Automated Pipe Spool Recognition in Cluttered Point Clouds

by

Thomas Czerniawski

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Master of Applied Science

in

Civil Engineering

Waterloo, Ontario, Canada, 2016

Author's Declaration

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

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

Abstract

Construction management is inextricably linked to the awareness and control of 3D geometry. Progress tracking, quality assurance/quality control, and the location, movement, and assembly of materials are all critical processes that rely on the ability to monitor 3D geometry. Therefore, advanced capabilities in site metrology and computer vision will be the foundation for the next generation of assessment tools that empower project leaders, planners, and workers.

3D imaging devices enable the capture of the existing geometric conditions of a construction site or a fabricated mechanical or structural assembly objectively, accurately, quickly, and with greater detail and continuity than any manual measurement methods. Within the construction literature, these devices have been applied in systems that compare as-built scans to 3D CAD design files in order to inspect the geometrical compliance of a fabricated assembly to contractually stipulated dtolerances. However, before comparisons of this type can be made, the particular object of interest needs to be isolated from background objects and clutter captured by the indiscriminate 3D imaging device. Thus far, object of interest extraction from cluttered construction data has remained a manual process.

This thesis explores the process of automated information extraction in order to improve the availability of information about 3D geometries on construction projects and improve the execution of component inspection, and progress tracking. Specifically, the scope of the research is limited to automatically recognizing and isolating pipe spools from their cluttered point cloud scans. Two approaches are developed and evaluated.

The contributions of the work are as follows: (1) A number of challenges involved in applying RANdom SAmple Consensus (RANSAC) to pipe spool recognition are identified. (2) An effective spatial search and pipe spool extraction algorithm based on local data level curvature estimation, density-based clustering, and bag-of-features matching is presented. The algorithm is validated on two case studies and is shown to successfully extract pipe spools from cluttered point clouds and successfully differentiate between the specific pipe spool of interest and other similar pipe spools in the same search space. Finally, (3) the accuracy of curvature estimation using data collected by low-cost range-cameras is tested and the viability of use of low-cost range-cameras for object search, localization, and extraction is critically assessed.

Acknowledgements

I am immensely grateful to Carl Haas and Scott Walbridge, for their generous mentorship, support, and friendship. Their editorial insights mark every page of this text. More importantly, their integrity and encouragement mark my life. Professor Haas is as remarkable as he is modest. He’s taught me the importance of taking initiative and forging my own path. His warmth and thoughtfulness gave me the courage to thrive in an environment of adversity. Scott’s candor and technical insight were refreshing and always welcome. His ability to ask the difficult questions without reserve made it a pleasure to meet with him.

Like Carl and Scott, Mohammad Nahangi interrupted his busy semesters to read my work and offer me substantial critical commentary. His enthusiasm for research was contagious and proved a reliable motivating force. My discussions with Chris Rausch, a real straight shooter, were invigorating and really encapsulated the spirit of academia. The conversations I have had with my friends Mark Tatangelo, Mark Charak, Yaman Nimer, and Alex Jegier were too few and far between, but it is through them I have come to appreciate the endlessly fascinating world outside the confines of the university.

Roya Cody, my loving girlfriend, with her unwavering devotion to academia, gave me the strength to persist. She constantly astounded me with what she was able to accomplish under deadline. My ability to get things done, I, in large part, attribute to her.

Finally, to my brother Bart, mother Alicja and father Andrzej; you provided me with an environment where a young curious mind could grow and learn. For that, I am eternally grateful. I know I’ve been somewhat worrisome, but I assure you, I work long hours because I enjoy doing it. Thank you for loving me. I love you.

Table of Contents

List of Figures

List of Tables

1.0 Introduction

The manufacturing sector in North America saw a massive shift from small scale enterprise to oligopolistic capitalism during the nineteenth and twentieth centuries. In 1800, a typical manufacturing operation consisted of a master artisan working alongside a few handicraft workers in the artisan's home. By 1880, economic growth and technological innovation created a system where manufacturers were likely to be owners or managers, supervising a workforce of hundreds or thousands of machine operators, all working in a multistory brick or stone factory powered by water or steam. With the increasing size and complexity of the industrial firm, the mechanization of production, and the subdivision of activities, ad hoc methods of scheduling, coordinating, and motivating the workforce were no longer sufficient. Diligent managers had to invent new management systems to assure adequate control of their production processes (Nelson 1975).

Brian Price (1987) wrote his doctoral thesis on Frank Gilbreth. In the beginning of the 1900s, Gilbreth was the owner of a building construction company in Boston that had a reputation for building things exceptionally fast. Gilbreth and his wife Lillian were influential social and human engineers at the forefront of a movement called scientific management. Their major contribution was the development of motion study. It involved systematic analysis and dissection of work processes resulting in faster work, not by forcing employees to work harder, but by eliminating unnecessary motions and creating improved work methods. Motion study was first applied to brick laying, typically reducing the number of motions from eighteen to four and a half exactly. This was achieved, in part, by placing the bricks and mortar on a raised platform on the scaffold so as to eliminate the motion of stooping, for example. The methods of analysis in motion study became more sophisticated over time and as technology developed. Within the Gilbreth's technical arsenal was the cinematographic cyclegraph method. A miniature electric light is mounted to a worker's hand as they execute a typical work process. Using time exposure photography, the movement of the light creates a bright line on a single time-exposed photograph, as can be seen in

Figure 1-1. This visualization, with its twists and turns, would reveal inefficient movement. The work process is modified until the shortest, smoothest lines are achieved.

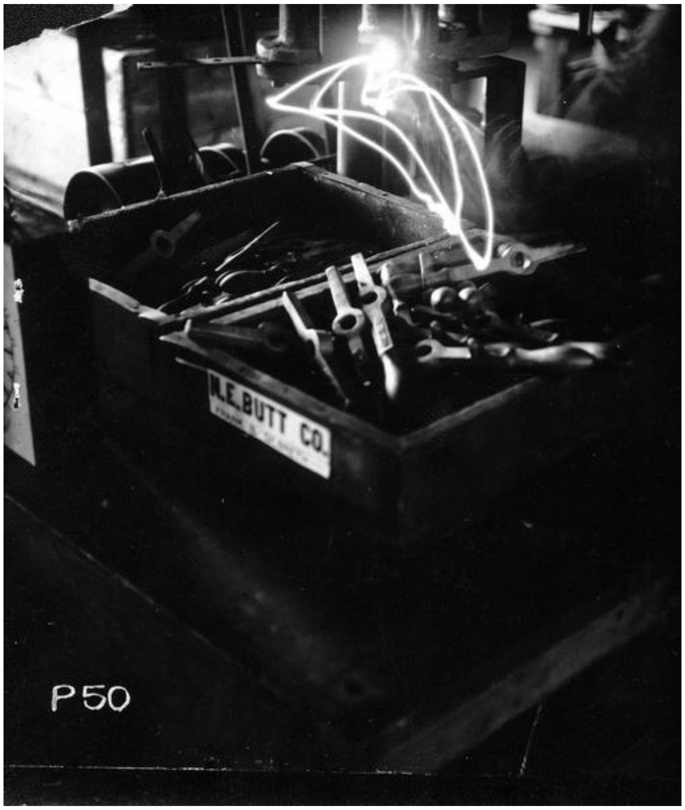


Figure 1-1: Cyclegraph of 2 cycles on drill press showing 'HABIT' positioning after transporting. Note the 'hesitation' before 'grasping' by Frank Bunker Gilbreth (1868-1924), retrieved from https://www.flickr.com/photos/kheelcenter/5279841396/ used under Creative Commons Attribution 2.0 Generic License https://creativecommons.org/licenses/by/2.0/

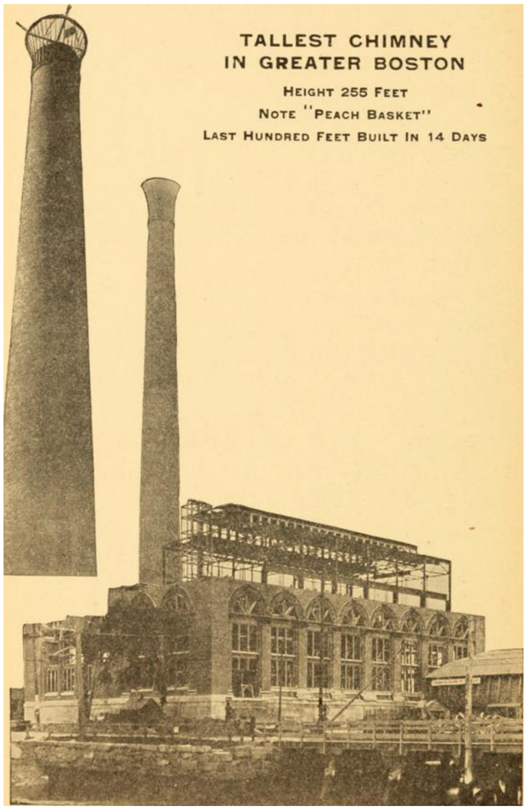


Figure 1-2: Tallest Structure in Greater Boston in 1901 - Field system by Frank Bunker Gilbreth (1868-1924), retrieved from https://www.flickr.com/photos... …/internetarchivebookimages/14580092249/ used under Creative Commons Attribution 2.0 Generic License https://creativecommons.org... /licenses/by/2.0/

Using motion studied construction methods and scientific management procedures, Frank Gilbreth's building company achieved some notable successes. For example, his company began the stonework for Prescott Hall in Cambridge, Massachusetts on March 27, 1896, completed the first floor on April 8, the second and third floor each in five days, and had the roof on by April 29. In 1901-02 Gilbreth constructed a power station and 255 feet tall chimney for the Cambridge Electric Light Company (Figure 1-2). The chimney was the tallest structure in the Boston area at the time, was begun on May 29, 1901, and completed on October 17 despite a total of nearly a month of work halts to let the mortar dry. In 1902, the contract for Lowell Laboratory for the Massachusetts Institute of Technology was signed June 28, work began July 1 and ended September 17, six weeks before the contract deadline. Finally, the cost of constructing a storehouse for the Plimpton Press in July 1911 had been lowered so much the Press decided to enlarge its size by forty percent and to give Gilbreth the contracts for two more storehouses. Gilbreth estimated that owners who applied even a fraction of his system could reduce costs by 10-20%.

Motion study was one of many powerful predecessor tools to modern lean management. The lean management paradigm is mainly attributed to the driving force of one production engineer at Toyota, Taiichi Ohno (Charron et al. 2015). Lean management focuses on the reduction of waste. The seven basic types of waste Ohno identified are: defects, overproduction, waiting, transporting, movement, inappropriate processing, and inventory. Shigeo Shingo worked with Ohno and was a student of Gilbreth's work. In Charron et al. an anecdote of Dr. Shingo provides insight into the lean management perspective,

> "The group stopped at a press that was forming metal. Dr. Shingo pulled out a stopwatch and watched the operation through one cycle. He asked the question: 'What is the percent of value added to non-value added?' One engineer said 100%-the worker was continuously working. Another said that it was 50%, and a third said 30%. Dr. Shingo looked at his watch, laughed, and said, 'Only 14% of the time is value added.' Dr. Shingo explained his calculation, stating that only when the press is forming the metal is real value added." (Charron et al. 2015) p. 49

A century ago, a paradigm shift took place in industrial production. Companies and facilities were growing in size and complexity at unprecedented rates, adopting mechanization, and subdividing skills and activities. These changes necessitated the creation of new management methods like motions study and lean management. Today, with advances in technology, building plants are more complex than ever, and the use of modularization and prefabrication is becoming more prevalent within industrial construction (Chandler 2013). These changes too, will require advances in management techniques; advances that will heavily rely on computers and sensors.

1.1 Problem statement

Among the many challenges of construction management are the awareness and control of 3D geometries. Progress tracking, quality assurance/quality control, and the location, movement, and assembly of materials are all critical processes that rely on 3D spatial data. The National Research Council instructs that improving the efficiency of these processes through effective interfacing of people, processes, materials, and information is one of the central opportunities for breakthrough improvement in construction (NRC 2009). Fiatech states in its construction and information management vision statement that sensor networks and communication technologies that provide seamless access to data, information, and knowledge needed for optimal decision making will be the key enabling tools in the management of project interfaces (Fiatech 2015). Integrating and automating the information flow regarding 3D geometries on dynamic and cluttered construction sites will require advanced capabilities in site metrology and 3D imaging, construction object detection and localization, data exchange, and design data to as-built comparison. These capabilities will be the foundation for the next generation of assessment tools that empower project leaders, planners, and workers.

3D imaging systems are a class of these sensor technologies growing in popularity. 3D imaging in the construction industry is often referred to as laser scanning and it has been profoundly affecting project surveying since the 1990s (Phair 2007). In addition to laser scanners, range-cameras are also a popular 3D imaging solution. Range-cameras use techniques such as structured light or time-of-flight to generate range images, which are 2D arrays (images) of depth values. 3D imaging devices enable the capture of existing physical geometric conditions objectively, accurately, quickly, and with greater detail and continuity than any manual methods. Current applications of laser scanners by construction firms include schedule and progress tracking (Turkan et al. 2012), creating complex as-built construction documents and 3D models (Patraucean et al. 2015), path planning, crane setup and clearance evaluation, quality assurance, retrofitting, controlling deformations, floor grading, steel column verticality, and base plate and tie point locations (FARO 2016; Jacobs 2006; Phair 2007).

Within the construction literature, laser scanning is being used as a source of data for advanced dimensional compliance control (Nahangi and Haas 2014). These new automated systems have proven potential for effectively and reliably detecting defects during complex fabrication and construction processes Figure 1-3. These tools compare as-built scans of fabricated assemblies with their 3D CAD design files. This comparison either confirms the assembly was fabricated correctly or identifies discrepancies.

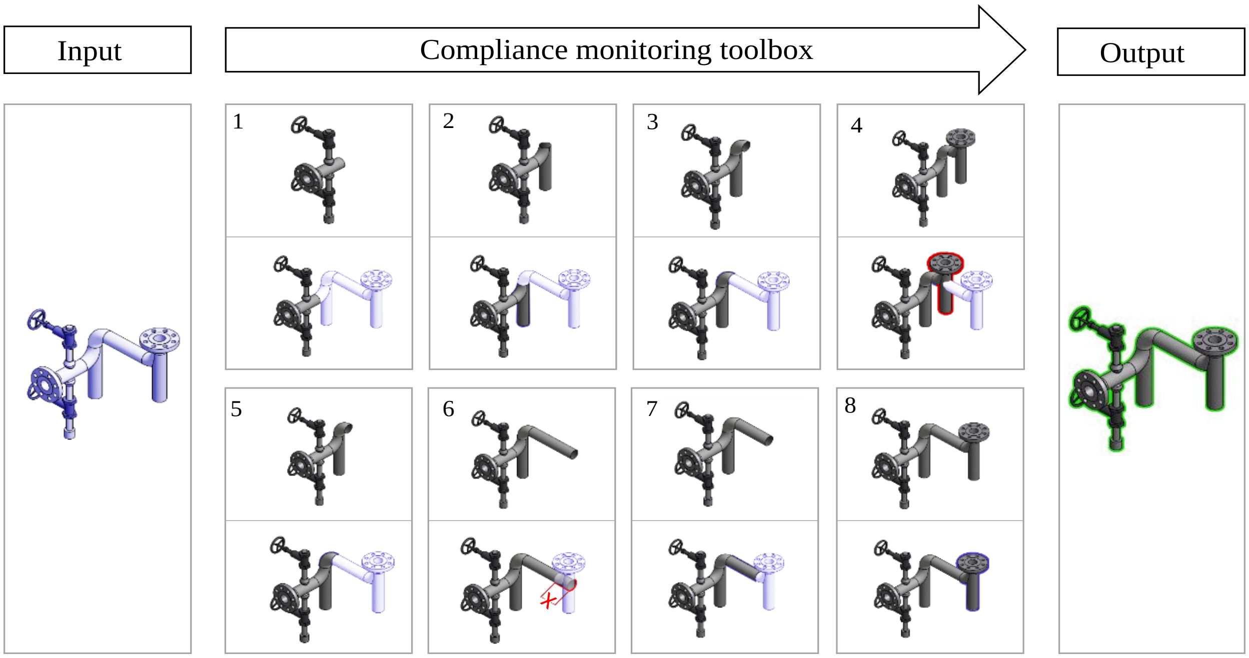


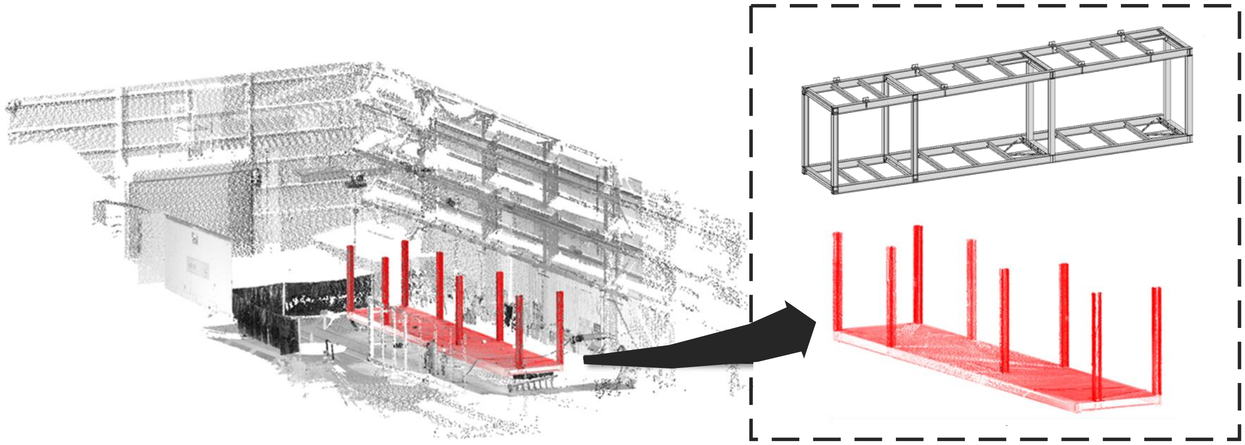
Figure 1-3: Example fabrication and assembly monitoring tool. The fabrication/assembly process is completed in eight steps as shown. The top row of each step is the fabricated state and the bottom row is the feedback provided by the dimensional compliance control system

Despite 3D imaging’s ability to provide massive amounts of spatial data, its potential is limited because extracting usable information from the collected data remains primarily a manual process. Before comparisons for dimensional compliance control can be made between as-built scans and 3D CAD design files, the particular object of interest needs to be isolated from the rest of the scan Figure 1-4. Manually extracting information from the raw 3D images in order to run analysis is painstaking, requires many person-hours and specialized personnel training, and is therefore not well suited for real-time or rapid decision making on a large scale. Automating the search and extraction of objects of interest from spatial data is the fundamental enabler for further developments in automated spatial analysis and information flow. Search and extraction involves many technical challenges that stem from the variability of spatial data and other operational realities such as local density, surface roughness, curvature, clutter, occlusion (Figure 1-4), missing/erroneous data, as well as range-sensor noise and inaccuracy. This thesis is an exploration of industrial pipe spool recognition and extraction from cluttered point clouds. Specifically, two questions are posed:

(1) What process can be used to automatically locate and extract pipe spools from cluttered point clouds?

(2) How does the accuracy of a 3D imaging device affect its ability to provide data from which objects can be successfully recognized?

These questions directly lead to the research objectives outlined in Section 2.5.



(a) cluttered points cloud

(b) comparison of as-built object of interest and design

Figure 1-4: Extracting points associated with structural frame module from cluttered 3D image of industrial fabrication facility for as-built object of interest to design file comparison

1.2 Thesis organization

This Thesis is divided into six sections.

Section 1 summarily explained the observable trends seen in management science over the past century and provided a brief problem statement and justification for the investigations conducted in this thesis.

Section 2 provides a comprehensive background on the current state of pipe spool fabrication. Topics of fabrication process and fabrication tolerance describe the current operational environment and frame the emerging automated dimensional control systems. Section 2.5 dictates the objectives, scope, and approach of the research within this thesis.

Section 3 is a literature review covering the current state-of-the-art in 3D imaging in construction, automated visual inspection, and industrial object recognition.

Sections 4.1 to 4.4 investigate two versions of a RANdom SAmple Consensus (RANSAC)-based pipe spool recognition framework that ultimately fail to provide reliable results. Sections 4.5 and 4.6 present and successfully validate a novel pipe spool recognition method that is based on local data level curvature estimation, clustering, and bag-of-features matching.

Section 5 presents an accuracy study for three 3D imaging devices that investigates the efficacy of using low-cost range-camera generated data for industrial object recognition.

Finally, Section 6 summarizes the findings of the research. Benefits and limitations of the proposed recognition frameworks are discussed, and potential avenues for further research are recommended.

2.0 Background

The Canadian construction industry contributes 120 billion dollars (2015) to the national gross domestic product, constituting 7.3% of the total. This is similar in magnitude to the health care and finance industries, at 111 billion and 114 billion dollars respectively (Statistics Canada 2016). The construction industry can be broken down into four industry segments (Halpin and Senior 2011): (1) heavy engineering construction, (2) residential construction, (3) building construction and (4) industrial construction. The term "industrial construction" is used to describe a wide range of facilities for basic industries, such as petroleum refineries, petrochemical plants, nuclear power plants (Figure 2-1), steel mills, and heavy manufacturing plants. Piping is the principle cost in these plants. The equally sizable mining, quarrying, and oil and gas extraction industry is the construction industry's second largest client, after the housing industry, ranked by capital expenditures for construction (Statistics Canada 2014).

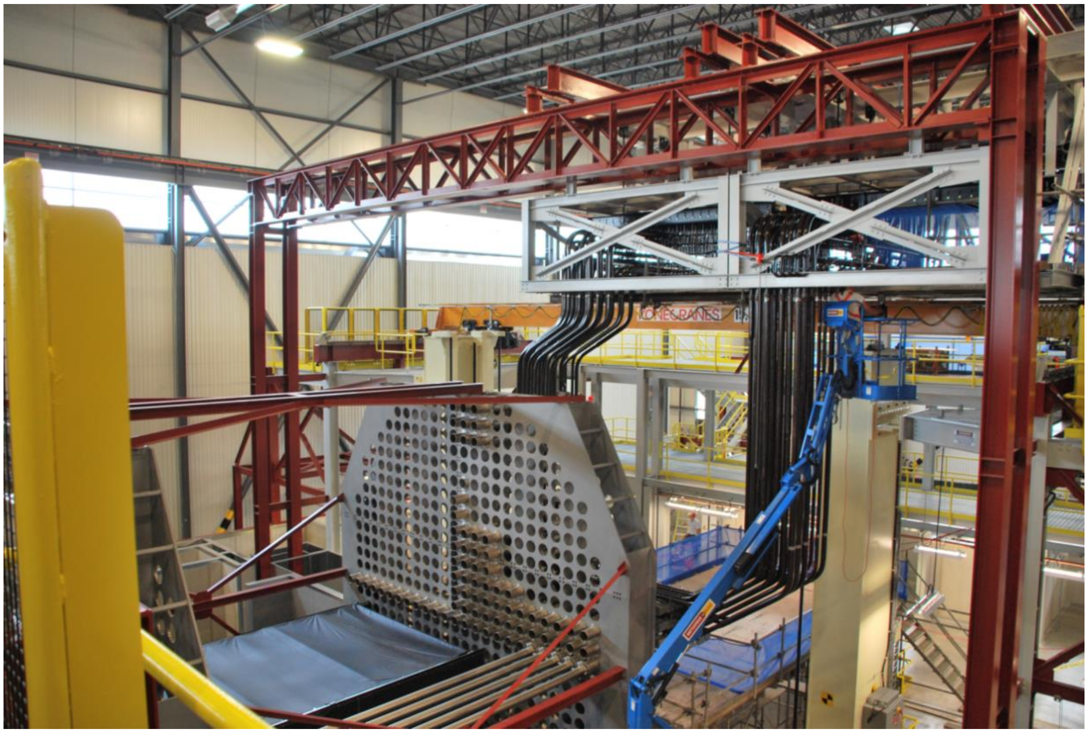


Figure 2-1: Nuclear reactor training mock-up with reactor face, fuel channels, and upper and lower feeders visible

2.1 Pipe spool fabrication

Pipe spool fabrication (Figure 2-2) falls under the category of industrial construction processes, and involves the creation of piping networks intended to carry water, steam, fluids, gases, chemicals, and fuel as part of heating, cooling, lubricating, and process piping systems. The prefabricated components of a piping system are called pipe spools. On a given project, pipe spool fabrication typically demands the largest percentage of direct labour (CII chartered Research Team 327 2015) and is commonly viewed as one of the more complex operations in project execution.

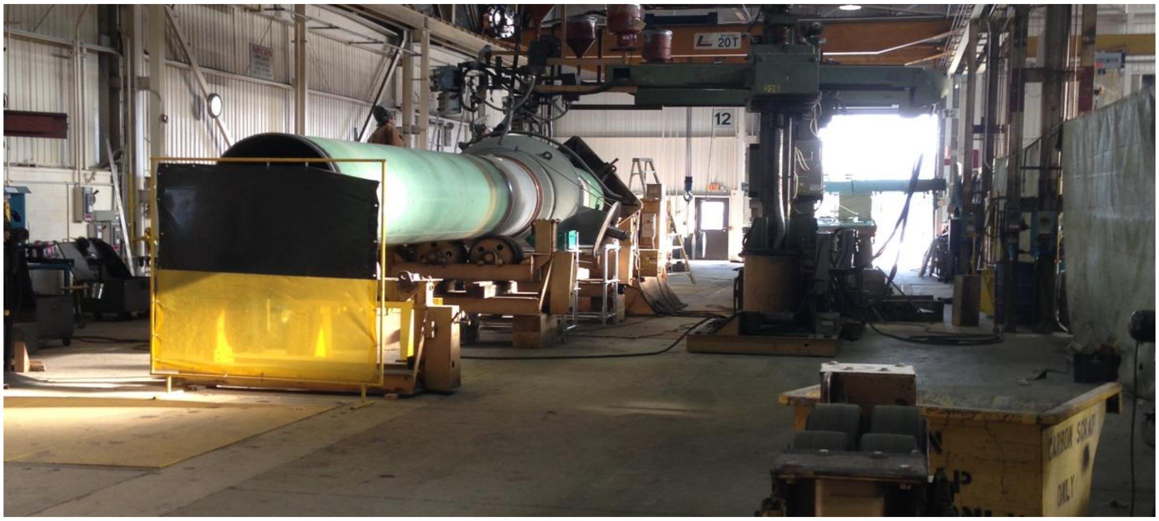


Figure 2-2: Large diameter pipe spool fabrication in piping fabrication facility, Cambridge, Ontario

The following sections are a summary and synthesis of information about the pipe spool fabrication process gathered from personal communications with Ricky Huynh, Cory Wilson, Andrew Giralt, Shaun Kauk, Chris Mullins, and Tom Elliot (personal communication, June 18, 2014 and November 18, 2014) during multiple visits to a pipe spool fabrication facility in southern Ontario, conversations with Ray Lemieux (personal communication, September 25, 2014) at a local pipe fitting union, and from reference material (Antaki 2005; Lucas and Welding Institute 1991; Nayyar 2000; ASME B31.1-2014 2014; Thielsch 1965). The presented process should not be interpreted as universal for all pipe spool fabrication as variations are possible.

Fabrication refers to the series of cutting, bending, forming, fitting, and welding of individual pipe components to each other and their subsequent treatment and examination to form a unit (pipe spool or piping assembly). This process may take place in a commercial pipe fabrication shop or an on-site fabrication shop, where subsets of the piping system are fabricated into subassemblies or modules. When built in commercial pipe fabrication shops away from their final installation site, this is known as off-site fabrication or prefabrication. These off-site fabrication shops typically offer a greater degree of control over fabrication quality and productivity than on-site fabrication. A summary of the process in BPMN notation can be found in Figure 2-3. Pipe diameters processed by these facilities range from between 2 inches to in excess of 48 inches. Commercial pipe fabrication shops often perform piping, as well as other architectural and structural metal work all within the same facility.

2.1.1 Pipe spool design and isometric drawings

Pipe spools vary by material, shape, configuration, type of joints, and many other properties. These are determined by the unique functions and loads of the intended application. Current industry practice is for the designer to prepare plans and sections or isometric drawings of the required piping system. These isometric sketches used by the craft workers are also often referred to as spool sheets, “Iso’s”, or shop drawings (Figure 2-4). An isometric drawing is the representation of an object in equal length projection showing length, width, and height axes. The axes in a true isometric drawing are 120° apart.

When complete, these drawings and auxiliary documents specify the: (1) routing and subdivision of the system into subassemblies, (2) number of field welds, (3) all necessary fabrication dimensions, (4) identification requirements, (5) code, classification, and inspection requirements, and (6) all special forming, welding, heat treatment, non-destructive examination (NDE), and cleaning requirements. How a system is divided into subassemblies depends on available materials, shipping limitations, heat treatment and welding clearance limitations, and occasionally scheduling directives. Minimizing field welds to simplify on-site installation work is often a priority, but must be balanced against added transportation and rigging costs. Once the designs are completed, they are sent to the fabricator along with the required

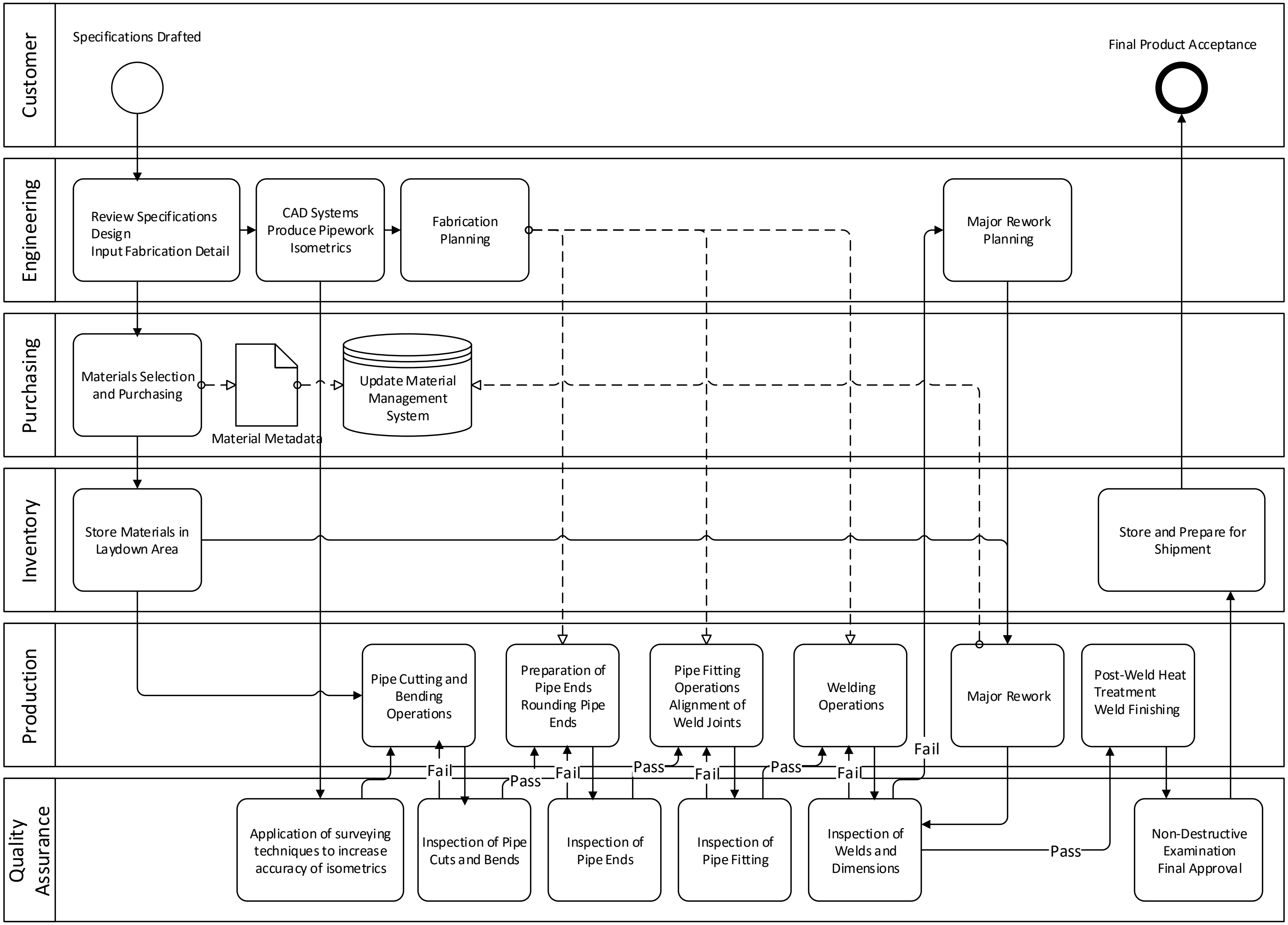
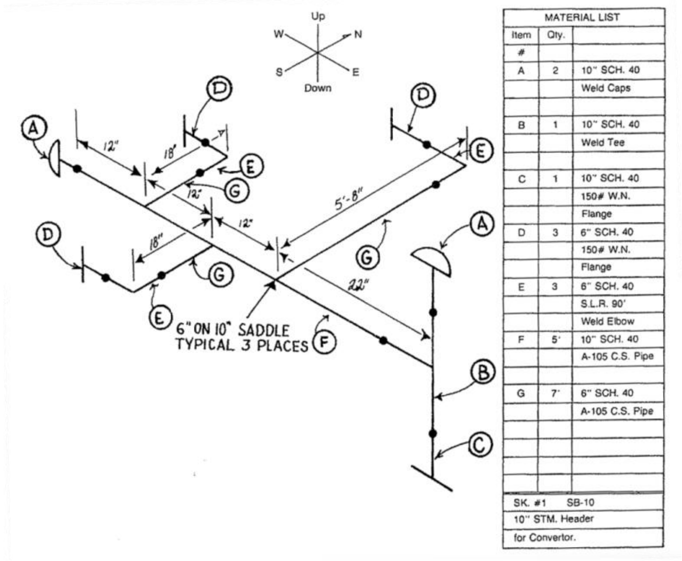

Customer
Engineering
Purchasing
Inventory
Production
Quality Assurance
Specifications Drafted
Final Product Acceptance
Review Specifications Design Input Fabrication Detail
CAD Systems Produce Pipework Isometrics
Fabrication Planning
Major Rework Planning
Materials Selection and Purchasing
Material Metadata
Update Material Management System
Store Materials in Laydown Area
Store and Prepare for Shipment
Pipe Cutting and Bending Operations
Preparation of Pipe Ends Rounding Pipe Ends
Pipe Fitting Operations Alignment of Weld Joints
Welding Operations
Major Rework
Post-Weld Heat Treatment Weld Finishing
Application of surveying techniques to increase accuracy of isometrics
Inspection of Pipe Cuts and Bends
Inspection of Pipe Ends
Inspection of Pipe Fitting
Inspection of Welds and Dimensions
Non-Destructive Examination Final Approval
Fail
Pass
Fail
Pass
Fail
Pass
Fail
Fail
Pass

Figure 2-3: Typical pipe spool fabrication process



MATERIAL LIST		
Item	Qty.	
#		
A	2	10" SCH. 40
		Weld Caps
B	1	10" SCH. 40
		Weld Tee
C	1	10" SCH. 40
		150# W.N.
		Flange
D	3	6" SCH. 40
		150# W.N.
		Flange
E	3	6" SCH. 40
		S.L.R. 90'
		Weld Elbow
F	5'	10" SCH. 40
		A-105 C.S. Pipe
G	7'	6" SCH. 40
		A-105 C.S. Pipe
SK. #1 SB-10		
10" STM. Header		
for Convertor.		

Figure 2-4: spool sheet and material list of a spool in a hot water heating system. From Drawing Interpretation and Plan Reading (pg. 46), by International Pipe Trades Joint Training Committee, Inc. 2000, Washington, D.C.

materials, such as pipes, flanges, fittings, and valves. Once all the material required for an isometric is available, that isometric is released to the shop for fabrication.

Drafting and input documentation for pipe spool fabrication is usually limited to two dimensional isometrics. The availability of 3D models on projects is still rare, but increasing. Anecdotally, when a 3D model exists, it is provided by the owner approximately 80% of the time, and made in house by the fabricator approximately 20% of the time.

2.1.2 Cutting, bevelling, and bending pipe segments

Typically, materials are supplied by the owner/client, sometimes the material comes precut, and sometimes it comes in bulk and needs to be cut in house. Cutting the raw piping material can be performed using mechanical methods such as saws, abrasive discs, lathes, and pipe-cutting machines; or thermal methods such as oxyfuel gas cutting or electric arc cutting. Cold mechanical cutting is the preferred method as it leads to cleaner connections and less thermal distortion during welding. After the pipes are cut, the cutting operators bevel the end surface of the pipes to make room for weld filler material if required. Bevels can be prepared using either the mechanical or thermal cutting methods. The V bevel is used in the vast majority of piping applications. Pipes are bent using a variety of methods using bending tables or bending machines with and without the application of heat.

2.1.3 Fitting

Then pipes are moved to fitting stations to be joined together by a steamfitter/pipefitter. First, a visual layout is produced, such as chalk lines and templates, that establishes the base line for locating and positioning the components and terminal dimensions of the subassembly. All weld surfaces are cleaned of rust, scale, grease, paint, and other foreign substances, which might contaminate the weld. Once cleaned, each weld joint is carefully aligned within required tolerances using alignment fixtures, spacers or jigs. Poor alignment usually results in a poor weld. Then the joint is tack welded (i.e., temporarily connected) to maintain the alignment. Once the pipes and other components of the spool, such as reducers, valves, and flanges, are fit, overhead cranes are used to move the assemblies to the welding station. Assemblies may move between fitting and welding stations several times as components are successively added to the assembly.

2.1.4 Welding

Welding (i.e., permanently fusing components to each other) constitutes the majority of work involved in the fabrication of modern piping systems. Currently the most commonly used welding processes for fabrication of piping are shielded metal arc welding (SMAW), submerged arc welding (SAW), gas shielded arc welding, gas tungsten arc welding (GTAW), Gas metal arc welding (GMAW), and flux core arc welding (FCAW). Other welding processes exist – e.g., robotic welding or friction stir welding (FSW) – however, they are used much less frequently in pipe spool fabrication. Welding is primarily performed via two methods: roll welding and position welding. In roll welding, the welder fixes one end of the pipe into a pipe turner or lays the pipe on a set of rollers and rotates the assemblies while welding them. The fitter or the welder does not change his or her position to perform the operation. The best efficiency in all shop welding processes is attained when the pipe axis is horizontal and the piece is rotated. Position welding is used when the pipes cannot be rotated by a turner or when components are not round in shape.

It requires the welder to maneuver around the object and weld in suboptimal positions. Position welding is a difficult procedure, and takes longer to perform than roll welding. Minimizing the number of position welds is one of the goals of pipe spool fabrication sequencing.

2.1.5 Miscellaneous post-weld processes

Based on the drawing requirements, after welding pipe, assemblies may be hydro-tested or undergo other processes such as stress relief heat treatment, surface treatment, surface finishing (sandblasting), cleaning, and painting.

2.1.6 Progress tracking

Production is tracked using Diameter Inches (DI), a metric based on the length of welding performed, scaled by size and thickness of pipe. When a spool is complete, the associated drawing is placed in a completed spool collection bin and the shop floor supervisor scans the barcode on the drawings and DIs are calculated and tracked automatically. Input labour hours and costs are recorded and correlated with the DI data to calculate and track productivity. Spool packages are tracked using a bar code system. Each drawing has a barcode and each drawing remains in physical proximity to the assembly during fabrication. The bar code provides a simple way to digitally track all projects. Information is tracked about the pipe spools on standard forms, and is organized within submission bins as seen in Figure 2-5. Forms and written procedures assure better control of fabrication processes.

