

<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 2. November 2020

# 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](#_Toc55155588)

[Declaration of Authorship III](#_Toc55155589)

[Table of Content IV](#_Toc55155590)

[List of Figures VI](#_Toc55155591)

[List of Abbreviations VIII](#_Toc55155592)

[List of Formula Symbols IX](#_Toc55155593)

[1 Introduction 1](#_Toc55155594)

[1.1 Einführung 1](#_Toc55155595)

[1.2 Aufbau der Arbeit 1](#_Toc55155596)

[2 State of The Art 2](#_Toc55155597)

[2.1 Augmented Reality 2](#_Toc55155598)

[2.1.1 Tracking in Augmented Reality 3](#_Toc55155599)

[2.1.2 Marker based Tracking 4](#_Toc55155600)

[2.2 Construction Progress in Augmented Reality 5](#_Toc55155601)

[2.2.1 Time Efficiency 5](#_Toc55155602)

[2.2.2 Data quality and accuracy 6](#_Toc55155603)

[2.2.3 Completion progress 7](#_Toc55155604)

[2.3 Point Cloud Capturing 9](#_Toc55155605)

[2.3.1 Light Detection and Ranging (LiDAR) 10](#_Toc55155606)

[2.3.2 Photogrammetry 11](#_Toc55155607)

[2.4 Point Cloud Representation 11](#_Toc55155608)

[2.4.1 Meshing 12](#_Toc55155609)

[2.4.2 Polygonization 12](#_Toc55155610)

[2.4.3 Voxelization 13](#_Toc55155611)

[2.5 Noise Filterring and Outliers Removal 14](#_Toc55155612)

[2.5.1 Neighborhood-based methods 14](#_Toc55155613)

[2.5.2 Principal Component Analysis 15](#_Toc55155614)

[2.5.3 Filter point cloud data from RGB information 16](#_Toc55155615)

[2.6 Point Cloud Mapping 17](#_Toc55155616)

[2.6.1 Iterative Closest Point (ICP) 17](#_Toc55155617)

[2.6.2 Coherent Point Drift (CPD) 19](#_Toc55155618)

[2.7 Performance Evaluation Method 20](#_Toc55155619)

[2.7.1 Frame based Metrics 20](#_Toc55155620)

[2.7.2 Object based metrics 21](#_Toc55155621)

[2.8 Resources used for the application 22](#_Toc55155622)

[2.8.1 Unity 22](#_Toc55155623)

[2.8.2 AR Foundation 23](#_Toc55155624)

[2.8.3 Open3D 24](#_Toc55155625)

[2.8.4 Unity Google Drive 25](#_Toc55155626)

[3 Methodology 26](#_Toc55155627)

[3.1 System Requirement 26](#_Toc55155628)

[3.2 Construction Progress Monitoring 27](#_Toc55155629)

[3.3 Point Cloud and Mesh Data Capturing 29](#_Toc55155630)

[3.4 Materials 31](#_Toc55155631)

[3.5 Boundary Condition 31](#_Toc55155632)

[3.6 Point Cloud Capturing Method 31](#_Toc55155633)

[Bibliography 32](#_Toc55155634)

[Anhang 35](#_Toc55155635)

# List of Figures

[Figure 2.1: Representation of the Reality-Virtuality continuum. [Milgram et al., 1995] 2](#_Toc55155636)

[Figure 2.2: Illustration of 3 groups of Augmented Reality devices. [Schmalstieg & Höllerer, 2016] 3](#_Toc55155637)

[Figure 2.2.3: Tracking methods for augmented reality. [Bostanci et al., 2013] 3](#_Toc55155638)

[Figure 2.4: Marker for tracking purpose 4](#_Toc55155639)

[Figure 2.5: Augmented-Reality based progress monitoring. [Lee & Pena-Mora, 2006] 6](#_Toc55155640)

[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](#_Toc55155641)

[Figure 2.7: A sample of a real project progress/inspection report. [Mani et al., 2009] 7](#_Toc55155642)

[Figure 2.8: Progress Deviation Visualize. [Lee & Pena-Mora, 2006] 8](#_Toc55155643)

[Figure 2.9: System process and work flow. [Fard & Peña-Mora, 2007] 8](#_Toc55155644)

[Figure 2.10: Point Cloud dataset of Enschede. [Stucker, 2017] 9](#_Toc55155645)

[Figure 2.11: Point cloud captured of a building. [Hinks et al., 2013] 9](#_Toc55155646)

[Figure 2.12: Basic structure of LiDAR. [Yoo et al., 2018] 10](#_Toc55155647)

[Figure 2.13: iPad camera with LiDAR scanner. (Picture acquired from Apple site) 10](#_Toc55155648)

[Figure 2.14: Point clouds based on feature points of object. 11](#_Toc55155649)

[Figure 2.15: (a) point cloud, (b) control mesh.[Yoon, 2006] 12](#_Toc55155650)

[Figure 2.16: Polygonization of point clouds. [Hidaka, 2016] 13](#_Toc55155651)

[Figure 2.17: Point clouds and binary voxel representation. [Gokberk et al., 2008] 13](#_Toc55155652)

[Figure 2.18: Workflow of FEM model reconstruction. [Hinks et al., 2013] 14](#_Toc55155653)

[Figure 2.19: (Left) Sparse outliers (Right) Point clouds data after outliers removal. [Rusu et al., 2008] 15](#_Toc55155654)

[Figure 2.20: Filtering Process of PCD based on RGB image. [C. Jia et al., 2019] 16](#_Toc55155655)

[Figure 2.21: Process from RGB image, maping, original PCD and filtered PCD. [C. Jia et al., 2019] 16](#_Toc55155656)

[Figure 2.22: Mapping of previous and current feature points. [Klein & Murray, 2007] 17](#_Toc55155657)

[Figure 2.23: Two separate point clouds of a rabbit. [He et al., 2017] 18](#_Toc55155658)

[Figure 2.24: Point cloud mapped after ICP. [He et al., 2017] 18](#_Toc55155659)

[Figure 2.25: Before and after SICP implementation. [Du et al., 2010] 19](#_Toc55155660)

