

<Typ der Arbeit>

<Name der Arbeit>

Li Yong, Teng

Prüfer:

Prof. Dr.-Ing. <Mustermann>

Prof. Dr.-Ing. <Mustermann>

Betreuer:

M. Sc. <Mustermann>

Institut für Produktionsmanagement und -technik

Technische Universität Hamburg-Harburg

Hamburg, den 22. January 2021

# Scope of work

<Kopie der Aufgabenstellung>

# Declaration of Authorship

I declare that the work presented here is, to the best of my knowledge and belief, original and the result of my own investigations, except as acknowledged, and has not been submitted, either in part or whole, for a degree at this or any other University. Formulations and ideas taken from other sources are cited as such. This work has not been published.

Hamburg, den <Datum einfügen> <Autor einfügen>

# Table of Content

[Scope of work II](#_Toc62236647)

[Declaration of Authorship III](#_Toc62236648)

[Table of Content IV](#_Toc62236649)

[List of Figures VII](#_Toc62236650)

[List of Abbreviations X](#_Toc62236651)

[List of Formula Symbols XI](#_Toc62236652)

[1 Introduction 1](#_Toc62236653)

[1.1 Einführung 1](#_Toc62236654)

[1.2 Aufbau der Arbeit 1](#_Toc62236655)

[2 State of The Art 2](#_Toc62236656)

[2.1 Augmented Reality 2](#_Toc62236657)

[2.1.1 Tracking in Augmented Reality 3](#_Toc62236658)

[2.1.2 Marker based Tracking 4](#_Toc62236659)

[2.2 Construction Progress in Augmented Reality 5](#_Toc62236660)

[2.2.1 Time Efficiency 5](#_Toc62236661)

[2.2.2 Data quality and accuracy 6](#_Toc62236662)

[2.2.3 Completion progress 7](#_Toc62236663)

[2.3 Point Cloud Capturing 9](#_Toc62236664)

[2.3.1 Light Detection and Ranging (LiDAR) 10](#_Toc62236665)

[2.3.2 Photogrammetry 11](#_Toc62236666)

[2.4 Point Cloud Representation 11](#_Toc62236667)

[2.4.1 Meshing 12](#_Toc62236668)

[2.4.2 Polygonization 12](#_Toc62236669)

[2.4.3 Voxelization 13](#_Toc62236670)

[2.5 Noise Filterring and Outliers Removal 14](#_Toc62236671)

[2.5.1 Neighborhood-based methods 14](#_Toc62236672)

[2.5.2 Principal Component Analysis 15](#_Toc62236673)

[2.5.3 Filter point cloud data from RGB information 16](#_Toc62236674)

[2.6 Performance Evaluation Method 17](#_Toc62236675)

[2.6.1 Frame based Metrics 17](#_Toc62236676)

[2.6.2 Object based metrics 18](#_Toc62236677)

[2.7 Resources used for the application 19](#_Toc62236678)

[2.7.1 Game Engine: Unity 19](#_Toc62236679)

[2.7.2 Tracking Framework: AR Foundation 20](#_Toc62236680)

[2.7.3 Analyzing Tool: Open3D 21](#_Toc62236681)

[3 Analysis 22](#_Toc62236682)

[3.1 Analysis of Construction Progress 22](#_Toc62236683)

[3.1.1 Ground Truth Knowledge 22](#_Toc62236684)

[3.1.2 Differentiation of Assembly Parts 24](#_Toc62236685)

[3.1.3 Assembly of Object Part 25](#_Toc62236686)

[3.2 Environmental Condition 26](#_Toc62236687)

[3.2.1 Lighting Condition 27](#_Toc62236688)

[3.2.2 Range of Experiment Environment 28](#_Toc62236689)

[3.3 Framework and Tools Analysis 28](#_Toc62236690)

[3.3.1 Augmented Reality Framework 29](#_Toc62236691)

[3.3.2 LiDAR Sensor 29](#_Toc62236692)

[4 Concept 31](#_Toc62236693)

[4.1 System Design 31](#_Toc62236694)

[4.1.1 Object Tracking Calibration 33](#_Toc62236695)

[4.1.2 Point Cloud Collection Method 34](#_Toc62236696)

[4.1.3 Mesh Collection Method 37](#_Toc62236697)

[4.1.4 Fitness Level 41](#_Toc62236698)

[4.1.5 Parts Combination 42](#_Toc62236699)

[4.2 Design of Experiment 42](#_Toc62236700)

[4.2.1 Percentage Threshold 42](#_Toc62236701)

[4.2.2 Design of Experiment Method 44](#_Toc62236702)

[5 46](#_Toc62236703)

[5.1 Criteria of Indicator Construction Progress 46](#_Toc62236704)

[5.1.1 Accuracy Graph 46](#_Toc62236705)

[6 Implementation 47](#_Toc62236706)

[6.1 Image Tracking Calibration 47](#_Toc62236707)

[6.1.1 AR Tracked Image Manager 47](#_Toc62236708)

[6.1.2 Reference Image Library 48](#_Toc62236709)

[6.1.3 CAD Model Position and Orientation 49](#_Toc62236710)

[6.1.4 AR Point Cloud Manager 49](#_Toc62236711)

[6.2 Analysis of data accuracy 49](#_Toc62236712)

[Bibliography 50](#_Toc62236713)

[Anhang 54](#_Toc62236714)

# List of Figures

[Figure 2.1: Representation of the Reality-Virtuality continuum. [Milgram et al., 1995] 2](#_Toc62236715)

[Figure 2.2: Illustration of 3 groups of Augmented Reality devices. [Schmalstieg & Höllerer, 2016] 3](#_Toc62236716)

[Figure 2.3: Tracking methods for augmented reality. [Bostanci et al., 2013] 3](#_Toc62236717)

[Figure 2.4: Marker for tracking purpose 4](#_Toc62236718)

[Figure 2.5: Augmented-Reality based progress monitoring. [Lee & Pena-Mora, 2006] 6](#_Toc62236719)

[Figure 2.6: An example of existing progress reporting techniques. Construction drawings and work schedules are hung on a construction site trailer’s wall to communicate progress with contractors and subcontractors. [Mani et al., 2009] 6](#_Toc62236720)

[Figure 2.7: A sample of a real project progress/inspection report. [Mani et al., 2009] 7](#_Toc62236721)

[Figure 2.8: Progress Deviation Visualize. [Lee & Pena-Mora, 2006] 8](#_Toc62236722)

[Figure 2.9: System process and work flow. [Fard & Peña-Mora, 2007] 8](#_Toc62236723)

[Figure 2.10: Point Cloud dataset of Enschede. [Stucker, 2017] 9](#_Toc62236724)

[Figure 2.11: Point cloud captured of a building. [Hinks et al., 2013] 9](#_Toc62236725)

[Figure 2.12: Basic structure of LiDAR. [Yoo et al., 2018] 10](#_Toc62236726)

[Figure 2.13: iPad camera with LiDAR scanner. (Picture acquired from Apple site) 10](#_Toc62236727)

[Figure 2.14: Point clouds based on feature points of object. 11](#_Toc62236728)

[Figure 2.15: (a) point cloud, (b) control mesh.[Yoon, 2006] 12](#_Toc62236729)

[Figure 2.16: Polygonization of point clouds. [Hidaka, 2016] 13](#_Toc62236730)

[Figure 2.17: Point clouds and binary voxel representation. [Gokberk et al., 2008] 13](#_Toc62236731)

[Figure 2.18: Workflow of FEM model reconstruction. [Hinks et al., 2013] 14](#_Toc62236732)

[Figure 2.19: (Left) Sparse outliers (Right) Point clouds data after outliers removal. [Rusu et al., 2008] 15](#_Toc62236733)

[Figure 2.20: Filtering Process of PCD based on RGB image. [C. Jia et al., 2019] 16](#_Toc62236734)

[Figure 2.21: Process from RGB image, maping, original PCD and filtered PCD. [C. Jia et al., 2019] 16](#_Toc62236735)

[Figure 2.22: Type I and Type II error illustration. 17](#_Toc62236736)

[Figure 2.23: Definition of core metric for object-based metric. [Bashir & Porikli, 2006] 18](#_Toc62236737)

[Figure 2.24: Unity XR platform structure. [Unity, 2019] 19](#_Toc62236738)

[Figure 2.25: AR Foundation chart. 20](#_Toc62236739)

[Figure 2.26: 3D data processing: load a point cloud, downsample it, and estimate normals. [Q.-Y. Zhou et al., 2018] 21](#_Toc62236740)

[Figure 2.27: Voxelization from a point cloud dataset in Open3D 21](#_Toc62236741)

[Figure 3.1: CAD model for ground truth knowledge. 23](#_Toc62236742)

[Figure 3.2: Assembly Parts of The Object 25](#_Toc62236743)

[Figure 3.3: Partially Assemble and Fully Assemble Middle Cover 26](#_Toc62236744)

[Figure 3.4: Low-light environment and Light-sufficient environment 27](#_Toc62236745)

[Figure 3.5: Workspace of the AR application 28](#_Toc62236746)

[Figure 3.6: Mesh Data from LiDAR Sensor from Apple. 30](#_Toc62236747)

[Figure 4.1: General illustration of AR application 32](#_Toc62236748)

[Figure 4.2: Virtual object superimpose to real-world object. 32](#_Toc62236749)

[Figure 4.3: Object Tracking Calibration 34](#_Toc62236750)

[Figure 4.4: Point cloud capturing in AR application 34](#_Toc62236751)

[Figure 4.5: Process Flow of Point Cloud Capturing Method. 35](#_Toc62236752)

[Figure 4.6: Point Cloud Capturing Illustration 36](#_Toc62236753)

[Figure 4.7: Flow Chart of the Point Cloud Capturing Algorithm 37](#_Toc62236754)

[Figure 4.8: Capturing mesh data in AR application 38](#_Toc62236755)

[Figure 4.9: Mesh Data Capturing illustration 38](#_Toc62236756)

[Figure 4.10: Flow Chart of mesh data collection method. 39](#_Toc62236757)

[Figure 4.11: Flow Chart of Mesh Data Collecting Algorithm 40](#_Toc62236758)

[Figure 4.12: General Flow Chart of Both Methods 41](#_Toc62236759)

[Figure 4.13: 2D Assembly Model 41](#_Toc62236760)

[Figure 4.14: Combination of Assembly Part. (GearShaft1) 42](#_Toc62236761)

[Figure 6.1: Outline of the structure for AR application. 47](#_Toc62236762)

[Figure 6.2: Reference Image Library for Marker 48](#_Toc62236763)

[Figure 6.3: CAD Model superimposing real-world object 48](#_Toc62236764)

List of Tables

[Table 2.1: Metrics to be computed based on defined quantities 18](#_Toc62236765)

[Table 2.2: Description of each AR Foundation, ARCore, and ARKit. 20](#_Toc62236766)

[Table 3.1: Part name of the CAD Model 23](#_Toc62236767)

[Table 3.2: Part Number with Bound Magnitude and Threshold Percentage 24](#_Toc62236768)

[Table 4.1: Percentage Threshold Value 43](#_Toc62236769)

[Table 5.1: Template of data collection 44](#_Toc62236770)

[Table 5.2: Allocation of TP, TN, FP and FN 44](#_Toc62236771)

[Table 6.1: Physical measurement of marker 48](#_Toc62236772)

[Table 6.2: Transformation adjustment of the Virtual Object in Scene 49](#_Toc62236773)

# List of Abbreviations

**Abbreviation Description**

AR Augmented Reality

VR Virtual Reality

SLAM Simultaneous Localization and Mapping

# List of Formula Symbols

# Introduction

Diese Dokumentvorlage wird vom Institut für Produktionsmanagement und -technik bereitgestellt. Sie ist nicht verbindlich.

