

Derivation of Minimum Required Model for Augmented Reality Based Stepwise Construction Assembly Control

by

Nicolas Jeanclos

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Civil Engineering

Waterloo, Ontario, Canada, 2018

© Nicolas Jeanclos 2018

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The global 3D imaging market is expected to reach \$26 billion by 2024 with an annual growth of 23.7% (3D Imaging Market Global Scenario, Market Size, Trend and Forecast, 2015 – 2024. 2018). Various industries are extensively involved in this emergence including the healthcare and entertainment industries, the architecture and construction industries. Additionally, global steel pipe demand is predicted to rise by 3.5% annually until 2019. The combination of the two growths raises the potential of 3D imaging technologies in the construction industry, especially in the piping industry. Thanks to the virtuous cycle between growth and innovation, development and applications of new 3D vision technologies and techniques has become a need for the construction industry facing harsh competition globally. Similarly, prefabrication has been boosted in the construction industry, reducing costs and optimizing time of fabrication. It also copes with the increased demand of small tolerances which sets the industry and its labor under high pressure. Thus, quality control is reinforced in fabrication facilities, and innovations can be deployed in that domain to preclude assemblies from any incompliance. Employing 3D scanners is one effective way to do so, and the recent emergence of handheld laser scanners has created the opportunity to develop efficient new methods to be used for quality control.

This thesis proposes a novel methodology for deriving 3D models for assemblies to be fabricated, breaking down a barrier that previously inhibited the utilization of small-range handheld 3D laser scanners. The process is applicable for industrial assembly lines, which present a stepwise fabrication process such as that for pipe spools. The methodology also aims at streamlining the fabrication flow for workers, and can provide as-built information to the management team. To do so, piping assemblies are thoroughly analyzed and decomposed at each and every step around the weld of interest: one part is being added with respect to the other. From this decomposition of a pipe spool, the challenge of the methodology is to shrink down to the minimum the amount of components that have to be investigated to control the geometry of the assembly. The key concept of solid of revolution is introduced and permits the derivation of the Minimum Required Model (MRM). Examples are generated and experiments are conducted to test the effectiveness of the presented method. This is mainly realized by implementing the algorithm within an in-house software, developed along with another PhD student, a master's student and a co-op student. The software enables the comparison of the acquired scene to the 3D model by segmenting piping components individually, and generating the as-modelled point cloud. Consequently, piping components can directly be segmented within the software, and the MRM can be derived and compared to the expected model.

In order to evaluate the efficiency of the method, three criteria are proposed: (1) the level of spatial complexity between the derived Minimum Required Model and the initial 3D model, (2) the capacity to

use a handheld scanner with or without the MRM, and finally (3) the accuracy of the comparison between the acquired scan and the 3D model.

Acknowledgments

First and foremost, who else could I thank more than Pr. Carl Haas. Two years ago, he gave me that incredible opportunity to come to Canada and study along his side during two years. He trusted me and supported me during all my time in Waterloo. Being one of Pr. Carl Haas student has been very enriching on all sides. His way of managing his students, his research projects and his job as a chair has been a daily model to me.

Secondly, I wish to thank my fellow project mates namely Mohammad-Mahdi Sharif, Caroline Kwiatek and Shang Kun Li (aka Wilson). We spent a lot of time together through those two years and we, altogether, manage to accomplish a great job and make this research project effective. With all of them, I spent awesome moments and shared a lot both professionally and personally. I learned a lot from these conversations and I realize the chance I had to work with three other people from three different countries. Mahdi has been a sort of mentor during my master and, as lab neighbour, he helped me a lot along the completion of my master. He has always been willing to discuss about anything, and dedicated a lot of time to work with me on the publication coming out of this thesis. Caroline and I started this program at the same time and we thus shared personal insights along the time. Finally, it has been such a pleasure to work with Wilson. He spent a lot of time to fill my ignorance of coding and implement my algorithm.

It is a pleasure to thank members of our industrial partner Aecon Industrial West, Scott Waters and Stacey Jenson Rose as well as Chaas Williams from Aecon Cambridge. They are the ones who support this research project and who make it real. Their willingness of applying this research on the field makes this research even more compelling and drives us to work even more, knowing that our concepts could have real industrial applications.

I am also thinking of my friends in Waterloo who filled my master's journey with a lot of fun and amazing moments. This diversity of friend I made here will shape my future. This multi cultural environment has been amazing and is probably the best experience I have been given the chance to live here. All the support from my friends in France was also much appreciated. This experience also proved me how much I care about them and how important they are in my life.

Finally, I want to tell my parents "thank you!" for all they did to me both by different manners. I owe them a lot and I will always remember their dedication during all my time spent as a student. Those two years helped me to realize how important they, and my family in general, matter to me. I think through the physical distance we developed a different relationship. I guess after this time, they realize I am completely emancipated and they can be certain I will be a good person.

Table of Contents

AUTHOR'S DECLARATION.....	ii
Abstract.....	iii
Acknowledgments.....	v
List of Figures.....	ix
List of Tables.....	xiii
1 Introduction.....	1
1.1 Problem Statement.....	2
1.2 Research Objectives.....	3
1.3 Research Approach.....	3
1.4 Scope.....	4
1.5 Thesis Organization.....	4
2 Research Background and Literature Review.....	6
2.1 Piping Fabrication Processes and Quality Inspection.....	6
2.1.1 Current Tolerance Control and Error Detection Within the Piping Industry.....	6
2.1.2 Aecon's Overall Pipe Spool Fabrication Process.....	7
2.1.3 Aecon's Current Tolerance Mapping and Control Approaches.....	12
2.1.4 Aecon's Current Metrics for Quantifying Non-conformance and Rework.....	13
2.1.5 Benefits and Feasibility of Augmented Reality Guided Assembly Framework.....	14
2.1.6 Piping Fabrication in Practice.....	20
2.1.7 Theoretical Implementation of the Stepwise Quality Control.....	25
2.2 Decision Making Method.....	32
2.2.1 Scanners Evaluated.....	32
2.2.2 Multi Attribute Utility Analysis.....	34
2.3 3D Imaging Technologies for Quality Control.....	36
2.3.1 Quality Control Processes in Pipe Fabrication.....	37
2.3.2 Automated Visual Inspection in Construction.....	38
2.3.3 New 3D Vision Technologies in Fabrication.....	39
2.3.4 Knowledge Gap and Problem Statement.....	39
2.3.5 Analogous Problem from Other Industries.....	40
3 Proposed Implementation Workflow.....	42
4 Segmentation of Piping Components.....	46
4.1 Introduction.....	46
4.2 Piping Components.....	46
4.3 Use of 3D Model.....	48
4.4 Generation and Use of PCF File.....	50
4.5 Segmentation of Components.....	53

4.5.1	Segmentation of Pipe and Cylindrical Component	54
4.5.2	Segmentation of Elbow	56
4.5.3	Segmentation of Tee	60
4.5.4	Segmentation of Valve.....	61
4.5.5	Segmentation of Eccentric Reducer	64
4.5.6	Segmentation of Flange	65
4.6	Conclusion	67
5	Derivation of Minimum Required Model	69
5.1	Parsing of the Piping Component File and Selection of Reference and Addition	69
5.1.1	Parsing of the Piping Component File	70
5.1.2	Identification of Reference and Addition.....	70
5.2	Determining the Property of Solid of Revolution for Reference and Addition	74
5.2.1	Solid of Revolution	74
5.2.2	Assessing the Solid of Revolution Property for Reference and Addition.....	75
5.3	Derivation of the Minimum Required Model	77
5.3.1	Analysis of the Structure for Reference and Addition	78
5.3.2	Principle of Geometric Control.....	79
5.3.3	Constituting the Minimum Required Model	81
5.4	Conclusion	83
6	Effectiveness of the Proposed Methodology.....	84
6.1	Designing the Models	84
6.1.1	Database of the 95 Models.....	84
6.1.2	Laboratory Steel Pipe Spool Assembly.....	86
6.2	Impact of MRM on Reducing the Spatial Complexity and Number of Components	87
6.2.1	Experiment on the Database Models.....	87
6.2.2	Experiment on the Steel Pipe Spool.....	88
6.3	Opening New Perspectives to Handheld Laser Scanners	89
6.3.1	Monte Carlo Simulation.....	91
6.3.2	Bounding Box Diagonal Length Analysis	95
6.4	Improving the Scan-vs-BIM Comparison.....	96
6.4.1	Dependency between the Number of Vertices and the Bounding Box Size	97
6.4.2	Comparison of Scanning from a Large vs. Small Volume with Structure IO.....	99
7	Thesis Summary, Conclusions and Future Work.....	108
7.1	Summary	108
7.2	Research Contributions and Conclusions.....	109
7.3	Limitations	109

7.4	Future Work.....	110
8	References.....	111
	Appendix A.....	115
	Appendix B.....	120
	Appendix C.....	131
	Appendix D.....	134

List of Figures

Figure 1: Example of Aecon data flow	8
Figure 2: Example of Aecon physical flow	9
Figure 3: Aecon Industrial West facility layout.....	10
Figure 4: Aecon revision flow	11
Figure 5: Example of a non-conformance report log page 1	15
Figure 6: Example of a non-conformance report log page 2	16
Figure 7: Example of a non-conformance report log page 3	17
Figure 8: Example of a non-conformance report log page 4	18
Figure 9: Example of a non-conformance report log page 5	19
Figure 10: Drawing shelf in the Aecon Industrial West facility	20
Figure 11: (a) Outdoor storage rack and (b) indoor cutting storage rack before being sent to the cutting machine.....	21
Figure 12: Cutting operation performed by an operator	21
Figure 13: Overview of Bay 1	22
Figure 14: Fitting station.....	22
Figure 15: Pipe loaded at a fitting station for being assembled	23
Figure 16: Pipe spool prepared for roll weld	23
Figure 17: Fitting table.....	24
Figure 18: Drawing after the QC personnel inspection	25
Figure 19: Implementation of Structure from Motion in the Aecon's fabrication physical flow	27
Figure 20: Simplified workflow of Structure from Motion	28
Figure 21: Layout of the AIW facility equipped with Structure from Motion	29
Figure 22: Implementation of Structure from Motion in the Aecon's fabrication data flow	30
Figure 23: Implementation of Structure from Motion in the Aecon's fabrication revision flow	31
Figure 24: FARO Sense M70	32
Figure 25: BLK 360.....	33
Figure 26: Artec Eva.....	33
Figure 27: Structure Sensor mounted on an iPad	34
Figure 28: Model compression using translational and rotational symmetries (Mitra and Pauly 2008)	41
Figure 29: Scope of thesis within developed workflow.....	43
Figure 30: SfM Segmentation page	49
Figure 31: Point cloud representing a straight pipe and an elbow	50
Figure 32: Extract of a manually created PCF.....	51

Figure 33: Extract of a PCF file	52
Figure 34: An example Piping Component File: (a) the described component is an elbow, (b) three key-points: two end-points and one center-point, (c) item description where, for instance the piping standard is specified, (d) x-coordinates of the key points, (e) y-coordinates of the key-points, (f) z-coordinate of the key points, (g) diameter about the key-points	53
Figure 35: Representation of a straight pipe	54
Figure 36: Geometric definition of a pipe in a PCF.....	55
Figure 37: Segmented straight pipe in the SfM Software	56
Figure 38: Representation of an elbow in 2D	56
Figure 39: Projection of a point P on a plane in a 3D space	58
Figure 40: Criteria for the elbow.....	59
Figure 41: Segmented elbow.....	60
Figure 43: Skeleton of a tee	61
Figure 42: Segmented tee.....	61
Figure 44: Tee	61
Figure 45: PCF description of a valve	62
Figure 46: bearings coordinate system.....	62
Figure 47: Segmented valve point cloud.....	64
Figure 48: Example of a segmented eccentric reducer point cloud	65
Figure 49: Representation of flanges on a pipe spool assembly	65
Figure 50: PCF description of a flange	66
Figure 51: Segmented flange	67
Figure 52: Bolt holes' alignment control performed by QC personnel	67
Figure 53: Minimum Required Model Workflow.....	69
Figure 54: Demonstration of piping components using key points and the hypothetical line to represent each component: (a) a pipe or any other cylindrical component with 2 end-points, (b) an elbow with two end-points and one center-point, (c) a tee with two end-points, one center-point and one branch-point, (d) a valve with two end-points, one center-point and one branch-point, and (e) a reducer-eccentric with two end-points.....	70
Figure 55: Photography of the steel pipe assembly located in the UW lab. The green oval surrounds the elbow. Each green arrow is pointing at an area of the assembly.	71
Figure 56: Illustration of REF, ADD and connection point on a pipe spool assembly in a fabrication facility. It respectively presents: (1) the original photograph, (2) the extracted assembly being tackled, (3) and the	

corresponding 3D model. (a) Reference consists of a flange, a pipe, and an elbow. (b) Addition is a straight pipe (c) The weld point shared between REF and ADD.....	72
Figure 57: Construction of the graph using the 3D model and the PCF file. (a) 3D model assembly. (b) Constructed graph. Each node corresponds to a key-point.	73
Figure 58: 3D models of SOR and SOR. (a-d) are solids of revolution. (e-j) are not solids of revolution.	74
Figure 59: Combination of: (a) 2 SOR along their axis of symmetry, and (b) 1 SOR and 1 SOR.	75
Figure 60: 2D-representation of a Pipe-Elbow-Pipe assembly (Addition) welded onto an Elbow-Pipe assembly (Reference). Reference is composed of a pipe (CD) and an Elbow (CBA) and Addition is made of a pipe (DE), an Elbow (EFG) and a pipe (GH). The unit vectors represent the normalized vectors connecting two consecutive points.	78
Figure 61: 3D model of the 2D-representation of the assembly presented in Figure 60. ADD is colored in red and REF is colored in blue.....	79
Figure 62: a revolute pair.....	80
Figure 63: Minimum Required Model of 3D model in Figure 61	82
Figure 64: 3D models of five scenario examples per case: components in blue belong to REF and those is red to ADD.....	84
Figure 65: 3D model of the laboratory steel pipe spool.....	86
Figure 66: LOC Reduction between initial 3D models and after applying MRM for (a) Case 1, (b) Case 2, (c) Case 3 and (d) Case 4	88
Figure 67: (a) Scanning range and maximum feasible length for scanning. a: acquisition device, b: only objects within the projected cube can be scanned, c: maximum allowable length for an object to be scanned, assemblies that have a principal length of more than 6.92 (m) cannot be scanned and controlled. d: infrared beams projected to capture the scene. (b) Principal length before and after applying MRM. D1 is the principal length of the assembly before applying MRM. D2 is the principal length of the assembly after applying MRM.....	90
Figure 68: Structure Sensor's depth precision	91
Figure 69: Monte Carlo simulation results using the assemblies from the database with three size categories	93
Figure 70: Comparing MRM utility within the three size groups.....	94
Figure 71: Point cloud in CloudCompare surrounded by its bounding box.....	95
Figure 72: Analysis of the bounding box diagonal length for the database scenarios	96
Figure 73: Monitor used as the center object of the four scans	97
Figure 74: Scans of the monitor without the clutter with different bounding box size: (a) scan 1, (b) scan 2, (c) scan 3, (d) scan 4.	97

Figure 75: Graph representing the number of vertices of a scanned monitor as a function of the scanner's bounding box size. The blue line is the curve binding the 4 data points. The dotted orange line stands for the trend line. Its equation and R2value are also displayed. 98

Figure 76: Assembling the red part on the blue part. The weld is located at the branch of the tee. The green oval surrounds the Minimum Required Model. 99

Figure 77: Segmenting the MRM on Stage 21 of the steel pipe spool assembly in the SfM software 100

Figure 78: Model point cloud derived from the Minimum Required Model 101

Figure 79: "Small" scan acquired with a cube volume of 1m3 101

Figure 80: "Large" scan acquired with a cube volume of 4x4x4m 102

Figure 81: Parallel representation of the two scans obtained after removing the clutter 102

Figure 82: SfM 3 points registration page. On the left, the model point cloud. On the right, the scan point cloud from the "big" scan..... 103

Figure 83: Comparison of model point cloud to scan point cloud from the "large" scan 104

Figure 84: SfM 3 points registration page. On the left, the model point cloud. On the right, the scan point cloud from the "small" scan. 104

Figure 85: Comparison of model point cloud to scan point cloud from the "small" scan 105

Figure 86: Comparison of the two point cloud. The scan from the "big" scan (colored points) is evaluated with respect to the scan from the "small" scan (white points). 106

List of Tables

Table 1: List of non-compliance categories and their descriptions.....	12
Table 2: Summarize of characteristics of different scanners	34
Table 3: Evaluation of the scanners based on their fit for purpose. 3 parameters are taken into account: the speed, the accuracy, and the range. The value is obtained by averaging the 3 parameters.....	35
Table 4: Cost of the hardware required to deploy the scanners investigated.....	35
Table 5: Weight of each attribute for the MAUT analysis.....	36
Table 6: Utility provided to each scanner for each attribute	36
Table 7: List of the principal piping components with a picture	47
Table 8: Piping Class Ratings based on the ASME B16.5 class and the corresponding ISO 7005 Pressure Nominal ratings.....	66
Table 9: All possible cases with respect to ADD/REF being or not being a SOR. The 1-value stands for SOR and the 0-value for SOR.....	82
Table 10: OR gate truth table.....	83
Table 11: LOC value for each type of piping component.....	87
Table 12: Average LOC Reduction for each Case.....	88
Table 13: Result of Level of Complexity for ID_C4_26	88
Table 14: Average LOC reduction for each case in the steel pipe spool assembly	89
Table 15: Total LOC comparison	89
Table 16: Categorization of the assemblies based on average length of pipes in the assemblies and the intended application. The length of pipes in each category was assumed to follow a normal distribution.....	91
Table 17: Distribution of points in the simulation and MRM utility in each group.	94
Table 18: Example of reduction of diagonal length for scenario ID_C1_08.....	95
Table 19: Scans properties	98
Table 20: Comparison of the measured maximum deviation	105

1 Introduction

Investment in innovation in construction companies is very low compared to others. In 2015, only 1% of revenues was spent in R&D in construction whereas 3.5% to 4.5% were spent in the auto and aerospace sectors. As a result, the industry is among the least digitized, and construction projects suffer from cost overruns and major schedule delays. As mentioned in a recent McKinsey report (Agarwal et al. 2016), some innovations could lead to high improvement of the industry operation. The major recommendation is to use higher-definition surveying and geolocation techniques in projects. Deploying 3D imaging techniques can allow high-quality 3D images to be gathered, which can be integrated with any project-planning tool, such as Building Information Modelling (BIM). As stated in a recent report (3D Imaging Market Size, Trends and Forecast. 2018), the 3D imaging market is expected to have an annually growth rate of about 25% until 2022, reaching a market valued at \$21bn. With such a market becoming more competitive, technologies are expected to gain accuracy and resolution with prices dropping down. In addition, industrial construction is categorized as one of the most expensive construction sectors and thus requires rigorous focus to avoid any incompliance. In particular, the piping industry accounts for a big amount in industrial construction. As the demand on infrastructure will keep growing in the next decade, global steep pipe demand is expected to rise 3.5% annually to 79.7 million metric tons in 2019 (Freedonia Group 2013).

By merging all this information together, the potentialities of applying 3D imaging technologies within the piping industry are straight-forward. The enhancement of piping fabrication processes using 3D scanners could have many positive impacts by improving quality and productivity, reducing rework or building up an innovative corporate image. 3D scanners can also be called Light Detection and Ranging (LIDAR) devices and include terrestrial and handheld laser scanners. Due to their light weight, some of most recent ones can be mounted on unmanned aerial vehicles (UAVs). Developing handheld laser scanners in the construction industry has the benefit of not requiring an expensive investment and providing valuable results for comparing as-built assemblies to as-modelled designs, or conducting on-site progress tracking.

In the following thesis, focus is brought on the deployment of handheld laser scanners in piping fabrication shops for assessing the geometric compliance of as-built pipe spool assemblies with regard to 3D Computer-Aided Design (CAD) models. A methodology is extensively developed, which leverages the available design information to produce a reduced 3D design model and takes into account the stepwise fabrication flow for assembling pipe spools. The derived model, named the Minimum Required Model (MRM), is applied in the perspective of 3D vision-based stepwise quality control in a piping fabrication environment.

1.1 Problem Statement

The emergence of modularization and prefabrication in the construction industry has enhanced the industry's performance. Modular construction reduces the duration of the construction phase, saves on construction costs and improves end-product quality (Jaillon and Poon 2014). Given the controlled nature of the factory environment, quality control can be performed more accurately, and contractors are held to a higher standard to achieve strict tolerances. As a result, construction companies have looked to other manufacturing industries – such as the automotive industry – and started to use their measurement processes by adapting them (Jaillon and Poon 2008).

Automation of construction processes in general and modular assembly specifically, would enable contractors to achieve a higher degree of accuracy with respect to geometrical deviations. For instance, automated machines are developed and utilized for welding and cutting (Gnanavel et al. 2017).

Unlike the automotive industry, due to the nature of construction industry, most of the work is still manual and labor intensive. Furthermore, in the U.S, the construction industry has the highest fatality rate across all industries with approximately a thousand deaths in 2015 (Fatal Occupational Injuries by Selected Characteristics, 2003-2014. 2003). The required high skill level coupled with the high accidents rate has led to a major challenge in supplying the required workforce in construction projects (Pan et al. 2012). Focusing on the piping industry, workers need to develop a variety of skills, including interpretation of spatially complicated assemblies from two-dimensional isometric drawings, correctly positioning components and proper welding.

In the piping industry, design files are transmitted in the form of a 2D drawing. These drawings are then printed and distributed to workers on the fabrication floor. Workers are trained to use these 2D drawings to build the 3D model of the design in their minds and then lay out the pipe spools based on their interpretation. Because of the complexity of this task, workers usually develop their own method to better understand (visualize) the drawing. One such method is to use a welding rod to quickly build a 3D model of the assembly. As such, utilization of 3D models (which could be digital or 3D-printed) along with the traditional 2D drawing has been proven to be effective when assembling components together (Goodrum et al. 2016). However, 3D model design during the prefabrication stages has been very limited due to the substantial changes in the internal workflows that it requires.

One of the major bottlenecks in modular fabrication is the inspection process. Whether it uses high end laser scanners, experienced Quality Control (QC) personnel or third party QC firms, it can only be applied at the end of the fabrication process. Moreover, detection of a geometric deviation at the end of the fabrication process can cause major delays and costs to the project (Safa et al. 2015). Recently developed

range cameras can potentially be a solution to this problem since they can offer accurate real time information on the as-built status, which can then be compared with the 3D model (Omar and Nehdi 2016). In addition to their ability to collect real time information, these scanners require a minimum training experience, making them a great solution to be adapted on the fabrication floor. Yet, range cameras have a limited range of acquisition, which is the primary reason of their limited application in the industry (Sharif et al. 2016).

1.2 Research Objectives

The primary objective of this thesis is to develop a framework for generating a Minimum Required Model (MRM) to be used for stepwise 3D imaging quality control during the pipe fabrication cycle. A MRM would create several efficiencies in the fabrication process by minimizing scanning requirements and relaxing constraints.

In addition to this primary objective, some secondary objectives are:

- to propose a methodology fully integrated into a worker-based user-friendly software;
- to overcome the short range limitation of handheld laser scanners;
- to develop the premises of a framework, which enables handheld laser scanners to be completely deployed in piping fabrication shops.

1.3 Research Approach

This thesis focuses on the development of a solution to derive the Minimum Required Model (MRM) in order to facilitate the utilization of handheld laser scanners for geometric control in piping fabrication shops. The developed methodology aims at breaking down the barriers that prevent the utilization of consumer grade scanners in the industry. Such barriers include their small range, and the low point cloud density on large scans.

To do so, the methodology is integrated into a software developed by the research project called *Structure from Motion* (SfM). The solution has been developed for more than two years through a research collaboration between the University of Waterloo and Aecon Industrial West. The main objective of this project is to develop an integrated software package to be used by workers in the fabrication shop to reduce rework and improve productivity by detecting defects before they occur. To do so, the SfM solution uses a handheld laser scanner to scan the as-built assembly, which is then compared to a 3D model of the assembly being fabricated. In addition, the SfM software enables the experimentation and the validation of the methodology. A database composed of about one hundred 3D piping models was created to test out the methodology.

The methodology was executed in a two-phased approach: (1) a method was developed to enable the segmentation of piping components in the appropriate 3D model format; (2) the algorithm for deriving the MRM was conceived and implemented in the SfM software.

Several impact factors were evaluated in order to measure the effectiveness of the proposed methodology.

1.4 Scope

Although the developed methodology could be applied to any kind of industrial assembly, the thesis focuses on industrial assemblies from the piping industry. Some processes as well as fabrication inputs/outputs are industry-oriented. As a result, the flow of piping fabrication is considered to be stepwise, which implies that each assembly is fabricated one step at a time by fitting together one component, or group of components, onto another.

The thesis and its inherent concepts were developed to be used with handheld consumer grade laser scanners and aim at solving the current challenges of those devices. The utilization of the methodology coupled with other acquisition devices such as terrestrial 3D laser scanners is not the original purpose of the methodology and may not be appropriate, because each technology has its inherent challenges which need to be tackled independently.

1.5 Thesis Organization

This thesis is organized into seven chapters:

- Chapter 1 presents the introduction to the problem statement, objectives, approach and scope of the research contained in this thesis.
- Chapter 2 provides a comprehensive overview of the literature and background information. The piping fabrication processes are extensively detailed to position the research in its context. A special attention is given to quality control during the fabrication. Relevant information surrounding 3D imaging technologies used to perform quality control is also provided.
- Chapter 3 presents the proposed methodology. The overall workflow is contextualized and broken down into the key steps involved. Objectives and constraints are also addressed.
- Chapter 4 details the methodology used to segment the piping components. Examples are provided and the process is explained.
- Chapter 5 focuses on the method used to derive the Minimum Required Model and the concepts involved.

- Chapter 6 analyzes the effectiveness of the proposed methodology by assessing the MRM with 3 metrics. The spatial complexity of the assembly, the possibility of scanning and the point cloud density are measured before and after using the methodology.
- Chapter 7 summarizes all the developments of this research into key conclusions. Future work is also discussed in order to strengthen and supplement the proposed methodology and research undertaken in this thesis.

2 Research Background and Literature Review

The following section entails a thorough description of the piping fabrication processes. First, the fabrication of a piping assembly is explained based on site visits. Secondly, the fabrication workflows, which determine the processes are detailed and provided. Finally, the theoretical implementation of the developed software is provided through improved workflows. This presentation enables a precise contextualization of this thesis. In the second part of this chapter, the decision making process which lead to the selection of Structure IO is provided. To conclude, a literature review is achieved and presents the 3D imaging technologies in quality control in the construction industry.

2.1 Piping Fabrication Processes and Quality Inspection

In order to fully understand the applications of the developed thesis, a detailed presentation of the piping fabrication processes is required. Therefore, a realistic presentation of piping fabrication is given in the next part, supported by in-shop pictures. Following it, the fabrication workflows are analyzed, and the potential implementation of the developed solution is explained.

2.1.1 Current Tolerance Control and Error Detection Within the Piping Industry

During fabrication, all materials and components are assumed to have specific dimensions, and the locations of elements are dimensioned on the isometric drawings to a theoretically exact position relative to one or more datum points. In reality, these fabricated dimensions and locations vary somewhat. The contractually acceptable amount of variance is the tolerance for that specific measure. Tolerance specifications assure installation of a system within a reasonable degree of accuracy. For specifying tolerances, designers and QC specialists typically use standardized dimensional tolerance guidelines. For any contract, the designer must clearly state: (1) the tolerances allowed, (2) the standards used, (3) how compliance will be verified, and (4) what the result of noncompliance will be. PFI ES-03 has been selected as the dimensional guideline standard for the purpose of the *Structure from Motion* project since it will cover more than 80% of the contractual obligation that the Aecon Industrial West (AIW) fabrication facility is interacting with. Currently, the predominant processes for monitoring the critical dimensions outlined in these standards involve manual assessment by certified QC personnel using direct contact measurement devices such as metal measuring tapes, calipers, custom gauges, squares, and straight edges. No measurement process is exact, and this uncertainty needs to be taken into account when verifying compliance with any allowable tolerance. Measurement uncertainty is simply a quantified doubt about the result of a measurement. Uncertainties and errors in measurement can come from many sources, such as the measuring device, the component being measured, the skill of the craft worker or inspector performing the measurement, the measurement process, and the environment. For example, when taking a simple distance measurement with a measuring tape, a number of things can go wrong: (1) the tape could be mismarked or the end hook could

be out of position, (2) the tape sags during the measurement, (3) the tape may not be perfectly aligned with the desired axis of measurement, (4) the craft worker measuring might have poor eyesight, or be making the reading in dim lighting conditions, or (5) the craft worker may round, adopting a “close enough” attitude.

In addition to measurement uncertainty, dimensional control problems also originate from existing poor design document creation practices. These include:

- Chain dimensioning establishes ambiguity and the potential for accumulated measurement error - e.g., dimensioning a spool’s multiple nozzles centre to centre instead of to a single base point.
- Problems can arise when copying values from guidelines or standards while neglecting to communicate if they are maximum or minimum values of an allowable range.
- Less important dimensions may have a plus-or-minus sign as a suffix to indicate that the dimension can vary, but the amount of the allowable variation is not clear.
- Errors can result from using units that the fabricator does not typically operate with.

Adhering to tolerance specifications can take on an additional layer of complexity when practical cost-benefit concerns are integrated into the decision making. For example, under a strict interpretation of the contract, a contractor would be required to re-fabricate an entire pipe spool assembly because it exceeded the specified tolerances by a quarter inch, but this could seriously delay construction progress, lead to litigation, make for a negative and adversarial work environment, and increase costs unnecessarily. For most large projects, the final tolerance inspection is performed by the regulatory agencies.

For a more in-depth discussion of tolerances in pipe spool fabrication and construction in general, see (Thielsch 1965, Markowitz 1973, Ballast 2007, Antaki et al. 2005) as well as ISO 4463-1:1989 2012 and ASME B31.1-2014 2014).

2.1.2 Aecon’s Overall Pipe Spool Fabrication Process

This section provides Aecon workflows in terms of: (1) data, (2) physical, and (3) revision. Data workflow corresponds to the general workflow associated with data transmission in AIW (Aecon Industrial West). The physical flow is how different process in the shop are located relative to each other. The revision workflow focuses on quality control and the drafting process for spooling. It also helps in integrating SfM in Aecon’s work flows.

Aecon Fabrication Data Flow (3032)

Version 0.0

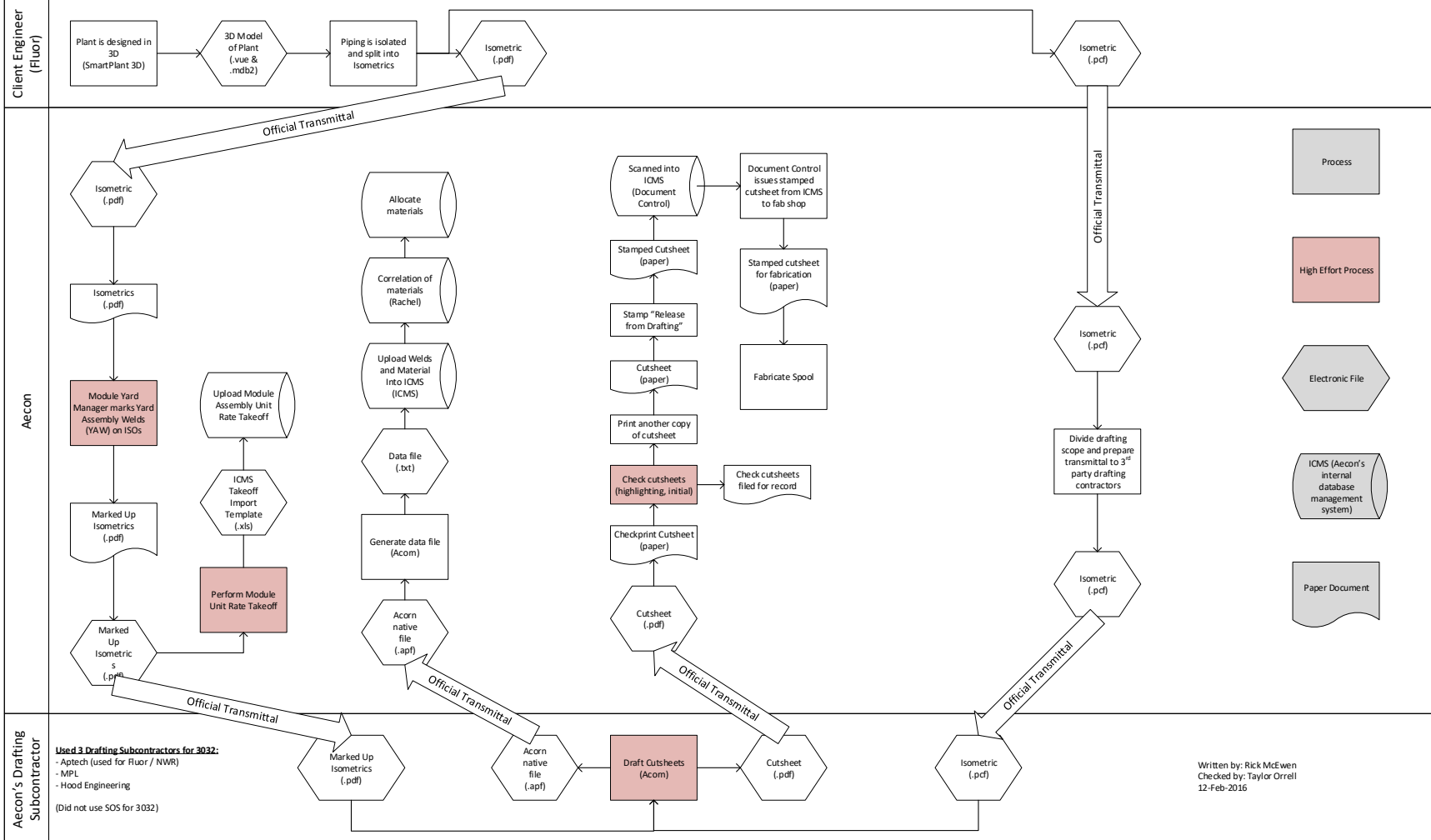


Figure 1: Example of Aecon data flow

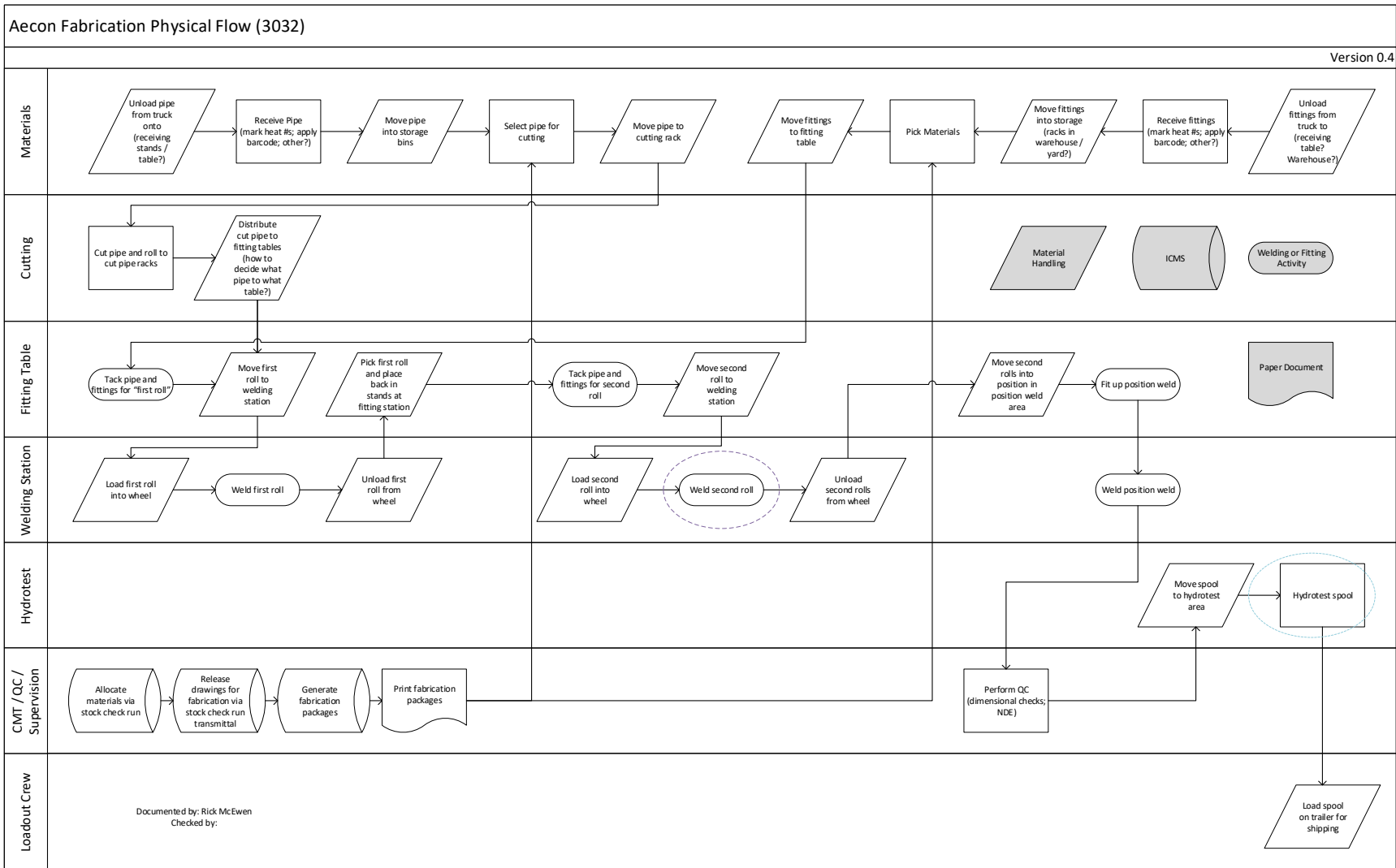
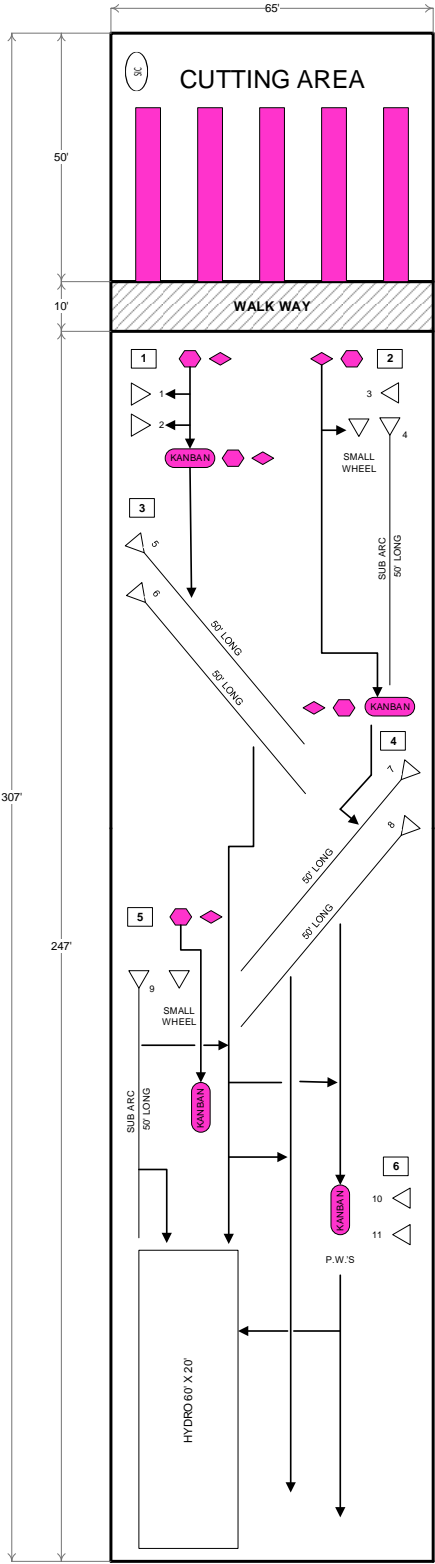







Figure 2: Example of Aecon physical flow

Layout "TO BE" General & Equipment Modules

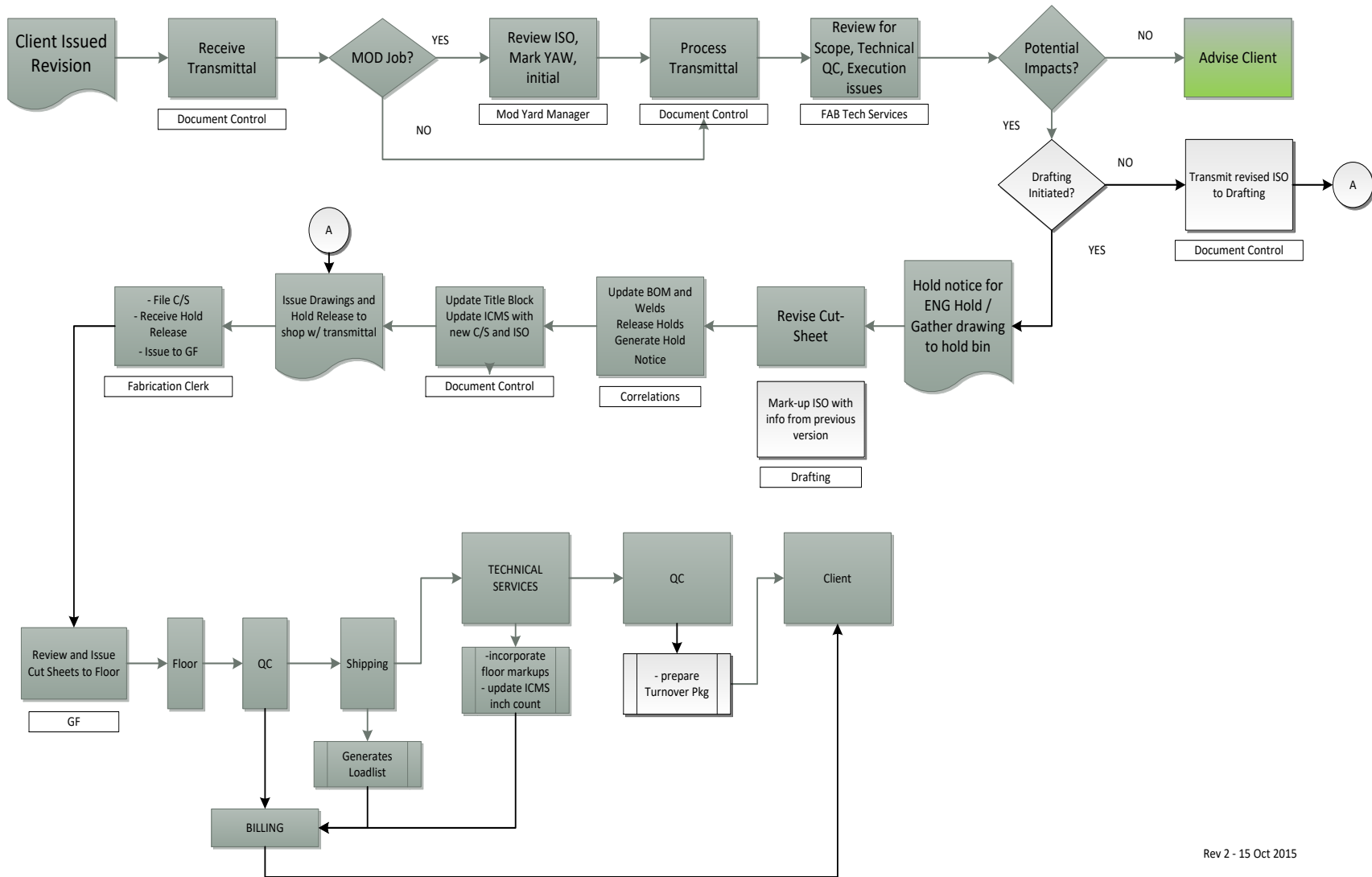


-  FITTER TABLE (6)
-  WELD STATION (11)
-  FITTINGS (RAW)
-  PIPE (RAW)
-  KANBAN

NOTE: 1. DRAWN TO SCALE
2. ALL DIMENSIONS ARE IN FEET

Figure 3: Aecon Industrial West facility layout

Revisions



Rev 2 - 15 Oct 2015

Figure 4: Aecon revision flow

2.1.3 Aecon’s Current Tolerance Mapping and Control Approaches

Occasionally, some work will be performed that upon examination will be found to be out of tolerance. This is an instance of non-compliance with the contract. Dimensional non-compliance or defects are primarily caused by pipe fitter error or by the thermal expansion and contraction of metals during welding. The degree of distortion depends on the metal or alloy, its size, shape, thickness, the tacking and alignment, the welding process, procedure, and sequence, the care taken by the welder, and the positioning of the welded components in the design. Regardless of whether the defect originates from fitting or welding, dimensional examination usually does not occur until after welding, i.e., after the components have been permanently joined. Remediation of this type of non-compliance is expensive; more-so if the dimensional defect goes undetected until final installation on-site. The following table illustrates a list of non-compliance categories and their descriptions.

