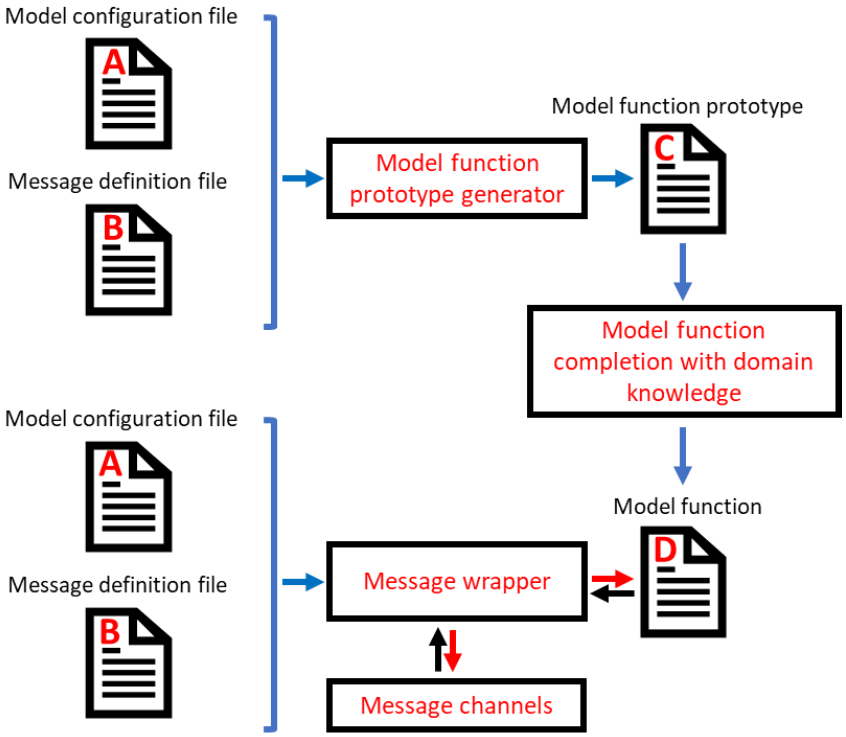
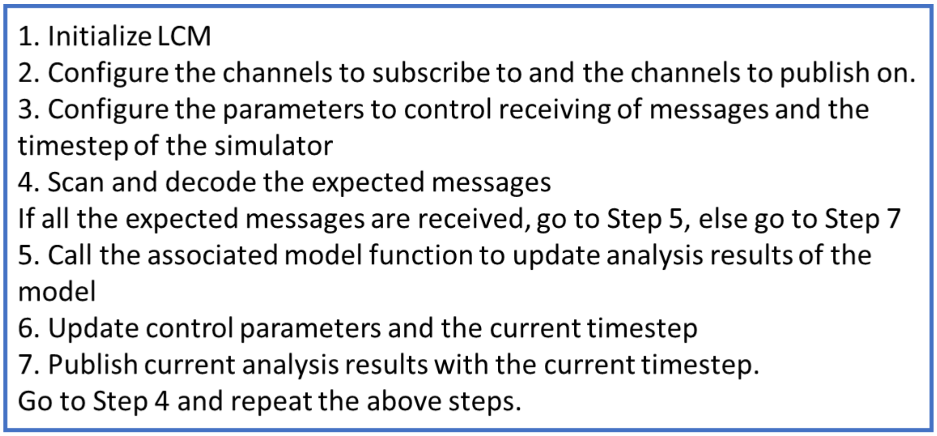


Figure 5.7 Procedures for developing an analysis model with a message wrapper

The message definition file includes the names of the variables in each message and the corresponding data types. It should be noted that even though LCM can decode the message and its variable types automatically, these variables need to be stored as local variables in the wrapper and thus the variable types still have to be provided in the message definition file. In the case of LCM, it is straightforward to prepare these two files by drawing a communication network and referring to the LCM message definitions. When these two files are ready, a model function prototype generator (Figure 5.7) is used to generate the function prototype of the model function.

This function prototype generator is implemented in a simple way that all the variables included in the input messages are listed as input arguments, all the variables in the output messages are listed as return values and the model name is used as the function name. Therefore, only the information about model name, channels to subscribe to, and channels to publish on in the model configuration file are used to generate the function prototype. Then model developers need to complete the created model function with only domain knowledge. After completing the model function, the message wrapper can be run to handle message exchange when it works with the same model configuration file, the same message definition file, and the completed model function.

Figure 5.8 shows how a message wrapper computes output messages based on the input messages and publishes the output messages on the specified channels. This process is very similar to that in the analysis models depicted in Figure 5.5. The only difference is that the model function and the code for message exchange are completely separated with the proposed message wrapper.



```
1. Initialize LCM
2. Configure the channels to subscribe to and the channels to publish on.
3. Configure the parameters to control receiving of messages and the
timestep of the simulator
4. Scan and decode the expected messages
If all the expected messages are received, go to Step 5, else go to Step 7
5. Call the associated model function to update analysis results of the
model
6. Update control parameters and the current timestep
7. Publish current analysis results with the current timestep.
Go to Step 4 and repeat the above steps.
```

Figure 5.8 An example implementation of the message wrapper for LCM

Therefore, domain users just need a little effort to develop a model since all they need to know is the domain knowledge to complete the model function and the relationship between different models. Besides, in the process of completing the model function, it is flexible for users to use any useful software and/or hardware to facilitate model development and/or accelerate the model.

In order to demonstrate the effectiveness, besides the results in [9,20], the proposed LCM-based framework and an LCM version of the wrapper are also used to replicate the active control algorithm described in [273]. As shown in Figure 5.9, the active control system is formalized as three models, where $P(t)$ represents the force caused by a wind excitation at time t, $D(t)$, $V(t)$ and $A(t)$ represent structure displacement, velocity and acceleration at time t, and $P_c(t)$ is the active control force at time t. The wind excitation function implemented Equation (1-5) in [20], the structure dynamics function implemented Equation (8) in [273], and the adaptive control function implemented Equation (12) in [273] where either displacement, velocity or acceleration can be used as the variable to be controlled. Two analysis results with and without active control are shown in Figure 5.10 and Figure 5.11. It can be observed that with active control, acceleration was successfully limited to the range of [-1.5, +1.5] m/s^2, and the displacement and the velocity responses were impacted correspondingly. The results demonstrated that the framework and the wrapper help discover the interdependency between the structure dynamics model and the adaptive control model. With domain knowledge from wind engineering and structural engineering, this distributed analysis model can be constructed conveniently by following the above fixed steps without knowing how to use LCM to exchange messages.

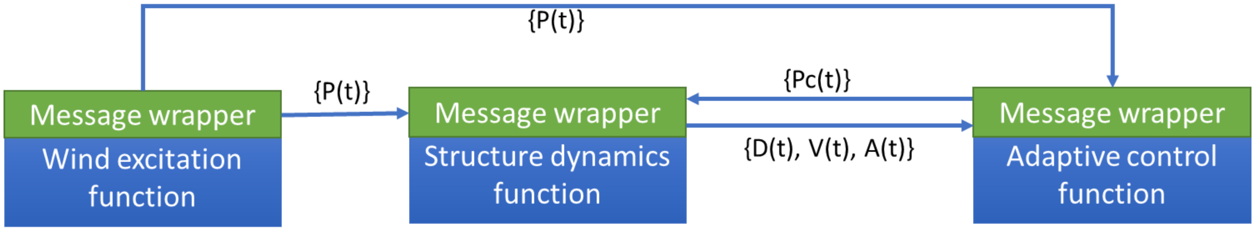


Figure 5.9 Distributed analysis design of an active control algorithm

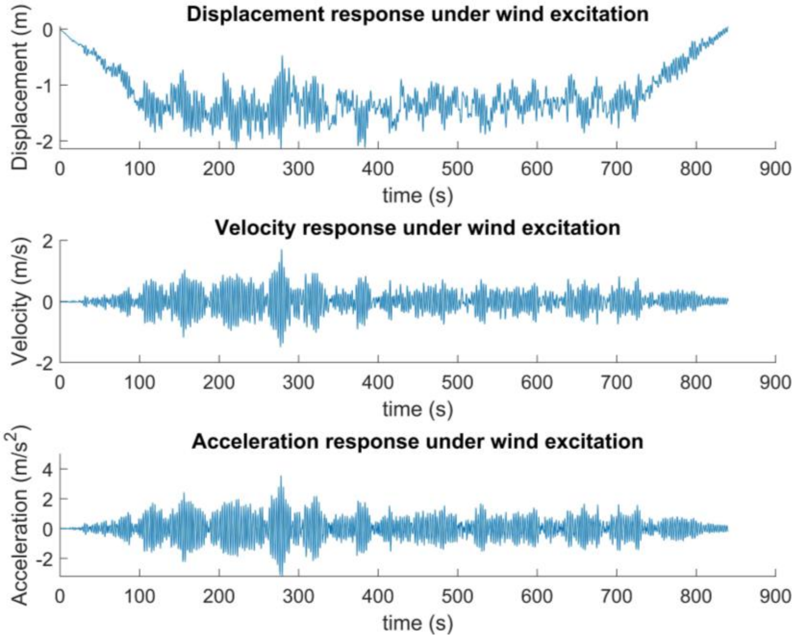


Figure 5.10 Analysis results without active control

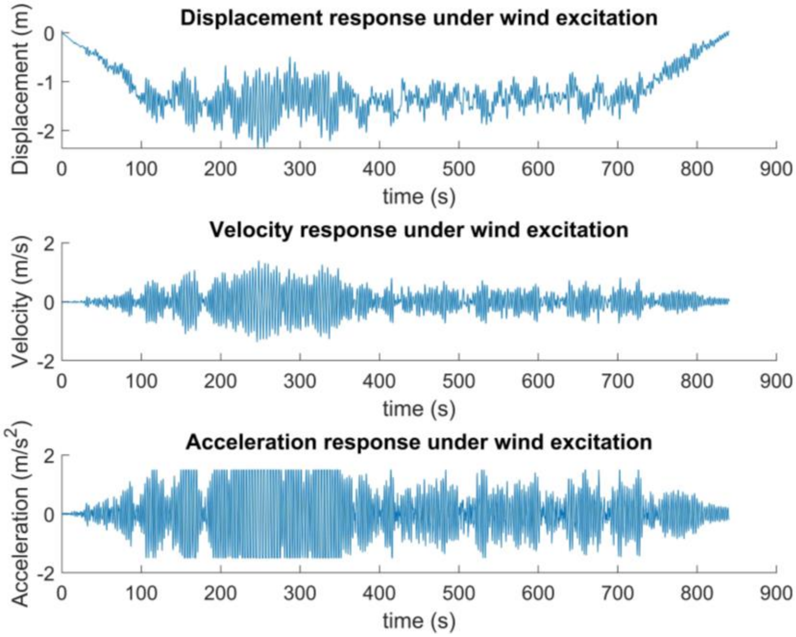


Figure 5.11 Analysis results with active control of acceleration

However, it should be noted that even though the message wrapper can be used as an extension to any data passing platform and help improve the scalability and extensibility in terms of implementation, current widely used mediator-based data passing platforms generally suffer from scalability problems in message communication. For those that adapt well with the analysis scale, such as LCM, they still lack the necessary time management and error-recovery mechanism for robust analysis. This challenge leads to the second proposed improvement that jointly uses peer-to-peer and mediator-based communication.