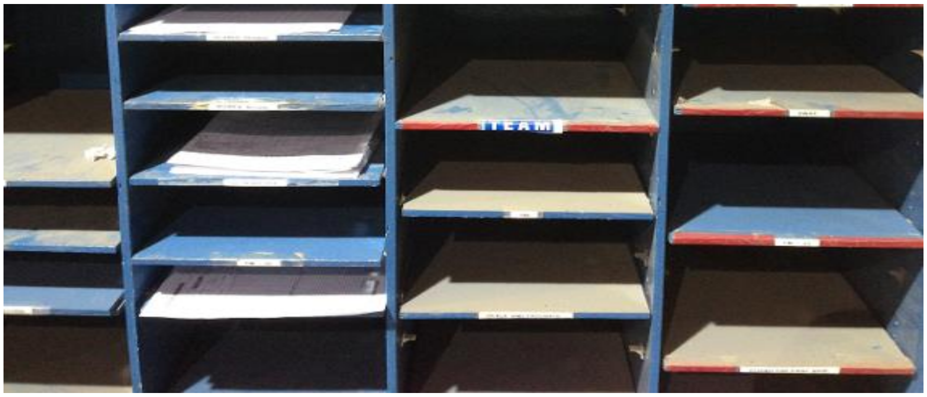
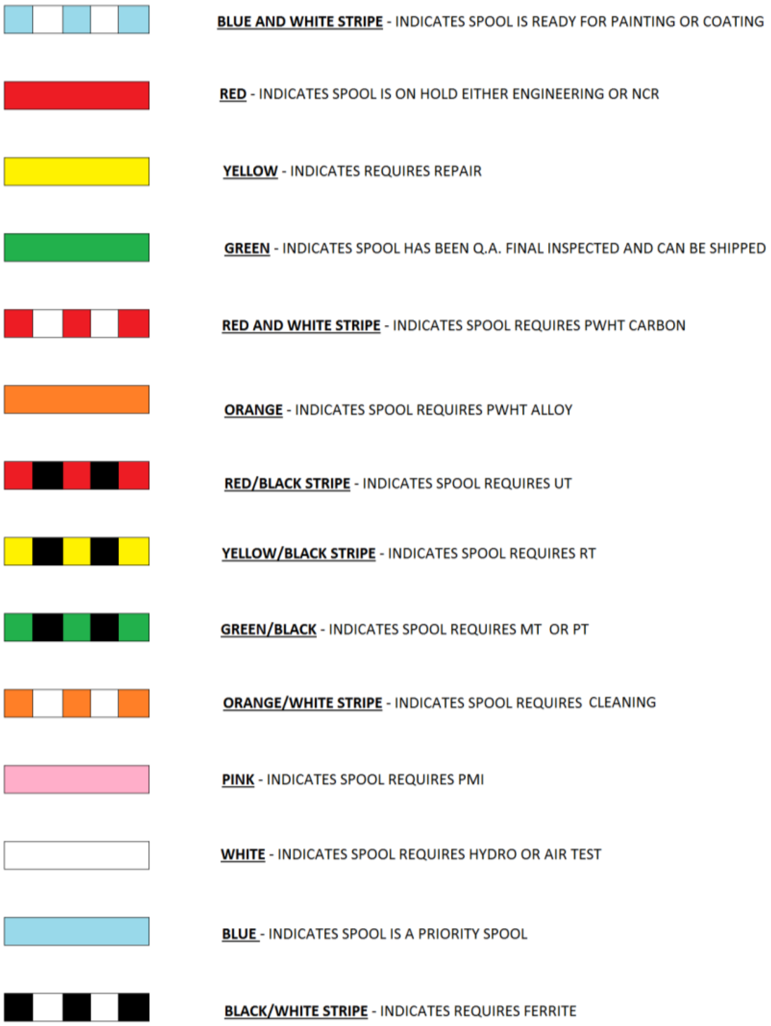


Figure 2-5: Pipe fabrication form submission and tracking bins

In order to control the flow of pipe spools through the fabrication process, it is typical for shops to use visual identifiers such as coloured ribbons (Figure 2-6), to identify the current state of the pipe assembly and any required special considerations. Example states include: spool on hold because of engineering on non-compliance report (NCR) issues, requires post-weld heat treatment (PWHT), requires ultrasonic testing (UT), requires radiographic testing (RT), requires magnetic particle testing (MT) or dye penetration testing (PT), or requires positive material identification (PMI).



BLUE AND WHITE STRIPE - INDICATES SPOOL IS READY FOR PAINTING OR COATING

RED - INDICATES SPOOL IS ON HOLD EITHER ENGINEERING OR NCR

YELLOW - INDICATES REQUIRES REPAIR

GREEN - INDICATES SPOOL HAS BEEN Q.A. FINAL INSPECTED AND CAN BE SHIPPED

RED AND WHITE STRIPE - INDICATES SPOOL REQUIRES PWHT CARBON

ORANGE - INDICATES SPOOL REQUIRES PWHT ALLOY

RED/BLACK STRIPE - INDICATES SPOOL REQUIRES UT

YELLOW/BLACK STRIPE - INDICATES SPOOL REQUIRES RT

GREEN/BLACK - INDICATES SPOOL REQUIRES MT OR PT

ORANGE/WHITE STRIPE - INDICATES SPOOL REQUIRES CLEANING

PINK - INDICATES SPOOL REQUIRES PMI

WHITE - INDICATES SPOOL REQUIRES HYDRO OR AIR TEST

BLUE - INDICATES SPOOL IS A PRIORITY SPOOL

BLACK/WHITE STRIPE - INDICATES REQUIRES FERRITE

Figure 2-6: Fabrication shop - ribbon identification chart

2.2 Fabrication tolerance

When a spool has been assembled, it goes through a series of verification activates. These can be generally categorized as inspection, non-destructive examination (NDE), and QA/QC. Inspection refers to the verification activities performed by stakeholders other than the fabricator, this being either the owner or a hired overseer. All other activities are usually performed by the fabricator. Inspection protocols are outlined in various industry codes and standards such as the ASME B31 Piping Code. Inspections may take the form of detailed visual examinations, witnessing of actual operations such as bending, welding, heat treatment, or NDEs, review or records, or a combination. The extent of inspection usually relies on the degree of confidence the inspector has in the fabricator.

Examination refers to the verification work performed by the fabricator, with the majority of inspection being NDE. Common NDEs are:

- Visual (alignment of surfaces, dimensions, surface conditions, weld profiles, markings, and evidence of leaks) (Antaki 2005)
- Radiographic testing (RT) (demonstrate integrity of welding)
- Ultrasonic testing (UT) (detect defects in welds and materials as well as determine material thickness)
- Liquid or dye penetration testing (PT) (surface examination)
- Magnetic particle testing (MT) (surface examination)
- Positive material identification (PMI) (ensure correct material usage)

The majority of verification activities are performed by the fabricator before the assembly is shipped. Once the assembly is received on-site by the customer, the assembly is assumed to be in a compliant state, so only minor visual checks for gross errors are performed.

2.2.1 Dimensional control

During fabrication, all materials and components are assumed to have specific dimensions and the locations of elements are dimensioned on the isometric drawings to a theoretically exact position relative to one or more datum points. In reality, these fabricated dimensions and locations vary somewhat. The contractually acceptable amount of variance is the tolerance for that specific measure. Tolerance specifications assure installation of a system within a reasonable degree of accuracy.

For specifying tolerances, designers and QC specialists typically use standardized dimensional tolerance guidelines (Figure 2-7) such as the systems published by American Society of Mechanical Engineers (ASME) or the International Organization for Standardization (ISO). For any contract, the designer must clearly state: (1) the tolerances allowed, (2) the standards used, (3) how compliance will be verified, and (4) what the result of noncompliance will be.

Currently, the predominant processes for monitoring the critical dimensions outlined in these standards involve manual assessment by certified QC personnel using direct contact measurement devices such as metal measuring tapes, calipers, custom gauges, squares, and straight edges. No measurement process is exact, and this uncertainty needs to be taken into account when verifying compliance with any allowable tolerance. Measurement uncertainty is simply a quantified doubt about the result of a measurement. Uncertainties and errors in measurement can come from many sources, such as the measuring device, the component being measured, the skill of the craftworker or inspector performing the measurement, the measurement process, and the environment. For example, when taking a simple distance measurement with a measuring tape, a number of things can go wrong: (1) the tape could be mismarked or the end hook could be out of position, (2) the tape sags during the measurement, (3) the tape may not be perfectly aligned with the desired axis of measurement, (4) the craft worker measuring might have poor eyesight, or be making the reading in dim lighting conditions, or (5) the craft worker may round, adopting a “close enough” attitude.

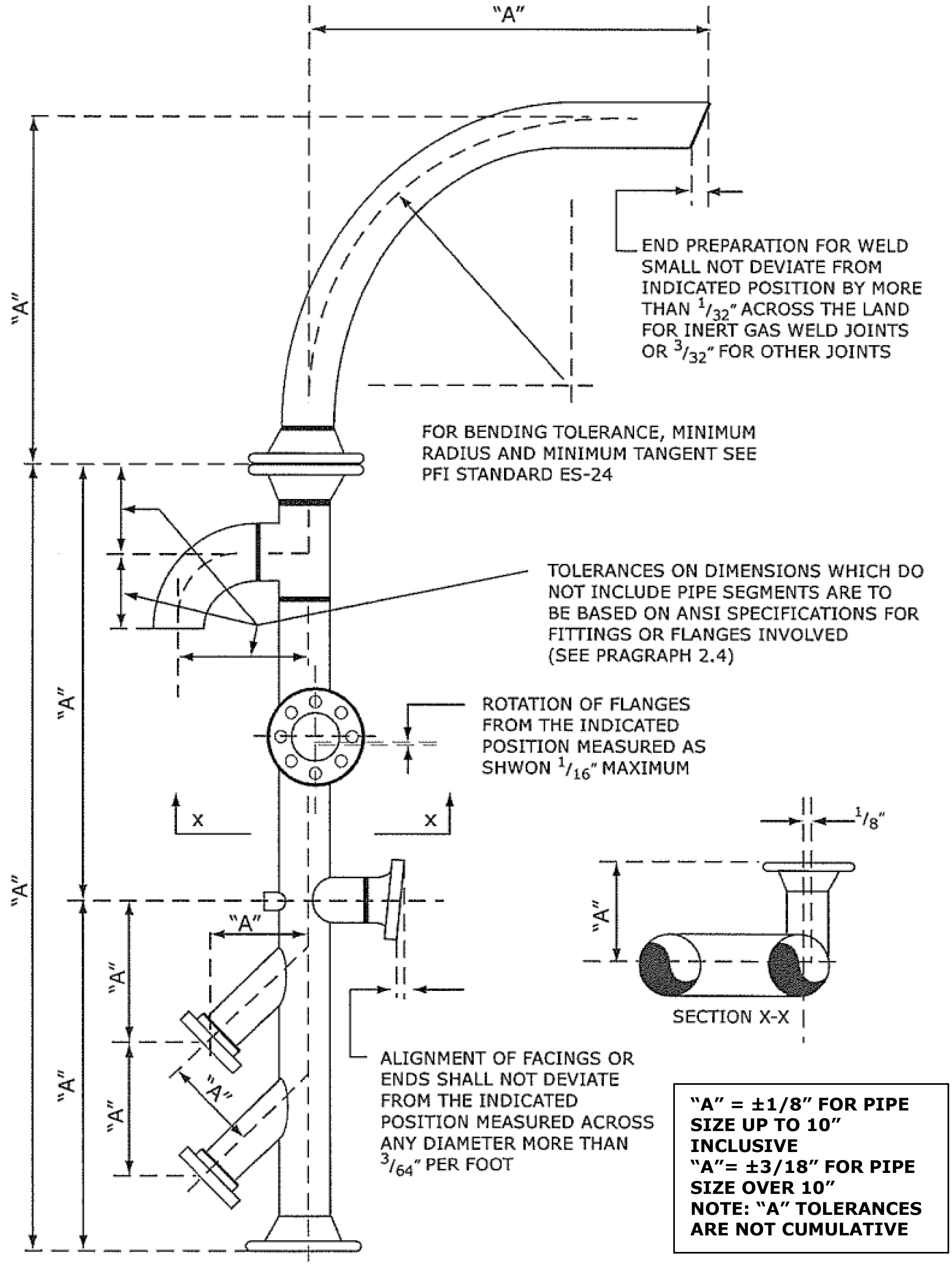

"A"
"A"
"A"
"A"
"A"
"A"
"A"
"A"
"A"
"A"
END PREPARATION FOR WELD SMALL NOT DEVIATE FROM INDICATED POSITION BY MORE THAN 1/32" ACROSS THE LAND FOR INERT GAS WELD JOINTS OR 3/32" FOR OTHER JOINTS
FOR BENDING TOLERANCE, MINIMUM RADIUS AND MINIMUM TANGENT SEE PFI STANDARD ES-24
TOLERANCES ON DIMENSIONS WHICH DO NOT INCLUDE PIPE SEGMENTS ARE TO BE BASED ON ANSI SPECIFICATIONS FOR FITTINGS OR FLANGES INVOLVED (SEE PRAGRAPH 2.4)
ROTATION OF FLANGES FROM THE INDICATED POSITION MEASURED AS SHWON 1/16" MAXIMUM
X
X
1/8"
SECTION X-X
ALIGNMENT OF FACINGS OR ENDS SHALL NOT DEVIATE FROM THE INDICATED POSITION MEASURED ACROSS ANY DIAMETER MORE THAN 3/64" PER FOOT
"A" = ±1/8" FOR PIPE SIZE UP TO 10" INCLUSIVE
"A"= ±3/18" FOR PIPE SIZE OVER 10"
NOTE: "A" TOLERANCES ARE NOT CUMULATIVE

Figure 2-7: Application of pipe fabrication tolerances from Pipe Fabrication Institute Standard ES-3

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

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

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

For a more in-depth discussion of tolerances in pipe spool fabrication and construction in general, see (Antaki 2005; Ballast 2007; International Bureau of Weights and Measures 1993; ISO 4463-1:1989 2012; ASME B31.1-2014 2014; Thielsch 1965).

2.2.2 Non-compliance with contractually stipulated tolerances

Occasionally, some work will be performed that upon examination will be found to be out of tolerance. This is an instance of non-compliance with the contract. Mistakes and defects during fabrication of pipe spools are caused by suboptimal examination practices; designers carelessly preparing specifications;

Table 2-1: Example of contractual non-compliance categories from industry

Non-Compliance Categories		Description
1	Weld defect	Flaws whose aggregate size, shape, orientation, location, or properties do not meet specified acceptance criteria
2	Dimensional defect	A dimensional discrepancy greater than the acceptable tolerances stated by the applicable code or contract
3	Drawing error	An error or omission on drawings used for production
4	Pressure test failure	The failure of a pressure test either hydro or pneumatic. Mechanical failures of equipment are not considered cause for an NCR but rather a retest
5	Fitting error	A dimensional or orientation discrepancy greater than the acceptable tolerances stated by the applicable code or contract discovered after the fitting process
6	Material defect (vendor error)	A defect in parent material which exceeds allowable tolerances and specifications of the applicable code
7	Missing MTR/Documentation (vendor error)	Required documentation and/or material test reports not available during receiving
8	Customer error	Customer error resulting in a non-compliance
9	Wrong material/consumable	Either material or consumables used in the process not as per the required specification or Welding Procedure
10	Wrong WPS used	Weld Procedure Specification used on production is not on the Approved Procedure List for the parent material welded
11	Damage part – general	Damage to production parts
12	PWHT error	An error either through documentation or during the Heat Treating process which is not acceptable per the applicable code
13	Machining error (by fabricator)	Machining completed by fabricator which falls outside that allowed by the applicable code or specification
14	Supplier/subcontractor error	Defect or error incurred by supplier or subcontractor which impacts compliancy of product to the applicable code and/or purchase order requirements.
15	Material not to specification (vendor error)	Material received either through purchasing or free issue does not meet the specification required
16	Painting defect	Defect in coating system which is not compliant to applicable code
17	Process non compliance	A violation of procedures or processes as stated in Traveler, inspection and Test Plan or QCPs
18	Contamination	Material has been contaminated either through contact or improper packaging and requires rework
19	Identification traceability error	Identification or traceability of components has come into question and cannot be positively linked to its accompanying documents
20	Regulatory non compliance	
21	MTR incorrect (vendor error)	Material Test Report is not in compliance either through error or omission with code stipulated
22	Customer supplied material NCR	Material issued by customer has Material Test Reports that are not in compliance either through error or omission with code stipulated
23	Damaged part – flange face	Flange face does not meet the requirements per the applicable code
24	Damaged part – ball valve	Ball valve does not meet the requirements per the applicable code
25	Welder not qualified	Welding personnel not qualified in accordance to the requirements per applicable standards
26	PO incorrect	Purchase Order supplied by fabricator does not meet the requisition
27	Material substitution	One material will be substituted by another
28	Miscellaneous	All other issues not covered in the above defect categories.

misplaced cost consciousness by the owner; and a fabricators eagerness to expand into performing work for which they lack the necessary experience in welding, metallurgy, quality control, or inspection. A list of non-compliance categories and their descriptions can be found in Table 2-1.

Dimensional non-compliance or defects are primarily caused by pipe fitter error or by the thermal expansion and contraction of metals during welding. The degree of distortion depends on the metal or alloy, its size, shape, thickness, the tacking and alignment, the welding process, procedure, and sequence, the care taken by the welder, and the positioning of the welded components in the design. Regardless of whether the defect originates from fitting or welding, dimensional examination usually does not occur until after welding, i.e., after the components have been permanently joined. Remediation of this type of non-compliance is expensive; more-so if the dimensional defect goes undetected until final installation on-site.

Client-side defect detection occurs when the contractor is installing pipe spools or modules in their final location and the interfaces or connection points of the assembly do not align with adjoining assemblies. Short of gross error, this misalignment can be characterized by 3 axes of translation (Figure 2-8) (i.e., centreline offset and face-to-face offset), and 2 axes of rotation (Figure 2-9) (i.e., flange out of parallel). Piping Code ASME B31.3 Paragraph 335.1.1 “Alignment” stipulates that a certain degree of misalignment can be remedied by cold-pulling or torquing the adjoining spools to bring them into alignment for joint assembly. This operation introduces a detrimental strain into the system, and so an engineer should determine whether the misalignment falls within a tolerance that is a function of: the inherent flexibility in the system, the length of pipe that lies in the direction perpendicular to the direction of a single degree of misalignment, and the other fit-up conditions in the balance of the system.

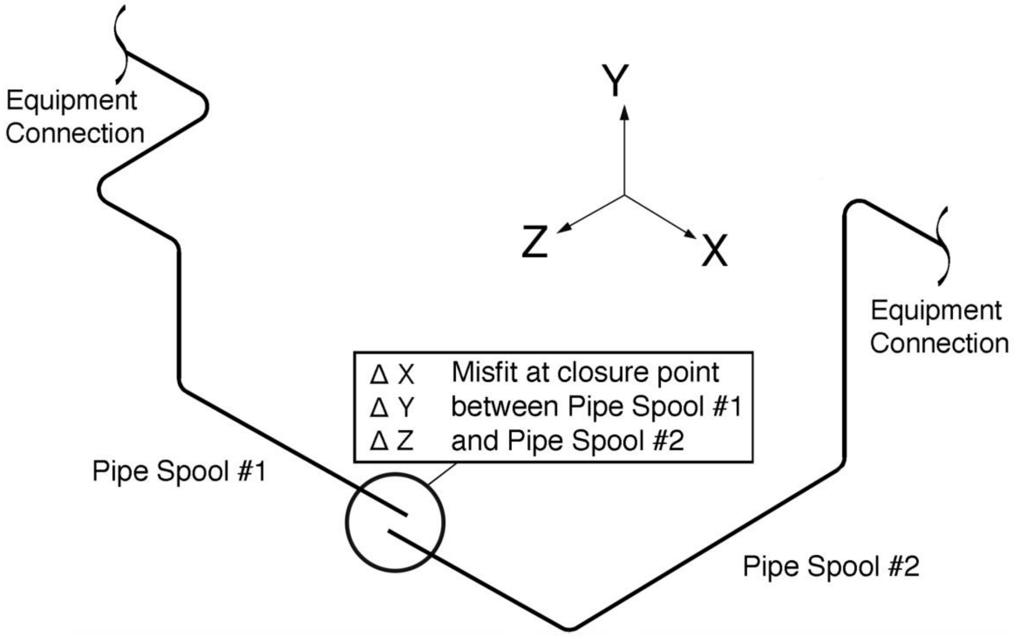


Figure 2-8: Adjoining spools' closure point translational interface misalignment. From ASME B31.3 Process Pipe Guide, Revision 2, p.125

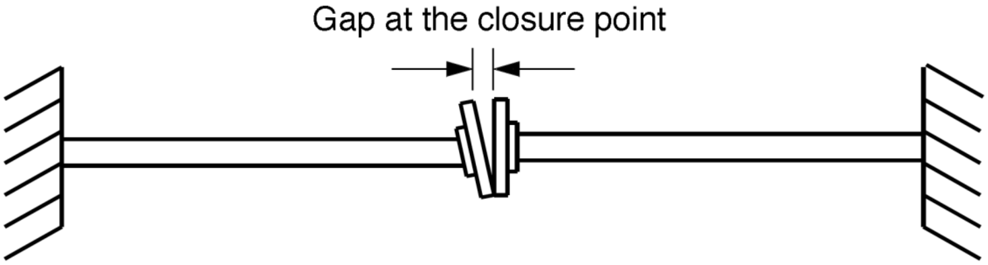


Figure 2-9: Adjoining spools' closure point rotational interface misalignment. From ASME B31.3 Process Pipe Guide, Revision 2, p.126

If the pipe closure point exceeds allowable alignment/cold-pull tolerance then the spool needs to be reworked. Depending on the nature of the fix, the component is either sent back to the fabricator with the instructions of the engineers or fixed on-site if the cost or liability is not prohibitive. This work can involve cutting, refitting, and rewelding, which must be followed with additional examination and inspection. Schedule delays may also result as industrial components are rarely interchangeable, and any delay in delivery and installation of such a unique component might hold back progress on-site.

Off-site facilities offer a greater degree of control to pipe spool fabrication as compared to on-site fabrication due to the more systematic and rigorous feedback control loops in place within these facilities. The biggest drawback is that occasionally after the spools are fabricated to a fixed size and shipped to site, they do not fit to previously constructed adjoining assemblies. This may be a result of poor workmanship, but more commonly attributable to misinterpreted isometrics or faulty dimensions on an isometric.

2.2.3 Rework

In the construction literature, rework is the wasteful effort involved in redoing work that has not yet yielded a product adequately conforming to contractual requirements (Hwang et al. 2009; Love and Li 2000). Rework directly and significantly contributes to cost and schedule overruns on construction projects (Hegazy et al. 2011; Love 2002). Specifically, research published by the Construction Industry Institute (CII) states that rework costs between 2% and 20% of a typical project's contract amount (Construction Industry Institute (CII) Research Team 252 2011). Using data from 178 construction projects, (Hwang et al. 2009) assessed the impact of rework from a contractor's perspective and concluded that it most greatly influenced cost increases on heavy industrial projects. It has been argued that the cause of rework on such projects is attributable to poor construction techniques and poor construction management policies (O'Connor and Tucker 1986). On a mining expansion megaproject in Alberta, it was discovered that errors and omissions in prefabrication and poor workmanship of prefabricated materials was a significant source of rework (Dissanayake et al. 2003). Systematic quality assessment of construction components during their lifecycle is important to reduce rework on projects (Love and Li 2000) and particular attention must be given to processes within prefabrication facilities to ensure they are meeting project requirements and mitigating field rework. Any automated quality assessment tools used for this purpose would need to have the capability of identifying errors and omissions in a timely and accurate manner, while using the most up-to-date design files, as many rework situations occur because field changes are not communicated to the fabricator effectively.

The method most widely employed to correct misalignment in vessel and piping components is the alternate heating and cooling of areas adjacent to the welds. Upon heating, the affected material expands with a corresponding reduction in yield strength, resulting in plastic flow if the heated area is restrained by the surrounding material. When the heated area cools, it shrinks, resulting in a permanent deformation. By repeating this procedure a number of times, the misalignment can be gradually corrected.

Rework is a substantial source of waste on construction projects, and systems for minimizing or eliminating it should be adopted. One such system is lean management.

2.3 Lean management

In the early 1950s, a Toyota Motor Corporation engineer named Taiichi Ohno created a systematized approach to think about process inefficiency and waste (Charron et al. 2015). His work developed into the Toyota Production System, which along with its derivative philosophies embodied in modern lean management, take scientific management to its natural conclusion. Rather than focusing on reducing time using time study, or reducing motion using motion study, lean management aims to reduce all forms of waste in productive systems. Lean management concepts have been applied to pipe spool fabrication (Wang et al. 2009), and the consequent modifications to shop layout can be seen in Figure 2-10 and Figure 2-11. The work cell layout reduces the transportation of pipe spools between fitting and welding stations, thus reducing motion waste. Waste is generally composed of unnecessary activities that do not add value to a product during its creation. "Waste can be broken down into eight forms: overproduction, excess inventory, defects, extra processing, waiting, inefficient motion, unnecessary transportation, and underutilized workers. Waste reveals inherent weaknesses in the current process in terms of capabilities and reliability." (Charron et al. 2015)

Rework in construction is an agglomeration of waste types, primarily resolving defects and any extra processing and transportation involved in doing so, and the inefficient motion that is the yet unquantified non-value add readjustment of pipes during normal fabrication processes. There is a need for new process control methods that reduce rework. The flow of information in the system is critical, and new methods of providing feedback to front line workers need to be developed in order to ensure that production remains in a continuous flow state, eliminating rework and ensuring actions are all value-adding.

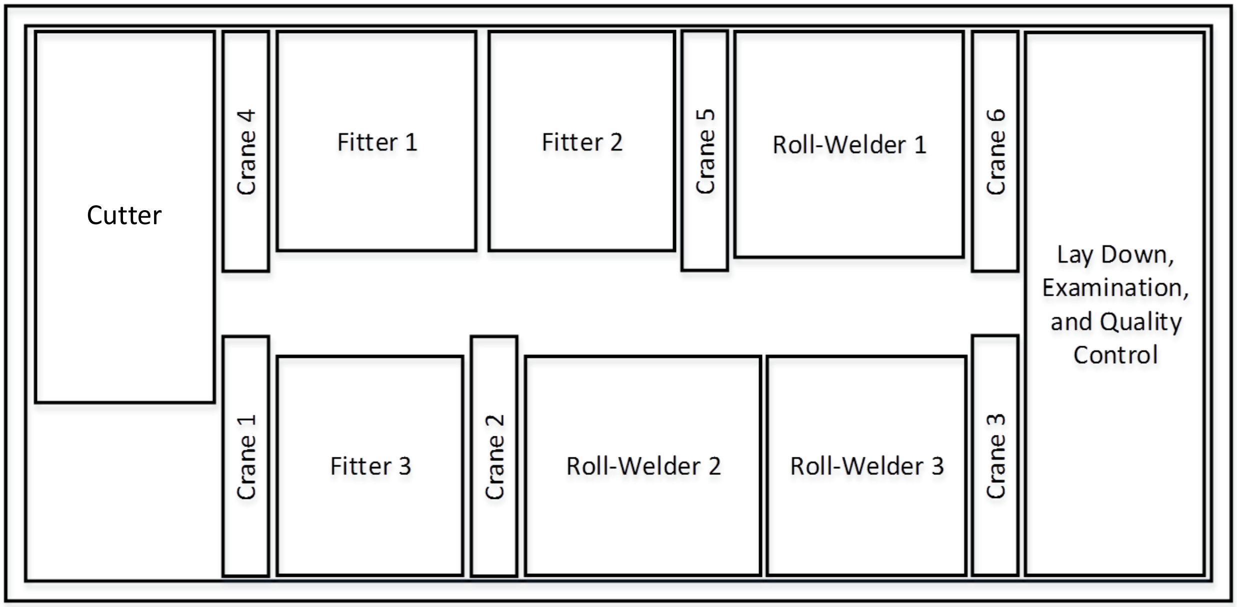


Figure 2-10: traditional pipe spool fabrication shop layout (Wang et al. 2009)

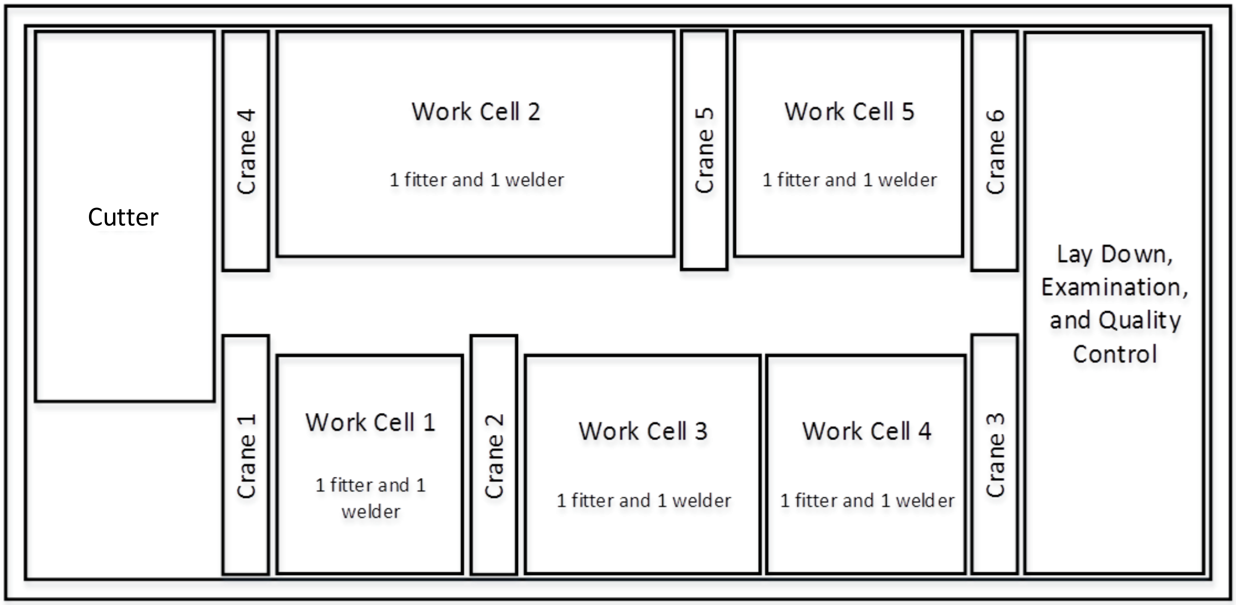


Figure 2-11: lean pipe spool fabrication shop layout (Wang et al. 2009)

2.4 Control systems theory applied to fabrication

Feedback has been described by Norbert Wiener as “the property of being able to adjust future conduct by past performance.” A feedback control system maintains a prescribed relationship of one system variable to another by comparing functions of these variables and using the difference as a means of control (De Silva 2009). Consider picking up and dropping a ball into a box. Here, the feedback control system involves the eyes capturing images of the motion of the hand through space, its relative position to the ball, and the performance of the grasping operation. These images are encoded and set to the brain for processing. The brain, knowing the intent of the motion can now compare what is expected (reference model) and what is actually happening (measured output based on visual sensor information) and control the movements of the muscles of the arm and hand in a way the difference between the reference and the output is always kept minimum. This is an example of a feedback control system. The difference between the reference and the data collected is called the error.

Figure 2-12 presents a standard error feedback control system configuration. The plant or the controlled system is the system that is controlled. The feedback element is typically a sensor that feeds the plant output back to be used by the controller.

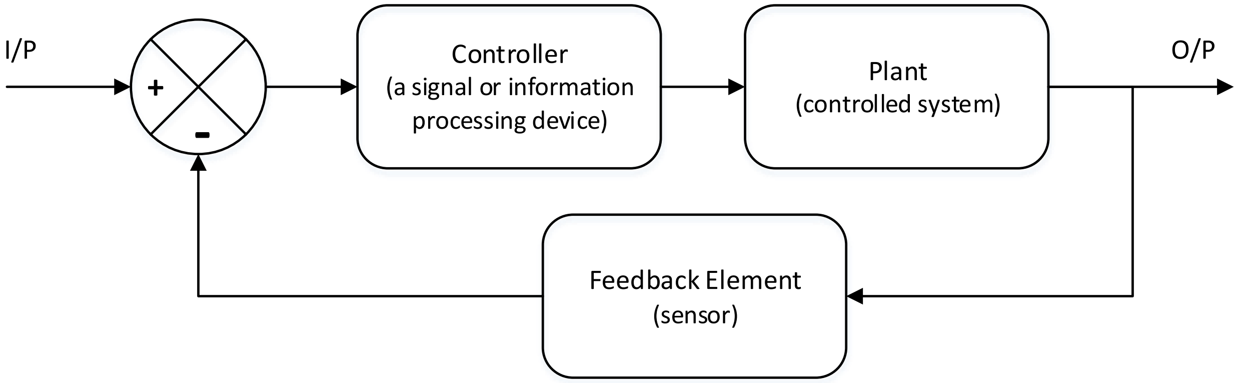


Figure 2-12: standard error feedback control system configuration

As construction fabrication operations become more complex, the biological feedback system outlined in the previous paragraphs becomes inadequate. In order to build lean companies, and reduce waste and rework, managers need to augment their eyes, gut feel, and manual note taking with sensor networks that automatically collect data about the plant and processes within. Manual data collection is untimely and subjective, and automated systems overcome these limitations. Although automated data collection systems are underutilized in construction fabrication, they are deployed extensively in manufacturing to track machine operation. A modern data collection system is composed of a data logger / transaction manager, a database, a report generator, and a factory viewer real-time interface (Wintriss Controls Group 2013). These programs typically operate together on a single server. The web browser acts as the front end to the system. The data logger is a program that gathers the raw production data, organizes it, and uploads it to the database. This raw data typically includes machine statuses: running, idle, unplanned downtime (with error codes entered manually by operator from a dropdown menu), planned downtime, changeover, offline. The database is then accessed by the web browser to display information in a format that is useful. The data collection system interfaces with the manufacturing plant's enterprise resource planning and manufacturing execution systems software to dynamically update schedules, calculate efficiencies to help reorganize production and allocate resources. Plant walk-throughs also benefit from the available display of real-time production data to focus the attention of management.

The tracking of machine statuses in manufacturing is much simpler then tracking the activity of craft workers in construction. The motion study methods of Frank Gilbreth focused solely on motion efficiency and required extensive manual processing, but were widely applicable to many types of work. The focus of this research and thesis will be to study how 3D imaging devices could provide useful information concerning as-built geometry in pipe spool fabrication environments.

2.5 Objective, scope, and approach of research

As the use of modularization and prefabrication becomes more prevalent within industrial construction, the scope of modularization will expand to include a greater diversity of systems and account for a larger portion of constructed facilities (Chandler 2013). As a result, the effective execution of prefabrication will play an increasingly central role in total cost and schedule management on construction projects.

Prefabrication errors and omissions are considered a significant source of rework (Dissanayake et al. 2003), and so have been the focus of many quality control tool development projects (Akinci et al. 2006; Bosché 2010; Kim et al. 2015).

Currently, dimensional compliance control is performed using direct contact measurement devices that are effective at evaluating whether basic assemblies are compliant with design specifications, but their effectiveness deteriorates as the assembly's geometrical complexity increases. Manual measurement is subjective, time-consuming, costly, and discontinuous. There is a need for automated and systematic dimensional compliance control tools that offer objective, fast, and continuous data collection for reliable quality control on industrial construction projects.

3D imaging systems are a class of sensor technologies well suited to provide data for these new dimensional compliance control tools. Current construction research applications of laser scanners are as diverse as schedule and progress tracking (Turkan et al. 2012) and automated compliance control (Nahangi and Haas 2014). However, extracting usable information from the collected data remains primarily a manual process. This is because the data capture is indiscriminate, and includes unwanted background objects and clutter in addition to the objects of interest. Automating the search and extraction of objects of interest from spatial data is the fundamental enabler for further developments in automated spatial analysis and information flow.

For the purpose of this work, the scope of the research is limited to pipe spools because of their significance to industrial construction. The objectives of the research follow directly from the problem statement presented in Section 1.1. The objectives are:

(1) Develop a process that can be used to automatically locate and extract pipe spools from cluttered point clouds

(2) Explore the potential of using low-cost range cameras for monitoring industrial pipe spool fabrication

Achieving these objectives will bring automated visual inspection of pipe spool fabrication closer to industrial implementation. An automated pipe spool recognition process will complete and fully automate the visual inspection workflow described in Sections 1.1 and 3.2 and the successful application of low-cost range-cameras to industrial inspection will decrease the required initial capital expenditures for fabricators in purchasing system equipment.

After performing an extensive literature review, two separate pipe spool recognition frameworks are developed and evaluated. RANdom SAmple Consensus (RANSAC) is a conceptually simple methodology that has achieved success in finding basic shape primitives in cluttered point sets (Section 3.3.4). In Sections 4.1 to 4.4, an attempt is made to extend and apply the RANSAC framework to locating pipe spools in cluttered point clouds. Then, in Sections 4.5 and 4.6, a novel method based on local data level curvature estimation, clustering, and bag-of-features matching is developed and validated. The techniques applied in the novel method were selected because pipe spools have unique curvature characteristics as compared to the majority of surrounding clutter. Finally, in Section 5.0, a study is conducted to explore the efficacy of using laser scanner, as well as low-cost range-camera generated data for industrial object recognition.

3.0 Literature review

Creating a dimensional compliance control system for pipe spool fabrication will require many of the components found in traditional manufacturing control systems. In feedback control, the controlled object has to be monitored and its response measured (e.g., using sensors) for feedback into a controller. The controller then compares the sensed signal with the designed or planned behaviour as specified externally, and uses the error between the two to generate corrective control signals (De Silva 2009).

For pipe spool fabrication, the externally specified design or planned state is detailed in the project's schedule, design drawings, specifications, and other contract files. A sensor or sensing system needs to be chosen that is capable of monitoring the project's state such that a meaningful comparison can be performed with the expected state as outlined in the contract. Once a sensing system is chosen, a way of extracting meaningful information from the raw data is needed. Ultimately, the calculated error between the sensed state and the planned state should initiate a control sequence that mitigates or eliminates the error.

The following literature review will begin with a study of 3D imaging systems and their application in the construction industry. Then, the current state of automated visual inspection is explored, and a gap in the technical execution of these processes is identified. Finally, a collection of concepts is outlined that will be combined and used in this thesis for developing a method of automated information extraction from sensed spatial data.

3.1 3D imaging in construction

Light detection and ranging (LiDAR) is an increasingly important technology from 3D computer vision used for metrology in the architecture, engineering, and construction (AEC) industry (Patraucean et al. 2015). LiDAR in the construction industry is often referred to as laser scanning and it enables the description of geometric conditions objectively, accurately, quickly, and with greater detail and continuity than any manual methods. Current applications of laser scanners by construction firms include schedule and progress tracking (Turkan et al. 2012), creating complex as-built construction documents and 3D models (Patraucean et al. 2015), path planning, crane setup and clearance evaluation, quality assurance, retrofitting, controlling deformations, floor grading, steel column verticality assessment, and base plate and tie point location identification (FARO 2016; Jacobs 2006; Phair 2007) (Figure 3-1).

