

WIRELESS VISUAL SENSOR NETWORK FOR 3D SCENE PERCEPTION
AND RECONSTRUCTION



A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy in Telecommunication
and Computer Engineering
Suranaree University of Technology
Academic Year 2021

การรับรู้และสร้างคืนภาพสามมิติบนเครือข่ายรับภาพแบบไร้สาย



นายวัชรพงศ์ อยู่ขวัญ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคมและคอมพิวเตอร์

มหาวิทยาลัยเทคโนโลยีสุรนารี

ปีการศึกษา 2564

WIRELESS VISUAL SENSOR NETWORK FOR 3D SCENE PERCEPTION
AND RECONSTRUCTION

Suranaree University of Technology has approved this thesis submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy.

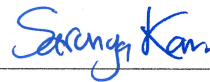
Thesis Examining Committee



(Asst. Prof. Dr. Krisana Chinnasarn)
Chairperson



(Asst. Prof. Dr. Paramate Horkaew)
Member (Thesis Advisor)



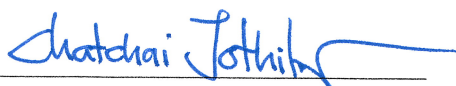
(Asst. Prof. Dr. Sarunya Kanjanawattana)
Member



(Assoc. Prof. Dr. Monthippa Uthansakul)
Member



(Assoc. Prof. Dr. Peerapong Uthansakul)
Member



(Assoc. Prof. Dr. Chatchai Jothityangkoon)
Vice Rector for Academic Affairs
and Quality Assurance



(Assoc. Prof. Dr. Pornsiri Jongkol)
Dean of Institute of Engineering

วัชรพงศ์ อยู่ขวัญ : การรับรู้และสร้างคืนภาพสามมิติบนเครือข่ายรับภาพแบบไร้สาย
(WIRELESS VISUAL SENSOR NETWORK FOR 3D SCENE PERCEPTION AND
RECONSTRUCTION) อาจารย์ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ ดร.ประเมศวร์ ห่อแก้ว,
104 หน้า.

คำสำคัญ: การประมาณความลึก/เอ็นโทรปี/การสร้างคืนภาพสามมิติ/การเทียบจุดสามมิติ

การประมาณค่าระยะลึกของฉากในภาพถ่ายเป็นขั้นตอนวิธีที่มีความสำคัญสำหรับการสร้าง
คืนภาพสามมิติ โดยทั่วไปแล้วในขั้นตอนนี้จะใช้เครื่องมือวัดระยะที่มีความแม่นยำสูงโดยตรงหรือการใช้
กล้องสเตอริโอ แต่ด้วยความซับซ้อนของภาพ การถูกวัตถุบดบัง รวมไปถึงสภาวะแสงที่มีผลกับการ
ถ่ายภาพ ทำให้การสร้างคืนพื้นผิวรวมไปถึงรูปร่างของวัตถุไม่สมบูรณ์หรือข้อมูลภาพมีการสูญหาย ใน
การแก้ไขความไม่สมบูรณ์ดังกล่าว จึงมีการใช้สถาปัตยกรรมแบบคอนโวลูชันในการสกัดระยะลึกจาก
ภาพสีในการสร้างพื้นผิวของภาพสามมิติเพิ่มเติม แต่โครงสร้างของคอนโวลูชันมีความคลุมเครือสูงทำ
ให้ยังคงเกิดข้อผิดพลาดขึ้น เพื่อแก้ไขปัญหาดังกล่าว งานวิจัยนี้จึงนำเสนอขั้นตอนวิธีในการพิพจน์
ระหว่างพอยท์คลาวด์ที่สกัดจากภาพระยะลึกที่ได้จากกล้องอินฟราเรดและการประมาณระยะจาก
ภาพสีโดยการดัดแปลง ResNet-50 เพื่อเพิ่มความแม่นยำในขั้นตอนการพิพจน์ Cross-Entropy
Iterative Closest Point (CEICP) ถูกใช้เพื่อเพิ่มประสิทธิภาพในการรวมพอยท์คลาวด์ทั้งสองชุดเข้า
ด้วยกัน จากนั้น ทำการประเมินผลการทดลองโดยใช้ฐานข้อมูลสาธารณะ ผลลัพธ์ที่ได้ทำให้เห็นว่า
ขั้นตอนวิธีที่นำเสนอมีประสิทธิภาพในการสร้างคืนภาพสามมิติ นอกจากนี้ ขั้นตอนวิธีถูกนำไป
ประมวลผลบนแบบจำลองเครือข่ายเน็ตเวิร์คแบบไร้สายเพื่อวัดประสิทธิภาพการทำงานของเครือข่าย
อีกด้วย

สาขาวิชา วิศวกรรมคอมพิวเตอร์
ปีการศึกษา 2564

ลายมือชื่อนักศึกษา วัชรพงศ์ อยู่ขวัญ
ลายมือชื่ออาจารย์ที่ปรึกษา ประเมศวร์ ห่อแก้ว

WATCHARAPHONG YOONWAN : WIRELESS VISUAL SENSOR NETWORK FOR 3D
SCENE PERCEPTION AND RECONSTRUCTION. THESIS ADVISOR : ASST. PROF.
PARAMATE HORKAEW, Ph.D., 104 PP.

Keywords: Depth Estimation/Entropy/lcp/Photogrammetry/Scene Reconstruction

The assessment of depth is a critical component of 3D scene comprehension. The majority of conventional systems rely on direct sensing of this data via photogrammetry or stereo imaging. As the complexity of the pictures increased, these modalities were hindered by factors such as occlusion and inadequate lighting conditions, etc. Due to the absence of data, rebuilt surfaces are typically left with voids. Consequently, surface regularization is frequently necessary as post-processing. With recent advancements in deep learning, depth inference from a monocular image has garnered significant interest. With promising results, numerous convolutional architectures have been developed to derive depth information from a monocular image. These networks may have learned and generalized confusing visual cues, resulting in imprecise estimate. This study provides an efficient method for merging point clouds recovered from depth values directly detected by an infrared camera and estimated using a modified ResNet- 50 from an RGB image of the same scene in order to address these concerns. CEICP, an information-theoretic alignment approach, was devised in order to assure the robustness and efficacy of detecting the correspondence between and aligning these point clouds. The experimental findings on a publicly available dataset revealed that the suggested method outperformed its competitors while creating high-quality surfacer representations of the underlying picture.

School of Computer Engineering
Academic Year 2021

Student's Signature ยอน อาน
Advisor's Signature พารมาเต หอคาอว

ACKNOWLEDGEMENT

This project would not have been possible without the support of many people. Many thanks to my adviser, Asst. Prof. Dr. Paramate Horkaew, who read my numerous revisions and helped make some sense of the confusion. Also thanks to my committee members, Asst. Prof. Dr. Krisana Chinnasarn, Asst. Prof. Dr. Sarunya Kanjanawattana, Assoc. Prof. Dr. Monthippa Uthansakul, and Assoc. Prof. Dr. Peerapong Uthansakul, who offered guidance and support. Prof. Dr. Chakchai So-in who offered a financial support.

This work was supported in part by the National Research Council of Thailand (NRCT) through the International Research Network Program under Grant IRN61W0006.

Watcharaphong Yookwan

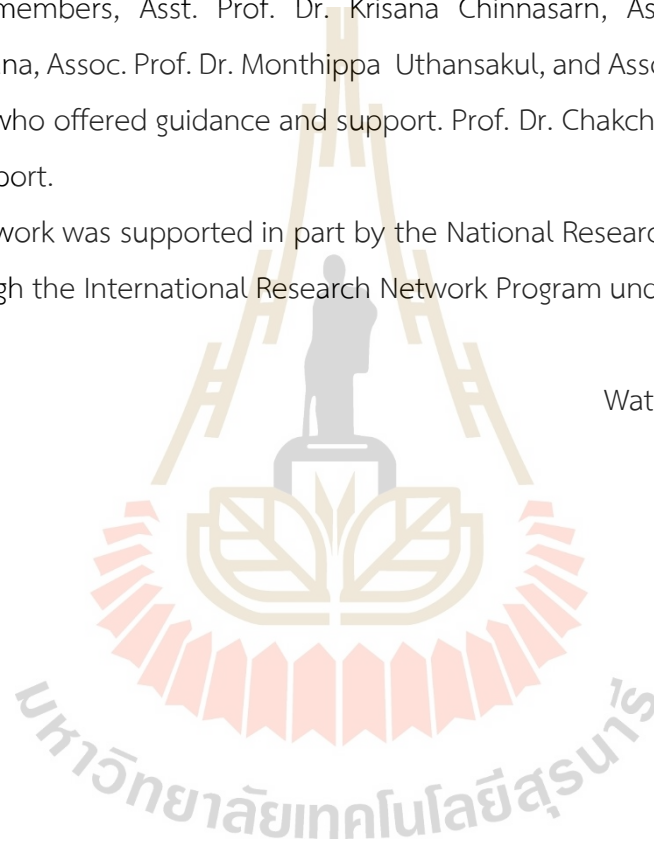


TABLE OF CONTENTS

	Page
ABSTRACT (THAI).....	I
ABSTRACT (ENGLISH).....	II
ACKNOWLEDGEMENT	III
TABLE OF CONTENTS.....	IV
LIST OF TABLES.....	VII
LIST OF FIGURES.....	VIII
CHAPTER	
1 INTRODUCTION	1
1.1 INTRODUCTION.....	1
1.2 STATEMENT OF PROBLEM.....	3
1.3 CONTRIBUTION.....	5
1.4 RESEARCH OBJECTIVE.....	6
1.5 SCHEDULE.....	6
2 BACKGROUND KNOWLEDGE	7
2.1 DEVICE AND INFRA-STRUCTURE.....	7
2.1.1 RGB-D Camera.....	7
2.1.1.1 Kinect™ Camera.....	8
2.1.1.2 Intel RealSense.....	9
2.1.2 Wireless Sensor Network.....	10
2.1.3 Wireless Sensor Network Simulator.....	11
2.1.4 Data transferring.....	12
2.1.5 Cloud Server.....	13
2.2 THEORETICAL BACKGROUND.....	14
2.2.1 Camera Calibration.....	14

TABLE OF CONTENTS (Continued)

	Page
2.2.1.1 Camera Model.....	15
2.2.1.2 Pinhole Camera.....	16
2.2.1.3 Camera Calibration Parameter	18
2.2.1.4 Extrinsic Parameter	19
2.2.1.5 Intrinsic Parameter.....	19
2.2.1.6 Distortion Camera Parameter	20
2.2.1.7 Radial Distortion Parameter	20
2.2.3 Point Cloud.....	21
2.2.4 Point Cloud Registration.....	22
2.2.5 Three-dimensional reconstruction	23
2.2.5.1 Polygon Meshes	24
2.2.5.2 Surface Reconstruction.....	25
2.3 LITERATURE REVIEW.....	27
2.3.1 Image information transferring and Data Communication	29
2.3.2 Camera Calibration.....	32
2.3.3 Image registration and Three-Dimensional Reconstruction	36
2.3.4 Surface Reconstruction.....	47
2.3.5 Wireless Visual Sensor Network Simulation	48
3 METHODOLOGY.....	52
3.1 Experiment Environment Set-up	53
3.1.1 RGB-D Camera set-up.....	53
3.2 CAMERA CALIBRATION.....	56
3.3 POINT CLOUD ACQUISITION.....	59
3.2.1 Point Cloud Extraction	59
3.2.2 Point Cloud Noise Reduction.....	60
3.3 DEPTH ESTIMATION FROM RGB	61
3.3.1 CNN Architecture.....	61

TABLE OF CONTENTS (Continued)

	Page
3.4 POINT CLOUD FUSION.....	63
3.4.1 Point Cloud Registration.....	64
3.4.2 Traditional Iterative Closest Point Registration (ICP).....	64
3.4.3 Cross-Entropy Iterative Closest Point.....	66
3.5 SURFACE RECONSTRUCTION.....	71
3.6 WIRELESS VISUAL SENSOR NETWORK SIMULATION.....	74
3.6.1 Type of Simulation.....	75
3.6.2 Wireless Visual Sensor Network Structure.....	76
4 EXPERIMENTAL RESULTS.....	79
4.1 CAMERA CALIBRATION RESULT.....	79
4.2 DEPTH ESTIMATION RESULT.....	81
4.3 POINT CLOUD FUSION RESULT.....	82
4.4 WIRELESS VISUAL SENSOR NETWORK RESULT.....	89
5 CONCLUSION AND DISCUSSION.....	93
5.1 CONCLUSION.....	93
5.2 DISCUSSION AND FUTURE WORKS.....	93
REFERENCES.....	96
APPENDIX I.....	102
BIOGRAPHY.....	104

LIST OF TABLES

Table	Page
1.1	Grant chart of dissertation..... 6
3.1	Traditional Iterative Closest Point 66
4.1	Intrinsic Parameters 80
4.2	Extrinsics Parameter..... 80
4.3	Depth Estimation Result..... 81
4.4	Point cloud fusion result comparison..... 86
4.5	Fusion result from different angle of camera placement using proposed approach..... 88
4.6	Average delivery time against No. Of sent images..... 90
4.7	Parameter setting and Controlled simulation..... 90

LIST OF FIGURES

Figures	Page
1.1 Wireless Visual Sensor Network Structure	3
1.2 A traditional 3D scene reconstruction pipeline would begin with the extraction of a point cloud from a series of depth photos, followed by triangulation, and then hole filling would typically come next in the process.....	4
1.3 The overarching idea of the cloud fusion technique that's being presented.....	5
2.1 Front-view of Kinect Camera.....	9
2.2 Intel RealSense.....	9
2.3 Example of simple wireless sensor network architecture	10
2.4 Example data transmitting structure.....	13
2.5 High level of cloud storage architecture.....	14
2.6 Overview of camera calibration (Zhang, 2020).....	15
2.7 Camera Model (Matlab, 2021).....	16
2.8 Pinhole Camera Model (Matlab, 2021).....	17
2.9 Image point and Real-World Point (MatlabTM, 2021)	18
2.10 Overview of World point and Image Point (Matlab, 2021).....	18
2.11 Explanation of Rotational and Translation Matrix (MatlabTM, 2021).....	19
2.12 Explanation of Skew (Remondino, F., 2006).....	20
2.13 Explanation of Distortion (Matlab, 2020).....	20
2.14 Example of unstructured Point cloud	22
2.15 Example of point cloud registration process using correspondence point.....	23
2.16 Three-Dimensional Reconstruction	24
2.17 Example of Polygon Meshes (Wikipedia contributor, 2021).....	25
2.18 Example of NURBS Surface model.....	26

LIST OF FIGURES (Continued)

Figures		Page
2.19	Overview of device and reconstruction techniques.....	27
2.20	PCB structure of image transferring (J. Loret).....	29
2.21	Wireless sensor network for proposed system.....	30
2.22	Overview of agricultural monitoring system.....	31
2.23	Experimental Result (Hold-Geoffroy. 2018)	33
2.24	Experimental Result (Y. Liu, 2018).....	34
2.25	Calibration Result (J. Huang, 2018).....	35
2.26	RGB-D tracking overall approach	37
2.27	Point Cloud Experimental Result (Z. Zhang, 2011).....	38
2.28	Reconstruction Result (Z. Chang, 2020)	39
2.29	Reconstruction Result (M. Lhuler, 2018)	40
2.30	Experimental Result (Chang, 2018).....	42
2.31	Experimental Result (C. Mineo, 2018).....	44
2.32	Experimental Result (K. Wang, 2014)	45
2.33	Experimental Result (Y. Cui, 2014).....	46
2.34	Experimental Result (R. A Newcombe, 2011).....	47
2.35	Three-dimensional tin surface.....	48
2.36	Random network configuration (Joao P. et. al, 2017)	49
2.37	Functional Architecture (Ahmed M et. al., 2009)	51
3.1	Overview of proposed framework.....	53
3.2	Pictures of system configuration, illustrating a Kinect TM for Xbox One TM (left) and a person holding a planar object (right).	54
3.3	WSN (Simulated) Structure for reconstruction System	55
3.4	Example of RGB-D input.....	55
3.5	Calibration plane.....	58
3.6	Camera position visualization.....	59
3.7	Result of Point cloud Acquisition process.....	60

LIST OF FIGURES (Continued)

Figures		Page
3.8	Visualization of ResNet-50 for Depth Estimation.....	61
3.9	Example of Depth Estimation Result	63
3.10	ICP Fundamental.....	65
3.11	Example of Fusion Result of indoor scene.....	70
3.12	Fusion result comparing with the original point cloud	71
3.13	Ball-pivot Algorithm (BPA)	72
3.14	from Ball pivot algorithm surface reconstruction before fusion.....	73
3.15	Result from Ball pivot algorithm surface reconstruction.....	74
3.16	Wireless Visual Sensor Network Architecture.....	77
3.17	3D Scene reconstruction on Wireless Visual Sensor Network Mechanism	78
4.1	Visualization of evaluation of depth estimation. Errors of the estimated depth by ResNet-50 from 3 scenes. Despite consistently low errors, they were inferior to direct extraction.....	82
4.2	Result of BPA with $r=0.0102$ (Before fusion).....	83
4.3	Result of BPA with $r=0.0131$ (Before fusion).....	84
4.4	Result of BPA with $r=0.0156$ (Before fusion).....	84
4.5	Result of BPA with $r=0.0209$ (Before fusion).....	85
4.6	Result of BPA with $r=0.0102$ (After fusion).....	85
4.7	Result of BPA with $r=0.0156$ (After fusion).....	86
4.8	Visualization of result comparison of recent method and the proposed approach.....	87
4.9	Convergence of the proposed approach comparing with the existing method.	88

LIST OF FIGURES (Continued)

Figures		Page
4.10	The average number of photos acquired per minute by the data collector using the BASELINE, GREEDY, and PDC schemes in comparison to the total number of objects (each image is conveyed in a 1000B-long packet).....	92



CHAPTER 1

INTRODUCTION

1.1 Introduction

In a wide variety of scientific and engineering applications, such as computer-aided geometric design (CAGD), graphics, computer vision, medical image analysis, computational modeling, augmented reality (AR), and digital multimedia, etc., the reconstruction of a three-dimensional (3D) scene is an essential component (R. J. Wilson, 1988). For example, in computer-aided diagnosis (CAD), the 3D information on an anatomical shape and its peripherals, possibly with associated lesions, reconstructed from the tomographic scan of a patient, are of great clinical value (S. Lee, et. al., 2005). This information can be helpful in determining whether or not a patient has a disease (D.C. Le, 2021). If a doctor is given this information, they will be able to establish an accurate diagnosis, as well as determine the prognosis for the condition, and they will also be able to execute therapeutic intervention. The topographic features of an underlying terrain can be determined through the use of a digital elevation model (DEM) in the field of remote sensing (RS).

Photogrammetry techniques, such as aerial laser altimetry, synthetic aperture radar, and other methods can be utilized to generate DEMs (Florinsky, 2003). A series of three-dimensional points, also known as a point cloud, sampled from an object's or form's surface can be used to characterize the object or shape. In most cases, the position of each point may be precisely determined by its Cartesian coordinate, which is written as (x, y, z) or by the equation $\mathbf{p} = [x \ y \ z]^T$. Typical photogrammetry can collect these three-dimensional points on an object's surface using either a passive or active approach. Examples of active approaches are structured light scanners and Light Detection and Ranging (LiDAR) scanners. The most recent breakthroughs in optical sciences have made it possible for us to tackle reverse engineering and rapid prototyping in a substantially more efficient manner, thanks in large part to these scanners. To this point, the accuracy of their reconstruction has

improved in tandem with the fidelity of the results provided by the most recent CAD application software. This has made it possible to speed the convergence of these essential technologies, which has led to a widespread use of those technologies in computer graphics and vision, specifically in the modeling, recognition, and analysis of real-world settings. As a result, three-dimensional scanners have been put to use in a wide variety of applications across all fields of data-driven science and at a variety of scales. Even up until the very moment that Metaverse was brought into existence, the vast proliferation of computerized point cloud analyses had already been triggered and attained by competing inventions and commercialization of low-cost real-time scanners such as the Microsoft™ Kinect (C. V. Nguyen, 2012). This was accomplished even before Metaverse came into existence. These innovations have had a significant impact on many different areas of research and development, such as the automotive industry, the design of machinery and artificial organs, archaeology, the military and defense, urban planning, and digital laboratories, amongst others.

The estimation of three dimensions using stereo vision has garnered a significant amount of interest, mostly as a result of the relatively low costs and amounts of resources required for its implementation. However, the quality of its estimation is greatly dependent on exact dense connection between cameras, which is in turn dependent on the scene. It's possible that holes or an inaccurate topology will be reconstructed as a result of mismatched pixels due to the features of the image texture. In contrast, monocular depth estimation (also known as MDE) draws conclusions about depth from a single image by analyzing motion or visual cues (Zhang et. al, 2013). Its primary benefit is that it eliminates the requirement to calibrate the alignment between cameras and, as a result, the mistakes that are caused by doing so. At this point, it does not have the ability to make certain judgments or have particular mental perceptions, both of which are often acquired through human experience. Due to these many features, it is no longer acceptable for use in shape critical applications. These advantages and disadvantages, as a result, served as the impetus for this research towards integrating depths determined from a variety of modalities.

A 3D reconstruction on a wireless visual sensor network is proposed in this study. The network's basic framework is depicted on

Figure 1.1 The algorithm to fusion three-dimensional point cloud with occluded area was created to complete the reconstruction with incomplete area.

Figure 1.1 depicts the overall layout for simulating a wireless vision sensor network.

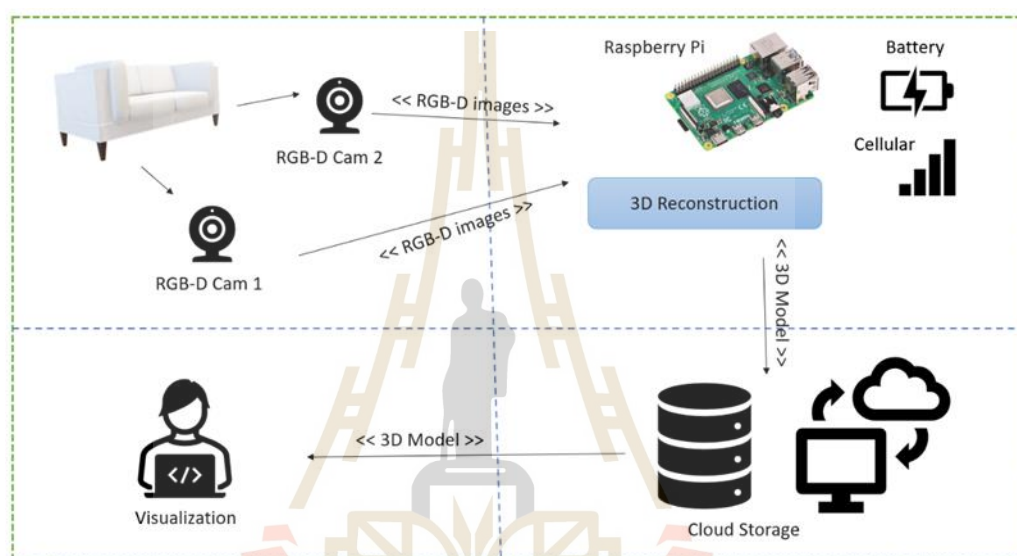


Figure 1.1 Wireless Visual Sensor Network Structure

The implementation of a wireless sensor network was divided into three levels in the simulation. Real-world object information is acquired in the perception layer utilizing an RGB-D camera (RGB image with Depth information) that is attached. The data is sent to a cloud server for storage via a cellular module. Finally, the collected data is sent to a local client for reconstruction of a three-dimensional model.

1.2 Statement of Problem

Because to occlusion and/or insufficient sampling, the majority of reconstructed scenes almost always have gaps in them, regardless of the cloud acquisition methods or approximation techniques that were utilized. As a direct result

of this, hole filling is frequently a necessary step in the post-processing stage, as shown in Figure 1.2

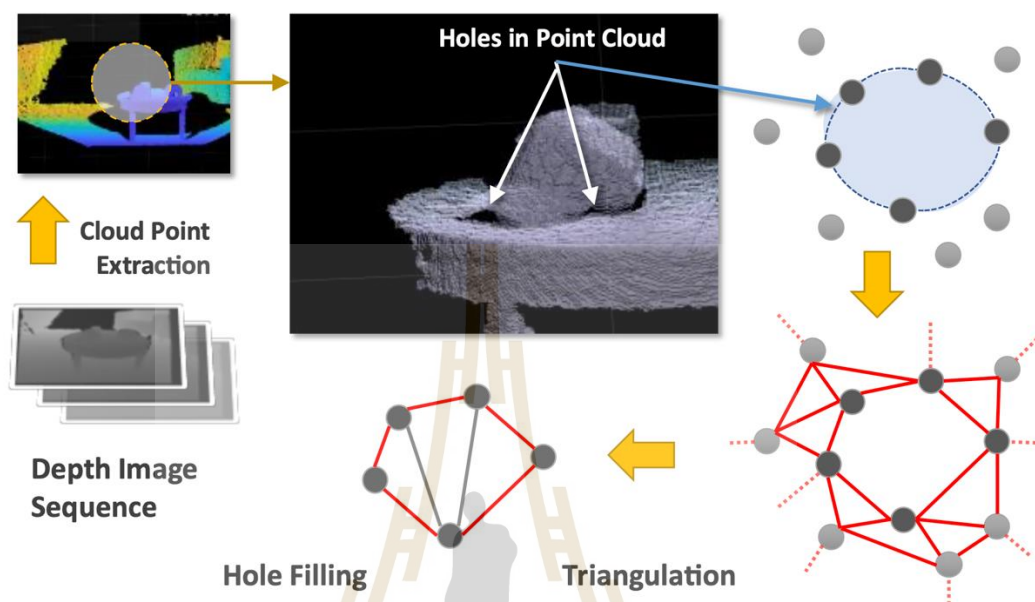


Figure 1.2 A traditional 3D scene reconstruction pipeline would begin with the extraction of a point cloud from a series of depth photos, followed by triangulation, and then hole filling would typically come next in the process.

In order to solve these problems, the authors of this study propose an approach to the reconstruction of 3D scenes by fusing information theoretically between point clouds that were gathered using several modalities (Z. Wang et. al, 2020). To be more specific, in order to reduce the need for hole filling, cloud points that were extracted from a depth scan were aligned and combined with those that were learned by a convolutional neural network (CNN) called ResNet-50 from an RGB image of the same scene. This was done in order to create a more accurate representation of the scene. Because the pixels in a color image are continuously distributed, the 3D points that are retrieved from their estimated depth can fill any hole that was present in the original point set, prior to the surface reconstruction. We updated ICP by including cross entropy (CE) function, which we later referred to as

CEICP, in order to robustly align these point sets. This modification was motivated by the work that Tsing and Kanade had done. Following that, a BPA was utilized in order to approximate the surface. The rolling ϵ -shape technique would correct any errors in the image-to-depth estimate process that may have occurred.

Figure 1.3 provides a concise summary of the overall approach that has been suggested.

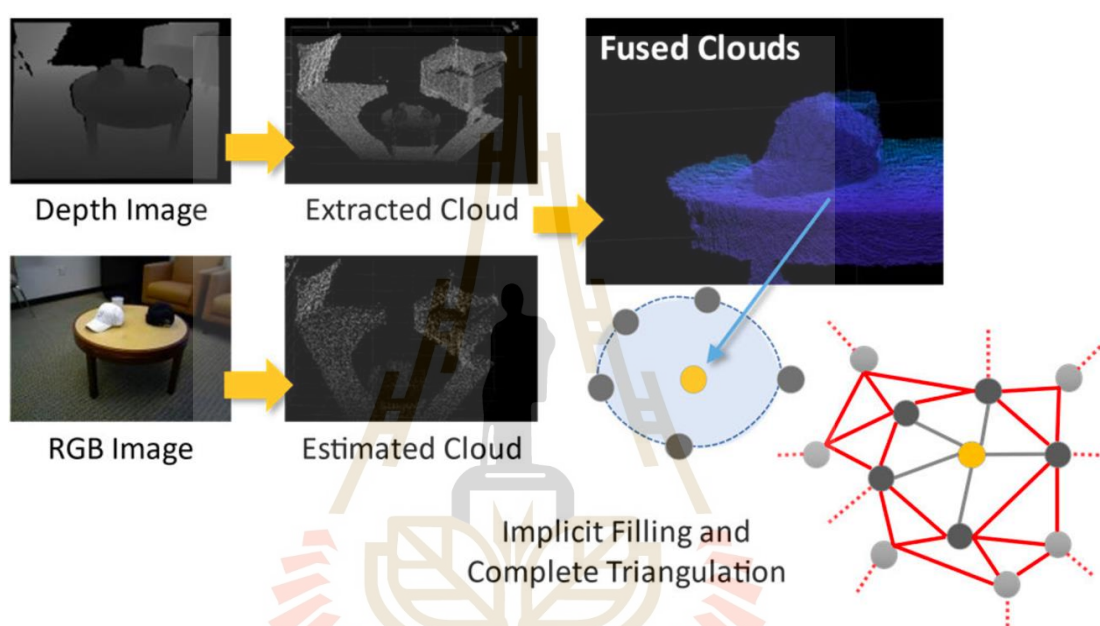


Figure 1.3 The overarching idea of the cloud fusion technique that's being presented.

