creative commons
COMMONS DEED

저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

공학석사 학위논문

Inspection of As-Built Piping System for Design Consistency using Point Clouds and CAD Model

점군 데이터와 CAD 모델을 이용한 배관 설계의 검사 및 정합성 검증

2015 년 2 월

서울대학교 대학원

조선해양공학과

Tharun Kumar Parepalli

Inspection of As-Built Piping System for Design Consistency using Point Clouds and CAD Model

지도 교수 김 태 완

이 논문을 공학석사 학위논문으로 제출함

2014 년 10 월

서울대학교 대학원

조선해양공학과

Tharun Kumar Parepalli

Tharun Kumar Parepalli 의

공학석사 학위논문을 인준함

2014 년 12 월

위 원 장 _____ **(인)**

부위원장 _____ **(인)**

위 원 _____ **(인)**

Abstract

Inspection of as-built piping system for design consistency using point clouds and CAD model

Parepalli Tharun Kumar
Department of Naval Architecture and
Ocean Engineering
The Graduate School
Seoul National University

With the increase of lifespan and complexity of plants, it has become necessary for the use of design models of plants for maintenance operations and requires design models to match the as-built models precisely. However, the designs do not reflect the built models to the precision required for maintenance operations. The consistency of design models with built models can be increased by capturing as-built models at a point where no further modification is required and correcting the design models to match the as-built models.

Terrestrial laser scanning is a promising technique to capture built environment because of its ability to capture millions of points within few seconds with accuracy in millimeters. But, manual reconstruction of as-built models from these enormous number of points (point cloud) is a time consuming process and involves high costs. Also,

there is no need of modification of entire CAD model, only that part of model which differs more than its tolerance from the designed is required. However, there has been little research in the comparison of CAD models and point clouds and none in case of CAD models and large complex point clouds. Therefore, My thesis will address this problem in case of piping system which accounts for the major portion of plants by inspecting as-built point cloud against design model by automating the process of finding whether the designed pipe is installed or not and if installed, as-built positions of pipes, flanges and elbows corresponding to their designed ones are found. This research can also be extended to find the position of tee junctions. The implementation of this research is expected to assist the generation of as-built models and thereby save time and costs incurred otherwise.

Installation of designed pipe was found with an accuracy of 81.65% and non-installation with 93.8%. Lengths of pipes were found with an average difference of 1.08cm and angles at bends with 0.74° compared to design values.

Keywords: As-built Piping, CAD Model, Laser Scanning, Point cloud

Student Number: 2012-23990

Contents

ABSTRACT ..... **i**

CONTENTS ..... **iii**

LIST OF TABLES ..... **vi**

LIST OF FIGURES ..... **viii**

1 INTRODUCTION **1**

1.1 MOTIVATION 1

1.2 RELATED RESEARCH 3

1.2.1 Recognition of Shapes without using Design Data 4

1.2.2 Recognition of Shapes using Design Data 5

1.3 SUMMARY OF THE RESEARCH 6

1.4 COMPOSITION OF THE THESIS 9

2 ANALYSIS OF THE INPUTS **10**

2.1 LASER SCANNING AND POINT CLOUD 10

2.1.1 Introduction 10

2.1.2 Scanning Measurement Set-up 14

2.1.3 Input Point Cloud 16

2.2 CAD MODEL 18

2.2.1 Designing of Piping in AVEVA Marine 18

2.2.2 Input CAD Model 21

3 ALGORITHM 23

3.1 PREPROCESSING 26

3.1.1 Registration of Point Cloud and CAD Model 26

3.1.2 Point Cloud Format Conversion 26

3.1.3 Cropping of Point Cloud by Block 27

3.2 FINDING CORRESPONDING PIPE 27

3.2.1 Finding Region of Interest of Pipe and Cropping Points 30

3.2.2 Dividing ROI into Partitions 36

3.2.3 Removing Points on Edges 37

3.2.4 Separating Partition into Clusters 41

3.2.5 Sorting of Clusters 42

3.2.6 Finding Cluster of Corresponding Pipe 43

3.3 FINDING FLANGES 51

3.3.1 Finding Tentative Pipe Points 55

3.3.2 Finding Tentative Non-pipe Points 56

3.3.3 Finding Shadowed Regions 56

3.3.4 Separating Non-pipe Points into Clusters 57

3.3.5 Finding Possible Flange Clusters 58

3.3.6 Comparison with Design Parameter 59

3.4 FINDING POSITIONS 61

3.4.1 Finding Position of Flange and End of Pipe at Flange 61

3.4.2 Finding Positions of Elbow, Tee and End of Pipe at Elbow 63

4 IMPLEMENTATION AND VERIFICATION OF ALGORITHM 66

4.1 IMPLEMENTATION 66
4.2 VERIFICATION METHODS AND TARGET PARTS OF THE PROGRAM 70
4.2.1 Finding Corresponding Pipe 77
4.2.2 Finding Positions of Flange, Elbow and Pipe 79
4.3 ANALYSIS OF THE PARAMETERS 81
4.4 RESULTS AND ANALYSIS OF THE TEST 83
4.4.1 Preprocessing 84
4.4.2 Finding Corresponding Pipe 86
4.4.3 Finding Positions of Flange, Elbow and Pipe 92
5 CONCLUSION AND FUTURE WORKS 99
5.1 CONCLUSION 99
5.2 FUTURE WORKS 101
BIBLIOGRAPHY 102

List of Tables

TABLE 2.1 PTX FORMAT 17
TABLE 2.2 PRIMITIVES OF AVEVA MARINE 20
TABLE 2.3 REPRESENTATION OF PRIMITIVES IN AVEVA MARINE 22
TABLE 3.1 PARAMETERS OF DEMONSTRATION PIPE 30
TABLE 3.2 TYPES OF FLANGES 52
TABLE 4.1 INPUT POINT CLOUD 74
TABLE 4.2 SOFTWARES USED FOR VERIFICATION 76
TABLE 4.3 TARGET PIPES FOR FINDING CORRESPONDING PIPE 78
TABLE 4.4 QUANTITY OF TARGET PARTS FOR FINDING CORRESPONDING PIPE 78
TABLE 4.5 TARGET PIPES FOR FINDING POSITION OF PIPE 80
TABLE 4.6 TARGET FLANGES FOR FINDING POSITION OF FLANGES 80
TABLE 4.7 TARGET ELBOWS FOR FINDING POSITION OF ELBOWS 80
TABLE 4.8 QUANTITY OF TARGET PARTS OF BLOCK 1 FOR FINDING POSITION 81
TABLE 4.9 ALL THE PARAMETERS OF ALGORITHM 82
TABLE 4.10 VALUES OF THE PARAMETERS 82
TABLE 4.11 ANALYSIS OF LENGTH OF PARTITION PARAMETER 83
TABLE 4.12 VALUES OF ALL PARAMETERS SET 84
TABLE 4.13 CORRESPONDING POINTS OF POINT CLOUD AND CAD MODEL 85
TABLE 4.14 INPUT POINT CLOUD BEFORE AND AFTER FORMAT CONVERSION 86
TABLE 4.15 NUMBER OF POINTS OF EACH BLOCK 86
TABLE 4.16 BLOCK 1 DESIGNED PIPE DETECTION 87
TABLE 4.17 BLOCK 1 DESIGNED PIPE FALSE DETECTION 87

TABLE 4.18 BLOCK 1 DIFFERENT PIPE DETECTION 87
TABLE 4.19 BLOCK 2 DESIGNED PIPE DETECTION 88
TABLE 4.20 BLOCK 2 DESIGNED PIPE FALSE DETECTION 88
TABLE 4.21 BLOCK 2 DIFFERENT PIPE DETECTION 88
TABLE 4.22 DESIGNED PIPE DETECTION ACCURACY 89
TABLE 4.23 ACCURACY OF FLANGES DETECTION OF BLOCK 1 92
TABLE 4.24 COMPARISON OF RESULTS OF PIPES WHOSE POSITIONS WERE FOUND 94
TABLE 4.25 COMPARISON OF RESULTS OF ELBOWS WHOSE POSITIONS WERE FOUND 95
TABLE 4.26 COMPARISON OF LENGTHS OF PIPES CORRECTLY FOUND WITH DESIGNED LENGTHS 96
TABLE 4.27 STATISTICS OF DIFFERENCES OF LENGTHS OF PIPES CORRECTLY FOUND AND THEIR DESIGNED LENGTHS 97
TABLE 4.28 COMPARISON OF ANGLES FOUND AT BENDS WITH DESIGNED ANGLES 97
TABLE 4.29 STATISTICS OF DIFFERENCES OF ANGLES FOUND AT ELBOWS AND THEIR DESIGN ANGLES 98

List of Figures

FIG. 1.1 CONCEPT OF MODELING OF AS-BUILT PIPING SYSTEM 6

FIG. 2.1 TERRESTRIAL LASER SCANNING APPLICATIONS. ADOPTED FROM [14]. 11

FIG. 2.2 LEICA SCANSTATION P20 LASER SCANNER. 12

FIG. 2.3 PRINCIPLE OF TERRESTRIAL LASER SCANNING. ADOPTED FROM [15] 12

FIG. 2.4 PHASE-SHIFT PRINCIPLE. ADOPTED FROM [14] 13

FIG. 2.5 PRINCIPLE OF A LASER TRIANGULATION SENSOR. ADOPTED FROM [14] 14

FIG. 2.6 NUMBER OF SCANS REQUIRED. ADOPTED FROM [17] 15

FIG. 2.7 SCAN AND REGISTERED LOCATIONS IN TOP VIEW OF DECIMATED TEST POINT CLOUD. 16

FIG. 2.8 PIPING DESIGN HIERARCHY IN AVEVA MARINE 18

FIG. 2.9 PIPE IN AVEVA MARINE 19

FIG. 2.10 REPRESENTATION OF THE PIPE IN AVEVA MARINE 19

FIG. 2.11 HIERARCHY IN RVT FORMAT 21

FIG. 2.12 HIERARCHY OF REPRESENTATION 11 22

FIG. 3.1 OVERVIEW OF ALGORITHM 23

FIG. 3.2 OVERVIEW OF FINDING CORRESPONDING PIPE 24

FIG. 3.3 OVERVIEW OF FINDING FLANGE 25

FIG. 3.4 BRIEF OVERVIEW OF FINDING CORRESPONDING PIPE 28

FIG. 3.5 OVERVIEW OF FINDING CORRESPONDING PIPE 29

FIG. 3.6 CAD MODEL OF DEMONSTRATION PIPE AND ITS BRANCH 30

FIG. 3.7 POINT CLOUD CONSISTING OF DEMONSTRATION BRANCH 30

FIG. 3.8 SIMPLE SCENARIO FOR INSTALLATION ERROR 32

FIG. 3.9 ROI OF THE SIMPLE SCENARIO 32
FIG. 3.10 ROI OF A PIPE 33
FIG. 3.11 MIN/MAX BOX ALLOWANCE IN TWO DIMENSIONS 35
FIG. 3.12 ROI POINTS OF DEMONSTRATION PIPE 36
FIG. 3.13 PARTITIONS OF DEMONSTRATION PIPE 37
FIG. 3.14 SCANNER ERRORS AT EDGES OF PLANE 38
FIG. 3.15 EDGE POINTS AND NON-EDGE POINTS 39
FIG. 3.16 EDGE POINTS REMOVAL 39
FIG. 3.17 POINT CLOUD DENSITY PARAMETERS OF SCANNER 40
FIG. 3.18 INPUT AND OUTPUT OF EDGE POINTS REMOVAL OF A PARTITION 40
FIG. 3.19 INPUT AND OUTPUT OF SEPARATION OF A PARTITION INTO CLUSTERS 42
FIG. 3.20 OVERVIEW OF FINDING CLUSTER OF CORRESPONDING PIPE 43
FIG. 3.21 DISTANCE OF DATA POINT FROM AXIS 45
FIG. 3.22 DATA POINTS CONTAINING LINE WITH NOISE AND ERRORS 47
FIG. 3.23 INLIERS AND OUTLIERS IN THE DATA POINTS 47
FIG. 3.24 CALCULATION OF CYLINDER PARAMETERS 48
FIG. 3.25 FLANGES OF SAME TYPE CONNECTED. ADOPTED FROM WIKIPEDIA 53
FIG. 3.26 OVERVIEW OF FINDING FLANGES 54
FIG. 3.27 DEMONSTRATION PIPE FOR FINDING FLANGES 54
FIG. 3.28 TENTATIVE PIPE AND NON-PIPE POINTS 55
FIG. 3.29 TENTATIVE PIPE POINTS OF DEMONSTRATION PIPE 56
FIG. 3.30 TENTATIVE NON-PIPE POINTS OF DEMONSTRATION PIPE 56
FIG. 3.31 SHADOWED REGIONS OF DEMONSTRATION PIPE. 57
FIG. 3.32 CLUSTERS OF NON-PIPE POINTS OF DEMONSTRATION PIPE 58

FIG. 3.33 POSSIBLE FLANGE CLUSTER OF DEMONSTRATION PIPE 58
FIG. 3.34 SELECTION OF REGION FOR COMPARISON 59
FIG. 3.35 POINTS FOR COMPARISON OF FLANGE 60
FIG. 3.36 COMPARISON OF FLANGE 60
FIG. 3.37 ENDS OF PIPE AND FLANGE 62
FIG. 3.38 OVERVIEW OF FINDING ENDS OF PIPE AND FLANGE 62
FIG. 3.39 DEMONSTRATION PIPELINE 63
FIG. 3.40 DEMONSTRATION PIPELINE AFTER FINDING AN END OF FLANGES, PIPES 64
FIG. 3.41 FINDING END POINT OF PIPE AT ELBOW AND TEE 65
FIG. 3.42 POSITIONS OF ELBOW, TEE AND PIPE 65
FIG. 4.1 FLOWCHART REPRESENTING IMPLEMENTATION OF ALGORITHM 67
FIG. 4.2 FLOWCHART REPRESENTING IMPLEMENTATION OF FINDING POSITION OF PIPE WITH AT LEAST ONE OF THE ENDS AS ELBOW 68
FIG. 4.3 FLOWCHART REPRESENTING FINDING POSITION OF PIPE WITH FLANGES ON BOTH ENDS 69
FIG. 4.4 SCAN LOCATIONS OF TEST SCENE IN TOP VIEW OF DECIMATED POINT CLOUD 71
FIG. 4.5 POINT CLOUD FROM LOCATION 1 71
FIG. 4.6 POINT CLOUD FROM LOCATION 2 72
FIG. 4.7 POINT CLOUD FROM LOCATION 3 72
FIG. 4.8 POINT CLOUD FROM LOCATION 4 73
FIG. 4.9 POINT CLOUD FROM LOCATION 5 73
FIG. 4.10 TOTAL POINT CLOUD OF SCENE. 74
FIG. 4.11 TEST BLOCK 1 75
FIG. 4.12 TEST BLOCK 2 75

FIG. 4.13 VERIFICATION OF FINDING CORRESPONDING PIPE 77

FIG. 4.14 POINT CLOUD AND CAD MODEL BEFORE REGISTRATION 85

FIG. 4.15 POINT CLOUD AND CAD MODEL AFTER REGISTRATION 85

FIG. 4.16 PIPES FOUND OF BOTH BLOCK 1 AND BLOCK 2 89

FIG. 4.17 OTHER COMPONENT RECOGNIZED AS DESIGNED PIPE 90

FIG. 4.18 SAME PIPE RECOGNIZED AS DIFFERENT PIPE 90

FIG. 4.19 PIPE RECOGNIZED AS NOT INSTALLED 91

FIG. 4.20 PIPE AND ELBOW COMBINATION RECOGNIZED AS PIPE 91

FIG. 4.21 POSITIONS OF PIPES, FLANGES AND ELBOWS OF BLOCK 1 93

FIG. 4.22 POSITIONS OF PIPES, FLANGES AND ELBOWS OF BLOCK 2 94

Chapter 1

1 Introduction

1.1 Motivation

According to [1], 3D CAD models of power plants, oil and gas refineries are used for simulation of maintenance operations, training and some of these require CAD models to match the actual built models precisely up to 1cm. Similarly, in case of offshore plants, as the complexity and life time increases, there is a need for the usage of CAD models for the effective management of plants during operations like repair and conversions. In case of ships, before the start of construction, the design involves three phases, basic, detail and production and through each phase, the design becomes more specific and sophisticated [2]. By the end of these phases, the design is not accurate and many revisions occur during construction [3]. These revisions should be reflected in design of all of those affected. But, in a complex project like ships, this does not happen typically [4]. Also, the required tolerance of CAD model compared to the built model is smaller than the required tolerance of construction compared to the design model. Therefore, inconsistency between the design model and construction more than the tolerance of design model compared to the built model arises.

In a ship, there are a variety of systems such as piping, electrical, HVAC (heating, ventilation, and air conditioning) etc. The current research deals with the piping system which plays an important role in the operations. Construction of piping involves fabrication of spools, assembly of spools to modules, installation of spools/modules at the site, on-site inspection and maintenance [5]. Due to the following non-exhaustive reasons, the final design of pipe spools differs from the as-built model slight to large in the amount of difference.

1) The design and construction of spools depends on the intended locations of the equipment, these spools/modules may connect to. However odds are that the equipment and thereby pipelines are not installed exactly as per design [6].
2 To allow for inaccuracies in position of equipment, few pipes are cut to lengths less than or more than the design values [6].

Therefore, there is a need to find these changes/errors and update the CAD model with that of as-built information depending on the change/error is necessary.

Popularity of laser scanning has been increasing because of its ability to capture as-built scenes in short time with accuracy in millimeters [7]. However, manual reconstruction of total 3D model from point cloud is very time consuming and involves high costs. Also, there is no need of modification of entire CAD model, only that part of model which differs more than its tolerance from the designed model is required. Hence, an algorithm is proposed to automate the process of finding whether the pipe designed is installed or not and if installed, the positions of pipes, flanges and elbows

are found in case of pipes with flange or elbow at the ends of pipes. The algorithm can also be extended to find pipes and tee junctions in case of pipes connecting to tee junction along with either of flange or elbow at ends. In the workflow of piping, the current research can be applied after the on-site inspection and maintenance of pipe spools. And modification of design of only those components who differ more than its tolerance from design can be carried out.

1.2 Related Research

Current problem of finding difference in positions of designed and installed pipes can be seen as the problem of finding cylinders in point cloud with measurement noise, error points and points belonging to other components. And finding relation between the design pipes and those cylindrical shapes in point cloud.

Random Sample Consensus (RANSAC) [8] and Hough transform are two commonly used techniques to deal with the problem of shape recognition in noisy data and outliers. The RANSAC approach essentially consists of two steps that are repeated in an iterative fashion. Firstly, candidate shape is built using randomly selected minimum number of points required to calculate parameters of the shape. Second step involves calculation of number of points (inliers) consistent with the model. Shape with most inliers is then chosen as the model in the original RANSAC. Without any improvement to the RANSAC, cost involved in finding the best shape can be very high. Hough transform involves mapping of data points to a discretized parameter

space of the shape. The parameters with most number of votes is then considered as the correct shape. Major drawback of this approach is its requirement of scanning whole parameter space of the shape.

Research related to current study can be divided into two categories. Firstly, the works related to recognition of shapes in point clouds without using any design data. The other ones are those works that used design information for recognition.

1.2.1 Recognition of Shapes without using Design Data

There have been many studies in the field of recognition of piping system (current research interest) without using the design data both automatic and semi-automatic. Few of them needs manual segmentation of points belonging to piping system to recognize each component, most of them deals with recognition of only cylindrical objects and to my knowledge, no study was carried on the recognition of flanges.

T. Rabbani [9] put forward an approach for recognition of cylinders in industrial installations. Segmentation of cylinders was performed by using smoothness constraint which finds smoothly connected areas in point clouds followed by application of sequential Hough transform to detect cylinders.

Kazuaki Kawashima, Satoshi Kanai and Hiroaki Date [10] proposed an automated normal-based region growing technique for segmentation of points belonging to pipe,

elbow and tee junction together in large complex point cloud. Following by recognition of pipe segments and finding parameters of pipes. Drawback of this approach is its sensitivity to the calculation of point normal, assumption of all cylindrical portions as pipes or part of elbows or of tee junctions, which resulted in high false recognition.

Joohyuk Lee, Hyojoo Son, Changmin Kim, Changwan Kim [11] were able to detect pipes, elbows and tee junctions of piping system using voronoi diagram. However, they manually segmented points belonging only to these components from total point cloud losing the generality and its applicability to real cases.

R. Schnabel, R. Wahl and R. Klein [12] were able to recognize planes, spheres, cylinders, cones and tori primitives. This is an improvisation of RANSAC algorithm aimed to reduce the number of trials required to detect the correct shapes and the time involved in calculation of inliers for each candidate. This was achieved by using the user input of minimum number of points required to support a primitive and probability of overlooking a primitive, and their novel sampling and scoring strategies.

1.2.2 Recognition of Shapes using Design Data

Aur´elien BEY, Rapha¨elle CHAINE et al. [1] proposed a method of reconstruction of consistent cylindrical parts of CAD models from point clouds using the design data. However, their implementation is limited to CAD model involving bends of either 90° or 0°. For every cylinder, using the design radius and nearest point to cylinder, parameters of the cylinder surface were calculated. In case of pipes connecting to similar

pipe with same axis which is the most common case, this method fails to find the exact location of pipe. Also, it cannot handle the case of existence of pipe different from designed.

Frédéric Bosché [13] proposed a method for an automated recognition of 3D CAD model objects and calculation of as-built dimensions for dimensional compliance control. It involves global registration of total point cloud and CAD model, recognition of object in point cloud corresponding to designed object and registration of each designed object with the corresponding object in point cloud. And as-built dimensions of corresponding object were calculated. However, they assumed that the shape dimensions are already complied with specified tolerance and verified only the pose tolerances of objects.