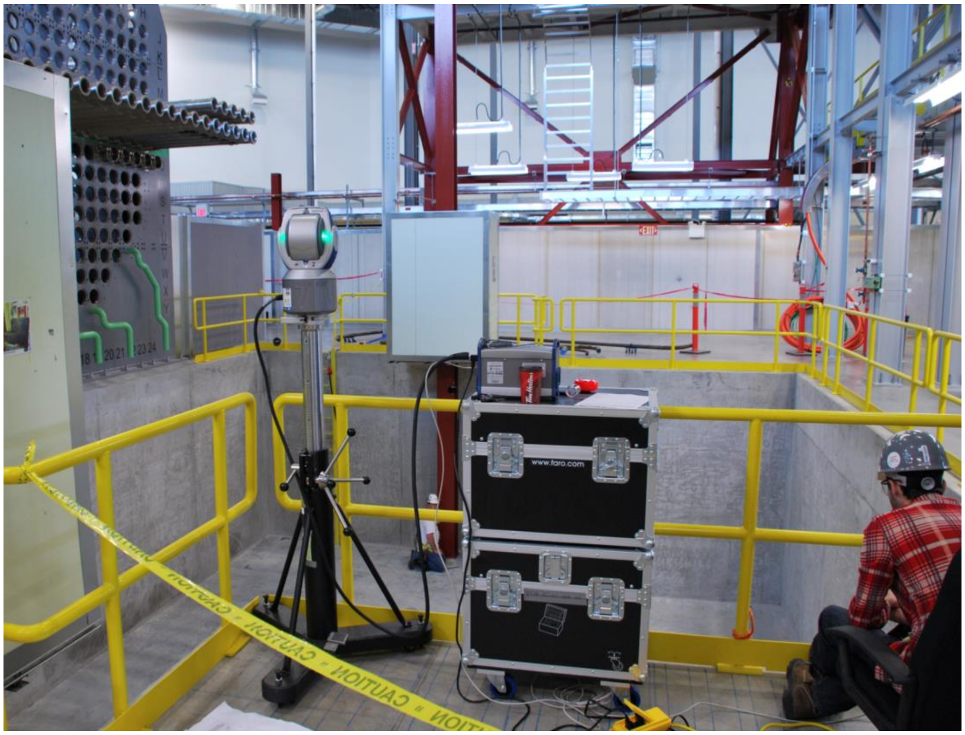


Figure 3-1: Survey and layout during construction using a laser tracker

3D imaging systems capture existing spatial conditions of a physical environment and generate digital representations such as point clouds or range images of their surroundings. Early iterations of range measurement devices were slow, provided only limited 3D information, and typically required substantial post-processing to provide useful information to the user. The recent emergence of non-contact active emission optical systems has substantially improved 3D data collection processes. Although these systems don't strictly conform to the traditional and basic definition of a sensor, they are referred to as range-sensors or sensors extensively in the literature. The imaging systems used in construction generally rely one of three technologies for inferring distance. Pulse time-of-flight systems use the time it takes for a laser pulse emitted by the device to travel from its transmitter to an observed object and then back to the receiver to calculate distance (Amann et al. 2001; Blais 2004). These devices can be used to measure up to several hundred meters with minimal degradation in accuracy. Phase-based systems calculate distance from the phase shift between the laser emitted by the device and the photoelectric current caused by the received laser (Amann et al. 2001). These devices are limited to collecting data at shorter distances (less than 100 m) but can achieve higher point density and faster data acquisition times (up to 976,000 data points per second (FARO 2015). Structured light systems implement a triangulation technology based on intersecting light rays in 3D space. They are similar to passive stereo camera systems, except one of the cameras is replaced by a projection device, either a single pulse or multiple emitters. The light emitted by the projector(s) uses a spatial or temporal coding strategy to derive correspondence between pixels in the camera and range values to objects in the scene (Drouin and Beraldin 2012). The captured data is stored in a 2D matrix of range values called a range-image, and commercially available low-cost optical range-cameras are capable of collecting these images at up to 30 frames per second (Microsoft 2015). Blais (2004) published a comprehensive review of range sensor development occurring between 1984 and 2004. A more recent survey of 3D reconstruction methods can be found in Gomes et al. (2014).

Despite the incredible capabilities of LiDAR laser scanners, they remain expensive and are not suitable for real-time applications due to their high data collection latency. For this reason, range-cameras, which are comparatively fast and cost-effective, although substantially less accurate, must be tested for purpose alongside laser scanners (Han et al. 2008). Microsoft’s first range sensing Kinect sensor was released in November 2010, followed by the release of Kinect v2 in 2013. The Kinect sensors are among the least expensive 3D imaging devices available in their product class. Kinect was designed as a human-computer game interface, but the sensor’s ability to capture 3D data has attracted the attention of researchers in 3D modelling and reconstruction. Kinect v2’s capabilities represent a major improvement over Kinect v1. A comparison of the two sensors can be found in Table 3-1.

Table 3-1: Technical specification comparison for   v1 and Kinect v2 (Gonzalez-Jorge et al. 2015)

Technical Specification	Kinect v1	Kinect v2
core depth sensor technology	infrared (IR) structured light for triangulation	IR indirect time-of-flight illuminator
depth resolution	320 × 240 (≈ 75,00 points) 30 fps	512 × 424 (≈ 200,00 points) 30 fps
field of view	57° horizontal × 43° vertical	70° horizontal × 60° vertical
measurement range	40 cm – 6 m	50 cm – 4 m

Results from a Kinect v1 sensor accuracy study concluded that: (1) Random error of depth measurements increases quadratically with increasing distance from the sensor and reaches 4 cm at the maximum sensing distance of 5 m. (2) The depth resolution decreases quadratically with increasing distance from the sensor. The point spacing in the depth direction (along the optical axis of the sensor) is as large as 7 cm at the maximum range of 5 m as can be seen in Khoshelham and Elberink (2012). Gonzalez-Jorge et al. (2015) concluded that although the precision of both the Kinect v1 and Kinect v2 decreased with increased scanning distance, the effect was less appreciable in the Kinect v2.

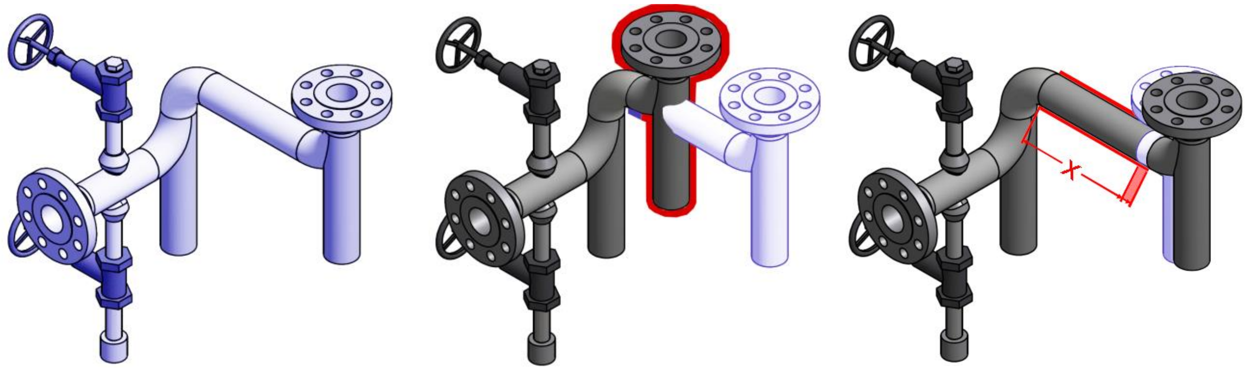
The low accuracy of the Kinect sensors remains prohibitive to practical application and laser scanners remain the technology of choice for dimensional compliance control. Section 5 presents a study that investigates the efficacy of using the Kinect sensors in industrial object recognition applications. In the next section, a review of automated visual inspection methods is presented, which covers a subset of the many emerging applications of these 3D imaging systems.

3.2 Automated visual inspection

Adhering to tolerance specifications during pipe spool fabrication requires rigorous inspection of work throughout the process. Laser scanners are well-suited for the inspection of assemblies for geometrical defects, which are a particular class of non-compliance. In the following sections, examples of geometrical defects encountered during pipe spool fabrication and operation are first provided. Then, the literature on automated visual inspection in construction is reviewed.

3.2.1 Geometrical defects in pipe spool fabrication

Geometrical defects are all unintended geometrical deviations of a pipe spool from the design intent. These generally, fall into two categories: (1) gross errors and (2) dimensional out-of-tolerance. Gross errors are mistakes during fabrication characterized by a misinterpretation of the design isometrics by the craft worker (Figure 3-2b). Dimensional out-of-tolerance refers to deviations during fabrication or distortions during handling or operation where a particular geometry or dimension in the design is not being adhered to (Figure 3-2c). In addition to these main classes of geometrical defects, Table 3-2 lists additional defects commonly observed in pipe spool fabrication, handling, and operation that have the potential to be monitored using laser scanning.



(a) 3D CAD design file

(b) gross error

(b) dimensional out-of-tolerance

Figure 3-2: Two classes of geometrical defects in pipe spools, (b) gross errors and (c) dimensional out-of-tolerance

Table 3-2: Defects with the potential to be monitored by laser scanner (Antaki 2005; Thielsch 1965)

Defect	Cause
gouges and dents	improper processing and storage; improper handling by cranes, chain falls, or other equipment
excessive ovality	improper cold bending
weld-defects	improper component fit-up; misalignment or spacing issues
thermal distortion	normal expansion and contraction of materials due to the thermal input from the welding process
creep damage	continuous increase in strain (deformation) of components under constant operating load at high temperatures

3.2.2 Automated visual inspection in construction

The function of QA/QC personnel is to perform lifecycle inspections to mitigate rework situations. Inspection is the process of determining if a product deviates from a given set of tolerance specifications. The predominant processes for monitoring the critical dimensions of an assembly involve a temporary production stoppage and manual direct contact measurement devices such as measuring tapes, calipers, custom gauges, squares, and straight edges. These processes can help fabricators evaluate whether basic assemblies are compliant with design specifications, but their effectiveness deteriorates as the assembly's geometrical complexity increases.

Automated inspection is desirable because manual inspection by humans is time-consuming, and can be excessively subjective, unreliable, and boring for humans to perform. Also, many industrial assemblies are not easily accessible for manual inspection. In addition to providing accurate dimensional evaluations, automated inspection processes would also automatically log defect rates, which would be an invaluable management tool. The use of CAD models in automated inspection started in the mid-1990s (Newman and Jain 1995a; Newman and Jain 1995b) and the natural utility of range data, which explicitly represents geometrical surface information critical for dimensional compliance control, also became evident. More recently, a methodology for using 3D imaging for quality control on dynamic construction projects was presented by Akinci et al. (2006). It focused on detecting defects early in the construction phase to eliminate costly rework downstream. The contribution of the work was the development of a formal method for comparing as-planned 3D design information (CAD model) with periodic imaging of critical construction components.

Building on this methodology, Nahangi et al. (2015) and Nahangi and Haas (2014) presented an automated approach for monitoring and assessing fabricated pipe spools and structural systems using automated scan-to-BIM registration. The method reliably detects the presence of dimensional non-compliance and has consistently quantified deviations with less than 10% error in experimental studies. The method requires two 3D imaging input files: (1) a point cloud of the as-built assembly generated using a 3D reconstruction technique such as LiDAR and (2) the tolerance specifications as represented by a 3D CAD design file. The files are input into an algorithm with the following three stages:

1. Preprocessing: involves converting the input 3D imaging files into a standard point cloud format and object of interest isolation from the as-built cluttered point cloud.
2. Registration: begins by importing the two input point clouds into a common 3D space. Since the input files do not share a common origin, they need to be aligned through a combination of course registration using principal component analysis and fine registration using iterative closest point (Besl and McKay 1992; Bosché 2012)
3. Dimensional non-compliance detection and quantification: analyzes the deviations between the superimposed files using a 3D cube local neighbourhood-based metric and outputs discrepancies.

Researchers have similarly created automated methods for monitoring and performing automated 3D image-to-BIM comparison of MEP systems (Bosché et al. 2014; Bosché et al. 2015), and general building and structural systems (Bosché and Haas 2008; Bosché et al. 2009; Bosché 2010; Golparvar-Fard et al. 2011; Nahangi et al. 2015; Tang et al. 2011; Turkan et al. 2012). Using 3D imaging for dimensional compliance assessment of construction components has proven potential to mitigate costly repair and rework while tracking progress.

Bosché et al. (2009) and Bosché (2010) developed an innovative method of inferring the presence of model objects in laser scans. Their approach compared as-built scans to simulated scans generated through a ray casting projection process (from the perspective of the scanner) performed using the 3D CAD model. The as-built scan and the synthetic scan are compared and a threshold is used for rejecting matching point pairs that are too far apart. Thus, the algorithm only recognizes and retains points in the as-built scan that coincide with the 3D CAD model objects. However, the process is limited by the initial coarse registration of the scan and model coordinates, which was performed manually. Turkan et al. (2012) employed Bosche’s algorithm to recognize built objects in scan data. The recognized objects were compared to the objects in a 4D as-planned model - a fusion of 3D CAD and schedule information - to track progress on construction sites. Golparvar-Fard et al. (2011) developed a machine-learning-based method for tracking construction progress using unordered photo collections. The photos are used to generate a 3D as-built model using scale-invariant feature transform (SIFT) (Lowe 2004) feature matching and then converted into a voxel representation (Section 3.4.1). The voxel's are traversed and labeled for occupancy and visibility using a combination of SIFT point detection and voxel reprojection back on all images that observe it to determine consistency among images. Pixel colour inconsistency indicates that the images are not observing the same object in the specified pixels and therefore no object is within the specific voxel. Occupancy and visibility information of voxels is then used to compare as-built data to the as-planned model. The process is similarly limited to Bosché's in that the initial model to as-built registration is done manually, but once the registration is complete, the locations of the voxel objects can be implicitly compared to the as-planned model objects because they share their coordinate space.

3D image-to-BIM comparison requires the superposition of the BIM onto the object of interest within the 3D image, i.e., registration of the object centred coordinate systems. However, unwanted clutter in the 3D image makes automated registration a challenge. Within the construction literature, this initial registration step has, predominantly, remained a manual process and must be resolved before the enormous amount of geometric data that 3D imaging makes available can be fully utilized. Visual inspection is typically concerned with a particular object in the scan data, but methods of geometrical analysis of these objects of interest fall short of full automation because of the absence of a reliable object recognition method. Only by fully automating these systems will they become unobtrusive enough for fabricators to adopt them. Once adopted, they will continuously collect data that can be mined for operational insights that will improve fabrication efficiency and mitigate rework.

3.3 Automated recognition of objects in three dimensional data

The problem of automatically extracting desired as-built data from cluttered scenes is closely related to 3D object recognition in the computer vision literature. 3D object recognition is the process of detecting the presence of an object in a captured image with similar characteristics to a reference image or model and mapping the 3D coordinates of the reference to the 3D coordinates (or world coordinates) of the detected object in 3D space (Brown 1992; Hoiem and Savarese 2011). Object recognition systems were used in the 1980s by automobile manufacturers for guiding welding robots and by electronics and microelectronics companies for assembly of small components and usually had very specific and limited applications (Horn 1989). Examples of early detection techniques include ellipse fitting to sparse range data for coarsely locating tubes on a flat table (Grimson et al. 1993) and using a black glove and black background to isolate and digitize non-black objects (Rusinkiewicz et al. 2002). Bin-picking robots are an example of more recent technology utilizing 3D object recognition (Scape technologies 2008). The systems use numerical descriptors of intrinsic and extrinsic 3D object features to guide a robotic arm through its grasping task (Balslev and Eriksen 2010). The key to this technology is the use of local features, which are used to identify points with distinct characteristics. The same local features are detected for both the desired object reference model as well as for the as-built scan. Points collected off of the target object in the scene will yield feature points similar to those computed for the reference model. This point matching is used to determine the pose of the object in the scene. For a comprehensive survey of existing surface-feature-based 3D object recognition methods see Guo et al. (2014). The following sections outline concepts important for the development of the object recognition algorithms presented in Section 4.

3.3.1 Feature space

Finding a 3D object in a cluttered scene requires that object-centred coordinate systems be generated and aligned. This process requires recognition of object and non-object components, which is difficult if the position of the object is unknown and not entered manually. Given the complexity of 3D data, concepts from human visual perception as well as information theory and data compression have been adapted to make object recognition tasks possible. In 1954, Newcomb and Attneave explored the role of abstraction in visual perception. They stated that any sort of physical invariance (smoothness, coherence, correlation) whatsoever constitutes a source of redundancy for an organism capable of abstracting the invariance and utilizing it appropriately. Redundancy provides no additional information and is therefore an opportunity for simplification. Computing data compression schemes have exploited this opportunity to improve the efficiency of data storage and data transfer. These methods compress data by removing redundancy from the original data in the source file (Salomon 2007). The resulting file is an abstraction of the original that maintains the information of the original but is substantially more manageable. The equivalent concept in object recognition is the abstraction of 3D data into feature or descriptor space.

3.3.2 Shape descriptors

The tools used for this abstraction process in the context of recognizing objects in cluttered scenes, must demonstrate (Iyer et al. 2005; Körtgen et al. 2003; Tangelder and Veltkamp 2008; Yang et al. 2008; Zhang and Lu 2004): (1) discrimination between objects of dissimilar geometry, (2) insusceptibility to noisy data, (3) invariance under transformation and rotation, (4) conciseness and ease of indexing, and (5) the ability to perform partial matching, i.e., describe parts of a point set (object of interest) independently of the rest of the point set in order to enable recognition of those specific parts.

There are two general descriptor types, global and local. Global descriptors abstract a 3D point set by considering the point set in its entirety. An example of a global descriptor is shape distribution (Osada et al. 2002), which is simply the probability distribution function of distance between randomly selected point pairs that reside in the 3D point set. Global descriptors work well for global-to-global matching, i.e., matching point sets representing single complete objects, but fail on the fifth requirement of performing partial matching. Therefore, global descriptors cannot abstract a cluttered scene in a way that allows a query object to be matched with the corresponding points in the scene, because the description of these points is contaminated by their connection with the rest of the scene. On the other hand, local descriptors abstract a 3D point set by considering the point set in subsets or regions. A popular subset type is the nearest neighbourhood, which is a collection of points in a spherical volume in 3D Euclidean space surrounding a query point. Determining the optimal nearest neighbourhood size to use for applying the shape descriptor is a critical problem in obtaining useful results from the abstraction process (Weinmann et al. 2015). Local descriptors allow for partial matching, and therefore, are ideal for object recognition in cluttered scenes.

For a detailed review of the state-of-the-art in shape descriptors see (Guo et al. 2014; Heider et al. 2012; Kazmi et al. 2013; Li et al. 2014; Li and Ben Hamza 2014; Tang and Godil 2012; Tangelder and Veltkamp 2008).

3.3.3 Bag-of-features

The bag-of-features (BoF) concept is largely inspired by the bag-of-words (BoW) (Blei et al. 2003) concept, which has been used in search engine text retrieval methods for quite some time. For illustrative purposes, the first ten entries in the BoW for Section 3.3.3 can be seen in Table 3-3. BoW aggregates word occurrences into a histogram called a “bag” and can be used to determine the similarity of text files. BoF is a generalization of this process that aggregates features into a finite, low dimensional, histogram that can then be used to determine similarity between objects. It was first adapted to image recognition in the seminal paper (Sivic and Zisserman 2003) and later applied to 3D non-rigid shape retrieval (Bronstein et al. 2011; Fehr et al. 2009; Li and Godil 2009) where it demonstrated impressive levels of performance. The object recognition process presented in Section 4.5 will utilize the BoF comparison framework.

Table 3-3: First 10 entries in bag-of-words for Section 3.3.3

feature	frequency of occurrence
the	9
of	8
in	7
a	5
and	5
bag	5
for	5
to	5
al	4
BoF	4

3.3.4 RANSAC

The RANdom SAmple Consensus (RANSAC) algorithm proposed by Fischler and Bolles (Fischler and Bolles 1981) is an iterative method used to find predefined models in noisy point data (Figure 3-3). The RANSAC algorithm assumes the input data is comprised of inliers (model of interest) and outliers (noise or unwanted points). Inliers are consistent with the predefined model, while outliers are distributed with a pattern that does not align with the model (MathWorks 2016). It begins by randomly drawing minimal point sets from the point data and constructing corresponding shape primitives. A minimal set is the smallest number of points required to uniquely define a predefined geometric primitive. For example, the minimal point set of a plane is a single point and a surface normal vector (Figure 3-4a). Or the minimal point set defining a cylinder is two points each with their respective normal vectors (Figure 3-4b). Each one of these minimal point sets uniquely defines the position and orientation of a shape primitive.

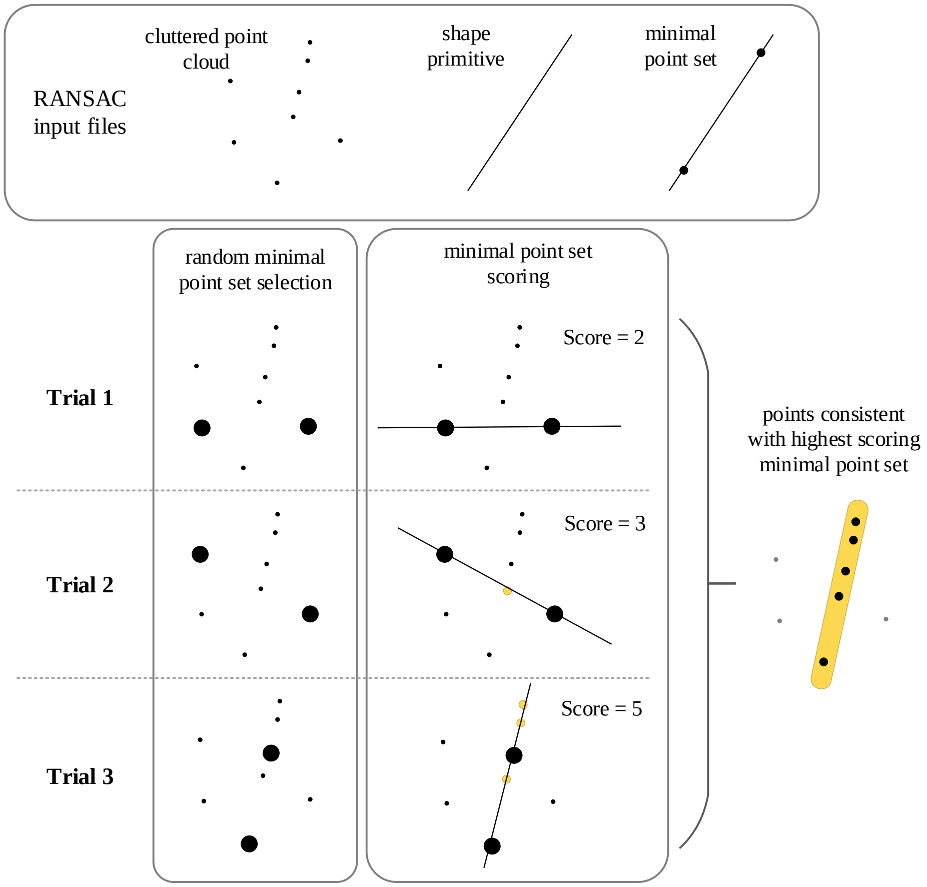


Figure 3-3: Basic RANSAC process for locating a line in a cluttered point set

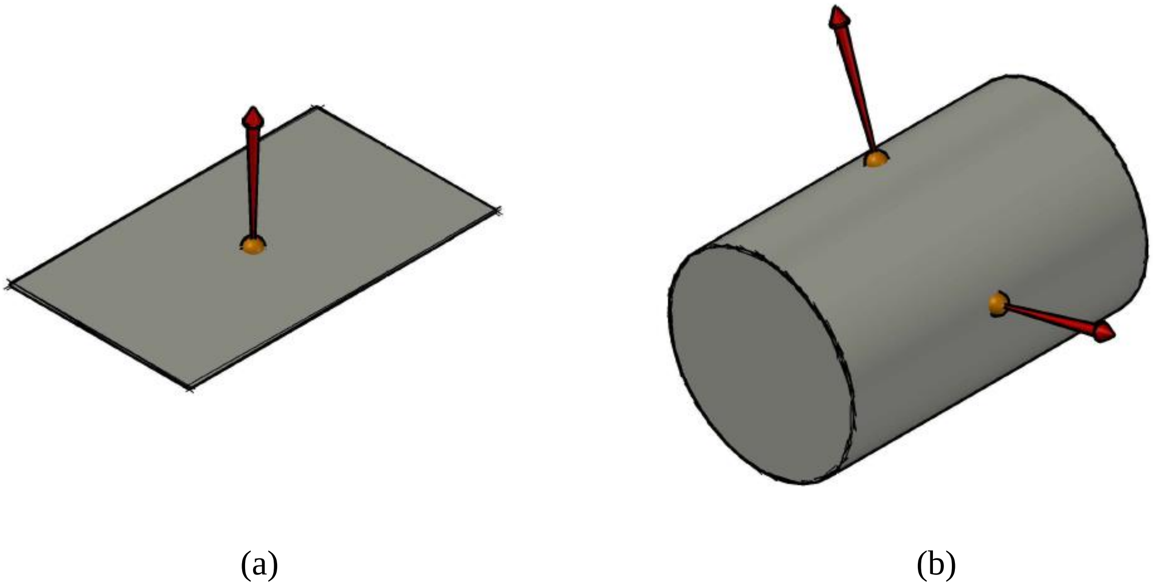


Figure 3-4: Minimal point sets for: (a) a plane and (b) a cylinder

These minimal sets are then evaluated by counting the number of points in the input data that are within some threshold distance of the shape primitive defined by the minimal point set. The shape primitive is considered detected by a minimal point set if the point set's score is above a threshold value. RANSAC is conceptually simple and very general and methods of managing the considerable computational demand of the RANSAC framework have been explored (Schnabel et al. 2007). The object recognition process presented in Section 4.1 will implement the RANSAC framework.

3.4 Object recognition in construction

The following section outlines a number of proposed industrial object recognition methods from point cloud data, with special attention given to pipe and pipe spool recognition methods.

3.4.1 Voxel methods

Voxel representations are widely used across many fields of computer-based modeling and graphic simulation. By dividing a 3D object, in point cloud form or otherwise, into an array of discrete voxels, subsequent computation can be simplified. Gilsinn et al. (2005) localized steel beams placed on a flat ground plane by characterizing voxels by their height above ground and adjacency. Voxels meeting specifications were included in a bounding box, which was then compared in size to the designed bounding box generated from a CAD model of the steel beam. Teizer et al. (2007) developed a more robust voxel-based algorithm, which used agglomerative hierarchical clustering on occupied voxels and compared the volume of clusters to reference objects. Erdõs et al. (2015) used voxelization and graph theory to segment and categorize piping objects in large-scale industrial point clouds, but were unable to categorize pipe junctions, T-sections or elbows. None of these studies validated their methods using point clouds with substantial amounts of clutter.

3.4.2 Circle fitting for pipe detection

Many researchers have reduced the problem of pipe detection to two dimensions by slicing a point cloud at pre-set intervals and searching these planes for circles. The method only works if the selected planes are perpendicular to the directions of the pipe runs. Ahmed et al. (2013) and Ahmed et al. (2014) assumed the pipe runs are orthogonal and align with the major axes of the building. Then for each plane, used the Hough transform to detect and "vote" on pipe locations. Liu et al. (2013) and Qiu et al. (2014) detected the primary pipe run directions using a Gaussian sphere and performed circle fitting on planes perpendicular to these directions using randomized Hough transform and RANSAC respectively. The method for determining the projection planes only works if great circles can be detected on the Gaussian sphere (where a great circle is the largest circle that can be drawn on the surface of any given sphere, and in the case of Gaussian spheres, is the result of projecting a cylinder's normal vectors onto a Gaussian sphere) which requires that the majority of objects within the scan be pipes running in a few set directions.

3.4.3 Training-based method

An innovative training-based method was presented by Pang and Neumann (2013) that used a local 3D Haar-like feature and an Adaboost training procedure for 3D object recognition. The initial detector produced a large number of false positive detections because the negative set used for training contained samples very different from the target object. However, the false positive detections were then used as a new negative set to retrain the detector in a process called boosting. The detector finally achieved precision of detection ranging from 40% to 100% for engineering objects, having the most difficulty with shape categories that have a very generic shape, which were easily confused with similar structures.

3.4.4 Spin-image matching

A local shape descriptor similar in nature to the one presented in this thesis is called the spin-image (Akinci et al. 2006; Johnson and Hebert 1999). Spin-images are a data level shape descriptor and are assigned to points by revolving a square patch about the point's normal vector and recording the configuration in which the surrounding points intersect the bins of the revolving square. Once spin-images are generated for points in the scanned scene and for points in the reference model, a correlation comparison of the spin-image shape descriptors is performed and high correlations indicate local surface correspondence and potential target object locations. Spin-images have been used for object recognition based on Kinect-like depth images (As'ari et al. 2014) and object recognition in terrestrial laser scan data (Date et al. 2012; Gordon et al. 2003). Spin-image-based object recognition has a high computational cost associated with the massive amount of image comparison involved. Potential target object locations are determined by comparing all spin-images of the model data with all spin-images of the scene data.

3.4.5 Curvature-based shape description

Measures of curvature play an important role in many shape analysis algorithms (Chua and Jarvis 1997; Gal and Cohen-Or 2006; Gatzke et al. 2005; Koenderink and van Doorn 1992; Salazar et al. 2010). Curvature-based descriptors have interfaced successfully with the efficient BoF abstraction framework (Li et al. 2014) and have demonstrated superior performance in object recognition when compared with other popular shape descriptors (Heider et al. 2012; Nagase et al. 2013). Curvature description has the added benefit of being spatially meaningful and therefore easily interpretable and understandable by the human user. In the real Euclidean space, curvature is defined as the rate of change of slope as a function of arc-length (Rosenfeld and Johnston 1973). Dealing with digital point sets, it is not immediately clear how to define a discrete analog of curvature. Researchers have tried to estimate various curvature values such as Gaussian, mean, and principal curvatures of point data by generating polygon models or by fitting parametric surfaces. Son et al. (2014) used nonuniform rational B-spline (NURBS) patch fitting (Piegl and Tiller 1997) on local nearest neighbourhoods to calculate maximum local curvatures as part of their method for segmenting 3D points corresponding to as-built pipelines in industrial laser scans. Collecting data using a Leica ScanStation C10, the framework achieved 100% precision and 100% recall for pipe identification, as well as a normalized mean radius classification error range of 2.74% to 3.68% for pipes of radius ranging from 76.2 mm to 304.8 mm. As impressive as the results are, the method's validation was limited to uncluttered datasets, and the choice to use parametric surface fitting as opposed to data level description negatively impacted computational efficiency.

Seibert et al. (2010) proposed a simple and efficient data level approach using conformal geometric algebra, which provides access to local curvature information within dense point sets without costly surface fitting or preprocessing. The directional curvatures at any point p on a smooth surface M describe all smooth curves on M containing p. The curvature estimation is reduced to the task of fitting a circle to each of a small discrete set of these directional curves. The minimum and maximum directional curvatures computed are called principal curvatures with corresponding orthogonal principal directions (Carmo 1976). This method was used by Dimitrov and Golparvar-Fard (2015) in a new region growing method for robust context-free segmentation of unordered point clouds, which capably segmented seven challenging point clouds of mechanical, electrical, and plumbing (MEP) systems. The object recognition process presented in Section 4.5 will make use of this curvature descriptor.

3.5 Data clustering

Data clustering is a central tool in the data mining and machine learning literature. The basic premise of clustering is to partition a set of data points into groups, which are as similar as possible. There are many clustering methods in the literature (Aggarwal and Reddy 2014) generally categorized into partitioning methods, hierarchical methods (Reddy and Vinzamuri 2013), and density-based methods (Ester et al. 1996; Ester 2013). These techniques have been applied to mining LiDAR data (Ghosh and Lohani 2013) and if applied to object recognition could potentially identify points representing the object of interest and partition away points, which are classified as clutter.

Density-based clustering is uniquely suited for this purpose because it is capable of detecting arbitrarily shaped clusters using the density of points as a guiding feature. Density-based spatial clustering of applications with noise (DBSCAN) (Ester et al. 1996; Ester 2013) is a popular method of density-based clustering and is applied to the object recognition framework presented in Section 4.5.

3.6 Plane removal

The search space in many industrial scans is largely comprised of massive planar objects (i.e., walls, floor, and ceiling). For cases where these planes are not the focus of the analysis, they clutter the search space and substantially slow the recognition of the object of interest. Therefore, quick removal of these planes before the object recognition process begins is desirable.

3.6.1 RANSAC

Basic RANSAC (Fischler and Bolles 1981) is comprised of two repeating steps: (1) minimal set selection and (2) minimal set evaluation. The minimal set for plane removal is a single point along with its normal vector, as this provides a complete description for a plane. RANSAC randomly samples minimal sets from the scan data, fits a plane using their description, and counts the number of points in the scan that are consistent with the fitted plane. After a given number of trials, a plane is considered to be recognized at the locations defined by the minimal set that achieved a score higher than a predefined threshold. Although basic RANSAC is conceptually simple, a direct application to plane recognition is computationally intensive. Methods for speeding up the RANSAC framework have been explored (Schnabel et al. 2007).

3.6.2 Hough transform

The general Hough transform (Ballard 1981) can be used to recognize planes within noisy data. It is comprised of three steps, (1) repeated transform mapping, (2) application of a “voting” rule, and (3) finding the shape parameters within the accumulated array of votes. Use of the 3D Hough transform for extraction of planar faces from point clouds has been investigated (Vosselman and Dijkman 2001). A randomized Hough transform is a variant of the 3D Hough transform that has proven to be especially effective for plane detection in point clouds (Borrmann et al. 2011; Xu et al. 1990).

3.6.3 Gaussian mapping

An elegant solution for identifying major planes within point cloud data includes mapping normal vectors to a Gaussian sphere (Liu and Xiong 2008; Wang et al. 2013). Each cluster on the Gaussian sphere represents a direction that is perpendicular to major sets of parallel planes. This technique for plane removal will be used to simplify the object recognition process in Section 4.5, because it is anticipated based on the literature to have the most combined potential for speed and efficacy.

4.0 Pipe spool recognition in cluttered point clouds

As stated in Section 1.1, the justification for the work presented in Sections 4.0 is that automating the process of extracting pipe spools of interest from point clouds (Figure 4-1) will be the fundamental enabler for further developments in automated industrial inspection systems.

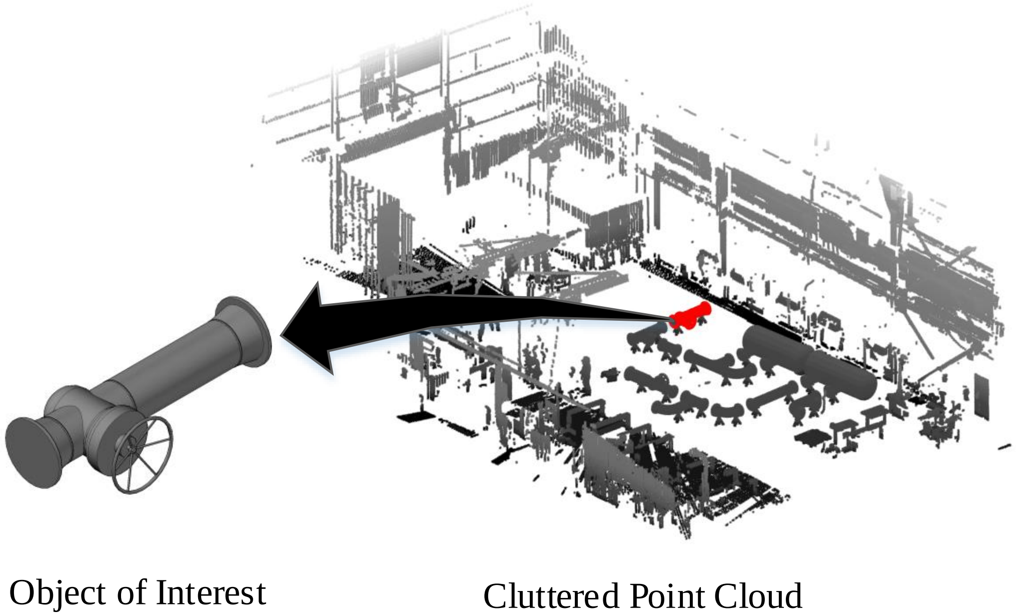


Figure 4-1: Extracting points associated with structural frame module from cluttered 3D image of industrial fabrication facility

The intent of the proposed methodologies is to automatically isolate the points in a cluttered laser scan that represent some specified pipe spool of interest. This isolation will allow for further analysis and inspection of the pipe spool's geometrical state. Figure 4-2 presents a basic conceptual illustration of the proposed methodology for isolating a pipe spool of interest from a cluttered laser scan.

The following section is broken into two parts: Sections 4.1 to 4.4 present and evaluate two RANSAC-based pipe spool recognition frameworks, and Sections 4.5 and 4.6 present and evaluate a novel pipe spool recognition method based on local data level curvature estimation, clustering, and bag-of-features matching.

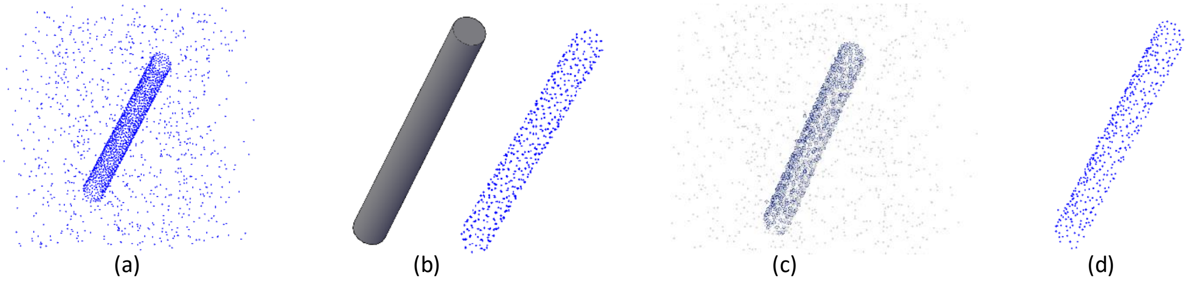


Figure 4-2: The conceptual framework for pipe spool isolation; (a) Laser-scanned point cloud is acquired, contains both the pipe spool of interest and surrounding noise and clutter; (b) A pipe spool of interest is specified using a 3D CAD design file; (c) The pipe spool is located in the cluttered laser scan; (d) The points representing the pipe spool are isolated from the cluttered laser scan.

4.1 RANSAC-based pipe spool recognition methodology

RANdom SAmple Consensus (RANSAC) (Fischler and Bolles 1981) is a general approach for primitive shape detection in cluttered point clouds. In general, it works by randomly selecting a set of points from a reference file describing the object of interest, selecting the same number of points from the cluttered laser scan, using a descriptor to check if the two sets are similar, and then recording a score for the set of points. Once a pre-set number of trials have been performed, the set of points with the highest score is chosen as the location for the object of interest. A summary of the method is presented in Figure 4-3. Each step of the method is discussed in detail in the following sections.

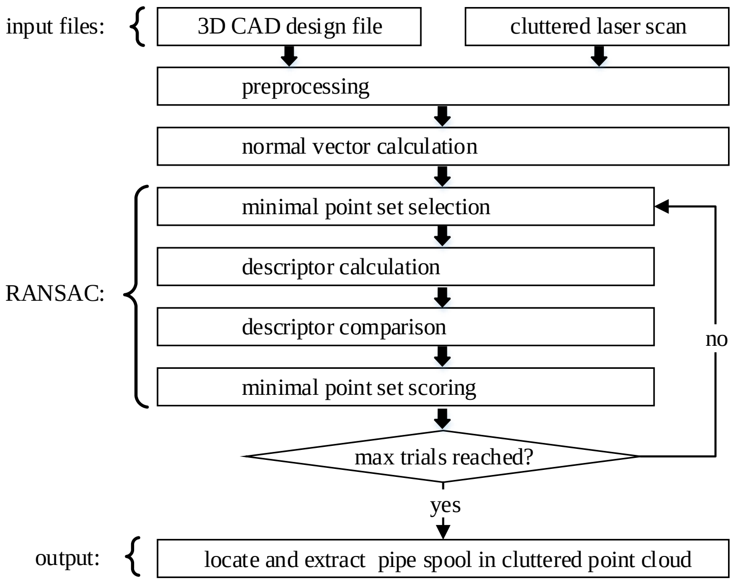
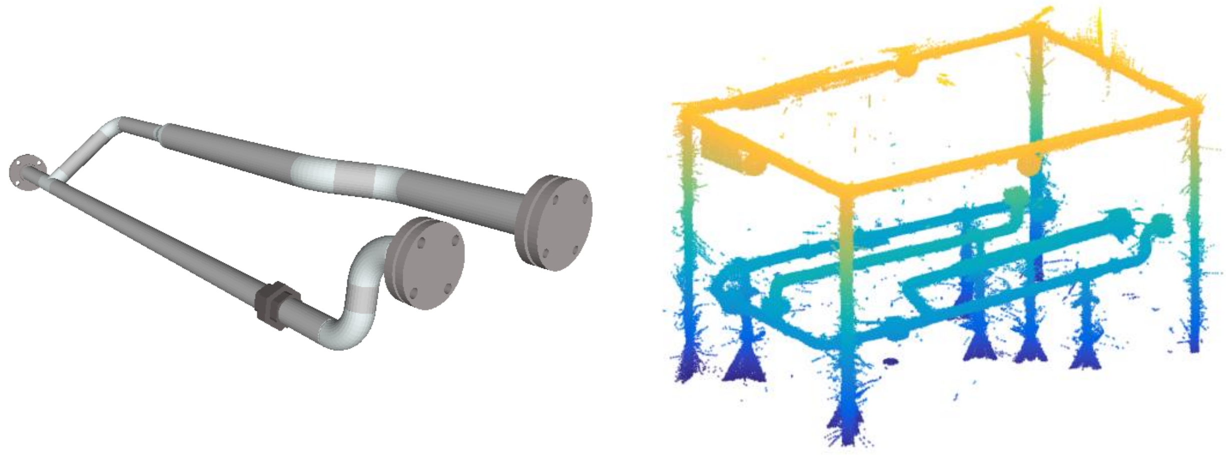


Figure 4-3: Proposed RANSAC-based pipe spool recognition framework

4.1.1 Preprocessing

The recognition framework requires two input files (Figure 4-4): (1) the 3D CAD design file for the pipe spool of interest and (2) the cluttered point cloud scan, P_{scan}, from which the as-built pipe spool of interest will be extracted.