[Figure 2.26: Given two sets of points, assign the correspondences and the transformation that maps one-point set to the other. [Myronenko & Song, 2010] 19](#_Toc55155661)

[Figure 2.27: Type I and Type II error illustration. 20](#_Toc55155662)

[Figure 2.28: Definition of core metric for object-based metric. [Bashir & Porikli, 2006] 21](#_Toc55155663)

[Figure 2.29: Unity XR platform structure. [Unity, 2019] 23](#_Toc55155664)

[Figure 2.30: AR Foundation chart. 24](#_Toc55155665)

[Figure 2.31: 3D data processing: load a point cloud, downsample it, and estimate normals. [Q.-Y. Zhou et al., 2018] 24](#_Toc55155666)

[Figure 2.32: Voxelization from a point cloud dataset in Open3D 25](#_Toc55155667)

[Figure 2.33: Google Drive API 25](#_Toc55155668)

[Figure 3.1: General illustration of AR application 28](#_Toc55155669)

[Figure 3.2: Virtual object superimpose to real-world object. 28](#_Toc55155670)

[Figure 3.3: Point cloud capturing in AR application 29](#_Toc55155671)

[Figure 3.4: Data collection process flow 30](#_Toc55155672)

[Figure 3.5: Capturing mesh data in AR application 30](#_Toc55155673)

List of Tables

[Table 1: Metrics to be computed based on defined quantities 21](#_Toc55155674)

[Table 2: Description of each AR Foundation, ARCore, and ARKit. 23](#_Toc55155675)

# 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.

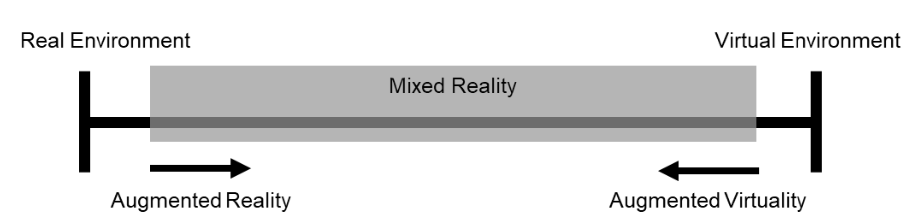


Figure 2.1: Representation of the Reality-Virtuality continuum. [Milgram et al., 1995]

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.

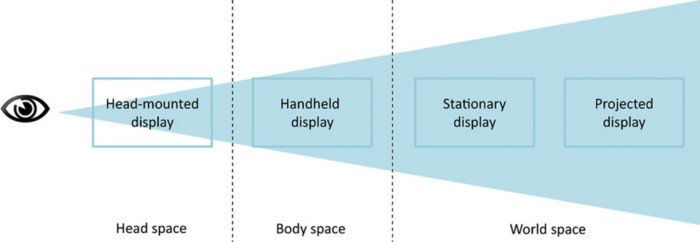


Figure 2.2: Illustration of 3 groups of Augmented Reality devices. [Schmalstieg & Höllerer, 2016]

### 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 2.2.3 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 2.2.3: 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 a 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

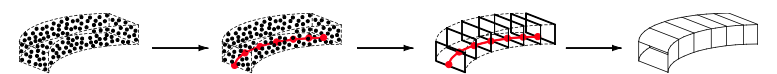
Description automatically generated

(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]

## Point Cloud Mapping

This sub-chapter will discuss about methods used for point cloud mapping. It is important to map the point cloud data to the real world, so that analysis can be more accurate and reliable. In order to have a high precision of mapping, one of the important aspects is to have a good tracking method, which will also be discussed further in this chapter. A good mapping strategy will produce a better analysis for location or object of interest in a real-world environment. Tracking and mapping is separated and run in two parallel task, where one task focuses on tracking to provide robust and stable data collection, and the other produces map of 3D feature point based on previous video frame. [Klein & Murray, 2007] Figure 2.22 shows the example of a point cloud data set matching to the real-world environment by comparing previous frame from the camera itself. This sub-chapter will discuss more deeply into method such as iterative closest point, coherent point drift and intersection over union.

A picture containing indoor, table, computer, desk

Description automatically generated

Figure 2.22: Mapping of previous and current feature points. [Klein & Murray, 2007]

### Iterative Closest Point (ICP)

Iterative Closest Point (ICP) is a registration point-based algorithm that matches two different data points, where the raw measurements such as X,Y,Z values from range images, intensity of points and interest points. [Stewart et al., 2003] This registration algorithm has been used in different application such as medical [Duncan & Ayache, 2000] and engineering field [Nguyen et al., 1999]. There are two steps that iterate through this registration point-based algorithm [Minguez et al., 2006]:

1. Using a fixed estimation, where each point of first image dataset and the closest point of second image dataset is found as a temporary match.
2. By using the constraint formed by the matches, a new estimation is computed.

The process is being repeated until the estimation is stabilized. However, there ae problems when the datasets come in the following circumstances. First, when the data quality of the data sets is relatively low, for example missing point in one image but appearing in the second image. The point will cause a mismatch that affect the overall result of the algorithm and converting the misalignments into a greater one. The goal of the algorithm is to determine a rigid transformation, which is translation and rotation based on two datasets. [He et al., 2017] Figure 2.23 and Figure 2.24 shows two separate point cloud data of two shapes of a rabbit, and the point clouds mapping after the ICP algorithm.

A picture containing drawing

Description automatically generated

Figure 2.23: Two separate point clouds of a rabbit. [He et al., 2017]

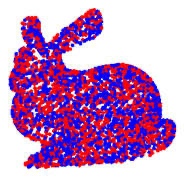


Figure 2.24: Point cloud mapped after ICP. [He et al., 2017]

There is an addition for this registration point-based algorithm, which is called Scaling Iterative Closest Point (SICP) [Du et al., 2010]. SICP take scale factor into the algorithm as well, which can be also represented in the equation as follow:

Where is a scale matrix, is an orthogonal matrix, and is a translation vector. Figure 2.25 shows the image of both before and after SICP algorithm is implemented.

