Spring 4-16-2020

Point to Pipe: Automatic Reconstruction and Classification of Pipes Using Lasergrammetry and Thermogrammetry for Building Information Modeling (BIM)

PATRICK BORGES RODRIGUES
SUNY College of Environmental Science and Forestry, pborgesr@esf.edu

POINT TO PIPE: AUTOMATIC RECONSTRUCTION AND CLASSIFICATION OF PIPES USING LASERGRAMMETRY AND THERMOGRAMMETRY FOR BUILDING INFORMATION MODELING (BIM)

by

Patrick Borges Rodrigues

A thesis
submitted in partial fulfillment
of the requirements for the
Master of Science Degree
State University of New York
College of Environmental Science and Forestry
Syracuse, New York
April 2020

Department of Sustainable Resources Management

Approved by:
Paul L. Crovella, Major Professor
Matthew Smith, Chair, Examining Committee
Christopher Nowak, Department Chair
S. Scott Shannon, Dean, The Graduate School

Acknowledgments

This thesis is a result of the contribution of many individuals over the last year and a half, and I would like to use this space to express my gratitude to them. It would not be possible for me to finish it without their help along the way.

First, I want to express my deep gratitude and admiration to my advisor, Dr. Paul Crovella, whose support started even before I applied for the master’s program at SUNY ESF. I was fortunate to have him as my advisor because of all his support, patience, and insightful comments on the development of my thesis, and my academic career overall. Other than that, I want to acknowledge his preoccupation with aspects related to my personal life, which helped me to keep moving forward.

Second, I want to thank Mr. Abraham Alpuerto Jr., who helped in the collection of the point clouds using his laser scanner. Other than that, I want to thank him for all the guidance and tips on the handling of the resulting point clouds and other aspects related to the registration of the images.

I am also grateful to my committee members. To Dr. Mohamad Razkenari and Dr. Endong Wang for the guidance and thoughtful insights on the preparation of my thesis proposal and final thesis. To Mr. Eric Wing, for accepting being the examiner of my committee. You are one of my main references in the BIM-world, and I am glad to have you as the reviewer of my work. Finally, to Dr. Matthew Smith for chairing my thesis defense.

I would like to thank the Physical Plant at SUNY ESF, in special to Mr. Joshua Arnold and Mr. Chad Lawrence, for helping me to identify the potential spaces that could be used for my research and for granting me access to the mechanical room I chose. Additionally, I would like to express my gratitude to the faculty and staff at SUNY ESF. Your support during my time at SUNY ESF was crucial for the successful completion of my degree.

Finally, I want to express all my gratitude to my family and friends, who, even 4,570 miles away, gave me all the love, support, and encouragement I needed to achieve this goal. This thesis is dedicated to you.

Table of Contents

List of Tables

List of Figures

List of Appendices

Abstract

P. B. Rodrigues. Point to Pipe: Automatic Reconstruction and Classification of Pipes Using Lasergrammetry and Thermogrammetry for Building Information Modeling (BIM), 140 pages, 9 tables, 23 figures, 2020. Automation in Construction citation style used.

Existing buildings account for 40% of global energy consumption, and two-thirds of them will still be operational in 2050. As most of these buildings lack the needed documentation for energy upgrades, it is essential to understand and represent the current conditions of their envelopes and mechanical systems. This project proposed a skeleton-based application for reconstructing and classifying pipes in existing buildings using point clouds from laser scanners and thermal images for Building Information Modeling (BIM) applications. MATLAB and Dynamo were used to process and model this information in Revit. Initial results indicate that the application is robust in identifying pipes and connections and that thermal images can be used to create sematic-rich models. These results can contribute to improving the capabilities of some of the commercially available software for pipe reconstruction in BIM and to expediting the digital reconstruction processes in existing buildings.

Key Words: Building Information Modeling (BIM), laser scanners, infrared cameras, semantic-rich models, skeletonization.

P. B. Rodrigues
Candidate for the degree of Master of Science, April 2020
Paul L. Crovella, Ph.D.
Department of Sustainable Resources Management
State University of New York College of Environmental Science and Forestry,
Syracuse, New York

CHAPTER 1: INTRODUCTION

BIM AND SUSTAINABLE DESIGN FOR EXISTING BUILDINGS

Among the current main goals of the construction industry is the improvement of the performance of existing buildings in terms of energy efficiency and energy use, a matter that is still not completely developed in Building Information Modeling (BIM) applications [1]. Existing buildings account for 40% of the total global energy consumption, which makes the construction industry one of the key players in energy conservation efforts [2].

The Global Status Report of the United Nations in 2017 [3] states that it is urgent *"to address energy and emissions from buildings and construction"* because two-thirds of the new buildings in the next 20 years are expected to be built in countries without mandatory building energy codes. For existing buildings, there is also a concern, as more than two-thirds of the current building area is expected to still be operational in 2050, and much of that was not built under the current stricter energy codes [4].

Especially after the energy crisis that started in 1973, energy consumption concerns have raised the awareness of the stakeholders involved in construction activities and, since then, many efforts have been made to improve the energy-performance of new buildings but also existing buildings. The oil embargo of 1973 led to the development of energy-conserving designs and the institution of code requirements for energy-efficient buildings given the rise in the awareness of the finiteness of energy resources by Architecture, Engineering, Construction, and Owner-Operated (AECOO) industry stakeholders [5]. Other than the finiteness of resources, climate change is another concern as 40% of the Greenhouse Gases (GHG) emissions are attributable to the AECOO industry [2].

Since the 1970s, requirements related to the energy efficiency of buildings, appliances, and equipment were included in many building codes and, then, spread throughout the U.S. [6]. Additionally, these codes have been updated periodically so that more efficient technologies and practices can be put in place to reduce building energy consumption [6]. As codes and regulations become stricter for the design and performance of new buildings, many best practices end up becoming the standard in new buildings over time. For existing buildings, however, the implementation of new technologies is more limited because there is a need to replace existing equipment [6].

In the AECOO industry, many of the currently available technologies may be used to help in the efforts of planning for construction activities, renovation projects, and operations and maintenance (O&M) in a manner that promotes sustainability and improved energy performance, as it is the case with BIM. BIM can be defined as an approach that uses a digital model of a building for information exchange through different phases of a building life-cycle, including design, construction, and facility management [7].

BIM has transformed the AECOO industry by allowing for the implementation of more efficient processes for managing the building life cycle in all its phases. Many researchers and professionals have advocated for the broader implementation of BIM in the AECOO industry to optimize design, construction, and operations and maintenance activities. In this sense, BIM uses Information Technologies (IT) for managing the building's life cycle through a process that involves many stakeholders, including architects, engineers, owners, real estate, fabricators, facility managers, regulatory compliance, sustainability, and others [8].

Given a series of constraints, which includes the costs of acquiring hardware and software, and the costs of training the employees in the new technology, a larger portion of the companies in the field have not yet fully changed their methodologies to BIM. In a survey conducted by Liu and Issa, 38.2% of the respondents said they use BIM in less than 10% of their projects, and a total of 56.4% of the respondents said they use BIM in less than 30% of their projects [9]. Added to that, when implemented, BIM use is not homogeneous throughout all phases in a building life cycle. If, on the one hand, the processes involved in the design and construction phases are relatively well-developed in BIM, with proposed manuals, workflows, frameworks and software applications, on the other hand, the processes involved in the operations and maintenance phases and renovations and retrofits are still in their infancy [1]. For post-construction phases, some frameworks have been proposed, but their use is still limited because of challenges relative to data requirements, lack of comprehensive workflows, the accuracy of documents, availability of as-is data, interoperability issues, among other [10,11].