## Einführung

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi. Nam eget dui. Etiam rhoncus. Maecenas tempus, tellus eget condimentum rhoncus, sem quam semper libero, sit amet adipiscing sem neque sed ipsum. Nam quam nunc, blandit vel, luctus pulvinar, hendrerit id, lorem. Maecenas nec odio et ante tincidunt tempus. Donec vitae sapien ut libero venenatis faucibus. Nullam quis ante.

## Aufbau der Arbeit

Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi. Nam eget dui. Etiam rhoncus. Maecenas tempus, tellus eget condimentum rhoncus, sem quam semper libero, sit amet adipiscing sem neque sed ipsum. Nam quam nunc, blandit vel, luctus pulvinar, hendrerit id, lorem. Maecenas nec odio et ante tincidunt tempus. Donec vitae sapien ut libero venenatis faucibus. Nullam quis ante. Etiam sit amet orci eget eros faucibus tincidunt. Duis leo. Sed fringilla mauris sit amet nibh.

# State of The Art

This chapter discusses the literature review of the thesis work for construction progress in augmented reality, theoretical framework for point cloud capturing, point cloud representation, noise filtering, and point cloud mapping.The theoretical framework covers the required essential knowledge to support the research and it is complemented with an extension of recent studies of related works. The chapter also concludes with the description of available performance evaluation method and resources to be implemented in the thesis work.

## Augmented Reality

Augmented reality, AR, is defined as a virtual object in a real environment, also provide local virtuality. [Van Krevelen & Poelman, 2010] According to [Milgram et al., 1995], AR is defined as a class on the reality-virtuality continuum, which is a real environment in one extreme end and a complete pure virtual environment in the other opposite, where everything in between is called a mixed reality. Figure 2.1 shows the illustration of the explanation. With that being said, Augmented Reality belongs to the mixed reality group which consist of the mix of both dimensions. As Augmented Reality merge real world with virtual objects, there is another group which called Virtual Reality which is fully allowing users to emerge in a virtual environment.

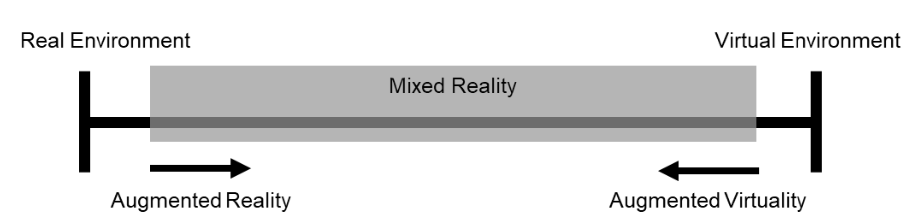


Figure 2.1: Representation of the Reality-Virtuality continuum. [Milgram et al., 1995]

Augmented Reality devices can be divided into several groups, head-mounted display, handheld display and spatial display. Head-mounted display allows users to wear the device on the head, where user can still be able to see the real-world environment alongside with virtual object through the display monitor of the device. Handheld displays employ small computing device which users can hold the device and access augmented reality world in the palm of their hands, a good example will be our daily use smartphones or tablets. Spatial display usually being used with projector based where users would not need to hold or wear the device in order to experience the augmented reality scene. Figure 2.2 shows the illustration of these three group of Augmented Reality devices.

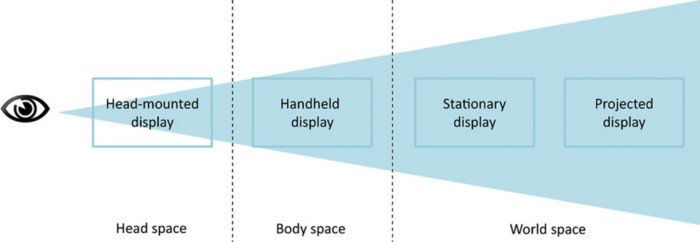


Figure 2.2: Illustration of 3 groups of Augmented Reality devices. [Schmalstieg & Höllerer, 2016]

### Tracking in Augmented Reality

Augmented reality tracking devices must have a higher accuracy for greater tracking, the registration accuracy depends on the distance of object of interest besides geometrical model [Van Krevelen & Poelman, 2010]. Indoor environment usually provides an easier tracking than outdoor environment as the setting in indoor environment is easily modelled and prepare, the lighting efficiency can be easily controlled. There are variety of tracking methods for different applications, Figure 2.3 show the categories of tracking in augmented reality, which includes indoor methods, outdoor methods, and fusion method [Bostanci et al., 2013].

Diagram

Description automatically generated

Figure 2.3: Tracking methods for augmented reality. [Bostanci et al., 2013]

According to [Behzadan et al., 2008], it is stated that for indoor application, the dimension of the environment are fixed and the physical movement of users are restricted and therefore movements are more predictable. There are many different types of indoor tracking methods, classic SLAM techniques [Klein & Murray, 2007] and vision-based tracking which include tracking using markers and object of interest.

On the other hand, outdoor environments are usually limitless in terms of orientation and position of the device. However, the lightning will sometimes bring a problem for camera tracking which it is not an issue for indoor environment. As shown in the figure, GPS is considered a reliable tracking option when the device is at outdoor environment. Inertial-based tracking is also made possible since the position of device can be larger and this work similar to our human ear, but instead the device uses accelerometers and gyros. All of the sensor tracking technologies are well develop and they have all advantage and disadvantages. Based on the research paper of [F. Zhou et al., 2008], a given example is that magnetic sensors will have a high update rate and the weight is very light, but they can be distorted by nearby metallic substance where it may vary the results of tracking. With all of the tracking methods, it makes possible and efficient for development of Augmented Reality application.

### Marker based Tracking

Marker based tracking is one of the visual tracking which attempts to track the head position by analyzing features detected in a live video stream based on a marker anchor. The device is able to calculate the camera position in relation to the marker features seen in the real-world environment. This has become a very low-cost sensor for Augmented Reality registration [Klein, 2006]. These markers have geometric or properties of color which make it easier to be read and identify in a video frame of an Augmented Reality device. Figure 2.4 shows an example of a marker that being used in this thesis work. Marker tracking is robust due to the constant update of relative position of camera and the marker position in each video frame of the device.



Figure 2.4: Marker for tracking purpose

## Construction Progress in Augmented Reality

Progress monitoring is important for a project management team to track the construction progress of the overall project in order to keep every detail of the planned progress in schedule. Early detection of any schedule delay or cost overrun in the field construction activities. [Mani et al., 2009] Currently, the construction projects is time consuming and labor intensive, which consist of discrepancies like charts, graphs and still photos which did not facilitate the communication and cause some distraction to decisions maker from performing a right and accurate decision. There is another way of representing the construction progress is the use of augmented reality, which means to put a virtual object into an immersive environment on a digital device. With the help of augmented reality, time which is a constraint will be overcome by using this technology. This technology could help in improving the project management in many other aspects like data accuracy, data analysis and data quality maintenance. The application of merging virtual and real environment can help in many field, there are proposed evaluation [Kamat & El-Tawil, 2005] of post-disaster buildings using augmented reality, which the application is able to provide real time feedback from the actual building view, where analysis can be detailed record to explore how a building might collapse if some structure is broken. This sub-chapter will discuss about the key performance index of construction progress and how augmented reality helps in improving the monitoring process.

### Time Efficiency

Current methods require data collection and extracting the data models manually from construction drawing, schedules and field report from the actual site. There are research [Mani et al., 2009] states that field personnel collect the progress data from the site and perform analysis before sending to project management team to make decisions, which will be a long run time consuming process. Figure 2.6 shows an example of existing progress reporting techniques, which all things were hung on a wall in order to let all team member to know about the construction progress. This reporting format does not solve problems when it comes to timely manner, all of the report would take times to sort out, prioritize and interprets. However, with the help of Augmented Reality, it can help people who does not have deep knowledge on construction situation to understand the site more. Figure 2.5 shows the illustration of progress monitoring using Augmented Reality as a tool. The progress monitoring is done by superimpose the real-world construction with the planned 3D model. By comparing the real-world construction and 3D model, work completed and work remaining can be clearly visualized.

A screenshot of a cell phone

Description automatically generated

Figure 2.5: Augmented-Reality based progress monitoring. [Lee & Pena-Mora, 2006]

A picture containing text, table

Description automatically generated

Figure 2.6: An example of existing progress reporting techniques. Construction drawings and work schedules are hung on a construction site trailer’s wall to communicate progress with contractors and subcontractors. [Mani et al., 2009]

### Data quality and accuracy

The data that has been manually collected and extracted may be appear in a low-quality manner. [Mani et al., 2009] The information of the construction progress collected tends to base on the people who interpret of what needed to be measured and it may affect the quality of the data and lead into data error since the ability of measuring progress is based on the experience of the person that collecting the data. Figure 2.7 shows a sample of real project progress example to show the percentage done of a work performed in an actual site. As the example shown, different subcontractor may have different judgment and decision to decide the complete percentage of work performed. This will result in an inconsistency data that brings a problem when it comes to decision making for the production management team.

A screenshot of a cell phone

Description automatically generated

Figure 2.7: A sample of a real project progress/inspection report. [Mani et al., 2009]

### Completion progress

Using augmented reality as a tool, it can be linked to a desired schedule so that if there is any discrepancy from the schedule it could be known in first glance. Figure 2.8 show the progress monitoring chart to detect deviation in construction progress. In this example, colors are used to represent the key performance index of the progress monitoring process. For example, green would represent ahead of schedule, whereas red represent behind schedule. Besides, the figure also shows the deviations based on a progress performance metrics such as Schedule Performance Index (SPI) and Cost Performance Index (CPI) which can be quantified and analyze later on. It is an additional improvement to use augmented reality to overcome the progress monitoring from Figure 2.7 manual input of the progress from sub-contractor. In this case, the progress can be software oriented and pre-determined so that everyone would share the same progress in the production management team. If all of the progress monitoring is automated, the augmented reality visualization techniques would give a better support for the production management team to perform the job in real-time.

A screenshot of a social media post

Description automatically generated

Figure 2.8: Progress Deviation Visualize. [Lee & Pena-Mora, 2006]

In one of the research topics [Fard & Peña-Mora, 2007], a series of vision-based methods has been automated based on the progress detection and visualized status. Figure 2.6 shows a proposed algorithm based on the research, which consist of 5 major steps, registration of the 3D model to the environment, analyze of the progress status, color coding, superimposing of the 3D model to environment and remove 3D object that has occluded in the environment.