Table 1: List of non-compliance categories and their descriptions

Non-Compliance Categories	Description
Weld defect	Due to the welding the size, shape, orientation, location, or properties do not meet specified acceptance criteria
Dimensional defect	A dimensional discrepancy greater than the acceptable tolerances stated by the applicable code or contract
Drawing error	An error or omission on drawings used for fabrication
Pressure test failure	The failure of a pressure test. Mechanical failures of equipment are not considered cause for an NCR but rather a retest
Fitting error	A dimensional discrepancy greater than the acceptable tolerances stated by the applicable code
Material defect	A defect in parent material which exceeds allowable tolerances and specifications of the applicable code
Customer error	Customer error resulting in a non-compliance
Wrong material	Material used in the procedure not as per required on the welding specification

Wrong WPS used	Weld Procedure Specification used on production is not on the approved procedure list
Damage part – general	Damage to production parts
Machining error (by fabricator)	Machining completed by fabricator which falls outside that allowed by the applicable code or specification
Supplier/Subcontractor error	Defect or error incurred by supplier or subcontractor which impacts the compliancy of the component
Painting defect	Defect in coating systems which is not compliant to applicable code
Contamination	Material has been contaminated either through contact or improper packaging and requires rework
Damaged part – flange face	Flange face does not meet the requirements per the applicable code
Damaged part – ball value	Ball value does not meet the requirements per the applicable code

2.1.4 Aecon’s Current Metrics for Quantifying Non-conformance and Rework

A rework activity is defined as an activity that is completed more than once or that removes work previously installed/created to correct non-conformance between the design and the as-built condition. Rework falls into two categories: corrective activities that are reported and tracked and unreported corrective activities that are performed by a craft worker during the fabrication of the component. This unreported rework is challenging to quantify as it is unreported. In heavy industrial projects, the highest impact on rework is caused by design error and omission (TFRF = 0.009) and owner change (TFRF = 0.005), while construction error has a TFRF of 0.002. TFRF is the total field rework factor and is defined as the total direct cost of field rework divided by the total construction phase cost (Hwang et al. 2009).

When non-conformance is detected in the work completed during an inspection at an Aecon facility, the inspector completes a one-page non-conformance report to document the incident. The report details the incidence of non-conformance, the proposed disposition, a statement on acceptance of the proposed disposition, and completion of disposition.

The inspector provides the identifying information about the spool being discussed in the report, including a NCR number, a spool number and a job number.

The inspector describes the non-conformance and recommends an action from the following choices: scrap, rework, use as is, and return. The inspector provides a root cause of the non-conformance.

If applicable, the inspector then provides a proposed disposition that outlines a proposed solution to the non-conformance situation. The non-conformance report and its current state is provided to the client for review and approval (Figure 5, Figure 6, Figure 7, Figure 8, and Figure 9).

Once the work outlined in the disposition is completed and recorded, the final signatures can be applied to the original non-conformance report, and the incident of non-conformance has been resolved.

2.1.5 Benefits and Feasibility of Augmented Reality Guided Assembly Framework

In a study conducted by CII research team titled “Innovative Delivery of Information to the Crafts”, the research team found that there is a significant improvement in craft performance when 2D information is augmented with the corresponding 3D information. Their field experiments provided evidence of the benefits of providing 3D information, including:

- A 16-percent improvement in time to completion using a two-sided isometric.
- A 39-percent reduction in the number of errors using a two-sided isometric.
- An 11-percent increase in direct work using a two-sided isometric.
- An ROI of \$33.47 saved for each dollar invested in the use of two-sided isometric drawings.

(For more information on 2-Sided isometrics please refer to (Goodrum et al., 2016)).

The University of Waterloo team was able to deliver an electronic version of the two-sided isometric with augmented reality features which is foreseen to provide an immediate efficiency gain from employing mobile technology to diffuse information to the craft workers, earlier than expected. This task was planned to be completed on the third phase of the project. The prototype is able to display the isometric drawings and 3d model of the selected pipe spool. The user is being able to navigate between these two screens by swiping the tablet screen or display both of them side by side. The user is being able to manipulate the 3d model using the tablet touch screen (a capability beyond the static 3D shape used in the aforementioned study).

NON-CONFORMANCE REPORT (NCR)

		<input checked="" type="checkbox"/> Shop <input type="checkbox"/> Field Comf:	
ISSUED TO: Aecon FAB Shop		NCR No.: 3032-F01-NCR-021	
DEPT / PLANT / LOCATION: SWP FAB SHOP		DATE: 28 OCT 15	
TAG No / LINE No.: 10-20-ISO-PL-3259-002		JOB NO.: 3032-F01- Fluor/NWR Unit 20	
DESCRIPTION OF NONCONFORMANCE: Orifice flange on SCN 60085 installed incorrectly, tap installed North 45 deg down, and not South 45 deg down as required per Isometric.			
RECOMMENDED ACTION: <input type="checkbox"/> SCRAP <input type="checkbox"/> REWORK <input checked="" type="checkbox"/> USE AS IS <input type="checkbox"/> RETURN			
ROOT CAUSE Misinterpretation of detail	INSPECTOR Tammy Murphy	SIGNATURE <i>T Murphy</i>	DATE 28 OCT 15
PROPOSED DISPOSITION (Completed by addressee) The spool containing the counterpart to this orifice flange has not yet been fabricated (SCN 60084). Aecon proposes, to avoid the risk of damage to the flange during rework, to leave spool 60085 as is, and revise the cut sheet for spool 60084 to show the tap at South 45 deg down, which would avoid interference with the tap on 60085.			
NAME: Tammy Murphy		SIGNATURE: <i>T Murphy</i>	DATE: 28 OCT 15
ACCEPTANCE OF PROPOSAL			
CLIENT APPROVAL: <input checked="" type="checkbox"/> YES <input type="checkbox"/> NO		AUTHORIZED INSPECTOR: <input type="checkbox"/> YES <input checked="" type="checkbox"/> NO	
	SIGNATURE	PRINT	DATE
Q.C REPRESENTATIVE:	<i>T Murphy</i>	Tammy Murphy	28 OCT 15
Fluor Representative:	<i>[Signature]</i>	BRENIE KADICKI	23 Nov 2015
NWR Representative:	<i>[Signature]</i>	BRIAN M. GARNEY	NOV 23/2015
COMPLETION OF DISPOSITION (ACTION COMPLETED): <i>Spool 60085 use as is</i>			
COMPLETED: <i>T Murphy</i>		PRINT: <i>Tammy Murphy</i>	DATE: <i>23-Nov-2015</i>
FINAL SIGN OFF	SIGNATURE	PRINT	DATE
Q.C REPRESENTATIVE:	<i>T Murphy</i>	Tammy Murphy	23-Nov-2015
Fluor Representative:	<i>[Signature]</i>	Dennis [unclear]	NOV 23/2015
NWR Representative:	<i>[Signature]</i>	BRIAN M. GARNEY	NOV 23/2015
DESIGN ENGINEER: (if applicable)	<i>[Signature]</i>	<i>[Signature]</i>	<i>[Signature]</i>

Figure 5: Example of a non-conformance report log page 1



Re: 3032-F01-NCR-021 
 Dave Arthurs  Dwayne Faulkner, Bernie Knierim

11/23/2015 01:26 PM

Dwayne / Bernie -

There is no impact on instrumentation functionality of the flange taps are on north or south side in orientation provided we have one tap on the upstream (high pressure) side of the orifice plate and one tap on the downstream (low pressure) side of the orifice plate. Additionally in this instance, the taps should be below horizontal per standard installation in liquid service.

I recommend moving forward with the proposed solution to the NCR to have the tap on spool 60084 located South at 45° down.

Cheers

Dave Arthurs, C.E.T., B.Mus | **FLUOR CANADA LTD.** | Senior Designer / Field Engineer – Control Systems |
 dave.arthurs@fluor.com | IO DC 40.4870 | Phone: +1.(780).640.2853 ext/ 5032 | www.fluor.com

Dwayne Faulkner Dave, As discussed prior, please see the attach... 11/23/2015 01:14:32 PM

From: Dwayne Faulkner/CA/Contr/FluorCorp
 To: Dave Arthurs/CA/FD/FluorCorp@FluorCorp,
 Date: 11/23/2015 01:14 PM
 Subject: 3032-F01-NCR-021

Dave,
 As discussed prior, please see the attached NCR for your review of the proposed disposition.

Regards
 Dwayne Faulkner | **FLUOR CANADA LTD.** | QA, Mod Yard – NWR Project |



Dwayne.Faulkner@fluorconstructors.com | Cell Phone 780.446-4334 3032-F01-NCR-021.pdf

Figure 6: Example of a non-conformance report log page 2

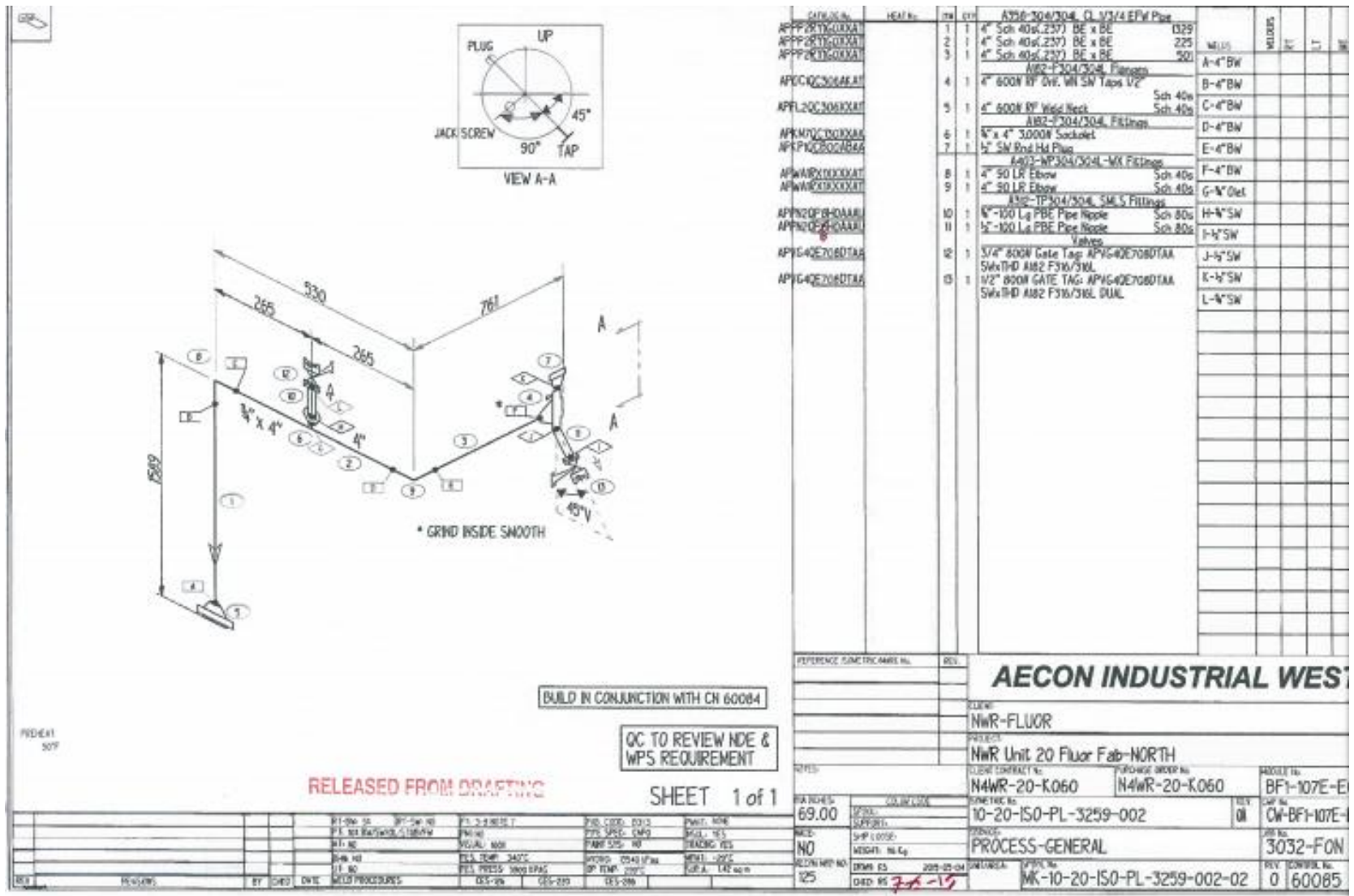


Figure 7: Example of a non-conformance report log page 3

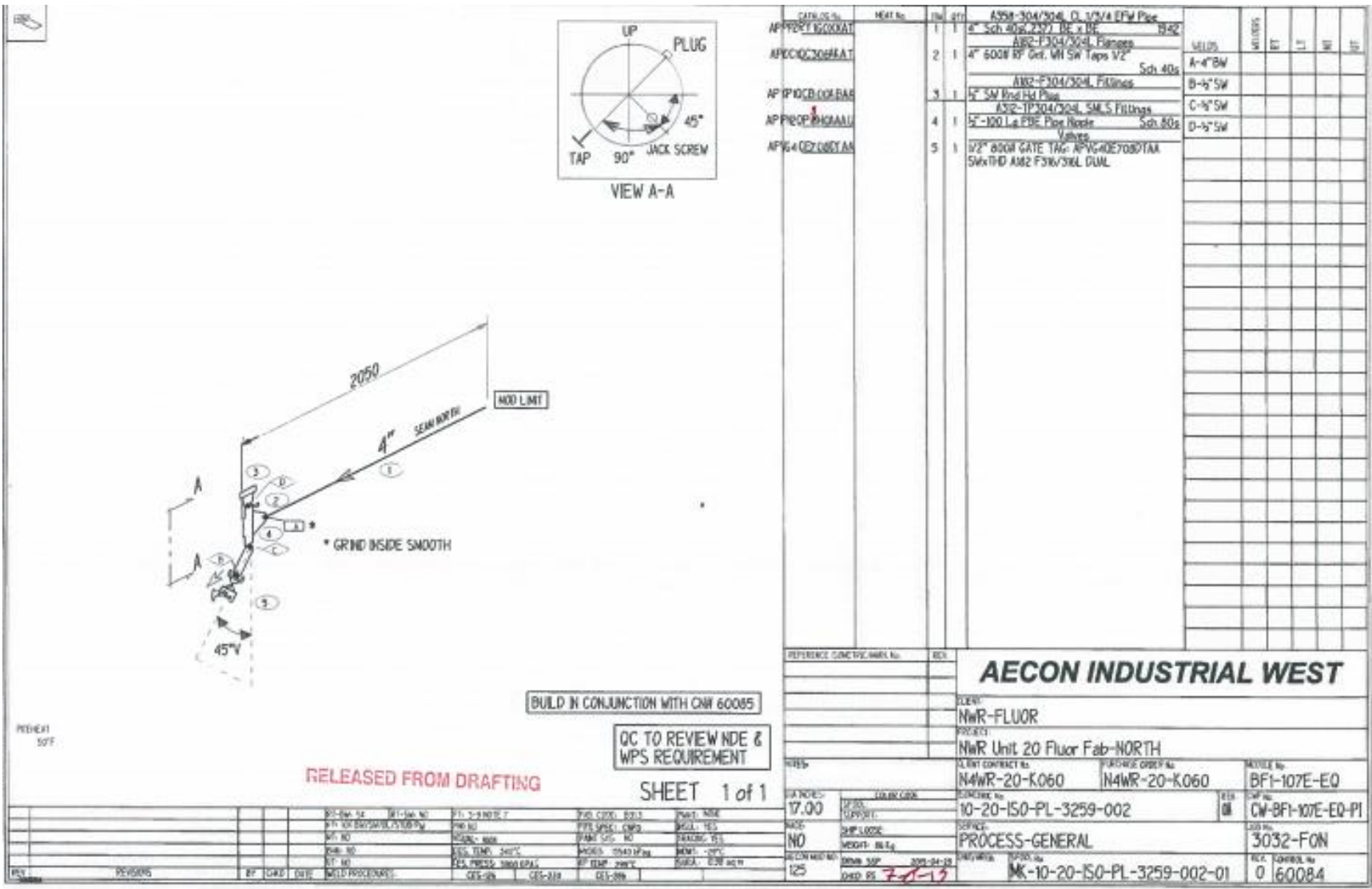


Figure 8: Example of a non-conformance report log page 4

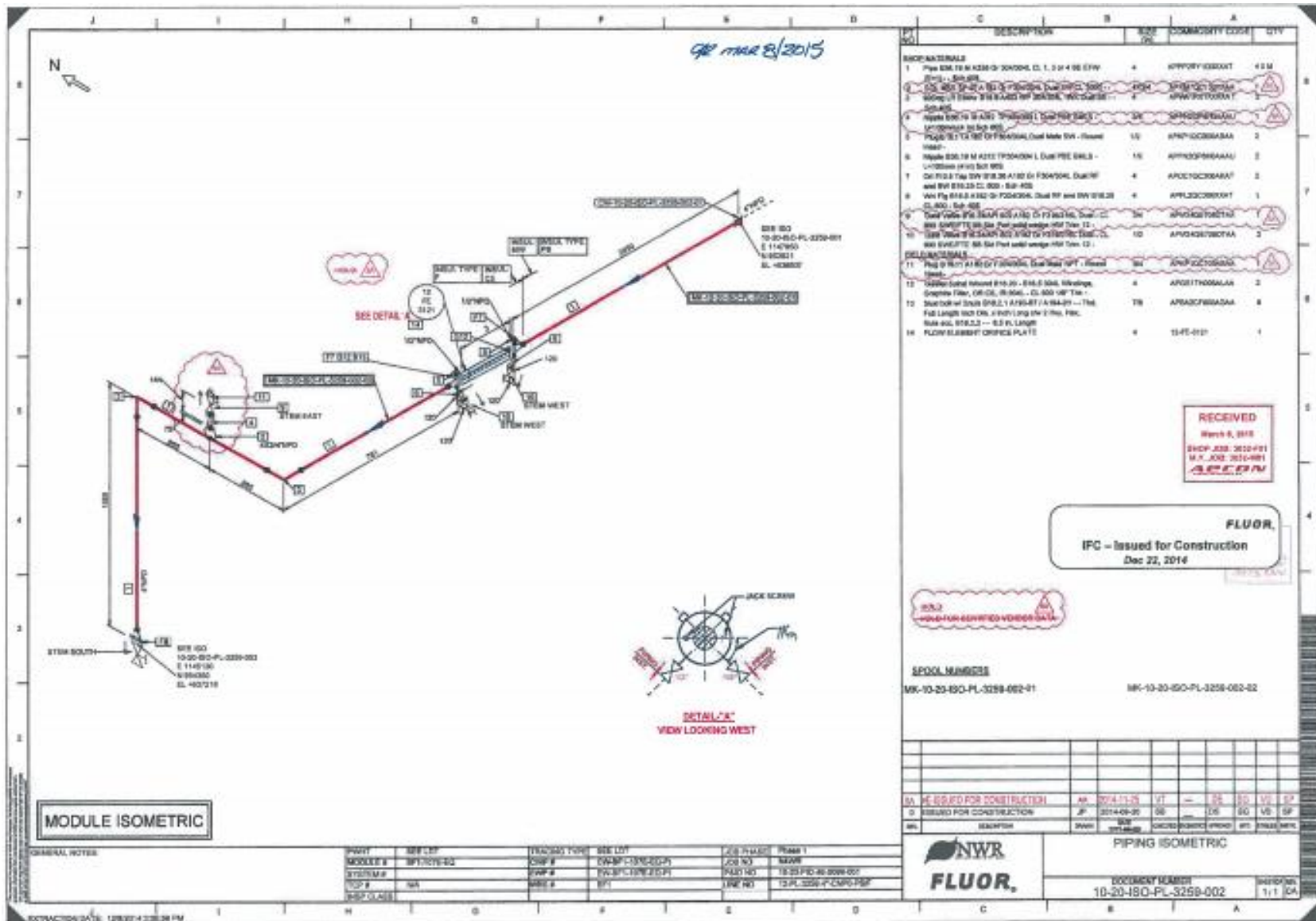


Figure 9: Example of a non-conformance report log page 5

2.1.6 Piping Fabrication in Practice

The following section illustrates the physical workflows provided in Figure 2 and Figure 3.

Once the drawings have been issued to the shop, the general foreman is responsible for properly distributing the drawings to the fitters and welders. Each worker has a ticket which allows him to perform certain types of fitting or welding. Also, drawings are dispatched between the different bays of the AIW facility. Figure 10 represents the shelf located in the AIW facility where workers pick up their job in the form of drawings.



Figure 10: Drawing shelf in the Aecon Industrial West facility

Once the drawing is brought into the shop, fabrication can begin and follows the physical workflows. Four main tasks are involved in the piping fabrication. After loading the materials from the storage rack to the cutting rack (Figure 11) and making sure no component is missing, the first step consists in cutting the components to the proper dimensions specified on the fabrication drawing (cutsheet). This task is performed semi-automatically by the cutter. His job is to correctly specify the required settings for the specific spool. Figure 12 shows the operator performing a cutting on a spool. The cutting information are typed into the monitor, and the machine cuts the pipe spool by itself.

Although this task could seem straightforward and not prone to defect, a common practice in the industry is to let extra tolerance to the spool so that it cannot be smaller than specified. As a result, the initial pipe is often longer than it should be. The reason to this practice is that it lets a chance to the fitters and welders to correct a potential mistake by giving the possibility to cut the extra length if the first attempt was miscellaneous.



Figure 11: (a) Outdoor storage rack and (b) indoor cutting storage rack before being sent to the cutting machine



Figure 12: Cutting operation performed by an operator

Once the cutting is performed, the spools are sent to the designated fitting station using roof cranes through each bay. Figure 13 shows the general layout of bay 1 in the AIW facility. The crane fixed at the roof is used to transport the spools between different stations. Following the fabrication flow, the spool goes to a fitting station such as the one shown in Figure 14.



Figure 13: Overview of Bay 1

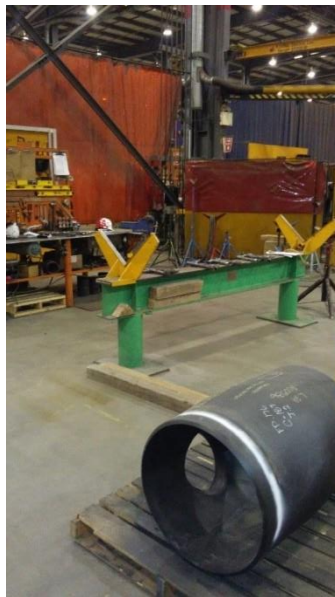


Figure 14: Fitting station

From there, the spool is unloaded by the fitter (Figure 15), and the second main task can begin: the fitting. Fitters are responsible for interpreting the cutsheet and performing the tack welds according to the drawing(s). Once a tack weld is performed by the fitter, the spool is shipped to the welding station for welding. One of the main tasks of fitters is to maximize the number of roll welds. This requires careful selection of sub-spools that can be tacked together, so the whole sub-spool can be rolled for all of its tacked welding locations. A complete spool is often too large or unwieldy to roll, and final welds to join sub-spools have to be done manually by awkwardly moving around the fixed spool. Roll welds are welds in which the

spool is rotating in a machine where the speed of the rotation is controlled by the welder. Roll welds have a better quality and take an order of magnitude less time to complete (Figure 16).



Figure 15: Pipe loaded at a fitting station for being assembled



Figure 16: Pipe spool prepared for roll weld

Another main task of the fitters is to perform continuous geometrical control of the assembly. After performing a tack weld, fitters are supposed to control the alignment, geometry and orientation of the fitting so that no weld is performed on a miscellaneous assembly. To do so, the fitting table is equipped with manual tools such as the ones shown in Figure 17. Fitters have at their disposal digital levels to measure the inclination of the assembled spool as well as other common hard tools.



Figure 17: Fitting table

As stated in the physical workflows, pipe spools go back and forth between the fitting station and the welding station. Once the assembly is complete, the quality control personnel are in charge of controlling the assembly. The person responsible for quality control has to check all of the lengths, angles, materials and grading of the spool with respect to the drawings. For example, Figure 18 represents a drawing that has just been controlled by the QC personnel. Cross marks are placed next to each length/angle right after they got controlled and are within tolerance. Once the entire spool has been investigated and is correct, the QC person stamps the drawing. If demanded, the assembly may then be sent for hydro tests before shipping. In the case where a defect is detected, the quality control personnel has to fill a Non Conformance Report (NCR) as previously shown in Figure 5 to Figure 9.

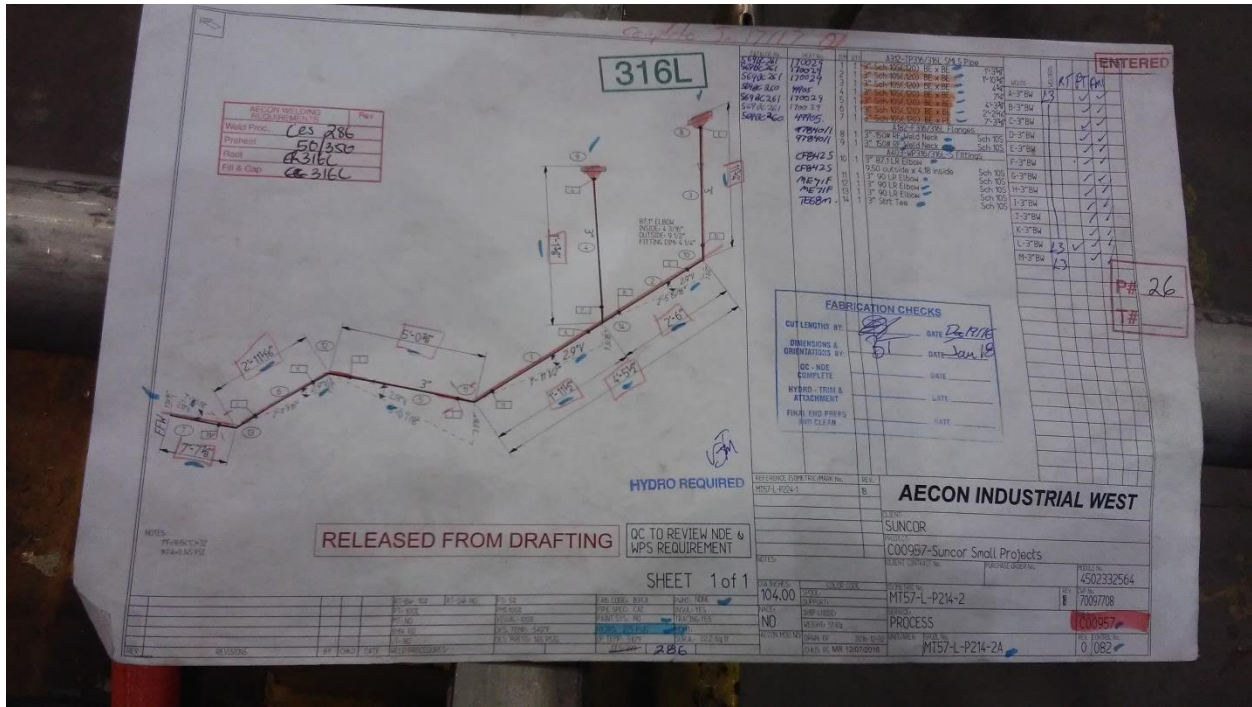


Figure 18: Drawing after the QC personnel inspection

One of the main shortcomings of the defect detection process is the poor level of accuracy and weight given to manual interpretation which makes it prone to human mistake. Integrating a 3D vision system for constant quality control has the potential to reduce the amount of rework detected at the end of the fabrication process by the quality control personnel. As briefly explained in 2.1.5, augmented reality could improve the efficiency of the fabrication by detecting defects right after tacking and before welding or at least right after they appear. This could drastically reduce the amount spent on fixing defects, and hence, making cost savings.

2.1.7 Theoretical Implementation of the Stepwise Quality Control

Following the shortcomings highlighted in the previous section, the following section describes the theoretical modifications that would be required in the current workflows in order to implement the proposed solution developed by the research team. Figure 19 shows the modification that would be required in the Aecon's fabrication physical flow to implement the solution developed by the UW team. The changes are represented by the yellow rectangle. Each one is represented by a sub-process that refers to the workflow presented in Figure 20. The idea is to implement the solution before going from the fitting station to the welding station. By this way, we make sure the assembly is correctly aligned and positioned before the weld is being done. To do so, the SfM package equips each fitting station. The current solution is using a handheld laser scanner (Structure IO) mounted on a tablet (iPad) to acquire the as-built assembly. The scan

is then remotely sent to a general computer – which could be located at the fitting station – where the post processing is finally achieved by the craft worker.

To perform such a stepwise quality control, some premises are required and are represented in the data flow (Figure 22) in yellow. Those changes consist in collecting to the client engineer not only the cutsheet of assemblies but also its 3D model and its Piping Component File (PCF). Also, because of the multiple revisions that can happen during the fabrication, the acquisition of a revised 3D model is necessary. These processes highlighted in the workflows are important for the effectiveness of the proposed quality control solution as well as for the future of the industry.

As such, the fabrication physical flow (Figure 19) is slightly modified by adding an extra step each time the assembly must be moved from the fitting station to the welding station. The extra step consists in applying the developed QC solution before welding the components which have just been fitted. Figure 20 provides the big picture of that step. The deployment of SfM in the AIW shop is then illustrated in Figure 21 with an iPad equipped with the scanner positioned at each deployable shop location. In addition, the fabrication data flow needs to be adapted (Figure 22). For a project, the client engineer should be asked to provide the 3D models. A SfM database could then be created to gather the cutsheets, the PCFs and the 3D models to enable an optimized use of SfM. Finally, as depicted in Figure 23, the fabrication revision flow would also have to be updated: instead of only revising the cutsheet when a revision happens, a revised 3D model should be provided to AIW after any revision, and the SfM database would be updated to match with the new version of the design.

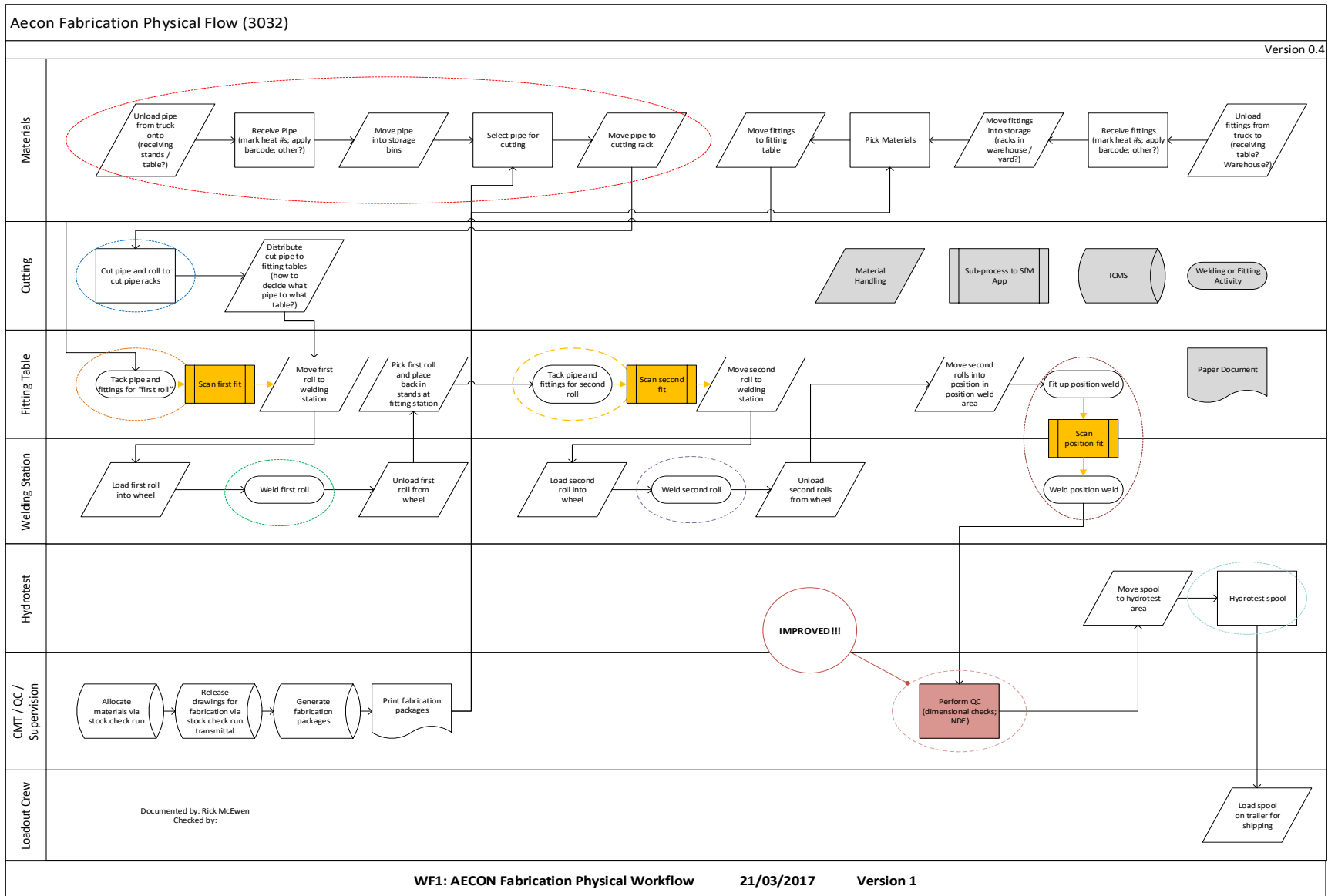


Figure 19: Implementation of Structure from Motion in the Aecon's fabrication physical flow

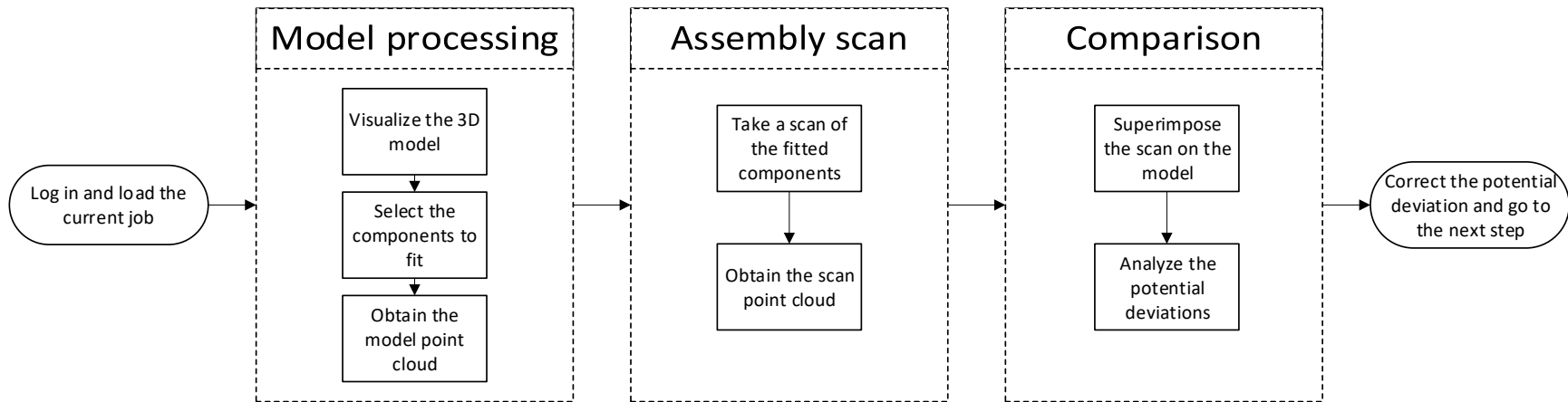


Figure 20: Simplified workflow of Structure from Motion

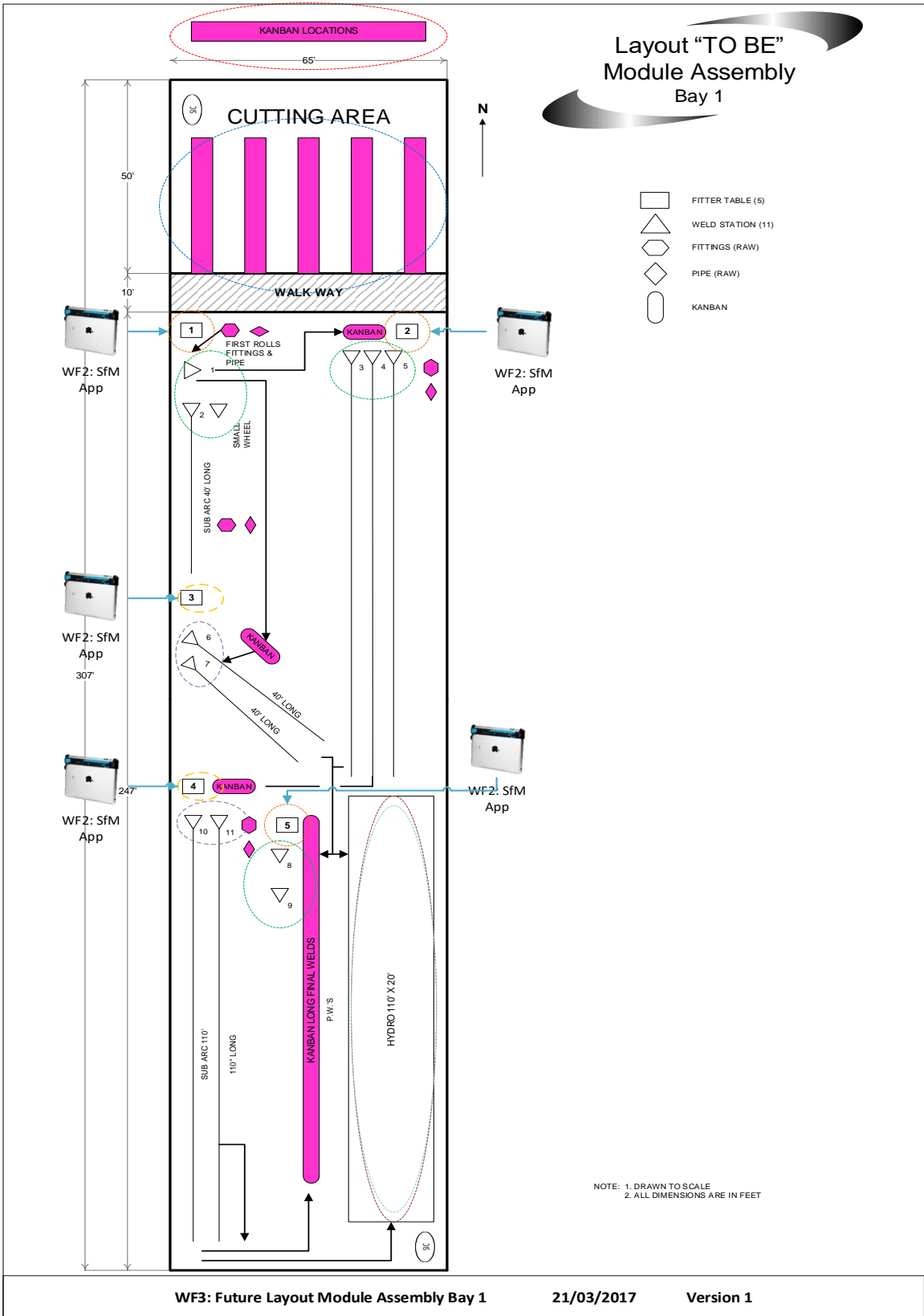


Figure 21: Layout of the AIW facility equipped with Structure from Motion

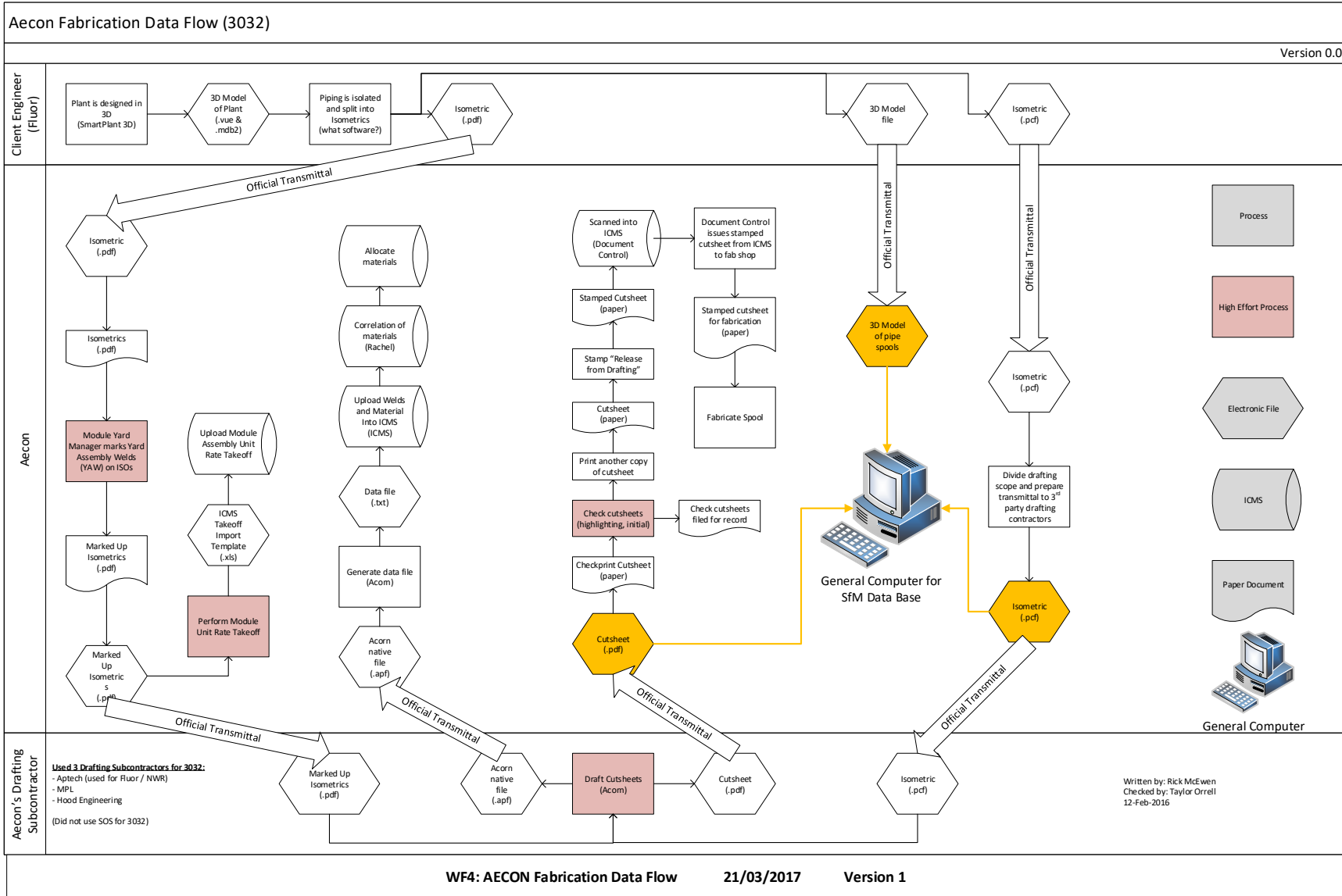


Figure 22: Implementation of Structure from Motion in the Aecon's fabrication data flow

2.2 Decision Making Method

To evaluate the best deployment option for the described scenario, focus was made on selecting the best scanner for the job, since scanning technology provides a wide spectrum of options to be evaluated. To simplify the analysis, only three criteria were used to evaluate the scanners: cost, fit for purpose, and deployability. Cost included acquiring enough hardware to deploy the system and not just the cost of an individual scanner. The computing system is also considered in the cost calculation. Fit for purpose refers to the scans produced and the scanner itself, taking into account the speed at which the scanner acquires points, the accuracy with which the points are captured and range at which the scanner works. Assessment of the fit for purpose of the scanners is based on the specifications provided by the manufacturers. Deployability accounts for how easily the scanner can be moved and how its physical properties suit the environment in which it is to be deployed.