Currently, BIM is mostly used for the projects of new buildings [1]. For renovating existing buildings using BIM, however, *"the literature suggests little maturity"* [12]. Many efforts, though, have been attempted in the last decade to develop BIM tools, workflows, and standards for existing buildings. These efforts can be explained by some of the needs the AECOO industry has faced in the last decades, which includes the need for bringing existing buildings to the level of energy performance required by many of the new codes and regulations.

Several researchers have studied the links between BIM and sustainability, including retrofitting projects, and found that a possible overlap between BIM and sustainability for existing buildings is in energy modeling and definition of scope for retrofitting projects, which may shorten the times required for achieving sustainability ratings and certifications [12]. Some best practices exist, however. For example, the *energiesprong* program developed in the Netherlands is helping to develop and implement net-zero refurbishment technologies for existing buildings by using accurate data collection through laser scanner technologies, BIM, and automation [13]. Other efforts include the development of green building certificates focused on renovations and operations and maintenance for existing buildings. Additionally, the Leadership in Energy and Environmental Design (LEED) presents certification options for existing buildings aiming to improve their shells, their mechanical systems, and their operations and maintenance [14].

STATEMENT OF THE PROBLEM

Although BIM is listed as one of the most promising technologies for the future of the AECOO industry, its adoption in some areas is still limited. This is especially true when considering the adoption of BIM in renovation projects *"because of technical, informational and organizational complications"* [12]. In its proposed BIM framework for building operations, maintenance and sustainability, McArthur [15] presents that the appropriate application of BIM for existing buildings requires overcoming four main challenges that include (i) identifying critical information for retrofitting, (ii) managing the transfer of information between software, (iii) creating the BIM model for an existing building, and finally (iv) handling uncertainty from incomplete documentation.

If, for new buildings, BIM models can be created from scratch, for existing buildings the creation of a BIM model requires the description of the as-is conditions of spaces and equipment, which is not always easy, especially when none or scarce documentation is available [1]. Even when documents are available, it is possible that they only reflect as-designed conditions or conditions that preceded major changes in spaces and equipment [16,17]. In other instances, it is still possible that different documents show different or conflicting conditions [18].

The format in which existing documents are available also has an impact on the efforts for reconstructing a BIM model [18]. If only 2D information is available, which includes physical copies, scanned images, pdfs, and even CAD files, there is a need to read and interpret the information before modeling the BIM model. In certain cases, previous BIM models may be available, but even in these cases, it is necessary to verify if the models appropriately describe current conditions. The creation of a BIM model from available documents in existing buildings is also considered a tedious and time-consuming process that requires proprietary formats and may result in inconsistent and non-integrated models [1]. Thus, integration of information is one of the greatest limitations for managing building renovation projects because, although having the data available in a digital format is important, it is more important to have it in digital formats that are compatible and have semantic meaning as to avoid conflicting information and duplicity, and to be used together for analysis [19].

In any of the cases, it is always possible that the accuracy of the information for existing buildings to represent actual as-is conditions may be compromised, and significant data collection efforts are needed. When the building does not have a BIM model, designers and facility managers need to work together to develop models for energy refurbishment applications [12]. Additionally, future developments should focus on automating the collection of as-built information and integrating it into BIM models [12].

Capturing as-is information in existing buildings is a challenging task because of aspects related to resource requirements, such as operators and equipment, time requirements, and all the involved risks in the process [20]. Over time, many processes have been developed to handle data collection requirements in existing buildings. Initial efforts involved the use of surveying equipment, including theodolites, for sizing spaces, and locating major equipment. These practices, although operationalizing the data collection processes in existing buildings, were characterized by the longer times required for data collection and by the risks posed to operators during surveying [21].

For capturing as-is conditions of large pipes in industrial facilities, for example, the need for targets and the long time it takes to capture information makes the use of theodolites less suitable if it is a hazardous environment [22]. Such hazardous environments include, for example, the risk of accidents and/or excessive exposure to harmful chemicals and extreme temperatures. Other concerns include the costs of such operations, as for some situations a full survey of a building can be prohibitively expensive [23].

Over time, however, more accurate and faster technologies for data collection in buildings were developed. Some of these technologies include Electronic Distance Measurement (EDM), Global Positioning System (GPS), photogrammetry, laser scanning, and building surveying [19]. With the advent of more advanced technologies, including photogrammetry and laser scanning (lasergrammetry), the requirements of long hours on-site and the risks posed to the operator were substantially diminished. Besides reducing the resource-intensive requirements of more traditional techniques for data collection, these two methods greatly improved the accuracy of the extracted data in existing buildings and reduced the possibility of human error in the process.

If the data collection techniques improved with the implementation of such methods, the same could not be said about the reconstruction techniques used to convert the point clouds resulting from photogrammetry and lasergrammetry into formats that were useful for renovation projects and O&M activities. For some time, although the data collected on-site was more accurate, the reconstruction phase of tridimensional models strongly relied on exclusively on manual user input, which still required many operator hours for modeling the building elements in CAD and BIM-based software, and was still prone to human error. The manual reconstruction of as-built models *"is time-consuming, labor-intensive, tedious, subjective, and requires skilled workers"*, which illustrates the need for automated and semi-automated techniques for creating as-built models [24].

With the consolidation of these data collection technologies, the focus of much of the researchers and practitioners was to develop frameworks, workflows, and computer applications to expedite the reconstruction processes when converting point clouds into models that were useful in other applications through semi-automatic and automatic reconstruction applications. Over time, many frameworks and applications were proposed, which considerably reduced the resource-intensive tasks related to the manual reconstruction processes in existing buildings.

In such cases, the elements of most interest in the scenes are usually walls, ceilings, floors, roofs, openings, i.e., doors and windows, structural elements, and some of the pipes and major equipment in HVAC systems. For most of these frameworks and applications, the requirements related to geometry reconstruction can be relatively well handled but, for almost all of them, the ability to embed sematic-rich information into the reconstructed models is still limited. For the case of pipes, specifically, although many commercially available programs for automatic reconstruction exist, none of them can automatically differentiate the existing systems, i.e., domestic hot water, hydronic system, sewer systems, steam, etc., as it is hard to embed semantic information into reconstructed models automatically. In its current state, the common practice for identifying building elements in the geometrical model is to use annotation categories in a process that is manual and requires expert judgment [1].

BACKGROUND AND NEED

As presented in the previous section, over time, some solutions were proposed to each of the problems still involved in the reconstruction of as-built BIM models for existing buildings, from data collection to the automatic reconstruction of models. First, more efficient data collection methods were developed, including photogrammetry and lasergrammetry; then frameworks and computer applications were developed to expedite the reconstruction efforts related to automatically converting the point clouds into useful formats for BIM applications.

As for data collection equipment, the current state of the technology involves the refinement of sensors, cameras, and laser scanners that can collect a larger amount of data in less time and with higher accuracies and less uncertainty. Some of the commercially available laser scanner options can collect millions of points and high-resolution images from the surveyed scenes in minutes, and provide accurate positioning of 3D points to the order of mm [17,25]. Other developments include the availability of handheld laser scanners, real-time registration of point clouds, more accurate positioning and registration when handling multiple scans, cloud processing and visualization, and noise removal.

When considering automating the reconstruction process in BIM-based applications, much current software that accompanies the scanners allows for the recognition and classification of coplanar points into different object classes including walls, ceilings, floors and roofs, which is useful for many applications, including energy simulation for existing buildings. When these point clouds are imported into some of the BIM-based software, some tools can reconstruct the geometries of some of these most common elements [26]. Other applications focus on the reconstruction of elements such as pipes, ducts, and structural members [21,27–29].

Efforts for automatically embedding semantic information to the reconstructed models also exist, although most of the proposed frameworks and applications are not enough developed to be commercially available. Examples of the current applications consider the merging of thermal information to building façades for assessing the energy performance of building envelopes, defining energy baseload for heating and cooling strategies, and finally for identifying leaks and thermal bridges through visualizing the reconstructed 3D models [30–35].