5.4.2 Hybrid Data Passing Between Message Wrappers

As shown in the recommended design in Figure 5.4, peer-to-peer communication is adopted to handle communication between different sub-models in each model and the model itself via message wrappers. This local communication generally needs more frequent and more extensive message exchange as compared to the model-model communication and is suggested to be implemented with UDP multicast that was also the transport utilized in LCM. The benefit to this approach is that there are no additional copies of messages that otherwise would increase linearly with the number of subscribers and result in a significant overhead if the number of subscribers is large. Moreover, even though LCM was originally designed for real-time robotic applications, based on previous experience [9,20], it was shown to work efficiently in timestep-based analyses. However, mediator-based communication between models is suggested to be implemented with TCP transport that provides reliable and ordered information delivery.

For the two types of communication, different marshaling methods can be chosen according to the tradeoff between transmission efficiency and marshaling cost. The message wrapper of a model function can be designed to decode the marshaled messages from its sub-models and marshal them in a different way and communicate them via the mediator. Generally, in order to simplify the platform design, the same message marshaling format would be shared between the two ways of communication. LCM marshaling is a good example. With the benefit of UDP multicast in LCM, the lcm-spy tool can be used to inspect traffic without additional cost. A similar traffic inspection tool can be developed for communication between local sub-models. However, for the inspection of communication between models, the inspection tool needs to be implemented as a separate inspection model that subscribes to all the channels and thus adds additional inspection cost.

For a platform only based on mediator-based communication, the load of the mediator increases with the number of messages that need to be delivered at any time and is usually the bottleneck of a large-scale distributed analysis. With the proposed hybrid communication, this issue can be greatly improved since all the sub-models will be handled by peer-to-peer communication that does not need the mediator and can be implemented efficiently. In addition, the mediator in the proposed solution is no different from the mediator in an RTI for HLA and can easily take advantage of the time management methods in HLA. The platform can be also integrated with some error recovery mechanism by using a certain number of historical messages from the models kept in the mediator. In this regard, the hybrid communication design jointly uses the ideas of LCM and HLA and capitalizes on their respective advantages.

5.5 Conclusions and Future Work

Distributed coupling analysis enables coupling analysis to identify deep interdependencies among different processes in the built environment. This paper provides a systematic review of existing standards, platforms and standalone data passing tools for distributed analyses, identifies the limitations in the existing tools, and proposes two recommendations on improving the design of a distributed coupling analysis platform. This survey study offers a reference for researchers when selecting tools for distributed analyses. Moreover, this article serves as a guiding document towards developing an improved distributed coupling analysis platform for complex process analyses by identifying the current limitations and providing feasible recommendations for future studies.

Chapter 6
Conclusions

6.1 Significance of Research

This research addresses three categories of scene understanding algorithms that make up fundamental building blocks for diverse potential applications either on construction sites or in operational buildings. It establishes a key step to improved construction and utilization and operation of the built environment.

For example, localization ability is not only the essential requirement for a robot to automatically execute a task such as construction progress monitoring [13], bridge bearing inspection [6], and automatic tunnel inspection [14], but it also provides a useful tool for indoor navigation in a complex environment such as a big building, a hospital or a museum. This is especially useful to those people with physical disabilities (PPD), who usually have limited access to external information and have difficulties finding their way to their destinations. The proposed localization algorithm can be modified and run on a smartphone with a camera to provide such individuals turn-by-turn instructions to their destinations. This can significantly help improve their independence and quality of life. The localization algorithm can also run with a drone-mounted camera, and together with the proposed barcode extraction framework, items in a warehouse or a distribution center can be scanned much more efficiently. This will not only liberate workers from the repeated and time-consuming work of scanning tens of thousands of barcodes but also improve their safety since they will not need to use a forklift to reach the vicinity of the barcodes at high locations to scan them manually.

In addition, the proposed distributed coupling analysis framework can be used to analyze building energy usage and performance deterioration to optimize building retrofit plans, analyze people’s response in fire emergencies to improve evacuation training and optimize safety exit distribution, and estimate the damage to a community caused by a natural disaster to optimize recovery strategy and improve community resilience.

6.2 Research Contributions

In general, this research explores and advances three categories of scene understanding, localization, object recognition, and distributed coupling analysis. The specific contributions in each category are summarized below.

1. Improved vSLAM algorithms for locating applications
    - A new RTLS was designed based on an OGM enhanced vSLAM algorithm, which provides high-accuracy localization and enables user interaction with the localization system for practical applications.
    - The proposed RTLS works in SLAM mode and localization mode and can switch between the two modes flexibly as needed to update or expand an existing map.
    - A sparse feature map and an occupancy grid map were used together to compensate each other.
    - A new fiducial landmark-based method was designed to evaluate the localization accuracy of a vSLAM system.
    - A scene-adaptive feature transform (SAFT), which self-adapts to currently observed scenes, was proposed to improve learning-based descriptors’ matching robustness for vSLAM applications.

- An integration framework was proposed to integrate the SAFT into a state-of-the-art feature-based SLAM and train it online.

2. 1D barcode extraction for asset tracking
   - A drone-assisted asset scan framework was designed for automatic asset tracking.
   - A 1D barcode extraction framework was designed to automatically extract 1D barcodes from video scan data collected in large-scale environments.
   - The 1D barcode extraction framework was tested in a warehouse environment and proved applicable for inventory management.

3. Distributed coupling analysis for deep interdependency discovery
   - Existing standards, platforms, and data passing tools were reviewed for distributed analyses, and limitations were summarized from the perspective of distributed analysis of built-environment processes.
   - An LCM-based distributed coupling analysis framework and a message wrapper were designed and tested with analysis models from wind engineering and structural engineering.
   - A hybrid data passing method was proposed to improve efficiency and time management for the design of an improved distributed analysis platform.

6.3 Future Directions

6.3.1 Localization

In order to improve vSLAM's robustness under motion blur, future research will explore the joint usage of IMU and camera, considering that IMU can provide pose estimation when vision cannot track features well and tracking results from vision can help eliminate the drift introduced into IMU-based pose estimation. Future research also includes direct SLAM and its integration with feature-based SLAM for robust pose estimation in challenging environments without sufficient features. For robot motion in 3D space, 3D occupancy grid mapping needs to be explored.

Topological maps also need to be explored for more efficient path planning in a large-scale environment.

6.3.2 Object Recognition

This research is limited to 1D barcode-based object recognition for applications in well-organized environments such as a warehouse or a distribution center. Future research will explore more general feature-based object recognition based on learning-based methods. This includes recognition of the objects that a robot needs to manipulate, human and static obstacles recognition and object velocity estimation for environment-aware path planning, and semantic segmentation of traversable area for more accurate local path planning.

6.3.3 Distributed Coupling Analysis

In order to further facilitate the usage of the proposed distributed coupling analysis framework for distributed analyses, future research includes implementation of different versions of the message wrapper to support different programming languages, development of a GUI for more convenient operation and visualization, implementation of more time management functionalities to support time-based, event-driven, and real-time analysis models, and dynamic distribution of the load of a mediator over multiple mediators and multiple devices.

Bibliography

[1] K.M. Lundeen, V.R. Kamat, C.C. Menassa, W. McGee, Scene understanding for adaptive manipulation in robotized construction work, Automation in Construction, 82 (2017) 16-30, https://doi.org/10.1016/j.autcon.2017.06.022.

[2] C.-J. Liang, K.M. Lundeen, W. McGee, C.C. Menassa, S. Lee, V.R. Kamat, A vision-based marker-less pose estimation system for articulated construction robots, Automation in Construction, 104 (2019) 80-94.

[3] J.-G. Juang, C.-Y. Yang, Document delivery robot based on image processing and fuzzy control, Transactions of the Canadian Society for Mechanical Engineering, 40 (5) (2016) 677-692.

[4] H. Grewal, A. Matthews, R. Tea, K. George, Lidar-based autonomous wheelchair, 2017 IEEE Sensors Applications Symposium (SAS), IEEE, 2017, pp. 1-6, ISBN: 1509032029.

[5] L. Zhang, F. Yang, Y.D. Zhang, Y.J. Zhu, Road crack detection using deep convolutional neural network, 2016 IEEE international conference on image processing (ICIP), IEEE, 2016, pp. 3708-3712, ISBN: 1467399612.

[6] H. Peel, S. Luo, A. Cohn, R. Fuentes, Localisation of a mobile robot for bridge bearing inspection, Automation in Construction, 94 (2018) 244-256, https://doi.org/10.1016/j.autcon.2018.07.003.

[7] P. Ibach, V. Stantchev, F. Lederer, A. Weiß, T. Herbst, T. Kunze, WLAN-based asset tracking for warehouse management, IADIS International Conference e-Commerce, Porto, Portugal, 2005, pp. 15-17.

[8] A. Thomas, C.C. Menassa, V.R. Kamat, Lightweight and adaptive building simulation (LABS) framework for integrated building energy and thermal comfort analysis, Building Simulation, Vol. 10, Springer, 2017, pp. 1023-1044, ISBN: 1996-3599, https://doi.org/10.1007/s12273-017-0409-5.

[9] S.-Y. Lin, L. Xu, W.-C. Chuang, S. El-Tawil, S.M.J. Spence, V.R. Kamat, C.C. Menassa, J. McCormick, Modeling Interactions in Community Resilience, Structures Conference, Fort Worth, Texas, 2018, https://doi.org/10.1061/9780784481349.001.

[10] iRobot Corporation, iRobot, 2019, Accessed 10/24/2019, https://www.irobot.com/.

[11] SuperDroid Robots Inc, Tactical Surveillance Robots, 2019, Accessed 10/24/2019, http://www.sdrtactical.com/Surveillance-Robots/.

[12] B.R. Mantha, C.C. Menassa, V.R. Kamat, Ambient data collection in indoor building environments using mobile robots, Proceedings of the International Symposium on Automation and Robotics in Construction (ISARC), Vol. 33, IAARC, Auburn, Alabama, USA 2016, p. 1, Retrieved from https://search.proquest.com/docview/1823082542?pq-origsite=gscholar.

[13] P. Kim, J. Chen, Y.K. Cho, SLAM-driven robotic mapping and registration of 3D point clouds, Automation in Construction, 89 (2018) 38-48, https://doi.org/10.1016/j.autcon.2018.01.009.

[14] E. Menendez, J.G. Victores, R. Montero, S. Martínez, C. Balaguer, Tunnel structural inspection and assessment using an autonomous robotic system, Automation in Construction, 87 (2018) 117-126, https://doi.org/10.1016/j.autcon.2017.12.001.

[15] L. Xu, V.R. Kamat, C.C. Menassa, Automatic extraction of 1D barcodes from video scans for drone-assisted inventory management in warehousing applications, International Journal of Logistics Research and Applications, 21 (3) (2018) 243-258, https://doi.org/10.1080/13675567.2017.1393505.

