

UC Irvine

UC Irvine Electronic Theses and Dissertations

Title

Encoded Visible Light Based Indoor Localization and Navigation Technology for Mobile Autonomous Robots

Permalink

<https://escholarship.org/uc/item/3gk3h5np>

Author

Ma, Shang

Publication Date

2017

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA,
IRVINE

Encoded Visible Light Based Indoor Localization and Navigation Technology
for Mobile Autonomous Robots

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Computer Engineering

By

Shang Ma

Dissertation Committee:
Professor Phillip Sheu, Chair
Professor Mohammad Abdullah Al Faruque
Dr Qiong Liu (Fuji Xerox Palo Alto Laboratory)

2017

*To my parents,
whose love and support
are without measure*

*and to my wife
Xing,
who is a source of joy.*

TABLE OF CONTENTS

	Page
LIST OF FIGURES	vi
LIST OF TABLES	viii
ACKNOWLEDGMENTS	ix
CURRICULUM VITAE	x
ABSTRACT OF THE DISSERTATION	xii
Chapter 1 Introduction	1
1.1 Location Estimation Requirement.....	2
1.2 Modeling Light in the Environment.....	3
1.3 Autonomous Robot Navigation with High Reliability.....	5
1.4 Multi-robot Coordination.....	6
1.5 Contribution.....	7
1.6 Dissertation Structure.....	8
Chapter 2 Visible Light Channel Modeling	10
2.1 General Optical Channel Model.....	10
2.2 Nondirected LOS Propagation Model.....	14
2.3 Nondirected Non-LOS Single-Reflection Propagation Model.....	18
2.3.1 Optical Power Distribution.....	18
2.3.2 Channel Impulse Response.....	25
2.3.3 Channel Delay Spread.....	26
2.4 OOK Modulation.....	31
2.5 Modeling Background Noise.....	34
Chapter 3 Coded Light Based Indoor Localization	40
3.1 Introduction.....	40
3.2 Projector Encoding.....	41
3.3 Location Discovery.....	43
3.3.1 Gray-coded Position Encoding.....	43
3.3.2 Embedded Synchronization Signal with High Reliability.....	44
3.3.3 Signal Detection & Position Reconstruction.....	45
3.4 Projector-Light Sensor Homography.....	46
3.5 System Evaluation.....	48
3.5.1 Localization Accuracy.....	49
3.5.2 Localization Latency.....	52

3.5.3	Localization Refresh Rate.....	54
3.5.4	Ambient Light Robustness.....	57
3.6	Comparison with Other Localization Systems.....	59
Chapter 4	Robust Indoor Navigation System for Autonomous Robots	64
4.1	Introduction.....	64
4.1.1	Topological Map versus Metric Map.....	64
4.1.2	Active Landmarks versus Passive Landmarks.....	67
4.2	Our approach.....	70
4.2.1	Topological map for global navigation.....	70
4.2.2	Photo-diode Enabled Active Landmarks	72
4.2.3	Landmark-defined Zone	73
4.2.4	PID Controller for Local Trajectory Smoothing.....	74
4.3	Robot Navigation Robustness Evaluation.....	76
4.3.1	Landmark Position Estimation Accuracy	77
4.3.2	Navigation Accuracy for Long Paths.....	78
4.3.3	Comparison with Existing Work	80
4.4	Discussion	81
4.4.1	Computation Efficiency	81
4.4.2	Cold Start Problem	82
4.4.3	Semantic Landmark Placement.....	82
4.4.4	Minimum Mutual Communication among Landmarks.....	82
4.4.5	Obstacle Avoidance	83
4.4.6	Sensor Deployment Strategy.....	83
4.4.7	Multi-robot Coordination and Navigation	84
Chapter 5	Multi-robot Motion Coordination.....	85
5.1	Introduction.....	85
5.2	Related work	86
5.2.1	Continuous Space Representation.....	86
5.2.2	Graph Representation	88
5.3	Our Contribution	89
5.4	Assumptions & Definitions.....	92
5.4.1	Assumptions	92
5.4.2	Definitions	93
5.5	Acyclic Graph	94
5.5.1	Chain.....	94
5.5.2	n -Star	94
5.5.3	Extended n -Star	98
5.5.4	Reachable Territory & General Acyclic Graph	101
5.5.5	Example Algorithm.....	110

5.5.6	Robot Capacity on Acyclic Graph	112
5.6	Graph with Cycles.....	115
5.6.1	Unicyclic Graph.....	116
5.6.2	(m, n) Tadpole Graph.....	118
5.6.3	Extended Tadpole Graph	119
5.6.4	General Cyclic Graph	122
5.6.5	Example Algorithm.....	123
5.6.6	Robot Capacity on Cyclic Graph	126
Chapter 6	Conclusion & Future Work.....	127
Bibliography	131

LIST OF FIGURES

Figure 1. An example scenario: an office with level floors and ceiling.....	6
Figure 2 Classification of visible light links.....	11
Figure 3 Transmission and reception in a visible light link with IM/DD.....	12
Figure 4 Modeling visible link as a baseband linear, time-invariant system.....	12
Figure 5 Propagation model of nondirected line-of-sight visible light link	16
Figure 6 Distribution of received power over nondirected LOS link with one LED.	17
Figure 7 Distribution of received power over nondirected LOS link with four LEDs.	18
Figure 8 Compare power spectral distribution to the measured spectral reflectance of various texture.....	19
Figure 9 Propagation model of reflected link.	21
Figure 10 Distribution of received power from first-order reflection with one LED.....	22
Figure 11 Distribution of received power from first-order reflection with four LEDs.....	23
Figure 12 Optical power distribution for single LED.....	24
Figure 13 Optical power distribution for four LEDs.	24
Figure 14 Impulse response of the reflective links with a LOS path.....	25
Figure 15 Impulse response of the reflective links without a LOS path.....	26
Figure 16 Mean delay spread for a single LED.	28
Figure 17 RMS delay spread for a single LED.	28
Figure 18 Maximum data rate distribution for a single LED.....	29
Figure 19 Mean delay spread for the case of four transmitters.....	30
Figure 20 RMS delay spread for the case of four transmitters.....	30
Figure 21 Maximum data rate distribution for the case of four transmitters.....	31
Figure 22 Transmitted waveforms for OOK: (a) NRZ and (b) RZ ($\eta=0.5$).	32
Figure 23 Power spectral densities of OOK-NRZ and OOK-RZ.....	33
Figure 24 Block diagram of OOK based optical system with AWGN.....	35
Figure 25 BER simulation for OOK-NRZ channel.	38
Figure 26 An example of gray code images for an 8×8 grid.	44
Figure 27 The DLP projector and a sensor module in Pilot.....	49
Figure 28 A test arena for calculating the homography.....	50
Figure 29 Localization errors of all 56 points.....	50
Figure 30 The empirical cumulative distribution function of localization error for 56 discrete points inside the projection.....	51
Figure 31 Experiment setup to evaluate system latency of Pilot.	52
Figure 32 Experiment setup to evaluate localization refresh rate in Pilot.....	54
Figure 33 Tracking refresh rate of Pilot at different movement speeds.	56
Figure 34 Experiment setup to evaluate system robustness.....	58
Figure 35. An Android application for robot navigation.	66
Figure 36. Active landmark for decoding projected light signal.....	69
Figure 37. Experiment setup for robot navigation.	70
Figure 38. The generated paths to navigate a robot.....	72
Figure 39. An illustration of the robot navigation with encoded projection.	74
Figure 40. Landmark position estimation error for 18 points.....	78

Figure 41. Experiment result of one navigation test from Landmark 7 to 25.....	79
Figure 42. Experiment result of one navigation test from Landmark 25 to 7.....	79
Figure 43. The empirical cumulative distribution function for the navigation error of the proposed system from all 20 tests.	80
Figure 44. The illustration of the subgraph structure, Chain.....	94
Figure 45. The illustration of the subgraph structure, Star.....	94
Figure 46. The illustration of the subgraph structure, Extended Star.....	98
Figure 47. The illustration <i>Origin, Stalk, Internal vertices, and Leaves</i>	100
Figure 48. Four reachable territories on an example tree.....	102
Figure 49. An example of close reachable territories.	105
Figure 50. An example of a stalk connecting a leaf and a junction.	114
Figure 51. An example of a stalk with junctions at both ends.	114
Figure 52. An example of a unicyclic graph.	116
Figure 53. An example of a pan graph.....	117
Figure 54. An example of an m - n tadpole graph.	118
Figure 55. An example of an extended tadpole graph.	119
Figure 56 An example of a stalk on cyclic graph.....	120
Figure 57 Another example of a stalk on cyclic graph.....	120

LIST OF TABLES

Table 1 System parameters for simulation.....	17
Table 2 Data Transmission Latency for Pilot.....	53
Table 3 Localization Accuracy under Different Conditions.....	58
Table 4 Comparisons with Representative Systems.....	63
Table 5 Comparisons with Representative Navigation Systems.....	80
Table 6 Potential Coordination Tasks for Robots on a Star.....	96
Table 7 Reachable Territory with its Vertices.....	103

ACKNOWLEDGMENTS

To my research advisor, Professor Phillip Sheu, for his tremendous guidance and support throughout my graduate school life.

To my thesis committee, Professor Mohammad Abdullah Al Faruque and Dr. Qiong Liu from Fuji Xerox Palo Alto Laboratory, for helping me shape this thesis.

To the members of the Coded Light project, especially Dr. Henry Tang, Mingming Fan, and Zhanqing Feng, for their collaboration and feedback.

To all the members at Fuji Xerox Palo Alto Laboratory, especially Dr. Lynn Wilcox, Dr. Don Kimber, Dr. Hideto Oda, and Dr. Chelhwon Kim, for their continuous support for many years.

CURRICULUM VITAE

Shang Ma

- 2009 B.A. in Business Administration, Zhejiang University
- 2012 M.S. in Computer Science, Tsinghua University
- 2013-14 Research Assistant, Northeastern University
- 2017 Ph.D. in Electrical Engineering and Computer Science,
University of California, Irvine

PUBLICATIONS

- S. Ma, Q. Liu, P. Sheu, "Foglight: Visible Light-enabled Indoor Localization System for Low-power IoT Devices," IEEE Internet of Things Journal. *To appear*. (Impact factor: 7.596)
- S. Ma, Q. Liu, and P. Sheu, "Robust Indoor Autonomous Navigation of Mobile Robot Using Encoded Projection," Intelligent Service Robotics, 2017. *Under review*.
- S. Ma, Q. Liu, P. Sheu, "Projected Visible Light for 3D Finger Tracking and Device Augmentation on Everyday Objects," IEEE Transactions on Multi-Scale Computing Systems. *Under review*.
- S. Ma, C. Kim, Q. Liu, P. Sheu, "Lift: Using Projected Coded Light for Finger Tracking and Device Augmentation," IEEE International Conference on Pervasive Computing and Communications (PerCom), Kona, Big Island, Hawaii, USA, 2017. (Acceptance rate: 16.5%)
- X. Hao, G. Zhang, S. Ma, "Deep Learning," International Journal of Semantic Computing, Vol 10(03), 417-439, 2016.
- S. Ma, Q. Liu, "Semantic Localization," Encyclopedia with Semantic Computing, 2016.
- S. Ma, Q. Liu, P. Sheu, "On hearing your position through light for mobile robot indoor navigation," IEEE Multimedia & Expo Workshops (ICMEW), Seattle, Washington, USA, 2016.
- M. Fan, Q. Liu, S. Ma, P. Chiu, "Smart Toy Car Localization and Navigation using Projected Light," IEEE International Symposium on Multimedia (ISM), Miami, Florida, USA, 2015.
- S. Ma, Q. Liu, H. Tang, "An Overview of Location Semantics Technologies and Applications," International Journal of Semantic Computing Vol 9 (03), 373-393, 2015.

S. Ma, J. Xia, M. Fan, “New Touch Interface Designs for Thumb Interaction,” The 39th Graphics Interface Conference (GI), Regina, Saskatchewan, Canada, 2013. *Poster*.

N. Correll, N. Farrow, S. Ma, “HoneyComb: A platform for computational robotic materials,” The 7th International Conference on Tangible, Embedded and Embodied Interaction, Barcelona, Spain, 2013.

S. Ma, H. Hosseinmardi, N. Farrow, R. Han, N. Correll, “Establishing Multi-Cast Groups in Computational Robotic Materials,” IEEE International Conference on Cyber, Physical and Social Computing, Besancon, France, 2012.

S. Ma, Y. Shi, “A Scalable Passive RFID-Based Multi-user Indoor Location System,” The 7th International Conference on Wireless Communication, Networking and Mobile Computing, Wuhan, China, 2011.

Y. Shi, Y. Suo, S. Ma, Y. Shi, “PuppetAnimator: A Performative Interface for Experiencing Shadow Play,” The 4th International Conference in Computer-Human Interactions, Guadeloupe, France, 2011.

ABSTRACT OF THE DISSERTATION

Encoded Visible Light Based Indoor Localization and Navigation Technology
for Mobile Autonomous Robots

By

Shang Ma

Doctor of Philosophy in Computer Engineering

University of California, Irvine, 2017

Professor Phillip Sheu, Chair

Reliable autonomous robot navigation is critically important when a mobile service robot is used in a domestic or industry environment. To guarantee an accurate and reliable navigation performance, many practical applications leverage artificial landmarks. In this dissertation, we introduce a visible light-enabled indoor localization system that relies on unique spatial encoding produced when the mechanical mirrors inside a projector are flipped based on gray-coded binary images. Then we present the first encoded projection based navigation system for indoor autonomous robots. At the same time, off-the-shelf photodiodes are used as landmarks and strategically placed on the ceiling to form a topological representation of the environment. The proposed technology makes use of a topological map for global path planning and encoded projection based location discovery for local smooth navigation. With this combination, the proposed scheme is efficient, scalable, and can be applied to mobile robots with limited computation resources in a large-scale workplace. Through experiments within real-world environment, we demonstrate that our proposed approach can locate a target device with an average

accuracy of 1.7 millimeters and allows robust autonomous robot navigation in practice with a navigation error of 18.5 millimeters. Furthermore, for dealing with the generally NP-hard multi-robot coordination problem, we discretize the environment into a topological graph and take advantage of its subgraph structure to guarantee the completeness and reliability for real-world multi-robot applications.

Chapter 1 Introduction

Mobile autonomous robots can be expected to perform various services in a plethora of environments, for example, office mail delivery, medical equipment transportation inside hospitals, and guiding in public supermarkets, museums, and big exhibition centers. Although we have observed substantial progress on the autonomous navigation of mobile robots, it remains one of the fundamental problems in a large amount of literature. Unsurprisingly, robot navigation system designers must address multiple, possibly competing, requirements by balancing tradeoffs in terms of computation, communication, accuracy, and reliability to satisfy design objectives. For instance, system designers might strive to minimize the engineering cost of modifying the environment and the computation requirement of the overall system, but at the same time attempt to maximize the reliability and repeatability of the resulting navigation system.

The basic requirements for the autonomous navigation of a mobile robot are environmental recognition, path planning, driving control, and location estimation/correction capabilities. The location estimation and correction capabilities are practically indispensable for the autonomous mobile robot to execute the given tasks efficiently. Even though indoor localization technology has been studied for decades, it still has problems for helping robots in the above simple scenario. For example, classical positioning technology based on wireless radio signals is still shy of its potential to provide truly ubiquitous and real-time location for many applications [1, 2]. Although radio technologies offer a great opportunity to measure internode distance and bearing or to perform simultaneous angle-distance measurements by utilizing heterogeneous radio

interfaces, wireless localization systems are still inaccurate and unreliable in indoor environments. It is primarily caused by three major challenges, namely, *multipath reflections* [3], *environmental dynamics* [4], and *device heterogeneity* [5]. With the available accuracy from this type of technology, we obviously cannot rely on the system to navigate robots inside workplace where human and robots coexist and collaborate.

1.1 Location Estimation Requirement

As a matter of fact, researchers and developers often face more challenges while developing indoor localization systems. Many mobile devices are battery-powered and have limited computation power and storage capability, whereas many practical applications have high demands on accurate and real-time location information and are inherently intolerant to small localization errors and system latency. These needs motivate researchers and developers to offer location-based services that deliver better delay performance and user experience. Simply because if a localization scheme incorrectly locates robots in a given scenario, the purpose of localization might be entirely defeated. Consequently, we prefer the system to be high-accuracy, low-latency, and have only minimum computation overhead. This pushes us into using wireless signals, such as visible light [6] or Ultra-wideband (UWB) [7] signals, rather than, for example, WiFi [8] and Bluetooth [9]. Even in the homes and workplaces where we could deploy more and more computing infrastructure, the cost, high power consumption, and low reliability for traditional radio signals is often prohibitive.

In this dissertation, we propose a new indoor localization technique that allows battery-powered devices to locate themselves with sub-centimeter accuracy, low signal processing overhead (*thus low latency*), and minimum power consumption. This technique operates by projecting a sequence of gray-coded binary images into the environment, which uniquely encode each pixel inside the projection area with a pair of coordinates $\langle x, y \rangle$. A given mobile device can locate itself by collecting and decoding a sequence of light intensity at a given place using an off-the-shelf light sensor. This sensor is simple and inexpensive, which allows it to be easily incorporated into many existing devices/applications, thus enabling them to recognize their own position inside the projection area. Additionally, the procedure required to restore a device's position from sensed light is fast (11.75 milliseconds) and power efficient, so they can be applied to battery-powered devices and deployed for large-scale applications over a long period of time. Further, the proposed system realizes fine-grained localization with sub-centimeter precision. This opens a huge field of new applications where high-accuracy location information is demanded. Finally, our experiments show the proposed approach is feasible. In a controlled lab study, the proposed positioning system can offer real-time localization services with 1.7 millimeters of accuracy across a variety of environments.

1.2 Modeling Light in the Environment

Overall, a positioning system based on white lighting LEDs, the main component inside the projector we use, has many advantages. Lights are typically placed at regular intervals on the ceiling, and have line of sight (LOS) to most positions within a room. Multipath transmission presents much less of a challenge in optical systems than RF, because if

reflected components are present, they typically have much lower power than the LOS path [10]. Further, as optical signals cannot pass through opaque obstacles, LED based positioning is free of interference from other systems working in different rooms. Installing a white LED based positioning system is potentially almost as “simple as changing a light bulb.” The advantages of positioning with LEDs also include low-cost frontends, simultaneous illumination and localization, and environmental friendliness. Because of the many advantages of LEDs, there are increasing number of papers published describing a variety of schemes for indoor positioning using LEDs [11]. The ranging techniques used in these papers include received signal strength (RSS) [12-15], time difference of arrival (TDOA) [16], and phase difference of arrival (PDOA) [17]. Most of the papers published so far have presented either practical evaluation or simulation localization results for very specific idealized scenarios in which parameters such as the optical power transmitted and the radiation pattern of each LED are precisely manually crafted. For the systems using TDOA or PDOA perfect synchronization has been assumed. For these idealized cases, very low estimation errors have been reported [12, 13, 16].

While the research so far on visible LED-based systems has shown that accurate localization can be possible, little has published on the channel characterization. The determination of these characteristics will allow optimization of the parameters governing optical positioning systems. Many studies on channel characteristics for infrared (IR) have been made since the first studies by Gfeller and Bapst in 1979 [18]. A computer simulation using a recursive method is present in [19]. Characterization of wireless channel by measuring in various room is reported in [20]. In this paper, we would like to perform one

of the first theoretical analysis of channel characteristics of visible light. We will report a series of simulation processes and results developed based on MATLAB and Simulink to present a comprehensive channel modeling and characterization study for both line-of-sight (LOS) and non-line-of-sight (non-LOS) paths. We consider various scenarios with different transmitter specifications (i.e. single versus multiple light source) and receiver specifications (location and rotation). For each setup, we obtain its channel impulse responses (CIRs), and present a channel characterization, in which various channel parameters are presented, including mean excess delay, root mean square (RMS) delay spread, maximum bit rate, and coherence bandwidth through simulation.

1.3 Autonomous Robot Navigation with High Reliability

Observing the promising resolution of visible light based indoor localization technology, we advocate an encoded projection based topological navigation system for an indoor autonomous service robot by presenting two tradeoffs we have investigated when building a practical delivery robot in a real-world office setting:

- 1) Should we use a continuous metric map or a semantic topological map?
- 2) For landmark-based navigation system, are active landmarks more suitable than passive landmarks for scenarios we are interested in, such as office setting, and hospital where human and robots coexist?

Note that we focus our attention on the case where the mobile robot moves in a common workplace with level floors and ceilings. We also assume that the robot has a clear view of

the ceiling. Figure 1 provides an example setting, where a robot with a projector on top is facing upwards and projecting encoded patterns onto the ceiling.

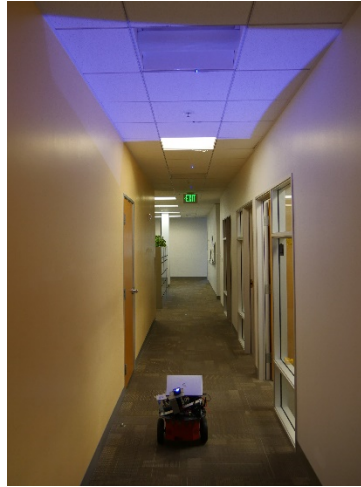


Figure 1. An example scenario: an office with level floors and ceiling.

1.4 Multi-robot Coordination

Further, indoor autonomous robot navigation is not just a theoretical exercise. It has many important real-world applications as well. It often manifests itself in the scenarios, such as office, hospital, warehouse, and more. Meanwhile, for many real-world applications, multiple robots are employed to accomplish design goals more efficiently. This, therefore, brings up another critical problems: how multiple robots can be coordinated to achieve their goals without causing any problems?

Although many algorithms have been proposed to solve this problem, they all have their own disadvantages. Some of them offer optimal solutions but do not scale up to large groups of robots. While some algorithm cannot guarantee completeness. Worse, we also

find that the following two problems, which are crucial to deploy real-life robot navigation system, have not been fully studied in prior research:

- 1) For a given representation of an environment, normally a floorplan, how should we design the potential paths so that there will be no mutual collision between multiple robots?
- 2) What is the maximum number of robots that can be operated inside the given environment without them colliding with each other?

In this work, we propose a general algorithmic framework for solving multi-robot coordination problems on graph representation. We demonstrate a new type of abstract representation for subgraph structures and generalize these subgraph topologies to more common cases step by step. We advocate that decomposing a topological map into subgraphs is a more intuitive and efficient way for path planners to perform coordination between multiple robots.

1.5 Contribution

This work makes three main contributions. Firstly, the key contribution is to advocate an encoded projection based localization system with the focus on low-power sensing devices featuring limited power consumption, communication capability, and computation resources. To my best knowledge, this is the first time that projection-based encoded light signal is exploited as an indoor localization method to obtain fine-grained, real-time location data for such devices. Secondly, this work introduces the encoded projection-based location discovery to the field of the indoor mobile service robot navigation. Here,

we reverse the traditional vision-based approach: we use a projector instead of a camera to encode the navigation space around the robot and we use active photodiode landmarks to decode the received light signal instead of retro-reflective tags or active LEDs. We also implemented a navigation system in a real-world scenario by combining the introduced landmark design and a topological representation of the environment. An Android application has also been developed, allowing users to navigate a robot to different goals by touch interaction on a tablet or using voice recognition. Users can also monitor the landmark-level location and moving direction of the robot in real-time. Finally, a general algorithmic framework is proposed to help deploy robot navigation system in real life by discretizing the environment into topological graphs and exploring particular subgraph structures with which a certain amount of robots can be freely manipulated without mutual collision.

1.6 Dissertation Structure

The rest of this dissertation is organized as follows: In Chapter 2, we perform one of the first theoretical analysis of channel characteristics of visible light communication. We will also report a series of simulation processes and results developed based on MATLAB and Simulink to present a comprehensive channel modeling and characterization study for both line-of-sight and non-line-of-sight paths. Chapter 3 details a new indoor localization system using coded visible light and depicts an experiment validation of the proposed system conducted in a real-world office setting. How to utilize the proposed positioning system for indoor autonomous robot navigation will be explored in Chapter 4. Finally, we answer the

two problems regarding multi-robot navigation and illustrate our framework in Chapter 5.
The conclusion and future work will be presented in Chapter 6.

Chapter 2 Visible Light Channel Modeling

To design, implement, and operate optical communication systems efficiently, it is important that the characteristics of the channel are well understood. Consequently, the prerequisite condition is to establish a model that can precisely describe its channel impulse response, which is then used to analyze the effects of channel distortion.

A considerable amount of work has been published on the channel characterization on infrared IR channel modeling [10, 12-19, 21], most of which depend on either recursive calculation methods [10, 12, 13], or Monte Carlo ray tracing approaches [14, 15, 18]. However, it has been reported that there exists significant difference between visible light channel and infrared communication, which implies that existing IR models cannot be applied to visible light communication (VLC). Meanwhile, most of existing channel modeling for VLC assume direction line of sight. In this case, the optical path loss can be easily calculated based on the knowledge of the location of the transmitter and the receiver, as well as the physical properties of them. However, indoor environment is not ideal. Receivers may use reflected signals off the room surface, such as walls, ceilings, and furniture. These reflections can be considered as unwanted signals or multipath distortions, which make the location estimation less accurate.

2.1 General Optical Channel Model

Different from radio wireless systems where various types of modulation techniques can be used, such as amplitude modulation (AM), pulse modulation (PM), or frequency modulation (FM), it is extremely difficult to collect appreciable signal power in a single electromagnetic mode for an optical wireless system. This spatially incoherent reception

makes it difficult to construct an efficient heterodyne or homodyne downconverter for AM, PM, and FM. As a result, *the intensity modulation (IM) with direct detection (DD)* become the most viable modulation/downconverter for optical wireless links, in which the desired waveform is modulated onto the instantaneous power of the carrier and a photodetector produces a current proportional to the square of the received electric fields.

As shown in Figure 2, the topologies of optical wireless system can be grouped into different classes, according to the degree of directionality of the transmitter and receiver and whether the link relies on the existence of a LOS path between them [20]. In this work, we will focus our attention on nondirected LOS and non-LOS link. LOS links require a clear path between the transmitter and receiver, which is designed to maximize power efficiency and minimize multipath distortion. However, non-LOS links increase link robustness and ease of use, allowing the system to operate even when barriers, such as people and cubicle partitions, stand between the transmitter and receiver.

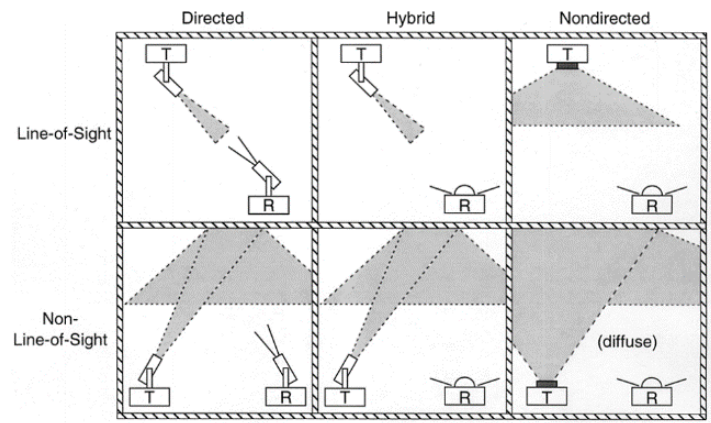


Figure 2 Classification of visible light links.

Non-LOS links, however, suffer from the effects of multipath propagation in the way similar to RF systems. This is because that multipath propagation introduces severe amplitude fades into electric field on the scale of a wavelength. This issue will be even worse if the size of the photodetector is proportional to one wavelength or less. This finding has encouraged the use of detector in model optical wireless receiver with a surface area typically millions of square wavelengths. A typical IM/DD-base optical wireless system can be seen in Figure 3, and its equivalent baseband model is also shown in Figure 4.

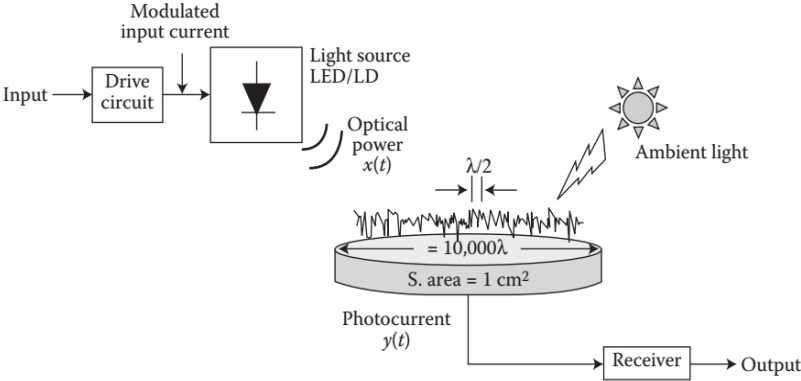


Figure 3 Transmission and reception in a visible light link with IM/DD.

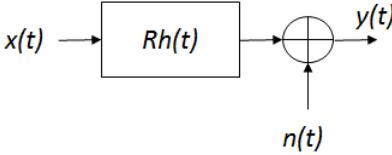


Figure 4 Modeling visible link as a baseband linear, time-invariant system.