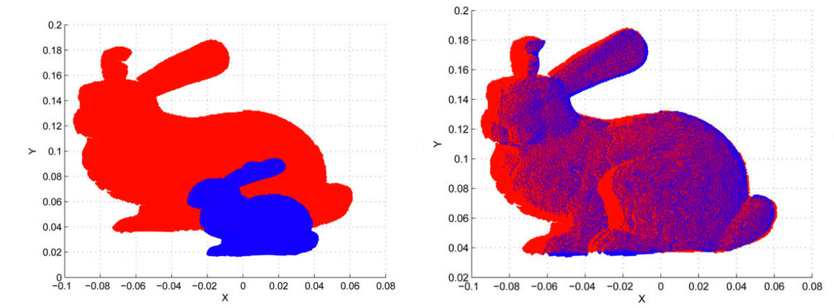


Figure 2.25: Before and after SICP implementation. [Du et al., 2010]

### Coherent Point Drift (CPD)

Coherent Point Drift (CPD) [Lu et al., 2015; Myronenko & Song, 2010] is a an alignment method, which two point sets as a probability density estimation problem. The aim of this method is to assign correspondences between two set of points data. Figure 2.26 shows an example of the problem, where two sets of points is given and alignment is needed to be done. The method is almost similar to ICP itself, which we have discussed in the previous sub-chapter. ICP has a few limitations, where the initial position of two-point sets is require, and both of the data sets should be adequately close in order to achieve a better and accurate result. This method is able to estimate complex non-linear and also non-rigid transformation [Myronenko et al., 2007], which will provide robust result in 3D point cloud mapping even with the presence of outliers and missing points. Coherent Point Drift (CPD) algorithm provide an appropriate solution when it comes to point cloud registration [Lu et al., 2015].

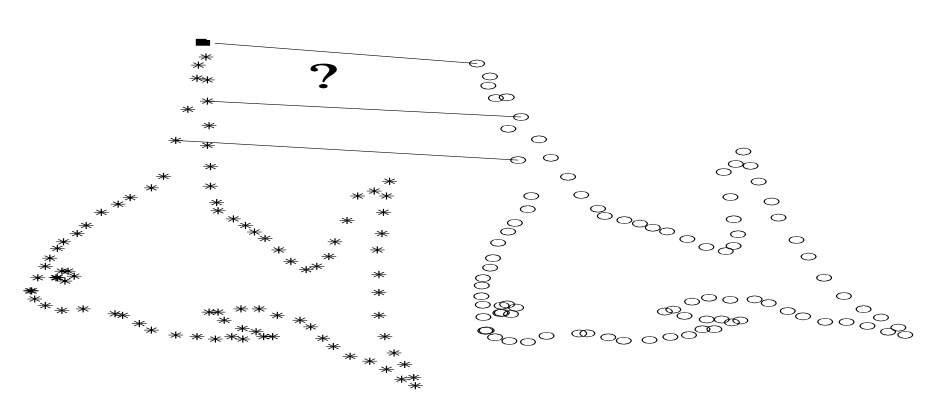


Figure 2.26: Given two sets of points, assign the correspondences and the transformation that maps one-point set to the other. [Myronenko & Song, 2010]

## 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 2.27 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 2.27: 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 2.28 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.

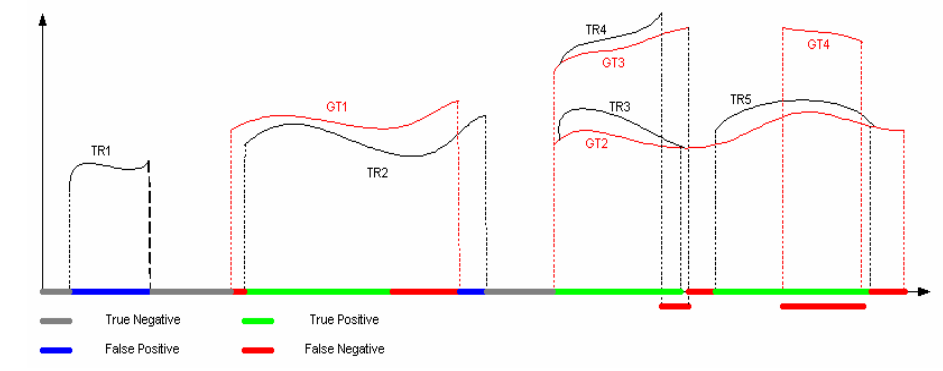


Figure 2.28: Definition of core metric for object-based metric. [Bashir & Porikli, 2006]

## 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.

### 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 2.29 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 2.29: Unity XR platform structure. [Unity, 2019]

### 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 and Figure 2.30 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 2.30: AR Foundation chart.

### 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 consist 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 2.31, 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 2.31: 3D data processing: load a point cloud, downsample it, and estimate normals. [Q.-Y. Zhou et al., 2018]

This library enable 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 2.32 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].

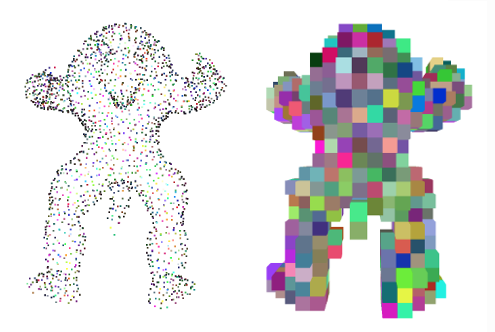


Figure 2.32: Voxelization from a point cloud dataset in Open3D

### Unity Google Drive

Unity Google Drive [Erlingus, 2020] is an API library for listing, searching, uploading, downloading, deleting and exporting files to user’s drive from within Unity game engine. This API Library supports all major platforms such as Windows, Mac, Linux, iOS and Android. Most importantly, it enables users to be able to acquire data wirelessly and accessible through a free data storage on cloud. Last but not least, it is an open source accessible SDK for developer to use for Unity projects. Figure 2.33 shows the interface tab of the SDK itself, which is directly connects to google drive API to enable data transferring within the drive and the Unity game engine. Certain settings and approval need to be done in the Google OAuth consent screen in order to create an access token key.