[16] Y. Angal, A. Gade, Development of library management robotic system, 2017 International Conference on Data Management, Analytics and Innovation (ICDMAI), IEEE, 2017, pp. 254-258, ISBN: 1509040838.

[17] C. Feng, Y. Xiao, A. Willette, W. McGee, V.R. Kamat, Vision guided autonomous robotic assembly and as-built scanning on unstructured construction sites, Automation in Construction, 59 (2015) 128-138, https://doi.org/10.1016/j.autcon.2015.06.002.

[18] Z. Tan, L. Zhen, L. Xiao, Q. Sun, Parcel Sorting Optimization in Double-Layer Automatic Sorting System, Proceedings of the KES International Symposium on Smart Transportation Systems, Springer, St. Julian's, Malta, 2019, pp. 251-259.

[19] H. Mohseni, S. Setunge, G. Zhang, R. Wakefield, Probabilistic deterioration prediction and cost optimization for community buildings using Monte-Carlo simulation, ICOMS Asset Management Conference, Asset Management Council Limited, Hobart, Australia, 2012, pp. 1-9, ISBN: 1329-7198.

[20] S.-Y. Lin, W.-C. Chuang, L. Xu, S. El-Tawil, S.M. Spence, V.R. Kamat, C.C. Menassa, J. McCormick, Framework for Modeling Interdependent Effects in Natural Disasters: Application to Wind Engineering, Journal of Structural Engineering, 145 (5) (2019) 04019025, https://doi.org/10.1061/(ASCE)ST.1943-541X.0002310.

[21] A. Ren, C. Chen, J. Shi, L. Zou, Application Of Virtual Reality Technology To Evacuation Simulation In Fire Disaster, Proceedings of the International Conference on Computer Graphics & Virtual Reality (CGVR), CSREA Press, Las Vegas, Nevada, USA, 2006, pp. 15-21, Retrieved from

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.519.9276&rep=rep1&type=pdf .

[22] S.L. Cutter, L. Barnes, M. Berry, C. Burton, E. Evans, E. Tate, J. Webb, A place-based model for understanding community resilience to natural disasters, Global environmental change, 18 (4) (2008) 598-606.

[23] T.S. Levitt, Qualitative navigation for mobile robots, International Journal of Artificial Intelligence, 44 (1990) 305-360.

[24] P.E. Hart, N.J. Nilsson, B. Raphael, A formal basis for the heuristic determination of minimum cost paths, IEEE transactions on Systems Science and Cybernetics, 4 (2) (1968) 100-107, https://doi.org/10.1109/TSSC.1968.300136.

[25] S.M. LaValle, Rapidly-exploring random trees: A new tool for path planning, TR 98-11, Computer Science Department, Iowa State University, 1998.

[26] Y. Cheng, X. Wang, M. Morelande, B. Moran, Information geometry of target tracking sensor networks, Information Fusion, 14 (3) (2013) 311-326, https://doi.org/10.1016/j.inffus.2012.02.005.

[27] J.A.P. Montañés, A.M. Rodríguez, I.S. Prieto, Smart indoor positioning/location and navigation: A lightweight approach, International Journal of Interactive Multimedia and Artificial Intelligence, 2 (2) (2013) 43-50, https://doi.org/10.9781/ijimai.2013.225.

[28] W. Lu, G.Q. Huang, H. Li, Scenarios for applying RFID technology in construction project management, Automation in Construction, 20 (2) (2011) 101-106, https://doi.org/10.1016/j.autcon.2010.09.007.

[29] J.M. Sardroud, Influence of RFID technology on automated management of construction materials and components, Scientia Iranica, 19 (3) (2012) 381-392, https://doi.org/10.1016/j.scient.2012.02.023.

[30] N. Li, B. Becerik-Gerber, Performance-based evaluation of RFID-based indoor location sensing solutions for the built environment, Advanced Engineering Informatics, 25 (3) (2011) 535-546, https://doi.org/10.1016/j.aei.2011.02.004.

[31] B. Ozdenizci, K. Ok, V. Coskun, M.N. Aydin, Development of an indoor navigation system using NFC technology, International Conference on Information and Computing (ICIC), IEEE, Phuket Island, Thailand, 2011, pp. 11-14, ISBN: 1612846882, https://doi.org/10.1109/ICIC.2011.53.

[32] S.S. Chawathe, Beacon placement for indoor localization using bluetooth, International Conference on Intelligent Transportation Systems, IEEE, Beijing, China, 2008, pp. 980-985, ISBN: 142442111X, https://doi.org/10.1109/ITSC.2008.4732690.

[33] A. Singer, M. Oelze, A. Podkowa, Mbps experimental acoustic through-tissue communications: MEAT-COMMS, International Workshop on Signal Processing

Advances in Wireless Communications (SPAWC), IEEE, Edinburgh, UK, 2016, pp. 1-4, ISBN: 1509017496, https://doi.org/10.1109/SPAWC.2016.7536815.

[34] M.-W. Park, C. Koch, I. Brilakis, Three-dimensional tracking of construction resources using an on-site camera system, Journal of Computing in Civil Engineering, 26 (4) (2011) 541-549, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000168.

[35] M.-W. Park, A. Makhmalbaf, I. Brilakis, Comparative study of vision tracking methods for tracking of construction site resources, Automation in Construction, 20 (7) (2011) 905-915, https://doi.org/10.1016/j.autcon.2011.03.007.

[36] M.-W. Park, I. Brilakis, Construction worker detection in video frames for initializing vision trackers, Automation in Construction, 28 (2012) 15-25, https://doi.org/10.1016/j.autcon.2012.06.001.

[37] B.R. Mantha, C.C. Menassa, V.R. Kamat, Robotic data collection and simulation for evaluation of building retrofit performance, Automation in Construction, 92 (2018) 88-102, https://doi.org/10.1016/j.autcon.2018.03.026.

[38] A.R. Jimenez, F. Seco, C. Prieto, J. Guevara, A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU, International Symposium on Intelligent Signal Processing (WISP), IEEE, Budapest, Hungary, 2009, pp. 37-42, ISBN: 1424450578, https://doi.org/10.1109/WISP.2009.5286542.

[39] S. Kohlbrecher, O. Von Stryk, J. Meyer, U. Klingauf, A flexible and scalable slam system with full 3d motion estimation, IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), IEEE, Kyoto, Japan, 2011, pp. 155-160, ISBN: 1612847692, https://doi.org/10.1109/SSRR.2011.6106777.

[40] G. Grisetti, C. Stachniss, W. Burgard, Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling, Proceedings of the International Conference on Robotics and Automation (ICRA), IEEE, Barcelona, Spain, 2005, pp. 2432-2437, https://doi.org/10.1109/ROBOT.2005.1570477.

[41] G. Grisetti, C. Stachniss, W. Burgard, Improved techniques for grid mapping with rao-blackwellized particle filters, IEEE Transactions on Robotics, 23 (1) (2007) 34-46, https://doi.org/10.1109/TRO.2006.889486.

[42] W. Hess, D. Kohler, H. Rapp, D. Andor, Real-time loop closure in 2D LIDAR SLAM, IEEE International Conference on Robotics and Automation (ICRA), IEEE, Stockholm, Sweden, 2016, pp. 1271-1278, ISBN: 1467380261, https://doi.org/10.1109/ICRA.2016.7487258.

[43] M. Filipenko, I. Afanasyev, Comparison of various slam systems for mobile robot in an indoor environment, International Conference on Intelligent Systems (IS), IEEE, Funchal, Madeira, Portugal, 2018, pp. 400-407, ISBN: 1538670976, https://doi.org/10.1109/IS.2018.8710464.

[44] A.J. Davison, I.D. Reid, N.D. Molton, O. Stasse, MonoSLAM: Real-time single camera SLAM, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 29 (6) (2007) 1052-1067, https://doi.org/10.1109/TPAMI.2007.1049.

[45] G. Klein, D. Murray, Parallel tracking and mapping for small AR workspaces, IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR), IEEE, Nara, Japan, 2007, pp. 225-234, ISBN: 142441749X, https://doi.org/10.1109/ISMAR.2007.4538852.

[46] R. Mur-Artal, J. Montiel, J.D. Tardós, Orb-slam: a versatile and accurate monocular slam system, IEEE Transactions on Robotics, 31 (5) (2015) 1147-1163, https://doi.org/10.1109/TRO.2015.2463671.

[47] R.A. Newcombe, S.J. Lovegrove, A.J. Davison, DTAM: Dense tracking and mapping in real-time, IEEE International Conference on Computer Vision (ICCV), IEEE, Barcelona, Spain, 2011, pp. 2320-2327, ISBN: 1457711028, https://doi.org/10.1109/ICCV.2011.6126513.

[48] J. Engel, J. Sturm, D. Cremers, Semi-dense visual odometry for a monocular camera, IEEE International Conference on Computer Vision (ICCV), IEEE, Sydney, NSW, Australia, 2013, pp. 1449-1456, ISBN: 1479928402, https://doi.org/10.1109/ICCV.2013.183.

[49] C. Forster, M. Pizzoli, D. Scaramuzza, SVO: Fast semi-direct monocular visual odometry, IEEE International Conference on Robotics and Automation (ICRA), IEEE, Hong Kong, China, 2014, pp. 15-22, ISBN: 1479936855, https://doi.org/10.1109/ICRA.2014.6906584.

[50] J. Engel, T. Schöps, D. Cremers, LSD-SLAM: Large-scale direct monocular SLAM, Proceedings of the European Conference on Computer Vision (ECCV), Springer, Zurich, Switzerland, 2014, pp. 834-849, ISBN: 331910604X, https://doi.org/10.1007/978-3-319-10605-2_54.

[51] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Columbus, OH, USA, 2014, pp. 580-587, https://doi.org/10.1109/CVPR.2014.81.

[52] R. Girshick, Fast R-CNN, Proceedings of the International Conference on Computer Vision (ICCV), IEEE, Santiago, Chile, 2015, pp. 1440-1448, https://doi.org/10.1109/ICCV.2015.169.

[53] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, IEEE transactions on pattern analysis and machine intelligence, 39 (6) (2015) 1137 - 1149, https://doi.org/10.1109/TPAMI.2016.2577031.

[54] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, Proceedings of the Conference on Computer Vision and Pattern

Recognition (CVPR), IEEE, Las Vegas, NV, USA, 2016, pp. 779-788, https://doi.org/10.1109/CVPR.2016.91.

[55] J. Redmon, A. Farhadi, YOLO9000: better, faster, stronger, Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Honolulu, HI, USA, 2017, pp. 7263-7271, https://doi.org/10.1109/CVPR.2017.690.

[56] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, arXiv preprint arXiv:1804.02767 (2018), Retrieved from https://arxiv.org/abs/1804.02767.

[57] E. Olson, AprilTag: A robust and flexible visual fiducial system, IEEE International Conference on Robotics and Automation (ICRA), IEEE, Shanghai, China, 2011, pp. 3400-3407, ISBN: 1612843859, https://doi.org/10.1109/ICRA.2011.5979561.