1.3 Contribution

- In this research, the fusion approach of the point cloud which is extracted from RGB images and depth images is proposed.
- To increase accuracy and reduce computational time of the algorithm, the fusion approach consists of modified iterative closest point using cross-entropy.

1.4 Research Objective

- Get an algorithm to reconstruct a three-dimensional point cloud from RGB-D images.
- Get a simulation of Wireless Visual Sensor Network system (WWSN) for handling three-dimensional scene perception and reconstruction.

1.5 Schedule

Table 1.1 Grant chart of dissertation

N o.	Task	Dec-21				Jan-22				Feb-22				Mar-22				Apr-22				May-22				Jun-22				Jul-22			
		w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
1	Review point cloud acquisition	█	█	█	█																												
2	Review Point cloud registration					█	█	█	█	█	█	█	█																				
3	Reconstruct RGB-D information to 3D point cloud scene									█	█	█	█	█	█	█	█	█	█	█	█												
4	Simulate WWSN architecture																					█	█	█	█								
5	Thesis Défense																									█	█	█	█				

CHAPTER 2

BACKGROUND KNOWLEDGE

In this section the background knowledge and literature review were performed. To complete the research, some theoretical background and literature review needed be described.

2.1 Device and Infra-Structure

In this research, the wireless sensor network (WSN) of hand-held three-dimensional model reconstruction system was designed. RGB-D camera was employed to collect image sequence and depth information. The collected data was firstly store in Raspberry Pi integrated with cellular module and individual battery. Then, the information is transferred wirelessly to cloud storage server for further process.

2.1.1 RGB-D Camera

RGB-D Sensors are a special sort of depth-sensing devices that work in combination with a camera that contains an RGB (red, green, and blue color) sensor. Such a camera detects red, green, and blue light in addition to depth. These sensors determine how far away an item is from the camera they are attached to. They are able to improve the conventional picture by adding depth information to it on a per-pixel basis. Depth information is information that is connected to the distance to the sensor. The fields of computer vision and computer graphics have been pushed in recent years to examine novel approaches that are founded on RGB-D photographs as a result of depth sensors. The depth information has the potential to make a significant contribution to the solution or simplification of a great number of challenging tasks. Some examples of these tasks include object detection, scene parsing, pose estimation, visual tracking, semantic segmentation, shape analysis, image-based rendering, and 3D reconstruction, to name just a few. For instance, after the depth formation of the scene has been achieved, the corresponding 3D model may be immediately constructed by using a mapping strategy. This can be done once the

depth formation of the scene has been obtained. Because of the use of mapping, this is now feasible. To put it another way, in order to get a high-quality geometric model, depth-based three-dimensional reconstruction does not demand that you carry out the routines of structure from motion (SFM), which is a method that is notoriously difficult to execute. As a result, the organizations that are now working on 3D reconstruction have an opportunity presented by RGB-D sensors. Consequently, an RGB-D Camera need to be taken into consideration. The RGB-D Cameras are the focus of this research project's investigation.

2.1.1.1 Kinect™ Camera

Kinect is a line of motion sensing input devices produced by Microsoft and first released in 2010 (TechTarget, 2021). The devices generally contain RGB cameras, and infrared projectors and detectors that map depth through either structured light or time of flight calculations, which can in turn be used to perform real-time gesture recognition and body skeletal detection, among other capabilities.

Kinect was originally developed as a motion controller peripheral for Xbox video game consoles, distinguished from competitors (such as Nintendo's Wii Remote and Sony's PlayStation Move) by not requiring physical controllers. The first-generation Kinect was based on technology from Israeli company PrimeSense and unveiled at E3 2009 as a peripheral for Xbox 360 codenamed "Project Natal". It was first released on November 4, 2010. The component of the Kinect Camera can be demonstrated in Figure 2.1 It consists of microphone array, status indicator LED, #D Depth sensor and vision camera.

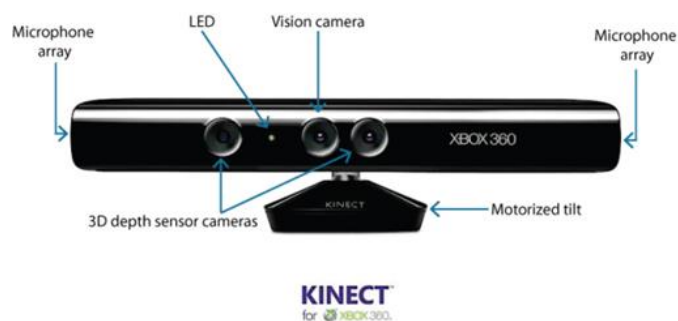


Figure 2.1 Front-view of Kinect Camera

2.1.1.2 Intel RealSense

The Intel RealSense Technology product line is a suite of depth and tracking technologies that was created to provide machines and other electronic devices the ability to have a feeling of depth (Plitch A., 2015). The technologies, which are owned by Intel, are applied in a wide range of goods catering to a large market. These products include autonomous drones, robotics, augmented reality and virtual reality (AR/VR), and smart home devices. Vision Processors, Depth and Tracking Modules, and Depth Cameras are the individual components that make up the RealSense product. It is supported by a Software Development Kit (SDK) that is open source and cross-platform. This makes it simpler for third-party software developers, system integrators, original design manufacturers, and original equipment manufacturers to provide support for cameras. An example of an Intel RealSense depth camera is shown in Figure 2.2 which gives a demonstration of the device.



Figure 2.2 Intel RealSense

In the system design, the image information is collected using RGB-D camera (RGB and Depth Information). Then, the collected data is firstly stored in the visual node i before transmitting to reconstruct in the cloud server through wireless visual sensor network simulation.

2.1.2 Wireless Sensor Network

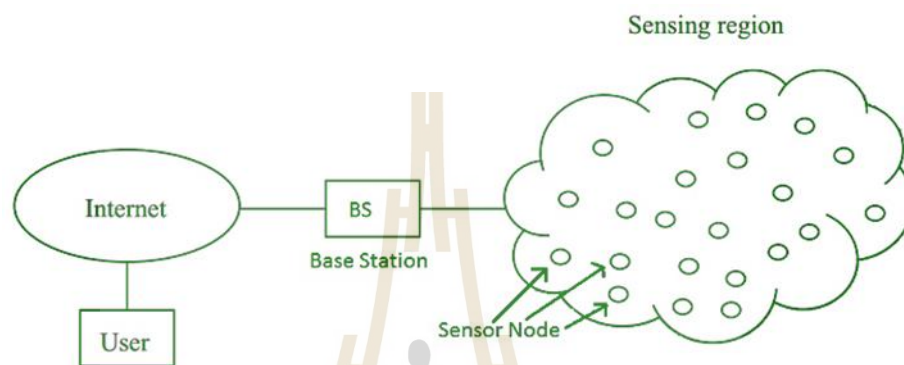


Figure 2.3 Example of simple wireless sensor network architecture

Wireless sensor networks, also known as WSNs, are networks that consist of sensors that are both spatially scattered and dedicated (Figure 2.3).

These sensors monitor and record the environmental conditions and then transmit the data that they have gathered to a centralized point. WSNs have the capability to measure aspects of the environment, including temperature, sound, levels of pollution, humidity, and wind.

These are very similar to wireless ad hoc networks in the sense that they allow sensor data to be transmitted wirelessly by relying on wireless connectivity and the spontaneous development of networks. The likes of temperature, sound, and pressure are some of the physical or environmental parameters that WSNs keep an eye on. Contemporary networks are bi-directional, meaning they can both gather data and enable users to manage the operation of sensors. Military applications, including as spying on the battlefield, were a driving force behind the development of these networks. These types of networks are utilized in a variety of commercial and

consumer applications, such as the monitoring and control of industrial processes and the health monitoring of machines.

2.1.3 Wireless Sensor Network Simulator

As embedded system and network technology has advanced, there has been an increasing interest in developing low-power devices that can provide fine-grained metering and control of living environments. This interest has been spurred on by the development of embedded system technology. Wireless Sensor Networks (WSNs), which are comprised of self-configurable sensors that are dispersed throughout space, are an ideal solution that satisfies all of the requirements. The sensors enable very low levels of energy consumption during the monitoring of a variety of environmental or physical parameters, including temperature, humidity, vibration, pressure, sound, motion, and so on.

Additionally, the sensors are able to communicate and relay data on the detecting environment to the base station. The vast majority of contemporary WSNs are bi-directional, which enables two-way communication. This makes it possible to receive sensing data from sensors and send it to the base station, as well as distribute orders from the base station to end sensors. Military uses, such as battlefield monitoring, were a driving force behind the creation of wireless sensor networks (WSNs). WSNs are now extensively employed in a variety of settings, including industrial settings, residential settings, and wildlife settings. Applications like as monitoring the health of structures, providing healthcare, automating the house, and tracking animals are examples of representative WSNs applications.

"Sensor nodes" are the building blocks of a typical Wireless Sensor Network (WSN), which might consist of several hundreds or even thousands of individual nodes. WSNs can have a topology that is any of three different types: a star network, a tree network, or a mesh network. Considering that every node is capable of wirelessly communicating with every other node, a typical sensor node is made up of several different components. These components include a radio transceiver with an antenna that is able to send or receive packets, a microcontroller that is able to process the data and schedule tasks that are relevant to it, a variety of sensors that

are able to collect data regarding the environment, and batteries that are able to provide an energy supply.

2.1.4 Data transferring

A wireless network is one that does not use any physical medium to function (TechWalla, 2019). Radio waves, microwaves, line-of-sight infrared, satellite communication, and other forms of wireless communication can be used to connect the various components of a wireless network, such as personal computers, laptops, servers, and printers. Radio waves are used by the majority of wireless network providers as demonstrated in Figure 7. Each node in a wireless network is equipped with an adapter or network card that is tailored to capture and transmit radio waves that have been finely tuned to a certain frequency. The adapters function in a manner quite similar to that of radio antennas.

The network will be equipped with a piece of hardware known as a wireless router. This piece of hardware will physically connect to the incoming network and, in turn, the Internet by means of high-speed cable or broadband Internet. The data that is physically sent is converted by the wireless router into radio waves, which it then sends out through its antennae. This procedure is also performed in reverse by the router, which takes information from wireless sources (such a computer) and converts it from radio waves into a language that can be used by an Internet connection that is physically attached.

The transmission of data involves converting information from its binary form of zeroes and ones into a medium that is comprised of radio waves. The newly converted data is subsequently broadcast, at which point wireless adapters listen in and convert the data received from the radio into a format that the computer can understand, using a combination of zeros and ones. Radio frequencies of 2.4 GHz or 5 GHz are utilized by wireless network technologies. The transmission of additional data is made possible by using a higher frequency. The 802.11 standard governs the operation of wireless networks, just as every other type of computer network has a specific code for the norms by which it abides. The example of data transmitting structure is shown in Figure 2.4.

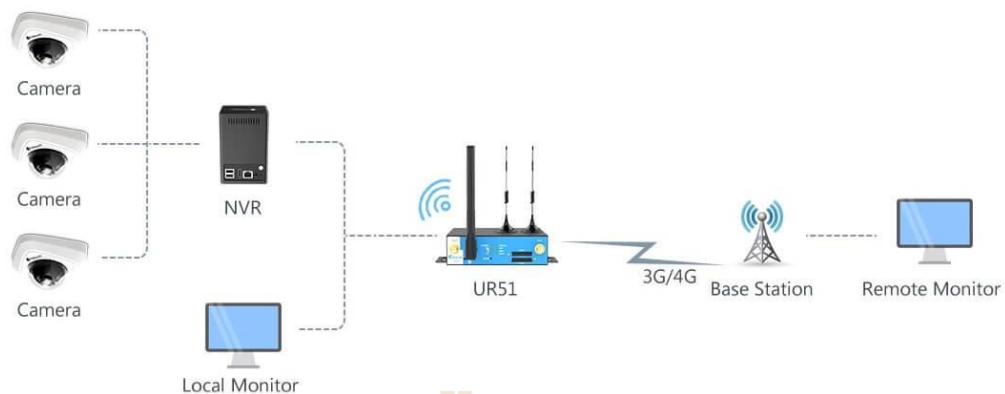


Figure 2.4 Example data transmitting structure

2.1.5 Cloud Server

"Cloud storage" is a method of storing data for computers that keeps the digital information in logical pools as opposed to physical ones (WikiContributor, 2021). The actual storage is distributed among a large number of servers, some of which may be found in more than one location. In most cases, the actual environment of the storage, which might span numerous servers, is owned and managed by the company that provides hosting services. It is the responsibility of these cloud storage providers to ensure that the data are always available and can be accessed, as well as to keep the physical environment safe, protected, and operational. In addition to this responsibility, the cloud storage providers must also ensure that the data can be accessed. Individuals and companies may buy or lease storage capacity from the providers in order to accomplish the task of storing user, organizational, or application data.

Access to cloud storage services can be gained through a colocated cloud computing service, an application programming interface (API) for a web service, or by applications that use the API. Some examples of these types of applications include cloud desktop storage, a cloud storage gateway, and Web-based content management systems. Using an application programming interface (API) provided by a web service is yet another method for accessing cloud storage services (API). Figure 2.5 is a visual representation of the extensive capacity of cloud storage.

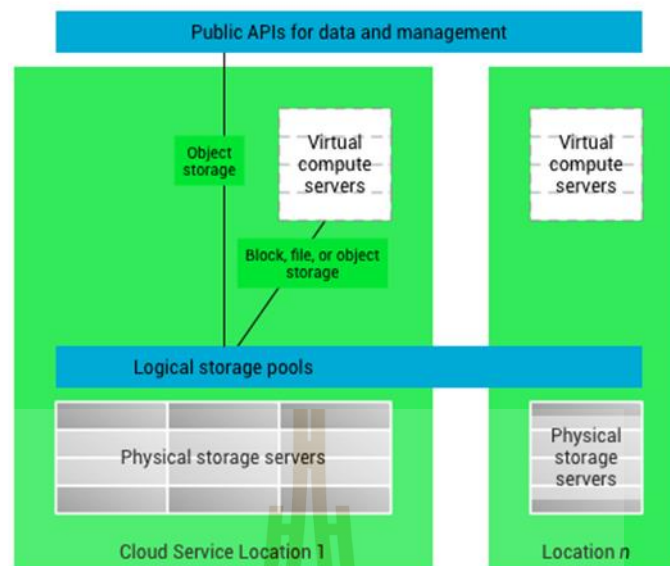


Figure 2.5 High level of cloud storage architecture

2.2 Theoretical Background

In this research, several necessary theoretical backgrounds for reconstruction process consist of camera calibration, morphological processing, 3D model polygon mesh, and 3D scene reconstruction are described.

2.2.1 Camera Calibration

To precisely extract depth of image and depth-image, the camera parameter must be calculated. So that, camera-calibration is required. The estimation of the parameters of an image or video camera's lens and image sensor is accomplished by a process known as geometric camera calibration, which is also known as camera resection (Zhang, 2020). Figure 2.5 illustrated these characteristics to adjust for lens distortion, quantify the size of an item in terms of world units, or establish where the camera is situated within the image. Applications such as machine vision make use of these activities to identify and quantify the objects in their field of view. They are also utilized in robotics, navigational systems, and the reconstruction of three-dimensional scenes.

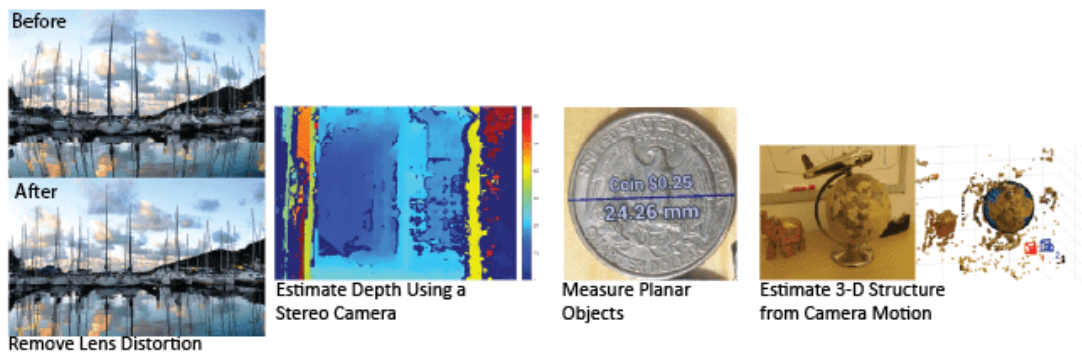


Figure 2.6 Overview of camera calibration (Zhang, 2020).

The intrinsic, the extrinsic, and the distortion coefficients are all types of camera parameters that need to have both the 3D world points and the 2D image points that correspond to those world points in order to estimate the camera parameters. These correspondences can be obtained by utilizing numerous pictures of a calibration pattern, such as a checkerboard, to do the calibration. It will be able to solve the camera settings if correspondences are used., in order to assess the precision of the calculated parameters, the result are follow:

- Plot the relative locations of the camera and the calibration pattern
- Calculate the reprojection errors.
- Calculate the parameter estimation errors.

Utilize the Camera Calibrator to carry out camera calibration and assess the degree to which the estimated parameters are accurate.

2.2.1.1 Camera Model

Both the pinhole camera model and the fisheye camera type have calibration algorithms included in the Computer Vision Toolbox™ (Matlab, 2021). The fisheye variant is compatible with cameras that have a field of vision (FOV) that is no greater than 195 degrees.

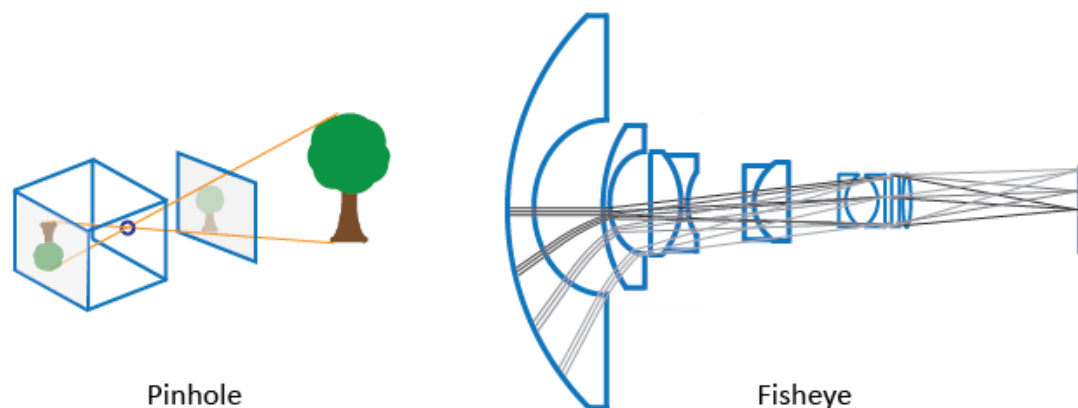


Figure 2.7 Camera Model (Matlab, 2021)

The model that was provided by Jean-Yves Bouguet serves as the foundation for the pinhole calibration algorithm (J. Bouguet, 2010). The pinhole camera type (Figure 2.5) and lens distortion are both accounted for in this approach. Because an ideal pinhole camera does not contain a lens, the lens distortion effect is not accounted for in the model of the pinhole camera. The radial and tangential lens distortions are accounted for in the whole camera model that the algorithm employs because this is necessary for an accurate representation of a genuine camera. The pinhole model is not suitable for simulating the behavior of a fisheye camera because of the severe distortion caused by fisheye lenses.

2.2.1.2 Pinhole Camera

A pinhole camera is a very basic type of camera that does not have a lens and just has a single, very narrow aperture. When light beams enter the camera and travel through the aperture, they produce a reversed image on the other side of the camera. The virtual picture plane is in front of the camera and it contains the image of the scene in its normal. Figure 2.8 shows a mechanic of pinhole camera. It consists of image plane, focal point, virtual image plane, and 3D real worked object.

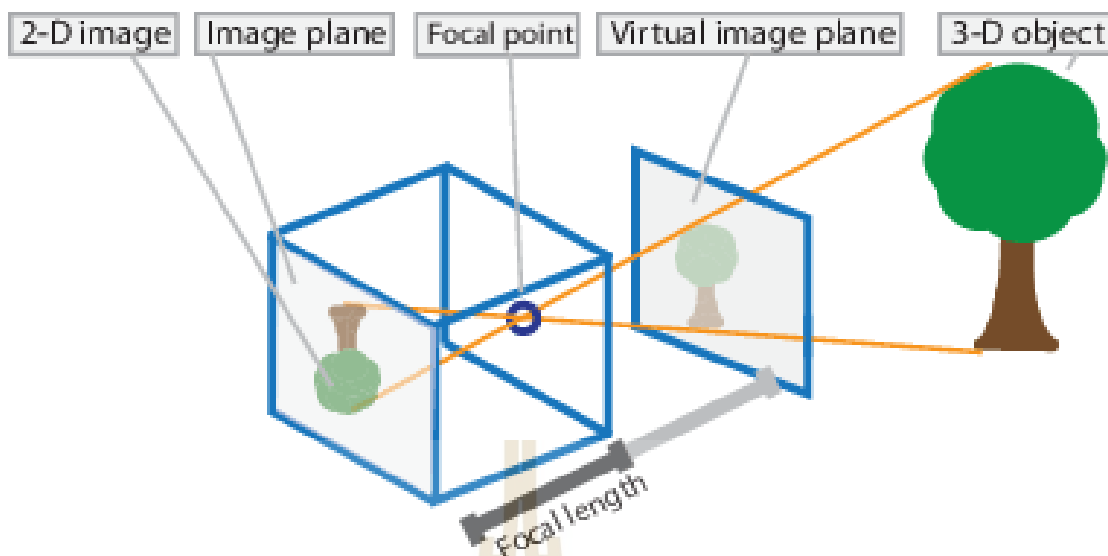


Figure 2.8 Pinhole Camera Model (Matlab, 2021)

The camera matrix is a four-by-three matrix that contains the representations of the pinhole camera's parameters. This matrix converts the three-dimensional scene into a picture on the plane. The camera matrix is computed by the calibration algorithm utilizing the extrinsic as well as the intrinsic characteristics. The position of the camera inside the three-dimensional scene is reflected by the extrinsic parameters. The optical center and focal length of the camera are both represented by the camera's intrinsic characteristics.

$$[x \ y \ 1] = [X \ Y \ Z \ 1]P \quad (2.1)$$

Where W is a scale factor. x and y are image point. X, Y , and Z are world point.

$$p = \begin{bmatrix} R \\ t \end{bmatrix} k \quad (2.2)$$

Let R and t are rotational matrix and translation matrix respectively. Through the utilization of the extrinsics parameters, Figure 2.9 shows the camera coordinates

of the world's points are altered. Through the utilization of the intrinsic parameters, the camera coordinates are mapped into the image plane.

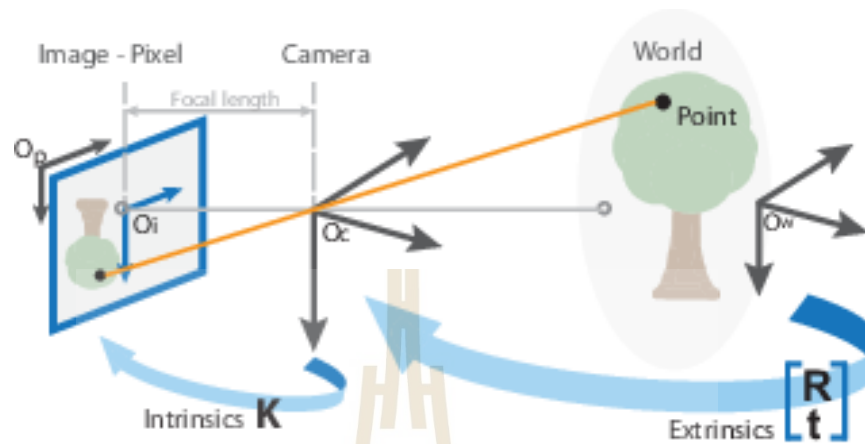


Figure 2.9 Image point and Real-World Point (Matlab™, 2021)

2.2.1.3 Camera Calibration Parameter

The camera matrix is computed by the calibration algorithm utilizing the extrinsic as well as the intrinsic parameters. Figure 2.10 demonstrated the extrinsic parameters constitute a strict transition from the coordinate system of the three-dimensional world to the coordinate system of the three-dimensional camera. A projective transformation from the coordinates of the three-dimensional camera to the coordinates of the two-dimensional image is represented by the intrinsic parameters.

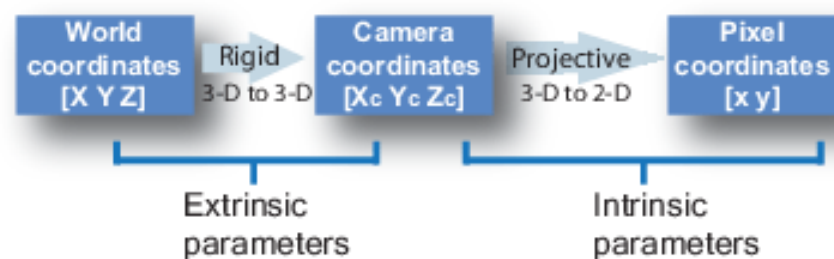


Figure 2.10 Overview of World point and Image Point (Matlab, 2021)

2.2.1.4 Extrinsic Parameter

In the Figure 2.11, the rotation, denoted by R , and the translation, denoted by t , make up the extrinsic parameters. The optical center of the camera serves as the starting point for the coordinate system, and the image plane is defined by the camera's x-axes and y-axes.

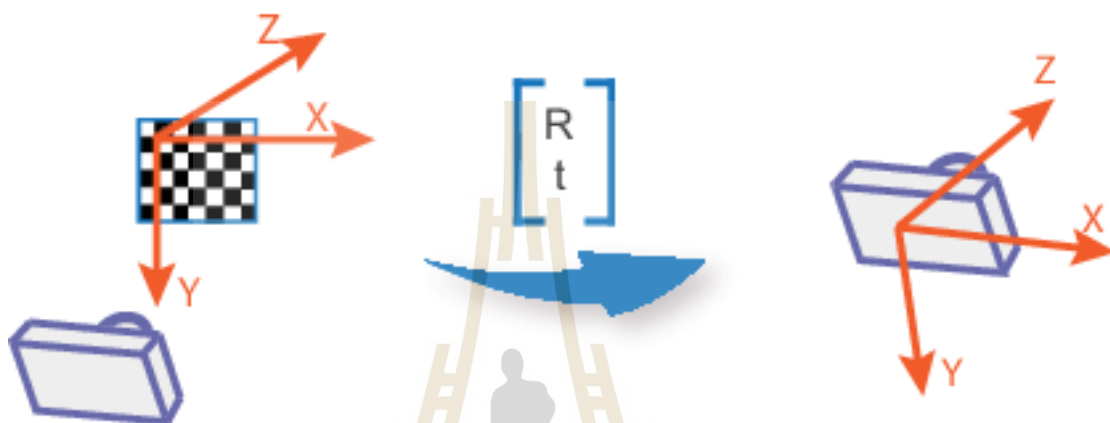


Figure 2.11 Explanation of Rotational and Translation Matrix (Matlab™, 2021)

2.2.1.5 Intrinsic Parameter

The focal length, the optical center, or principal point, and the skew coefficient are all examples of intrinsic characteristics. Other intrinsic parameters include the axial tilt.

Figure 2.12 shows skew Coefficients: A sensor's pixels may not be precisely square, which might lead to a slight distortion in either the X or Y direction. On a sensor, the skew coefficient refers to the number of pixels that are packed into one unit of length in each direction.

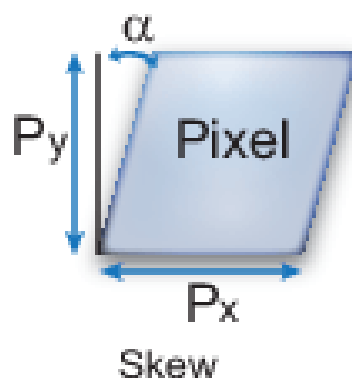


Figure 2.12 Explanation of Skew (Remondino, F., 2006)

2.2.1.6 Distortion Camera Parameter

Due to the fact that an ideal pinhole camera does not contain a lens, lens distortion is not taken into consideration by the camera matrix. The radial and tangential lens distortions are accounted for in the camera model so that it can faithfully simulate the behavior of a real camera.

2.2.1.7 Radial Distortion Parameter

When light rays bend closer to the borders of a lens than they do at the optical center of the lens, a phenomenon known as radial distortion occurs. The greater the degree of distortion, the more compact the lens.