Graphical user interface, text, application

Description automatically generated

Figure 2.33: Google Drive API

# Methodology

This research is based on a development of method to track construction progress. The master plan of the project as a whole is to enable 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 is then enables the operator to make intuitive decision to determine the overall construction progress. As a result, this AR assistance may increase the productivity for the management planning and commisioning of the construction process for the operating site. This chapter will discuss about the limitation and criterion of the overall AR system. In this thesis work, three major method has been used to compare and analyzed, which is point cloud comparing method, mesh data comparing method, and iterative closest point method. All of the resources and method will be discussed in this chapter.

## System Requirement

By looking at the overall aim of the thesis work, 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. There are several requirements that need to be set for this particular AR application:

* Indoor environment with sufficient artificial lightning
* Enough space that make dynamic movement possible
* Object of interest has been pre-defined
* Object Tracking is assumed to be near perfect
* Hand-held iPad to run the AR application

The environment status is crucial for the AR application to work, as insufficient lightning 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 even crucial in order for the device camera to detect the marker and perform good tracking. Figure 3.1 shows an example of how the lightning 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.

Secondly, as 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. On the other hand, the indoor environment shall consist of enough space to accommodate the movement of 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 AR device and capture the data of the scene.

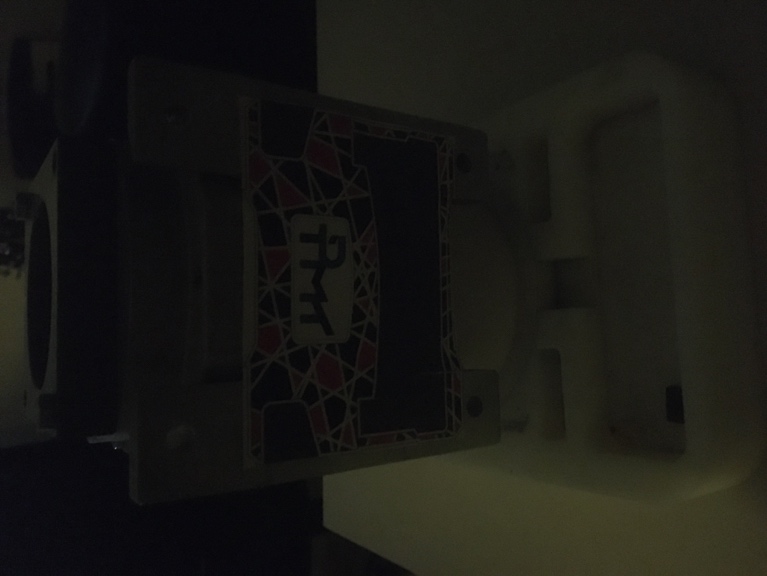
 

Figure 3.1: Low-light environment and Light-sufficient environment

Operator must already acquire an object of interest virtual model to have a general ground truth knowledge comparison with the real model. 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.

The tracking of the object is provided from AR foundation, the marker and virtual object is assumed to be near perfect. This means that the transformation and rotation of the virtual object placed is assumed to be exact position and orientation of the real-world object. The tracking efficiency and algorithm is not covered in the scope of this thesis work.

Last but not least, iPad with LiDAR sensor is being used for the AR application. 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. However, capturing point cloud in this thesis work does not use the LiDAR sensor by iPad. This is due to limitation of the open source library, where the full LiDAR sensor could not be access by the time of this thesis is written.

In conclusion, the requirement for the AR application for this thesis can be formulated as follow:

1. Working space ranges from 1 meter to 3.5 meters.
2. Object tracking is assumed to be nearly perfect.
3. Indoor environment with sufficient lightning.
4. Operator is able to move around the object of interest dynamically.
5. The latest iPad Pro with LiDAR sensor is being used.

## 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 3.1 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 3.2.
4. Based on the object of interest and the virtual object, data is captured
5. Analysis is made based on the data captured.
6. Construction progress percentage will then be presented in other analysis tool.