[58] S. Garrido-Jurado, R. Muñoz-Salinas, F.J. Madrid-Cuevas, M.J. Marín-Jiménez, Automatic generation and detection of highly reliable fiducial markers under occlusion, Pattern Recognition, 47 (6) (2014) 2280-2292.

[59] R. Muniz, L. Junco, A. Otero, A robust software barcode reader using the Hough transform, International Conference on Information Intelligence and Systems, IEEE, Bethesda, MD, USA, 1999, pp. 313-319, ISBN: 0769504469, http://doi.org/10.1109/ICIIS.1999.810282.

[60] E. Ohbuchi, H. Hanaizumi, L.A. Hock, Barcode readers using the camera device in mobile phones, International Conference on Cyberworlds IEEE, Tokyo, Japan, 2004, pp. 260-265, ISBN: 0769521401, https://doi.org/10.1109/CW.2004.23.

[61] R. Adelmann, M. Langheinrich, C. Floerkemeier, A Toolkit for Bar-Code-Recognition and -Resolving on Camera Phones - Jump Starting the Internet of Things, Proceedings of the workshop on Mobile and Embedded Interactive Systems (MEIS'06) at Informatik 2006, GI Lecture Notes in Informatics Series (LNI), Dresden, Germany, 2006.

[62] O. Gallo, R. Manduchi, Reading 1D barcodes with mobile phones using deformable templates, IEEE transactions on pattern analysis and machine intelligence, 33 (9) (2011) 1834-1843, https://doi.org/10.1109/TPAMI.2010.229.

[63] O. Gallo, R. Manduchi, Reading challenging barcodes with cameras, Proceedings of the Workshop on Applications of Computer Vision (WACV), IEEE, Snowbird, UT, USA, 2009, pp. 1-6, ISBN: 1424454972, https://doi.org/10.1109/WACV.2009.5403090.

[64] J. Liyanage, Efficient decoding of blurred, pitched, and scratched barcode images, Proceedings of the 2nd International Conference on Industrial and Information Systems, 2007.

[65] Inlite Research Inc, ClearImage SDK, 2005.https://www.inliteresearch.com/.

[66] C. Xiong, X. Lu, H. Guan, Z. Xu, A nonlinear computational model for regional seismic simulation of tall buildings, Bulletin of Earthquake Engineering, 14 (4) (2016) 1047-1069, https://doi.org/10.1007/s10518-016-9880-0.

[67] P. Latcharote, K. Terada, M. Hori, F. Imamura, A prototype seismic loss assessment tool using integrated earthquake simulation, International journal of disaster risk reduction, 31 (2018) 1354-1365, https://doi.org/10.1016/j.ijdrr.2018.03.026.

[68] Y. Wang, C. Chen, J. Wang, R. Baldick, Research on resilience of power systems under natural disasters—A review, IEEE Transactions on Power Systems, 31 (2) (2015) 1604-1613, https://doi.org/10.1109/TPWRS.2015.2429656.

[69] C. Barrett, R. Beckman, K. Channakeshava, F. Huang, V.A. Kumar, A. Marathe, M.V. Marathe, G. Pei, Cascading failures in multiple infrastructures: From transportation to communication network, 2010 5th International Conference on Critical Infrastructure (CRIS), IEEE, 2010, pp. 1-8, ISBN: 1424480817, https://doi.org/10.1109/CRIS.2010.5617569.

[70] S. Jain, C. McLean, Simulation for emergency response: a framework for modeling and simulation for emergency response, Proceedings of the 35th conference on Winter simulation: driving innovation, Winter Simulation Conference, 2003, pp. 1068-1076, ISBN: 0780381327, https://doi.org/10.1109/WSC.2003.1261532.

[71] G. Bunea, F. Leon, G.M. Atanasiu, Postdisaster evacuation scenarios using multiagent system, Journal of Computing in Civil Engineering, 30 (6) (2016) 05016002, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000575.

[72] H. Xie, N.N. Weerasekara, R.R. Issa, Improved System for Modeling and Simulating Stadium Evacuation Plans, Journal of Computing in Civil Engineering, 31 (3) (2016) 04016065, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000634.

[73] K. Liu, X. Shen, N.D. Georganas, A.E. Saddik, A. Boukerche, SimSITE: The HLA/RTI based emergency preparedness and response training simulation, Proceedings of the International Symposium on Distributed Simulation and Real-Time Applications, IEEE, Chania, Greece, 2007, pp. 59-63, ISBN: 0769530117, https://doi.org/10.1109/DS-RT.2007.33.

[74] F. Fiedrich, An HLA-based multiagent system for optimized resource allocation after strong earthquakes, Proceedings of the 38th conference on Winter simulation, Winter Simulation Conference, 2006, pp. 486-492, ISBN: 1424405017, https://doi.org/10.1109/WSC.2006.323120.

[75] A.S. Tanenbaum, M. Van Steen, Distributed systems: principles and paradigms, Prentice-Hall, 2007, ISBN: 0132392275.

[76] D.S. Committee, IEEE Standard for Distributed Interactive Simulation--Application Protocols, IEEE Std, Vol. 1278.1-2012, 2012, pp. 1-1437, https://standards.ieee.org/standard/1278_1-2012.html.

[77] IEEE Std. 1516-2000, IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-- Framework and Rules, IEEE Std 1516-2010 (Revision of IEEE Std 1516-2000), 2010, pp. 1-38, https://doi.org/10.1109/IEEESTD.2010.5553440.

[78] E.T. Powell, J.R. Noseworthy, The test and training enabling architecture (TENA), in: A. Tolk (Ed.), Engineering principles of combat modeling and distributed simulation, Vol. 449, John Wiley & Sons, Inc., Hoboken, New Jersey, 2012, ISBN: 9780470874295 https://doi.org/10.1002/9781118180310.ch20.

[79] OMG, Data Distribution Service (DDS), Version 1.4, 2015, Accessed Dec. 19, https://www.omg.org/spec/DDS/1.4.

[80] K.M. Lundeen, V.R. Kamat, C.C. Menassa, W. McGee, Autonomous motion planning and task execution in geometrically adaptive robotized construction work, Automation in Construction, 100 (2019) 24-45, https://doi.org/10.1016/j.autcon.2018.12.020.

[81] C. Yuan, S. Li, H. Cai, V.R. Kamat, GPR Signature Detection and Decomposition for Mapping Buried Utilities with Complex Spatial Configuration, Journal of Computing in Civil Engineering, 32 (4) (2018) 04018026, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000764.

[82] A. Llarena, R. Rojas, I Am Alleine, the Autonomous Wheelchair at Your Service, Intelligent Autonomous Systems 13, Springer, 2016, pp. 1613-1626 https://doi.org/10.1007/978-3-319-08338-4_116.

[83] V. Sommer, Service robot for the automatic suction of dust from floor surfaces, Vol. 6,370,453, U.S. Patent, 2002.

[84] L. Xu, V.R. Kamat, C.C. Menassa, Automatic Barcode Extraction for Efficient Large-Scale Inventory Management, Proceedings of the International Workshop on Computing in Civil Engineering (IWCCE), ASCE, Seattle, Washington, USA, 2017, pp. 340-348, https://doi.org/10.1061/9780784480830.042.

[85] S.-K. Kim, J.S. Russell, K.-J. Koo, Construction robot path-planning for earthwork operations, Journal of Computing in Civil Engineering, 17 (2) (2003) 97-104, https://doi.org/10.1061/(ASCE)0887-3801(2003)17:2(97).

[86] C. Feng, S. Dong, K. Lundeen, Y. Xiao, V. Kamat, Vision-based articulated machine pose estimation for excavation monitoring and guidance, ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction, Vol. 32, Vilnius Gediminas Technical University, Department of Construction Economics & Property, 2015, p. 1, https://doi.org/10.22260/ISARC2015/0029.

[87] X. Liang, M. Xu, L. Xu, P. Liu, X. Ren, Z. Kong, J. Yang, S. Zhang, The amphihex: A novel amphibious robot with transformable leg-flipper composite propulsion mechanism, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, Vilamoura, Portugal, 2012, pp. 3667-3672, ISBN: 1467317365, https://doi.org/10.1109/IROS.2012.6386238.

[88] L. Xu, S. Zhang, N. Jiang, R. Xu, A hybrid force model to estimate the dynamics of curved legs in granular material, Journal of Terramechanics, 59 (2015) 59-70, https://doi.org/10.1016/j.jterra.2015.03.005.

[89] H.-S. Lee, K.-P. Lee, M. Park, Y. Baek, S. Lee, RFID-based real-time locating system for construction safety management, Journal of Computing in Civil Engineering, 26 (3) (2011) 366-377, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000144.

[90] J. Park, K. Kim, Y.K. Cho, Framework of automated construction-safety monitoring using cloud-enabled BIM and BLE mobile tracking sensors, Journal of Construction Engineering and Management, 143 (2) (2016) 05016019, https://doi.org/10.1061/(ASCE)CO.1943-7862.0001223.

[91] A. Carbonari, A. Giretti, B. Naticchia, A proactive system for real-time safety management in construction sites, Automation in Construction, 20 (6) (2011) 686-698, https://doi.org/10.1016/j.autcon.2011.04.019.

[92] T. Cheng, M. Venugopal, J. Teizer, P. Vela, Performance evaluation of ultra wideband technology for construction resource location tracking in harsh environments, Automation in Construction, 20 (8) (2011) 1173-1184, https://doi.org/10.1016/j.autcon.2011.05.001.

[93] J. Song, C.T. Haas, C.H. Caldas, Tracking the location of materials on construction job sites, Journal of Construction Engineering and Management, 132 (9) (2006) 911-918, https://doi.org/10.1061/(ASCE)0733-9364(2006)132:9(911).

[94] Y. Ham, K.K. Han, J.J. Lin, M. Golparvar-Fard, Visual monitoring of civil infrastructure systems via camera-equipped Unmanned Aerial Vehicles (UAVs): a review of related works, Visualization in Engineering, 4 (1) (2016) 1, https://doi.org/10.1186/s40327-015-0029-z.

[95] K.K. Han, M. Golparvar-Fard, Appearance-based material classification for monitoring of operation-level construction progress using 4D BIM and site photologs, Automation in Construction, 53 (2015) 44-57, https://doi.org/10.1016/j.autcon.2015.02.007.

[96] I. Brilakis, M.-W. Park, G. Jog, Automated vision tracking of project related entities, Advanced Engineering Informatics, 25 (4) (2011) 713-724, https://doi.org/10.1016/j.aei.2011.01.003.

[97] H.M. Khoury, V.R. Kamat, Evaluation of position tracking technologies for user localization in indoor construction environments, Automation in Construction, 18 (4) (2009) 444-457, https://doi.org/10.1016/j.autcon.2008.10.011.

[98] H. Cai, A.R. Andoh, X. Su, S. Li, A boundary condition based algorithm for locating construction site objects using RFID and GPS, Advanced Engineering Informatics, 28 (4) (2014) 455-468, https://doi.org/10.1016/j.aei.2014.07.002.