1.3 Summary of the Research

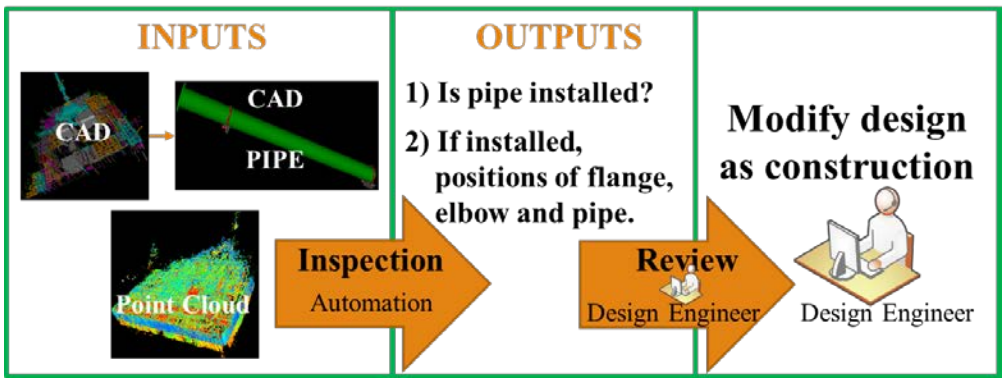


Fig. 1.1 Concept of Modeling of As-built Piping System

Fig. 1.1 shows the flow of processes in modeling of as-built piping system. Starting

with the current research - automatic inspection, there is review of outputs and modification of design. For each pipe in CAD model, the current research verifies whether the pipe is installed or not and if installed, axis is found in case of all the pipes. Whereas position of pipe is found in case of pipes connecting to either of flange or elbow on both ends by finding the components connecting pipe. In case of design pipes not installed, radius and axis of pipe existing in its location is found. The review of the outputs can be carried out by viewing the axes of pipes, pipe points, flange points and elbow points found along with the total point cloud and CAD model. And can judge if the modification in CAD model is required or not depending on the difference in design and built positions, and tolerance of the component. Also, the outputs can be used for manual modification of design as construction.

Automation of finding as-built positions was carried with the following research.

1) Analysis of the input data

Designing of piping specifically in AVEVA Marine (one of the mostly used shipbuilding 3D CAD) was studied. Following observations were made from the analysis of piping design and laser scanned point clouds.

1 Pipe spools are often flanged to facilitate easy connection to other spools and flanges of same type are used to connect two spools.

2 The difference in position of as-designed and as-built pipes do not vary greatly after total construction. However, due to non-updating of design, component in the design and construction may be different.

3 In case of large and complex scanned scenes, few pipes are not scanned

along total length due to shadowing by other objects.

4 To allow for inaccuracies in position of equipment, few pipes are cut to lengths less than or more than the designed.

5 Pipes extend into few components like in case of few types of flanges.

2) Proposition of Algorithm

Based on the above observations, it was assessed that, to find accurate position of pipe, components next to pipe has to be found. Hence, after finding designed flanges and elbows connecting pipes, the design data of pipe and the components connected were used to find accurate position of pipes. Flanges were found by finding the axis of pipe connecting and comparing the design radius of cylindrical portions of flange with the radius found. Elbows were found by finding the axes of pipes connecting them.

3) Implementation and application of program

A software program was developed based on this research and tested with industrial CAD data and its corresponding as-built point cloud data. The results of the test were verified manually by overlapping axes, pipe points, flange points and elbow points found with total decimated point cloud and CAD model, and also by comparing lengths of pipes found, angles found at bends with their design values.

This research addresses the as-built modelling of pipe spools by automating the process of finding the existence of designed pipe in construction, finding position of pipes connecting on both ends by either of flange or elbow by finding their positions. It is expected that with the application of this research, time and costs involved in

modeling of as-built piping system decreases.

1.4 Composition of the Thesis

This thesis contains five chapters and has been organized in the following format. In chapter 2, a brief introduction to laser scanning and input point cloud format is presented, and also discusses the design of piping in AVEVA Marine and its RVT format. Chapter 3 proposes an algorithm to find the existence of designed pipe in point cloud. And if designed pipe exists, the algorithm finds the axes of all pipes and position is found in case of pipes connecting flanges, elbows on both ends by finding them. Chapter 4 deals with the implementation of software program, analysis of the parameters involved, and verification and analysis of the test results. Finally, chapter 5 presents the conclusion and future works.

Chapter 2

2 Analysis of the Inputs

2.1 Laser Scanning and Point Cloud

2.1.1 Introduction

Laser scanning is a method where a surface is sampled or scanned using laser technology. It analyses a real world or object environment to collect data on its shape and possibly its appearance (e.g. color) [14].

Laser scanning technology can be divided into two categories, static and dynamic. In case of static, as the name suggests, the scanner is kept in a fixed position during the data acquisition. Whereas, in dynamic, the scanner is mounted on a mobile platform. These dynamic scanners require additional positioning systems. Airborne laser scanning is an example of dynamic scanning.

Terrestrial laser scanning is the ground based scanning. Because of its advantages like non-contact measurements, high accuracy, long range and fast data acquisitions, many disciplines like shown in Fig. 2.1 finds their applications in laser scanning.

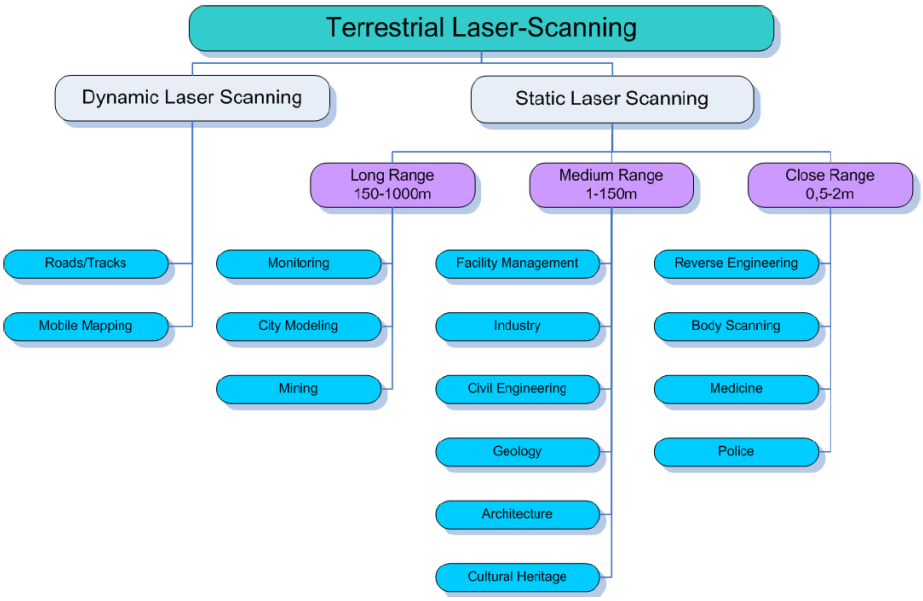


Fig. 2.1 Terrestrial Laser Scanning Applications. Adopted from [14].

Offshore plant facility management is the application the current research aimed at. An example of a medium range scanner is shown in Fig. 2.2. These scanners provide visualization of scene by recording horizontal and vertical angles, distance measurement from the hit point on surface with local coordinate system at the center of scanner as shown in Fig. 2.3, and also intensity of the reflected laser beam. These coordinates are converted to Cartesian coordinates. Thus, each point has X, Y, and Z coordinates and intensity values. In some cases, a camera is also attached to the scanner internally or externally and each point has extra color information represented as sum of Red (R), Green (G) and Blue (B) colors. Collection of all these points is called point cloud.

Fig. 2.2 Leica ScanStation P20 Laser Scanner.

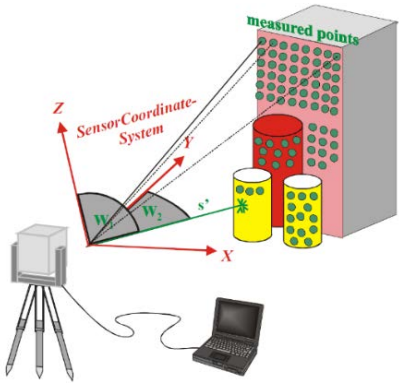


Fig. 2.3 Principle of Terrestrial Laser Scanning. Adopted from [15]

The measuring distance techniques can be divided into two, active and passive. Passive scanners do not emit any radiation, but depends on detecting reflected ambient radiation. Active scanners emit radiation and detects its reflection and measures distances using varied principles. The three major principles are Time of Flight, Phase Shift and Triangulation. In all these types, a laser pulse is emitted and a sensor detects a portion of the pulse that is reflected from a given surface.

1) **Time of Flight:** The distance to the surface is calculated by measuring the time

taken for the pulse to reach scanner after reflection from surface using equation (2.1). Long range laser scanning is typically performed by the scanners based on this technology. These scanners has the longest range and typically have slow acquisition rates. The accuracy depends on the system's ability to measure the time taken accurately.

$$D = \frac{(c.t)}{2} \tag{2.1}$$

c = velocity of light in air

t = time taken for the pulse to reflect

2) **Phase Shift:** Scanners using this technology measure the distance by measuring the phase difference between the emitted and reflected signals. As the wave-form is cyclic as shown in Fig. 2.4, an ambiguity problem exists if phase shift exceeds 2π. Thus, the effective range of these systems is limited and to over-come this, multiple frequency pulse is being used. These scanners are among the fastest laser scanners.

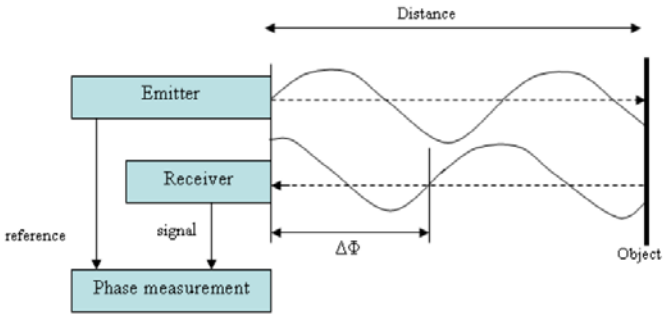


Fig. 2.4 Phase-shift Principle. Adopted from [14]

3) **Triangulation:** This technology is used when individual objects need to be

scanned with high level accuracy. Typical configuration of the setup is as shown in Fig. 2.5. In this setup, distance is measured using the known angle (α) between emitted laser beam and line connecting emitter and lens, distance (D) between emitter and lens, and by calculating angle (β) between reflected laser beam and line connecting emitter and lens.

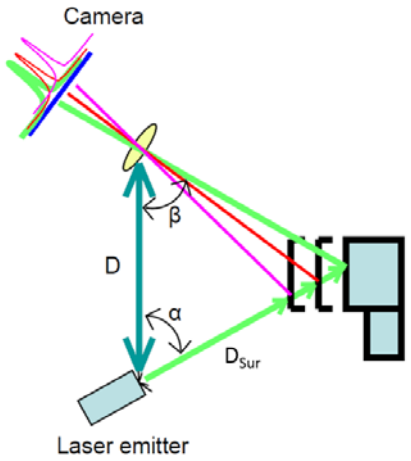


Fig. 2.5 Principle of a Laser Triangulation Sensor. Adopted from [14]

2.1.2 Scanning Measurement Set-up

Terrestrial laser scanners can measure the distance with precisions in the order of millimeters. According to [16], quality of individual point depends on four major factors: instrument mechanism, atmospheric conditions, object surface properties and scan geometry. Given a scanner and scene to be scanned, there is very little one can do to improve the first three factors. Scan geometry for a point depends on its position

and orientation of its surface relative to scanner position. The precision of each point decreases as the distance of point from the scanner increases and also decreases with increasing angle of incidence of laser beam on the surface [16]. Thus, by carefully choosing the scan locations in view of the specifications of scanner and geometry of scene, noise in the resultant point cloud can be minimized. A simple room of 400 X 400 m with small height and no occlusions, as shown in Fig. 2.6 requires 15 scans for a scanner of range 80m and maximum angle of incidence 70° [17].

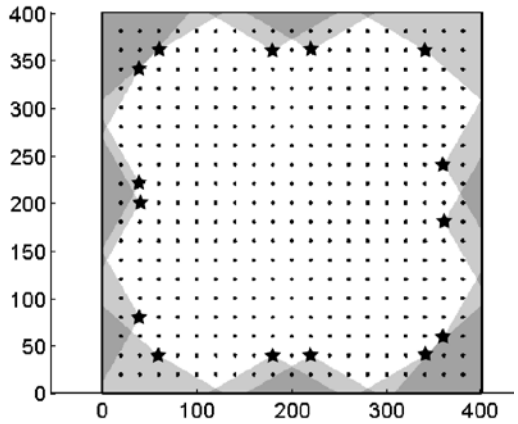


Fig. 2.6 Number of Scans Required. Adopted from [17]

Hence, few scene needs scans from more than one location and points in all these scans will be in their respective local coordinate systems. All these points must be transformed to a common coordinate system for further processing and this process is called registration. Scan locations and registered location in case of test point cloud in top view is shown in Fig. 2.7.

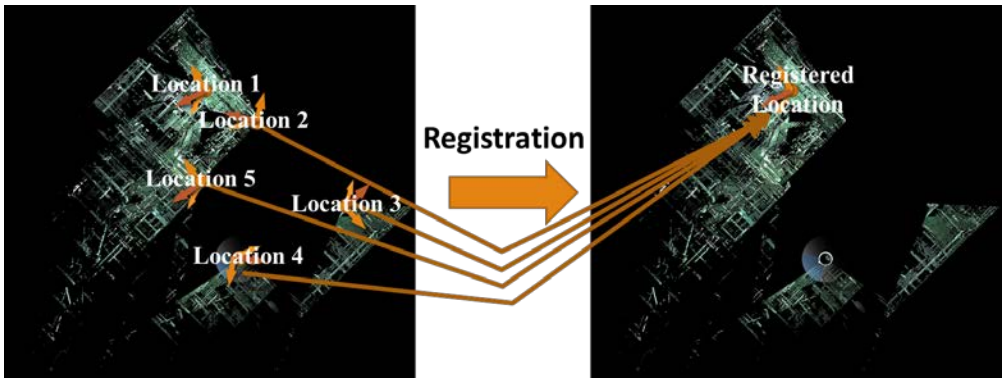


Fig. 2.7 Scan and Registered Locations in Top View of Decimated Test Point Cloud.

2.1.3 Input Point Cloud

Each point in the cloud has position (X, Y, Z), intensity of reflected laser beam (I) and color (R, G, B) information. The scanner used to scan the test scene is Leica Scanstation P20 shown in Fig. 2.1. Input point cloud is an ASCII based PTX formatted file, explained in detail in section 2.1.3.1.

2.1.3.1 PTX Format (Point Cloud Format of Leica Scanners)

Table 2.1 PTX Format

number of rows **(Cloud #1)**
number of columns (number of points = number of rows * number of columns)
st1 st2 st3 ; scanner registered position
sx1 sx2 sx3 ; scanner registered axis 'X'
sy1 sy2 sy3 ; scanner registered axis 'Y'
sz1 sz2 sz3 ; scanner registered axis 'Z'
r11 r12 r13 0 ; transformation matrix - rotation
r21 r22 r23 0 ; transformation matrix - rotation
r31 r32 r33 0 ; transformation matrix - rotation
tr1 tr2 tr3 1 ; transformation matrix - translation
X1, Y1, Z1, I1, R1, G1, B1 (Coordinates, Intensity and Color)
X2, Y2, Z2, I2, R2, G2, B2
..
number of rows **(Cloud #2)**
number of columns
..
..

Actual location of each point of a cloud is obtained by transforming points with transformation matrix of its cloud.

2.2 CAD Model

Most domestic shipyards use private 3-dimensional (3D) computer-aided design software for shipbuilding such as AVEVA'S AVEVA Marine or Intergraph's Smart Marine 3D. This research was carried out using AVEVA Marine. The design of Piping in AVEVA Marine which is the subject of interest is explained below in detail.

2.2.1 Designing of Piping in AVEVA Marine

The design hierarchy of piping in AVEVA Marine is as shown in Fig. 2.8. In this software, pipe that we have been referring is called tube and it is one of the components. Pipe in this software is a collection of branches, which in turn owns components, which define the route of total pipeline. In this section, when I refer pipe, I mean the pipe in this software. This pipe is referred as pipeline in other chapters. For pipe shown in Fig. 2.9, the design hierarchy looks as shown in Fig. 2.10 with tubes between the components automatically implied depending on the specifications of components.

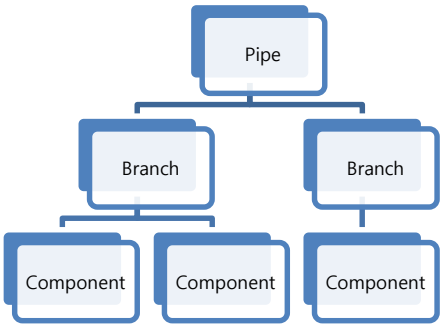


Fig. 2.8 Piping Design Hierarchy in AVEVA Marine

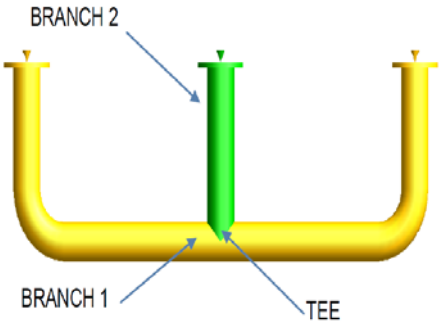


Fig. 2.9 Pipe in AVEVA Marine

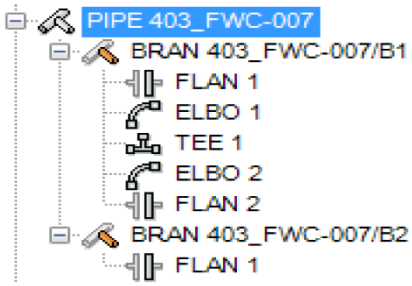
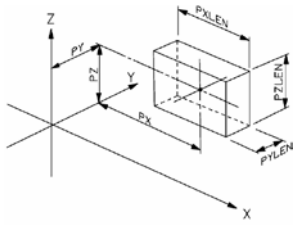


Fig. 2.10 Representation of the Pipe in AVEVA Marine

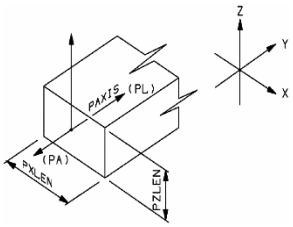
2.2.1.1 Designing of Piping Components

Piping components are modeled using primitives. There are 18 primitives in total shown below, a separate cylinder to represent tubes and line with two types of representations included. There are negatives of these primitives also to create holes.

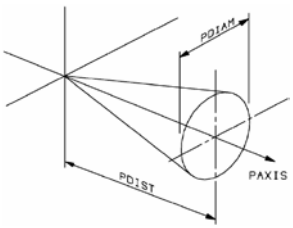
Table 2.2 Primitives of AVEVA Marine



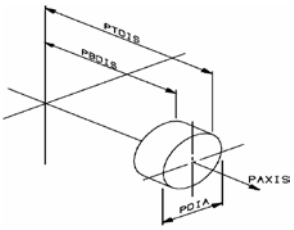
Rectangular Box 1



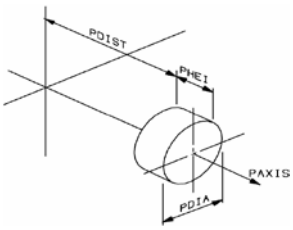
Rectangular Box 2



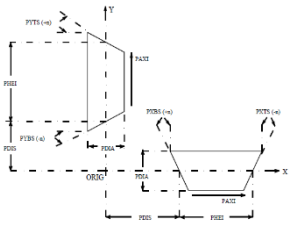
Cone



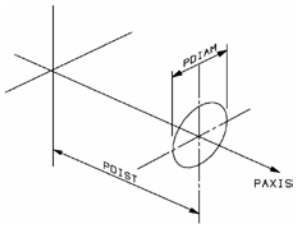
Cylinder 1



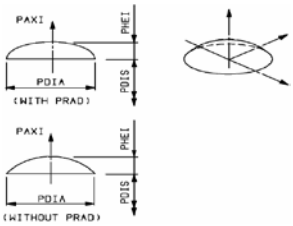
Cylinder 2



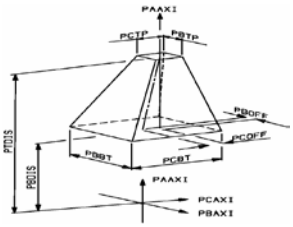
Slope Bottomed Cylinder



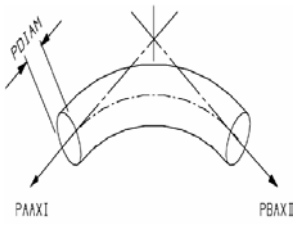
Disc



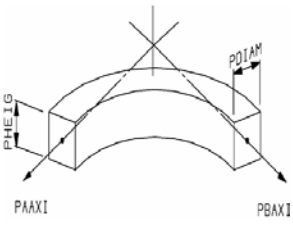
Dish



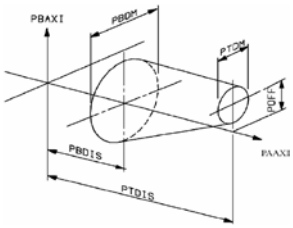
Pyramid



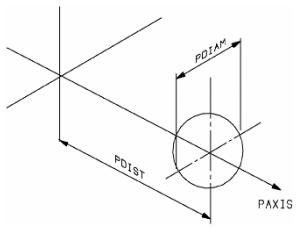
Circular Torus



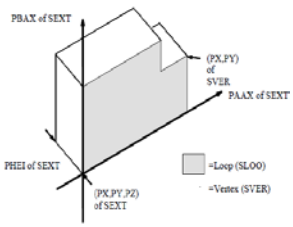
Rectangular Torus



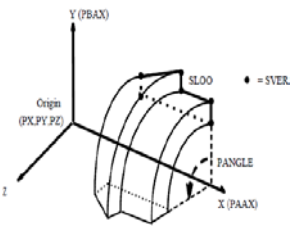
Snout



Sphere



Extrusion



Revolution

2.2.2 Input CAD Model

The input for design data to the algorithm is an RVT (ASCII format of Review model of AVEVA Marine) formatted file. Data in RVT format is hierarchical in nature. The hierarchy is same as design hierarchy. In case of pipeline design, the hierarchy is as below.