2.2.1 Scanners Evaluated

Four scanners were evaluated for the purpose of this decision making. The scanners were selected to be demonstrative of their respective scanner categories and styles. The FARO Sense M70 (Figure 24) was selected as a standard tripod laser scanner, the BLK 360 (Figure 25) was chosen as a portable lower cost laser scanner, the Artec Eva (Figure 26) was chosen as a high-end portable structured light scanner and the Structure Sensor/IO (Figure 27) was chosen as an inexpensive super portable structured light scanner.

The FARO Sense M70 is a \$25,000 tripod laser scanner that captures points at 500,000 points per second. It is accurate up to 3mm and scans to a 70m range. It is the least portable scanner investigated as it is mounted on a tripod and is the heaviest scanner investigated. The FARO Sense M70 must be used in conjunction with a laptop or desktop computer.



Figure 24: FARO Sense M70

The BLK 360 is a small \$16,000 dome-shaped laser scanner. It captures points at a speed of 300,000 points per second. It is accurate up to 4mm and has a 30m scanning range. The scanner is portable, it is light weight and can be placed on any flat surface. The BLK 360 works with an iPad.



Figure 25: BLK 360

The Artec Eva is a high-end \$15,000 structured light scanner. It resembles a clothes iron in shape and size. It captures 2,000,000 points per seconds with an accuracy of 0.1mm and a scanning range of 0.4m. The scanner is portable but must maintain a wired connection to a Microsoft Surface tablet or equivalent computing device.



Figure 26: Artec Eva

The Structure Sensor is an inexpensive \$400 structured light scanner. It captures up to 1,000,000 points per second with an accuracy of 0.5mm and a 0.4m by 3.5m scanning range. It is the most portable scanner investigated as it can be mounted onto either an iPad or a Microsoft Surface tablet.



Figure 27: Structure Sensor mounted on an iPad

Table 2 summarizes the 4 different scanners investigated.

Table 2: Summarize of characteristics of different scanners

	Type	Operating System Compatibility	Cost	Speed	Accuracy	Range
FARO Focus M70	Tripod laser scanner	Windows	\$25,000	0.5 M pts/sec	3 mm	70 m
BLK 360	Portable dome laser scanner	iOS	\$16,000	0.36 M pts/sec	4 mm	40 m
ARTEC EVA	Handheld structured light scanner	Windows	\$15,000	2 M pts/sec	0.1 mm	0.4 m
Structure Sensor	Tablet-mountable structured light scanner	iOS/Windows	\$400	1 M pts/sec	0.5 mm	0.4 m x 3.5 m

2.2.2 Multi Attribute Utility Analysis

For the implementation of the solution described in Table 2, the fit for purpose of the scanner was deemed to be the most important factor, accounting for 61% of the decision. To minimize disruption to the existing work flow scans must be obtained quickly, while being accurate. The scanning range was also deemed important as needing to keep the scanner at a consistent distance from the object being scanned would make the scanning process too arduous to be implementable. In addition to the distance between the scanner and the object, the scanning area of the scanner was also an important factor in assessing the scanner's range.

Each scanner was rated based on its specifications for speed, accuracy and range and how they pertain to this use case. The evaluation across the 3 categories was averaged as the 3 criteria were deemed to be of equal importance. Table 3 summarizes the evaluation of the scanners based on their fit for purpose.

Table 3: Evaluation of the scanners based on their fit for purpose. 3 parameters are taken into account: the speed, the accuracy, and the range. The value is obtained by averaging the 3 parameters.

Scanner	Speed	Accuracy	Range	Average
FARO Focus M70	0.9	0.8	0.5	0.75
BLK 360	0.75	0.5	0.5	0.58
ARTEC EVA	1	1	0.6	0.87
Structure Sensor	1	1	0.7	0.90

The scanner with the best fit for purpose score was determined to be the Structure Sensor.

The cost of the scanner accounted for 25% of the decision as the shop environment is not ideal for fragile technology and a high replacement cost for damaged technology is not desirable. The determination of cost was based on the expected number of scanners and corresponding computing devices required to deploy the system in a shop. Table 4 summarizes the cost of the hardware required to deploy the scanners investigated.

Table 4: Cost of the hardware required to deploy the scanners investigated

Scanner	Cost	Computing System	Cost of Computing System	Number of Devices Needed	Total Cost of System
FARO Focus M70	\$25,000	Laptop	\$700	4	\$102,800
BLK 360	\$16,000	iPad	\$600	8	\$132,800
ARTEC EVA	\$15,000	Windows Tablet	\$1,500	15	\$247,500
Structure Sensor	\$400	iPad/Windows Tablet	\$1,500	30	\$57,000

The structure sensor provided the lowest cost scanning system.

Lastly, deployability accounted for 14% of the decision as the ability to move the scanner would be nice to facilitate scanning from the best angles, but was deemed to not be crucial as lifting the scanner above the shop floor hazards may help prolong the life of the hardware. The structure sensor was determined to be the most deployable option for the use case 1 environment.

A summary table of the qualities evaluated and their weights is summarized below. The weights were determined using the Multi Attribute Utility Theory (MAUT) with fit for purpose being four times as important as cost and three times as important as deployability. Cost was also determined to be twice as important as deployability. Results are presented in Table 5.

Table 5: Weight of each attribute for the MAUT analysis

Category	Weight
Cost	25%
Fit for Purpose	61%
Deployability	14%

Table 6 summarizes the utility provided to each scanner for each attribute.

Table 6: Utility provided to each scanner for each attribute

	Cost	Fit for Purpose	Deployability	Utility
FARO Focus M70	0.4	0.73	0.4	0.604
BLK360	0.3	0.58	0.6	0.515
Artec Eva	0.2	0.87	0.8	0.692
Structure Sensor	1	0.90	1	0.939

Based on the decision making approach outlined, the Structure Sensor currently being used was determined to be the most appropriate scanner for this application.

To put it in a nutshell, the solution using Structure IO on an iPad sending the as-built information to a general computer has been proven to be the most effective for our purpose which consists in developing a 3D vision based stepwise quality control method for piping fabrication. However, this solution contains challenges which must be tackled before being able to implement the solution in a shop environment. The main challenge is the short range these sensors possess, which currently deters the industry from using them for quality control applications. In the next section, a literature review is done around the deployment of 3D imaging technologies for quality control in the construction industry.

2.3 3D Imaging Technologies for Quality Control

This section is broken down into four sections: (1) an overview of the current quality control processes in pipe fabrication and their impact on obtaining the 3D model of the design, (2) utilization of 3D vision

technologies for visual inspection, (3) novel imaging tools and techniques to improve QC in construction industry, and (4) knowledge gap and problem statement of the proposed paper.

2.3.1 Quality Control Processes in Pipe Fabrication

The predominant processes for monitoring the geometrical correctness of fabrication assemblies in the piping industry involves manual inspection by certified QC personnel using direct contact measurement devices, such as, metal measuring tapes, calipers, custom gauges, squares, and straight edges. These manual methods of inspection can generate measurement uncertainties, which have to be taken into account when verifying assemblies' compliance. Uncertainties and errors in measurement can come from many sources, such as the measuring device, the component being measured, the skill of the craft worker or inspector performing the measurement, the measurement process, and the environment.

In addition to measurement uncertainty, dimensional control problems also originate from poor existing document creation practices. For example, chain dimensioning establishes ambiguity and the potential for accumulated measurement error, or the drawings include units that may not be legible to the fabricator, which may cause delays.

Another problem is the fragmented process from the design stage to the fabrication. Although not universal, some of the major steps include: (1) 3D design and modeling, (2) transmitting isometric drawings, (3) production drawing creation (referred to as “cutsheets”), and (4) work package distribution and fabrication. Depending on the contract, a different company may do each of these steps. As such, a major challenge is to address the interoperability issues, such as, different software packages used in the modeling process (Leite et al. 2016). Companies have developed different strategies as how to address the interoperability issue, including using their in-house workforce to re-model the design using 2D drawings, which can be very costly and inefficient (Holzer 2007) or, in many projects, not taking advantage of the 3D models. However, 3D modelling and especially Building Information Modelling (BIM) have been proven to be of a major importance in construction projects. Communication, coordination, visualization and clash detection are among the major advantages BIM models (Chen and Wu 2014, Farnsworth et al. 2015). As a result, in order to remedy the interoperability challenge and keep the benefits of using 3D-models, software independent CAD file formats have emerged. For instance, a commonly used format is Stereolithographic format (*.Stl) (Garcia et al. 1999), which is compatible with a variety of software packages and only contains the geometrical information of an assembly (Lee et al. 2016).

Furthermore, a file data transfer method is used in the piping industry to pass along the most important geometrical information and management information related to the pipe spool assembly. Piping Component File (PCF) is a type of file which contains this important information (Fung et al. 2014). PCFs

are formatted as a text file and hence are not software dependent. End-points of the components, diameters, length, type of weld, and specific description are some instances of the PCF, which is segmented in paragraphs specifying each component or weld.

2.3.2 Automated Visual Inspection in Construction

Automated inspection is desirable because manual inspection by humans is time-consuming, and can be excessively subjective, unreliable, and not interesting for humans to perform. Also, many industrial assemblies are not easily accessible for manual inspection. Emergence of CAD models in the mid 1990's (Newman and Jain 1995) and advanced 3D imaging technologies allowed for accurate dimensional evaluations as automated parts of the fabrication process. Bosché et al. (Bosché et al. 2015) presented a methodology for using 3D laser scanners and comparing the as-built data with the BIM model for construction projects.

Building on this methodology, Nahangi et al. (Nahangi et al. 2015) presented an automated approach for monitoring and assessing fabricated pipe spools and structural systems using automated scan-to-BIM registration. The method reliably detects the presence of dimensional non-compliance and has consistently quantified deviations with less than 10% error in experimental studies. The method requires two 3D imaging input files: (1) a point cloud of the as-built assembly generated using a 3D reconstruction technique such as Light detection and ranging (LiDAR), and (2) the tolerance specifications as represented by a 3D CAD design file. LiDAR, often referred to as laser scanning in the construction industry, is an increasingly important technology. However, the acquired data can be too dense and a number of methods have been developed to automatically find objects of interest in the data (Sharif et al. 2017). Similar methods include the combination of BIM and LiDAR to build a real-time construction quality control system (Wang et al. 2015). It is the state-of-the-art that 3D laser scanning technologies are commonly used in construction to inspect the quality of a project (Tang and Akinici 2012, Bosché 2010, Malamas et al. 2003).

Researchers have created automated methods for monitoring and performing automated 3D image-to-BIM comparison of mechanical, electrical, and plumbing (MEP) systems (Bosché et al. 2013, Bosché et al 2015), and general building and structural systems (Bosché et al. 2009, Bosché 2010, Golparvar-Fard et al. 2011). Using 3D imaging for dimensional compliance assessment of construction components has the potential to mitigate costly repair and rework while tracking progress (Turkan et al. 2012).

Laser scanning technologies have thus been under extensive investigation and have been successfully applied in the industry (Tang et al. 2011). Yet, most of the time, laser scanners are merely used at the end of the fabrication process to control and validate the assembly, because 3D imaging technologies require engineering work and are usually labor-intensive (Akinici et al. 2006). Consulting companies have to come

in the fabrication facility to scan the assembly and then have to process the data before providing a final report. The process is expensive and takes several days (Anil et al. 2013). Since the data captured by laser scanning typically includes a large number of point clouds, the derivation of as-built models in a quick and cost-effective manner is a major challenge (Brikalis et al. 2010, Tang et al. 2010, Bhatla et al. 2012).

2.3.3 New 3D Vision Technologies in Fabrication

With the advancements of technology, new methods have emerged such as augmented reality (AR) systems. Combined with BIM, AR can provide a full 3D interactive solid model of the design, giving the workers a visual understanding of the design. Wang et al. (Wang et al. 2014) describe some of the AR functionalities such as superimposing a 3D colored model onto the standard 2D drawing plan. As information can be made available in real-time, AR can be used to expedite tasks efficiently and effectively (Hou and Wang 2011). Head mounted displays (HMDs) can potentially be fully integrated in QC processes (Makris et al. 2013, Wang et al. 2016). In (Aiteanu et al. 2003), a welding helmet is presented which provides augmented visual information such as drawings and quality assistance. Schwerdtfeger et al. (Schwerdtfeger et al. 2008) employ laser projectors for stepwise fabrication. Laser beams of the component's boundaries are projected on the surface where it is to be assembled. Molleda et al. (Molleda et al. 2013) developed a 3D imaging system for dimensional quality inspection that utilizes range images. Another category of acquisition devices for collecting as-built information is Unmanned Aerial Vehicles (UAVs). Seibert and Teizer used an UAV system to survey earthwork of large infrastructure projects (Seibert and Teizer 2014). Other uses include and are not limited to project tracking, quality inspection, and safety inspection (Irizarry and Costa 2016).

2.3.4 Knowledge Gap and Problem Statement

Recently, structured light scanners have emerged as potential 3D imaging acquisition tools, but due to their short scanning range, no realistic utilization has been developed for construction assemblies (Fathi et al. 2015). Despite their range that for most of them does not exceed four meters, structured light scanners such as the Structure IO are inexpensive and potentially accurate. This means that workers can use them with little fear of breaking them and that a precise quality control and geometric compliance can be performed (Golparvar-Fard et al. 2011). Combining the two characteristics opens the path to stepwise quality control of construction assemblies. However, the range constraint has to be addressed in order to be able to use range cameras for construction assemblies.

Construction assemblies are often larger than the intrinsic structured light scanner range of a few meters (Kim et al. 2008). To face this major challenge, the solution proposed in the following thesis is to shrink down the dimensions of the required scan by redefining the boundaries of the model. The 3D model is derived to contain the minimum required geometrical information to enable the geometric control of the

assembly. A method is developed to automatically derive the Minimum Required Model (MRM) and allows structured light scanners to be effectively used for construction assembly control. To achieve the derivation of the MRM, an assembly guidance process is designed for workers and facilitates the stepwise tracking of the assembly.

The proposed method combines STL and PCF file formats to prevent any possible interoperability challenges, to improve the control guidance and the quality control (QC) of the assembly in a stepwise manner. As new components are added to partially-built assemblies composed of one or more components, the geometric control can be applied at each step to prevent any ongoing deviation from being propagated throughout the completion of the assembly.

2.3.5 Analogous Problem from Other Industries

Completely different areas are facing similar problems and proposed somewhat similar approaches. This is the case in the 3D computer gaming industry where scenes need to be generated in real-time because complete gaming maps would be too computationally heavy for any GPUs. One major approach is the use of procedural modeling techniques (Patow and Besuievsky 2013). Procedural modeling describes a family of techniques that generate geometry from a set of rules. Therefore, those techniques are used in a variety of open world games such as *Grand Theft Auto*. As such, Steinberger et al. (Steinberger et al. 2014) applies a grammar-based procedural modeling technique in order to dynamically generate the real-time view of a city. However, it is still challenging to render large environments because of the huge amount of storage that would be required. To tackle the issue, they develop a specific methodology to only generate the required geometry to render the current view in real-time. They somehow manage to predict the next frame by taking advantage of the “frame-to-frame coherence”. Instead of building up a new tree to represent the next frame, specific procedures are set to only update the tree from the previous frame. Consequently, they are facing similar problems where a new model need to be derived with a minimum amount of geometry since the GPU memory usage has be restrained. The derivation of the next frame follows a set of rules to minimize the number of changes in between two consecutives trees.

A parallel can be drawn between the frame-to-frame coherence concept used by the aforementioned research and the proposed step-by-step derivation of the Minimum Required Model. The big picture challenge of the MRM derivation is to reduce the 3D model geometry up to its minimum, because it is not necessary to check the entire model to control the geometric compliance of an assembly. Indeed, like for the frame-to-frame coherence, when a stepwise quality control is done, the just-assembled components are assured to be compliant. Similarly, a set of rules can be applied to minimize the amount of geometry in the next derived 3D model.

Another parallel that can be acknowledge between this thesis and gaming industry is the utilization of geometric symmetries. Among many applications, they are particularly used to perform improved model acquisition as only a base patch of an object need to be detected (Mitra et al. 2013). Consequently, model representations can be shrunk down using the detected symmetries which factored out models redundancies (Simari et al. 2006). Another benefit is that the symmetrical patch can directly be inferred thanks to symmetrical transformation. Therefore, higher model representations can be rendered and can carry more graphical details. This process is named model compression and is illustrated in Figure 28.

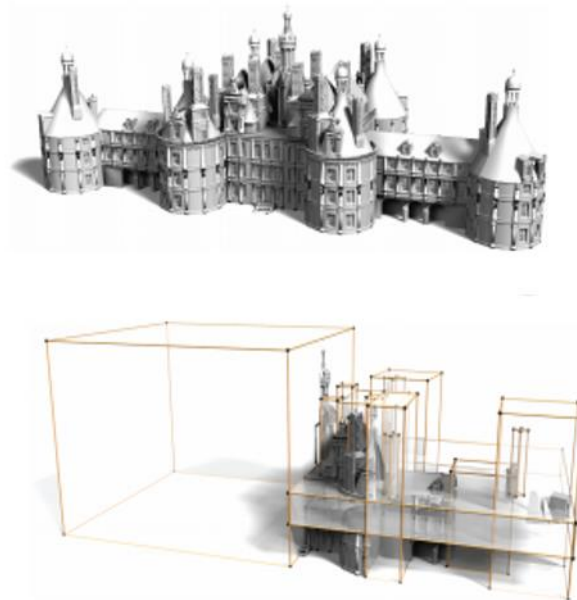


Figure 28: Model compression using translational and rotational symmetries (Mitra and Pauly 2008)

As a conclusion, development of algorithms capable of deriving the next model/scene by storing the minimum amount of information has been a major challenge and, for this reason, has been under extensive investigation. Unlike the gaming industry, the construction industry has not faced a similar challenge yet. But, with the development of large 3D models combined with the increased demand on quality, new processes for quality control, such as the stepwise process described in this thesis, are likely to be implemented in coming years. Such problems will most certainly become a matter of concern sooner than expected. Methods for performing an inexpensive, accurate (through 3D imaging techniques), and productive quality control will become necessary in the upcoming construction manufacturing world as it has been in gaming for real-time scene generation.

3 Proposed Implementation Workflow

The thesis methodology is presented below. Figure 29 illustrates the main steps of the overall workflow being developed and the parts on which this thesis is focused. The workflow begins with the reception of the required inputs: (1) the CAD (Computer Aided Design) model, and (2) the Piping Component File (PCF). Once those two files are saved in the SfM data-base, the assembly of the pipe spool can start.

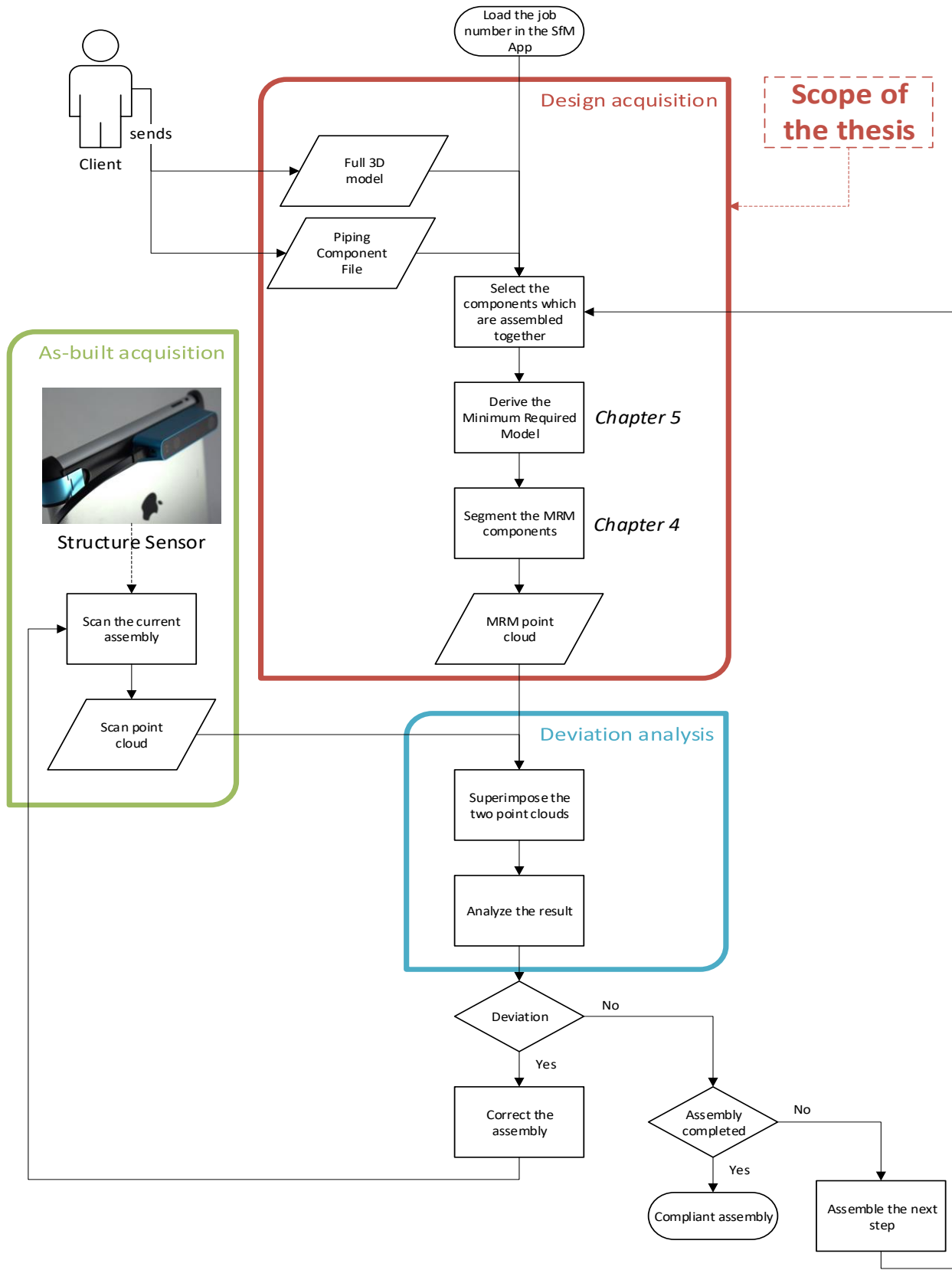


Figure 29: Scope of thesis within developed workflow

The worker chooses what stage of fabrication he wants to tackle and selects his components. From this selection, the derived Minimum Required Model is derived by: (1) segmenting each individual piping component, and (2) calculating the components which are part of the MRM based on geometrical properties. The MRM is produced in the format of a point cloud and serves as model in the subsequent Scan-vs-BIM comparison. Following the derivation of the model, the as-built assembly is scanned and remotely transferred in the format of a point cloud to the SfM computer. The scan is then superimposed on the 3D design in order to proceed to the deviation analysis. This process is repeated until the assembly is completed so that each step is controlled and the geometry of the assembly is assured.

The purpose of this thesis is to study the derivation of the Minimum Required Model, but it is worth keeping in mind that the generated 3D design only has some potential when integrated to a complete fabrication solution.

Therefore, Chapter 4 presents the methodology for segmenting each individual piping component within the *Structure from Motion* software. The methodology takes advantage of both CAD model and Piping Component File (PCF) to retrieve each component's geometric characteristics. In Chapter 5, the derivation of the MRM is thoroughly detailed. Every theoretical aspect of the methodology is provided in this section, and each point is explained in detail.

Chapter 6 presents the validation of the proposed methodology by quantifying its effectiveness based on three different metrics.

- The first metric is the reduction of the spatial complexity of the 3D design resulting from deriving the MRM. In this section, a parameter is created entitled Level of Complexity (LOC). LOC is evaluated on a data-base of around a hundred 3D models as well as on the steel pipe spool of the University of Waterloo lab.
- Secondly, the reduction of the assembly principal length is measured before and after applying the methodology. It gives an overview of the potential applications of the MRM in real situations using the chosen scanning package. Consequently, based on the utilized device maximal scanning range, areas of potentialities are created, splitting assemblies whether they can, or not, be scanned before and/or after applying the MRM methodology. When assemblies can just be scanned with the Structure IO if reduced to MRM, then the methodology is considered effective.
- Finally, as the application of the MRM enables to reduce the size of the assembly and thus of the required scan volume, the scan density can be improved. As a result, the deviation analysis becomes more accurate as the scan is denser. The structural deviation measuring the distance between points on the same surface but located at a different position can be diminished by using the MRM.

As presented in the general workflow, the scope of the thesis is the required design scan volume acquisition process. However, the evaluation of the methodology is partially based on the improvement on the next step, that is, improving the accuracy of the deviation analysis and enabling the utilization of Structure Sensor for quality control in a piping fabrication facility.

Consequently, Chapter 4 and 5 focus on the elements encapsulated in the red frame (Figure 29) and explain in detail how is the MRM correctly derived. Chapter 6 evaluates the improvement on the design acquisition, the as-built acquisition, and the deviation analysis. By this way, the implementation of the developed methodology is proven to enhance on every aspect the overall solution proposed in the *Structure from Motion* software.

4 Segmentation of Piping Components

4.1 Introduction

Segmentation of a 3D point cloud design model is a key process in a 3D vision-based stepwise assembly quality control. In a serial assembly fabrication, components are assembled one after the other in a very specific order. The fitter, according to his experience, chooses a specific assembly order to minimize the assembly time; that is, his goal is to optimize the number of roll welds along the assembly.

To this extent, the segmentation of the 3D model has to enable the craft worker to select in the software any piping component involved in the stage of fabrication.

However, as shown in Figure 29, the client only provides the contractor with the full 3D CAD model, which is, most of the time, in a non-editable format. By converting the CAD file (.STL) into a point cloud (.XYZ), one only has a single point cloud representing the whole originally designed 3D model assembly, with no distinction between different components, and no possibility for retrieving specific subsets of the assembly. Addressing this challenge implies developing a solution that allows for the generation of the required subsets of point cloud. To this end, each piping component type has to be segmented individually to take into account each geometric peculiarities.

To achieve such a task, pre-processing the inputs is necessary. This implies being able to segment each piping component independently within the complete assembly point cloud. A method is developed that uses the properties of Piping Component Files (PCF) as well as Stereolithographic (STL) design models. One major benefit is that the segmentation method doesn't rely on any software, because both the 3D design and the PCF are software-independent types of files.

The following sections detail how the two specific inputs – (1) the 3D design of the whole assembly which contains the shape and a 3D visual representation of the job, and (2) the PCF of the assembly which contains all the engineering information of the component parts of the assembly – are used. Then, each segmented type of piping component is studied independently.

4.2 Piping Components

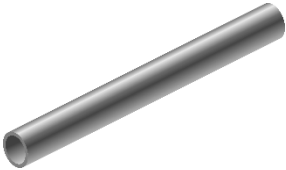







Before all, it is required to briefly present the most common piping components as they will be mentioned many times in the rest of the document. Table 7 shows the most common piping components with a realistic picture.

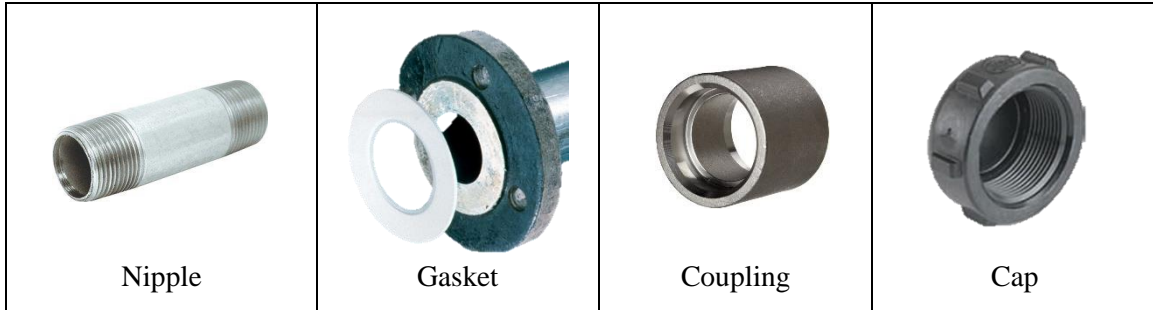
Below are some specific details for each component:

- Pipe: a straight pipe is a cylinder with a unique diameter.

- Elbow: elbows can have different angle such as 90° or 45° (a 90° elbow is represented in Table 7), and radii of curvature.
- Tee: tees connect three branches.
- Valve: a valve can be seen as a tee but instead of connecting another branch to the main route, it connects a spindle with a tap.
- Flange: flanges have a nominal diameter and an outside diameter induced by their pipe standard and pressure class.
- Concentric reducer: a concentric reducer is a cylinder with different diameters on both ends.
- Eccentric reducer: it can be seen as concentric reducer except its two centers are out of alignment.
- Olet: this component is usually added anywhere along a pipe spool and enables to branch out the main route. Olet is not considered in the thesis because their location is undetermined and most of the time it tends to be replaced by tees.
- Nipple: it connects two pipes together when they have to be screwed.
- Gasket: it serves as a joint in between two flanges that are bolted together. Gaskets are very thin and thus are skipped on purpose in the methodology, because they don't account for a geometrically significant step of fabrication.
- Coupling: it can be seen as a nipple when welding is involved.
- Cap: it "closes" the pipe spool at its ends. Caps can either be screwed or welded.

Table 7: List of the principal piping components with a picture

 <p>Pipe</p>	 <p>Elbow</p>	 <p>Tee</p>	 <p>Valve</p>
 <p>Flange</p>	 <p>Concentric reducer</p>	 <p>Eccentric reducer</p>	 <p>Olet</p>



Many variations of those components exist but look similar. Other components are “support components” and are used to hold the spools or maintain components before they get welded. Every piping component can be found in the Isogen Symbol Key Definitions (Isogen Symbol Key (SKEY) Definitions, 2007).

4.3 Use of 3D Model

Use of 3D models is a challenge in the pipe industry and in construction in general because of the numerous revisions a job can have. Each change is recorded and the 2D drawing is typically modified, but the 3D model, unless specified in the contract, is usually not updated. Issuing a revised 3D model is considered too time-consuming and too expensive. That is why in most cases, the management team chooses to simply record the change in the database by red-lining the 2D drawing. However, shifting this process towards updating the 3D design model could bring tremendous benefits both to the client and to the contractor such as avoiding conflicts between stakeholders, providing 3D visualization to workers and enabling as-built to as-design comparison with 3D scans. Considering that the described process is reached, the proposed method can be performed using the appropriate 3D design file without any mistake.

For the 3D design model of the piping assembly, the SfM solution uses the STL format. The STL assembly file is converted within the SfM solution to a point cloud as a XYZ file (.xyz). The software enables the user to choose the number of points of the model point cloud. Finally, on the segmentation page, the 3D model gives the user the opportunity to visualize its selection. It surrounds the selected component with an envelope that fits its size as shown in Figure 30.

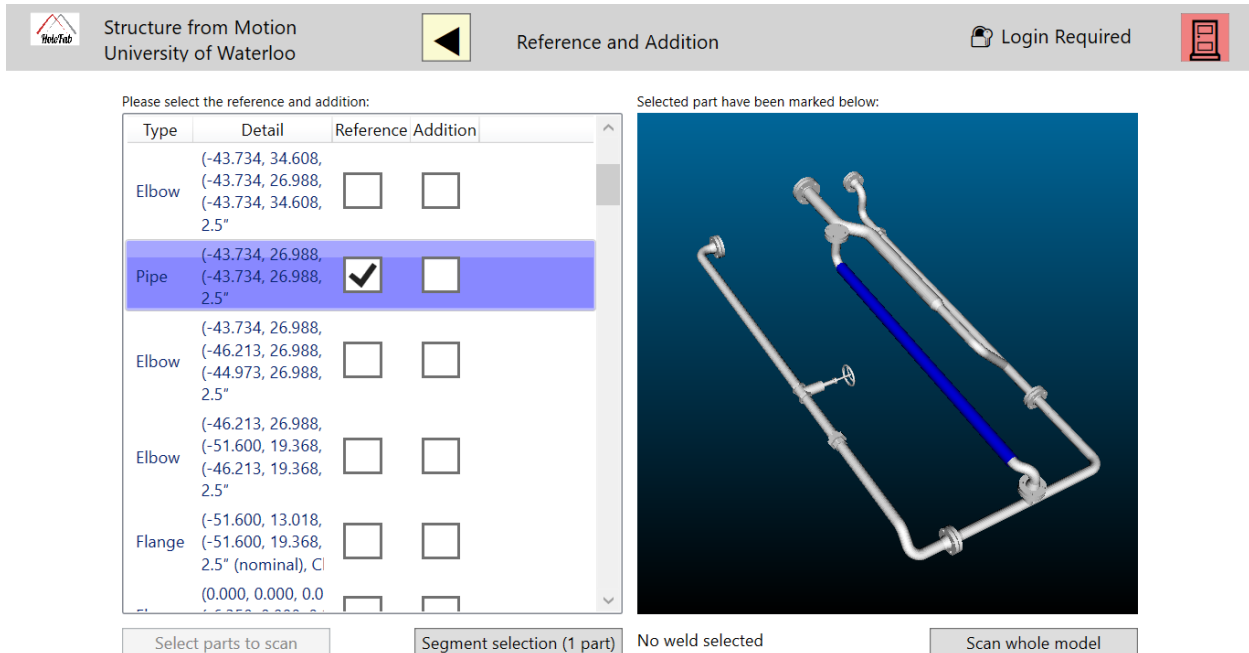


Figure 30: SfM Segmentation page

Once the selection has been made according to the fitter's choice, the 3D model is displayed as a point cloud in a XYZ file. A straight pipe and an elbow have been selected in the segmentation page and the resulting point cloud is shown in Figure 31.

The point cloud is set as the model point cloud and will be used to assess the compliance of the as-built assembly. The model point cloud density is defined as a setting (e.g. 10,000). The points are distributed onto the surface of the components based on the surface of the initial mesh. Consequently, more points will be part of a larger mesh.

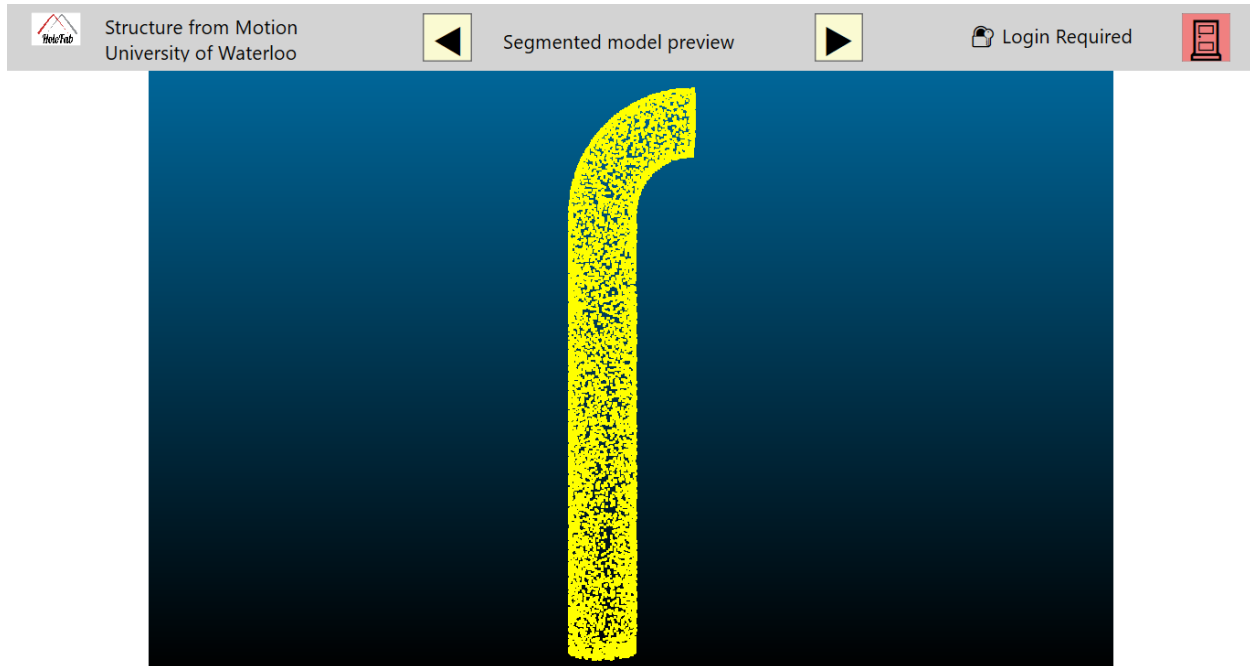


Figure 31: Point cloud representing a straight pipe and an elbow

4.4 Generation and Use of PCF File

In addition to being issued the 3D model of the piping projects, piping fabrication shops usually receive another design/management file called Piping Component File (PCF). PCFs gather geometric information of the designed assembly, mainly the end points of the different components, as well as various information regarding specific details of components and, depending on the design, managerial specifications. PCFs are simply text files, readable by any hardware with any notepad software. For each component, a PCF includes various mandatory records:

- Type: pipe, elbow, flange or valve etc...
- Geometry: coordinates, diameter of key points
- (optional) Description: semantic description, pressure class etc...

In fact, the generation of the PCF relies on the generation of the CAD file or BIM. For example, in the case of a semantic BIM, it is possible to include specific details such as the name/type of worker assigned to a weld. Besides, an updated version of PCF called “Super PCF” has been recently launched by Isogen Inc., the company which initiated PCFs (Piping Data Exchange using the Super PCF. 2008). “Super PCF” are able to deal with more data to produce more detailed PCFs.

In the current version of SfM, a PCF can simply be manually generated by inputting the end points coordinates (in metric unit) of each component and weld with their associated diameter (in imperial unit).

The word “FABRICATION-ITEM” has to be typed in for each component of the assembly because, in complex assemblies, some components out of the “real” assembly can be part of the PCF. Typically, it is the case for any piping support or components that would just be mentioned in the cutsheet as “piping continuation” but would belong to another cutsheet. As a result, Figure 32 displays a manually created PCF from the steel pipe spool assembly located in the University of Waterloo (UW) lab.

```

1000e.pcf - Notepad
File Edit Format View Help
END-POSITION-NULL
  CO-ORDS      -43.73372    50.77968    199.99198
END-POSITION-NULL
  CO-ORDS      -65.80886    25.43556    280.90876
END-POSITION-NULL
  CO-ORDS      -105.57002   20.34032    282.22194
END-POSITION-NULL
  CO-ORDS      8.29564     12.405      262.6233
FLANGE
  END-POINT    -43.73372    50.77968    199.99198    2.5
  END-POINT    -43.73372    48.87468    199.99198    2.5
  FABRICATION-ITEM
GASKET
  END-POINT    -43.73372    48.55718    199.99198    2.5
  END-POINT    -43.73372    48.87468    199.99198    2.5
  FABRICATION-ITEM
FLANGE
  END-POINT    -43.73372    48.55718    199.99198    2.5
  END-POINT    -43.73372    42.20718    199.99198    2.5
  FABRICATION-ITEM
PIPE
  END-POINT    -43.73372    42.20718    199.99198    2.5
  END-POINT    -43.73372    34.60750    199.99198    2.5
  FABRICATION-ITEM
WELD
  END-POINT    -43.73372    34.60750    199.99198    2.5
  END-POINT    -43.73372    34.60750    199.99198    2.5
  FABRICATION-ITEM
ELBOW
  END-POINT    -43.73372    34.60750    199.99198    2.5
  END-POINT    -43.73372    26.98750    192.37198    2.5
  CENTRE-POINT -43.73372    26.98750    199.99198
  FABRICATION-ITEM

```

Figure 32: Extract of a manually created PCF

As shown in Figure 32, PCF are initiated with “END-POSITION-NULL” followed by a single point – described with its coordinates. Those points account for the extreme end-points of the assembly where no component is attached to. Those points are not necessary for the generation of the PCF but are recommended because they facilitate the registration between the PCF and the STL coordinate systems. Since in some situations, the STL and the PCF have an offset. With this process, the offset is automatically calculated and

a file (AssemblyNumber.sfm) is created and stored in the associated design folder. As long as this offset file is located in the design folder, the two coordinate systems are registered together, and no manual selection for matching the coordinates together is required. Those points are very useful when an assembly is used for the first time.