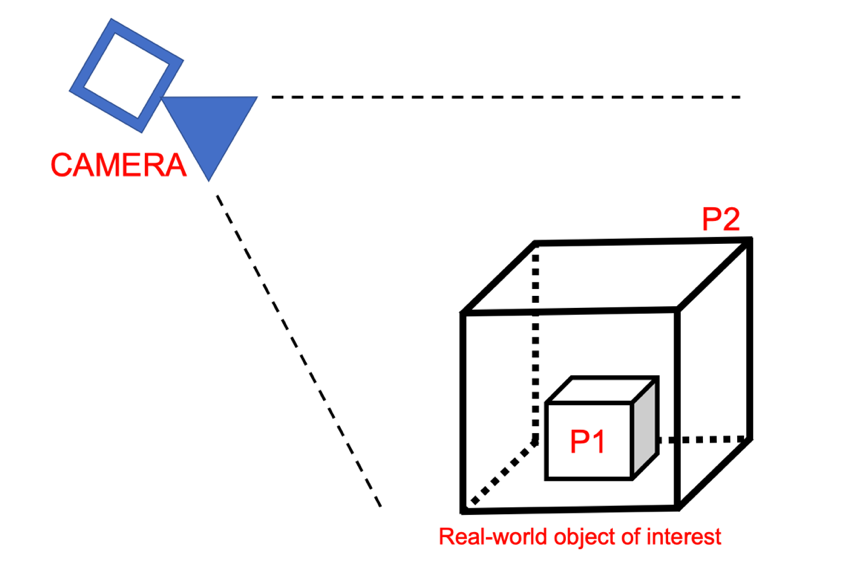


Figure 3.2: General illustration of AR application

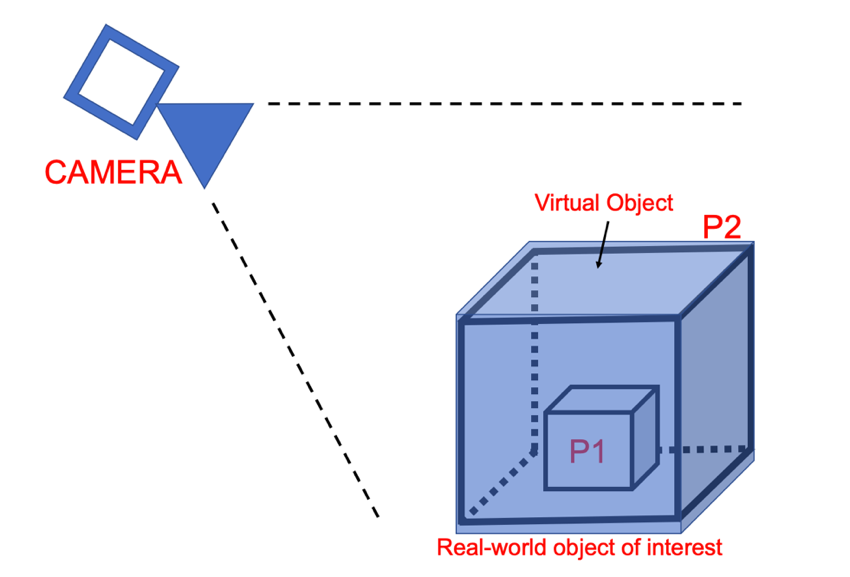


Figure 3.3: Virtual object superimpose to real-world object.

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.

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 3.1 and Figure 3.2, 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.

### Ground Truth Knowledge

Before discussing about ways of capturing point cloud data and mesh data, ground truth knowledge plays a crucial role in this thesis work. 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 3.4 shows the CAD model that has being use in this thesis work. 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 43 separate parts to assemble.

A picture containing light, toy, sitting, table

Description automatically generated

Figure 3.4: CAD model for ground truth knowledge.

Table 3.1: Part name of the CAD Model

|  |  |  |  |
| --- | --- | --- | --- |
| Housing Screw 1 | Pass feeder 1 | Big Housing (57310000) | Bearing (6200) |
| Housing Screw 2 | Pass feeder 2 | Middle Case (57313000) | Bearing (6200) X |
| Housing Screw 3 | Pass feeder 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) | 57397010 |
| Housing Screw 8 | Outer Hexagon Screw 2 | Bearing (6201) | 5739700 |
| Housing Screw 9 | Outer Hexagon Screw 3 | Bearing (6202) | Radial shaft seal |
| Oil Inlet Screw | Outer Hexagon Screw 4 | Bearing (6204) | Adapter |
| Oil Outlet Screw 1 | Oil Outlet Screw 2 | Bearing (6300) |

### 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 has been designed in a good and structured fashion, there will be still position and rotation error and this will lead to inaccurate result. Besides, the application will need developers to firstly determine the initial position and rotation of the virtual object after the marker is detected by the device. 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 all possible error, the object tracking is based on the position of the camera itself with the marker position. Figure 3.5 shows an example of why object tracking calibration is important in the AR application, as the virtual object is not accurately superimpose to the real-world object of interest, it will cause 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 3.1, the implementation of object tracking will be then further discussed in the next chapter.

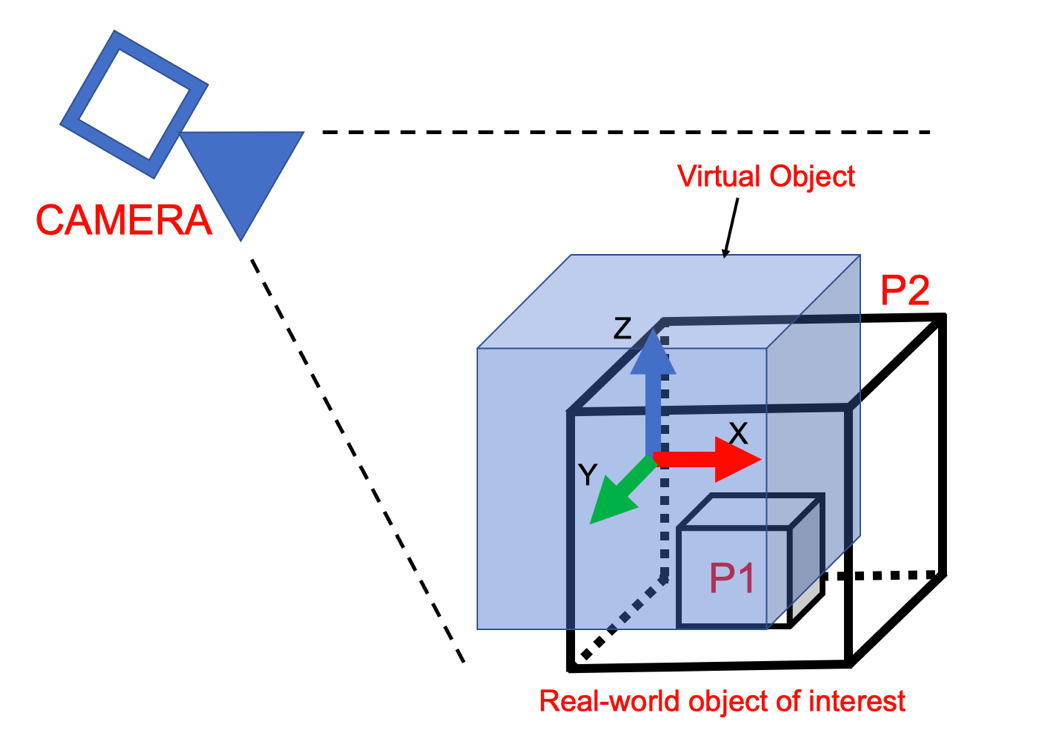


Figure 3.5: Object Tracking Calibration

### Capturing Point Cloud Data and Mesh Data

Figure 3.4 shows the process flow of the data collecting part for the thesis work. This part of the work is done all in the AR application, where the application will detect the marker (As shown in Figure 2.4) which will be used for tracking in AR scene. Then, the object tracking will start by superimpose the CAD model to the object of interest. Based on the feature points or mesh data of the object of interest, the AR application will then collect the point cloud data from the AR scene. While the object tracking is running, the AR application will then detect the intersection of the CAD Model and object of interest, if the feature points is overlapping with the CAD model within a tolerance distance, in this case if it is within 1mm distance, the point cloud data is then stored and save for analyze. The data is then uploaded to an accessible storage for offline analysis, in this thesis work, the data is uploaded to Google Drive. Figure 3.3 shows an example of what being explained, where the green cross represent 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.