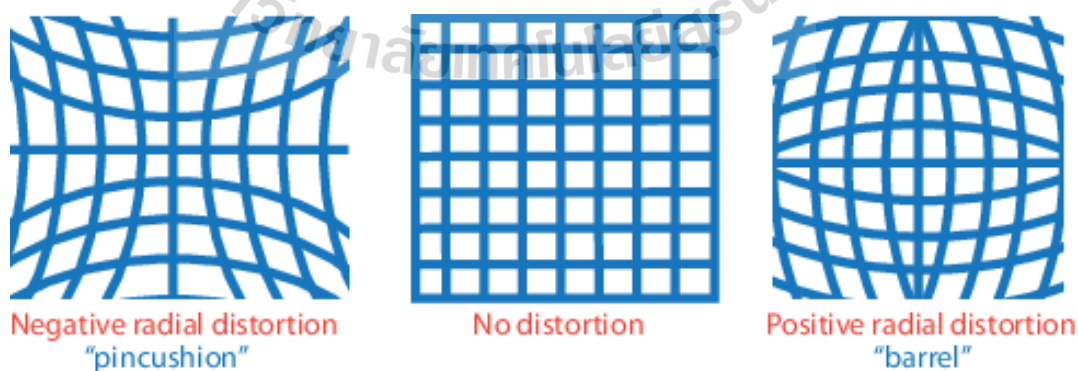


Figure 2.13 Explanation of Distortion (Matlab, 2020)

Radial distortion is a problem that can occur in a transmission that follows a straight line, as was previously explained. Radial distortion can be broken down into two primary categories (Figure 2.13). The first one is a barrel distortion, which is also known as a negative displacement. When points are pushed from their correct position toward the center of the image, a phenomenon known as barrel distortion takes place. The second kind of radial distortion is known as a positive displacement, and it takes place whenever points are moved further away from the optical axis. Another name for this kind of distortion is pincushion distortion. The wide-angle lenses are more likely to have barrel distortion, whereas the narrow-angle lenses are more likely to have pincushion distortion.

2.2.3 Point Cloud

A point cloud is the collective representation of data points that are spread out throughout a three-dimensional region. It's feasible that the points reflect a form or entity that's three dimensions deep. Cartesian coordinates are independently allotted to each point location in the space (X, Y, Z) . Both photogrammetry software and 3D scanners acquire data from a high number of points situated on the outside of the object being scanned. This information is then used to build point clouds, which are often used for visualization purposes. Example of extracted point cloud is shown in Figure 2.14 The creation of 3D CAD models for manufactured parts, the conduct of metrology and quality inspections, and the execution of a wide variety of tasks involving visualization, animation, rendering, and mass customization are just some of the many uses for point clouds, which are the output of 3D scanning processes.

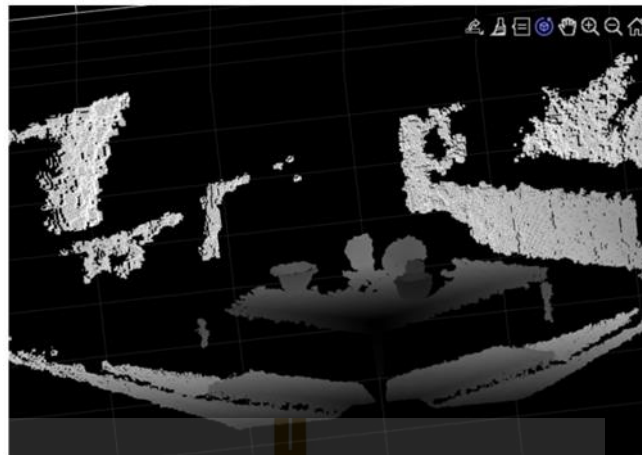


Figure 2.14 Example of unstructured Point cloud

2.2.4 Point Cloud Registration

The process of finding a spatial transformation (such as scaling, rotation, and translation) that aligns two-point clouds is referred to as point-set registration in the fields of computer vision, pattern recognition, and robotics. Specifically, point-set registration is used to describe the process. This procedure is also referred to as scan matching and point-cloud registration (Figure 2.15). Finding a transformation of this kind will allow you to accomplish multiple goals at the same time, including the merging of multiple data sets into a globally consistent model (or coordinate frame), as well as the mapping of a new measurement to a known data set in order to identify features or to estimate its pose. The goal of finding this transformation is to achieve multiple goals at once. Finding a transformation is the path to success for achieving both of these objectives. The raw 3D point cloud data that is often gathered comes from a variety of sources, the key ones being Lidars and RGB-D cameras. Producing 3D point clouds may also be accomplished with the use of computer vision methods such as triangulation, bundle correction, and more recently, monocular image depth estimation accomplished by deep learning. Other computer vision algorithms include Feature extraction from an image, which can yield two-dimensional pixel coordinates that can be used in 2D point set registration, which is used in image processing and

feature-based image registration. Corner detection is one example of a computer vision algorithm that can extract features from an image. Both methods of picture registration may be accomplished with the help of a point set. Point cloud registration has a wide variety of applications, some of which include autonomous driving, motion estimation and 3D reconstruction, object detection and pose estimation, robotic manipulation, simultaneous localization and mapping (SLAM), panorama stitching, virtual and augmented reality, medical imaging, and many more

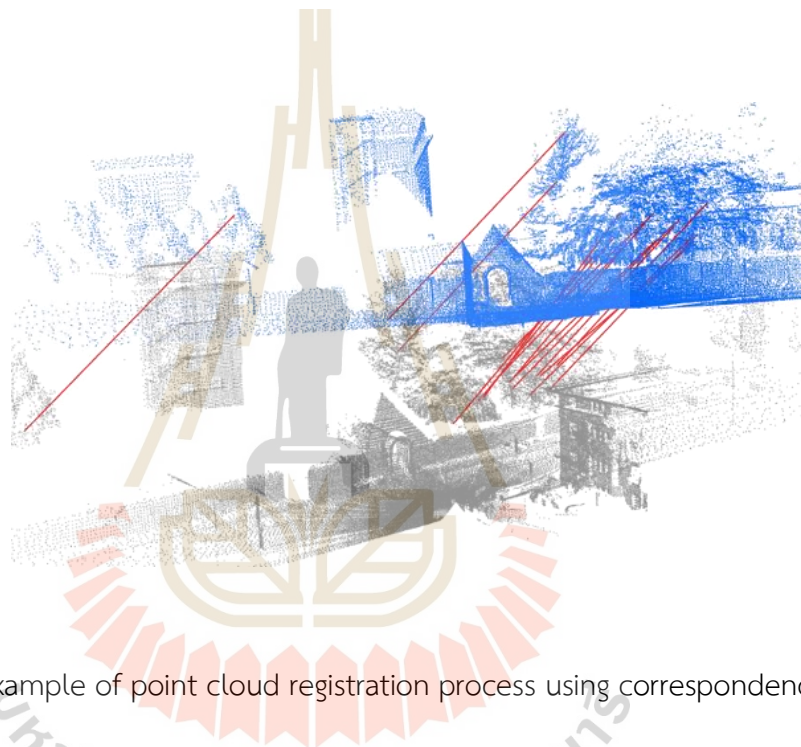


Figure 2.15 Example of point cloud registration process using correspondence point

2.2.5 Three-dimensional reconstruction

The process of capturing the form and look of real-world objects in three dimensions is referred to as "3D reconstruction" in the fields of computer vision and computer graphics. This process can be completed using either active or passive ways, depending on your preference. Non-rigid or spatio-temporal reconstruction is the term used to describe the situation in which the model is permitted to vary its shape throughout the course of time.

Reconstruction in three dimensions has traditionally been considered a challenging scientific aim. One may determine the three-dimensional profile of any object by using 3D reconstruction, as well as know the three-dimensional coordinate

of any point on the profile. Computer-aided geometric design (CAGD), computer graphics, computer animation, computer vision, medical imaging, computational science, virtual reality, digital media, and many other fields all rely heavily on the core technology of three-dimensional object reconstruction. This is because three-dimensional object reconstruction is both a general scientific problem and a core technology. For example, the information about the patients' lesions can be presented in 3D on the computer. This provides a novel and accurate approach to diagnosis, and as a result, it has essential clinical value. Reconstructing digital elevation models is possible through the utilization of techniques such as aerial laser altimetry and synthetic aperture radar. In the Figure 2.16, the demonstration of 3D reconstruction is performed. The 3D model can be constructed from multi plane or multi point of view.



Figure 2.16 Three-Dimensional Reconstruction

2.2.5.1 Polygon Meshes

When applied to the realm of three-dimensional computer graphics and solid modeling, the term "polygon mesh" refers to a collection of vertices, edges, and faces that, when taken as a whole, constitute the geometry of a polyhedral object (Wikipedia contributor, 2021). The rendering process is simplified when the faces are made up of triangles (triangle mesh), quadrilaterals (quads), or other basic convex polygons (n-gons), since these shapes may be broken down into the simplest possible

configurations. However, the faces may also be built in a less specific manner of concave polygons, or even polygons having holes cut out of them. The study of polygon meshes is one of the important subfields that can be found within both computer graphics, more especially 3D computer graphics, and geometric modeling. Both of these subjects have substantial subfields. Alternative representations of polygon meshes are required for a wide variety of applications and goals, and this is because polygon meshes are so often used. Meshes are capable of undergoing a large variety of operations, some of which include but are not limited to smoothing, simplification, and Boolean logic, amongst many more. Meshes may also be simplified. Because methods are available, ray tracing, collision detection, and rigid-body dynamics are all able to be performed using polygon meshes. When a model is turned into a wireframe representation by drawing the edges of the mesh rather than the faces of the model, this is known as "edge-drawing". The example of different resolution of point cloud is demonstrated in.

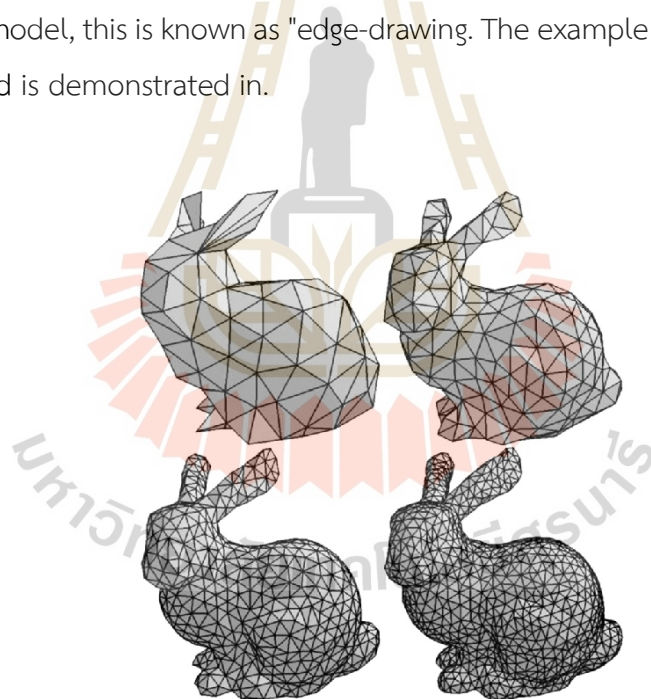


Figure 2.17 Example of Polygon Meshes (Wikipedia contributor, 2021)

2.2.5.2 Surface Reconstruction

Although point clouds may be directly displayed and inspected, the process known as surface reconstruction (Figure 2.17) is often used to turn point

clouds into a polygon mesh or triangle mesh models, NURBS surface models, or CAD models. Converting a point cloud into a 3D surface may be done in a variety of different ways. While some methods, such as Delaunay triangulation, alpha shapes, and ball pivoting, construct a network of triangles over the existing vertices of the point cloud, other methods convert the point cloud into a volumetric distance field and then reconstruct the implicit surface defined by doing so using an algorithm called marching cubes.

One of the inputs that go into the creation of a digital elevation model of the landscape in geographic information systems is point clouds. Another use for them is the generation of three-dimensional representations of urban settings. It is common practice to collect a series of RGB images using a drone. These images can then be processed using a computer vision algorithm platform to generate RGB point clouds. These point clouds can then be used to derive distances and volumetric estimates.

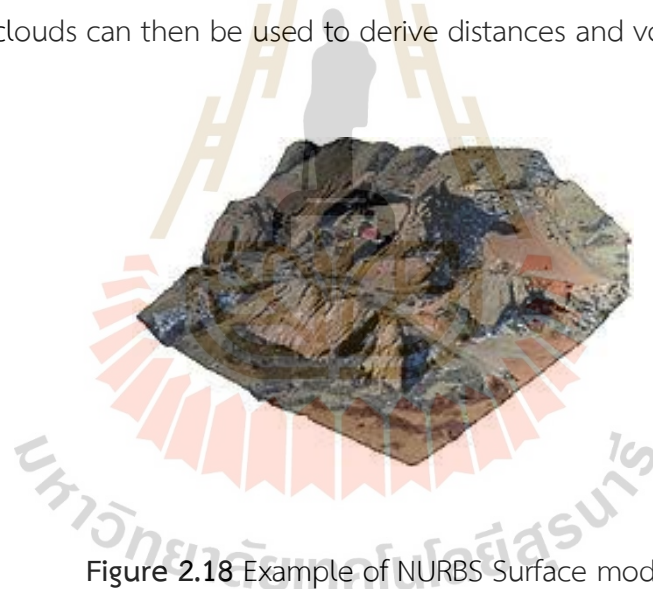


Figure 2.18 Example of NURBS Surface model

2.3 Literature Review

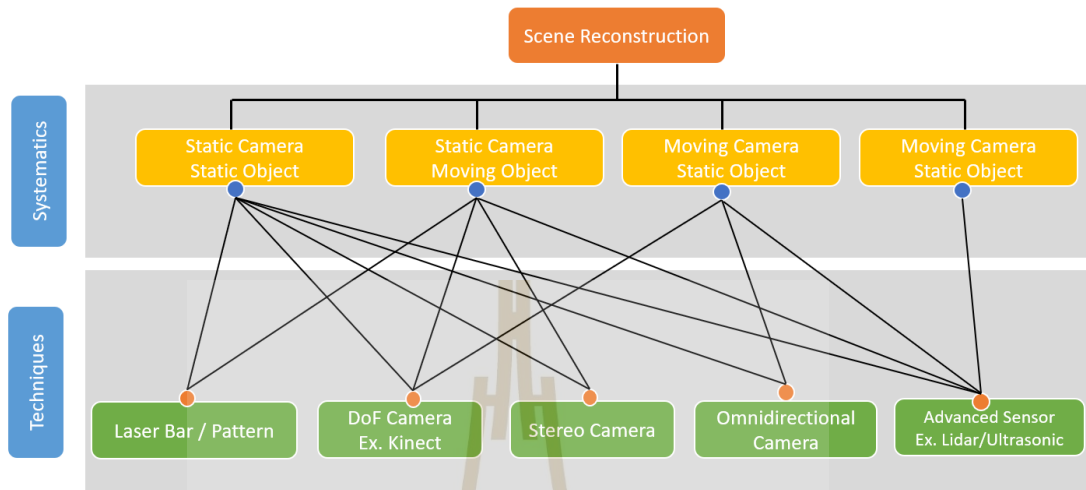


Figure 2.19 Overview of device and reconstruction techniques

The actual scene that is taking place in the real world is obscured at each and every time instance, and significant motion results in significant frame-to-frame changes, which may lead to an incomplete picture reconstruction. Therefore, the following categories of system designing approaches are used to classify the study activities that have been done in the field of three-dimensional scene reconstruction in the real-world environment: 1) Camera and Object in a Static Position (SCSO). 2) Camera Held Still, Subject in Motion (SCMO). 3) Camera in Motion, Object in the Same Place (MCSSO). 4) Camera in Motion, Subject Also in Motion (MCMO) as described in Figure 2.19.

Reconstructing dynamic scenes requires a variety of approaches, tools, and equipment, all of which are determined by the camera and the subject being reconstructed. The most typical uses of a gadget may be broken down into the following five modules: 1) Laser Pattern. 2) DoF Camera. Ex. Kinect Camera, RGB-Camera. 3) Stereo Camera. 4) Omni-directional Camera or 360-degree camera. 5) The Most Recent Sensor. Ex. Lidar-sensor, Ultrasonic. The following is an explanation of how devices and systematics design may be applied in practice: The SCSO design is

one that is utilized often. In order to accomplish a scene reconstruction, each of the aforementioned equipment was utilized due to the ease with which it could be done. On the other hand, if you capture still objects with a stationary camera, you run the risk of making an occlusion mistake from the point of view. As a result, the SCSO is dissatisfied with the performance in the real-world application. SCMO is a reconstruction approach that has gained popularity in recent years. It is possible to divide it into three distinct groups according on the nature of the thing that is being pieced back together. 1) the creation of a three-dimensional model from a static object. 2) the development of a three-dimensional model from a non-rigid object. In this course, the real-world scene is taken using either many cameras, a single RGB-D camera, or a monocular camera. Monocular cameras only record in one dimension. It is possible that the format of the collected data will differ depending on the scanning devices used; the data may either be gathered in the form of a point cloud or it may be represented as straightforward RGB information in pixel format. The second phase is called data pre-processing, and it involves applying filters to the data in order to reduce the noise that was introduced as a result of the collecting devices. 3) A three-dimensional reconstruction of the object's articulated motion in its various states If they start with a known priory, several state-of-the-art tracking and reconstruction algorithms that concentrate on articulated motion are able to reach high levels of performance. Priory-based reconstruction methods are dependent on additional inputs such as the geometric topology of the 3D model that needs to be reconstructed. Similarly, the animation of many three-dimensional facial reconstructions depends on blend shapes, probabilistic models, and other such things to estimate the dynamic changes. On the other hand, if the video camera does not have a high enough frame rate, SCMO will not be able to obtain any information about the item. It may result in a mistake or the loss of information. The MCSO reconstruction approach has a very high level of precision. The reconstructed model often improves in terms of its correctness as a result of the inclusion of many perspectives of the item. The MCSO comes equipped with a Depth-of-Field camera such as a Kinect camera, RGB-D, omnidirectional Camera, as well as sophisticated sensors such as Lidar-sensors, ultrasonic acoustic sensors, and so on. The problem of registering the required item

between frames continues to be a barrier. The MCMO strategy is the most difficult one to implement. because of the dynamic viewpoint that it provides between the camera and the item.

Nevertheless, the reconstruction results are tending to the most accurate to a real-world object. By using advanced sensor like Lidar-sensor, omnidirectional camera, DoF camera, a detail of reconstructed objects is nearly completed. Owing to expensive devices and uncertainty registration procedure, MCMO is challenging.

In this section, the literatures are separated into 3 parts, consist of 1) Image data communication, 2) Camera Calibration, and 3) Three-dimension scene reconstruction.

2.3.1 Image information transferring and Data Communication

In this review section, the research paper with image data transferring through Wireless Sensor Network (WSN) was proposed.

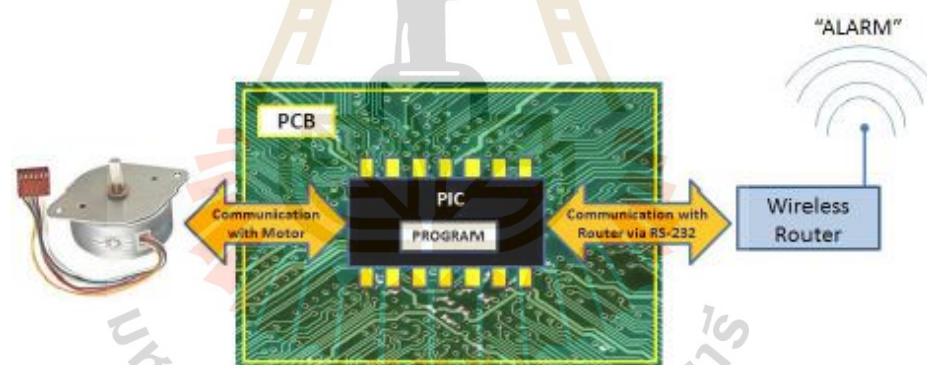


Figure 2.20 PCB structure of image transferring (J. Loret)

J. Loret et al. (2011) presented WSN that makes use of an image processing system in each wireless node to detect any unusual status of the leaves that could be caused by a deficiency, pest, disease, or other harmful agent in the vineyard. This could be done in order to protect the vines from potential damage. When the wireless sensor identifies any sign of illness in the leaves of the vine, it will trigger an alert that will be sent to the sink node in order to notify the farmer as shown in

Figure 20. They have investigated the WSN from the perspective of the sensing coverage area as well as the radio coverage area with the various spot (Figure 21). They have demonstrated with the sensor network traffic that the dispersed architecture enables reduced bandwidth usage as well as greater scalability. This is in comparison to when all video streams are broadcast across the network. The communication structure between router and sensor/motor is illustrated in Figure 19. The RS-232 was employed as a connection protocol between PIC program and router for data transferring.

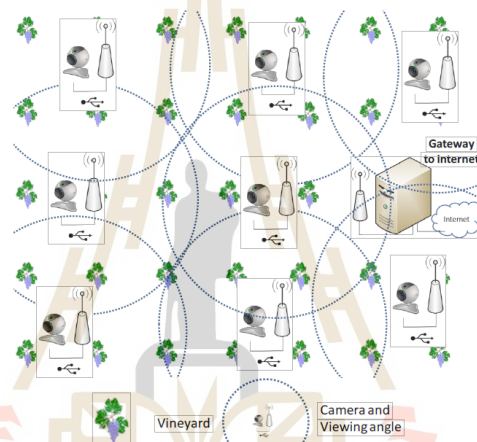


Figure 2.21 Wireless sensor network for proposed system

Taking into consideration the data collected in the sensor network traffic measurement section about the average value of the traffic (4.26 kbps), as well as the fact that IEEE 802.11g has a theoretical bandwidth rate of 54 Mbps, but an effective bandwidth rate of 27 Mbps, they are able to conclude that there will not be any limitations placed on the number of nodes that are able to operate within the WSN, at least not from a theoretical standpoint.

A. Javier et. al. (2015) proposed the implementation of wireless sensor networks for the purpose of integrating data monitoring and video surveillance in precision agriculture across scattered crops.

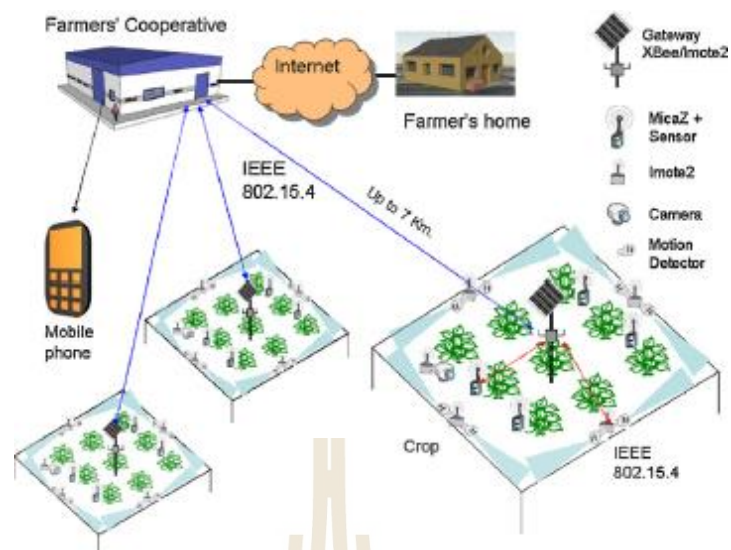


Figure 2.22 Overview of agricultural monitoring system

In this study, the Figure 2.21 shows the integrated system based on Wireless Sensor Figure 2.22 illustrated network for monitoring crops, conducting video surveillance, and controlling the cultivation process. The implementation of precision agriculture through the use of IEEE 802.15.4 as a technology that is efficient and cost-effective is implied by this network. As a result, the method has been created to carry out all of these duties not only in a single crop but also in deployments taking into consideration scattered crops that are located several kilometers apart from the premises of the farmer's cooperative. Additional elements are taken into consideration and given the necessary amount of engineering, such as the amount of energy consumed by the video devices or the end-to-end transmission delays. All of these needs are met by the comprehensive ISSPA system, which offers an effective and well-coordinated communication infrastructure between the many sensing nodes installed in the crops and the end-user. The ISSPA methodology makes it easier to maintain a well-organized crop monitoring system and to find trespassers in a timely manner. A comprehensive performance assessment study that illustrates the viability of the ISSPA system for use in precision agricultural applications has also been carried out as part of this body of work. In addition to that, a test-bed scenario has been established and is now being run. Several hardware prototypes for agricultural monitoring devices have

been developed and tested. Additional detection capabilities are supplied by infrared motion sensors, and intruder identification is achieved by utilizing video sensors in conjunction with the system. Each apparatus comes equipped with its own control, which may be customized with a unique set of application software modules. When it comes to the parameters that were chosen, the lifetime of the devices, and the transmission delay, the findings that were acquired experimentally in the real-world situation are comparable to those that were obtained through analysis and computer simulation.

According to the reviewed paper, the image information was transferred on wireless sensor networks with high quality. Due to the connection stable problems, their some transmitting issues which caused information loss. It also reduces the accuracy of the computational process.

2.3.2 Camera Calibration

To determine the depth information of the two-dimensional image, the camera is calibrated for obtaining camera parameters. Recently, there are many camera calibration methods which compatible with RGB-D camera. In this section, the camera calibration methods were described.

In their study, Yannick Hold-Geoffroy et. al. (2018) offered the first examination of human sensitivity to estimate errors for camera pitch, roll, and field of vision in the context of inserting virtual objects. In order to achieve this goal, they carried out a large-scale user study on Mechanical Turk. The purpose of the study was to determine the accuracy with which participants were able to differentiate between two images containing virtual objects that were composited with ground truth and distorted camera parameters. Their research shows that people are not always good at detecting significant inaccuracies, particularly when the roll is exaggerated or when the area of vision is overestimated. A CNN-based single picture calibration estimation approach that offers state-of-the-art performance was also described by these researchers. This method enables applications such as image retrieval, geometrically consistent 2D object transfer, and virtual 3D object insertion. In the course of the inquiry, it was found out that the learnt model is seeking for semantically significant vanishing lines, drawing similarities with geometrically based strategies for auto-

calibration. In the end, they utilize the findings of the user survey to establish a distance function that is based on human perception. This function is then used to compare the CNN to other techniques that have been taken in the past. Figure 2.23 shows the heatmap visualization of the experimental result.

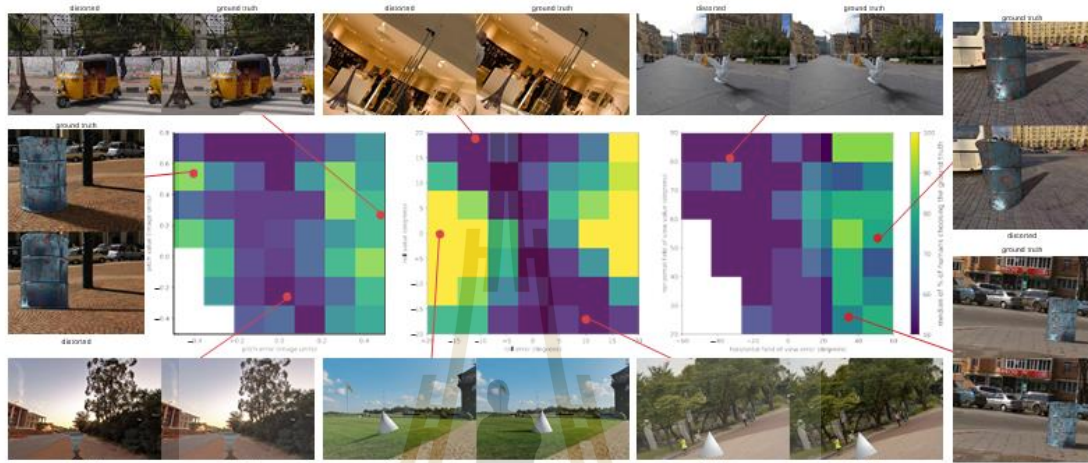


Figure 2.23 Experimental Result (Hold-Geoffroy, 2018)

Y. Liu et. al (2018) suggested developing an efficient 3D tracker for use in tracking objects in RGB-D films. They investigated a mean-shift tracker with a three-dimensional extension and uncovered its mechanism. They suggest two useful approaches, explicit occlusion management and 3D context-based model adaptation, both of which are based on the tracker and considerably increase the tracker's resilience. The efficiency of the strategy that was presented has been proved by a large number of experimental findings. Even if the procedure is able to operate successfully the vast majority of the time, there is a possibility that it may fail when dealing with long-term blockage. It should be noted that the discriminative capacity of color histograms is restricted. Figure 2.24

shows the adaptation in the face of distractions on the left, the pixels that are colored green represent locations that are contained within the target sphere. The weight of the points obtained without taking into account the effects of any distractor (AAD). The weights of distracting spots on the wall have been suppressed using AAD, which can be found in the middle of the bottom. When 3D PDF is used without AAD,

the mode of distractor is brought to the forefront. The mode of the distractor in the 3D PDF on the right becomes less clear when it is combined with AAD.