Diagram

Description automatically generated

Figure 2.9: System process and work flow. [Fard & Peña-Mora, 2007]

## Point Cloud Capturing

There are many ways when it comes to point cloud capturing. The methodology can be classified into three, aerial capture, terrestrial capture [Hinks et al., 2013] and hand-held capture. Aerial capture where large scale data is being collected to do a mapping of big map area where the accuracy of the point clouds is not so important. Figure 2.10 shows a point cloud data example of aerial capture from a city of Enschede. [Stucker, 2017] Terrestrial capture which a small area of compound is being capture for construction sector and development purpose. An example for terrestrial capture is as shown as Figure 2.11, where the building is captured in point cloud form.

A picture containing person, flower, food

Description automatically generated

Figure 2.10: Point Cloud dataset of Enschede. [Stucker, 2017]

A picture containing building, outdoor, front, tower

Description automatically generated

Figure 2.11: Point cloud captured of a building. [Hinks et al., 2013]

Hand-held capture where the method focusses on accuracy and high precise area of small object for digital analysis and reconstruction. This sub chapter will discuss about the ways of collecting point cloud data in real world.

### Light Detection and Ranging (LiDAR)

LiDAR is a sensing process which collects measurement from the environment, which emits laser light to the real world. LiDAR is able to measure the roundtrip time for the laser to bounce back to determine the depth and distance of the object accurately. Since LiDAR provide an accurate dataset, this system has various application as a high resolution and highly accurate measurement techniques. [Yoo et al., 2018] Figure 2.12 shows the basic structure of LiDAR, where a laser pulse is generated and hits the object, then the receiver detects the reflected laser pulse, and finally the data acquisition calculate the unit distance of the object from the scanner itself.

A close up of a map

Description automatically generated

Figure 2.12: Basic structure of LiDAR. [Yoo et al., 2018]

LiDAR sensor has been widely used in various field, for example autonomous driving, building construction operations, forestry and robotics. In 2020, Apple introduced a new feature to the iPad, where LiDAR sensor is implemented on a hand-held system, which combine camera and motion sensor data, and the iPad is able to understand the scene more, enables cross-fade the virtual and reality world in an accurate manner. However, the LiDAR data provided from the company is limited by users in an open-source format by the time of this thesis being written. Figure 2.13 shows the image of a 2020 iPad pro with LiDAR 3D scanner. The recent technology of AR Foundation also enable user to capture meshes from the real-world scene using LiDAR sensor that being built on am iPad 2020. The latest AR Foundation also provides a library that enable users to generate mesh based on the LiDAR sensor information, which means it will bring a much more accurate data compared to the traditional mesh generator.

A close up of electronics

Description automatically generated

Figure 2.13: iPad camera with LiDAR scanner. (Picture acquired from Apple site)

### Photogrammetry

Comparing to 3D scanning from LiDAR, photogrammetry uses photographs to gather point clouds data. Many images have been taken in a different angle to capture the object of interest, and the overlap image will provide a data information to form the depth of the object. The advantage of using photogrammetry is that it is less expensive compare to 3D scanning, and it is able to produce an object of interest in full color and texture. Photogrammetry is also a very accessible as the equipment and software is not as complicated as 3D scanning. On the other hand, it has a few drawbacks, which is accuracy and precision. Data processing will be needed for the point clouds data in photogrammetry. Normal smartphones or camera are able to collect point clouds data by using software as an intermediate source to generate the data. There are many libraries that offer the function of generating the point clouds based on feature points of the object of interest. Figure 2.14 shows an example of a point cloud that has been gathered through photogrammetry based on feature points of object. As mentioned previously, AR Foundation has a function to capture the point clouds based on photogrammetry.

A cup of coffee on a table

Description automatically generated

Figure 2.14: Point clouds based on feature points of object.

## Point Cloud Representation

Point cloud is defined in a coordinate system and describes a three-demensional object. Point clouds are gathered through scanning of 3D object using various measuring techniques. Each of the points has an XYZ value that determine the position in the coordinate system. Point clouds consist of millions of points and it is a collection of coordinate points in space. Depending on application, point clouds usually are filtered and processed in order to then highlight the measuring process with certain parameter. Most of the time, point clouds are converted into polygon mesh in order to represent the surface of the scanned object.

One of the major advantage of using point clouds is that it is easy to display the information and filter unwanted points. Since the scale or rotation of points is less important, computational is much easier to handle huge amount of data. Point clouds are the essential step to digitalize real world object. Based on the raw information of point clouds, further steps can be taken on the application. The points of the point cloud can go through several representation, such as meshing, polygonization, and voxelization.

### Meshing

Meshing provide a pleasant visualization for point cloud representation. Nearest neighbors of a point cloud are assumed to represents a small piece of the object’s surface. [Linsen, 2001] The amount of point clouds data is usually large, which will cause measuring error. In order to eliminate such errors, smoothing operator to the point clouds must be applied in meshing representation. Meshes are created using computer algorithm, which one of the input domain geometry is by using point clouds. Techniques such as *Moving Least Squares* (MLS) [Fleishman et al., 2005] can be used as a robust statistics method for surface smoothing. Meshing is done by approximation algorithm, where each data point from point cloud is projected to nearby neighbor’s point. The approximation will be computed based on the projection parameters and adjusted to the least approximation error. [Yoon, 2006] Figure 2.15 shows the approximation process of point clouds computed into control mesh.

A close up of an animal

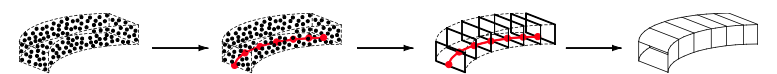
Description automatically generated

(a) (b)

Figure 2.15: (a) point cloud, (b) control mesh.[Yoon, 2006]

### Polygonization

Polygonization is one of the methods to present an object from point cloud data. The 3D models must be prepared using scanned data, where the set of point cloud is being converted to meaningful polygons. This method is widely used in civil infrastructure, [Hidaka, 2016] which maintenance is a critical issue for government in order to prolong the service life. Most of reconstruction methods [Kazhdan et al., 2006] may fill the incomplete structure, however, the accuracy of polygonization is not completely guaranteed although some of the missing region of object can be estimated. Figure 2.16 shows the polygonization of point cloud using lofting operation. [Hidaka, 2016] This method includes four major steps, which is gathering input point cloud data, center line identification, cross section creation and finally polygonization.



(a) Input point cloud (b) Center-line identification (c) Cross-sections creation (d) Polygonization

Figure 2.16: Polygonization of point clouds. [Hidaka, 2016]

### Voxelization

Voxelization defines the point clouds as 3D boxes in 3D space. [Ruchay et al., 2019] Figure 2.17 shows a sample point cloud data and corresponding voxel representation. [Gokberk et al., 2008] Each voxel, a point is chosen to approximate all the points that lie on that particular voxel. The center of the average points is usually taken as an approximation. However, voxelization method usually leads to information loss of point clouds. Voxelization is used widely for FEM construction, in which point clouds data is semi-automatically computed and the models reconstructed based on voxel grids with critical parameter as voxel size and number of voxel grid. [Ruchay et al., 2019] Figure 2.18 shows the workflow of voxelization used in FEM model reconstruction. The point clouds for FEM model reconstruction is gathered through LiDAR sensor, which then being carried forward for reconstructing procedure. Details of the FEM construction of voxelization work can be found in [Hinks et al., 2013].

A close up of a cage

Description automatically generated

Figure 2.17: Point clouds and binary voxel representation. [Gokberk et al., 2008]

A screenshot of a cell phone

Description automatically generated

Figure 2.18: Workflow of FEM model reconstruction. [Hinks et al., 2013]

## Noise Filterring and Outliers Removal

As mentioned before in the sub chapter, that it is easier to remove unwanted points and display desired information from the point clouds raw data. Point clouds obtained with 3D scanners or imaged-based reconstruction are often corrupted with huge amount of noise and outliers. [Rakotosaona et al., 2020] This sub chapters will discuss the ways and methods to filter the point cloud gathered, which includes filtering algorithm, sparse outliers’ removal, normal and curvature estimation.

Noise filtering is very crucial in point cloud processing, as the object of interest to be analyzed must have a clean point cloud data in order to bring good result and simulation for the real-world application. Fundamentally, a good point cloud filtering method should have the following behavior: [Rakotosaona et al., 2020]

1. Eliminates noise without deleting important details
2. Self- tuning. (e.g.: Noise model is not required to estimate surface)
3. Avoid degrading input data.

The filtering and denoise method that will be discussed in this sub-section will be neighborhood-based methods, principal component analysis, and filtering point cloud data from RGB information.

### Neighborhood-based methods

The non-local image filter [Wang et al., 2006] is one of the neighborhood-based methods to recover original data from noisy measurement. The concept is as shown in the equation below, where is the observed value, is the object of interest value, and is the noise perturbation at pixel .

This method defines the intensity value of pixel in an image depending on the weight average of neighbor point cloud data with the same intensity value.[Schall et al., 2006] This method is widely use in image processing, which an image , where is pixel position and is the intensity value. The smoothed value of intensity is average computed as follow:

Sparse outliers which caused by measurement errors may corrupt the overall point clouds data. [Rusu et al., 2008] These outliers will cause complication for the estimation of local point clouds data. Fortunately, sparse outliers removal module is able to correct the irregularities by computing the mean and standard deviation of nearest neighbor distance. Figure 2.19 shows the overall result for sparse outliers removal based on neighborhood-based method.

A picture containing indoor, old, photo, table

Description automatically generated

Figure 2.19: (Left) Sparse outliers (Right) Point clouds data after outliers removal. [Rusu et al., 2008]

### Principal Component Analysis

Principal Component Analysis (PCA) method is used to effectively smooth the noise in point cloud data without losing original feature of the object during 3D scanning. [C. C. Jia et al., 2018] Principal Component Analysis consist of three essential steps, first, the normal vector is estimated using the algorithm where the surface variation is estimated. Then, the point cloud data is being separated into different region by comparing the surface variation. Finally, the filter algorithm is being applied based on the new normal vector of the point clouds data and adopted to smooth the regions that are being separated.

However, there are problems when using Principal Component Analysis. [Belton, 2008] First, neighborhood point cloud data may consist various discrete surface entities, where the algorithm will estimate that there is only one surface structured. The second problem by using Principal Component Analysis is that the point clouds data information can be lost, such as object surface curvature through surface variation.

### Filter point cloud data from RGB information

Point clouds data that has been collected by RGB camera is usually distort with points and outliers that do not belong to the object of interest. The point clouds that generated from the image taken by the RGB camera has limitation as it depends on light intensity and also different viewing angles. Figure 2.20 shows the overview of the method to process point clouds data from RGB information [C. Jia et al., 2019]. The image taken from RGB camera is processed and required point clouds data is being extracted and filtered. This method is able to segment the object of interest and outliers, by modifying the image from RGB information to depth image, and extract the object of interest, which remove the outlier noise. Figure 2.21 shows the results of this particular method of filtering, where the RGB image is mapped and filtered. This method uses four different angle cameras to capture the RGB data on the same spot, which provide a sufficient computation for depth input and point cloud data based on feature points of the scanned object.