[99] Y. Fang, J. Chen, Y. Cho, P. Zhang, A point cloud-vision hybrid approach for 3D location tracking of mobile construction assets, Proceedings of the International Symposium on Automation and Robotics in Construction (ISARC), Vol. 33, IAARC, Auburn, Alabama, USA., 2016, p. 1, Retrieved from https://pdfs.semanticscholar.org/fcb0/6a7fa831497b250712f4eb3a7be10dbc092c.pdf.

[100] S. Woo, S. Jeong, E. Mok, L. Xia, C. Choi, M. Pyeon, J. Heo, Application of WiFi-based indoor positioning system for labor tracking at construction sites: A case study in Guangzhou MTR, Automation in Construction, 20 (1) (2011) 3-13, https://doi.org/10.1016/j.autcon.2010.07.009.

[101] J. Song, C.T. Haas, C.H. Caldas, A proximity-based method for locating RFID tagged objects, Advanced Engineering Informatics, 21 (4) (2007) 367-376, https://doi.org/10.1016/j.aei.2006.09.002.

[102] L.-C. Wang, Enhancing construction quality inspection and management using RFID technology, Automation in Construction, 17 (4) (2008) 467-479, https://doi.org/10.1016/j.autcon.2007.08.005.

[103] E. Ergen, B. Akinci, B. East, J. Kirby, Tracking components and maintenance history within a facility utilizing radio frequency identification technology, Journal of Computing in Civil Engineering, 21 (1) (2007) 11-20, https://doi.org/10.1061/(ASCE)0887-3801(2007)21:1(11).

[104] J. Qi, G.-P. Liu, A Robust High-Accuracy Ultrasound Indoor Positioning System Based on a Wireless Sensor Network, Sensors, 17 (11) (2017) 2554, https://doi.org/10.3390/s17112554.

[105] J. Park, Y.K. Cho, D. Martinez, A BIM and UWB integrated mobile robot navigation system for indoor position tracking applications, Journal of Construction Engineering and Project Management, 6 (2) (2016) 30-39, https://doi.org/10.6106/JCEPM.2016.6.2.030.

[106] Y. Deng, H. Hong, H. Deng, H. Luo, BIM-based Indoor Positioning Technology Using a Monocular Camera, Proceedings of the International Symposium on Automation and Robotics in Construction (ISARC), Vol. 34, IAARC, Taipei, Taiwan, 2017, Retrieved from https://pdfs.semanticscholar.org/8a36/319cd0985d061b39eef6c70f98f742a09734.pdf.

[107] Y. Fang, Y.K. Cho, S. Zhang, E. Perez, Case study of BIM and cloud–enabled real-time RFID indoor localization for construction management applications, Journal of Construction Engineering and Management, 142 (7) (2016) 05016003, https://doi.org/10.1061/(ASCE)CO.1943-7862.0001125.

[108] H. Durrant-Whyte, T. Bailey, Simultaneous localization and mapping: part I, IEEE Robotics & Automation Magazine, 13 (2) (2006) 99-110, https://doi.org/10.1109/MRA.2006.1638022.

[109] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, J.J. Leonard, Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age, IEEE Transactions on Robotics, 32 (6) (2016) 1309-1332, https://doi.org/10.1109/TRO.2016.2624754.

[110] S. Thrun, W. Burgard, D. Fox, Probabilistic robotics, MIT press, 2005, ISBN: 0262303809.

[111] W. Burgard, A. Derr, D. Fox, A.B. Cremers, Integrating global position estimation and position tracking for mobile robots: the Dynamic Markov Localization approach, Proceedings of the International Conference on Intelligent Robots and Systems (IROS), Vol. 2, IEEE, Victoria, BC, Canada, 1998, pp. 730-735, ISBN: 0780344650, https://doi.org/10.1109/IROS.1998.727279.

[112] D. Fox, W. Burgard, F. Dellaert, S. Thrun, Monte carlo localization: Efficient position estimation for mobile robots, Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence, Vol. 1999, AAAI, Orlando, Florida, USA, 1999, pp. 343-349, Retrieved from http://www.aaai.org/Papers/AAAI/1999/AAAI99-050.pdf.

[113] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? the kitti vision benchmark suite, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Providence, RI, USA, 2012, pp. 3354-3361, ISBN: 1467312282, https://doi.org/10.1109/CVPR.2012.6248074.

[114] T. Naseer, W. Burgard, C. Stachniss, Robust Visual Localization Across Seasons, IEEE Transactions on Robotics, 34 (2) (2018) 289-302, https://doi.org/10.1109/TRO.2017.2788045.

[115] S. Park, T. Schöps, M. Pollefeys, Illumination change robustness in direct visual SLAM, IEEE International Conference on Robotics and Automation (ICRA), IEEE, Singapore, Singapore, 2017, pp. 4523-4530, ISBN: 150904633X, https://doi.org/10.1109/ICRA.2017.7989525.

[116] W. Tan, H. Liu, Z. Dong, G. Zhang, H. Bao, Robust monocular SLAM in dynamic environments, IEEE International Symposium on Mixed and Augmented Reality (ISMAR), IEEE, Adelaide, SA, Australia, 2013, pp. 209-218, ISBN: 1479928690, https://doi.org/10.1109/ISMAR.2013.6671781.

[117] H. Kim, A. Handa, R. Benosman, S.-H. Ieng, A.J. Davison, Simultaneous mosaicing and tracking with an event camera, British machine vision conference (BMVC), Vol. 43, BMVA, Nottingham, UK, 2008, pp. 1-12, Retrieved from http://www.bmva.org/bmvc/2014/files/abstract066.pdf.

[118] H. Kim, S. Leutenegger, A.J. Davison, Real-time 3D reconstruction and 6-DoF tracking with an event camera, European Conference on Computer Vision (ECCV), Springer,

Amsterdam, The Netherlands, 2016, pp. 349-364, https://doi.org/10.1007/978-3-319-46466-4_21.

[119] S. Yang, Y. Song, M. Kaess, S. Scherer, Pop-up slam: Semantic monocular plane slam for low-texture environments, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, Daejeon, South Korea, 2016, pp. 1222-1229, ISBN: 1509037624, https://doi.org/10.1109/IROS.2016.7759204.

[120] P. Lichtsteiner, C. Posch, T. Delbruck, A 128 ×128 120 dB 15 μs Latency Asynchronous Temporal Contrast Vision Sensor, IEEE journal of solid-state circuits, 43 (2) (2008) 566-576, https://doi.org/10.1109/JSSC.2007.914337.

[121] Y. He, S. Chen, Advances in sensing and processing methods for three-dimensional robot vision, International Journal of Advanced Robotic Systems, 15 (2) (2018) 1729881418760623, https://doi.org/10.1177/1729881418760623.

[122] R. Mur-Artal, J.D. Tardós, Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras, IEEE Transactions on Robotics, 33 (5) (2017) 1255-1262, https://doi.org/10.1109/TRO.2017.2705103.

[123] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, ORB: An efficient alternative to SIFT or SURF, IEEE International Conference on Computer Vision (ICCV), IEEE, Barcelona, Spain, 2011, pp. 2564-2571, ISBN: 1457711028, https://doi.org/10.1109/ICCV.2011.6126544.

[124] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A.Y. Ng, ROS: an open-source robot operating system, International Conference on Robotics and Automation Open-Source Software Workshop, Vol. 3, IEEE, Kobe, Japan, 2009, p. 5, Retrieved from https://www.willowgarage.com/sites/default/files/icraoss09-ROS.pdf.

[125] D. Gálvez-López, J.D. Tardos, Bags of binary words for fast place recognition in image sequences, IEEE Transactions on Robotics, 28 (5) (2012) 1188-1197, https://doi.org/10.1109/TRO.2012.2197158.

[126] P. Mihelich, ROS package openni_launch, 2013, Accessed 10/24/2018, http://wiki.ros.org/openni_launch.

[127] P. Bovbel, T. Foote, Ros package pointcloud_to_laserscan, 2015, Accessed 10/24/2018, http://wiki.ros.org/pointcloud_to_laserscan.

[128] ROS PoseStamped Message, Accessed 10/24/2018, http://docs.ros.org/lunar/api/geometry_msgs/html/msg/PoseStamped.html.

[129] ROS LaserScan Message, Accessed 10/24/2018, http://docs.ros.org/api/sensor_msgs/html/msg/LaserScan.html.

[130] A.K. Singh, A.J. Amiri, 2D Grid Mapping and Navigation with ORB SLAM, GitHub repository, 2017, Accessed 10/24/2018, https://github.com/abhineet123/ORB_SLAM2.

[131] Alkaid-Benetnash, Map save/load for ORB SLAM2, GitHub repository, Accessed 10/24/2018, https://github.com/Alkaid-Benetnash/ORB_SLAM2.

[132] ROS OccupancyGrid Message, Accessed 10/24/2018, http://docs.ros.org/melodic/api/nav_msgs/html/msg/OccupancyGrid.html.

[133] J.E. Bresenham, Algorithm for computer control of a digital plotter, IBM Systems journal, 4 (1) (1965) 25-30, https://doi.org/10.1147/sj.41.0025.

[134] D. Hershberger, D. Gossow, J. Faust, ROS rviz, 2018, Accessed 10/24/2018, http://wiki.ros.org/rviz.

[135] K. Conley, rostopic, Accessed 1/15/2019, http://wiki.ros.org/rostopic?distro=indigo.

[136] B. Clipp, J. Lim, J.-M. Frahm, M. Pollefeys, Parallel, real-time visual SLAM, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) IEEE, Taipei, Taiwan, 2010, pp. 3961-3968, ISBN: 1424466768, http://doi.org/10.1109/IROS.2010.5653696.

[137] N. Pradhananga, J. Teizer, Automatic spatio-temporal analysis of construction site equipment operations using GPS data, Automation in Construction, 29 (2013) 107-122, https://doi.org/10.1016/j.autcon.2012.09.004.

[138] J. Sturm, N. Engelhard, F. Endres, W. Burgard, D. Cremers, A benchmark for the evaluation of RGB-D SLAM systems, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, Vilamoura, Portugal, 2012, pp. 573-580, ISBN: 1467317365, https://doi.org/10.1109/IROS.2012.6385773.

[139] R. Akhavian, A.H. Behzadan, Smartphone-based construction workers' activity recognition and classification, Automation in Construction, 71 (2016) 198-209, https://doi.org/10.1016/j.autcon.2016.08.015.

[140] F. Zafari, A. Gkelias, K. Leung, A survey of indoor localization systems and technologies, arXiv preprint arXiv:1709.01015 (2017), Retrieved from https://arxiv.org/abs/1709.01015.

[141] S. Yang, S. Scherer, Monocular Object and Plane SLAM in Structured Environments, arXiv preprint arXiv:1809.03415 (2018), Retrieved from https://arxiv.org/abs/1809.03415.

[142] P. Meadati, P. Juneja, Integration of Geotagged Photos with BIM, Accessed 10/24/2018, http://ascpro.ascweb.org/chair/paper/CPRT153002017.pdf.

[143] D. Sato, U. Oh, K. Naito, H. Takagi, K. Kitani, C. Asakawa, Navcog3: An evaluation of a smartphone-based blind indoor navigation assistant with semantic features in a large-scale environment, Proceedings of the International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS), ACM, Baltimore, Maryland, USA, 2017, pp. 270-279, ISBN: 1450349269, https://doi.org/10.1145/3132525.3132535.