(a) 3D CAD design file

(b) cluttered point cloud scan

Figure 4-4: Search and extraction algorithm input files (a) 3D CAD design file for the object of interest and (b) the raw point cloud scan from which the as-built object of interest will be extracted

Typically, the 3D CAD design file will be obtained in a solid model format that needs to be converted into a point cloud format. First, the 3D CAD design file is exported as an ".STL" file. The vertices within the STL file form a sparse point cloud. The density of the point cloud is increased through a process of triangular mesh surface subdivision (Figure 4-5) (Cignoni et al. 2014). The result is a point cloud, P_{design}, representing the 3D CAD design file.

4.1.2 Normal vector calculation

The descriptor used in the RANSAC methodology will rely on local point surface normal vectors (Figure 4-6). A normal vector is calculated for each point in both input point clouds, P_{design} and P_{scan}. The calculation of these normal vectors is performed in two steps: (1) find nearest neighbours for each point and (2) estimate the normal vector by performing plane Principal Component Analysis (PCA) (Smith 2002) on the set of nearest neighbours for each point.

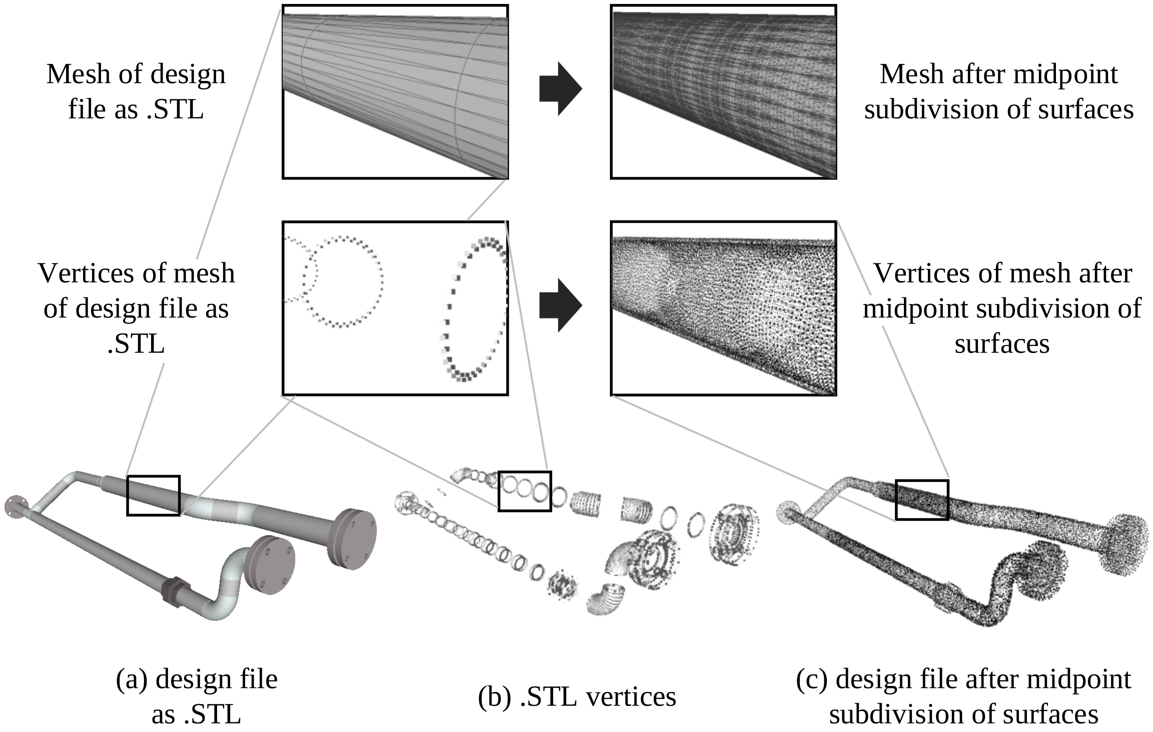

Mesh of design file as .STL
Mesh after midpoint subdivision of surfaces
Vertices of mesh of design file as .STL
Vertices of mesh after midpoint subdivision of surfaces
(a) design file as .STL
(b) .STL vertices
(c) design file after midpoint subdivision of surfaces

Figure 4-5: Midpoint subdivision of surfaces, used to increase the density of the input .STL design file

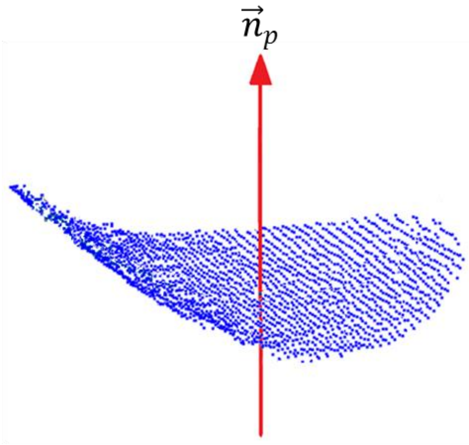


Figure 4-6: Local surface normal vector for point in point cloud

4.1.2.1 Finding nearest neighbours

The nearest neighbours for each point are located using the k-nearest neighbour (KNN) algorithm (Cover and Hart 1967; Jain et al. 2000) with k=10 and is supported by a KD-Tree space-partitioning data structure (Bentley 1975; Friedman et al. 1977). The point and its nearest neighbours are then stored in an array for reference during surface normal vector estimation.

4.1.2.2 Surface normal vector estimation

A comparison of surface normal estimation methods for range sensing applications was carried out by Klasing et al. (2009) and concluded that as long as a KD-Tree data structure is maintained and updated, the plane Principal Component Analysis (PCA) (Smith 2002) is the universal method of choice because of its superior performance in terms of both quality and speed. Performing PCA on the 10 nearest neighbours, the resulting eigenvector with the smallest corresponding eigenvalue is the estimated normal vector $\vec{n}_p$ of the selected point.

4.1.3 RANSAC

Once normal vectors for the points in each point cloud have been calculated, the RANSAC method is used to find hypothesis locations of the pipe spool in the cluttered laser scan. RANSAC is performed in a series of four steps: (1) minimal point set selection, (2) descriptor calculation, (3) descriptor comparison, and (4) minimal point set scoring. A summary of the RANSAC-based algorithm is shown in Figure 4-7.

4.1.3.1 Minimal point set selection, description, and comparison

The first step in executing RANSAC for pipe spool recognition is to select minimal point sets from both P_{design} and P_{scan}. The minimal point set in RANSAC, is the smallest number of points that can be used to apply a transformation to P_{design}. A unique transformation requires at least three reference points, so the minimal point set will be a set of three points. These three points, along with a point set descriptor uniquely define a position of P_{design}. So, three random points $\{f_{pc}^1, f_{pc}^2, f_{pc}^3\} \in P_{pc}$ are selected from each point cloud, with the superscripts denoting the three individual points. Each member is a vector such that,

$f^i = \langle p^{i1}, p^{i2}, p^{i3}, n^{i1}, n^{i2}, n^{i3} \rangle$, in which p^i's are the point coordinates and n^i's are the normal vector components in a 3D global coordinate space.

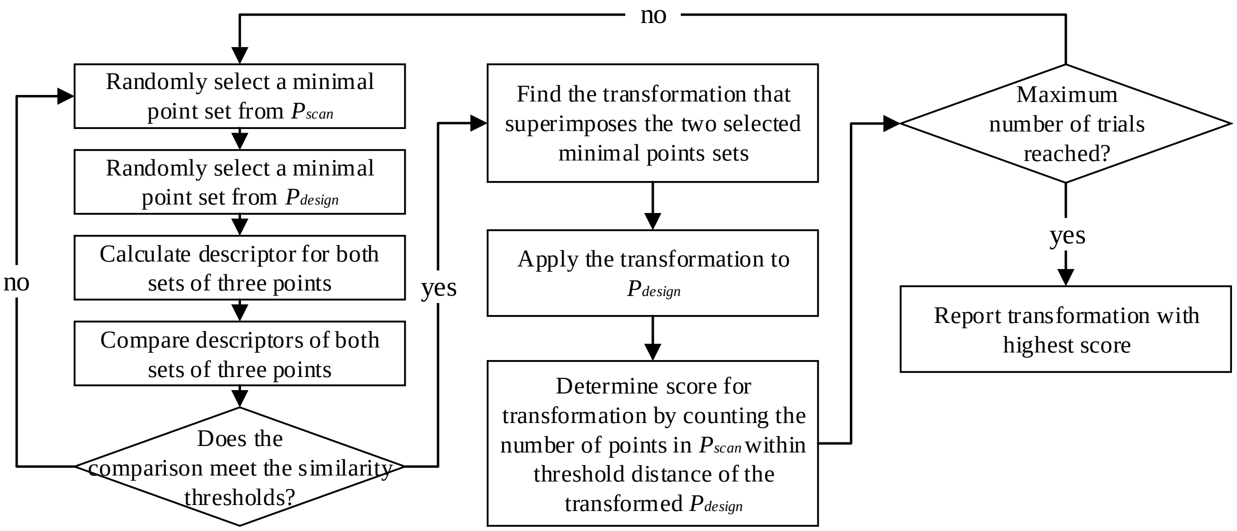

no
Randomly select a minimal point set from P_{scan}
Randomly select a minimal point set from P_{design}
Calculate descriptor for both sets of three points
Compare descriptors of both sets of three points
Does the comparison meet the similarity thresholds?
no
yes
Find the transformation that superimposes the two selected minimal points sets
Apply the transformation to P_{design}
Determine score for transformation by counting the number of points in P_{scan} within threshold distance of the transformed P_{design}
Maximum number of trials reached?
yes
Report transformation with highest score

Figure 4-7: Summary of the RANSAC-based matching algorithm

The point set descriptor (Figure 4-8) is comprised of two components: (1) the area of the triangle created by the three points, and (2) the volume of the parallelepiped formed by the three normal vectors.

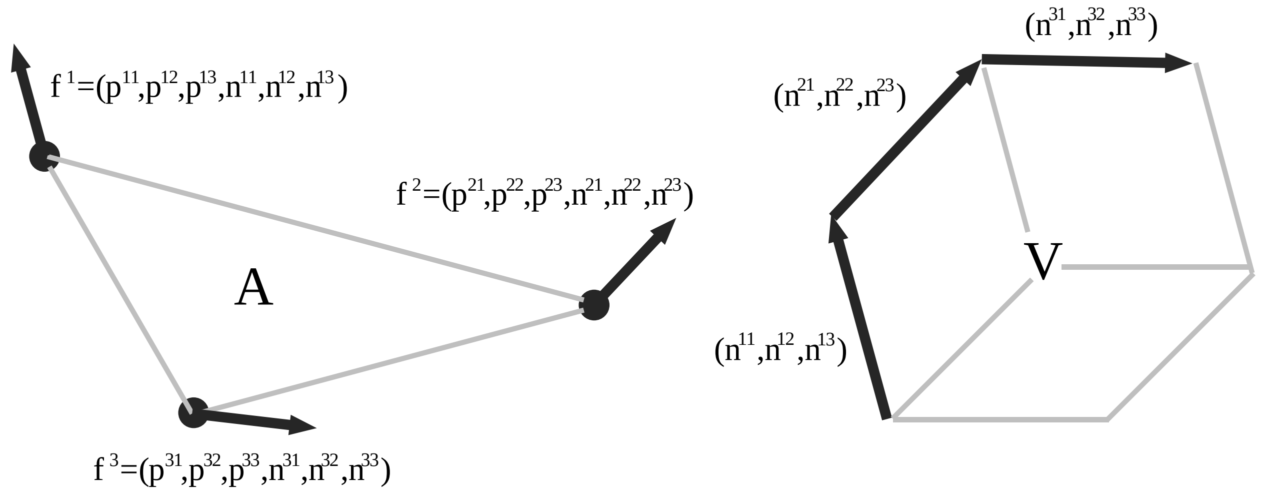

$f^1=(p^{11},p^{12},p^{13},n^{11},n^{12},n^{13})$
$f^2=(p^{21},p^{22},p^{23},n^{21},n^{22},n^{23})$
A
$f^3=(p^{31},p^{32},p^{33},n^{31},n^{32},n^{33})$
(n^{31},n^{32},n^{33})
(n^{21},n^{22},n^{23})
V
(n^{11},n^{12},n^{13})

Figure 4-8: Point set descriptors, (A) the area of the triangle created by the three points and (V) the volume of the parallelepiped formed by the three normal vectors

For the selected subset $\{f_{pc}^1, f_{pc}^2, f_{pc}^3\}$ the area parameter is calculated as follows:

$$A = \frac{1}{2}|(\hat{x}^1 - \hat{x}^2) \times (\hat{x}^1 - \hat{x}^3)| \qquad \text{Eq. 4-1}$$

where, $\hat{x}^i$'s are the vectors that connect the three points $(f_{pc}^1, f_{pc}^2, f_{pc}^3)$, and A is the area of the triangle made by the three selected points $(f_{pc}^1, f_{pc}^2, f_{pc}^3)$. The volume of the parallelepiped made (V) by the three normal vectors, is calculated as follows:

$$V = \overrightarrow{n^1} \cdot \left(\overrightarrow{n^2} \times \overrightarrow{n^3}\right) \qquad \text{Eq. 4-2}$$

The area and volume parameters for the original dataset P_{pc} are denoted by A_{pc} and V_{pc}, respectively. In special cases, the volume equals zero when the three normal vectors are belonging to a plane (coplanar); and the volume equals 1 when the normal vectors are orthogonal in Cartesian coordinate space (i.e.,, $n^i . n^j = 0$). In other words, the volume is between 0 and 1 ($0 \leq V \leq 1$).

Having selected and described two minimal point sets using the area and volume descriptor, the two minimal point sets are compared using Eq. 4-3 and Eq. 4-4.

$$|A_{scan} - A_{design}| \leq A_{th} \qquad \text{Eq. 4-3}$$

$$|V_{scan} - V_{design}| \leq V_{th} \qquad \text{Eq. 4-4}$$

where, A_{th} and V_{th} are the area and volume thresholds, respectively.

The algorithm continues to select, describe and compare random minimal point sets until both the area and volume thresholds are met for a pair of point sets. Once the comparison criteria are met, a hypothesis transformation is generated.

4.1.3.2 Minimal point set scoring

A rigid transformation is composed of a rotational component $\vec{R}$, and a translational component $\vec{T}$. The transformation is thus denoted as $g = (\vec{R}, \vec{T})$. For finding the required transformation to match

$(f_{design}^{1}, f_{design}^{2}, f_{design}^{3})$ to $(f_{scan}^{1}, f_{scan}^{2}, f_{scan}^{3})$, single value decomposition (SVD) of the covariance matrix is used (CII chartered Research Team 327 2015; Nahangi et al. 2014).

Once the transformation is calculated, P_{design} is transformed by $g = (\vec{R}, \vec{T})$ and the resulting matrix is set as the new hypothesis location:

$$P_{design}' \coloneqq g(\vec{R}, \vec{T}) \times P_{design} \qquad \text{Eq. 4-5}$$

The final step of the RANSAC process is to check the score of the resulting hypothesis by counting the number of transformed model points which lie within some threshold distance of points within the scene (Figure 4-9). For that purpose, a Euclidean distance threshold (inlier threshold) is set. The points that are closer than the threshold are counted as inliers.

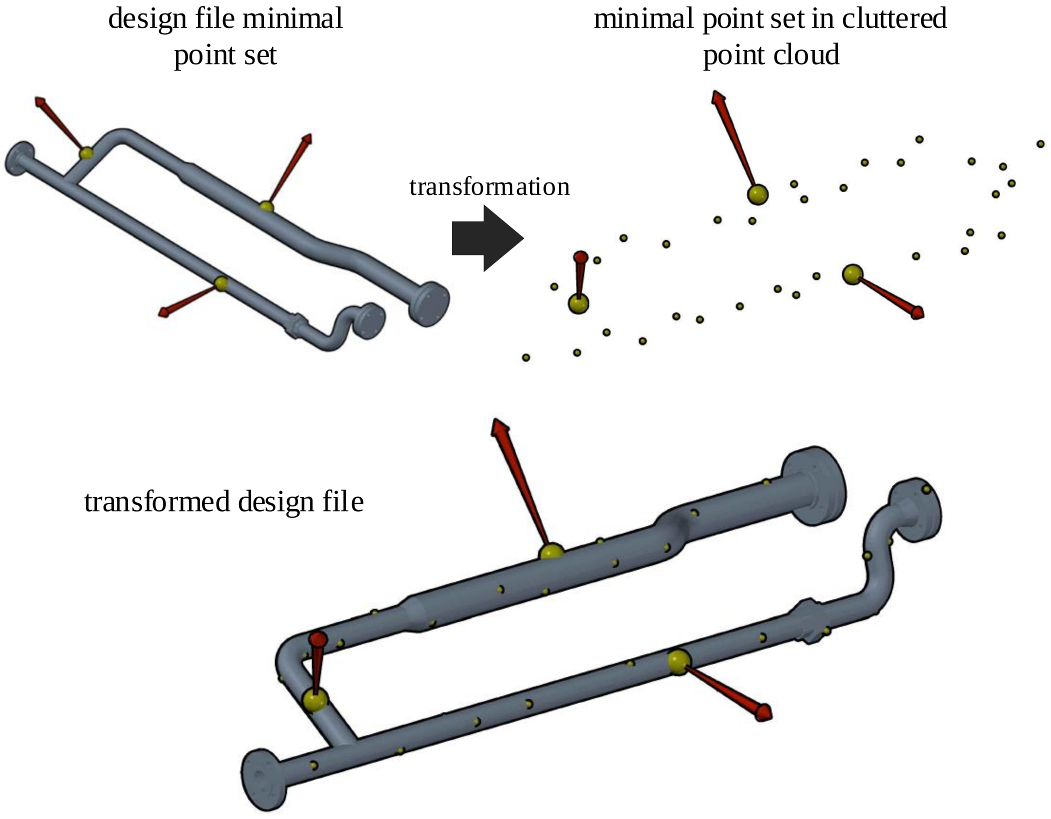

design file minimal point set
minimal point set in cluttered point cloud
transformation
transformed design file

Figure 4-9: Transformation of design file to cluttered point cloud based on minimal point set match

The entire process, from minimal point set selection to scoring is repeated until a predefined number of hypotheses have been scored. Once this predefined number of trails has been met, the hypothesis location with the highest score is chosen as the location of the object of interest.

4.2 Basic RANSAC pipe spool recognition algorithm evaluation

4.2.1 Test setup

The algorithm was tested on a real laser scanned point cloud of a single pipe spool in a university laboratory (Figure 4-4b). The scan data, P_{scan}, was collected using a FARO laser scanner (Table 4-1).

Table 4-1: Technical specifications of FARO LS 880 HE scanner

Measurement	Range	Accuracy
Distance	0.6-40 m	0.6 mm (@ maximum resolution)
Field of View	Horizontal: 360° Vertical: 320°	Horizontal: 0.009° Vertical: 0.00076°

The preprocessing of P_{design} was performed in Meshlab (Cignoni et al. 2014), using the "subdivision surfaces: midpoint" function. The input files were scaled such that the resulting point set units were centimeters. The algorithm was implemented and programmed in MATLAB 2015 using standard functions and toolboxes whenever possible. A sensitivity analysis was performed by varying the comparison thresholds, using $A_{th} = \{1, 2, 4, 8, 16, 32, 64\}$ and $V_{th} = \{0.01, 0.02, 0.04, 0.08, 0.16\}$ with units in centimeters. The inlier threshold used for scoring was 5cm.

For instances when the point set comparison thresholds A_{th} and V_{th} were low, a timeout threshold needed to be set because the algorithm would require excessive amount of time to find point sets that are similar enough to proceed to hypothesis scoring. Thus, if the algorithm compared 500,000 different point sets with none meeting the comparison thresholds, the execution would cancel.

The algorithm was set to generate 1000 hypotheses meeting the comparison thresholds before selecting the hypothesis with the highest score.

4.2.2 Results and discussion

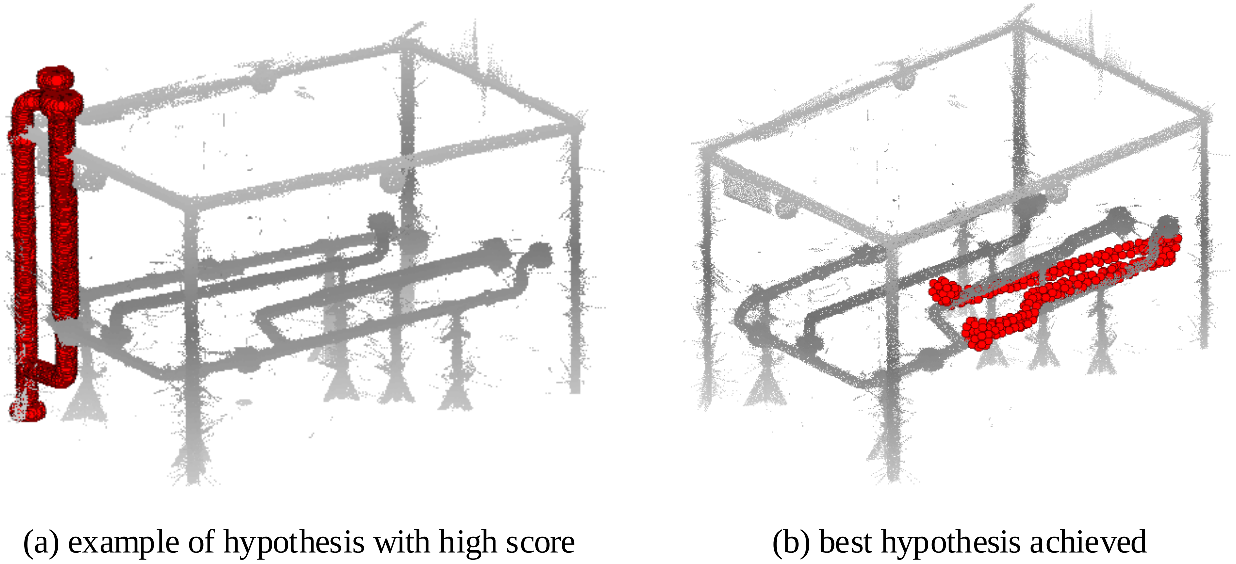
The basic RANSAC algorithm implementation failed to reliably locate the pipe spool of interest in the cluttered point cloud scan. The results of the sensitivity analysis can be found in Table 4-2. The execution time (benchmarked on a 3.7 GHz 12 core processor with 32 GB RAM) ranged from 15 to 85 minutes depending on the comparison thresholds. Higher thresholds allowed for faster algorithm execution. In the instances when there was no hypothesis generated, (i.e., naN in the table), the timeout threshold was triggered.

The size of P_{design} was 3619, thus the hypothesis score indicating a correct pipe spool recognition was expected to be approximately 3619. None of the parameter combinations achieved this score. The highest scores were achieved with $A_{th} = \{4,8\}$ and $V_{th} = \{0.04, 0.08\}$. When A_{th} and V_{th} where large, the algorithm achieved lower scores because the hypothesis transformations were excessively random and less discriminative.

Table 4-2: Basic RANSAC implementation descriptor thresholds sensitivity analysis, best hypothesis scores after 1000 trials

		A_{th} (cm^2)							
		0.5	1	2	4	8	16	32	64
V_{th} (unit^3)	0.005	naN	naN	naN	naN	naN	1103	845	636
	0.01	naN	620	naN	92	908	1042	692	824
	0.02	naN	naN	623	279	712	854	905	624
	0.04	naN	184	531	1319	1232	826	730	872
	0.08	134	256	812	1027	1269	1012	791	608
	0.16	367	410	606	853	798	871	766	923

Figure 4-10 presents two example hypothesis locations of P_{design}. Figure 4-10a is the transformation resulting from the algorithm run using $A_{th} = 8$ and $V_{th} = 0.04$. The high score of 1232 was achieved because one of the pipe branches in P_{design} aligned perfectly with a beam in the structural frame in P_{scan}. Although a high score was achieved, the hypothesis location was incorrect. Figure 4-10b was the best result achieved in terms of hypothesis location accuracy, but as can be seen, the alignment was still extremely poor.


(a) example of hypothesis with high score

(b) best hypothesis achieved

Figure 4-10: example transformations (i.e., pipe spool hypothesis locations) applied to P_{design} by the basic RANSAC algorithm

Although the basic RANSAC approach has the advantage of being conceptually simple, its direct application to the 3D pipe spool recognition problem is computationally very expensive. The probability of selecting the 3 points in the scan that correspond perfectly with the three points selected from the 3D CAD design file is

$$P(ideal\_trial) = \frac{3!}{(n-2)\times(n-1)\times n} = \frac{3!}{(2262321-2)\times(2262321-1)\times 2262321} = 5.1819\times 10^{-19}$$

Where, $n = size(P_{scan}) = 2{,}262{,}321\ points$

Although, a correct hypothesis selection might be achievable, the computation time would be prohibitive to practical implementation. A second version of the RANSAC recognition algorithm is proposed in the following section, and attempts to reduce n and improve the probability of selecting a set of matching points.

4.3 RANSAC-based pipe spool recognition methodology with curvature-based down sampling

The basic form of RANSAC is very computationally expensive because of the incredibly large number of possible minimal set selections. An efficient form of RANSAC was presented in (Schnabel et al. 2007) for primitive shape detection. Their method was used to detect planes, spheres, cylinders, cones and tori. The benefit of this efficient form, is the reduction of the number of possible minimal point sets by limiting selection using a distance criteria. The work demonstrated that the basic RANSAC framework can be improved by reducing the search space represented by P_{scan}. Randomly downsampling the input point clouds, does not work, as it does not improve the probability of selecting matching point sets.

Discriminative down sampling retains the points of a select few areas in P_{design} and downsamples P_{scan} such that the same areas on the pipe spool of interest are also retained. In this section, a curvature characterization algorithm is utilized to discriminatively down sample a point cloud such that only points with a particular curvature description are input into the RANSAC process. The premise is, if for example, only the pipe elbow joints are retained in P_{design} and P_{scan}, then the probability of selecting two corresponding minimal point sets will improve substantially.

4.3.1 Curvature characterization

In order to discriminatively down sample both point clouds a curvature-based shape descriptor is used to characterize and then filter the points within each input file. As seen in Figure 4-11, the proposed curvature characterization algorithm has two primary steps: (1) normal vector extraction and (2) local curvature calculation. Each step is explained in the following sections. The MATLAB code can be found in Appendix F.

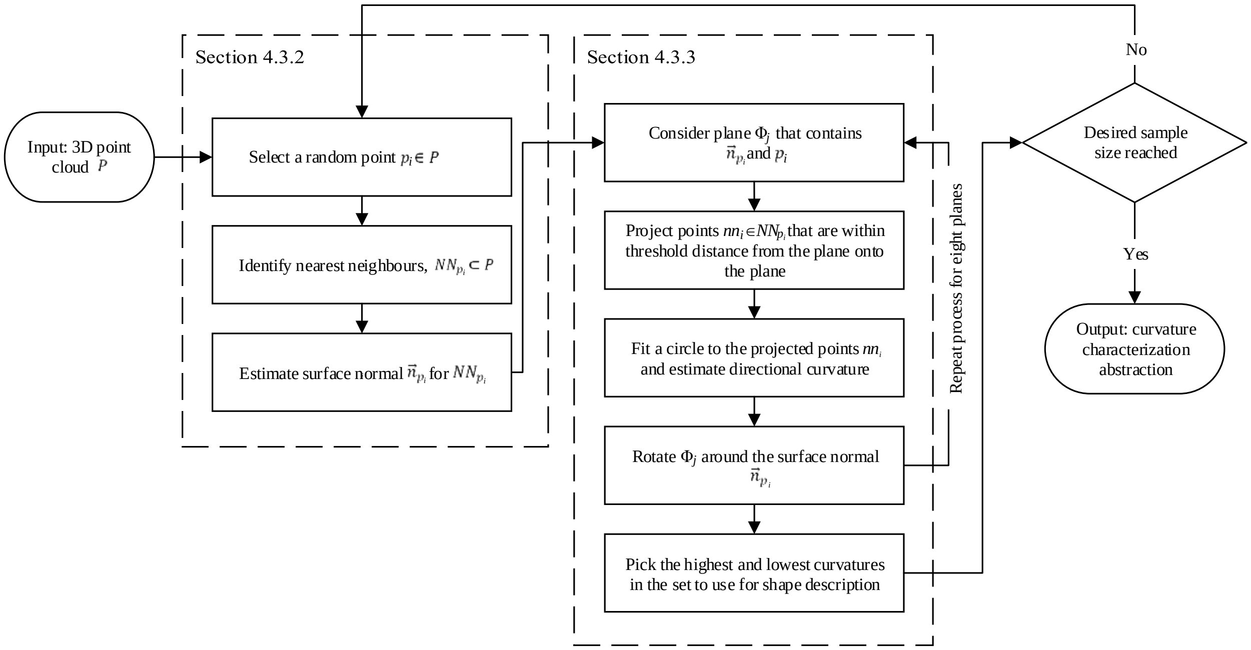

Section 4.3.2
Section 4.3.3
No
Input: 3D point cloud P
Select a random point $p_i \in P$
Identify nearest neighbours, $NN_{p_i} \subset P$
Estimate surface normal $\vec{n}_{p_i}$ for NN_{p_i}
Consider plane Φ_j that contains $\vec{n}_{p_i}$ and p_i
Project points $nn_i \in NN_{p_i}$ that are within threshold distance from the plane onto the plane
Fit a circle to the projected points nn_i and estimate directional curvature
Rotate Φ_j around the surface normal $\vec{n}_{p_i}$
Pick the highest and lowest curvatures in the set to use for shape description
Repeat process for eight planes
Desired sample size reached
Yes
Output: curvature characterization abstraction

Figure 4-11: Flowchart for curvature characterization

4.3.2 Normal vector estimation

First, using $p_i \in p$ as the centre, a nearest neighbourhood subset $NN_{p_i} \subseteq P$ is isolated (Section 3.3.2) from P. The nearest neighbourhood selection is supported by a KD-Tree space-partitioning data structure (Bentley 1975; Friedman et al. 1977). Such structures are widely used for indexing in search engines (Philbin et al. 2007). The NN_{p_i} comprises all points in P within radius r of p_i.

The surface normal vector of local surface point subsets has been the unanimous choice of reference to compute surface descriptors. A comparison of surface normal estimation methods for range sensing applications was carried out by (Klasing et al. 2009) and concluded that as long as a KD-Tree data structure is maintained and updated, the plane Principal Component Analysis (PCA) (Smith 2002) is the universal method of choice because of its superior performance in terms of both quality and speed. Performing PCA on NN_{p_i}, the resulting eigenvector with the smallest corresponding eigenvalue is the estimated normal vector $\vec{n}_{p_i}$ of NN_{p_i}. This procedure for normal vector extraction is illustrated in Figure 4-12.

4.3.3 Local curvature estimation

In order to find the principal curvatures, eight planes Φ_j = {Φ_1, Φ_2…Φ_8} passing through p_i and rotating uniformly at π/8 radians, $\theta = \left\{\frac{\pi}{8}, \frac{2\pi}{8}, \dots, \frac{7\pi}{8}\right\}$, around the surface normal $\vec{n}_{p_i}$ are generated (Figure 4-13). The normal $\vec{n}_{\Phi 1}$ to the first plane Φ_1 is set as the eigenvector with the largest eigenvalue calculated during the plane PCA of NN_{p_i} above. Smaller rotation intervals provide more accurate principal curvature characterization, however, if two planes in the set Φ_j are parallel with the first and second principal components of the NN_{p_i}, the improvement in accuracy is minimal.

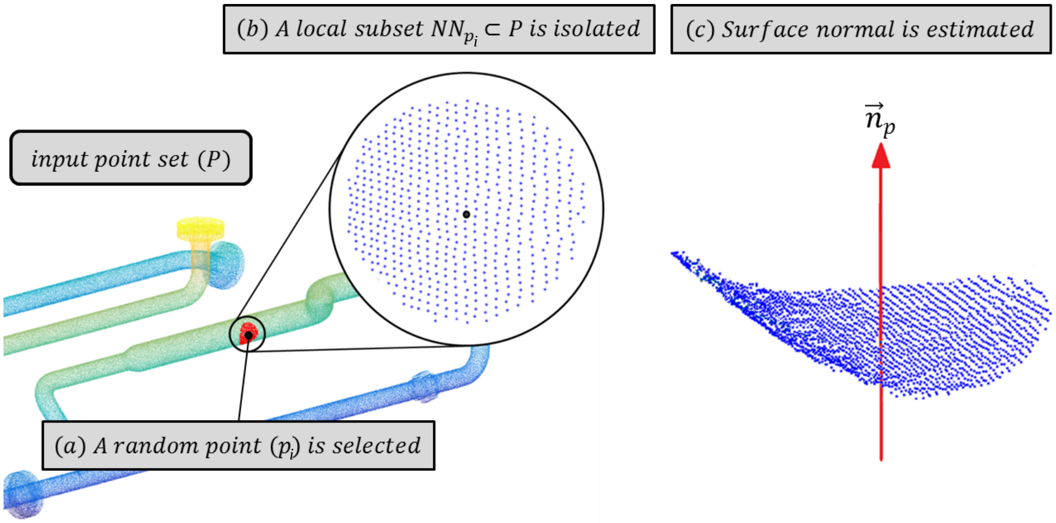

(b) A local subset $NN_{p_i} \subset P$ is isolated
(c) Surface normal is estimated
input point set (P)
$\vec{n}_p$
(a) A random point (p_i) is selected

Figure 4-12: Normal vector extraction procedure: (a) A random point p is first selected, (b) a neighbourhood region is then isolated $NN_p \subset P$ and (c) the normal vector $\vec{n}_p$ is estimated using plane PCA

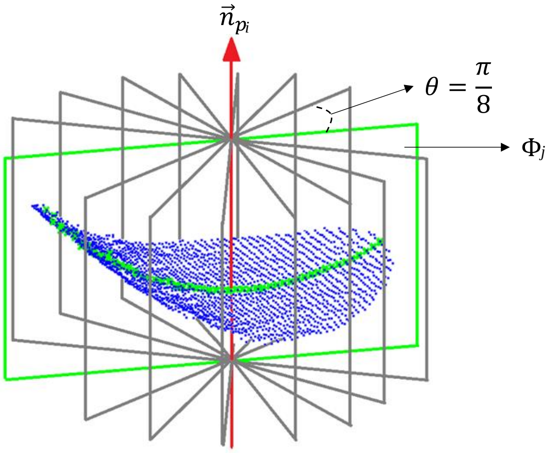


Figure 4-13: The planes are rotated about the normal vector $\vec{n}_p$ to find the principal curvature.

For each of the planes Φ_j, the distance d between every point in $nn_i \in NN_{p_i}$ and its corresponding projection $nn_i \rightarrow nn_i' \in \Phi_j$ on the plane is calculated (Figure 4-14). $\overrightarrow{nn_\iota nn_\iota'} \parallel \vec{n}_\Phi$, therefore the equations providing the coordinate elements of nn_i' are as follows:

$$\begin{cases} x_i' = x_i + d \ \times \vec{n}_{\mathrm{x}\Phi} \\ y_i' = y_i + d \ \times \vec{n}_{\mathrm{y}\Phi} \\ z_i' = z_i + d \ \times \vec{n}_{\mathrm{z}\Phi} \end{cases} \qquad \text{Eq. 4-6}$$

The scalar equation of the plane Φ_i is:

$$\vec{n}_{\mathrm{x}\Phi}(x - x_i') + \vec{n}_{\mathrm{y}\Phi}(y - y_i') + \vec{n}_{\mathrm{z}\Phi}(z - z_i') = 0 \qquad \text{Eq. 4-7}$$

Eliminating the unknown nn_i' coordinate elements by substituting Eq. 4-6 into Eq. 4-7 yields:

$$d = \left| \frac{\vec{n}_{\mathrm{x}\Phi}(x_i - x) + \vec{n}_{\mathrm{y}\Phi}(y_i - y) + \vec{n}_{\mathrm{z}\Phi}(z_i - z)}{\vec{n}_{\mathrm{x}\Phi}{}^2 + \vec{n}_{\mathrm{y}\Phi}{}^2 + \vec{n}_{\mathrm{z}\Phi}{}^2} \right| \qquad \text{Eq. 4-8}$$

Points $nn_i \in NN_p$ that have $(d \leq threshold)$ to the plane Φ_j are selected for circle fitting, $nn_i(circle\ fitting) \subseteq nn_i$. Back-substituting d into Eq. 4-6, $nn_i'(circle\ fitting)$ are calculated.

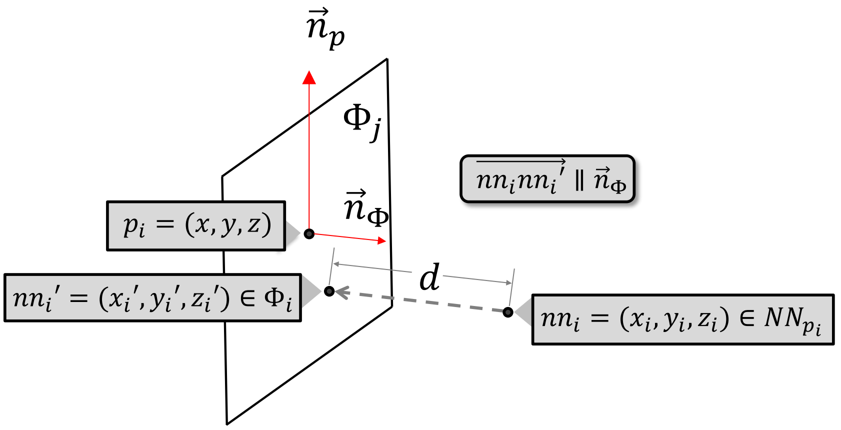

$\vec{n}_p$
Φ_j
$\overrightarrow{nn_i nn_i'} \parallel \vec{n}_\Phi$
$\vec{n}_\Phi$
$p_i = (x, y, z)$
d
$nn_i' = (x_i', y_i', z_i') \in \Phi_i$
$nn_i = (x_i, y_i, z_i) \in NN_{p_i}$

Figure 4-14: Projecting a point on a plane Φ_i. Point p is the original selected point and $\vec{n}_p$ is the calculated normal vector to p. Point p_i is a point in NN_p distance d from the plane . Point p_i' is the projection of point p_i onto the plane Φ_i.

$nn_i'(circle\ fitting)$ must now be transformed to the XY plane to reduce the dimensionality of the data to allow for circle fitting. This alignment procedure is in the form of a rotation matrix that maps $\vec{n}_{\mathrm{x}\Phi}$ onto the global $z\ axis$. Circle fitting is performed by the hyper-accurate algebraic fit method (Al-Sharadqah and Chernov 2009). Circle fitting provides a circle radius (Figure 4-15), which is converted to a directional curvature value for the local surface.