For pipe applications specifically, some of the current developments include using thermal images for identifying leaks and material deposition in pipes, although most of them focus on large diameter pipes (>300mm) in industrial and large-scale facilities including power plants, oil and gas utilities [36,37]. Thus, the development of automatic methods for reconstructing and embedding semantic data to BIM models was recognized as an important field for research. This can leverage many benefits to AECOO practitioners by reducing the efforts for creating BIM models of as-is conditions of existing buildings.

PURPOSE OF THE STUDY

The purpose of this study is to present the development and application of an automatic method for reconstructing and classifying round pipes in existing buildings for BIM applications. The first objective is to develop an algorithm for geometric reconstruction of pipes in existing buildings for BIM applications by processing point clouds from laser scanners. The second objective is to develop an algorithm for classifying the pipes' systems in the BIM models using information from thermal images from infrared cameras. To test the proposed application, three scenes from a mechanical room in one of the academic buildings at the State University of New York College of Environmental Science and Forestry (SUNY ESF), Illick Hall, were used.

RESEARCH QUESTIONS

- To which extent can thermal information be used in the classification of mechanical systems comprised of round pipes in automatic reconstruction applications for BIM?
- What are the effects of insulation around the pipes on the ability of the proposed application to correctly classify the mechanical systems?

SIGNIFICANCE TO THE FIELD

Stakeholders in the AECOO industry can benefit from the research because it aims to expedite the reconstruction process of the as-is conditions of pipes in existing buildings while also providing semantic rich information such as pipe system classification. Thus, the proposed application can reduce some of the cost and time requirements related to more traditional techniques and improve the planning for retrofitting projects and operations and maintenance activities in existing buildings.

As for extending this research, the present work can be improved to include the use of other sensors and variables not considered herein, it can be adapted to reconstruct rectangular ducts and conduits, and finally, it can be improved in terms of its computational performance and processing times.

CHAPTER 2: LITERATURE REVIEW

INTRODUCTION

Although the introduction of BIM in the AECOO industry aimed to improve the performance of all phases of a building life cycle, the extent to which the technology has been applied to existing buildings, for both operations and maintenance activities and renovations and retrofits purposes, has not yet reached the desired levels. Many of the challenges related to using BIM for existing buildings result from insufficient and/or inaccurate documentation about the as-is conditions of these buildings, which often require collecting as-is information before commencing such activities.

Traditional techniques for collecting as-is conditions data in buildings include the use of theodolites, total stations, measuring tapes, and other handheld devices that heavily rely on the operator. New developments allowed for the introduction of photogrammetry and more recently, 3D scanners, which considerably reduced the time for collecting data while increasing the accuracy of the geometric information of the reconstructed models. The reconstruction process of the tridimensional models, on the other hand, continued to rely exclusively on user intervention, and this process continued to be time-consuming and costly.

The introduction of automatic and semi-automatic methods for reconstructing the geometries of existing buildings from point clouds, however, drastically reduced the times required for reconstructing the models while reducing the number of man-made errors in the process. Even though the use of these methods improved the reconstruction process overall, they are usually limited as they do not provide semantic-rich information about the reconstructed models but only geometric information.

The literature review will address the efforts being made to produce as-built BIM models for existing buildings and the possibility to embed semantic information into these models using thermal images. The first section will address the use of BIM for existing buildings. The second section will discuss data collection strategies for existing buildings, which include 3D mapping techniques, thermography, and the fusion of geometric and thermal data. Finally, the third section will focus on the efforts for creating the as-built BIM models from the collected data.

BIM FOR EXISTING BUILDINGS

BIM

Before the development of current technologies for building information exchange and interoperability, traditional facility delivery methods were not integrated and relied on 2D drawings communication, which frequently resulted in errors, omissions, delays, increased costs, and even lawsuits [8]. The urge to eliminate such issues led to the development of new project delivery methods and of new technologies for the proper management of project information, which, although facilitating information sharing, could not significantly reduce the impacts and frequency of the aforementioned problems due to the 2D nature of the information [8]. With the introduction of BIM, however, many of these issues were resolved, and the AECOO industry experienced improvements in collaboration, project visualization, and exchange of information.

Many definitions of BIM can be found in the literature [38]. The National Building Information Modeling Standard™ (NBIMS) [2] presents that the term BIM may refer to a *product* that represents a building – Building Information Model; an *activity* used to create the Building Information Model – Building Information Modeling; or a *system* that focuses on business structures to provide better project results – Building Information Management.

Alternatively, the International Organization for Standardization (ISO) [39] defines BIM as the *"use of a shared digital representation of a built object […] to facilitate design, construction and operation processes to form a reliable basis for decisions"*.

Other definitions exist and may represent BIM in a narrower or broader perspective, 'little BIM' vs. 'BIG BIM', respectively [40]. The main difference between the two approaches is that while 'little BIM' sees the model as an information management repository, the 'BIG BIM' approach divides BIM *"into interrelated functional, informational, technical and organizational/legal issues"* [40]. In any case, BIM has improved the management of building information through efficient information access and visualization in all phases of a building's life cycle [17].

In its current state, BIM has promoted the use of more efficient construction delivery methods such as Integrated Project Delivery – IPD, which allows for closer cooperation among the project teams [8]. Although projects will benefit differently from BIM application, BIM can benefit a project in many areas, including (i) preconstruction benefits to the owner through a better definition of scope, and collaboration; (ii) design benefits through accurate visualization, consistent drawings, enhanced collaboration among disciplines, cost extraction, and improved energy performance and sustainability; (iii) construction and fabrication benefits through the use of design models for fabrication, quick reaction to change, early error identification, and synchronization of design and planning; and (iv) post-construction benefits, including improved commissioning and information handover, better management and operations of facilities, and integration [8].

As BIM models are tridimensional, they provide an accurate representation of building components, with the advantage of also functioning as a database that allows for the management of spaces and assets, quantity take-offs and estimates, building energy analysis, among others [41]. Other than that, BIM can promote more efficient and collaborative project delivery methods, which can reduce costs and result in improved communication, early decision making, improved coordination, and support to complex analysis [41].

Although BIM is intended to be used throughout the life cycle of a building, many researchers have found that BIM implementation is more frequent in the design and construction phases of a project than in the Facility Management (FM) activities, as many issues still limit the application of BIM in FM [11,42]. First, although some guidelines for using BIM for FM applications exist, such as the one developed by the U.S. General Service Administration (GSA), many case studies have shown that it has been challenging for project teams and FM staff to integrate BIM and FM [43]. Second, liability concerns and uncertainty related to intellectual property have limited data exchange and the use of full life-cycle BIM approaches [44]. Finally, there is a lack of understanding of the information needs of operations and maintenance personnel by the part of designers and constructors because facility managers usually do not participate in early phases of the building life cycle [9,10].

This verification of the limited application of BIM during O&M, however, shows how much room there is for improvements. Current approaches in the AECOO industry focus on optimizing design and construction phases, even though most of a facility life cycle's costs and environmental impacts occur during O&M [2]. Many researchers present that overall, the life cycle costs associated with O&M can be up to 7 times higher than the costs associated with design and construction phases, or between 60% to 85% of all life cycle costs [45–47].

BIM and Existing Buildings

As mentioned, the use of BIM in the AECOO industry occurs preferentially during design and construction phases as many issues still limit its wider application during O&M phases. One such issue is the existence of information in a format that is useful for BIM. For example, more than 80% of the residential buildings in Europe were built before 1990, and thus most of them do not have BIM models reflecting their as-is conditions [40].

Among the main characteristics that make BIM suitable for existing buildings is the fact that the embedment of information into an as-built model of a building increases the ability to operate the building and its systems more efficiently, and the ability to better plan for retrofitting projects and even demolition projects. Many examples can be found in the literature, such as the use of BIM for planning for maintenance activities in existing buildings, for project retrofitting, and for studying the impacts of different alternatives in deconstruction projects [1,47,48].