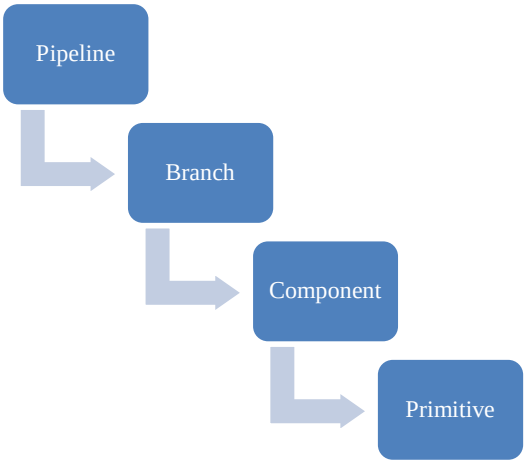


Fig. 2.11 Hierarchy in RVT Format

Each primitive is represented in 11 ways depending on the type of primitive as shown in Table 2.3. All primitives have primitive type, transformation matrix from primitive specific position at origin to actual position of primitive, bounding box data.

Table 2.3 Representation of primitives in AVEVA Marine

Representation 1	Pyramid
Representation 2	Box
Representation 3	Rectangular Torus
Representation 4	Circular Torus
Representation 5	Dish
Representation 6	Spherical cap
Representation 7	Cone, Sloped cylinder, Snout
Representation 8	Cylinder
Representation 9	Sphere
Representation 10	Line Segment
Representation 11	Extrusion, Revolution surface

In addition, in case of representations 1 to 10, there are shape parameters and in case of primitive 11, there is hierarchy in representation as shown in Fig. 2.12.

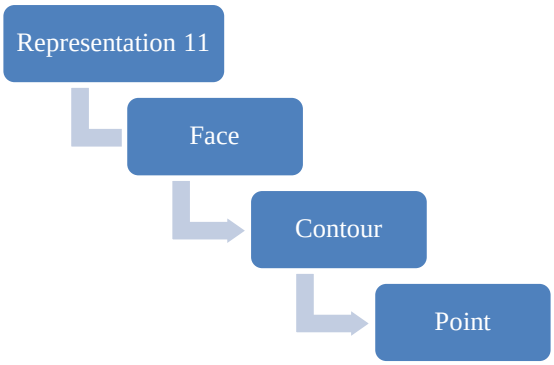


Fig. 2.12 Hierarchy of Representation 11

Chapter 3

3 Algorithm

This chapter explains all the steps involved in the algorithm starting with the pre-processing, finding corresponding pipe, finding flanges, and finding positions of flanges, elbows and pipes. Overview of the algorithm is shown in Fig. 3.1.

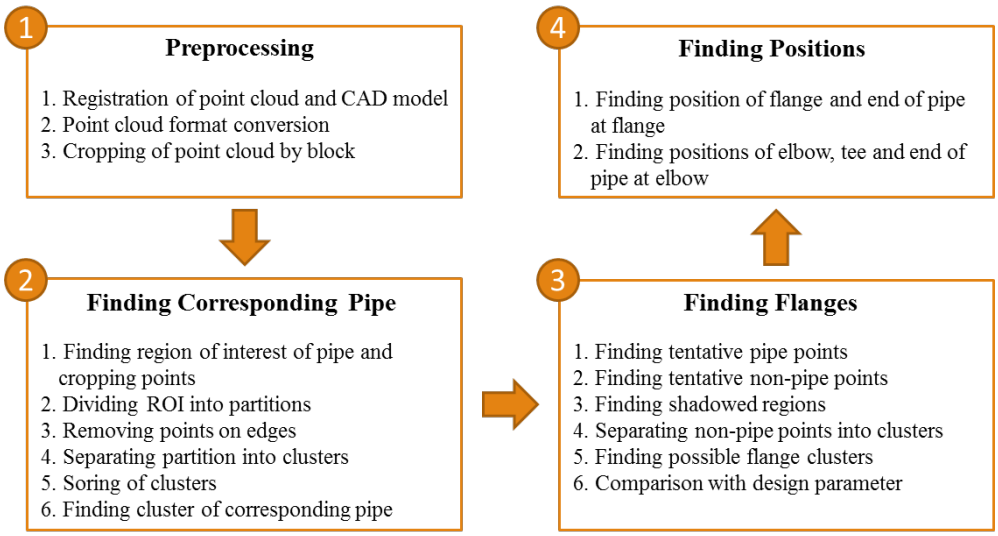


Fig. 3.1 Overview of Algorithm

First part contains all the preprocessing steps. Outputs of these steps are point clouds of each block registered with CAD model in desired format. In the second part, existence of designed pipe in point cloud is verified and if exists, installed axis is

found. If the designed pipe doesn't exist, radius and axis of pipe in the vicinity of designed pipe is found. Inputs and outputs of each step are shown in Fig. 3.2.

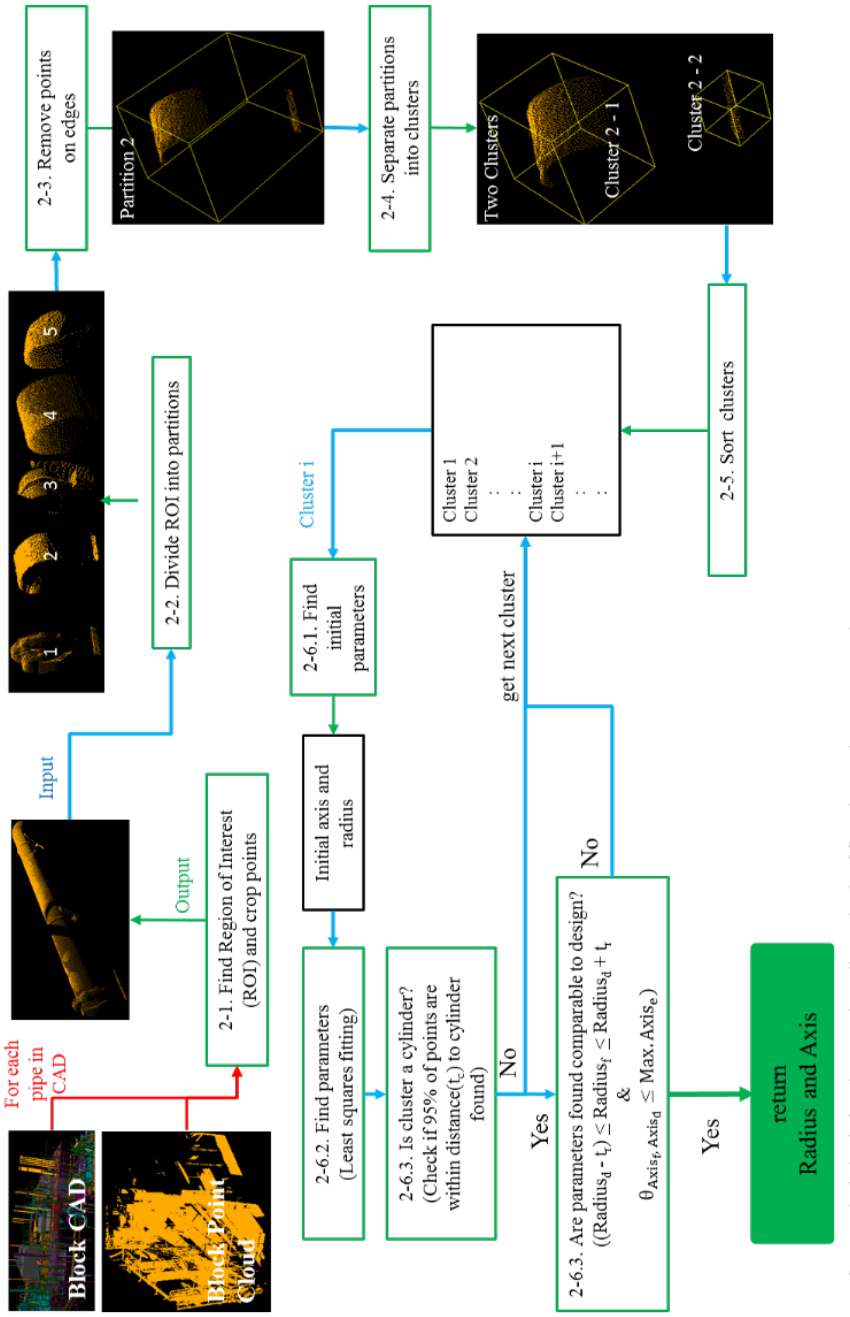


Fig. 3.2 Overview of Finding Corresponding Pipe

In the third part, with the inputs of points in ROI of pipe after removing points on edges, and radius and axis of found pipe, flange/s connecting pipe is/are found. All the steps in the process are shown in Fig. 3.3.

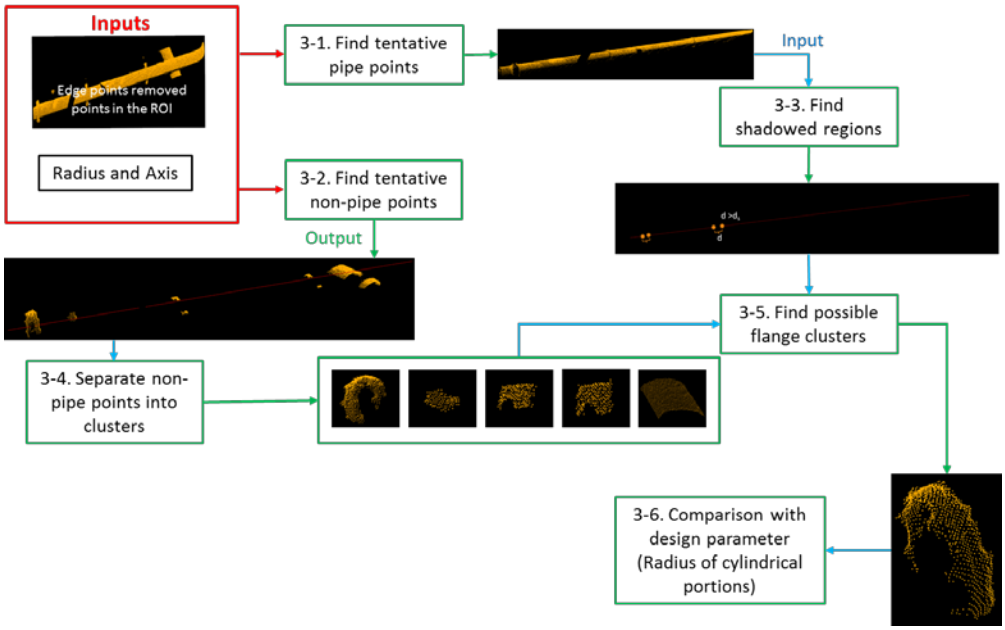


Fig. 3.3 Overview of finding flange

In the fourth part, positions of flange, elbow and pipe are found by using the radius and axis found of pipe, and design parameters of the components.

3.1 Preprocessing

This section refers to the first section in the overview shown in Fig. 3.1.

3.1.1 Registration of Point Cloud and CAD Model

As stated before, each scanning position has its own coordinate system. In case of a scene scanned from more than one location, all the scans are registered to put all the points in a coordinate system. This coordinate system may not match the coordinate system of the CAD model. Thus, to use the information of CAD effectively, it is necessary to match the coordinate systems of both the point cloud and CAD model. Three points in the point cloud that are constructed accurately as per the design are taken and their corresponding points are chosen from the CAD model. Using these three corresponding point pairs, transformation matrix of coordinate system of point cloud to match the coordinate system of CAD model is calculated.

3.1.2 Point Cloud Format Conversion

As mentioned in section 2.1.3, the input point cloud is an ASCII based file with position, intensity of reflected laser beam and color information for each point. However, the algorithm uses only position information. Also, due to existence of millions of points, handling of large size files on a normal PC becomes difficult. Hence, to save the memory, the ASCII file is converted to binary PCD (Point Cloud Data, the format

of Point Cloud Library) [18] format without the intensity and color.

3.1.3 Cropping of Point Cloud by Block

In this step, the total point cloud is cropped by each block. The simple reason for doing this is to reduce the search space of each design component. The inputs to this step are the total point cloud and RVT formatted file of piping system of each block. For each block, this is achieved by calculating the minimum and maximum values of X, Y and Z coordinates of all the primitives in piping system and using these min/max values, the points inside the min/max box are segmented with an allowance to all min and max of X, Y and Z values using [18]. The allowance depends on the accuracy of construction and will be discussed later in detail in section 3.2.1.2.

3.2 Finding Corresponding Pipe

This section refers to the second section in the overview shown in Fig. 3.1.

As mentioned in section 1.1, it is very possible that most of the designed pipes are not installed accurately, some are not at all installed and in some cases pipe different from design pipe exists. Thus, this part of algorithm finds the existence of designed pipe and the axis of pipe will be found if exists. If the designed pipe is not found, the axis and radius of a pipe present in ROI is found. Pipe that is found other than the designed could be

1. Pipe that is installed differently from designed.

2. Another designed pipe that exists in the ROI of pipe.

Ambiguity between the two of above can be cleared in manual review of results of automation. Brief overview of this part is shown in Fig. 3.4 and detail overview is shown in Fig. 3.2 and Fig. 3.5.

$Radius_f$ - Radius found
$Radius_d$ - Radius designed
$Axis_f$ - Axis found
$Axis_d$ - Axis designed
$Max.Axis_e$ - Maximum expected angle between design and built axis
$Axis_b$ - As-built axis

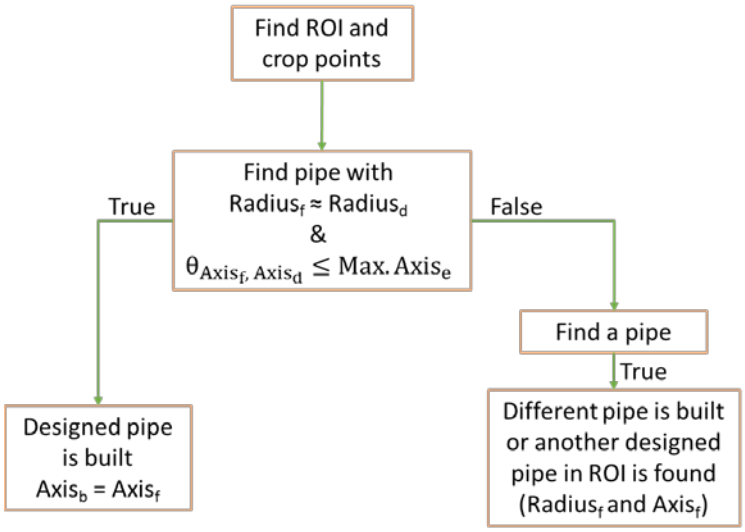


Fig. 3.4 Brief Overview of Finding Corresponding Pipe

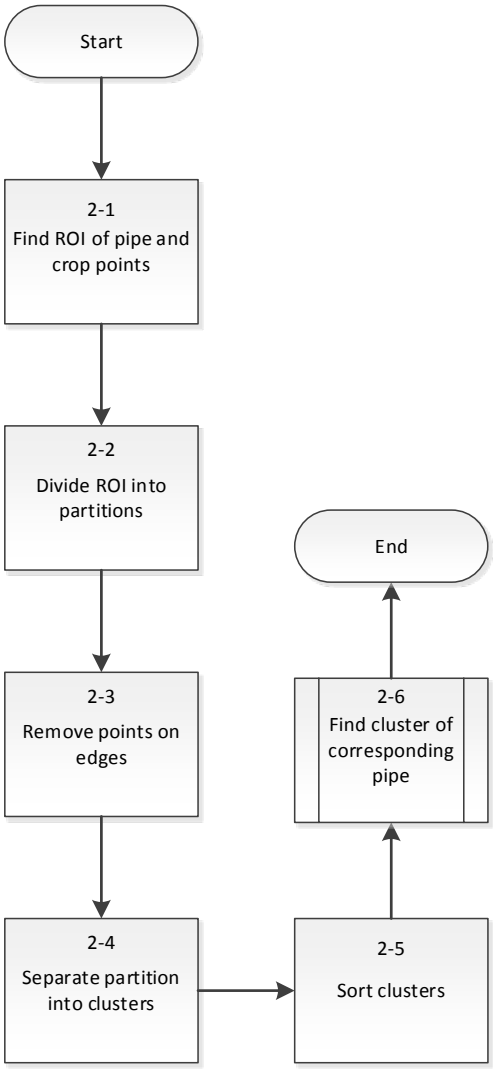


Fig. 3.5 Overview of Finding Corresponding Pipe

For each step of this part of algorithm, an output related to pipe highlighted (longest pipe in branch) in Fig. 3.6 will be presented for clear explanation. The parameters of the demonstration pipe are shown in Table 3.1.

Table 3.1 Parameters of Demonstration Pipe

	Radius	Length
Demonstration pipe	108.15mm	3.004m

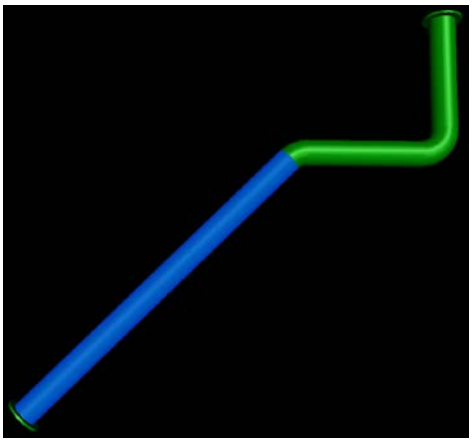


Fig. 3.6 CAD Model of Demonstration Pipe and its Branch

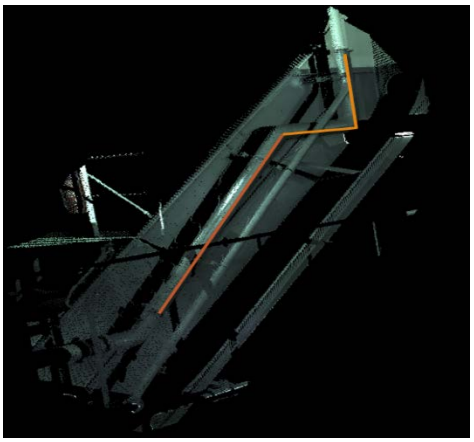


Fig. 3.7 Point Cloud Consisting of Demonstration Branch

3.2.1 Finding Region of Interest of Pipe and Cropping Points

This section refers to the step 2-1 in the overviews shown in Fig. 3.2 and Fig. 3.5.

3.2.1.1 Region of Interest of Pipe

As mentioned before, the premise of this research is that designed pipe and built pipe are differed only slightly in position as the installation is complete and no modification will be done. Hence, if a designed pipe is installed, it can be found in the

vicinity of designed pipe. This section deals with the finding of region in the point cloud, the pipe will be present, depending on the inaccuracies in installation and this region of a pipe is called Region of Interest (ROI) of the pipe.

Suppose, a pipe is to be installed such that the end points are at position 1 and position 2. Starting the installation at position 1, the error in the installation can occur at positioning of pipe at position 1 and also in the inclination of pipe that may or may not match the other end to position 2. Similar errors are possible if the installation is started at position 2. A simple illustration of the scenario in two dimensions with error considered in one direction is shown in Fig. 3.8 and its ROI is a rectangle as shown in Fig. 3.9. Thus, the ROI of a pipe is cuboid in shape as shown in Fig. 3.10, with pipe at center and allowances l_A in the direction of axis and $l_{A\perp}$ in any two mutually perpendicular directions and perpendicular to axis. These allowances depends on the accuracy of fabrication and installation.