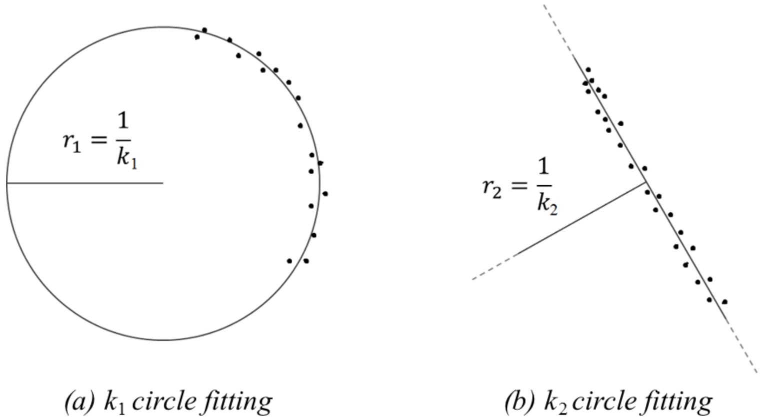


Figure 4-15: Circle fitting, (a) Φ_i circle fitting results providing the smallest radius circle i.e., k_1 highest curvature value, and (b) Φ_i circle fitting results providing the largest radius circle i.e., k_2 lowest curvature value

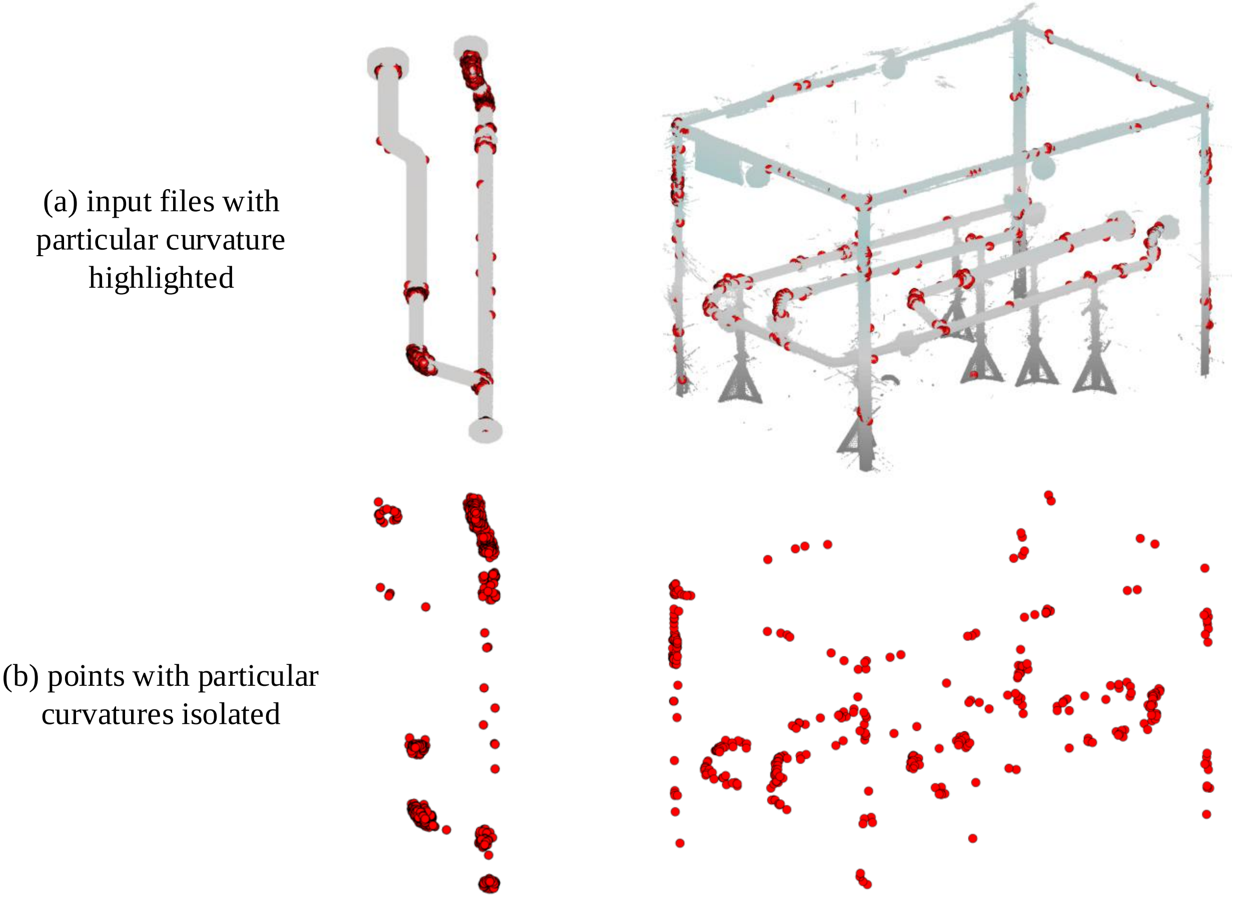
The maximum curvature value from the set of eight calculated is identified as the principal curvature k_1, and the perpendicular plane is the corresponding principal curvature k_2. Finally, ε_{k1} and ε_{k2} are computed and stored as the mean fit error from the circles in the two principal planes. The procedure is repeated until the desired sample size has been characterized.

4.4 RANSAC pipe spool recognition algorithm with curvature-based down sampling evaluation

4.4.1 Test setup

As for the basic RANSAC framework, the algorithm with the added curvature-based down sampling step was tested on a real laser scanned point cloud of a single pipe spool in a university laboratory (Figure 4-4b). The scan data, P_{scan}, was collected using a FARO laser scanner (Table 4-1).

The preprocessing of P_{design} was performed as before and the algorithm was implemented and programmed in MATLAB 2015 using standard functions and toolboxes whenever possible. The curvature characterization algorithm was used to describe 5000 points from P_{scan} and 1000 points from P_{design} (Figure 4-16). The curvature values calculated were used to identify points falling on pipe elbow joints. Since $curvature = \frac{1}{radius\ of\ circle}$, therefore $k1 = \frac{1}{radiusof\ pipe}$ and $k2 = \frac{1}{radius\ of\ bend}$. Therefore the points within a threshold value of $k1 = 1/6$ and $k2 = 1/10.5$ were retained from the original point clouds to be used for selecting minimal point sets for the RANSAC algorithm.


(a) input files with particular curvature highlighted

(b) points with particular curvatures isolated

Figure 4-16: Curvature-based discriminative down sampling for RANSAC recognition- points with curvatures similar to elbow-joints isolated

A sensitivity analysis was performed by varying the comparison thresholds, using $A_{th} = \{1, 2, 4, 8, 16, 32, 64\}$ and $V_{th} = \{0.01, 0.02, 0.04, 0.08, 0.16\}$. The inlier threshold used for scoring was 5cm.

For instances when the point set comparison thresholds A_{th} and V_{th} were low, a timeout threshold needed to be set because the algorithm struggles to find point sets that are similar enough to proceed to hypothesis scoring. Thus, if the algorithm compared 500,000 different point sets with none meeting the comparison thresholds, the execution would cancel.

The algorithm was set to generate 100 hypotheses meeting the comparison thresholds before selecting the hypothesis with the highest score.

4.4.2 Results and discussion

Even after the curvature-based down sampling, the RANSAC algorithm implementation failed to reliably locate the pipe spool of interest in the cluttered point cloud scan. The results of the sensitivity analysis can be found in Table 4-3. The instances when there was no hypothesis generated, (i.e., naN in the table), the timeout threshold was triggered.

The size of P_{design} was 3619, thus the hypothesis score indicating a correct pipe spool recognition was expected to be approximately 3619. None of the parameter combinations achieved this score.

Table 4-3: Basic RANSAC implementation descriptor thresholds sensitivity analysis, best hypothesis scores after 1000 trials

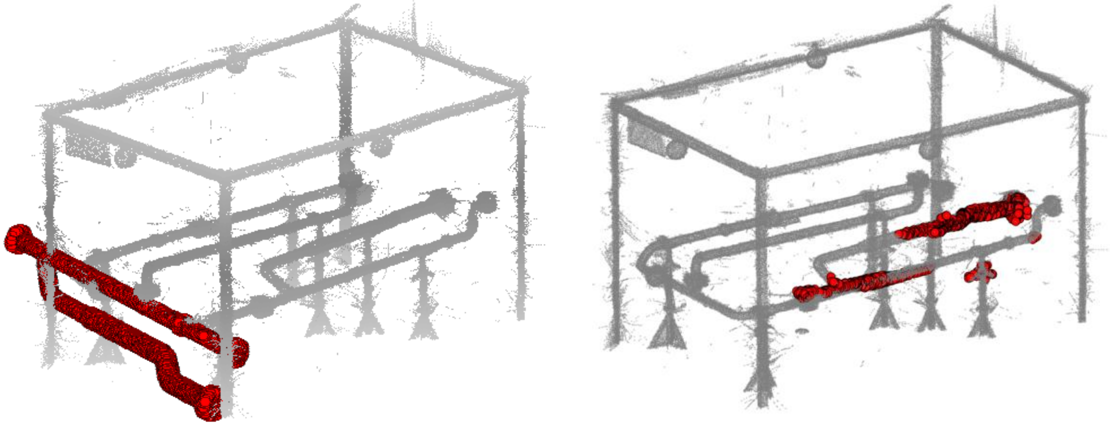
		A_{th} (cm^2)							
		0.5	1	2	4	8	16	32	64
V_{th} (unit^3)	0.005	naN	naN	437	902	1231	1248	992	702
	0.01	naN	naN	834	317	922	1051	764	977
	0.02	832	436	641	1236	1874	926	939	743
	0.04	naN	892	633	1420	1231	972	857	981
	0.08	255	323	870	1080	1396	1016	878	694
	0.16	420	554	729	950	1026	1065	858	1041

Figure 4-17 presents two example hypothesis locations of P_{design}. Figure 4-17a demonstrates a typical erroneous hypothesis, a performance quality similar to the basic RANSAC. Figure 4-17b is the best hypothesis achieved ($A_{th} = 8$ and $V_{th} = 0.02$) executing either RANSAC framework, and is the only instance the recognition was performed successfully. Repeating the execution with the same comparison thresholds failed to repeat the recognition.

Although the theoretical probability of selecting the 3 points in the scan that correspond perfectly with the three points selected from the 3D CAD design file is better within the curvature down sampled search space, the probability still is not favorable.

$$P(ideal\_trial) = \frac{3!}{(n-2)\times(n-1)\times n} = \frac{3!}{(356-2)\times(356-1)\times 356} = 1.3411 \times 10^{-7}$$

Where, $n = size(downsampled\ P_{scan}) = 356\ points$



(a) example of hypothesis with high score

(b) best hypothesis achieved

Figure 4-17: Example transformations (i.e., pipe spool hypothesis locations) applied to P_{design} by the RANSAC algorithm aided by curvature-based down sampling

The RANSAC performs well in the literature on primitive shapes such as planes, spheres, cylinders, cones and tori, but evidently fails to achieve reliable recognition for pipe spools. A more effective data structure and strategy for comparing minimal point sets is required. Additionally, requiring a user to manually enter a specific curvature type (i.e., elbow joint curvature specification) is a difficult and subjective step that needs to be eliminated.

The following section, introduces a novel pipe spool recognition method based on local data level curvature estimation, clustering, and bag-of-features matching. The validation of the novel method demonstrates its effectiveness at extracting pipe spools from cluttered point clouds.

4.5 Pipe spool recognition using a curvature-based shape descriptor

In the following sections, a recognition method for locating and extracting pipe spools from cluttered point clouds is presented. No prior assumptions are made about pipe spool size, configuration, or position within the point cloud scan (P_{scan}). The expected amount and type of clutter in P_{scan} is what would reasonably be expected in a pipe spool fabrication facility (e.g., the building structure, tables, humans, machinery, other pipe spools, etc.). The recognition framework requires two input files (Figure 4-18): (1) the 3D CAD design file for the pipe spool of interest, and (2) P_{scan} from which the as-built pipe spool of interest will be extracted. No other input by the user is required. The entire process is performed in seven steps:

1- Preprocessing of input files

2- Major planes are removed from P_{scan} (e.g., walls, floor, ceiling, etc.)

3- The points in P_{scan} are filtered using a curvature-based shape descriptor; accepted points form a hypothesis space

4- Points in the hypothesis space are clustered into hypothesis objects

5- Using BoF, each hypothesis object is compared to the 3D CAD design file

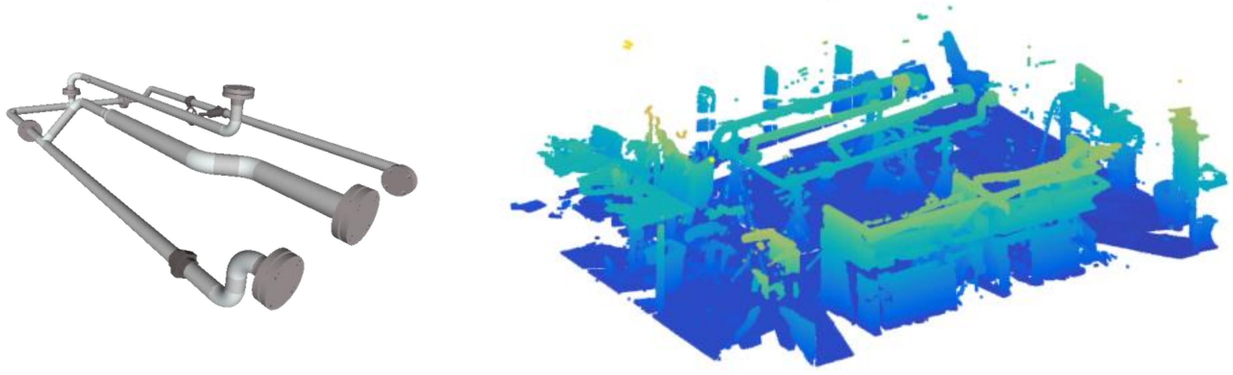
6- The 3D CAD design file is registered to the hypothesis object most similar to the 3D CAD design file

7- The pipe spool of interest (p_{obj}) is extracted from P_{scan}.

The main MATLAB function used to execute pipe spool recognition and extraction can be found in Appendix B.

4.5.1 Preprocessing

First, the 3D CAD design file is exported as an STL file. The vertices within the STL file form a sparse point cloud. The density of the point cloud is increased through a process of triangular mesh surface subdivision. Then, both P_{scan} and P_{design} are subsampled using Poisson-disc subsampling in order to



(a) 3D CAD design file

(b) cluttered point cloud scan

Figure 4-18: Search and extraction algorithm input files (a) 3D CAD design file for the object of interest and (b) the raw point cloud scan from which the as-built object of interest will be extracted

reduce the size of the files, while achieving a more uniform point density. A Poisson-disc radius (pdr) of 5mm was used, $pdr = 5mm$. Additionally, the Poisson-disc subsampling ensures that P_{scan} and P_{design} have the same density, which means that the pipe spool of interest in P_{scan} will have approximately the same amount of points as P_{design}. This will be important for hypothesis space clustering (Section 4.5.4).

4.5.2 Plane removal

A substantial portion of the search space in P_{scan} will be planes, e.g., walls, floor, ceiling, etc. These planar objects are obviously not the pipe spool to be located, so removing them from the scan reduces the search space. The plane removal process (Figure 4-19) used for this work applied Gaussian mapping (Liu et al. 2013; Qiu et al. 2014; Wang et al. 2013). First, surface normal vectors are calculated for points in P_{scan} and mapped to normal vector space (Gaussian sphere). Then by using DBSCAN, dense collections of points on the Gaussian sphere that represent points on major parallel planes can be identified and isolated. However, any plane removal process capable of extracting major planes can be used. The MATLAB code used to execute plane removal can be found in Appendix A.

4.5.3 Curvature-based filtering

A characterization algorithm is now used to describe the curvatures of two sets of random point samples $p_{scan} \subseteq P_{scan}$ and $p_{design} \subseteq P_{design}$. The curvatures of p_{scan} are then filtered based on their similarity to the curvatures of p_{design}. The accepted points produce a hypothesis space.

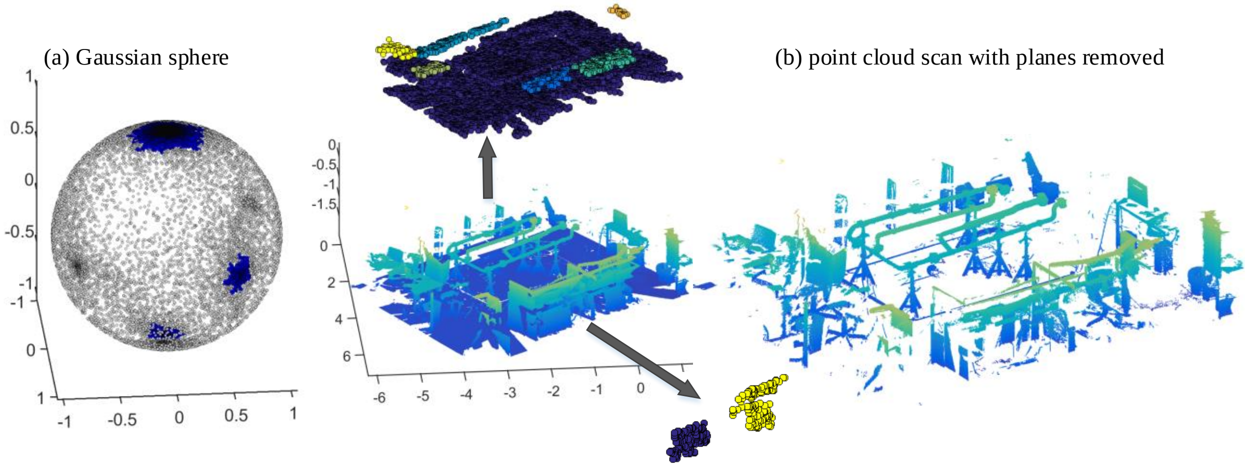


Figure 4-19: Plane removal using Gaussian sphere. (a) Surface normal vectors are calculated for P_{scan} and mapped to normal vector space. The space (Gaussian sphere) is clustered using DBSCAN. The major clusters represent parallel major planes in the original point cloud. (b) Major planes are removed from the raw scan.

4.5.3.1 Curvature characterization

The curvature characterization process applied here is identical to the characterization algorithm presented in Section 4.3. The MATLAB code used for execution can be found in Appendix F.

4.5.3.2 Nearest neighbourhood size for curvature characterization

For analysing cylindrical objects, the amount of data available for $k1$ circle fitting reaches a maximum when $r = 2 \times cylinder\ radius$ (Figure 4-20). Using $r > 2 \times cylinder\ radius$ is undesirable for a number of reasons: (1) No noticeable improvement in estimating $k1$ as compared to $r \approx 2 \times cylinder\ radius$, (2) the probability of including clutter or adjoining pipe fittings or valves that will contaminate the local surface curvature characterization increases as r increases. Conversely, r also needs to be large enough not to be affected by local sensor specific noise levels.

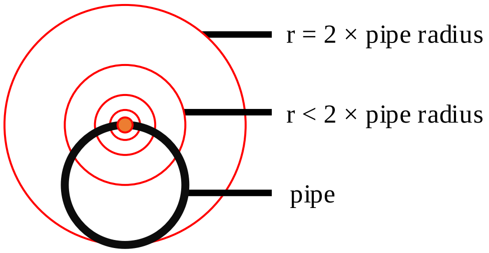


Figure 4-20: k1 cross-section with nearest neighbour size for a point on a pipe

For running curvature characterization of p_{scan} and p_{design}, there are two possible strategies for managing r. With a variable r implementation, p_i is characterized using a series of growing r values until k_{1i} becomes stable and is set as the k_{1i} for p_i (Figure 4-21). This method minimizes the probability of including clutter points in surface curvature estimation while mitigating the impact of sensor noise. However, performing multiple curvature characterizations for each sample point in p_{scan} and p_{design} is computationally expensive.

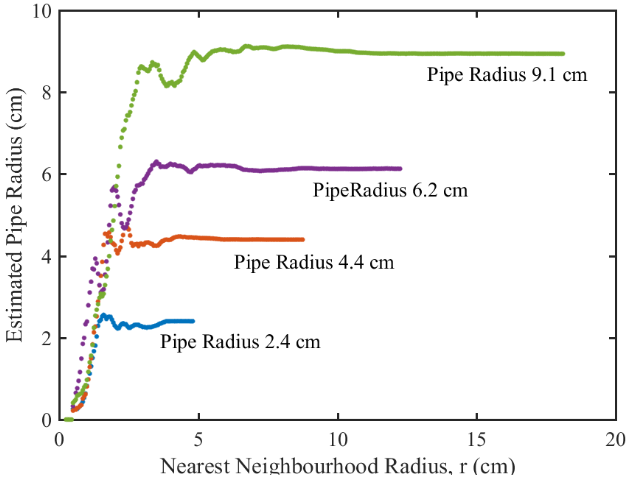


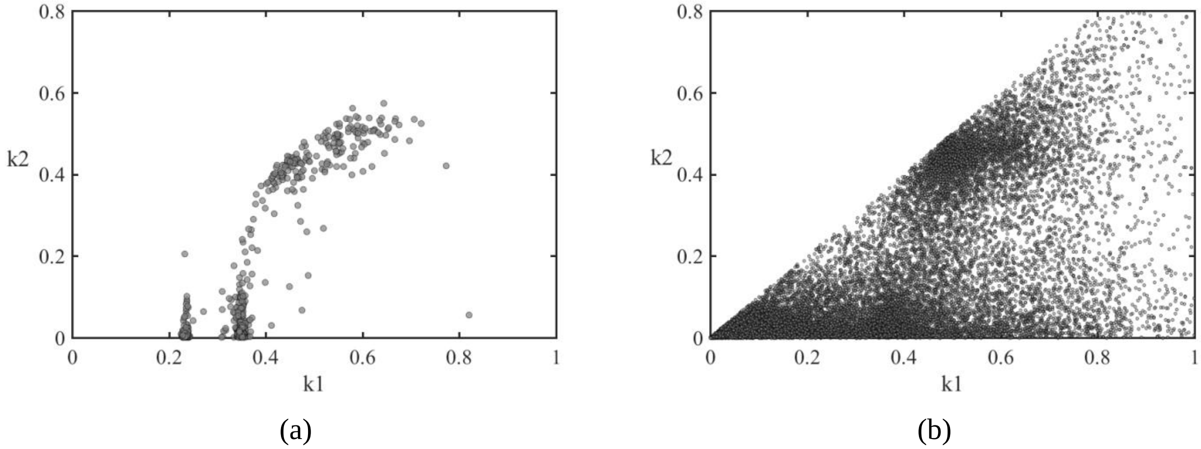
Figure 4-21: Nearest neighbourhood size's effect on estimated pipe radius for four pipes. Each pipe's radius estimation reaches stability and terminates when nearest neighbourhood size equals pipe diameter. The characteristics of the laser scanner used are provided in Table 4-1.

The second possible strategy sets a constant r value for characterization of p_{scan} and p_{design}. For this thesis, this was the strategy of choice because reducing execution time of the algorithm was a priority. r is set by performing curvature characterization on a small subset (approx. 10 points) of p_{design} using the variable r implementation, $r = \{1 \times pdr, 2 \times pdr, 3 \times pdr, etc\}$. This provides a quick estimate of the pipe radii critical for the extraction methodology. For characterizing laser scans of pipe radii between 2.4 cm and 9.1 cm, empirical results (Figure 4-21) suggest the relationship: $lower\ bound\ r = \frac{pipe\ radius}{2} + 1$. Therefore, the optimal constant r can be set using the median pipe radii calculated. The MATLAB code used to determine nearest neighbourhood size can be found in Appendix E.

4.5.3.3 Hypothesis space generation

Once principal curvatures $k1$ and $k2$ for p_{scan} and p_{design} have been calculated (Figure 4-22), p_{scan} is filtered based on the similarity of its curvatures to p_{design} and the quality of the circle fit ε_{k1} that estimated those curvatures. Similarity of curvature is determined based on the mean three nearest neighbour Euclidean distance in curvature space (Figure 4-22) from points in p_{design} to points in p_{scan}. This parameter is denoted by d_{mean}. Points in p_{scan} with $d_{mean} < \theta_{sim}$ (Figure 4-23) and $\varepsilon_{k1} < \theta_{error}$

(Figure 4-24) create the hypothesis space (Appendix G). θ_{sim} (similarity threshold) and θ_{error} (error threshold) together represent the strength of the filter, which will be affected by the level of noise in P_{scan} and are therefore 3D imaging device specific.



(a) (b)

Figure 4-22: Scatter plots of characterized principal curvatures, (a) plot for the 3D CAD design file sample p_{design} and, (b) plot for the cluttered point cloud scan sample p_{scan}

4.5.4 Hypothesis space clustering

The points in the hypothesis space are now clustered into a set of discrete hypothesis (Figure 4-25) objects using DBSCAN (Ester et al. 1996). The parameters of DBSCAN, Eps_{hyp} and

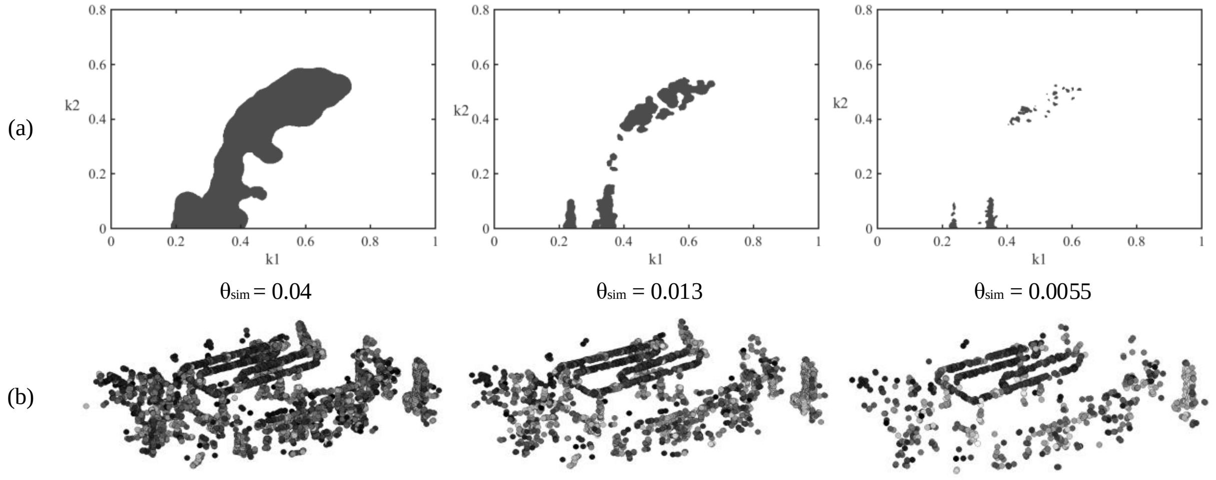


Figure 4-23: Point filtering (a) area in scatter plot where $d_{mean} < \theta_{sim}$ and (b) corresponding points in xyz space with $d_{mean} < \theta_{sim}$ coloured by curvature characterization circle fit quality (darker points have lower circle fit error).

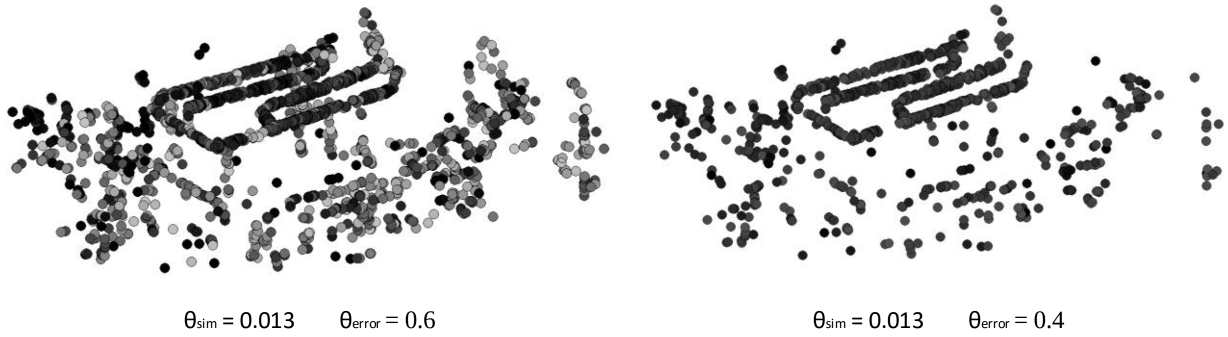


Figure 4-24: Point filtering using the error threshold ($\varepsilon_{k1} < \theta_{error}$).

$MinPts_{hyp}$ are set by simulating Hyp 1, the cluster in hypothesis space representing our pipe spool of interest. The simulated Hyp 1 is denoted by Hyp^{*} 1. We estimate the number of points in Hyp 1 as follows:

$$size(p_{scan}) \times \frac{size(P_{design})}{size(P_{scan})} \times FAR \approx size(\,Hyp\ 1) \qquad \text{Eq. 4-9}$$

$$size(p_{scan}) \times \frac{size(P_{design})}{size(P_{scan})} \approx size(p_{obj}) \qquad \text{Eq. 4-10}$$

where, p_{obj} is the set of points in p_{scan} that is the pipe spool of interest, and FAR is the filter's acceptance rate, which is the approximate percentage of points from p_{obj} that will be accepted by the filter thresholds. The filter's acceptance rate is 3D imaging device specific. The procedure used to calculate FAR for the case study in Section 4.6.2 can be found in Appendices C and D.

The $Hyp^{*}\,1 \subseteq P_{design}$, is processed with DBSCAN with $MinPts = 1$ while incrementally growing Eps until DBSCAN produces a single cluster. The minimum Eps that allows DBSCAN to yield a single cluster from $Hyp^{*}\,1$, is set as Eps_{hyp}. Using Eps_{hyp}, $MinPts_{hyp}$ is set by calculating the mean number of points within radius Eps_{hyp} of points in $Hyp^{*}\,1$. Eps_{hyp} and $MinPts_{hyp}$ are then used to cluster the hypothesis space. The MATLAB code used to determine Eps_{hyp} and $MinPts_{hyp}$ can be found in Appendix H. The MATLAB code used to cluster the hypothesis space can be found in Appendix I. The resulting set of clusters comprise the set of discrete hypothesis objects (Figure 4-25) that will be compared to p_{design}.

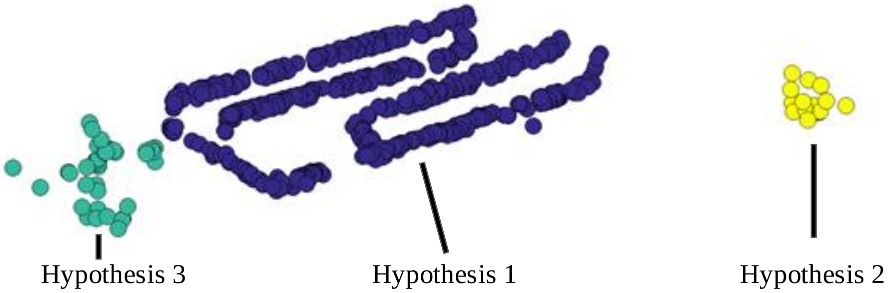
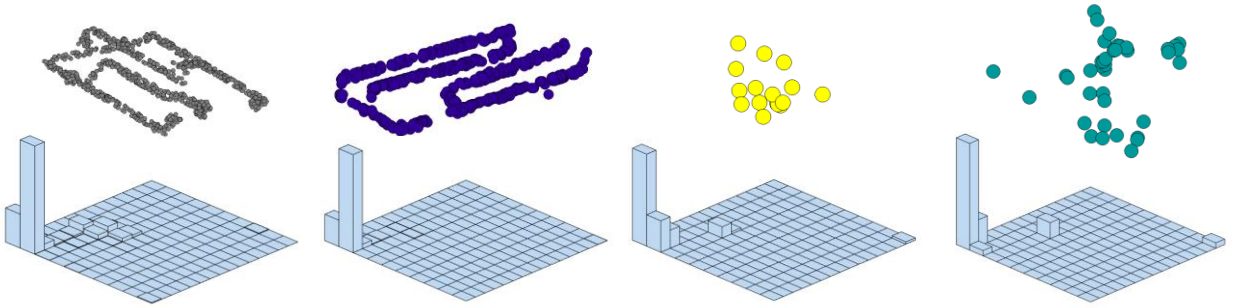


Figure 4-25: Set of discrete hypothesis objects clustered from hypothesis space

4.5.5 BoF comparison and final extraction

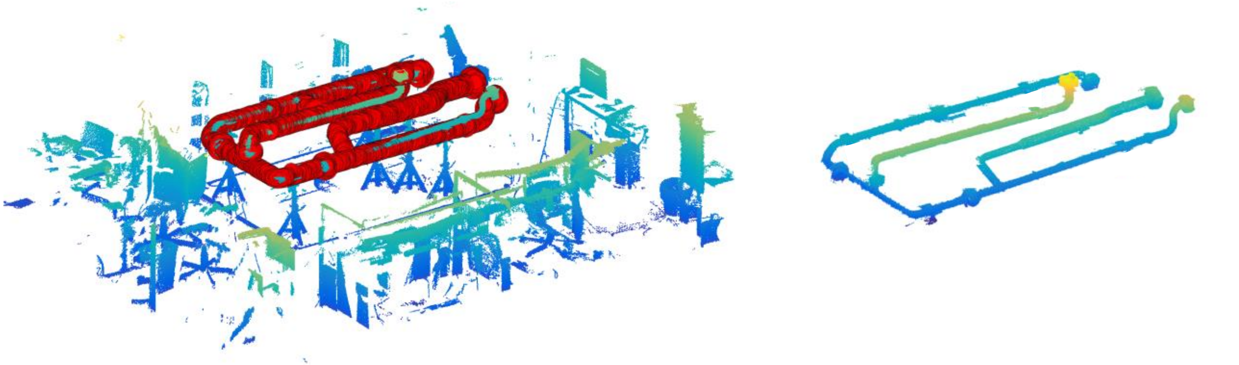
The set of discrete hypothesis objects provided by DBSCAN are now each compared to the design file (Appendix J). The curvature descriptions ($k1$ and $k2$) associated with the points in each cluster are binned into bivariate histograms (Figure 4-26). These histograms are the BoF for each hypothesis and describe the frequency by which each type of curvature occurs in each cluster. Each cluster's histogram is compared to the BoF formed from p_{design}, and the cluster with the smallest difference, i.e., greatest similarity, is selected as the location of the as-built object of interest. The bin size used to the create the histograms is proportional to the size of p_{obj}.



p_{design} BoF Hypothesis 1 BoF Hypothesis 2 BoF Hypothesis 3 BoF

Figure 4-26: BoF for all hypothesis objects along with the BoF for the sample of the 3D CAD design file

Nearest neighbours of the selected cluster provide a rough extraction. A global-to-global registration using principal component alignment and iterative closest point (ICP) (Besl and McKay 1992; Nahangi and Haas 2014) (Appendix K) is performed using P_{design} and the newly extracted rough as-built object of interest. This process registers P_{design} with the coordinate system of the P_{scan} (Figure 4-27a). Finally, retrieving points from P_{scan} that are within threshold distance of points in the registered P_{design} results in the final as-built object of interest extraction (Figure 4-27b).



(a) input files registered (b) extracted as-built pipe spool of interest

Figure 4-27: Registration and extraction, (a) the 3D CAD design file registered to the pipe spool of interest in the point cloud scan and (b) the extracted pipe spool of interest

4.6 Pipe spool recognition using a curvature-based shape descriptor validation

The algorithm was tested on two different point clouds. The first point cloud (Figure 4-28) is a collection of 11 synthetic pipe spools (Figure 4-29) introduced into a real laser scanned fabrication facility with a total of 7 million points before plane removal. The second is a real laser scanned point cloud of a pipe spool in a university laboratory (Figure 4-18) with a total of 1.1 million points before plane removal. For every experiment, the size of p_{design} was set to 1000.

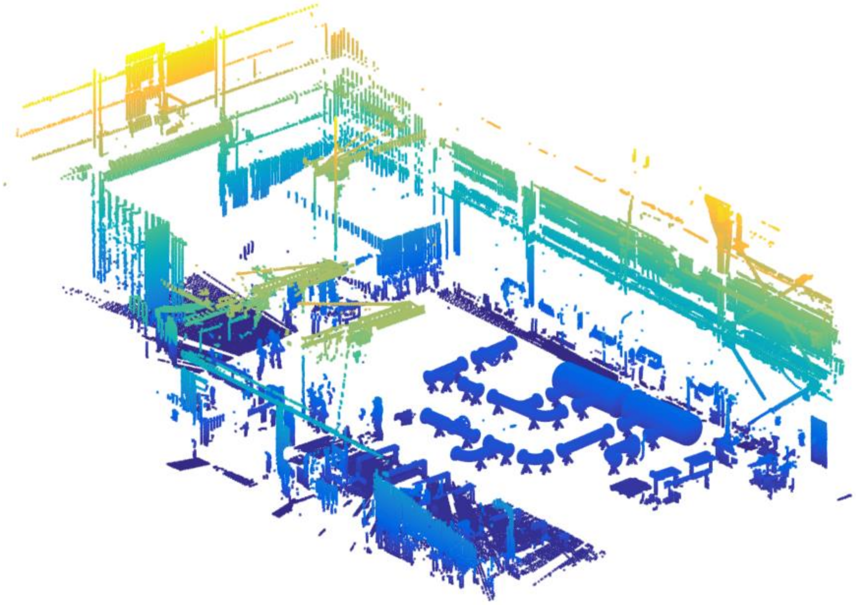


Figure 4-28: Collection of synthetic pipe spools in point cloud scan of fabrication facility with planes removed

4.6.1 Synthetic pipe spool location and extraction

4.6.1.1 Test setup

In order to test the methodology's ability to differentiate between similar pipe spools in a point cloud during extraction, 11 synthetic pipe spools are each extracted in turn from a scan of a fabrication facility (Figure 4-28). $size(p_{scan})$ is varied to illustrate the improvement in the method's ability to differentiate between similar hypotheses as the number of points characterized in the scan increases. For each extraction, the size of p_{scan} is set by using Eq.4-10 with $size(p_{obj}) = \{50, 100, 200, 400, 800\}$.

Therefore, approximately $size(p_{obj})$ points are sampled off of the pipe spool of interest in the point cloud scan. Curvature filter parameters were set at, $\theta_{error} = 0.4$ and $\theta_{sim} = 0.0025$, but were not critical to separate the pipe spools from the background as the accuracy of curvature characterization for spool points was artificially high.

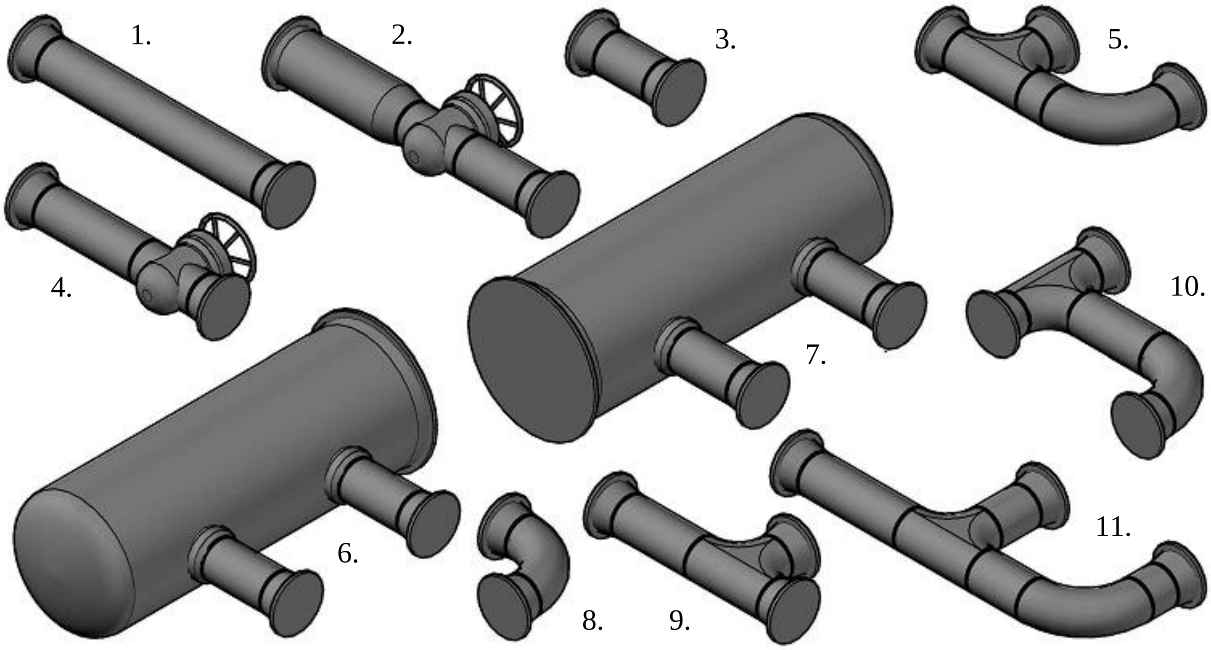


Figure 4-29: Collection of 3D CAD design files for synthetic pipe spools and their corresponding pipe spool identification numbers

Pipe Spools 3, 6, 7, and 8 (Figure 4-29) were specifically included in the analysis to illustrate the extraction situations with which the algorithm would struggle. Pipe Spools 3 and 8 are small simple components that exist as subsets within other spools, e.g., Pipe Spools 2 and 9 containing straight sections which may be mistaken for pipe spool 3. Pipe Spools 6 and 7 will be a challenge because they are only differentiable by their absolute orientation in space, but are identical in their curvature descriptions.