A close up of a device

Description automatically generated

Figure 2.20: Filtering Process of PCD based on RGB image. [C. Jia et al., 2019]

A picture containing food

Description automatically generated

Figure 2.21: Process from RGB image, maping, original PCD and filtered PCD. [C. Jia et al., 2019]

## Performance Evaluation Method

This sub-chapter discuss about the method used for evaluating the performance of the research method used. There are many researches which present metrics and algorithms evaluation for tracking system. The evaluation is statistical detection and estimation theory to evaluation tracking task using frame based as well as evaluation based on object of interest. Both of the methods require ground truth knowledge to be able to perform the evaluation. The evaluation method is based on a statistical hypothesis of Type I (α) and Type II (β) error, which consist of true negative, true positive, false negative and false positive. The method will be further discussed in the sub chapters of each metrics method. Figure 2.22 shows an illustration of the Type I and Type II error being implemented. Type I errors are assimilated with false positives, this happen when the hypothesis is true but being rejected. For example, in a construction progress scene, where the object is not being built and completed, but however the system validate that the object is available. Type II errors on the other hand are assimilated with false negatives, this will happen when the hypothesis is being false but being accepted. We will take the same construction scene that we discussed before as another example, where the object is being completed and built, but the system validates the object is not available.

A picture containing diagram

Description automatically generated

Figure 2.22: Type I and Type II error illustration.

### Frame based Metrics

Frame based metrics uses a method of starting with the initial frame of the sequence, and compute every frame in the sequence. [Bashir & Porikli, 2006] Firstly the true and false detection are being computed as follow:

* True Negative (TN): Total number of frames that ground truth and system agreed that the object is absence.
* True Positive (TP): Total number of frames that the ground truth and system agreed that the object is present.
* False Negative (FN): Total number of frames that the ground truth contains presence of the object, but the system register that the object is absence.
* False Positive (FP): Total number of frames that the system registered presence of the object, but the ground truth contains presence of the object.

(Table) shows an example of how the metrics are computed, where total ground truth (TG) is the total number of frames for ground truth object and TF is the total number of frames in the sequence.

Table 2.1: Metrics to be computed based on defined quantities

|  |  |
| --- | --- |
| Tracker Detection Rate |  |
| False Alarm Rate |  |
| Accuracy |  |
| Detection Rate |  |

### Object based metrics

Object based metrics evaluates the complete sequence and lifespan of individual system or application. The correspondence of ground truth is far more than one application track; therefore, a mapping has to be done before the evaluation. With that being said, the mapping of object and also frame based metrics are computer. Figure 2.23 shows an example from a research [Bashir & Porikli, 2006] where the metrics is being computed based on Euclidean distance of the current object and ground truth knowledge to determine the core metric (TN, TP, FN and FP). If the threshold of Euclidean distance has reached certain value, the decision is being made whether if it is True Positive or False Negative, and vice versa.

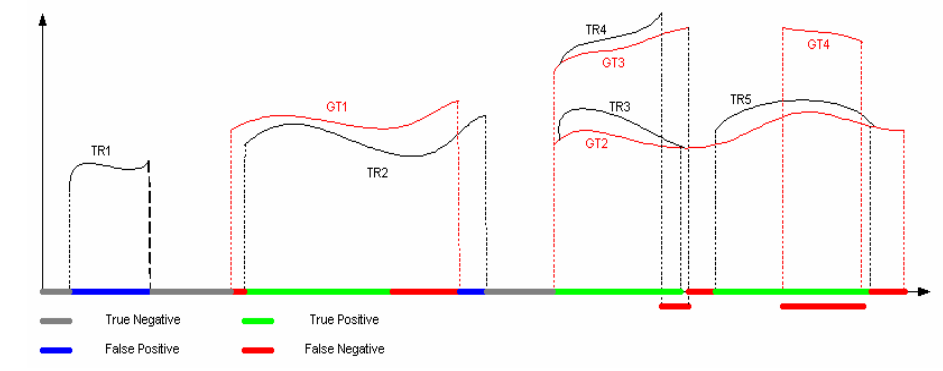


Figure 2.23: Definition of core metric for object-based metric. [Bashir & Porikli, 2006]

## Resources used for the application

In order to have a up and running application, a development platform and tools need to be prepared and well researched. This sub-chapter will discuss about the resources used for Augmented Reality Application in this thesis work, which will include the Integrated Development Environment (IDE), tools for the IDE, and supporting frameworks for creating the application.

### Game Engine: Unity

Unity is an IDE that enable designers and developers to work together to create application. This platform enables users to create games, application for automotive, transportation, manufacturing, film, animation and even architecture, engineering and construction. Unity also supports a very wide range of platform products, which include macOS, iOS, Android, Windows, Xbox One, Oculus, Nintendo Switch and many more. [Unity, 2019]

Unity is also able to support numerous AR and VR platforms, such as Unity MARS, and AR Foundation. This technology is possible with the use of Unity engine, which created in 2005 by the San Francisco-based Unity Technologies, which is primarily for the video-game industry. [Chen, 2019] It is widely used in most mobile application in AR, including the popular augmented-reality game Pokémon-Go. Besides, AR application is not merely just for gaming, but also good for simulation and construction progress in engineering field.

Unity provide an XR platform, which enables developer to work in AR or VR application. Figure 2.24 shows the structure of Unity XR platform, which the provider framework define the implementation and device specific SDK, which then able to handle translation of platform specific representation. The middle layer from the figure is an interface layer, which optimize core engine implementation from Unity. The developer framework, which is AR foundation framework, is a layer that expose the functionality of the subsystem from provider framework, which is also game object based and developer friendly to use the framework effectively.

Diagram

Description automatically generated

Figure 2.24: Unity XR platform structure. [Unity, 2019]

### Tracking Framework: AR Foundation

AR Foundation is an open-source AR Foundation allows user to work with Augmented Reality platforms in a multi-platform, such as iOS and android within Unity. In fundamental level, AR foundation uses separate packages (ARKit for iOS and ARCore for Android) for the target platform officially supported by Unity. In short, AR Foundation enables users to write the application once, and build for both Android and iOS platform. However, AR Foundation does not implement all features from ARKit and ARCore, AR Foundation provides a scripting API and MonoBehaviours for making both of ARCore and ARKit application that use core functionality to share between Android and iOS platform. Table 2.2 and Figure 2.25 shows the summary of both of the SDK for ARKit, ARCore and AR Foundation. AR Foundation contains APIs that support various features from the packages, for instance, device tracking which track the device’s position, rotation and orientation in real-world environment. AR Foundation also provide other features API such as plane detection, point clouds, light estimation, face tracking, meshing and even body tracking. AR Foundation package wraps API and enhance with advance utilities, such as creating GameObject in unity platform to represent detected features in real time environment.

Table 2.2: Description of each AR Foundation, ARCore, and ARKit.

|  |  |  |  |
| --- | --- | --- | --- |
| **Details** | AR Foundation | Unity ARCore SDK | Unity ARKit Plugin |
| **Description** | Wraps ARKit and ARCore low-level APIs into a cohesive framework | Provides native APIs for all essential AR feature supported on ARCore for Android platform | Plugin for building ARKit experiences in Unity that exposes the objective- APIs in C# for use within Unity. |

Diagram

Description automatically generated

Figure 2.25: AR Foundation chart.

### Analyzing Tool: Open3D

Open3D [Q.-Y. Zhou et al., 2018] is an open source library that support development of software that deals with 3D data. Frontend of this library uses a set of data structures and algorithm in programming languages such as C++ and Python. This library consists of core features such as 3D data processing algorithms, scene reconstruction, surface alignment and many more. An example of a data processing task is shown in Figure 2.26, where a set of point cloud is loaded, down sampled, and normal is estimated. Besides, this library also provides implementation of multiple surface registration method, such as iterative closest point. Open3D ease up the workflow of point-cloud data analysis and it is useful for broad community of developers who are dealing with complex 3D data.

A picture containing game

Description automatically generated

Figure 2.26: 3D data processing: load a point cloud, downsample it, and estimate normals. [Q.-Y. Zhou et al., 2018]

This library enables users to perform data analysis on point-cloud data, which including cropping point cloud based on point of interest, visualizing point cloud to a desirable view, point cloud outliers removal which filter unwanted noise from the point clouds data, surface construction which helps in reconstructing missing part by estimating the normal of the point cloud data and many more. Open3D also supports various type of point clouds data, with the format of: .xyz, *.xyzn*, *.xyzrgb*, *.pts*, *.ply* and *.pcd*, which is consider very flexible for an open source library. There is possibility to modify point cloud into desire format, such as meshes and voxel, Figure 2.27 shows the implementation of voxelization from a point cloud to voxel. For more thorough and view for the library function, it can be found here in [Q.-Y. Zhou et al., 2018].

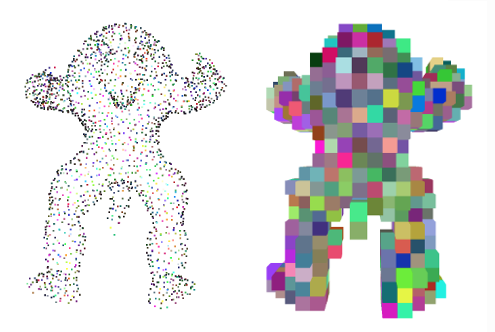


Figure 2.27: Voxelization from a point cloud dataset in Open3D

set

# Analysis

This research is based on the development of a method to track construction progress. The master plan of the project as a whole is to enable the operator to track the construction progress with an augmented reality system that enables simulation visualization in real-time in the operating site. The AR experience then enables the operator to make intuitive decisions to determine the overall construction progress. As a result, this AR assistance aims to increase the productivity for the management planning and commissioning of the construction process for the operating site. This chapter will discuss the analysis part of the overall work, which will also include limitations and criteria of the overall AR system. Two major methods have been used to compare and analyzed, which are the point cloud comparing method and mesh data comparing method. All of the resources and methods will be discussed in this chapter.

## Analysis of Construction Progress

This section will discuss the analysis part of the construction progress using the Augmented Reality application. Section 3.1.1 will discuss the requirement of ground-truth knowledge of the object of interest and Section 3.1.2 will then discuss the differentiation of big and small parts of the assembly for the object of interest. Section 3.1.3 will then discuss about the basis of assembly for the object of interest.

### Ground Truth Knowledge

Operator must already acquire a virtual model to have a general ground truth knowledge comparison with the real model. That means the virtual CAD model is pre-defined in the application before the scanning of object of interest start. Moreover, object parts can be falsely taken as a mounted part where the parts are not available. The hypothesis is that with the help of the AR application, the human errors are able to reduce, so that the construction process can be track semi-automatically.

The CAD model should be also accurate to the real-world object in this case, this allows components of the same part to be distinguished from one another. The procedure described forms the standard course. This part is also important where the result is highly dependent on the accuracy and precision of the model itself. The dimension and orientation of the virtual model plays also a crucial role in the application itself; this is to be done because the virtual object will match with the real-world object in the AR application. Thus, dimension and accuracy of each part in the CAD model have to be nearly perfect to match with the real-world object.

Before discussing about ways of capturing point cloud data and mesh data, ground truth knowledge plays a crucial role in the AR application. As mentioned before, a pre-defined model has to be known in order to compare the model with the real-world object of interest. Every part of the model is defined separately in the backend of the AR application, so that the AR application is able to differentiate which point or meshes belongs to which specific part of the CAD model. Figure 3.1 shows the CAD model that has being use in this research. The CAD model consist of the parts as shown in Table 3.1. This information will be then used to map the feature points or meshes to respective parts in the CAD model. As shown in the table, this CAD Model consist of 41 separate parts to assemble. Every part of the engine is accurately and precisely digitalized into the CAD Model.