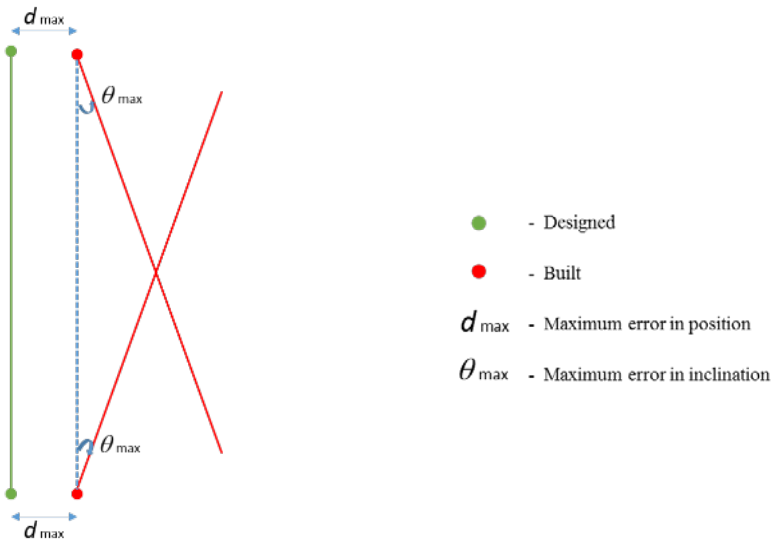


Fig. 3.8 Simple Scenario for Installation Error

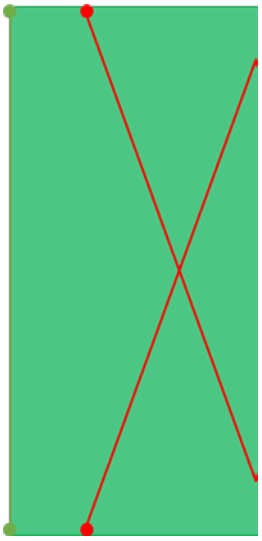


Fig. 3.9 ROI of the Simple Scenario

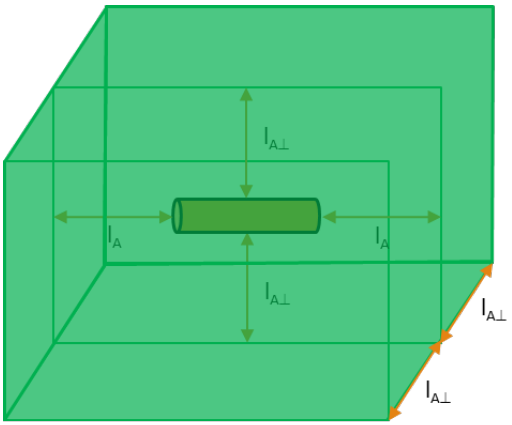


Fig. 3.10 ROI of a Pipe

3.2.1.2 Cropping Points in ROI of Pipe

This section contains the strategy for cropping of points particular to input CAD model type. As mentioned in section 2.1, the hierarchy of input CAD model is pipeline, branch and component and component is made of primitives. Pipeline can have many branches and a branch can have many pipes. Pipe is a component which has only one cylinder primitive. Few types of flanges and gaskets also have only one cylinder primitive but there lengths are very small. The goal of this step is to crop points in ROI of a pipe that is cuboid in shape as shown in Fig. 3.10. Cropping is the most time consuming part of algorithm. To reduce the time taken, pipeline points are cropped first, then branch points are cropped from pipeline points and then pipe points are cropped from branch points. Similar to cropping of points of a block, min/max values of pipeline are calculated and allowances L_A, $L_{A\perp}$ as in equation (3.2) are added for max

values and subtracted for min values, and points are cropped. The same allowances are used in case of block also. Similar is done in case of branches also. However for a pipe, allowance of only l_A, $l_{A\perp}$ along axis and axis perpendicular directions respectively are needed. So, pipe and branch points are transformed to a coordinate system with pipe axis as z-axis, in which case, allowances of l_A, $l_{A\perp}$ are just the same along axis and axis perpendicular directions. From Fig. 3.11, it can be seen that the inequality (3.1) holds. Hence, for cropping of points in cases of block, pipeline and branch, allowances L_A and $L_{A\perp}$ are taken as in equation (3.2).

$$L_A, L_{A\perp} < \sqrt{{l_A}^2 + l_{A\perp}^2} \tag{3.1}$$

l_A - Allowance along axis for ROI

$l_{A\perp}$ - Allowance perpendicular to axis for ROI

L_A - Allowance for min/max box of ROI along axis

$L_{A\perp}$ - Allowance for min/max box of ROI perpendicular to axis

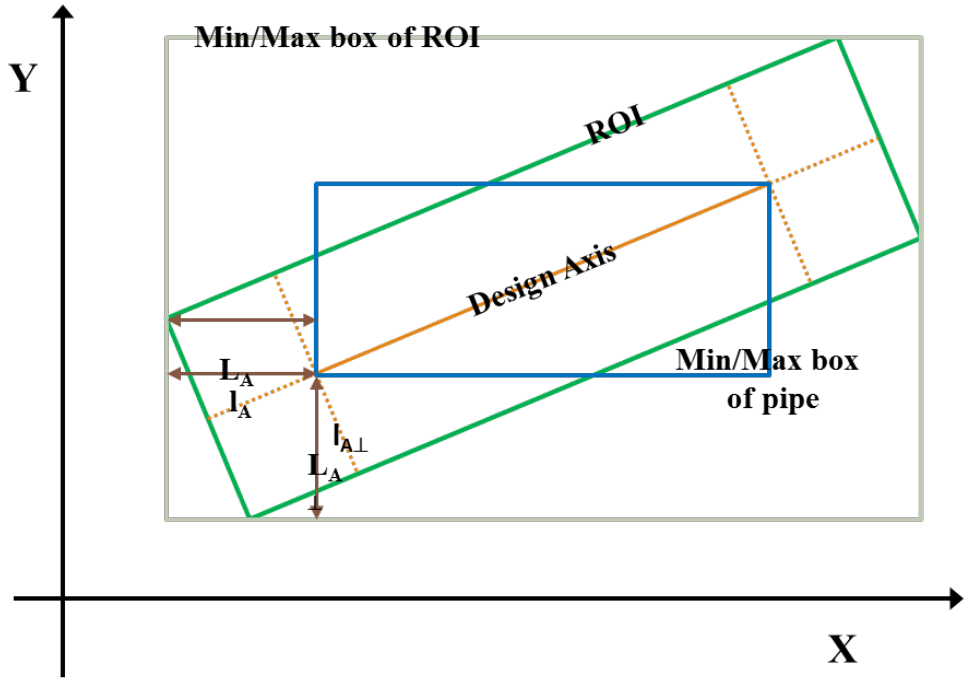


Fig. 3.11 Min/Max Box Allowance in two dimensions

$$L_A = L_{A\perp} = \sqrt{l_A{}^2 + l_{A\perp}^2} \tag{3.2}$$

The final output of this step for demonstration pipe is shown in Fig. 3.12.

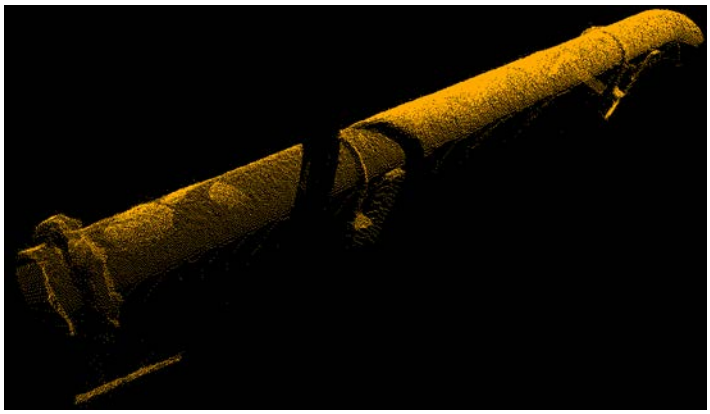


Fig. 3.12 ROI Points of Demonstration Pipe

3.2.2 Dividing ROI into Partitions

This section refers to the step 2-2 in the overviews shown in Fig. 3.2 and Fig. 3.5.

In this step, total ROI is divided into different partitions along length of designed pipe. The idea behind this step is, the total length of pipe is not needed to calculate the parameters and to find a group of points belonging only to the pipe (without points belonging to other components). The advantages of doing this are

1. The radius and axis of pipe can be easily found using the group of points belonging only to pipe.
2. Computational cost is reduced.

The length of each partition (lp) should be big enough to calculate the axis and radius accurately and small enough to be able to find at least one partition without points from other components.

Few partitions of demonstration pipe are shown in Fig. 3.13.

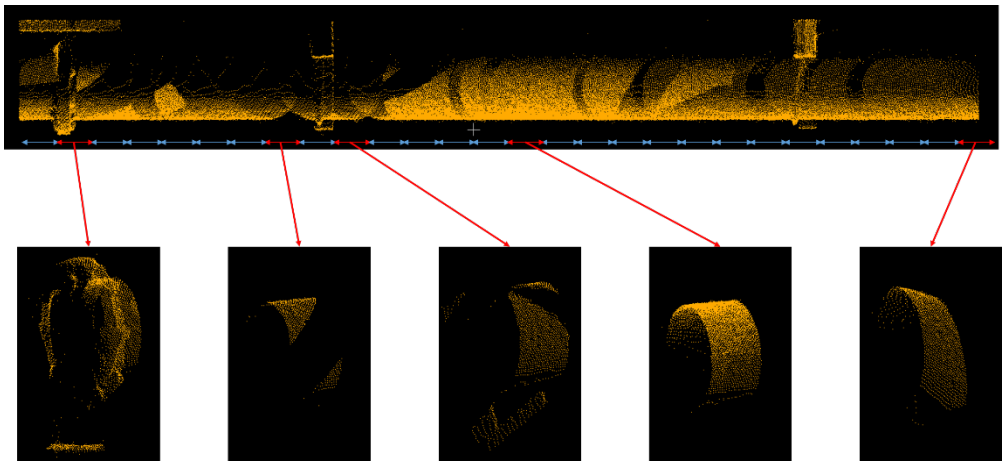


Fig. 3.13 Partitions of Demonstration Pipe

3.2.3 Removing Points on Edges

This section refers to the step 2-3 in the overviews shown in Fig. 3.2 and Fig. 3.5.

3.2.3.1 Mixed Pixel Effect

As mentioned in section 2.1.1, time of flight scanners emits laser beam and measures distance by calculating time interval between emitted and received beams. The laser beam has a finite cross section and at the edges of objects, the beam hits only a part on the edge and the other part hits other objects. The distance calculated in this scenario is the average of all the positions where the laser beam hits. Hence, this error is to be minimized for the accurate calculation of parameters of pipe and here, this is done by removing those points. Simple scenario of the case described above in case of a rectangular cross section scene is shown in Fig. 3.14. Perfect pixel

is the position where total cross section of beam is hit, the distance calculated here is accurate and mixed pixel is the one where only some portion of beam is hit. Distance calculated in this case is average of the distance to the position on the edge and other object where the other portion of beam is hit.

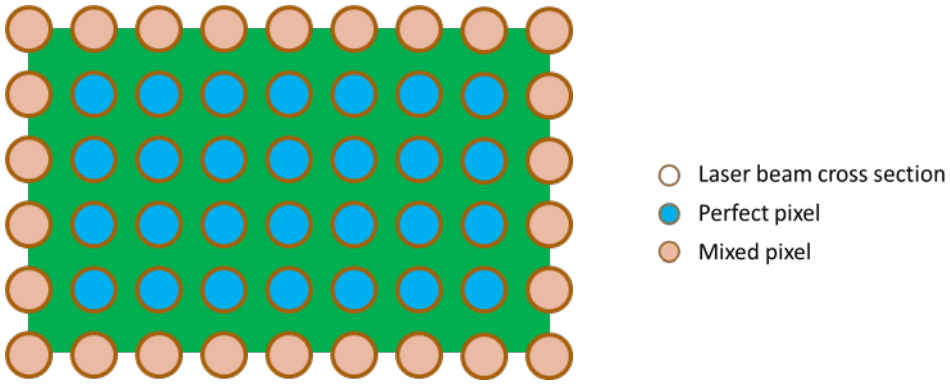


Fig. 3.14 Scanner Errors at Edges of Plane

In case of pipes, the surface is cylindrical and the mixed pixel effect will be more as the angle of incidence of laser beam with the surface normal increases.

3.2.3.2 Removing Edge Points

The basic idea of this step is that the point density on edges due to errors becomes less compared to the density of non-edge points. This can be visualized for a pipe in Fig. 3.15. These edge points are removed by removing those points with number of points less than a minimum number (n) within a given distance (d_e) using radius outlier removal algorithm implemented in Point Cloud Library [18].

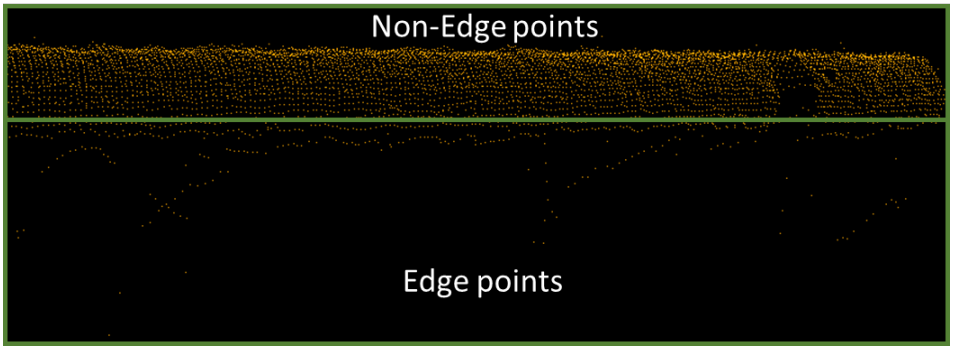


Fig. 3.15 Edge Points and Non-edge Points

Parameter n is set to 4, to remove those points that do not have adjacent points from both horizontal and vertical directions. A simple scenario of this process can be seen in Fig. 3.16. This particular step is carried out on all the partitions.

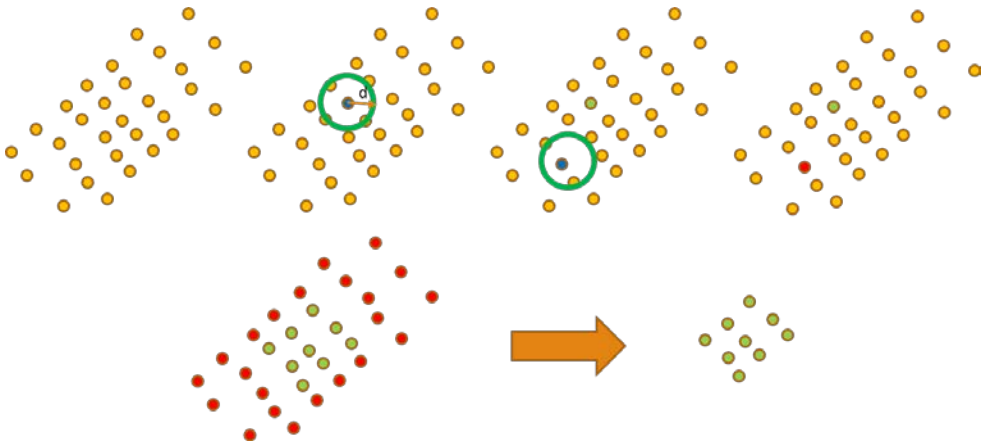


Fig. 3.16 Edge Points Removal

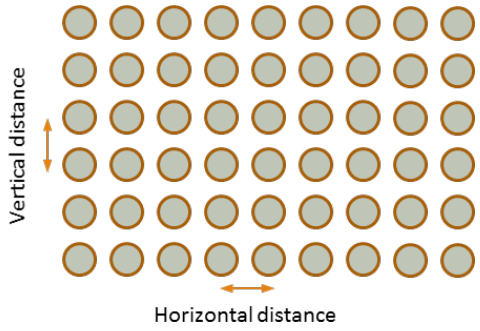


Fig. 3.17 Point Cloud Density Parameters of Scanner

Parameter d_e is chosen depending on the density of point cloud. Scanners can scan the scenes with different densities. The density is defined by setting the horizontal and vertical distance between the hit points shown in Fig. 3.17. These distances vary with the distance of hit points from the scanner and also with the shape of surface. The surface is cylindrical in case of pipes. The distance d_e should be set by examining the points on pipes with maximum distance from scanner position.

For one of the partition of the demonstration pipe, the output of this step is shown in Fig. 3.18 along with the input partition.

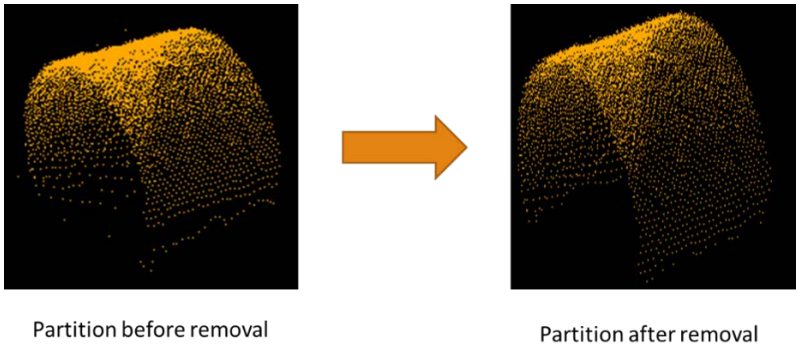


Fig. 3.18 Input and Output of Edge Points Removal of a Partition

3.2.4 Separating Partition into Clusters

This section refers to the step 2-4 in the overviews shown in Fig. 3.2 and Fig. 3.5.

The aim of this step is to divide points from different pipes in a partition into different groups. The reason for doing this is that in ROI of a pipe, there could be other pipes also. Thus, in a partition, there could be points from other pipes. If each pipe is separated from the other pipe and also other components, it becomes easy to find the parameters of each pipe. The separation is done using Euclidean Cluster Algorithm in PCL [18].

3.2.4.1 Euclidean Cluster Algorithm

The algorithm initializes a cluster, then a point is randomly chosen and distance to nearest point is checked. If this distance (d_c) is less than a threshold, both the points belong to same cluster and the process is repeated with the point added. This is repeated for all the points added to cluster till no more points can be added. Another point from the remaining points is chosen and the above process is repeated till a minimum number of points are left.

There could be small portions of same pipe, other pipe and also group of error points after dividing into clusters. These small groups could possibly be misrecognized as pipes and incase of groups belonging to actual pipe, the parameters found will not be accurate. Hence, these small groups should be removed and so, a minimum number of points in a cluster should also be set. If there are clusters with less than set minimum

number of points, the cluster will be removed.

Value of d_c depends on the least density of point cloud which is at maximum distance from scanner position. Minimum size of cluster was set to 250 which is a small number compared to high density of point clouds.

An input partition after removal of edge points and its output after separating into clusters are shown in Fig. 3.19.

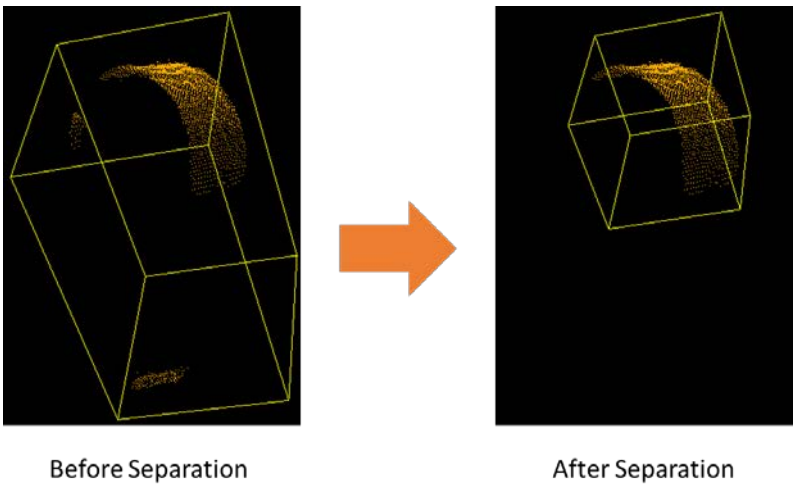


Fig. 3.19 Input and Output of Separation of a Partition into Clusters

3.2.5 Sorting of Clusters

This section refers to the step 2-5 in the overviews shown in Fig. 3.2 and Fig. 3.5

In this step, all the clusters from all partitions are sorted in the decreasing order of number of points in the cloud. The reason for doing this is, in the later steps, each cluster is analyzed if it's a cylinder and if it is, parameters are found. One more reason

is that, as mentioned before, there could be other pipes also and it is highly likely that only small portion of other pipes are inside ROI, as we assumed the difference in positions of as-designed and as-built is small. Also, more the number of points in a cluster, greater the accuracy of parameters found. Hence, the clusters are sorted to find the pipe we are searching for faster and to calculate the parameters accurately.

3.2.6 Finding Cluster of Corresponding Pipe

This section refers to the step 2-6 in the overviews shown in Fig. 3.2 and Fig. 3.5

The overview of this process can be seen in Fig. 3.20

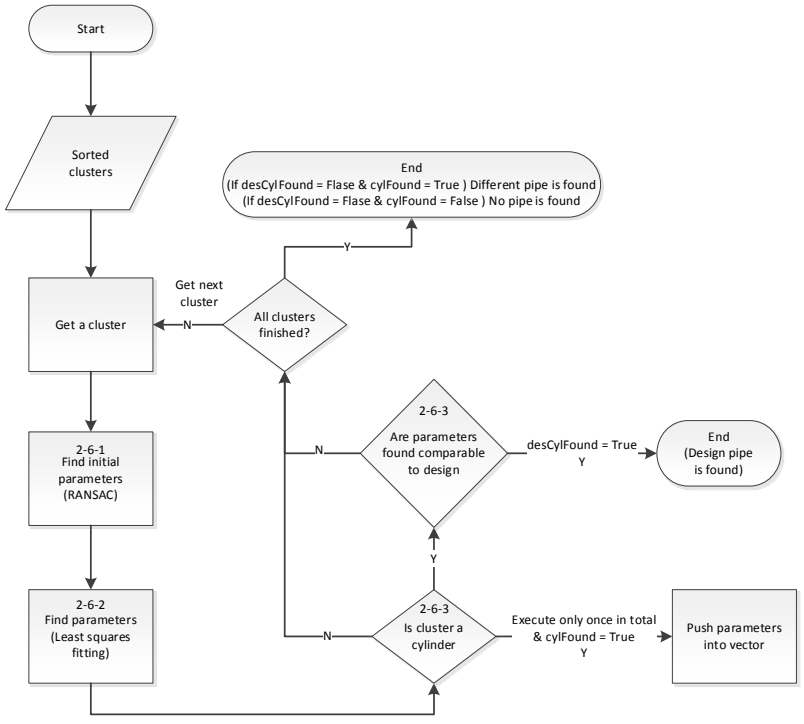


Fig. 3.20 Overview of Finding Cluster of Corresponding Pipe

3.2.6.1 Least Squares Fitting

This section refers to the step 2-6-2 in the overviews shown in Fig. 3.2 and Fig. 3.20. The step before is explained next to make sense of the approach.

This is the most used and simple technique to find parameters of known model (line, cylinder etc..) given a set of data points $\{(x_1, y_1), (x_2, y_2)..... (x_n, y_n)\}$ contaminated by noise but not the outliers. Hence, this technique finds the applicability in the current problem in case of cluster that contains points only from pipe. Here, we wish to find a function f(x), that best predicts the value y = f(x) for every given x. The procedure of finding the parameters of model by minimizing the sum of squares of residuals of points from the model is least squares fitting. In most of the cases, the residual is taken as $| f(x_i) - y_i |$.

There are different techniques for minimizing sum of squares. In the present work, Levenberg-Marquardt method, which is a combination of gradient descent method and Gauss-Newton method is used due to its rapid convergence. It improves the values of parameters iteratively starting from given initial parameters.

In case of pipes, the model is cylinder and parameters are axis and radius. Axis is represented by a point on axis and direction vector. In total, there are seven variables (a, b, c, α, β, γ, r). With (a, b, c) representing a point on the axis, (α, β, γ,) represents direction of axis and r, the radius of cylinder.