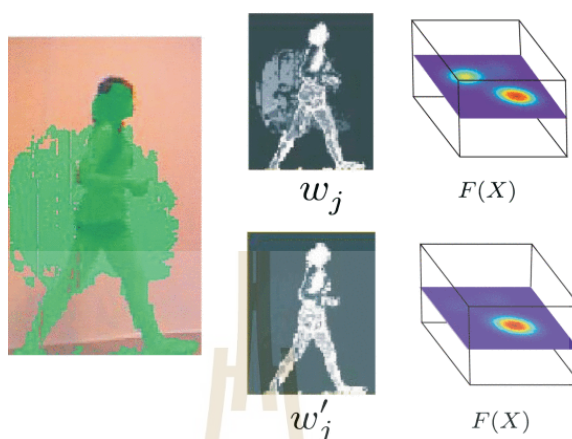


Figure 2.24 Experimental Result (Y. Liu, 2018)

Jen-Hui Chuang et. al. (2018) devised an innovative method for camera calibration that is based on a geometric point of view. The proposed method solves two major problems that are associated with Zhang's method, which is widely used. These problems include a lack of clear hints of appropriate pattern poses and a limitation on the applicability of the method that is imposed by the assumption of a fixed focal length. The proposed method resolves both of these problems. A closed-form solution to the calibration of extrinsic and intrinsic parameters based on the analytically tractable principal lines is the primary contribution of this study. The principal point is determined as the place where such lines meet. Each of the lines can then conveniently represent the relative three-dimensional orientation and position (each line only has one degree of freedom) between the image plane and the corresponding calibration plane for a particular WCS-IPCS pair. This can be done by comparing the relative orientation of the image plane to the orientation of the calibration plane. As a consequence of this, the calculations related with the calibration may be substantially simplified, and relevant recommendations can be easily developed to prevent outliers in the computation. Experimental results for both synthetic and real data clearly validate the correctness and robustness of the proposed

approach, with both comparing favorably with Zhang's method. This is especially true in terms of the possibilities to screen out problematic calibration patterns as well as the ability to cope with the situation of varied focal length. Moreover, the experimental results validate the correctness and robustness of the proposed approach. There are four examples in Figure 2.25 taken from eight good calibration photos, and the values range from 0 degrees to 180 degrees. Calibration pictures were used to obtain eight primary lines, each of which was used to calibrate the image.

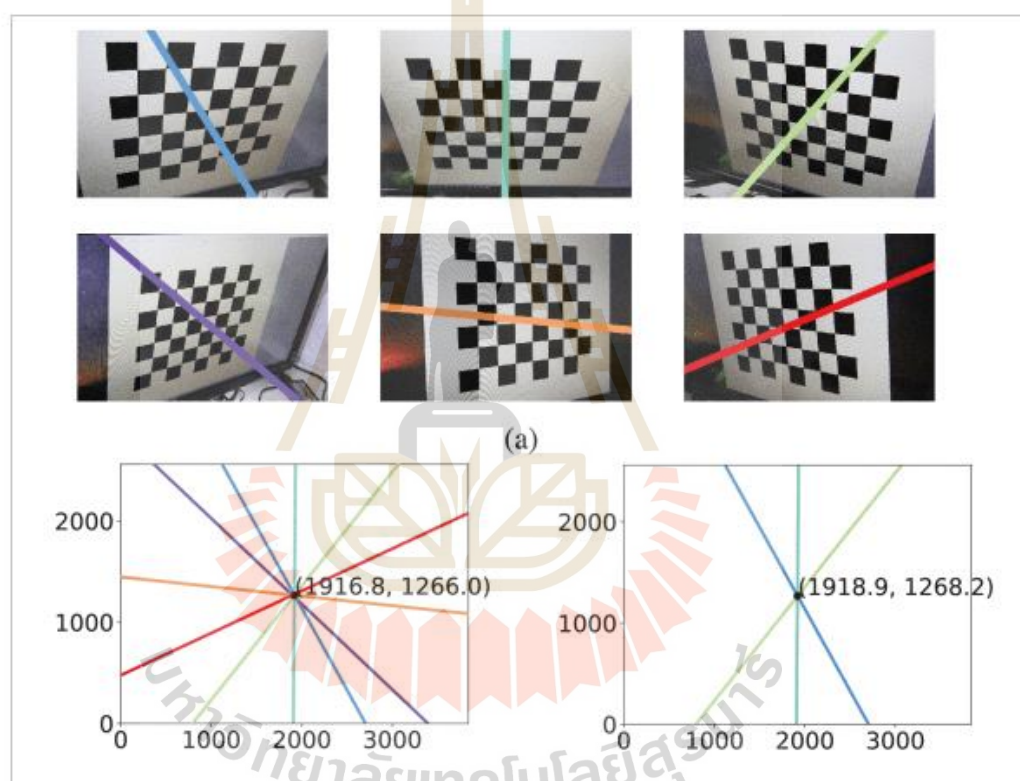


Figure 2.25 Calibration Result (J. Huang, 2018)

Shaoyan Gai et. al. (2015) suggested a unique 3D dual camera calibration. The reconstruction error that is employed in the suggested technique is better appropriate to dual-camera calibration. 1) The method comprises of using dual-camera calibration. When compared to the mistake in reprojection that the traditional approach produced, the precision achieved by the new method is a substantial improvement. 2) During the process of calibrating the system, they employ Zhang's approach to determine the

principal single camera parameters, the Centroid Distance Increment Matrix to determine the R, T matrix, and the space intersection method to determine the outcomes of the 3D reconstruction. The task is made more adaptable and secure by all of the processes described above. During the process of calibration, the calibration template will be moved into a few different places; nevertheless, there are no specific requirements about the position of the calibration template. First and foremost, the procedure is effective in terms of calibrating two cameras simultaneously. It has the potential to boost the efficiency of the 3D reconstruction.

The above-mentioned literatures, they proposed techniques to calibrate the camera parameter using the chess board pattern with different viewpoint. The corner of each pattern was detected and used as calibration input. The result shows that the camera parameters were calculated accurately. On the other hand, they still need the chessboard pattern with variety of viewpoint to get calibrated parameters.

2.3.3 Image registration and Three-Dimensional Reconstruction

S. Song et. al. (2013) presented an assessment of numerous baseline techniques using 2D detectors, 3D detectors, optical flow, and ICP, as well as provided a unified tracking benchmark for both RGB and RGB-D tracking. They provide a straightforward occlusion management approach that is based on the depth map, and they also assess numerous advanced RGB tracking algorithms. Both of these are done based on the depth map. The findings indicate that it is possible for trackers to improve their performance and deal with occlusion in a more reliable manner if they make use of depth data. They have high hopes that the single benchmark, which will make experimental assessment more uniform and more freely available, would give fresh insights to the field. Figure 2.26 shows algorithm for RGBD tracking based on a 2D picture patch. The combined confidence from the detector and the optical flow tracker is displayed on the two-dimensional confidence map. The target depth histogram is used to generate an estimated Gaussian for the one-dimensional depth distribution. A threshold derived from the 1D Gaussian is applied to the 2D confidence map before the computation of the 3D confidence map may begin. Figure 2.26 demonstrated the location of the target, shown in the output by the green bounding

boxes, is the point in which confidence is the highest. The depth value of an Occluder, shown by the blue enclosing box, allows it to be identified.

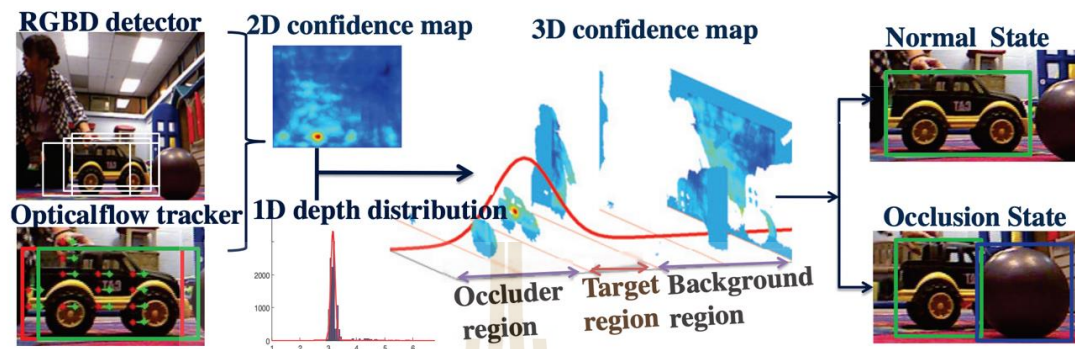


Figure 2.26 RGB-D tracking overall approach

Arnulfo León Reyesa et. al. (2016), this article demonstrates a way for developing a prototype of a 3D scanning device that is capable of representing virtual 3D objects using distance measurements received by a 1D optical distance sensor. The method is offered as part of this article. An electromechanical platform, data collecting hardware (which is controlled by the PIC18F4550 microcontroller), and a graphical user interface that is coded in Matlab make up the essential components of the device. The findings that were achieved by scanning a variety of solid items are quite encouraging and demonstrate the usefulness of the technique that was provided as well as the device's capacity for excellent operation.

Zhiyi Zhang et al. (2011) presented a 3D scanning technology that was introduced that was based on the geometric structure of monocular vision. According to the findings of the experiments, when using a 10-mW laser and the natural environmental light illumination was less than 500 lx, it was possible to obtain at least 4000 valid vertex data in one second across a variety of resolutions. This was the case regardless of the resolution chosen. Within a distance of three meters, the highest error and the average error are both around two millimeters, while the relative error stayed at approximately 0.08 percent throughout. Therefore, using this technology, it is possible to essentially re-construct the 3D model, and regular users will be happy with

both the scanning accuracy and speed. The findings of scanning the terracotta warriors and clay pots are shown in Figure 2.27 respectively. The scanned point cloud of the terracotta warriors can be seen in Figure 2.27 (a), and the depth map of the surface model can be seen in Figure 2.27 (b), which was produced from Figure 2.27 (a). The model contains a total of 39,460 vertices in addition to 64,918 polygons. Figure 2.27 (c) displays the scanned point cloud of clay pots, and Figure 2.27 (d) displays the depth map of the surface model derived from Figure 2.27 (a). Both figures may be found in the same Figure 2.27 (a). The model contains a total of 23,7823 vertices in addition to 28,9861 polygons. Pseudo coloration is used to convey information about depth in the picture; the range from far to near corresponds to the colors blue, green, and red.

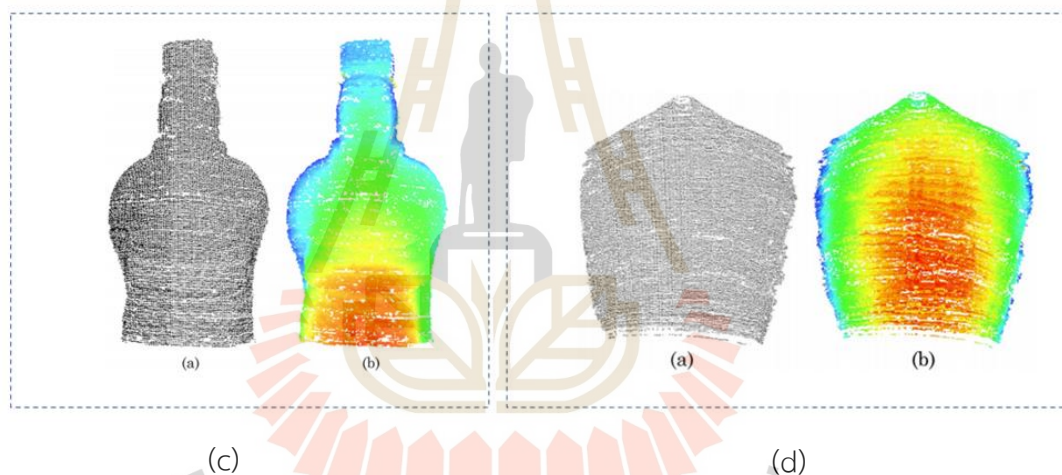


Figure 2.27 Point Cloud Experimental Result (Z. Zhang, 2011)

Chenyang Zhang et. al. (2020) proposed a novel geometric constraint model of RGB-D SLAM with points and lines was suggested by the researchers. The following is a rundown of the method's most significant contributions to the field: (1) In regard to the point, in addition to the 2D re-projection error of points, the per-depth was leveraged to its full potential and the constraint error of per-depth was added to the approach. This is due to the fact that an RGB-D camera is able to offer the per-depth

information. Their technique was distinct from the RGB-D SLAM systems since it was only based on the 2D re-projection error of point and line features. For the line, they constructed the suggested method by making the most of the 2D and 3D geometric information included inside the lines. (2) Using the 2D and 3D information of points and lines, they created their geometric constraint model and then extended it to the BA model. The outcome of the camera posture estimation that yields the least value should be considered the best possible one. The tracking loss of partial frames that happened in the OPF-SLAM was indicated in Figure 2.28 by a yellow rectangle. This information was gleaned from the experimental findings of our dynamic SLAM in real-world settings. Despite the fact that ORB-SLAM2 carried out trials in both of the image sequences, the camera pose estimate has a poor level of precision.

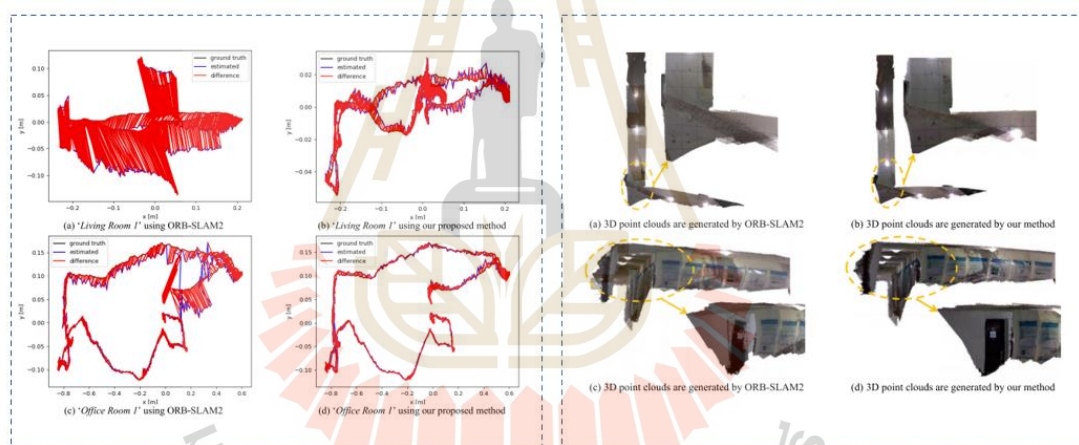


Figure 2.28 Reconstruction Result (Z. Chang, 2020)

Maxime Lhuillier et. al. (2018) proposed the first surface reconstruction approaches that simultaneously ensure consistent visibility while maintaining a low genus. They demonstrate surface improvements, including the elimination of holes, as well as a quantitative reduction in the genus. The topological noise that was caused by an operation of the original technique may be eliminated with a simple adjustment. A second technique, albeit more difficult than the first, changes the topology simplification with the help of a user parameter while the visibility consistency is being optimized. Other contributions include an acceleration of the manifold test by using

the orientability of the 3D Delaunay triangulation (of the input points), and a more efficient removal of surface singularities, which improves escapes from local extrema. Both of these improvements were made possible thanks to the work of the authors. An investigation of the non-manifoldness of the region close to a surface vertex or edge led to the conclusion that this feature should be eliminated. In the field of combinatorial topology, the local extrema may be explained in part by the constraints of an operation known as "shelling." To begin, the input point cloud has a low density. This is helpful in a few different circumstances, including the setup of dense stereo as well as big scale sceneries with limited processing resources. Second, the points are rebuilt using movies that were captured by many consumer cameras (or a spherical camera) put on a helmet while the subject was walking (or riding) through complicated landscapes. The images at a single site are depicted in Figure 2.29, together with their estimated and ground truth surfaces. Experiments conducted utilizing a fabricated city setting. Images taken at a certain place can be seen at the top. The middle section exhibits top views of the SfM result and the M3 surface. At the bottom are several local perspectives of the ground truth and the M3 surfaces.



Figure 2.29 Reconstruction Result (M. Lhuler, 2018)

Sheng et. al. (2018) presented a lightweight surface reconstruction approach for online 3D scanning point cloud data geared toward 3D printing has been developed. this method was proposed. An online lightweight surface reconstruction algorithm has been proposed. This algorithm is made up of three sub-algorithms: a point cloud update algorithm (PCU), a rapid iterative closest point algorithm (RICP), and an improved Poisson surface reconstruction algorithm. The goals of this algorithm are to generate a lightweight 3D model and achieve a low level of algorithmic complexity (IPSR). The point cloud data is denoised using the PCU, which results in a version that uses less memory and processing power. A quick and precise registration between two different sets of point cloud data may be accomplished with the help of the RICP. Postprocessing of the PDE patch creation based on biharmonic-like fourth order PDEs is done to repair the mesh holes on the rebuilt lightweight mesh, which results in an improvement to the 3D model's visualization. The IPSR is utilized to produce the lightweight mesh. In addition, a dynamic visualization framework for point cloud data that is based on WebSocket is provided in order to enable real-time point cloud data transfer in combination with the 3D scanning process in an online environment. This is accomplished by using WebSocket. Using this method, a thorough and dynamic display of the point cloud data may be accomplished in the browser. In a setting with a high level of concurrency, the Web Worker technique makes it possible to maintain fluidity and a high level of rendering quality. The next step involves developing, on the basis of the suggested technology, an online customized customisation system that is geared for 3D printing. The number of iterations for the "Huba" model reduced from 15 to 8, and the time decreased by 46.8 percent from 77 s to 41 s as a result of using the RICP. In comparison, the time for the "Totoro" model decreased by 45.8 percent, going from 59 s to 32 s. Table 4.2 contains a listing of these findings. Figure 2.30 demonstrates that after one iteration, the MSEs of the two models decrease by 0.039 mm and 0.036 mm, respectively. An increase in algorithmic efficiency is demonstrated by the experimental study as a result of preregistering the point cloud data in the RICP. Figure 2.30 depicts the implications of the registrations made by the RICP and the ICP on the visualisation of the data.

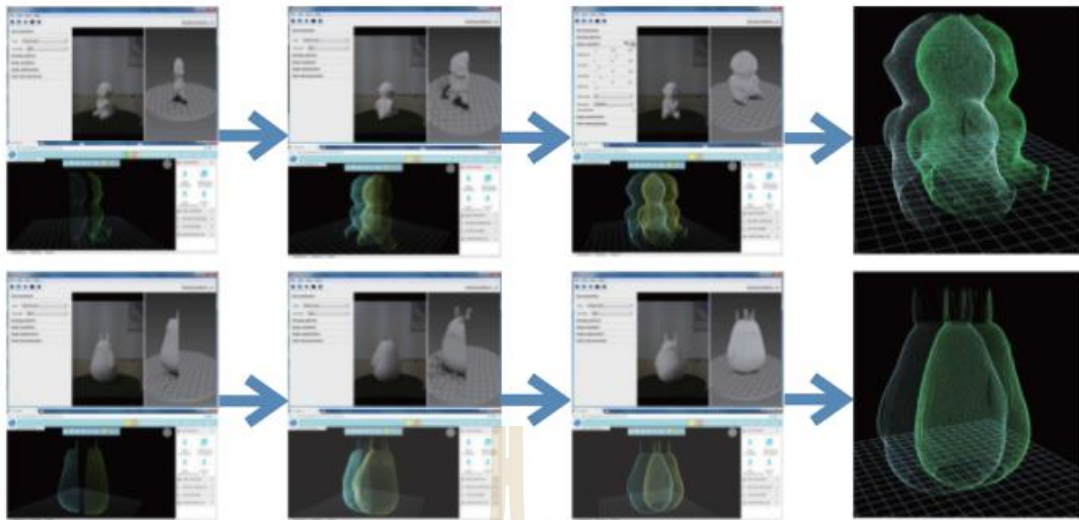


Figure 2.30 Experimental Result (Chang, 2018)

Carmelo Mineo et. al. (2018) presented a new approaches were introduced in order to improve the boundary of meshed surfaces after they were derived using point cloud data received from 3D scanning. The BPD method divides the unlabeled points of a surface point cloud into two categories—boundary points and interior points—so that the data may be processed more efficiently. The currently available detection methods have been honed to perfection in order to recognize points that correspond to sharp edges and wrinkles. The BPD algorithm is geared at the detection of boundary points, and it is able to do this task more effectively than the other approaches that are now available. The RBS algorithm locates the corners of each closed boundary and then separates each closed boundary into the edges that constitute it. Through the use of spatial FFT-based filtering, every edge is smoothed off. The fact that the proposed methods are not based on any threshold values, which means they might be appropriate for certain point clouds but not appropriate for others, is a significant benefit of these algorithms. The difficulty of choosing a certain polynomial function order for optimal polynomial curve fitting is solved by using FFT-based edge reconstruction instead of the traditional approach. The algorithms were put through their paces in order to examine the findings and determine the amount of time required for the execution of point clouds that were produced from laser scanned

measurements on a turbofan engine turbine blade that had a variety of member points. It has been demonstrated that the BPD algorithm is very robust for out-of-plane noise that is lower than 25 percent of the cloud resolution, and that it can produce satisfactory results when the noise is lower than approximately 75 percent of the total. This was accomplished by adding artificial noise to the model. The identification technique will miss some boundary spots if the noise values are between 25 and 75 percent of the cloud resolution; nevertheless, it will not create any outliers. In addition to this, quantitative findings about the functionality of the RBS algorithm were discussed. When compared to polynomial edges, the reconstruction edges that were calculated using the novel method provided a fit that was 4.7 times more accurate with the boundary points. In addition to this, they adhere to the rebuilt surface mesh contour, which results in a 77 percent improvement as compared to the polynomial fitting edges. The cloud points were decimated in order to create four distinct point cloud versions, each of which had a target resolution that was correspondingly equivalent to 4, 16, 8, or 34 millimeters. An extra point cloud with configurable point resolution ranging from 2 to 34 millimeters was produced as well. Testing the algorithms in such controlled environments and analyzing the results was made easier by the point clouds that were created using these techniques. The points that were identified within three spheres that were centered at preset places and had radii equal to 16, 32, and 64 mm were eliminated from the clouds in order to bring about the introduction of clearly defined internal boundaries. As a result, every cloud has three holes, denoted by the letters H1, H2, and H3, whose radii are approximately equivalent to those of the initial producing spheres. The five-point clouds that were produced as a consequence are depicted in the top row of plots in Figure 2.31. These plots illustrate the identified boundary spots with darker point markers in the shape of a circle. Both the exterior and the interior border locations have been pinpointed successfully. The resolution of the point cloud is one of the factors that determines the smallest hole's observable radius. This topic is covered in Section 2. Due to the fact that the resolution of these clouds is quite near to that of the holes' radii, only four spots of the 16 mm and 32 mm radius holes (H1 and H2) are found in the clouds shown in Figure 2.31 c

and d. On the cloud with a resolution of 34 millimeters, H1 cannot be found (Figure 2.31 d).

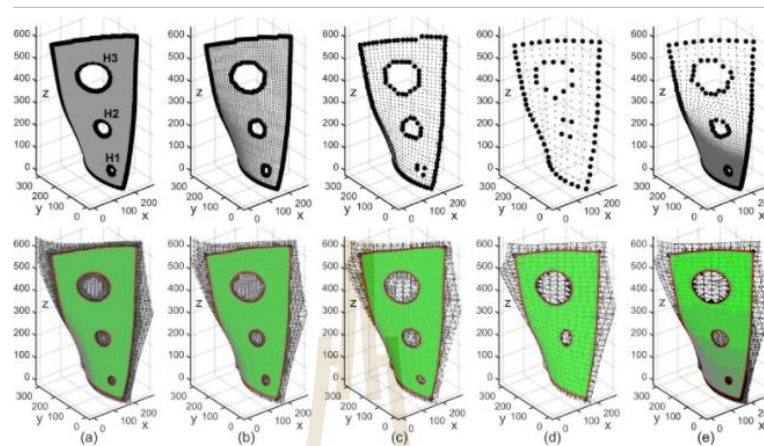


Figure 2.31 Experimental Result (C. Mineo, 2018)

K. Wang et. al. (2014) proposed an innovative and robust three-dimensional reconstruction method using an RGB-D camera was suggested. They combine the SFM approach with the use of visual and geometric cues to provide a more robust registration, particularly in circumstances when depth information is unavailable. They came up with a Prior-based Multi-Candidates RANSAC (PMCSAC) technique in order to deal with the repetitive textures. The goal of this approach was to make feature matching more reliable and effective. In addition to this, they make use of 3D information to assist in the detection of the loop closure and undertake global refinement in order to get rid of the drift issue. Combining multi-view stereo with mesh deformation methods is an excellent way to fill the missing geometry caused by depth missing. The findings of the experiments show that the approach is capable of producing superior 3D reconstruction outcomes compared to the current state of the art. Their system is not yet capable of functioning well in real-time environments. Due to the fact that the suggested system was developed using the code that had not been optimized, there is a lot of potential for the system to be sped up.

Figure 2.32 illustrated 3D models reconstructed using their approach, using the bag dataset, the bear dataset, and the human head dataset, respectively. The data on the collected depth that corresponds to the image. An assortment of perspectives on the rebuilt models.

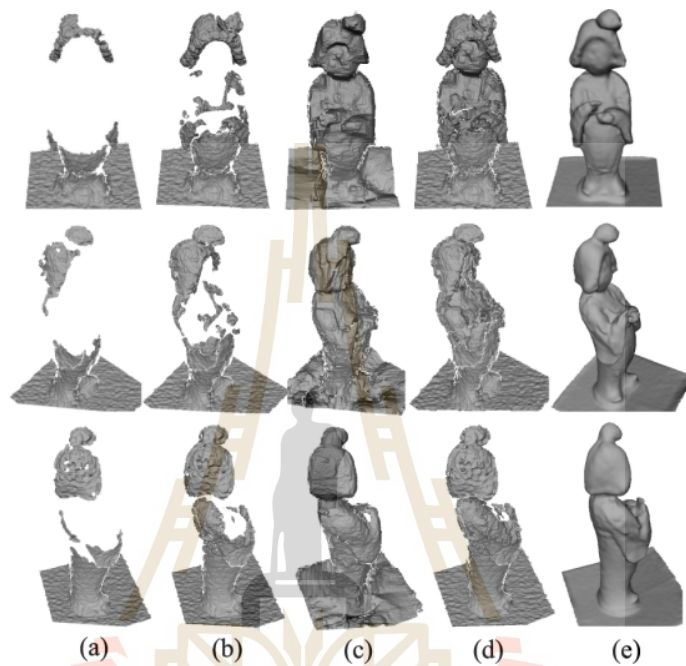


Figure 2.32 Experimental Result (K. Wang, 2014)

Y. Cui et. al. (2010) proposed an alignment of depth scans that were obtained from around an item using a time-of-flight camera was offered as a method for doing 3D object scanning. These ToF cameras have the capability of measuring depth scans at a video rate. They have the potential for low-cost manufacturing in large numbers because to the very basic technology that they use. It is possible that a scanning solution that is based on such a sensor and is both cost-effective and efficient may make 3D scanning technology more accessible to regular consumers. The degree of random noise produced by the sensor is high, and there is a non-trivial systematic bias. These two factors provide a problem for the algorithm. It revealed that 3D form representations of stationary objects can also be collected using a Time-of-Flight sensor, which, at first appearance, seems to be entirely unsuited for the job at hand.

The key to successfully accomplishing this goal is the effective integration of 3D super resolution with a novel probabilistic multi-scan alignment technique that has been adapted specifically for ToF cameras. Figure 2.33

illustrated Antique skull (a); despite the fact that the raw ToF data had a lot of mistakes, our system was able to generate a 3D model of fair quality (c) (b). When compared to the results of a laser scan (d), the reconstruction error (e) reveals that under no circumstances was the error greater than 2.5 cm, and throughout the majority of the surface, it was less than 1.0 cm. (Note: these are raw aligned scans; hole filling has not been done)

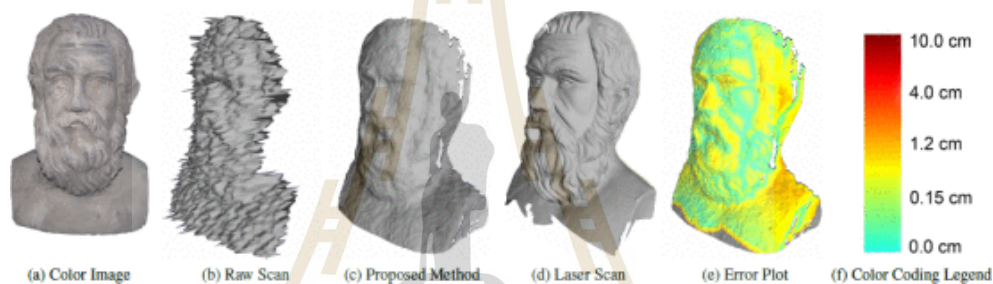


Figure 2.33 Experimental Result (Y. Cui, 2014)

R. A. Newcombe et. al. (2011) presented an utilizing low-cost depth camera and commodity graphics technology, the researcher was able to develop a method for accurate real-time mapping of complicated and arbitrary interior environments under varying illumination circumstances. They do this in real time by combining all of the depth data that is being broadcast from a Kinect sensor into a single global implicit surface model of the scene that is being viewed. Tracking the live depth frame in relation to the global model using a coarse-to-fine iterative closest point (ICP) approach, which makes use of all of the observed depth data that is now available, concurrently allows the current sensor posture to be derived. They show that tracking against a developing whole surface model has many benefits over frame-to-frame tracking, including the ability to acquire tracking and mapping findings in continuous time inside room-sized scenarios with little drift and high precision. In addition to this,

they provide both qualitative and quantitative findings in relation to different facets of the tracking and mapping system. An interesting new development in the field of augmented reality (AR), in particular, is expected to come from the modeling of natural environments in real-time using just commodity sensor and GPU technology (Figure 2.34). It enables the reconstruction of dense surfaces in real-time, with a degree of detail and resilience that is superior to any approach that has been offered so far making use of passive computer vision.