A common reason is that to have a one-to-one comparison between the virtual object and real object, in this case, the engine itself. The CAD model and the engine part is provided from the institute for this thesis work. The marker is attached to the front part of the engine to enable AR application to recognize the object. The engine also been put onto a mount to enable the object to be upright standing position for the tracking purpose. This however disclosed certain part of the engine to be exposed virtually. In this case, the AR application will not be able to capture the outer shaft of the engine itself.

A picture containing light, toy, sitting, table

Description automatically generated

Figure 3.1: CAD model for ground truth knowledge.

Table 3.1: Part name of the CAD Model

|  |  |  |  |
| --- | --- | --- | --- |
| Housing Screw 1 | Passfeder 1 | Big Housing (57310000) | Bearing (6200) |
| Housing Screw 2 | Passfeder 2 | Middle Case (57313000) | Bearing (6200) X |
| Housing Screw 3 | Passfeder 3 | Gear (57371420) | Bearing (6209) |
| Housing Screw 4 | Locking ring 1 | Gear (57373070) | Middle shaft (57330000) |
| Housing Screw 5 | Locking ring 2 | Gear (57371350) | Spacer (57382020) |
| Housing Screw 6 | Locking ring 3 | Gear shaft (57372220) | Spacer (57392010) |
| Housing Screw 7 | Outer Hexagon Screw 1 | Gear shaft (57372400) | Radial shaft seal |
| Housing Screw 8 | Outer Hexagon Screw 2 | Bearing (6201) | Adapter |
| Housing Screw 9 | Outer Hexagon Screw 3 | Bearing (6202) |
| Oil Inlet Screw | Outer Hexagon Screw 4 | Bearing (6204) |
| Oil Outlet Screw 1 | Oil Outlet Screw 2 | Bearing (6300) |

### Differentiation of Assembly Parts

As mention in the previous section, there are many assembly parts needed to be analyzed to track the construction progress. The assembly parts consist of different parts and sizes. In order to have a more precise and accurate data analysis, the assembly parts are divided into two categories.

The bound size of the CAD model part is acquired in the Unity Game Engine, where it runs in the backend of the application. Table 3.2 shows the summary of boundary magnitude for different CAD model parts, which categorize into larger part and a smaller part. The table is shown in ascending in the bound magnitude of the CAD model itself. The categorization of larger part and smaller part has a threshold value of 2000, which means as long as the value is larger than 2000, it is considered as a larger part. Figure 3.2 shows the assembly parts of the object, , based on the figure itself, it consists of parts that are very small and different sizes of parts

Table 3.2: Part Number with Bound Magnitude and Threshold Percentage

|  |  |
| --- | --- |
| **Part Name** | **Bound Magnitude** |
| Oil Outlet Screw 2 | 503 |
| Oil Outlet Screw 1 | 507 |
| Passfeder 1 | 513 |
| Passfeder 2 | 513 |
| Passfeder 3 | 513 |
| Oil Inlet Screw | 675 |
| Spacer (57382020) | 1085 |
| Spacer (57392010) | 1129 |
| Housing Screw 1 | 1160 |
| Housing Screw 2 | 1160 |
| Housing Screw 3 | 1160 |
| Housing Screw 4 | 1160 |
| Housing Screw 5 | 1160 |
| Housing Screw 6 | 1160 |
| Housing Screw 7 | 1160 |
| Housing Screw 8 | 1160 |
| Housing Screw 9 | 1160 |
| Outer Hexagon Screw 1 | 1387 |
| Outer Hexagon Screw 2 | 1387 |
| Outer Hexagon Screw 3 | 1387 |
| Outer Hexagon Screw 4 | 1387 |
| Bearing (6300) | 1866 |
| **Other Parts** | >2000 |

In addition, the AR application will only detect seen object in the AR scene, which means the hidden part of the assembly part will not be detected in the AR application. With that being said, assembly parts that are hidden during the construction progress will show no data at the AR application. Further details of the discussion will be discussed in the next chapter.

A picture containing indoor, different, electronics, variety

Description automatically generated

Figure 3.2: Assembly Parts of The Object

### Assembly of Object Part

The AR application tracks the assembly of the real-world object; therefore, assembly plays a big role in this procedure. In order to have an accurate data, the differential part have to be assembled correctly with the right position and right orientation. Besides the right orientation and right position, the part has to be completely installed and attached to the right place. Figure 3.3 shows the difference of fully fastened middle cover of the engine and a partially installed middle cover. This will affect the accuracy of the result as the part is not fully assemble into the right place. Therefore, every installation or assembly must be correct into the right place so that the application will be able to retrieve the data correctly and accurately.

All things considered, the operator must have prior knowledge of the assembly process and the AR application is just a supporting tool to track the construction progress in site. Another factor is that knowing the sequence of the assembly process, so that nothing in internal is left out before installing the outer part. Section 3.1.2 had discussed about the AR application ability, where the application can only be able to track visible parts of the object of interest.

A picture containing indoor, sitting, table, plane

Description automatically generated A picture containing indoor, table, sitting, small

Description automatically generated

Figure 3.3: Partially Assemble and Fully Assemble Middle Cover

On the other hand, there is also another scenario which will affect the accuracy of the gathered data, which is assembly wrong parts onto the wrong position. The operator must be able to distinguish the assembly part to put it into the right place of the object. The scope of the AR application does not detect if there is wrong part been place onto a wrong place. If the wrong part falsely assembled, the AR application will assume that the assembly is completed, therefore the parts must be well-known by the operator before the assembly begin. A common mistake is that fastening the screw with different size and the application assumed that the screw part is there. Although this is a rare case, but there is still possibility to happen and it may affect the accuracy of the data.

## Environmental Condition

This section analyzes the environmental condition of construction progress using the Augmented Reality application. By looking at the overall aim of the problem statement, there are criterion, boundary and limitation for the developed AR application. The limitation can be explained by looking into the application itself, which include the overall environment and operator. As part of this work was the assembly of an engine and tracking the construction progress. There are several environment requirements that need to be set for this particular AR application:

* An indoor environment with sufficient artificial lightning
* Enough space that makes dynamic movement possible

Section 3.2.1 will discuss about the lighting condition of environment, where the methods is applicable in the AR application. Section 3.2.2 will analyze the area of experiment design.

### Lighting Condition

The environment status is crucial for the AR application to work, as insufficient lighting will cause an incorrect tracking position relative to the camera and world co-ordinates, and this may impact the results of the overall work. On the other hand, since the initial object tracking is based on a marker, sufficient lightning is crucial for the device camera to detect the marker and perform good tracking. Figure 3.4 shows an example of how the lighting will influence the object tracking based on marker affect the overall tracking, it is nearly impossible for AR application to track a marker in a low light condition. In the case under consideration, all of the data for this thesis work is gathered with sufficient lighting conditions in an indoor environment.

A person in a dark room

Description automatically generated A close up of a computer

Description automatically generated

Figure 3.4: Low-light environment and Light-sufficient environment

There are two ways of method to be evaluated in the AR application, point cloud capturing and mesh data collecting, where point cloud capturing requires the object to be seen during the session, which means that the object must be in a very sufficient lightning condition in order to capture the data correctly and accurately.

On the other hand, mesh data is collected by using the LiDAR sensor from the iPad device. LiDAR does not require sufficient lighting to track the surrounding object, but however, the CAD model is needed to be placed in the right position, which it is also important to perform marker tracking correctly. With that being said, sufficient lightning is still needed in order for tracking the superimposing CAD model on the real-world object. The bottom line is that both of the methods would require sufficient lighting to have a more accurate and precise data collection.

### Range of Experiment Environment

The AR application will be used in an indoor environment, the functionality of the device, such as GPS tracking shall be omitted from the design. GPS tracking is not taking into consideration due to the small working area that this thesis work is being carried out. On the other hand, the indoor environment shall consist of enough space to accommodate the movement of the operator to scan the object of interest. This requirement is crucial so that every part from the object of interest will be covered and scanned by the device to get an optimal result.

Besides, it is also important to ensure that every part of the object of interest is captured, for example, the operator has to walk around the object of interest with an AR device and capture the data of the scene. Figure 3.5 shows the illustration of the workspace required. Space ranges minimum between 1 meter and 3.5 meters to ensure that the device is not too close or too far from the object of interest. If the object is too close, the camera would not be able to capture the overall view, and it may cause inaccuracy of the data collected. On the other hand, if the device is held too far away, the application will fail to capture the detail feature of the object.

A picture containing photo, flying, colorful, air

Description automatically generated

Figure 3.5: Workspace of the AR application

## Framework and Tools Analysis

This section will discuss the framework and tools used for the research. Section 3.3.1 will discuss the analysis of Augmented Reality Framework and Section 3.3.2 will discuss the use of LiDAR sensor. The concept of how to implement the framework and sensor will then be discuss in the next chapter.

### Augmented Reality Framework

The Augmented Reality application uses the tracking framework from AR Foundation Library. The library provides a reliable image tracking where it can be implemented into the AR application and use accordingly. The AR application uses the tracking framework directly from the AR foundation to perform the marker tracking in the AR scene.

AR foundation provides world tracking, where it can track the device’s position and orientation in physical space. With that being said, the device camera position is taken into consideration to determine the current object of interest position. The tracking provided from AR foundation in this application is assumed to be nearly perfect, as it is not in the scope of the problem statement. However, further calibration of the image tracking is needed which will be discussed in the next chapter.

### LiDAR Sensor

The AR application uses LiDAR sensor to collect mesh data, hypothetically, this provides an accurate data compared to a classical feature points collection. With the application of LiDAR sensor, the device is able to interact physically with the real world and collect the data by knowing the depth of the object and features of the object.

AR Foundation provides a library tool that enable the application to collect the mesh data using LiDAR sensor. The addition of LiDAR sensor enables the application to collect data much faster than the classical feature points collection. As discussed in previous chapter in Section 2.3.1, the iPad combines camera and motion sensor data, and the iPad is able to understand the scene more, enables cross-fade the virtual and reality world in an accurate manner. The LiDAR sensor was instead meant to be used for augmented reality for the new iPad 2020. It can detect and classify features of object in the environment.

The application can only collect the mesh data from the LiDAR sensor on the device and there are not any APIs available for developers to get the raw depth data from the sensor. Further detail of the concept of collecting mesh data using LiDAR sensor will be discussed in the next chapter. The latest technology enable operator to capture the data more efficient due to the computing speed and LiDAR sensor also helps a lot in collecting meshes data which will be discussed further in this chapter. Therefore, a strategy concept is needed to be constructed to use the data for tracking the construction progress. Further discussion will be discussed in the next chapter regarding the concept of using the LiDAR sensor data for meshing purposes.

 A picture containing dome

Description automatically generated

Figure 3.6: Mesh Data from LiDAR Sensor from Apple.