The as-built model is one of the key requirements for the proper implementation of BIM-enabled FM [11]. Different than the as-designed documents and models that reflect the design intents of the designers, the as-built documents and models are representations of the building at the moment of the survey, which may differ from the as-designed documents [49]. Most of the changes between the as-built models and as-designed models are a result of undocumented changes that take place during the building life cycle [50]. These undocumented changes and deviations from the as-built documents have associated high-costs imposed on building owners [17]. These costs were estimated in a 2004 National Institute of Standards & Technology (NIST) report [51], which presents that $4.8 billion per year is spent on labor charges to verify and validate existing as-built documentation, and $1.5 billion per year is spent due to information delays and inadequate information in FM applications.

Some trends can be observed in the development of as-built models. First, for new buildings, many owners and facility managers now require the delivery of as-built BIM models at the end of the construction phase [8,17]. Second, for existing buildings, the digitization, and exchange of geometric and semantic information as BIM models for facility management purposes is becoming more frequent [17]. Alternatively, even though these models can help owners and facility managers, they have not been used more broadly in FM applications because they have insufficient information, or they have superfluous information [52].

When properly managed, though, the use of BIM for FM in existing buildings benefits many areas, including real-time data access and component location, space management, planning for non-capital construction, energy monitoring, and training staff [17]. The development and use of COBie (Construction Operations Building information exchange) also intend to help facility managers by providing comprehensive information about building assets using an open format that allows for the management and exchange of information using simple spreadsheets [53].

Sustainability concerns about the built environment, especially when considering building energy consumption, also play an important role in the development and use of BIM in existing buildings. The development of new "green" technologies and their use in buildings has allowed for substantial reductions in energy consumption in new and existing buildings. For example, following the oil embargo of 1973, residential energy consumption has decreased 37% per household and 27% per capita, and commercial energy use per square foot has decreased 25%, even though the sizes of new buildings and the number of electric equipment in buildings have increased [6].

For new buildings, applying currently available technologies for energy efficiency can reduce energy consumption and GHG emissions by 30% to 40% over the entire building lifespan, while for existing buildings the options are more limited due to the need of replacing building equipment and components [6]. To put this into perspective, Ham [54], citing a report from the Department of Energy (DoE) in 2010, presents that 87% of the residential buildings and 74% of the commercial buildings in the U.S. were built before 2000, under less strict energy-performance regulations, and that these buildings waste 35% of their total energy input.

Many efforts are still needed, though, as *"the construction industry is in the middle of a growing crisis"* because buildings consume 40% of the raw materials and 40% of the total global energy, and contribute to 40% of GHG emissions and 20% of landfills' waste on a global scale [2]. Added to that, there is still much waste associated with construction activities. For example, in 2007 the Construction Industry Institute estimated that around 57% of construction-related efforts are wasted each year, which accounts for over US$ 600 billion/year [2].

Current efforts for reducing energy consumption in existing buildings have focused on accurately representing the as-is building conditions to improve the quality and effectiveness of retrofits [35]. In this sense, BIM has also been used in simulating energy consumption in existing buildings, in identifying problematic areas (leaks, thermal bridges, etc.) in the building envelope, and finally in revamping heating, ventilation, and air conditioning (HVAC) systems in existing buildings [45,55–57].

DATA COLLECTION STRATEGIES IN EXISTING BUILDINGS

3D Mapping

Capturing and modeling 3D information from existing structures is a challenging task that has been addressed with the use of technologies such as EDM, GPS, photogrammetry, and laser scanning [58]. Recent developments in the creation of 3D models for existing buildings are a result of the development of technologies such as terrestrial laser scanners and photogrammetry. In both cases, it is possible to create a point cloud, a set of data points in space that conserves geometric and spatial relationships among them, that can be used for BIM applications.

3D mapping techniques can be divided into two main categories based on the type of sensor used: active and passive [59]. Active mapping techniques use active sensors that transmit energy that is reflected by the surface of the objects, while passive mapping techniques use passive sensors that do not transmit energy, but rather use natural light from the surroundings to collect information about the objects [60]. Active mapping techniques allow for the direct acquisition of point clouds with high geometric accuracy, as it is the case for terrestrial laser scanners, while in passive mapping techniques a set of images is used to reconstruct a 3D model, as it is the case with photogrammetry and videogrammetry [59].

Given their higher precision and speed when compared to passive mapping techniques, some active mapping techniques that use laser scanners and Light Detection and Ranging (LiDAR) instruments have recently found extensive application in construction. The use of such instruments, conversely, is associated with higher costs and training requirements for operators when compared to traditional passive mapping techniques [59].

Lasergrammetry

Lasergrammetry can be defined as the process of directly acquiring 3D information about objects and scenes by projecting a laser beam onto the surfaces of these objects and measuring angles and distances from the reflecting surfaces to the scanner [49]. When using laser scanners to map existing scenes, two main approaches can be used: Terrestrial Laser Scanning (TLS) and Aerial Laser Scanning (ALS), known as LiDAR [61]. The only difference between the two systems is the fact that in airborne (ALS) applications, the scanner is mounted in a flying device while the terrestrial counterpart is not [19].

This equipment has found extensive application in many areas, indoor and outdoor building modeling, 3D city modeling, bridge modeling, energy efficiency studies, pipe modeling, tree and vegetation cover modeling, and cultural heritage modeling [27,61–66]. Other interesting applications include the use of mobile laser scanners in autonomous cars [67]. As the purpose of this project is to assess the ability of the proposed algorithm to reconstruct the geometry of pipes in interior spaces in existing buildings, the focus of this section will be on terrestrial laser scanners.

The functioning of terrestrial laser scanners is based on the measurement of thousands, or millions, of distances and angles from the laser scanner to the surfaces of real-world objects in the scene, which results in a point cloud [66]. These three measurements, distance, horizontal and vertical angles, then, are used to determine the x, y, z coordinates of each point in the scene.

Laser scanners can be classified under two different types based on their principles of operation: ranging scanners, which includes scanners based on time-of-flight and phase comparison, and triangulation scanners [68]. In time-of-flight scanners, a high number of laser pulses is emitted and reflected by the surfaces of the objects in the scene, and the distances between the scanner and each of the points are determined using the travel time between the transmission of the pulse and reception of the reflected pulse and the speed of the light [68]. Additionally, two angles are measured in the process for each point, a vertical and a horizontal angle. To collect this pair of angles, the scanner is rotated around its vertical axis, and it is equipped with a rotating mirror that moves the laser pulses up and down. Then, the distance and vertical and horizontal angles are used to determine the x, y, z coordinates of each point in the point cloud. The operation of phase-based scanners is similar to the operation of time of flight-based scanners, with the difference that the distances are determined using the phase shift between the emitted and reflected signals of the laser beam modulated by a harmonic wave [68]. Finally, triangulation-based scanners use a transmitting device at one end of the instrument that emits the laser beam, which is reflected by the object surface, and one or two charge-coupled devices (CCD) cameras at the other end of the instrument, which detect the reflected signal [68].

Other than the location, i.e., x, y, and z coordinates, of each point in a scene, laser scanners can measure other parameters, which includes the intensity of the returned signal and the colors of each point [69]. In this context, intensity refers to the strength of the backscattered signal and varies with the material [70]. Intensity values are usually employed to support the visualization of point clouds, although they are rarely used for most applications [70]. The colors of the points are another parameter frequently captured in modern laser scanners. Color collection is usually made with the help of high-quality images from cameras associated with the laser scanners [69]. These images are also used in the projection of the RGB colors from the imaging cameras onto the geometric point cloud from the laser scanner, which helps in visualizing the point cloud and in the identification of the materials' textures [68]. Some modern scanners also have other sensors associated, which includes, for example, thermal sensors [71].