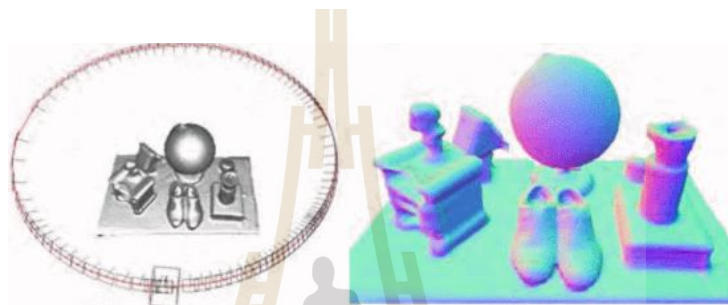


Figure 2.34 Experimental Result (R. A Newcombe, 2011)

The mentioned reconstruction techniques provide a good result with low errors. Some techniques use high performance sensors such as Lidar and SLAM, so that, it may cause lack of device access. Not only high-performance device was used but also point of view (POV) was varied. In the real world, it hard to get all the POV of the desired object. Thus, the mentioned reconstruction techniques are a disadvantage at this point.

2.3.4 Surface Reconstruction

Wei Ma et al. presented a novel approach to filtering that makes use of a spatial sorting algorithm in conjunction with an improved ball-pivoting algorithm. When compared to older filtering approaches, the parameters are simpler to comprehend and adjust. An improved BPA that retrieved bottom boundary points directly without the requirement for a 3D TIN model was able to successfully solve the issue of output loss in the bare ground zone. In addition, the performance of this method is often stable, which may be attributed to the straightforward approach for parameter selection. The findings of the trials indicate that this tactic has a high degree

of dependability and resilience in its execution. Because of this characteristic, it is possible to prevent making errors in interpolation or having uncertainty.

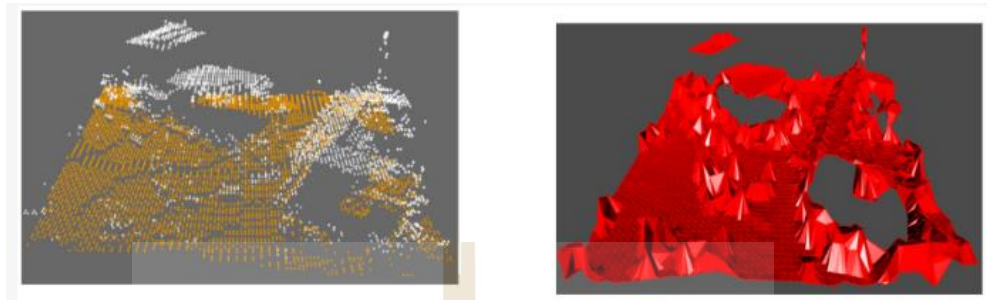


Figure 2.35 Three-dimensional tin surface

The raw point cloud of the 3D TIN structure, as well as the form that it ultimately became. It is possible for a primary raw point cloud's class to be either ground (brownness points) or non-ground. This distinction is made based on the cloud (white points). When a standard 3D alpha shape is used in the development of a 3D TIN, the resulting form is demonstrated in Figure 2.35

2.3.5 Wireless Visual Sensor Network Simulation

Joao Paolo et. al (2017) presented a Wireless visual sensor network for smart city applications: A relevance-based approach for multiple sinks mobility. In today's globe, big cities all around the world are faced with challenges that would have been unimaginable in years gone by. New problems are always cropping up as a result of the rapid rate of population growth; nevertheless, technology may be employed to alleviate these problems and improve the quality of life in large cities. Under those circumstances, surveillance is a service that is in great demand, and the majority of governments are currently employing a wide variety of tools to ensure adequate levels of safety. Wireless Visual Sensor Networks (WVSN) may be used to monitor every section of a city without the expense of stringing wires all over it. This is possible because to recent technological advancements. However, there has to be an effective method for compiling all of the data gleaned from the sensors and cameras, preferably one that uses less power and has a more consistent lag time. In

this study, a new method for positioning numerous mobile sinks in WWSN networks that are placed along highways and streets is proposed. Because source nodes with greater sensing relevance are anticipated to send more data packets, a relevance-based strategy was developed with the intention of positioning sinks in closer proximity to these nodes. Because the suggested algorithm is able to identify prohibited and unconnected regions, it can ensure that sinks will be positioned in locations where they are authorized. This ensures that the technique is particularly suited for the implementation of actual smart city applications.

In addition, the suggested method was tested with a variety of different combinations of sensor nodes and sinks. The purpose of this project was to give average performance data that might further witness to the efficacy of the solution that was devised.

During the initial random verification, a standard distribution of sensor nodes was taken into consideration. In the "Network 3" scenario (Figure 2.36), which describes a city with 20 horizontal and 20 vertical streets, a total of 64 sensors are distributed across the city, with 4 sensors located in each of the city's 16 blocks. It is possible that some of the 64 sensors may be source nodes for that scenario; however, the determination of which sensors will be sources and the sensing relevancies of those source nodes will be determined in a random fashion.

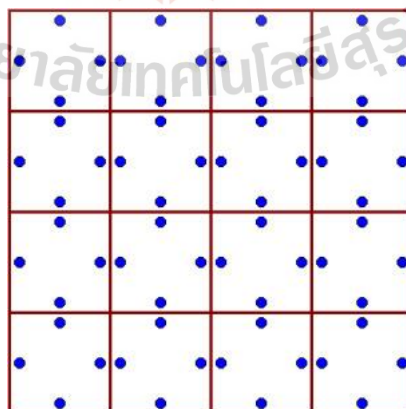


Figure 2.36 Random network configuration (Joao P. et. al, 2017)

Ahmed M. et al (2009) present a new field, wireless multimedia sensor networks in research entitled “Multimedia sensor networks: an approach based on 3D real-time reconstruction”. They present a variety of issues, and the consumption of resources is compounded by the fact that multimedia data are being transmitted. They described the design of their system as well as the preliminary performance of their system, which was specifically adapted for the use of video surveillance in this study. Optimizing system resources, in particular the bandwidth of the network, and providing the ability for full video data fusion and exploitation are the primary objectives that have motivated the development of our approach. Real-time three-dimensional reconstruction of the scene being witnessed was their original concept. However, despite the fact that this design offers a number of benefits (some of which were mentioned before), it mandates that the sensor nodes carry out a number of extra responsibilities in addition to their capturing responsibilities.

In this specific piece of research, the authors paid particular attention to this aspect by developing and testing an actual capture device. The latter makes use of a Fox Board card measuring 66 by 72 millimeters and weighing 37 grams, which is interfaced with a camera and a Wi-Fi USB key. Based on the results of the experiments that we carried out, it is clear that the suggested capture device is more than capable of simply meeting the criteria of the target application, provided that videos of a low or medium resolution are employed. In addition, despite the fact that we are utilizing an "ancient" Fox Board, we are certain that more recent versions are capable of managing films of a high definition.

This significant finding motivates them to proceed with our work in two primary directions: (a) first, they intend to make our capture device out of a different material (for example, a Stargate Board that is fitted with a PXA-255 XScale 400 MHZ RISC processor); this will be one of the main directions. They anticipate performances that are superior to those of Fox Board. (b) Secondly, they plan to validate their proposal by conducting an in-depth study of both the fusion server and the end-user server from two different points of view: the performance point of view, in order to support the significant workload generated by a significant number of sensor nodes; and the exploitation point of view, through the development of intuitive and interactive tools

that are dedicated to the exploitation of data. The overall architecture of their proposition is seen in Figure 2.37. It is made up of three primary components that are referred to as the capture device, the fusion server, and the end-user server.

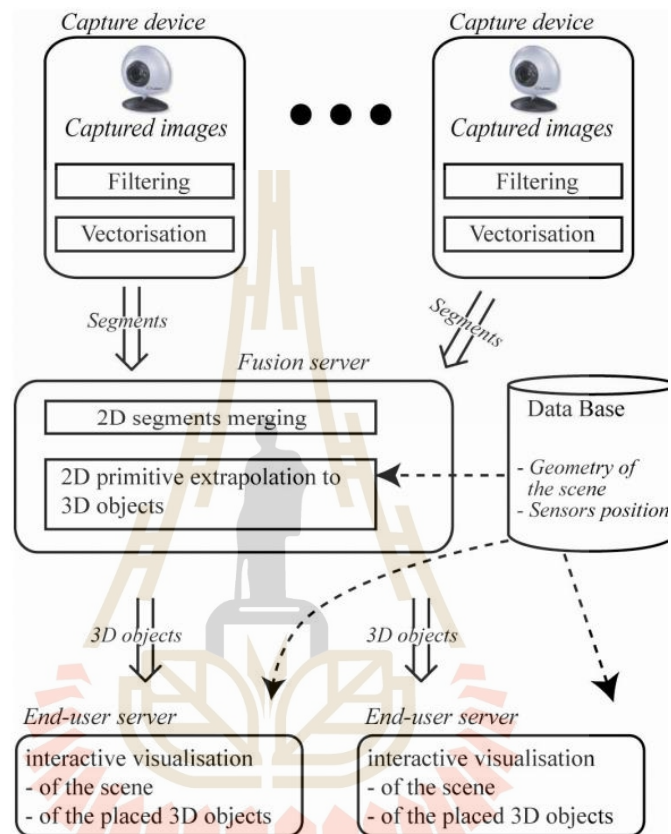


Figure 2.37 Functional Architecture (Ahmed M et. al., 2009)

CHAPTER 3

METHODOLOGY

In this research, an approach to the reconstruction of 3D meshes employing multi-modal point clouds, such as those obtained by depth scan and those inferred from the associated RGB picture, is presented as a possible solution. The figure seen in Figure 3.1 provides an overview of its description.

There are two data streams that make up the multi-modal input. In the beginning, a group of cloud points were taken from the scene's depth picture and placed into a separate file. After taking into consideration the camera characteristics and performing a calibration based on an acquisition, the physical coordinate of a given point was derived based on its depth. The natural light (RGB) image of the same scene was used as the input for the second pipeline in the process of inferring another set of point clouds using a deep learning (DL) network that had been trained before. After that, an outlier reduction technique known as spatial noise filtering was applied to the initial point cloud. After that, the CEICP that was developed was utilized to combine both point clouds, and then 3D TIN was ultimately rebuilt from those point clouds by utilizing BPA. A more in-depth explanation of this procedure may be found in the following subsections. To be more explicit, they are as follows: A) camera calibration; B) point cloud acquisition; C) depth estimate using deep learning derived from an RGB picture; D) entropy-based point cloud fusion; E) surface reconstruction utilizing BPA; and F) Simulation of a Wireless Visual Sensor Network.

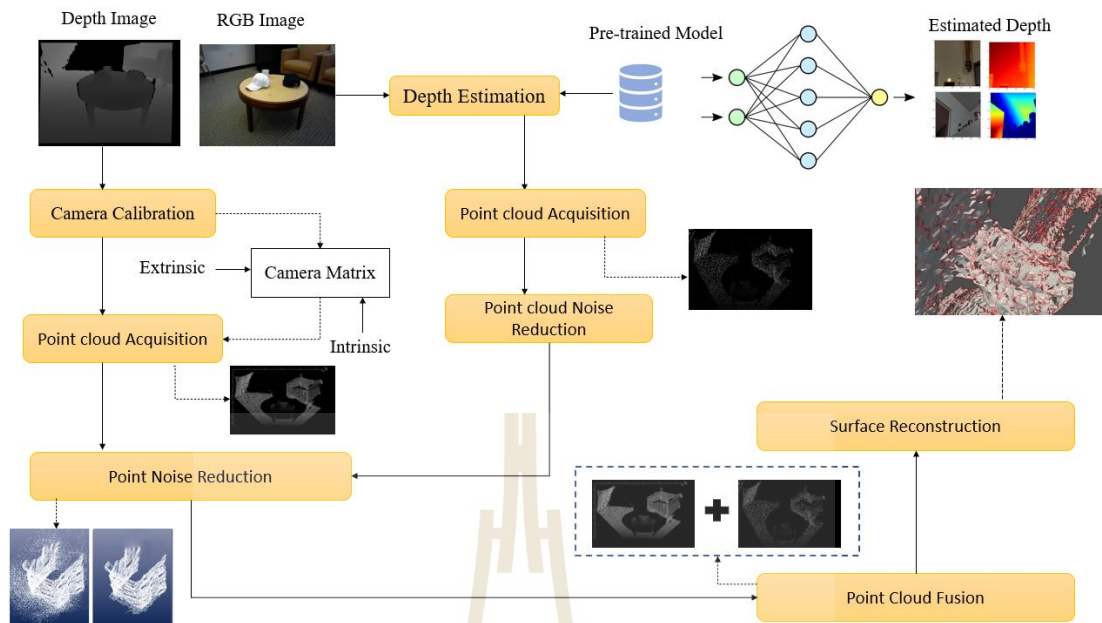


Figure 3.1 Overview of proposed framework

3.1 Experiment Environment Set-up

3.1.1 RGB-D Camera set-up

In the Figure 3.2, displays images depicting the configuration of the system. Kinect™ for Xbox One™ served as the study's RGB-D camera throughout its whole. It was made up of an infrared (IR) depth camera as well as an RGB camera, both of which had a spatial resolution of 640 x 480 pixels and a frame rate of 30 frames per second (FPS), respectively.

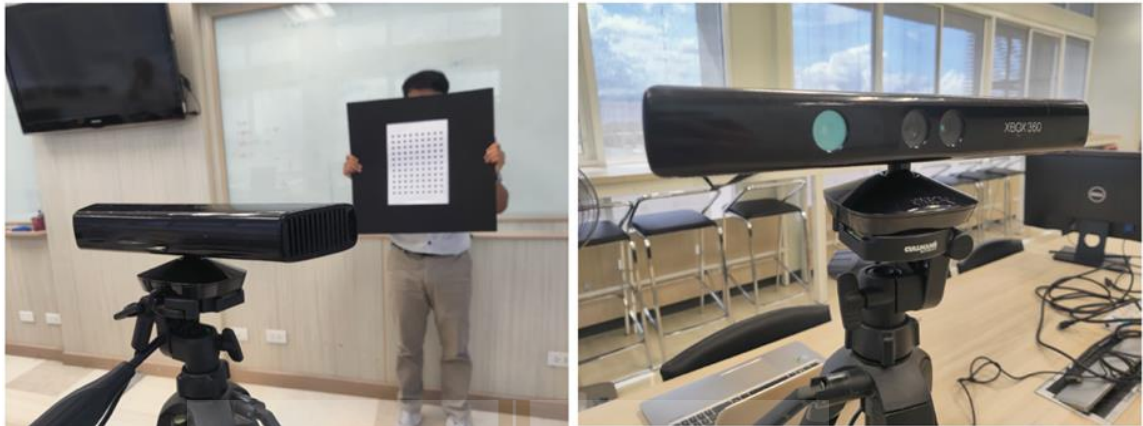


Figure 3.2 Pictures of system configuration, illustrating a Kinect™ for Xbox One™ (left) and a person holding a planar object (right).

Although it was anticipated that the proposed system could be generalized to fusing a static scene with a moving camera, so as to capture it at different aspects for better 3D coverage, the results are reported only for a pair of depth and RGB images, taken at the same time in the subsequent experiments. This is because it was anticipated that the proposed system could be generalized to fusing a static scene with a moving camera (Figure 3.3).

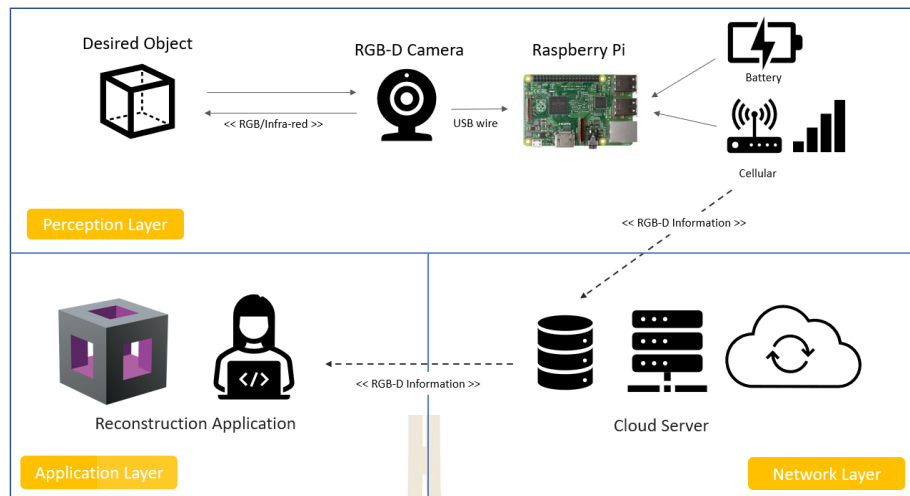


Figure 3.3 WSN (Simulated) Structure for reconstruction System

Simulated Wireless visual sensor network implementation was separated into three layers. In the perception layer, real-world object information is collected using RGB-D camera (RGB image with Depth information) connected with Raspberry Pi 4. The information is transferred to cloud server storage via cellular connection. Finally, the collected data is transferred to local client for three-dimensional model reconstructing. The result in Figure 3.4 show the RGB-D camera is moving around the desired object with the limited angle of viewpoint. This caused the occluded area occurs on the other side of object or obstacle object.



Figure 3.4 Example of RGB-D input

3.2 Camera Calibration

Camera Calibration is an important stage in the process of scene reconstruction from an 2D image. It investigates the possibility of determining the geometric factors that regulate the picture capturing. In this research, the factors that were taken into account were those of the camera (i.e., focal length, primary point, and skew of axis) and its geometry (i.e., rotation and translation), which were afterwards referred to as the intrinsic afterward and extrinsic parameters, respectively. These parameters were then approximated using known physical locations in the actual world and their projection on the picture plane after a calibration pattern with well-defined geometry had been provided. In order to get an accurate approximation of these parameters, it is necessary to rectify the distortion caused by the characteristics of the lens by employing below equation.

$$\begin{aligned} u &= x \cdot (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\ v &= y \cdot (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \end{aligned} \quad (3.1)$$

where the coordinates of a depth picture before and after rectification were denoted by the values (x, y) and (u, v) , respectively. The values k_1, k_2 , and k_3 represented the radial distortion coefficients, while r represented the distance from the coordinates (x, y) to the center of the lens, as specified by

$$r^2 = x^2 + y^2 \quad (3.2)$$

Afterwards, the intrinsic camera parameter is computed. The term "intrinsic parameters" refers to information that is unique to a camera and includes focal length and optical centers. For the purpose of providing a fundamental description of a camera lens, the focal length is commonly expressed as (mm). The term "focal length" does not relate to the size of a particular lens; rather, it describes the point at an optical distance at which light rays converge to form a clear and detailed picture for the underlying digital sensor. Focal length has nothing to do with the size of a lens. As

a representation of the case (f_x, f_y) . The singular point at which light rays continue to travel in the same direction after passing through the curvature of a lens is referred to as the optical center of the lens. Any other point on a lens will cause the light rays to be bent toward or away from the optical center depending on whether or not the lens is holding a convex or concave shape. In the case it is represented with the following notation (c_x, c_y) . The intrinsic parameters are contained in a 3 x 3 matrix as shown below.

$$\text{camera matrix } K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

Extrinsic parameters were used to characterize the relative location of the camera in 3D space. They were governed by a rotation matrix, which specified them. R , as well as a translation vector denoted by t , which moved the camera all the way from its starting point to where it is now. Imagine for a moment if the image was projected in perspective. Once the camera has been calibrated, a point that was captured by it and shown at (u, v) on the sensor will correspond to that which is located at (x, y, z) in the actual coordinate systems of the world coordinates.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R^{-1} \left(K^{-1} \begin{bmatrix} u_h \\ v_h \\ h \end{bmatrix} - t \right) \quad (3.4)$$

where h is a homogeneous coordinate and $u_h = u \cdot h$ and $v_h = v \cdot h$. Note that h depended on the distance from a point to the sensor in the camera coordinate system. However, since both u_h and v_h also varied as h , so during the calibration, h could be eliminate.

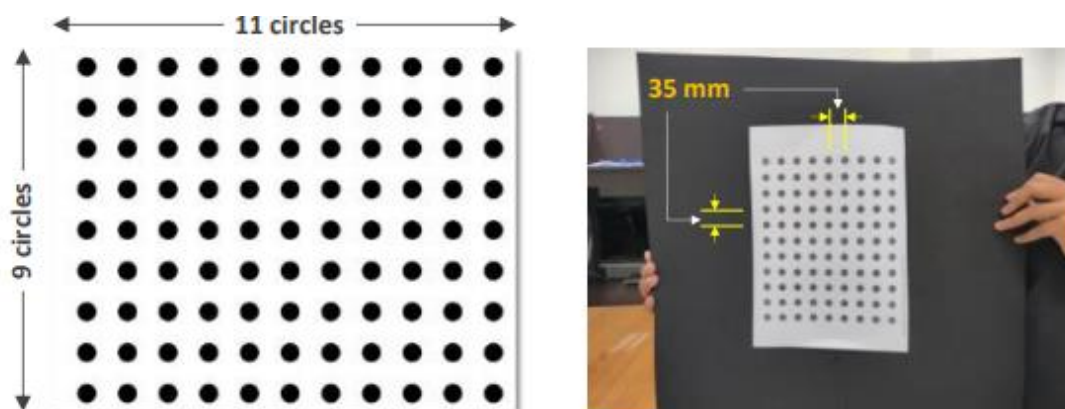


Figure 3.5 Calibration plane

Figure 3.5 shows the image that was used for the calibration process. It had 99 black circles on a white backdrop that were evenly dispersed at 9x11 and placed at 35mm intervals from one another. As can be seen in

Figure 3.5 it was printed out and then adhered to a piece of black cardboard, which acted as the calibration plane. These images of the aircraft were shot with the cameras aimed in two separate directions, each providing a unique perspective. Each perspective resulted in the acquisition of eight distinct plane orientations; four of these were used for the purpose of establishing the camera settings, while the other four were employed for testing purposes. The size of their image were 1030 by 1380 pixels. Figure 3.6 demonstrates the position of real-world camera (right), The real position of the calibration plane (left).

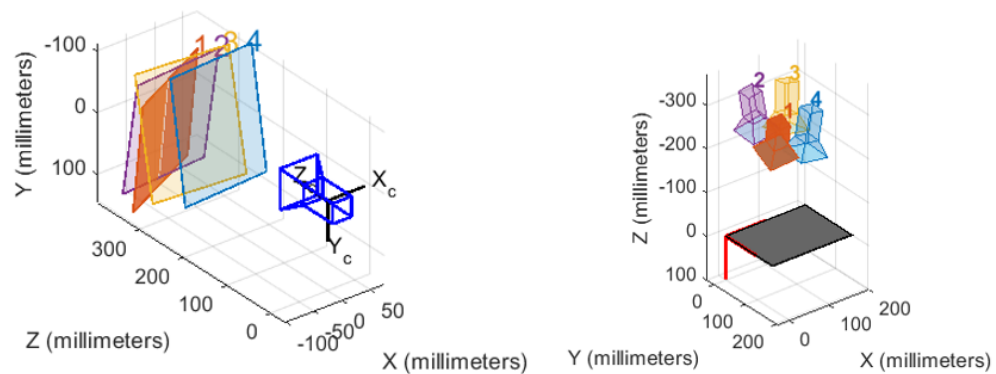


Figure 3.6 Camera position visualization

3.3 Point Cloud Acquisition

3.2.1 Point Cloud Extraction

After the camera has been computed and acquired, the point cloud in three dimensions will be retrieved from the depth picture. The camera projection matrix is made up of three components: the rotation matrix \mathbf{R} , the translation vector \mathbf{t} , and the intrinsic matrix \mathbf{K} . It is defined to convert from the coordinates of the globe to the coordinates of the screen as follows. Figure 3.7 shows the point cloud was derived from this depth image, which is an example of a depth image. It can be observed that the lens distortion and noise have been mostly controlled, but there are still some outliers and certain spots with depth that isn't defined.

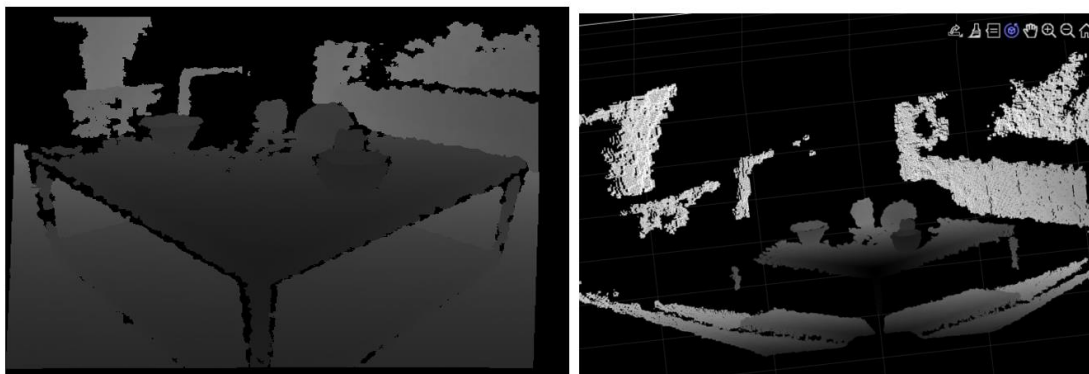


Figure 3.7 Result of Point cloud Acquisition process

3.2.2 Point Cloud Noise Reduction

Point cloud meshing may not give results that are satisfactory. On the other hand, if an excessive amount of regularization was used, the rebuilt mesh can be missing certain essential elements or have erroneous geometry. In order to solve this problem, also known as maintaining their features while denoising the data, a statistically based structural adaptive filter was applied to the retrieved points in a direct manner. In the event that the distribution of a point cloud was found to be locally Gaussian, an outlier would be eliminated if it was found to reside outside of this distribution. Let's choose a random point in the cloud and call it p . After that, the below equation, was used to determine the typical distance between it and its N closest neighbors.

$$d(\mathbf{p}) = (1/N) \sum_{\mathbf{q} \in \Omega_N(\mathbf{p})} \|\mathbf{p} - \mathbf{q}\| \quad (3.5)$$

Let the mean and standard deviation of d over all points be denoted by the symbols μ and σ , respectively. After then, a location would be designated an outlier if the average distance to its N closest neighbors was more than or equal to a predetermined threshold. To put it another way, if we start with the original point cloud denoted by P_0 , we may define its smoothed version as P .

$$\mathbf{P} = \{\mathbf{p} \in \mathbf{P}_0 \mid d(\mathbf{p}) \leq \mu + t \cdot \sigma\} \quad (3.6)$$

where t represented some kind of empirical threshold. In this particular research, the values for N and t were determined to be 4.0 and 1.0, respectively. This filter was resilient and adaptable to both sparse points and large outliers due to the fact that the threshold, t , was determined based on the statistics of distances.