When PCFs are automatically generated from a 3D CAD software, such as *AutoCAD Plant 3D* or *AutoCAD Inventor*, the content for each component is much denser and contains more details regarding each component. An example of such a PCF is shown Figure 33. In Figure 34, the PCF of an elbow with its common attributes created from *AutoCAD Plant 3D* is provided.

```

PIPE
COMPONENT-IDENTIFIER 1
END-POINT 1251600.000 917184.910 638953.522 2 PL
END-POINT 1251600.000 917184.910 637775.000 2 PL
FABRICATION-ITEM
INSULATION-SPEC MW
INSULATION ON
PAINTING-SPEC UNDEFINED
TRACING-SPEC NT
MISC-SPEC3 UNDEFINED
PIPING-SPEC ABB3
WEIGHT 7.773
UCI {0001388C-0000-0000-F617-FA627354AB04}
ITEM-CODE APPP1DF2H0XXAC
ITEM-DESCRIPTION Pipe B36.10 M A106 Gr B PE SMLS - - Sch XS
SPOOL-IDENTIFIER MK-10-20-ISO-LS-5250-006-01
CUT-PIECE-LENGTH 1178.52
ELBOW
COMPONENT-IDENTIFIER 2
END-POINT 1251600.000 917224.622 638993.222 2 SW
END-POINT 1251600.000 917184.897 638953.522 2 SW
CENTRE-POINT 1251600.000 917184.910 638993.234
SKEY EL**
ITEM-CODE APKA1DD130XXAA
ITEM-DESCRIPTION 90Deg. Elbow B16.11 A105N SW CL 3000 - -
UCI {00013885-0000-0000-F939-B83773545104}
INSULATION-SPEC MW
INSULATION ON
PAINTING-SPEC UNDEFINED
TRACING-SPEC NT
MISC-SPEC3 UNDEFINED
PIPING-SPEC ABB3
FABRICATION-ITEM
WEIGHT 1.755
WELD
COMPONENT-IDENTIFIER 3

```

Figure 33: Extract of a PCF file

From any 3D CAD software, the PCF is generated using the 3D model coordinate system. Points are read from the negative to the positive values along the coordinate system. The x-axis is considered first followed by the y-axis and finally the z-axis. Consequently, as soon as a point is detected, the entire component is positioned in the PCF and so on. Welds are also taken into account, detected and positioned in the same order. As a result, linearly inferring the flow of the assembly can lead to misinterpretations. Rather, one should look for same or very close key-points coordinates to reconstruct the flow of assembly.

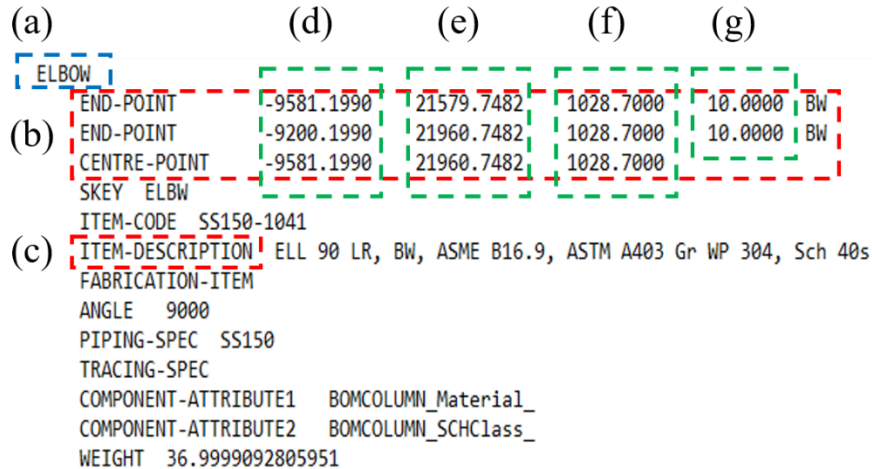


Figure 34: An example Piping Component File: (a) the described component is an elbow, (b) three key-points: two end-points and one center-point, (c) item description where, for instance the piping standard is specified, (d) x-coordinates of the key points, (e) y-coordinates of the key-points, (f) z-coordinate of the key points, (g) diameter about the key-points.

4.5 Segmentation of Components

The general idea of the proposed segmentation is to construct an envelope surrounding the point cloud of the component of interest. This component's point cloud originates from the complete design point cloud derived from the STL input file and converted into an XYZ file. To do so, the most accurate means is to reconstruct each component geometry using its geometric properties from the PCF.

For every component, three parts need to be coded to extract its point cloud.

- First, the detection of the component in the PCF file to generate a new object class with its respective attributes.
- Secondly, the maths to segment a specific component are encoded.
- Finally, criteria are implemented such as distance function or angle function that determine whether or not a specific point in the entire assembly point cloud is part of a component.

To give a clear view of these functions, some parts of the code are provided in Appendix A.

For every component, each process is explained, and particular attention is given to component specifications. Straight pipes as well as every cylindrical piping component are first investigated. Then, elbows, which are more complex to properly extract, are presented. Segmentation of tees is then detailed. Considered approximately identical to tees, valves segmentation follows the previous section. Eccentric reducers are then tackled, and finally, flanges segmentation is explained.

4.5.1 Segmentation of Pipe and Cylindrical Component

A straight pipe is a cylindrical component and thus the segmentation of straight pipes and other cylindrical components such as caps, unions, couplings, nipples or concentric reducers is identical. In this section, straight pipe also accounts for any other cylindrical component.

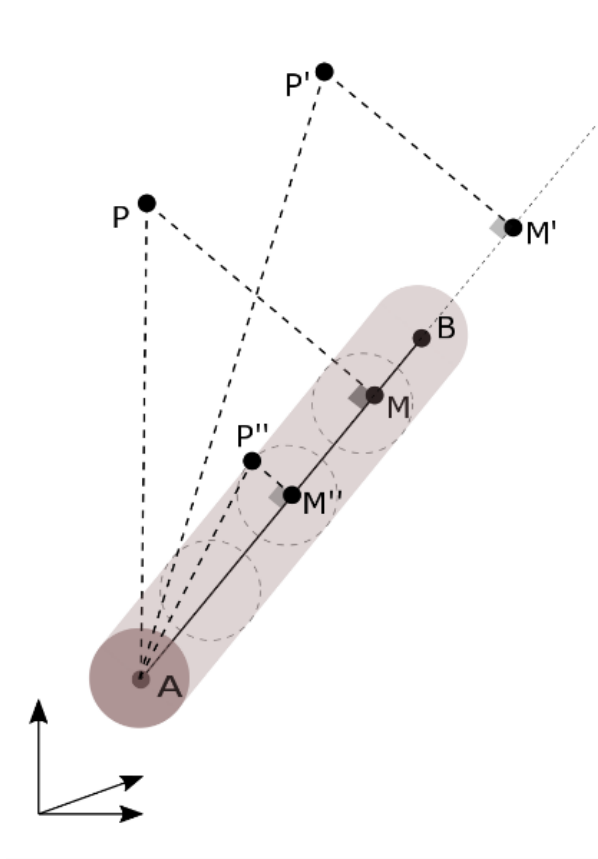


Figure 35: Representation of a straight pipe

A straight pipe can simply be represented in the form of a cylinder. Geometrically, any cylinder can be completely determined by its two end-points with their respective diameter. Apart for concentric reducers, the diameter for each end-point is identical. Figure 35 shows the two end-points, A and B, of a pipe going from point A to point B with a specific diameter defined in the PCF. The coordinates of those two points and diameters are extracted from the PCF.

Considering the point cloud generated from the original design file, the code calculates the projection of the distance from the first end-point A to a particular point P onto the lines made by the two end-points of pipe [AB]. The projection point is called M. If M is found to be out of the segment [AB] then the point is directly disregarded. For instance, point P' in Figure 35 is excluded because M' doesn't belong to [AB]. If the projection point is included in [AB], then the distance [PM] is calculated.

As highlighted in Figure 36, the diameter of the pipe is indicated in inches. The conversion of this distance is made and a small tolerance is added to make sure every point of the pipe is selected. For example, the distance for a 2 inches diameter is determined as followed:

$$\begin{aligned}
 Distance_{mm} &= PipeRadius_{mm} + 10\% \text{ of } PipeRadius_{mm} \\
 &= 1.1 * \frac{PipeDiameter_{mm}}{2} \\
 &= \frac{1.1}{2} * 25.4 * PipeDiameter_{inch} \\
 &= 14 * PipeDiameter_{inch} \\
 &= 28 \text{ mm}
 \end{aligned}$$

Thus,

$$if [PM] \leq Distance_{mm}, P \in \{Pipe\}$$

To illustrate the previous statement, in Figure 35, P doesn't belong to the pipe but P'' does. The method assures the extraction of every point belonging to the pipe and no more.

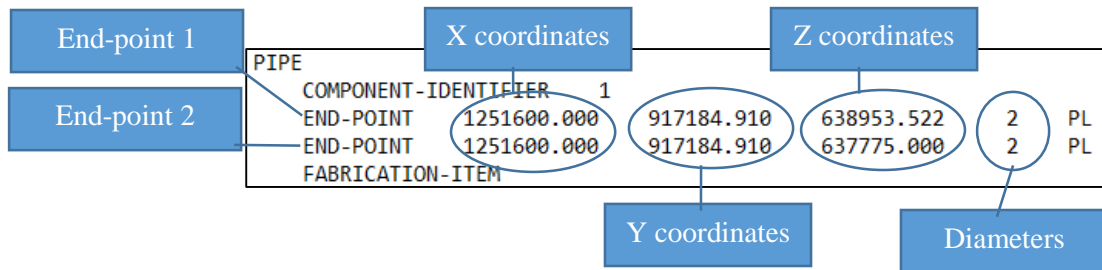


Figure 36: Geometric definition of a pipe in a PCF

As a result, Figure 37 illustrates the result of the segmentation of a straight pipe in the SfM software.

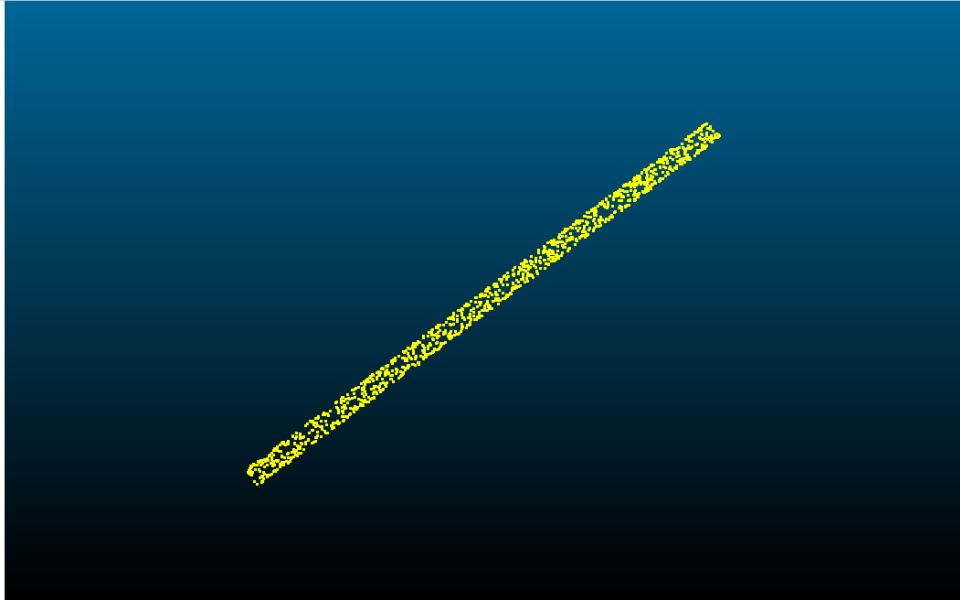


Figure 37: Segmented straight pipe in the SfM Software

4.5.2 Segmentation of Elbow

An elbow is a complex component as it can be very diverse with different angle, different curvature and length. The description of an elbow in a PCF is only described with three points:

- End-point A with its diameter
- End-point B with its diameter
- Center-point C

Most elbows are 90° -elbows. In order to be as general as possible a random elbow is built in Figure 38. The angle of the elbow is described with θ value.

The elbow can also be seen as an arc of a circle. The arc belongs to a circle and has a range angle, called γ in Figure 38. In 3D, a circle is defined with three attributes:

- A centre
- A normal vector
- A radius

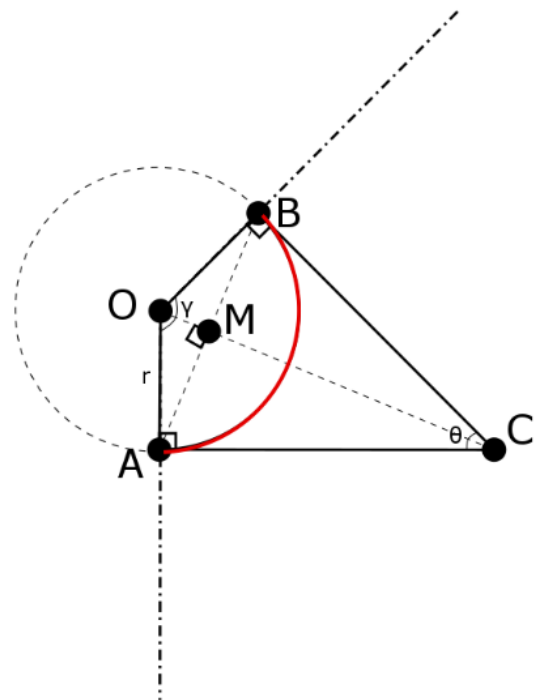


Figure 38: Representation of an elbow in 2D

Consequently, with the inputs provided in the PCF, we need to determine those three attributes. The code of the function is given in Appendix A. The first step consists in calculating the mid-point M defined as follows:

$$M = \frac{A + B}{2}$$

Secondly, the two tangents to the arc [AC] and [BC] are defined. By applying the cross product on the two segments the unit normal vector of the circle in the 3D space is obtained:

$$\vec{n} = \frac{\overrightarrow{AC} * \overrightarrow{BC}}{\|\overrightarrow{AC}\| * \|\overrightarrow{BC}\|}$$

The next step is to extract the angle of the elbow. In the pipe industry, every elbow is denominated with its angle, (i.e. Elbow 90, Elbow 135). This value stands for the angle between the two vectors \overrightarrow{CA} and \overrightarrow{CB} and is called θ in Figure 38. Below is the equation used to determine θ .

$$\theta = \arccos\left(\frac{\overrightarrow{CA} \cdot \overrightarrow{CB}}{\|\overrightarrow{CA}\| * \|\overrightarrow{CB}\|}\right)$$

Then, γ , which will be used to define the angular constraint to apply on the points from the point cloud, has to be determined from θ . As noted in Figure 38, the following relation can be used:

$$\gamma = \pi - \theta$$

Every angle has been determined. Consequently, the coordinates of the center of the circle still have to be found out. To do so, the distance from an end-point to the mid-point M, which was calculated previously, is computed. This distance is named HalfChord:

$$\overrightarrow{HalfChord} = \overrightarrow{AM}$$

Then, the unit vector along \overrightarrow{CM} is extracted. Note that this vector also goes through O, center of the circle.

$$\overrightarrow{VectorToCenter} = \frac{\overrightarrow{CM}}{\|\overrightarrow{CM}\|}$$

The distance $\|\overrightarrow{OM}\|$ is computed according to the trigonometric properties in the triangle defined by A, O and M.

$$\|\overrightarrow{OM}\| = \|\overrightarrow{HalfChord}\| * \tan\left(\frac{\theta}{2}\right)$$

However, this equation just provides the distance from M to O. Point O coordinates are still unknown. The previous equations make possible the determination of x, y, and z coordinates. Point O is calculated in this equation:

$$O = M + \|\overrightarrow{OM}\| * \overrightarrow{VectorToCenter}$$

The determination of the radius is then straightforward:

$$\begin{cases} \overrightarrow{radius1} = \overrightarrow{OA} \\ \overrightarrow{radius2} = \overrightarrow{OB} \end{cases}$$

Note that the two radius can have a few gaps. To assure a good representation of the elbow, the average of $\overrightarrow{radius1}$ and $\overrightarrow{radius2}$ is selected for radius.

The arc of the circle is now entirely known and defined by its geometric characteristics. The torus shape in the 3D point cloud can now be computed. Indeed, the points that belong to an elbow are part of a torus shape constrained by a range angle. As explained above, the range angle is equal to the γ value. The torus diameter is retrieved from the PCF and corresponds to the value associated to the end-points.

Every point belongs to the 3D space. Thus, a projection of the point onto the plane defined by the arc of the circle must be computed. As shown in Figure 39, a point P is divided into its vertical and horizontal projection. The green line represents the vertical distance to the plane. This length is parallel to the blue line, standing for the normal vector of the plane. This vector was determined previously and was called \vec{n} . Finally, the red line is the total distance from a point on the plane to the point-of-interest P.

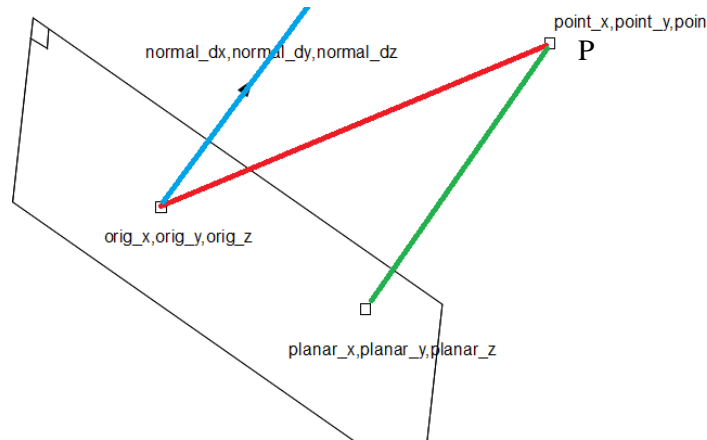


Figure 39: Projection of a point P on a plane in a 3D space

The procedure for retrieving elbow's points is detailed in Appendix A. The representation of the elbow is shown in Figure 40. The distance to be calculated is the distance from point P to the arc going from A to

B. In order to get this length, the distance from point P to the plane is obtained using the following dot product:

$$DistanceFromP2Plane = \overline{OP} \cdot \vec{n}$$

Afterward, one still needs to calculate the distance from the projection of P onto the plane to the arc going from A to B. This point is point q and is determined with the following expression:

$$q = P - (\vec{n} \cdot \overline{OP})\vec{n}$$

At this point, the distance from that point to the arc can be computed.

$$DistanceFromq2Arc = \|\overline{Oq}\| - RadiusOfCircle$$

Finally, the distance coloured in red on Figure 40 and going from P to this arc is given in the following equation.

$$DistanceFromP2Arc = \sqrt{DistanceFromP2Plane^2 + DistanceFromq2Arc^2}$$

This value can be compared to the radius of the elbow specified in the PCF. Basically, if the distance is smaller than the radius – plus a specific tolerance – then the point passes the test and goes to the angular criterion. Otherwise, the point is excluded and doesn't belong to the elbow.

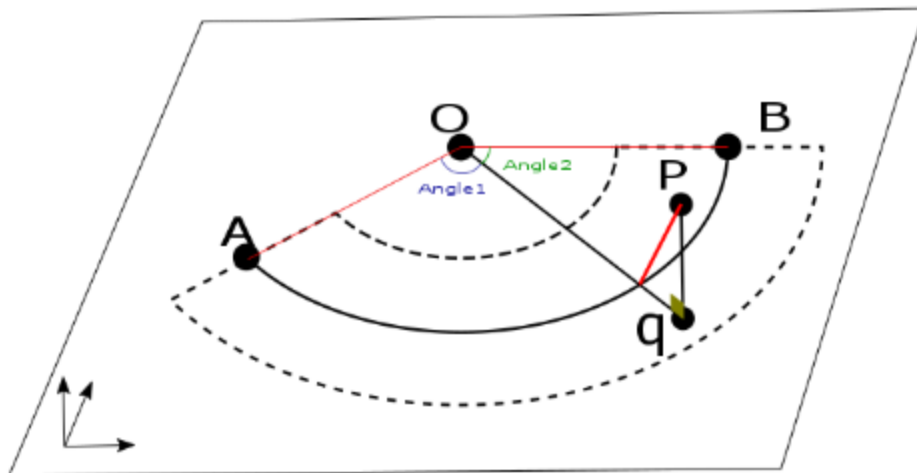


Figure 40: Criteria for the elbow

For the selected points, their position has to be analyzed in comparison to the elbow. Thus, the determination of its position around the circle is crucial to select the point. To do so, the angle between $\overline{radius1}$ and \overline{OP} is first calculated. Second, the angle between $\overline{radius2}$ and \overline{OP} is measured. Finally, the sum of the two

angles is compared to the angle of the elbow γ . The equations used to obtain those values are given in the following equations.

$$\begin{cases} Angle1 = \widehat{AOP} = \text{acos}\left(\frac{\overrightarrow{radius1} \cdot \overrightarrow{OP}}{\|\overrightarrow{radius1}\| * \|\overrightarrow{OP}\|}\right) \\ Angle2 = \widehat{BOP} = \text{acos}\left(\frac{\overrightarrow{radius2} \cdot \overrightarrow{OP}}{\|\overrightarrow{radius2}\| * \|\overrightarrow{OP}\|}\right) \end{cases}$$

If the point belongs to the elbow, then the sum of the angle is approximately equal to γ , which can be written as:

$$\text{For } P \in \text{Elbow}, (Angle1 + Angle2) \approx (\gamma + \varepsilon)$$

where ε is a very small value taking into account the potential inaccuracy in the PCF data. In the code, the arbitrary value of 0.01 is used for ε . If this criterion is met, then point P belongs to the elbow and is represented in the point cloud of the component (e.g. Figure 41).

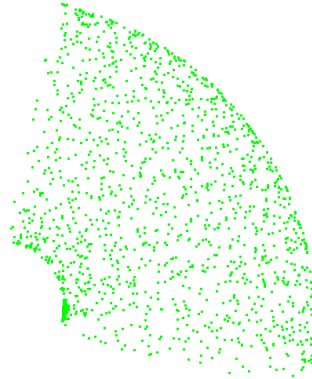


Figure 41: Segmented elbow

4.5.3 Segmentation of Tee

In the piping industries, a tee is a very common fitting component. It is often used to divide or combine fluid flows. Tees can connect pipes of different diameter, change the direction of a pipe run, or both. Most common tees have a similar design to the one in Figure 44.

In PCFs, tee's geometric information is:

- two end-points with their diameter

- one branch-point with its diameter
- the centre-point

The two end-points define the main axis of the tee. The branch-point is the other part of the tee, and the centre-point is the intersection between the branch pipe and the main pipe of the tee.

Consequently, a tee can be seen as two straight pipes interlocking each other and converging at the center-point. The skeleton of a tee has been represented on Figure 43. Hence, segmentation of a tee is processed as the segmentation of a first straight pipe going from end-point 1 to end-point 2, and the segmentation of a second straight pipe going from branch-point to center-point. This process remains robust to any angle for the branch.

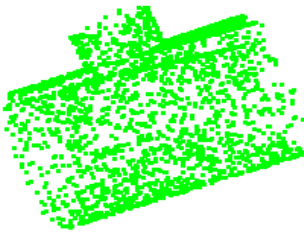


Figure 43: Segmented tee

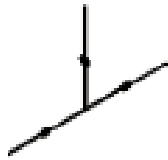


Figure 42: Skeleton of a tee



Figure 44: Tee

The curvature from the main axis to the branch can change from a tee to another. To address the issue, we add a tolerance distance so that a point that belongs to the curve of a very curved tee would be segmented as well in the tee. Unfortunately, PCFs don't mention this specificity. But since the segmentation is processed with the 3D model only, there is no risk of clutter points interfering in the process. Increasing the tolerance to capture the points in the corner of the tee doesn't jeopardize the accuracy of the process. Figure 42 above represents the point cloud of a segmented tee.

4.5.4 Segmentation of Valve

Theoretically, valves seem like tees because they can be constructed from three end-points and a centre-point. Instead of branching out from a branch-point, valves are equipped with a spindle terminating with a tap used to regulate the flow of liquid/gas in the pipe spool. A physical representation of a typical valve can be found in Table 7.

Unlike tees, the PCF description of valves (Figure 45) is less straight-forward to process, because neither their centre-point nor branch-point is mentioned. Rather, their spindle direction is stated using bearing coordinates.

```

VALVE
END-POINT          79750.0260    -40478.1017    36974.4784    12.0000    FL
END-POINT          80099.2760    -40128.8517    37468.3925    12.0000    FL
SKEY   VGFL
ITEM-CODE   SS150-3922
ITEM-DESCRIPTION   Globe Valve, 150 LB, RF, ASME B16.10, Hand Wheel
FABRICATION-ITEM
PIPING-SPEC   SS150
TRACING-SPEC
COMPONENT-ATTRIBUTE1   BOMCOLUMN_Material_
COMPONENT-ATTRIBUTE2   BOMCOLUMN_SCHClass_150
WEIGHT   410.001814388098
SPINDLE-DIRECTION   SOUTH
DIRECTION   SOUTH 30 EAST

```

Figure 45: PCF description of a valve

In the example above, the spindle direction is “South 30 East” which only points out the direction of the theoretical branch-point (or spindle-point) but doesn’t provide its exact location. Furthermore, the bearings are bounded to the main core of the valve which, itself, is randomly located in the global coordinate system. The bearing angle is therein a local angle instead of being global.

Before going further on the bearing functioning, Figure 46 illustrates the bearing coordinate system with regard to the Cartesian one. A vector (in blue) is placed in the coordinate system and projected onto the (\vec{x}, \vec{y}) plane.

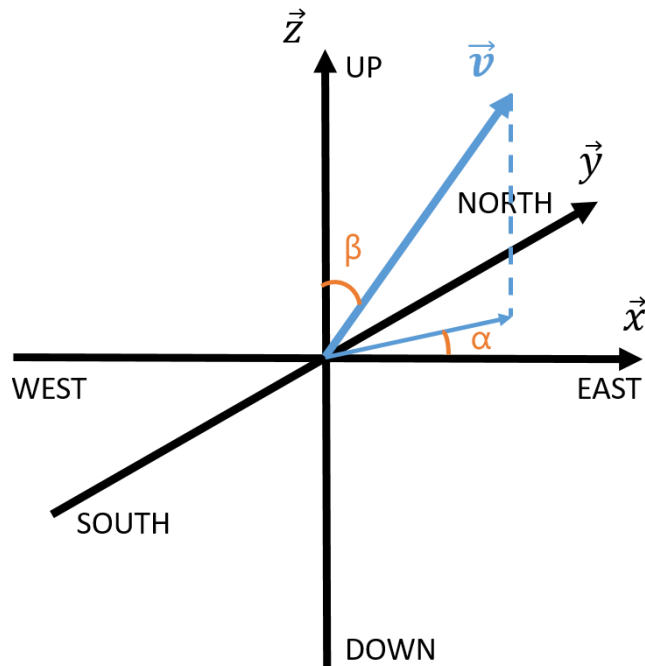


Figure 46: bearings coordinate system

PCF bearings are segmented as follow:

1. Bearing 1
2. Angle between bearing 1 and bearing 2
3. Bearing 2

In the example plotted on Figure 46, \vec{v} bearing would typically be: EAST α NORTH, where α is the angle value in degree and smaller or equal than 45° . Otherwise, bearing 1 and bearing 2 would be inverted to become, in the example above, NORTH α EAST. The 2 bearings chosen to represent the direction of the valve spindle are arbitrary and, for instance, \vec{v} could have been projected onto the (\vec{y}, \vec{z}) plane and another angle would have been prompted in the PCF.

However, bearings describe the local direction of the spindle, based on the initial orientation of the valve (direction of the axis bounding the two end-points). From this description, the challenge is to retrieve the global angle in order to determine the global direction to be able to obtain some coordinates for the spindle-point. This will enable to generate an envelope around the spindle axis to extract its model points.

Because the spindle is perpendicular to the valve vector and is located at the valve's center, one first calculates the valve's center-point as the midpoint of the 2 end-points. Consequently, as explained above, according to the plane of projection, the angle will not be the same. Three cases are thus differentiated depending on which bearing direction is null. Let us study the situation where the \vec{z} bearing is left equal to 0. We need to find a vector that is perpendicular to the valve vector. The following equations translate the previous sentence. Let us use (x, y, z) to denote the unknown vector and (a, b, c) for the valve vector.

$$(x, y, z) \text{ perpendicular to } (a, b, c) \Leftrightarrow \text{dotprod}((x, y, z), (a, b, c)) = 0 \Leftrightarrow \begin{cases} x = \text{bearing}_{\vec{x}} \\ y = \text{bearing}_{\vec{y}} \\ z = -\frac{x * a + y * b}{c} \end{cases}$$

We then assume the length of the spindle to be three times the valve's main core length.

Finally, denoting (u, v, w) being the centre-point, the spindle-point coordinates can be derived as follows:

$$\begin{cases} x_{spindle} = u + \frac{x}{\|(x, y, z)\|} * \text{length}_{spindle} \\ y_{spindle} = v + \frac{y}{\|(x, y, z)\|} * \text{length}_{spindle} \\ z_{spindle} = w + \frac{z}{\|(x, y, z)\|} * \text{length}_{spindle} \end{cases}$$

When the bearing is on one of the main axis (e.g. NORTH), the spindle point is calculated by simply estimating the offset to the centre-point. Its coordinates are computed in the following equation:

$$\begin{cases} x_{spindle} = u + bearing_{\bar{x}} * length_{spindle} \\ y_{spindle} = v + bearing_{\bar{y}} * length_{spindle} \\ z_{spindle} = w + bearing_{\bar{z}} * length_{spindle} \end{cases}$$

Then are shown some bearings examples:

$$\text{NORTH} = \begin{cases} bearing_{\bar{x}}=0 \\ bearing_{\bar{y}}=1 \\ bearing_{\bar{z}}=0 \end{cases} ; \text{SOUTH 30 DOWN} = \begin{cases} bearing_{\bar{x}}=0 \\ bearing_{\bar{y}}=-1/2 \\ bearing_{\bar{z}}=-\frac{\sqrt{3}}{2} \end{cases} ; \text{WEST 45 UP} = \begin{cases} bearing_{\bar{x}}=-\frac{\sqrt{2}}{2} \\ bearing_{\bar{y}}=0 \\ bearing_{\bar{z}}=\frac{\sqrt{2}}{2} \end{cases}$$

Once the bearing has been computed, and the spindle point has been calculated, the same process used for tees is applied. Note that it is not our intention to necessarily segment the entire spindle with the tap. Rather, we intend to retrieve enough points from the spindle to control the spindle's position because this position can cause conflicts. For instance, when placing it into a complex clustered module, a wrong spindle position may engender rework to fix a conflict. Figure 47 shows the segmented valve point cloud.

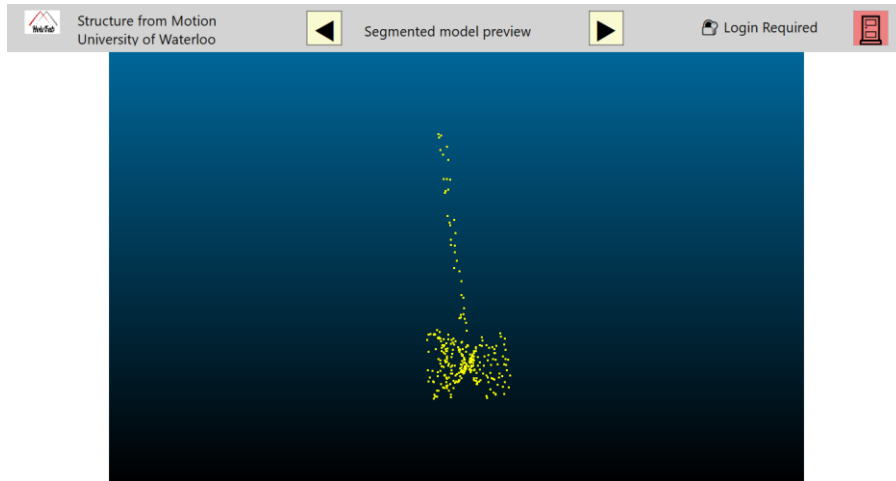


Figure 47: Segmented valve point cloud

4.5.5 Segmentation of Eccentric Reducer

As presented in Table 7, an eccentric reducer is a particular component that, at the same time, reduces/increases the spool diameter and shifts the axis to a parallel one. In term of segmentation, eccentric reducers don't create a major problem, but they will be challenging for the upcoming chapter. Their distinctive feature is that they only have two end-points but are not a cylindrical component. Further explanations are provided in Chapter 5. Figure 48 shows an example of a segmented eccentric reducer point cloud.

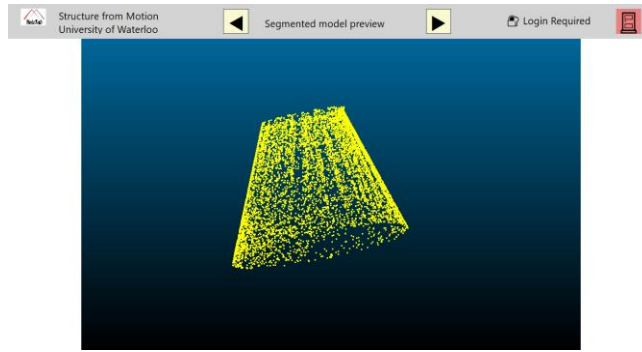


Figure 48: Example of a segmented eccentric reducer point cloud

4.5.6 Segmentation of Flange

Flanges are piping components used to make a connection between two pipes or between a pipe and any type of fitting. Most of the time, flanges are placed at special locations to facilitate dismantling of piping assemblies or specific components (such as valves). Used as connectors, they are successfully used for high pressure piping applications.

Two flanges are mechanically joined together through their flat surface via a certain number of bolts (Figure 49). A gasket, used as a seal, is placed in between to maintain the impermeability of the system. The other flange end is usually welded to its following piping component.

Flanges are assigned a pressure class which, owing to tables, provides the outside diameter given the nominal pipe size (NPS). Every standard has its own tables. SfM utilizes the ASME/ANSI B16.5 tables which can be found in References (Flanges and Bolt Dimensions ASME/ANSI B16.5 - Class 150 to 2500).



Figure 49: Representation of flanges on a pipe spool assembly

As shown in Figure 50, the PCF mainly provides: (1) the two end-points with their respective NPS, (2) and the pressure class (highlighted in yellow).

```

FLANGE
END-POINT      -9889.1740    17699.4716    5105.4616    4.0000    FL
END-POINT      -9812.9740    17699.4716    5105.4616    4.0000    BW
SKEY    FLWN
ITEM-CODE    SS150-1150
ITEM-DESCRIPTION    FLANGE WN, 150 LB, RF, ASME B16.5, ASTM A182 Gr F304
FABRICATION-ITEM
PIPING-SPEC    SS150
TRACING-SPEC
COMPONENT-ATTRIBUTE1    BOMCOLUMN_Material_
COMPONENT-ATTRIBUTE2    BOMCOLUMN_SCHClass_150
WEIGHT    7.48435090265808

```

Figure 50: PCF description of a flange

Pressure class is given in terms of class which, for instance, goes from 150 to 2500 in the ASME B16.5 standards. Using ISO 7005 Pressure Nominal, Table 8 provides the associated pressure ratings in bars. The main advantage of using class numbers is that they provide a proportional relationship between different class numbers.

Table 8: Piping Class Ratings based on the ASME B16.5 class and the corresponding ISO 7005 Pressure Nominal ratings

Flange Class	150	300	400	600	900	1500	2500
Flange Pressure Nominal (in bars)	20	50	68	110	150	260	420

Class tables provided by the ASME B16.5 thereby can be used to reliably extract the outside diameter of any flange. For the time being, each table is directly stored within the SfM code. However, an optimal way would be to use an external file that would gather every existing standards tables. The program would then automatically access the suitable standard file and use the proper pressure class table.

Once the outside diameter is retrieved, the flange can be segmented. The flange is considered as a cylindrical component and follows the same segmentation rules as described in 4.5.1. However, the diameter taken into account for the segmentation is the outside one as it surrounds the entire flange. Figure 51 displays a segmented flange.



Figure 51: Segmented flange

Point cloud representation of flanges may not allow a clear visualization of bolt holes. Furthermore, bolt position is not mentioned in the PCF which prevents the described methodology from controlling their alignment. That is why, manual quality control, as performed in Figure 52, is still required to check the bolt holes alignment on flanges.

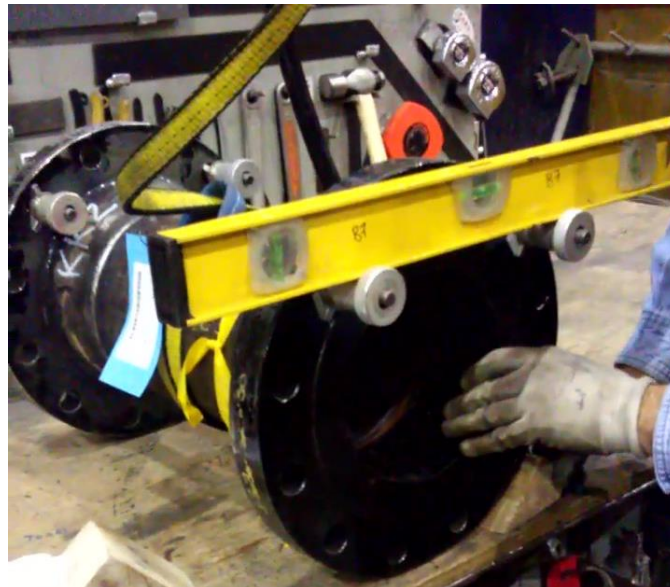


Figure 52: Bolt holes' alignment control performed by QC personnel

4.6 Conclusion

Segmentation of components in a pipe spool assembly is a crucial preprocess on the path of a step-by-step assembly quality control. As shown in Figure 29, as soon as a model has to be derived, segmentation is performed to correctly generate the point cloud.

Although being called in the SfM application software after the MRM process, the segmentation process is explained earlier in this thesis, because it must be considered as a prerequisite for any model derivation. Indeed, before applying any rule on the model to extract the required components, we need to make sure that each one can be properly generated.

The objective of this chapter was to correctly detail how each component can be independently retrieved from the whole-spool design point cloud. As demonstrated, the vast majority of the piping components can be segmented in the developed solution due to the combination of the 3D model and the Piping Component File. The two files are associated to bring in the PCF geometric data into the 3D model representation. However, some components remain unconsidered either because of the lack of information in the PCF or because of their particular shape. Besides olets, the method is not looking at any pipe support such as pipe clamps, pipe shoes, pipe saddles or pipe cradles. Many times, those components are present in the as-built scenes and cannot be avoided when scanning. As a result, they are likely to cause occlusion in the scan point cloud. In a few situations, the author has encountered pipe supports in the PCF, but modelling them remains complex. As they are assigned a specific location in pipe assemblies, pipe supports could potentially be segmented in order to improve the potential of the described process.

5 Derivation of Minimum Required Model

[Most of this chapter is extracted from a peer-reviewed publication by the author (Jeanclos et al. 2018)].

Following the thorough presentation of the segmentation process, this chapter focuses on the methodology for deriving the Minimum Required Model (MRM). It presents the theoretical concepts and general objectives of the method as well as its implementation. The major objective of the proposed method consists in reducing the number of components to be scanned for a 3D vision based quality control. The determination of these components is achieved to prevent any geometrical ambiguity in the MRM; that is, by ensuring the final derived assembly geometry, the whole assembly compliance is guaranteed. The entire process of MRM derivation is shown in Figure 53. The process is divided into three different sections. In Section 5.1, the basic concepts of the method are explained which includes explaining the use of Piping Component Files as well as a description of the appropriate categorization of components into Reference (REF) and Addition (ADD). In Section 5.2, the concept of Solid of Revolution (SOR) is presented and the determination of REF/ADD in SOR is explained. Finally, Section 5.3 studies how the MRM is derived using REF and ADD components.

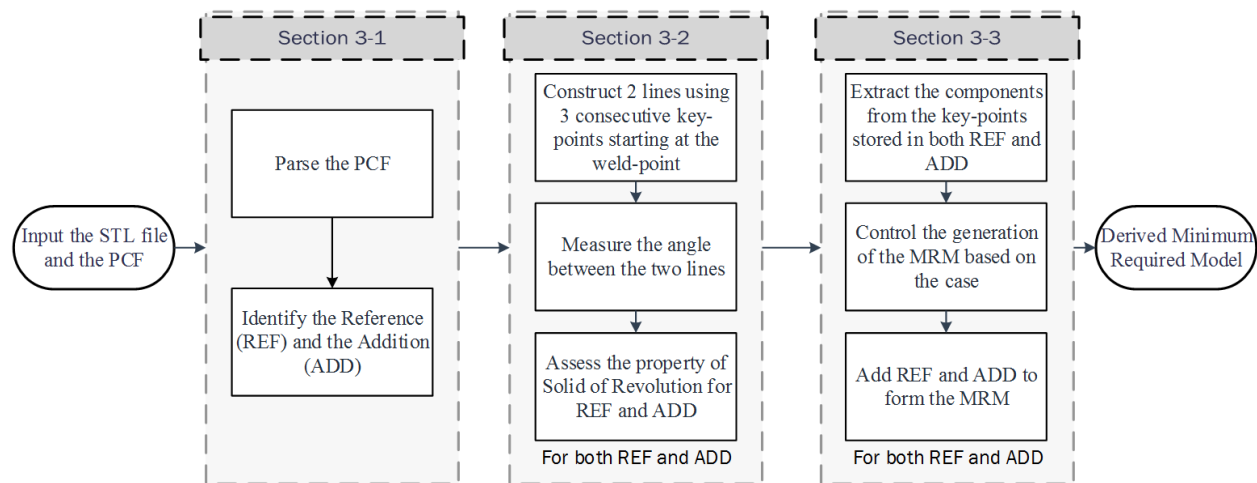


Figure 53: Minimum Required Model Workflow

5.1 Parsing of the Piping Component File and Selection of Reference and Addition

A major aspect of the MRM algorithm is its use of Piping Component Files (PCFs) and Stereolithographic files (STLs). Combining the 3D model, in the format of a software independent file such as a STL along with a PCF makes the method completely software independent. It solves the software compatibility challenge encountered by construction companies (Chien et al. 2014). The first part of this section details how a PCF is parsed to extract the appropriate information with regard to the use made by the MRM. The precise identification of Reference and Addition is explained afterwards.

5.1.1 Parsing of the Piping Component File

As discussed in the thesis background chapter, a common practice in the piping industry is to transmit the PCF of the assembly along with the 2D drawings and/or the 3D model.

Transmitting the PCF can be seen as a safe means to send the drafting information because it contains all the components information including their type, coordinates, size, shape, and weld location. The file is segmented into paragraphs where each one represents a different component. Explanations are provided in Chapter 4.

In order to gather the proper components to derive the MRM, the components' key-points are to be computed. The combination of the key-points allows a complete description of the component. For example, a pipe has two key-points while an elbow needs three key-points to be modelled. A representation of the most common components with their associated key-points is provided in Figure 54. As noticed in Chapter 4, a key-point can either be an end-point, a centre-point or a branch-point. Each type of component has its own key-points label.

Using the key points, an envelope is calculated and encompasses the corresponding component in the STL file. Thus, the envelope provides the possibility to segment a specific component out of the assembly. Consequently, employing PCFs allows users to select and segment only the components they are working with as opposed to the complete assembly.