As mentioned, the output of the scanning process is a point cloud. A point cloud can be defined as a set of x, y, and z coordinates of points that are used to describe the spatial distribution of objects in a scene. Usually, the structure of the point clouds can be described in a table containing, in each row, the x, y, and z coordinates of each point, and it is also possible to have intensity values, I, and RGB color information.

As for almost all cases, because a single scan cannot capture information about all the elements in a scene, multiple scans from different positions are usually required [28]. If the location of each scan is properly selected, the individual scans will present overlapping regions that will be used to merge individual scans into a single point cloud in a process called registration. The registration process can be defined as the alignment of two or more scans of the same scene and begins with bringing the data from each scan into a common coordinate system [72]. Various registration methods exist, each one with its advantages and disadvantages. Among the most popular registration processes is the use of artificial targets that are used for co-registration of multiple scans into a single point cloud [72]. In such cases, the radii of the spheres are known and are used to determine the center of each target [72]. When multiple individual scans contain the same target, the coordinates of identical points in these targets are used to align these individual scans.

Among the main challenges for the use of 3D scanners are restrictions related to line-of-sight, which results in only 2.5 dimensions (2.5D) of an object being captured in a single scan, and partial occlusion, which is caused by objects in the scenes hiding other objects and regions of interest, which requires multiple scans for each space [28]. Other problems related to extracting information from point clouds from laser scanners include varying point density resulting from the laser scanner itself and the registration process, undersampling caused by partial occlusion, and the presence of noise [66]. Finally, there are still problems related to the presence of noise in the edges of objects in the models as a result of the quick change in the angle formed between the laser beam and the surface normal during data collection [73].

In the AECOO industry, laser scanners have gained a lot of attention because they can be used to rapidly and accurately extract 3D information from a scene [24]. In some current applications, the association of laser scanner with other technologies is explored, including the merging of terrestrial laser scanner data with LiDAR data, the construction of IFC (Industry Foundation Classes) models, and the printing of architectural elements using 3D printers [58]. In others, algorithms are developed to process and integrate data from both terrestrial and aerial laser scanners to generate complete detailed 3D models [61].

Cultural heritage applications are also of interest, especially given the non-destructive and non-contact nature of laser scanning techniques. Examples include the use of AM-LR (Amplitude-Modulated Laser Rangefinder) scanner, a scanner developed by the ENEA National Research Laboratories in Frascati (Italy), for cultural heritage applications, and the use of laser scanners for assessing the conservation state of historical buildings [62,74].

Many of these applications aim to provide a comprehensive 3D model of the built environment using technologies such as CAD (Computer-Aided Design), BIM, and others. Such an example includes the combination of terrestrial laser scanners and total stations to create models of exterior and interior spaces of buildings in BIM for data management purposes [63]. Yet, it is possible to combine laser scanner data, digital images, and other sources of data to create Historic Building Information Modeling (HBIM) models with a high level of details that can be used for augmented reality and virtual reality applications [75]. Other applications include the use of laser scanners and photogrammetry for BrIM (Bridge Information Modeling), in a process that allows for the creation of BIM models of bridges, simulations, and the storing of results and reports associated with the structural behavior of the bridge [65].

Finally, for the reconstruction of pipes in industrial plants, laser scanners have allowed the collection of geometric data from pipes with higher resolutions, accuracy, and speed [28]. Additionally, laser scanners also have the potential to automate the reconstruction process in 3D applications [24,73]. Alternatively, some drawbacks still limit the broader application of laser scanner in such environments, which include complex occlusion, noise due to reflective surfaces, and problems related to locating the scanner in congested areas [28]. Yet, because point clouds from laser scanners have a large number of points, including the ones that represent complex geometries of some objects and noise, the reconstruction of the digital model can be time-consuming and costly [27].

Photogrammetry

Photogrammetry can be defined as the process of extracting geometric information of objects from photographs using a single photo, multiple photos, analog or digital images, or video images (videogrammetry) [17]. It is one of the non-contact processes for acquiring measurements from 3D objects using passive sensors, being classified as one of the methods in Image-Based Modeling (IBM) [76]. IBM methods use mathematical formulas to extract 3D object information from 2D images, and they are widely used in many areas, including architecture and city modeling [76].

One such widely used photogrammetric technique is the Structure-from-Motion (SfM), in which a set of images from different viewpoints are used to reconstruct 3D objects [77]. SfM is a low-cost photogrammetric method that provides high-resolution 3D models, and it differs from traditional photogrammetry in the sense that the user does not need to use targets to solve for camera positioning and orientation [78]. In SfM, camera positioning and orientation are solved automatically by using a set of images with a high level of overlap, which is a result of a moving sensor capturing images from different positions [78].

Conversely, because SfM does not use targets with known locations in the scenes, the resulting point clouds lack scale and orientation and are presented in a relative coordinate system that requires further alignment [78]. Common practices for solving such an issue include the use of physical targets with well-defined centroids and the use of auxiliary scales that are introduced in the scenes for scaling the reconstructed point cloud or mesh [78,79].

SfM is the most used 3D reconstruction method using unordered sets of images, but even though it has experienced advances in its capabilities over the years, the method still lacks robustness, accuracy, completeness, and scalability [77]. Due to such limitations, it is common that a set of initially registrable images are not registered, or a broken model results [77].

Rabbani and van den Heuvel [73], when discussing the use of photogrammetry in industrial environments, present that there are some limitations in the process. These limitations include the need for intersecting images for the same point, which is needed to retrieve 3D information; the presence of clutter and uncontrolled lighting; and finally the differences in information quality from the edges and the surfaces of objects, as the edges contain more contrast and, consequently, provide more information [73].

Before the advent of laser scanning technologies, photogrammetry was commonly used to model industrial plants because it is faster and more convenient than performing manual measurements [73]. When compared to laser scanners, however, image acquisition is more flexible and less costly, especially because the required sensors are portable and cheaper [73,76].

In terms of spatial accuracy, even though the point clouds from laser scanners are more accurate, the results from photogrammetric surveys can help project managers to extract as-built semantic information and can be used during tasks that do not require higher levels of position accuracy [60].

Examples of the use of photogrammetry in the construction industry include the use of photogrammetric point clouds and site images for construction progress monitoring using BIM models [80,81]; city and historical building modeling using aerial photogrammetry [82–84]; and building structure modeling [85,86]. For architectural applications, fully automated 3D reconstruction of architectural elements can be developed using SfM [87]. Additionally, photogrammetry can be used in HBIM applications to produce models with high-levels of detail using photogrammetry software, BIM software, and Trimble Sketchup [88].

For as-built applications, manual measurements and photogrammetry can be used to verify the dimensions of interior and exterior spaces in existing as-built BIM models [17]. In such cases, photogrammetry presents some advantages over manual processes, which include reduced time and effort for data collection, facilitated access to elements in the building geometry, and simultaneous generation of 2D and 3D information [17].

Infrared Thermography

Different than photogrammetry, which is based on the visible spectrum of light, Infrared Thermography is a technique based on the reconstruction of images from infrared light, which is invisible to the human eye [64]. The thermal camera detects the amount of energy radiated by the objects in the scene, which is proportional to the objects' temperatures, and produces a thermal map in which the color of each pixel represents the amount of detected energy in a particular location [64]. Thus, a thermal image can be understood as a 2D matrix that stores temperature information in rows and columns [59].

The detector of the thermal camera, a transducer that absorbs the energy the objects emit, converts the absorbed energy into a signal that can be stored in the thermal map [89]. These Infrared detectors can be grouped into two main groups based on their functioning: thermal detectors and photon detectors [89]. Thermal detectors absorb the infrared energy emitted by the objects and warm up, then the changes in temperature-dependent properties of the conductor's materials are used to determine temperature differences in the object [89]. Photon detectors, on the other hand, absorb the energy that the objects emit and release electrical carriers, at the expense of requiring cryogenic temperatures to cool the sensors [89].