[144] B.R. Mantha, C.C. Menassa, V.R. Kamat, C.R. D'Souza, Evaluation of preference-and constraint-sensitive path planning for assisted navigation in indoor building environments, Journal of Computing in Civil Engineering, 34 (1) (2019) 04019050, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000865.

[145] L. Xu, C. Feng, V.R. Kamat, C.C. Menassa, Enhancing visual SLAM with occupancy grid mapping for real-time locating applications in indoor GPS-denied environments, ASCE International Conference on Computing in Civil Engineering, ASCE, Atlanta, Georgia, 2019, pp. 344-351, https://doi.org/10.1061/9780784482438.044.

[146] J. Park, Y. Cho, K. Kim, Field construction management application through mobile BIM and location tracking technology, Proceedings of the International Symposium on Automation and Robotics in Construction (ISARC), IAARC, Auburn, AL, USA, 2016, https://doi.org/10.22260/ISARC2016/0011.

[147] Z. Ma, S. Cai, N. Mao, Q. Yang, J. Feng, P. Wang, Construction quality management based on a collaborative system using BIM and indoor positioning, Automation in Construction, 92 (2018) 35-45, https://doi.org/10.1016/j.autcon.2018.03.027.

[148] K.-P. Lee, H.-S. Lee, M. Park, H. Kim, S. Han, A real-time location-based construction labor safety management system, Journal of Civil Engineering and Management, 20 (5) (2014) 724-736, https://doi.org/10.3846/13923730.2013.802728.

[149] H. Jiang, P. Lin, Q. Fan, M. Qiang, Real-time safety risk assessment based on a real-time location system for hydropower construction sites, The Scientific World Journal, 2014 (2014), http://doi.org/10.1155/2014/235970.

[150] A. Hammad, J.H. Garrett, H.A. Karimi, Location-based computing for infrastructure field tasks, in: A. Hammad (Ed.), Telegeoinformatics: Location-based computing and services, CRC Press, Boca Raton, 2004, pp. 243-266, ISBN: 9780429210402 https://doi.org/10.1201/b12395.

[151] A. Montaser, O. Moselhi, RFID indoor location identification for construction projects, Automation in Construction, 39 (2014) 167-179, https://doi.org/10.1016/j.autcon.2013.06.012.

[152] J. Teizer, D. Lao, M. Sofer, Rapid automated monitoring of construction site activities using ultra-wideband, Proceedings of the International Symposium on Automation and Robotics in Construction (ISARC), IAARC, Kochi, India, 2007, pp. 23-28, https://doi.org/10.22260/ISARC2007/0008.

[153] M. Lu, W. Chen, X. Shen, H.-C. Lam, J. Liu, Positioning and tracking construction vehicles in highly dense urban areas and building construction sites, Automation in Construction, 16 (5) (2007) 647-656, https://doi.org/10.1016/j.autcon.2006.11.001.

[154] W.-S. Jang, M.J. Skibniewski, Embedded system for construction asset tracking combining radio and ultrasound signals, Journal of Computing in Civil Engineering, 23 (4) (2009) 221-229, https://doi.org/10.1061/(ASCE)0887-3801(2009)23:4(221).

[155] A. Mutka, D. Miklic, I. Draganjac, S. Bogdan, A low cost vision based localization system using fiducial markers, Proceedings of the World Congress, Vol. 41, IFAC, Seoul, Korea, 2008, pp. 9528-9533, ISBN: 1474-6670, https://doi.org/10.3182/20080706-5-KR-1001.01611.

[156] M.H. Lee, I.K. Park, Blur-invariant feature descriptor using multidirectional integral projection, ETRI Journal, 38 (3) (2016) 502-509, https://doi.org/10.4218/etrij.16.0115.0631.

[157] F. Tombari, A. Franchi, L. Di Stefano, BOLD features to detect texture-less objects, Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2013, pp. 1265-1272, http://doi.org/10.1109/ICCV.2013.160.

[158] L. Xu, C. Feng, V.R. Kamat, C.C. Menassa, An occupancy grid mapping enhanced visual SLAM for real-time locating applications in indoor GPS-denied environments, Automation in Construction, 104 (2019) 230-245, https://doi.org/10.1016/j.autcon.2019.04.011.

[159] S. Cebollada, L. Payá, M. Juliá, M. Holloway, O. Reinoso, Mapping and localization module in a mobile robot for insulating building crawl spaces, Automation in Construction, 87 (2018) 248-262, https://doi.org/10.1016/j.autcon.2017.11.007.

[160] Z. Zhang, Iterative point matching for registration of free-form curves and surfaces, International Journal of Computer Vision, 13 (2) (1994) 119-152, https://doi.org/10.1007/BF01427149.

[161] D. Fox, S. Thrun, W. Burgard, F. Dellaert, Particle filters for mobile robot localization, in: A. Doucet, N. de Freitas, N. Gordon (Eds.), Sequential Monte Carlo methods in practice, Springer, New York, NY, 2001, pp. 401-428, ISBN: 9781441928870 https://doi.org/10.1007/978-1-4757-3437-9_19.

[162] X. Zhang, M. Li, J.H. Lim, Y. Weng, Y.W.D. Tay, H. Pham, Q.-C. Pham, Large-scale 3D printing by a team of mobile robots, Automation in Construction, 95 (2018) 98-106, https://doi.org/10.1016/j.autcon.2018.08.004.

[163] K. Asadi, H. Ramshankar, H. Pullagurla, A. Bhandare, S. Shanbhag, P. Mehta, S. Kundu, K. Han, E. Lobaton, T. Wu, Vision-based integrated mobile robotic system for real-time applications in construction, Automation in Construction, 96 (2018) 470-482, https://doi.org/10.1016/j.autcon.2018.10.009.

[164] S. Ullman, The interpretation of structure from motion, Proceedings of the Royal Society B: Biological Sciences, 203 (1153) (1979) 405-426, https://doi.org/10.1098/rspb.1979.0006.

[165] B. Triggs, P.F. McLauchlan, R.I. Hartley, A.W. Fitzgibbon, Bundle adjustment-a modern synthesis, International Workshop on Vision Algorithms (IWVA), Springer, Corfu, Greece, 1999, pp. 298-372, https://doi.org/10.1007/3-540-44480-7_21.

[166] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, P. Sayd, Real time localization and 3d reconstruction, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 1, IEEE, New York, NY, USA, 2006, pp. 363-370, ISBN: 0769525970, https://doi.org/10.1109/CVPR.2006.236.

[167] K. Konolige, M. Agrawal, FrameSLAM: From bundle adjustment to real-time visual mapping, IEEE Transactions on Robotics, 24 (5) (2008) 1066-1077, https://doi.org/10.1109/TRO.2008.2004832.

[168] C. Mei, G. Sibley, M. Cummins, P.M. Newman, I.D. Reid, A Constant-Time Efficient Stereo SLAM System, Proceedings of the British Machine Vision Conferenc, BMVA Press, London, UK, 2009, pp. 1-11, https://doi.org/10.5244/C.23.54.

[169] J. Civera, A.J. Davison, J.M. Montiel, Inverse depth parametrization for monocular SLAM, IEEE Transactions on Robotics, 24 (5) (2008) 932-945, https://doi.org/10.1109/TRO.2008.2003276.

[170] H. Lim, J. Lim, H.J. Kim, Real-time 6-DOF monocular visual SLAM in a large-scale environment, IEEE International Conference on Robotics and Automation (ICRA), IEEE, Hong Kong, China, 2014, pp. 1532-1539, ISBN: 1479936855, https://doi.org/10.1109/ICRA.2014.6907055.

[171] H. Strasdat, J. Montiel, A.J. Davison, Scale drift-aware large scale monocular SLAM, Proceedings of Robotics: Science and Systems, Vol. 2, MIT Press, Zaragoza, Spain, 2010, p. 5, https://doi.org/10.15607/RSS.2010.VI.010.

[172] G. Grisetti, R. Kummerle, C. Stachniss, W. Burgard, A tutorial on graph-based SLAM, IEEE Intelligent Transportation Systems Magazine, 2 (4) (2010) 31-43, https://doi.org/10.1109/MITS.2010.939925.

[173] H. Abbaspour, M. Moskowitz, Basic Lie theory, World Scientific Publishing Company, 2007, ISBN: 9813101563.

[174] N. Fioraio, K. Konolige, Realtime visual and point cloud slam, Proceedings of the RGB-D Workshop on Advanced Reasoning with Depth Cameras at Robotics: Science and Systems Conference, Vol. 27, MIT Press, Los Angeles, USA, 2011, Retrieved from https://pdfs.semanticscholar.org/22af/9ca7263fa7d35e5d53c448bb41f26cc5fde8.pdf.

[175] F. Endres, J. Hess, J. Sturm, D. Cremers, W. Burgard, 3-D mapping with an RGB-D camera, IEEE Transactions on Robotics, 30 (1) (2014) 177-187, https://doi.org/10.1109/TRO.2013.2279412.

[176] T. Whelan, R.F. Salas-Moreno, B. Glocker, A.J. Davison, S. Leutenegger, ElasticFusion: Real-time dense SLAM and light source estimation, The International Journal of Robotics Research, 35 (14) (2016) 1697-1716, https://doi.org/10.1177/0278364916669237.

[177] H. Strasdat, A.J. Davison, J.M. Montiel, K. Konolige, Double window optimisation for constant time visual SLAM, IEEE International Conference on Computer Vision (ICCV), IEEE, Barcelona, Spain, 2011, pp. 2352-2359, ISBN: 1457711028, https://doi.org/10.1109/ICCV.2011.6126517.

[178] T. Naseer, M. Ruhnke, C. Stachniss, L. Spinello, W. Burgard, Robust visual SLAM across seasons, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, Hamburg, Germany, 2015, pp. 2529-2535, ISBN: 1479999946, https://doi.org/10.1109/IROS.2015.7353721.

[179] L. Lacasa, B. Luque, F. Ballesteros, J. Luque, J.C. Nuno, From time series to complex networks: The visibility graph, Proceedings of the National Academy of Sciences of the United States of America, 105 (13) (2008) 4972-4975, https://doi.org/10.1073/pnas.0709247105.

[180] D.G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision, 60 (2) (2004) 91-110, https://doi.org/10.1023/B:VISI.0000029664.99615.94.

[181] H. Bay, T. Tuytelaars, L. Van Gool, Surf: Speeded up robust features, European Conference on Computer Vision (ECCV), Springer, Graz, Austria, 2006, pp. 404-417, https://doi.org/10.1007/11744023_32.

[182] M. Calonder, V. Lepetit, C. Strecha, P. Fua, Brief: Binary robust independent elementary features, European Conference on Computer Vision (ECCV), Springer, Heraklion, Crete, Greece, 2010, pp. 778-792, https://doi.org/10.1007/978-3-642-15561-1_56.