We use the following equations to summarize the above model:

$$\begin{aligned}
y(t) &= Rx(t) \otimes h(t) + n(t) \\
&= \int_{-\infty}^{\infty} Rx(\tau)h(t-\tau)d\tau + n(t)
\end{aligned} \tag{1}$$

where $y(t)$ is the resulting photocurrent, R is the photodetector responsivity, $x(t)$ is the instantaneous transmitted optical power, $h(t)$ is the baseband channel impulse response, $n(t)$ is system noise, and \otimes represents the convolution.

In many applications, optical wireless link is operated in the presence of intense visible background light, for example sun light. While received ambient light can be minimized by optical filtering, it still adds shot noise, which can be modeled as white, Gaussian, and independent. It is also worth mentioning that when little or no ambient light is present, the dominant noise source would be receiver preamplifier noise, which is also signal-independent. This stands in contrast to the signal-dependent Poisson noise considered in photon-counting channel models. Ambient noise sources will be discussed in detail later.

In [22], the channel impulse response $h(t)$ was modeled by Gfeller and Bapst as follows

$$h(t) = f(x) = \begin{cases} \frac{2t_0}{t^3 \sin^2(FOV)} & t_0 \leq t \leq \frac{t_0}{\cos(FOV)} \\ 0 & \text{else} \end{cases} \tag{2}$$

where t_0 is the minimum delay.

2.2 Nondirected LOS Propagation Model

In general, for an optical wireless link where an LED is used as a source and large-area photodetectors are used as receivers, its angular distribution of the radiant intensity pattern is usually assumed to follow a Lambertian radiation pattern given by the following distribution

$$R_0(\phi) = \begin{cases} \frac{(m+1)}{2\pi} \cos^m(\phi) & \phi \in [-\pi/2, \pi/2] \\ 0 & \phi \geq \pi/2 \end{cases} \quad (3)$$

where m is called the *Lambertian order*, which is related to the LED semi-angle at half-power $\Phi_{1/2}$ and can be given as follows

$$m = \frac{-\ln 2}{\ln(\cos \Phi_{1/2})} \quad (4)$$

Consequently, the radian intensity can be given as

$$S(\phi) = Pt \frac{(m+1)}{2\pi} \cos^m(\phi) \quad (5)$$

Another important component in an optical wireless link is the photodetector. From analysis above, we see that ideally a large-area detector would be more suitable for indoor optical wireless system so that the multipath propagation issue can be alleviated. But in practice, it would cause many problems, for instance, increased cost, increased junction

capacitance, and increased noise on the receiver side. To build a fully functional optical wireless link, a non-imaging concentrator is commonly used in order to increase overall effective collection area. Yet, the optical gain of an ideal non-imaging concentrator can be modeled as

$$g(\Psi) = \begin{cases} \frac{n^2}{\sin^2 \Psi_c} & 0 \leq \Psi \leq \Psi_c \\ 0 & \Psi > \Psi_c \end{cases} \quad (6)$$

where n is the internal refractive index, and $\Psi_c \leq \pi/2$ is the field of view of the receiver (FOV). Based on the constant radiance theorem, we can also know that the FOV of a receiver system is related to the collection area of lens and the active photodetector area [23], and this relation can be illustrated as

$$A_{coll} \sin\left(\frac{\Psi_c}{2}\right) \leq A_r \quad (7)$$

Having the above equations, for a typical indoor environment with an optical wireless link having a Lambertian source and a receiver with a non-imaging concentrator of gain $g(\Psi)$, the channel gain can be represented as the following equation

$$H_{LOS}(\bullet) = \begin{cases} \frac{A_r(m+1)}{2\pi d^2} \cos^m(\phi) g(\Psi) \cos \Psi & 0 \leq \Psi \leq \Psi_c \\ 0 & else \end{cases} \quad (8)$$

where d and ϕ are the distance and the angle of a receiver with respect to the transmitter, respectively, as shown in Figure 5.

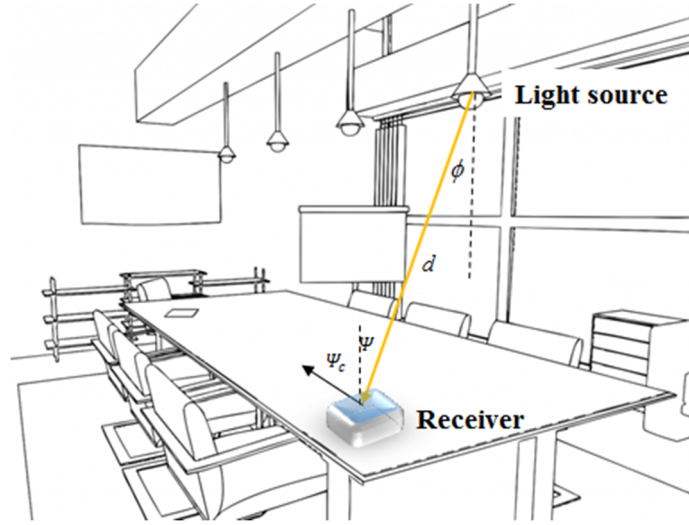


Figure 5 Propagation model of nondirected line-of-sight visible light link

As a result, for any receiver inside the projection area of the transmitter, its received power will become

$$P_{LOS} = P_t \times H_{LOS}(\bullet) \quad (9)$$

where P_t is the transmitted power over the channel of the light source.

A simulation is performed to graphically present the received power of simulated pairs of transmitters and receivers over nondirected visible light link for a specific room with the size of $5 \times 5 \times 3 \text{ m}$. System parameters for the calculation of the channel dc gain are shown in Table 1. Figure 6 illustrates the case where a single LED is installed on the ceiling at the center of the room, while Figure 7 shows the case where four LEDs are evenly distributed in the room.

Table 1 System parameters for simulation.

	Parameters	Values
Transmitter	Room size	5×5×3 m ³
	Reflectance factor	0.8
	Semi-angle at half power	70°
	Lambert's order	1
	Transmitted power	20 mW
	Numbers of LEDs	60×60
	Azimuth	0°
	Elevation	-90°
	Position (single LED)	(0, 0, 3)
	Position (four LEDs)	(-1.25, -1.25, 3), (-1.25, 1.5, 3) (1.25, -1.25, 3), (1.25, 1.25, 3)
Receiver	Half-angle FOV	60°
	Internal refractive index	1.5
	Active area (A _R)	1cm ²
	Azimuth	0°
	Elevation	90°
	Receive plane above floor	0.8 meter

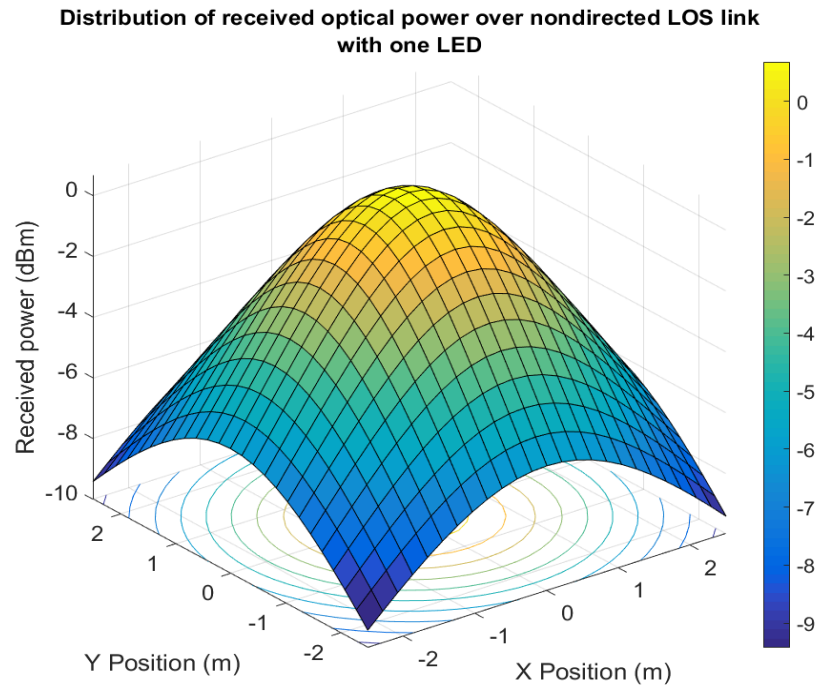


Figure 6 Distribution of received power over nondirected LOS link with one LED.

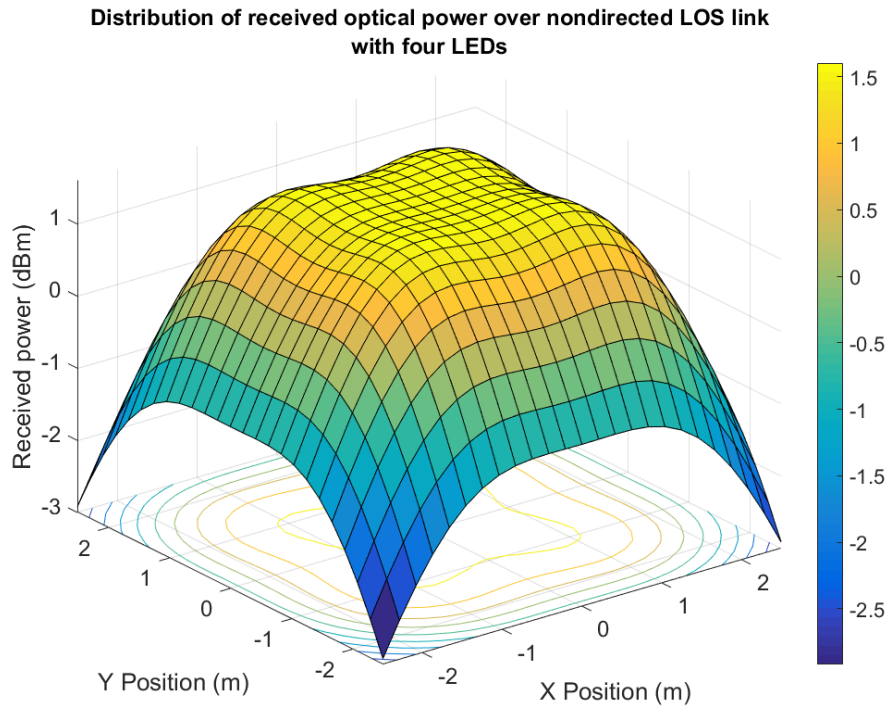


Figure 7 Distribution of received power over nondirected LOS link with four LEDs.

2.3 Nondirected Non-LOS Single-Reflection Propagation Model

Although determining the distribution of optical power over line-of-sight links throughout a room is adequate for basic power budget calculations, it does not allow the power penalty due to multipath propagation to be accurately predicted, since multiple reflections are not taken into consideration in the LOS propagation model.

2.3.1 Optical Power Distribution

Different from RF signal, visible light does not pass through opaque barriers but will certainly reflect off walls, ceilings, and various other objects inside a room, such as people

and furniture. Even worse, in many applications there is a need to offer wireless services to multiple users within a relatively large room. In this case, a source with a wide optical footprint needs to be adopted, which will add more reflected light into the environment. Yet, for indoor applications, nondirected non-LOS configuration is considered to be more flexible. It uses wider beam transmitters, wide FOV receivers, and scatters from surfaces within the room, thus offering a broader coverage area. However, non-LOS optical wireless links incur a high optical path loss (thus higher transmit power) and suffer from intersymbol interference (ISI) introduced by multipath propagation. The path loss can be increased even further if a temporary obstruction, such as moving people, obscures the receiver such that the main signal path is blocked. This situation is commonly referred to as shadowing [24].

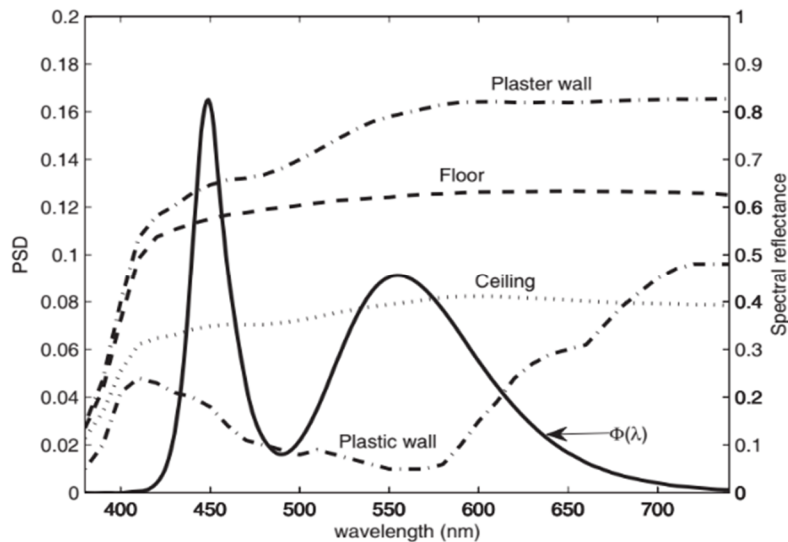


Figure 8 Compare power spectral distribution to the measured spectral reflectance of various texture.

As shown in Figure 8, the power spectral distribution (solid line, which corresponds to the left axis) is compared to the measured spectral reflectance (which corresponds to the right

axis) of plaster and plastic wall, floor, and ceiling, concluding that the reflectivity depends on both wavelength and the texture of the objects [25]. And the difference of reflectivity of different surfaces leads to different delay spread and inter-symbol interference. Figure 8 also shows that the plaster wall has the higher reflectivity than floor and ceiling. In this section, we further analyze the non-LOS optical link within the same simulated indoor environment we use in the previous section and we will focus our attention on the reflection from the four walls around the simulated environment.

With the consideration of reflections from the walls, we can further revise our model on the channel gain to the following equation:

$$P_r = P_t H_{LOS}(\bullet) + \int_{\text{Reflections}} P_t dH_{ref}(\bullet) \quad (10)$$

where P_r is the received power at the photodetector, P_t is the transmitted power over the channel of the light source, and $H_{LOS}(\cdot)$ is channel gain over the LOS optical link. Different from before, we add the received power of reflections from all the walls. Here, the DC channel gain of the first-order reflection can be represented as follows:

$$H_{ref}(\bullet) = \begin{cases} \frac{A_t(m+1)}{2(\pi d_1 d_2)^2} \rho dA_{\text{wall}} \cos^m(\phi_r) \cos(\alpha_{ir}) \\ \quad \times \cos(\beta_{ir}) g(\Psi) \cos(\Psi_r) & 0 \leq \Psi_r \leq \Psi_c \\ 0 & \Psi_r > \Psi_c \end{cases} \quad (11)$$

where, as illustrated in Figure 9, d_1 is the distance between the transmitter and a particular reflective point, and d_2 is the distance between this reflection point and the receiver. ρ is used to represent the reflectance factor of the surface where the reflective point situates. dA_{wall} is used to denote a reflective area of small region. ϕ_r denotes the angle of irradiance to a reflective point, while α_{ir} and β_{ir} are the angle of irradiance to the same reflective point and the angle of irradiance to the same receiver, respectively. Finally, Ψ_r is the angle of incidence from the reflective surface. It can be seen from the equation that the reflection characteristics of objects for a given environment depend on various factors, including the transmitted signal wavelength, the angle of incidence ϕ_r , and the texture of the objects, such as material and roughness, relative to the wavelength.

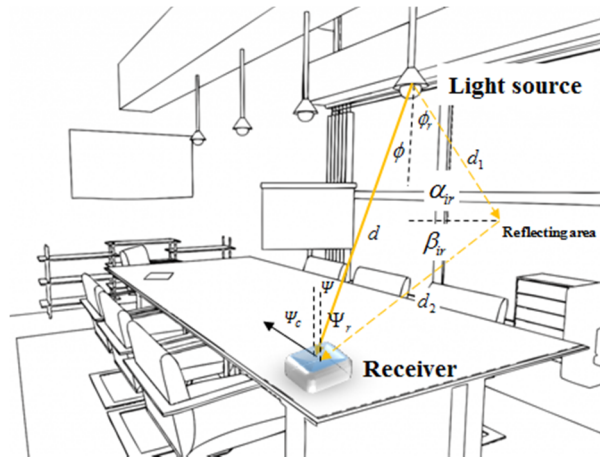


Figure 9 Propagation model of reflected link.

In order to accurately predict the path loss for non-line-of-sight links, we perform another simulation to analyze the channel characteristics of nondirected non-LOS link for the same simulated environment (see Table 1) focusing on the first-order reflected path. The simulation setup is also shown in Figure 9. Here, the surfaces of the walls are divided into

small areas with the size of ΔA . And there are totally N reflecting elements. Having this, each reflecting element can be considered as a receiver, which will 1) receive light signal through the nondirected LOS link from the transmitter, and 2) re-emit the collected light signal scaled by reflectivity ρ . Figure 10 & 11 demonstrate the optical power distribution of the first reflection at the receiving surface from totally four walls in the simulated indoor environment with one and four LED sources, respectively.

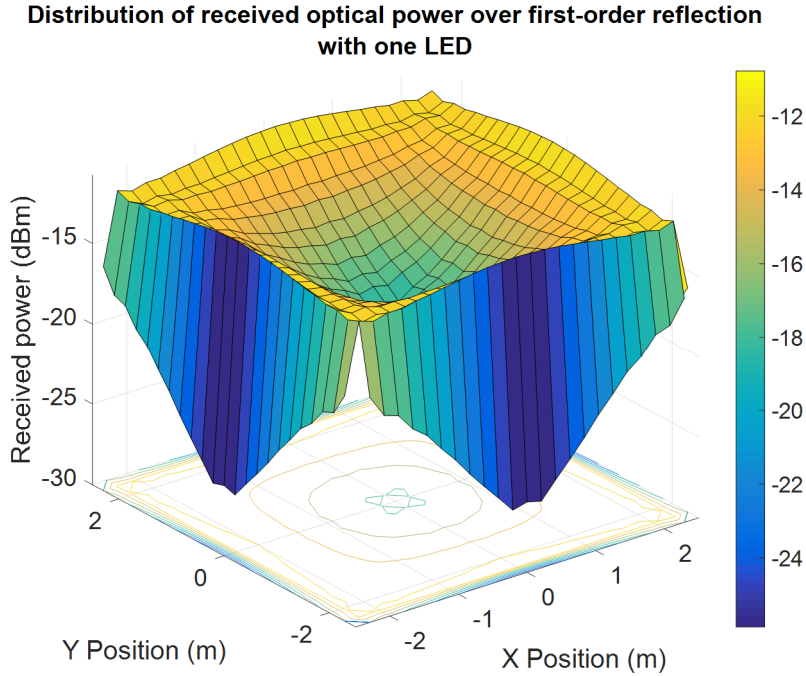


Figure 10 Distribution of received power from first-order reflection with one LED.

Distribution of received optical power over first-order reflection with four LEDs

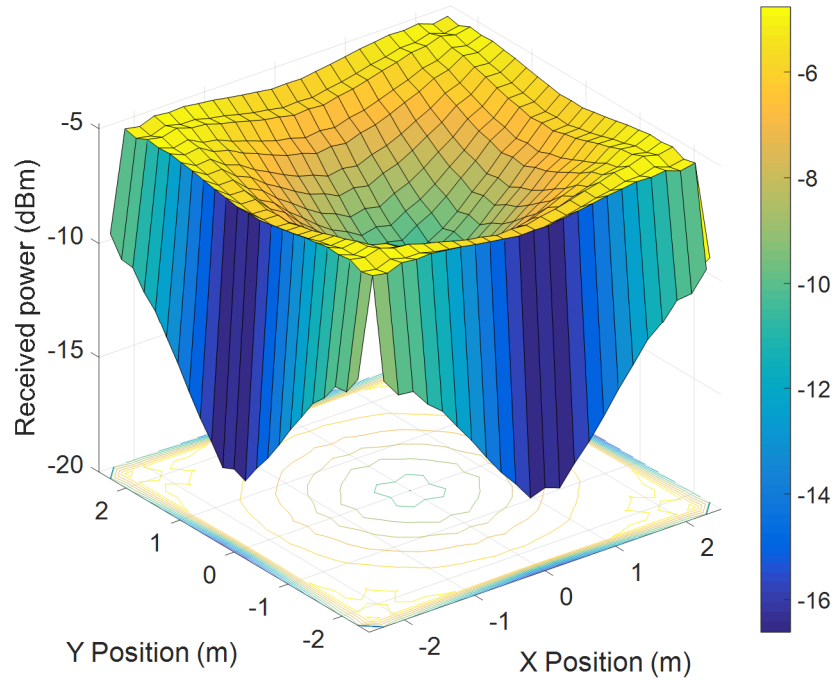


Figure 11 Distribution of received power from first-order reflection with four LEDs.

Additionally, we combine the result from the first simulation, which only includes the LOS optical link (seen in Figure 6 & Figure 7), and the reflected optical power from four walls together. Figure 12 illustrates the resulting power distribution in the simulated environment, including both LOS component and non-LOS first-order reflection with only one LED in the middle of the room. Figure 13 shows the case with four LEDs.

Distribution of received optical power over nondirected LOS link plus first-order reflection with one LED

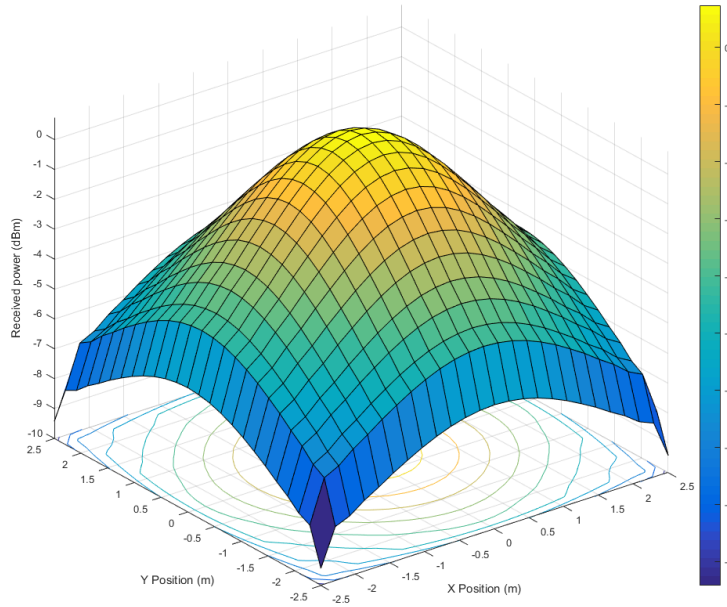


Figure 12 Optical power distribution for single LED.

Distribution of received optical power over nondirected LOS link plus first-order reflection with four LEDs

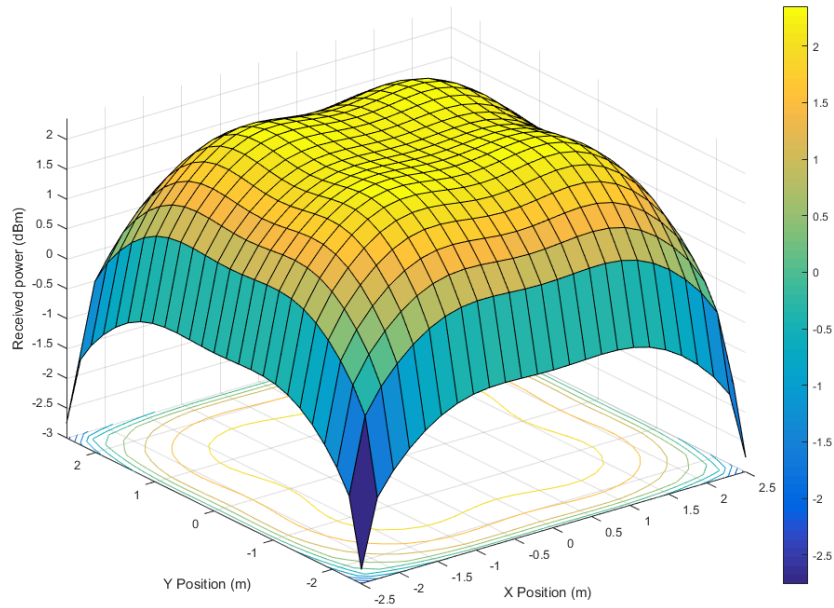


Figure 13 Optical power distribution for four LEDs.

2.3.2 Channel Impulse Response

Further, the channel impulse response with first-order reflection from the surrounding walls can be described as

$$h_{nlos}^1(t) = \sum_{k=1}^{\square} \frac{(m+1)A_r \Delta A}{2\pi d_{1k}^2 d_{2k}^2} \rho \cos^m(\phi_{r_k}) \cos(\alpha_{ir_k}) \times \cos(\beta_{ir_k}) \delta\left(t - \frac{d_{1k} + d_{2k}}{c}\right) \quad (12)$$

Given the system parameters in Table 1, we calculate the impulse response of the simulated LOS link and the first-order diffused reflection as shown in Figure 14 and Figure 15.

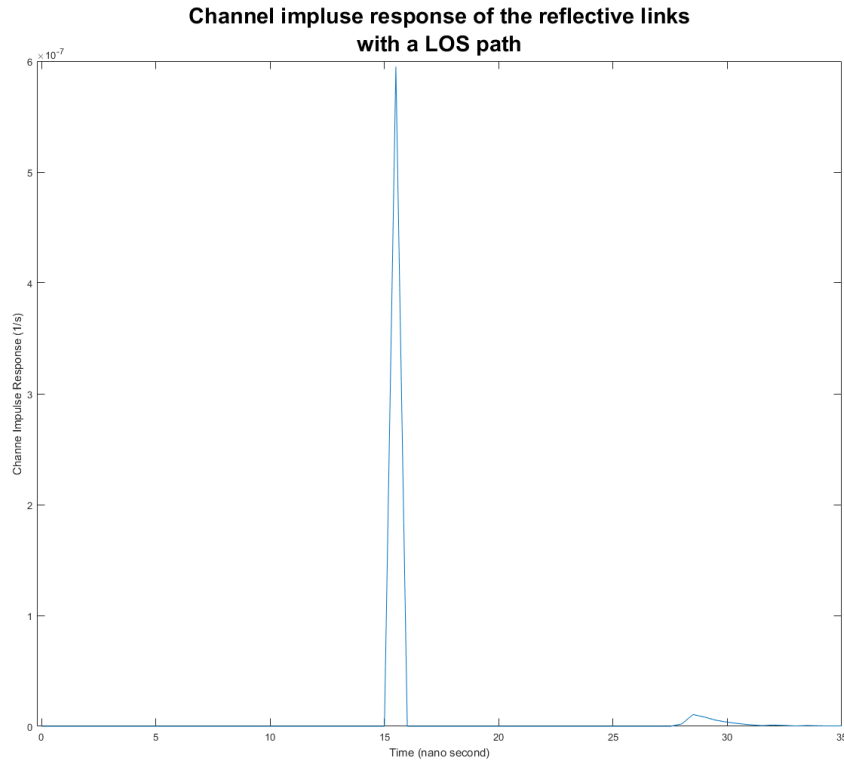


Figure 14 Impulse response of the reflective links with a LOS path.

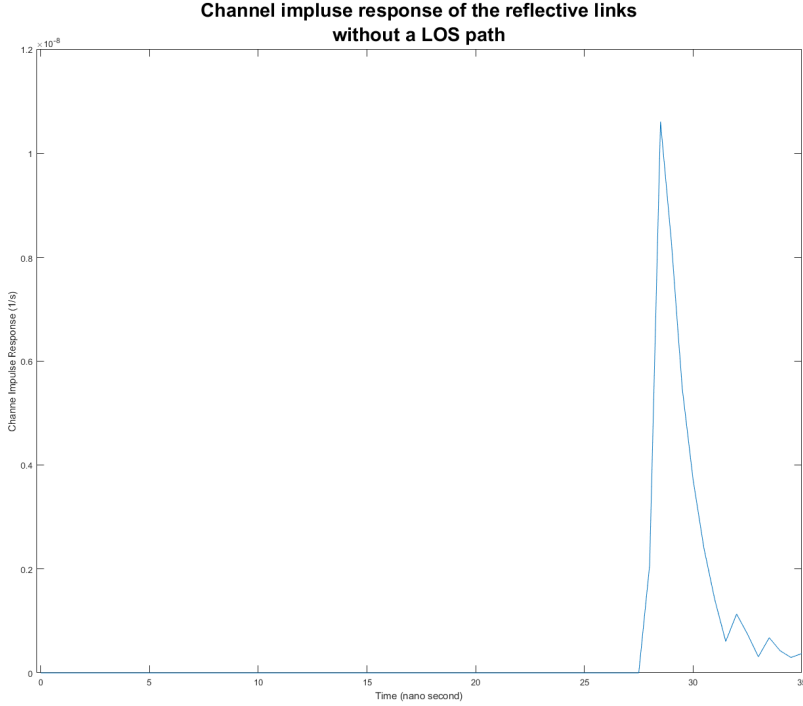


Figure 15 Impulse response of the reflective links without a LOS path.