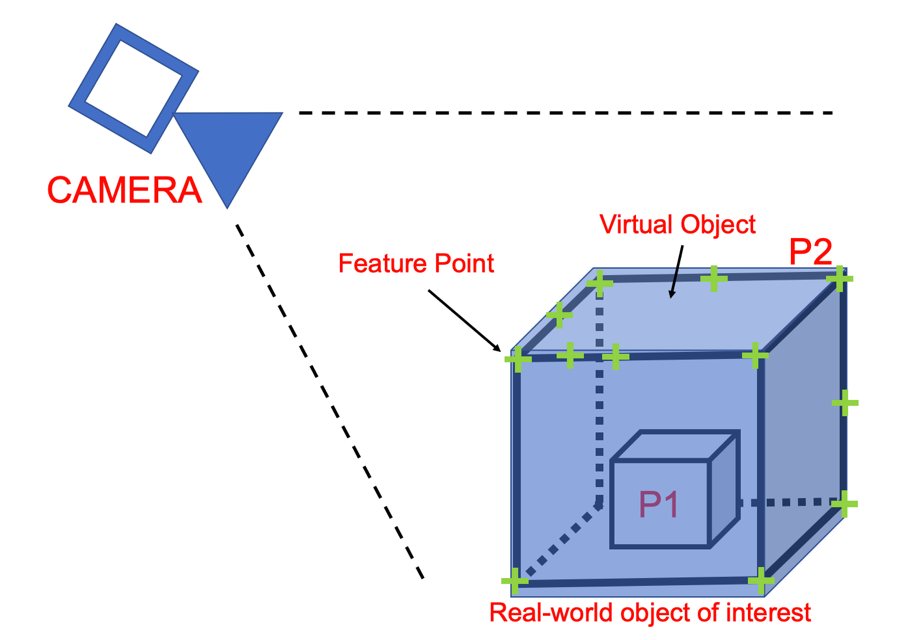


Figure 3.6: Point cloud capturing in AR application

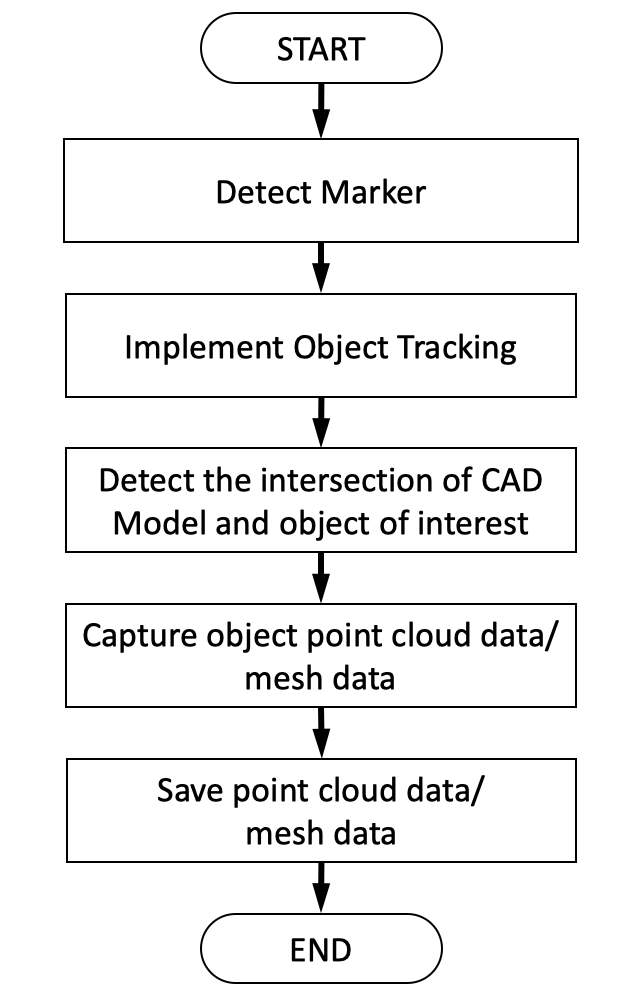


Figure 3.7: Data collection process flow

With that being said, the mesh data is also being collected by the same manner, but different AR scene in the application itself. Instead of capturing feature points from the real world object of interest, the LiDAR sensor will capture the mesh data of the real world object of interest, 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 3.5 shows an example of how mesh data is being collected in AR application.

Diagram

Description automatically generated

Figure 3.8: Capturing mesh data in AR application

### Analysis of data collected

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. Likewise, point cloud data and mesh data results will be compiled and show in a statistical approach as discussed in Section 2.7. Figure 3.8 shows the process flow chart for the analysis part of the thesis work.