In case of cylinder, residual is the difference of distance (d) of data point (x, y, z) from the axis shown in Fig. 3.21 and radius (r).

$$\text{Residual} = \left| \left| \overrightarrow{(P_{D_I} - P_A)} \, X \, \overrightarrow{D_A} \right| - \, r \, \right| = |d_i - r|$$

$$d_i = \sqrt{m_i^2 + n_i^2 + \, o_i^2}$$

$$m_i = \, \gamma(y_i - b) - \, \beta(z_i - c)$$

$$n_i = \, \alpha(z_i - c) - \, \gamma(x_i - a)$$

$$O_i = \, \beta(x_i - a) - \, \alpha(y_i - b)$$

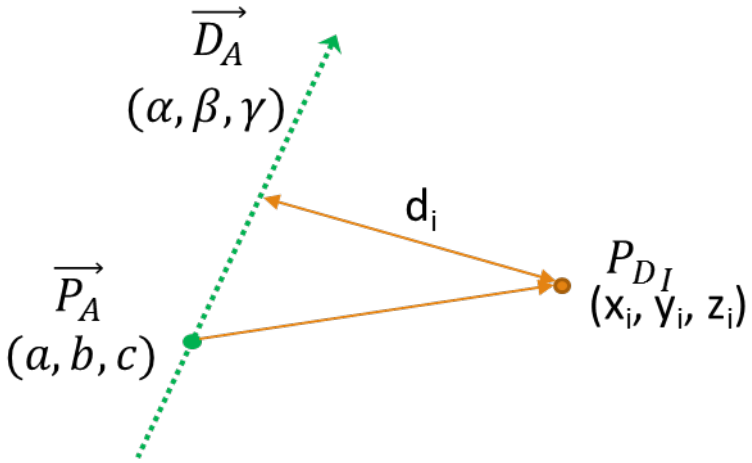


Fig. 3.21 Distance of Data Point from Axis

$$\text{Minimizing function} = \sum_{i=1}^{n} (d_i - r)^2$$

3.2.6.2 Finding Initial Parameters

This section refers to the step 2-6-1 in the overviews shown in Fig. 3.2 and Fig. 3.20.

Random Sample Consensus (RANSAC) [8] is used to find the initial parameters. Given an input of data points, the algorithm consists of two steps

1. Random selection of a minimal sample set of points and calculation of parameters of the model using the set of points selected.
2. Number of input data points compatible with the model calculated with a certain threshold is calculated.

And these steps are repeated till

1. Model with minimum number of compatible points are found.
2. Maximum number of models computed.

This is the RANSAC first put forward by Fishler and Bolles [19]. And this is the RANSAC used here without any modifications. This method is very useful when there are outliers in the data. Outliers are the points that do not belong to the model and inliers are those points that belong to the model. An example of data points containing points belonging to a line with some noise and few error data points are shown in Fig. 3.22 and inliers and outliers in this example are indicated in Fig. 3.23.

Fig. 3.22 Data Points Containing Line with Noise and Errors

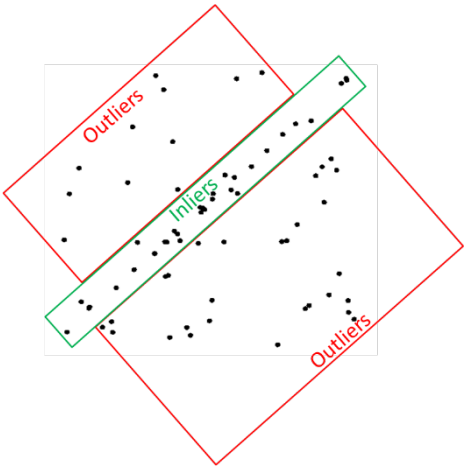


Fig. 3.23 Inliers and Outliers in the Data Points

Minimum number of data points with normals required to calculate cylinder model is two. Point P_1 with normal $\overrightarrow{n_1}$ and P_2 with normal $\overrightarrow{n_2}$ are shown in Fig. 3.24. Parameters, radius 'r' is calculated using expression (3.3) below and axis using expression (3.4)

$$\overrightarrow{(P_1 - r * \overrightarrow{n_1}) - (P_2 - r * \overrightarrow{n_2})} \;||\; (n_1 \; X \; n_2) \tag{3.3}$$

$$\overrightarrow{(P_1 - r * \overrightarrow{n_1}) - (P_2 - r * \overrightarrow{n_2})} \tag{3.4}$$

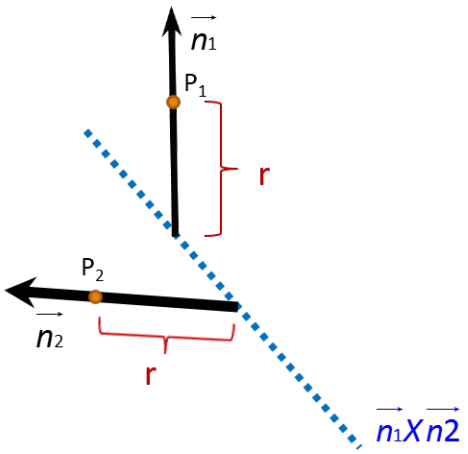


Fig. 3.24 Calculation of Cylinder Parameters

The normal of each point is estimated by fitting a plane to few neighboring points and assuming the normal of plane is the normal of point. These neighboring points can be set in two different methods

1. K nearest neighbors (KNN): In this method, for a point, K fixed number of points with least distances from point is chosen from cluster.
2. Fixed distance neighbors (FDN): For a given point, this method chooses all the points with in a fixed distance from the point.

As the number of points in KNN is fixed, the area of interest chosen increases with the decrease in point density and decreases with increase in point density. In a cloud, as the distance from scanner increases, the point density decreases and the noise in the measurement increases. Hence, KNN calculates better normals than FDN, where area of interest does not change with the point density which reflects measurement noise [9] and thus, KNN was used to find normals in this research.

As there are errors in data points and hence in normals estimation, the parameters calculated in all the cases may not be the actual parameters of model. Also, to find the best model, finding models and their compatible points for all the cases is a very time consuming process. Hence, this technique is used only to find initial parameters of the model.

The input to this step of algorithm are cluster of points and some of these clusters have all the points belonging to pipe and in some cases, only few and in some cases, none. As stated before, main idea of this algorithm is to find a cluster containing points belonging only to pipe (cylinder).

As the parameters found here are only used as initial parameters for least squares fitting which can calculate the parameters accurately, the number of compatible points can be set to a small number depending on the size of cluster (0.3 times the size of cluster is set in the current research). Threshold can be chosen relatively higher than scanner error to allow for errors in scanner measurement and normal calculations of points (1cm is set for an inaccuracy of 3mm in the current research). In case of clusters with only pipe points, number of iterations required will be very small as all the points belong to pipe. Hence, the time taken in case of clusters with other points can be greatly reduced by setting maximum number of models to be computed to less (2000 is set in the current research). To find the accurate radius and axis of perfect cluster, K should be high as there is measurement noise (150 is set in the current research).

3.2.6.3 Finding Corresponding Cluster

This section refers to the step 2-6-3 in the overviews shown in Fig. 3.2 and Fig. 3.20.

In this step, the cluster being processed is verified if it's actually a cylinder and if it is a cylinder, the cylinder parameters are compared with that of the design pipe, we are searching for. In the least squares fitting step, for every input cluster, a best fit cylinder is found even when the cluster is not a cylinder. Hence, this step is very important. To check if the cluster is a cylinder, cluster is checked if at least 95% of total points of cluster fall within a threshold (t_c) to the cylinder found.

And to check, if the cylinder found is of the pipe, we are looking for, radius of the cylinder found is checked if it falls within threshold (t_r) to design radius and axis within the maximum deviation expected (a_d). And these steps are repeated through all sorted clusters until a cluster with the given constraints is found. If no such cluster is found, the cluster with maximum points and is a cylinder is returned. This cylinder's radius and axis can be used during the manual modeling of point cloud.

Value of t_c should be carefully chosen depending on the accuracy of scanner, Least squares fit accuracy and smallest radius of pipe existing. If this is not carefully chosen, even the elbows may be regarded as pipes. Threshold (tr) , to verify if the pipe found is designed pipe, depends on the accuracy at which radius are found and also minimum difference of radius of pipes existing in ROI. Maximum deviation expected (a_d) should

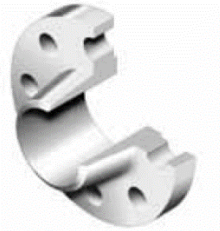
be small enough to avoid finding pipes connecting to the pipe through routing components, as in most of the cases, they have similar radius but different axis.

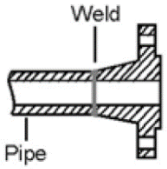
3.3 Finding Flanges

This section refers to the third section in the overview shown in Fig. 3.1.

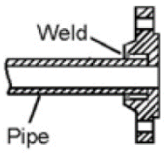
According to ASME B16.5, flanges that are used mostly are weld-neck, socket weld, slip-on, threaded, lap-joint and blind types [20] [21] shown in Table 3.2.

Table 3.2 Types of Flanges

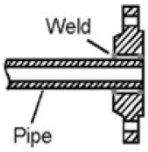




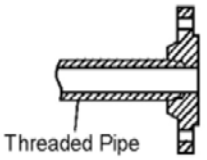
Weld-neck Flange



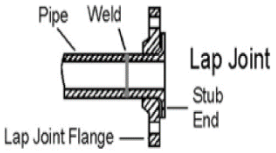
Socket Weld Flange



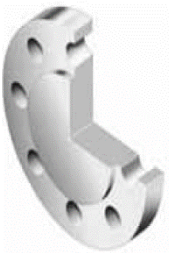
Slip-on Flange

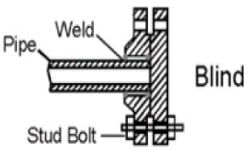


Threaded Flange



Lap Joint Flange





Blind Flange

Leaving out the blind flanges which are used to close the ends of piping. All other flanges when connected to flanges of same type have cylinders on either side of gasket, which is at center between them as shown in Fig. 3.25. This is the basis of finding flanges.

Fig. 3.25 Flanges of Same Type Connected. Adopted from Wikipedia

Overview of this part can be seen in Fig. 3.3 and Fig. 3.26. Inputs of this part of algorithm are radius and axis found in previous part of algorithm and points in ROI of pipe after removing points on the edges.

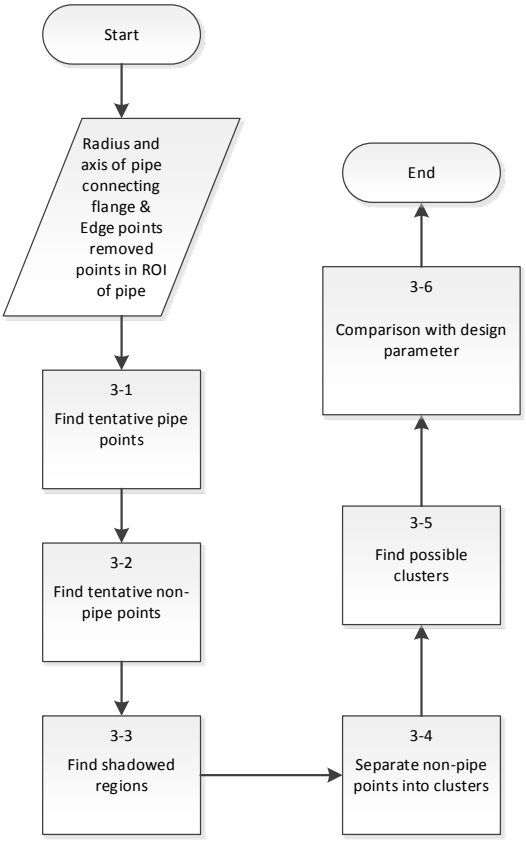


Fig. 3.26 Overview of Finding Flanges

A pipe of length 4.5m and radius 5.7cm will be used for demonstration of all the steps in this section from here. Points in its ROI after removing edge points are shown in Fig. 3.27

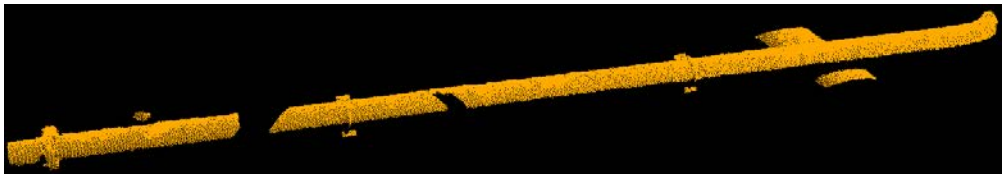


Fig. 3.27 Demonstration Pipe for Finding Flanges

3.3.1 Finding Tentative Pipe Points

This section refers to the step 3-1 in the overviews shown in Fig. 3.3 and Fig. 3.26.

In this step, all the points in ROI compatible with the parameters found are found and considered as tentative pipe points. As there can be little inaccuracies in the calculated radius and axis, to include all the points of the pipe as tentative pipe points, a threshold d_f is set in finding the compatibility. A simple scenario of this can be seen in Fig. 3.28, points in green region are compatible points and those in red regions are not compatible.

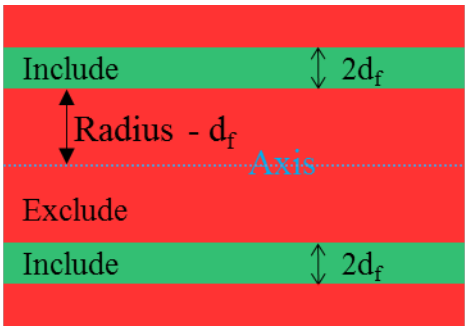


Fig. 3.28 Tentative Pipe and Non-pipe Points

Output of this step for demonstration pipe can be seen in Fig. 3.29. d_f must be chosen such that the points on cylindrical portions of flange are not included. Value of d_f = 2cm is set in the current research. As the difference of radii of pipe and the connected flange are generally much more than this value.

Fig. 3.29 Tentative Pipe Points of Demonstration Pipe

3.3.2 Finding Tentative Non-pipe Points

This section refers to the step 3-2 in the overviews shown in Fig. 3.3 and Fig. 3.26.

All remaining points in ROI are regarded as tentative non-pipe points.

Output of this step for demonstration pipe can be seen in Fig. 3.30. It can be observed that, these points include points from other pipes, supports of the pipe and flange connected.

Fig. 3.30 Tentative Non-pipe Points of Demonstration Pipe

3.3.3 Finding Shadowed Regions

This section refers to the step 3-3 in the overviews shown in Fig. 3.3 and Fig. 3.26.

In this step, tentative pipe points are projected onto the axis found and region between two of these consecutive projections is regarded as shadowed region if distance

between consecutive projected points is more than d_s.

Value of d_s for flanges is set to 15mm modestly, since length of flanges connected to other flange is much more than this value.

Output of this step for demonstration pipe is shown in Fig. 3.31. Two shadowed regions are found, one region is due to flange and the other due to shadowing by other objects.

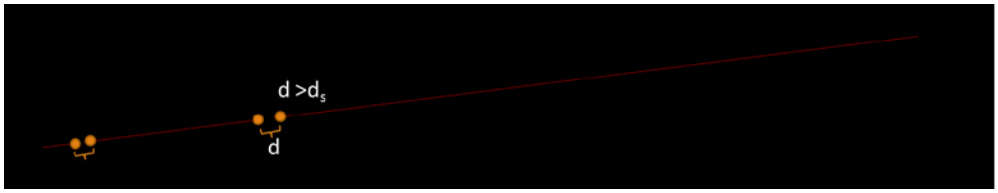


Fig. 3.31 Shadowed Regions of Demonstration Pipe.

3.3.4 Separating Non-pipe Points into Clusters

This section refers to the step 3-4 in the overviews shown in Fig. 3.3 and Fig. 3.26.

In this step, all the tentative non-pipe points are separated into different clusters similar to the separation of points in a partition in section 3.2.4.

Output of this step for demonstration pipe can be seen in Fig. 3.32. It can be seen that clusters include supports of pipe, other pipes and flange.

Fig. 3.32 Clusters of Non-pipe Points of Demonstration Pipe

3.3.5 Finding Possible Flange Clusters

This section refers to the step 3-5 in the overviews shown in Fig. 3.3 and Fig. 3.26.

This step involves finding of those clusters shadowing the pipe. This is achieved by finding the center of projections of points in cluster. And verifying if the projections lie in the shadowed regions found before. As the later step involves the verification of precise definition of flange, in this step, clusters whose centers of projections on axis lies in the shadowed regions are regarded as possible flanges.

Output of this step in case of demonstration pipe is just the flange cluster shown in Fig. 3.33.

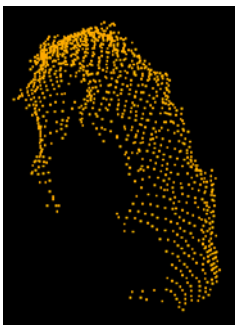


Fig. 3.33 Possible Flange Cluster of Demonstration Pipe

3.3.6 Comparison with Design Parameter

This section refers to the step 3-6 in the overviews shown in Fig. 3.3 and Fig. 3.26.

Here, points in cluster whose projection lies in either of regions of length a_f on axis leaving length a_g for gasket at center of shadowed region in which cluster's projection center lies are found as shown in Fig. 3.34.

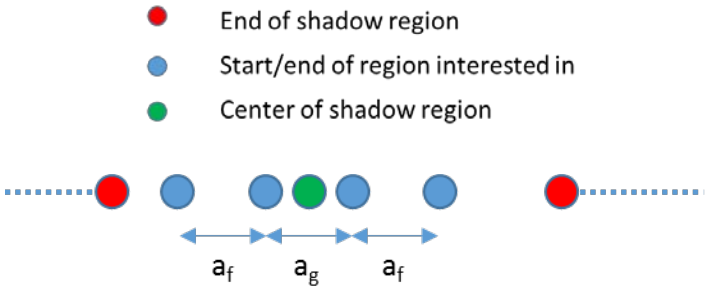


Fig. 3.34 Selection of Region for Comparison

Value of a_f is set to 1cm in the current research which is smaller compared to lengths of flanges to avoid including the other end portion of flange containing a different primitive other than cylinder and/or bolts. Value of a_g is set to 1cm, which is typically the maximum length of gasket.

Points found for demonstration pipe are shown in Fig. 3.35.

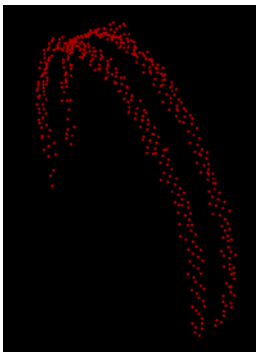


Fig. 3.35 Points for Comparison of Flange

Next, distances of these points are measured from axis of pipe and their mean and standard deviation values are calculated. If this mean lies within 1cm to design radius of cylindrical portion of flange and standard deviation less than 1cm, then the cluster is regarded as flange as shown below.

F_r	Design radius of cylindrical portion of flange
μ_d	Mean of distances of all points to axis
σ_d	Standard deviation of distances of all points to axis

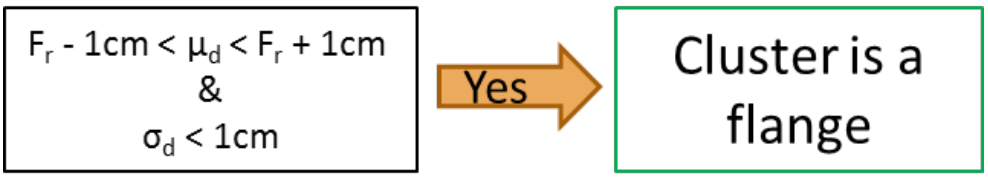


Fig. 3.36 Comparison of Flange

In case of flange found of demonstration pipe, μ_d = 98mm and F_r = 100mm, σ_d = 3.6mm.

3.4 Finding Positions

This section refers to the fourth section in the overview shown in Fig. 3.1.

In this part of algorithm, position of pipe is found by finding the positions of elbow or/and flange adjacent to pipe in cases, where

1) Pipe is connected to other pipes on both ends with two flanges of same type.
2) Pipe is connected to another pipe to an end with two flanges of same type and on the other end to an elbow.
3) Pipe is connected to elbows at both ends.

In piping system, there are a variety of components pipe may be connected to and pipes may be extended into these components like in case of flanges as shown in Fig. 3.37. Also, lengths of installed pipes may vary from the designed values and total length of pipe may not be scanned. With all the constraints above, to find position of pipes accurately, there is a need to find the positions of components pipe connecting to which may not match the designed components. After finding the positions of components connecting pipe, design information of pipe and components connected to pipe is used to find the end points of pipe accurately.

3.4.1 Finding Position of Flange and End of Pipe at Flange

As the two flanges in cluster are same, the center of shadowed region in which flange's

projection center lies is regarded as an end of flange where cylindrical portion starts neglecting very thin gasket existing between flanges. For finding end of pipe at flanges, the distance (d) as shown in Fig. 3.37 is calculated from the design data. End point of pipe is calculated by finding the point at same distance d along the axis found from the flange start position. Direction of pipe is determined by the other end of pipe. The whole process of this is described in Fig. 3.38.