2.3.3 Channel Delay Spread

As discussed above, non-LOS links suffer from intersymbol interference caused by multipath propagation, which can be characterized by the root-mean-square (RMS) delay spread of the signal. As in Eq. (10), for multipath cases, the total received optical power includes both the line-of-sight ones and the first-order reflected links. It can be given as the following equation:

$$P_{rT} = \sum_{i=1}^M P_{LOS,i} + \sum_{j=1}^N P_{ref,j} \quad (13)$$

where M represents the number of LOS optical links from the transmitters to a specific point in the environment and N represents the number of reflection optical links to the same point. $P_{LOS,i}$ denotes the received optical power from the i th LOS link and

$P_{ref,j}$ indicates the received power from the j th reflected link. In general, delay spread is a measure of how much signal propagation smears the signal over time due to scattering from objects at various distances. If delay spread is large compared to the symbol time, interference between conjunctive symbols will happen, limiting the maximum data rate of the communication channel. The RMS delay spread is one of important performance criteria for the upper bound of the data transmission rate for a specific communication channel. **Error! Reference source not found.** presents then mean delay spread observed in the simulation, which is the first moment of the power delay profile and can be defined by

$$\mu = \frac{\sum_{i=1}^M P_{LOS,i} t_{LOS,i} + \sum_{j=1}^N P_{ref,j} t_{ref,j}}{P_{rT}} = \frac{\int th^2(t) dt}{\int h^2(t) dt} \quad (14)$$

We then calculate the root mean square of the channel delay time, which is the square root of the second central moment of the power delay profile and can be represented as the following equation

$$D_{RMS} = \sqrt{\mu^2 - (\mu)^2} = \left[\frac{\int (t - \mu)^2 h^2(t) dt}{\int h^2(t) dt} \right]^{1/2} \quad (15)$$

Figure 16 and Figure 17 illustrate the mean delay spread and RMS delay spread of the simulated optical link in the case of a single transmitter situated at the center of the room, respectively.

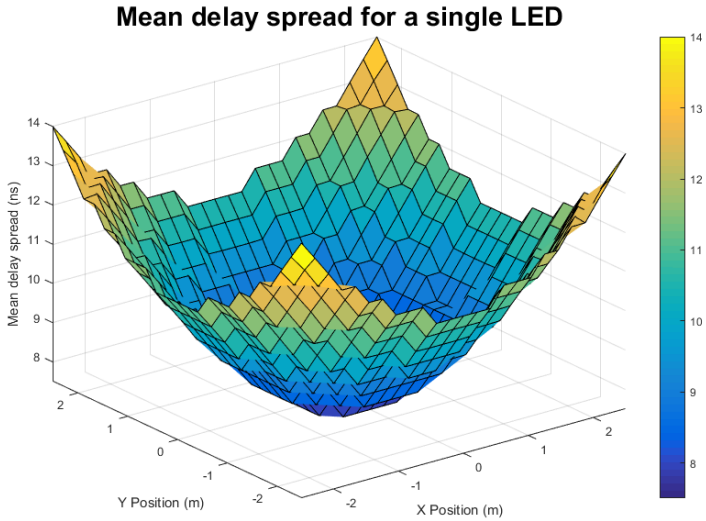


Figure 16 Mean delay spread for a single LED.

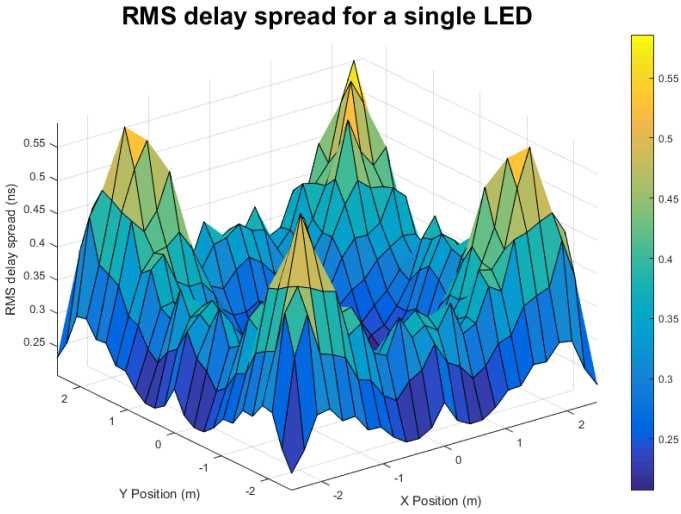


Figure 17 RMS delay spread for a single LED.

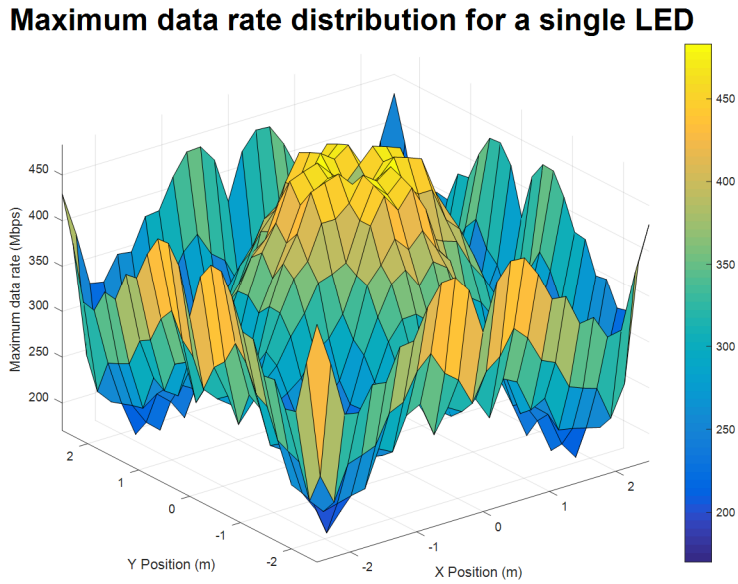


Figure 18 Maximum data rate distribution for a single LED.

Finally, the RMS delay spread determines the maximum bit rate that can be transmitted through a communication channel [26] and it can be given by the following equation when an equalizer is not used

$$R_{\max} \leq \frac{1}{10 \times D_{RMS}} \quad (16)$$

Thus, we can plot the maximum data rate distribution for the simulated system, as shown in Figure 18, from which we can see that the maximum achievable data rate in our simulated channel without requiring an equalizer is about 170 Mbps.

Similar to the single transmitter case, we further perform our simulation for the case of four transmitters placed at the locations of $(-1.25, -1.25, 3)$, $(-1.25, 1.5, 3)$, $(1.25, -1.25, 3)$,

and (1.25, 1.25, 3). Here, both the LOS optical links from all four transmitters and the reflected links from the surrounding four walls have been taken into consideration.

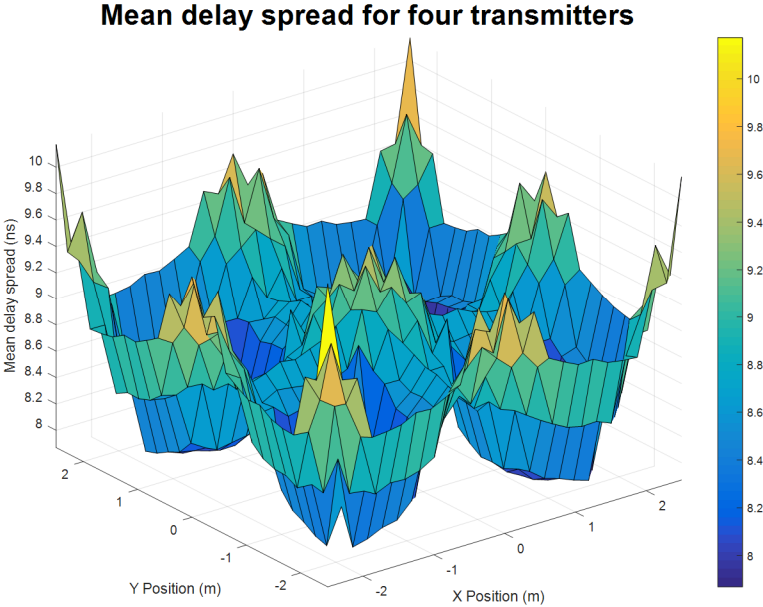


Figure 19 Mean delay spread for the case of four transmitters.

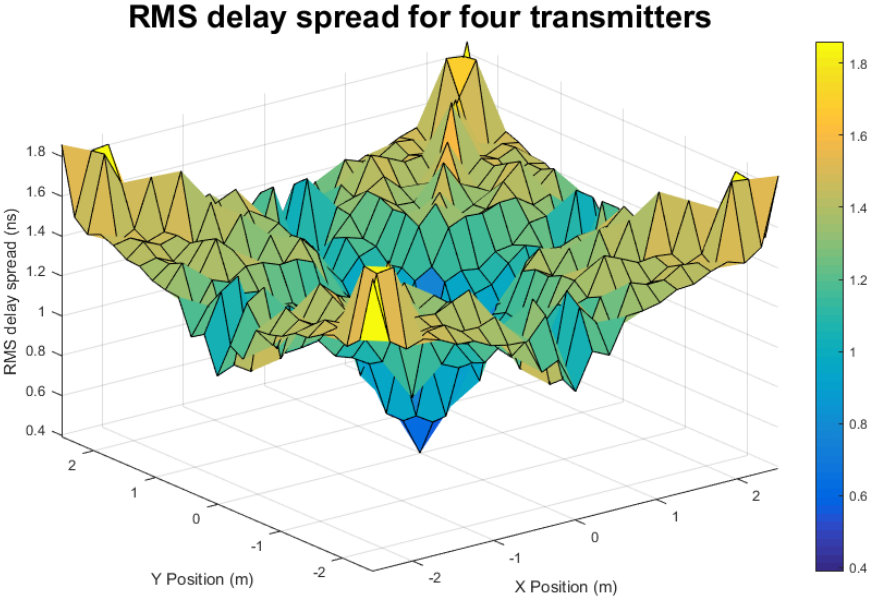


Figure 20 RMS delay spread for the case of four transmitters.

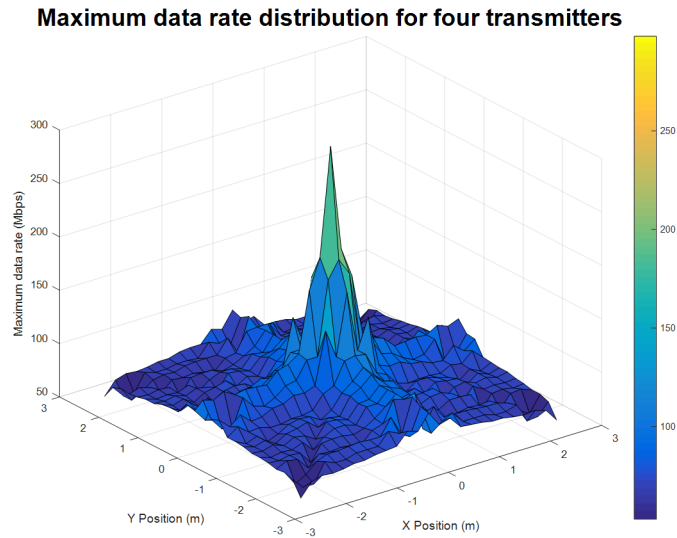


Figure 21 Maximum data rate distribution for the case of four transmitters.

Figure 19 demonstrates the mean delay spread distribution for the case of four transmitters while Figure 20 calculates the RMS delay spread distribution, from which we can see that the maximum value is 1.86 nanosecond and the average value is 1.29 nanosecond. We also plot the maximum data rate distribution for the given settings in Figure 21 and a 53 Mbps can be achieved without the need for an equalizer, which is much less than the 170 Mbps from the case of a single transmitter.

2.4 OOK Modulation

As we present above, different from RF systems where the amplitude, frequency and phase of the carrier signal can be modulated, in most optical systems operating below 2.5 Gbps data rates, it is the intensity of the optical carrier that is modulated. While for systems which have even higher transmission data rates, external modulation will be required. In

this section, the most widely used modulation technique in optical systems, OOK modulation, will be discussed in terms of power efficiency and bandwidth efficiency. Its spectral properties, error probability, and the power and bandwidth requirements will be presented.

As one of the baseband modulation schemes, OOK is more tolerant to the effects of the multipath channel. It does not translate the data to a much higher carrier frequency before modulating the intensity of the light source. Therefore, a significant portion of the signal power is restricted to the dc region. Due to this simplicity, OOK is the most reported scheme for IM/DD in optical communication system: a bit '0' is denoted by the absence of a light pulse; a bit '1' is represented by a light pulse which could either occupy the entire, also known as non-return-to-zero (NRZ), or part of the bit duration, also known return-to-zero (RZ). An example of such mapping of OOK-NRZ and OOK-RZ with a duty cycle $\eta = 0.5$ for average transmitted power of P_{avg} is given in Figure 22.

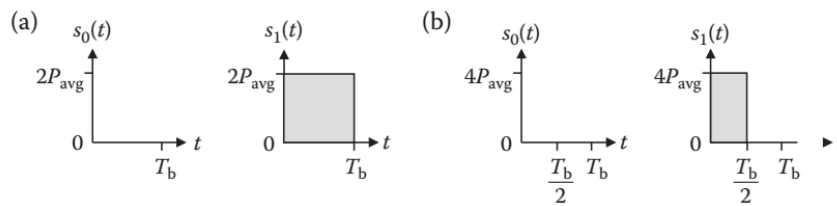


Figure 22 Transmitted waveforms for OOK: (a) NRZ and (b) RZ ($\eta=0.5$).

Further, the electrical power spectral densities (PSDs) of the OOK-NRZ and OOK-RZ ($\eta = 0.5$) assuming independency and identically distributed (IID) one and zeros are given by

$$S_{OOK-NRZ}(f) = (P_{avg}R)^2 T_b \left(\frac{\sin \pi f T_b}{\pi f T_b} \right)^2 \left[1 + \frac{1}{T_b} \delta(f) \right] \quad (17)$$

and

$$S_{OOK-RZ(\eta=0.5)}(f) = (P_{avg}R)^2 T_b \left(\frac{\sin \pi f T_b / 2}{\pi f T_b / 2} \right)^2 \left[1 + \frac{1}{T_b} \sum_{n=-\infty}^{\infty} \delta(f - \frac{n}{T_b}) \right] \quad (18)$$

, respectively. Here, P_{avg} is the average power, R is the sensitivity of the photodetector, T_b is the bit duration, and $\delta(\cdot)$ is the Dirac delta function. Additionally, we plot the PSDs of OOK-NRZ and OOK-RZ in Figure 23, where we use the same average transmitted power P_{avg} for both cases, and both the power axis and the frequency axis are normalized to the average electrical power multiplied by the bit duration, $(P_{avg}R)^2$, and the bit rate, $1/T_b$, respectively.

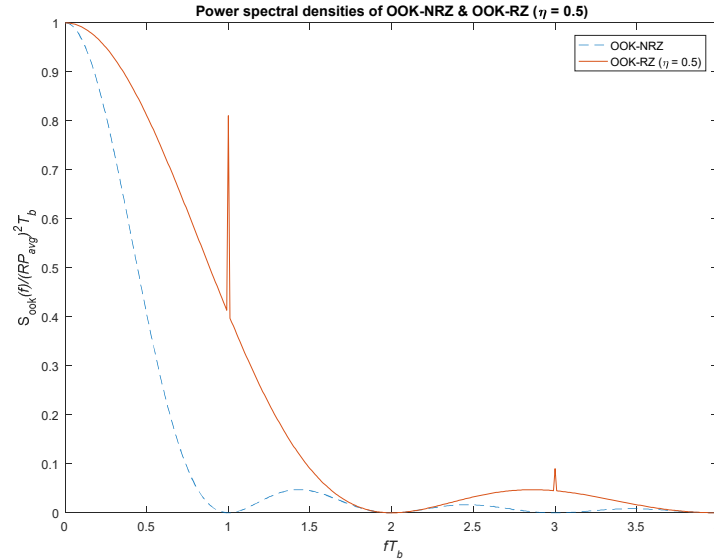


Figure 23 Power spectral densities of OOK-NRZ and OOK-RZ.

Figure 23 demonstrates that for a given average transmitted power, OOK-RZ will have twice the average power of OOK-NRZ. This can be seen by calculating the areas under the two curves. Further, for baseband modulation schemes, the bandwidth requirement is generally defined as the span from DC to the first null in the PSD of the transmitted signal. Unsurprisingly, as shown in Figure 23, the OOK-RZ in our simulation has twice the bandwidth requirement of OOK-NRZ, which is due to the fact that the pulses in OOK-RZ are only half as wide as in OOK-NRZ. From Figure 23 we can also see that OOK-NRZ has spectral nulls at multiples of the bit rate, while, on the other hand, OOK-RZ has discrete terms at odd multiples of the bit rate. Finally, both OOK-NRZ and OOK-RZ have significant power components at DC and low frequencies. This implies that electrical high-pass filtering would not be effective in reducing the interference produced by artificial light sources, because high cut-on frequencies cannot be used without introducing significant baseline wander.

2.5 Modeling Background Noise

As shown in Figure 4, a typical optical communication channel often can be affected by additive white Gaussian noise (AWGN). Consequently, a continuous-time filter with an impulse response $r(t)$ is normally used. This filter is often designed to be matched to the transmitted pulse shape $p(t)$, and followed by a sampler and voltage comparator which is set midway between expected bit '1' and '0' levels. We have plotted this revised model in Figure 24.

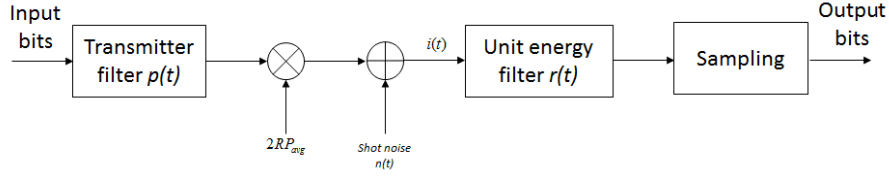


Figure 24 Block diagram of OOK based optical system with AWGN.

Here, the transmitter filter has a unit-amplitude rectangular impulse response $p(t)$, with a duration of one bit. Further, the output of the transmitter filter is scaled by the peak detected signal photocurrent $2RP_{avg}$ and P_{avg} is the average received optical power. Similar to the linear model in Figure 4, the signal-independent shot noise $n(t)$ is then added to the receiver and modelled as white and Gaussian, with a double-sided power spectral density $N/2$, given by the following equation

$$\frac{N}{2} = qI_B \quad (19)$$

where q is the electron charge and I_B is the average photocurrent generated by the background light.

The received signal at the input of the matched unit energy filter can be modelled as

$$i(t) = \begin{cases} I_p + n(t) & \text{'1' transmitted} \\ n(t) & \text{'0' transmitted} \end{cases} \quad (20)$$

where $n(t) \sim N(0, \sigma^2)$ is the additive white Gaussian noise due to ambient light with double-sided power spectral density $N/2$, zero mean, and a variance of σ^2 , and I_p is the peak photocurrent.

Having this, a digital symbol '1' will be assumed to have been received if the resulting signal is above the threshold level and '0' otherwise. Therefore, the probability of error of the channel can be represented as

$$P_e = p(0) \int_{Th}^{\infty} p(i/0) di + p(1) \int_0^{Th} p(i/1) di \quad (21)$$

where Th is the threshold signal level, $p(0)$ and $p(1)$ are the probabilities of bit '0' and bit '1', respectively. Besides, the marginal probabilities are defined as:

$$p(i/0) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-i^2}{2\sigma^2}\right) \quad (22)$$

$$p(i/1) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(i - I_p)^2}{2\sigma^2}\right) \quad (23)$$

For equiprobable symbols, both $p(0)$ and $p(1)$ are chosen to be equal to 0.5; therefore, the optimum threshold point is $Th = 0.5I_p$ and the conditional probability of errors will be reduced to

$$P_e = Q\left(\frac{Th}{\sigma}\right) \quad (24)$$

where $Q(\cdot)$ is Marcum's Q -function, which is the area under the Gaussian tail, represented by

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\alpha^2/2} d\alpha \quad (25)$$

Additionally, in the case of a matched filter, Eq. (20) can be replaced by

$$y_i = \begin{cases} E_p + n_i & \text{'1' transmitted} \\ n_i & \text{'0' transmitted} \end{cases} \quad (26)$$

Meanwhile, the variance of the noise samples at the output of the filter is dependent only on the PSD of the noise input and the energy in the impulse response of the matched filter. Thus, if the input is additive white Gaussian with a double-sided PSD $N/2$, the variance of the noise at the output of the filter can be denoted as:

$$\sigma^2 = \frac{N}{2} \int_{t=0}^{T_b} r^2(t) dt \quad (27)$$

Thus, the standard deviation can be represented as:

$$\sigma = \sqrt{\frac{NE_p}{2}} \quad (28)$$

Finally, having Eq. (24), Eq. (26), and Eq. (28), we can have the following

$$P_{bit_error_ook} = Q\left(\sqrt{\frac{E_b}{N}}\right) \quad (29)$$

where E_b denotes the average energy per bit and can be represented as

$$E_b = \frac{E_p}{2} = 2(RP_{avg})^2 T_b \quad (30)$$

and E_b/N is commonly referred to as the Signal-to-Noise ratio per bit (SNR).

Based on the optical communication model with OOK modulation and a matched filter discussed above, we perform another simulation to investigate the bit error rate (BER) for OOK-NRZ with randomly generated 100000 symbols and plot the result in Figure 25.

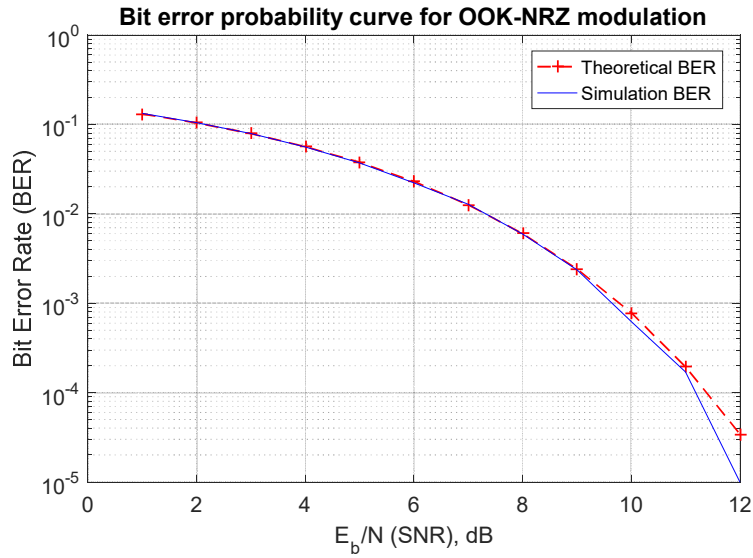


Figure 25 BER simulation for OOK-NRZ channel.

On the other hand, for the case of OOK-RZ modulation, the average energy per bit is increased by a factor of $1/\eta$, and can be described as

$$E_b = \frac{E_p}{2} = (2(RP_{avg})^2 T_b / \eta) \quad (31)$$

Comparing Eq. (30) and Eq. (31), we can easily see that OOK-RZ requires less electrical power (3 dB) than OOK-NRZ in order to achieve the same signal-to-noise ratio. But this comes at the expense of doubling the bandwidth.

Chapter 3 Coded Light Based Indoor Localization

3.1 Introduction

In the context of indoor localization for location-based services, there exist several approaches to provide indoor location information. More specifically, Zou et al. [27] applied an online sequential extreme learning machine based indoor localization algorithm on WiFi signals so that the proposed system reduce the time consumption and manpower cost for offline site survey in traditional fingerprinting-based wireless signal indoor localization systems. Wu et al. [28] proposed a genetic algorithms based localization scheme to estimate the location of unknown RFID nodes, given that RFID technology has been used in a variety of IoT applications. Alletto et al. [29] placed Bluetooth Low Energy (BLE) beacons in different rooms of a museum as wireless landmarks. An application running on the user's mobile device is used to determine the user's location based on the value of Received Signal Strength Indicator (RSSI) so that personalized cultural contents can be provided to visitors.

Prior systems that employ visible light to determine indoor position include [6, 30-34]. For these systems, high precision localization schemes often require well-shaped LEDs for propagation modeling [6, 33, 34] or ultra-dense deployment [33] (multiple LEDs within the camera's field-of-view). In contrast, our system features an easy-to-deploy approach in which off-the-shelf Digital Light Processing (DLP) projectors and light sensors are used to provide high-resolution localization. Since projecting binary images into the environment is the inbuilt function of the projector, the proposed technology does not require any augmentation on the projector itself.

Also on the topic of indoor localization, there exists a large body of literature including UWB [7, 35], WiFi [8, 36, 37], Bluetooth [9, 38], ZigBee [39], radar [40], and magnetic sensing [41].

In this dissertation, we advocate an encoded projection based localization system with the focus on low-powered mobile devices featuring limited power consumption, communication capability, and computation resources. We call it *Pilot*. To the author's best knowledge, this is the first time that projection-based encoded light signal is exploited as an indoor localization method to obtain fine-grained, real-time location data for such devices. And a performance comparison between *Pilot* and the state-of-the-art existing work is given in Table III.

3.2 Projector Encoding

Pilot takes advantage of particular properties of DLP projectors in order to assign unique coordinates to each pixel inside the projection area. Foremost, a DLP projector normally has a so-called Digital Micromirror Device (DMD) chip in it, which consists of millions of micro optical mirrors arranged with diamond pixel array geometry and configuration. These micro mirrors can be flipped between two states (*on* and *off*) independently at a high frequency. This fast-flipping property allows it to be used to modulate light using OOK modulation scheme by changing projected images.

Another important property that is exploited is that a sequence of gray-coded binary patterns can be used to encode the projection area by assigning a unique pair of

coordinates to each pixel in it. So, while projecting a sequence of gray-code patterns by flipping the corresponding mirrors inside the projector, the light signals coming out of each mirror unambiguously transmit its pixel coordinates on the projector plane into the environment. These two properties work in concert to enable Pilot to provide fine-grained location information simply but reliably.

To capture light intensity at different locations, we use an off-the-shelf light sensor. This is particularly well suited to locate mobile devices, which are normally low-cost, battery-operated, and have limited computation resources. If mass-produced, this sensor and its required peripheral circuit might cost less than two dollars.

Our method has one important constraint shared by all existing visible light-based indoor localization systems: it can only locate devices inside the projection area, or rather it has a line-of-sight requirement. This condition needs to be met not only because we use visible light to convey localization information, but also because we envision the projector to be integrated with existing lighting fixtures in the environment, providing both location discovery and smart environment illumination *simultaneously*.

Sensing in Pilot is surprisingly robust in a wide variety of use contexts. The same sensing technique can be used for TV controllers, mobile robots, rice cookers, temperature controllers, and many other low-powered IoT devices. The only notable restriction is that these devices must be *seen* by the projected light, thus blocking the light will prevent the device from locating itself. So, for example, a smart shopping cart augmented with our

sensor could be used for customer behavior tracking, but if the sensor is covered by a shopping item, the cart is not likely to locate itself inside the shop. However, this is not necessarily a negative quality. If customers are selecting personal products, they might not want to be tracked by the system. A good example of this is the pharmacy area in a retail store, where shoppers would like to protect their privacy about which products they are viewing and purchasing.

3.3 Location Discovery

3.3.1 Gray-coded Position Encoding

Pilot is based on the projection of a set of patterns onto the desired environment, such as walls, tables, and much more. These patterns are specially designed so that each pixel inside the projection area has its own *codeword* (a sequence of binary digits), and there is a direct mapping from the assigned codeword to the corresponding coordinates of the pixel on the projected pattern. Codewords are simply numbers, which can be mapped in the pattern by using gray levels, colors, or geometrical representations.

We adopted an easy but robust coding strategy called gray code [42], which is one of the common time-multiplexing pattern projection techniques. In this case, a set of patterns is successively projected onto the surface. The codeword for a given pixel is formed by the sequence of illuminance values for that pixel across the projected patterns along time. The advantage of this design is that the structure of each pattern can be very simple, thereby the procedure to decode the coordinates from the codeword being simple as well.

In Pilot, only two illumination levels are used (*therefore OOK-NRZ*), which are coded as '0' (*black*) and '1' (*white*). Each pixel has its own codeword formed by a sequence of 0s and 1s that correspond to its value in each projected pattern. Thus, a codeword is obtained once the sequence is completed. This design helps Pilot achieve high accuracy for localization due to three factors.

First, since multiple patterns are projected, the codeword basis tends to be small (two in Pilot) and only a small set of primitives (0 and 1) are used, which are easily distinguishable between each other. Second, a coarse-to-fine paradigm is followed, meaning the position of a pixel is being encoded more precisely while the patterns are successively projected. Finally, an important advantage of gray code is that consecutive codewords have a Hamming distance of 1, meaning it is more robust against noise. Figure 26 demonstrates an example in which three vertical (left three) and three horizontal patterns (right three) are used to resolve an 8×8 unit grid.



Figure 26 An example of gray code images for an 8×8 grid.

3.3.2 Embedded Synchronization Signal with High Reliability

Pilot can be used in a variety of indoor environments. More specifically, we use Manchester coding [43] to transmit the data patterns through a visible light channel to remove the dependency on the DC voltage of the received light intensity, which can be influenced severely by ambient lighting conditions. This scheme improves the signal-to-noise ratio of

the overall system. Further, Pilot does not require any additional communication channel for synchronization but instead has embedded synchronization frames inside each data package, allowing the sensor units to be synchronized with the data source automatically.

As a matter of fact, this encoding scheme also helps with the flickering issue of visible light communication. This is usually due to long runs of '0's or '1's and may cause serious detrimental physiological changes in humans if the fluctuation in the brightness of the light is perceivable to human eyes [44]. In Pilot, each '1' or '0' is followed by its reverse bit (except the header, which has three '1's in a row), and the number of '1's in each packet is only one more than the number of 0s. This design keeps switching between '1' and '0', freeing the system from fluctuation so that users always see the projector as a stable light source.