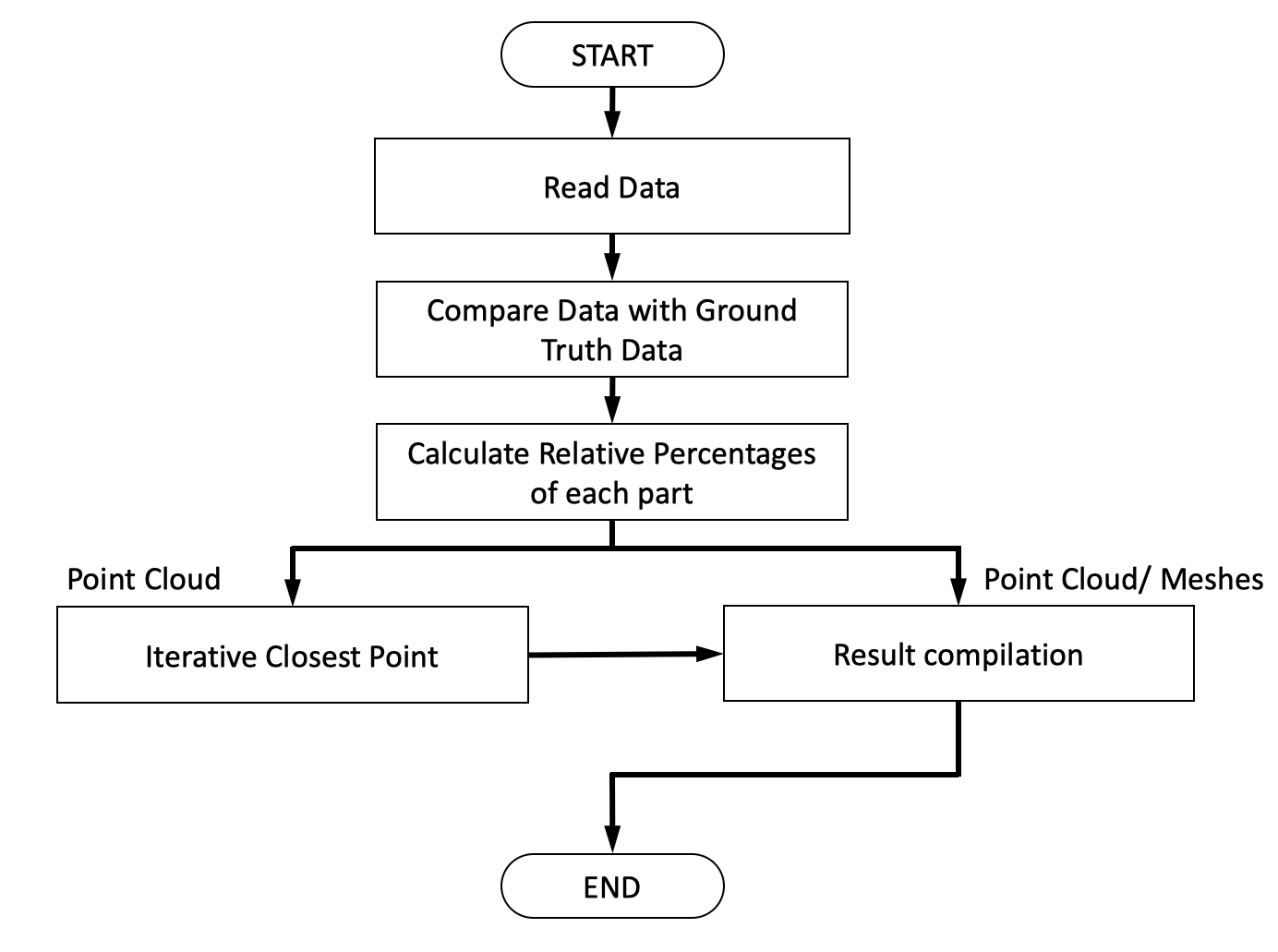


Figure 3.9: Process flow of analysis part.

All things considered with the general process flow of the AR application, the next sub-chapter will discuss more about the library and helping tools for developing the application.

## Materials

This sub-chapter will discuss about materials, library and tools used for this thesis work.

In this thesis work, an (ENGINE) is being used to determine the overall construction progress. (Figure) shows the engine part provided from IPMT, this engine acts as a tool to illustrate the construction progress of an engine building process. Besides, the CAD model of this specific (engine) is also provided as shown in (Figure).

## Boundary Condition

## Point Cloud Capturing Method

# Bibliography

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

Bashir, F., & Porikli, F. (2006). Performance Evaluaion of Object Detection and Tracking Systems. *Proceedings from the 9th IEEE Workshop on PETS*, *Pets*, 7–14.

Behzadan, A. H., Timm, B. W., & Kamat, V. R. (2008). General-purpose modular hardware and software framework for mobile outdoor augmented reality applications in engineering. *Advanced Engineering Informatics*, *22*(1), 90–105. https://doi.org/10.1016/j.aei.2007.08.005

Belton, D. (2008). Improving and extending the information on principal component analysis for local neighborhoods in 3D point clouds. *ISPRS Congress*, 477–484.

Bostanci, E., Kanwal, N., Ehsan, S., & Clark, A. F. (2013). User Tracking Methods for Augmented Reality. *International Journal of Computer Theory and Engineering*, *5*(1), 93–98. https://doi.org/10.7763/ijcte.2013.v5.654

Chen, S. (2019, November). How Virtual Reality is Augmenting Realty. *The New York Times*.

Du, S., Zheng, N., Xiong, L., Ying, S., & Xue, J. (2010). Scaling iterative closest point algorithm for registration of m-D point sets. *Journal of Visual Communication and Image Representation*, *21*(5–6), 442–452. https://doi.org/10.1016/j.jvcir.2010.02.005

Duncan, J. S., & Ayache, N. (2000). Medical image analysis: Progress over two decades and the challenges ahead. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(1), 85–106. https://doi.org/10.1109/34.824822

Erlingus. (2020). *Unity Google Drive*. https://github.com/Elringus/UnityGoogleDrive

Fard, M. G., & Peña-Mora, F. (2007). Application of visualization techniques for construction progress monitoring. *Congress on Computing in Civil Engineering, Proceedings*, *40937*(July 2007), 216–223. https://doi.org/10.1061/40937(261)27

Fleishman, S., Cohen-Or, D., & T.Silva, C. (2005). Robust Moving Least-squares Fitting with Sharp Features. *Association for Computing Machinery, Inc.* https://doi.org/0730-0301/05/0700- 0544

Gokberk, B., Dutagaci, H., Ulas, A., Akarun, L., & Sankur, B. (2008). Representation plurality and fusion for 3-D face recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *38*(1), 155–173. https://doi.org/10.1109/TSMCB.2007.908865

He, Y., Liang, B., Yang, J., Li, S., & He, J. (2017). An iterative closest points algorithm for registration of 3D laser scanner point clouds with geometric features. *Sensors (Switzerland)*, *17*(8). https://doi.org/10.3390/s17081862

Hidaka, N. (2016). *POLYGONIZATION OF POINT CLOUD OF ELONGATED CIVIL*. *December*.