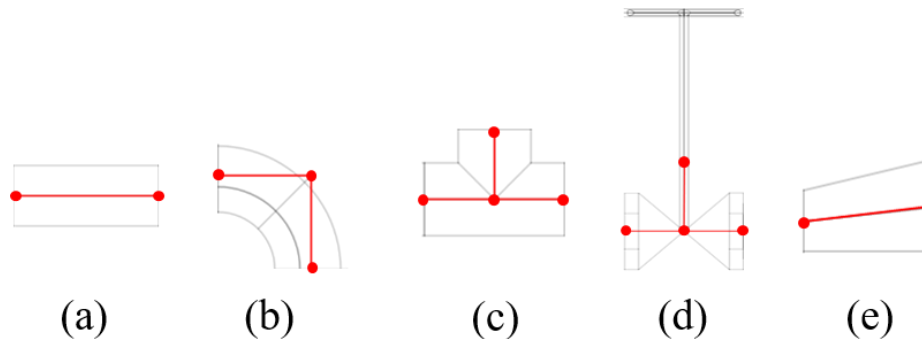


Figure 54: Demonstration of piping components using key points and the hypothetical line to represent each component: (a) a pipe or any other cylindrical component with 2 end-points, (b) an elbow with two end-points and one center-point, (c) a tee with two end-points, one center-point and one branch-point, (d) a valve with two end-points, one center-point and one branch-point, and (e) a reducer-eccentric with two end-points.

The rest of the piping components presented in Table 7 are composed of two key-points only and thus may be represented as a straight pipe (cylindrical components).

5.1.2 Identification of Reference and Addition.

The derivation of the Minimum Required Model is to be integrated with a stepwise process of fabrication, meaning that workers would control the compliance of the fabricated assembly at each step. One of the

important steps for pipe fitters is to choose the sequence of the assembly. The selection of the sequence depends on a number of variables including maximizing number of roll welds. As roll weld machines are around 1.5m high, the idea is to limit the height of the current assembly as long as possible to remain smaller than the tolerable length. Roll welds allow for ergonomic and efficient welding by welders and thus same money over regular welding for a joint. When, by following linearly the flow of the assembly, the height is greater than the acceptable one, the fitter may rather choose to switch to another area of the assembly to enable the welder to perform a roll weld.

In the example provided in Figure 55, an elbow is analyzed to this extent. Let us consider all the other areas assembled already. The question is whether the elbow encircled should first be welded to the area on its left or to the one above. The first solution would involve rotating the left part around the axis described by the arrow pointing left. As it can be seen, the height would be way too long to rotate the resultant assembly. On the contrary, rotating the above assembly around the arrow pointing up would be totally feasible as the required height would be close to 50cm. Considering the dimensions, unlike the second option, the first option would not be in accordance with a roll weld. Therefore, a good strategy would be to first roll weld the elbow to the above assembly, and then completely manually weld the resultant assembly to the left part.



Figure 55: Photography of the steel pipe assembly located in the UW lab. The green oval surrounds the elbow. Each green arrow is pointing at an area of the assembly.

Regardless of the sequence, there would always be a component, or group of components, taken as Reference (REF) which would be leveled, and the next component, or group of components, would be

added, called Addition (ADD). Figure 56 further illustrates the concepts of REF and ADD from a real example that was photographed during a visit in the AIW fab shop. Reference – part (a) in (2) and blue part in (3) – consists of a flange, a pipe and an elbow; Addition – part (b) in (2) and red part in (3) – is a straight pipe. The weld point – part (c) in (2) – is located between the elbow and the pipe.

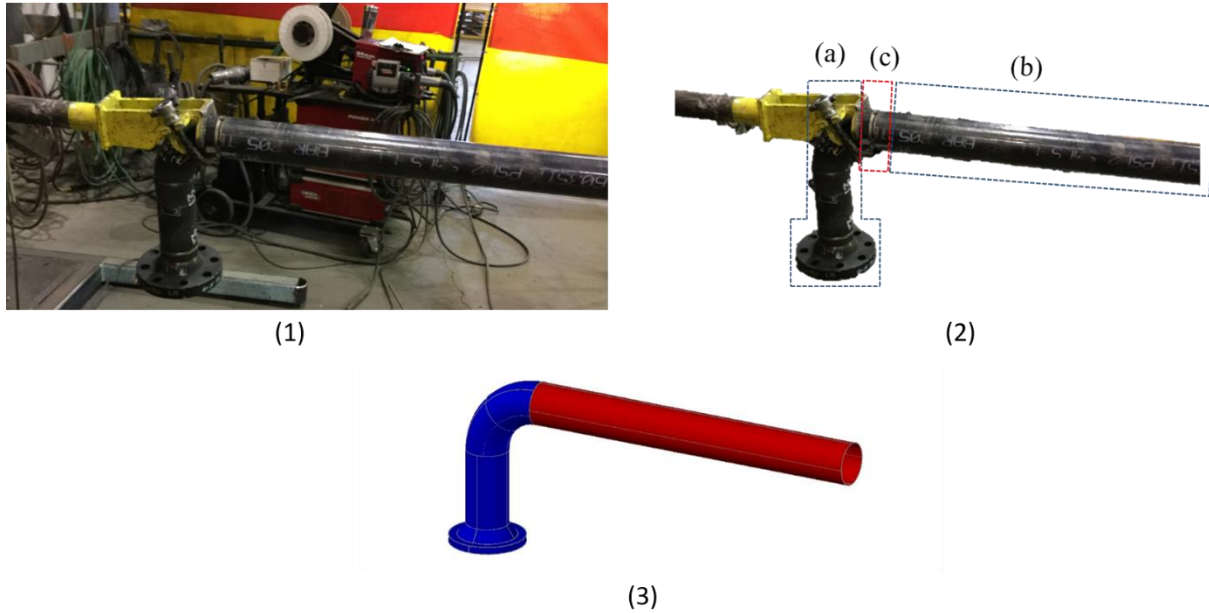


Figure 56: Illustration of REF, ADD and connection point on a pipe spool assembly in a fabrication facility. It respectively presents: (1) the original photograph, (2) the extracted assembly being tackled, (3) and the corresponding 3D model. (a) Reference consists of a flange, a pipe, and an elbow. (b) Addition is a straight pipe (c) The weld point shared between REF and ADD.

As previously mentioned, the sequence by which an assembly is built depends on the worker. For example, in Figure 56, a worker could decide to first weld the pipe in the box (a) to the flange in box (a). Then, he would weld the elbow in the box (a) and the pipe in the box (a). Finally, the pipe in the box (b) and the resultant sub-assembly of box (a) would be assembled. The proposed method is robust to any combination of REF and ADD since the algorithm takes advantage of the welding point between REF and ADD. It does not rely on the components available in each group. The weld point accounts for the starting point of REF and ADD. It is worth noting that a weld point is shared by two components and thus belongs to both REF and ADD. Also, it is the unique shared point between REF and ADD (Figure 56 box (c)).

Any given assembly can be represented in a tree, binary sorted according to the sequence of the components. The tree is made of nodes where each stands for a key-point. It begins at the root with the selected weld, which is also a key-point shared between REF and ADD, and goes down until all key-points are stored. This representation improves the mental visualization of pipe assemblies and facilitates the explanation of the methodology. Each step possesses its own binary search tree (BST) as the root is given by the selected

weld, and the tree will stop based on the components selection. As a result, the tree construction can be considered as a dynamic process. Once the worker selects his REF and his ADD, the weld is automatically found by comparing the common key-points. The tree is thus initiated by positioning the weld point at its root. Reading through the PCF, one of the branch is first analyzed. The other component's key-point is positioned as the child of the root. The algorithm then searches for the approximate same key-point within the PCF and extracts the second component's key-point which becomes the child of the previous node. If a component has more than two key-points, then a branch is created, meaning that the first key-point will have two children: the second key-point, and the branch-point.

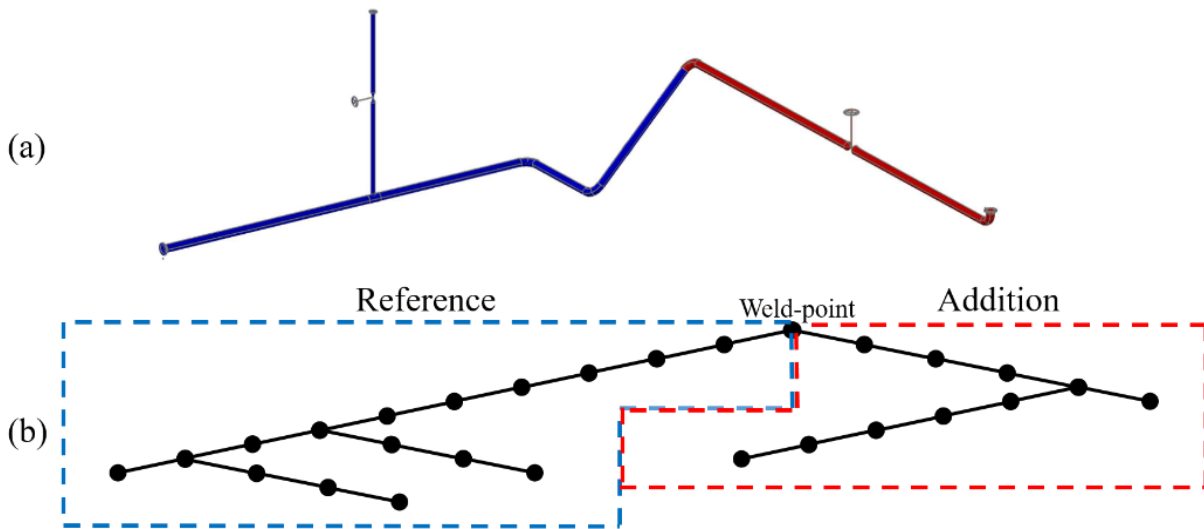


Figure 57: Construction of the graph using the 3D model and the PCF file. (a) 3D model assembly. (b) Constructed graph. Each node corresponds to a key-point.

Figure 57 illustrates the tree construction of an assembly that has pipes, elbows, flanges, a tee, and two valves. The algorithm starts reading the key-points information from the weld point and follows the flow of the assembly by reading into its associated PCF. As shown, the weld point is located between the blue part (REF) and the red part (ADD), between a pipe and an elbow. Considering the Reference tree (in blue), the weld moves on to the adjacent point, standing for the second end-point of the pipe. Then, it follows linearly the path with an elbow, a pipe, an elbow and another pipe. At that point, the first end-point of the tee is divided into two children: one is the branch-point and the other is the second key-point.

Depending on the properties of the selected assembly, the tree generation may be terminated before every component gets analyzed and stored. Indeed, along the constitution of the tree on each side, REF and ADD are continuously evaluated to determine whether or not they are a Solid of Revolution (SOR). This is the objective of the two upcoming sections.

5.2 Determining the Property of Solid of Revolution for Reference and Addition

5.2.1 Solid of Revolution

To evaluate whether REF/ADD is a Solid of Revolution (SOR) or not ($\overline{\text{SOR}}$), a clear explanation has to be given around the concept of solid of revolution. By definition, a solid of revolution is an object that can be generated by rotating any arbitrary shape around a straight line (Jeffery 1915). Ten 3D models are displayed in Figure 58. The first four models have been generated by rotating respectively a rectangle (a), a trapeze (b), a spline (c), and an association of lines (d) around a straight line corresponding to the axis of symmetry. Physically, solids of revolution have a specific inertia property described by the shape of the inertia tensor. Using the Cartesian coordinate system and considering the cylindrical symmetry to be around the \vec{z} -axis, the inertia tensor is thereby defined as followed:

$$I = \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_1 & 0 \\ 0 & 0 & I_3 \end{bmatrix}$$

where I_1 and I_3 are the principal moments of inertia. For a SOR, the products of inertia are null and the principal moments of inertia around \vec{x} and \vec{y} are equal. More information related to moments of inertia and specifically to the inertia for solids of revolution can be found in (Diaz et al. 2006).

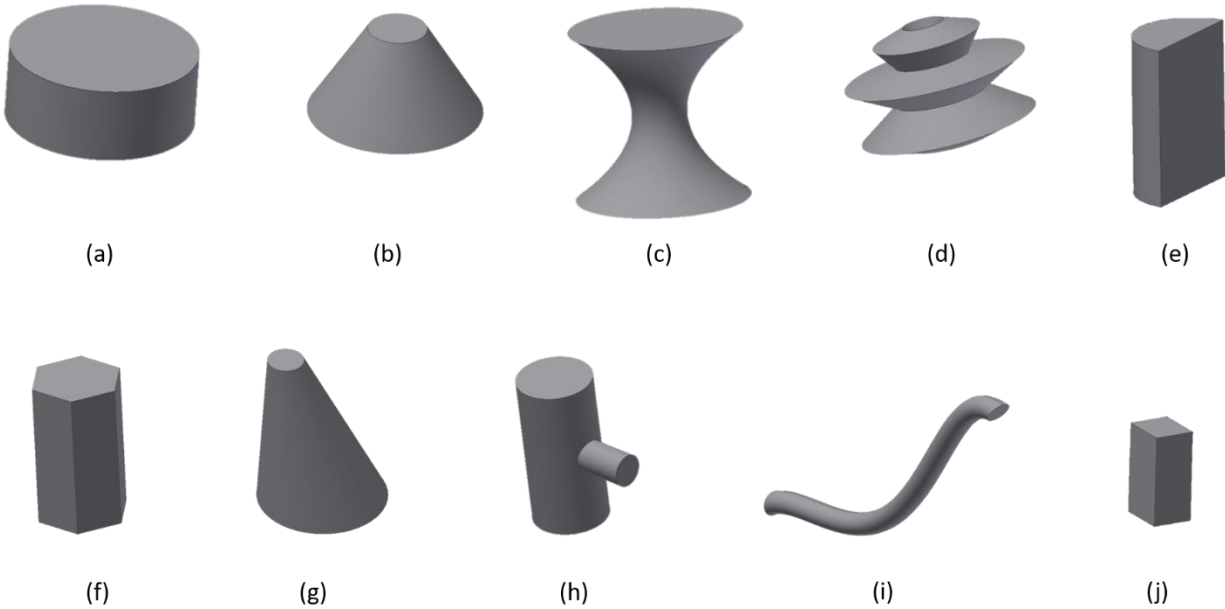


Figure 58: 3D models of SOR and $\overline{\text{SOR}}$. (a-d) are solids of revolution. (e-j) are not solids of revolution.

According to the previous definition and the followed explanations; every piping cylindrical object (pipe, flange, cap, and coupling) is a SOR. Elbows, tees, valves or eccentric reducers are not SOR ($\overline{\text{SOR}}$).

Considering that piping components are assembled by aligning them together (sharing the same axis such as shown with \vec{w}_1 and \vec{u}_1 in Figure 60), the SOR becomes a conservative property. This means that assembling two SOR piping components will result in a SOR assembly. Also, if a \overline{SOR} piping component is added to a SOR component the assembly becomes \overline{SOR} . Equation 1 summarizes this concept:

$$\forall (S_1, S_2) \in \{SOR\}, \forall S_3 \in \{\overline{SOR}\}, \begin{cases} (S_1 + S_2) \in \{SOR\} \\ (S_1 + S_2 + S_3) \in \{\overline{SOR}\} \end{cases}$$

,where S_1 and S_2 denote to a piping component that is a solid of revolution, and S_3 denotes to a component that is not a solid of revolution. To illustrate Equation 1, Figure 59 shows two assemblies: (a) the first one combines two solids of revolution that have been joined along their axis of symmetry; (b) the second presents a semi-cylinder (\overline{SOR}) associated with a solid of revolution along their axis of generation. As explained by Equation 1, the SOR property remains in (a) but is lost in (b).

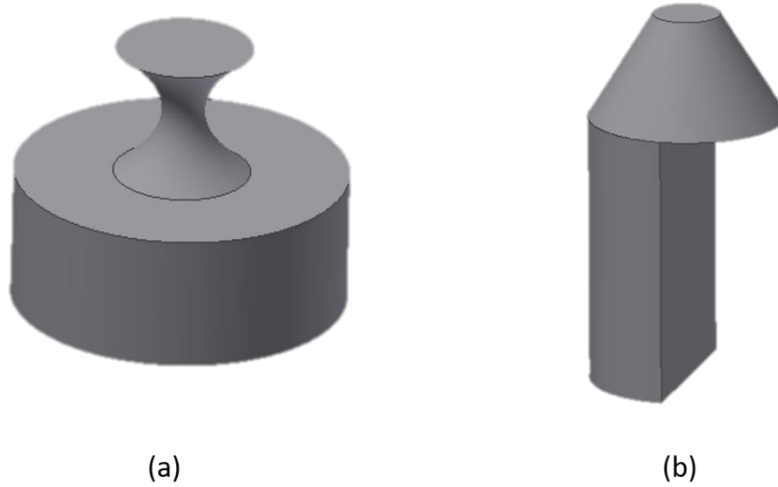


Figure 59: Combination of: (a) 2 SOR along their axis of symmetry, and (b) 1 SOR and 1 \overline{SOR} .

Fitting piping components can be envisioned as previously described. Indeed, each component is or is not a SOR and fittings are, for most of them, done by aligning the two axis of generation. As a result, the next section describes the applied method to assess this property on REF and ADD.

5.2.2 Assessing the Solid of Revolution Property for Reference and Addition

Finding out the SOR property permits the creation of the key-points list for REF and ADD. The combined list will be used to extract the MRM. Algorithm 1 describes how this process is performed. The SOR property is defined as a Boolean variable, which is initially set as true, and can be changed to false for specific conditions.

Algorithm 1. Pseudo code for Solid of Revolution (SOR) determination and extraction of key-points for one part of the selected assembly (REF or ADD)

```

SOR=true
i=1
P0 is the weld-point
KeyPointsList = {P0}
while Pi has at least 1 child
if Pi has 2 children
    SOR=false
    break
else
    Calculate the angle between  $\overrightarrow{P_{i-1}P_i}$  and  $\overrightarrow{P_iP_{i+1}}$ 
    if (angle>0)
        SOR=false
        break
    end if
end if
add Pi to KeyPointsList
i=i+1
end while
if all Pi are in KeyPointsList
    KeyPointsList = {P0}
end if
KeyPointsList
end

```

The appraisal of the SOR property is accomplished by evaluating the angle between two consecutive lines virtually drawn by bounding two consecutive key-points together. Those lines can qualitatively be depicted by the lines connecting two consecutive points in the tree described in Figure 57. The next bullet points sequentially describe the aforementioned algorithm. The entire code of this process can be found in Appendix B.

- First, the list gathering the key-points is initiated and filled with the weld point P₀.
- The first neighbour (P₁) is analyzed. Its number of children is assessed. If P₁ only has 1 neighbour (P₀) then it signifies that P₁ is an end-point. Thus, the selected part (REF or ADD) is only composed of a single component defined by solely two key-points. Except eccentric reducers, all piping components defined in that way are SOR. The only stored key-point will then be the weld-point P₀. (If an eccentric reducer is instead detected, the points are shifted to the other weld neighbour to still evaluate the angle and conclude that the component is not a solid of revolution. Otherwise, no angle would ever be calculated and the program would conclude that because only 2 key-points were detected the part is a SOR.)

- As soon as P_1 possesses 2 children (3 neighbours), then the part is automatically considered as \overline{SOR} and once again only the weld point P_0 is stored in the key points list. This would typically be the case for tees or valves. Indeed, when welding a single tee to an assembly, P_1 has 3 neighbours: the weld point, the end-point of the tee, and the branch point.
- When P_1 is found to have 2 neighbours (1 children), two normalized vectors are created binding P_1 to its neighbours P_0 and P_2 . As written in Algorithm 1, the angle is evaluated using the inverse cosine function between these vectors. The detection of a non-zero angle reveals the presence of a \overline{SOR} component in the inspected group of components. Following Algorithm 1, this implies that the current key-point P_1 is not aligned with its previous and next neighbor ($\overrightarrow{P_0P_1}$ and $\overrightarrow{P_1P_2}$ don't form a straight line) The situation happens when an elbow or an eccentric reducer is in the assembly. For instance, if a 90-degree elbow is in the chain of components, when calculating the angle for its first key-point, an angle of 90 degrees will be found. As a result, the whole group of components would be \overline{SOR} . Detecting a non-zero angle stops the algorithm and returns only P_0 within the key points list for the investigated part (REF or ADD).
- If the angle is null, P_1 is stored into the key-points list and P_2 is investigated as described. The algorithm runs, and so on, until all key-points P_i are processed.
- For both Reference and Addition, the algorithm will be terminated as soon as a non-zero angle is detected. In the situation where every key-point has been stored in the list (when REF and ADD are SOR), the algorithm will reduce the list to its initial configuration with only the weld point P_0 .

For both REF and ADD, the algorithm runs and their respective final key-points list, depending on the SOR property, are generated. Once the SOR property is determined, the final components of the MRM can be extracted. The next subsection leads to the procurement of the definitive list of components defining the Minimum Required Model.

5.3 Derivation of the Minimum Required Model

The objective of MRM is to eliminate the geometrical ambiguities in the sub-assembly being fabricated – all the components belonging to REF and ADD for a specific stage of fabrication are included in the sub-assembly – in order to lessen its complexity by reducing its scan volume and the number of components. Consequently, instead of analyzing and scanning the entire assembly to perform a geometrical control, the inspection can be performed on the reduced assembly (MRM) only.

In the following subsections, the rules of component selection and the constitution of the Minimum Required Model are explained.

5.3.1 Analysis of the Structure for Reference and Addition

As explained in the previous section, a flow of vectors is constructed. Each vector is associated to a node and its direction is dictated by the component direction in space. After being normalized, the unit vectors can be used to determine the calculate angle value (Algorithm 1). To facilitate the understanding, an example is provided in Figure 60 in the form of a 2D-representation. The associated 3D model is shown in Figure 61 to get a clear idea of the presented example. REF is composed of a pipe and an elbow and makes the group of key-points (DCBA). ADD is made up of a pipe, an elbow, and a pipe, grouping the key-points (DEFGH). The weld-point (D) is the initial position for both chains (REF and ADD). From there, the unit vector representing the first pipe of REF is built (\vec{u}_1) between D and C. While going to the next point, the vector is built in order to be able to calculate the angle (\vec{v}_1 is built at C and the angle between \vec{u}_1 and \vec{v}_1 is evaluated). On this example, the angle is null, so the next point B is analyzed. The process is then repeated for ADD.

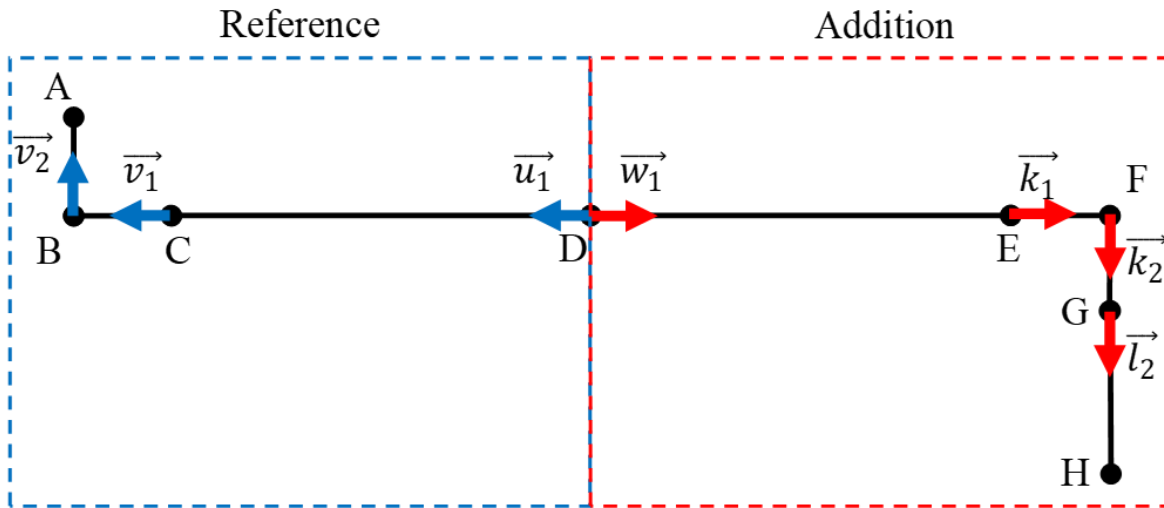


Figure 60: 2D-representation of a Pipe-Elbow-Pipe assembly (Addition) welded onto an Elbow-Pipe assembly (Reference). Reference is composed of a pipe (CD) and an Elbow (CBA) and Addition is made of a pipe (DE), an Elbow (EFG) and a pipe (GH). The unit vectors represent the normalized vectors connecting two consecutive points.

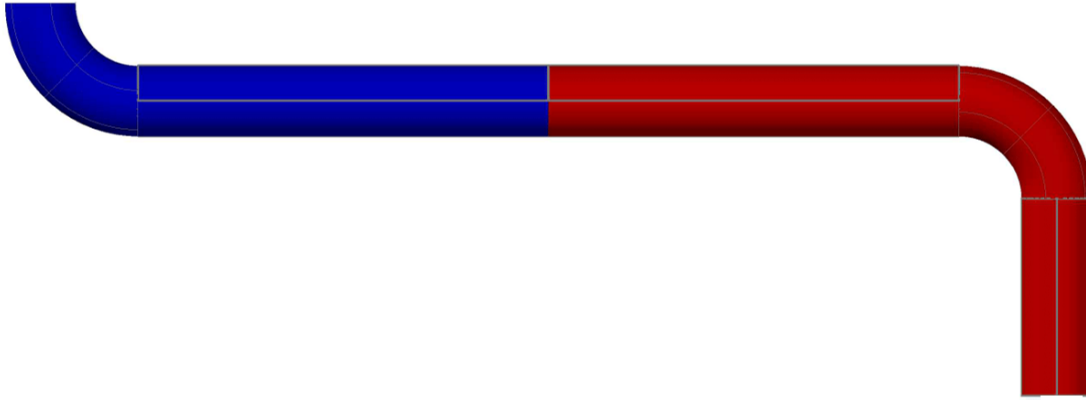


Figure 61: 3D model of the 2D-representation of the assembly presented in Figure 60. ADD is colored in red and REF is colored in blue.

It is crucial to understand that the geometric compliance of (DCBA) and (DEFGH) have been established in the hypothetical previous stages of fabrication. In other words, each sub-assembly is correct: first, the geometric compliance of (DCBA) was controlled when welding the blue pipe (DC) to the blue elbow (CBA); second, the compliance of (DEFG) was ascertained when welding the red pipe (DE) to the red elbow (EFG); finally, ((DEFG)H) was checked when welding the red pipe and elbow (DEFG) to the red pipe (GH). Consequently, the stage under investigation in Figure 60 is the assembly of ADD (DEFGH) with REF (DCBA). When applying Algorithm 1, the key-points list of REF and ADD are respectively composed of points D,C and B and points D, E and F (since a non-zero angle is respectively detected at point B and point F. As a consequence, the red pipe (GH) is removed from the model, because it doesn't necessitate controlling the geometrical correctness of the current stage. The elimination of some components in the MRM is further justified in the next section along with the example depicted in Figure 60.

5.3.2 Principle of Geometric Control

To assess the geometric alignment, every axis of ADD has to be controlled with respect to the ones of REF. The objective is then to constrain the vectors from ADD to the ones from REF. The representation of vectors is provided in Figure 60. To this extent, the vectors are compared together within each sub-assembly to only keep the ones needed. The process of isolating the unit vectors is done using the two rules described as follows:

$$\begin{aligned} & \text{If } \vec{a}_1 = \vec{b}_1 \text{ then only keep } \vec{b}_1 \\ & \text{If } \vec{a}_1 \neq \vec{a}_2 \text{ then keep } (\vec{a}_1 \text{ and } \vec{a}_2) \end{aligned}$$

where \vec{a}_1 and \vec{b}_1 denote two unit vectors connecting two consecutive key-points. The analysis of vectors and their selection enables the constitution of the sub-set of points used to form the MRM and control the correctness of the assembly. Applying the two rules described above to Figure 60 would result in keeping \vec{v}_1 , \vec{v}_2 , \vec{k}_1 , and \vec{k}_2 and removing all the remaining vectors.

To assess the alignment of ADD to REF, controlling the position of \vec{k}_1 with respect to \vec{v}_1 is required. By constraining them together, a revolute pair is created and removes five of the six degrees of freedom. In other words, after fixing \vec{k}_1 , ADD has only one more degree of freedom left around \vec{v}_1 . The representation of a revolute pair is provided in Figure 62. A revolute pair keeps the axes of two rigid bodies together. Two rigid bodies constrained by a revolute pair have an independent rotary motion around their common axis. Therefore, a revolute pair removes five degrees of freedom (DOF) in spatial mechanism (DOF = 1).

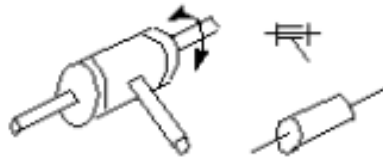


Figure 62: a revolute pair

However, ADD can still rotate around \vec{k}_1 . Therefore, an additional constraint must be set to remove the last DOF. For this purpose, the position of \vec{k}_2 is fixed based on \vec{v}_2 . ADD is thus completely constrained to REF. Finally, \vec{l}_2 does not have to be analyzed since ADD is fully constrained to REF.

The outcome on the kinematic torsor $\overrightarrow{V_{ADD/REF}}_H$ is presented and expressed as the base $(\vec{k}_1, \vec{k}_1 * \vec{z}, \vec{z})$ following the two steps described in the previous paragraph. The first constraint removes five of the six DOFs. Except the rotational velocity ω around \vec{k}_1 , it cancels five velocities around any point belonging to the rotational axis such as point F. When moving the velocity to point H, ω creates a linear velocity on the \vec{z} -axis. Consequently, a second constraint is defined to remove the rotational velocity ω and set each velocity of the kinematic torsor to zero. By this way, the end-point of ADD, point H has no motion left. More generally, any point on the second axis \vec{k}_2 could have been picked to define the kinematic tensor presented as follows.

$$\begin{aligned} \overrightarrow{V_{ADD/REF_H}} &= \left\{ \begin{array}{l} \overrightarrow{\Omega_{ADD/REF}} \\ \overrightarrow{V_H \in ADD/REF} \end{array} \right\}_{(\overrightarrow{k_1}, \overrightarrow{k_1 * \vec{z}}, \vec{z})} \xleftrightarrow{\text{Fixing } \overrightarrow{k_1 / \vec{v}_1}} \overrightarrow{V_{ADD/REF_H}} \\ &= \left\{ \begin{array}{l} \omega \ 0 \quad 0 \\ 0 \ 0 \ HF * \ \omega \end{array} \right\}_{(\overrightarrow{k_1}, \overrightarrow{k_1 * \vec{z}}, \vec{z})} \xleftrightarrow{\text{Fixing } \overrightarrow{k_2 / \vec{v}_2}} \overrightarrow{V_{ADD/REF_H}} = \left\{ \begin{array}{l} 0 \ 0 \ 0 \\ 0 \ 0 \ 0 \end{array} \right\}_{(\overrightarrow{k_1}, \overrightarrow{k_1 * \vec{z}}, \vec{z})} \end{aligned}$$

However, in the situation where all the unit vectors are equal – this means that all vectors are aligned and belong to the same axis –, for REF or ADD, then no constraint needs to be set up to remove the potential rotation around the first axis. Indeed:

- if REF has just one unit vector, there is no \vec{v}_2 to assess a potential \vec{k}_2 . This stands for situations where REF is SOR. For example, if REF is a pipe and a flange, then there is no \vec{v}_2 in REF to assess any potential second axis in ADD.
- respectively, if ADD has just one unit vector there is no \vec{k}_2 to be constrained to a potential \vec{v}_2 . This stands for situations where ADD is SOR. For example, with ADD being a single straight pipe, the composition of REF doesn't matter as there is no \vec{k}_2 .

The described situations stand for scenarios where either REF or ADD is SOR. The idea to bear in mind is that each point is analyzed until REF and ADD are fully constrained one over the other as best as possible. Once all 6 degrees of freedom have been removed, the analysis can be stopped, and the investigated points can be stored and utilized to derive the unambiguous model. In the previous example, after analyzing \vec{k}_2 with respect to \vec{v}_2 – that is, F with respect to B –, no more motion remains between REF and ADD. This is the conceptual reason why no other point needs to be investigated.

5.3.3 Constituting the Minimum Required Model

Once the key-points lists are finalized for REF and ADD, the constitution of the Minimum Required Model can be done. The outcome is to extract the components that are associated to the embedded key-points. Because of the parsing of PCF, each key-point is directly associated to, at least, one component. Therefore, a component is retrieved when one of its end-points is detected in a key-points list. Building on the two key-points lists for REF and ADD, a Minimum Required Reference (MRR) and a Minimum Required Addition (MRA) are created using the detected components.

For instance in Figure 60, point C is stored in the REF key-points list. Point C both belongs to the blue pipe and the blue elbow. It results that the two components are stored in MRR. On the contrary, none of the end-points from the red pipe (GH) was stored in the ADD key-points list because the analysis was stopped after point F. Therefore, the red pipe (GH) is not part of MRA.

The Minimum Required Model is obtained by adding MRR and MRA. Equation 4 below provides the mathematical expression of the MRM once REF_s and ADD_s have been derived.

$$MRM = MRR + MRA$$

The derived 3D model is only made up with the minimum required components to control the geometry and the alignment of the complete assembly.

Following the previous example from Figure 60 and Figure 61, the MRM is given below in Figure 63. The red pipe (GH) has been removed from the original 3D model because it is not required to assess the alignment and orientation of the assembly.

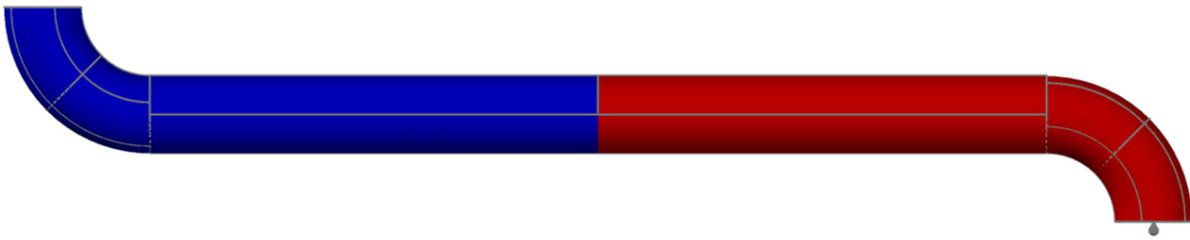


Figure 63: Minimum Required Model of 3D model in Figure 61

Finally, four different situations can be encountered and impact the constitution of the Minimum Required Model. These potential situations are based on the result of the Boolean variable SOR defined in Algorithm 1. Whether SOR is true or false for REF and for ADD leads to different approaches in the way of deriving the MRM. Table 9 summarizes the 4 situations which will be used to design the experimental 3D models.

Table 9: All possible cases with respect to ADD/REF being or not being a SOR. The 1-value stands for SOR and the 0-value for SOR.

Case	Reference	Addition	Outcome
1	1	1	MRM composed of the closest component to the weld of REF and ADD
2	0	1	
3	1	0	
4	0	0	MRM composed of components from the weld to the component associated to the non-zero angle for REF and ADD

As described above, the same approach is employed when at least one group of components identified to be SOR (SOR = 1).

Using the logic gate terminations, the overall function for deriving the MRM could be envisioned as an OR gate which behaves according to the truth table below (Table 10):

Table 10: OR gate truth table

Input		Output
A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

By describing the truth table, a high output (1) results if one or both the inputs to the gate are high. If neither input is high, a low output (0) results. By simply associating the low output to the outcome from case 4 (Table 9) and the high output to the other cases outcome, the derivation of the MRM can be conceived as an OR logic gate.

5.4 Conclusion

To conclude, the methodology for deriving the MRM has been entirely explained in this chapter. The genesis of the MRM was detailed and thorough explanations were provided throughout Chapters 4 and 5. The algorithm was described in the format of pseudo-code and examples rendered clear illustrations of the proposed concepts. In the two previous chapters, the complete methodology was studied, starting with an unviolated PCF and heading to a correct derived Minimum Required Model.

6 Effectiveness of the Proposed Methodology

To assess the previous methodology, the MRM impact was measured and estimated. Three potential outcomes of the MRM are surveyed: (1) the spatial complexity reduction, (2) the use of the 3D mobile laser scanner, (3) the obtained scan point cloud density. The evaluation of those three criteria was conducted using an experimental data set of 3D models comprised of: (1) a database of 95 models created by the author and split into the 4 potential situations that can be encountered (Table 9), and (2) a steel pipe spool assembly located in the UW laboratory.

6.1 Designing the Models

The following section expounds on how the models were designed and used.

6.1.1 Database of the 95 Models





















Case 1	 ID_C1_06	 ID_C1_07	 ID_C1_08	 ID_C1_09	 ID_C1_10
Case 2	 ID_C2_06	 ID_C2_07	 ID_C2_08	 ID_C2_09	 ID_C2_10
Case 3	 ID_C3_11	 ID_C3_12	 ID_C3_13	 ID_C3_14	 ID_C3_15
Case 4	 ID_C4_26	 ID_C4_27	 ID_C4_28	 ID_C4_29	 ID_C4_30

Figure 64: 3D models of five scenario examples per case: components in blue belong to REF and those in red to ADD.

Designing a significant amount of 3D models originates from the need to test the methodology out on the 4 potential situations described in Table 9. Once implemented into the SfM Code, the algorithm had to be validated and examined for each situation on realistic piping models. In addition, the 3D models in our possession were not sufficient enough for trying out an important amount of cases.

Consequently, the requirement of building up a database became obvious. To do so, *AutoCAD Plant 3D 2018* was used. Prior to generating the 3D models, they had to be designed cautiously to count enough models within each situation (Case 1, 2, 3, and 4).

For cases 1, 2, and 3, both REF and ADD, or one of both, need to be SOR. This constraint restrains the possibilities of diverse assemblies. Consequently, designing a substantially higher number of models for

case 4 is a voluntary decision and corresponds to an industry reality since most of the sub-assemblies include at least one $\overline{\text{SOR}}$ component (i.e. elbow, tee, valve, and reducer-eccentric). The odds are greater that REF/ADD will be $\overline{\text{SOR}}$ than the opposite. This is why case 4 includes more scenarios. For this reason, the number of scenarios designed for cases 1, 2, and 3 is significantly smaller than it is for case 4 (15 versus 50).

In case 4, both REF and ADD are SOR. Thus, no constraint is set to choose components; that is, every type of piping component can be part of REF/ADD as long as one $\overline{\text{SOR}}$ component is in the chain of components. The potential situations are therein countless. Even if two identical components are welded together, the other components located in the group of components matter as well as their position in the chain. For example, assembling a single elbow (REF) to another single elbow (ADD) is different than fitting a pipe and an elbow (REF) to a single elbow (ADD) and is also different than welding an elbow and a pipe (REF) to a single elbow (ADD). This simple case illustrates the impossibility to test out every existing scenario. Therefore, a larger number of scenarios was designed for this specific case. It includes scenarios with small amount of components as well as larger assemblies containing more components. The entire list of design models can be visualized in Appendix C.

Once all the scenarios were designed, *AutoCAD Plant 3D 2018* was used to create the 3D models. The software enables to create piping assemblies by using components belonging to piping catalogs. Thus, their shape and design are monitored and correspond to real piping components. The database was generated using the SS150 pressure class. An overview of some 3D models is provided Figure 64. Five examples of each case are provided. An ID number is assigned to each scenario to facilitate referring to them throughout this manuscript.

In designing these scenarios, the author endeavored to take into account the occurrences of SOR and $\overline{\text{SOR}}$ group of components in realistic pipe spool assemblies. The database was reviewed by industry's experts and was judged realistic and consistent. Its significance comes from its diversity that enables to prove, first, the correctness of the developed methodology, and second, its effectiveness.

6.1.2 Laboratory Steel Pipe Spool Assembly



Figure 65: 3D model of the laboratory steel pipe spool

The lab pipe assembly accounts for a real end-application of the proposed methodology as it was actually fabricated in a fab shop. It is made of 52 components distributed between 4 branches. All components are welded together except between two flanges that are bolted together. The assembly contains 22 pipes, 13 flanges, 12 elbows, 2 tees, 2 unions, and 1 valve. The diversity of the components makes the assembly a valuable case in point. The assembly was designed onto *AutoCAD Inventor* and its Piping Component File was realized afterward manually.

In order to apply the algorithm to the steel pipe spool, the fabrication stages were inferred based on the knowledge and experience of the author. Forty-six stages were required to assemble the entire assembly. Those stages are described in Appendix D. Among the stages, 13 belong to case 1, 9 to case 2, 13 to case 3, and 11 to case 4.

The results coming from this particular example are very representative of the potential utilization of the MRM along the entire fabrication process. The impacts are threefold:

1. Reduction of the spatial complexity of the model (Section 6.2)
2. Opening of new potentialities for consumer grade scanners (Section 6.3)

3. Improvement of the scan point cloud density (Section 6.4)

6.2 Impact of MRM on Reducing the Spatial Complexity and Number of Components

The effectiveness of the method is first evaluated in terms of the spatial complexity reduction and the decrease in the number of components. To this end, the concept of Level of Complexity (LOC) is introduced. LOC is a value that attempts to gather the spatial complexity and the number of components together. LOC is a driver of 3D scanning requirements. Each component is assigned a LOC detailed in Table 11.

Table 11: LOC value for each type of piping component

Type of component	Level of Complexity
Pipe	1
Flange	1
Elbow	2
Concentric reducer	1
Eccentric reducer	2
Tee	2
Valve	2
Coupling	1
Cap	1

For each component, key-points are established, and their associated unit vector is calculated. The number of unaligned vectors within a component provides the LOC for that component. Once each component has been assigned a value, the LOC of the full assembly can be calculated using Equation 5.

$$LOC_{assembly} = \sum LOC_{components}$$

Consequently, the LOC handles at the same time the spatial complexity of the assembly and the number of components. The LOC before and after applying the MRM is calculated for all of the designed scenarios and is tabulated in Figure 66.

6.2.1 Experiment on the Database Models

In the four cases, all the MRMs have a reduced LOC. The reduction of the Level of Complexity is important when a \overline{SOR} component is found as the first component of REF and/or ADD. The more components the initial assembly has, the more likely the MRM becomes useful. Thus, as shown in Table 12, the mean reduction of LOC for case 2,3 and 4 is higher than case 1. This is due to the fact that all the components in case 1 are SOR and, as explained earlier, a limited amount of assemblies have this quality. Case 1, 2 and 3, have at least REF or ADD being SOR. As a result, the MRM can be reduced to the only two components

being welded regardless of the other components. In case 4, the reduction can be very important if the first two components are $\overline{\text{SOR}}$, but can also be very small when the first $\overline{\text{SOR}}$ components are far from the weld.