Figure 3.6 shows the mesh data collected from LiDAR Sensor from an example application provided from apple, *LiDAR Mesh Export*. It can be seen that not all of the feature is being captured, where only blocks of meshes being show in the device data. However, the data collection is very fast with this latest technology. Therefore, a strategy concept is needed to be constructed to use the data for tracking the construction progress. Further discussion will be discussed in the next chapter regarding the concept of using the LiDAR sensor data for meshing purposes.

# Concept

This chapter will discuss the concept of the software architecture of the AR application. Based on the previous analysis chapter, it is known that AR application is an aiding tool to evaluate the construction progress. The development of concept is as follow:

* Mobile recognition of point cloud/mesh data with the digital assistance system
* Comparison with the known 3D Model
* Derivation of the indicator construction progress for the whole assembly

By using the available tools and framework, the concept is generated to develop the Augmented Reality application for the use of automatic recognition of construction progress. There are two methods used for the Augmented Reality application, which is capturing point cloud data and capturing mesh data. Section 4.1 will discuss the concept of capturing point cloud data method and Section 4.2 will discuss the concept of capturing mesh data method.

## System Design

The aim of the AR application is to track the construction progress by comparing the known model and real-world object of interest. The AR application will collect the data collected from the object of interest. Figure 4.1 shows an illustration of how the AR application generally works. An example based on the figure can be explained as follow:

1. AR device’s camera capturing the object of interest.
2. AR device detected the object of interest.
3. AR application will superimpose a virtual object upon the object of interest as shown in Figure 4.2.
4. Data is captured based on the intersection of real and virtual object.
5. Analysis is made based on the data captured.
6. Construction progress percentage will then be presented in other analysis tool.

Shape, rectangle

Description automatically generated

Figure 4.1: General illustration of AR application

There are also assumption and boundary condition needed to be clarified for this application. The condition is as follow:

1. Superimposed virtual object is assumed and taken to nearly perfect fit from the real-world object.
2. Undetectable object due to virtually occlusive which is inside the object of interest is assumed to be presence.

Shape

Description automatically generated

Figure 4.2: Virtual object superimpose to real-world object.

As mentioned before, the method used to monitor the construction progress is to compare the virtual object with the real-world object. Therefore, the object tracking is then assumed to be nearly perfect fit to each other to enable better analysis for the construction progress.

Secondly, virtually occlusive object is also assumed to be presence if it is undetectable. For example, referring to Figure 4.1 and Figure 4.2, object P1 is inside of object P2, which it is virtually impossible for the AR device’s camera to detect object P1, either the camera angle is taken from side to side or from top to bottom. In this case, P1 is assumed to be presence in the monitor of construction progress. By looking at the logic of a normal construction progress, the part that needed to be complete, in this case, the inner part of the overall model, must been construct and complete before the next part to be completed, which is the enclosure of the model. This is to be taken into consideration when operators are using the AR application. The thesis is based on two major part, which is data collecting part and analysis part. Data collecting part is done by developing the AR application, where analysis part is done offline, where the data collected is being analyze and results are plotted. More detail of the method will be discussed in the next chapter of this documentation, which is point cloud capturing method and mesh data capturing method.

### Object Tracking Calibration

Object Tracking calibration is required to reduce the error as small as possible in the system. Although the system and library tools have been designed in a good and structured fashion, there will be still position and rotation errors and this will lead to an inaccurate result. Besides, the application will need to firstl determine the initial position and rotation of the virtual object.

This is done to ensure the virtual object is directly superimposed on the object of interest. Once the marker-based object tracking is established, the AR application will then perform the tracking based on the relative position of the AR camera and the marker within the AR scene. The calibration is necessary to compensate for all possible error, the object tracking is based on the position of the camera itself with the marker position. Figure 4.3 shows an example of why object tracking calibration is important in the AR application, as the virtual object is not accurately superimposing to the real-world object of interest, it will cause an inaccurate result. Therefore, object tracking calibration protocol is needed to achieve a reliable, accurate, and reproduceable result. The marker is been attached to the object of interest as shown in Figure 3.4, the implementation of object tracking will be then further discussed in the next chapter.

With that being said, to have a one-to-one comparison of CAD model and the real-world object, tracking calibration is very important so that the intersection of every part of the object is mapped on each other. As mentioned before, the scope of the problem statement does not cover the tracking performance, the tracking is presumed to be perfect mapped between virtual and real object.

Diagram

Description automatically generated

Figure 4.3: Object Tracking Calibration

### Point Cloud Collection Method

Point cloud collection is one of the two method that being used widely, AR foundation uses the feature points of the real-world scene and collect point clouds data with position data. Figure 4.4 shows an example of the point cloud scene method, where the green cross represents the point cloud data captured based on the feature point of the real-world object of interest and the data is stored when the overlapping distance is within the tolerance between CAD model and object of interest. Figure 4.5 shows the process flow of the point cloud capturing method.

Diagram

Description automatically generated

Figure 4.4: Point cloud capturing in AR application

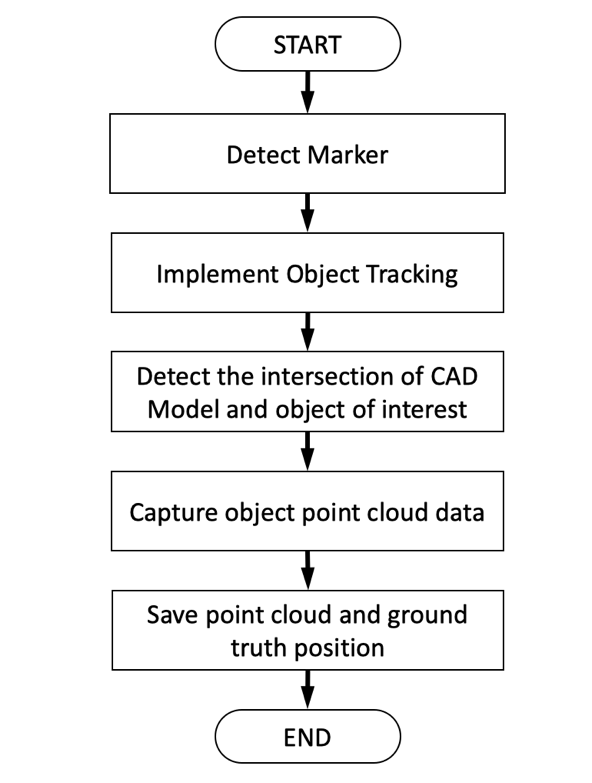


Figure 4.5: Process Flow of Point Cloud Capturing Method.

At the start of the Augmented Reality scene, the application will detect the marker that attached to the object of interest and implement object tracking in the device. Then, the CAD model is superimposed onto the object of interest in the Augmented Reality scene, as shown in Figure 4.4. During the AR session, users have to take the device around the object of interest to collect require point cloud data. Operator must scan every side of the object in order to have an accurate and precise data.

The tracking of the application is assumed to be as a perfect tracking, and it is not in the scope of the work. The application will then detect the intersection of CAD Model and the object of interest, follow by capturing and saving both of the point cloud and ground truth data. Collider is also added to the CAD Model in order for the AR application to detect while the application ray cast into the scene.

AR foundation generate point cloud data based on the feature points of the real-world scene, the scene can be illustrate as shown in Figure 4.6. If there are any changes of point cloud in scene, the application will perform a ray casting on the scene where it will hit the CAD model. Based on the ray-cast return value, the application will then perform an iteration where if there are any feature points detected in that area. If the distance, d, is lesser than a certain fixed distance, the point is classified to the parts. The ground truth knowledge is also collected through the Raycast hit.

Figure 4.7 shows what has been discussed in a flow chart of the software architecture. The point cloud data is then collected and uploaded to file management system where further analysis can be done. On the other hand, the saved point cloud should also mark visually in the device’s screen to show operator that the point has been recorded. The position data of point cloud data will be served to be analyze offline using Open3D as discussed in Section 2.7.3. The concept of analyzing the point cloud data will be discussed in **{Section 4.2.1}**

Diagram

Description automatically generated

Figure 4.6: Point Cloud Capturing Illustration

Diagram

Description automatically generated

Figure 4.7: Flow Chart of the Point Cloud Capturing Algorithm

### Mesh Collection Method

Similar to point clouds collection method, where the application will detect the marker (As shown in Figure 2.4) which will be used for tracking in AR scene. The mesh data is also being collected by the same manner, but different AR scene in the application itself. Instead of capturing feature points, LiDAR sensor captures the mesh data of the real-world object, and with the same process, if the intersection of the CAD Model is overlapping within a tolerance value between the virtual object and real object, the mesh data is then captured and saved. Figure 4.8 shows an illustration of how the AR application will looks like in the scene.

Diagram

Description automatically generated

Figure 4.8: Capturing mesh data in AR application

There is similarity with the previous point cloud capturing method, but however, the concept is slightly different, which we will discuss further in this sub-chapter. This method require data from the LiDAR sensor where the device will be able to collect the mesh directly from the real-world scene. Figure 4.10 shows the flow chart of the mesh data collection method. The additional factor of this concept is that everything is done online, where the percentage can be calculated based on the matching points of the data.

One additional concept is that, when the intersection of both of the virtual and real object reaches a certain percentage threshold, the device will be able to visually present the changes of material in the CAD model to show the availability of the construction progress. The implementation will be discussed in the next chapter. Figure 4.9 shows the illustration of the mesh data collection concept.

Diagram

Description automatically generated

Figure 4.9: Mesh Data Capturing illustration

Diagram

Description automatically generated

Figure 4.10: Flow Chart of mesh data collection method.

This method uses a similar concept from previous point cloud capturing concept, where the device will shoot a ray from the camera position to the object of interest. Taking a single raycast as an example. The algorithm is designed as a concept in this way:

1. RaycastAll into the application camera scene
2. Classified Real Mesh Hit, RHit
3. Calculate distance, d
4. If the threshold is reached within the value, pick RHit and sort into part name.
5. Pick every CAD hit point before the classified RHit

For an example in Figure 4.9, it is shown that the Ray hits two part of the CAD model and there is one hit on the mesh surface. The application will then calculate the distance of the registered mesh data and hit points of the CAD model. If the requirement is fulfilled, the program will register and classified the respective RHit part. Furthermore, the previous CAD hits from the ray will also being registered as a CAD hits as a ground truth hit. Figure 4.11 shows the algorithm that will be used for the mesh data collection concept.