4.6.1.2 Results and discussion

The results for the series of synthetic pipe spool extraction tests can be found in Table 4-4. Correct extraction was defined as the search query P_{design} being accurately registered to the corresponding pipe spool in P_{scan}. Incorrect registrations are defined as the search query P_{design} being incorrectly registered to a pipe spool in P_{scan} not corresponding to the search query. For instances of incorrect registration, the erroneous pipe spool match with the highest share of extraction attempts was reported next to each search query.

As $size(p_{obj})$ increased, the algorithms ability to differentiate between pipe spools increased accordingly. Without the four difficult case pipe spools mentioned, the average successful extraction rates were 60% 69% and 90% for $size(p_{obj}) = \{50, 200, 800\}$, respectively. For incorrect registrations, each search query tended to incorrectly match to a small set of incorrect hypotheses. As the object of interest sample increased, the incorrect registration cases for each search query tended to be comprised of a single "most similar" alternative hypothesis. There were no instances in which the search query P_{design} was mistakenly registered to clutter that was not a pipe spool. For actual industrial applications the observed BoF comparison accuracy is adequate because incorrect registration will not necessarily equate to a failed extraction. After registration to the false match occurs, a simple registration quality of fit test would reveal an incorrect match, and the algorithm could attempt registration to the hypothesis cluster with the second highest similarity, and so forth until an acceptable registration quality of fit identifies the correct spool.

Table 4-4: Result of varying object of interest sample size on extraction performance for synthetic dataset

* Search queries designed to confuse algorithm, based on similarity to other pipe spools (see Figure 4-29).

$^+$ Average excludes search query spool IDs marked with *.

$size(p_{obj})$ = 50				$size(p_{obj})$ = 200				$size(p_{obj})$ = 800			
Correct Extraction		Incorrect Registration		Correct Extraction		Incorrect Registration		Correct Extraction		Incorrect Registration	
Search Query Spool ID	% of Attempts	To Spool ID	% of Attempts	Search Query Spool ID	% of Attempts	To Spool ID	% of Attempts	Search Query Spool ID	% of Attempts	To Spool ID	% of Attempts
1	100%	-	-	1	100%	-	-	9	100%	-	-
2	90%	4	10%	2	100%	-	-	11	100%	-	-
11	70%	5	30%	11	90%	5	10%	1	100%	-	-
5	60%	10	30%	7*	80%	6	80%	2	100%	-	-
6*	50%	7	50%	5	80%	10	10%	4	90%	2	10%
8*	50%	10	30%	4	50%	2	50%	5	90%	10	10%
4	40%	2	50%	8*	50%	5	50%	8*	80%	5	20%
7*	40%	6	60%	6*	40%	7	60%	6*	50%	7	50%
10	40%	5	30%	9	30%	11	40%	10	50%	5	50%
9	20%	11	50%	10	30%	5	40%	7*	40%	6	60%
3*	0%	10	50%	3*	0%	9	70%	3*	0%	9	50%
Average	51%			Average	59%			Average	73%		
Average$^+$	60%			Average$^+$	69%			Average$^+$	90%		

4.6.2 Real pipe spool location and extraction

4.6.2.1 Test setup

The algorithm was also tested on a real laser scanned point cloud of a single pipe spool in a university laboratory (Figure 4-18). The scan data was collected using a FARO laser scanner (Table 4-1). A sensitivity study of the parameters (1) FAR and (2) size of p_{obj} was conducted to test how the methods ability to extract pipe spool objects is effected by the sample size characterized from P_{scan} as well as the percentage of points accepted or rejected by the curvature filter. FAR was varied from 10% to 90% at 10% increments. Size of p_{obj} was varied from 25 to 1600 quadratically.

FAR is dictated by the θ_{sim} and θ_{error} chosen. The noisier the data, the lower the corresponding acceptance rate for each threshold pair. This relationship is 3D imaging device specific. For the laser scanner used to conduct the experiment, the optimum pair of thresholds was determined for each acceptance rate empirically using a separate sensitivity analysis (Appendix C and D). This relationship is illustrated in Figure 4-30.

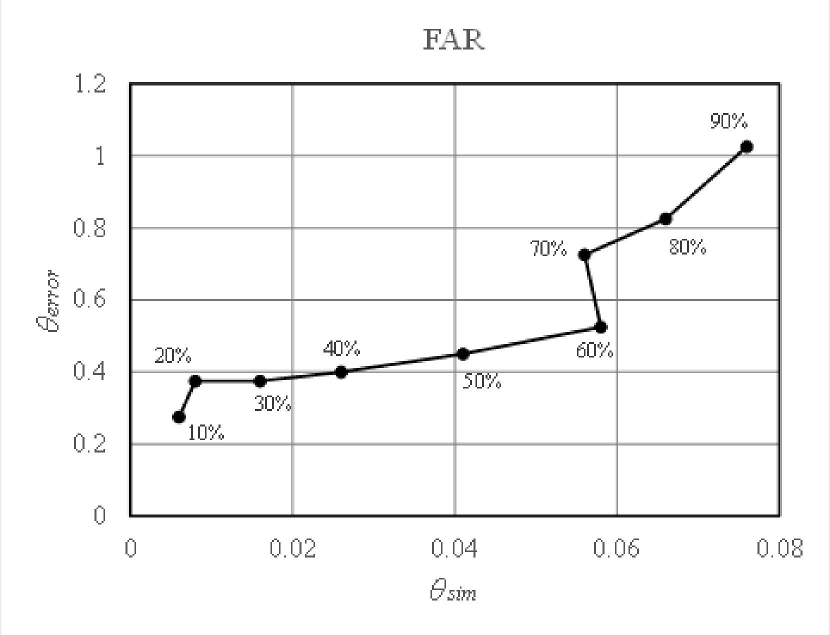


Figure 4-30: optimal combinations of θ_{error} and θ_{sim} and associated filter's acceptance rates; determined empirically through sensitivity study; for FARO LS 880 HE scanner

4.6.2.2 Results and discussion

The results for the real laser scanned pipe spool extraction tests can be found in Table 4-5. A large component of the execution time is comprised of curvature characterization and rises linearly with size of p_{obj}. Ultimate extraction accuracy was highest when the size of p_{obj} was high and the hypothesis space FAR was low. The large amount of points sampled allowed the filter to reject a greater amount of points whilst maintaining a dense enough hypothesis space for successful clustering and registration. Extraction was poor when the size of p_{obj} was low and the FAR was low because the hypothesis space was too sparse for successful clustering and subsequent registration. When the FAR was high the extraction was poor because, like the original scan, the clutter in the hypothesis space did not allow for location of the search query. Extraction success rates varied from 0% to 100%.

Table 4-5: Sensitivity analysis for pipe spool extraction, parameters varied: Hypothesis space FAR and object of interest sample

		FAR									Execution Time (min)
		10%	20%	30%	40%	50%	60%	70%	80%	90%	
$size(p_{obj})$	25	0%	0%	0%	10%	50%	50%	20%	30%	10%	1.1
	50	0%	30%	80%	60%	60%	60%	40%	40%	30%	1.3
	100	20%	40%	60%	70%	70%	40%	50%	30%	20%	1.9
	200	20%	90%	70%	60%	70%	60%	30%	40%	60%	2.9
	400	70%	50%	90%	60%	60%	80%	60%	50%	20%	5.0
	800	80%	100%	80%	90%	70%	70%	70%	70%	40%	9.2
	1600	100%	100%	100%	100%	90%	80%	60%	70%	60%	17.6

4.7 Discussion of recognition methodology’s interface with subsequent analysis procedures

The pipe spool recognition methodology presented in this section completes the automated visual inspection workflow described in Sections 1.1 and 3.2 (Figure 4-31). However, the intricacies of the transition from recognition to geometrical analysis need to be further studied. For the final extraction process described in Section 4.5.5 and discussed for case studies in Sections 4.6.1.2 and 4.6.2.2, the assumption is that the as-built pipe spool being extracted was fabricated within tolerance and does not

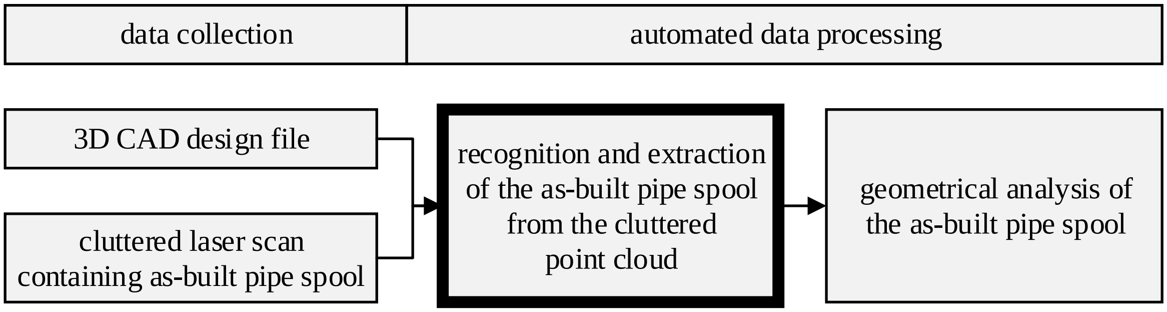

data collection
automated data processing
3D CAD design file
cluttered laser scan containing as-built pipe spool
recognition and extraction of the as-built pipe spool from the cluttered point cloud
geometrical analysis of the as-built pipe spool

Figure 4-31: The automated workflow from pipe spool visual inspection

deviate geometrically from the design file. However, the extraction process would be more complicated if it did deviate geometrically from the design file. Specifically, the process of registering the design file to the object of interest and the subsequent point extraction might be compromised as these two steps rely on congruence between the global configurations of the design file and object of interest. Besides this, the methodology is not entirely undermined by geometrical defects so long as the BoF for the as-built pipe spool remains largely unaffected. If the BoF of the as-built pipe spool remains similar to the BoF of the design file despite geometrical defects, the as-built pipe spool will still be detected as a hypothesis cluster (Section 4.5.5). Expansions need to be made to the presented methodology to further understand and accommodate situations involving geometrical defects of varying degrees.

Once the points representing the as-built pipe spool of interest have been identified, those points are copied to a separate file and enables the execution of the geometrical analyses presented in Section 3.2.2.

5.0 Curvature identification using low-cost range-cameras

The search and extraction algorithm presented in Section 4.5 relies on accurate curvature characterization of the point cloud data. The application of low-cost range-cameras to object recognition is desirable as it would minimize the cost of purchasing equipment. However, low-cost range-cameras have limited accuracy. Results from a Kinect sensor accuracy study concluded that: (1) Random error of depth measurements increases quadratically with increasing distance from the sensor and reaches 4 cm at the maximum sensing distance of 5 meters. (2) The depth resolution decreases quadratically with increasing distance from the sensor. The point spacing in the depth direction (along the optical axis of the sensor) is as large as 7 cm at the maximum range of 5 m as can be seen in (Khoshelham and Elberink 2012). Point density of the collected scans are also adversely affected by distance. Considering the density of the resulting point cloud to be the number of points per unit area, the point density on the XY plane is inversely proportional to the squared distance from the imaging device (Khoshelham and Elberink 2012). Therefore, an object being detected by the device will be represented by fewer and fewer points as it moves away from the sensor.

Accuracy of curvature characterization is related to the accuracy of the 3D imaging device used, the amount of surface noise in the data, the distance of the object of interest from the sensor, the size of the object of interest, and the size of the nearest neighbourhood parameter in the curvature characterization algorithm described in Section 4.3. An experiment was set up to measure the effects of these variables on the execution of the algorithms and to explore the viability of using low-cost range-cameras for accurate object recognition and extraction. Optimal nearest neighbourhood sizes, the associated accuracy of curvature characterization, and finally, the accuracy of object recognition were all determined for 3D imaging sensors collecting data of pipe-objects at different distances.

5.1 Experimental setup

Data acquisition was performed using Kinect for Windows v1, Kinect for Windows v2 (discussed in Section 3.1), and a FARO LS 840 HE laser scanner. Technical properties and features of the Kinect range-cameras used in this study are summarized in Table 3-1. Range and accuracy specifications for the FARO laser scanner used in this study are summarized in Table 4-1.

The object extraction algorithm requires that the curvatures of the objects in the cluttered scene be characterized accurately enough so that differentiation and the appropriate extraction can be performed. To test the algorithm’s ability to differentiate between similar curvatures and the objects they represent, four PVC pipes of various raddii were used as objects of interest (Table 5-1). The first 3D imaging device was positioned at one end of a hallway. As seen in Figure 5-1, single perspective images were taken of the 4 pipes starting at a distance from the imaging device of 0.5 m and at each 0.25 m interval to a maximum distance of 3.75 m. The process was repeated for each of the other two 3D imaging devices. A total of 42 images were generated.

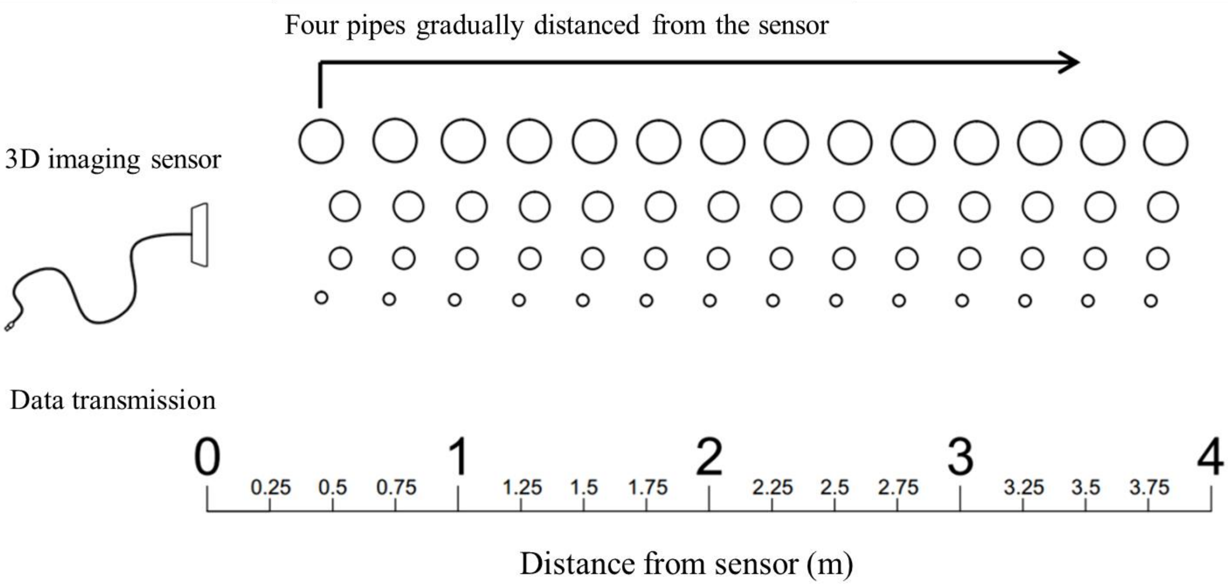


Figure 5-1: Experimental setup plan view

Table 5-1: Properties of the pipes used for the experiments

Descriptor	Pipe 1	Pipe 2	Pipe 3	Pipe 4
Index used for results analysis	P1	P2	P3	P4
Actual radius (*cm*)	2.41	4.43	6.09	8.775
Length (*cm*)	40	40	40	40

5.2 Optimal nearest neighbourhood specification analysis

When running the object extraction algorithm, the user must know what nearest neighbourhood size to specify. The known user inputs are the type of imaging device being used and the curvatures of the search query. In order to explore the relationship between nearest neighbourhood size and the accuracy of the associated output, the images collected are manually segmented, and each individual pipe point cloud is analyzed using a series of nearest neighbourhood sizes. The accuracy of the curvature characterization is studied.

Each of the four pipes was extracted by manually segmenting the collected images resulting in 168 individual pipe point clouds. The curvature characterization algorithm was run on each of the extracted point clouds using 10 nearest neighbourhood sizes (1-10 cm) generating 1680 curvature histograms. The result is a histogram of k_1 curvatures. Figure 5-2 shows sample curvature histograms for the 6.09 cm radius pipe at 1.5 m from the sensor.

The curvature histograms for the entire range of distances are combined into summary charts for each device, pipe radius and nearest neighbourhood size combination. Figure 5-3 shows the 5 cm nearest neighbourhood size curvature characterization results for the 6.09 cm radius pipe data collected by the Kinect 2 at 0.5-3.75 m from the sensor.

Understanding that each individual pipe point cloud represents a single curvature type that would be entered as part of a search query in the object extraction algorithm, the histogram must be abstracted to an actual pipe radius (3D CAD model curvature) and curvature acceptance threshold (Figure 5-4). The curvature acceptance threshold must be as small as possible to discriminate against clutter in the scene

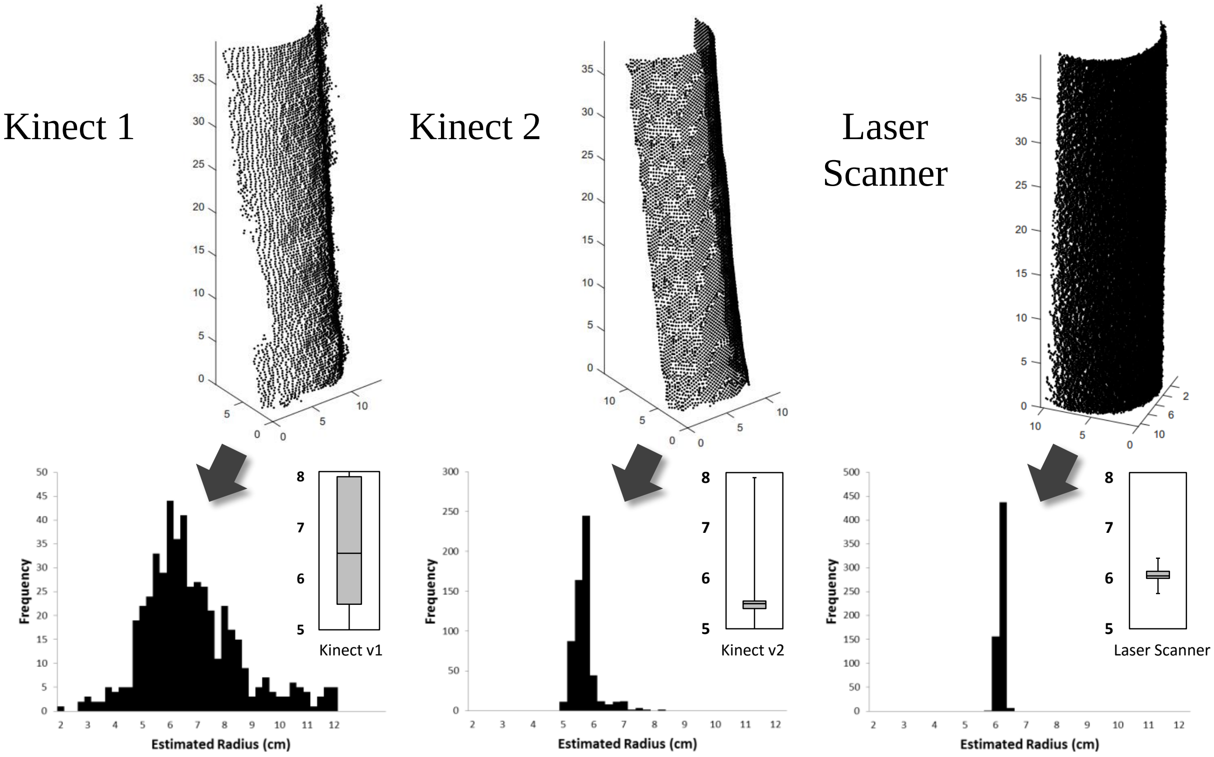


Figure 5-2: Individual pipe point clouds for P3 at 1.5 meters from sensor and generated curvature histograms and distribution box plot

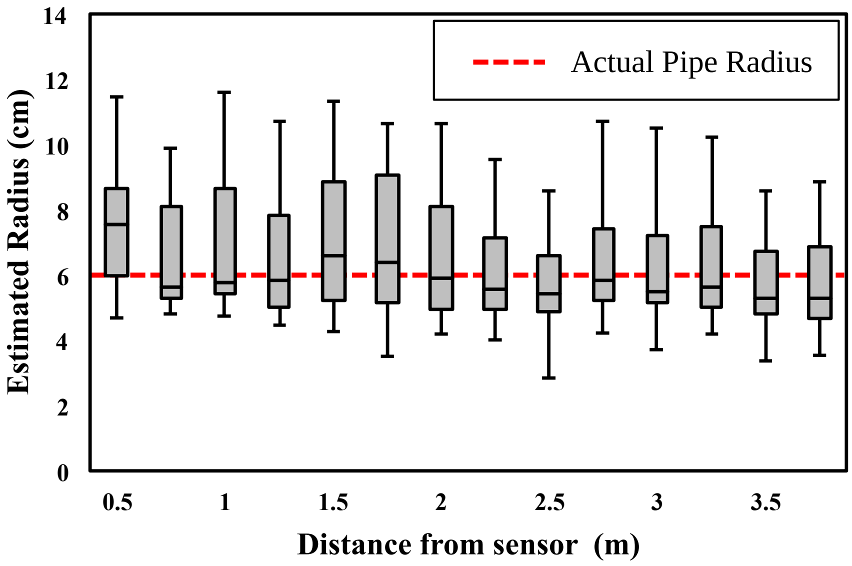


Figure 5-3: Curvature characterization for P3 with nearest neighbourhood size 5 cm, Kinect 2 data source

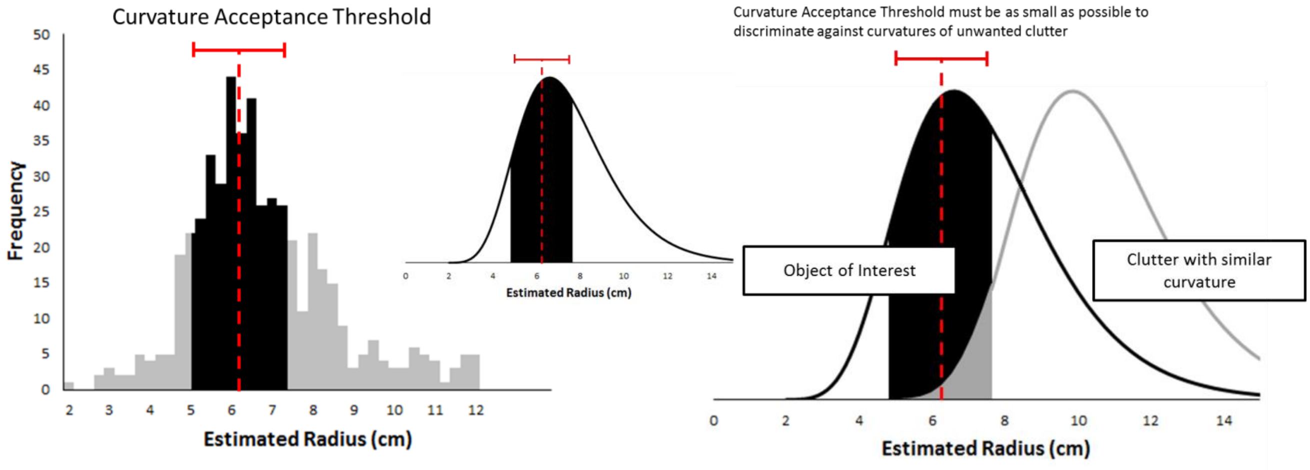


Figure 5-4: Curvature Acceptance Threshold specification for Kinect 1 curvature histogram for 6.09 cm radius pipe at 1.5 meters

that has curvatures similar to those in the search query. For each of the 1680 curvature histograms, the actual pipe radius and curvature acceptance threshold specifications resulting in a 50% point acceptance rate are calculated. The curvature acceptance threshold summary charts are presented in Figure 5-5. Each boxplot represents a 3D imaging device/pipe radius/nearest neighbourhood size combination and is comprised of 14 curvature acceptance threshold values (one for each distance 0.5-3.75 m). Using the curvature acceptance thresholds, the optimal nearest neighbourhood size is determined for each 3D imaging device and pipe radius combination.

The optimal nearest neighbourhood sizes for each 3D imaging device / pipe radius combination were determined by selecting the nearest neighbourhood size that yielded the lowest maximum curvature acceptance threshold. The optimal nearest neighbourhood sizes for the laser scanner were, however, chosen differently because (1) the improvement in accuracy after nearest neighbourhood size 4 cm was minimal and (2) processing time is quicker for smaller nearest neighbourhoods (see Table 5-2). The optimal nearest neighbourhood sizes are shown in Table 5-2. The resulting curvature characterizations obtained from implementing the optimal nearest neighbourhood sizes are summarized in Figure 5-6. Using the median as the discrete curvature estimate, the average error of characterization was 18% for Kinect 1, 10% for Kinect 2, and 2% for the laser scanner. The associated curvature acceptance threshold –

indicator of curvature estimate spread and inverse discriminative ability - was ± 2 cm for Kinect 1, ± 1.1 cm for Kinect 2, and ± 0.25 cm for the laser scanner.

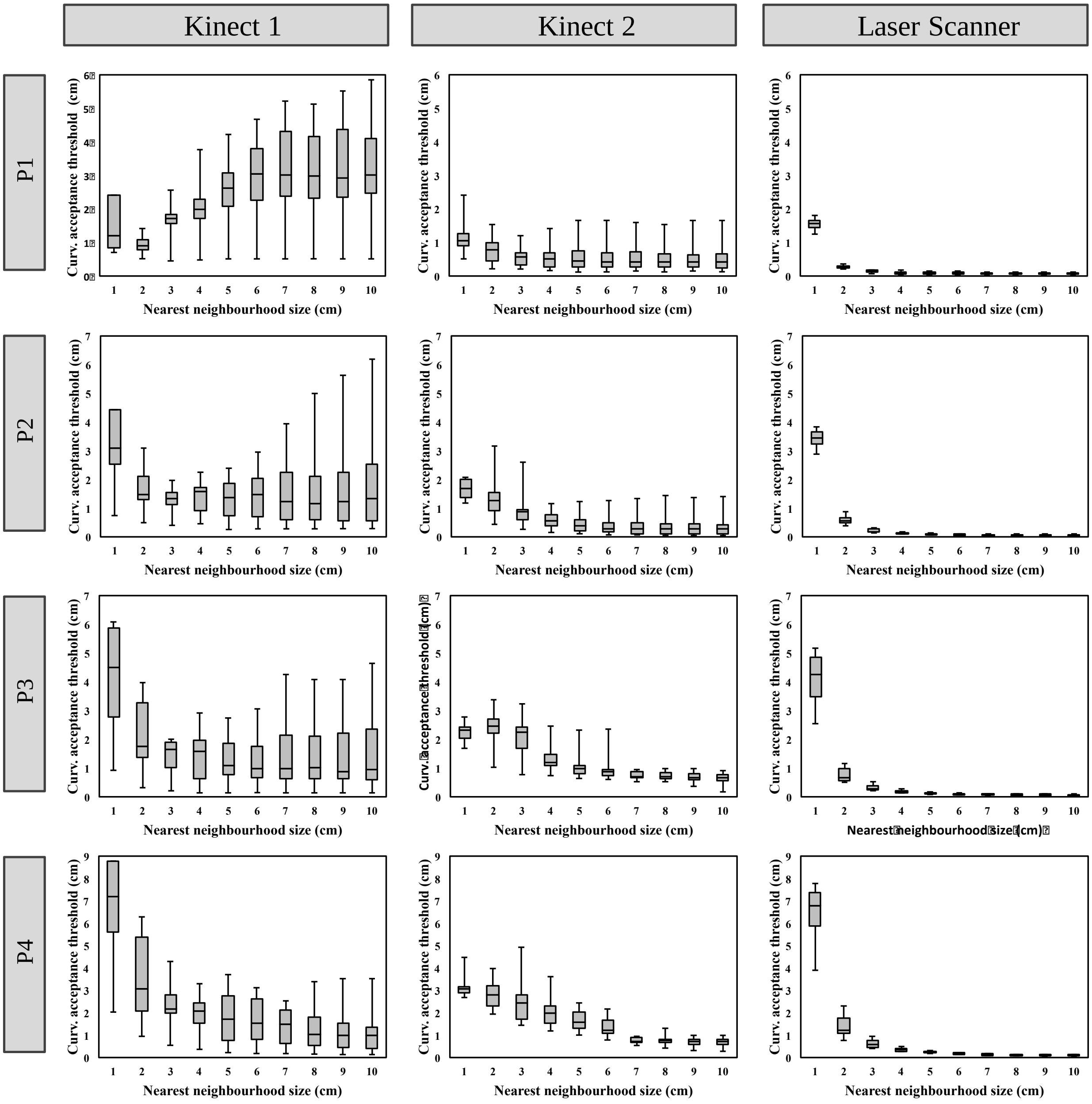


Figure 5-5: Curvature acceptance threshold for 50% point acceptance rate summary charts for nearest neighbourhood sizes 1-10 cm, for Kinect 1, Kinect 2, and laser scanner and pipes with radius 2.41 cm (P1), 4.43 cm (P2), 6.09 cm (P3), and 8.775 cm (P4)

Table 5-2: Optimal nearest neighbourhood size for extracting pipe with radius 2.41 cm (P1), 4.43cm (P2), 6.09 cm (P3), and 8.775 cm (P4) using various sensors

3D imaging device	P1	P2	P3	P4
Kinect 1	2 cm	3 cm	3 cm	7 cm
Kinect 2	3 cm	4 cm	10 cm	9 cm
Laser Scanner	4 cm	4 cm	4 cm	4 cm

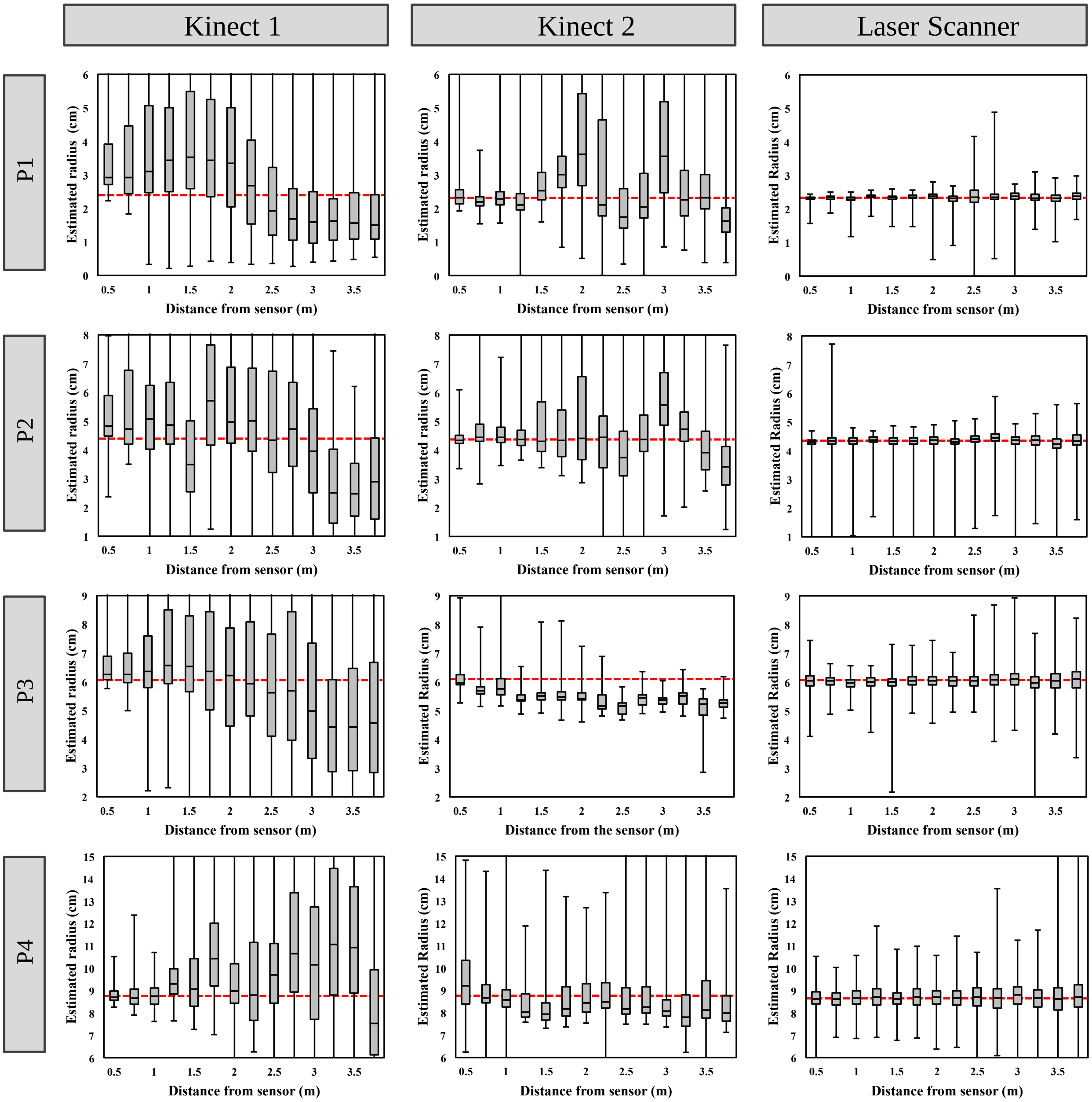


Figure 5-6: Curvature characterization using optimal nearest neighbourhood sizes, for Kinect 1, Kinect 2, and laser scanner for pipes with radius 2.41 cm (P1), 4.43 cm (P2), 6.09 cm (P3), and 8.775 cm (P4)

5.3 3D imaging device pipe extraction results

The object extraction algorithm was executed on the original images collected by the 3D imaging devices. Using the optimal nearest neighbourhood sizes and their associated curvature acceptance thresholds, the algorithm's ability to differentiate between pipes and extract the desired search query were determined. For each 3D imaging device/search query combination, the algorithm would extract a single object from the cluttered scene. The results are presented in Table 5-3, where the percentages represent the proportion of images (distance from sensor 0.5-3.75 m) for each 3D imaging device that yielded each type of extracted object with the specified search query.

Table 5-3: Pipe extraction statistics, percentages represent the proportion of images (distance from sensor 0.5-3.75 m) for each 3D imaging device that yielded each type of extracted object with specified search query.

Search Query	2.41 cm Pipe (P1)				4.43 cm Pipe (P2)				6.09 cm Pipe (P3)				8.775 cm Pipe (P4)			
Extracted Object	P1	P2	P3	P4	P1	P2	P3	P4	P1	P2	P3	P4	P1	P2	P3	P4
Kinect 1	21%	29%	14%	36%	0%	43%	36%	21%	0%	0%	50%	50%	0%	0%	7%	93%
Kinect 2	79%	21%	0%	0%	0%	71%	29%	0%	0%	0%	100%	0%	0%	0%	0%	100%
Laser Scanner	100%	0%	0%	0%	0%	100%	0%	0%	0%	0%	100%	0%	0%	0%	0%	100%

The laser scanner achieved perfect discriminative ability for extracting the specified pipes from the captured images, extracting the search query pipe correctly in 100% of the executions. The Kinect v2 outperformed the Kinect v1, in that when queried, the two larger pipes were extracted correctly 100% of the time, while the smaller of the two pipes were extracted correctly 75% of the time on average. The Kinect v1 performed poorly, with an average correct extraction rate of only 52%. The extraction process, when providing an erroneous object extraction, consistently demonstrated a bias towards pipes of larger radius and larger surface area.

Although the range-cameras did not demonstrate sufficiently reliable object-of-interest discrimination for industrial applications, the apparent trend in the 3D imaging device's ability to differentiate between similar curvatures suggests that future iterations of Kinect devices could achieve comparable capabilities to the laser scanner and would be suitable for input to the automated object extraction framework.

6.0 Summary, conclusions, and recommendations

6.1 Thesis summary

The application of emerging sensor and communication technologies to industrial pipe spool fabrication has proven potential for providing a greater degree of production control. 3D imaging devices, particularly laser scanners, are being used to monitor 3D geometries and check dimensional compliance during fabrication. However, many of these monitoring tools are limited because they require the object of interest being monitored to be manually isolated from the rest of the background data and clutter captured by the indiscriminate laser scanner. Automated information extraction processes that exploit the massive databases created by emerging sensor networks and communication technologies have the potential to substantially improve the availability of information about 3D geometries on construction projects and in turn, improve the execution of components inspection, and progress tracking. In this thesis, the problem of automatically recognizing and isolating pipe spools from their cluttered point cloud scans is studied. Two approaches were developed and evaluated.

6.1.1 RANSAC

RANSAC has proven to effectively recognize simple shape primitives in cluttered point cloud data within the computer vision literature. One of the recognition frameworks in this thesis attempted to generalize the method to recognize pipe spools. Although the basic RANSAC approach has the advantage of being conceptually simple, its direct application to the 3D pipe spool recognition problem is shown to be prohibitively expensive computationally. For the problem of locating a pipe spool in a point cloud of approximately two million points, the sheer volume of random hypothesis locations tested during the recognition process made it impossible to find the correct location of pipe spools of interest.

In an attempt to decrease the number of possible minimal point set matches, a method for discriminately down sampling the point cloud using curvature was presented. The curvature characterization and filtering was able to successfully sample points from a few key areas in the scan, but a subsequent RANSAC execution failed to provide reliable results.

6.1.2 Recognition using a curvature-based shape descriptor

A novel method for extracting pipe spools from cluttered point clouds using local data level curvature estimation, density-based clustering, and bag-of-features matching was developed. The methodology was validated on two test point clouds, demonstrating both an ability to successfully extract pipe spools from point clouds and to differentiate between similar pipe spools in the same point cloud. Successful extraction rates achieved range from 90%-100%.

6.1.3 Curvature identification using low-cost range-cameras

Three 3D imaging devices were used to capture point clouds of pipe objects at distances ranging from 0.5 to 3.75 m. The radii of the pipe objects were characterized and the accuracies of the devices were evaluated. Although the range-cameras did not demonstrate sufficiently reliable curvature discrimination for industrial applications, the apparent trend in the device's ability to differentiate between similar curvatures suggests that future iterations of Kinect devices could achieve much better accuracy and would be suitable for input to the automated object extraction framework.

6.2 Research contributions and conclusions

The contributions of the work are as follows: (1) A number of challenges involved in applying RANSAC to pipe spool recognition are identified. (2) An effective spatial search and pipe spool extraction algorithm based on local data level curvature estimation, density-based clustering, and bag-of-features matching is presented. (3) The accuracy of curvature estimation using data collected by low-cost range-cameras and the viability of use of low-cost range-cameras for object search, localization, and extraction are tested.