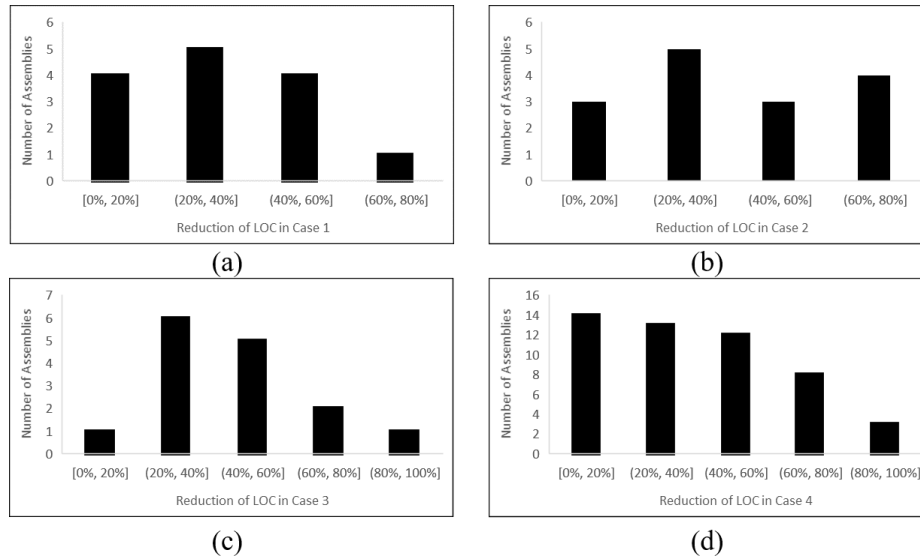


Figure 66: LOC Reduction between initial 3D models and after applying MRM for (a) Case 1, (b) Case 2, (c) Case 3 and (d) Case 4

Table 12: Average LOC Reduction for each Case

Case Number	Case 1	Case 2	Case 3	Case 4
Mean reduction of LOC	31%	40%	47%	41%

For example, in ID_C4_26, REF is composed of an elbow, a pipe and an elbow and ADD is an eccentric reducer, a pipe and an elbow. The weld is between the elbow and the reducer-eccentric. Thus, the MRM is only composed of the elbow and the reducer-eccentric. The result of LOCs for ID_C4_26 is detailed in Table 13.

Table 13: Result of Level of Complexity for ID_C4_26

$\text{LOC}_{\text{InitialModel}}$	LOC_{MRM}	Reduction of LOC
10	4	60%

The reduced LOC is important since it directly correlates with the time required for inspection of an assembly. Furthermore, the lower the LOC is, the more accurate the acquisition result will be.

6.2.2 Experiment on the Steel Pipe Spool

As for the steel pipe spool, each LOC is evaluated for each stage. Table 14 shows the percentage of LOC reduction for each case of the steel pipe spool assembly. The difference between case 1 and case 4 is

striking. This is due to the fact that case 4 stages usually happen after a certain number of components have been assembled, whereas case 1 stages are, most of the time, located at the beginning of the fabrication. For instance, welding a pipe to a flange is accomplished at the first stages. Thus, the initial LOC is already minimal (LOC = 2). On the contrary, welding two sub-assemblies together such as in stage 23 includes a high number of component (LOC = 11) which can be reduced with the MRM (LOC = 5) producing a high percentage of reduction (55%).

Table 14: Average LOC reduction for each case in the steel pipe spool assembly

Case Number	Case 1	Case 2	Case 3	Case 4
Mean reduction of LOC	13%	40%	26%	44%

Thanks to this assembly, it is possible to evaluate the overall assembly LOC reduction using the MRM. To do so, the total LOC of the initial model is computed by adding the initial LOC of each stage. The same process is applied to the MRM and the percentage reduction is calculated. The values are presented in Table 15.

Table 15: Total LOC comparison

Total initial LOC	Total MRM LOC	Reduction of LOC
261	150	43%

The percentage of 43% of LOC reduction is certainly the best way to estimate the effectiveness of the MRM on the reduction of spatial complexity. This way of calculating the average LOC reduction permits taking into account the higher importance of some stages rather than others. Instead of averaging out each reduction, this method renders the value each stage has as well as the total reduction. Moreover, at the end, what matters the most is the saved scanning time of the as-built assembly which is directly correlated to the overall LOC reduction. Therefore, it can be evaluated that, using the MRM methodology, 43% of the scanning time would be cut down compared to applying the same QC process by scanning all current components.

With the above example, the point is brought on the overall effectiveness of the MRM as it has value when implemented along the complete assembly fabrication.

6.3 Opening New Perspectives to Handheld Laser Scanners

As explained earlier, MRM is designed to be employed in a step-wise assembly control system for construction workers. To that end, measuring its performance with respect to limitations of consumer grade scanning devices is a matter of concern.

One of the major challenges of real-time mobile scanners are their limited scanning range and accuracy (Figure 67). While being cost effective and easy to use, the accuracy of this type of sensors decreases as objects are placed further away. For instance, the structured light sensor used on the SfM solution, called Structure Sensor, has a specific graph describing the relationship between the scanning distance and the depth precision (z-axis) (Figure 68). The range is only defined until 3.5m from the scanned object. At this distance, the depth precision is equal to 4cm. Regarding the precision on x and y-axis, it is linearly proportional to the observed depth and can be calculated as followed:

$$X/Y_{Precision} = \tan^{-1} \left(\frac{1 \text{ pixel}}{\text{Structure IO Focal Length}} \right) * \text{DistanceToObject}$$

The Structure IO focal length can be found in the device’s datasheet and is equal to 570. Using the 3.5m depth (DistanceToObject), the x/y precision is 6.1mm. Obviously, a strong correlation exists between the precision and the accuracy of the acquisition device. At least, to be able to obtain an accurate scan, the precision must be high enough.

Consequently, based on how depth and precision are bounded, it will be significantly beneficial to reduce the length of objects that need to be scanned. Figure 67 illustrates the example of the Structure Sensor with its limited range of 4x4x4m. It results in a maximum feasible length to obtain as-built information of 6.92m.

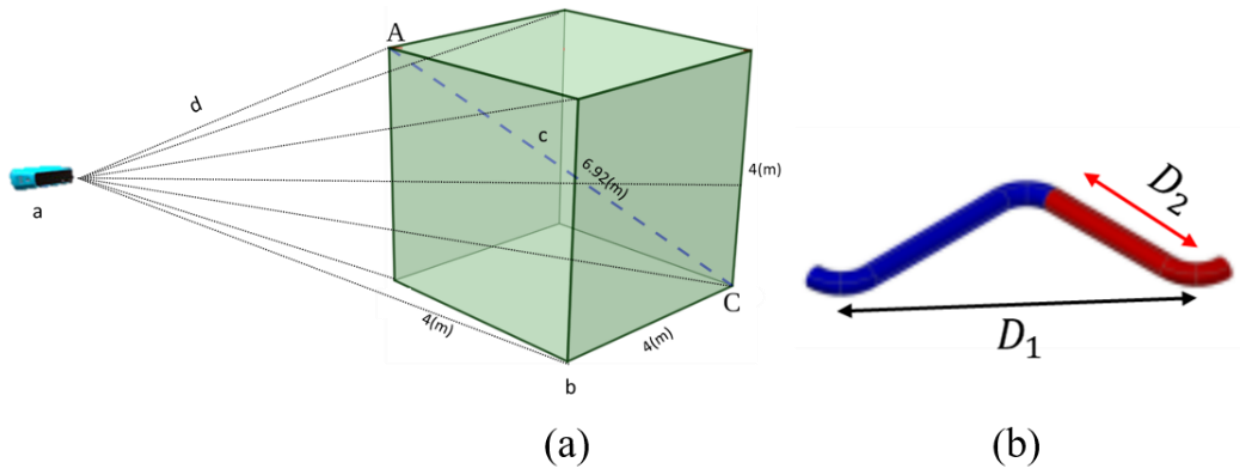


Figure 67: (a) Scanning range and maximum feasible length for scanning. a: acquisition device, b: only objects within the projected cube can be scanned, c: maximum allowable length for an object to be scanned, assemblies that have a principal length of more than 6.92 (m) cannot be scanned and controlled. d: infrared beams projected to capture the scene. (b) Principal length before and after applying MRM. D_1 is the principal length of the assembly before applying MRM. D_2 is the principal length of the assembly after applying MRM.

By defining principal length as the shortest line connecting the two ends of the assembly, a performance measure can then be defined (Equation 6) for MRM with respect to a components’ length.

$$U_{MRM} = \frac{(K_1 - K_2)}{K_1} \times 100$$

where K_1 denotes to the principal length of the assembly before applying MRM and K_2 denotes to the principal length of the component after applying MRM.

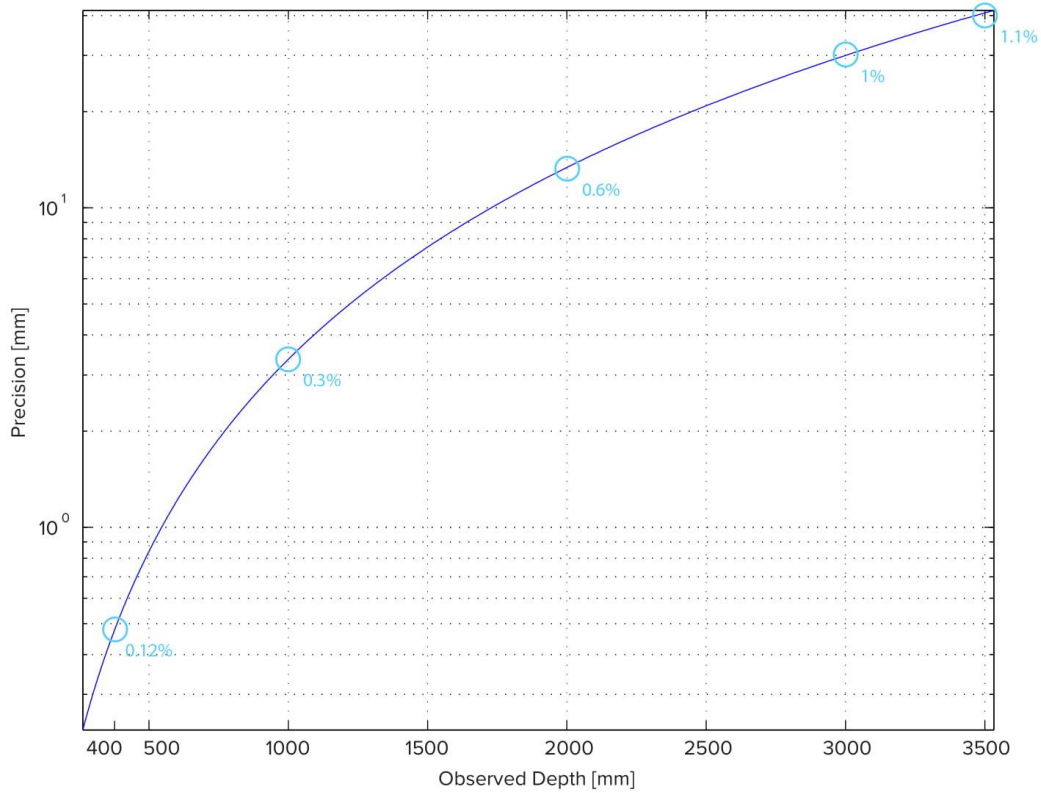


Figure 68: Structure Sensor's depth precision

6.3.1 Monte Carlo Simulation

To more accurately quantify the effectiveness of MRM on addressing the above challenges, Monte Carlo simulations were performed on the created 3D models database. It was assumed that the only contributor to assemblies' length are pipes. In order to simulate variability of pipes' length, an expert's advice was sought and the assemblies were divided into 3 groups based on their length (Table 16). Each type of length matches with specific applications and thus can be associated to a particular industry. Along with the variation of lengths, pipes' diameters were also modeled to account for the variability. However, as they don't directly influence the required scan volume, no special consideration is established in this experiment.

Table 16: Categorization of the assemblies based on average length of pipes in the assemblies and the intended application. The length of pipes in each category was assumed to follow a normal distribution.

Attributes:	Small	Medium	Large
-------------	-------	--------	-------

Mean Length (M)	1.5	3.5	5
Standard deviation	0.5	1	2
Application	Mostly residential and non-industrial	Nuclear and water supply	Oil and gas, large industrial projects, and pipe lines

Monte Carlo simulations were designed using a Normal distribution to estimate pipes' lengths. Independent simulations were run for pipe's length 1 (PL1), pipe's length 2 (PL2), pipe length's 3 (PL3) ... Each simulation was made of 1,000 independent events providing for each PL 1,000 lengths. Those results were then averaged out to obtain a mean length for each PL. Finally, PLs are added up to get total lengths.

For each database scenario, the number of pipes is computed for the initial model and the MRM. Based on the geometry of the initial model, the principal assembly length (PAL) is computed. For instance, if two pipes are aligned, PAL is obtained by simply adding up PL1 and PL2. But if they are perpendicular, PAL is computed by assuming PL1 equal to PL2 – where PL1 is the PL of the pipe left in the MRM – and PAL is equal to the square root of 2 multiplied by PL1.

A new Monte Carlo simulation is run for each of the 95 scenarios within each size category. Therefore, a total of 95 Monte Carlo simulations are done for each pipe's application, achieving a total of 285 independent simulations (95 different simulations for each of the 3 applications).

To calculate U_{MRM} for each assembly, the principal lengths were then calculated and recorded for both initial model and MRM for each scenario within each size category. The comparison of MRM principal length and initial model one is plotted in Figure 69. The utility functions are also plotted on the graph to render a visual estimation of the MRM performance.

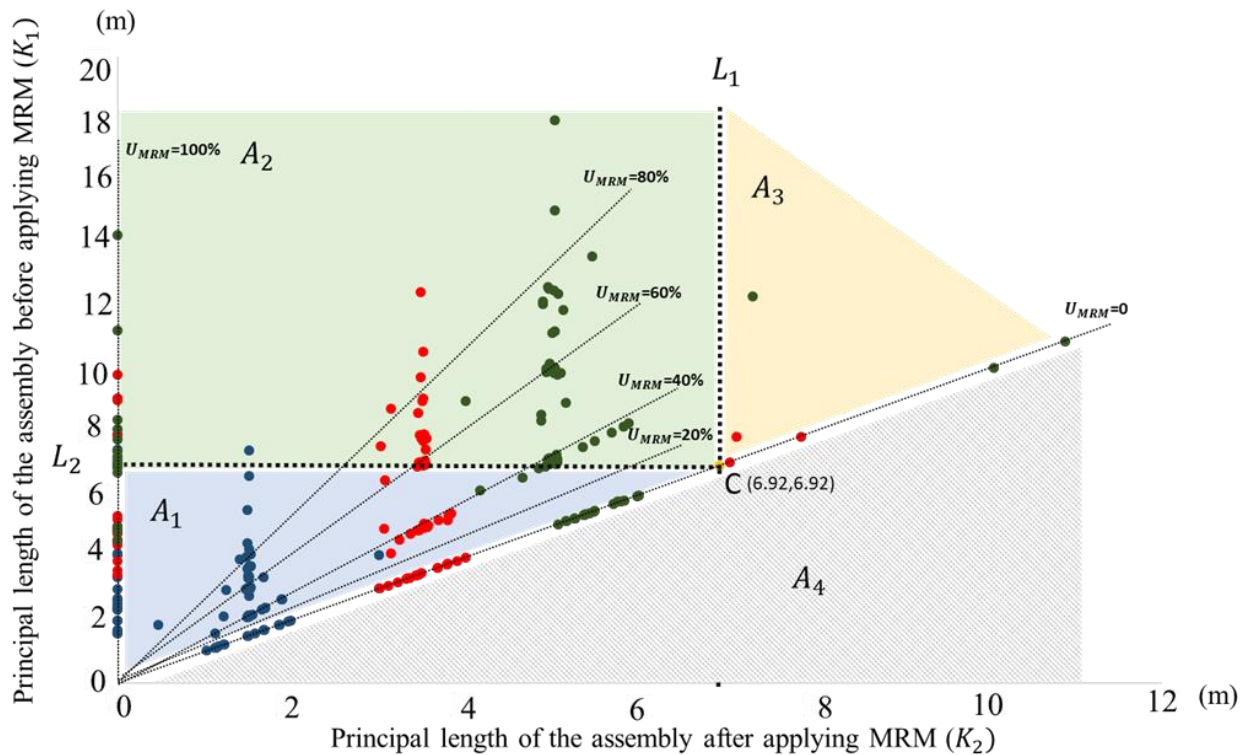


Figure 69: Monte Carlo simulation results using the assemblies from the database with three size categories

Each point in Figure 69 corresponds to an assembly existing in the database and has 2 values associated with it, K_1 and K_2 . Points in blue color show the K_1 and K_2 values for scenarios simulated for small assemblies, red points correspond to medium-size assemblies, and finally, green points belong to assemblies made up with large pipes. As shown, the line $U_{MRM}=0$, divides the space into 2 areas: A_4 and A_1 , A_2 and A_3 . As explained earlier, in the worst-case scenario, applying MRM will not reduce the LOC, and thus the principal components' length will not change ($K_1=K_2$). As anticipated, no point exists in the A_4 area. The remainder of space is divided into 3 areas by drawing the lines L_1 ($K_2=6.92$) and L_2 ($K_1=6.92$).

- Points in the A_3 area correspond to scenarios where each principal length of both the initial assembly and the MRM exceeds the scanner's range. Thus, obtaining the as-built information is infeasible. These assemblies are too large to be scanned and employing MRM will have no utility.
- Points in the A_2 area correspond to assemblies that can only be scanned if MRM is applied. In other words, this area shows assemblies whose as-built information could not be obtained without the use of MRM.

- Finally, points in the A_1 area can be scanned whether MRM is applied or not. However, applying MRM on some of the assemblies can reduce the principal length in the assembly and thus increase the accuracy of the obtained information.

The diagonal line drawings in this area indicate how much the principal length was reduced. For example, if a point lies between lines $U_{MRM}=40\%$ and $U_{MRM}=60\%$, it has had a reduction of more than 40% and less than 60% in the assembly’s principal length which directly correlates with the accuracy of the obtained as-built information. Table 17 shows the distribution of points within A_1 , A_2 and A_3 .

Table 17: Distribution of points in the simulation and MRM utility in each group.

Area	Ratio	MRM’s Utility
A_1	61.0%	Increased acquisition accuracy
A_2	36.9%	Only feasible to inspect with MRM
A_3	2.1%	No Utility

Furthermore, in order to investigate the application and the performance of MRM within the three categories, Figure 70 was prepared.

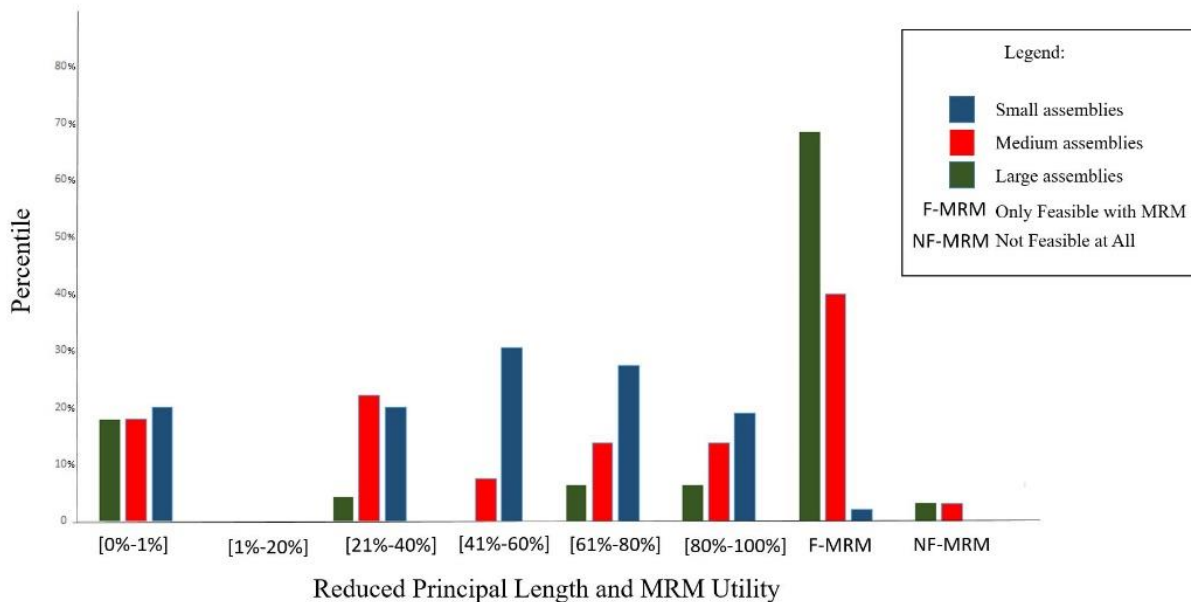


Figure 70: Comparing MRM utility within the three size groups

As can be seen, regardless of the assembly’s size, MRM will not be useful for the assemblies that have a specific geometry ($U_{MRM}=0\%$). For instance, ID_C1_07 initial model is composed of a flange and a pipe

welded to a concentric reducer. Applying MRM only removes the flange, which doesn't impact the principal length. Besides, the results show that applying MRM has the most impact on accuracy for assemblies with small dimensions. Finally, more than 70% of the assemblies are infeasible to obtain as-built information in large assemblies. This shows that applying MRM is critical for any step-wise assembly process control for large assemblies.

6.3.2 Bounding Box Diagonal Length Analysis

Using the same 95 scenarios, another analysis was run using the real dimension of the 3D models determined by inputting them into *CloudCompare*. Figure 71 below gives the representation of the *CloudCompare* bounding box that is displayed around a point cloud.

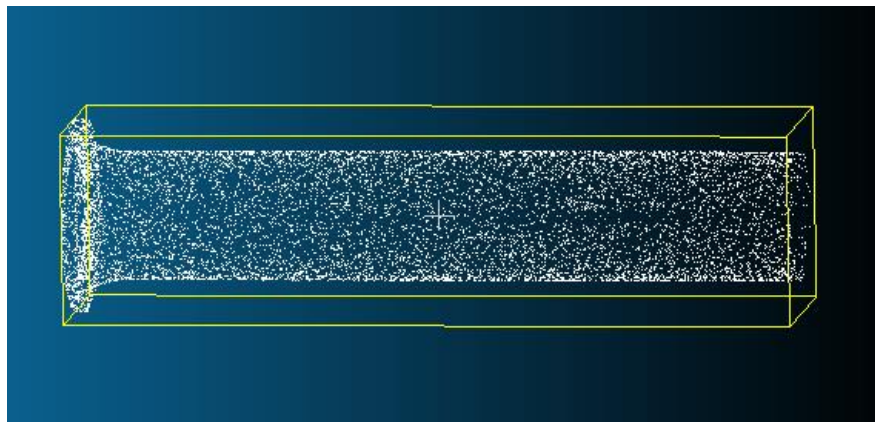


Figure 71: Point cloud in CloudCompare surrounded by its bounding box

The maximum dimension of the box is extracted for each of the 95 initial models and 95 MRMs. Knowing that the scanner's bounding box is a cube, the required cube dimension is estimated by calculating the required diagonal length of the bounding box. To this end, the maximum dimension is multiplied by $\sqrt{3}$. The reduction of the diagonal length is then computed for each scenario. One example (ID_C1_08) is depicted in Table 18.

Table 18: Example of reduction of diagonal length for scenario ID_C1_08

	<i>Result in meters (m)</i>	Bounding box maximum dimension	Bounding box diagonal length	Reduction of diagonal length
ID_C1_08	Initial model	2.41	4.20	68%
	MRM	0.76	1.32	

Once the reduction is computed for all scenarios, the similar chart presented earlier can be drawn. The 95 points are plotted in Figure 72. As explained earlier, the chart is segmented into A_1 , A_2 , A_3 , and A_4 . The distribution of the data is displayed directly on the chart.

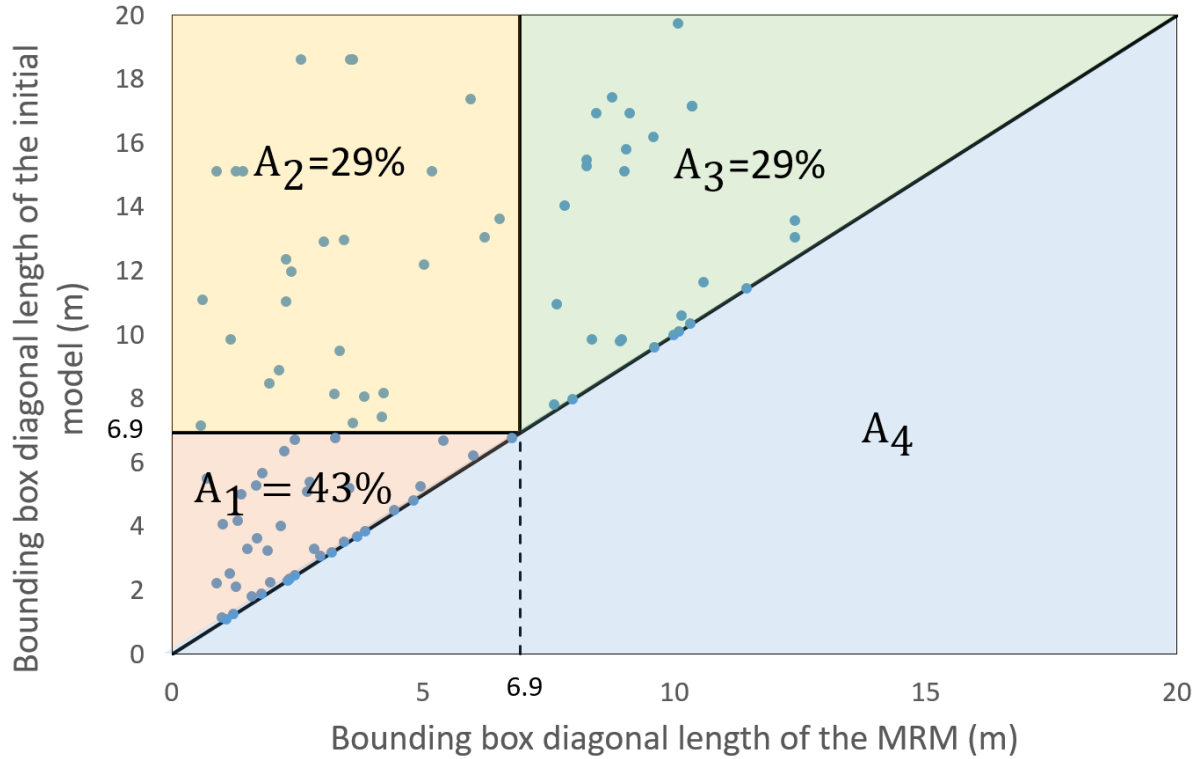


Figure 72: Analysis of the bounding box diagonal length for the database scenarios

The design of the scenarios was performed independently of the proposed analysis. Thus, no trick was applied to reach this result. How data is distributed among regions can therein be considered realistic. Besides, in addition to belonging to the same piping pressure class, all components used in the models were designed to remain within realistic dimensional lengths with regard to their diameter.

The above representation differs from the one obtained with Monte Carlo simulations because it compares the diagonal of the required bounding box whereas the first one focuses on the principal length of the obtained assembly and assumes the principal length of the assembly matches with the bounding box diagonal. In practice, aligning the principal length with the scanner's bounding box is not quite manageable. However, both approaches are valuable and provide a significant performance measurement of the MRM methodology.

6.4 Improving the Scan-vs-BIM Comparison

Last but not least, the point cloud density is measured, whether or not the MRM is applied. Use of MRM shrinks down the initial model. The potential impact of this reduction is a considerable improvement of the point cloud density enabling a more accurate deviation analysis (comparison of the scan point cloud to the model point cloud). Indeed, since the model is smaller, the required scan is smaller. Thus, the scanner's bounding box can be significantly diminished. Because the size of the scan affects the point cloud density

, and thus the precision of the deviation analysis, MRM is found to be very effective for improving the Scan-vs-BIM comparison.

6.4.1 Dependency between the Number of Vertices and the Bounding Box Size

So as to prove the influence of the scanner's bounding box size on the obtained point cloud density, an experiment was carried out. Four scans were acquired from the same position pointing at the same object: a monitor (Figure 73). All 4 scans were taken facing the monitor without moving around and only the bounding box was changed.



Figure 73: Monitor used as the center object of the four scans

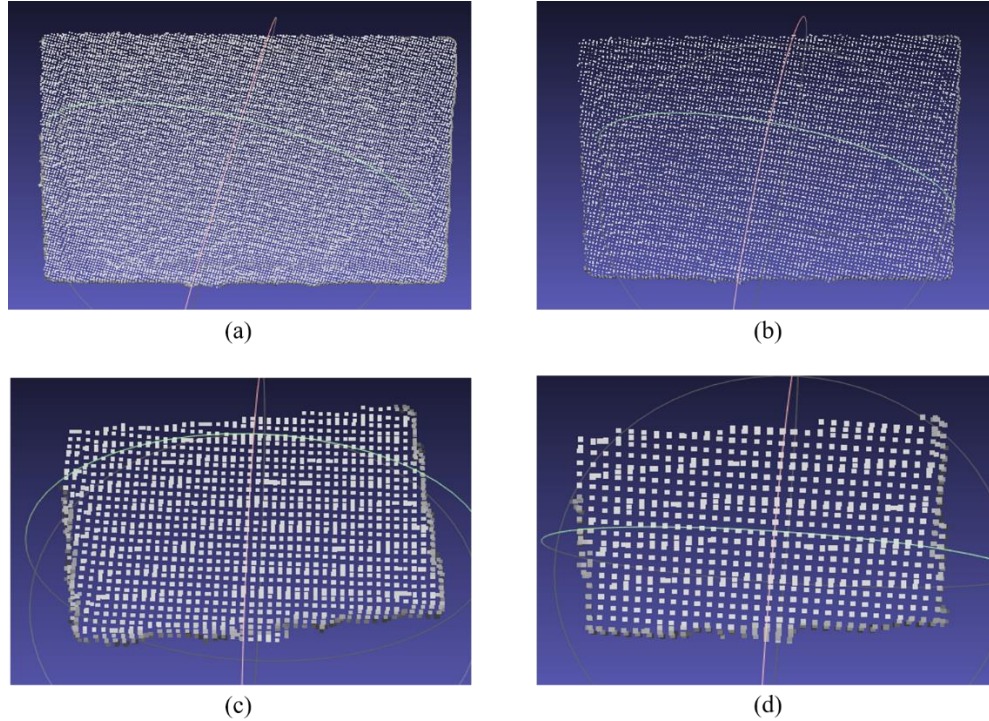


Figure 74: Scans of the monitor without the clutter with different bounding box size: (a) scan 1, (b) scan 2, (c) scan 3, (d) scan 4.

The scans were then post-processed to remove the clutter (all the points which do not belong to the monitor). The four scans are presented in Figure 74. The raw scans were imported into *CloudCompare* to estimate the scan's bounding box size. The given dimension is assumed to match with the scanner's bounding box size. Then, the number of vertices on the monitor was computed by removing all the clutter from the initial scans. By this way, a relationship between the two parameters can be established. Table 19 below summarizes the scans properties.

Table 19: Scans properties

	<i>Smaller</i> → <i>Larger</i>			
	Scan 1	Scan 2	Scan 3	Scan 4
Bounding box maximum dimension (m)	0.5984	0.8724	1.9662	2.7459
Number of vertices	21829	10277	1830	944

The above data clearly underlines the falling-off in the point cloud density— represented by the number of vertices – when the bounding box is enlarged. To illustrate this phenomena, a graph was drawn by using the above data and is presented in Figure 75.

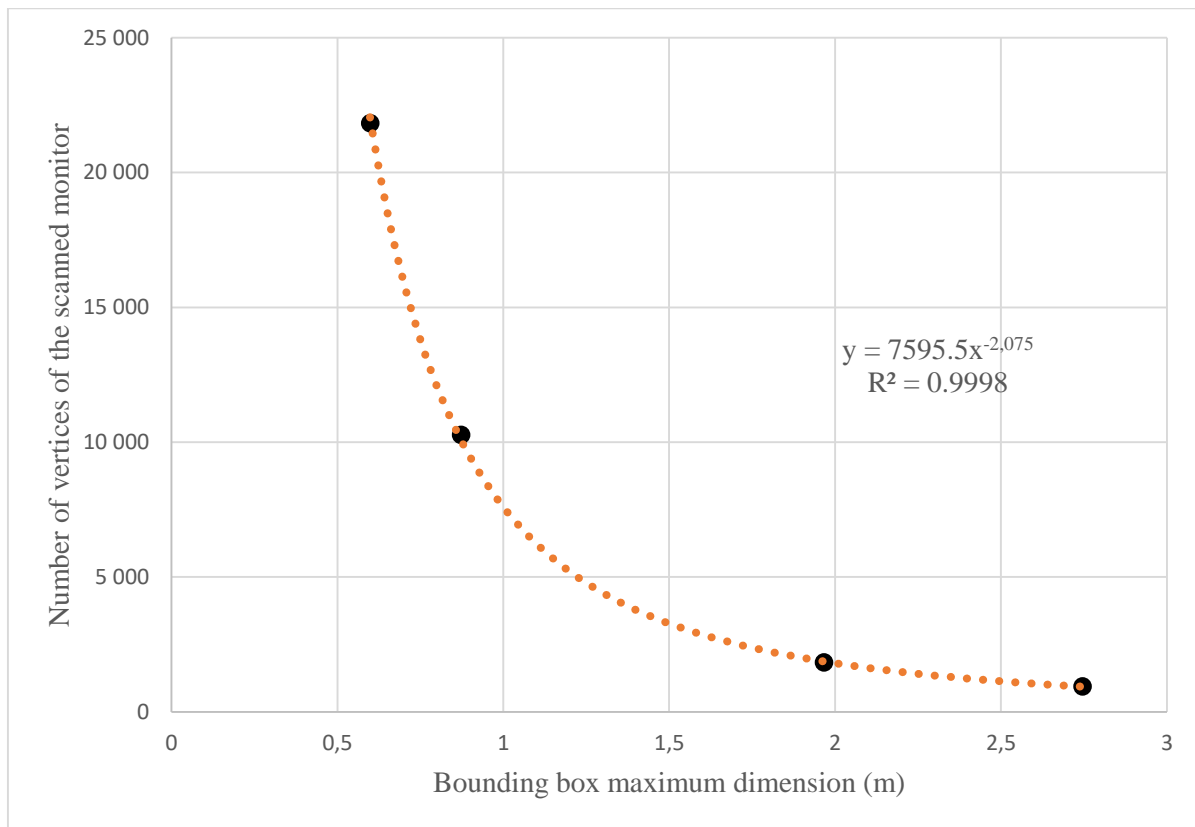


Figure 75: Graph representing the number of vertices of a scanned monitor as a function of the scanner's bounding box size. The blue line is the curve binding the 4 data points. The dotted orange line stands for the trend line. Its equation and R^2 value are also displayed.

The data points were linked together in order to visualize the shape of the relationship. A trend line was then associated to the curve. The best matching function is a power function proportional to $\frac{1}{x^2}$. By modeling the data with the rendered function, the R²-value of 0.9998 appears to be high enough to state the existence of a correlation between the two factors.

However, the equation is most probably size-dependent, meaning that the number of points obtained after removing the clutter relies on the size of the scanned object. The scanned monitor's dimensions are 45*30cm creating a surface of 1350cm². Consequently, a relationship may exist between the scanned surface area, the bounding box size, and the point cloud density and may be further investigated to excerpt a proper connection from the Structure IO.

6.4.2 Comparison of Scanning from a Large vs. Small Volume with Structure IO

Since a correlation between the bounding box size and the scan point cloud density has been established, the concept can henceforth be proven onto pipe spool assemblies.

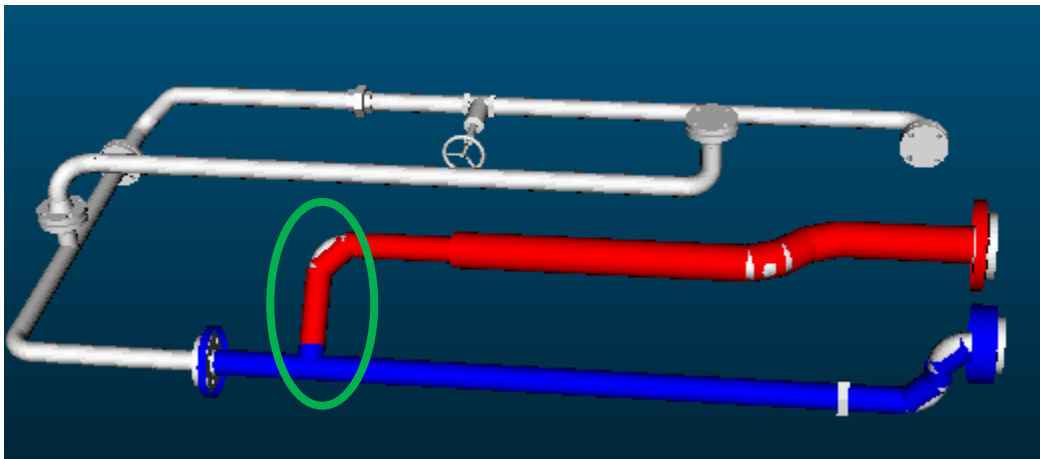


Figure 76: Assembling the red part on the blue part. The weld is located at the branch of the tee. The green oval surrounds the Minimum Required Model.

The objective of this section is twofold:

1. Provide a real application of the Minimum Required Model on the model point cloud generation.
2. Prove the importance of reducing the scan volume with consumer grade scanners such as Structure IO.

To this end, the steel pipe spool assembly is chosen as an experimental assembly. In particular, focus is made on fabrication stage 21 displayed on Figure 76. Following the color rule, red components (ADD) are assembled to blue components (REF). The weld is performed at the branch of the tee and the MRM is

surrounded by the green oval. Using the SfM software as shown in Figure 77, the MRM components can be segmented.

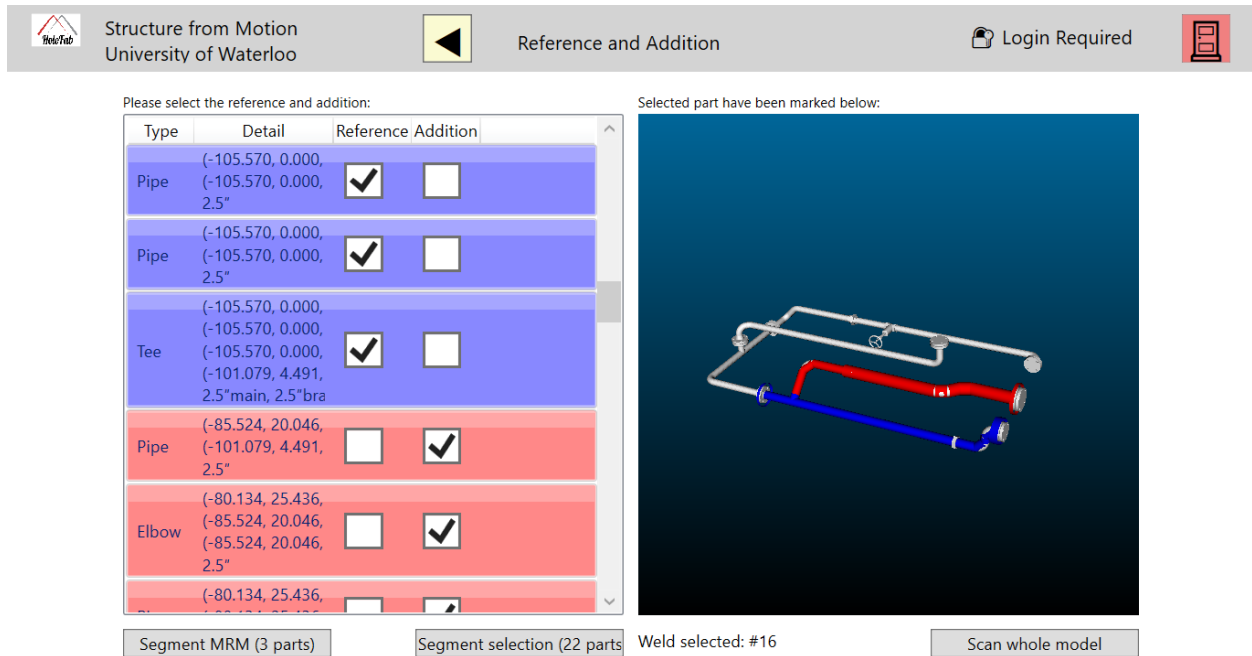


Figure 77: Segmenting the MRM on Stage 21 of the steel pipe spool assembly in the SfM software

The result is presented in . It is composed of the tee (in blue) from REF, and the pipe and the elbow (in red) from ADD. The point cloud is defined by 345 vertices. The vertices are randomly distributed among the meshes of the original model based on the mesh surface. Owing to algorithmic functions, the intrinsic density could be increased to obtain a denser model point cloud. Although the model point cloud number of vertices can account for a minimum barrier to reach, the following study mainly focuses on the scan point cloud number of vertices.

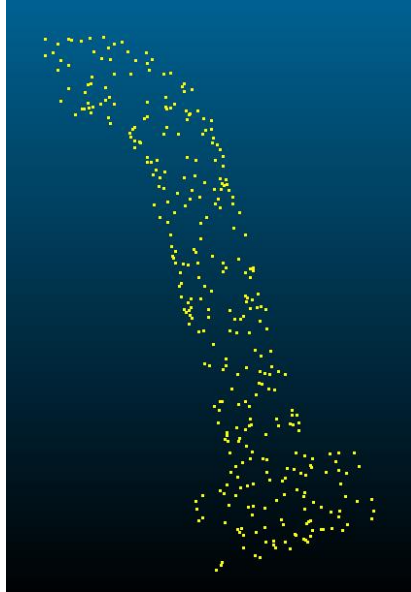


Figure 78: Model point cloud derived from the Minimum Required Model

The following experiment is using two different scan point clouds. The first scan (Figure 79) has been acquired with the Structure IO with a small cube of 1x1x1m (denoted “small” scan). The second scan (Figure 80) was acquired with the maximum scan volume of 4x4x4m (denoted “large” scan). Once the two acquisitions were completed, the clutter – all points not belonging to the MRM – was manually removed. Consequently, this process renders two different point clouds representing the same components.

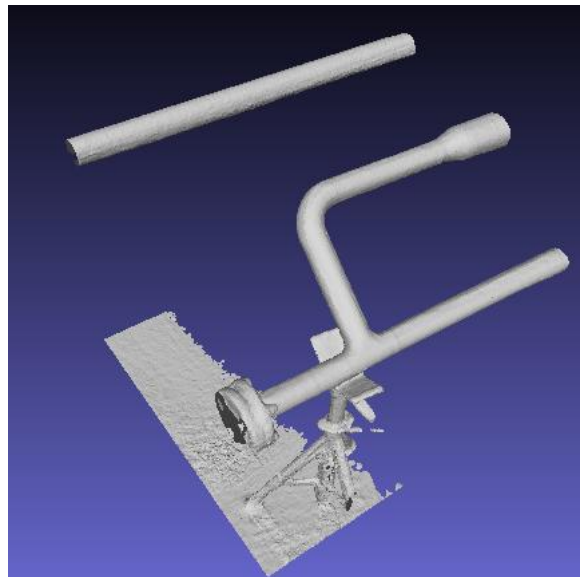


Figure 79: "Small" scan acquired with a cube volume of 1m³

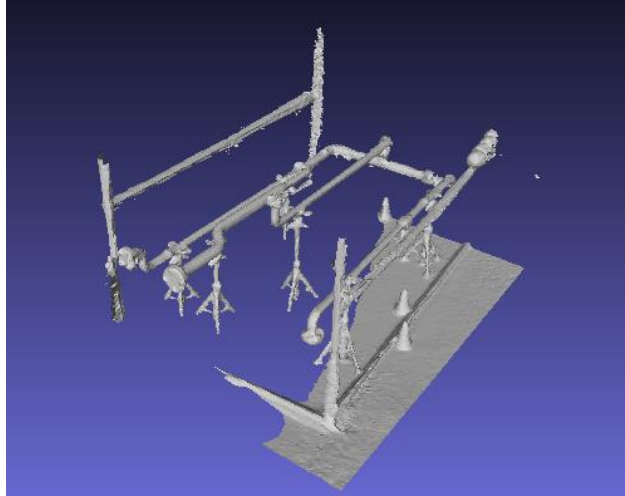


Figure 80: "Large" scan acquired with a cube volume of 4x4x4m

The two scans can be visualized in Figure 81 and are intentionally shown in parallel so that one can realize the density contrast.