3.3.3 Signal Detection & Position Reconstruction

The sensor unit in Pilot includes one or two light sensors to decode light signals. The number of light sensors on a single sensor unit depends on the target application, as detailed later. Each light sensor is connected via a signal conditioning circuit to the same Arduino Micro microcontroller, using its general digital I/O port. The microcontroller is an Atmel Atmega32U4 chip running at 16 MHz and contains 26 general digital I/O ports, each of which can be used to collect light intensity by configuring it for input and evaluating the voltage level. The signal conditioning circuit contains an amplifier and voltage comparator, which not only increases the signal strength even when the sensor unit is far away from the projector, but also removes unwanted noise. In our prototype implementation, the sensor

unit can detect the projector's location signals from up to 4.5 meters. Finally, the firmware running on the microcontroller collects light signals every 250 microseconds and restores the original position data, which can then be sent out to a host PC for event triggering and data logging via a WiFi module. We use RN-XV WiFly module from Sparkfun in our current setup.

3.4 Projector-Light Sensor Homography

Pilot is designed to provide fine-grained 2D physical coordinates for battery-powered mobile devices with high accuracy. In attempting to answer this challenge, another key component in the system is projector-light sensor calibration.

To obtain physical coordinates of the sensor unit, the position, orientation, and optical parameters of the projector relative to the interaction surface should be known. Consider that there exists a point (x', y') in the projector plane, that, in our case, is a pixel on the DMD mirror array inside the projector, and it is projected to a point on a projection screen, for instance, a flat table, with physical coordinates (x, y) . (Here, we assume that the origin on the table is already known.) The relationship between (x', y') and (x, y) is determined by a planar projective transformation whose parameters further depend on the position and orientation of the projector relative to the table, and the optical parameters of the lens inside the projector. The following equation demonstrates this relationship, where H is the perspective transformation between the projector plane and the table, and is also known as the homography between the pixel coordinates on the projector plane and the Euclidean coordinates in the physical space.

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = H \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} \quad (32)$$

This *planar homography*, or *projective transformation*, defines a non-singular linear relationship between points on planes, which plays an important role in the geometry of multiple views.

It is evident that the correspondence condition $X = HX'$ is projective relationship because it has only involves projective geometric relationship, such as the intersection of lines and planes. In other words, the homography encodes the correspondence between the images of the points in one view to the images of the same points in the other view and only depends on the intrinsic and extrinsic parameters of the devices used to capture the two views and the parameters of the 3D plane. Since planar surfaces are ubiquitous in the environment, estimating homography induced by a plane between two views is an important step in many applications, such as camera calibration, augmenting reality, stitching and warping images, and other applications that use perspective geometry such as image registration and image mosaicing. It has also been used widely in tracking applications using multiple cameras and to build structured light system, which usually consists of a projector and a camera. In our system, we use the homography relationship between the projector plane and the light sensor plane to infer the transformation matrix between these two planes, which can be used to transform pixel coordinates on the projector DMD chips to the physical coordinate in the space.

In our proposed system (x', y') coordinates can be observed and decoded by a light sensor and (x, y) can be measured relative to a user-defined origin. Therefore, H can be calculated with the following steps: (1) marking a certain number of points in the projection area; (2) measuring the distances between these points and the origin of the physical plane; and (3) collecting the pixel coordinates of these points using a light sensor.

Further, this transformation matrix can be applied to future sensor readings to obtain the Euclidean coordinates of a projected point. We have followed these steps to calculate the homography in our experiments and evaluate the performance of this design in the sense of localization accuracy in the next section.

3.5 System Evaluation

To assess the feasibility of our localization approach, we created a proof-of-concept system seen in Figure 27. For evaluating the different features of our Pilot localization technique, a comprehensive evaluation was conducted. These experiments are used to evaluate the following features of this proposed system.

- Localization Accuracy
- Localization Latency
- Localization Refresh Rate
- Ambient Light Robustness

Different types of experiments have been designed to examine these features respectively. All these experiments are conducted under an indoor environment. During the evaluation,

a DLP projector was installed behind the ceiling (Figure 27 Middle), and a flat office table was used as an interaction space. The distance between the projector and the office table is 2 meters, and the resulting projection area is 1380 mm × 860 mm (54.3" × 33.85").

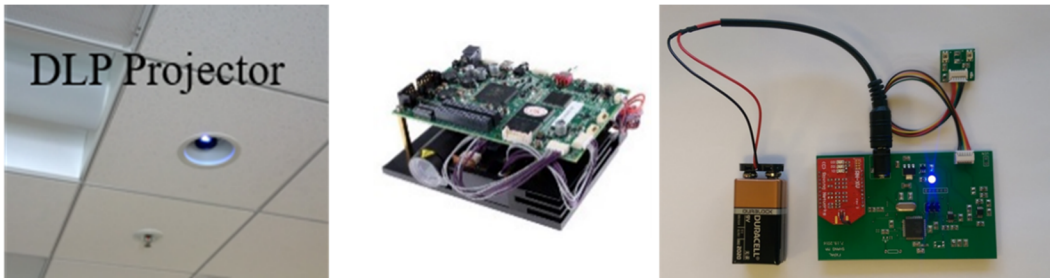


Figure 27 The DLP projector and a sensor module in Pilot.

3.5.1 Localization Accuracy

3.5.1.1 Experiment Design

To evaluate the localization accuracy of our proposed system, a test arena was developed. A commercial cutting mat with grid markers was placed on the table to provide ground truths data. The size of the cutting mat itself is 36"× 48" and the grids on top of it are located at a 0.5" distance.

As shown in Figure 28, 64 points (black dots marked on the mat) were chosen for data collecting, and they were selected to be uniformly distributed in the projection area. A sensor unit was placed at all these points to collect pixel coordinates on the projection plane, and their physical distances regarding the center were also measured. The homography under this setup was calculated based on these two sets of data.

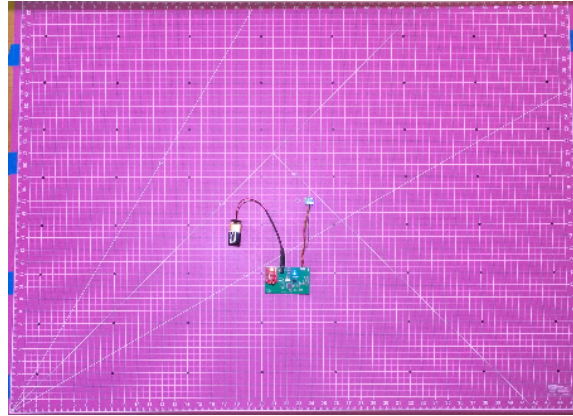


Figure 28 A test arena for calculating the homography.

We also collected the pixel coordinates of another 56 points, which are different from the 64 points from the first step, and applied the homography calculation from to them. The physical coordinates of these 56 points were also measured as ground truth. The difference between the ground truth and the calculation from Pilot for these new 56 points shows the localization error. These errors are plotted in Figure 29 for better understanding.

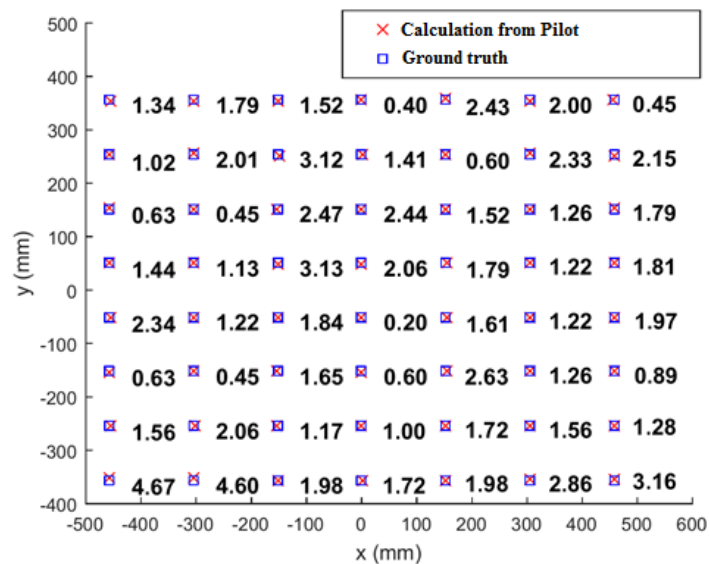


Figure 29 Localization errors of all 56 points.

3.5.1.2 Results

A 2-D plot (Figure 29) shows how much of the errors are at different locations inside the projection area. The error is defined as the Euclidean distance between the computed coordinates from Pilot and the ground truth. It is clear from this figure that the errors are small and the accuracy of the localization technique is high. The empirical cumulative distribution function (CDF) for this experiment is also illustrated in Figure 30. For 90 percent of these points, Pilot has achieved a localization deviation less than 2.875 millimeters. For the whole projection space, Pilot achieved an average error of 1.707 millimeters, and the standard deviation is 0.918 millimeters. This demonstrates that Pilot achieves its goal of sub-centimeter device localization in practice.

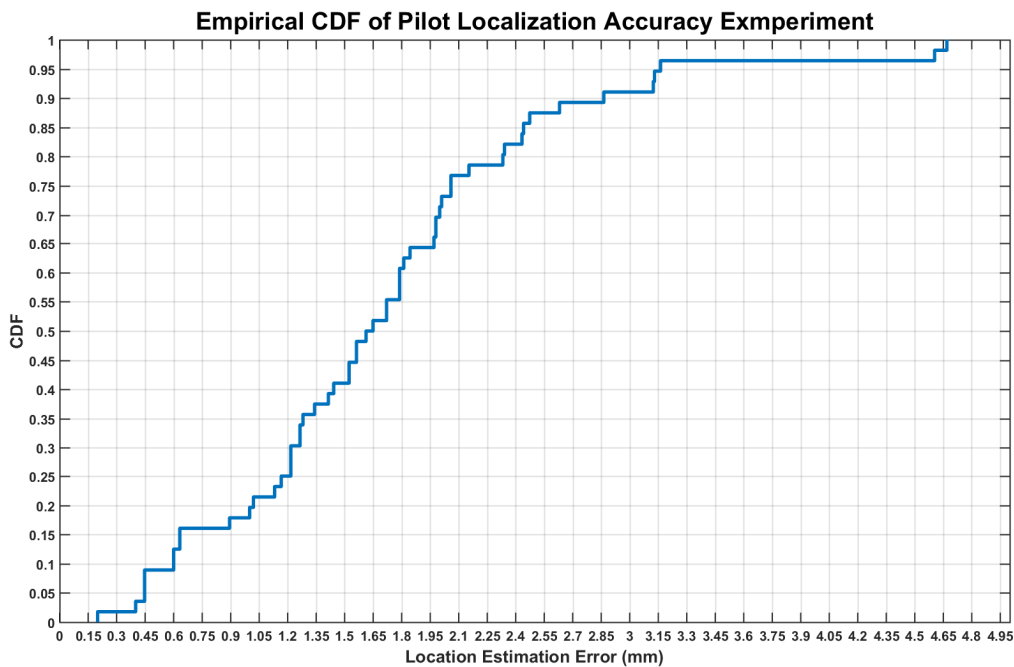


Figure 30 The empirical cumulative distribution function of localization error for 56 discrete points inside the projection.

3.5.2 Localization Latency

Another fundamental question that we aim to answer in this thesis is: *can we use Pilot for real-time location-based applications and services?* We should note that the latency of Pilot comes from multiple levels. Foremost, the visible light packet itself is 47 frames in our current implementation, and 250 microseconds for each frame. This is 11.75 milliseconds long in total. Further, the microcontroller adds a certain delay while decoding the collected light intensity, restoring the original physical coordinates, and then packing position data for transmission. Finally, to make Pilot portable and compatible with existing mobile devices, we chose WiFi for communication between Pilot and target devices in our evaluation. Open Sound Control (OSC) [45] is also used as the data transmission protocol because of its simplicity and robustness. The WiFi connection is the communication bottleneck of our current implementation because the WiFi bandwidth is shared between the Arduino board in Pilot with other devices in the testing environment, including desktops, laptops, mobile phones, and a variety of other wireless devices. Therefore, we designed the following experiment to measure the latency of the proposed system.

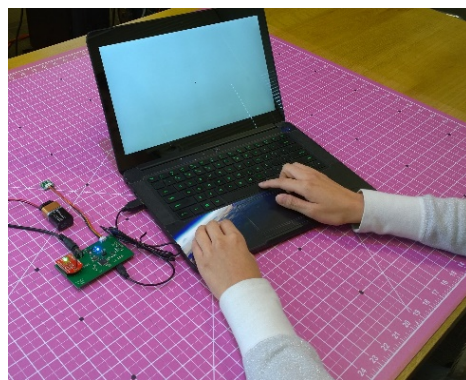


Figure 31 Experiment setup to evaluate system latency of Pilot.

3.5.2.1 Experiment Design

We used a simple setup to measure the delay between the time when Pilot receives a “start” command for position decoding and the time when a laptop (2.2 GHz CPU, 8 GB memory) receives the decoded position through WiFi connection and applies the homography to the sensor readings to get the final position. Figure 31 demonstrates our setup for this experiment. A program is developed to run on the laptop recording a timestamp of a keypress event from the keyboard. Whenever the user presses the “space” key on the keyboard, the corresponding timestamp is recorded by the program, which also notifies Pilot to start the position detection process through serial communication running at 115200 baud. In turn, Pilot will send the detected position data to the same laptop through WiFi connection once it finishes data processing. The time interval between this program detecting the keyboard event and it receiving a position package is considered as system latency.

Table 2 Data Transmission Latency for Pilot.

Communication Channel	Transmission Latency (millisecond)
WiFi	31.46
Serial	23.2

3.5.2.2 Results

Our evaluation consisted of two phases. First, we transmitted location data from the sensor unit to the host PC using WiFi connection. We then ran the second phase of this experiment by repeating this process, but this time we used serial communication to collect position data from Pilot to the host PC. Six participants were recruited from our lab to collect 1200

groups of time difference measurements, and an average latency of 31.46 milliseconds was reported when using WiFi connection, as seen in Table 2. Whereas, the delay decreases to 23.2 milliseconds when serial communication (115200 baud) is used.

3.5.3 Localization Refresh Rate

Another research question we would like to answer in this paper is: *does Pilot support mobility?* An important feature of location-based service, especially within the *Fog Computing* paradigm, is that objects equipped with sensors could be moving all the time. Therefore, *service continuity* should be supported in various scenarios. At the localization level, the refresh rate of Pilot plays an important role to ensure location-based services are available at all times. Six participants were invited for the following evaluation to find out the number of positions that Pilot can detect at different movement speeds.

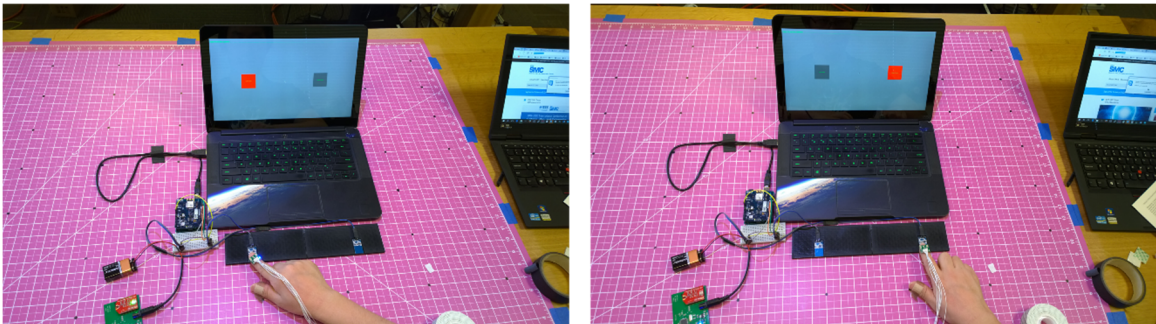


Figure 32 Experiment setup to evaluate localization refresh rate in Pilot.

3.5.3.1 Experiment Design

Figure 32 demonstrates the setup of this evaluation, and the procedure to conduct this experiment is detailed as followed. All six participants were asked to situate a sensor unit on one of their fingers and touch two touchpads in a fixed order (left one first, then right

one). They were also asked to perform this action at different speed modes: *normal*, *medium*, and *as fast as possible*. The time when these two touch sensors are activated were used as timestamps indicating the beginning and ending of a single test, and the time difference of clicking these two pads consecutively and the physical distance between them were used to calculate the movement speed of user's finger. To make our experiment as accurate as possible, the participants were given a demonstration of the system first before being allowed to practice with it. Once they were familiar with the procedure, we started the experiment.

The distance between these two pads is 20 centimeters. This distance is chosen to be long enough so that participants must move their fingers for a measurable period of time, but not so long that they have to move or stretch their body to access the second touchpad, which obviously will decrease their movement speed and affect the experiment accuracy. The time interval between two touch events is detected by another microcontroller (Arduino Uno) and sent to the laptop through serial communication (115200 baud) for data logging. A program running on the same laptop is used to count the number of position packages during this time interval.

We repeated the above set of experiments 10 times for each speed case. Participants could use any finger for the touch action, but they must use the same one during a single test. Thirty tests were performed for each user, totaling 180 tests for all six participants. Participants were allowed to take a rest and change to another finger between the tests.

3.5.3.2 Results

Figure 33 illustrates the tracking performance of Pilot with different moving time, varying among 955.504 milliseconds, 547.692 milliseconds, and 435.105 milliseconds, respectively.

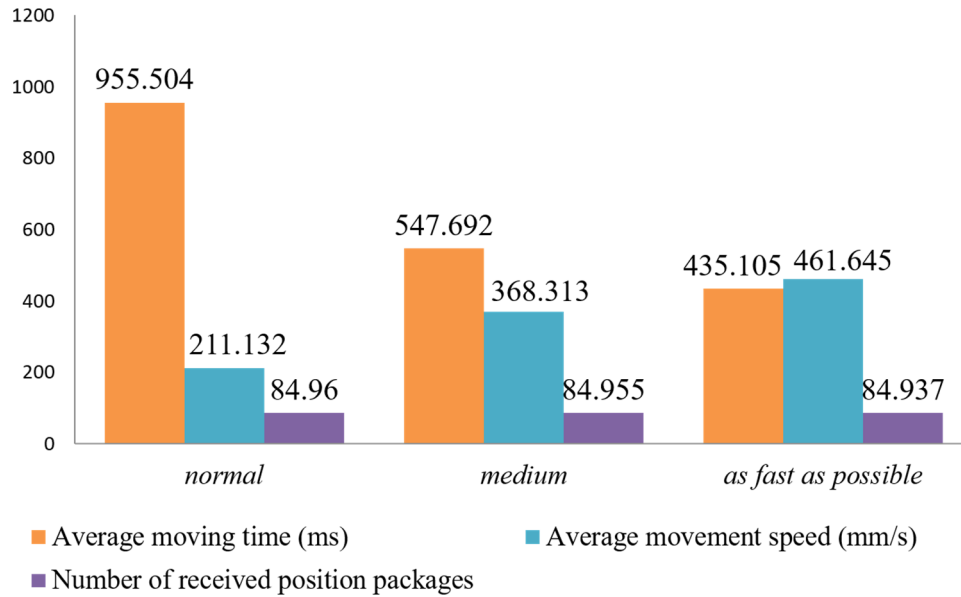


Figure 33 Tracking refresh rate of Pilot at different movement speeds.

Figure 33 also shows average movement speeds calculated by the following equation:

$$V = \frac{\text{the distance between two touchpads (20 cm)}}{\text{the time interval between two touch events}} \quad (33)$$

which are 211 mm/s, 368 mm/s, and 461 mm/s, and the average number of position packages received at these three different speeds is 84.96, 84.955, and 84.937. Each package contains the position data sensed by the sensor unit on the user's finger. This agrees with the theoretical maximum refresh rate of Pilot, which is 85.1 Hz and decided by the total

number of patterns the projector can send every second and the length of a position packet.

This quantity can be calculated simply by this equation:

$$r = \frac{4000(\text{total number of patterns per second})}{47(\text{number of patterns for a single position})} \approx 85.1 \quad (34)$$

This setup simulated a scenario where mobile devices have spatial movement while sensing and collecting information. Our findings underscore that Pilot can maintain a high refresh rate even when the target devices are moving at a high speed, indicating that our proposed localization technique can meet this challenge and provide continuous location-based services without disconnection.

3.5.4 Ambient Light Robustness

Since Pilot is visible light driven, we investigated its behavior and robustness to ambient light across different lighting conditions. We placed a sensor unit at a *known* place inside the projection area and collected the restored positions generated by Pilot, as shown in Figure 34. We also measured the ambient light intensity around this sensor unit by resting a light meter next to it. The metric used here to quantify the system reliability is the percentage of correct readings generated by Pilot.

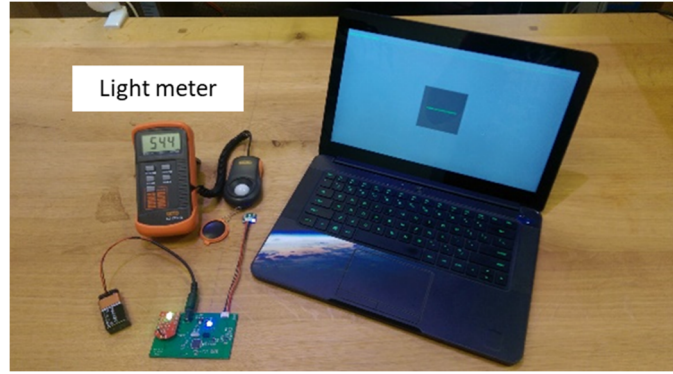


Figure 34 Experiment setup to evaluate system robustness.

As shown in Table 3, we collected eight groups of sensor readings, generating a total of 8000 positions. The first column indicates different light intensity of the environment. Since the proposed system is driven by a visible light projector, the projector itself will increase the light intensity of the projection area by a certain range. Thus, the second column in the table illustrates the light intensity of the projection space after the DLP projector is turned on. The third column logs the result from this experiment, showing the percentage of accurate position readings at different lighting setups. This experiment revealed that our localization technique can work reliably when ambient light is less than 345 Lux. However, if the environment is brighter than this threshold, Pilot fails to work properly. We will explore how to solve this problem in Section 3.7.

Table 3 Localization Accuracy under Different Conditions.

Ambient Light Intensity (Lux)	Ambient Light + Pilot Intensity (Lux)	Accuracy Percentage (%)
232	384	100
312	444	100
336	464	98.7
345	472	94.4
349	482	22
355	487	0
400	538	0
483	614	0

3.6 Comparison with Other Localization Systems

Finally, we also compared the experiment results obtained by our prototype implementation to those from state-of-art systems in the literature. Note that the way indoor localization systems are evaluated and compared can be rather tricky. Previous studies have shown that a plethora of factors may affect performance of a particular indoor positioning technology during system evaluation [46], such as setup of evaluation environment, evaluation points selected in a given environment, and the amount of time that has been devoted to developing the system under investigation. Since one of the goals of this work is to promote high-accuracy, low-latency indoor localization technology for low-power devices, we propose the following evaluation criteria for such scenario: accuracy, precision, and hardware complexity.

Accuracy is arguably the most important requirement for positioning systems. Specifically, *mean distance error*, which represents the average Euclidean distance between the true and reported coordinates of selected evaluation points, has been adopted as a performance metric by many existing studies. One good example is the Microsoft Indoor Localization Competition 2014-2017 [47], where more than 100 teams from both academia and industry working on indoor localization technology have been invited to deploy their systems in realistic, unfamiliar environments and compete with each other in terms of accuracy.

As those who are familiar with positioning systems will recognize, however, it is not enough to measure the performance of a positioning technique only by observing its

accuracy. To be more specific, the reported coordinate is basically an estimate, which heavily depends on the sample size considered in the evaluation. A few data points that represent inevitable, occasional failures of a localization system can overly influence the value of *mean distance error*. Therefore, we also reported a measure of precision, which demonstrates how consistently a particular system works. In other words, it is a measure of the robustness of the positioning system as it reveals the variation in its performance over many trials. Here, the cumulative probability function of the distance error between the estimated and true location is used to illustrate the precision of the systems under consideration.

Moreover, localization system designers often need to address multiple, possibly competing, requirements by balancing tradeoffs in terms of accuracy, precision, and system complexity. Here, system complexity can be attributed to *hardware complexity* and *software complexity*. Given that we do not have access to the source code of the solutions under comparison, we only examined their hardware requirements.

For a typical indoor localization system, hardware components usually consist of mobile nodes, external base stations, and central computing infrastructures. More specifically, mobile nodes are often rested on target devices and could be one of the following: radio transceivers (e.g. WiFi, Bluetooth, ZigBee, or UWB), RFID tags, or light sensors with necessary microprocessors and other circuit elements. Interestingly, modern smartphones have packed a variety of sensors, wireless communication, and desirable computing capability into a small form factor, which encourages many researchers to use them as

mobile nodes in various applications. Although two candidates [48, 49] in Table 4 adopted this method for its convenience and the devices they chose may seem too demanding for some applications, we consider the overall solutions inspiring and could be remodeled to fit more scenarios with a certain effort. Therefore we included them in the comparison list, with an emphasis on the contributing hardware components instead of the smartphone as a whole.

Meanwhile, WiFi access points, RFID readers, and LEDs serving as base stations can be easily found in the literature. Both cloud servers and dedicated desktop computers have been used as central computing infrastructures for location estimation. Given that base stations and backend servers can be connected to the grid without concern for power consumption, we only point them out in the last column for reference.

Closely related to system complexity, building cost represents another dimension of localization system evaluation. However, the cost of a particular system may depend on many factors, including man power, time, space, and energy. Considering that most existing studies did not reveal the details about their deployment overhead and development cost, we think it is not appropriate for us to make any statement on this matter.

Furthermore, in this work we focus our attention on the case of low-computation devices. Indoor localization technologies using computing vision, laser scanners, and pedestrian dead reckoning (PDR) are not considered, given the diversity of the appearance of low-power devices, the size and power consumption of laser scanners, and the requirement of working pedestrians, respectively.

Finally, it can be seen from Table 4 that this evaluation study allows us to closely compare our proposed system with other strategies for indoor localization purpose. Even though these candidates do not cover every single research and industry effort in this field, they are representative of the most promising indoor localization technologies that can be integrated into either existing or future low-power devices. And based on this analysis, we believe we can safely say that Pilot holds significant promise for location-based pervasive applications with appealing accuracy, robustness, and simplicity. As shown in the third and fourth column, Pilot achieves the highest accuracy and precision compared to all other candidates. Further, Pilot is simple in design with the need of only off-the-shelf components: light sensors, entry-level microcontrollers, and DLP projectors installed in the environment.

Table 4 Comparisons with Representative Systems.

Reference	The Proposed System	Xie et al. [32]	Quantitec Intranav [51]	Hammer et al. [50]	Shimosaka et al. [39]	Beder and Klepal [48]	Dahlgren and Mahmmud [49]
Technical Approach	Encoded visible light	Spatially coded infrared LED	UWB TDoA	Sound ToF	ZigBee fingerprinting	WiFi fingerprinting + Bayesian filter	BLE RSSI Trilateration + Particle filter
Accuracy	0.0017m	0.041m	0.168m	1.22m	1.5m	1.56m	2.337m
Precision	95% within 0.00315m	95% within 0.092m	60% within 0.14m; 90% within 0.31m	80% within 1.1m; 90% within 2.4m	95% within 3.38m	Unspecified	28.2% within 2m; 92.5% within 4m
Mobile Node Complexity	A light sensor, a MCU, and a wireless link (optional)	A light sensor, a MCU, and a wireless link (optional)	A UWB transceiver	A microphone, a RF transceiver, and a DSP computation unit	A MCU and a ZigBee module	WiFi connection and signal strength measurement	A cellphone supporting BLE
Infrastructure Complexity	A DLP projector	An infrared LED, a customized lamp shade, a step motor, and a MCU	Eight UWB anchors within 600 m2 and a cloud server	Six loudspeakers, a backend PC, a RF transceiver	Six ZigBee modules within an area of 63.22 m2	2D floor map, ten access points within 300 m2, a backend server	Ten BLE modules within 88 m2

Chapter 4 Robust Indoor Navigation System for Autonomous Robots

4.1 Introduction

Reliable autonomous navigation is critically important when a mobile service robot is used in a domestic or industry environment. Unsurprisingly, robot navigation system designers must address multiple, possibly competing, requirements by balancing tradeoffs in terms of computation, communication, accuracy, and reliability to satisfy design objectives. For instance, system designers might strive to minimize the engineering cost of modifying the environment and the computation requirement of the overall system, but at the same time attempt to maximize the reliability and repeatability of the resulting navigation system. Here, we advocate an encoded projection based topological navigation system for an indoor autonomous service robot by presenting several tradeoffs we have investigated when building a practical delivery robot in a real-world office setting.

4.1.1 Topological Map versus Metric Map

The first trade-off we investigated is between *topological map* and *metric map*. On one end, navigation systems that do not require any environment modification but heavily rely on an environment map can be easily found. Such systems usually include a map building module, which creates a map of the environment using laser scanners [52, 53], cameras [54-56], or both [57]. They would also contain a localization module to determine the robot's position and orientation on the map based on real-time data acquisition. Although a certain degree of accuracy and reliability has been achieved within such systems, map building is computationally expensive and does not scale well to large-scale environments.