Even though the use of Infrared Thermography started with military purposes in mind, many civil applications are benefitting from the technology [90]. As the use of infrared thermography allows us to "see" the heat, it results in faster and more correct decisions that promote efficiency, safety, and economy [90]. Many areas benefit from the use of Infrared Thermography, including aerial infrared imaging, building surveys, predictive maintenance, cultural heritage, medical applications, and research and development, among others [89]. The application of Infrared Thermography in Building Science has many pros and cons, needs and areas of application in existing buildings, which includes energy savings in buildings through identifying problems associated with the buildings envelope, structural analysis, moisture detection, microclimate monitoring, and Heating, Ventilation and Air Conditioning (HVAC) performance monitoring [91].

The wider application of infrared thermography in different fields is a result of the fact that these measurements are non-invasive [64]. It has been used for more than 50 years for monitoring building energy performance of both the building envelope and physical plant, and its current widespread use is a result of lower costs and better equipment [91].

Conversely, infrared thermography still has some limitations, which include the impossibility to generate georeferenced temperature data and isotherms due to the distortion of the camera's lenses [64]. Thus, many applications have been limited to qualitative studies where the images allow for the detection of hot and cold areas, and the identification of leaks and thermal bridges [64]. Additionally, because most of the old building elements lack thermal performance information, the use of as-designed building information for energy analysis is likely to overestimate the energy performance of these elements, which makes it necessary to collect and reflect the as-is conditions of the building elements for proper energy modeling [92].

For building applications specifically, the use of IR thermography can support preventive maintenance and energy savings for indoor and outdoor spaces, being its use indoors more accurate and its use outdoors more productive [91]. Some examples of current outdoor applications aim to assess the energy efficiency of building façades [31,93,94]. Other applications either use thermal images aiming to produce 3D models and orthothermograms or use existing BIM models to improve their semantic information including updating R-values of BIM elements [35,95–99]. Applications involving the indoor environment can also focus on identifying potential problems related to energy performance or can focus on HVAC monitoring [31,100].

In his dissertation, Ham [54] presents the development five methods based on the use of thermal cameras, including (i) a new automated method for reconstructing 3D scenes using digital and thermal images, which results in spatial-thermal 3D models; (ii) a new Energy Performance Augmented Reality (EPAR) environment to determine deviations between actual and expected energy performance in buildings; (iii) a new method for determining the actual thermal resistance in buildings using the previously developed 3D models, and detecting problems related to condensation; (iv) a method for determining the heat loss through problematic areas and the total savings that can result from solving these problems; and finally (v) a new method for updating the thermal resistance of the BIM model using the actual performance of the building previously determined.

Fusion of 3D Mapping Techniques and Thermal Imagery

To overcome the limitations associated with the lack of geometrical information in thermal images, many studies have proposed the inclusion of metric data from other sources such as photogrammetry and laser scanners. The merging of such information has allowed the measurements of distances and areas to achieve a deeper understanding of the geometry of and thermal performance of the objects of interest.

With the development of laser scanning technologies, many of the limitations associated with photogrammetry were solved, which does not mean that the use of images in 3D information extraction lost their utility [73]. On the contrary, because no single technique can satisfy the requirements for all objects and environments, the combined use of both photogrammetry and laser scanning offers the best cost-effective solution [73,76].

Most of the current laser scanners have visible cameras built-in, and some models already have thermal sensors embedded [71]. Other applications try to overcome the limitations of using less expensive, and consequently less accurate sensors, by merging sets of less expensive sensors, including cell phone cameras, multiple digital cameras, RGB-D images from videogame consoles and so on. Cho et al. [35] present a comprehensive review of some of the current techniques being used for 3D as-is energy modeling for buildings.

Recently, the merging of thermal and geometric data has become a topic of research interest, especially when the geometric data is the main source of information, as it is the case for 3D building modeling [101]. Many examples can be found for building energy performance evaluation for both indoor and outdoor spaces and using both photogrammetry and lasergrammetry associated with infrared thermography.

For building energy performance applications, examples of approaches using photogrammetry and infrared thermography include the use of visible and thermal cameras, depth cameras and thermal cameras, and even smartphone and thermal cameras [59,101–107]. In some cases, RGB and thermal images are merged, but using drone-mounted cameras for the built environment [108]. Yet, other applications present the use of laser scanners and infrared thermography [32,34,64].

Some applications use all three sensors: laser scanner, visible camera, and thermal camera. One such example includes the construction of hybrid systems composed of a laser scanner, a color camera, and a thermal camera for creating semantic-rich 3D models for indoor scenes [109]. Other applications include the development of robots equipped with a laser scanner, a thermal camera, and an RGB camera to generate 3D thermal models of indoor environments in a manner that the information from the three sensors is automatically joined into a common reference system [110].

Finally, there are examples of integrated applications of these sensors for purposes other than energy efficiency analysis in buildings, such as the use of terrestrial laser scanners and thermal cameras to detect anomalies in the architectural and structural elements in historical buildings [74]. In this particular application, it was possible to identify cracks on the building façade, different materials, and hidden elements resulting from historical rebuilding works, and structural defects [74].

AUTOMATIC AND SEMI-AUTOMATIC RECONSTRUCTION OF AS-BUILT BIM MODELS

Scan-to-BIM: Creating As-Built BIM Models

As presented previously, there is a need for BIM models representing the current conditions of existing buildings to produce better results in the efforts to manage building operations, plan for renovations and retrofits, and simulate building performance. On many occasions, collecting data using 3D imaging techniques is followed by some level of modeling, which is usually a manual process, and involves cleaning, segmenting, and fitting geometric primitives to the elements in the scene [69].

The process of using laser scanners to collect data that is used for BIM modeling is known as Scan-to-BIM. Three tasks are involved in this process: (i) *data collection*, when the point clouds are collected with the help of the laser scanner, (ii) *data preprocessing*, in which noise is removed and the different scans are registered into a single point cloud, and (iii) *modeling the BIM*, in which the point cloud is used to produce the semantic-rich BIM model [24]. For such applications, the U.S. General Service Administration (GSA) [69], in its GSA BIM Guide Series vol. 03: BIM Guide For 3D Imaging, provides a comprehensive reference describing the different phases and requirements for data collection, processing, and modeling in public projects requiring documentation of as-is conditions in buildings.

To accomplish this, manual, automatic, or semi-automatic methods can be used. Manual processes are more common than semi-automatic and automatic methods, and the main difference between semi-automatic and automatic methods is the fact that while the former requires significant user input, the latter does not [24,111].

In most of the cases, the use of these manual processes is labor-intensive, time-consuming, tedious, may require skilled workers, and is likely to provide different results if performed by different modelers [24,111]. Yet, depending on the input and output requirements of each project, the problems associated with as-built reconstruction can vary. On the input side, the existence of prior data can enhance the reconstruction process when used along with collected data, while on the output side the requirements relative to the level of detail and semantics, and the type of objects influence the reconstruction efforts [24].

Some companies have specialized in producing as-built BIM models and offer specific services for scanning and/or modeling BIM models using the Scan-to-BIM process [112–114]. In such cases, the company may be hired for both scanning the scenes and modeling the as-built BIM, or for modeling the BIM from a point cloud produced by someone else. Depending on the complexity and requirements of each project, these processes may require longer times, which may take up to several months [24].

Given the fact that automation is expected to solve some of the main problems associated with the manual modeling process in Scan-to-BIM applications, it is an area of active research interest [111]. More recently, many applications and plugins were developed to automate the reconstruction of as-built BIM models from point clouds [115,116].