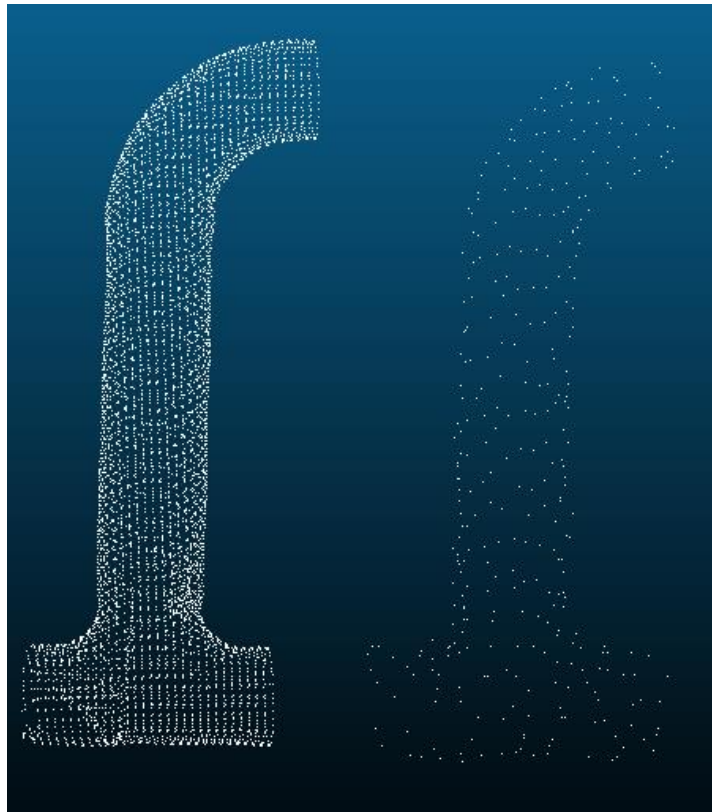


Figure 81: Parallel representation of the two scans obtained after removing the clutter

Each scan point cloud is used to perform a deviation analysis. The result is analyzed with the aim of proving the advantage of scanning a smaller assembly thanks to the MRM algorithm. In addition, the two scan point clouds are compared together to reveal some potential causes of the differences.

6.4.2.1 Comparison using the “large” scan

The first deviation analysis is performed on the “large” scan data. For the purpose of scanning this volume, the bounding box was extended to its limit (4x4x4m). After preprocessing the raw scan, the minimum required scan was loaded into the SfM software to be compared to the model point cloud of Figure 78. The scan point cloud is displayed with red points, on the right side of Figure 82, and is composed of 625 vertices.

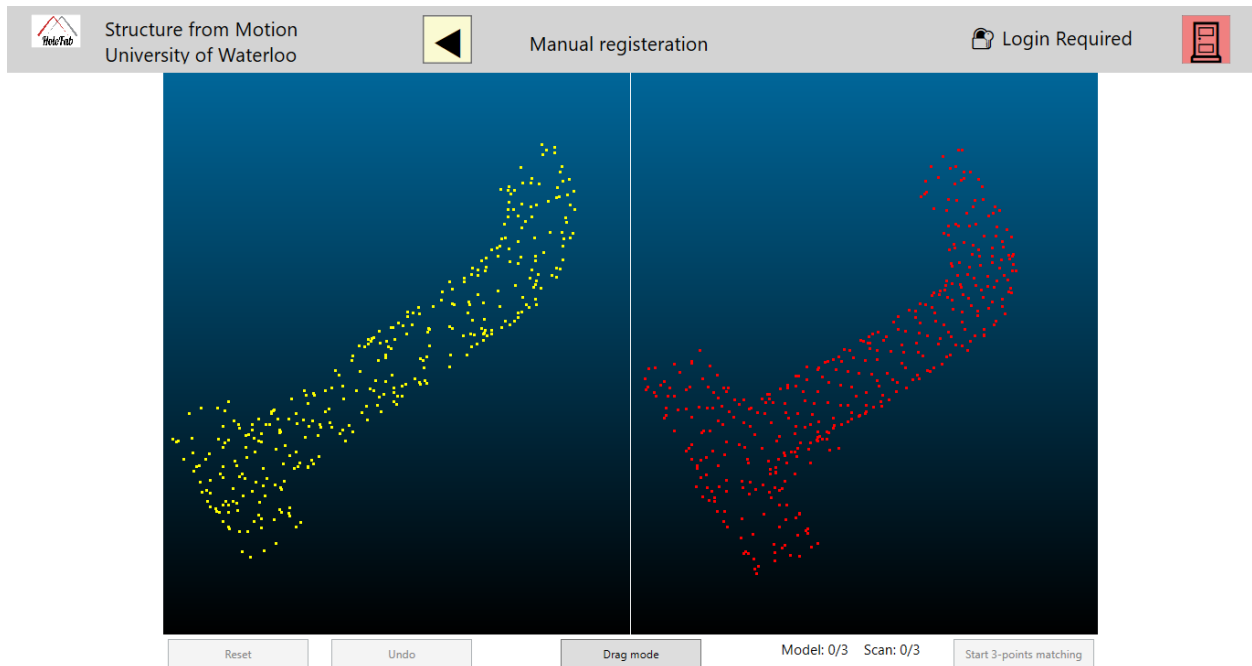


Figure 82: SfM 3 points registration page. On the left, the model point cloud. On the right, the scan point cloud from the “big” scan

The two point clouds are superimposed by matching 3 points. Finally, the clutter is removed to keep the 10 closest neighbours to each model point. Every point not detected in the neighborhood of a model point is removed from the scan point cloud. Once completed, the result of the deviation analysis is given by the software and shown in Figure 83. The maximum deviation detected on the heat map is 0.981mm.

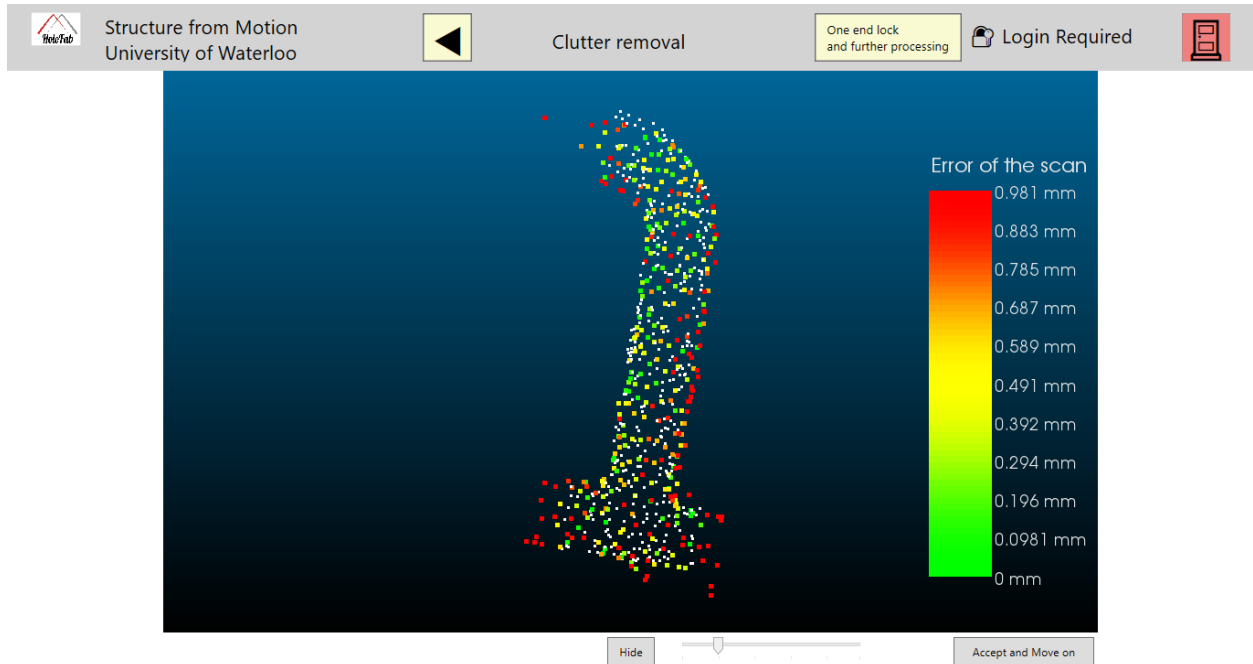


Figure 83: Comparison of model point cloud to scan point cloud from the "large" scan

6.4.2.2 Comparison using the "small" scan

The same process is performed with the "small" scan.

Using the "small" scan, the scan point cloud is made up of 6,632 vertices and is shown on the right side of Figure 84. As noted, the point cloud is about 10 times denser than the one from the "large" scan.

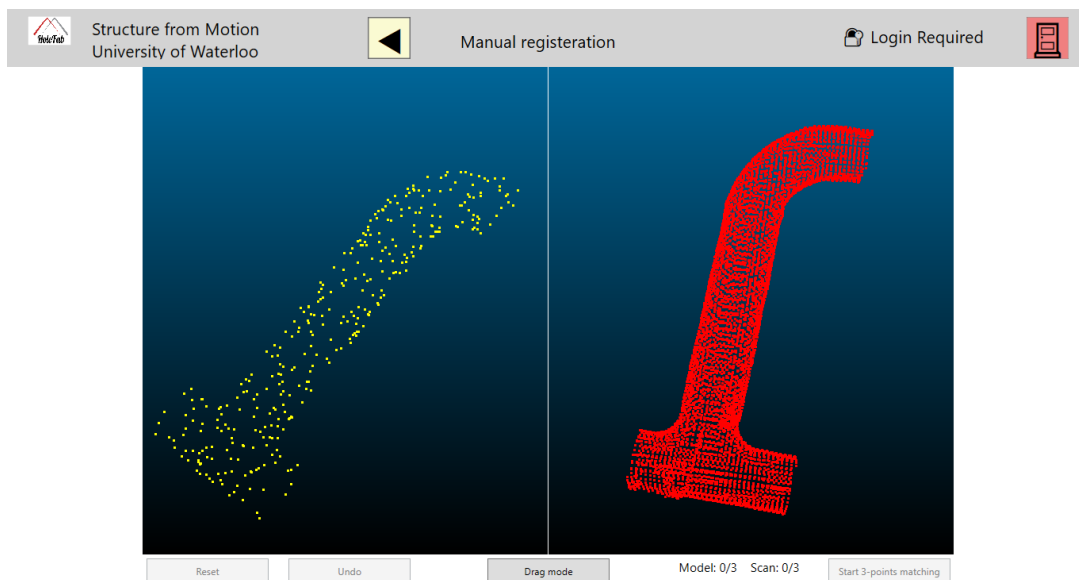


Figure 84: SfM 3 points registration page. On the left, the model point cloud. On the right, the scan point cloud from the "small" scan.

The result of the comparison, once clutter removal has been applied, is presented in Figure 85.

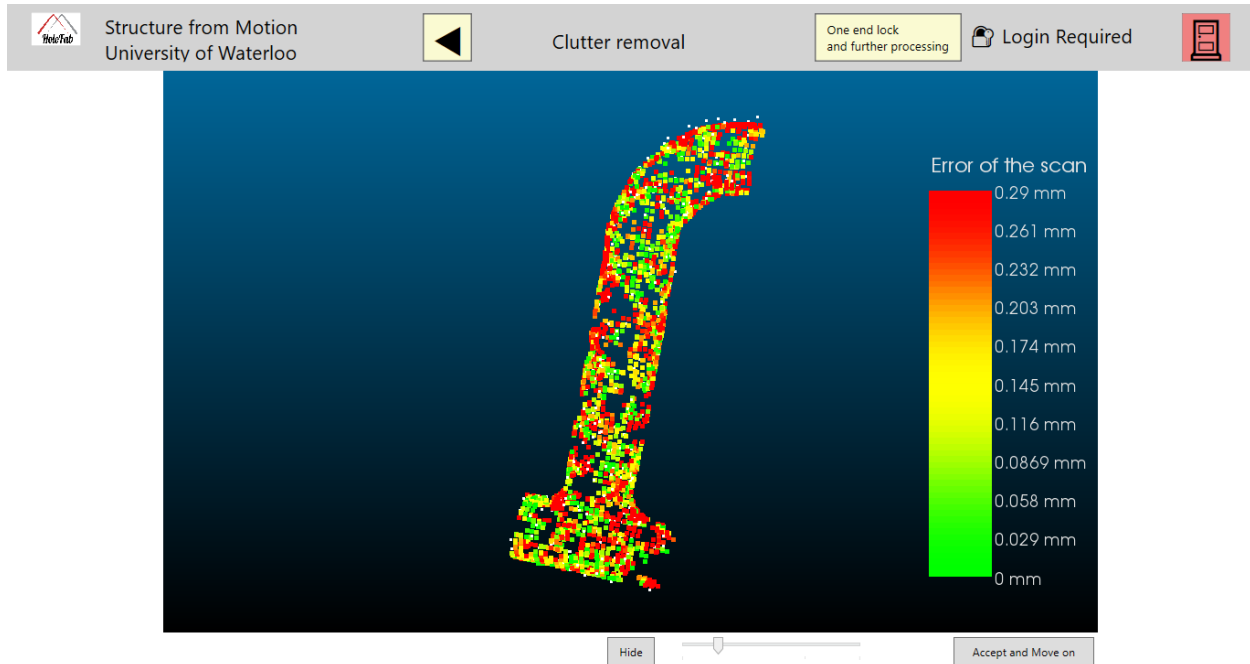


Figure 85: Comparison of model point cloud to scan point cloud from the "small" scan

This time, the maximum deviation displayed by the heat map is 0.29mm, mostly located on the components surfaces. The explanation can be found in the density difference between the two point clouds. In other words, since the analysis is run on the scan point cloud, denser than the model, some of its points can be far from a point belonging to the model point cloud (up to 0.29mm). However, as the number of closest neighbours used in the clutter removal has been taken at 10, one can infer that each point of the model point cloud has its 10 closest neighbours within a sphere of 0.29mm radius. Because the scan point cloud is very dense, this value is very acceptable.

Comparing this result to the previous one highlights the sensitivity of the result to the point cloud density and thus to the initial scan volume. The fact is that some outliers cannot be removed with the clutter removal when the density is too low. For instance, with a 10th closest neighbour being far from a model point, increasing the maximum value of the heat map will automatically be surged.

Table 20: Comparison of the measured maximum deviation

Maximum deviation with the "big" scan	Maximum deviation with the "small" scan	Improvement of the deviation analysis
0.981	0.29	70%

To summarize the result, the maximum deviations are compared and its improvement is calculated (Table 20). A 70% reduction of the measured maximum deviation is reached simply by reducing the scan volume from 64m^3 to 1^3 which can be critical when controlling tight tolerances assemblies.

6.4.2.3 Comparison of the two scan parts

To further investigate this result, a comparison has been realized on *CloudCompare* between the two scans and is provided in Figure 86.

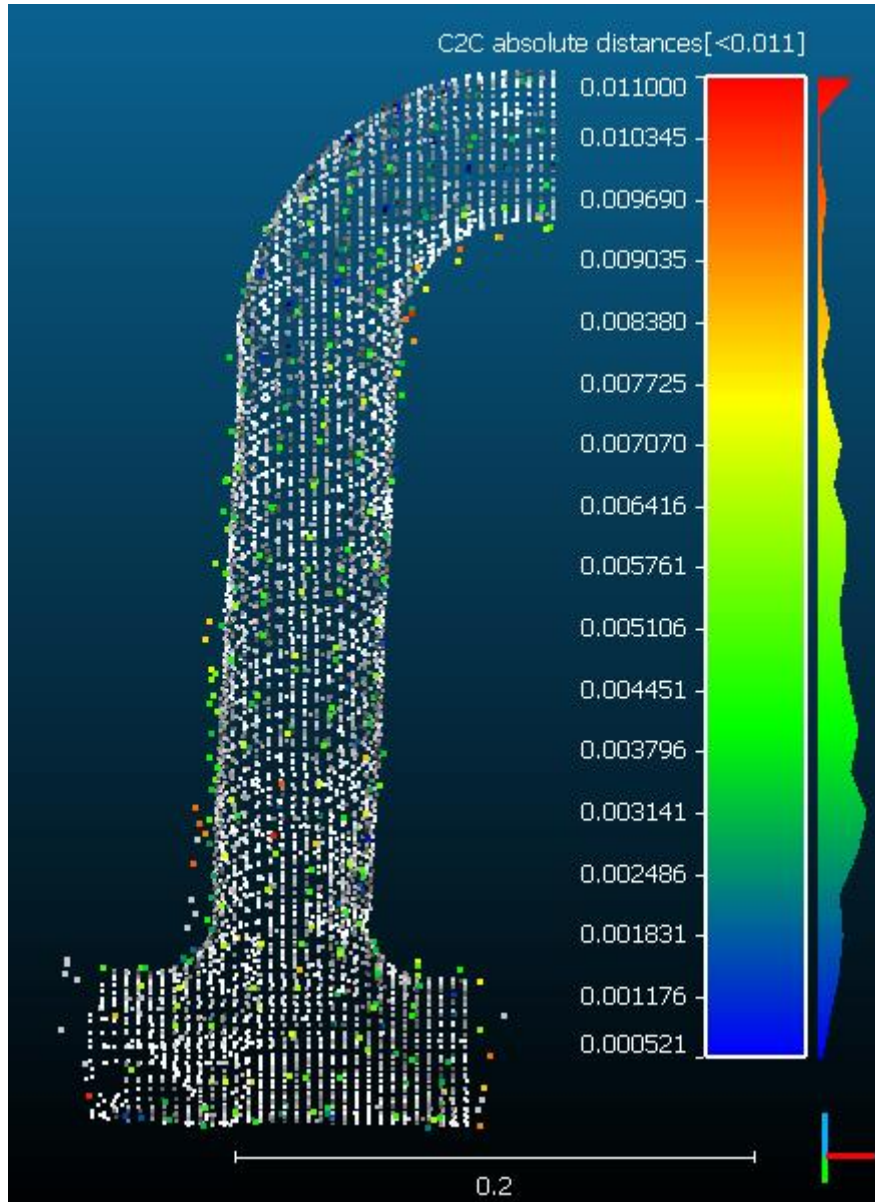


Figure 86: Comparison of the two point cloud. The scan from the "big" scan (colored points) is evaluated with respect to the scan from the "small" scan (white points).

The maximum deviation is found to be 11mm and is located in the elbow's inner curvature. Indeed, this area seems to have a deviation when comparing the two scans. Almost all points from the "large" scan have been found to have an approximate deviation of 3-4mm which can substantially influence the interpretation of a Scan-vs-BIM analysis. In general, the points from the "large" scan appear to be located above those from the "small" scan. One explanation may be the decline of the precision when increasing the depth (Figure 68) going from 0.3% precision for an observed depth of 1m up to 1.1% for a 3.5m observed depth. However, other parameters may influence the difference between those two scan point clouds which could be extensively investigated. This also depends on the scanner acquisition technology and its inherent parameters.

As a conclusion, the important outcome of this section is to show the density impact of the scan point cloud on the deviation analysis. The denser the scan is the more accurate the analysis is. As the density of consumer grade scanners such as the Structure IO can be directly associated with the scan volume, the finding is that utilizing the Minimum Required Model in deviation analysis can significantly enhance the accuracy of the analysis by cutting the structural deviation, thereby bringing more confidence in the outcomes.

7 Thesis Summary, Conclusions and Future Work

7.1 Summary

Due to higher requirements of accuracy and economical consequences of any defect, controlling the quality of industrial assemblies in fabrication shops has become crucial. Manual tools are usually utilized to measure dimensions and check alignments. To reduce the number of tasks performed manually, 3D vision technologies have become common practice.

Yet, these methods are often applied at the end of the fabrication process, because they can be cumbersome, time consuming and require engineering knowledge. In fact, when laser technologies are used to inspect the quality of industrial construction assemblies, the work is carried out by specialists who need to process the scanned data for a certain period of time, and who usually draft a report aimed at managers. The consequences are that those techniques only benefit the company late in the assembly process thus increasing the impact of fitting and rework risk.

As a result, this thesis develops a straight-forward methodology that can easily be implemented in a worker-designed device to streamline the quality control process. By using the Minimum Required Model on a real-time mobile scanner, workers can perform the inspection every time components are assembled. The proposed method is necessary for any 3D visual stepwise quality control process.

Along this thesis, the overall methodology is envisioned into a general process and fabrication workflows that utilize consumer grade scanners (e.g. Structure IO) to inspect pipe spool assemblies within a manufacturing environment. The methodology gives a significant role to Piping Component Files. Building on the piping inputs, the derivation of the Minimum Required Model is thoroughly detailed in Chapter 5.

The effectiveness of the Minimum Required Model in reducing the spatial complexity of the original assembly, in facilitating the utilization of consumer grade scanning devices, and in improving the scan point cloud density was shown Chapter 6. The methodology was then tested on multiple industrial assembly models with different configurations and dimensions (length, diameter) as well as on a real steel pipe spool assembly.

When designing the 3D model database, the objective was to make sure that the assemblies are realistic and simulate real scenarios encountered in the industry. The value provided is based on the designed models and may change with different designs. However, the provided framework to obtain the MRM is universal and can be applied to any assembly.

7.2 Research Contributions and Conclusions

The contributions of the work are as follows: (1) an innovative method was developed that uses a non-software dependent 3D model and its Piping Component File to accurately segment piping components from their complete model; (2) a worker-oriented 3D visual stepwise quality control for piping assemblies designed in fabrication shop was proposed; (3) building on the geometrical properties of piping assemblies and the limitations of handheld laser scanners, a solution to reduce the 3D model to its minimum was realized.

Following from these contributions are four conclusions:

1. The utilization of two software independent inputs in the methodology renders an important adaptability to the overall methodology that can thus be used independently from the original design model format. In particular, PCFs have been found to be very effective and trustworthy input files. They provide flexible data that can be easily post-processed.
2. Based on the designed database and the steel laboratory pipe spool, the MRM impact has been demonstrated to be more important on large and spatially complex assemblies.
3. The experiments have shown a consequent increase in the utility of consumer grade scanners (i.e. Structure IO) with the developed methodology paving the way to their emergence for industrial applications. In other words, when 3D models are not reduced, those devices may not have the intrinsic range to scan the entire model, whereas using the MRM, the required scanning size would potentially be reduced allowing those scanners to be used.
4. Consumer grades scanners that use structured light technology (i.e. Structure IO) to acquired point clouds have their density varying based on the scanning depth. Scanning smaller sized assemblies can thus enhance the as-built point cloud density, which benefits the deviation analysis accuracy. Consequently, most of the emerging Scan-vs-3D model applications could use the MRM with positive effect.

7.3 Limitations

Despite the continuous move toward digitization of the construction industry, a lot of companies are still not using 3D models for their design. In a lot of projects, providing 3D data to the contractor is not a mandatory duty, and thereby, none of the key players is urged to use a 3D design of the project. As the entire method is built on 3D files, if no data can be obtained from a project then it becomes impossible to apply the proposed stepwise control process.

Another potential limitation to the overall method is the scanner precision. Indeed, based on the required assembly tolerances, the scanner may not be able to acquired precise enough data. As presented in the

thesis, the Structure IO's precision decreases logarithmically with the distance. The low level of precision could simply prevent the utilization of those devices for industrial applications. The accuracy itself is complex to quantify as it depends on multiple factors, which are hard to measure, but the intrinsic precision unfortunately lowers the odds to obtain an accurate scan.

Last but not least, quantifying the potential geometric deviations could be bothersome, since some lengths, which would have been used to visually estimate the deviation, are removed from the deviation analysis. In addition, automatic deviation measurement and orientation detection have not yet been implemented in the SfM software. For now, the worker has to use a heat map to estimate the location of the deviation and quantify it based on individual points color. An additional solution should therein be proposed in the software to (semi) automatically display the information. Otherwise, the deviation analysis could become useless to the worker regardless to any other considerations.

7.4 Future Work

In order to truly evaluate the effectiveness of the method, future work could be conducted and would consist of applying the proposed methodology for each step of the construction assembly fabrication during the construction phase. Following workers' steps of assembling would enable observation of the real stages of fabrication and thus applying the methodology for each stage of the evaluated assembly. Several industrial assemblies would be investigated under the proposed measurements of effectiveness, and the utility of the MRM would be measured based on real fabrication shop situations.

Other benefits of the MRM should be analyzed such as time and cost savings, which could be measured and compared to the same process without the MRM, hence, obtaining a monetary impact factor for the MRM. Performing stepwise quality control using the MRM with a consumer grade scanning device may reduce even more rework and measurement time. Also, the use of such a technique would certainly reduce the number of non-compliances detected at the end of the fabrication, which could either be caused by a worker's mistake or by the propagation of tolerances along the assembly. Therefore, quantifying the savings of the described methodology in the overall construction process of industrial assemblies could be done as future work.

Finally, other domains of applicability of the MRM may be investigated. For manufactured industrial assemblies where QC is performed, the MRM overall workflow could potentially be tested. Considering the rise of digitization in industries and, in particular, the recourse to 3D models for every type of product, applying the MRM for 3D visual stepwise geometric quality control may be of great value and cost efficient in plenty of situations.

8 References

- 3D Imaging Market Global Scenario, Market Size, Trend and Forecast, 2015 – 2024. 2018. <https://www.variantmarketresearch.com/report-categories/semiconductor-electronics/3d-imaging-market>
- 3D Imaging Market Size, Trends and Forecast. 2018. <https://www.alliedmarketresearch.com/3D-imaging-market>
- Agarwal, Rajat, Shankar Chandrasekaran, and Mukund Sridhar. 2016. Imagining Construction's Digital Future. McKinseyandCompany, no. June: 1–13.
- Aiteanu, D., Hillers, B., and Gräser, A. 2003. A step forward in manual welding: demonstration of augmented reality helmet. *In Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*, p. 309.
- Akinci, B., Boukamp, F., Gordon, C., Huber, D., Lyons, C., and Park, K. 2006. A formalism for utilization of sensor systems and integrated project models for active construction quality control. *Automation in construction*, **15**(2): 124-138.
- Anil, E. B., Tang, P., Akinci, B., and Huber, D. 2013. Deviation analysis method for the assessment of the quality of the as-is Building Information Models generated from point cloud data. *Automation in Construction*, **35**: 507-516.
- Antaki, G. A., Monahon, T. M., and Cansler, R. W. 2005. Risk-Based Inspection (RBI) of Steam Systems. *In ASME 2005 Pressure Vessels and Piping Conference*, pp. 65-71.
- Ballast, D. K. 2007. *Handbook of construction tolerances*. John Wiley and Sons.
- Bhatla, A., Choe, S. Y., Fierro, O., and Leite, F. 2012. Evaluation of accuracy of as-built 3D modeling from photos taken by handheld digital cameras. *Automation in construction*, **28**: 116-127.
- Bosché, F., Ahmed, M., Turkan, Y., Haas, C. T., and Haas, R. 2015. The value of integrating Scan-to-BIM and Scan-vs-BIM techniques for construction monitoring using laser scanning and BIM: The case of cylindrical MEP components. *Automation in Construction*, **49**: 201-213.
- Bosché, F. 2010. Automated recognition of 3D CAD model objects in laser scans and calculation of as-built dimensions for dimensional compliance control in construction. *Advanced engineering informatics*, **24**(1): 107-118.
- Bosché, F., Guillemet, A., Turkan, Y., Haas, C. T., and Haas, R. 2013. Tracking the built status of MEP works: Assessing the value of a Scan-vs-BIM system. *Journal of computing in civil engineering*, **28**(4).
- Bosché, F., Haas, C. T., and Akinci, B. 2009. Automated recognition of 3D CAD objects in site laser scans for project 3D status visualization and performance control. *Journal of Computing in Civil Engineering*, **23**(6): 311-318.
- Brilakis, I., Lourakis, M., Sacks, R., Savarese, S., Christodoulou, S., Teizer, J., and Makhmalbaf, A. 2010. Toward automated generation of parametric BIMs based on hybrid video and laser scanning data. *Advanced Engineering Informatics*, **24**(4): 456-465.
- Chen, L., and Luo, H. 2014. A BIM-based construction quality management model and its applications. *Automation in construction*, **46**: 64-73.
- Chien, K. F., Wu, Z. H., and Huang, S. C. 2014. Identifying and assessing critical risk factors for BIM projects: Empirical study. *Automation in Construction*, **45**: 1-15.
- Diaz, R. A., Herrera, W. J., and Martinez, R. 2006. Moments of inertia for solids of revolution and variational methods. *European journal of physics*, **27**(2): 183.
- Farnsworth, C. B., Beveridge, S., Miller, K. R., and Christofferson, J. P. 2015. Application, advantages, and methods associated with using BIM in commercial construction. *International Journal of Construction Education and Research*, **11**(3): 218-236.

- Fatal occupational injuries by selected characteristics, 2003-2014. 2003.
https://www.bls.gov/iif/oshwc/cfoi/all_worker.pdf
- Fathi, H., Dai, F., and Lourakis, M. 2015. Automated as-built 3D reconstruction of civil infrastructure using computer vision: Achievements, opportunities, and challenges. *Advanced Engineering Informatics*, **29**(2): 149-161.
- Flanges and Bolt Dimensions ASME/ANSI B16.5 - Class 150 to 2500.
https://www.engineeringtoolbox.com/flanges-bolts-dimensions-d_464.html
- Freedonia Group. 2013. World Electric Motors - Industry Market Research, Market Share, Market Size, Sales, Demand Forecast, Market Leaders.
- Fung, M., Fung, A., Fung, R. H., and Fung, P. K. 2014. The Advantage of Using a File Data Transfer Method in a Plant Design. *In ASME 2014 Pressure Vessels and Piping Conference*.
- Garcia, M. A., Llanos, D. R., and de Prada, C. 1999. A configurable ACSL-based interface generator for simulated systems. *Simulation*, **73**(4): 206-212.
- Gnanavel, C., Saravanan, R., Chandrasekaran, M., and Jayakanth, J. J. 2017. Improvement of Productivity in TIG Welding Plant by Equipment Design in Orbit. *In IOP Conference Series: Materials Science and Engineering*, **183**(1).
- Golparvar-Fard, M., Bohn, J., Teizer, J., Savarese, S., and Peña-Mora, F. 2011. Evaluation of image-based modeling and laser scanning accuracy for emerging automated performance monitoring techniques. *Automation in Construction*, **20**(8): 1143-1155.
- Golparvar-Fard, M., Peña-Mora, F., and Savarese, S. 2011. Integrated sequential as-built and as-planned representation with D 4 AR tools in support of decision-making tasks in the AEC/FM industry. *Journal of Construction Engineering and Management*, **137**(12): 1099-1116.
- Goodrum, P. M., Miller, J., Sweany, J., and Alruwaythi, O. 2016. Influence of the format of engineering information and spatial cognition on craft-worker performance. *Journal of Construction Engineering and Management*, **142**(9).
- Holzer, D. 2007. Are you talking to me? Why BIM alone is not the answer.
- Hou, L., and Wang, X. (2011). Experimental framework for evaluating cognitive workload of using AR system for general assembly task. *In Proceedings of the 28th International Symposium on Automation and Robotics in Construction*.
- Hwang, B. G., Thomas, S. R., Haas, C. T., and Caldas, C. H. 2009. Measuring the impact of rework on construction cost performance. *Journal of Construction Engineering and Management*, **135**(3): 187-198.
- Irizarry, J., and Costa, D. B. 2016. Exploratory study of potential applications of unmanned aerial systems for construction management tasks. *Journal of Management in Engineering*, **32**(3): 05016001.
- Isogen Symbol Key (SKEY) Definitions, Alias piping solutions, 2007.
http://www.academia.edu/29513874/ISOGEN_Symbol_Key_SKEY_Definitions
- Jaillon, L., and Poon, C. S. 2014. Life cycle design and prefabrication in buildings: A review and case studies in Hong Kong. *Automation in Construction*, **39**: 195-202.
- Jaillon, L., and Poon, C. S. 2008. Sustainable construction aspects of using prefabrication in dense urban environment: a Hong Kong case study. *Construction Management and Economics*, **26**(9): 953-966.
- Jeanclos, N., Sharif, M. M., Li, S. K., Kwiatek, C., and Haas, C. 2018, June. Derivation of Minimum Required Model for Augmented Reality Based Stepwise Construction Assembly Control. *In Workshop of the European Group for Intelligent Computing in Engineering*, Springer, Vol I, pp. 336-358.
- Jeffery, G. B. 1915. On the steady rotation of a solid of revolution in a viscous fluid. *Proceedings of the London Mathematical Society*, **2**(1): 327-338.

- Kim, C., Son, H., Kim, H., and Han, S. H. 2008. Applicability of flash laser distance and ranging to three-dimensional spatial information acquisition and modeling on a construction site. *Canadian Journal of Civil Engineering*, **35**(11): 1331-1341.
- Lee, J., Lee, K., Nam, B., and Wu, Y. 2016. IoT Platform-based iAR: a Prototype for Plant O&M Applications. *In Mixed and Augmented Reality (ISMAR-Adjunct)*, 2016 IEEE International Symposium on, pp. 149-150.
- Leite, F., Cho, Y., Behzadan, A. H., Lee, S., Choe, S., Fang, Y., and Hwang, S. 2016. Visualization, information modeling, and simulation: Grand challenges in the construction industry. *Journal of Computing in Civil Engineering*, **30**(6): 04016035.
- Makris, S., Pintzos, G., Rentzos, L., and Chryssolouris, G. 2013. Assembly support using AR technology based on automatic sequence generation. *CIRP Annals-Manufacturing Technology*, **62**(1): 9-12.
- Malamas, E. N., Petrakis, E. G., Zervakis, M., Petit, L., and Legat, J. D. 2003. A survey on industrial vision systems, applications and tools. *Image and vision computing*, **21**(2) 171-188.
- Markowitz, W. M. 1973. SI, the international system of units. *Geophysical surveys*, **1**(2): 217-241.
- Mitra, N. J., Pauly, M., Wand, M., and Ceylan, D. 2013. Symmetry in 3d geometry: Extraction and applications. *In Computer Graphics Forum*, **32**(6):1-23.
- Mitra, N. J., and Pauly, M. 2008. Symmetry for architectural design. *Advances in Architectural Geometry*, 13-16.
- Molleda, J., Usamentiaga, R., García, D. F., Bulnes, F. G., Espina, A., Dieye, B., and Smith, L. N. 2013. An improved 3D imaging system for dimensional quality inspection of rolled products in the metal industry. *Computers in Industry*, **64**(9): 1186-1200.
- Nahangi, M., Czerniawski, T., Haas, C. T., Walbridge, S., and West, J. 2015. Parallel systems and structural frames realignment planning and actuation strategy. *Journal of Computing in Civil Engineering*, **30**(4): 04015067.
- Newman, T. S., and Jain, A. K. 1995. A survey of automated visual inspection. *Computer vision and image understanding*, **61**(2): 231-262.
- Omar, T., and Nehdi, M. L. 2016. Data acquisition technologies for construction progress tracking. *Automation in Construction*, **70**: 143-155.
- Pan, W., Gibb, A. G., and Dainty, A. R. 2012. Strategies for integrating the use of off-site production technologies in house building. *Journal of Construction Engineering and Management*, **138**(11): 1331-1340.
- Patow, G., and Besuievsky, G. 2013. Challenges in procedural modeling of buildings. *In Proceedings of the Eurographics Workshop on Urban Data Modelling and Visualisation*, pp. 25-28.
- Piping Data Exchange Using the Super PCF. 2018.
<https://hexagonppm.com/products/pfc-product-family/isogen/super-pcf>
- Safa, M., Shahi, A., Nahangi, M., Haas, C., and Noori, H. 2015. Automating measurement process to improve quality management for piping fabrication. *In Structures*, **3**: 71-80. Elsevier.
- Schwerdtfeger, B., Pustka, D., Hofhauser, A., and Klinker, G. 2008. Using laser projectors for augmented reality. *In Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, pp. 134-137. ACM.
- Sharif, M.M., Nahangi, M., Haas, C., West, J., Ibrahim, M. 2016. A preliminary investigation of the applicability of portable sensors for fabrication and installation control of industrial assemblies. *In Canadian Society for Civil Engineering*.
- Sharif, M. M., Nahangi, M., Haas, C., and West, J. 2017. Automated Model-Based Finding of 3D Objects in Cluttered Construction Point Cloud Models. *Computer-Aided Civil and Infrastructure Engineering*, **32**(11): 893-908.
- Siebert, S., and Teizer, J. 2014. Mobile 3D mapping for surveying earthwork projects using an Unmanned Aerial Vehicle (UAV) system. *Automation in Construction*, **41**: 1-14.

- Simari, P., Kalogerakis, E., and Singh, K. 2006. Folding meshes: Hierarchical mesh segmentation based on planar symmetry. *In* Symposium on geometry processing, **256**: 111-119.
- Steinberger, M., Kenzel, M., Kainz, B., Wonka, P., and Schmalstieg, D. 2014. On-the-fly generation and rendering of infinite cities on the GPU. *In* Computer graphics forum, **33**(2): 105-114.
- Tang, P., Anil, E. B., Akinci, B., and Huber, D. 2011. Efficient and effective quality assessment of as-is building information models and 3D laser-scanned data. *In* Computing in Civil Engineering, pp. 486-493.
- Tang, P., Huber, D., Akinci, B., Lipman, R., and Lytle, A. 2010. Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. *Automation in construction*, **19**(7): 829-843.
- Tang, P., and Akinci, B. 2012. Automatic execution of workflows on laser-scanned data for extracting bridge surveying goals. *Advanced Engineering Informatics*, **26**(4): 889-903.
- Thielsch, H. 1965. *Defects and failures in pressure vessels and piping*. New York: Reinhold.
- Turkan, Y., Bosche, F., Haas, C. T., and Haas, R. 2012. Automated progress tracking using 4D schedule and 3D sensing technologies. *Automation in Construction*, **22**: 414-421.
- Wang, J., Sun, W., Shou, W., Wang, X., Wu, C., Chong, H. Y., and Sun, C. 2015. Integrating BIM and LiDAR for real-time construction quality control. *Journal of Intelligent and Robotic Systems*, **79**(3-4): 417-432.
- Wang, X., Ong, S. K., and Nee, A. Y. C. 2016. Multi-modal augmented-reality assembly guidance based on bare-hand interface. *Advanced Engineering Informatics*, **30**(3): 406-421.
- Wang, X., Truijens, M., Hou, L., Wang, Y., and Zhou, Y. 2014. Integrating Augmented Reality with Building Information Modeling: Onsite construction process controlling for liquefied natural gas industry. *Automation in Construction*, **40**: 96-105.