At the other extreme, many researchers proposed to reduce the computation complexity and bound localization uncertainty by using pre-deployed landmarks in the environment. One good example of such cases is to place high-contrast tapes on the ground, establishing fixed paths for robot navigation. This method essentially reduces the computation overhead to the minimum because only binary signals (on/off the tape) need to be processed. This approach is proven to be highly reliable and has been used in a variety of industrial manufacturing applications. It does, however, increase the engineering cost and results in a system with little flexibility. Since once these tapes are fixed on the floor, for example, rerouting them would not be trivial.

Interestingly, from many practical use scenarios in literature, we find that some tasks do not need an exact global geometric location of the robot regarding the environment, but rather require an accurate local position relative to known points of interest. Consider an example of delivering small packages inside large buildings, such as airports, warehouses, factories, and hospitals. Directing a service robot from the loading zone to a specific office or a sequence of offices on a routine path without bumping into anyone or entering an office can be sufficiently helpful in modern life as long as the navigation can be performed reliably. Certainly, this can be achieved with the line-following scheme. Though, placing high-contrast tapes on the floor of many public places might be highly unacceptable to their owners.

Meanwhile, a common navigation task is usually specified (or could be) in terms of the qualitative/topological relationship between the robot and the environment. For instance,

when giving a robot directions from the Green Room to the Robotics area on the map in Figure 35, the path could be described by saying, “go down this hallway, make a left at the first intersection, keep walking to the end, then turn right and travel to the end of the hallway.” One can argue that the same directions can be given in a more metric way, such as “go straight for 45 meters, turn left, move forward another 40 meters, then turn right and continue for 20 meters.” Although both are valid directions detailing the same starting point to the same final destination, this is hardly the best way to specify the path from one location to another.

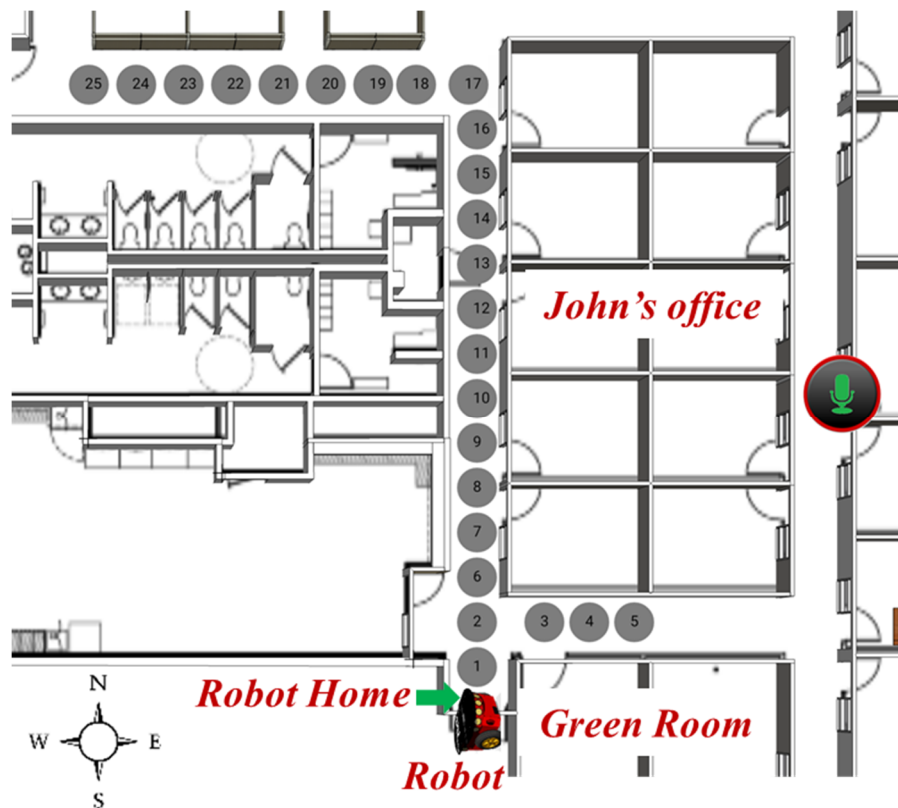


Figure 35. An Android application for robot navigation.

These two navigation examples depict the difference between a topological map and a metric map. We argue it is advantageous if the map used for an autonomous robot can be defined in terms that easily relate to the robot motion capabilities or its available behaviors. A number of previous studies [58-64] have investigated how a topological map based navigation system can be used to facilitate indoor robot navigation.

4.1.2 Active Landmarks versus Passive Landmarks

Another important issue that needs to be considered in a landmark-based topological navigation approach is to decide what type of landmark is suitable for a given environment. Intuitively, landmarks are selected to be easily identified and can be natural or artificial. Natural landmarks are objects or features that can be found in the environment but have their own original functions other than facilitating robot navigation [59, 63, 64]. Limitations of this type of landmark include high variability for different environments and low robustness for recognition. Worse, these features may not always be available for some scenarios. These restrictions make existing features insufficient for safe and reliable navigation.

As a result, previous studies turn to artificial landmarks to allow more accurate localization and navigation. For example, reflective tape [65], color landmark [66], passive RFID tags [67], barcode markers [68], MR-code [69], retro reflective targets [70], and infrared light [71] have all been explored as landmarks.

Artificial landmarks can also be grouped into two classes: passive markers [65-70], which do not consume electricity, and active markers [71], which consume electrical power. Passive landmarks are usually cheaper and easier to scale because no additional wiring is required. However, in order to improve robustness and accuracy of recognition, these landmarks tend to be large in size. On the other hand, active landmarks are costly and require more engineering to the environment but can reduce the computational complexity for landmark localization and provide high reliability. Selecting a suitable type of landmark is the second tradeoff we have studied. In our current prototype implementation, we use low-cost photodiodes as landmarks, and the main motivations are fourfold:

- 1) Reliability
- 2) Speed
- 3) Accuracy
- 4) Appearance

Based on the high-accuracy localization scheme we present in Chapter 3, we have specifically designed networked sensor units using off-the-shelf components which can detect and decode the projected encode light. By placing these photodiode-enabled landmarks onto the ceiling of the environment, we do not need to face the much more challenging task of detecting whatever features happen to be present in the environment, leading to an applicable system in a variety of places. By reversing the traditional camera-landmark approach and instead using a robot-mounted projector and photodiodes, we bound the potential uncertainty of common vision-based systems, such as varying ambient illumination, the resolution and refresh rate of the camera, and also reduce the computation load. In this manner, we have achieved an average accuracy of 8.65 millimeters for landmark localization with entry-level microcontrollers.

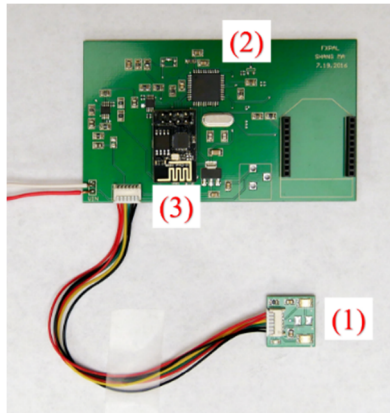


Figure 36. Active landmark for decoding projected light signal.

Moreover, it is important to understand that there are some scenarios where the size and appearance of landmarks matter. Consider, for example, a museum guiding robot which moves along a routine trajectory and presents video clips about artwork to visitors at certain locations. Here, it would not be appropriate to use a plethora of landmarks, like QR-code markers on the floor for two reasons. One, they are not physically appealing and might affect the overall design of the exhibition. Second, with people walking around all the time, they tend to be easily worn down. Additionally, QR-code markers may not attach to the ceilings. Given the location and strength of light sources in most of these environments, ceilings tend to be darker than the walls or the floors, and localization techniques based on these types of markers can be severely affected by ambient lighting, low albedo surfaces, and shadows [72]. But the photodiodes we used are tiny and barely discernible (see Figure 36). They can also be embedded in the ceiling so long as their active area is exposed.

4.2 Our approach

4.2.1 Topological map for global navigation

The effectiveness of topological representation of the environment and the promising resolution of encoded projection based landmarks encourage us to combine them together to build a navigation system that can:

- 1) robustly recognize landmarks;
- 2) accurately estimate the 2D position of the robot relative to a landmark;
- 3) successfully detect the robot has arrived at the region defined by a particular landmark;
- 4) reliably navigate a robot from one landmark to another while keeping track of the robot's progress.

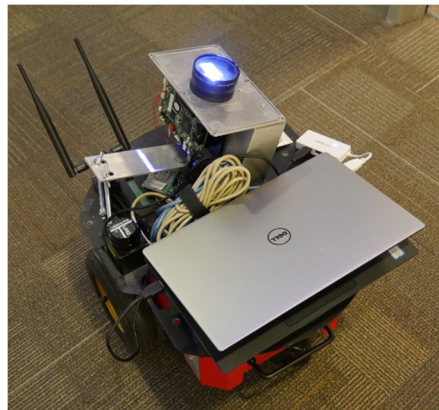


Figure 37. Experiment setup for robot navigation.

As we mentioned earlier, we assume the mobile robot has a clear view of the ceiling. With this, a DLP projector is mounted on the robot to project gray-coded binary patterns to the ceiling (seen in Figure 37). Each pixel in the projection area is individually encoded by the

projected patterns and can be decoded to restore its coordinates inside the projection area at a high frequency. Low-cost photodiodes serve as landmarks to: (1) use the MAC (Media Access Control) addresses of its WiFi module (see Figure 36) as identification (ID); and (2) detect the projected patterns, restore its own coordinates, and send the resulting position to the robot for pose estimation. In this manner, the projector serves a dual purpose, both for activating the landmarks for global localization and for local robot trajectory smoothing.

Users are provided with graphical interface (see Figure 35) monitoring the location of the landmarks and the real-time progress of the robot. Whenever a navigation task needs to be performed, for example, delivering a document from *John's office* to *Green Room*, The user can click on the landmark representing *John's office* on the interface, which triggers the system to generate a feasible path from the robot's current region to this place. The resulting path consists of a list of MAC addresses of the landmarks, which the robot will come across, and a sequence of actions the robot should perform at each landmark.

Further, this generated path will be sent to the robot, which will then move autonomously to each of these landmarks and perform the associated action once it arrives at a particular landmark until it reaches *John's office*, waiting outside. John can put the package on the robot and then navigate the robot to *Green Room* in a similar manner. An example of such paths from *Robot Home* to *John's office* and then from *John's office* to *Green Room* is illustrated in Figure 38. Note that the IDs are shown for clarity and we only transmit MAC addresses and actions in our implementation.

From *Robot Room* to *John's office* From *John's office* to *Green Room*

The initial position and orientation of the robot:		The current position and orientation of the robot:	
Location: 0006669c5617		Location: 0006669c4d67	
Orientation: north		Orientation: north	
The path from <i>Robot Room</i> to <i>John's office</i>:		The path from <i>John's office</i> to <i>Green Room</i>:	
ID: MAC	Action	ID: MAC	Action
1: 0006669c5617	move straight	12: 0006669c4d67	turn around
2: 0006666df2a7	move straight	11: 0006669c48a3	move straight
6: 0006669c4c7a	move straight	10: 0006669c546d	move straight
7: 0006669c4fa3	move straight	9: 0006669c548d	move straight
8: 0006669c4fa0	move straight	8: 0006669c4fa0	move straight
9: 0006669c548d	move straight	7: 0006669c4fa3	move straight
10: 0006669c546d	move straight	6: 0006669c4c7a	move straight
11: 0006669c48a3	move straight	2: 0006666df2a7	turn left
12: 0006669c4d67	stop	3: 0006669c5042	stop

Figure 38. The generated paths to navigate a robot.

4.2.2 Photo-diode Enabled Active Landmarks

Our landmark design includes a photodiode to decode light signals coming from the projector on the robot. This photodiode is connected to an Atmel Atmeg32U4 microcontroller running at 16 MHz (seen in Figure 36). In our prototype, the landmark can detect the projector's location signals from up to 4.5 meters, and the ceiling height in our testing environment is about 2.72 meters. Finally, the software running on the microcontroller collects the projected location signals and restores the position data of the particular pixel where the landmark currently is located. This position data will be sent out to the robot for local PID control and global path planning via a low-cost WiFi module from Espressif [73].

For the encoded location discovery to operate correctly for accurate local trajectory smoothing, the projector must be calibrated to obtain physical coordinates of the landmarks with respect to a world coordinate system, such as the projector on the robot.

Then, the distance and orientation of the robot relative to a landmark in the physical space can be computed accordingly to build suitable PID controllers. As previously mentioned, we currently only consider the scenario where both the floor and the ceiling are level. Point correspondences between the projection plane and the ceiling plane are then used to find the perspective projection matrix, which encodes the relationship between the projection area and the ceiling.

We conducted this standard calibration by firstly fixing the projector onto a robot, then placing a sensor unit at 20 different points evenly distributed within the projection area on the ceiling, and collecting its readings respectively. We also measured the physical coordinates of these points relative to the center of the projection area. The resulting matrix has been applied to all the decoded position data of landmarks in our experiments, described later, to obtain the Euclidean coordinates of a particular landmark with respect to the projector. In actual operation, this calibration could be performed a single time while setting up the robot for a given environment.

4.2.3 Landmark-defined Zone

As shown in Figure 38, the generated path sent to the robot is a list of intermediate landmarks the robot would come across on its path to a goal and associated actions, which the robot would perform at these landmarks. These intermediate landmarks form a path providing the robot with the right directions to reach the final destination. Once the robot receives the generated list, it then follows this path while sending a notification message back to the tablet whenever it has arrived a specific landmark so that users can monitor its

progress on the tablet. Although our proposed encoded projection landmark design is small and fast, the sensing and processing are not instantaneous. For the robot to determine if it is within a landmark zone, we define a circular sub-region inside the projection area as a hot spot, as shown in Figure 39. In this way, once the position data from a landmark have been found to be within the range of this sub-region, the robot can be sure that it has arrived at this landmark zone. Then the robot would notify the tablet and perform the associated action with this landmark, such as *turning left/right*, *turning around*, *moving forward*, or *making a stop for loading or unloading*.

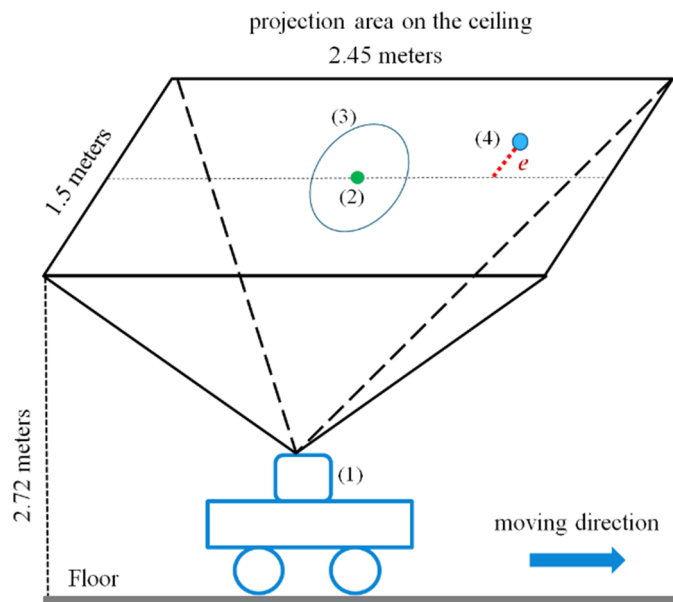


Figure 39. An illustration of the robot navigation with encoded projection.

4.2.4 PID Controller for Local Trajectory Smoothing

In order to control the smoothness of the robot motion, we have implemented two PID controllers in our system: one for heading control, and the other for rotation control.

The first PID controller is implemented based on two facts. Firstly, the robot we used for prototyping uses a differential drive. Secondly, the projector has been aligned well with the robot so that the relationship between the robot and the projection area is fixed, and the longer side of the projection area is in parallel with the straightforward moving direction of the robot, as shown in Figure 39. With this setup, whenever the robot receives position data from an upcoming landmark, its relative physical distance away from the perpendicular bisector in the projection, error e in Figure 39, will be computed based on the homography calculation and be used to adjust the robot heading in real-time.

We use a second PID controller to rotate the robot when turning is necessary. Whenever rotation is needed, the robot would rotate itself either to the left or right until the position data from the upcoming landmark indicates that the robot heading has become straight. It is important to note that although turning around is always 180 degree, turning left/right is not necessarily an exact 90 degrees. As shown in Figure 35, due to limitations of the environment, landmark #17 is not installed straight to the north of landmark #16. A robot coming from landmark #16 will adjust its heading to the left in order to reach #17 with the first PID controller ending up with an angle of about 10 degree West. If landmark #18 is the next target, the robot would need to make a left at this point with a rotation angle of 80 degrees. In the experiments, our robot can successfully achieve this by simply rotating until landmark #18 is straight ahead of it. It is also worth mentioning that all the actions associated with landmarks are only performed when the robot is already inside a landmark zone.

4.3 Robot Navigation Robustness Evaluation

To demonstrate the capabilities of the proposed navigation system and evaluate its performance in a real-world environment, we conducted a series of navigation experiments in a real-life environment (see Figure 1). Notice that the testing environment is a typical office setting. It is basically symmetric and possesses glass doors, walls, and various places that look alike, making it difficult for a robot to perform accurate navigation with some other technologies [53], such as laser scanners or vision-based systems.

We used a Pioneer 2DX robot from Adept [74] and equipped the robot with a DLP projector facing upwards. We also put a laptop (2.2 GHz CPU, 8 GB memory, ROS Kinetic, Ubuntu 16.04) on the robot to process navigation tasks from users and to control the robot motion based on the real-time position data from landmarks. An Android tablet was provided to the users, allowing touch interaction and voice recognition for task assignment. Both the communication between the tablet and the robot as well as the communication between the landmarks and the robot was completed over the WiFi network in our laboratory. Open Sound Control (OSC) [45] is used as the data transmission protocol. Figure 37 depicts the setup of these experiments.

Given that the robot does not have access to any IMU sensors or a compass, the initial position and orientation of the robot was explicitly set. In this experiment, we placed the robot facing north at the zone defined by landmark #1, called *Robot Home*, as shown in Figure 35. In spite of this, anytime in the middle of the experiments, the robot can be aware of its orientation by keeping track of the rotations it has already made. The Android

application was used to drive the robot among these landmarks using our proposed navigation method. A Hokuyo's URG-04LX-UG011 laser scanner (see Figure 37) was used to evaluate the outcome of the proposed navigation system by measuring the distance between the robot and one side of the hallway (all glasses on the other side and laser scanners fail to work properly.). A calibration setup, as described previously, was conducted ahead of time to calculate the transformation matrix between the projection plane and the ceiling of our lab.

4.3.1 Landmark Position Estimation Accuracy

To evaluate the position estimation accuracy of the proposed encoded projection location discovery for each landmark, another set of 18 points (different from the 20 points used for homography calculation) inside the projection area on the ceiling were selected to 1) collect their pixel coordinates in the projection area using a sensor unit and apply the calculated transformation matrix to them; and 2) measure their physical positions as ground truth. The difference between these two quantities is considered as the landmark localization error. A 2D plot is used in Figure 40 demonstrating the error for these points. A mean error of 8.65 millimeters for landmark localization was reported, and the standard deviation was 7.3 millimeters.

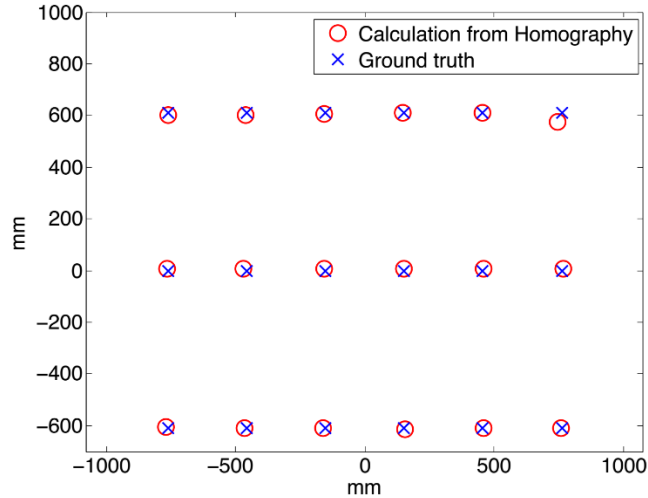


Figure 40. Landmark position estimation error for 18 points.

4.3.2 Navigation Accuracy for Long Paths

We also evaluated the navigation accuracy of the intended system by deploying 25 landmarks on the ceiling in our lab and repeatedly navigating the robot along the formed path. The calculated homography was applied to the received position data from each landmark. The distance between the robot and the wall were measured by the laser scanner and compared to the ground truth, which is the physical path formed by all the landmarks. Additionally, as shown in Figure 35, the landmark #17 is at the corner where a rotation is required. This makes the distance measurement complicated. Therefore we only collected the data from landmark #7 to #16 and landmark #18 to #25 to evaluate the performance of our navigation scheme. The robot was driven along this path back and forth continuously for 20 times at a speed of 25 *cm/sec* . We show both the ground truth and the robot's real trajectory in one navigation test from landmark #7 to #25 and vice versa in Figure 41 and Figure 42, where ground truth is formed by the physical landmarks (red), and the robot's real trajectory is color-coded in blue.

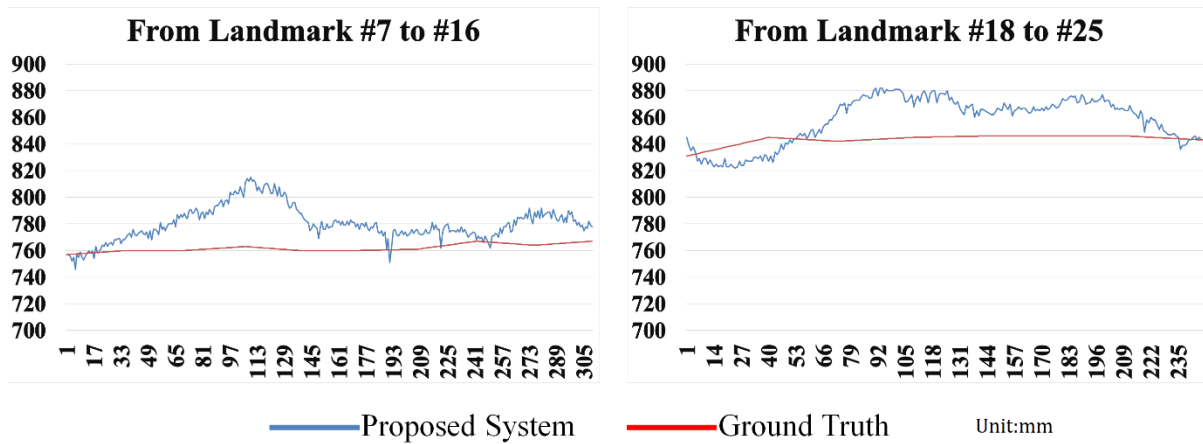


Figure 41. Experiment result of one navigation test from Landmark 7 to 25.

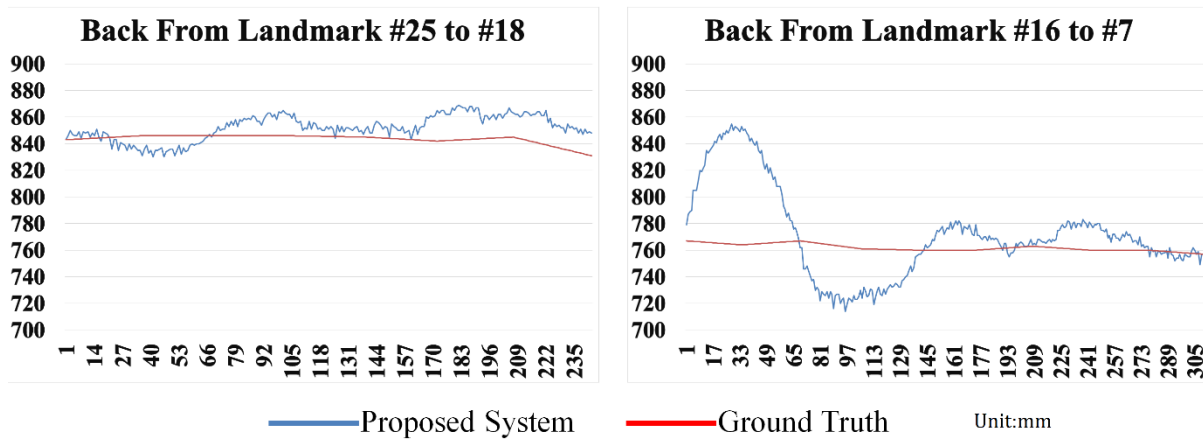


Figure 42. Experiment result of one navigation test from Landmark 25 to 7.

Figure 43 illustrates the empirical cumulative distribution function for all 20 navigation tests. For all tests, the robot can successfully recognize the landmark zones, make necessary rotations, and eventually reach its goals. The average navigation deviation was 18.5 millimeters with a standard deviation of 24.39 millimeters. The measured maximum deviation of the robot from the desired trajectory was 90.45 millimeters, which occurred when the PID controller tried to adjust the robot's heading around landmark #16.

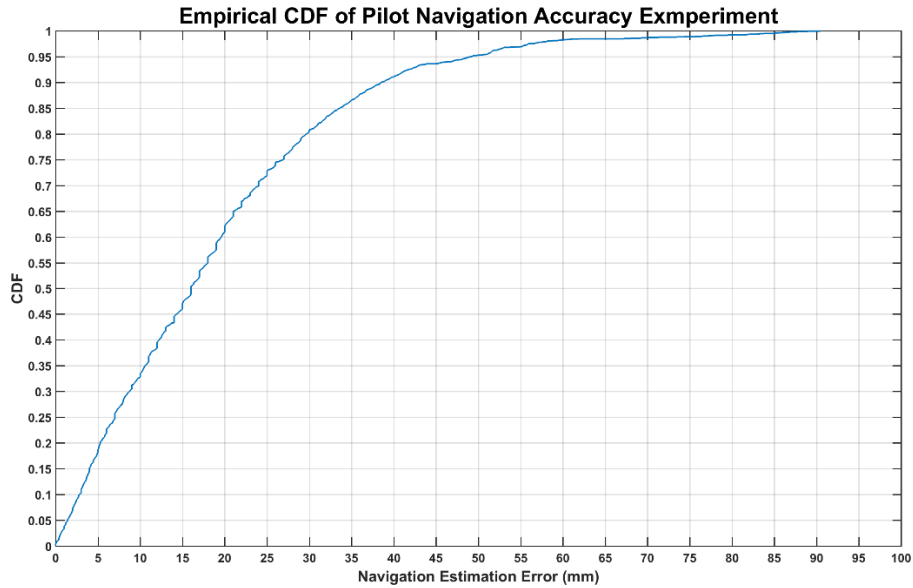


Figure 43. The empirical cumulative distribution function for the navigation error of the proposed system from all 20 tests.

4.3.3 Comparison with Existing Work

Similarly, we compared the experiment results obtained by our prototype system to the ones from other systems in the literature. Note that it is difficult to compare all existing robot navigation systems due to the difference of underlying technologies and target applications, not to mention the impact of infrastructure deployment. For instance, the number of access points used in a WiFi-based location system would significantly affect the system performance. We therefore only excerpt the performance of the cutting-edge systems we can find and summarize our findings in the following table.

Table 5 Comparisons with Representative Navigation Systems.

Technical Approach	Reported Accuracy	Frame Rate	Maximum Receiving Range
Stargazer [75]	2 cm	20 Hz	6.5 m
NorthStar [76]	5cm - 15cm	10 Hz	6 m
The proposed system	1.8 cm	85 Hz	18 m

4.4 Discussion

Although our results are preliminary because they were obtained in a relatively small scale, they are still promising. It is important to recognize that classical positioning technologies based on wireless radio signals, computer vision, or laser scanners can greatly suffer from multipath reflection, environmental similarity, or heavy computation. Nonetheless, our experiments are in a typical office setting that shows the proposed system can provide precise and reliable navigation without metric measurements and other types of sensors, such as odometry or inertial sensors. Some important properties of the implemented system have been brought to light.

4.4.1 Computation Efficiency

The encoded projection-based location discovery scheme is computation-efficient. The robot in the system only needs to unpack the received message to obtain positions of activated landmarks, control its local motion between two landmarks, and determine if it has arrived at a landmark zone. Considering that autonomous navigation is a fundamental function of mobile service robots, our design significantly reduces the computational complexity of such tasks so the robot can focus on its application-specific missions. Although we are currently using a visible light-based DLP projector, leveraging an infrared projector with proper signal modulation would significantly reduce the power consumption of the projection system and increase the signal-to-noise ratio. We will leave the development of such modifications for future work.

4.4.2 Cold Start Problem

The photodiode-based landmarks are pre-deployed on the ceiling, forming a topological presentation of the environment. Although a map-building stage can be conducted beforehand by a navigator driving the robot to collect the topological relationship between consecutive landmarks automatically, we manually input this information into our system with a configure file, and the Android application will load this file and build the map rapidly and spontaneously.

4.4.3 Semantic Landmark Placement

Considering that our system only requires a qualitative map of the environment, the deployment of the landmarks can be flexible. We advocate a strategy of placing landmarks at semantically meaningful places, such as “*kitchen*” or “*printing room*” so that later on users can complete navigation tasks by saying these keywords to the robot or the mobile application. Furthermore, the topology can be updated whenever rerouting or covering a larger area is needed by relocating or adding landmarks.