[183] J. Chan, J. Addison Lee, Q. Kemao, BIND: binary integrated net descriptors for texture-less object recognition, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Honolulu, HI, USA, 2017, pp. 2068-2076, https://doi.org/10.1109/CVPR.2017.322.

[184] X. Han, T. Leung, Y. Jia, R. Sukthankar, A.C. Berg, Matchnet: Unifying feature and metric learning for patch-based matching, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Boston, MA, USA, 2015, pp. 3279-3286, ISBN: 1467369640, https://doi.org/10.1109/CVPR.2015.7298948.

[185] S. Zagoruyko, N. Komodakis, Learning to compare image patches via convolutional neural networks, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Boston, MA, USA, 2015, pp. 4353-4361, ISBN: 1467369640, https://doi.org/10.1109/CVPR.2015.7299064.

[186] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, F. Moreno-Noguer, Discriminative learning of deep convolutional feature point descriptors, Preceedings of the IEEE International Conference on Computer Vision (ICCV), IEEE, Santiago, Chile, 2015, pp. 118-126, ISBN: 1467383910, https://doi.org/10.1109/ICCV.2015.22.

[187] A. Shrivastava, A. Gupta, R. Girshick, Training region-based object detectors with online hard example mining, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Las Vegas, NV, USA, 2016, pp. 761-769, https://doi.org/10.1109/CVPR.2016.89.

[188] V. Balntas, E. Johns, L. Tang, K. Mikolajczyk, PN-Net: Conjoined triple deep network for learning local image descriptors, arXiv preprint arXiv:1601.05030 (2016), Retrieved from https://arxiv.org/abs/1601.05030.

[189] J. Yoon, S.J. Hwang, Combined group and exclusive sparsity for deep neural networks, Proceedings of the International Conference on Machine Learning, JMLR. org, Sydney, NSW, Australia, 2017, pp. 3958-3966, Retrieved from https://dl.acm.org/citation.cfm?id=3306090.

[190] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, C. Zhang, Learning efficient convolutional networks through network slimming, Proceedings of the IEEE International Conference on Computer Vision (ICCV), IEEE, Venice, Italy, 2017, pp. 2736-2744, https://doi.org/10.1109/ICCV.2017.298.

[191] M. Brown, G. Hua, S. Winder, Discriminative learning of local image descriptors, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 33 (1) (2011) 43-57, https://doi.org/10.1109/TPAMI.2010.54.

[192] X. Sun, X. Ren, S. Ma, H. Wang, meprop: Sparsified back propagation for accelerated deep learning with reduced overfitting, Proceedings of the International Conference on Machine Learning (ICML), JMLR. org, Sydney, NSW, Australia, 2017, pp. 3299-3308, Retrieved from https://dl.acm.org/citation.cfm?id=3306022.

[193] E. Rosten, T. Drummond, Machine learning for high-speed corner detection, European Conference on Computer Vision (ECCV), Springer, Graz, Austria, 2006, pp. 430-443, https://doi.org/10.1007/11744023_34.

[194] E. Rosten, R. Porter, T. Drummond, Faster and better: A machine learning approach to corner detection, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 32 (1) (2008) 105-119, https://doi.org/10.1109/TPAMI.2008.275.

[195] A. Bonarini, W. Burgard, G. Fontana, M. Matteucci, D.G. Sorrenti, J.D. Tardos, Rawseeds: Robotics advancement through web-publishing of sensorial and elaborated extensive datasets, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems Workshop on Benchmarks in Robotics Research, Vol. 6, IEEE, Beijing, China, 2006, p. 93, Retrieved from http://chrome.ws.dei.polimi.it/images/e/ea/Bonarini_2006_IROS.pdf.

[196] A. Handa, T. Whelan, J. McDonald, A.J. Davison, A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM, IEEE International Conference on Robotics and Automation (ICRA), IEEE, Hong Kong, China, 2014, pp. 1524-1531, ISBN: 1479936855, https://doi.org/10.1109/ICRA.2014.6907054.

[197] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, Proceedings of the ACM International Conference on Multimedia, ACM, Orlando, Florida, USA, 2014, pp. 675-678, ISBN: 1450330630, https://doi.org/10.1145/2647868.2654889.

[198] G. Yang, Z. Chen, Y. Li, Z. Su, Rapid relocation method for mobile robot based on improved ORB-SLAM2 algorithm, Remote Sensing, 11 (2) (2019) 149, https://doi.org/10.3390/rs11020149.

[199] H. Kato, K.T. Tan, D. Chai, Barcodes for mobile devices, Cambridge University Press, 2010, ISBN: 1139487515.

[200] S. Wachenfeld, S. Terlunen, X. Jiang, Robust recognition of 1-d barcodes using camera phones, International Conference on Pattern Recognition, IEEE, Tampa, FL, USA, 2008, pp. 1-4, ISBN: 1424421748, https://doi.org/10.1109/ICPR.2008.4761085.

[201] J. Pons, Drone Ready?, 2014, Accessed 8/12/2016, http://www.scanman.co.za/downloads/whitepaperdronereadyscanman.pdf.

[202] S. Wachenfeld, S. Terlunen, X. Jiang, Robust 1-D barcode recognition on camera phones and mobile product information display, Workshop of Mobile Multmedia Processing, Springer, Tampa, Florida, USA, 2010, pp. 53-69, https://doi.org/10.1007/978-3-642-12349-8_4.

[203] C. Zhang, J. Wang, S. Han, M. Yi, Z. Zhang, Automatic real-time barcode localization in complex scenes, International Conference on Image Processing, IEEE, Atlanta, GA, USA, 2006, pp. 497-500, ISBN: 1424404800, https://doi.org/10.1109/ICIP.2006.312435.

[204] Z. Wang, A. Chen, J. Li, Y. Yao, Z. Luo, 1D Barcode Region Detection Based on the Hough Transform and Support Vector Machine, International Conference on Multimedia Modeling, Springer, Miami, FL, USA, 2016, pp. 79-90, ISBN: 3319276735, https://doi.org/10.1007/978-3-319-27674-8_8.

[205] A. Zamberletti, I. Gallo, S. Albertini, L. Noce, Neural 1D barcode detection using the Hough transform, Information and Media Technologies, 10 (1) (2015) 157-165, https://doi.org/10.11185/imt.10.157.

[206] M. Katona, L.G. Nyúl, A novel method for accurate and efficient barcode detection with morphological operations, International Conference on Signal Image Technology and Internet Based Systems (SITIS), IEEE, Naples, Italy, 2012, pp. 307-314, ISBN: 1467351520, https://doi.org/10.1109/SITIS.2012.53.

[207] M. Katona, L.G. Nyúl, Efficient 1D and 2D barcode detection using mathematical morphology, International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing, Springer, Uppsala, Sweden, 2013, pp. 464-475, https://doi.org/10.1007/978-3-642-38294-9_39.

[208] X.J. Juett, X. Qi, Barcode localization using bottom-hat filter, NSF Research Experience for Undergraduates, 19 (2005).

[209] P. Bodnár, L.G. Nyúl, Improving barcode detection with combination of simple detectors, International Conference on Signal Image Technology and Internet Based Systems (SITIS), IEEE, Naples, Italy, 2012, pp. 300-306, ISBN: 1467351520, https://doi.org/10.1109/SITIS.2012.52.

[210] D.-T. Lin, M.-C. Lin, K.-Y. Huang, Real-time automatic recognition of omnidirectional multiple barcodes and dsp implementation, Machine Vision and Applications, 22 (2) (2011) 409-419.

[211] P. Bodnár, L.G. Nyúl, Barcode detection with uniform partitioning and distance transformation, IASTED International Conference on Computer Graphics and Imaging, ACTA Press, Innsbruck, Austria, 2013, pp. 48-53, https://doi.org/10.2316/P.2013.797-022.

[212] A. Zamberletti, I. Gallo, S. Albertini, Robust angle invariant 1d barcode detection, Asian Conference on Pattern Recognition, IEEE, Naha, Japan, 2013, pp. 160-164, ISBN: 0730-6512, https://doi.org/10.1109/ACPR.2013.17.

[213] C. Creusot, A. Munawar, Real-time Barcode Detection in the Wild, IEEE Winter Conference on Applications of Computer Vision, IEEE, Waikoloa, HI, USA, 2015, pp. 239-245, ISBN: 1550-5790, https://doi.org/10.1109/WACV.2015.39.

[214] L. Ott, P. Lambert, B. Ionescu, D. Coquin, Animation movie abstraction: Key frame adaptative selection based on color histogram filtering, International Conference on Image Analysis and Processing Workshops, IEEE, Modena, Italy, 2007, pp. 206-211, ISBN: 0769529216, https://doi.org/10.1109/ICIAPW.2007.12.

[215] Y. Li, C.-C.J. Kuo, A robust video scene extraction approach to movie content abstraction, International Journal of Imaging Systems and Technology, 13 (5) (2003) 236-244.

[216] M. Brown, D.G. Lowe, Automatic panoramic image stitching using invariant features, International Journal of Computer Vision, 74 (1) (2007) 59-73.

[217] D. Steedly, C. Pal, R. Szeliski, Efficiently registering video into panoramic mosaics, IEEE International Conference on Computer Vision (ICCV), Vol. 2, IEEE, Beijing, China, 2005, pp. 1300-1307, ISBN: 076952334X, https://doi.org/10.1109/ICCV.2005.86.

[218] D. Chai, F. Hock, Locating and decoding EAN-13 barcodes from images captured by digital cameras, International Conference on Information Communications & Signal Processing, IEEE, Bangkok, Thailand, 2005, pp. 1595-1599, ISBN: 0780392833, https://doi.org/10.1109/ICICS.2005.1689328.

[219] R.O. Duda, P.E. Hart, Use of the Hough transformation to detect lines and curves in pictures, Communications of the ACM, 15 (1) (1972) 11-15.

[220] C. Harris, M. Stephens, A combined corner and edge detector, Alvey Vision Conference, Vol. 15, University of Manchester, Manchester, UK, 1988, p. 50.

[221] A. Zamberletti, I. Gallo, M. Carullo, E. Binaghi, Neural Image Restoration for Decoding 1-D Barcodes using Common Camera Phones, International Conference on Computer Vision Theory and Applications, Vol. 1, INSTICC Press, Angers, France, 2010, pp. 5-11.

[222] DJI, Phantom 4, 2017. http://www.dji.com/phantom-4.

[223] S. Kerber, J.A. Milke, Using FDS to simulate smoke layer interface height in a simple atrium, Fire technology, 43 (1) (2007) 45-75, https://doi.org/10.1007/s10694-007-0007-7.

[224] R. Zobel, P. Tandayya, H. Duerrast, Modelling and simulation of the impact of tsunami waves at beaches and coastlines for disaster reduction in Thailand, International Journal of Simulation, 7 (4-5) (2006) 40-50, http://ijssst.info/Vol-07/No-4-5/Paper6.pdf.

[225] B.M. Ginting, R.-P. Mundani, Parallel Flood Simulations for Wet–Dry Problems Using Dynamic Load Balancing Concept, Journal of Computing in Civil Engineering, 33 (3) (2019) 04019013, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000823.