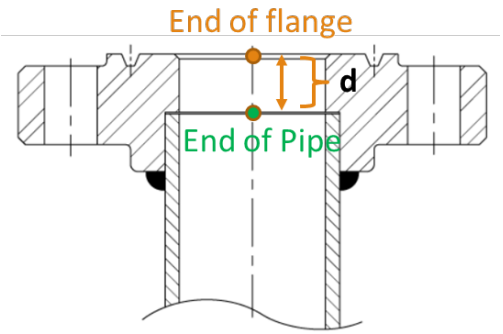


Fig. 3.37 Ends of Pipe and Flange

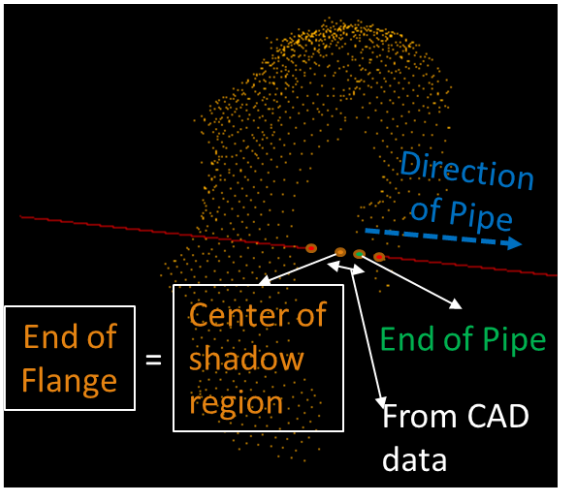


Fig. 3.38 Overview of Finding Ends of Pipe and Flange

3.4.2 Finding Positions of Elbow, Tee and End of Pipe at Elbow

As mentioned before, pipe spools are often flanged due to its easy maintenance. Here, ends of pipe at other components elbow and tee junction are found. Radius and axis of pipes found along with the design parameters of elbow and tee are used to find the positions of pipe, elbow and tee. This process is explained using a pipeline shown in Fig. 3.39.

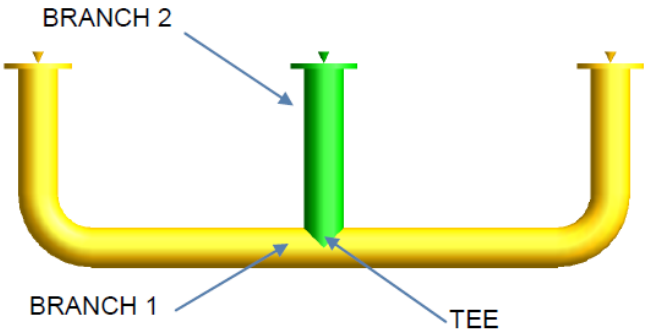


Fig. 3.39 Demonstration Pipeline

Previous to this step, axis and radius of pipe, an end of pipe and flange connected to pipe are found. Hence for this pipeline, the information that will be found till previous steps is as shown in Fig. 3.40.

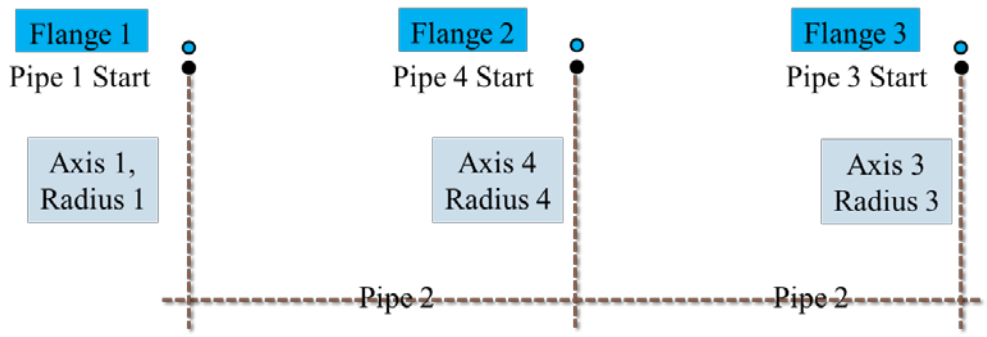


Fig. 3.40 Demonstration Pipeline after Finding an End of Flanges, Pipes

Elbow and tee components in this pipeline are found by intersection of pipes connecting the respective component. The axes of pipes connecting component may not intersect. In that case, center of shortest distance line segment between the axes of pipes connecting is regarded as the position of component. Axes parallel to axes of found connecting pipes and passing through position of component are regarded as the axes of pipes connecting the component. If the pipes connecting components are not found, then the component is regarded as not installed.

End point of pipe at elbow can be calculated as shown in Fig. 3.41 using rout radius of design elbow and angle found between axes of pipes connecting at elbow. While calculating end point of pipe at elbow, position of elbow is taken as the intersection of shortest distance line segment between the axes of pipes connecting at elbow with the pipe's axis, whose end point is being calculated. In case of tee junction, the end point of pipe can be calculated similarly by using the primitives of tee junction. In the present case, tee junction component has no physical existence.

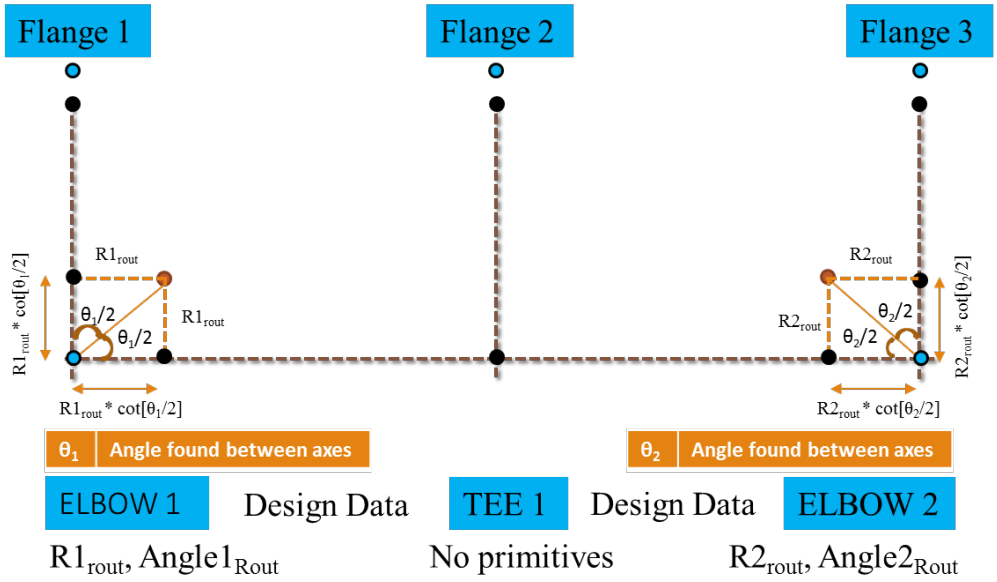


Fig. 3.41 Finding End Point of Pipe at Elbow and Tee

Output of this process for demonstration pipeline can be seen in Fig. 3.42.

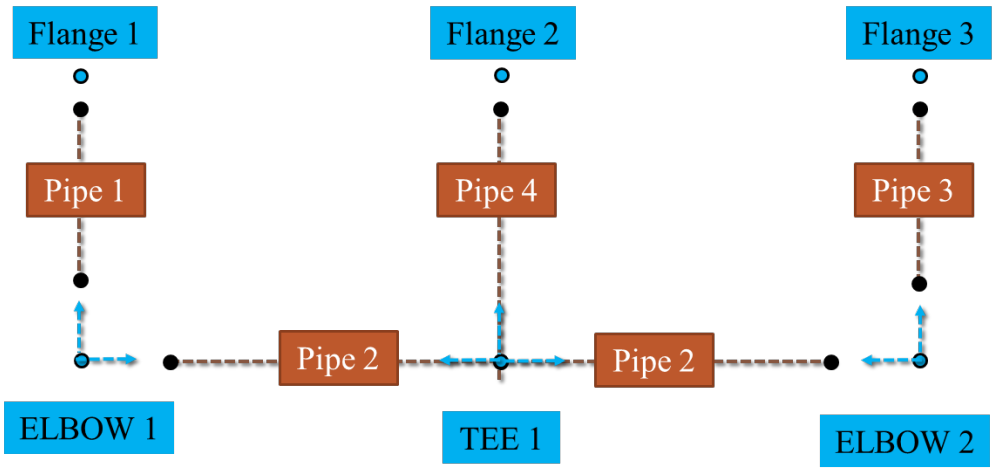


Fig. 3.42 Positions of Elbow, Tee and Pipe

Chapter 4

4 Implementation and Verification of Algorithm

In this chapter, we will discuss the program developed based on the contents suggested in chapters 2 and 3.

4.1 Implementation

The program in this study was implemented in Microsoft Visual Studio 2013® based on Point Cloud Library (PCL) 1.7.2 [18]. The inputs to the program are RVT formatted file of machinery part of a block from AVEVA Marine® for as-designed data and PCD (Point Cloud Data, the format of PCL library) formatted point cloud file for as-built information.

Flowchart representing the brief implementation of total algorithm is shown in Fig. 4.1.

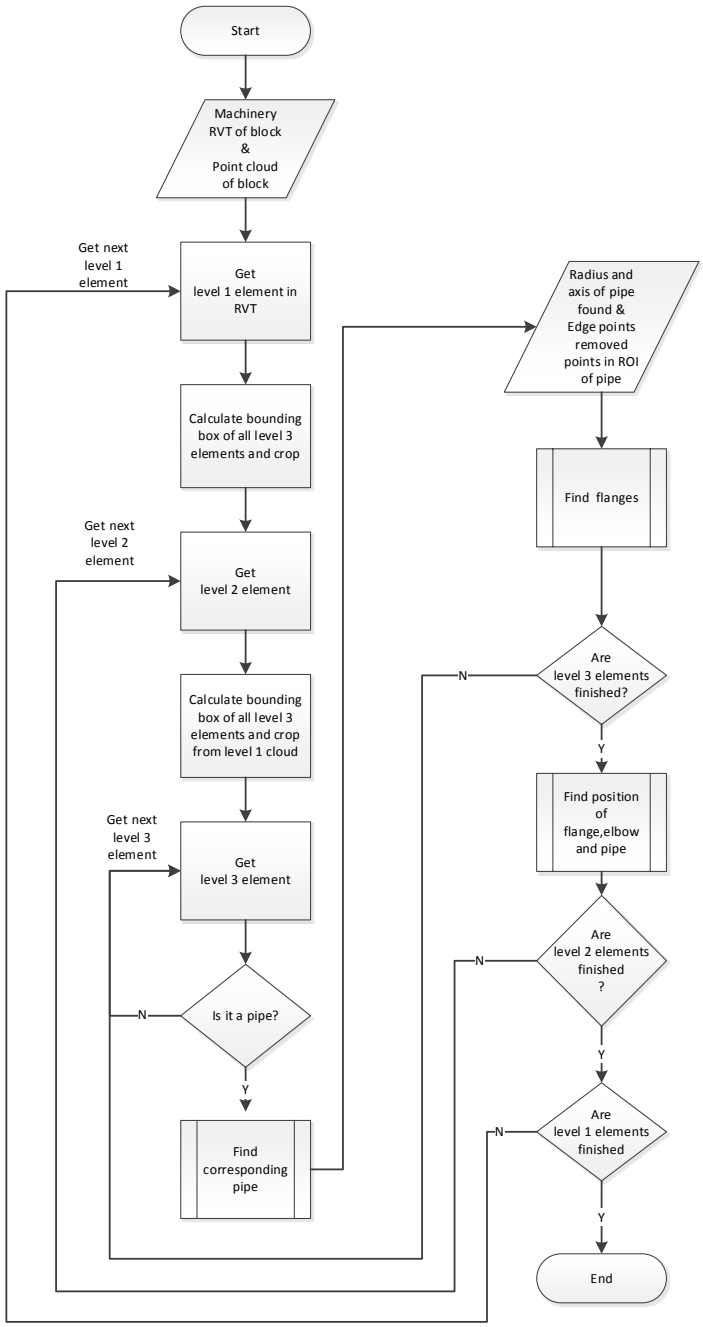


Fig. 4.1 Flowchart Representing Implementation of Algorithm

Position of pipe is found by finding the position of connecting flange and elbow. Pipe with flanges on either ends is represented as a branch (level 2 element) in the RVT format. As the branch containing elbow does not contain a pipe connected with flange on both ends. And also flanges, elbows connecting pipe are next to the pipe in the components list. Finding positions of flange, elbow and pipe are implemented in two scenarios. First one is the case of branch containing at least one elbow. Its brief representation is shown in Fig. 4.2.

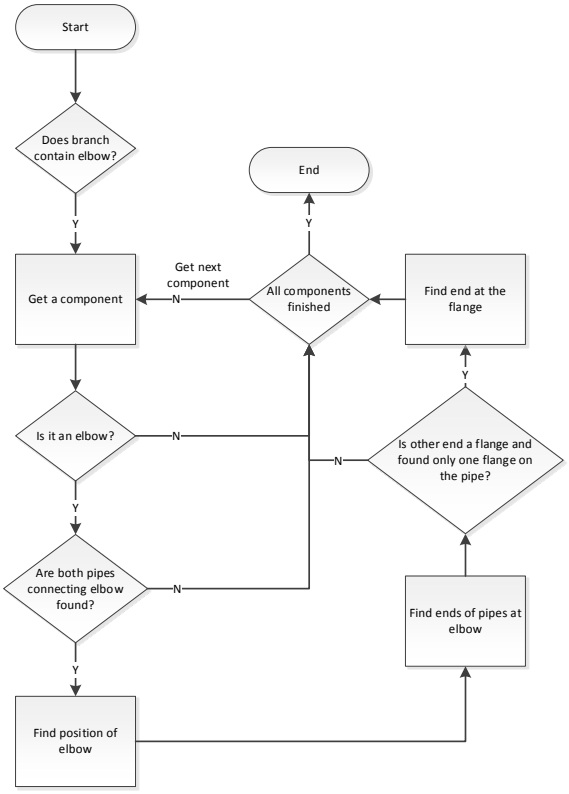


Fig. 4.2 Flowchart Representing Implementation of Finding Position of Pipe with At Least One of the Ends as Elbow

And the other is all other remaining branches and its brief representation is shown in Fig. 4.3.

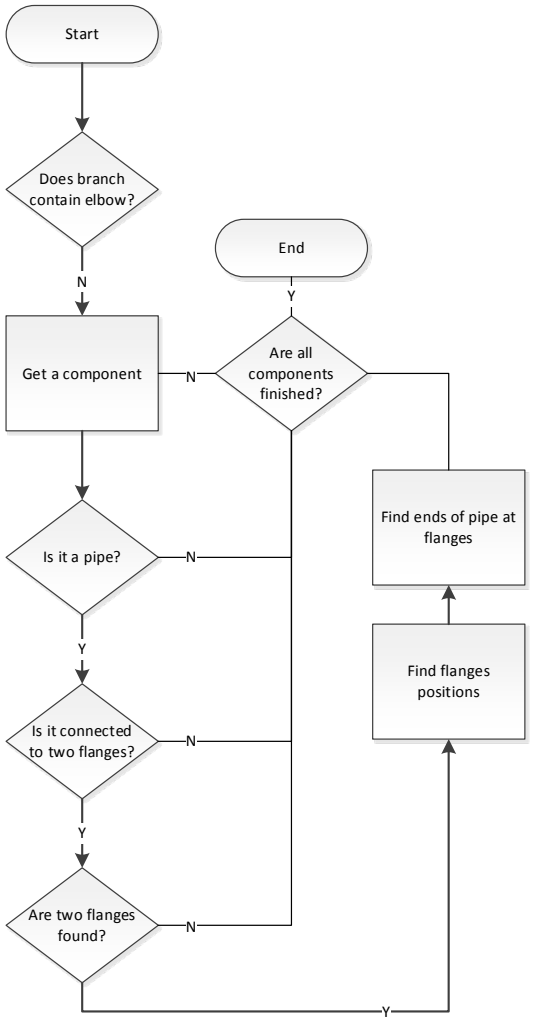


Fig. 4.3 Flowchart Representing Finding Position of Pipe with Flanges on Both Ends

4.2 Verification Methods and Target Parts of the Program

We tested the algorithm using real point cloud and corresponding CAD data based on the program referenced in section 4.1 and verified the program.

The test scene is a part of engine room of fully constructed LNG carrier waiting for delivery. The scene consists of 7 partially scanned blocks. It was scanned from 5 different locations as shown in Fig. 4.4. Point clouds from locations 1, 2, 3, 4 and 5 are shown in Fig. 4.5, Fig. 4.6, Fig. 4.7, Fig. 4.8 and Fig. 4.9 respectively. The maximum distance of objects in the scene to scanner in all scans is less than 50m. The scanner has position accuracy of 3mm at 50m and 2mm standard deviation up to 50m. Large errors do exists on the edges. And the density of point cloud is such that maximum distance of non-edge point to nearest point is 6mm. Registered total point cloud is show in Fig. 4.10.