3.3 Depth Estimation from RGB

3.3.1 CNN Architecture

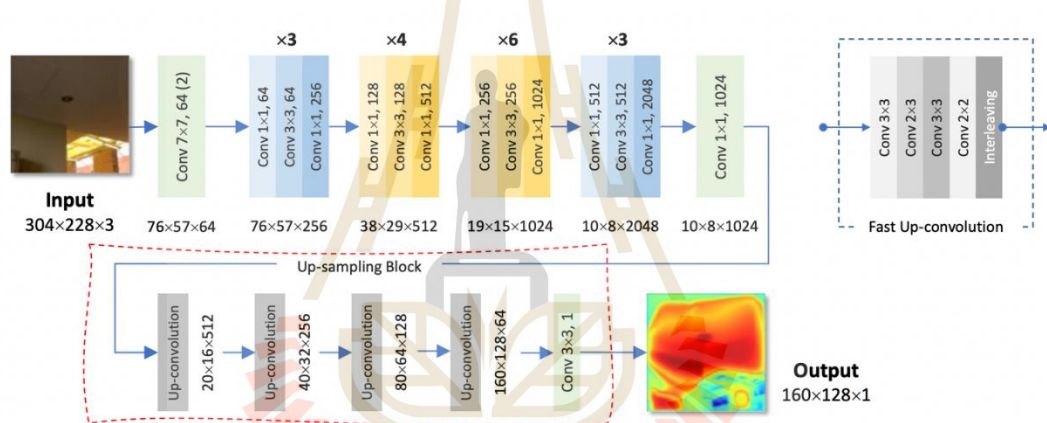


Figure 3.8 Visualization of ResNet-50 for Depth Estimation

The process of estimating depth from images is an essential step in computer vision as well as in a wide variety of other applications, such as simultaneous localization and mapping (SLAM), navigation, object recognition, and semantic segmentation, etc. In particular, the assignment is an important step in deducing the geometrical aspects of the scene that is under the surface. The quality of the cloud points that are extracted, however, is highly dependent on the surface properties, its continuity, its texture pattern and repetitions, and the lighting environment. This is because the acquisition of depth from a stereo camera is dependent on the analysis of projected light patterns on an object surface using epipolar geometry. It is common

knowledge that the presence of certain elements, such as voids, ambiguity, and degraded or missing features, etc., all lead to an inaccurate 3D reconstruction. We hypothesized that these inaccuracies may be reduced if the data were combined with considerably more regularized depths and interpreted using other methods, such as a visual cue. As a result, reconstruction via fusion is going to be one of the most important contributions made by this work. In order to accomplish this goal, the second set of point cloud data that would be fused was inferred using CNN from an RGB picture. In this subsection, as opposed to those that were previously recovered using an infrared camera, depth values were calculated from a image taken with natural light. Inspired by a method of self-supervision that was proposed by Godard et al., in which the depth map was calculated using a mix of network topologies, this research was carried out. The approach made its predictions about depth by utilizing a fully connected U-Net, and it made its predictions regarding poses between picture pairs by utilizing a pose network that used ResNet-18 as its encoder. In addition, the pre-trained version of ImageNet was used to initialize the weights.

For the purposes of this investigation, a customized version of the ResNet-50 (Figure 3.8) was used. The network was trained with RGB pictures and their associated depth images, which served as input and target, respectively, during the training process. The KITTI dataset was combed through to gather imaging data, all of which were of the size 304 by 228 pixels. The network was set up using a configuration of 22 layers, a batch size of 32, a learning rate (LR) of 0.0002, and 30 epochs. The appearance-based loss function was utilized all during the training process. In addition, we implemented a modified minimum reprojection loss, which was computed for each individual pixel, and eigen splits were utilized in order to provide an approximation of the final depth map. In a manner analogous to that described in the preceding paragraph, a resulting depth map was turned into a dense point cloud. Taking an information-theoretic approach, the next section provides a detailed description of how to combine the point cloud that was captured by an infrared camera with the point cloud that was estimated using a modified version of ResNet-50. The result from the depth estimation with modified ResNet-50 is demonstrated below. Figure 3.9 is examples of RGB photos (on top), together with their corresponding ground truth and

estimated depth images (middle and bottom) (bottom). It is important to note that these estimations correspond to ground plane perception, the vanishing point, and the relative sizes of items in the scene, among other things.

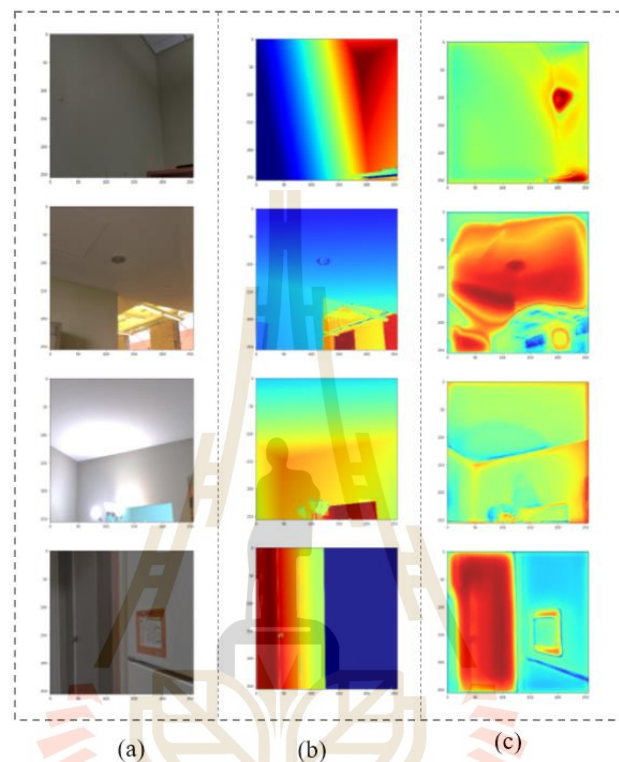


Figure 3.9 Example of Depth Estimation Result

The depth map image was used in to extract set of point cloud using eq. (3.4). Then, the inferred point set is employed to perform fusion in the next process.

3.4 Point Cloud Fusion

It is common knowledge that combining the geometrical data collected from many sensors may significantly improve the quality of the 3D model reconstruction in comparison to the results achieved by using only one sensor on its own. The information that is absent from one collection of data might be completed or suggested by the information that is available in another group. Therefore, in order to verify that the reciprocal is valid, it is necessary to build a thick correlation between

the source clouds and the destination clouds. The iterative closest point (ICP) technique is one of the most successful approaches, and it is frequently used in the research that has been published. The fundamental algorithm for this method, as well as several proposed variations, are detailed below.

It is important to point out that ICP was often used to align dynamic objects. This is something that should be highlighted here. But in the example that was given, it was presumed that there was no movement at all between the cloud points that were extracted and those that were calculated. Despite this, their outward appearances were distinct from one another as a result of complementary interpretations of depths, which were separately based on epipolar geometry and visual cues learned by a CNN. Not only could voids arise in one set but not in the other, but it's also possible that their geometrical aspects do not coincide with one another. These required a unique approach to be taken with regard to their similarity metric, which is what is being presented here.

3.4.1 Point Cloud Registration

If the correspondences are already known before the optimization process begins, for instance through the use of methods for feature matching, then the optimization process simply needs to estimate the transformation. Correspondence-based registration is the name given to this particular mode of registration. On the other hand, if the correspondences are not known, then the optimization must be performed in order to jointly find out the correspondences and transformation at the same time. A simultaneous pose and correspondence registration are an example of this particular kind of registration.

3.4.2 Traditional Iterative Closest Point Registration (ICP)

The ICP algorithm is extremely useful for a variety of applications, including but not limited to the following: reconstructing an object from multiple surfaces; aligning an anatomical model to a patient-specific scan; localizing a moving robot; and optimizing the path planning of said robot (especially in situations where an equipped wheel odometry is unreliable due to slippery terrain). ICP's primary objective is to identify the transformation that results in a given source point being sent to the point that is either the closest possible match to it or the point that is the

most possible match to it. This is done in an effort to reduce the overall gap between the two-point sets with regard to some metric. On the other hand, these matched pairings could not all be perfect correspondences depending on the original orientation and capture range; hence, the transformations might not be unique at initially. Therefore, the ICP will continue to do this procedure, which involves iteratively updating the correspondences, until coverage is achieved. The conventional ICP method is successful, however it has a low level of efficiency. It has a slow convergence rate, which is notably noticeable when compared to a pair of cloud points with better resolutions. As a result, this research also offered a modified ICP based on cross-entropy and designated it as CEICP. This was done as another contribution to the topic. Figure 3.10 illustrated ICP seeks to establish the closest pairs (dashed lines) (b) between the source (Q) and destination (P) points (a), and then iteratively ($t > t+1$) finds the transformation that best matches them (c). The traditional ICP algorithm is demonstrated in Table 3.1.

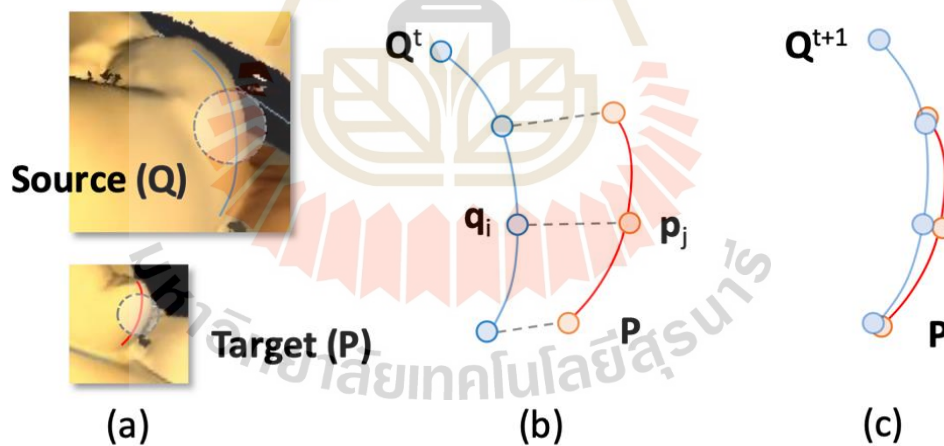


Figure 3.10 ICP Fundamental

Table 3.1 Traditional Iterative Closest Point

ALGORITHM	Iterative Closest Point
INPUT	Reference point set P , New point set Q
OUTPUT	Fused point set X
BEGIN	
1:	COMPUTE $center_P$ and $center_Q$
2:	TRANSLATE Q into P
3:	WHILE $d(T) > Error_{max}$ OR $iter < max_{iter}$ DO
4:	FOR EACH p_i IN P
5:	COMPUTE $nearest_point(p_i, Q) \rightarrow S_i$
6:	ENDFOR
7:	TRANSFORM $\min_T \sum_i \ q_i - T(p_i)\ ^2 \rightarrow T$
8:	COMPUTE $transform_point_set(P, T) \rightarrow P$
9:	INCREASE $iter + 1 \rightarrow iter$
10:	END WHILE
END	

3.4.3 Cross-Entropy Iterative Closest Point

The ICP makes an effort to align several point sets, with regard to a distance measure, whenever it does a normal registration. However, such a measure is often sensitive, which results in ICP performing badly whenever either dataset has a considerable number of outliers or has a higher noise floor. This is because such a measure is sensitive to general variations in the data. This is notably the case in photogrammetry, which evaluates depths rather than immediately determining them on the surface of an item. In addition, the point clouds that were used in this study were of the same scene; however, they were acquired using two different imaging modalities, which resulted in differences not only in precision but also in the interpretation of depths. These differences were achieved by correlating projected light patterns and by deep learning from visual cues, respectively. Because of this, their outward looks were distinct, and the degree to which they are alike may be ascertained by comparing the information they share with one another.

The objective of the CEICP that has been presented is to locate the right correspondence between two different sets of cloud points. To put it another way, it was the one that provided the probability distributions of random variables selected from these sets that were the most appropriate. Cross-entropy (CE) between

distributions P and Q of the same underlying event (i.e., a 3D scene) quantifies the average amount of data units (or bits) necessary to uniquely identify a co-existing event (p, q) , where $p \in P$ and $q \in Q$ are the distributions P and Q . This is the definition of cross-entropy. Let's say that Q was an estimated probability distribution of the real distribution, P , and that P was the genuine distribution. An index mapping between a point, p , and the one that is closest to it, q , according to the present rigid transformation, was determined throughout each iteration of the process. After that, a fresh transformation was computed in an effort to send all of the points in P to the points in Q that are geographically closest to them. The procedure was repeated while maximizing CE in order to achieve convergence; this is analogous to the standard ICP. The equation that expresses the cross-entropy of the distribution Q in comparison to the distribution P across a certain sample space known as $H(P, Q)$ is as follows:

$$H(P, Q) = E_{PQ}[-\log f(p, q)] \quad (3.7)$$

where $E_{PQ}[\cdot]$ is an expected value operator, with respect to the joint probability of both distributions. The above definition may be formulated using Kullback-Leibler divergence, $D_{KL}(Q | P)$, of Q from P , which is also known as the relative entropy of Q with respect to P , i.e.,

$$H(P, Q) = H(Q) + D_{KL}(Q | P) \quad (3.8)$$

Let $H(Q)$ is an entropy of $H(Q)$ The relative entropy D_{KL} can be determined for probability distributions P and Q that are both specified on the same support, X , by measuring the additional information that is necessary to encode samples from P using a coding that is optimized for Q . Due to the fact that X was a surface embedded on R , the information I was projected onto the norm of its L value. In addition, because $H(Q)$ remained the same during ICP, optimizing Eq. (3.9) to its maximum is comparable to optimizing the information obtained by ICP.

$$I = - \sum_i f(\|p_j - q_i\|) \log f(\|p_j - q_i\|) \quad (3.9)$$

It is obvious that Eq (3.9). was optimized to its full potential when both P and Q were in perfect alignment. In addition, the sample evaluations of Eq (3.9). that were carried out during CEICP are presented in Figure. However, determining for q_i the optimum correspondence (i, j) that maximizes I required a significant amount of computing effort. Instead, each distinct piece of information was inferred by the use of w_{ij} in Algorithm I. Let P and Q be the input point clouds, where P was produced from an infrared depth picture and Q was inferred by ResNet-50 from a image of the same scene. After that, we looked for a transformation, denoted by $T = R | t$, that would provide the best correspondence possible between these point sets. First, for each point q_i in Q , determine the point p_j in P that is closest to it in terms of the Euclidean distance between them, which is provided in Eq.(3.10).

$$d_{ij} = \|p_j - q_i\| \quad (3.10)$$

Calculate the central element, $H(p, q)$ of the pair $x_{ij} \in X$ by utilizing Eq. (3.8) and Eq. (3.10) The pair was considered to be an outlier and removed from the support, X , if the distance d_{ij} was greater than a previously defined threshold known as T_d . Following that, for each of the remaining pairings (i, j) , their contribution to the alignment was determined by the information weight, w_{ij} , which was then stated by the Eq. (3.11)

$$w_{ij} = -f(d_{ij}) \log f(d_{ij}) \quad (3.11)$$

where f represented the estimated Gaussian distribution of d at each iteration of the process. Due to the existence of this correlation, the rotation (R) and

translation (\mathbf{t}) matrices that were responsible for the transformation from P to Q may be determined by employing the singular value decomposition technique (SVD). In order to do this, the centroids of P and Q were determined by taking into consideration just the locations that were supported by X , and they were correspondingly represented as \mathbf{c}_p and \mathbf{c}_q . After that, both point clouds were translated using Eq. (3.12), such that their respective centroids would coincide with the origin. These new point clouds were then designated as P' and Q' , respectively.

$$\mathbf{c}_p = \frac{1}{N_j} \sum_j \mathbf{p}_j \mathbf{c}_q = \frac{1}{N_i} \sum_i \mathbf{q}_i \quad (3.12)$$

The covariance matrix, weighted by \mathbf{W} , was evaluated, and decomposed by using SVD, as Eq. (3.13).

$$\mathbf{W} = \text{diag} [w_{ij}] \quad (3.13)$$

$$\mathbf{P}'\mathbf{W}\mathbf{Q}'^T = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T \quad (3.14)$$

Lastly, Eq (3.15) and Eq (3.16) were used to determine the matrices \mathbf{R} and \mathbf{t} that most effectively pushed Q to Q^* , which is the value that is closest to P .

$$\mathbf{R} = \mathbf{V}\mathbf{U}^T \quad (3.15)$$

$$\mathbf{t} = \mathbf{c}_p - \mathbf{R}\mathbf{c}_q \quad (3.16)$$

At each iteration, the error of a resultant transformation \mathbf{T} , i.e., $E(\mathbf{R}, \mathbf{t})$, was given by Eq. (3.17)

$$E(\mathbf{R}, \mathbf{t}) = \sum_{(i,j) \in X} w_{ij} \|\mathbf{p}_j - (\mathbf{R}\mathbf{q}_i + \mathbf{t})\| \quad (3.17)$$

It was formulated in terms of the weighted distances that separated P and Q^* . This process was continued by CEICP until either the accuracy requirements were satisfied, convergence was achieved, or the number of iterations reached their maximum (t_{MAX}). After both Q^* and P were finished, the point clouds from each were combined into a single one. An example of CEICP result and the corresponding fusion is illustrated in and Figure 3.11. It demonstrates Fusion of point clouds obtained from a depth image and calculated using a modified version of ResNet-50, illustrating instances in which the point clouds were complementary to one another and in which their holes coincided.

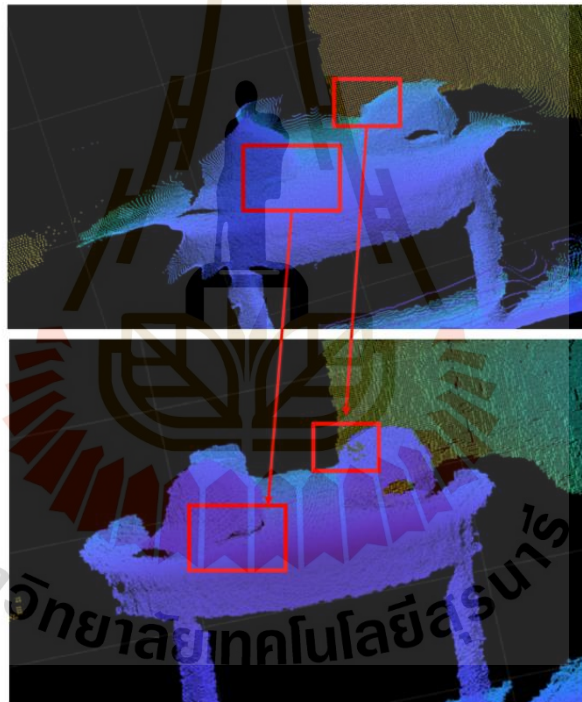


Figure 3.11 Example of Fusion Result of indoor scene

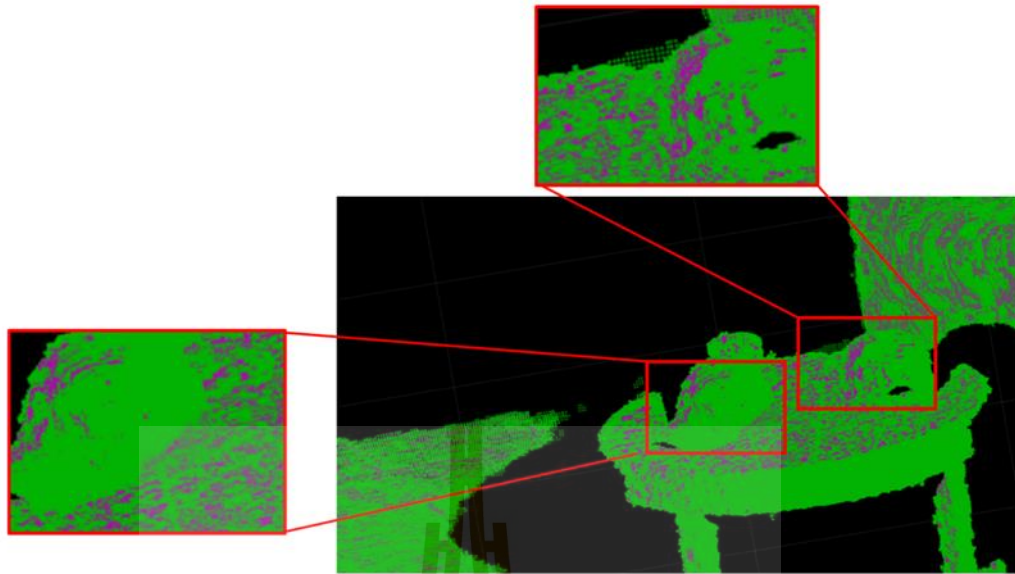


Figure 3.12 Fusion result comparing with the original point cloud

3.5 Surface Reconstruction

This subsection provides a full description of the Ball Pivot Algorithm (BPA-based) surface reconstruction that was implemented. A 3D TIN of a scene was built using the fused point cloud that was obtained from the step before this one. In the method that was developed (Bernardini et. al, 1998), a shape was represented in the form of a rolling ball. It was founded on the concept that if a sampled dataset P is sufficiently dense, then a sphere of a particular radius cannot transit through it without colliding with one or more points inside it (Lotem Nadir, 2022). This assumption served as the foundation for the model (H. Seo et. al, 2019). As a result, mounting the

initial three points with a ball is the first step in the BPA. This ball will continue to retain its interactions with two of these spots while it pivots until it makes contact with another point.

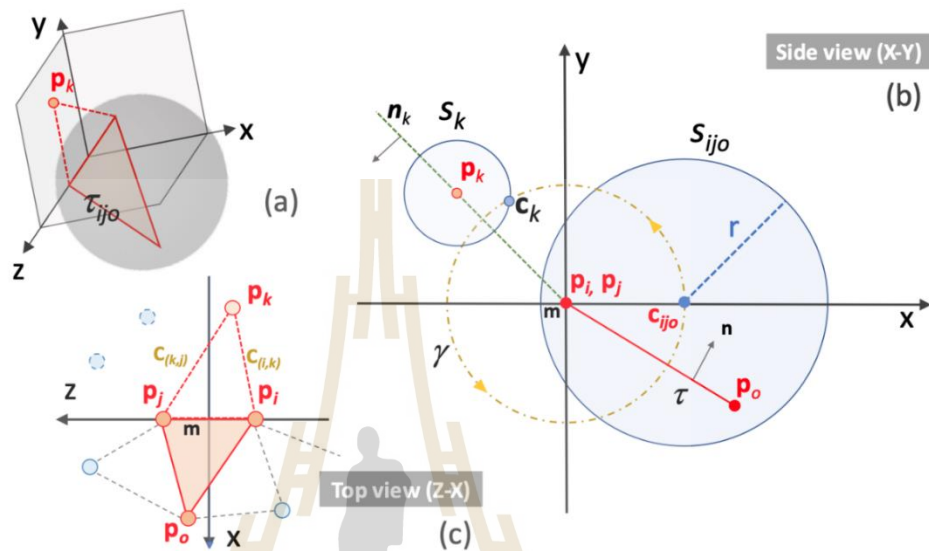


Figure 3.13 Ball-pivot Algorithm (BPA)

The BPA algorithm was started by selecting a seed triangle from the existing point cloud. To be more specific, it is defined by a chosen point and the two points that are immediately next to it. A ball with a radius of r was pivoting on the edge (p_i, p_j) of the triangle τ_{ij0} , which lay on the z – axis, given that the triangle τ_{ij0} consisted of the vertices p_i , p_j and p_o . Before moving on to the next step, the sphere was examined to see whether it included any more points in the cloud. if it did not, a new triangle was selected. The local coordinate that is displayed in Figure 3.13 is given in such a way that the origin is at the same location as the midpoint of this edge (m). At first, this r – ball made its initial contact with the $(x - y)$ plane at the circle S_{ij0} , which was centered at c_{ij0} . The center of the ball traveled along the trajectory as it pivoted on the edge (p_i, p_j) , which means it travelled around m with a radius $\|c_{ij0} - m\|$. When the ball found a new point, p_k , it intersected the $(x - y)$ plane at a new circle, S_k , and its center shifted to c_k as a result of this discovery. n_k was responsible for defining the

orientation of the new intersecting line from m to p_k , and as a result, the newly discovered triangle (p_i, p_j , and p_k). This procedure was done several times, each time turning on a side that had not yet been explored, until all points had been thoroughly explored. After that, the generated mesh provided an approximation of the 3D surface underneath it. The fact that this method only needed linear amounts of time and storage made it incredibly effective. However, its difficulties included accurately determining the ball radius or managing point clouds that were sampled less frequently than one and also included those that included an excessive number of voids. Despite this, the multimodal fusion technique that was suggested was successful in resolving these more recent problems. The example result of Ball-pivot algorithm surface reconstruction with holes are illustrated in Figure 3.14 and Figure 3.15.

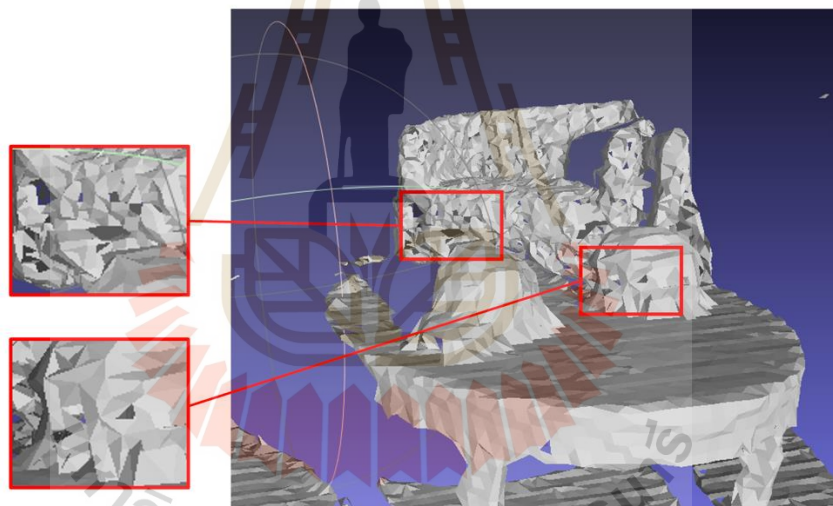


Figure 3.14 from Ball pivot algorithm surface reconstruction before fusion

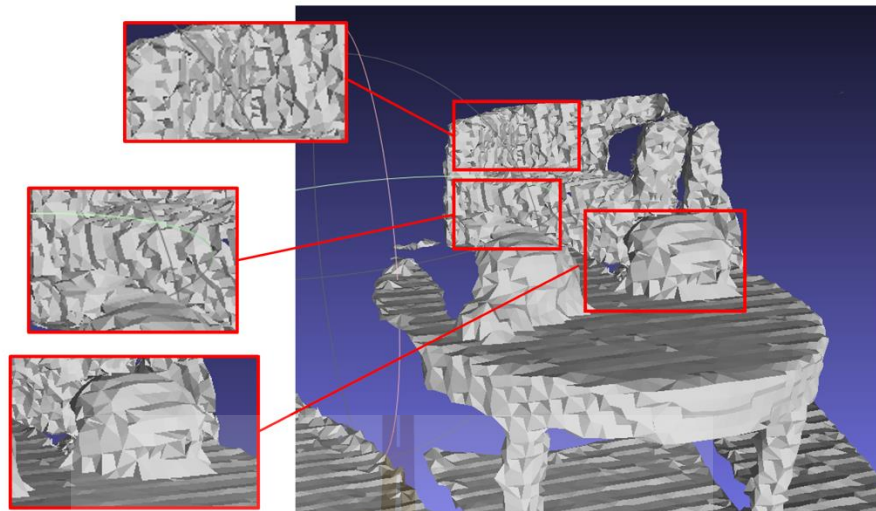


Figure 3.15 Result from Ball pivot algorithm surface reconstruction

3.6 Wireless Visual Sensor Network Simulation

The Wireless Sensor Network (WSN) is now a popular research subject. There are still many network details that have not been established and standardized in WSNs. Putting together a testbed for WSNs may be highly expensive. It can be time-consuming and expensive to conduct genuine trials on a testbed. In addition, reproducibility is severely hindered due to the fact that several factors have an influence on the findings of the experiment at the same time. It is difficult to focus on one facet. In addition, doing genuine tests always takes a significant amount of time. As a result, the simulation of WSNs is an essential part of the development of WSNs. Evaluation on a very wide scale is possible for everything from protocols and schemes to brand new concepts. Users of WSNs simulators are given the ability to tune adjustable settings in order to isolate distinct causes.

As a consequence of this, simulation is necessary for the research of WSNs because it is the standard method for testing novel applications and protocols out in the field. Because of this, there has been a recent uptick in the creation of simulators.

However, drawing reliable inferences from simulation research is not a process that can be considered simple. The accuracy of the simulation models and the appropriateness of a specific tool to implement the model are the two most important

features of WSNs simulators. (1) The correctness of the simulation models and (2) the appropriateness of a particular tool to implement the model. In order to arrive at reliable conclusions, it is essential to begin with a "proper" model that is founded on sound assumptions. The essential choice that has to be made is between performance and scalability and accuracy and the importance of details. In the next section of this study, various mainstream WSN simulators will each be discussed in further depth before being compared to one another.

3.6.1 Type of Simulation

- Discrete-event and Trace-driven

Since discrete-event simulation can quickly replicate hundreds of jobs operating on various sensor nodes, it is a popular choice for usage in wireless sensor networks (WSNs). Some of these components are included in the discrete-event simulation. This simulation has the ability to list upcoming occurrences, each of which may be modeled using different routines. The global variables, which are responsible for describing the current state of the system, are able to reflect the current time of the simulation, which enables the scheduler to make accurate predictions regarding this time period. Input routines, output routines, beginning routines, and trace routines are all a part of this simulation. Additionally, this simulation offers dynamic memory management, which allows new things to be added to the model while allowing the removal of older ones. Users are able to inspect the code in a step-by-step manner without interfering with the functioning of the program thanks to the debugger breakpoints that are supplied in discrete-event simulation.

Trace-Driven Simulation, on the other hand, offers a diverse set of services. In the real system, simulations of this sort are utilized rather frequently. The outcomes of the simulation have a higher level of trustworthiness. It offers a more realistic workload, and the detailed information it gives enables users to do in-depth research on the simulation model. Throughout most cases, the values of the inputs remain identical in this simulation. Nevertheless, this simulation has a few limitations that you should be aware of.