[226] NRC, National earthquake resilience: Research, implementation, and outreach, National Academies Press, 2011, ISBN: 0309186773.

[227] NRC, Grand challenges in earthquake engineering research: A community workshop report, National Academies Press, 2011, ISBN: 0309214521.

[228] NIST, Community resilience planning guide for buildings and infrastructure systems, Volume I, National Institute of Standards and Technology (2016), http://dx.doi.org/10.6028/NIST.SP.1190v1.

[229] NIST, Community resilience planning guide for buildings and infrastructure systems, Volume II, National Institute of Standards and Technology (2016), http://dx.doi.org/10.6028/NIST.SP.1190v2.

[230] M. Koliou, J.W. van de Lindt, T.P. McAllister, B.R. Ellingwood, M. Dillard, H. Cutler, State of the research in community resilience: progress and challenges, Sustainable and Resilient Infrastructure (2017) 1-21, https://doi.org/10.1080/23789689.2017.1418547.

[231] D. Mitsova, Integrative Interdisciplinary Approaches to Critical Infrastructure Interdependency Analysis, Risk Analysis (2018), https://doi.org/10.1111/risa.13129.

[232] ROS.org, ROS-Introduction, 2018, Accessed 12/9/2018, http://wiki.ros.org/ROS/Introduction.

[233] A.S. Huang, E. Olson, D.C. Moore, LCM: Lightweight communications and marshalling, IEEE/RSJ international conference on Intelligent robots and systems (IROS), IEEE, Taipei, Taiwan, 2010, pp. 4057-4062, ISBN: 1424466768, https://doi.org/10.1109/IROS.2010.5649358.

[234] M. Hori, T. Ichimura, Current state of integrated earthquake simulation for earthquake hazard and disaster, Journal of Seismology, 12 (2) (2008) 307-321, https://doi.org/10.1007/s10950-007-9083-x.

[235] M. Hori, Introduction to computational earthquake engineering, World Scientific, 2011, ISBN: 1848163991.

[236] A. Sahin, R. Sisman, A. Askan, M. Hori, Development of integrated earthquake simulation system for Istanbul, Earth, Planets and Space, 68 (1) (2016) 115, https://doi.org/10.1186/s40623-016-0497-y.

[237] S.B. Miles, S.E. Chang, Modeling community recovery from earthquakes, Earthquake Spectra, 22 (2) (2006) 439-458, https://doi.org/10.1193/1.2192847.

[238] S. Miles, S. Chang, ResilUS--Modeling Community Capital Loss and Recovery, World Conference on Earthquake Engineering, IAEE, Beijing, China, 2008, Retrieved from http://www.iitk.ac.in/nicee/wcee/article/14_09-01-0095.PDF.

[239] M. Mandiak, P. Shah, Y. Kim, T. Kesavadas, Development of an integrated GUI framework for post-disaster data fusion visualization, International Conference on Information Fusion, Vol. 2, IEEE, Philadelphia, PA, USA, 2005, p. 7 pp., ISBN: 0780392868, https://doi.org/10.1109/ICIF.2005.1591984.

[240] C. Nan, I. Eusgeld, Adopting HLA standard for interdependency study, Reliability Engineering & System Safety, 96 (1) (2011) 149-159, https://doi.org/10.1016/j.ress.2010.08.002.

[241] V.R. Kamat, J.C. Martinez, Comparison of simulation-driven construction operations visualization and 4D CAD, Proceedings of the Winter Simulation Conference, Vol. 2, IEEE, San Diego, CA, USA, 2002, pp. 1765-1770, ISBN: 0780376145, https://doi.org/10.1109/WSC.2002.1166463.

[242] A.H. Behzadan, V.R. Kamat, Integrated information modeling and visual simulation of engineering operations using dynamic augmented reality scene graphs, Journal of Information Technology in Construction (ITcon), 16 (17) (2011) 259-278, https://www.itcon.org/paper/2011/17.

[243] S. Dong, V.R. Kamat, Robust mobile computing framework for visualization of simulated processes in augmented reality, Proceedings of the Winter Simulation Conference, IEEE, Baltimore, MD, USA, 2010, pp. 3111-3122, ISBN: 1424498643, http://doi.org/10.1109/WSC.2010.5679004.

[244] E. Azar, C.C. Menassa, A conceptual framework to energy estimation in buildings using agent based modeling, Proceedings of the Winter Simulation Conference, IEEE, Baltimore, MD, USA, 2010, pp. 3145-3156, ISBN: 1424498643, https://doi.org/10.1109/WSC.2010.5679007.

[245] A.H. Behzadan, C.C. Menassa, A.R. Pradhan, Enabling real time simulation of architecture, engineering, construction, and facility management (AEC/FM) systems: a review of formalism, model architecture, and data representation, Journal of Information Technology in Construction, 20 (2015) 1-23, https://www.itcon.org/paper/2015/1.

[246] Y. Yong, J. Yicheng, Marine simulator and distributed interactive simulation technology, Computer Simulation, 17 (6) (2000) 66-68, http://en.cnki.com.cn/Article_en/CJFDTotal-JSJZ200006021.htm.

[247] L. Arguello, J. Miró, Distributed interactive simulation for space projects, ESA bulletin (102) (2000) 125-130, http://www.esa.int/esapub/bulletin/bullet102/Arguello102.pdf.

[248] P.T. Grogan, O.L. De Weck, Infrastructure system simulation interoperability using the High-Level Architecture, IEEE Systems Journal, 12 (1) (2015) 103-114, https://doi.org/10.1109/JSYST.2015.2457433.

[249] K. DAI, W. ZHAO, H.-l. ZHANG, G.-q. SHI, Y.-l. ZHANG, TENA Based Implementation Technology for Virtual Test, Journal of System Simulation, 5 (2011), Retrieved from http://en.cnki.com.cn/Article_en/CJFDTotal-XTFZ201105006.htm.

[250] IoTivity, IoTivity, Accessed 12/18/2018, https://www.iotivity.org.

[251] D.C. Miller, J.A. Thorpe, SIMNET: The advent of simulator networking, Proceedings of the IEEE, 83 (8) (1995) 1114-1123, https://doi.org/10.1109/5.400452.

[252] DIS Steering Committee, IEEE standard for distributed interactive simulation-application protocols, IEEE Standard, Vol. 1278.1a-1998, 1998, pp. 1-52, https://standards.ieee.org/standard/1278_1a-1998.html.

[253] D. McGregor, D. Brutzman, J. Grant, Open-DIS: An open source implementation of the DIS protocol for C++ and Java, Simulation Interoperability Workshop (SIW) of Simulation Interoperability Standards Organization (SISO), paper 08F-SIW-051, Orlando, Florida, 2008, Retrieved from http://open-dis.org/SisoDISPaper.pdf.

[254] VT MAK, VR-Link, Accessed 12/18/2018, https://www.mak.com/products/link/vr-link#vr-link-supports-dis.

[255] J.W. Hollenbach, Inconsistency, Neglect, and Confusion: A Historical Review of DoD Distributed Simulation Architecture Policies, Proceedings of the Spring Simulation Interoperability Workshop, SISO, San Diego, CA, USA, 2009, pp. 23-27, Retrieved from https://www.sisostds.org/DesktopModules/Bring2mind/DMX/Download.aspx?Command=Core_Download&EntryId=28991&PortalId=0&TabId=105.

[256] J.S. Dahmann, F. Kuhl, R. Weatherly, Standards for simulation: As simple as possible but not simpler the high level architecture for simulation, Simulation, 71 (6) (1998) 378-387, https://doi.org/10.1177/003754979807100603.

[257] HLA Working Group, IEEE standard for modeling and simulation (M&S) high level architecture (HLA)-framework and rules, IEEE Standard, 2000, pp. 1516-2000, https://standards.ieee.org/standard/1516-2010.html.

[258] I. Eusgeld, C. Nan, Creating a simulation environment for critical infrastructure interdependencies study, IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), IEEE, Hong Kong, China, 2009, pp. 2104-2108, ISBN: 1424448697, https://doi.org/10.1109/IEEM.2009.5373155.

[259] I. Eusgeld, C. Nan, S. Dietz, "System-of-systems" approach for interdependent critical infrastructures, Reliability Engineering & System Safety, 96 (6) (2011) 679-686, https://doi.org/10.1016/j.ress.2010.12.010.

[260] S. Hwang, R. Starbuck, S. Lee, M. Choi, S. Lee, M. Park, High level architecture (HLA) compliant distributed simulation platform for disaster preparedness and response in facility management, Proceedings of the Winter Simulation Conference, IEEE, Washington, DC, USA, 2016, pp. 3365-3374, ISBN: 1509044841, https://doi.org/10.1109/WSC.2016.7822367.

[261] ONERA, CERTI, Accessed 12/23/2018, http://savannah.nongnu.org/projects/certi/.

[262] Calytrix Technologies, Portico, Accessed 12/23/2018, http://porticoproject.org.

[263] MÄK Technologies, MÄK High Performance RTI, Accessed 12/23/2018, http://www.mak.com/products/link/mak-rti.

[264] Pitch Technologies, Pitch pRTI, Accessed 12/23/2018, http://pitchtechnologies.com/products/prti/.

[265] H. Behner, B. Lofstrand, The New HLA Certification Process in NATO, Symposium on M&S Technologies and Standards for Enabling Alliance Interoperability and Pervasive M&S Applications, NATO Modelling and Simulation Group (NMSG) - M&S Coordination Office, Lisbon, 2017, https://www.researchgate.net/publication/325967907_The_New_HLA_Certification_Process_in_NATO.

[266] R. Cozby, Foundation initiative 2010, ARMY TEST AND EVALUATION COMMAND ABERDEEN PROVING GROUND MD, 1998, https://apps.dtic.mil/dtic/tr/fulltext/u2/a355375.pdf.

[267] OMG, "Data Distribution Service (DDS) Version 1.0, 2004, Accessed 12/19/2018, https://www.omg.org/spec/DDS/1.0/About-DDS/.

[268] Real-time Innovations, RTI connext DDS professional, 2014, Accessed 12/19/2018, https://www.rti.com/products/connext-dds-professional.

[269] OMG, Open DDS, 2018, Accessed 12/19/2018, http://opendds.org/.

[270] B. Lazarov, G. Kirov, P. Zlateva, D. Velev, Network-Centric Operations for Crisis Management Due to Natural Disasters, International Journal of Innovation, Management and Technology, 6 (4) (2015) 252, https://doi.org/10.7763/IJIMT.2015.V6.611.

[271] P.P. Ray, M. Mukherjee, L. Shu, Internet of things for disaster management: State-of-the-art and prospects, IEEE Access, 5 (2017) 18818-18835, https://doi.org/10.1109/ACCESS.2017.2752174.

[272] B.R. Mantha, C.C. Menassa, V.R. Kamat, A taxonomy of data types and data collection methods for building energy monitoring and performance simulation, Advances in Building Energy Research, 10 (2) (2016) 263-293, https://doi.org/10.1080/17512549.2015.1103665.

[273] A. Reinhorn, G. Manolis, C. Wen, Active control of inelastic structures, Journal of engineering mechanics, 113 (3) (1987) 315-333.