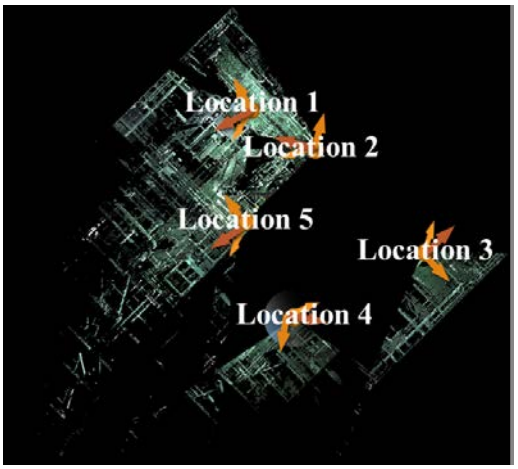


Fig. 4.4 Scan Locations of Test Scene in Top View of Decimated Point Cloud

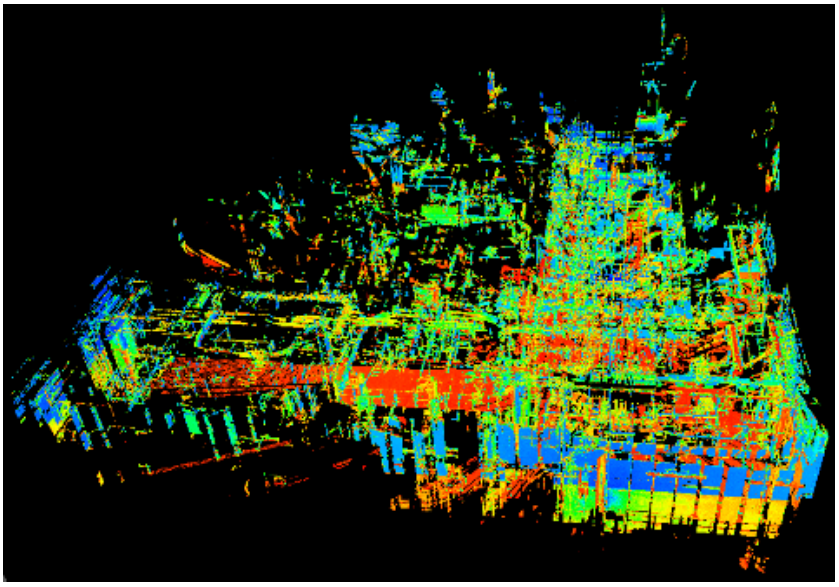


Fig. 4.5 Point Cloud from Location 1

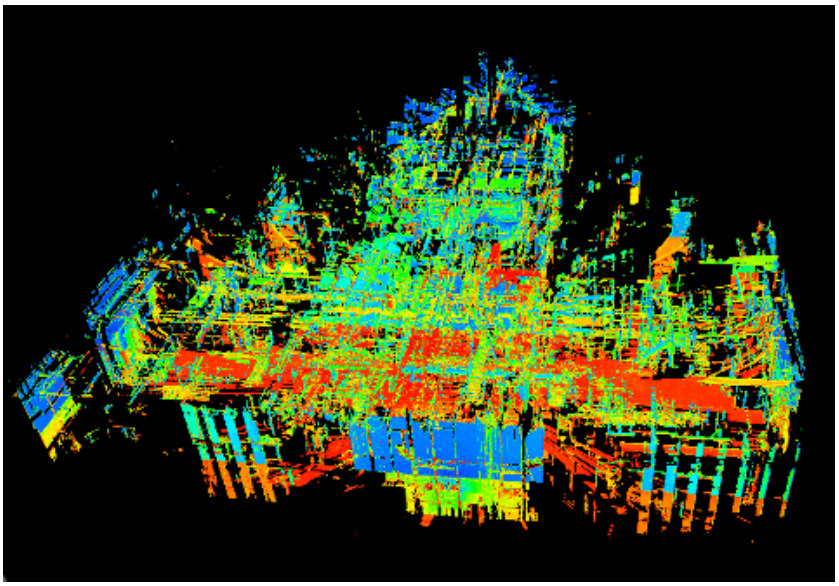


Fig. 4.6 Point Cloud from Location 2

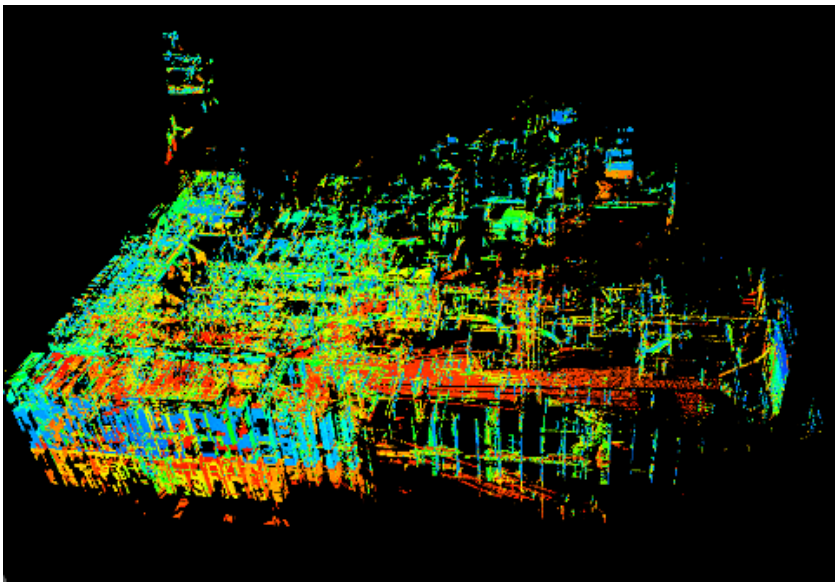


Fig. 4.7 Point Cloud from Location 3

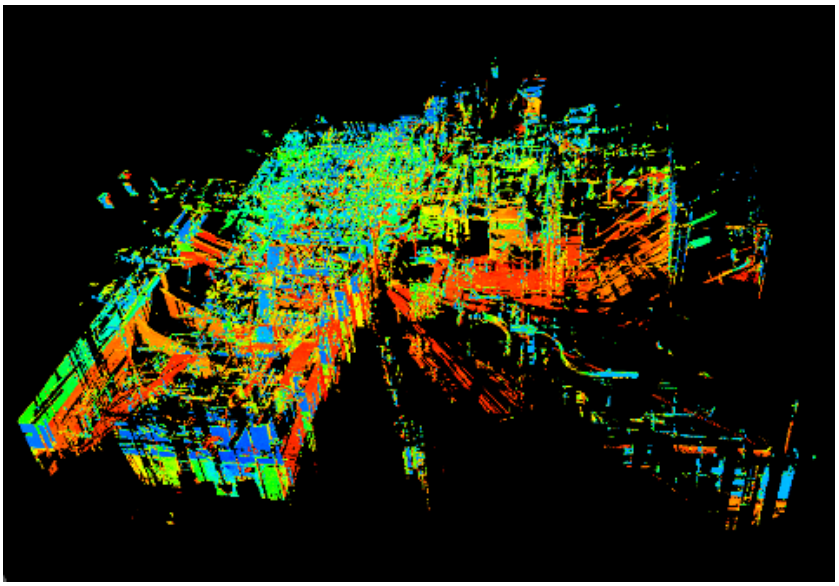


Fig. 4.8 Point Cloud from Location 4

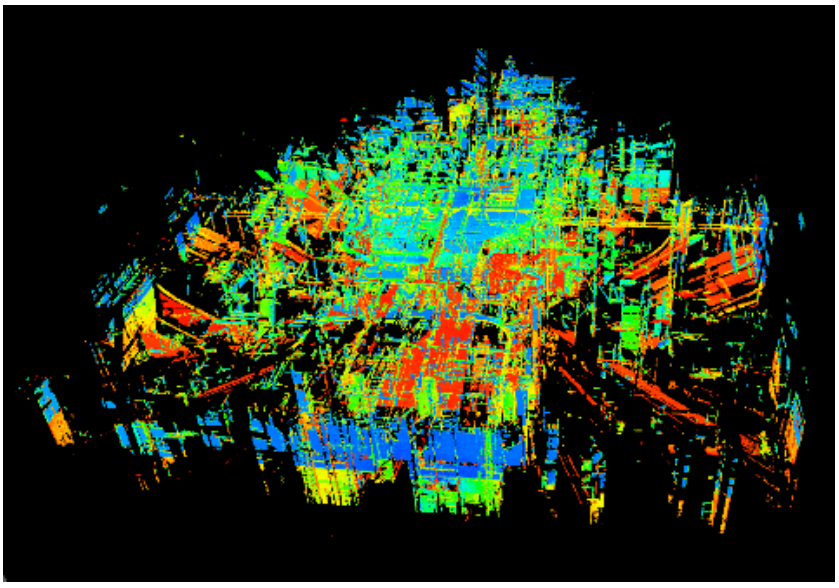


Fig. 4.9 Point Cloud from Location 5

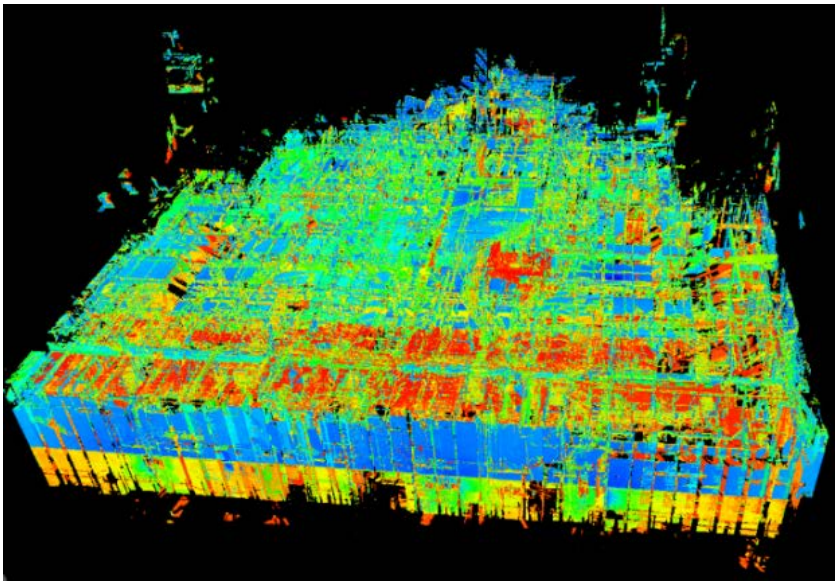


Fig. 4.10 Total Point Cloud of Scene.

Size of the point cloud and number of points in it cloud are shown in Table 4.1.

Table 4.1 Input Point Cloud

Number of points	0.75 billion
Input file size	33.5 GB

Among the seven partial scanned blocks, the algorithm was verified with two blocks shown in Fig. 4.11 and Fig. 4.12.

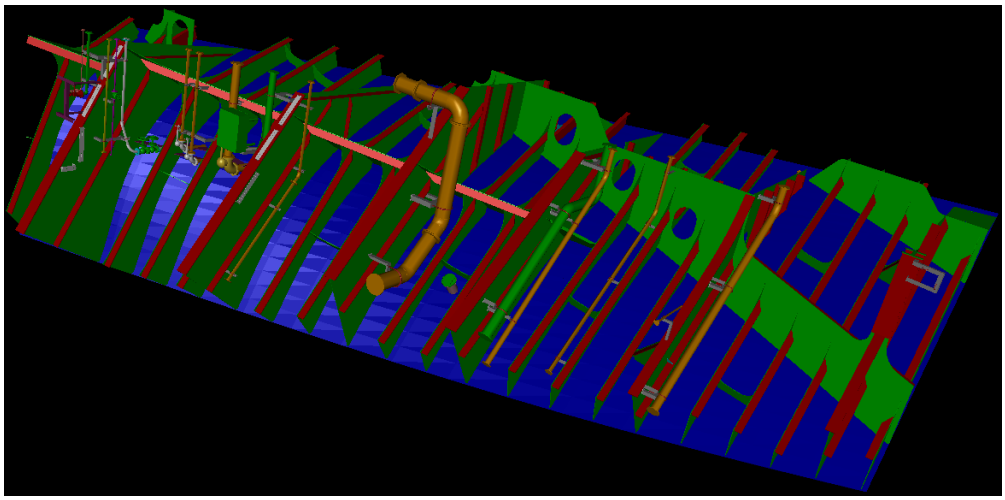


Fig. 4.11 Test Block 1

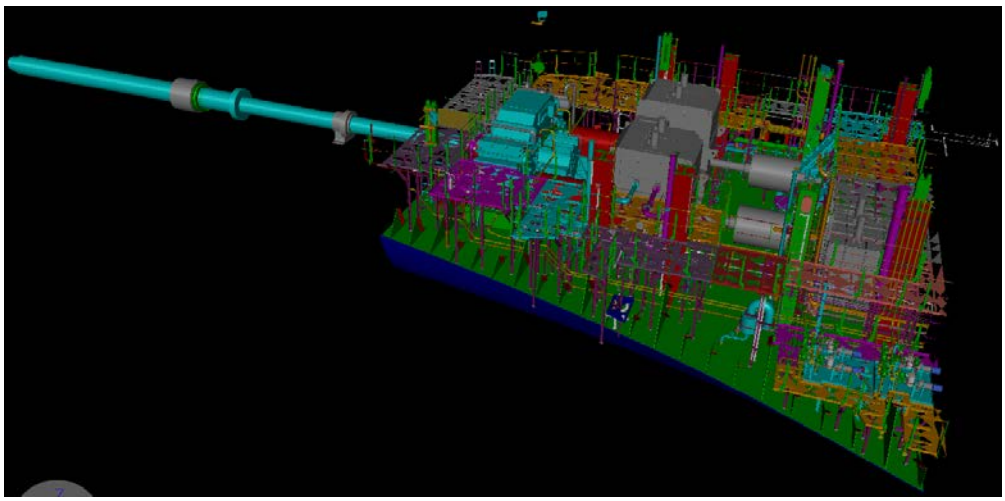


Fig. 4.12 Test Block 2

Softwares used for verification of the algorithm are shown in Table 4.2.

Table 4.2 Softwares Used for Verification

S.No	Software	Use
1	CloudCompare	Visualization of point cloud and CAD model, and measurements in point cloud
2	Autodesk Navisworks Manage	Visualization of point cloud and CAD model
3	Autodesk ReCap	Visualization of point cloud and scene images

Algorithm has two parts as following

1) Finding corresponding pipe

2) Finding positions of flange, elbow and pipe

Target parts and verification methods of both the parts are different and they are explained in detail in the following sections.

4.2.1 Finding Corresponding Pipe

4.2.1.1 Verification Method

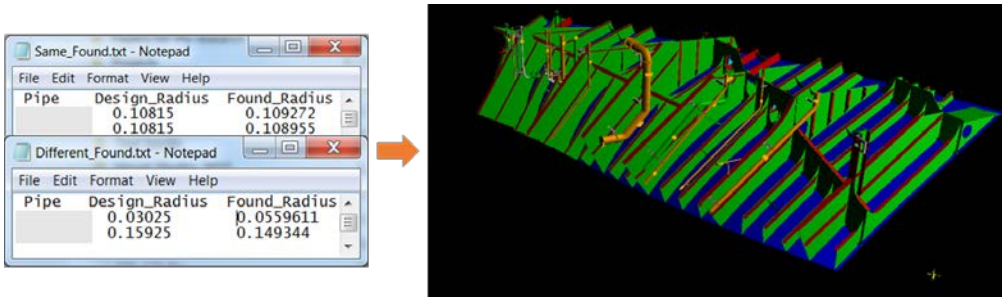


Fig. 4.13 Verification of Finding Corresponding Pipe

Fig. 4.13 is the verification method of this part of algorithm. The program is facilitated to output two text files named “Same_Found” and “Different_Found”. Former file contains all the designed pipes found in the point cloud with design radius and found radius printed against pipe name as shown in Fig. 4.13. And the latter file contains all the designed pipes that are not found in the point cloud. But there exists another pipe in the ROI of designed pipe, which could be a pipe that is installed instead of designed pipe or another designed pipe. Similar to the former file, design radius and found radius are printed against pipe name as shown in Fig. 4.13. Also, it is facilitated to output cluster that found the corresponding pipe. Steps of verification are as given below

1) Radius difference between the found radius and design radius was verified.
2) Cluster that found the corresponding pipe was overlapped with the CAD model and verified if cluster belongs to the same pipe it represents.

4.2.1.2 Target Parts of Verification

Table 4.3 Target Pipes for Finding Corresponding Pipe

Targets for finding corresponding pipe
Pipes with length $\geq$ partition length* & radius $\geq$ 1cm

(Refer section 3.2.2 for partition length*). Table 4.3 is the verification subject designed in this part of study. Pipes with lengths greater than partition length and radii greater than 1cm are the targets of the test. Since the algorithm cannot find a cluster belonging only to the pipe for lengths smaller than partition length, the target pipes are just those pipe with lengths greater than partition length. Length of partition affects the accuracy of radius and axis found, which can decide whether the pipe found is corresponding to the designed pipe or not. Partition length is a parameter. Radius is limited to 1cm, because in some cases, group of error points can be misrecognized as pipe. Table 4.4 represents the number of target pipes in both the blocks. The algorithm was verified using these pipes.

Table 4.4 Quantity of Target Parts for Finding Corresponding Pipe

Block	Target pipes of the program
Block 1	40
Block 2	564

4.2.2 Finding Positions of Flange, Elbow and Pipe

4.2.2.1 Verification Method

The program is facilitated to output PCD (Point Cloud Data, format of PCL) formatted files of following

a) Points belonging to pipe.

b) Points belonging to pipe projected on its axis.

c) Points belonging to cylindrical portions of flange.

d) Points belonging to elbows

It is also facilitated to output text files of the following

a) Lengths of pipes whose position is found along with the design length.

b) Angles found between the axes of pipes connecting elbow along with the design elbow angle.

This part of algorithm was verified in 3 ways as below

1) Verified manually after overlapping points belonging to pipe, cylindrical portions of flange and elbow with the CAD model.

To verify how accurate the axis found are,

1) Lengths of pipes found were compared with their design values.

2) Angles found between pipes connecting elbows were compared with the design angles of elbow.

4.2.2.2 Target Parts of Verification

Table 4.5 Target Pipes for Finding Position of Pipe

Targets for finding position of pipe
Found designed pipes connected to either elbow or flange on both ends

Table 4.6 Target Flanges for Finding Position of Flanges

Targets for finding position of flanges
Flanges connecting pipes

Table 4.7 Target Elbows for Finding Position of Elbows

Targets for finding position of elbow
Elbows connecting to pipes found on both ends

Table 4.5, Table 4.6 and Table 4.7are the verification subjects of this part of algorithm. Pipes found same as designed in the previous part of algorithm are a superset of the targets. Among the pipes found same as designed, pipes whose both ends are connected to either elbow (elbow connecting to another found pipe) or flange are the targets. In case of elbows, elbow connecting to two found pipes are the targets.

Whereas in case of flanges, flange connecting to a pipe found and a similar flange are the targets.

The test of this part of algorithm was verified with both the blocks of LNG carrier. In case of block 1, a portion was fully scanned and the target parts in that portion are given in Table 4.8. Block 2 is a complex multilayered block. Only small portions of pipes and flanges were scanned. Also as the number of points belonging to flange are small. It was difficult to judge the number of flanges fully scanned. Hence the accuracy was verified with only block 1.

Table 4.8 Quantity of Target Parts of Block 1 for Finding Position

Type	Targets of the test
Flange	15
Elbow	6
Pipe	11

4.3 Analysis of the Parameters

All the parameters in algorithm whose values are not set in the description of algorithm are shown in Table 4.9 along with their major dependencies. Of all the parameters, partition length is the only parameter that is independent of accuracy of scanner, density of point cloud and installation inaccuracy. Parameters that depend on the scanner accuracy and density of point cloud were set by analyzing the input point cloud, running few experiments and inferences. Set values of all the parameters and how the

values were set is shown in Table 4.10

Table 4.9 All the Parameters of Algorithm

Part of algorithm	Parameter	Dependency
Region of Interest	l_A	Installation inaccuracy
	$l_{A\perp}$	
Partition	l_p	-
Edge points removal	d_e	Density of point cloud and accuracy of scanner
Clustering	d_c	Density of point cloud and accuracy of scanner
Is cluster a cylinder	t_c	Accuracy of scanner and partition length
Is cluster a part of design pipe	t_r	Accuracy of scanner and partition length
	a_d	Installation inaccuracy & to avoid detection of other pipes in ROI

Table 4.10 Values of the Parameters

Parameter	Value	How the value was set?
$\mathbf{l_A}$	16cm	Input from the data provider
$\mathbf{l_{A\perp}}$	10cm	
$\mathbf{d_e}$	10mm	Maximum distance of a point to nearest is 6mm and scanner inaccuracy is 3mm. After few experiments
$\mathbf{d_c}$	9mm	
$\mathbf{t_c}$	4mm	Inaccuracy of scanner is 3mm. After few experiments
$\mathbf{t_r}$	3mm	Inaccuracy of scanner is 3mm. After few experiments
$\mathbf{a_d}$	20°	Least angle of elbow in input data is 20°

Effect of length of partition (l_p) was analyzed by finding difference of angles found

at elbow and the design angles of elbows. This test was carried on Block 1, which contains fully scanned pipes with l_p = 8cm, 10cm and 12 cm. Although the difference for each pipe is not proportional, the average of absolute difference in angles is proportional to length of partition and is shown in Table 4.11.

Table 4.11 Analysis of Length of Partition Parameter

	l_p = 8 cm	**l_p = 10 cm**	**l_p = 12 cm**
Average absolute difference of angle	1.07°	0.75°	0.54°

As the length of partition also influences the smallest length of target pipes. Detailed results for both blocks were tested with 12cm partition length.

4.4 Results and Analysis of the Test

The parameters shown in Table 4.12 were set for the detailed results presented in this section.

Table 4.12 Values of All Parameters Set

Part of algorithm	Parameter	Value
Region of Interest	l_A	16cm
	$l_{A\perp}$	10cm
Partition	l_p	12cm
Edge points removal	d_e	10mm
Clustering	d_c	9mm
Is cluster a cylinder	t_c	4mm
Is cluster a part of design pipe	t_r	3mm
	a_d	20°

4.4.1 Preprocessing

4.4.1.1 Registration of Point Cloud and CAD Model

Three points of four corners of a rectangular cross section in CAD model were chosen and their corresponding points in point cloud were manually found. Coordinates of these three corresponding points are shown in Table 4.13. Point cloud and CAD model before registration is shown in Fig. 4.14 and after registration is shown in Fig. 4.15.

Table 4.13 Corresponding Points of Point Cloud and CAD Model

Point	Coordinate	Point Cloud(m)	CAD(m)
P1	**x**	-8.9	24
	y	-20.13	0
	z	1.135	8.076
P2	**x**	1.423	53.6
	y	11.531	15.295
	z	4.071	10.576
P3	**x**	22.988	53
	y	-10.404	-15.295
	z	3.936	10.576

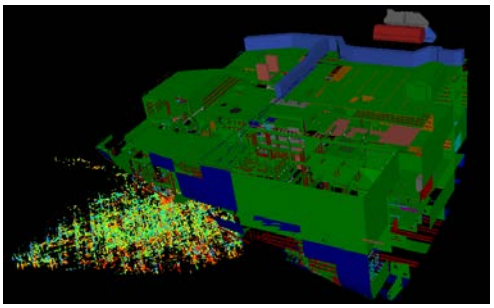


Fig. 4.14 Point Cloud and CAD Model Before Registration

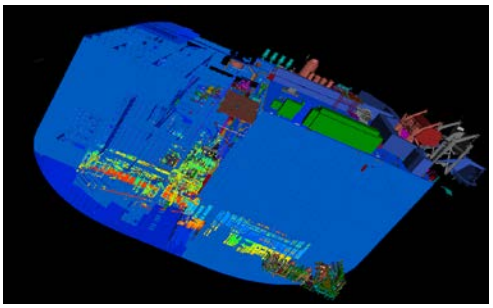


Fig. 4.15 Point Cloud and CAD Model After Registration

4.4.1.2 Point Cloud Format Conversion

Based on the format of input point cloud PTX (Point cloud format of Leica scanners) mentioned in 2.1.3.1 and PCD (Point Cloud Data of PCL) of [18], a program was developed to convert ASCII based PTX format to PCD binary format. The result of this step can be seen in Table 4.14.

Table 4.14 Input Point Cloud Before and After Format Conversion

Format	Size
PTX format (X, Y, Z, I, R, G, B)	33.5 GB
Binary PCD format (X, Y, Z)	1.69 X 5 = 8.45 GB

4.4.1.3 Cropping of Point Cloud by Block

Based on the shape of Region of Interest (ROI) described in 0 and from the inputs of production engineers of construction of the input data, it was found that all the pipes can be found along with flanges if connected in cuboid with pipe at center (refer 0) with parameters l_A = 16cm and $l_{A\perp}$ = 10cm.

Table 4.15 Number of Points of Each Block

Block	Number of points
Block 1	9,654,983
Block 2	103,317,673

4.4.2 Finding Corresponding Pipe

4.4.2.1 Results

A pipe is regarded as scanned only if a minimum length of partition length and minimum angle the pipe's surface can make with axis is 90°.

Results of this part of program for block 1 are shown in Table 4.16, Table 4.17 and

Table 4.18.

Table 4.16 Block 1 Designed Pipe Detection

	True	False	% of Accuracy
Designed pipe exists	22	1	95.65
Designed pipe not exists (many pipes were not scan targets)	17	1	94.44

Table 4.17 Block 1 Designed Pipe False Detection

Recognized other component as designed pipe		**Recognized scanned designed pipe as different pipe**	**Recognized pipe as not installed**
Other component is pipe	**Other component is non-pipe**		
0	0	0	1

Table 4.18 Block 1 Different Pipe Detection

Different pipe	**Another non-pipe component**	**Another pipe component**	**Pipe and elbow combination**	**Same designed pipe**
1	0	0	0	1

Results of this part of program for Block 2 are shown in Table 4.19, Table 4.20 and Table 4.21.

Table 4.19 Block 2 Designed Pipe Detection

	True	False	% of Accuracy
Designed pipe exists	107	28	79.26
Designed pipe not exists(many pipes were not scan targets)	422	28	93.78

Table 4.20 Block 2 Designed Pipe False Detection

Recognized other component as designed pipe		**Recognized scanned designed pipe as different pipe**	**Recognized pipe as not installed**
Other component is pipe	**Other component is non-pipe**		
6	1	5	14

Table 4.21 Block 2 Different Pipe Detection

Different pipe*	**Another non-pipe component**	**Another pipe component**	**Pipe and elbow combination**	**Same designed pipe**
1	8	3	2	7

All the pipes found are shown with their tentative pipe points in Fig. 4.16. Total accuracy of designed pipe detection combining both the blocks is shown in Table 4.22. In both the blocks, there are many non-scan targeted pipes.

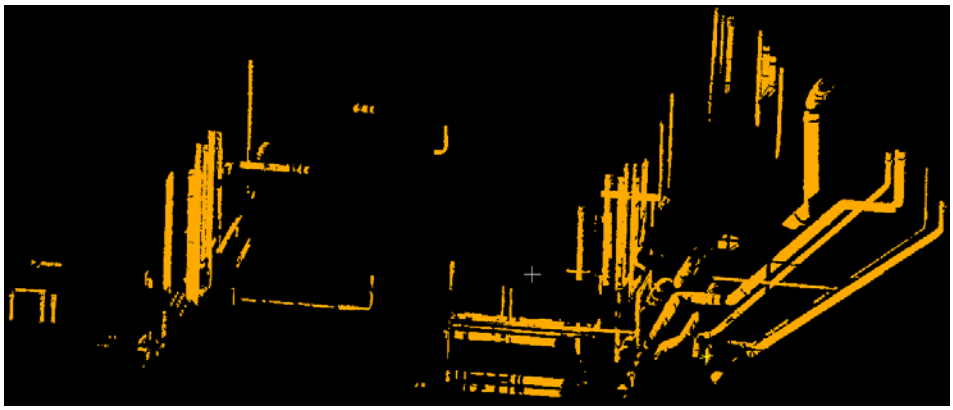


Fig. 4.16 Pipes Found of Both Block 1 and Block 2

Table 4.22 Designed Pipe Detection Accuracy

	% of Accuracy
Designed pipe exists	81.65
Designed pipe not exists (many pipes are not scanned)	93.80

4.4.2.2 Analysis

It is observed that the reasons for each case of false designed pipe detection as below.