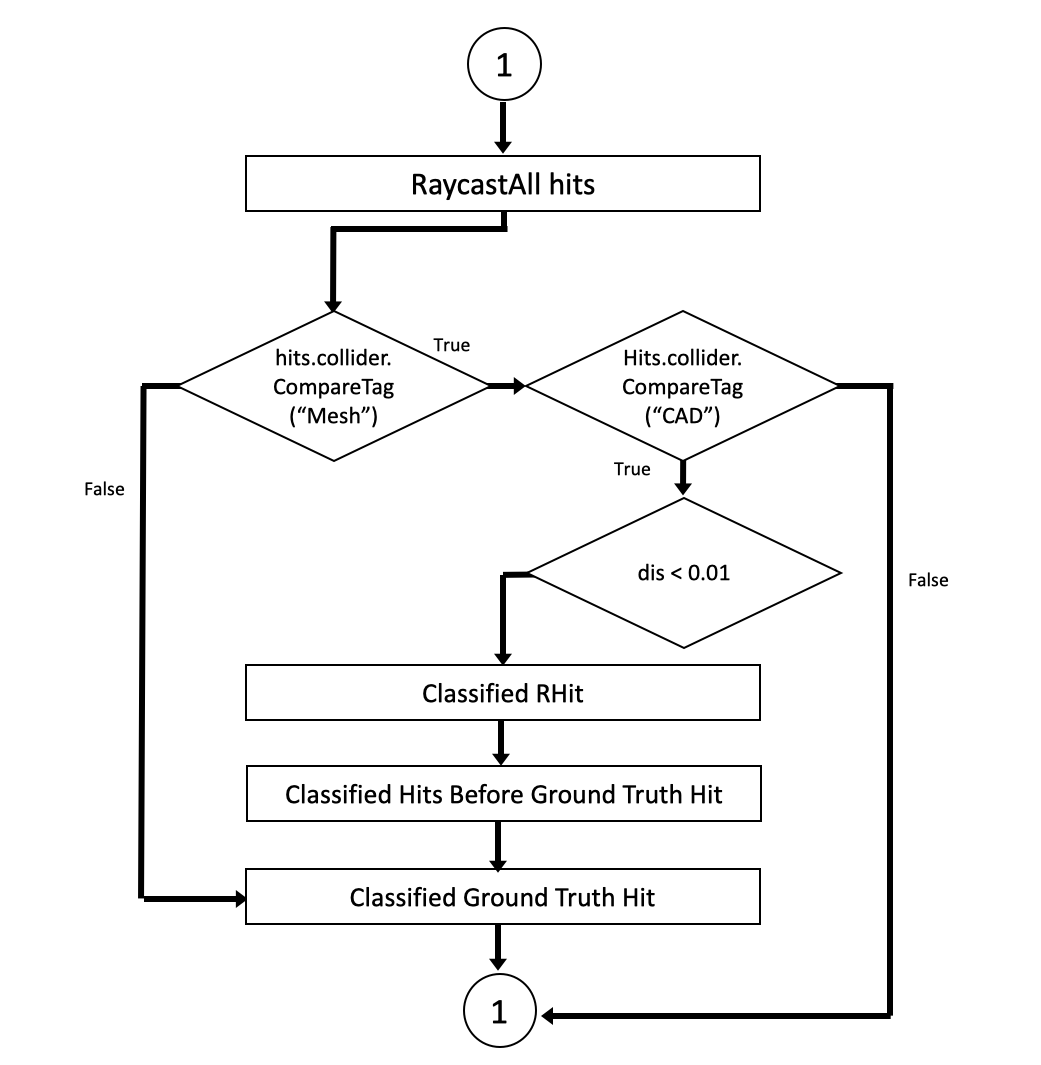


Figure 4.11: Flow Chart of Mesh Data Collecting Algorithm

After collecting the data from the AR scene, analysis part is done to monitor the construction progress of the object of interest. Two types of data are collected, which is point cloud data and mesh data, these relative data will have different approach of performing the analysis part. Firstly, point cloud data collected will have the data of each point cloud position as well as orientation. These data will be processed at the backend of the AR application, where the data is compared with the ground truth data from the CAD model. Then, the relative percentages of each part will also be calculated and recorded. The process after this is being done offline, which is out of the AR application.

As mentioned previously, there are two types of data, which will be handle differently. Point cloud data will be analyze using Open3D as discussed in Section 2.8.3, the library will perform Iterative closest point to match the point cloud data with ground truth data. This is done as a control method, where result will be analyzed and compare where which method suits the best for monitoring the construction progress. In short, Figure 4.12 show the overall flowchart of both of the methods. Data analysis will be discussed in Section 4.2.1

Diagram

Description automatically generated

Figure 4.12: General Flow Chart of Both Methods

### Fitness Level

Fitness level is calculated based on the intersection of CAD model and real object of interest. As discussed previously, there will be a separate analysis of larger part and smaller assembly parts. For example, Figure 4.13 shows the concept of a 2D assembly model, which consist of a bolt and a housing. The device is able to capture the housing fully, but however, only part of the bolt can be captured. This is due to the application is unable to get the data for hidden parts, which result in showing less percentage of intersection. The percentage threshold let the program determine the availability of the part in the system. Therefore, a certain percentage threshold should introduce depends on larger or smaller parts in order to have a more reliable application.

Diagram

Description automatically generated with medium confidence

Figure 4.13: 2D Assembly Model

### Parts Combination

In Section 3.3.2, we discuss about the mesh data that collected from the LiDAR sensor. It will cause a slight problem where the mesh data collected is in a form of blocks and vertices, which does not reflect the actual features of the object. Therefore, certain concept and strategy is needed to solve the issue. The concept of solving the blocks and vertices of the mesh is by combining some of the assembly parts and treat it as one object parts. Figure 4.14 shows the combination of part and renamed it into *GearShaft1.* This only applied for the inner part for the construction progress of this particular object. For the outer parts of the assembly progress will be still a normal procedure, as outer layer is able to distinguish easily by the mesh data provided from the LiDAR sensor. Figure 4.14 shows the assembly part of two ring connector, two bearings, a gear shaft and an adjusting spring (Passfeder). This is to be done to solve the issue of scanning inner part during the assembly process.

A picture containing metalware, screw, gear

Description automatically generated

Figure 4.14: Combination of Assembly Part. (GearShaft1)

## Design of Experiment

In this subchapter, we will discuss the design of the experiment and the concept of analyzing the data collected from the AR application. Section 4.2.1 will discuss the percentage threshold of the system to show the availability of the assembly parts. Then, Section 4.2.2 will discuss the design of experiment method to arrange the data that is collected to proceed to the analysis process. Last but not least, Section 4.2.3 will discuss different scenario of the experiment to be validated.

### Percentage Threshold

Percentage of the collected data is calculated based on the data collected. As discussed previously, only part that are visible by the device are considered in our data. The percentage is calculated based on the intersection of the data with between real and virtual object, which is the true positive value of the data over the total number of ground truth data. For visible part, the AR application will show the output of what the system defined. In this thesis work, there is 41 different parts, but however it is not all visible. The AR application is able to determine available parts by scanning the object through the scene.

The point cloud and mesh data collected is being converted into fitness level, and the AR application will be able to distinguish and make judgement based on the fitness level as follow:

X : Point Cloud/ Mesh Data of part

GT : Ground Truth Data of part

When the data collected reached a certain threshold, the system will register the part as available. However, there are 41 parts in the CAD model itself, where every part is different size and dimension, this may cause a slight problem if the application has been set to a constant percentage threshold. In order to subdue the percentage threshold problem, a different percentage threshold is introduced, where the threshold is set upon the part size of the CAD model. All things considered; there will be two percentage thresholds where it is based on the size of the part as state in Section 3.1.2. It comes with a conclusion that the percentage threshold is state as follow:

Table 4.1: Percentage Threshold Value

|  |  |
| --- | --- |
|  | **Percentage Threshold** |
| Larger Part,  Boundary Magnitude > 2000 | 80% |
| Smaller Part,  Boundary Magnitude < 2000 | 30% |

By using this method to calculate the determined percentage threshold, larger CAD part will have the correct amount of fitness level, therefore, for larger parts (E.g.: Big House and Middle Cover), the percentage threshold is set at 80% to show the availability of the part for the system to decide. In one way or another, the smaller parts would require a lower percentage, since it is most of the time being hidden into the larger parts. As discussed in Section 4.1.4, a good example is a screw that has been inserted into larger part, where a certain part of the object is covered, this is done to provide more robust and resilient results.

### Design of Experiment Method

The key performance index for each method is determined based on the method discussed at Section 2.6.1, which is Tracker Detection Rate. The data is being collected and organize in a template as shown in Table 4.2. The first row in Table 4.2 shows an example of how the data is being recorded. The data will be as followed, where the part name, Nd and NGT, is number of detections of part and number of ground truth parts. %T represent the percentage threshold, and %R is the percentage calculated as follow:

Where = Part detection number,

= Ground truth part numbers

*User Input* is needed to be input manually, which depends on the availability of the part of the testing scene. *System Define* is being input automatically when the percentage is higher than threshold percentage.

Table 4.2: Template of data collection

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test** | **Part Name** | **Nd** | **NGT** | **%T** | **%R** | **User**  **Input** | **System**  **Define** | **TP** | **TN** | **FP** | **FN** |
| 1 | Housing Screw 1 | 200 | 400 | 30% | 50% | X | X | 200 | 0 | 0.5 | 0 |
| 1 | Housing Screw 2 | 100 | 500 | 30% | 20% |  |  | 0 | 100 | 0 | 0 |

The value of TP, TN, FP, FN is being allocated as the following scenario as shown in Table 4.3. For example, when *User Input* and *System Define* is positive, the value from is being recorded in TP. Another example is that when *User Input* is positive and the *System Define* is negative, the value from is being recorded in FN.

Table 4.3: Allocation of TP, TN, FP and FN

|  |  |  |
| --- | --- | --- |
|  | User Input | System Define |
| True Positive, TP | X | X |
| True Negative, TN |  |  |
| False Positive, FP |  | X |
| False Negative, FN | X |  |

For analysis purpose, the percentage used here is not dynamic to show the relationship of the percentage and accuracy of the system definition. The key performance index is accuracy which based on the value of all TP, TN, FP, FN with the formula from state of the art at Section 2.6.1. The analysis will be done offline based on the template and finally a graph is plotted and analyzed.

# 

# Implementation

This chapter discusses the implementation of the automatic recognition of construction progress and the implementation from back-end to the front end of the AR application. There are two methods of capturing the data in this thesis work, which is point cloud capturing method and mesh capturing. Unity is used as an IDE for the application and the marker used for tracking purpose is provided from the institute IPMT. The structure of the thesis work is as shown in Figure 6.1, where the object tracking is being calibrated and implemented, follow by data collection and finally analysis part for evaluation outside of AR application.

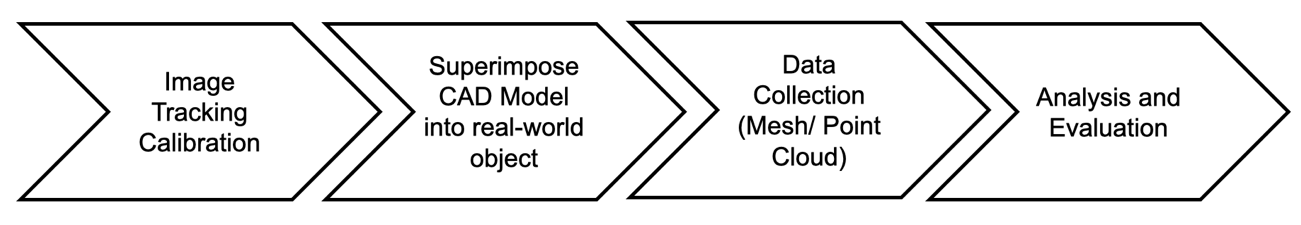


Figure 6.1: Outline of the structure for AR application.

Calibration of object tracking will be discussed in Section 4.1, where the approach of the method will be introduced. Section 4.2 and section 4.3 will be discussing about the methods of point cloud and mesh capturing method in the AR application. Last but not least, the result collected is computed offline and the result will be evaluated in the next chapter.