Following from these contributions are three conclusions:

(1) One-to-one comparison of minimal point sets for RANSAC applied to finding pipe spools in cluttered point clouds is an inadequate comparison strategy

(2) Curvature can be the basis for a local shape descriptor that can successfully differentiate between a pipe spool object and surrounding clutter in a typical industrial fabrication facility scene

(3) The success of object recognition by the curvature-based recognition algorithm is dependent on the accuracy of the 3D imaging device used to collect the data

6.3 Limitations

6.3.1 Recognition using a curvature-based shape descriptor

Many of the parameters in the curvature-based recognition algorithm are 3D imaging device specific. Currently a recalibration of these parameters is necessary for each new scanner used. The complexity and specificity of the algorithm is prohibitive to generalization for recognition of other non-pipe spool industrial assemblies. The method is only suitable for pipe spool recognition. Spools comprised mostly of flanges, valves and instrumentation were not tested; it is expected the algorithm would not perform as well on this subclass of spools.

Only two case studies were used to validate the algorithm. Case studies did not include situations of substantial occlusion.

6.4 Future work

Recommended future work includes, test alternative point sets and descriptor types for the RANSAC framework and explore alternative data structures such as hash tables to facilitate the comparison of minimal point sets.

In applying the curvature-based recognition algorithm, for implementations not prioritizing speed, studying the improvement in performance of a variable NN_p size characterization is another recommended area for future research. The 3D imaging device specific parameters in the algorithm change as the level of noise in the data changes. In order to understand these parameters in more detail, a metric for quantifying the noise that affects them needs to be determined. If an effective metric is found, a method for automating the determination of these 3D imaging device specific parameters should be created. The curvature characterization algorithm is well suited for parallel computing and therefore future iterations of the code should leverage parallel graphical processing unit (GPU) processing to reduce computation time. Finally, the performance of the proposed method should be compared in future research with a library of other popular shape descriptor methods such as object recognition based on spin-images (Johnson and Hebert 1999).

Appendix A

MATLAB code: Major plane removal from laser scans of industrial facilities

```
function [Scan_PlanesRemoved] = RemovePlanes( Scan )
%function [Scan_PlanesRemoved] = RemovePlanes(Scan) removes the major planes from within Scan,
%
%       coded with the intention of removing floors, cieling and walls from within
%       a fabrication facility
%
%       input:
%       Scan = input point cloud (units in cm)
%
%       Major plane removal is performed in two clustering steps:
%       (1) Guassian sphere is clustered to identify sets of parallel
%       planes
%       (2) Sets of parallel planes are spatially segemented to isolate
%       individual planes
%
Scan_pt=pointCloud(Scan);
ScanNormals=pcnormals(Scan_pt,12); %estimate surface normals for each point in the point cloud
ClusteringSpace=[Scan_pt.Location,ScanNormals];
ClusteringSpace_reduced=datasample(ClusteringSpace,20000); %sample 20,000 points from point cloud
for Gaussian sphere clustering operation
%(1) Guassian sphere is clustered to identify sets of parallel planes
[Class,trash]=dbscan(ClusteringSpace_reduced(:,4:6),1000,0.02); %1000 is the minimum number of
points representing a set of parallel planes and 0.02 is 2 times the SquareRoot(surface area of a
sphereical cap of the Gaussian sphere created by a 10 degree normal vector deviation divided by
1000)
ClassList=unique(Class);
% ---- align principal axes of the scan with the xyz axes
for i=2:length(ClassList);
    R=[1,0,0;0,1,0;0,0,1];
    if dot(mean(ClusteringSpace_reduced(Class==ClassList(i),4:6)),[0,0,1])<0.2
```

```
angle=atan2(norm(cross(mean(ClusteringSpace_reduced(Class==ClassList(i),4:6)),[1,0,0])),dot(mean(
ClusteringSpace_reduced(Class==ClassList(i),4:6)),[1,0,0]));
    R=vrrotvec2mat([0;0;1;angle]);
    break
    end
end
% ----
ClusteringSpace_reduced(:,1:3)=ClusteringSpace_reduced(:,1:3)*R;
Scan_pt=pointCloud(Scan_pt.Location*R);
DirectionVector=zeros(length(ClassList)-1,3);
for i=2:length(ClassList);
    DirectionVector(i-1,:)=mean(ClusteringSpace_reduced(Class==ClassList(i),4:6))*R; %calculate
the mean normal vector direction for each cluster of the Gaussian sphere
    ParallelPlanes=ClusteringSpace_reduced(Class==ClassList(i),1:3);
    %(2) Sets of parallel planes are spatially segemented to isolate individual planes
    [Class_sub,trash]=dbscan(ParallelPlanes,100,100); %planes are represented by at least 100
points, and are spatially seperated by at least 100cm
    ClassList_sub=unique(Class_sub);
    % --- remove each plane from the scan
    for e=2:length(ClassList_sub);
        roi=[min(ParallelPlanes(Class_sub==ClassList_sub(e),1:3))'-
20,max(ParallelPlanes(Class_sub==ClassList_sub(e),1:3))'+20]; %region of interest
        sampleIndices=findPointsInROI(Scan_pt,roi);
        sub=select(Scan_pt,sampleIndices);
        left=select(Scan_pt,setdiff(1:length(Scan_pt.Location),sampleIndices));
        range(sampleIndices)
        [trash1,trash2,outlierIndices]=pcfitplane(sub,2.5,DirectionVector(i-1,:),10); %Plane
removal parameters: 2.5 cm maximum spatial deviation of points on a plane, and maximum 10 degree
normal vector deviation
        sub=select(sub,outlierIndices);
        Scan_PlanesRemoved=pointCloud([left.Location;sub.Location]);
    end
    % ---
end
end
```

Appendix B

MATLAB code: Pipe spool recognition and extraction main function

```
function [ ExtractedObject, transformation ] = ExtractSpool( Design_File, Scan, Scan_Sample_Size,
acceptanceRate)
%function ExtractSpool is a pipe spool recognition method used for locating
%and extracting pipe spools from cluttered point clouds.
%
%   input:
%   Design_File = 3D CAD design file point cloud of pipe spool of interest
%   Scan        = cluttered point cloud scan that includes the pipe spool
%                 of interest
%   Scan_Sample_Size = number of points from Scan that will go through
%                      curvature characterization and be used to find the
%                      pipe spool of interest
%   acceptanceRate   = number of points off the object of interest in Scan
%                      to be accepted through the curvature-based filter.
%                      (used to set filter thresholds)
%
%The extraction process is performed in five steps:
%   (1) The points in Scan are filtered using a curvature-based shape
%       descriptor
%   (2) Points in the hypothesis space are clustered into hypothesis
%       objects
%   (3) Using BoF, each hypothesis object is compared to the 3D CAD design
%       file
%   (4) The 3D CAD design file is registered to the hypothesis object most
%       similar to the 3D CAD design file
%   (5) The pipe spool of interest is extracted from Scan
%
%   *all units in the code are in centimeters

%access filter thresholds (parameters are sensor specific and accessed from
%empirically determined optimal values)
[SimilarityThreshold,ErrorThreshold]=accessThresholds(acceptanceRate);

pdr=0.5; %pdr is the Poisson-disc Sampling Radius used to sample the search space (UNITS)
Design_File_Sample_Size=1000; %Specify the # of points to analyze in the Design_File

%Generate a KD Tree for Design_File such that Nearest Neighbours can
%be found more efficiently
OCKDT_Design_File=KDTreeSearcher(Design_File);
scale=DetermineScale(Design_File,OCKDT_Design_File); %Determine reasonable scale value to use for
curvature characterization

%analyze the Design_File for curvature (K1 and K2 principal curvatures) at
%"Design_File_Sample_Size" number of local surface patches
Design_File_Curvature_Signature=CurvSearch(Design_File,OCKDT_Design_File,pdr,Design_File_Sample_S
ize,scale);

%Convert the Design_File_Curvature_Signature into a polygon that is used to

%Generate a KD Tree for the search space such that Nearest Neighbours can
%be found more efficiently
OCKDT_Scan=KDTreeSearcher(Scan);

%analyze the Scan for curvature (K1 and K2 principal curvatures) at
%"Scan_Sample_Size" number of local surface patches
Scan_Curvature_Probe=CurvSearch(Scan,OCKDT_Scan,pdr,Scan_Sample_Size,scale);

%Classify and filter the analyzed points using 3NN and distance threshold
%"SimilarityThreshold"
Object_Hypothesis_Map1=DesiredCurvNN(Scan_Curvature_Probe,Design_File_Curvature_Signature,Similar
ityThreshold);
%Classify and filter the analyzed points using circle fit error and error threshold
%"ErrorThreshold"
Object_Hypothesis_Map=Object_Hypothesis_Map1(Object_Hypothesis_Map1(:,6)<ErrorThreshold,:); %this
is where you filter using error
```

```
%Cluster the Object_Hypothesis_Map and generate a number of discrete
%hypotheses for the object of interest
[DBSCAN_NumPts,DBSCAN_eps]=determineClustParam(Design_File,length(Scan),Scan_Sample_Size,acceptan
ceRate);
Hypotheses=GenerateHypotheses(Object_Hypothesis_Map,DBSCAN_NumPts,DBSCAN_eps);
HypothesesNN=cell(length(Hypotheses),3);
HypothesesMatch=cell(length(Hypotheses),5);

%Calculate the similarity between each hypothesis object and
%Design_File by comparing frequencies in each BoF
%AND extract NN for each cluster in order to have hypothesis with greater
%point density for viewing
for i=1:length(Hypotheses)

[HypothesesMatch{i,1},HypothesesMatch{i,2}]=ComparePlots(Design_File_Curvature_Signature,Hypothes
es{i});
    HypothesesNN{i,1}=rangesearch(OCKDT_Scan,Hypotheses{i}(:,1:3),2*scale);
    HypothesesNN{i,2}=cell2mat(HypothesesNN{i,1}');
    HypothesesNN{i,3}=unique(HypothesesNN{i,2}','rows');
    HypothesesMatch{i,3}=Scan(HypothesesNN{i,3},:);
    HypothesesMatch{i,4}=Hypotheses{i}(:,1:3);
    HypothesesMatch{i,5}=Hypotheses{i}(:,4:5);
end

%Sort the hypotheses in descending order of similarity to the Design_File
%and plot the top 5 if there are at least 5 hypotheses, if not then plot
%all
[trash,HypothesesMatchReorderedIndex]=sort([HypothesesMatch{:,1}],'descend');
HypothesesMatch=HypothesesMatch(HypothesesMatchReorderedIndex,:);
Hypotheses=Hypotheses(HypothesesMatchReorderedIndex,:);
size(HypothesesMatch{1,3},1)
if size(HypothesesMatch,1)<5
    for e=1:size(HypothesesMatch,1)
        figureTitle=sprintf('Similarity = %d size =
%d',HypothesesMatch{e,1},length(HypothesesMatch{e,4}));
        figure
        showPointCloud(HypothesesMatch{e,3})
        title(figureTitle)
    end
else
    for e=1:5
        figureTitle=sprintf('Similarity = %d size = 
%d',HypothesesMatch{e,1},length(HypothesesMatch{e,4}));
        figure
        showPointCloud(HypothesesMatch{e,3})
        title(figureTitle)
    end
end
% The section below is for registering the Design_File to the cluttered
% point cloud once a rough extraction of points has been obtained.
RoughExtract=HypothesesMatch{1,3};
%downsample to improve registration speed
PC1=datasample(Design_File,2000);
PC2=datasample(RoughExtract,2000);
%register
[Design_File_Fine_Transformed,transformation]=autoReg(PC1,PC2,Design_File);

ExtractedObjectIDX1=rangesearch(OCKDT_Scan,Design_File_Fine_Transformed(:,1:3),2*scale);
ExtractedObjectIDX=cell2mat(ExtractedObjectIDX1');
ExtractedObjectIDX=unique(ExtractedObjectIDX','rows');
ExtractedObject=Scan(ExtractedObjectIDX,:);
end
```

Appendix C

MATLAB code: Retrieve SimilarityThreshold and ErrorThreshold given acceptanceRate as per Appendix D

```
function [ SimilarityThreshold,ErrorThreshold ] = accessThresholds( acceptanceRate )
%function accessThresholds returns two filter thresholds for input
%acceptanceRate. Values derived empiracally for FARO LS 880 HE laser
%scanner
%
%   input:
%   [acceptanceRate] = the curvature filter's acceptance rate, the
%   approximate percentage of points from object of interest that will be
%   accepted by the filter's threshold.
%
thresholds=[0.1,0.006,0.275;
 0.2,0.008,0.375;
 0.3,0.016,0.375;
 0.4,0.026,0.4;
 0.5,0.041,0.45;
 0.6,0.058,0.525;
 0.7,0.056,0.725;
 0.8,0.066,0.825;
 0.9,0.076,0.1.025];

SimilarityThreshold=interp1(thresholds(:,1),thresholds(:,2),acceptanceRate);
ErrorThreshold=interp1(thresholds(:,1),thresholds(:,3),acceptanceRate);


end
```

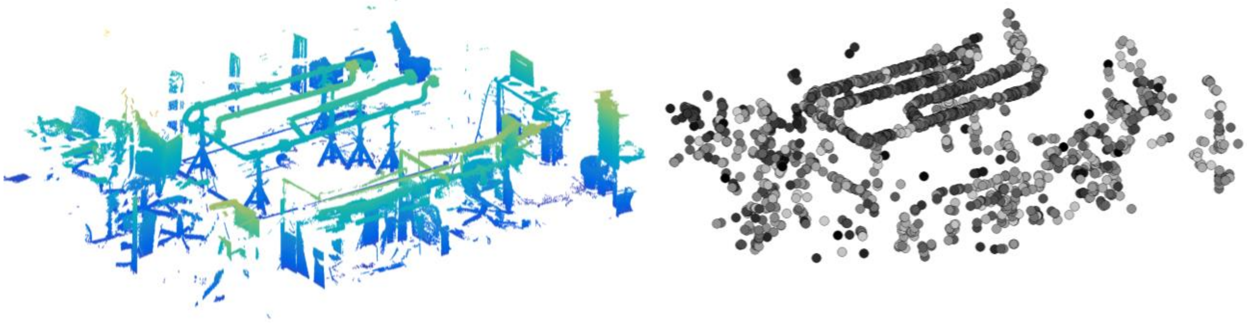
Appendix D

Sensitivity analysis determining optimal thresholds for each filter acceptance rate

The filter acceptance rate (FAR) (i.e., the approximate percentage of points from p_{obj} that will be accepted by the filter) is dictated by the $SimilarityThreshold$ (θ_{sim}) and $ErrorThreshold$ (θ_{error}). This relationship is sensor specific, as the noisier the data, the lower the corresponding acceptance rate for each specific threshold pair. A sensitivity analysis was conducted to determine the optimal filter thresholds for a FARO LS 880 HE scanner. The filtering processes was performed on a typical point cloud (Figure D-1) created by the scanner. The parameter θ_{sim} was varied from 0.0005 to 0.098 and parameter θ_{error} was varied from 0.075 to 1.05. For each parameter pair, two results were obtained, (1) the percentage of all points accepted by the filter, i.e., all points constituting the hypothesis space, that lie on the pipe spool of interest:

$$metric\ 1 = \frac{number\ of\ points\ in\ hypothesis\ space\ on\ pipe\ spool\ of\ interest}{size(hypothesis\ space)}$$

and (2) the filter acceptance rate (FAR) (the approximate percentage of points from p_{obj} that will be accepted by the filter). Both these measures were obtained by manually segmenting the pipe spool of interest from the rest of the hypothesis space. The resulting sensitivity analysis tables can be found in Figure D-2 and Figure D-3.



(a) input scan

(a) hypothesis space achieved using θ_{sim} and θ_{error}

Figure D-6-1: (a) typical point cloud created using a FARO LS 880 HE scanner processed through curvature filter to create (b) hypothesis space

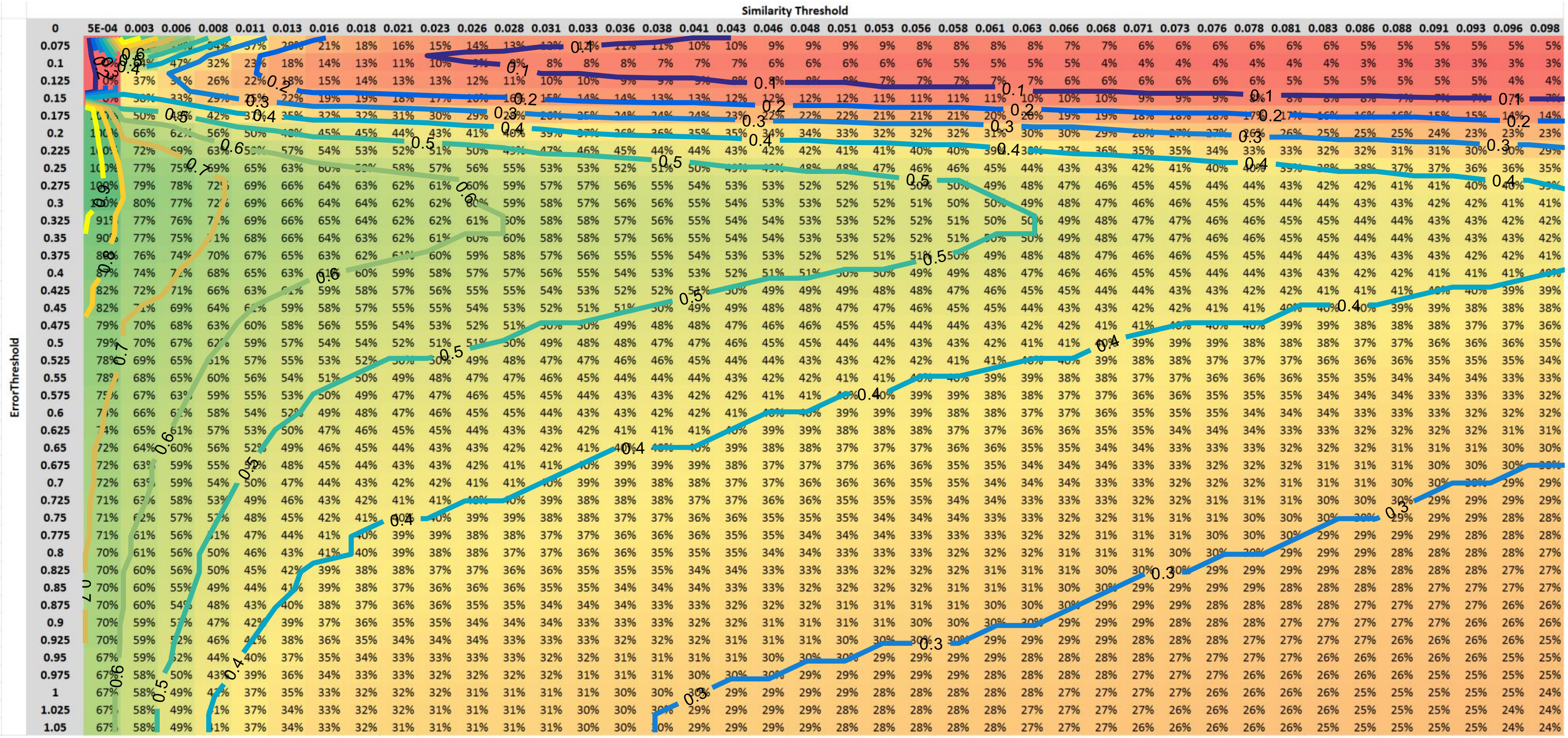
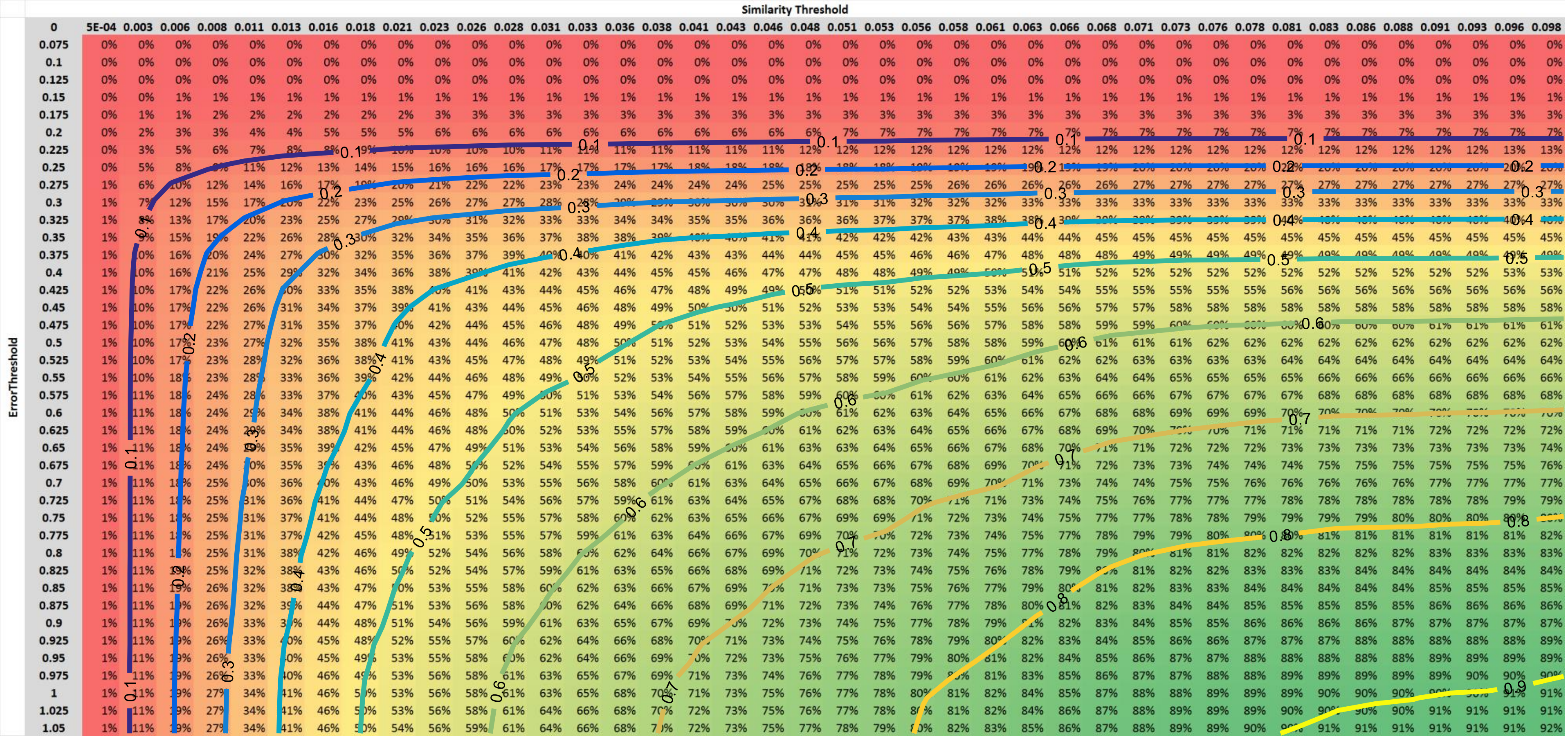


Figure D-6-2: sensitivity analysis change in metric 1 as θ_{sim} and θ_{error} are varied



		Similarity Threshold																																						
	0	5E-04	0.003	0.006	0.008	0.011	0.013	0.016	0.018	0.021	0.023	0.026	0.028	0.031	0.033	0.036	0.038	0.041	0.043	0.046	0.048	0.051	0.053	0.056	0.058	0.061	0.063	0.066	0.068	0.071	0.073	0.076	0.078	0.081	0.083	0.086	0.088	0.091	0.093	0.096	0.098
ErrorThreshold	0.075	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	0.1	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	0.125	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	0.15	0%	0%	1%	1%	1%	1%	1%	1%	1%	1%	1%	1%	1%	1%	1%	1%	1%	1%	1%	1%	1%	1%	1%	1%	1%	1%	1%	1%	1%	1%	1%	1%	1%	1%	1%	1%	1%	1%	1%	1%
	0.175	0%	1%	1%	2%	2%	2%	2%	2%	2%	3%	3%	3%	3%	3%	3%	3%	3%	3%	3%	3%	3%	3%	3%	3%	3%	3%	3%	3%	3%	3%	3%	3%	3%	3%	3%	3%	3%	3%	3%	3%
	0.2	0%	2%	3%	3%	4%	4%	5%	5%	5%	6%	6%	6%	6%	6%	6%	6%	6%	6%	6%	6%	7%	7%	7%	7%	7%	7%	7%	7%	7%	7%	7%	7%	7%	7%	7%	7%	7%	7%	7%	7%
	0.225	0%	3%	5%	6%	7%	8%	8%	9%	10%	10%	10%	10%	11%	11%	11%	11%	11%	11%	11%	12%	12%	12%	12%	12%	12%	12%	12%	12%	12%	12%	12%	12%	12%	12%	12%	12%	12%	12%	13%	13%
	0.25	0%	5%	8%	9%	11%	12%	13%	14%	15%	16%	16%	16%	17%	17%	17%	17%	18%	18%	18%	18%	18%	18%	19%	19%	19%	19%	19%	19%	20%	20%	20%	20%	20%	20%	20%	20%	20%	20%	20%	20%
	0.275	1%	6%	10%	12%	14%	16%	17%	19%	20%	21%	22%	22%	23%	23%	24%	24%	24%	24%	25%	25%	25%	25%	25%	26%	26%	26%	26%	26%	27%	27%	27%	27%	27%	27%	27%	27%	27%	27%	27%	27%
	0.3	1%	7%	12%	15%	17%	20%	22%	23%	25%	26%	27%	27%	28%	28%	29%	29%	30%	30%	30%	31%	31%	31%	32%	32%	32%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%
	0.325	1%	8%	13%	17%	20%	23%	25%	27%	29%	30%	31%	32%	33%	33%	34%	34%	35%	35%	36%	36%	36%	37%	37%	37%	38%	38%	39%	39%	39%	39%	39%	39%	40%	40%	40%	40%	40%	40%	40%	40%
	0.35	1%	9%	15%	19%	22%	26%	28%	30%	32%	34%	35%	36%	37%	38%	38%	39%	40%	40%	41%	41%	42%	42%	42%	43%	43%	44%	44%	45%	45%	45%	45%	45%	45%	45%	45%	45%	45%	45%	45%	45%
	0.375	1%	10%	16%	20%	24%	27%	30%	32%	35%	36%	37%	39%	40%	40%	41%	42%	43%	43%	44%	44%	45%	45%	46%	46%	47%	48%	48%	48%	49%	49%	49%	49%	49%	49%	49%	49%	49%	49%	49%	49%
	0.4	1%	10%	16%	21%	25%	29%	32%	34%	36%	38%	39%	41%	42%	43%	44%	45%	45%	46%	47%	47%	48%	48%	49%	49%	50%	51%	51%	52%	52%	52%	52%	52%	52%	52%	52%	52%	52%	52%	53%	53%
	0.425	1%	10%	17%	22%	26%	30%	33%	35%	38%	40%	41%	43%	44%	45%	46%	47%	48%	49%	49%	50%	51%	51%	52%	52%	53%	54%	54%	55%	55%	55%	55%	55%	56%	56%	56%	56%	56%	56%	56%	56%
	0.45	1%	10%	17%	22%	26%	31%	34%	37%	39%	41%	43%	44%	45%	46%	48%	49%	50%	50%	51%	52%	53%	53%	54%	54%	55%	56%	56%	57%	57%	58%	58%	58%	58%	58%	58%	58%	58%	58%	58%	58%
	0.475	1%	10%	17%	22%	27%	31%	35%	37%	40%	42%	44%	45%	46%	48%	49%	50%	51%	52%	53%	53%	54%	55%	56%	56%	57%	58%	58%	59%	59%	60%	60%	60%	60%	60%	60%	60%	61%	61%	61%	61%
	0.5	1%	10%	17%	23%	27%	32%	35%	38%	41%	43%	44%	46%	47%	48%	50%	51%	52%	53%	54%	55%	56%	56%	57%	58%	58%	59%	60%	61%	61%	61%	62%	62%	62%	62%	62%	62%	62%	62%	62%	62%
	0.525	1%	10%	17%	23%	28%	32%	36%	38%	41%	43%	45%	47%	48%	49%	51%	52%	53%	54%	55%	56%	57%	57%	58%	59%	60%	61%	62%	62%	63%	63%	63%	63%	64%	64%	64%	64%	64%	64%	64%	64%
	0.55	1%	10%	18%	23%	28%	33%	36%	39%	42%	44%	46%	48%	49%	50%	52%	53%	54%	55%	56%	57%	58%	59%	60%	60%	61%	62%	63%	64%	64%	65%	65%	65%	65%	66%	66%	66%	66%	66%	66%	66%
	0.575	1%	11%	18%	24%	28%	33%	37%	40%	43%	45%	47%	49%	50%	51%	53%	54%	56%	57%	58%	59%	60%	60%	61%	62%	63%	64%	65%	66%	66%	67%	67%	67%	67%	68%	68%	68%	68%	68%	68%	68%
	0.6	1%	11%	18%	24%	29%	34%	38%	41%	44%	46%	48%	50%	51%	53%	54%	56%	57%	58%	59%	60%	61%	62%	63%	64%	65%	66%	67%	68%	68%	69%	69%	69%	70%	70%	70%	70%	70%	70%	70%	70%
	0.625	1%	11%	18%	24%	29%	34%	38%	41%	44%	46%	48%	50%	52%	53%	55%	57%	58%	59%	60%	61%	62%	63%	64%	65%	66%	67%	68%	69%	70%	70%	70%	71%	71%	71%	71%	71%	72%	72%	72%	72%
	0.65	1%	11%	18%	24%	29%	35%	39%	42%	45%	47%	49%	51%	53%	54%	56%	58%	59%	60%	61%	63%	63%	64%	65%	66%	67%	68%	70%	71%	71%	72%	72%	72%	73%	73%	73%	73%	73%	73%	73%	74%
	0.675	1%	11%	18%	24%	30%	35%	39%	43%	46%	48%	50%	52%	54%	55%	57%	59%	60%	61%	63%	64%	65%	66%	67%	68%	69%	70%	71%	72%	73%	73%	74%	74%	74%	75%	75%	75%	75%	75%	75%	76%
	0.7	1%	11%	18%	25%	30%	36%	40%	43%	46%	49%	50%	53%	55%	56%	58%	60%	61%	63%	64%	65%	66%	67%	68%	69%	70%	71%	73%	74%	74%	75%	75%	76%	76%	76%	76%	76%	77%	77%	77%	77%
	0.725	1%	11%	18%	25%	31%	36%	41%	44%	47%	50%	51%	54%	56%	57%	59%	61%	63%	64%	65%	67%	68%	68%	70%	71%	71%	73%	74%	75%	76%	77%	77%	77%	78%	78%	78%	78%	78%	78%	79%	79%
	0.75	1%	11%	18%	25%	31%	37%	41%	44%	48%	50%	52%	55%	57%	58%	60%	62%	63%	65%	66%	67%	69%	69%	71%	72%	73%	74%	75%	77%	77%	78%	78%	79%	79%	79%	79%	80%	80%	80%	80%	80%
	0.775	1%	11%	18%	25%	31%	37%	42%	45%	48%	51%	53%	55%	57%	59%	61%	63%	64%	66%	67%	69%	70%	70%	72%	73%	74%	75%	77%	78%	79%	79%	80%	80%	80%	81%	81%	81%	81%	81%	81%	82%
	0.8	1%	11%	18%	25%	31%	38%	42%	46%	49%	52%	54%	56%	58%	60%	62%	64%	66%	67%	69%	70%	71%	72%	73%	74%	75%	77%	78%	79%	80%	81%	81%	82%	82%	82%	82%	82%	83%	83%	83%	83%
	0.825	1%	11%	19%	25%	32%	38%	43%	46%	50%	52%	54%	57%	59%	61%	63%	65%	66%	68%	69%	71%	72%	73%	74%	75%	76%	78%	79%	80%	81%	82%	82%	83%	83%	83%	84%	84%	84%	84%	84%	84%
	0.85	1%	11%	19%	26%	32%	38%	43%	47%	50%	53%	55%	58%	60%	62%	63%	66%	67%	69%	70%	71%	73%	73%	75%	76%	77%	79%	80%	81%	82%	83%	83%	84%	84%	84%	84%	84%	85%	85%	85%	85%
	0.875	1%	11%	19%	26%	32%	39%	44%	47%	51%	53%	56%	58%	60%	62%	64%	66%	68%	69%	71%	72%	73%	74%	76%	77%	78%	80%	81%	82%	83%	84%	84%	85%	85%	85%	85%	85%	86%	86%	86%	86%
	0.9	1%	11%	19%	26%	33%	39%	44%	48%	51%	54%	56%	59%	61%	63%	65%	67%	69%	70%	72%	73%	74%	75%	77%	78%	79%	81%	82%	83%	84%	85%	85%	86%	86%	86%	86%	87%	87%	87%	87%	87%
	0.925	1%	11%	19%	26%	33%	40%	45%	48%	52%	55%	57%	60%	62%	64%	66%	68%	70%	71%	73%	74%	75%	76%	78%	79%	80%	82%	83%	84%	85%	86%	86%	87%	87%	87%	88%	88%	88%	88%	88%	89%
	0.95	1%	11%	19%	26%	33%	40%	45%	49%	53%	55%	58%	60%	62%	64%	66%	69%	70%	72%	73%	75%	76%	77%	79%	80%	81%	82%	84%	85%	86%	87%	87%	88%	88%	88%	88%	88%	89%	89%	89%	89%
	0.975	1%	11%	19%	26%	33%	40%	46%	49%	53%	56%	58%	61%	63%	65%	67%	69%	71%	73%	74%	76%	77%	78%	79%	80%	81%	83%	85%	86%	87%	87%	88%	88%	89%	89%	89%	89%	89%	90%	90%	90%
	1	1%	11%	19%	27%	34%	41%	46%	50%	53%	56%	58%	61%	63%	65%	68%	70%	71%	73%	75%	76%	77%	78%	80%	81%	82%	84%	85%	87%	88%	88%	89%	89%	89%	90%	90%	90%	90%	90%	91%	91%
	1.025	1%	11%	19%	27%	34%	41%	46%	50%	53%	56%	58%	61%	64%	66%	68%	70%	72%	73%	75%	76%	77%	78%	80%	81%	82%	84%	86%	87%	88%	89%	89%	89%	90%	90%	90%	90%	91%	91%	91%	91%
	1.05	1%	11%	19%	27%	34%	41%	46%	50%	54%	56%	59%	61%	64%	66%	68%	70%	72%	73%	75%	77%	78%	79%	80%	82%	83%	85%	86%	87%	88%	89%	89%	90%	90%	91%	91%	91%	91%	91%	91%	92%

Figure D-3: sensitivity analysis change in FAR as θ_{sim} and θ_{error} are varied

Using the two sensitivity analysis matrices in Figures D-2 and Figure D-3, we run the MATLAB code,

```
function [ OptimalBins ] = findOptimalThresholds( ss1, ss2 )
%function findOptimalThresholds finds the SimilarityThreshold and the Error
%Threshold pair that yields the highest fraction of:
%
%     points on pipe spool of interest accepted through filter
%                           divided by
%           total points accepted into the hypothesis space
%
%                                    for every corresponding FAR, or filter
%                                    acceptance rate.
%
%   input:
%   [ss1] = sensitivity study results showing how the number of points on
%           pipe spool of interest accepted through filter are affected by
%           filter thresholds.
%   [ss2] = sensitivity study results showing how the filter acceptance
%           rate is affected by filter thresholds.
%

OptimalBins=zeros(9,3);
for i=1:9
    maxSS1=max(ss1(ss2>(i/10-0.01)&ss2<(i/10+0.01)));
    [rows,cols]=find(ss1== maxSS1&ss2>(i/10-0.01)&ss2<(i/10+0.01));
    OptimalBins(i,2)=rows(1,1);
    OptimalBins(i,3)=cols(1,1);
    OptimalBins(i,1)=ss2(OptimalBins(i,2),OptimalBins(i,3));
end

end
```

Which finds the parameters θ_{sim} and θ_{error} for FAR = {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9} that maximize metric 1. The resulting graph of optimal parameters can be found in Figure D-4.