Automatic and Semi-Automatic Modeling of BIM models from Point Clouds

Other than the previously mentioned problems associated with sustainability and energy consumption in new and existing buildings, and the limited use of processes that contemplate the entire building life cycle, productivity in the construction industry is another concern. As presented in Sacks et al. [8], based on data from the U.S. Census Bureau for 2016, between 1967 and 2015, the labor productivity index of the manufacturing industry grew by over 100% while it remained almost constant for on-site construction.

For the case of improving the productivity of the tasks associated with collecting and modeling the as-is conditions in existing buildings, advances in 3D scanner technology accelerated the collection of denser and more accurate point clouds. Additionally, the development of automatic and semi-automatic methods for reconstructing the geometry of these buildings and their systems decreased the times and user intervention requirements. As the concept of as-built BIM models is relatively new, however, these software applications are still fragmented [24].

There are comprehensive reviews on current efforts and the state of the art of automatic as-built BIM modeling, in which the main steps involved in data collection, processing point clouds, and as-built BIM modeling are presented [49,117]. Over time, many authors have proposed and developed methods for automatically modeling the as-built and as-is conditions in existing buildings. These conditions may involve the modeling of exterior conditions, façades, interior conditions for both architectural/structural elements and building systems, or a mix of the two. In many cases, these applications use the concepts of the Manhattan-World assumption, in which the major elements (floors, walls, and ceilings) are parallel to the planes defined by two out of the three mutually orthogonal directions, x, y and z, i.e., x-y, x-z, y-z planes [111].

Some of the available applications include the development of automatic approaches for modeling parametric models of indoor environments using complex point cloud datasets [26]. Other applications use semi-automatic methods for segmenting, extracting, and reconstructing floors, walls, and ceilings in the IFC format using point-clouds as an input [111]. Yet, there are occasions in which the method for modeling the as-built BIM model for indoor environments uses point clouds that are segmented and their boundaries are traced automatically, but the final BIM model is modeled manually [118].

Huber et al. [119] describe four methods their group was developing to automatically recognize and model planar elements, including floors, walls, and ceilings, but also openings, i.e., doors and windows, even in face of clutter and partial occlusion. The authors also focused on developing IFC models to represent as-built information. As a result of this work, the authors developed an algorithm able to automatically identify and model many of the aforementioned indoor planar elements to create semantically rich models using point clouds, even when clutter was present and partial occlusion occurred [50].

There are also automated applications for generating parametric models using a combination of laser scanners and videogrammetry [120]. In such cases, point clouds are used during the automatic recognition and classification of objects, and the georeferenced videos are used during the human-assisted model assembly in which the operator can solve any ambiguities the automated application could not properly identify [120]. The examples shown in this section refer to the modeling of architectural and structural elements in a building only. The next section focuses on automatic pipe modeling, the concentration of this project.

Automatic and semi-automatic modeling of Pipes

There is a growing demand for as-built 3D models of pipes [21,22,28]. These models are important because pipes usually represent a major part of a facility, which has direct impacts on operations/maintenance activities, and because they can be used for planning and renovations [21]. Additionally, as the replacement of facility equipment has become more frequent, the development and use of 3D models from laser scanner data are expected to improve such tasks [27].

Initial efforts included the use of digital images and CAD-based applications for reconstructing straight and curved pipes in tridimensional formats [22,23,121,122]. More recently, the use of laser scanners, with and without the combined use of photogrammetry, has been more commonly used, especially for BIM applications [16,123,124].

The geometric elements that characterize pipes, i.e., cylinders connected by elbows and tees, are a key difference between pipes and other building structures, which allows for the development of various reconstruction methods for pipes [117]. When no as-designed BIM models are used in the reconstruction process, it is possible to use geometric features, such as curvature of the pipes, to detect pipes, and use the geometric skeletons of pipes and methods such as the Hough transform to detect and model the pipes' geometries [117].

Some methods use laser scanner data to automatically reconstruct the geometry of straight pipes, elbows, and tees using skeletonization and topological thinning [21]. Others, reconstruct parametric models of pipes using point clouds by first estimating the pipes' skeleton, and then approximating the pipes' diameters using a circular arc and two line segments tangent to the arc [125]. The interesting feature of this last application is the fact that it is intended for bent pipes that do not follow the more common structure of straight cylindrical pipes.

It is also possible to use the points that represent the surfaces of the pipes, along with information about the local curvature of these pipes, to extract geometrical information about the pipes [28]. One such an interesting feature of this framework is the fact that it works even when a single scan is available, which can help in many practical applications that need to deal with incomplete point clouds due to partial occlusion [28]. Also, there are automatic methods for reconstructing pipes based on normal tensors and the discriminant analysis method associated with cylinder fitting for determining the geometric attributes of straight pipes and fittings such as Ys, tees, and elbows [27].

Applications using the Hough transform include the use of area-based adaptive Hough transform and circle fitting algorithms to reconstruct pipes in industrial plants [126]. Interesting features of the method include the estimation of both single and multiple cylinder orientation, the classification of pipes that do not follow the standard x, y, and z directions, and the use of intersections between pairs of cylinders to estimate connections [126].

Currently, various commercially available solutions exist for the automatic, or semi-automatic, reconstruction of 3D pipes. Their main advantage is the fact that they eliminate, or significantly reduce, the modeling efforts on the part of the operator [21]. Some of these solutions, although creating geometric primitives for cylinders from point clouds, are not readily available for BIM-based software, as it is the case of Trimble Realworks [111]. Other applications, including Trimble Edgewise and Scan to BIM, can be used as plugins that can be used with BIM-based software [111,115,116].

When dealing with semi-automatic modeling software, the existence of complex configurations of pipes impairs the process, as it is hard for the software to identify and determine the correct radii, lengths, and orientations of the pipes [21]. This is especially an issue in situations when large sets of 3D scans are used, which makes the processing time consuming and labor-intensive.

Limitations related to line-of-sight, total or partial occlusion of elements, and complex pipe configuration, on the other hand, may still require user intervention even in automatic modeling software [126]. In this sense, recent developments in some software have focused on autocompletion algorithms. As an example, EdgeWise (ClearEdge3D) [29] has tools for auto-connecting pipes and even inserting components such as valves, flanges, reducers, and other components into any pipe run, the latter requiring user input. Added to that, the software's quality assurance tools list information about the geometry of each pipe and the fit accuracy of the modeled elements so the user can have more confidence in the results [29].

As it occurs in other areas, it is important to develop metrics to evaluate the performance of newly proposed methods for automating the creation of as-built BIM models [24]. Tang et al. [24] proposed performance measures that can be used to assess the algorithm design, the environmental conditions, and the modeling performance when developing algorithms for modeling as-built BIM models. These measures include, among others, the degree of automation, the requirements for inputs and outputs, the ability of the algorithm to learn, how the algorithm performs in face of occlusion, clutter, and the different reflectance levels found in different materials, and finally how well the algorithms produce accurate data and identify relationships among elements in the scenes [24].

CHAPTER 3: METHODOLOGY

The study followed a quantitative design and aimed to assess the feasibility of using thermal point clouds along with geometrical point clouds in the automatic reconstruction of BIM models of pipes. The thermal point clouds were built with the use of photogrammetric software and were merged with geometrical point clouds from laser scanners. Then, an application was proposed for automatically modeling pipes and automatically classifying the systems these pipes represent using MATLAB and Dynamo for Revit. The measurement of the accuracy and reliability of the proposed application was made through the 18 evaluation measures presented in Tang et al., and the comparison of the outcome of the application with the actual systems in the surveyed building [24].

This section describes the methods used to collect and process data from point clouds from laser scanners and 2D infrared and visible images from thermal cameras. Additionally, it describes the main steps involved in the development of the proposed application, including data requirements and assumptions. Figure 1 presents a graphical description of the steps involved in the process.