## Image Tracking Calibration

The AR application uses marker to implement image tracking and therefore, it is first needed to be calibrated. This section will discuss about essential elements of image tracking, which include the reference image library provided from AR foundation which will be discussed in Section 4.1.2. The orientation calibration of the CAD model into real-world scenario will be discussed in Section 4.1.3.

### AR Tracked Image Manager

The AR tracked image manager in Unity is responsible of creating Game Object for the detected image in the real-world environment. The manager is instructed to look for a set of reference image that compiled in a reference image library, which we will discuss in the next sub-section. After the image from the library is detected, the manager will then generate and create a Game Object to the real-world environment. In this thesis work, the manager tracks the marker image and superimpose the CAD model onto the real-world object, which is an engine provided from IPMT. When the image is detected, AR Foundation will create a CAD model to represent on the marker itself. Note that the CAD model is imported from glb file by UniGLTF.

### Reference Image Library

The identification of the object is based on image tracking in the real-world scenario. Firstly, the reference image library is being created in Unity and the physical size of the marker has to be measured. The marker that being used in this thesis is as shown in Figure 2.4 and the size of the marker is as shown as follow in Table 6.1. This parameter is being input into the assets to be used in the AR application. Figure 6.2 shows the implementation in Unity where the physical size settings is being input into the field of the reference image library named *BoxSingleMarker.*

Table 6.1: Physical measurement of marker

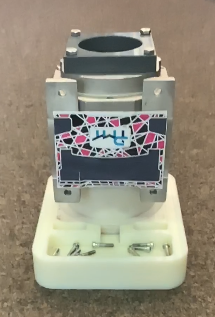
|  |  |
| --- | --- |
| Physical Size (X)/ meters | 0.1340000 |
| Physical Size (Y)/ meters | 0.1010234 |

Graphical user interface, text, application

Description automatically generated

Figure 6.2: Reference Image Library for Marker

The reference image library is then being referred by the AR tracked Image Manager. Once the AR application detect the image on the engine, the CAD model will be superimposed on the real environment engine as shown in Figure 6.3. The material of the CAD Model is set to semi-transparent so that when the device is tracking the object, it is still visible to see the real-world engine in the AR scene.

 A picture containing appliance, stacked, stack

Description automatically generated

Figure 6.3: CAD Model superimposing real-world object

### CAD Model Position and Orientation

After the reference image library is being configured, the CAD model does not stay on top of the real-world object and translation and orientation need to be configure. This is to be done to ensure the virtual object is one-to-one matching and superimpose with the real object in the environment. One of the reasons is the placement of the marker and the orientation of the marker placed on the engine itself. The prefab transformation is being adjusted as shown in Table 6.2. After the adjustment and calibration, the virtual engine is visually paired with the real-world engine and the scene is then ready for the next implementation. As discussed in Chapter 3, the calibration is important because at the implementation phase of the AR application, the data is collected based on the accuracy of the virtual object position to the real-world object position.

Table 6.2: Transformation adjustment of the Virtual Object in Scene

|  |  |  |  |
| --- | --- | --- | --- |
|  | **X** | **Y** | **Z** |
| **Position** | 0 | -0.092 | 0.048 |
| **Rotation** | 180 | 180 | 0 |

### AR Point Cloud Manager

In AR Foundation, the point cloud manager creates a sets of feature points, which is called point clouds. The point clouds consist of 3D location in the world, and each point in the environment can be track between frames. (Figure) shows an e

## Analysis of data accuracy

This section discussed about the accuracy of the data collected using the AR developed AR application. Section 3.2.1 will cover the analysis of the ground truth data for the object of interest. Section 3.2.2 will discuss the analysis of object tracking calibration. Both of the factors are very important to the accuracy of the data collected. The usage of LiDAR sensor will also be discussed and analyze in Section 3.2.3. The assembly of the object part will then be discussed in section 3.2.4.

# Bibliography

[Habe02] Haberfellner, R.; Daenzer, W.F.: Systems Engineering. Methodik und Praxis. Verl. Industrielle Organisation, Zürich, 2002.

[Bashir06] Bashir, F., & Porikli, F.: Performance Evaluation of Object Detection and Tracking Systems. Proceeding from the 9th IEEE Workshop, 2006.

[Behza08] Behzadan, A. H., Timm, B. W., & Kamat, V. R.: General-purpose modular hardware and software framework for mobile outdoor augmented reality applications in engineering. University of Michigan, Am Arbor, 2008

[Belton08] Belton, D. Improving and extending the information on principal component analysis for local neighborhoods in 3D point clouds. Department of Spatial Science, Australia, 2008.

[Bost13] Bostanci, E., Kanwal, N., Ehsan, S., & Clark, A. F. User Tracking Methods for Augmented Reality. *International Journal of Computer Theory and Engineering*, 2013.

[Chen19] Chen,S. How Virtual Reality is Augmenting Realty. The New York Times, United States, 2019.

[Du10] Du, S., Zheng, N., Xiong, L., Ying, S., & Xue, J. Scaling iterative closest point algorithm for registration of m-D point sets. *Journal of Visual Communication and Image Representation*, 2010.

[Dunc00] Duncan, J. S., & Ayache, N. Medical image analysis: Progress over two decades and the challenges ahead. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.

[Erling20] Erlingus. Unity Google Drive. <https://github.com/Elringus/UnityGoogleDrive>, 2020.

[Fard07] Fard, M. G., & Peña-Mora, F. Application of visualization techniques for construction progress monitoring. *Congress on Computing in Civil Engineering, 2007.*

[Fleish05] Fleishman, S., Cohen-Or, D., & T.Silva, C. (2005). Robust Moving Least-squares Fitting with Sharp Features. *Association for Computing Machinery, 2005.*

[Gokb08] Gokberk, B., Dutagaci, H., Ulas, A., Akarun, L., & Sankur, B. Representation plurality and fusion for 3-D face recognition. *IEEE Transactions on Systems, Man, and Cybernetics, 2008.*

[He17] He, Y., Liang, B., Yang, J., Li, S., & He, J. An iterative closest points algorithm for registration of 3D laser scanner point clouds with geometric features, Switzerland, 2017.

[Hida16] Hidaka, N. Polygonization of Point Cloud of Elongated Civil, 2016.

[Hinks13] Hinks, T., Carr, H., Truong-Hong, L., & Laefer, D. F. Point cloud data conversion into solid models via point-based voxelization. Journal of Surveying Engineering, 139(2), 72–83, 2013.

[Jia18] Jia, C. C., Wang, C. J., Yang, T., Fan, B. H., & He, F. G. A 3D Point Cloud Filtering Algorithm based on Surface Variation Factor Classification. Procedia Computer Science, 54–61, 2018.

[Jia19] Jia, C., Yang, T., Wang, C., Fan, B., & He, F. A new fast filtering algorithm for a 3D point cloud based on RGB-D information. PLoS ONE, 14(8), 1–21, 2019.

[Kamat05] Kamat, V. R., & El-Tawil, S. Rapid post-disaster evaluation of building damage using augmented situational visualization. Construction Research Congress 2005: Broadening Perspectives - Proceedings of the Congress, 2005.

[Kazh06] Kazhdan, M., Bolitho, M., & Hoppe, H. Poisson Surface Reconstruction. Johns Hopkins University, Baltimore MD, USA, 2006.

[Klein06] Klein, G. Visual Tracking for Augmented Reality. System, January,182, 2006.

[Klein07] Klein, G., & Murray, D. Parallel Tracking and Mapping for Small AR Workspaces, 2007.

[Lee06] Lee, S., & Pena-Mora, F. Visualization of construction progress monitoring. Joint International Conference on Computing and Decision Making in Civil and Building Engineering, 2527–2533, 2006.

[Linse01] Linsen, L. Point cloud representation. University of Karlsruhe, Germany Technical Report, Faculty of Informatics, 1–18, 2001.

[Lu15] Lu, M., Guo, Y., & Ma, Y. Accelerated Coherent Point Drift for Automatic 3D Point Cloud Registration. College of Electronic Science and Engineering National University of Defense Technology Changsha, Hunan, 2015.

[Mani09] Mani, G. F., Feniosky, P. M., & Savarese, S. D4AR-A 4-dimensional augmented reality model for automating construction progress monitoring data collection, processing and communication. Electronic Journal of Information Technology in Construction, 2009.

[Milgr95] Milgram, P., Takemura, H., Utsumi, A., & Kishino, F. Augmented reality: a class of displays on the reality-virtuality continuum, 1995.

[Rako20] Rakotosaona, M. J., La Barbera, V., Guerrero, P., Mitra, N. J., & Ovsjanikov, M. PointCleanNet: Learning to Denoise and Remove Outliers from Dense Point Clouds. *Computer Graphics Forum*, *39*(1), 185–203, 2020.

[Ruch19] Ruchay, A. N., Dorofeev, K. A., & Kalschikov, V. V. Accuracy analysis of 3D object reconstruction using point cloud filtering algorithms. *CEUR Workshop Proceedings*, *2391*, 169–174, 2019.

[Rusu16] Rusu, R. B., Marton, Z. C., Blodow, N., Dolha, M., & Beetz, M. Towards 3D Point cloud based object maps for household environments. *Robotics and Autonomous Systems*, *56*(11), 927–941, 2008.

[Schall06] Schall, O., Belyaev, A., & Seidel, H. P. Feature-preserving denoising of time-varying range data. *ACM SIGGRAPH: Sketches, 2006*.

[Schm16] Schmalstieg, D., & Höllerer, T. *Augmented reality: principles and practice*, 2016.

[Stuck17] Stucker, C. *Semantic Point Cloud Filtering by*. *July*, 2017.

[Uni19] Unity. *Explore Unity*. unity.com, 2019.

[Van10] Van Krevelen, D. W. F., & Poelman, R. (2010). A Survey of Augmented Reality Technologies, Applications and Limitations. *International Journal of Virtual Reality*, *2010.*

[Wang06] Wang, J., Guo, Y., Ying, Y., Liu, Y., & Peng, Q. Fast non-local algorithm for image denoising. *Proceedings - International Conference on Image Processing, ICIP*, *2006.*

[Yoo18] Yoo, H. W., Druml, N., Brunner, D., Schwarzl, C., Thurner, T., Hennecke, M., & Schitter, G. MEMS-based lidar for autonomous driving. *Elektrotechnik und Informationstechnik*, *135*(6), 408–415, 2018.

[Yoon06] Yoon, S.-H. (2006). A Surface Displaced from a Manifold. *Computer Science and Engineering, Seoul National University, Korea*, 2006.

[Zhou08] Zhou, F., Been-Lirn Duh, H., & Billinghurst, M. Trends in Augmented Reality Tracking, Interaction and Display: A Review of Ten Years of ISMAR Feng. *Center for Human Factors and Ergonomics, Nanyang Technological University, Singapore 2Department*, *Cambridge*, *UK*, 2008.

[Zhou18] Zhou, Q.-Y., Park, J., & Koltun, V. (2018). *Open3D: A Modern Library for 3D Data Processing*, 2018.

# Anhang

#### Zeichnungsstandards am IPMT