4.4.4 Minimum Mutual Communication among Landmarks

In contrast to sensor network-based robot navigation systems [77], all landmarks in our system work asynchronously, sending position data to the robot without the knowledge of other landmarks. This design removes the burden of synchronization between landmarks. Also, a landmark only sends out data when the onboard photodiode detects the projected light. This means that when the robot is moving, only the landmarks inside the projection area are active and taking up the bandwidth while others remain asleep.

4.4.5 Obstacle Avoidance

The robot does not need to access GPS, IMU, a compass, odometry, or other range/bearing measuring equipment. Both global path planning and local trajectory smoothing rely on an encoded projection based location discovery mechanism. Still, people may move around inside a target environment. Therefore, reactive obstacle avoidance using sonar range sensors or infrared distance sensors can be utilized to stop the robot in front of humans.

4.4.6 Sensor Deployment Strategy

An important point to be addressed is the optimal placement of the landmarks when deploying the proposed navigation system in a real-world environment. This problem can be further divided into two sub-problems: 1) how many landmarks are needed; and 2) where these landmarks should be placed so that the robot can perform any foreseeable tasks.

In our prototype system, the projector gives us a projection area of 2.45 by 1.5 meters, as shown in Figure 39. To guarantee that the robot can always move from one landmark to another without using any other sensors, we have placed our landmarks in a way that the projector can always see at least two landmarks. In other words, the distance between two consecutive landmarks is about 1.2 meters. Although simple, this deployment scheme increases the system cost because it needs more landmarks. Surely, we can loosen this restriction by increasing the distance between landmarks and only requiring one landmark on the scene. This is partially valid because, instead of always using the upcoming landmark for robot heading control, we could also use a landmark the robot recently

passes. However, if the trajectory in front of the robot is not straight, for example when rotation is needed at corners, landmarks should be installed in such special places.

The restriction can be additionally relaxed if other types of sensors are available, such as odometry, obstacle avoidance sensors, or laser scanners. These sensors can be leveraged to navigate the robot between landmarks even when no landmarks can be seen by the projector, but this would increase the uncertainty of the overall system.

4.4.7 Multi-robot Coordination and Navigation

In our current experiment, we asked the Android application to send the navigation tasks to the robot directly and configured all the landmarks to send its decoded positions to the robot as well. However, a back-end server can be added to this architecture to allow the coordination between multiple robots and multiple user applications. More specifically, multiple users can navigate several robots simultaneously through their own GUI applications, which could run on desktop computers or mobile devices. These applications will send the user's navigation tasks to a mediate server, which would assign different missions to different robots. All the robots will also send their real-time global location back to the server so the server can monitor all the robots simultaneously and intelligently perform global path planning to avoid situations where two robots bump into each other at an intersection. This server will also update the user's GUI applications, if necessary, so they can monitor the progress of their assigned robot. We will have a thorough discussion on this subject in the remainder of this work.

Chapter 5 Multi-robot Motion Coordination

5.1 Introduction

The multi-robot motion coordination (MRMC) problem is gaining increasing attention because many practical applications of autonomous robots require the use of multiple team members. The most common motivations for developing multi-robot system solutions are that:

- 1) The task complexity is too high for a single robot to accomplish;
- 2) The task is inherently distributed;
- 3) Building several resource-bounded robots is much easier than having a single powerful robot;
- 4) Multiple robots can solve problems faster using parallelism;
- 5) The introduction of multiple robots increases robustness through redundancy.

A critical issue in these mobile robot teams is coordinating the motions of multiple robots interacting in the same workspace. Regardless of the mission of the robots, they must be able to effectively share the workspace to prevent interference between the team members. Solutions to the motion coordination problem are approached in a variety of ways, depending upon the underlying objectives of the robot teams. In some cases, the paths of the robots are explicitly planned and coordinated in advance, as might be needed in a busy warehouse management application, for example. In other cases, planning is relaxed and emphasis is placed on mechanisms to avoid collision, applicable for tasks such as automated hospital meal deliveries. In yet other situations, the robots could have mechanisms with little pre-planning that focus on coordinating robot motions in real-time

using reactive, behavior-based, or control-theoretic approaches, such as those used in a convoying or formation-keeping application.

5.2 Related work

5.2.1 Continuous Space Representation

Motion coordination for a single robot on a map requires finding a collision-free path from an initial position to a destination place. This single robot path planning problem can be effectively solved by the A* algorithm when an optimal path or solution is required. Though the search space is bounded by the size of map, the computational cost for the algorithm like A* to find an optimal path can be high when the map is large. In contrast, multi-robot motion coordination is a more challenging problem. By planning the paths for all mobile robots simultaneously on the same map, multi-robot motion coordination can be considered as a single compound robot coordination in a composite search space, which is a Cartesian product of the search space for each robot. As a result, finding an optimal solution in MRMC can become intractable as the number of robots increases and has been proved to be NP-hard [78].

Existing MRMC approaches basically fall into two classes: *centralized coordination* and *decoupled coordination*. The first one is pretty straight-forward. The idea is that all m robots in the scene are treated as one single composite robot with m independent bodies that are not necessarily connected to each other, hence transforming the multi-robot problem into a single-robot one. In this way, the standard motion planning methods, such as, cell-decomposition, can then be used to find a path in the configuration space of the

composite robot. However, this compound robot has a degree of freedom equal to the sum of the degrees of freedom of all individual robots. This implies that the dimension of its configuration space is usually large, and, therefore, the time complexity and memory need of centralized coordination methods tend to be high [79]. And the collision of, for example, l robots ($l < m$), can be considered as a self-collision of separate parts of the compound robot.

Decoupled methods normally include two stages: firstly, they plan the paths for each robot independently while ignoring the existence of other robots; then in a second stage, they try to coordinate these paths in a way that mutual robot collision can be avoided. This scheme significantly reduces the computation complexity, therefore, decoupled coordination can handle problems with large number of robots and is implemented more extensively in the literature. Among them, collision between robots can be avoided through two techniques: prioritizing the robots and velocity tuning.

The technique of prioritizing the robots assigns priorities to each robot based on the properties of the targeted application itself. Then the moving paths of these robots will be computed in the order of their priorities [80]. A fully distributed cooperative path planning scheme is proposed in [81], where dynamic conflicts between robots can be solved by the heuristic adjustment of priority values. [82] also applies a heuristic algorithm to assign higher priorities to robots than can move to their goals directly. [83], however, utilized a randomized hill-climbing search to perform priority assignment while at the same time using A* algorithm to calculate the optimal path for each robot. Another method that can be used to

resolve collision is velocity turning, in which, each robot can be forced to stop or decelerate its motion to avoid colliding with other robots [84].

5.2.2 Graph Representation

The methods we have described above deal with multi-robot coordination in a continuous space, where robots can freely move among any points on the map. Although having many use cases in realistic scenarios, these methods suffer from high computation requirement when the amount of robots in the target system is large. A scheme that have been proposed by some researchers to overcome this issue is to downsize the configuration space in a way that the entire workspace can be modeled as a tree or a graph, also known as a topological map as we show in Chapter 4. By doing this, the original space is discretized into roads and stops, which correspond to edges and vertices in the graph. In this way, the overall computation can be reduced and many existing graph algorithms can be utilized for multi-robot motion coordination.

Another advantage of this scheme is that the graph representation guarantees that all the robots will only move on the predefined routes (the edges on the topological map), and stop at certain points (the vertices on the topological map) performing necessary actions to complete their paths. Therefore, the MRMC issue can be turned into an ordering problem where all the robots are coordinated to move toward their target either sequentially or concurrently, without collision with existing obstacles in the environment or other robots.

This graph representation scheme has inspired a number of algorithms of motion coordination for a group of robots on a topological map. [85] introduces predefined subgraphs such as stacks, halls, rings, and cliques, which turns the MRMC problem into the question of reaching destinations of connected subgraphs. A multi-phase algorithm is proposed in [86] to take advantage of the spanning tree representation of the environment to plan paths for a number of robots by moving all the robots to the leaves of the spanning tree first. Although this algorithm is complete, the maximum number of the robots that can be controlled is less than the number of the leaves in the resulting spanning tree. Due to the fact that the construction of the spanning tree for a given graph is not unique, the performance of this proposed algorithm varies. Besides, other structural features in a graph have also been utilized to develop effective algorithm for MRMC problems. For example [87] takes advantage of the grid structure, while [88, 89] proposes macros for efficient path planning on θ graphs.

5.3 Our Contribution

Although the multi-robot motion coordination problem has received a considerable amount of attention in the recent years, the following two problems, however, have not been investigated in prior research:

- 1) What types of structures a general graph should have to be “reachable” for that maximum amount of robots?
- 2) What is the maximum number of robots a given topological graph that represents a specific environment can accommodate while it is still being “reachable”?

Here, we use the notion “*reachable*” to represent that for a given graph and a given number of robots, these robots can be navigated from *any* initial configuration to *any* final configuration. And a configuration is the arrangement of these robots on the vertices of the topological graph such that one robot only occupies one vertex and all robots should have their own vertices.

Both graph representation of the environment and these two questions have real-world applications especially in the scenarios we have discussed above, where a group of indoor autonomous robots can be taken advantage of to achieve various tasks in a workplace with corridors. Considering that operations performed by robots need to be done as efficient as possible, the underlying MRMC system should always be able to find feasible solutions to navigate all the robots from their current positions to the desired targets without human intervention. Further, given that in most realistic scenarios, the cost of building such a system is a big concern for system designers, the two questions listed above should be answered as early as possible before deploying the whole system.

Following the graph representation scheme, we assume that a given workplace, such as warehouses, office settings, hospitals, can be divided into vertices. And each vertex can be occupied by a robot or empty without a robot, which we refer to a *hole*, throughout this work. A group of robots will be accepting user requests to move from their current places to new places (either holes or nodes occupied by another robots). Since user requests can be hard to predict, this requires any configuration of these robots can be reached from any

initial configuration through navigating the robots along the edges on the topological map without colliding with obstacles or other robots.

As one of the general data structures, there are two types of graphs: acyclic graph and cyclic graph. Acyclic graphs, also known as trees, do not have loops, therefore providing less flexibility to move robots around. Cyclic graphs, on the other hand, contain loops. In this chapter, we first investigate tree-type (acyclic) graphs, which serves as a basic for multi-robot motion coordination problem and then graphs with cycles, both of which can be extended to more general topological maps. Regarding acyclic tree-type graphs, we start our investigation from simple data structure, *Chain*, which only consists of a sequence of vertices. Then, we will discuss *n-Star* and *Extended n-Star*, both of which build the foundation of an important concept of this chapter, namely *Reachable Territory*. Based on this, we will investigate:

- i. What kind of topologies a general acyclic graph must have to be reachable?
- ii. For a given tree, what is the maximum number robots that can be moved from *any* initial configuration of their position to *any* target configuration without collision?

Further, we will investigate how multi-robot motion coordination can be extended to graphs with circles. This will be started by introducing several small graphs, including *Unicyclic Graph*, *Pan Graph*, *Tadpole Graph*, and *Extended Tadpole Graph*. Based on the study on these small graphs, we will extend our findings to general graphs with circles and ask the same questions as we do for acyclic graphs.

5.4 Assumptions & Definitions

5.4.1 Assumptions

As we discussed above, in lots of applications, a topological representation has various advantages over a geometric one, for example low computation cost, providing semantic meaning to both robots and users. Therefore, properties of a given topological map, such as the number of vertices, the degree of vertices, and the existence of edges between any pair of vertices are important to the MRMC problem. For the rest of the chapter, we would like to have our investigation built on top of the following assumptions:

- 1) The graph is a topological representation of an environment where a group of robots will be accommodated. Therefore, the graph is finite, planar, and connected.
- 2) The robot will move on the edges connecting two vertices and can only stop or change its orientation at the positions of vertices.
- 3) All the robots will start from some vertices (*initial configuration*) and will end at some other vertices (*final configuration*) based on user's navigation requests. One robot can only occupy one vertex and one vertex can only accommodate one robot at a time.
- 4) One edge can be only occupied by one robot at any given time.
- 5) Two robots cannot swap their positions either on vertices or edges.
- 6) For topological maps where trees are used, there will be no loop in these trees, while for cases where cyclic graphs are used, there will be at least one loop and at least one other node that is not in the loop.
- 7) All the graphs are undirected, because robots can move in both directions.

- 8) A robot can move from its current vertex u to its neighbor v , if v is unoccupied or if the robot at vertex v is moving to a vertex other than u and v , which evacuates v .
- 9) The weight of edges is not considered.
- 10) The terms “acyclic graph” and “tree” are interchangeable in this work.

5.4.2 Definitions

To preserve self-containment character of this dissertation, several graph theory notions that build the foundation of this work are revisited. Here, the standard terminologies from this book [90] are adopted.

Let $G = (V, E)$ be an undirected graph with vertices V and edges E . The number of all the vertices, denoted by $|G|$, is the order of G . The degree of a vertex u , which is represented by $d(u)$, is the number of vertices that have been connected to u . Leaves are vertices whose degree is 1, and internal vertices are those with degree of 2. The distance between two vertices u and v is denoted by $dist(u, v)$, representing the length of the shortest path between them or the number of edges between these two vertices.

Definition 1. A *Reachable Graph* (acyclic or cyclic) for m robot, RG^m , is the one on which *any* configuration of m robots can be achieved by navigating the robots from *any* initial configuration of these robots in a way that no robot collides with other robots.

5.5 Acyclic Graph

5.5.1 Chain

A chain, as shown in Figure 44, represents a narrow corridor on the topological map. It is one of the simplest subgraphs in real environments. Formally it consists of a chain of vertices, each linked only to its predecessor and/or its successor.

A configuration of a chain corresponds to an ordering of the robots that reside in it. Chains are narrow, therefore robots in the chain cannot pass each other, and so the ordering cannot be changed without the robots exiting and re-entering the chain.

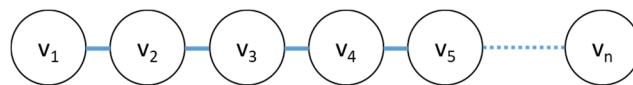


Figure 44. The illustration of the subgraph structure, Chain.

5.5.2 n -Star

Junctions on trees are vertices whose degree is larger than two. An n -Star is one type of trees which have one junction vertex, n leaves, and no internal vertices. Figure 45 shows an example of a 6-star with 6 leaves, where v_1 is a junction

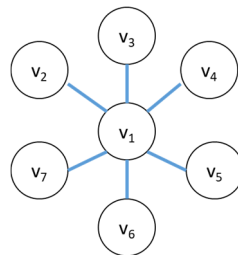


Figure 45. The illustration of the subgraph structure, Star.

Apparently, with the help of a junction, robots from different leaves can swap their positions. For example, robots at v_2 and v_4 on the star in Figure 45 can swap their positions by first moving one of them to v_3 through the junction, assuming both the junction and v_3 are not occupied by other robots, and navigating the second robot to the original place of the first one. And then the first robot can be moved to the original position of the second robot through the junction as well.

In this example, robots from different leaves are interchanged by moving them through the junction. Meanwhile, other leaves, v_3 in the previous example, can be used for situating the robots temporarily during the multi-robot coordination tasks so that the desired path for moving the target robots can be clear.

Lemma 1. m -Star, a star type of trees with m leaves, is reachable for at most $m-1$ robots.

Proof. To begin with, for a star with m leaves, there are $m+1$ vertices in total including m leaves and one junction. Let us consider the case where there are two holes (vertices without robots), because if a star can accommodate $m-1$ robots, it will certainly be able to accommodate any amount of robots less than $m-1$. For a star with $m+1$ vertices and $m-1$ robots, the possibilities of the combination of an initial configuration and a final configuration for all the robot can be found in one of the four cases in Table 6.

Table 6 Potential Coordination Tasks for Robots on a Star

Final Configuration Initial Configuration	The final positions of all robots are leaves	The final position of one robot is the junction, and the rest are leaves.
All robots start at leaves.	Case A	Case C
One robot starts at a junction, and the rest start at leaves.	Case B	Case D

Case A: In this case, all the robots have leaves as their starting points and they have been asked to move to another leaves. Since there are only $m-1$ robots, two of all the vertices will be holes including one of the leaves and the junction. Therefore, each of the robot can be navigated to its destination through this empty junction if the destination is empty as well. Otherwise, if the destination is not empty, the coordination scheme can

- 1) First move the robot at the destination to the empty leaf;
- 2) Then move the original robot to its destination, which has been cleared by the first step.

Following this method, the coordination scheme would be able to move all the robots to their destinations one by one, which means that the whole star is reachable according to our definition in Section 5.4.2.

Case B: In this scenarios, one of the robots starts at the junction at the center of the star structure, while others start from the leaves. And the destinations of all the robots are on the leaves. Since there are only $m-3$ robots on the leaves, there will be 3 leaves without robots on them (*holes*). By moving the robot at the center to one of the holes, we can turn

this problem to the previous case which has been proved to be reachable. Therefore, in this case, the star is reachable as well.

Case C: In this case, all the robots are at the leaves at the beginning. And their final targets are leaves as well except one of them should be moved to the junction. Similar to *Case A*, the coordination algorithm can move all the robots that have leaves as their targets first and then move the last robot to the junction. Since all the other robots have already been moved to their destination, this one at the junction will not cause any blocking. Thus, in this case, the star is reachable.

Case D: For this scenario, one of the robots starts at the junction and the target positions of all the robots are leaves except one which should be navigated to the junction. This robot may or may not be the same robot which starts at the junction. Meanwhile, since there are totally m leaves and two of them are empty, the coordination scheme can first move the robot at the junction to one of the empty leaves. This converts this case to the *Case C*, which has been proved to be reachable. Consequently, *Case D* is reachable as well.

However, if the amount of the robots is larger than $m-1$, this implies that there will be holes less than 2, which means 1 hole or 0 hole. Apparently, the tree will not be reachable for 0 hole. For the case of 1 hole, it is not hard to show that the tree will still not be reachable. For example, if the hole is the junction. Then all the leaves should have been occupied by robots. And any one of them can only move to the junction but cannot be navigated to other leaves. Further, if the hole is at one of the leaves, we can first move the robot at the junction

to this particular leaf and this case turns to be the same as the former. Both of these cases are not reachable.

5.5.3 Extended n -Star

An *extended n -Star* is a generalization of an n -Star. Like an n -Star, it has one junction and n vertices as leaves, but an extended n -Star can have internal vertices between leaves and the junction. Formally, it consists of n chains of vertices, all connected together by a junction in the center. Figure 46 shows an example of an extended n -star with 6 leaves, where v_1 is a junction, $\{v_2, v_3, v_4, v_5, v_6, v_7, v_{13}\}$ are internal vertices, and $\{v_8, v_9, v_{10}, v_{11}, v_{12}, v_{14}\}$ are leaves.

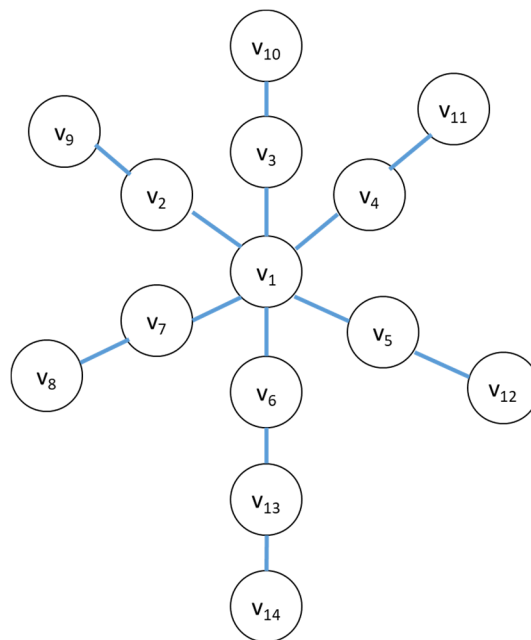


Figure 46. The illustration of the subgraph structure, Extended Star.

Definition 2. The *origin* of a tree is defined as the vertex which has the highest degree.

We can see that based on this definition, the origin of a given tree may not be unique. In other words, there might be several vertices which have the same highest degree among all the vertices. Therefore, we propose another definition to help us finalize the selection.

Definition 3. The *level* of a vertex is defined as the distance from the origin of the tree to this particular vertex along the shortest path between them.

Consequently, for a given tree where several vertices have the same degree which is also the highest among all the vertices, we choose the one that will minimize the overall levels for all the other vertices.

Definition 4. A *Stalk* is the longest branch starting from the origin (*not included*) on a tree.

Corollary 1. The leaf on the stalk of a given tree has the largest level, l_{max} .

Proof. Based on the definition of a stalk, it is the longest branch on a given tree (seen in Figure 47). In other words, the leaf of this branch will be the farthest vertex from the origin on the tree, which indicates the distance of it to the origin, also called its level, is the largest.

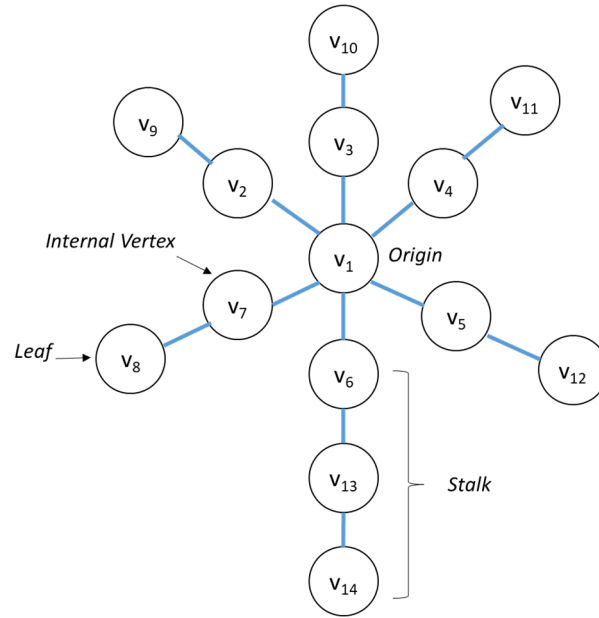


Figure 47. The illustration *Origin*, *Stalk*, *Internal vertices*, and *Leaves*.

Lemma 2. For an extended n -Star tree, the maximum number of robots it can accommodate while still being reachable is $m = |G| - l_{max} - 1$.

Proof. To prove the extended star can always provide feasible solutions for m robots from any initial configuration to any final configuration, let us consider the worst case. That is a robot t located at the leaf of the stalk need to be navigated to the leaf of another connected chain of the tree. Given that the robot t will need to be navigated through its own branch and the junction to reach the other branch where the target is situated. At least there should be a clear path connecting from the current place of the robot t to the junction. Further, in order to move this robot from the junction to its target, which is the leaf node on the target branch, there should be another empty vertex on any branch other than the original and target ones so that the robot can move to this empty vertex temporarily and all

the robots on the target branch can be pushed into the original branch to clear the rest of the path for robot t .

For the first part of the path which connects the original place of the robot to the junction, the coordination scheme needs l_{max} empty vertex, which includes the junction itself. To clear the second part of the path, the robot need to be moved to another branch, which requires one more hole. Overall, $l_{max} + 1$ holes are required for the worst case. Considering that the total number of the vertices on the tree is denoted by $|G|$, which is also the degree of the tree, we can conclude that in order to make sure the tree is reachable, the number of robots should be equal to or less than $m = |G| - (l_{max} + 1) = |G| - l_{max} - 1$.

On the other hand, if the number of holes is less than $l_{max} + 1$, for example l_{max} , the robot will only be able to move the junction and no further, thus making the whole tree unreachable.

5.5.4 Reachable Territory & General Acyclic Graph

Based on our investigation about simple subtree structure, we can now consider more general trees which may include multiple junctions. Since obviously not every tree will contain only one junction, we propose to partition a given general tree into different territories formed around each junction. In this way, we can determine if a general tree is reachable by following a two-step coordination scheme:

- 1) If the robots can be moved around inside a territory formed by its junction; or in other words, the reachability of a single territory is verified;
- 2) If the coordination scheme can move the robots from one territory to another.

Here, to facilitate our discussion on this two-phase multi-robot coordinate scheme, we propose a new definition, called *Reachable Territory*.

Definition 5. *Reachable Territory* of a tree is a set of vertices around a junction, in which the following equation holds true:

$$RT_j = \{u \mid \text{dist}(u, j) \leq H-1, \text{ for any } u \in V\}.$$

Here j is a junction on the tree, V is the set of vertices of the whole tree, H is the total number of holes (*empty vertices*) on the tree, and RT_j is the resulting reachable territory around j .

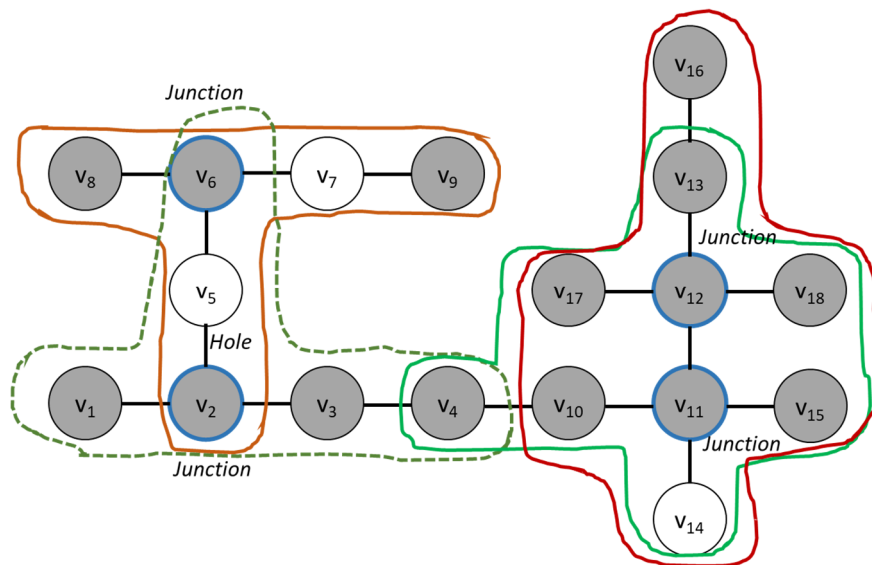


Figure 48. Four reachable territories on an example tree.

From the definition we can see that a reachable territory includes a set of vertices that are close enough to a specific junction. Figure 48 illustrates an example of a general tree with 18 vertices, three of which are holes (v_5, v_7, v_{14}). As shown in the figure, there are four

junctions, $\{v_2, v_6, v_{11}, v_{12}\}$, and therefore four reachable territories, $\{RT_1, RT_2, RT_3, RT_4\}$, which have been listed in Table 7.

Table 7 Reachable Territory with its Vertices

Reachable Territory	Junctions	Vertices
RT_1	v_6	$v_2, v_5, v_6, v_7, v_8, v_9$
RT_2	v_2	$v_1, v_2, v_3, v_4, v_5, v_6$
RT_3	v_{11}	$v_4, v_{10}, v_{11}, v_{12}, v_{13}, v_{14}, v_{15}, v_{17}, v_{18}$
RT_4	v_{12}	$v_{10}, v_{11}, v_{12}, v_{13}, v_{14}, v_{15}, v_{16}, v_{17}, v_{18}$

Further, based on the definition of reachable territory and Lemma 2, we can also have the following.

Lemma 3. If a reachable territory has H holes in it, it will certainly be reachable, where H is the total number of holes of the tree.

Proof. Given the definition of reachable territory, we can see that reachable territory can be considered as an extended star with less vertices:

- 1) A reachable territory has only one junction;
- 2) A reachable territory can have internal vertices and leaves;
- 3) All the leaves inside a specific reachable territory is no further than $H-1$ from the junction.

Having what we have proved for *Lemma 2*, for a given reachable territory to be reachable, it needs at least $l_{max}+1$ empty nodes, where l_{max} is the level of the leaf on the longest branch. While in the definition of reachable territory, we have limited the vertices which belong to a reachable territory to be equal to or closer than $H-1$. This implies that a reachable territory will need at least $(H-1)+1$ holes to be reachable, which gives H . Thus, *Lemma 3* is proved.

Take the reachable territory consisting of $\{v_2, v_5, v_6, v_7, v_8, v_9\}$ as an example. Overall, there are 3 holes on the tree in total and six vertices in this particular territory. If we move the robot at v_{11} to v_{14} and move $\{v_2, v_3, v_4, v_{10}\}$ all to the right one step, we will have another hole at v_2 , and there will be three holes $\{v_2, v_5, v_7\}$. Any robots on the rest three vertices in this territory will be able to move freely to any position inside it.

Based on our discussion about reachable territory, let us consider more carefully how such concept can be used to decompose a general acyclic graph, also known as a tree. A general acyclic graph normally has more than one junction, as shown in Figure 48. Consequently, we can partition a given acyclic graph into several reachable territories, each of which consists of a junction on the graph and vertices that satisfies the requirements from *Definition 5*. As proven in *Lemma 3*, if there are enough holes in a single reachable territory, all the robots would be able to move freely inside that particular territory. In the rest of the subsection, we will identify other conditions under which robots can be navigated among reachable territories.