Appendix A

Definition of pipe

```
if (item.Split()[0] == "PIPE")
{
    SfMPoint endPoint1 = null;
    SfMPoint endPoint2 = null;
    double diameter1 = 0;
    double diameter2 = 0;
    foreach (string attribute in parsedDictionary[item].Keys)
    {
        if (attribute.Split()[0] == "END-POINT")
        {
            string[] attributeSplitted =
parsedDictionary[item][attribute].Split((string[])null,
StringSplitOptions.RemoveEmptyEntries);
            double x = Convert.ToDouble(attributeSplitted[0]);
            double y = Convert.ToDouble(attributeSplitted[1]);
            double z = Convert.ToDouble(attributeSplitted[2]);
            double diameter = Convert.ToDouble(attributeSplitted[3]);
            if (endPoint1 == null)
            {
                endPoint1 = new SfMPoint(x, y, z);
                diameter1 = diameter;
            }
            else
            {
                endPoint2 = new SfMPoint(x, y, z);
                diameter2 = diameter;
                break;
            }
        }
    }
    allComponents.Add(new SfMPipe(endPoint1, endPoint2, diameter1,
diameter2));
}
```

Extraction of elbow from the PCF

```
else if (item.Split()[0] == "ELBOW")
{
    SfMPoint endPoint1 = null;
    SfMPoint endPoint2 = null;
    SfMPoint centrePoint = null;
    double diameter1 = 0;
    double diameter2 = 0;
    foreach (string attribute in parsedDictionary[item].Keys)
    {
        if (attribute.Split()[0] == "END-POINT")
        {
            string[] attributeSplitted =
parsedDictionary[item][attribute].Split((string[])null,
StringSplitOptions.RemoveEmptyEntries);
            double x = Convert.ToDouble(attributeSplitted[0]);
            double y = Convert.ToDouble(attributeSplitted[1]);
            double z = Convert.ToDouble(attributeSplitted[2]);

```



```

        double diameter = Convert.ToDouble(attributeSplitted[3]);
        if (endPoint1 == null)
        {
            endPoint1 = new SfMPoint(x, y, z);
            diameter1 = diameter;
        }
        else
        {
            endPoint2 = new SfMPoint(x, y, z);
            diameter2 = diameter;
            if (centrePoint != null)
                break;
        }
    }
    else if (attribute.Split()[0] == "CENTRE-POINT")
    {
        string[] attributeSplitted =
parsedDictionary[item][attribute].Split((string[])null,
StringSplitOptions.RemoveEmptyEntries);
        double x = Convert.ToDouble(attributeSplitted[0]);
        double y = Convert.ToDouble(attributeSplitted[1]);
        double z = Convert.ToDouble(attributeSplitted[2]);
        centrePoint = new SfMPoint(x, y, z);
        if (endPoint2 != null)
            break;
    }
}
allComponents.Add(new SfMElbow(endPoint1, endPoint2, centrePoint,
diameter1, diameter2));
}

```

Distance of a line to a segment for pipes

```

/// <summary>
/// Determine whether a point is within the cylindrical spatial boundary defined
by a pipe
/// </summary>
/// <param name="pipe">the pipe of interest, defines a cylindrical spatial
boundary</param>
/// <returns>True if the point is within the cylindrical spatial boundary defined
by the pipe, vise versa</returns>
public bool isInPipe(SfMPipe pipe)
{
    SfMPoint v = pipe.endPoint2 - pipe.endPoint1;
    SfMPoint w = this - pipe.endPoint1;

    double c1 = w * v;
    if (c1 <= 0)
        return false;

    double c2 = v * v;
    if (c2 <= c1)
        return false;

    double b = c1 / c2;
    SfMPoint pointB = pipe.endPoint1 + b * v;

```

```

        return distance(this, pointB) < PipeSpoolManager.toleratedInchToMillimeter *
pipe.diameter1;
    }

```

Distance of a point to a circle for elbows

```

    /// <summary>
    /// Determine whether a point is within the spatial boundary defined by an elbow
    /// </summary>
    /// <param name="elbow">the pipe of interest, defines an elbow shaped (part of a
torus) spatial boundary</param>
    /// <returns>True if the point is within the spatial boundary defined by the
elbow, vise versa</returns>
    public bool isInElbow(SfMElbow elbow)
    {
        SfMPoint v = this - elbow.circleCentre;
        double perpendicularDistance = v * elbow.normal;
        SfMPoint q = this - perpendicularDistance * elbow.normal;
        SfMPoint projectedToCenter = q - elbow.circleCentre;
        double parallelDistance = Math.Sqrt(projectedToCenter * projectedToCenter) -
elbow.radius;
        double distance = Math.Sqrt(parallelDistance * parallelDistance +
perpendicularDistance * perpendicularDistance);

        if (distance > PipeSpoolManager.toleratedInchToMillimeter * elbow.diameter1)
            return false;

        SfMPoint oq = q - elbow.circleCentre;

        double angle1 = Math.Acos(elbow.radius1 * oq / (Math.Sqrt(elbow.radius1 *
elbow.radius1) * Math.Sqrt(oq * oq)));
        double angle2 = Math.Acos(elbow.radius2 * oq / (Math.Sqrt(elbow.radius2 *
elbow.radius2) * Math.Sqrt(oq * oq)));

        if (Math.Abs(angle1 + angle2 - elbow.angle) < 0.01)
            return true;

        return false;
    }

```

Calculation of the parameters of an elbow

```

public class SfMElbow
{
    /// <summary>
    /// first endpoint of the elbow
    /// </summary>
    public SfMPoint endPoint1;
    /// <summary>
    /// second endpoint of the elbow
    /// </summary>
    public SfMPoint endPoint2;
    /// <summary>
    /// centre point of the elbow
    /// </summary>

```

```

public SfMPoint circleCentre;
/// <summary>
/// diameter of the elbow at the first endpoint
/// </summary>
public double diameter1;
/// <summary>
/// diameter of the elbow at the second endpoint
/// </summary>
public double diameter2;

public SfMPoint radius1;

public SfMPoint radius2;
/// <summary>
/// Normal vector that the circle lies on
/// </summary>
public SfMPoint normal;
/// <summary>
/// The radius of curvature of the elbow
/// </summary>
public double radius;
/// <summary>
/// The angle between the two pipes that the elbow connect to
/// </summary>
public double angle;

/// <summary>
/// Constructor of a elbow, input: 2 endpoints and 2 corresponding diameters
/// </summary>
/// <param name="point1">first endpoint of the new elbow</param>
/// <param name="point2">second endpoint of the new elbow</param>
/// <param name="diameter1">diameter of the new elbow at the first
endpoint</param>
/// <param name="diameter2">diameter of the new elbow at the second
endpoint</param>
public SfMElbow(SfMPoint endPoint1, SfMPoint endPoint2, SfMPoint centrePoint,
double diameter1, double diameter2)
{
    this.endPoint1 = endPoint1;
    this.endPoint2 = endPoint2;

    this.diameter1 = diameter1;
    this.diameter2 = diameter2;

    SfMPoint chordMidPoint = 0.5 * (endPoint1 + endPoint2);

    SfMPoint tangent1 = endPoint1 - centrePoint;
    SfMPoint tangent2 = endPoint2 - centrePoint;
    normal = SfMPoint.normalize(SfMPoint.cross(tangent1, tangent2));

    double theta = Math.Acos(tangent1 * tangent2 / (Math.Sqrt(tangent1 *
tangent1) * Math.Sqrt(tangent2 * tangent2)));
    angle = Math.PI - theta;

    SfMPoint halfChord = endPoint1 - chordMidPoint;

    SfMPoint vectorToCentre = SfMPoint.normalize(chordMidPoint - centrePoint);
    double om = Math.Sqrt(halfChord * halfChord) * Math.Tan(theta / 2);

```

```

circleCentre = chordMidPoint + om * vectorToCentre;

SfMPoint r = endPoint1 - circleCentre;
radius = Math.Sqrt(r * r);

radius1 = endPoint1 - circleCentre;
radius2 = endPoint2 - circleCentre;
}

```

Determination of the belonging of point to a flange

```

/// <summary>
/// Determine whether a point is within the cylindrical spatial boundary defined
by a flange
/// </summary>
/// <param name="flange">the flange of interest, defines a cylindrical spatial
boundary</param>
/// <returns>True if the point is within the cylindrical spatial boundary defined
by the flange, vice versa</returns>
public bool isInFlange(SfMFlange flange)
{
    SfMPoint v = flange.endPoint2 - flange.endPoint1;
    SfMPoint w = this - flange.endPoint1;

    double c1 = w * v;
    if (c1 <= 0)
        return false;

    double c2 = v * v;
    if (c2 <= c1)
        return false;

    double b = c1 / c2;
    SfMPoint pointB = flange.endPoint1 + b * v;
    return distance(this, pointB) < PipeSpoolManager.toleratedInchToMillimeter *
flange.diameter1 * flangeDiameter(flange.classNumber, flange.diameter1);
}

```

Appendix B

```
/// <summary>
/// A pipe spool component key point graph.
/// Containing set of geometric key points as its nodes, and an edge between the
nodes if they are connected in the pipe spool.
/// Used to obtain an consistent ordering of the key points and derive the minimum
required planning horizon of the pipe spool.
/// Implemented for the Structure from Motion project.
/// </summary>
public class SfmKeyPointGraph
{
    /// <summary>
    /// Set of nodes, i.e. set of all key points in all the selected pipe spool
components
    /// </summary>
    public List<SfmKeyPointGraphNode> nodes;
    /// <summary>
    /// Coordinates of the calculated weld point
    /// </summary>
    public SfmPoint weldPoint;
    /// <summary>
    /// Number of weld encountered. Every weld s encountered twice due to the double
counting in the nested loop.
    /// If weld count is 0, then no common key point between the selected reference
and addition parts.
    /// If weld count is 2, then there is a unique common key point between the
reference and addition parts.
    /// ^(The derivation of the Minimum Required Model can be carried out using the
unique weld as the starting point)
    /// If weld count is 4, then there are more than one common key poitin between the
reference and addition parts.
    /// </summary>
    public int weldCount;
    /// <summary>
    /// Constructor of a key point graph. Construct an empty graph
    /// </summary>
    public SfmKeyPointGraph()
    {
        nodes = new List<SfmKeyPointGraphNode>();
    }

    /// <summary>
    /// Overrides the default ToString() method,
    /// returns a string that is a user friendly representation of the key point
graph by listing out all its nodes in a sequential manner
    /// </summary>
    /// <returns>a string that is a user friendly representation of the key point
graph</returns>
    public override string ToString()
    {
        string graphString = "";
        foreach (SfmKeyPointGraphNode node in nodes)
            graphString += node.ToString() + "\n";
        return graphString;
    }

    /// <summary>
```

```

    /// Extract all the key points (e.g. end points, centre point, and branch point
    for a tee) from the list of components,
    /// then generate all the internal (within the component) edges.
    /// After this step, most of the key points will be included twice in the graph
    (once from each component it belongs to).
    /// </summary>
    /// <param name="referenceComponents">List of selected reference
components</param>
    /// <param name="additionComponents">List of selected addition components</param>
    public void extractKeyPoints(List<object> referenceComponents, List<object>
additionComponents)
    {
        foreach (object component in referenceComponents)
        {
            if (component is SfmCylindricalGeneric)
            {
                SfmCylindricalGeneric cylindricalGeneric = component as
SfmCylindricalGeneric;
                SfmKeyPointGraphNode endNode1 = new
SfmKeyPointGraphNode(cylindricalGeneric.endPoint1, true);
                SfmKeyPointGraphNode endNode2 = new
SfmKeyPointGraphNode(cylindricalGeneric.endPoint2, true);
                endNode1.addNeighbour(endNode2);
                endNode2.addNeighbour(endNode1);
                nodes.Add(endNode1);
                nodes.Add(endNode2);
            }
            else if (component is SfmElbow)
            {
                SfmElbow elbow = component as SfmElbow;
                SfmKeyPointGraphNode endNode1 = new
SfmKeyPointGraphNode(elbow.endPoint1, true);
                SfmKeyPointGraphNode endNode2 = new
SfmKeyPointGraphNode(elbow.endPoint2, true);
                SfmKeyPointGraphNode centreNode = new
SfmKeyPointGraphNode(elbow.pcfCentre, true);
                endNode1.addNeighbour(centreNode);
                endNode2.addNeighbour(centreNode);
                centreNode.addNeighbour(endNode1);
                centreNode.addNeighbour(endNode2);
                nodes.Add(endNode1);
                nodes.Add(endNode2);
                nodes.Add(centreNode);
            }
            else if (component is SfmFlange)
            {
                SfmFlange flange = component as SfmFlange;
                SfmKeyPointGraphNode endNode1 = new
SfmKeyPointGraphNode(flange.endPoint1, true);
                SfmKeyPointGraphNode endNode2 = new
SfmKeyPointGraphNode(flange.endPoint2, true);
                endNode1.addNeighbour(endNode2);
                endNode2.addNeighbour(endNode1);
                nodes.Add(endNode1);
                nodes.Add(endNode2);
            }
            else if (component is SfmTee)
            {

```

```

        SfMTee tee = component as SfMTee;
        SfMKeyPointGraphNode endNode1 = new
SfMKeyPointGraphNode(tee.endPoint1, true);
        SfMKeyPointGraphNode endNode2 = new
SfMKeyPointGraphNode(tee.endPoint2, true);
        SfMKeyPointGraphNode centreNode = new
SfMKeyPointGraphNode(tee.centrePoint, true);
        SfMKeyPointGraphNode branchNode = new
SfMKeyPointGraphNode(tee.branchPoint, true);
        endNode1.addNeighbour(centreNode);
        endNode2.addNeighbour(centreNode);
        branchNode.addNeighbour(centreNode);
        centreNode.addNeighbour(endNode1);
        centreNode.addNeighbour(endNode2);
        centreNode.addNeighbour(branchNode);
        nodes.Add(endNode1);
        nodes.Add(endNode2);
        nodes.Add(centreNode);
        nodes.Add(branchNode);
    }
}
foreach (object component in additionComponents)
{
    if (component is SfMCylindricalGeneric)
    {
        SfMCylindricalGeneric cylindricalGeneric = component as
SfMCylindricalGeneric;
        SfMKeyPointGraphNode endNode1 = new
SfMKeyPointGraphNode(cylindricalGeneric.endPoint1, false);
        SfMKeyPointGraphNode endNode2 = new
SfMKeyPointGraphNode(cylindricalGeneric.endPoint2, false);
        endNode1.addNeighbour(endNode2);
        endNode2.addNeighbour(endNode1);
        nodes.Add(endNode1);
        nodes.Add(endNode2);
    }
    else if (component is SfMElbow)
    {
        SfMElbow elbow = component as SfMElbow;
        SfMKeyPointGraphNode endNode1 = new
SfMKeyPointGraphNode(elbow.endPoint1, false);
        SfMKeyPointGraphNode endNode2 = new
SfMKeyPointGraphNode(elbow.endPoint2, false);
        SfMKeyPointGraphNode centreNode = new
SfMKeyPointGraphNode(elbow.pcfCentre, false);
        endNode1.addNeighbour(centreNode);
        endNode2.addNeighbour(centreNode);
        centreNode.addNeighbour(endNode1);
        centreNode.addNeighbour(endNode2);
        nodes.Add(endNode1);
        nodes.Add(endNode2);
        nodes.Add(centreNode);
    }
    else if (component is SfMFlange)
    {
        SfMFlange flange = component as SfMFlange;
        SfMKeyPointGraphNode endNode1 = new
SfMKeyPointGraphNode(flange.endPoint1, false);

```

```

        SfMKeyPointGraphNode endNode2 = new
SfMKeyPointGraphNode(flange.endPoint2, false);
        endNode1.addNeighbour(endNode2);
        endNode2.addNeighbour(endNode1);
        nodes.Add(endNode1);
        nodes.Add(endNode2);
    }
    else if (component is SfMTee)
    {
        SfMTee tee = component as SfMTee;
        SfMKeyPointGraphNode endNode1 = new
SfMKeyPointGraphNode(tee.endPoint1, false);
        SfMKeyPointGraphNode endNode2 = new
SfMKeyPointGraphNode(tee.endPoint2, false);
        SfMKeyPointGraphNode centreNode = new
SfMKeyPointGraphNode(tee.centrePoint, false);
        SfMKeyPointGraphNode branchNode = new
SfMKeyPointGraphNode(tee.branchPoint, false);
        endNode1.addNeighbour(centreNode);
        endNode2.addNeighbour(centreNode);
        branchNode.addNeighbour(centreNode);
        centreNode.addNeighbour(endNode1);
        centreNode.addNeighbour(endNode2);
        centreNode.addNeighbour(branchNode);
        nodes.Add(endNode1);
        nodes.Add(endNode2);
        nodes.Add(centreNode);
        nodes.Add(branchNode);
    }
}
}

/// <summary>
/// Merge all the key point nodes with the same (or within 5 millimeters)
coordinates into the same node.
/// All the neighbours of the nodes that are being merged become neighbours of
the new node
/// </summary>
public void mergeDuplicates()
{
    weldCount = 0;
    List<SfMKeyPointGraphNode> oldNodes = new List<SfMKeyPointGraphNode>(nodes);
    nodes.Clear();
    for (int i = 0; i < oldNodes.Count; i++)
    {
        Boolean duplicateFound = false;
        // Find and merge all the nodes with the same (or within 5 millimeters)
coordinates
        for (int j = 0; j < oldNodes.Count; j++)
            if (i != j && SfMPoint.distance(oldNodes[i].keyPoint,
oldNodes[j].keyPoint) < PipeSpoolManager.minimumAcceptableLength)
            {
                SfMKeyPointGraphNode newNode;
                if (oldNodes[i].isReference && oldNodes[j].isReference)
                    newNode = new SfMKeyPointGraphNode(oldNodes[i].keyPoint,
true);
                else if (!oldNodes[i].isReference && !oldNodes[j].isReference)

```



```

        newNode = new SfmKeyPointGraphNode(oldNodes[i].keyPoint,
false);
        else
        {
            newNode = new SfmKeyPointGraphNode(oldNodes[i].keyPoint,
true);

            weldPoint = newNode.keyPoint;
            weldCount++;
        }

        List<SfmKeyPointGraphNode> newNeighbours = new
List<SfmKeyPointGraphNode>(oldNodes[i].neighbours);

        newNeighbours.AddRange(oldNodes[j].neighbours);

        newNode.neighbours = newNeighbours;

        if (i < j)
            nodes.Add(newNode);

        duplicateFound = true;
    }
    // If duplicate is not found for a node, add it directly to the new list
    if (!duplicateFound)
        nodes.Add(oldNodes[i]);
}

// Remake the list of nodes, to fix the reference mismatch issue
for (int i = 0; i < nodes.Count; i++)
{
    SfmKeyPointGraphNode node = nodes[i];
    List<SfmKeyPointGraphNode> newNeighbours = new
List<SfmKeyPointGraphNode>();
    foreach (SfmKeyPointGraphNode neighbour in node.neighbours)
        for (int j = 0; j < nodes.Count; j++)
            if (i != j && SfmPoint.distance(neighbour.keyPoint,
nodes[j].keyPoint) < PipeSpoolManager.minimumAcceptableLength)
                newNeighbours.Add(nodes[j]);
    node.neighbours = newNeighbours;
}
}

/// <summary>
/// Given a selected node, order all the nodes in the graph in a consistent
sequential manner by traversing the nodes
/// The traversal is stopped whenever the component have more than two neighbours
/// </summary>
/// <param name="selectedNode">Selected node, the starting point of the
traversal</param>
public List<SfmKeyPointGraphNode>
getMinimumRequiredModelKeyPoints(SfmKeyPointGraphNode selectedNode)
{
    List<SfmKeyPointGraphNode> fullKeyPointNodes = new
List<SfmKeyPointGraphNode>();
    fullKeyPointNodes.Add(selectedNode);
    // If the selected node has no neighbours, then the ordered node list have
only such node, and is already sorted
    if (selectedNode.neighbours.Count == 0)

```

```

        return fullKeyPointNodes;
    // If the selected node has one and only one neighbour, then it is an end
point.
    // Go one direction and add all the traversed nodes in order
    else if (selectedNode.neighbours.Count == 1)
    {
        SfmKeyPointGraphNode previousNode = selectedNode;
        SfmKeyPointGraphNode currentNode = selectedNode.neighbours[0];

        List<SfmKeyPointGraphNode> branchKeyPointNodes = new
List<SfmKeyPointGraphNode>();
        while (currentNode.neighbours.Count > 1)
        {
            if (currentNode.neighbours.Count > 2)
            {
                fullKeyPointNodes.AddRange(branchKeyPointNodes);
                return fullKeyPointNodes;
            }
            else
            {
                if (SfmPoint.distance(currentNode.neighbours[0].keyPoint,
previousNode.keyPoint) < PipeSpoolManager.minimumAcceptableLength)
                {
                    double angle =
SfmPoint.angleFromVectors(previousNode.keyPoint - currentNode.keyPoint,
currentNode.keyPoint - currentNode.neighbours[1].keyPoint);
                    previousNode = currentNode;
                    currentNode = currentNode.neighbours[1];
                    Console.WriteLine(angle);
                    if (angle > 0.001)
                    {
                        fullKeyPointNodes.AddRange(branchKeyPointNodes);
                        break;
                    }
                }
                else
                {
                    double angle =
SfmPoint.angleFromVectors(previousNode.keyPoint - currentNode.keyPoint,
currentNode.keyPoint - currentNode.neighbours[0].keyPoint);
                    previousNode = currentNode;
                    currentNode = currentNode.neighbours[0];
                    Console.WriteLine(angle);
                    if (angle > 0.001)
                    {
                        fullKeyPointNodes.AddRange(branchKeyPointNodes);
                        break;
                    }
                }
                branchKeyPointNodes.Add(currentNode);
            }
        }
        return fullKeyPointNodes;
    }
    // If the selected node has two neighbours, then it is where two selected
component connects.

```

```

// Go one direction and add all the traversed nodes in order, and then go the
other direction and do the same
else if (selectedNode.neighbours.Count == 2)
{
    SfmKeyPointGraphNode previousNode = selectedNode;
    SfmKeyPointGraphNode currentNode = selectedNode.neighbours[0];
    bool SolidOfRevolution = true;

    List<SfmKeyPointGraphNode> branchKeyPointNodes = new
List<SfmKeyPointGraphNode>();

    //Dealing with Eccentric reducer when it is the entire part
if (currentNode.neighbours.Count == 1)
{
    foreach (object component in PipeSpoolManager.allComponents)
    {
        if (component is SfmReducerEccentric)
        {
            SfmReducerEccentric ReducerEccentric = component as
SfmReducerEccentric;
            if (SfmPoint.distance(currentNode.keyPoint,
ReducerEccentric.endPoint1) < PipeSpoolManager.minimumAcceptableLength ||
SfmPoint.distance(currentNode.keyPoint,
ReducerEccentric.endPoint2) < PipeSpoolManager.minimumAcceptableLength)
            {
                previousNode = selectedNode.neighbours[1];
                currentNode = selectedNode;
            }
        }
    }
}

while (currentNode.neighbours.Count > 1)
{
    if (currentNode.neighbours.Count > 2)
    {
        fullKeyPointNodes.AddRange(branchKeyPointNodes);
        SolidOfRevolution = false;
        break;
    }
    else
    {
        if (SfmPoint.distance(currentNode.neighbours[0].keyPoint,
previousNode.keyPoint) < PipeSpoolManager.minimumAcceptableLength)
        {
            double angle =
SfmPoint.angleFromVectors(previousNode.keyPoint - currentNode.keyPoint,
currentNode.keyPoint - currentNode.neighbours[1].keyPoint);
            previousNode = currentNode;
            currentNode = currentNode.neighbours[1];
            Console.WriteLine(angle);
            if (angle > 0.001)
            {
                fullKeyPointNodes.AddRange(branchKeyPointNodes);
                SolidOfRevolution = false;
                break;
            }
        }
    }
}

```

```

        else
        {
            double angle =
SfMPoint.angleFromVectors(previousNode.keyPoint - currentNode.keyPoint,
currentNode.keyPoint - currentNode.neighbours[0].keyPoint);
            previousNode = currentNode;
            currentNode = currentNode.neighbours[0];
            Console.WriteLine(angle);
            if (angle > 0.001)
            {
                fullKeyPointNodes.AddRange(branchKeyPointNodes);
                SolidOfRevolution = false;
                break;
            }
        }
        branchKeyPointNodes.Add(currentNode);
    }
}

if (SolidOfRevolution)
{
    return new List<SfMKeyPointGraphNode> { selectedNode };
}

previousNode = selectedNode;
currentNode = selectedNode.neighbours[1];

SolidOfRevolution = true;

branchKeyPointNodes = new List<SfMKeyPointGraphNode>();

//Dealing with Eccentric reducer when it is the entire part
if (currentNode.neighbours.Count == 1)
{
    foreach (object component in PipeSpoolManager.allComponents)
    {
        if (component is SfMReducerEccentric)
        {
            SfMReducerEccentric ReducerEccentric = component as
SfMReducerEccentric;
            if (SfMPoint.distance(currentNode.keyPoint,
ReducerEccentric.endPoint1) < PipeSpoolManager.minimumAcceptableLength ||
SfMPoint.distance(currentNode.keyPoint,
ReducerEccentric.endPoint2) < PipeSpoolManager.minimumAcceptableLength)
            {
                previousNode = selectedNode.neighbours[0];
                currentNode = selectedNode;
            }
        }
    }
}
while (currentNode.neighbours.Count > 1)
{
    if (currentNode.neighbours.Count > 2)
    {
        fullKeyPointNodes.AddRange(branchKeyPointNodes);
        SolidOfRevolution = false;
    }
}

```

```

        return fullKeyPointNodes;
    }
    else
    {
        if (SfMPoint.distance(currentNode.neighbours[0].keyPoint,
previousNode.keyPoint) < PipeSpoolManager.minimumAcceptableLength)
        {
            double angle =
SfMPoint.angleFromVectors(previousNode.keyPoint - currentNode.keyPoint,
currentNode.keyPoint - currentNode.neighbours[1].keyPoint);
            previousNode = currentNode;
            currentNode = currentNode.neighbours[1];
            if (angle > 0.001)
            {
                fullKeyPointNodes.AddRange(branchKeyPointNodes);
                SolidOfRevolution = false;
                break;
            }
        }
        else
        {
            double angle =
SfMPoint.angleFromVectors(previousNode.keyPoint - currentNode.keyPoint,
currentNode.keyPoint - currentNode.neighbours[0].keyPoint);
            previousNode = currentNode;
            currentNode = currentNode.neighbours[0];
            if (angle > 0.001)
            {
                fullKeyPointNodes.AddRange(branchKeyPointNodes);
                SolidOfRevolution = false;
                break;
            }
        }
        branchKeyPointNodes.Add(currentNode);
    }
}
}
if (SolidOfRevolution)
{
    return new List<SfMKeyPointGraphNode> { selectedNode };
}
// If the selected node has 3 or more neighbours, the program can conclude
that the first component is not a solid of revolution
else
    return fullKeyPointNodes;

return fullKeyPointNodes;
}

/// <summary>
/// Getting the list of component that contains at least a key point in the
minimum required model using a nested loop.
/// </summary>
/// <param name="keyPoints">List of key point nodes in the minimum required
model</param>
/// <returns>List of components in the minimum required model</returns>
public static List<object>
getMinimumRequiredModelComponents(List<SfMKeyPointGraphNode> keyPointNodes)

```

```

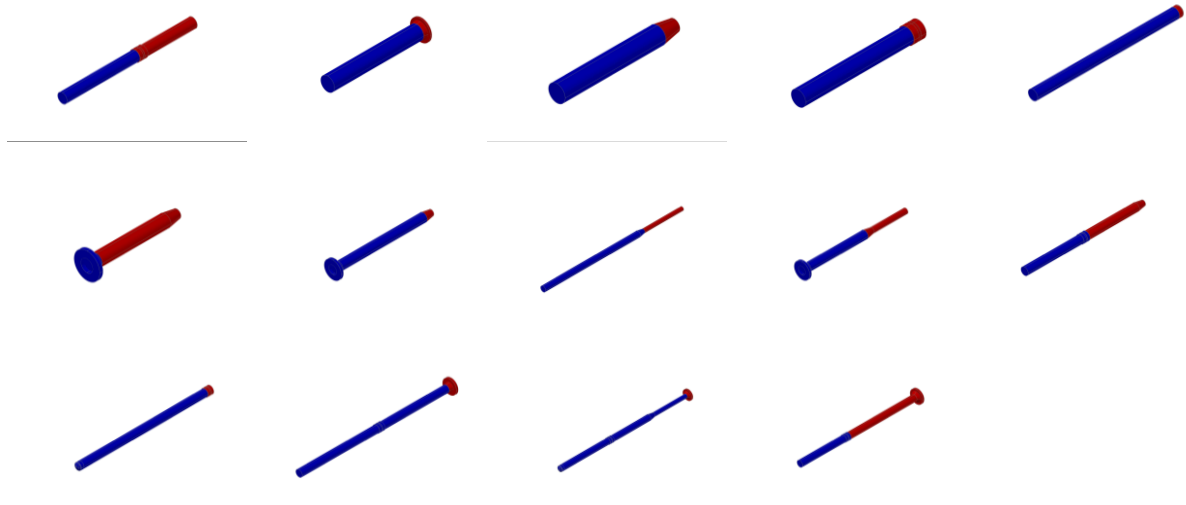
{
    List<object> components = new List<object>();
    foreach (object component in PipeSpoolManager.allComponents)
    {
        if (component is SfMCylindricalGeneric)
        {
            SfMCylindricalGeneric cylindricalGeneric = component as
SfMCylindricalGeneric;
            foreach (SfmKeyPointGraphNode keyPointNode in keyPointNodes)
                if (SfmPoint.distance(keyPointNode.keyPoint,
cylindricalGeneric.endPoint1) < PipeSpoolManager.minimumAcceptableLength ||
                    SfmPoint.distance(keyPointNode.keyPoint,
cylindricalGeneric.endPoint2) < PipeSpoolManager.minimumAcceptableLength)
                {
                    components.Add(component);
                    break;
                }
        }
        else if (component is SfMElbow)
        {
            SfMElbow elbow = component as SfMElbow;
            foreach (SfmKeyPointGraphNode keyPointNode in keyPointNodes)
                if (SfmPoint.distance(keyPointNode.keyPoint, elbow.endPoint1) <
PipeSpoolManager.minimumAcceptableLength ||
                    SfmPoint.distance(keyPointNode.keyPoint, elbow.endPoint2) <
PipeSpoolManager.minimumAcceptableLength ||
                    SfmPoint.distance(keyPointNode.keyPoint, elbow.pcfCentre) <
PipeSpoolManager.minimumAcceptableLength)
                {
                    components.Add(component);
                    break;
                }
        }
        else if (component is SfmFlange)
        {
            SfmFlange flange = component as SfmFlange;
            foreach (SfmKeyPointGraphNode keyPointNode in keyPointNodes)
                if (SfmPoint.distance(keyPointNode.keyPoint, flange.endPoint1) <
PipeSpoolManager.minimumAcceptableLength ||
                    SfmPoint.distance(keyPointNode.keyPoint, flange.endPoint2) <
PipeSpoolManager.minimumAcceptableLength)
                {
                    components.Add(component);
                    break;
                }
        }
        else if (component is SfmTee)
        {
            SfmTee tee = component as SfmTee;
            foreach (SfmKeyPointGraphNode keyPointNode in keyPointNodes)
                if (SfmPoint.distance(keyPointNode.keyPoint, tee.endPoint1) <
PipeSpoolManager.minimumAcceptableLength ||
                    SfmPoint.distance(keyPointNode.keyPoint, tee.endPoint2) <
PipeSpoolManager.minimumAcceptableLength ||
                    SfmPoint.distance(keyPointNode.keyPoint, tee.centrePoint) <
PipeSpoolManager.minimumAcceptableLength ||
                    SfmPoint.distance(keyPointNode.keyPoint, tee.branchPoint) <
PipeSpoolManager.minimumAcceptableLength)
        }
    }
}

```

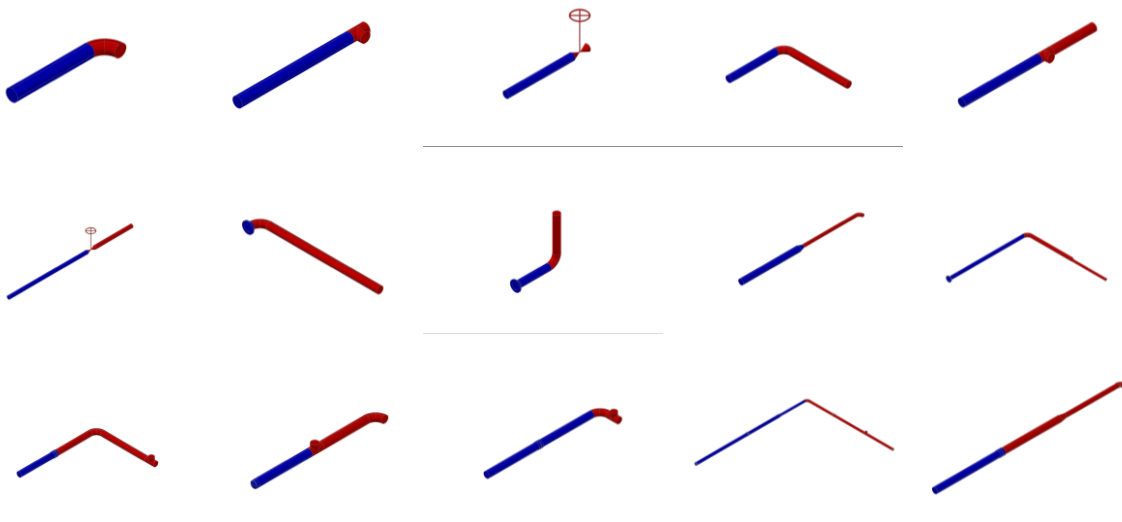
```
        {
            components.Add(component);
            break;
        }
    }
    return components;
}
```

Appendix C

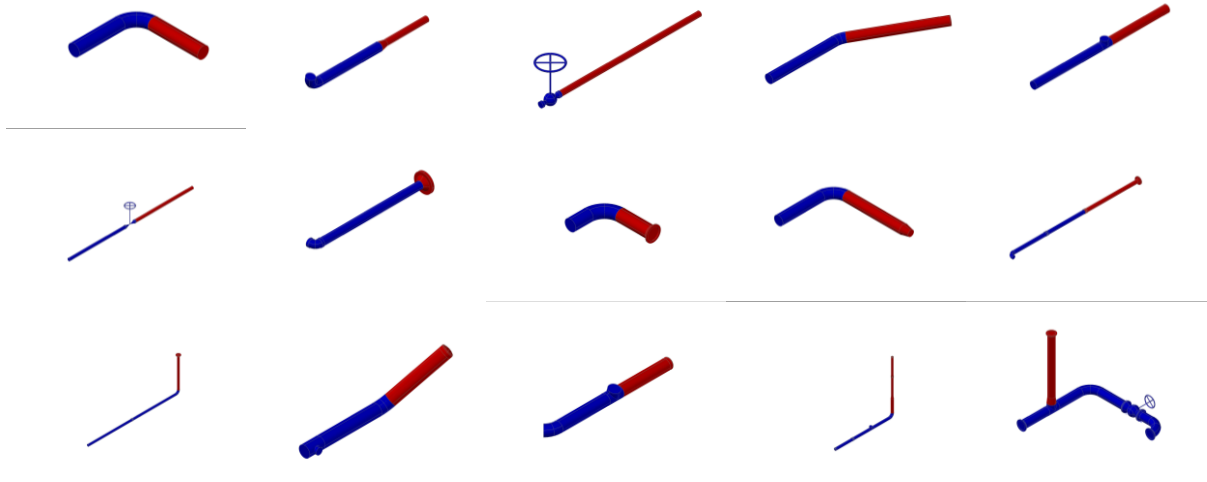
Class 1: 14 Cases



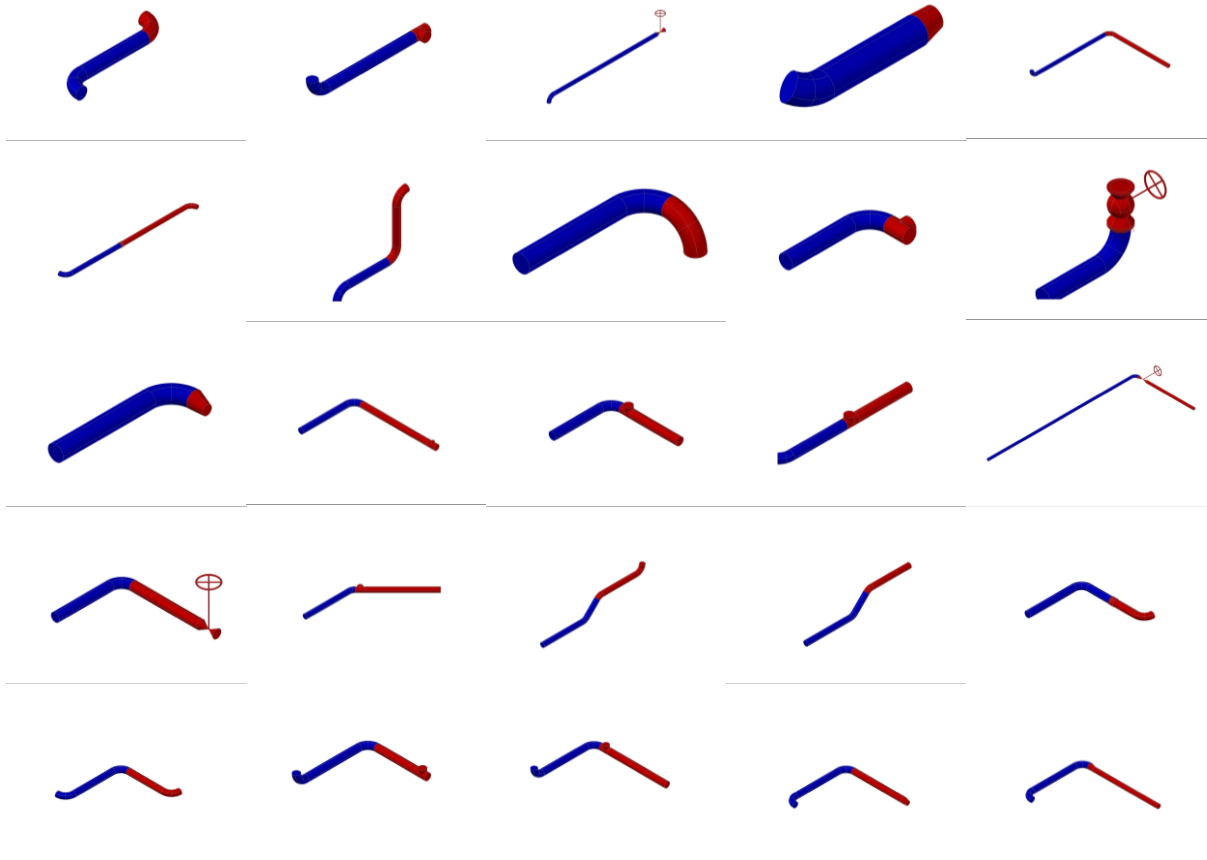
Class 2: 15 Cases



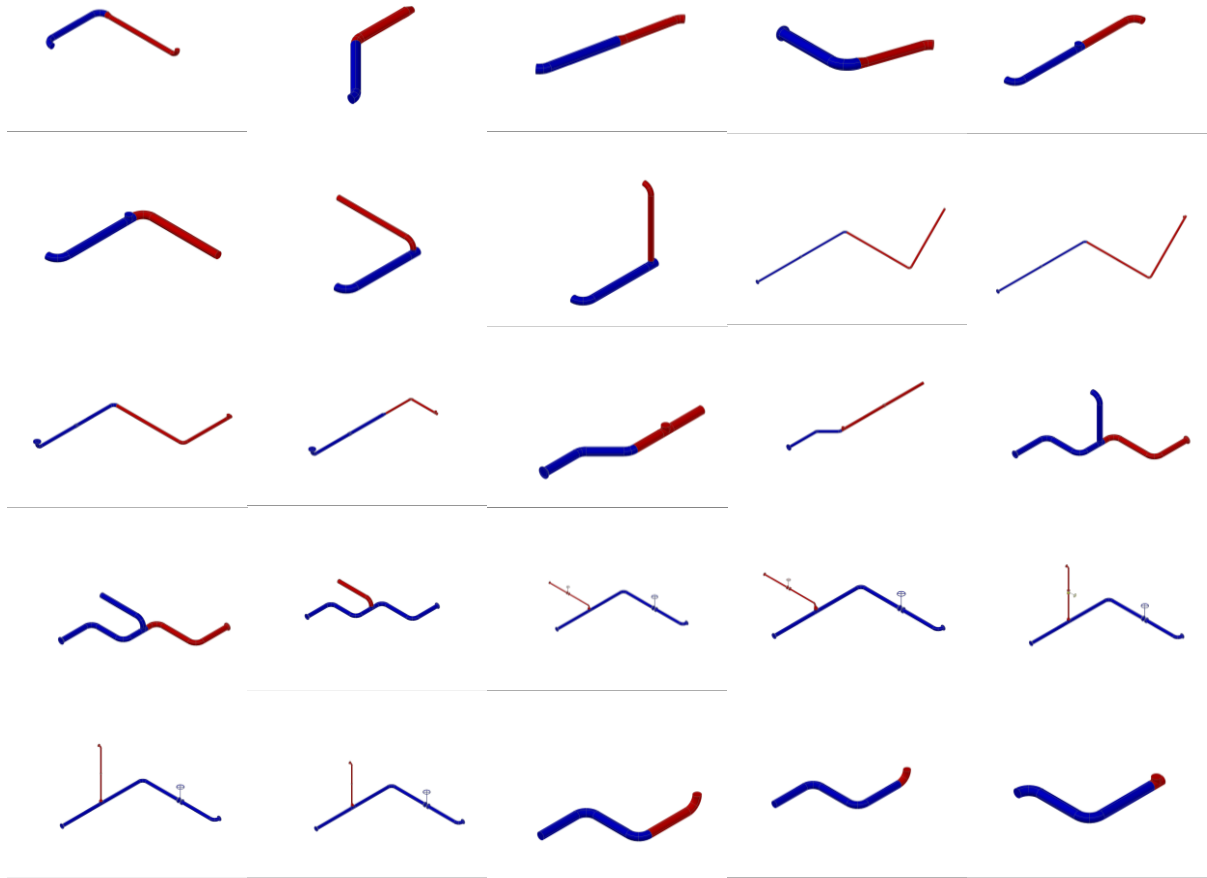
Class 3: 15 Cases





















Class 4: 50 Cases


















Class 4 continued



Appendix D

Case		Reference	Addition	Picture	Level Of Complexity		
		Assembly 5124-001-1:1			Initial Assembl	MRM	OC REDUCTIO
1	Stage 1	Pipe	Flange		2	2	0%
1	Stage 2	Flange + Pipe	Pipe		3	2	33%
3	Stage 3	Flange + Pipe + Pipe	Tee		5	3	40%
2	Stage 4	Flange + Pipe + Pipe + Tee	Pipe		6	3	50%
2	Stage 5	Flange + Pipe + Pipe + Tee + Pipe	Union		7	2	71%
2	Stage 6	Flange + Pipe + Pipe + Tee + Pipe + Union	Pipe		8	2	75%
3	Stage 7	Elbow	Flange		3	3	0%
2	Stage 8	Flange + Elbow	Pipe		4	3	25%
4	Stage 9	Flange + Elbow + Pipe	Elbow		6	5	17%
4	Stage 10	Flange + Pipe + Pipe + Tee + Pipe + Union + Pipe	Elbow + Pipe + Elbow + Flange		14	7	50%
1	Stage 11	Pipe	Flange		2	2	0%
1	Stage 12	Flange + Pipe	Pipe		3	2	33%
3	Stage 13	Flange + Pipe + Pipe	Elbow		5	3	40%
2	Stage 14	Flange + Pipe + Pipe + Elbow	Pipe		6	3	50%
1	Stage 15	Pipe	Reducer		2	2	0%
1	Stage 16	Pipe + Reducer	Pipe		3	2	33%
3	Stage 17	Pipe + Reducer + Pipe	Elbow		5	3	40%
3	Stage 18	Pipe	Elbow		3	3	0%

4	Stage 19	Elbow + Pipe + Reducer + Pipe	Elbow + Pipe		8	7	13%
4	Stage 20	Pipe + Elbow + Pipe + Reducer + Pipe + Elbow	Pipe + Elbow + Pipe + Pipe + Flange		14	5	64%
4	Stage 21	Pipe + Pipe + Tee + Pipe + Union + Pipe + Elbow + Pipe + Elbow + Pipe + Reducer + Pipe + Elbow + Pipe + Elbow + Pipe			27	5	81%
Assembly 5124-001-2:1							
1	Stage 22	Pipe	Flange		2	2	0%
3	Stage 23	Flange + Pipe	Tee		4	3	25%
2	Stage 24	Flange + Pipe + Tee	Pipe		5	3	40%
2	Stage 25	Flange + Pipe + Tee + Pipe	(Flange)		6	3	50%
1	Stage 26	Pipe	Flange		2	2	0%
1	Stage 27	Flange + Pipe	Pipe		3	2	33%
3	Stage 28	Flange + Pipe + Pipe	Elbow		5	3	40%
4	Stage 29	Flange + Pipe + Tee + Pipe (+ Flange)	Elbow + Pipe + Pipe + Flange		11	5	55%
Assembly 5124-001-3:1							
1	Stage 30	Pipe	Flange		2	2	0%
3	Stage 31	Flange + Pipe	Elbow		4	3	25%
3	Stage 32	Pipe	Elbow		3	3	0%
1	Stage 33	Pipe	Union		2	2	0%
1	Stage 34	Pipe + Union	Pipe		3	2	33%
3	Stage 35	Pipe + Union + Pipe	Valve		5	3	40%
2	Stage 36	Elbow	Flange		3	3	0%

3	Stage 37	Pipe	Elbow + Flange		4	3	25%					
4	Stage 38	Pipe + Union + Pipe + Valve	Pipe + Elbow + Flange		9	5	44%					
4	Stage 39	Flange + Pipe + Elbow	Pipe + Elbow		7	5	29%					
4	Stage 40	Flange + Elbow + Pipe + Valve + Pipe + Union + Pipe	Elbow + Pipe + Elbow + Pipe + Flange		16	7	56%					
Assembly 5124-001-4:1												
1	Stage 41	Pipe	Flange		2	2	0%					
3	Stage 42	Flange + Pipe	Elbow		4	3	25%					
2	Stage 43	Elbow	Flange		3	3	0%					
4	Stage 44	Flange + Elbow	Elbow		5	4	20%					
3	Stage 45	Pipe	Elbow + Pipe + Flange		5	3	40%					
4	Stage 46	Flange + Pipe + Elbow + Pipe	Elbow + Elbow + Flange		10	5	50%					
				TOTAL LOC	261	150	43%					
13	Case 1	pipe	22					This average is a better way to estimate				
9	Case 2	elbow	12									
9	Case 2	tee	2									
13	Case 3	union	2									
11	Case 4	flange	13									
46	Total stages	valve	1									
			Total						52			
		13%	LOC reduction for case 1									
		40%	LOC reduction for case 2									
		26%	LOC reduction for case 3									
		44%	LOC reduction for case 4									