- Simulator and Emulator

It is common practice to employ Simulator in the process of developing and testing protocols for WSNs, particularly in the preliminary stages of these systems. The expense of simulating networks with thousands of nodes is quite minimal, and the simulation may be completed in an extremely little amount of execution time. For the purpose of simulating WSNs, there are both generic and specialist simulators available for usage. Emulator is the name given to the piece of software that, to carry out the simulation, makes use of both hardware and firmware. The software implementation and the hardware implementation can be combined in emulation. Since an emulator implements in actual nodes, it may thus deliver a performance that is more precise. Emulators often have excellent scalability, which allows them to imitate many sensor nodes all at the same time.

3.6.2 Wireless Visual Sensor Network Structure

illustrates the process that must be followed. Let's call the group of visual sensors that have previously been calibrated C , where $C = \{C_1, C_2, C_3, \dots, C_N\}$. They are located in a given region. The images set I that C developed looks like this: $I = \{I_1, I_2, I_3, \dots, I_N\}$. The number of visual sensors that may be employed to reconstruct a 3D scene is restricted to a maximum of M , where M is a smaller number than N . N is the total number of sight sensors that are at disposal. The overall of Wireless Visual Sensor Network demonstrates in Figure 3.16.

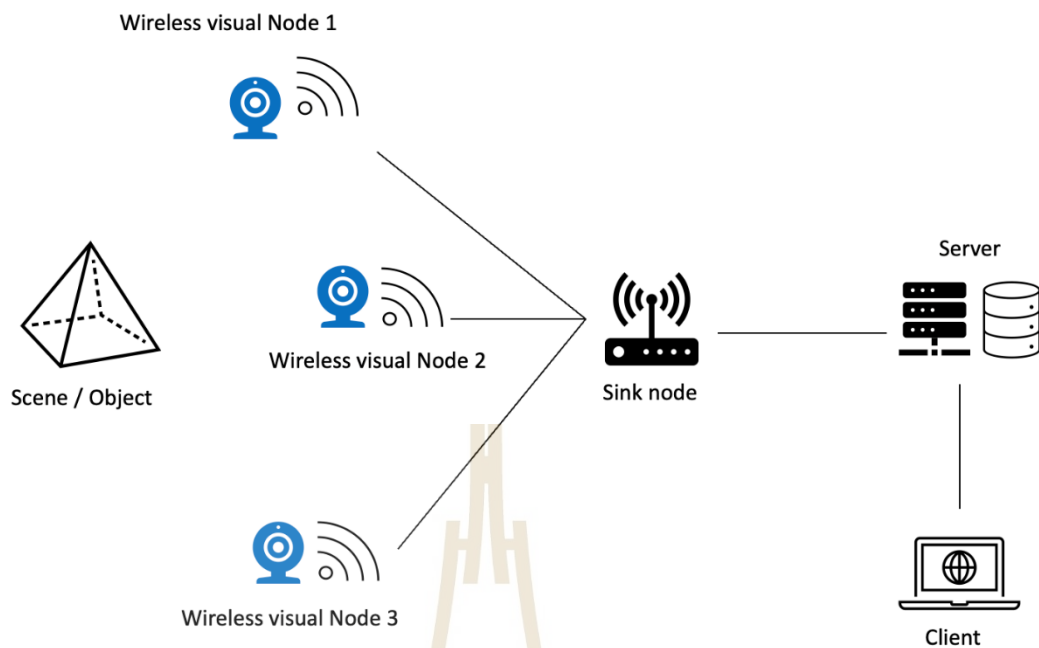


Figure 3.16 Wireless Visual Sensor Network Architecture

A full-fledged Wireless Visual Sensor Network was constructed with MATLAB™ Simulink for the purposes of this study. The simulation process begins with the construction of the hardware architecture of the transmitting nodes and continues with the modeling of the communication channel as well as the design of the receiving master nodes. As can be seen in Figure 3.16, the simulated system model specifies that there are three sensor nodes fitted with an embedded Linux operating system that has the capacity to display imagesKinect™ for Xbox One™. It consisted of an infrared (IR) depth and RGB cameras, whose spatial resolution and frame rate were 640x 480 pixels. Image size is 901 kilobytes, and the file is in the data raw format known as portable pixel map (ppm). The transmitted data result is illustrated in the next Section.

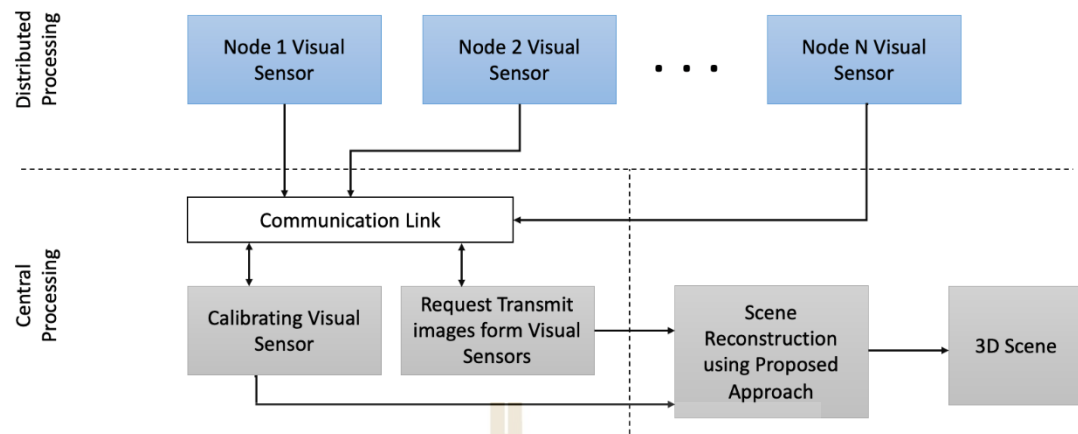
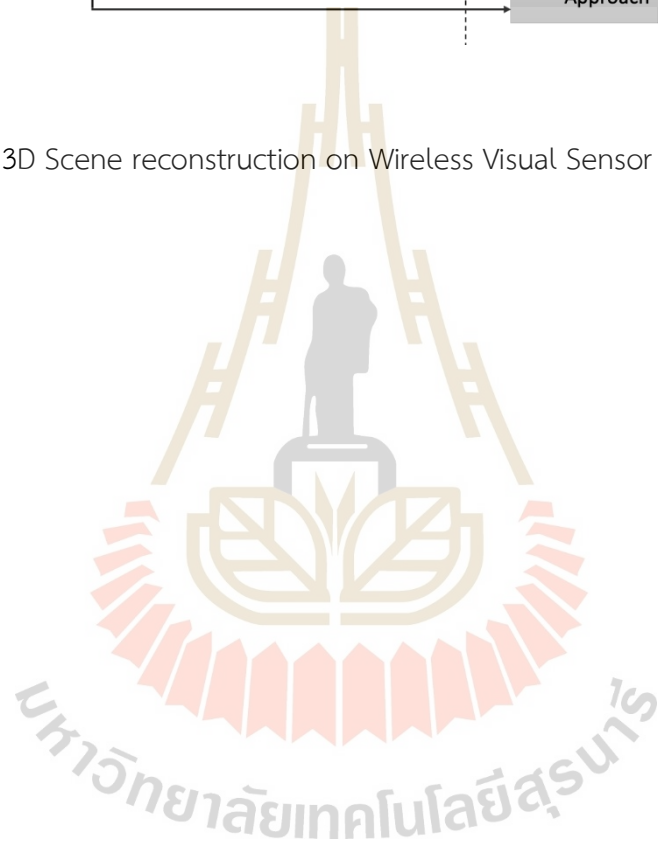


Figure 3.17 3D Scene reconstruction on Wireless Visual Sensor Network Mechanism



CHAPTER 4

EXPERIMENTAL RESULTS

In this part, the experimental surface reconstructions of chosen scenes that were created using the suggested technique are presented, and their results are discussed. It is broken down into four pieces based on the essential processes that go into it, which include camera calibration, depth estimation using deep learning, cloud point fusion, and surface reconstruction to round off the list. In particular, because fusion that is based on cross-entropy is the contribution of this research, it was benchmarked not only against the method that is considered to be the standard, but also against a number of other algorithms that are considered to be state-of-the-art.

MATLAB™ v.2020a was employed these tests to implement camera calibration, depth estimation, the extraction of point clouds, the noise reduction of those point clouds, and the fusing of those point clouds. Python version 3.10 was used in the writing of the surface reconstruction. All of the code was run on a Windows® computer that has an Intel®Core™ i7-7700HQ processor operating at 2.81 GHz and 8 gigabytes of RAM.

4.1 Camera Calibration Result

In Table 4.1 and Table 4.2 illustrated two experiments which conducted in order to evaluate the performance of this procedure by testing its resilience against noise while retrieving camera parameters. The results of these experiments are presented below. On a turn-by-turn basis, Gaussian noises with zero mean and 0–1.0 standard deviations were added to the calibration image. These noises were applied to the image in a variety of orientations and viewpoints. Table 3.1 and Table 4.1, respectively, present the results of the analysis of the retrieved intrinsic and extrinsic parameters. In the first case, f_x and f_y represented the focal length along their respective axes; (u_0, v_0) represented the lens center; and k_1 and k_2 represented

the lens' radial distortion in millimeters squared (mm^2) and millimeters squared (mm^4) respectively. In the latter, the translations were denoted by t_x , t_y and t_z , while the rotations were denoted by r_x , r_y and r_z , in accordance with their respective axes.

Table 4.1 Intrinsic Parameters

Noise	f_x	f_y	u_0	v_0	k_1	k_2
0	500.000	500.000	300.000	250.000	0.0500	0.1000
0.2	500.013	500.009	300.132	249.855	0.0503	0.1023
0.4	499.738	500.343	300.273	249.762	0.0443	0.1048
0.6	498.836	501.538	299.317	249.346	0.0634	0.0936
0.8	501.329	500.938	299.021	250.829	0.0849	0.1274
1.0	502.474	497.638	301.327	251.043	0.1382	0.0632

Table 4.2 Extrinsic Parameter

Noise	f_x	f_y	u_0	v_0	k_1	k_2
0	60.173	-10.193	17.757	30.0027	19.9969	18.0053
0.2	60.237	-9.661	18.329	29.9965	20.0053	18.0072
0.4	59.538	-10.472	18.517	30.0051	20.003	17.9929
0.6	60.621	-10.613	17.104	30.0062	20.0089	17.9892
0.8	60.126	-10.169	17.973	30.002	21.67	17.999
1.0	0.3535	0.3422	0.5001	0.0035	40794	0.0071

Except for radial distortions, which were extremely sensitive and displayed considerable errors (bold red), at increasing noise levels, the values for the majority of the recovered intrinsic parameters were well within 0.5 percent errors. Radial distortions were the exception to this rule. Nevertheless, the errors of the extrinsic parameters were held to a maximum of 5 percent over the whole spectrum of noise levels. According to this, the noise level should be maintained at or below 0.5.

4.2 Depth Estimation Result

The errors in depth estimation caused by the proposed CNN, which is based on the modified ResNet-50, are compared to the actual measurements that are published in the KITTI indoor dataset in Figure 4.1. These errors are presented for each scene. The root-mean-square error (RMSE) and the mean absolute relative error (REL), expressed as a percentage, were, respectively, 6.328, 7.944, and 5.647 for scenes 1, 2, and 3, and 0.206, 0.361, and 0.161 for each of those scenes (Table 4.3). The RMSE of the estimations came in at 6.640, which was 0.963 on average. In spite of the fact that its homogeneity was excellent, the RLE of 0.243 and 0.086 suggested that its precision was subpar in comparison to that which could be obtained directly from the depth image.

Table 4.3 Depth Estimation Result

Training Data	RMSE (lin.)	RMSE (log.)	Abs. Rel.
Scene 1	6.328	0.284	0.206
Scene 2	7.944	0.339	0.361
Scene 3	5.647	0.236	0.161

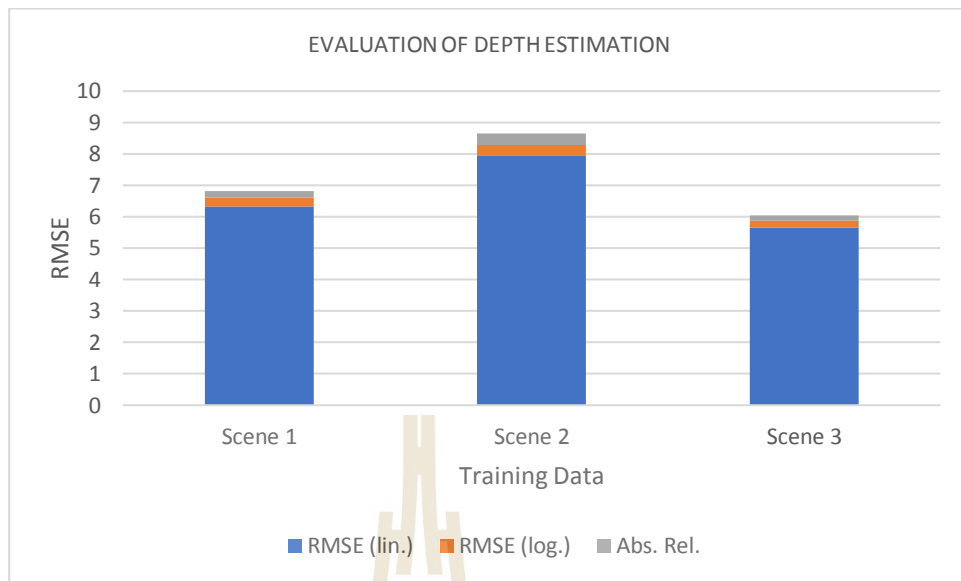


Figure 4.1 Visualization of evaluation of depth estimation. Errors of the estimated depth by ResNet-50 from 3 scenes. Despite consistently low errors, they were inferior to direct extraction.

4.3 Point Cloud Fusion Result

The above result makes it abundantly clear that the depth information learnt and approximated by ResNet-50 was dependable but not very accurate. This is evidenced by the fact that the information was reliable. As a result, it was complementary to the information that was obtained from the infrared scan by means of fusion in this particular investigation. Existing works, such as basic ICP, picky ICP (PICP), RICP (E. Trucco et. al, 1999), multi-resolution ICP (MRICP) (T. Jost et. al, 2003), fractional ICP (FICP) (J. M. Phillips et. al, 2007), and hue ICP (H. Men et. al, 2010), were used as benchmarks for the proposed CEICP fusion in order to assess its efficacy (HICP). These approaches were evaluated using the RGB-D dataset that could be found on KITTI's website (A. Geiger et al, 2013). This data collection was comprised of three different scenarios, each of which included one hundred frames. The error metric that is being shown in TABLE IV is the average Hausdorff distance that was produced by each approach for the scenes that were being analyzed. It is abundantly clear that the proposed CEICP outperformed its competitors and was, for the most part, comparable

to the MRICP. The comparison between 3D TINs reconstructed from the point cloud, before and after CEICP fusion are illustrated in Figure 4.2 to Figure 4.7. The ball radii were varied from r_F to $r_F + \sigma_D$, where r_F and σ_D are mean and standard deviation of distance to a nearest neighbor in the fused point cloud.

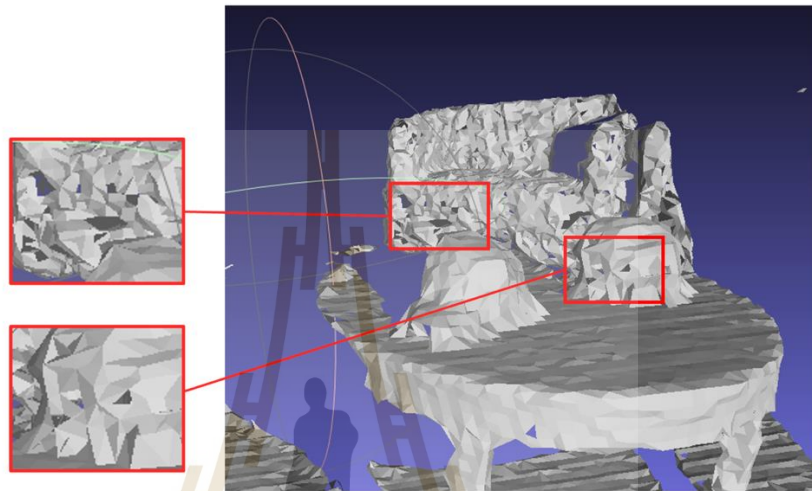


Figure 4.2 Result of BPA with $r=0.0102$ (Before fusion)

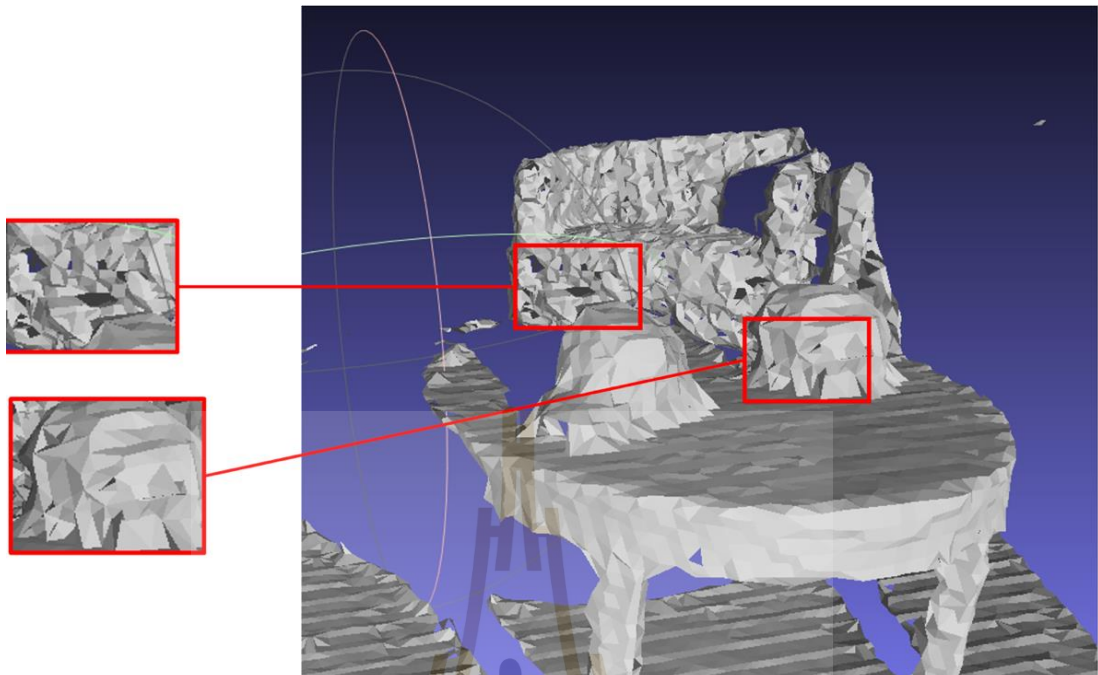


Figure 4.3 Result of BPA with $r=0.0131$ (Before fusion)

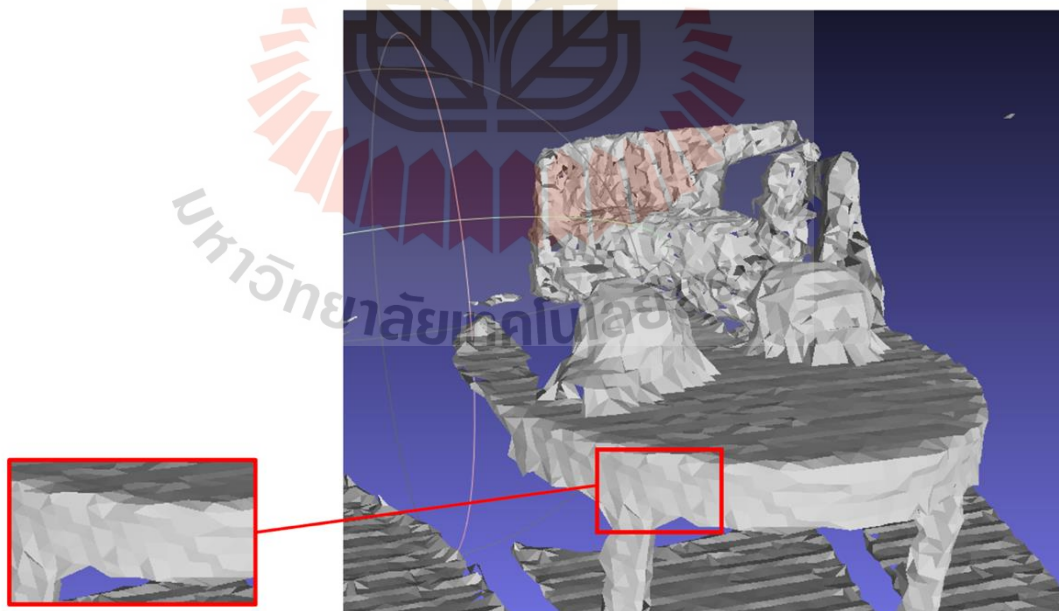


Figure 4.4 Result of BPA with $r=0.0156$ (Before fusion)

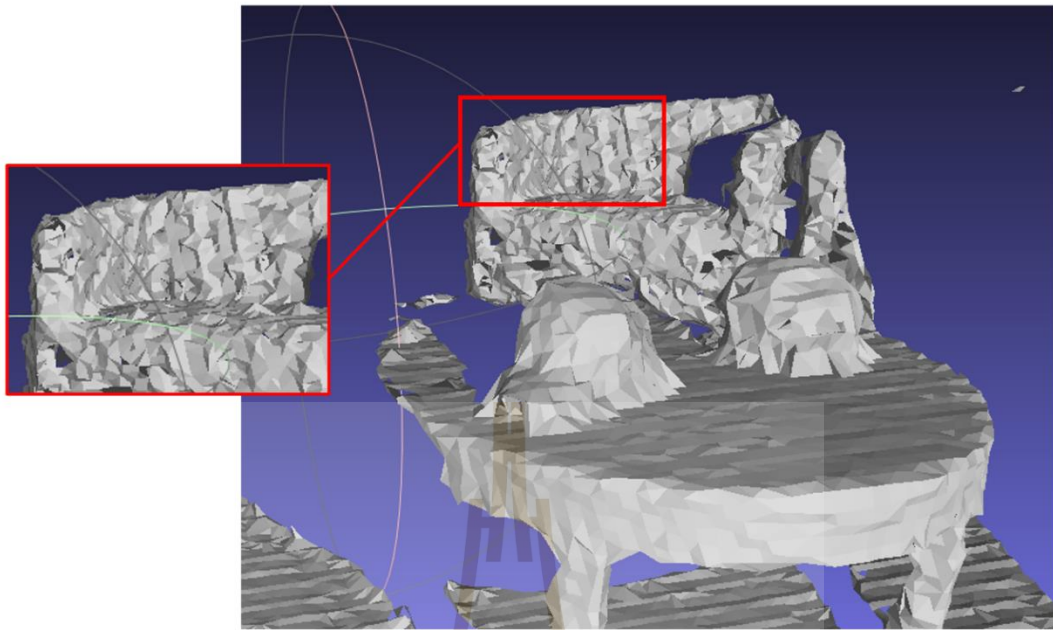


Figure 4.5 Result of BPA with $r=0.0209$ (Before fusion)

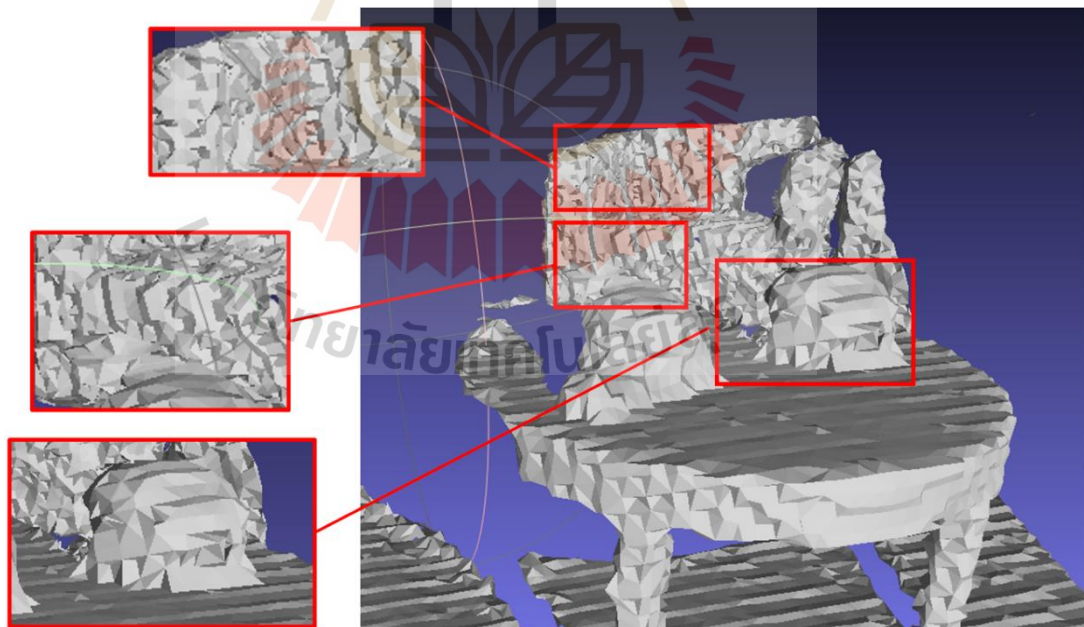


Figure 4.6 Result of BPA with $r=0.0102$ (After fusion)

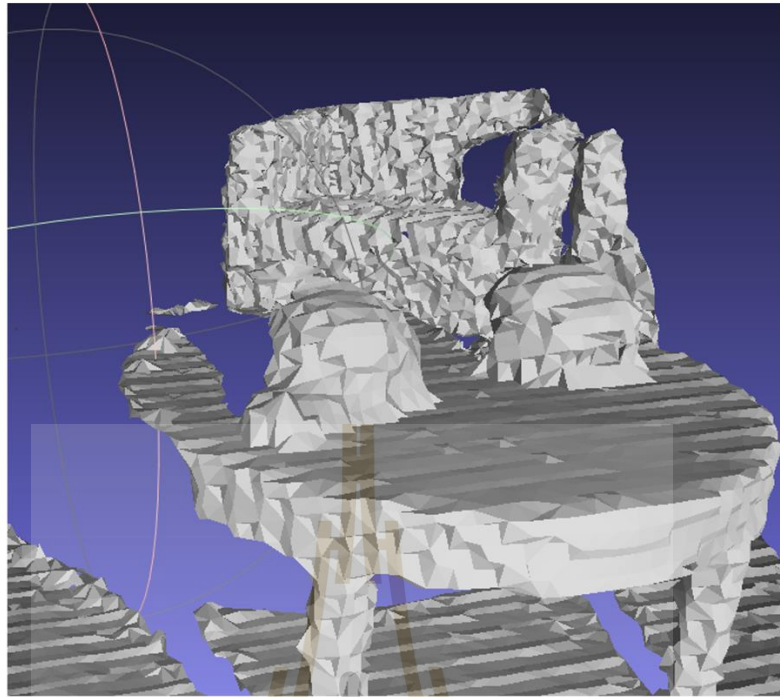


Figure 4.7 Result of BPA with $r=0.0156$ (After fusion)

Table 4.4 Point cloud fusion result comparison

	ICP	PICP	RICP	MRICP	FICP	HICP	Proposed
Scene1	8.21	97.25	19.91	4.91	6.51	9.54	4.361
Scene2	10.12	19.33	21.13	6.64	15.78	12.09	5.5484
Scene3	23.84	85.51	32.39	12.51	39.7	18.97	7.6874

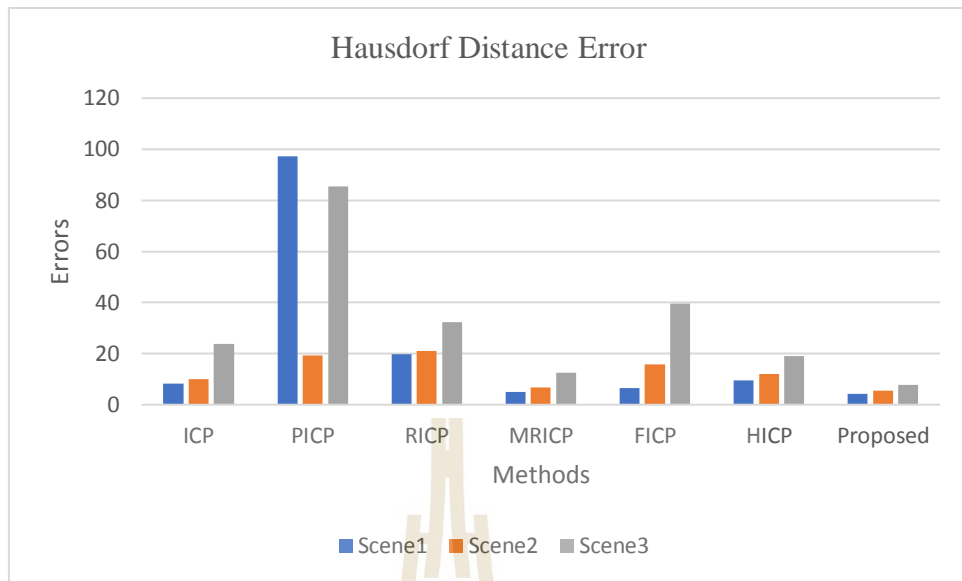


Figure 4.8 Visualization of result comparison of recent method and the proposed approach

In addition to this, the convergence of the proposed merger was evaluated against that of its counterparts, as shown in Figure 4.9. PICP and MRICP are plotted in this graph. PICP and MRICP respectively made use of hierarchical point selection and KD-tree search in order to expedite registration in comparison to a standard ICP.