Definition 6. Any two reachable territories are considered *close* if there is no junction vertices between them. That is,

$$RT_1 \cong RT_2, \text{ if } u \text{ is not a junction vertex, for any } u \in \text{Path}(J_1, J_2)$$

Here we use symbol ' \cong ' to indicate two junctions to be close to each other. RT_1 and RT_2 are two reachable territories determined by two junctions J_1 and J_2 , respectively. $\text{Path}(J_1, J_2)$ represents all the vertices that are on the shortest path from junction vertices J_1 to J_2 . An example of two close reachable territories is shown in Figure 49, where the two reachable territories determined by junction vertices J_1 and J_2 are close, since there are no other junction vertices between them. This is also the case for J_2 and J_3 , and J_3 and J_4 .

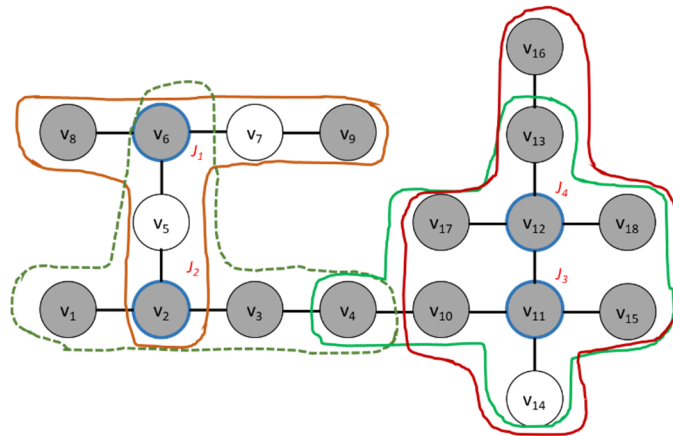


Figure 49. An example of close reachable territories.

Based on the concept of two reachable territories being close, another notion can be introduced to facilitate our investigation on how to determine the reachability of a general acyclic graph, namely *Joint Reachable Territory*.

Definition 7. Two reachable territories are *joint* if they are close and the distance between their junction vertices is no more than $H-2$, where H is number of holes on the tree. That is,

$$RT_m \equiv RT_n, \text{ if and only if } RT_m \equiv RT_n \text{ and } \text{dist}(J_m, J_n) \leq H-2$$

Here, we use the symbol ' \equiv ' to represent two territories being joint, and RT_m and RT_n are two reachable territories determined by two junctions J_m , and J_n , respectively. Based on this definition, there is only one pair of joint reachable territories, $\{RT_3 \equiv RT_4\}$ in Figure 49,

Corollary 3. If RT_1 and RT_2 are joint reachable territories, any robot in RT_1 can also access any vertices in RT_2 .

Proof. We will prove the correctness of this corollary by contradiction. Consider the RT_1 and RT_2 in Figure 49, where the distance between J_1 and J_2 is 2, which is one more than 1 ($H-2 = 3-2 = 1$). According to Definition 7, these two are not joint reachable territories. The worst case of a robot traveling between them would be the case that the starting point of the robot is at one leaf of a territory, for example v_9 in RT_1 , while its target is the other end of the second territory, say v_4 in RT_2 . From Lemma 3, we have shown that the robot can be freely navigated inside a reachable territory by first moving all the holes on the tree, H in total, to that particular territory. Now we want to move a robot from one reachable territory RT_1 defined by junction J_1 to another reachable territory defined by J_2 . Apparently, to make this movement feasible, the path from J_1 to J_2 should be clear, which has a distance of 2. And in order to move the robot to v_4 , both v_3 and v_4 should be empty as well. In total, this operation will require 4 holes, which are $\{v_5, v_2, v_3, v_4\}$. However, there are only 3 holes

in the whole tree. Thus, this operation is not feasible. Following the same method, we can easily show that any robot in RT_3 can be moved to any vertices in RT_4 , since these two are joint reachable territories. The distance between their junctions is 1, which is no more than $H-2$, where $H = 3$ in our example graph.

By combining this two-step coordination scheme we have shown above, the definition of joint reachable territory, and the results of Corollary 3, we can extend our investigation from one reachable territory to a general graph with multiple territories. The following theorem study the conditions under which **any** configuration of robots on a general acyclic graph can be rearranged to **any** other configuration as final destination without mutual collisions.

Theorem 1. For any general acyclic graph with H holes and more than one reachable territories, the whole graph is reachable if and only if the following is true:

- 1) Any two close reachable territories are joint ones;
- 2) Every vertex on the graph belongs to one or more reachable territories.

Proof. \Rightarrow : The original statement in this direction is that if a general acyclic graph is reachable, any vertex on the graph should belong to at least one reachable territory and any two close reachable territories should be joint territories as well. We now show the correctness of this statement by contradiction for each of these two conditions separately.

- 1) Suppose the graph is reachable but two of its reachable territories, for example RT_a and RT_b , are not joint such that the junction vertices of them have a distance larger

than $H-2$, say $H-1$. Considering both RT_a and RT_b are reachable territories and there are H holes on the graph, from *Lemma 3* we have seen that we can easily prove that a robot, t , inside RT_a is able to move freely inside this particular territory. Since the statement also says that the whole graph is reachable, t should be able to move to any position on the graph, say a vertex inside RT_b away from RT_a . In order to make this happen, the path from the junction node of RT_a to the one of RT_b should be empty. Thus, including the destination of the robot, there should be at least $H+1$ ($(H-1)+1+1$) holes, which contradicts to the statement in which there are only H holes for the whole graph. Thus this concludes that if a given acyclic graph is reachable, any two close reachable territories are joint.

- 2) The second condition enforces that there is no isolated vertices on the graph. Here, “isolated vertices” means “vertices do not belong to any reachable territory”. Again, we prove this statement by contradiction. Assume that there is a reachable acyclic graph, G and one of its vertices, u , does not belong to any reachable territory. According to Definition 5, any vertex belongs to a reachable territory should be no further than $H-1$ to the junction. Since u is isolated, the distance between u and the closest junction, J , should be at least H . Suppose that there is a robot, t , inside the reachable territory defined by J and it needs to be navigated to u . Considering that the distance between u and J is H , this navigation operation would require $H+1$ empty vertices along this path. Apparently, there is only H holes on the graph and therefore, the vertex u cannot be isolated.

\Leftarrow : In this direction, the two conditions are satisfied and we now show that if this is the case, the acyclic graph will be reachable. Assume there is a robot t at the vertex u on the graph. According to the second condition, u (on which the robot sits) belongs to at least one of the reachable territories. Without loss of generality, we assume the closest junction to u is J_m and the reachable territory defined by J_m is RT_m . That is, u belongs to RT_m . Consequently, based on our discussion for Lemma 3, the robot t can move to any vertices inside RT_m . Meanwhile, we also have the first condition, which assures that any two close reachable territories are joint reachable territories. This means that not only can the robot move freely inside RT_m , it can be navigated to any vertices inside the joint reachable territories of RT_m , say RT_n . Following the same scheme, the robot will be able to move to any vertices of any other reachable territories of RT_n and *etc.* Revisiting the second condition, we have that every vertex belongs to at least one reachable territory, we can conclude that this robot will be able to reach any vertex on the graph. Equally, this graph is reachable.

Formally, we define the following expression as the operation of moving a robot at the vertex of t_m inside of RT_m to the vertex of t_n which belongs to RT_n :

$$\text{Move}\left(\frac{t_m}{RT_m}, \frac{t_n}{RT_n}\right).$$

Given any acyclic graph, which has k reachable territories, $\{RT_1, RT_2 \dots RT_k\}$, according to the first condition, any two close reachable territories are joint reachable territories. In other words, we have

$$RT_i \equiv RT_{i+1} \text{ or } Move\left(\frac{t_i}{RT_i}, \frac{t_{i+1}}{RT_{i+1}}\right) \text{ is feasible}$$

where $i \in 1, 2, \dots, k - 1$.

Given that the robot t belongs to RT_m , the coordination scheme can move the robot to any reachable territory l by moving it from one to another as follows:

$$Move\left(\frac{t_m}{RT_m}, \frac{t_l}{RT_l}\right) = Move\left(\frac{t_m}{RT_m}, \frac{t_{m+1}}{RT_{m+1}}\right) \rightarrow Move\left(\frac{t_{m+1}}{RT_{m+1}}, \frac{t_{m+2}}{RT_{m+2}}\right) \rightarrow \dots \rightarrow Move\left(\frac{t_{l-1}}{RT_{l-1}}, \frac{t_l}{RT_l}\right)$$

This demonstrates that with the help of joint reachable territory, a robot at any place of the graph can be navigated to any other place. Or put it another way, any two robots on the graph can switch their positions. This means that the whole graph is reachable under such conditions.

5.5.5 Example Algorithm

An example algorithm for any general graph without loops can be seen as follows. This algorithm has assumed that the given graph meets the requirements given in Theorem 1.

Algorithm 1 MOTION_ON_TREE(G, R, A, T)

```
1:  $J \leftarrow$  Tree of junctions //junctions  $j_p$  and  $j_q$  are connected if  $Near(j_p, j_q)=1$ 
2:  $Q \leftarrow$  SORT( $R, T$ ) // Sort robots based on the depth of targets from largest to smallest
3:  $U \leftarrow$  NULL
4: for all  $r \in Q$  do
5:    $t \leftarrow$  TARGET( $T[r], U$ ) //  $t$  is the deepest available node in subtree of  $T[r]$ 
6:    $P \leftarrow$  SHORTEST_PATH( $J, A[r].junction, t.junction$ )
   //  $n$  is a node in Influence Zone (IZ) of  $n.junction$ 
7:   if  $P.length = 1$ 
8:      $P' \leftarrow$  SHORTEST_PATH ( $G, A[r], t$ )
9:   else
10:     $P' \leftarrow$  SHORTEST_PATH ( $G, A[r], P[1]$ )
11:    $v \leftarrow$  AVAILABLE_NEIGHBOR( $P[0], P', U$ )
   //  $v$  is an available neighbor of  $P[0]$  that is not in  $P'$ 
12:   if  $v \neq$  NULL
13:     MOVE( $G, A, r, v, U$ ) //move robot  $r$  from  $A[r]$  to  $v$ 
14:   for  $i$  from 1 to  $P.length-2$ 
15:      $P' \leftarrow$  SHORTEST_PATH ( $G, A[r], P[i+1]$ )
16:      $v \leftarrow$  AVAILABLE_NEIGHBOR( $P[i], P', U$ )
17:     MOVE( $G, A, r, v, U$ )
18:    $P' \leftarrow$  SHORTEST_PATH ( $G, A[r], t$ )
19:    $v \leftarrow$  AVAILABLE_NEIGHBOR( $P[P.length-1], P', U$ )
20:   if  $v \neq$  NULL
21:     MOVE( $G, A, r, v, U$ )
22:   MOVE( $G, A, r, t, U$ )
23:    $U \leftarrow \{r\} + U$ 
24: for all  $r \in U$  do
25:    $P \leftarrow$  SHORTEST_PATH( $G, A[r], T[r]$ )
26:   move  $r$  from  $A[r]$  to  $T[r]$  along  $P$ 
```

G : graph; R :list of robots; A : start position of robots; T : targets of robots;
line 7-13: move robot r from $A[r]$ to the neighbor of the first junction $P[0]$
line 14-17: move robot r from the neighbor of $P[0]$ to the neighbor of $P[P.length-2]$
line 18-22: move robot r from the neighbor of $P[P.length-2]$ to the target

Algorithm 2 TARGET (n, U)

```
1:  $t \leftarrow n$ 
2: for  $d \in$  descendants of  $n$ 
3:   if  $d.depth > t.depth$ 
4:     if there is no robot on  $d$ 
5:        $t \leftarrow d$ 
6:   else
7:      $r \leftarrow$  robot on  $d$ 
8:     if  $r$  is not in  $U$ 
9:        $t \leftarrow d$ 
10: return  $t$ 
```

Algorithm 3 AVAILABLE_NEIGHBOR(j, P, U)

```
1: for  $n \in$  neighbors of  $j$ 
2:   if  $n$  is not in  $P$ 
3:     if there is no robot on  $n$ 
4:       return  $n$ 
5:     else
6:        $r \leftarrow$  robot on  $n$ 
7:       if  $r$  is not in  $U$ 
8:         return  $n$ 
9: return NULL
```

Algorithm 4 MOVE(G, A, r, v, U)

```
1:  $Q \leftarrow$  empty nodes in subtree of  $v$ 
2: SORT( $Q$ )
3: for  $n \in Q$  do
4:    $s \leftarrow$  BFS_ROBOT( $G, n, r$ )
   //find nearest robot  $s$ ,  $s$  is not in  $U$ , And  $r$  is not in path from  $n$  to  $s$ 
5:   if  $s \neq$  NULL
6:     move  $s$  to  $n$ 
7:  $P \leftarrow$  SHORTEST_PATH( $G, r, v$ )
8: for  $i$  from 1 to  $P.length-1$  do
9:   if  $P[i]$  is not empty
10:     $e \leftarrow$  BFS_EMPTY( $G, P[i], r$ )
    //find nearest empty node  $e$ , and  $r$  is not in path from  $P[i]$  to  $e$ 
11:     $P' \leftarrow$  SHORTEST_PATH( $G, P[i], e$ )
12:    for  $j$  from 1 to  $P.length-2$  do
13:      move robot on  $P[j]$  to  $P[j+1]$ 
14:    move  $r$  to  $P[i]$ 
15:  $A[r] = v$ 
```

line 1-6 tries to push all the empty space out of the subtree of v so that the empty spaces will not be blocked by robot r after it's moved to position v

5.5.6 Robot Capacity on Acyclic Graph

In the last section, we have demonstrated what kind of topology a general acyclic graph should have in order to be reachable. This is based on the assumption that there are certain number of holes on the tree such that there is enough space for the robots to move around. But we have not shown how many holes are necessary for a given graph. Or put it another way, what is the maximum number of robots a given acyclic graph can accommodate while

it is still being reachable. This is the second question we would like to answer in this script, and we call it *robot capacity problem*. Here we present another theorem which gives the upper bound of the number of robots that any acyclic graph can accommodate. Given that the number of the holes is equal to the number of vertices on the graph except those with robots on it, this also defines the lower bound of the number of holes.

Theorem 2. The maximum number of robots that an acyclic graph can afford while still being reachable, or the *robot capacity* of a given acyclic graph, is given by the following equation:

$$m = |G| - l_{max} - s$$

where $|G|$ is the degree of the tree, or the number of vertices. Same as the previous section, l_{max} represents the length of the stalk on the graph. s , however, may have two different values depending on the topology of the given graph. On one hand, if one end of the stalk is connected to a junction and the other end is a leaf, same as the one in extended n -star from Section 5.5.3, s would take the value of 1. On the other hand, if both ends of the stalk are connected to junctions, s should be equal to 2. Figure 50 gives an example of an acyclic graph with a stalk which connects a leaf to a junction, while Figure 51 illustrates the case where both ends of the stalk are junctions.

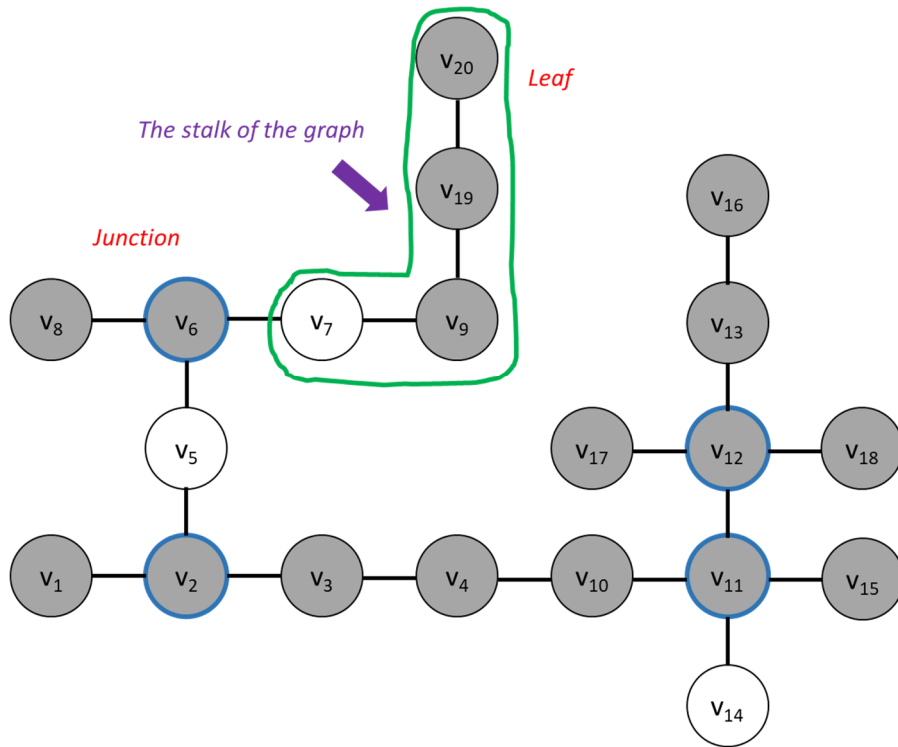


Figure 50. An example of a stalk connecting a leaf and a junction.

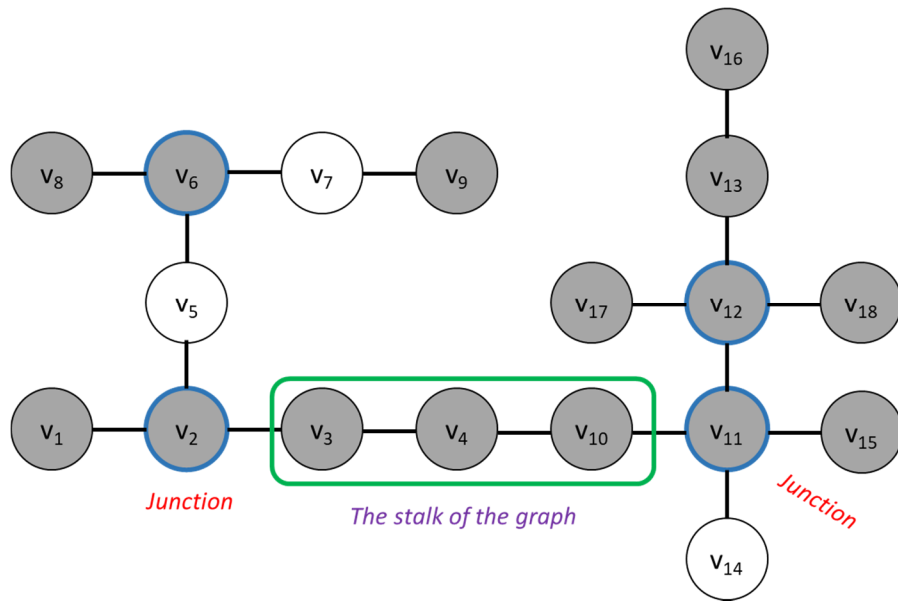


Figure 51. An example of a stalk with junctions at both ends.

Theorem 2 can be interpreted as the upper bound of the number of the robots or equally the least number of the holes for the whole graph to be reachable.

Corollary 4. The least number of holes on the graph to make it reachable is $l_{max} + s$, where

$$s = \begin{cases} 1 & \text{if there is a leaf at one end of the stalk} \\ 2 & \text{if the stalk is connected to junctions at both ends} \end{cases}$$

It is also worth mentioning that both Theorem 2 and Corollary 4 stay true when there is only one junction on the graph. For example, we have shown in Section 5.5.3 that for extended n -star tree which has only one junction, its robot capacity is $|G| - l_{max} - 1$, or in other words, the least number of holes is $l_{max} + 1$. This is consistent with Theorem 2 and Corollary 4.

5.6 Graph with Cycles

In the previous section, we have shown how a general acyclic graph, or a tree, can be partitioned into several reachable territories and what conditions such graphs need to meet to be reachable for a group of robots. We have also discussed how to calculate the maximum number of robots an acyclic graph can accommodate. However, we have only investigated acyclic graphs. In real-world environments, system designers often need to deal with circles on the topological maps during sensor deployment. In this section, we will continue our discussion of reachable graphs and robot capacity but on graphs with cycles.

We start from a series of small graphs and then extend our findings to more general cyclic graphs.

5.6.1 Unicyclic Graph

A unicyclic graph is a connected graph with exactly one cycle and no leaves. Figure 52 gives an example of a unicyclic graph with 6 vertices. Although unicyclic graphs have simple topological structure and may not occur often in real environment, it serves as an easy illustration of how robots can be navigated on a circle.

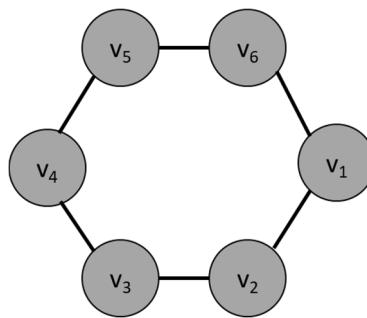


Figure 52. An example of a unicyclic graph.

Consider a single circle with m vertices. Apparently, the total number of robots it can accommodate will be m , while the number of different permutations of robots is $m!$. But, the only motion a group of robots can do on a circle is either clockwise rotation or counter-clockwise one, no matter if there are holes on it. This indicates that a group of vertices placed in a circle is not able to afford $m!$ permutation, simply because robots cannot change their relative positions.

Lemma 4. A group of robots on a circle can be rearranged if and only if there is an extra empty vertex connected to the circle forming a *pan* graph.

Proof. Figure 53 gives an example of a pan graph where an extra empty vertex, v_0 , is connected to the original unicyclic graph. In this way, any robots from the original group $\{r_1, r_2, r_3, r_4, r_5, r_6\}$, whose original positions are $\{v_1, v_2, v_3, v_4, v_5, v_6\}$, can exit from the loop and reenter it when necessary. In this way, the whole group of robots can reorder their sequence and achieve all $m!$ permutation.

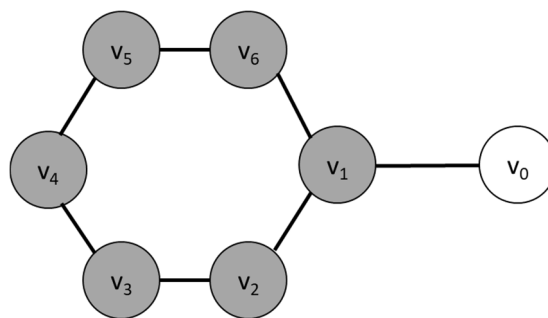


Figure 53. An example of a pan graph.

Similar to how trees with a star structure play an important role for multiple robot coordination on acyclic graphs, pan graph is a basic component for cyclic graphs on which a sequence of robots can exchange their positions making the whole graph reachable. Therefore, we will extend this concept to more general graphs.

5.6.2 (m, n) Tadpole Graph

A (m, n) tadpole graph, which is also known as a dragon graph, is a graph obtained by combining a unicycle graph with m vertices and a chain with n vertices, as shown in Figure 54.

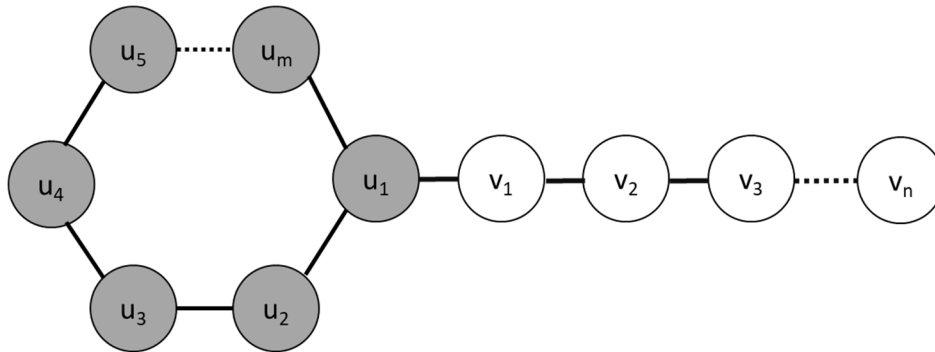


Figure 54. An example of an m - n tadpole graph.

Lemma 5. A tadpole graph with m vertices in the circle and n vertices on the tail is reachable if and only if there are n holes on the graph.

Proof. From Lemma 4 we can see that a $(m, 1)$ tadpole graph with one hole is reachable for m robots but not for $m + 1$. If we add another empty vertex at the end of the chain forming a $(m, 2)$ tadpole graph with 2 holes, however, this whole graph is still only reachable for m robot but not $m + 1$. Since even if we add one more robot at one of the two vertices on the tail, this robot still cannot enter the circle simply because there are only m vertices for the circle. Following this, we can keep adding more empty vertices on the tail but would not increase the robot capacity of the graph.

5.6.3 Extended Tadpole Graph

Similar to how we have extended our discussion on an n -Star to an extended n -Star in Section 5.5.3, here we extend our conclusion on a (m, n) tadpole graph to *extended tadpole graphs*.

Definition 8. An *extended tadpole graph* is a graph with only one circle but multiple chains of vertices, each of which is connected to one vertex in the circle.

An example extended tadpole graph is shown in Figure 55.

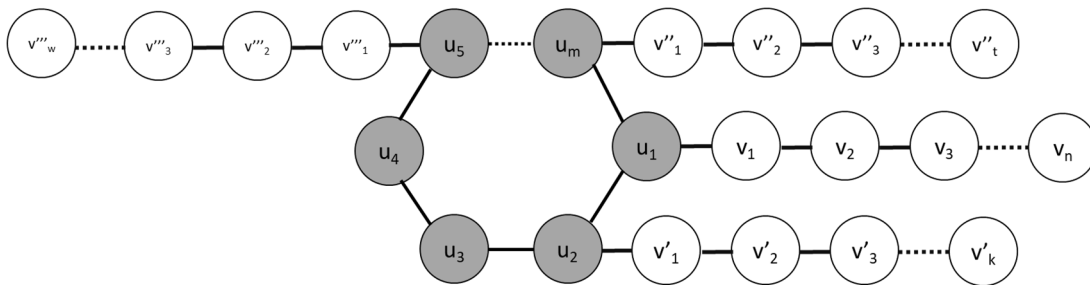


Figure 55. An example of an extended tadpole graph.

Further, we also extend our definition of a stalk from acyclic graphs to cyclic graphs. And, we still use the notion l_{max} to represent the length of a stalk for a given cyclic graph.

Definition 9. A stalk on a cyclic graph is the longest chain connecting two vertices on two different circles or connecting one vertex on a circle and a leaf (*included*). And its length is noted as l_{max} .

An example stalk which connects two circles is given in Figure 56. Another which connects a circle with a leaf is given in Figure 57.

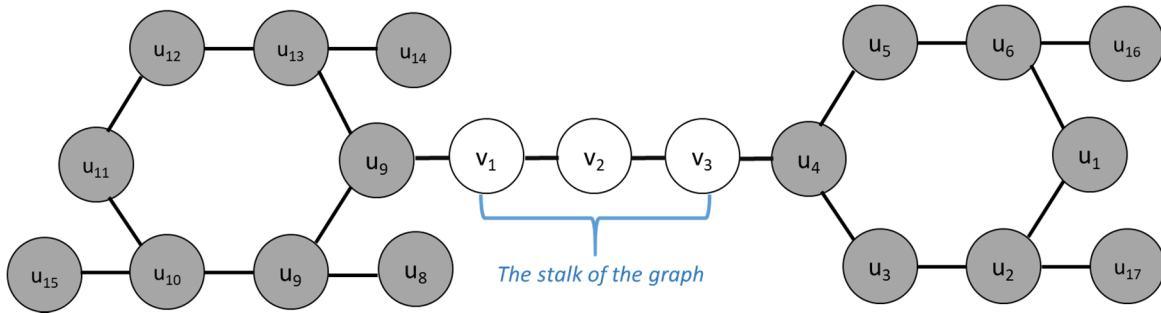


Figure 56 An example of a stalk on cyclic graph.

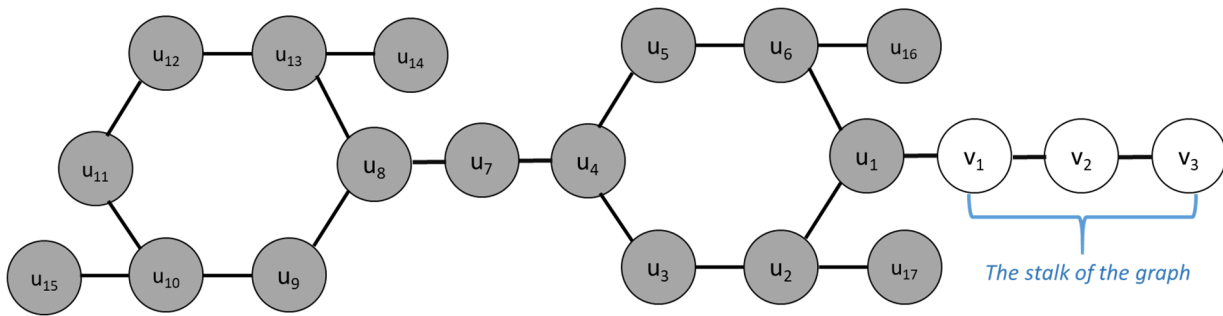


Figure 57 Another example of a stalk on cyclic graph.

Based on our finding on tadpole graphs and new definition of extend tadpole graphs, we can have the following conclusion.

Lemma 6. An extended tadpole graph is reachable if and only if there are at least l_{max} holes on it.