Hinks, T., Carr, H., Truong-Hong, L., & Laefer, D. F. (2013). Point cloud data conversion into solid models via point-based voxelization. *Journal of Surveying Engineering*, *139*(2), 72–83. https://doi.org/10.1061/(ASCE)SU.1943-5428.0000097

Jia, C. C., Wang, C. J., Yang, T., Fan, B. H., & He, F. G. (2018). A 3D Point Cloud Filtering Algorithm based on Surface Variation Factor Classification. *Procedia Computer Science*, *154*(1985), 54–61. https://doi.org/10.1016/j.procs.2019.06.010

Jia, C., Yang, T., Wang, C., Fan, B., & He, F. (2019). A new fast filtering algorithm for a 3D point cloud based on RGB-D information. *PLoS ONE*, *14*(8), 1–21. https://doi.org/10.1371/journal.pone.0220253

Kamat, V. R., & El-Tawil, S. (2005). Rapid post-disaster evaluation of building damage using augmented situational visualization. *Construction Research Congress 2005: Broadening Perspectives - Proceedings of the Congress*, 1251–1260. https://doi.org/10.1061/40754(183)122

Kazhdan, M., Bolitho, M., & Hoppe, H. (2006). Poisson Surface Reconstruction. *Johns Hopkins University, Baltimore MD, USA*.

Klein, G. (2006). Visual Tracking for Augmented Reality. *System*, *January*, 182. http://www.robots.ox.ac.uk/~gk/publications/Klein2006Thesis.pdf

Klein, G., & Murray, D. (2007). *Parallel Tracking and Mapping for Small AR Workspaces*.

Lee, S., & Pena-Mora, F. (2006). Visualization of construction progress monitoring. *Joint International Conference on Computing and Decision Making in Civil and Building Engineering*, 2527–2533. https://doi.org/10.1061/(ASCE)0887-3801(2009)23:6(391)

Linsen, L. (2001). Point cloud representation. *University of Karlsruhe, Germany Technical Report, Faculty of Informatics*, 1–18. https://doi.org/10.1007/s13398-014-0173-7.2

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

Mani, G. F., Feniosky, P. M., & Savarese, S. (2009). D4AR-A 4-dimensional augmented reality model for automating construction progress monitoring data collection, processing and communication. *Electronic Journal of Information Technology in Construction*, *14*(June), 129–153.

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

Minguez, J., Montesano, L., & Lamiraux, F. (2006). Metric-based iterative closest point scan matching for sensor displacement estimation. *IEEE Transactions on Robotics*, *22*(5), 1047–1054. https://doi.org/10.1109/TRO.2006.878961

Myronenko, A., & Song, X. (2010). Point set registration: Coherent point drifts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(12), 2262–2275. https://doi.org/10.1109/TPAMI.2010.46

Myronenko, A., Song, X., & Carreira-Perpiñán, M. Á. (2007). Non-rigid point set registration: Coherent point drift. *Advances in Neural Information Processing Systems*, 1009–1016. https://doi.org/10.7551/mitpress/7503.003.0131

Nguyen, V. D., Nzomigni, V., & Stewart, C. V. (1999). Fast and robust registration of 3D surfaces using low curvature patches. *Proceedings - 2nd International Conference on 3-D Digital Imaging and Modeling, 3DIM 1999*, 201–208. https://doi.org/10.1109/IM.1999.805350

Rakotosaona, M. J., La Barbera, V., Guerrero, P., Mitra, N. J., & Ovsjanikov, M. (2020). PointCleanNet: Learning to Denoise and Remove Outliers from Dense Point Clouds. *Computer Graphics Forum*, *39*(1), 185–203. https://doi.org/10.1111/cgf.13753

Ruchay, A. N., Dorofeev, K. A., & Kalschikov, V. V. (2019). Accuracy analysis of 3D object reconstruction using point cloud filtering algorithms. *CEUR Workshop Proceedings*, *2391*, 169–174. https://doi.org/10.18287/1613-0073-2019-2391-169-174

Rusu, R. B., Marton, Z. C., Blodow, N., Dolha, M., & Beetz, M. (2008). Towards 3D Point cloud based object maps for household environments. *Robotics and Autonomous Systems*, *56*(11), 927–941. https://doi.org/10.1016/j.robot.2008.08.005

Schall, O., Belyaev, A., & Seidel, H. P. (2006). Feature-preserving denoising of time-varying range data. *ACM SIGGRAPH 2006: Sketches, SIGGRAPH ’06*, *June*. https://doi.org/10.1145/1179849.1179919

Schmalstieg, D., & Höllerer, T. (2016). *Augmented reality: principles and practice*.

Stewart, C. V., Tsai, C. L., & Roysam, B. (2003). The dual-bootstrap iterative closest point algorithm with application to retinal image registration. *IEEE Transactions on Medical Imaging*, *22*(11), 1379–1394. https://doi.org/10.1109/TMI.2003.819276

Stucker, C. (2017). *Semantic Point Cloud Filtering by*. *July*.

Unity. (2019). *Explore Unity*. unity.com

Van Krevelen, D. W. F., & Poelman, R. (2010). A Survey of Augmented Reality Technologies, Applications and Limitations. *International Journal of Virtual Reality*, *9*(2), 1–20. https://doi.org/10.20870/ijvr.2010.9.2.2767

Wang, J., Guo, Y., Ying, Y., Liu, Y., & Peng, Q. (2006). Fast non-local algorithm for image denoising. *Proceedings - International Conference on Image Processing, ICIP*, *0*, 1429–1432. https://doi.org/10.1109/ICIP.2006.312698

Yoo, H. W., Druml, N., Brunner, D., Schwarzl, C., Thurner, T., Hennecke, M., & Schitter, G. (2018). MEMS-based lidar for autonomous driving. *Elektrotechnik und Informationstechnik*, *135*(6), 408–415. https://doi.org/10.1007/s00502-018-0635-2

Yoon, S.-H. (2006). A Surface Displaced from a Manifold. *Computer Science and Engineering, Seoul National University, Korea*. https://doi.org/10.1007/11802914

Zhou, F., Been-Lirn Duh, H., & Billinghurst, M. (2008). 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*. https://doi.org/10.1109/ISMAR.2008.4637362

Zhou, Q.-Y., Park, J., & Koltun, V. (2018). *Open3D: A Modern Library for 3D Data Processing*. http://arxiv.org/abs/1801.09847

# Anhang

#### Zeichnungsstandards am IPMT