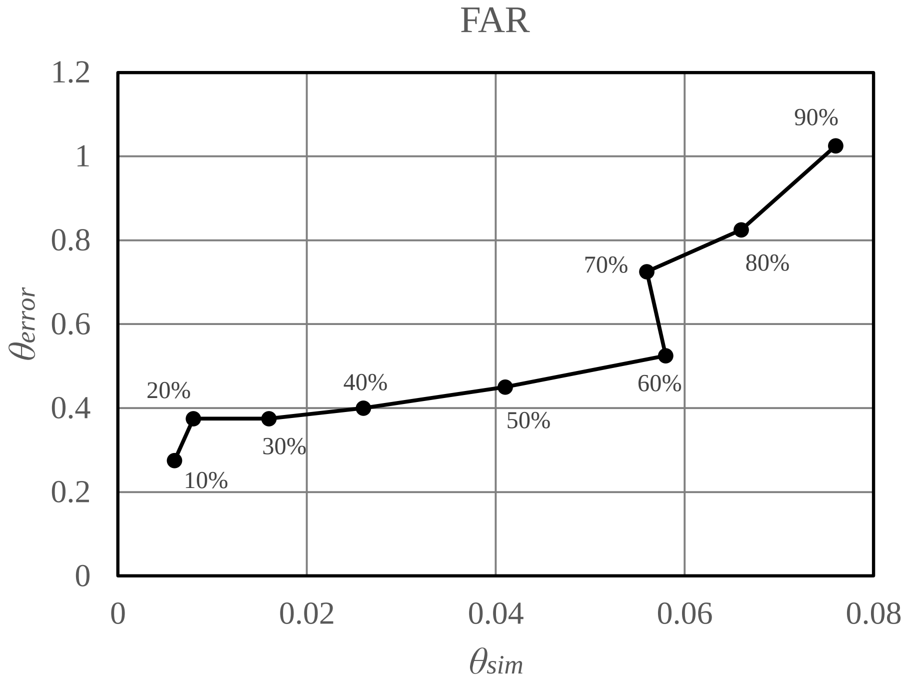


Figure D-4: optimal parameters $\boldsymbol{\theta_{sim}}$ and $\boldsymbol{\theta_{error}}$ for each FAR interval

Appendix E

MATLAB code: Determine nearest neighbourhood radius for curvature analysis

```
function [ scale ] = DetermineScale( Design_File, OCKDT_Design_File )
%
% function [ scale ] = DetermineScale( Design_File, OCKDT_Design_File )
% sets a reasonable scale value for nearest neighbourhood analysis of laser
% scanned data of pipe spools
%
%   input:
%   Design_File = 3D CAD design file point cloud of pipe spool (unit cm)
%   OCKDT_Design_File = KD-tree data structure for Design_File
%
pdr=0.5; %pdr is the Poisson-disc Sampling Radius used to sample the search space (UNITS)
Design_File_Sample_Size=10; %Specify the # of points to analyze in the Design_File
OCKDT_Design_File=KDTreeSearcher(Design_File);
scale=3*pdr; %smallest possible scale
Diameter=999; %initialize diamter
while scale<Diameter;

Design_File_Curvature_Signature=CurvSearch(Design_File,OCKDT_Design_File,pdr,Design_File_Sample_S
ize,scale);
    Diameter=2/median(Design_File_Curvature_Signature(:,4));
    scale=scale+pdr; %increase scale until median pipe diameter is reached
end
scale=Diameter/4+1; %set scale based on laser scan empirically derived lower bound estimate
end
```

Appendix F

MATLAB code: Curvature characterization algorithm

```
function [down]=CurvSearch(origCloud,OCKDT,pdr,sampleSize,scale)
%CurvSearch is a function that analyzes local surface patches existing in a point
%cloud [origCloud] and outputs the principal curvatures of each local surface patch.
%
%   input:
%   [origCloud]  = is the original Point Cloud matrix
%   [OCKDT]      = is the KD tree data structure for the input [origCloud]
%   [pdr]        = is the poisson-disc radius used to downsample the input
%                   [origCloud]
%   [sampleSize] = is the desired output point cloud size eg. 100 points or the
%                  number of local surface patches to be analysed
%   [scale]      = size of nearest neighbourhood defining the size of the local
%                  surface patches
%
%The curvature characterization algorithm is a repeating 3 step process
%   (1) nearest neighbourhood identificaiton and normal vector estimation
%   (2) curvature characterization
%       (a) project points within threshold distance of plane onto the
%           plane
%       (b) fit circle to the project points
%       (c) rotate plane
%   (3) identify highest and lowest curvatures as principal curvatures
%
down=zeros(sampleSize,7);
rot=8; %how many planes are used to find principal curvature values/directions
i=0;
while i<sampleSize
    k1=0;
    k2=0;
    sample=datasample(origCloud,1,'replace',false); %randomly chooses a single point from the
point cloud
```

```
    %(1) nearest neighbourhood identificaiton and normal vector estimation
    neN=rangesearch(OCKDT,sample,scale); %find points in origCloud within radius 'scale' of sample
    neN=cell2mat(neN);
    neN=origCloud(neN,:);
    if size(neN,1)<3
        continue; %if the nearest neighbourhood has less than 3 points, select new sample
    end
    normal=pca(neN); %find the principal componenets of the nearest neighbourhood
    %normal(:,3) = should be the ambiguous direction normal vector for sample
    if size(normal,2)<3
        normal(:,3)=[0;0;1];
    end
    n=normal(:,3)';
    planeN=normal(:,2)'; %set as normal vector for first plane
    curvature=zeros(2,rot);
    %uniformly rotate plane about normal vector, project points onto plane
    %and fit a circle
    %(2) curvature characterization
    for e=1:rot
        %(a) project points within threshold distance of plane onto the
        %    plane
        P(1)=-planeN(1)/planeN(3);
        P(2)=-planeN(2)/planeN(3);
        P(3)=planeN(1)*sample(1)/planeN(3)+planeN(2)*sample(2)/planeN(3)+sample(3);
        d=(neN(:,3)-P(1).*neN(:,1)-P(2)*neN(:,2)-P(3))./(P(1).*planeN(1)+ P(2).*planeN(2)-planeN(3));
        curveSet=[neN(:,1)+planeN(1)*d,neN(:,2)+planeN(2)*d,neN(:,3)+planeN(3)*d];
        %find index for points within threshold distance pdr*1.5 of plane
        dc=abs(d)<pdr*1.5;
        curveSet=curveSet(dc,:); %set of points in neN which are pdr*1.5 from the current plane, projected onto the plane
        %rotate the plane of points so that they exist in 2D
        v=cross(planeN,[0,0,1]);
        s=norm(v);
        c=dot(planeN,[0,0,1]);
        vx=[0,-v(3),v(2);v(3),0,-v(1);-v(2),v(1),0];
        R=eye(3)+vx+vx^2*((1-c)/s^2);
        RcurveSet=R*curveSet';
        RcurveSet=RcurveSet';
        RcurveSet=RcurveSet(:,1:2);
        rcsLength=size(RcurveSet);
        if rcsLength(:,1)<3
            curvature(e)=0;
            continue
        end
        %fit a circle to the data
        %(b) fit circle to the project points
        %Al-Sharadqah, A., & Chernov, N. (2009). Error analysis for circle fitting algorithms.
Electronic Journal of Statistics, 3, 886-911.
        %http://people.cas.uab.edu/~mosya/cl/MATLABcircle.html
        CPar=HyperSVD(RcurveSet);
        %Store circle fit error information
        Error=abs(sqrt((RcurveSet(:,1)-CPar(1)).^2+(RcurveSet(:,2)-CPar(2)).^2)-CPar(3));
        curvature(2,e)=rms(Error);
        %extract curvature information from fitted circle
        curvature(1,e)=1/CPar(3);
        %(c) rotate plane
        %rotate plane http://www.mathworks.com/matlabcentral/fileexchange/34426-rotate-vector-s--
about-axis
        crosskv(1) = n(2)*planeN(3) - n(3)*planeN(2);
        crosskv(2) = n(3)*planeN(1) - n(1)*planeN(3);
        crosskv(3) = n(1)*planeN(2) - n(2)*planeN(1);
        planeN = cos(pi()/rot)*planeN + (crosskv)*sin(pi()/rot)+ n*(dot(n,planeN))*(1 -
cos(pi()/rot));
    end
    %(3) identify highest and lowest curvatures as principal curvatures
    %extract principal curvatures
    [k1,in]=max(curvature(1,:));
    if in<=rot/2
        k2=curvature(1,in+rot/2);
        k2error=curvature(2,in+rot/2);
    else
        k2=curvature(1,in-rot/2);
        k2error=curvature(2,in-rot/2);
    end
    %store principal curvatures and associated circle fit errors
    i=i+1;
    down(i,4)=k1;
    down(i,5)=k2;
    down(i,6)=curvature(2,in);
    down(i,7)=k2error;
    down(i,1:3)=sample;
end
```

Appendix G

MATLAB code: Curvature-based similarity filter

```
function [ hypothesis ] = DesiredCurvNN( sample,Design_File_Curvature_Signature, tt)
%DesiredCurv classifies and filters points in [sample] based on the curvature
%similarity threshold. Points with mean 3NN distance below threshold are outputted
%
%   input:
%   [sample] = is the sample of the point cloud for which curvatures have
%              been calculated
%   [Design_File_Curvature_Signature] = is the dataset that the sample will
%                                       be compared to using 3NN
%   [tt] = is the 3NN distance similarity threshold
%
threshold=1000/length(Design_File_Curvature_Signature)*tt;
[IDX,D]=knnsearch(Design_File_Curvature_Signature(:,4:5),sample(:,4:5),'K',3);
hypothesis=sample(mean(D,2)<threshold,:); %returns a n x 5 matrix, first three columns xyz and
4th and 5th columns are principal curvatures
end
```

Appendix H

MATLAB code: DBSCAN clustering parameter determination

```
function [ Set_pts_final, Set_eps_final ] = determineClustParam( Design_File,ScanSize,Scan_Sample_Size, filterRate)
%function [determineClustParam] determines a set of DBSCAN parameters that
%are likely to produce a good hypothesis cluster in the hypothesis space
%
%   input:
%   [Design_File] = 3D CAD design file point cloud of pipe spool of interest
%   [ScanSize] = number of points in cluttered input point cloud scan
%   [Scan_Sample_Size] = number of points sampled from cluttered input
%                        point cloud scan for which curvature
%                        characterization will take place
%   [filterRate] = predicted number of points on object of interest that
%                  will be accepted by the filter
%
% The DBSCAN parameters are set by simulating the correct hypothesis object
% in the hypothesis space using [Design_File]. This can be done because the
% poisson-disc sampling preprocessing step ensures that the density of both
% the input point cloud and the cluttered input point cloud scan are
% approximately the same.

CADSize=length(Design_File);
percObj=CADSize/ScanSize; %fraction of cluttered input point cloud scan that is the pipe spool of interest
sampleObj=Scan_Sample_Size*percObj; %number of points that will be sampled from the pipe spool of interest in the scan
objClust=sampleObj*filterRate; %number of points that will comprise the correct hypothesis object in hypothesis space
Set_eps_collect=zeros(5,1);
for i=1:5
    Clust1=datasample(Design_File,floor(objClust));
    eps=5;
    a=0;
    mEps=0;
    %find Eps such that clustering the simulated hypothesis yields a single
    %cluster. Continue until at least two instances of Eps yield a single
    %cluster.
    while a<2
        eps=eps+3;
        [cl,ty]=dbscan(Clust1,1,eps); %minPts is set to 1
        if length(unique(cl))==1
            a=a+1;
            mEps=mEps+eps;
        end
    end
    mEps=mEps/2;
    Set_eps=mEps*1.2+10; %1.2 and 10 account for small occlusions and non-uniform point density
in the real laser scanned hypothesis space
    Set_eps_collect(i)=Set_eps;
    Set_pts_collect=zeros(10,1);
    %calculate minPts that is associated with the Eps calculated above
    %this is done by randomly sampling 10 different points from
    %[Design_File] and calculating the local point density in those regions
    %and selecting the median
    for p=1:10
        point=datasample(Clust1,1);
        idx=rangesearch(Clust1,point,Set_eps_collect(i));
        NN=cell2mat(idx);
        NN=Clust1(NN,:);
        Set_pts_collect(p,1)=length(NN);
    end
    Set_pts(i)=median(Set_pts_collect);
end
Set_pts_final=median(Set_pts);
Set_eps_final=median(Set_eps_collect);
end
```

Appendix I

MATLAB code: Hypothesis space clustering

```
function [ hypotheses ] = GenerateHypotheses( acceptedPoints,pts,eps )
%GenerateHypotheses clusters the search space comprised of points that were
%classified as having curvatures similar to the design file.
%
%   input:
%   [acceptedPoints] =  the points from the search space that have met the
%                       curvature similarity and circle fit criteria
%   [pts]            =  MinPts
%   [eps]            =  Eps
% minPts and eps as per -  Ester, M., Kriegel, H. P., Sander, J., & Xu, X.
%(1996, August). A density-based algorithm for discovering clusters in
%large spatial databases with noise. In Kdd (Vol. 96, No. 34, pp. 226-231).

[Class,Type]=dbscan(acceptedPoints(:,1:3),pts,eps); %cluster based on density
NumClusters=length(unique(Class))-1 % determines the number of clusters found (-1 to take out the
outlier class)
hypotheses=cell(NumClusters,1);
for i=1:NumClusters
    %output all clusters as hpothesis objects
    hypotheses{i}=acceptedPoints(Class==i,:);
end
end
```

Appendix J

MATLAB code: BoF-based matching

```matlab
function [ Similarity,DiffM ] = ComparePlots( Plot1,Plot2 )
%function ComparePlots provides a metric of similarity for two scatter plots by
%generating a bof i.e., a bivariate histogram and comparing bin frequencies
%
%       input:
%       [Plot1] = is the first scatter plot
%       [Plot2] = is the second scatter plot
%

%%
%ensures the two scatter plots are converted to histograms with identical bins and boundaries
min1x=min(Plot1(:,4));
max1x=max(Plot1(:,4));
min1y=min(Plot1(:,5));
max1y=max(Plot1(:,5));
min2x=min(Plot2(:,4));
max2x=max(Plot2(:,4));
min2y=min(Plot2(:,5));
max2y=max(Plot2(:,5));

Plot1(length(Plot1)+1,4:5)=[min2x,min2y];
Plot1(length(Plot1)+1,4:5)=[max2x,max2y];
Plot2(length(Plot2)+1,4:5)=[min1x,min1y];
Plot2(length(Plot2)+1,4:5)=[max1x,max1y];

%%
%convert plots to histograms and normalize

X1=[Plot1(:,4),Plot1(:,5)];
X2=[Plot2(:,4),Plot2(:,5)];
N1=hist3(X1,[20,40]);
N2=hist3(X2,[20,40]);
```

```
N1=N1/length(Plot1);

N2=N2/length(Plot2);


%%

%calculate the difference between the numbers in each of the histogram's

%bins


DiffM=abs(N1-N2);

Similarity=floor((1/sum(DiffM(:)))*100);


end
```

Appendix K

MATLAB code: Register 3D CAD design file to pipe spool of interest in cluttered scan

```
function
[PCmoving_Registered,tformfinal]=autoReg(PCmoving_downsampled,PCfixed_downsampled,PCmoving_original)
%function autoReg is used to register (i.e., superimpose) Point
%Cloud PCmoving_downsampled to Point Cloud PCfixed_downsampled
%
%   input:
%   [PCmoving_downsampled] = downsampled point cloud that will move during
%                            registration
%   [PCfixed_downsampled]  = downsampled point cloud that will be fixed in
%                            place during registration
%   [PCmoving_original]    = Original point cloud that will be registered
%                            to the fixed point cloud
%

%Move PCmoving such that the centroid of the two point clouds are the same
[PCmoving_downsampled_Normalized,PCfixed_downsampled_Normalized,Centroid] =
alignCentroids(PCmoving_downsampled,PCfixed_downsampled);

%aligns the principal components of the two input point clouds ambiguously,
%and outputs both the resulting movedPointCloud and the rotation cell array
%that defines the 4 possible associated configurations.
[PCmoving_downsampled_Course_Transformed,R]=transpca(PCmoving_downsampled_Normalized,PCfixed_downsampled_Normalized);
minerror=10^5; %initialize fit error with high number
PointCloudfixed=pointCloud(PCfixed_downsampled_Normalized);

% try each of the 4 possible principal component alignments and select the
% option with the lowest RMSE
for i=1:4
    PointCloudmoving=pointCloud(PCmoving_downsampled_Course_Transformed*R{i+2});
    PCmovingCourse=PCmoving_downsampled_Course_Transformed*R{i+2};
    [tform, PointCloudMovingReg,
rmse]=pcregrigid(PointCloudmoving,PointCloudfixed,'Extrapolate',true);   %ICP performs on each
combination
    PCmovingFine=PCmovingCourse*tform.T(1:3,1:3);
    if rmse<minerror
        minerror=rmse;
        tform.T(1:3,1:3)=R{1}*R{2}*R{i+2}*tform.T(1:3,1:3);
        tform.T(4,1:3)=tform.T(4,1:3)+Centroid;
        tformfinal=tform.T; %store transformation
        PCmoving_original_mean=mean(PCmoving_original);
        %register original point cloud using calculated transform
        PCmoving_original_normalized(:,1)=PCmoving_original(:,1)-PCmoving_original_mean(:,1);
        PCmoving_original_normalized(:,2)=PCmoving_original(:,2)-PCmoving_original_mean(:,2);
        PCmoving_original_normalized(:,3)=PCmoving_original(:,3)-PCmoving_original_mean(:,3);
        PCmoving_Registered=PCmoving_original_normalized*tform.T(1:3,1:3);
        PCmoving_Registered(:,1)=PCmoving_Registered(:,1)+tform.T(4,1);
        PCmoving_Registered(:,2)=PCmoving_Registered(:,2)+tform.T(4,2);
        PCmoving_Registered(:,3)=PCmoving_Registered(:,3)+tform.T(4,3);
    end
end

function [ PCmoving_Course,PCfixed_Course, PCfixed_Centroid ] = alignCentroids( PCmoving,
PCfixed)
%Moves PCmoving such that the centroid of the two point clouds are the same
PCmoving_Centroid=mean(PCmoving);
PCfixed_Centroid=mean(PCfixed);

PCmoving_Course(:,1)=PCmoving(:,1)-PCmoving_Centroid(1);
PCmoving_Course(:,2)=PCmoving(:,2)-PCmoving_Centroid(2);
PCmoving_Course(:,3)=PCmoving(:,3)-PCmoving_Centroid(3);

PCfixed_Course(:,1)=PCfixed(:,1)-PCfixed_Centroid(1);
PCfixed_Course(:,2)=PCfixed(:,2)-PCfixed_Centroid(2);
PCfixed_Course(:,3)=PCfixed(:,3)-PCfixed_Centroid(3);
End
```

```
function [ PCmoving_Rotation2,R ] = transpca( PCmoving_Normalized,PCfixed_Normalized )
%transpca( PCmoving_Course_Translation,PCfixed ) aligns the principal
%components of the two input point clouds ambiguously, and outputs both the resulting
%movedPointCloud and the rotation cell array that defines the 4 possible
%associated configurations.

PCmovingPC=pca(PCmoving_Normalized);
PCfixedPC=pca(PCfixed_Normalized);

R=cell(4);
R{1}=vrrotvec2mat(vrrotvec(PCfixedPC(:,1),PCmovingPC(:,1)));
PCmoving_Rotation1=PCmoving_Normalized*R{1};

PCmovingPC2=pca(PCmoving_Rotation1);

R{2}=vrrotvec2mat(vrrotvec(PCfixedPC(:,2),PCmovingPC2(:,2)));
PCmoving_Rotation2=PCmoving_Rotation1*R{2};



R{3}=[1,0,0;0,1,0;0,0,1];
R{4}=vrrotvec2mat([PCfixedPC(:,1);pi]);
R{5}=vrrotvec2mat([PCfixedPC(:,2);pi]);
R{6}=vrrotvec2mat([PCfixedPC(:,3);pi]);

end
```

Bibliography

Aggarwal, C. C., and Reddy, C. K. (2014). *Data clustering : algorithms and applications*. Boca Raton, FL, CRS Press.

Ahmed, M., Haas, C. T., Haas, R. (2013). Autonomous Modeling of Pipes within Point Clouds. *Proceedings of the 30th ISARC*, Montréal, Canada, 1093-1100.

Ahmed, M., Haas, C., Haas, R. (2014). Automatic Detection of Cylindrical Objects in Built Facilities. *Journal of Computing in Civil Engineering*, 28(3), 04014009.

Akinci, B., Boukamp, F., Gordon, C., Huber, D., Lyons, C., Park, K. (2006). A Formalism for Utilization of Sensor Systems and Integrated Project Models for Active Construction Quality Control. *Automation in Construction*, 15(2), 124-138.

Al-Sharadqah, A., and Chernov, N. (2009). Error Analysis for Circle Fitting Algorithms. Electronic *Journal of Statistics*, 3, 886-911.

Amann, M., Bosch, T., Lescure, M., Myllyla¨, R., Rioux, M. (2001). Laser Ranging: A Critical Review of Usual Techniques for Distance Measurement. *Optical Engineering*, 40(1), 10.

Antaki, G. A. (2005). *Fitness-for-Service and integrity of piping, Vessels, and tanks: ASME Code Simplifed*. New York, NY, McGraw-Hill.

As'ari, M. A., Sheikh, U. U., Supriyanto, E. (2014). 3D Shape Descriptor for Object Recognition Based on Kinect-Like Depth Image. *Image and Vision Computing*, 32(4), 260-269.

Ballard, D. H. (1981). Generalizing the Hough Transform to Detect Arbitrary Shapes. *Pattern Recognition*, 13(2), 111-122.

Ballast, D. K. (2007). *Handbook of Construction Tolerances*. 2nd Ed., Hoboken, N.J, Wiley.

Balslev, I., & Eriksen, R. D. (2010). U.S. Patent No. 7,822,264. Washington, DC: U.S. Patent and Trademark Office.

Bentley, J. L. (1975). Multidimensional Binary Search Trees used for Associative Searching. *Communications of the ACM*, 18(9), 509-517.

Besl, P. J., and McKay, H. D. (1992). A Method for Registration of 3-D Shapes. *Pattern Analysis and Machine Intelligence*, IEEE Transactions On, 14(2), 239-256.

Blais, F. (2004). Review of 20 Years of Range Sensor Development. *Journal of Electronic Imaging*, 13(1), 231.

Blei, D. M., Ng, A. Y., Jordan, M. I. (2003). Latent Dirichlet Allocation. *The Journal of Machine Learning Research*, 3, 993-1022.

Borrmann, D., Elseberg, J., Lingemann, K., Nüchter, A. (2011). The 3D Hough Transform for Plane Detection in Point Clouds: A Review and a New Accumulator Design. *3D Research*, 2(2), 1-13.

Bosché, F., Ahmed, M., Turkan, Y., Haas, C. T., Haas, R. (2015). The Value of Integrating Scan-to-BIM and Scan-Vs-BIM Techniques for Construction Monitoring using Laser Scanning and BIM: The Case of Cylindrical MEP Components. *Automation in Construction*, 49, 201-213.

Bosché, F., Guillemet, A., Turkan, Y., Haas, C., Haas, R. (2014). Tracking the Built Status of MEP Works: Assessing the Value of a Scan-Vs-BIM System. *Journal of Computing in Civil Engineering*, 28(4), 05014004.

Bosché, F. (2010). Automated Recognition of 3D CAD Model Objects in Laser Scans and Calculation of as-Built Dimensions for Dimensional Compliance Control in Construction. *Advanced Engineering Informatics*, 24(1), 107-118.

Bosché, F., Haas, C., Akinci, B. (2009). Automated Recognition of 3D CAD Objects in Site Laser Scans for Project 3D Status Visualization and Performance Control. *Journal of Computing in Civil Engineering*, 23(6), 311-318.

Bosché, F., and Haas, C. (2008). Automated Retrieval of 3D CAD Model Objects in Construction Range Images. *Automation in Construction*, 17(4), 499-512.

Bosché, F. (2012). Plane-Based Registration of Construction Laser Scans with 3D/4D Building Models. *Advanced Engineering Informatics*, 26(1), 90-102.

Bronstein, A. M., Bronstein, M. M., Guibas, L. J., Ovsjanikov, M. (2011). Shape Google: Geometric Words and Expressions for Invariant Shape Retrieval. *ACM Transactions on Graphics* , 30(1), 1.

Brown, L. (1992). A Survey of Image Registration Techniques. *ACM computing surveys (CSUR)*, 24(4), 325.

Carmo, M. (1976). *Differential geometry of curves and surfaces*. Englewood Cliffs, N.J, Prentice-Hall.

Chandler, G. (2013). Smaller, Better Faster: Fred Haney's Vision Turns Modular Construction Theory on its Head. *Oilsands Review*, 16.

Charron, R., Harrington, H. J., Voehl, F., Wiggin, H. (2015). *The Lean Management Systems Handbook*, CRC Press, Taylor & Francis Group, Boca Raton, FL.

Chua, C. S., and Jarvis, R. (1997). Point Signatures: A New Representation for 3d Object Recognition. *International Journal of Computer Vision*, 25(1), 63-85.

Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., Ranzuglia, G. (2014). MeshLab: An open-source mesh processing tool. <http://meshlab.sourceforge.net/> 2014).

CII chartered Research Team 327. (2015). Innovative Delivery of Information to the Crafts. Construction Industry Institute Implementation Resource, 327-1.

CII chartered Research Team 252. (2011). A Guide to Construction Rework Reduction. Construction Industry Institute Implementation Resource, 252-2b.

Cover, T. M., and Hart, P. E. (1967). Nearest Neighbor Pattern Classification. *Information Theory*, IEEE Transactions On, 13(1), 21-27.

Date, H., Kaneta, Y., Hatsukaiwa, A., Onosato, M., Kanai, S. (2012). Object Recognition in Terrestrial Laser Scan Data using Spin Images. *Computer-Aided Design and Applications*, 9(2), 187-197.

De Silva, C. W. (2009). *Modeling and control of engineering systems*, Boca Raton, FL, CRC Press.

Dimitrov, A., and Golparvar-Fard, M. (2015). Segmentation of Building Point Cloud Models Including Detailed Architectural/Structural Features and MEP Systems. *Automation in Construction*, 51, 32-45.

Dissanayake, M., Fayek, A., Wolf, H., Tol, A. (2003). Developing a Standard Approach for Construction Field Rework Assessment. *In Construction Research Congress@ sWind of Change: Integration and Innovation* (pp. 1-8). ASCE.

Drouin, M., and Beraldin, J. (2012). Active 3D Imaging Systems. In Pears, N., Liu, Y., Bunting, P., (Eds.) *3D Imaging, Analysis and Applications*. New York, NY, Springer.

Erdos, G., Nakano, T., Horváth, G., Nonaka, Y., Váncza, J. (2015). Recognition of Complex Engineering Objects from Large-Scale Point Clouds. *CIRP Annals-Manufacturing Technology*, 64(1), 165-168.

Ester, M., Kriegel, H., Sander, J., Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Kdd*, 96(34) , 226-231.

FARO. (2016). As built documentation. <http://www.faro.com/measurement-solutions/applications/as-built-documentation> .

FARO. (2015). Features of the focus 3D laser scanner. <http://www.faro.com/products/3d-surveying/laser-scanner-faro-focus-3d/features#main> 2015).

Fehr, J., Streicher, A., Burkhardt, H. (2009). A Bag of Features Approach for 3D Shape Retrieval. *Advances in Visual Computing*, Springer, 34-43.

Fiatech. (2015). Tech roadmap: Construction summary. <http://fiatech.org/tech-roadmap/strategic-elements/construction> .

Fischler, M. A., and Bolles, R. C. (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications with ACM*, 24(6), 381-395.

Friedman, J. H., Bentley, J. L., Finkel, R. A. (1977). An Algorithm for Finding Best Matches in Logarithmic Expected Time. *ACM Transactions on Mathematical Software (TOMS)*, 3(3), 209-226.

Gal, R., and Cohen-Or, D. (2006). Salient Geometric Features for Partial Shape Matching and Similarity. *ACM Transactions on Graphics (TOG)*, 25(1), 130-150.

Gatzke, T., Grimm, C., Garland, M., Zelinka, S. (2005). Curvature maps for local shape comparison. In *Shape Modeling and Applications*, 2005 International Conference, IEEE, 244-253.

Ghosh, S., and Lohani, B. (2013). Mining Lidar Data with Spatial Clustering Algorithms. *International Journal of Remote Sensing*, 34(14), 5119-5135.

Gilsinn, D., Cheok, G. S., Witzgall, C., Lytle, A. (2005). *Construction Object Identification from LADAR Scans: An Experimental Study using I-Beams*, US Department of Commerce, Technology Administration, National Institute of Standards and Technology, .

Golparvar-Fard, M., Peña-Mora, F., Savarese, S. (2011). Integrated Sequential as-Built and as-Planned Representation with D4AR Tools in Support of Decision-Making Tasks in the AEC/FM Industry. *Journal of Construction Engineering and Management*, 137(12), 1099-1116.

Gomes, L., Regina Pereira Bellon, O., Silva, L. (2014). 3D Reconstruction Methods for Digital Preservation of Cultural Heritage: A Survey. *Pattern Recognition Letters*, 50, 3-14.

Gonzalez-Jorge, H., Rodríguez-Gonzálvez, P., Martínez-Sánchez, J., González-Aguilera, D., Arias, P., Gesto, M., Díaz-Vilariño, L. (2015). Metrological Comparison between Kinect I and Kinect II Sensors. *Measurement*, 70, 21-26.

Gordon, C., Boukamp, F., Huber, D., Latimer, E., Park, K., Akinci, B. (2003). Combining reality capture technologies for construction defect detection: A case study. In *EIA9: E-Activities and Intelligent Support in Design and the Built Environment*, 9th EuropIA International Conference, 99-108.

Grimson, W. E. L., Lozano-Perez, T., Noble, N., White, S. J. (1993). An automatic tube inspection system that finds cylinders in range data. In *Computer Vision and Pattern Recognition*, Proceedings CVPR'93., 1993 IEEE Computer Society Conference on, 446-452.

Guo, Y., Bennamoun, M., Sohel, F., Lu, M., Wan, J. (2014). 3D Object Recognition in Cluttered Scenes with Local Surface Features: A Survey. *Pattern Analysis and Machine Intelligence*, IEEE Transactions on, 36(11), 2270-2287.

Halpin, D. W., and Senior, B. A. (2011). *Construction Management*. 4th Ed., Hoboken, NJ, Wiley.

Han, S. H., Kim, C., Kim, H., Son, H. (2008). Applicability of Flash Laser Distance and Ranging to Three-Dimensional Spatial Information Acquisition and Modeling on a Construction Site. *Canadian Journal of Civil Engineering*, 35, 1331.

Hegazy, T., Said, M., Kassab, M. (2011). Incorporating Rework into Construction Schedule Analysis. *Automation in Construction*, 20(8), 1051-1059.

Heider, P., Pierre-Pierre, A., Li, R., Mueller, R., Grimm, C. (2012). Comparing Local Shape Descriptors. *The Visual Computer*, 28(9), 919-929.

Hoiem, D., and Savarese, S. (2011). *Representations and Techniques for 3D Object Recognition and Scene Interpretation*. San Rafael, CA, Morgan & Claypool.

Horn, D. (1989). Machine Vision: The Guiding Light. *Mechanical Engineering*, 111(6), 40.

Hwang, B., Thomas, S., Haas, C., Caldas, C. (2009). Measuring the Impact of Rework on Construction Cost Performance." *Journal of Construction Engineering and Management*, 135(3), 187-198.

International Bureau of Weights and Measures. (1993). *International Vocabulary of Basic and General Terms in Metrology*. 2nd Ed., International Organization for Standardization, Genève, Switzerland.

ISO 4463-1:1989. (2012). *Measurement Methods for Buildings: Setting-Out and Measurement*. International Organization for Standardization, Geneva, Switzerland.

Iyer, N., Jayanti, S., Lou, K., Kalyanaraman, Y., Ramani, K. (2005). Three-Dimensional Shape Searching: State-of-the-Art Review and Future Trends. *Computer-Aided Design*, 37(5), 509-530.

Jacobs, G. (2006). Uses of Scanning in Construction and Fabrication. *Professional Surveyor Magazine*, <http://hds.leica-geosystems.com/en/Scanning-in-Construction_54173.htm>

Jain, A. K., Duin, R. P., Mao, J. (2000). Statistical Pattern Recognition: A Review. *Pattern Analysis and Machine Intelligence*, IEEE Transactions On, 22(1), 4-37.

Johnson, A. E., and Hebert, M. (1999). Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes. *Pattern Analysis and Machine Intelligence*, IEEE Transactions On, 21(5), 433-449.

Kazmi, I. K., You, L., Zhang, J. J. (2013). A survey of 2D and 3D shape descriptors. In *Computer Graphics, Imaging and Visualization (CGIV)*, 2013 10th International Conference, IEEE, 1-10.

Khoshelham, K., and Elberink, S. O. (2012). Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications. *Sensors*, 12(2).

Kim, M., Cheng, J. C. P., Sohn, H., Chang, C. (2015). A Framework for Dimensional and Surface Quality Assessment of Precast Concrete Elements using BIM and 3D Laser Scanning. *Automation in Construction*, 49(Part B), 225-238.

Klasing, K., Althoff, D., Wollherr, D., Buss, M. (2009). Comparison of surface normal estimation methods for range sensing applications. In *Robotics and Automation*, 2009. ICRA'09. IEEE International Conference on, 3206-3211.

Koenderink, J. J., and van Doorn, A. J. (1992). Surface Shape and Curvature Scales. *Image and Vision Computing*, 10(8), 557-564.

Körtgen, M., Park, G., Novotni, M., Klein, R. (2003). 3D shape matching with 3D shape contexts. In *The 7th central European seminar on computer graphics*, 3, 5-17.

Li, X., and Godil, A. (2009). Exploring the bag-of-words method for 3D shape retrieval. In *Image Processing (ICIP)*, 2009 16th IEEE International Conference On, IEEE, 437-440.

Li, B., Godil, A., Johan, H. (2014). Hybrid Shape Descriptor and Meta Similarity Generation for Non-Rigid and Partial 3D Model Retrieval. *Multimedia Tools and Applications*, 72(2), 1531-1560.

Li, C., and Ben Hamza, A. (2014). Spatially Aggregating Spectral Descriptors for Nonrigid 3D Shape Retrieval: A Comparative Survey. *Multimedia Systems*, 20(3), 253-281.

Liu, Y., Zhang, J., Hou, J., Ren, J., Tang, W. (2013). Cylinder Detection in Large-Scale Point Cloud of Pipeline Plant. In *Visualization and Computer Graphics*, IEEE Transactions On, 19(10), 1700-1707.

Liu, Y., and Xiong, Y. (2008). Automatic Segmentation of Unorganized Noisy Point Clouds Based on the Gaussian Map. *Computer-Aided Design*, 40(5), 576-594.

Love, P. (2002). Influence of Project Type and Procurement Method on Rework Costs in Building Construction Projects. *Journal of Construction Engineering and Management*, 128(1), 18-29.

Love, P. E. D., and Li, H. (2000). Quantifying the Causes and Costs of Rework in Construction. *Construction Management & Economics*, 18(4), 479-490.

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2), 91-110.

Lucas, W., and Welding Institute. (1991). *Process pipe and tube welding : a guide to welding process options, Techniques, Equipment, NDT and codes of Practice*. Abington Publishing, Woodhead Publishing in association with the Welding Institute, Cambridge, England.

MathWorks. (2016). Using RANSAC for estimating geometric transforms in computer vision. <http://www.mathworks.com/discovery/ransac.html> .

Microsoft. (2015). Kinect for windows v2 sensor. <http://www.microsoftstore.com/store/msca/en_CA/pdp/Kinect-for-Windows-v2-Sensor/productID.298907900> .

Nagase, M., Akizuki, S., Hashimoto, M. (2013). 3-D feature point matching for object recognition based on estimation of local shape distinctiveness. In *Computer Analysis of Images and Patterns*, Berlin Heidelberg, Springer, 473-481.

Nahangi, M., Czerniawski, T., Haas, T. C., Walbridge, S., West, J. (2015). Parallel Systems and Structural Frame Realignment Planning and Actuation Strategy. *Journal of Computing in Civil Engineering*, 10.1061/(ASCE)CP.1943-5487.0000545, 04015067.

Nahangi, M., and Haas, C. T. (2014). Automated 3D Compliance Checking in Pipe Spool Fabrication. *Advanced Engineering Informatics*, 28(4), 360.

Nahangi, M., Yeung, J., Haas, C. T., Walbridge, S., West, J. (2014). Automatic Realignment of Defective Assemblies using an Inverse Kinematics Analogy. *Journal of Computing in Civil Engineering*, 10.1061/(ASCE)CP.1943-5487.0000477, 04015008.

Nayyar, M. L. (2000). *Piping Handbook*. 7th Ed., New York, NY, McGraw-Hill Education.

Nelson, D. (1975). *Managers and workers: Origins of the new factory system in the United States, 1880-1920*. Wisconsin, University of Wisconsin Press Madison.

Newcomb, T. M., and Attneave, F. (1954). Some Informational Aspects of Visual Perception. *Psychological Review*, 61(3), 183-193.

Newman, T. S., and Jain, A. K. (1995a). A Survey of Automated Visual Inspection. *Computer Vision and Image Understanding*, 61(2), 231-262.

Newman, T. S., and Jain, A. K. (1995b). A System for 3D CAD-Based Inspection using Range Images. *Pattern Recognition*, 28(10), 1555-1574.

NRC. (2009). *Advancing the Competitiveness and Efficiency of the U.S. Construction Industry*, Washington, D.C. < http://www.nap.edu/catalog/12717/advancing-the-competitiveness-and-efficiency-of-the-us-construction-industry>

O'Connor, J., and Tucker, R. (1986). Industrial Project Constructability Improvement. *Journal of Construction Engineering and Management*, 112(1), 69-82.

Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D. (2002). Shape Distributions. *ACM Transactions on Graphics (TOG)*, 21(4), 807-832.

Pang, G., and Neumann, U. (2013). Training-based object recognition in cluttered 3d point clouds. In *3D Vision*, 2013 International Conference On, IEEE, 87-94.

Patraucean, V., Armeni, I., Nahangi, M., Yeung, J., Brilakis, I., Haas, C. (2015). State of Research in Automatic as-Built Modelling. *Advanced Engineering Informatics*, 29(2), 162-171. doi:10.1016/j.aei.2015.01.001.

Phair, M. (2007). 3-D Laser Scanning Finds its Way to the Job Site. *Building Design & Construction*, 48(2), 24-25.

Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. In *Computer Vision and Pattern Recognition* 2007, IEEE Conference On, 1-8.

Piegl, L., and Tiller, W. (1997). *The NURBS Book*. 2nd Ed., New York, NY, Springer.

Qiu, R., Zhou, Q., Neumann, U. (2014). Pipe-Run Extraction and Reconstruction from Point Clouds. *Computer Vision–ECCV 2014*, Springer, 17-30.

Reddy, C., and Vinzamuri, B. (2013). A Survey of Partitional and Hierarchical Clustering Algorithms. In Aggarwal, C. C., and Reddy, C. K. (Eds.), *Data clustering : algorithms and applications*, Boca Raton, FL, CRS Press.

Rosenfeld, A., and Johnston, E. (1973). Angle Detection on Digital Curves. *Computers*, IEEE Transactions on, 100(9), 875-878.

Rusinkiewicz, S., Hall-Holt, O., Levoy, M. (2002). Real-time 3D model acquisition. In *29th Annual Conference on Computer Graphics and Interactive Techniques(SIGGRAPH '02)*, New York, NY, USA, 438.

Salazar, A., Cerón, A., Prieto, F. (2010). 3D Curvature-Based Shape Descriptors for Face Segmentation: An Anatomical-Based Analysis. *Advances in Visual Computing*, Springer, 349-358.

Salomon, D. (2007). Introduction. In *Data Compression: The Complete Reference*. 4th Ed., Springer London, 1.

Scape technologies. (2008). KUKA KR16 SCAPE bin-picker product sheet. <http://www.scapetechnologies.com/images/products/Standard_Bin_Picker_product_sheet_2.pdf> .

Schnabel, R., Wahl, R., Klein, R. (2007). Efficient RANSAC for Point-Cloud Shape Detection. In *Computer Graphics Forum*, 26(2), 214-226.

Seibert, H., Hildenbrand, D., Becker, M., Kuijper, A. (2010). Estimation of curvatures in point sets based on geometric algebra. In *VISAPP* (1), 12-19.

Sivic, J., and Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos. In *Computer Vision 2003*, Ninth IEEE International Conference On, 1470-1477.

Smith, L. I. (2002). *A Tutorial on Principal Components Analysis*. Cornell University, USA, 51(52), 65.

Son, H., Kim, C., Kim, C. (2014). Fully Automated as-Built 3D Pipeline Extraction Method from Laser-Scanned Data Based on Curvature Computation. *Journal of Computing in Civil Engineering*, 29(4), B4014003.

Statistics Canada. (2016). Gross domestic product at basic prices, by industry (monthly). <http://www.statcan.gc.ca/tables-tableaux/sum-som/l01/cst01/gdps04a-eng.htm> .

Statistics Canada. (2014). Capital expenditures for construction by sector, by province and territory (Canada). <http://www.statcan.gc.ca/tables-tableaux/sum-som/l01/cst01/busi01a-eng.htm> .

Tang, P., Anil, E., Akinci, B., Huber, D. (2011). Efficient and effective quality assessment of as-is building information models and 3D laser-scanned data. *Proceedings of the ASCE International Workshop on Computing in Civil Engineering*, 19-22.

Tang, S., and Godil, A. (2012). An evaluation of local shape descriptors for 3D shape retrieval. In *IS&T/SPIE Electronic Imaging*, 82900N-82900N-15, International Society for Optics and Photonics.

Tangelder, J. W., and Veltkamp, R. C. (2008). A Survey of Content Based 3D Shape Retrieval Methods. *Multimedia Tools and Applications*, 39(3), 441-471.

Teizer, J., Caldas, C. H., Haas, C. T. (2007). Real-Time Three-Dimensional Occupancy Grid Modeling for the Detection and Tracking of Construction Resources. *Journal of Construction Engineering and Management*, 133(11), 880-888.

ASME B31.1-2014 (2014). Power Piping: ASME Code for Pressure Piping, B31, American Society of Mechanical Engineers, New York, NY.

Thielsch, H. (1965). *Defects and failures in pressure vessels and piping*. New York, Reinhold Pub. Corp. .

Turkan, Y., Bosché, F., Haas, C. T., Haas, R. (2012). Automated Progress Tracking using 4D Schedule and 3D Sensing Technologies. *Automation in Construction*, 22, 414-421.

Vosselman, G., and Dijkman, S. (2001). 3D Building Model Reconstruction from Point Clouds and Ground Plans. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 34(3/W4), 37-44.

Wang, P., Mohamed, Y., Abourizk, S., Rawa, A. (2009). Flow Production of Pipe Spool Fabrication: Simulation to Support Implementation of Lean Technique. *Journal of Construction Engineering and Management*, 135(10), 1027-1038.

Wang, Y., Hao, W., Ning, X., Zhao, M., Zhang, J., Shi, Z., Zhang, X. (2013). Automatic Segmentation of Urban Point Clouds Based on the Gaussian Map. *The Photogrammetric Record*, 28(144), 342-361.

Weinmann, M., Urban, S., Hinz, S., Jutzi, B., Mallet, C. (2015). Distinctive 2D and 3D Features for Automated Large-Scale Scene Analysis in Urban Areas. *Computers and Graphics*, 49, 47-57.

Wintriss Controls Group. (2013). Automated Collection of Real-Time Production Data. ShopfloorConnect. < http://www.automation.com/pdf_articles/SFC_White_Paper.pdf>

Xu, L., Oja, E., Kultanen, P. (1990). A New Curve Detection Method: Randomized Hough Transform (RHT). *Pattern Recognition Letters*, 11(5), 331-338.

Yang, M., Kpalma, K., Ronsin, J. (2008). A Survey of Shape Feature Extraction Techniques. *Pattern Recognition*, 43-90.

Zhang, D., and Lu, G. (2004). Review of Shape Representation and Description Techniques. *Pattern Recognition*, 37(1), 1-19.