Recognized other component as designed pipe: When actual pipe is scanned relatively smaller surface or not scanned at all and other components with similar parameter exists in ROI. An example of this, in case of pipe running parallel to similar pipe can be seen in Fig. 4.17

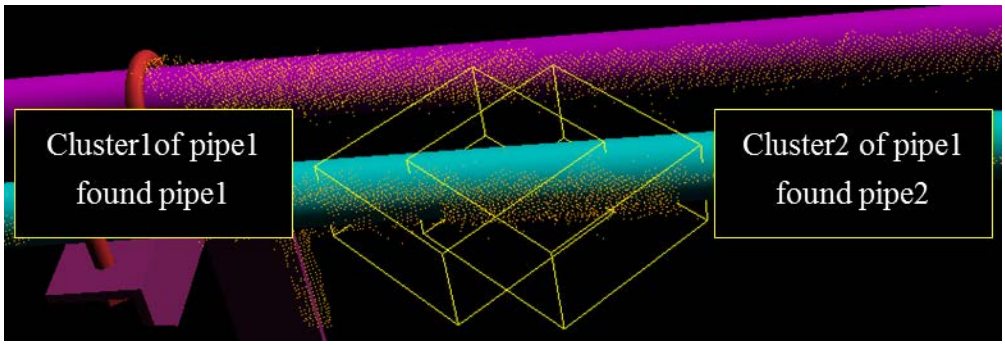


Fig. 4.17 Other Component Recognized as Designed Pipe

Recognized scanned designed pipe as different pipe: When scanned area is small or/and large errors exists, accuracy of radius calculated is less than t_r (3mm here). An example of this case can be seen in Fig. 4.18, where a pipe consists of points from two scans and have more errors at edges those cannot be removed with edge point removal technique which was proposed for single scan.

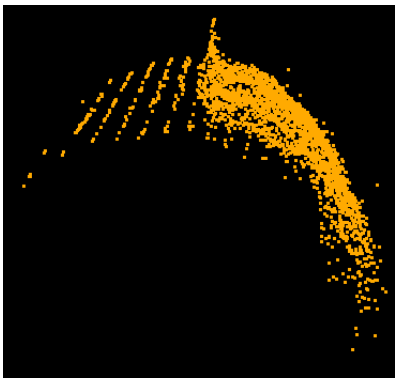


Fig. 4.18 Same Pipe Recognized as Different Pipe

Recognized pipe as not installed: When length and/or radius of pipe is small and/or large errors exists. An example of this case where a continuous length of partition with only pipe points is not available is shown in Fig. 4.19.

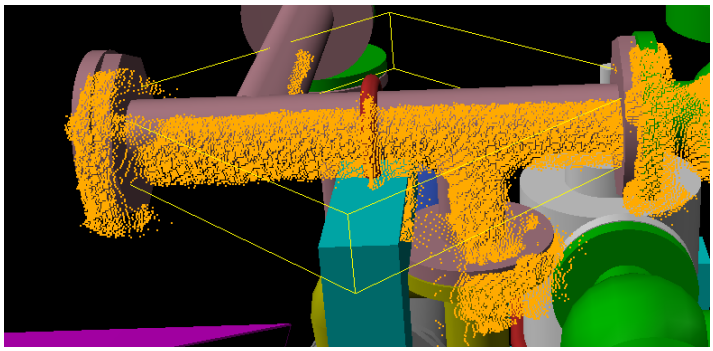


Fig. 4.19 Pipe Recognized as Not Installed

In case of different pipe detection, we were already aware of detection of planes from non-pipe component as cylinder when small length/breadth of plane exists in the cluster. But left it for future work. This can be handled by limiting the angle, the cylindrical surface can make with its axis. Reasons for detection of pipe and elbow combination as different pipe is the existence of only small part of elbow together with pipe in a cluster as shown in Fig. 4.20 and in case of pipes connected to elbows with small bend angle.

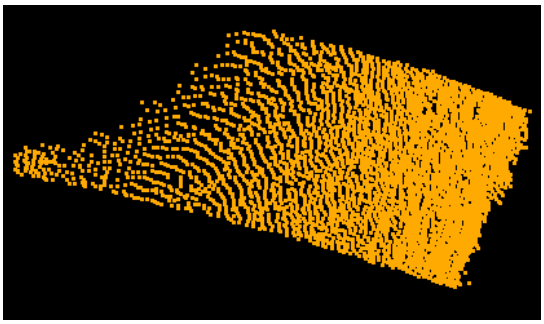


Fig. 4.20 Pipe and Elbow Combination Recognized as Pipe

All these inaccuracies can be verified at the manual review of results similar to

verification procedure suggested in section 4.2.1.1.

4.4.3 Finding Positions of Flange, Elbow and Pipe

4.4.3.1 Results and Analysis

In case of block 1, out of 15 flanges connecting to similar flanges scanned, 14 flanges are found. Elbows were found in all the cases as it just depends on the connecting pipes. Positions of pipes were found in all the cases. All the flanges, elbows and pipes whose positions were found are shown in Fig. 4.21.

Table 4.23 Accuracy of Flanges Detection of Block 1

Number of flanges scanned	Number of flanges found	Accuracy
15	14	93.3

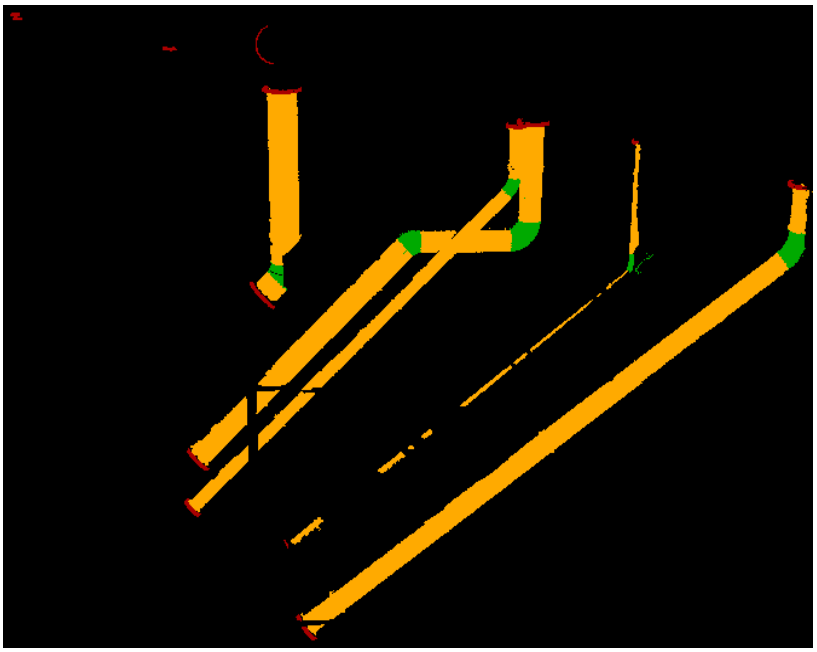


Fig. 4.21 Positions of Pipes, Flanges and Elbows of Block 1

All pipes, flanges and elbows of block 2whose positions are found are shown in Fig. 4.22.

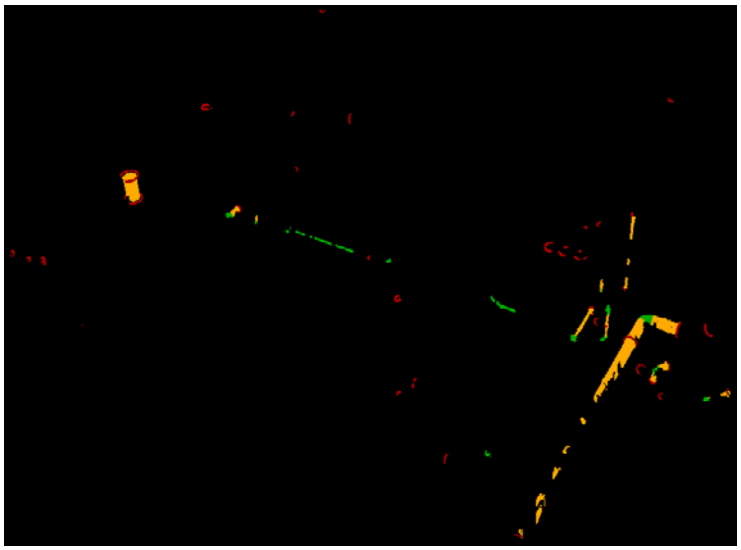


Fig. 4.22 Positions of Pipes, Flanges and Elbows of Block 2

Comparison of all the pipes in both block 1 and block 2 whose positions were found is shown in Table 4.24.

Table 4.24 Comparison of Results of Pipes Whose Positions were Found

Correctly detected	21
Wrongly detected	5

In case of two pipes connected by an elbow with flanges at other ends, a pipe was only partially scanned, but the pipe was found with large deviation in axis. Also, in case of 2 pipes with bends on both sides whose route angles of both bends are 20° and 30° respectively, the pipes were found with large deviation in axis because elbow was part of the cluster that found the pipe. Also, in case of a pipe (pipe 1) connected to another (pipe 2) through elbow. A pipe parallel to pipe 2 is misrecognized as pipe 2

and hence, the length calculated for pipe 1 was wrong. This case can be handled by limiting the shortest distance between the axes connecting elbow to find the position of elbow.

Comparison of all the elbows in both block 1 and block 2 found is shown in Table 4.25

Table 4.25 Comparison of Results of Elbows Whose Positions were Found

Correctly detected	21
Wrongly detected	5

In all the five wrongly detected cases, the route angles of elbows are less than 30°. In one of the axis or both axis of pipes connecting, elbow together with pipe was recognized as pipe. Hence, there was large deviation in axis and hence the position of elbow.

4.4.3.2 Verification

To verify accuracy of axes and positions found,

1) Lengths of 21 correctly found pipes were compared with the design values and they are shown in Table 4.26. And their average, minimum and maximum values are shown in Table 4.27.

Table 4.26 Comparison of Lengths of Pipes Correctly Found with Designed Lengths

S.No	Design length (cm)	Found length (cm)	|Design - Found| (cm)
1	600.00	597.30	2.70 (largest)
2	70.88	70.52	0.36
3	60.15	60.87	0.72
4	94.62	93.88	0.74
5	29.11	28.94	0.17 (smallest)
6	30.68	28.78	1.90
7	30.68	31.02	0.34
8	190.19	188.85	1.34
9	15.01	15.94	0.93
10	21.88	24.13	2.25
11	49.14	51.60	2.46
12	48.00	46.79	1.21
13	300.47	300.81	0.34
14	94.65	93.96	0.69
15	96.38	96.63	0.25
16	449.64	448.54	1.10
17	48.22	50.58	2.36
18	21.71	22.03	0.32
19	190.84	189.75	1.09
20	507.50	507.94	0.44
21	36.49	37.42	0.93

Table 4.27 Statistics of Differences of Lengths of Pipes Correctly Found and their Designed Lengths

Average absolute difference	1.08 cm
Minimum absolute difference	0.17 cm
Maximum absolute difference	2.7 cm

2) Angles found between pipes at elbows with route angle greater than 30° are compared with design values and they are shown in Table 4.28 and their average, minimum and maximum values are shown in Table 4.29.

Table 4.28 Comparison of Angles Found at Bends with Designed angles

S.No	**Design angle (°)**	**Found angle (°)**	**|Design - Found| (°)**
1	90.00	89.54	0.46
2	90.00	89.09	0.91
3	45.00	45.29	0.29
4	90.00	89.76	0.24
5	90.00	89.00	1
6	90.00	89.55	0.45
7	90.00	89.58	0.42
8	90.00	87.66	2.34 (largest)
9	90.00	88.95	1.05
10	90.00	89.75	0.25
11	90.00	89.23	0.77
12	90.00	89.37	0.63
13	90.00	88.39	1.61
14	90.00	89.67	0.33
15	38.80	39.81	1.01

S.No	Design angle (°)	Found angle (°)	|Design - Found| (°)
16	45	44.53	0.47
17	90	89.91	0.09 (smallest)
18	45	46.78	1.78
19	45	45.12	0.12
20	53	52.74	0.26
21	51.20	50.17	1.03

Table 4.29 Statistics of Differences of Angles Found at Elbows and their Design Angles

Average absolute difference	0.74°
Minimum absolute difference	0.09°
Maximum absolute difference	2.34°

Chapter 5

5 Conclusion and Future Works

5.1 Conclusion

This study was done to automate the process of finding positions of as-built pipes, flanges and elbows corresponding to their designed ones which is required for as-built modeling of installed piping system. Modification/Modeling of only those components whose position differs more than the tolerance of the component is required. However with the existing manual as-built modeling, all the components are modified/remodeled. Therefore, an algorithm to automate the process of finding positions of components mentioned corresponding to their designed ones was suggested. The algorithm is extendable to tee junction. The positions are found in case of pipes connecting other pipes through either of flange or elbow on both ends.

To automate the process of finding position of installed pipes corresponding to their designed ones, designing of piping specifically in AVEVA Marine was studied. It was observed that pipe spools are often flanged to facilitate easy connection to other spools and flanges of same type are used to connect two pipe spools. Also, it was observed that the difference in position of as-designed and as-built pipes do not vary much.

However, due to non-updating of design after revisions, a different component other than the designed may exist in the construction. Also, it was observed that few pipes are cut to lengths less than or more than design values.

Based on above observations, for each designed pipe, a Region of Interest (ROI) was calculated and corresponding installed pipe was found by finding components the pipe connected to. Hence, the misrecognition of cylinders and planes from other components as pipes were greatly reduced compared to previous works.

From this research, a program was developed and tested with industrial CAD data and its corresponding as-built point cloud data. This research was verified manually by overlapping axes, pipe points, flange points and elbow points with total decimated point cloud, and also by comparing lengths of pipes found, angles found at bends with the design values. Existence of designed pipe in point cloud was found with an accuracy of 81.65% and non-existence with 93.8%, two pipes that exists differently from the designed pipe were detected. Average difference in lengths of pipes found was 1.08cm, and angles of elbows found was 0.74° in case of bends more than 30° compared to their design values.

Based on the above results, this research is expected to find positions of pipes, flanges and elbows in cases mentioned above which can be used to judge whether to modify the construction or not accurately. Also, axis found in case of all pipes is expected to help judge to some extent and be a guideline for modification.

5.2 Future Works

Firstly, we will develop program involving tee junctions and find the applicability of research for tee junctions. One more foremost priority is to implement distinction of planes from cylinders which can be done by limiting the minimum angle, the surface can make with its axis to be regarded as cylinders. In addition, to increase the applicability of the research, we would like to add more types of components that pipes can connect to, to find positions.

Bibliography

[1] Aur´elien Bey, Rapha¨elle Chaine, Rapha¨el Marc, Guillaume Thibault, Samir Akkouche, Reconstruction of Consistent 3D Cad Models from Point Cloud Data Using A Priori CAD Models, Calgary, 2011.

[2] M.-J. Son, "Decision Support System for Job Assignment in Shipbuilding Design Using Discrete-event-based Business Process Simulation," Doctorate degree thesis of Seoul National University, 2013.

[3] Moyst, Howard; Das, Biman "Factors Affecting Ship Design and Construction Lead Time and Cost," *Journal of Ship Production,* Vol. 21, No. 3, pp. 186-194, 2005.

[4] Cox, Joe Sanderson and Andrew, "The Challenges of Supply Strategy Selection in a Project Environment: Evidence from UK Naval Shipbuilding," *Supply Chain Management,* Vol. 13, No. 1, pp. 16 - 25, 2008.

[5] C. H. Kim, S. W. Kwon, and C. Y. Cho, "Development of Automated Pipe Spool Monitoring System using RFID and 3D Model for Plant COnstruction Project," *KSCE Journal of Civil Engineering,* Vol. 17, No. 5, pp. 865-876, 2013.

[6] W. M. Huitt, *Piping Design Part 3: Installation, Cleaning, Testing & Verifification,* 2007.

[7] San José Alonso, J.I., Martínez Rubio, J., Fernández Martín, J.J., García Fernández, J, Comparing Time-of-Flight and Phase-shift. The Survey of the Royal Pantheon in the Basilica of San isidoro (León)

[8] Bolles, Martin A. Fischler and Robert C, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Communications of the ACM,* Vol. 24, No. 6, pp. 381-395, 1981.

[9] Shah, Tahir Rabbani, "Automatic Reconstruction of Industrial Installations

Using Point Clouds and Images," Doctorate degree thesis of Delft University of Technology, 2006.

[10] Kazuaki Kawashima, Satoshi Kanai and Hiroaki Date, "As-built Modeling of Piping System from Terrestrial Laser-scanned Point Clouds Using Normal-based Region Growing," *Journal of Computational Design and Engineering*, Vol. 1, No. 1, pp. 13-26, 2014.

[11] Joohyuk Lee, Hyojoo Son, Changmin Kim, Changwan Kim, "Skeleton-based 3D Reconstruction of As-built Pipelines from Laser-scan Data," *Automation in Construction*, Vol. 35, pp. 199-207, 2013.

[12] Schnabel, R., Wahl, R. and Klein, R., "Efficient RANSAC for Point-Cloud Shape Detection," *Computer Graphics Forum*, Vol. 26, No. 2, pp. 214-226, 2007.

[13] Frédéric Bosché, "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*, Vol. 24, pp. 107-118, 2010.

[14] Erwin, Heine, 3D-Riskmapping: Training Tools for Advanced Terrestrial Laserscanning Application in Risk Awareness, Vienna, 2008.

[15] Rudolf STAIGER, "Terrestrial Laser Scanning Technology, Systems and Applications," *FIG Regional Conference*, Marrakech, 2003.

[16] Sylvie Soudarissanane, Roderik Lindenbergh, Massimo Menenti, Peter Teunissen, "Scanning Geometry: Influencing Factor on the Quality of Terrestrial Laser Scanning," *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 66, No. 4, pp. 389-399, 2011.

[17] Sylvie Soudarissanane, Roderik Lindenbergh, Optimizing Terrestrial Laser Scanning Measurement Set-up, *ISPRS*, 2011.

[18] Cousins, Radu Bogdan Rusu and Steve, "3D is Here: Point Cloud Library (PCL)," *ICRA Communications*, CA, 2011.

[19] Robert C Holies, Martin A.Fischler, A Ransac-based Approach to Model Fitting and its Application to Finding Cylinders in Range data, SRI Interational, CA, 1981.

[20] W. Sölken, "Definition and Details of Flanges - Types of Flanges," [Online]. Available: http://www.wermac.org/flanges/flanges_welding-neck_socket-weld_lap-joint_screwed_blind.html.

[21] *Piping Components and Fittings*, SOLAS, 2014.

[22] AVEVA, *AVEVA Marine Pipework Modelling, Training Guide.*

[23] AVEVA, *AVEVA Marine Piping Catalogues and Specifications, Training Guide.*

초 록

플랜트의 연한과 복잡도가 증가함에 따라 플랜트 유지보수 과정에서의 플랜트의 설계 모델의 활용도가 증가하였으며, 동시에 설계 모델과 실제 플랜트 간의 정밀한 정합도가 요구되고 있다. 그러나 실제 설계 모델은 유지보수 과정에서 요구하는 실제 건설된 플랜트와의 정합성을 제대로 반영하지 못하고 있다. 플랜트의 건설이 완료된 시점에서 스캔 된 데이터를 이용하여 설계 모델을 수정한다면, 요구되는 설계 모델과 건설 모델 간의 설계정합성을 요구되는 수준까지 끌어올릴 수 있다.

지상 레이저 스캐닝 기법은 수백만의 점들을 수초 내에 수 밀리미터의 오차로 스캔 할 수 있으며, 이러한 특성 때문에 건설 완료된 환경을 스캔하는데 매우 적합하다. 스캔 된 많은 수의 점들(점군)을 이용하여 수작업으로 CAD 모델을 재생성하는 작업은 시간과 비용이 많이 소비된다. 또한 전체 모델에 대한 재생성 보다는 설계 CAD 모델을 기준으로 허용 오차 이상으로 다른 일부 모델에 대한 수정만 필요한 경우가 대부분이다. 그러나 설계 CAD 모델과 실제 제품의 스캔 점군을 비교한 연구는 매우 적으며, 플랜트와 같이 크고 복잡한 제품에 대한 연구는 전무하다. 본 연구에서는 플랜트 의장에 대부분을 차지하는 배관시스템에서의 설계 CAD 모델과 건설된 플랜트로부터 스캔 된 점군을 비교하는 연구를 수행했다. 배

관파이프 부재의 존재 유무와 위치, 배관파이프 부재의 위치, 플랜지와 곡관의 존재 유무와 주요 위치를 CAD 모델과 스캔 점군 정보를 이용하여 찾는 과정을 자동화 하였다. 본 연구는 배관시스템 내의 티 부재를 찾는 데도 응용될 수 있다. 본 연구의 프로그램 구현 결과는 건설된 모델 재생성에 도움이 될 것으로 기대되며, 그에 따라 시간과 비용을 줄일 수 있을 것으로 기대된다.

본 연구의 결과를 통해 설치된 설계 모델의 파이프 부재중 중 81.65%가 검출 되었으며, 미설치된 설계 파이프 부재는 93.8% 검출되었다. 설치된 파이프는 설계 모델 기준으로 길이는 평균 1.08cm 오차 내로 측정되었으며, 설치 각도는 평균 0.74 도 이내로 측정되었다.

주요어: As-built Piping, CAD Model, Laser Scanning, Point cloud

학번: 2012-23990