Proof. According to the definition of extended tadpole graphs and the property of the stalk on such graphs, we know that on a given extended tadpole graph, the leaf node on the stalk is the furthest vertex away from the circle. And that distance is l_{max} . Based on this, we can see that when a robot needs to be navigated from one vertex to another on such graph, the worst case is to move a robot from the leaf on the stalk to the leaf of another chain. Because there is only one circle on the graph, in order to move the robot from one chain to another, the robot need to be on the circle first. This means that at least the path from the robot itself to the vertex on the graph which the chain of the robot is connected to should be clear. Or in other words, this requires at least l_{max} empty nodes to move the robot from its current place to the circle. After that, all the robots on the circle can move simultaneously so that the robot can move to the vertex on the circle to which the chain of its target is connected. This step does not require any extra holes.

Further, after the robot moves to the circle, all the robots on the target chain can move out of this particular chain so that when the robot can have a clear way to reach to the leaf. And since the first step already requires l_{max} empty vertices and all the other chains has less or equal number of vertices than the stalk. So we can be sure that there is enough space for the robots on the target chain.

After the robot navigates to its target position, we can move other robots back to their original positions by reverse operation. In this way, we can show that we can move any robot to any position on an extended tadpole graph as long as there are at least l_{max} holes on it. Following the same principle, we can be sure that any initial configuration of multiple

robots on such a graph can be manipulated to any final configuration. Or put it another way, such a graph is reachable.

5.6.4 General Cyclic Graph

Similar to how we extend the reachability of an extended n -star to any general acyclic graph in Section 5.5.4, we can have the correspondent conclusion for cyclic graphs.

Theorem 3. For a general cyclic graph which has at least one circle and one extra vertex that is not on any circle, there should be at least l_{max} holes to make the overall graph reachable. Here, l_{max} is the length of the stalk of the graph.

Proof. From the previous subsection we have seen that this condition is good enough to guarantee an extended tadpole graph to be reachable. And an extended tadpole graph can be considered as a general cyclic graph but with only one cycle. More general cyclic graph may contain more than one cycle on it. Since l_{max} is the length of the stalk, which is the longest path consisting of vertices that are not in any circle, any other paths on the graph which has less or equal number of vertices can be evacuated for robot coordination simply by moving the robots to existing holes on the graph.

Further, as we have shown in Section 5.4.1, one of our assumptions is that all the graphs discussed here are connected. There are no isolated vertices or a group of vertices. Therefore, for any circle on the graph, it is guaranteed that there will be at least one extra

vertex connected to it. And Lemma 4 already shows that any circle with an extra empty node is reachable.

Therefore, any circle on the graph is reachable, and any path that is connected to two circles at both ends or one circle and one leaf is reachable. We can conclude that any general cyclic graph with at least one circle and at least one vertex which is not in any circle will be reachable as long as l_{max} vertices are not occupied by the robots.

5.6.5 Example Algorithm

An example algorithm for any general graphs with loops can be seen as follows. This algorithm has assumed that the given graph meets the requirements given in Theorem 3.

Algorithm 1 MOTION_ON_GRAPH(G, R, A, T)

```
1:  $T \leftarrow$  Tree of  $G$  where all cycles are reduced to nodes
2:  $Q \leftarrow$  SORT( $T$ ) // sort nodes based on their depth from the smallest to the largest
3:  $L \leftarrow$  NULL
4: for  $n \in T$ 
5:   if  $n$  is a cycle in  $G$ 
6:      $L[n] \leftarrow$  all the nodes in this cycle in the order of clockwise
7: for  $n \in Q$ 
8:   if  $L[n] \neq$  NULL
9:     replace  $n$  with  $L[n]$ 
10:  $U \leftarrow$  NULL
11: for  $n \in$  reverse of  $Q$ 
12:   if  $\exists r, T[r] = n$ 
13:      $t \leftarrow$  TARGET( $T[r], Q, T, L, U$ )
        //  $t$  is the deepest available node in subtree of  $T[r]$ 
14:      $P \leftarrow$  SHORTEST_PATH( $G, A[r], t$ )
15:      $c \leftarrow$  CYCLE ( $L, A[r]$ ) // get the cycle which contains  $A[r]$ 
16:      $p \leftarrow 0$ 
17:     while ( $p < P.length$  && CYCLE ( $L, P[p]$ ) == NULL)  $p++$ 
18:     if ( $p == P.length$ )
19:       MOVE ( $r, A, P, t, G$ ) // if there is no cycle in  $P$ , move robot  $r$  to target  $t$  along  $P$ 
20:       continue
21:      $n_{in} \leftarrow P[p]$  // entering node of the first cycle
22:     MOVE ( $r, A, P, n_{in}, G$ ) // move the robot  $r$  in to the first cycle
23:     while  $A[r] \neq t$  // move the robot  $r$  among cycles along  $P$ 
24:        $c \leftarrow$  CYCLE ( $L, A[r]$ )
25:       while ( $p < P.length$  &&  $c ==$  CYCLE ( $L, P[p]$ ))  $p++$ 
26:        $n_{out} \leftarrow P[p-1]$  // exiting node of the cycle or  $t$ 
27:        $n_{in} \leftarrow$  NULL // entering node of the next cycle
28:       if  $n_{out}$  in another cycle other than  $c$ 
29:          $n_{in} \leftarrow n_{out}$ 
30:       else
31:         while ( $p < P.length$  && CYCLE ( $L, P[p]$ ) == NULL)  $p++$ 
32:         if  $p < P.length$ 
33:            $n_{in} \leftarrow P[p]$ 
34:         CLEAR( $n_{out}, n_{in}, r, P, A, G, L$ ) // clear all the robots between  $n_{out}$  and  $n_{in}$ 
35:         move all robots  $r' \in c$  clockwise until  $A[r'] == n_{out}$ 
36:         MOVE ( $r, A, P, n_{in}, G$ ) // move the robot  $r$  in to the next cycle if  $n_{in} \neq$  NULL
37:        $U \leftarrow \{r\} + U$ 
38: for all  $r \in U$  do
39:    $P \leftarrow$  SHORTEST_PATH( $G, A[r], T[r]$ )
40:   move  $r$  from  $A[r]$  to  $T[r]$  along  $P$ 
```

Algorithm 2 TARGET (n, Q, T, L, U)

```
1: if CYCLE ( $L, n$ )  $\neq$  NULL
2:    $c \leftarrow$  CYCLE ( $L, n$ )
3:   for  $n' \in$  reverse of  $Q$ 
4:     if  $n'$  is a node in subtree of  $n$  or  $c$ 
5:       return  $n'$ 
6:     else if  $c \neq$  NULL &&  $n' \in L[c]$ 
7:       return  $n'$ 
8:   return  $n$ 
```

Algorithm 3 CYCLE (T, L, n)

```
1: for  $c \in T$ 
2:   if  $c == n$ 
3:     return NULL
4:   if  $n$  in  $L[c]$ 
5:     return  $c$ 
6: return NULL
```

Algorithm 4 MOVE (r, A, P, v, G)

```
1:  $P \leftarrow$  SHORTEST_PATH( $G, r, v$ )
2: for  $i$  from 1 to  $P.length-1$  do
3:   if  $P[i]$  is not empty
4:      $e \leftarrow$  BFS_EMPTY( $G, P[i], r$ )
       //find nearest empty node  $e$ , and  $r$  is not in path from  $P[i]$  to  $e$ 
5:      $P' \leftarrow$  SHORTEST_PATH( $G, P[i], e$ )
6:     for  $j$  from 1 to  $P.length-2$  do
7:       move robot on  $P[j]$  to  $P[j+1]$ 
8:     move  $r$  to  $P[i]$ 
9:  $A[r] = v$ 
```

Algorithm 5 CLEAR ($n_{out}, n_{in}, r, P, A, G, L$)

```
1:  $c \leftarrow$  CYCLE ( $L, n_{out}$ )
2: move all robots  $r' \in c$  clockwise until  $A[r]$  is not in  $P$ 
3:  $P' \leftarrow$  SHORTEST_PATH( $G, n_{out}, n_{in}$ )
4: if  $\exists n, n$  is in  $P$  and  $n$  is not empty
5:    $e \leftarrow$  BFS_EMPTY( $G, n, n_{in}$ ) //find nearest empty node  $e$ , and  $n_{out}$  is not in path from  $n$  to  $e$ 
6:   move all the robots between  $n$  and  $e$  towards  $e$  until there is no empty space
```

5.6.6 Robot Capacity on Cyclic Graph

Similarly, we can also determine the maximum number of robots that a general cyclic graph can afford while it is still being reachable. Since the number of robots can be calculated by subtracting the number of holes from the number of vertices, we can have the following theorem for robot capacity on cyclic graph based on Theorem 3.

Theorem 4. The maximum number of robots that a cyclic graph can accommodate while still being reachable, or the *robot capacity* of a general cyclic graph, is given by:

$$m = |G| - I_{stalk} - 1$$

where I_{stalk} is the number of internal vertices on the stalk of a given graph, and its value depends on the topology of the graph. For one thing, if the stalk of a particular cyclic graph has its both ends connected to circles, $I_{stalk} = l_{max}$. Figure 56 gives an example of such case, where $\{v_1, v_2, v_3\}$ are vertices on the stalk. In this case, the maximum number of robot that can be accommodated on such graph is $m = 20 - 3 - 1 = 16$. Or in other words, at least 4 holes are required to make the overall graph reachable.

On the other hand, if stalk is connected to a circle at one end and a leaf at the other end, $I_{stalk} = l_{max} - 1$. This case is illustrated in Figure 57, where with the same amount of vertices for the overall graph, however, only 3 holes are required to guarantee that graph to be reachable, and therefore the robot capacity for this particular case is $m = 20 - (3 - 1) - 1 = 17$.

Chapter 6 Conclusion & Future Work

There is a growing literature on visible light enabled indoor localization. Yet, there is still a lack of proper visible light channel modeling for the potential applications. In this thesis, we take a system perspective and build visible light channel model specifically for indoor localization purpose. We consider both single and multiple transmitter-receiver pairs. Meanwhile, we also analyze the performance of the resulting communication link such as signal-to-noise ratio and throughput.

Based our investigation on visible light channel modeling, we design a high-accuracy indoor localization technology based on encoded visible light using DLP projectors, called Pilot. The salient distinguishing feature of Pilot is that all pixels inside the projection area have been encoded with location information and the only step needed to restore the corresponding coordinates for a particular sensor unit is to collect a sequence of sensor readings and convert the codeword to its original coordinates. This important property lays the foundation of its encouraging localization accuracy (1.7 millimeters) and sets it apart from all existing indoor localization system.

Further, our technology is sufficiently low-powered and inexpensive to enable integration into existing location-based applications. Communication in Pilot takes place in a unidirectional fashion from the DLP projector to the sensor unit. This simplicity allows us to maintain a minimum system design, not requiring a central communication infrastructure or a network server for heavy computation.

Finally, with Pilot, indoor localization and tracking occur in real time, as illustrated in the system evaluation section. This property can be particularly critical for real-time location based services, where low latency is highly preferred.

At present, Pilot also has several drawbacks. Foremost, it can only provide 2D location information. Further, we only test our current implementation with one projector, and investigating how to stitch multiple DLP projectors together to provide indoor locations at a larger scale needs to be explored. A potential approach is to assign unique identification to each projector and add a few naming frames at the beginning of the projected patterns so that a target light sensor can easily differentiate the source of received light signals. Modulating the light from the projectors with different carrier signals would also be necessary to avoid interference.

In addition, considering gray codes have a $O(\log_2 n)$ relationship between the number of required patterns and the total number of pixels of the projecting devices, projectors with more pixels and higher projection rate can be explored to improve localization accuracy and system refresh rate. Moreover, although our experimental results are promising, robustness need to be improved for real word cases. For example, we are working on the next version of Pilot, in which the location information is transmitted through a modulated visible light channel with a 480 kHz carrier. This could significantly improve the signal-to-noise ratio, thereby the overall performance of Pilot, which would allow us to track target devices under various challenging conditions.

Finally, the reason that Pilot is suitable for low-powered mobile devices is that the sensor unit itself only needs to decode time-multiplexing light signals to restore its own location using simple light sensors. Through our extensive experiments, the designed hardware system consumes ~ 25 mA current while decoding the received light signals at the frequency of 84 Hz. This does not include the power consumption of the WiFi module for data transmission, which in practice consumes about 0.65 Watts for transmission. Other low-powered communication technologies, such as Bluetooth Low Energy, can be utilized to reduce this part to around 0.45 Watts [91]. However, the DLP projector itself still consume a certain amount of power, around 15 Watts in our current prototype. Although they can be connected to the grid without the worry of running out of power, more effort is needed to evaluate the likelihood of reducing energy consumption.

Further, observing the promising performance of the proposed localization technology, the encoded projection based location discovery combined with a topological map is applied to a mobile robot in a real-world scenario for the first time. Our proposed navigation system used a topological map to perform global path planning, and the encoded projection location discovery was used for local motion control. Photodiodes are installed on the ceiling to serve as landmarks. From extensive experiments, we showed how the proposed system enables accurate navigation of a robot from one landmark to another without using other types of sensing technologies, such as laser scanners, cameras, or metric information of the environment, and achieves an average navigation deviation of 18.5 millimeters in practice. However, as shown in Section 4.4, more research questions should be investigated to make the overall system more suitable for real world cases.

Finally, for many real-world applications, multiple robots are employed to accomplish design goals more efficiently. However, for a topological map based multiple mobile robot navigation, it is crucial to make sure that the map has a proper topology and enough extra empty places to avoid mutual collision between robots. For the last chapter of this thesis, we extend our investigation on reliable indoor robot navigation and propose a general algorithmic framework for solving multi-robot coordination problems on both acyclic and cyclic topological graphs. We have demonstrated a new type of abstract representation for subgraph structures and generalized these subgraph topologies to more general cases step by step. We advocate that decomposing a topological map into subgraphs is a more intuitive and efficient way for path planners to perform coordination between multiple robots.

Bibliography

1. Rabaey JM, Ammer MJ, Da Silva JL, Patel D, Roundy S. PicoRadio supports ad hoc ultra-low power wireless networking. *Computer*. 2000;33(7):42-8.
2. Connolly P, Boone D. Indoor location in retail: Where is the money. ABI Research Report. 2013.
3. Bahl P, Padmanabhan VN, editors. RADAR: An in-building RF-based user location and tracking system. INFOCOM 2000 Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies Proceedings IEEE; 2000: Ieee.
4. Chen Y, Lymberopoulos D, Liu J, Priyantha B, editors. FM-based indoor localization. Proceedings of the 10th international conference on Mobile systems, applications, and services; 2012: ACM.
5. Zhang Z, Zhou X, Zhang W, Zhang Y, Wang G, Zhao BY, et al., editors. I am the antenna: accurate outdoor AP location using smartphones. Proceedings of the 17th annual international conference on Mobile computing and networking; 2011: ACM.
6. Hu P, Li L, Peng C, Shen G, Zhao F, editors. Pharos: Enable physical analytics through visible light based indoor localization. Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks; 2013: ACM.
7. Alavi B, Pahlavan K. Modeling of the TOA-based distance measurement error using UWB indoor radio measurements. *IEEE communications letters*. 2006;10(4):275-7.
8. Chintalapudi K, Padmanabha Iyer A, Padmanabhan VN, editors. Indoor localization without the pain. Proceedings of the sixteenth annual international conference on Mobile computing and networking; 2010: ACM.
9. Bargh MS, de Groot R, editors. Indoor localization based on response rate of bluetooth inquiries. Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments; 2008: ACM.
10. Barry JR, Kahn JM, Krause WJ, Lee EA, Messerschmitt DG. Simulation of multipath impulse response for indoor wireless optical channels. *IEEE journal on selected areas in communications*. 1993;11(3):367-79.
11. Hassan NU, Naeem A, Pasha MA, Jadoon T, Yuen C. Indoor positioning using visible led lights: A survey. *ACM Computing Surveys (CSUR)*. 2015;48(2):20.
12. Lopez-Hernandez F, Betancor M. DUSTIN: algorithm for calculation of impulse response on IR wireless indoor channels. *Electronics Letters*. 1997;33(21):1804-6.
13. Carruthers JB, Kannan P. Iterative site-based modeling for wireless infrared channels. *IEEE Transactions on Antennas and Propagation*. 2002;50(5):759-65.
14. Lopez-Hernandez F, Perez-Jimenez R, Santamaria A. Monte Carlo calculation of impulse response on diffuse IR wireless indoor channels. *Electronics Letters*. 1998;34(12):1260-2.
15. Lopez-Hernandez F, Perez-Jimenez R, Santamaria A. Modified Monte Carlo scheme for high-efficiency simulation of the impulse response on diffuse IR wireless indoor channels. *Electronics Letters*. 1998;34(19):1819-20.
16. Perez-Jimenez R, Berges J, Betancor M. Statistical model for the impulse response on infrared indoor diffuse channels. *Electronics Letters*. 1997;33(15):1298-300.
17. Perez-Jimenez R, Melián VM, Betancor MJ, editors. Analysis of multipath impulse response of diffuse and quasi-diffuse optical links for IR-WLAN. INFOCOM'95 Fourteenth

Annual Joint Conference of the IEEE Computer and Communications Societies Bringing Information to People Proceedings IEEE; 1995: IEEE.

18. Lopez-Hernandez FJ, Perez-Jimenez R, Santamaria A. Ray-tracing algorithms for fast calculation of the channel impulse response on diffuse IR wireless indoor channels. *Opt Eng.* 2000;39(10):2775-80.
19. Chowdhury MS, Zhang W, Kavehrad M. Combined deterministic and modified monte carlo method for calculating impulse responses of indoor optical wireless channels. *Journal of Lightwave Technology.* 2014;32(18):3132-48.
20. Kahn JM, Barry JR. Wireless infrared communications. *Proceedings of the IEEE.* 1997;85(2):265-98.
21. Carruthers JB, Kahn JM. Modeling of nondirected wireless infrared channels. *IEEE transactions on communications.* 1997;45(10):1260-8.
22. Gfeller FR, Bapst U. Wireless in-house data communication via diffuse infrared radiation. *Proceedings of the IEEE.* 1979;67(11):1474-86.
23. O'Brien DC, Katz M, Wang P, Kalliojarvi K, Arnon S, Matsumoto M, et al., editors. Short-range optical wireless communications. *Wireless world research forum;* 2005.
24. Kahn JM, Krause WJ, Carruthers JB. Experimental characterization of non-directed indoor infrared channels. *IEEE Transactions on Communications.* 1995;43(234):1613-23.
25. Lee K, Park H, Barry JR. Indoor channel characteristics for visible light communications. *IEEE Communications Letters.* 2011;15(2):217-9.
26. Xu F, Khalighi A, Caussé P, Bourennane S. Channel coding and time-diversity for optical wireless links. *Optics express.* 2009;17(2):872-87.
27. Zou H, Jiang H, Lu X, Xie L, editors. An online sequential extreme learning machine approach to WiFi based indoor positioning. *Internet of Things (WF-IoT), 2014 IEEE World Forum on;* 2014: IEEE.
28. Wu T-Y, Liaw G-H, Huang S-W, Lee W-T, Wu C-C. A GA-based mobile RFID localization scheme for internet of things. *Personal and Ubiquitous Computing.* 2012;16(3):245-58.
29. Alletto S, Cucchiara R, Del Fiore G, Mainetti L, Mighali V, Patrono L, et al. An indoor location-aware system for an IoT-based smart museum. *IEEE Internet of Things Journal.* 2016;3(2):244-53.
30. Yang Z, Wang Z, Zhang J, Huang C, Zhang Q, editors. Wearables can afford: Light-weight indoor positioning with visible light. *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services;* 2015: ACM.
31. Kuo Y-S, Pannuto P, Hsiao K-J, Dutta P, editors. Luxapose: Indoor positioning with mobile phones and visible light. *Proceedings of the 20th annual international conference on Mobile computing and networking;* 2014: ACM.
32. Xie B, Tan G, He T, editors. Spinlight: A high accuracy and robust light positioning system for indoor applications. *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems;* 2015: ACM.
33. Peng C, Shen G, Zhao F. Epsilon: A Visible Light Based Positioning System.
34. Zhang C, Tabor J, Zhang J, Zhang X, editors. Extending mobile interaction through near-field visible light sensing. *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking;* 2015: ACM.
35. Günes E, Ibarra G. IntraNav–Low-Cost Indoor Localization & Navigation System.

36. Rea M, Cordobés de la Calle H, Giustiniano D, Lenders V. Robust WiFi Time-of-Flight Positioning System. 2016.
37. Su W, Liu E, You J, Wang R. Precise Indoor Localization Platform Based on WiFi-GeoMagnetic Fingerprinting and Aided IMU.
38. Wang Y, Yang Q, Zhang G, Zhang P, editors. Indoor positioning system using Euclidean distance correction algorithm with bluetooth low energy beacon. Internet of Things and Applications (IOTA), International Conference on; 2016: IEEE.
39. Shimosaka M, Saisho O, Sunakawa T, Koyasu H, Maeda K, Kawajiri R. ZigBee based wireless indoor localization with sensor placement optimization towards practical home sensing. *Advanced Robotics*. 2016;30(5):315-25.
40. Dobrev Y, Vossiek M, Shmakov D, editors. A Bilateral 24 GHz wireless positioning system for 3D real-time localization of people and mobile robots. *Microwaves for Intelligent Mobility (ICMIM)*, 2015 IEEE MTT-S International Conference on; 2015: IEEE.
41. Huang Q, Yang T. A Magnetic Field Based Lightweight Indoor Positioning System for Mobile Devices. 2015.
42. Inokuchi S, editor Range-imaging system for 3D object recognition. *Proc of 7th International Conference on Pattern Recognition*, 1984; 1984.
43. Manchester code. . Available from: https://en.wikipedia.org/wiki/Manchester_code.
44. Berman SM, Greenhouse DS, Bailey IL, Clear RD, Raasch TW. Human electroretinogram responses to video displays, fluorescent lighting, and other high frequency sources. *Optometry & Vision Science*. 1991;68(8):645-62.
45. Open Sound Control. Available from: <http://opensoundcontrol.org/introduction-osc>.
46. Lymberopoulos D, Liu J, Yang X, Choudhury RR, Handziski V, Sen S, editors. A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned. *Proceedings of the 14th international conference on information processing in sensor networks*; 2015: ACM.
47. Lymberopoulos D, Liu J. The Microsoft Indoor Localization Competition: Experiences and Lessons Learned. *IEEE Signal Processing Magazine*. 2017;34(5):125-40.
48. Beder C, Klepal M, editors. Fingerprinting based localisation revisited: A rigorous approach for comparing RSSI measurements coping with missed access points and differing antenna attenuations. *Indoor Positioning and Indoor Navigation (IPIN)*, 2012 International Conference on; 2012: IEEE.
49. Dahlgren E, Mahmood H. Evaluation of indoor positioning based on Bluetooth Smart technology. Master of Science Thesis in the Programme Computer Systems and Networks. 2014.
50. Hammer F, Pichler M, Fenzl H, Gebhard A, Hesch C. An acoustic position estimation prototype system for underground mining safety. *Applied Acoustics*. 2015;92:61-74.
51. Quantitec Intranav. Available from: <https://intranav.com>.
52. Dissanayake MG, Newman P, Clark S, Durrant-Whyte HF, Csorba M. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on robotics and automation*. 2001;17(3):229-41.
53. Thrun S, Burgard W, Fox D. *Probabilistic robotics*: MIT press; 2005.
54. Lemaire T, Berger C, Jung I-K, Lacroix S. Vision-based slam: Stereo and monocular approaches. *International Journal of Computer Vision*. 2007;74(3):343.

55. Sola J, editor Multi-camera VSLAM: from former information losses to self-calibration. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Diego, CA, USA; 2007.
56. Davison AJ, Reid ID, Molton ND, Stasse O. MonoSLAM: Real-time single camera SLAM. IEEE transactions on pattern analysis and machine intelligence. 2007;29(6):1052-67.
57. Huh S, Shim DH, Kim J, editors. Integrated navigation system using camera and gimbaled laser scanner for indoor and outdoor autonomous flight of UAVs. Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on; 2013: IEEE.
58. Liu M, Pradalier C, Pomerleau F, Siegwart R, editors. The role of homing in visual topological navigation. Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on; 2012: IEEE.
59. Ko DW, Kim YN, Lee JH, Suh IH, editors. A scene-based dependable indoor navigation system. Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on; 2016: IEEE.
60. Fraundorfer F, Engels C, Nistér D, editors. Topological mapping, localization and navigation using image collections. Intelligent Robots and Systems, 2007 IROS 2007 IEEE/RSJ International Conference on; 2007: IEEE.
61. Lin Z, Kim S, Kweon IS, editors. Recognition-based indoor topological navigation using robust invariant features. Intelligent Robots and Systems, 2005(IROS 2005) 2005 IEEE/RSJ International Conference on; 2005: IEEE.
62. Bandera A, Urdiales C, Sandoval F, editors. An hierarchical approach to grid-based and topological maps integration for autonomous indoor navigation. Intelligent Robots and Systems, 2001 Proceedings 2001 IEEE/RSJ International Conference on; 2001: IEEE.
63. Maddern W, Milford M, Wyeth G, editors. Towards persistent indoor appearance-based localization, mapping and navigation using cat-graph. Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on; 2012: IEEE.
64. Booi O, Terwijn B, Zivkovic Z, Krose B, editors. Navigation using an appearance based topological map. Robotics and Automation, 2007 IEEE International Conference on; 2007: IEEE.
65. Beinhofer M, Müller J, Krause A, Burgard W, editors. Robust landmark selection for mobile robot navigation. Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on; 2013: IEEE.
66. Yoon K-J, Kweon I-S, editors. Artificial landmark tracking based on the color histogram. Intelligent Robots and Systems, 2001 Proceedings 2001 IEEE/RSJ International Conference on; 2001: IEEE.
67. Tsukiyama T. RFID based navigation system for indoor mobile robots. IFAC Proceedings Volumes. 2011;44(1):1084-9.
68. George L, Mazel A, editors. Humanoid robot indoor navigation based on 2D bar codes: Application to the NAO robot. Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on; 2013: IEEE.
69. Zheng R, Yuan K, editors. MR code for indoor robot self-localization. Intelligent Control and Automation, 2008 WCICA 2008 7th World Congress on; 2008: IEEE.
70. User's Guide Localization system StarGazer™ for Intelligent Robots. Available from: <http://www.hagisonic.com>.

71. Lee S, Song J-B, editors. Use of coded infrared light as artificial landmarks for mobile robot localization. Intelligent Robots and Systems, 2007 IROS 2007 IEEE/RSJ International Conference on; 2007: IEEE.
72. Mautz R, Tilch S, editors. Survey of optical indoor positioning systems. Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on; 2011: IEEE.
73. Esp8266. Available from: <https://espressif.com/en/products/hardware/esp8266ex/overview>.
74. Adept. Available from: http://www.mobilerobots.com/Mobile_Robots.aspx.
75. Hagisomic. User's Guide Localization System StarGazer™ for Intelligent Robots 2008. Available from: <http://www.hagisomic.com>.
76. Robotics E. 2010. Available from: <http://www.evolution.com>.
77. Batalin MA, Sukhatme GS, Hattig M, editors. Mobile robot navigation using a sensor network. Robotics and Automation, 2004 Proceedings ICRA'04 2004 IEEE International Conference on; 2004: IEEE.
78. Canny J. The complexity of robot motion planning: MIT press; 1988.
79. Cao YU, Fukunaga AS, Kahng A. Cooperative mobile robotics: Antecedents and directions. Autonomous robots. 1997;4(1):7-27.
80. Van Den Berg JP, Overmars MH, editors. Prioritized motion planning for multiple robots. Intelligent Robots and Systems, 2005(IROS 2005) 2005 IEEE/RSJ International Conference on; 2005: IEEE.
81. Regele R, Levi P, editors. Cooperative multi-robot path planning by heuristic priority adjustment. Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on; 2006: IEEE.
82. Buckley SJ, editor Fast motion planning for multiple moving robots. Robotics and Automation, 1989 Proceedings, 1989 IEEE International Conference on; 1989: IEEE.
83. Bennewitz M, Burgard W, Thrun S. Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots. Robotics and autonomous systems. 2002;41(2):89-99.
84. Saha M, Isto P, editors. Multi-robot motion planning by incremental coordination. Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on; 2006: IEEE.
85. Ryan MRK. Exploiting subgraph structure in multi-robot path planning. Journal of Artificial Intelligence Research. 2008;31:497-542.
86. Peasgood M, McPhee J, Clark C. Complete and scalable multi-robot planning in tunnel environments. IFAC Proceedings Volumes. 2006;39(20):26-31.
87. Wang K-HC, Botea A, editors. Tractable Multi-Agent Path Planning on Grid Maps. IJCAI; 2009: Pasadena, California.
88. Surynek P, editor An application of pebble motion on graphs to abstract multi-robot path planning. Tools with Artificial Intelligence, 2009 ICTAI'09 21st International Conference on; 2009: IEEE.
89. Sturtevant NR, Buro M, editors. Improving Collaborative Pathfinding Using Map Abstraction. AIIDE; 2006: Marina del Rey.
90. Diestel R. Graph theory: Springer-Verlag Berlin and Heidelberg GmbH & amp; 2000.
91. BlueCreation BC118. Available from: <http://www.digikey.com/product-detail/en/bluecreation/BC118/1495-1003-1-ND/4860037>.