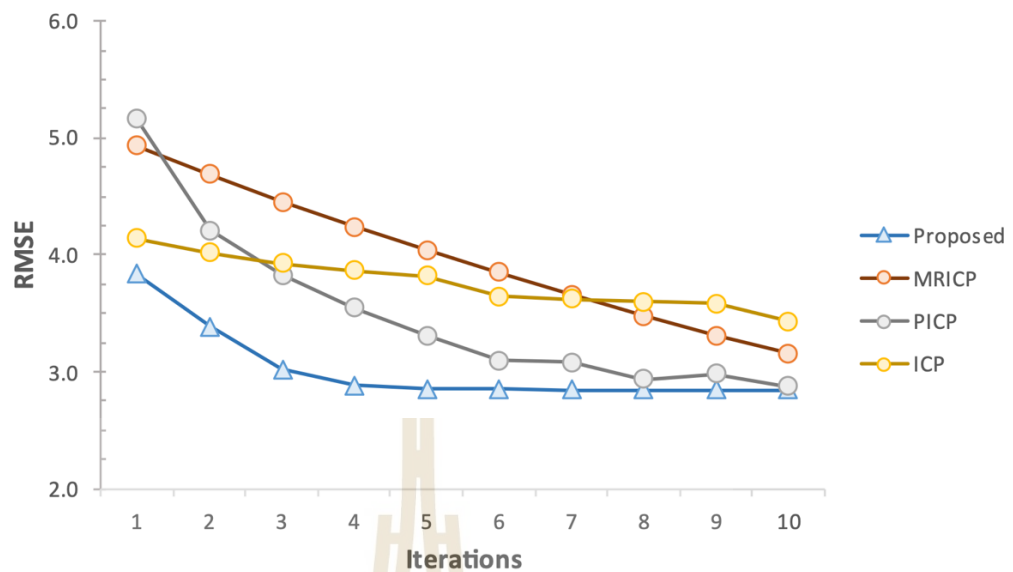


Figure 4.9 Convergence of the proposed approach comparing with the existing method.

Moreover, to demonstrate the experimental result from the depth camera, the extracted point cloud using the calibrated parameters are used to fused with inferred depth from the captured RGB image. The result in Table 4.5 show that the average errors of width, length, and height are 0.67, 0.66, and 0.69 respectively. It can conclude that the camera calibration process can increase the accuracy of reconstruction and fusion.

Table 4.5 Fusion result from different angle of camera placement using proposed approach.

	Width (cm)	Length (cm)	Height (cm)
Position 1 (0°)	0.63	0.59	0.65
Position 2 (45°)	0.57	0.61	0.62
Position 3 (90°)	0.81	0.78	0.82
Average	0.67	0.66	0.69

4.4 Wireless Visual Sensor Network Result

The experimental findings of image transmission from the node to the sink are presented in Table 4., which can be found below. According to the findings, the total quantity of photos that are transmitted has a considerable impact on the delivery time. This is because the size of the picture captured by the vision sensor is 901 kilobytes, the maximum speed of communication is 250 kbps, and each packet has a header that is that size.

To be more specific, we will suppose that each image that is taken by a camera has a resolution of 800 by 600 pixels and is compressed using the JPEG format into a file that is smaller than 1000 bytes in size. As a result, it is possible to transport a single image using a single packet of size 1000 bytes. In addition, we are operating under the assumption that the properties of a picture may be outlined using only a few tens of bytes (40 bytes). As a result, just a few hundred kilobytes are required to send data summaries for thousands of photos. The data gathering strategies discussed earlier have been included into our system as an application agent that operates on top of the UDP transmission protocol. This agent's duties include the creation of data summaries, the transmission of those Internet connected server housing the data collector, and the receipt of requests for images. In conclusion, we model the wireless communications using the following parameters: the MAC protocol is IEEE 802.11 without RTS/CTS handshaking, the physical transmission rate is 11 Mbps, the nominal radio range is set to 100 meters, and the two-ray ground propagation model is used. In the case when it is not specified differently, the simulated field typically has a total of four roadside APs. It is assumed that the wired cables that connect the roadside access points to the data collector have an unlimited bandwidth.

Table 4.6 Average delivery time against No. Of sent images

Number of images transmitted	Average time of delivering	
	Seconds	Minutes
1	138	2.30
2	336	5.60
3	489	8.25
4	605	10.18
5	819	13.67

Table 4.7 Parameter setting and Controlled simulation

Parameter	Value
image buffer size	200
camera sampling rate	2 s
validity time \mathcal{T}	10 s
observation period T	600 s
control period \mathcal{S}	60 s
r_L	3
r_H	8

The length of time for each simulation is set at five hours. However, in order to eliminate the possibility of temporary impacts, steady-state statistics aren't gathered until after the first 30 minutes have been subtracted. For the purpose of establishing confidence intervals with a 95% level of accuracy, we run each simulation five times.

As a last step in our investigation of the relative merits of various data collecting strategies, we assess how well these methods make use of the available bandwidth. Figure 4.10 illustrates this point by depicting the typical number of photos received by the data collector in a given minute in comparison to the total number of objects that are fitted with cameras. As was to be predicted, the higher the number of objects and the greater the number of increases received is due of the increased

frequency of connections. On the other hand, GREEDY is able to drastically cut down on the amount of data traffic while simultaneously increasing the amount of network coverage since it requests only photos that do not include redundant information. Intuitively, the k value and the number of received pictures are both reduced the more they are decreased. In a similar fashion, PDC cuts down on the amount of messages that are sent, however the level to which this reduction occurs may be contingent on the criteria that are employed in PDC to lower the likelihood of the system requesting fresh photos. In the end, we additionally investigate the protocol overheads caused by data summaries and picture queries, which are measured in terms of bytes per minute. In particular,

Figure 4.10 The average number of photos acquired per minute by the data collector using the BASELINE, GREEDY, and PDC schemes in comparison to the total number of objects (each image is conveyed in a 1000B-long packet). depicts the protocol overheads for the various iterations of GREEDY and PDC schemes that were examined earlier as a function of the number n of objects that are fitted with cameras. For the sake of clarity, we would like to remind you that an image tag is comprised of 40 bytes, and a data summary can include tags for a maximum of 200 images (i.e., the size of the local data storage), whereas the replies list the identifiers of the images that have been requested, and they typically consume a few hundreds of bytes at most. According to the findings, all of the strategies send around the same quantity of data summaries. This is due to the fact that the mobility profiles are the primary factor that determines this number. On the other hand, the signaling traffic that is caused by images requests diminishes as the value of k increases. This is due to the fact that this also restricts the maximum number of items that may be included in each request.

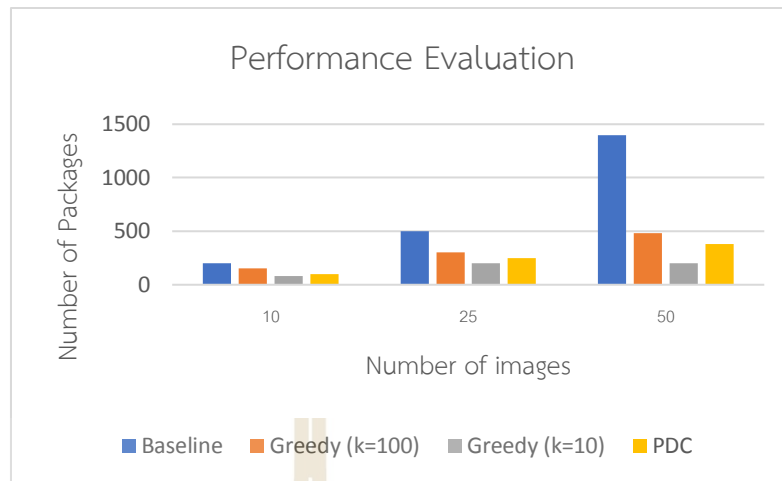


Figure 4.10 The average number of photos acquired per minute by the data collector using the BASELINE, GREEDY, and PDC schemes in comparison to the total number of objects (each image is conveyed in a 1000B-long packet).

CHAPTER 5

CONCLUSION AND DISCUSSION

5.1 Conclusion

Recently, three-dimensional scene reconstruction task is still challenge. Many up-to-date approaches needs improvement for increasing its accuracy and reducing computational time. Therefore, there are numbers of research works which aims to speed up and gain more accuracy. In this research, the framework for three-dimensional scene reconstruction on wireless visual sensor network using RGB-D camera is proposed. Firstly, the camera is calibrated using the calibration pattern for intrinsic and extrinsic camera parameter computing. Secondly, the set of point cloud is extracted from depth camera using the previous camera parameter. The RGB image is used to infer the point cloud based on deeply learn ResNet-50 model. Thirdly, both set of point cloud is fused together using modified iterative closest point algorithm, it's called Cross-Entropy Iterative Closest Point (CEICP). Fourthly, the Ball-Pivot Algorithm (BPA) is performed to reconstruct its surface. The process is simulated on wireless sensor network simulator.

5.2 Discussion and Future works

The outcomes that were described above can be explained in the following manner. A more realistic calibration might be the first step in resolving visible distortion, which is often shown in a reconstruction that is based on a straightforward pinhole camera model. If this is not the case, then straight lines in an image that has been collected (and subsequent objects that have been rebuilt) may seem deformed; this effect will be exacerbated the more distant they were from the lens center. Furthermore, missing data that was caused by imperfect lighting, noise, and outliers in the point cloud that was extracted from an infrared depth map were remedied by using a statistical filter and by fusing it with that estimated by deep learning from the

RGB image that was captured from the same scene. This was done in order to complete the data.

It was recommended that CEICP should register both point clouds before the fusion process began because they were received through distinct modalities. This would guarantee that the right correspondence was maintained. The results of the many evaluations showed that it was superior to the current state of the art. In instance, the suggested CEICP provided a correlation that was significantly more accurate while also having a quicker convergence rate. This is because in this particular instance, the degree of informational similarity was considerably more relevant than the geographic distance. However, because there were local minima in the entropy function, the two datasets had to start off having very few differences between them and being substantially aligned. This turned out to be the case in our environment, and we can credit the camera arrangement as well as the Deep Learning from the picture that corresponded to it. However, it is important to highlight that even after the fusion of the datasets, there were still some missing data as a result of coincident gaps in both of the datasets. It is common knowledge that reconstruction using only one of the modalities is inadequate, and this idea has been widely accepted. Just lately, another strategy that is analogous to ours was taken in mentioned research. They suggested an architecture for refining the disparity map, which merged monocular and stereo depth pictures. The former was pieced back together using a modified version of the VGG-16, which served as an auto encoder. The bilinear unification that was produced as a consequence was then improved using a minimal spanning tree (MST). In contrast to our approach, fusion was carried out, with the semantic prior of the scene serving as the basis. Because of this, empirical weights for known object classes and their distance are necessary, which reduces the generalizability of the method. As an additional alternative, the merging of point clouds directly with other suggestions was also taken into consideration. From the initial point cloud that was provided, first the point-wise features and then the voxel-wise features were extracted. The relevant proposals for voxel dense storage (VDS) and point sparse storage (PSS) have been combined. Second, proposal-aware fusion was used to combine the VDS and PSS proposals' semantic elements that had been derived from both proposals.

Finally, regression refinement was accomplished through the use of proposal classification and regression. Despite the fact that the fusion was performed directly on the point cloud, it was deduced from the information that was only available in a single modality. In addition, deep area of interest (ROI) fusion necessitated the tagging of objects within the image, which, in contrast to our approach, renders it inappropriate for use in scenarios involving surfaces that cannot be separated and the complexity of the proposed approach is need to be reduced.

In conclusion, the fused point cloud was utilized in conjunction with the BPA in order to recreate the scene's ultimate surface. It is clear that following CEICP fusion, there were far fewer apparent holes, when the ball radii were no less than $r_F + \sigma_D$, although those from a single cloud still presented strong indications of missing data. This was the case even if there were significantly less holes overall. It is important to keep in mind that even if a larger ball could eventually get rid of those flaws, it might also cause the loss of certain details in the process. Despite the fact that the trials were performed on a public dataset depicting an interior scenario for the purpose of benchmarking, there was no reduction in the capacity to generalize the results. Having said that, analytical insights may very well gain from that on far more complicated situations, or when using a moving camera, provided that their ground-truth measurements and the conclusions of their peers were accessible.

It is hoped that it will also be applicable to the reconstruction of sceneries obtained through remote sensing techniques, such as satellite images and aerial photography, as well as anatomical objects obtained through medical tomographic imaging. Another potential research path that should be taken into consideration is data fusion with other entities than points, such as voxel intensities and fiducial markers, and deep learning of these other entities. In addition, the treatments of their geometrical qualities, such as feature preserving mesh filtering approaches, inhomogeneous distribution, and sparse data collection, etc., have yet to be completely examined in the future. This to be done in the near future. Furthermore, the more complex shape of object i.e., engraved object, the archaeological site, the scene with different light effect which can be caused the accuracy will be investigated.

REFERENCES

- R. Laing, M. Leon and J. Isaacs, "Monuments Visualization: From 3D Scanned Data to a Holistic approach, an Application to the City of Aberdeen," 2015 19th International Conference on Information Visualisation, 2017, pp. 512-517
- Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., & Fitzgibbon, A. (2011, October). Kinectfusion: Real-time dense surface mapping and tracking. In 2011 10th IEEE international symposium on mixed and augmented reality (pp. 127-136). IEEE.
- Whelan, T., Kaess, M., Fallon, M., Johannsson, H., Leonard, J., & McDonald, J. (2012). Kintinuous: Spatially extended kinectfusion.
- TechTarget, T. (2011, March 25). Kinect. SearchHealthIT.[https:// searchhealthit.techtarget.com/definition/Kinect](https://searchhealthit.techtarget.com/definition/Kinect)
- Piltch, A. (2015, January 14). Intel RealSense 3D: What It Is and What You Do With It. Tom's Guide. <https://www.tomsguide.com/us/intel-realsense-guide,news-20286.html>
- Wikipedia contributors. (2021, November 4). Raspberry Pi. Wikipedia. [https:// en.wikipedia.org/ wiki/ Raspberry_Pi](https://en.wikipedia.org/wiki/Raspberry_Pi)
- Wikipedia contributors. (2021, June 2). Laser diode. In Wikipedia, The Free Encyclopedia. Retrieved 04:24, June 14, 2021, from [https://en.wikipedia.org/ w/ index.php?title=Laser_diode&oldid=1026457016](https://en.wikipedia.org/w/index.php?title=Laser_diode&oldid=1026457016)
- Kikuta, H., Iwata, K., & Nagata, R. (1986). Distance measurement by the wavelength shift of laser diode light. *Applied optics*, 25(17), 2976-2980.
- Rao, S. M., Heitz, J. J., Roger, T., Westerberg, N., & Faccio, D. (2014). Coherent control of light interaction with graphene. *Optics letters*, 39(18), 5345-5347.
- Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11), 1330-1334. Camera Calibration. Camera Calibration - MATLAB & Simulink. (n.d.). [https:// www.mathworks.com/ help / vision/camera-calibration.html](https://www.mathworks.com/help/vision/camera-calibration.html).

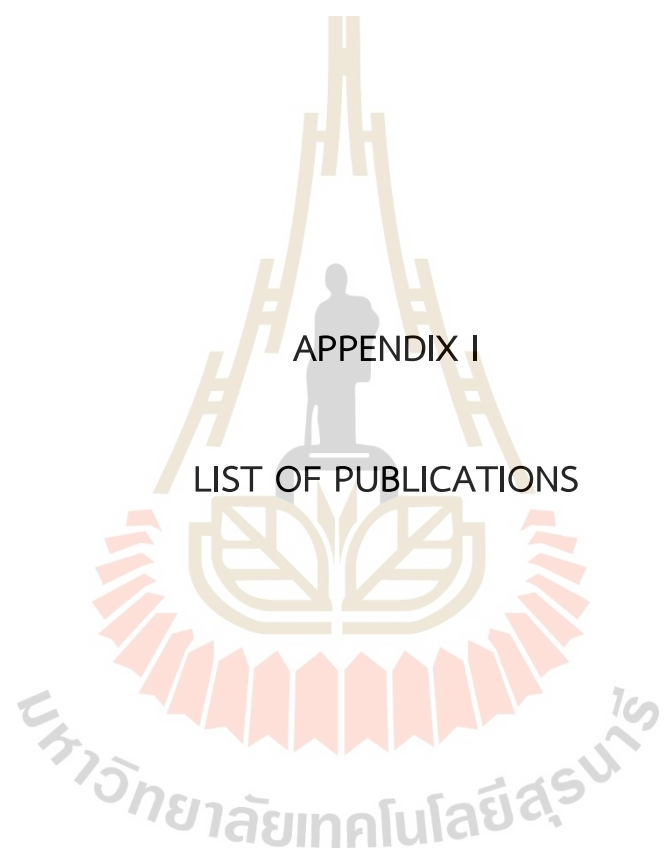
- He, L., Chao, Y., Suzuki, K., & Wu, K. (2009). Fast connected-component labeling. *Pattern recognition*, 42(9), 1977-1987.
- Chatzis, V., & Pitas, I. (2000). Interpolation of 3-D binary images based on morphological skeletonization. *IEEE transactions on medical imaging*, 19(7), 699-710.
- Mudrova, M., & Procházka, A. (2005, November). Principal component analysis in image processing. In *Proceedings of the MATLAB technical computing conference, Prague*.
- Wikipedia contributors. (2021, June 5). Polygon mesh. In *Wikipedia, The Free Encyclopedia*. Retrieved 05:10, June 14, 2021, from https://en.wikipedia.org/w/index.php?title=Polygon_mesh&oldid=1027043893
- Hold-Geoffroy, Y., Sunkavalli, K., Eisenmann, J., Fisher, M., Gambaretto, E., Hadap, S., & Lalonde, J. F. (2018). A perceptual measure for deep single image camera calibration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2354-2363).
- J. -H. Chuang, C. -H. Ho, A. Umam, H. -Y. Chen, J. -N. Hwang and T. -A. Chen, "Geometry-Based Camera Calibration Using Closed-Form Solution of Principal Line," in *IEEE Transactions on Image Processing*, vol. 30, pp. 2599-2610, 2021, doi: 10.1109/TIP.2020.3048684.
- Gai, S., Da, F., & Dai, X. (2018). A novel dual-camera calibration method for 3D optical measurement. *Optics and Lasers in Engineering*, 104, 126-134.
- Reyes, A. L., Cervantes, J. M., & Gutiérrez, N. C. (2013). Low cost 3D scanner by means of a 1D optical distance sensor. *Procedia Technology*, 7, 223-230.
- Zhang, Z., & Yuan, L. (2012). Building a 3D scanner system based on monocular vision. *Applied optics*, 51(11), 1638-1644.
- Rocchini, C. M. P. P. C., Cignoni, P., Montani, C., Pingi, P., & Scopigno, R. (2001, September). A low cost 3D scanner based on structured light. In *Computer Graphics Forum* (Vol. 20, No. 3, pp. 299-308). Oxford, UK and Boston, USA: Blackwell Publishers.
- Zhang, C. (2021). PL-GM: RGB-D SLAM With a Novel 2D and 3D Geometric Constraint Model of Point and Line Features. *IEEE Access*, 9, 9958-9971.
- Lhuillier, M. (2018). Surface reconstruction from a sparse point cloud by enforcing visibility consistency and topology constraints. *Computer Vision and Image Understanding*, 52-71.

- Sheng, B., Zhao, F., Yin, X., Zhang, C., Wang, H., & Huang, P. (2018). A lightweight surface reconstruction method for online 3D scanning point cloud data oriented toward 3D printing. *Mathematical Problems in Engineering*, 2018.
- Mineo, C., Pierce, S. G., & Summan, R. (2019). Novel algorithms for 3D surface point cloud boundary detection and edge reconstruction. *Journal of Computational Design and Engineering*, 6(1), 81-91.
- R. J. Wilson and S. Chiang, "Image processing techniques for obtaining registration information with scanning tunneling microscopy," *J. Vac. Sci. Technol. A, Vac., Surf., Films*, vol. 6, no. 2, pp. 398-400, Mar. 1988.
- S. Lee, P. Horkaew, W. Caspersz, A. Darzi, and G.Z. Yang, "Assessment of shape variation of the levator ani with optimal scan planning and statistical shape modeling," *Journal of Computer Assisted Tomography*, vol. 29, no. 2, pp. 154-162, 2005.
- D.C. Le, J. Chansangrat, N. Keeratibharat, and P. Horkaew, "Symmetric reconstruction of functional liver segments and cross-individual correspondence of hepatectomy," *Diagnostics*, vol. 11, no. 5., 852, 2021.
- Florinsky, V. Igor. "Digital Terrain Analysis in Soil Science and Geology", *Digital Elevation Models*, pp. 31-41. 2003.
- C. V. Nguyen, S. Izadi and D. Lovell, "Modeling kinect sensor noise for improved 3d reconstruction and tracking". In 2012 second international conference on 3D imaging, modeling, processing, visualization & transmission, pp. 524-530. Oct, 2013.
- J. Zhang, and X. Lin, "Advances in fusion of optical imagery and LiDAR point cloud applied to photogrammetry and remote sensing", *International Journal of Image and Data Fusion*, 8(1), pp. 1-31. 2017.
- P. Liang, Z. Fang, B. Huang, H. Zhou, X. Tang, and C. Zhong, "PointFusionNet: Point feature fusion network for 3D point clouds analysis", *Applied Intelligence*, 51(4), pp. 2063-2076, 2021.
- Z. Wang, Y. Xu, Q. He, Z. Fang, G. Xu, and J. Fu, "Grasping pose estimation for SCARA robot based on deep learning of point cloud" , *The International Journal of Advanced Manufacturing Technology*, 108(4), pp. 1217-1231, 2020.

- J. Ying, and X. Zhao, "Rgb-D Fusion For Point-Cloud-Based 3d Human Pose Estimation" In 2021 IEEE International Conference on Image Processing (ICIP), pp. 3108-3112, Sep, 2021.
- C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "Densefusion: 6d object pose estimation by iterative dense fusion", In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 3343-3352, 2019.
- S. Vosselman, "Fusion of laser scanning data, maps, and aerial photographs for building reconstruction", In IEEE International Geoscience and Remote Sensing Symposium, Vol. 1, pp. 85-88. Jun, 2002.
- C. H Lin, C. Kong, and S. Lucey, (2018, April). Learning efficient point cloud generation for dense 3d object reconstruction. In proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, No. 1.
- Z. Ouyang, Y. Liu, C. Zhang, and J. Niu, (2017, December). A cgans-based scene reconstruction model using lidar point cloud. In 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC), pp. 1107-1114.
- R. Scona, M. Jaimez, Y. R. Petillot, M. Fallon, and D. Cremers, "Staticfusion: Background reconstruction for dense rgb-d slam in dynamic environments", In 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 3849-3856. May, 2018.
- K. Tateno, F. Tombari, I. Laina, and N. Navab, "Cnn-slam: Real-time dense monocular slam with learned depth prediction", In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 6243-6252, 2017
- B. Li, Y. Wang, Y. Zhang, W. Zhao, J. Ruan, and P. Li, "GP-SLAM: laser-based SLAM approach based on regionalized Gaussian process map reconstruction", *Autonomous Robots*, 44(6), pp. 947-967, 2020.
- Q. Y. Zhou, J. Park, and V. Koltun, Fast global registration. In European conference on computer vision, pp. 766-782, Oct, 2016.

- R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration", In 2009 IEEE international conference on robotics and automation, pp. 3212-3217, May, 2009.
- S. Gold, A. Rangarajan, C. P. Lu, S. Pappu, and E. Mjolsness, "New algorithms for 2D and 3D point matching: pose estimation and correspondence", Pattern recognition, 31(8), pp. 1019-1031, 1998.
- H. Chui, and A. Rangarajan, "A new point matching algorithm for non-rigid registration", Computer Vision and Image Understanding, 89(2-3), pp. 114-141, 2003.
- Y. Tsin, and T. Kanade, "A correlation-based approach to robust point set registration", In European conference on computer vision, pp. 558-569. May, 2004.
- X. Lu, S. Wu, H. Chen, S. K. Yeung, W. Chen, and M. Zwicker, "GPF: GMM-inspired feature-preserving point set filtering", IEEE transactions on visualization and computer graphics, 24(8), pp. 2315-2326, 2017.
- O. Hirose, "A Bayesian formulation of coherent point drift" IEEE transactions on pattern analysis and machine intelligence, 43(7), pp. 2269-2286, 2020.
- C. Kim, H. Son, and C. Kim, "Fully automated registration of 3D data to a 3D CAD model for project progress monitoring", Automation in Construction, 35, pp. 587-594, 2013.
- F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," IEEE Transactions on Visualization and Computer Graphics, vol. 5, no. 4, pp. 349-359, Oct, 1999.
- H. Seo, T. Kin, and T. Igarashi, "A Mesh-Aware Ball-Pivoting Algorithm for Generating the Virtual Arachnoid Mater," in Proceeding of Medical Image Computing and Computer Assisted Intervention (MICCAI 2019), Lecture Notes in Computer Science, vol. 11768, pp. 592-600.
- X. Guo, J. Xiao, and Y. Wang, "A survey on algorithms of hole filling in 3D surface reconstruction." Vis Comput, vol. 34, pp. 93-103, 2018.
- I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari and N. Navab, "Deeper Depth Prediction with Fully Convolutional Residual Networks," presented at 2016 Fourth International Conference on 3D Vision (3DV), pp. 239-248, 2016.

- P. Horkaew, G.Z. Yang, "Construction of 3D dynamic statistical deformable models for complex topological shapes," in *Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, pp. 217–224, 2004.
- W. Ma and Q Li, "An Improved Ball Pivot Algorithm-Based Ground Filtering Mechanism for LiDAR Data," *Remote Sensing*, vol. 11, no. 10, 1179, 2019.
- S. Gai, F. Da, X. Dai, "A novel dual-camera calibration method for 3 D optical measurement", *Optics and Lasers in Engineering*, 104, pp. 126-134, 2018.
- J. Digne and C. Franchis, "The Bilateral Filter for Point Clouds," *Image Processing On Line*, vol. 7, pp. 278–287, 2017.
- C. Godard, O.M. Aodha, M. Firman and G.J. Brostow, "Digging Into Self-Supervised Monocular Depth Estimation," presented at 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 3827-3837, Feb. 2020.
- A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset". *The International Journal of Robotics Research*, 32(11), pp. 1231-1237, 2013.
- S. Kullback, "Information Theory and Statistics," New York, NY, USA, Dover Publications, 1968.
- Lotem Nadir, "Ball-Pivoting Algorithm [Online]. Available: <https://github.com/Lotemn102/Ball-Pivoting-Algorithm>, Accessed on: May 10, 2022.
- T. Zinßer, J. Schmidt, and H. Niemann, "A refined ICP algorithm for robust 3 - D correspondence estimation", In *Proceedings 2003 international conference on image processing*, pp. 695, Sep, 2003.
- E. Trucco, A. Fusiello, and V. Roberto, "Robust motion and correspondence of noisy 3-D point sets with missing data", *Pattern recognition letters*, 20(9), 889-898, 1999
- T. Jost, and H. Hugli, "A multi-resolution ICP with heuristic closest point search for fast and robust 3D registration of range images" In *Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings*, pp. 427-433, Oct, 2003.
- J. M. Phillips, R. Liu, and C. Tomas, "Outlier robust ICP for minimizing fractional RMSD". In *Sixth International Conference on 3-D Digital Imaging and Modeling (3DIM 2007)*, pp. 427-434. Oct, 2010.
- H. Men, B. Gebre, and K. Pochiraju, "Color point cloud registration with 4D ICP algorithm". In *2011 IEEE International Conference on Robotics and Automation*, pp. 1511-1516, May 2010.



APPENDIX I

LIST OF PUBLICATIONS

List of Publications

W. Yookwan, K. Chinnasarn, C. So-In and P. Horkaew, "Multimodal Fusion of Deeply Inferred Point Clouds for 3D Scene Reconstruction using Cross-Entropy ICP," in IEEE Access, 2022, doi: 10.1109/ACCESS.72022.3192869.



BIOGRAPHY

Watcharaphong Yookwan Received The B.Sc. Degree In Computer Science And The M.Sc. Degree In Informatics From Burapha University, Chon Buri, Thailand, In 2016 And 2018, Respectively. He Is Currently Pursuing The Ph.D. Degree In Computer Engineering With The Suranaree University Of Technology, Nakhon Ratchasima, Thailand. His Research Interests Include Image Processing And Digital Geometry Processing.