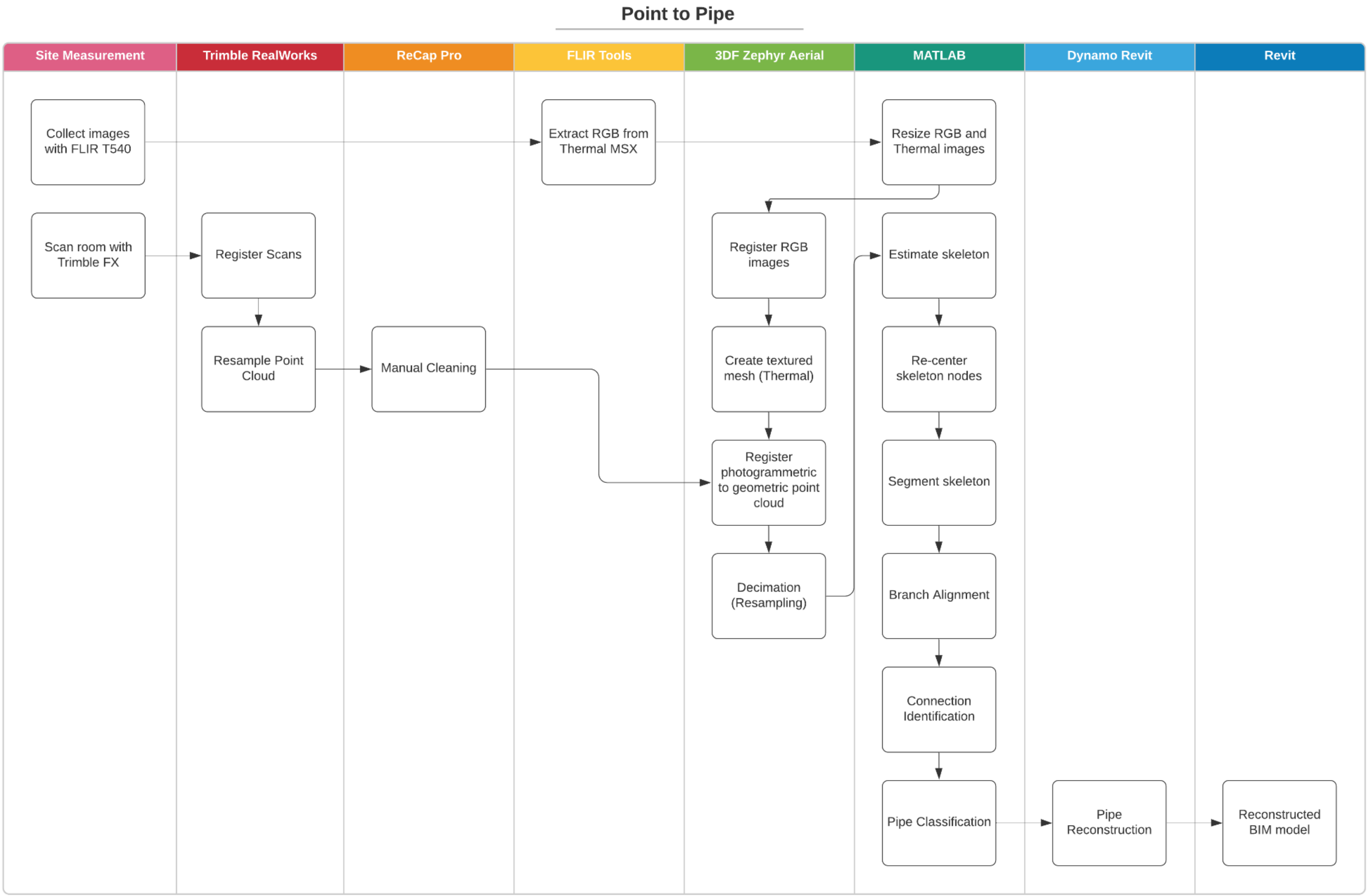


Figure 1 – Methodology.

SITE DESCRIPTION

To test the proposed application, one mechanical room of Illick Hall, at the State University of New York College of Environmental Science and Forestry (SUNY ESF) was used. The mechanical room used has 2912.6 sq. ft of gross floor area and three scenes were produced from this room.

Illick Hall is an academic building located on the north side of the ESF's campus (Figure 2) and it is used primarily for teaching, researching, and service. The building has 140,870 SF of gross floor area at an original cost of $4,957,000 at the time of its dedication on October 11, 1968 [127]. The reason for choosing this building is related to the fact that most of the equipment in place in the mechanical rooms was installed before the energy crisis of 1973, which may make it a candidate for future renovations.

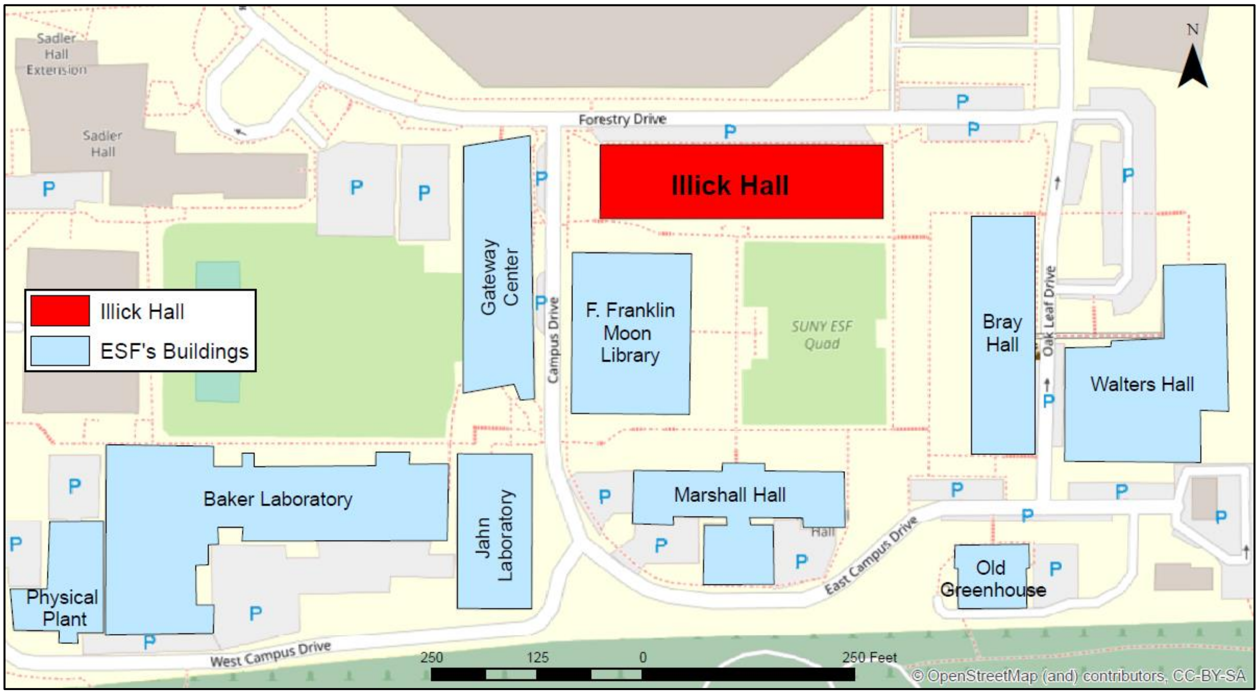


Figure 2 - SUNY ESF's map.

Figure 3 and Figure 4 show the location of the mechanical room within the building and the spaces from which data were collected, respectively. Figure 5 shows pictures of the three spaces within the selected room. In these spaces, the systems that can be found are glycol supply and return, chilled water supply and return, domestic hot water and cold water, condensate, and high, medium, and low-pressure steam.

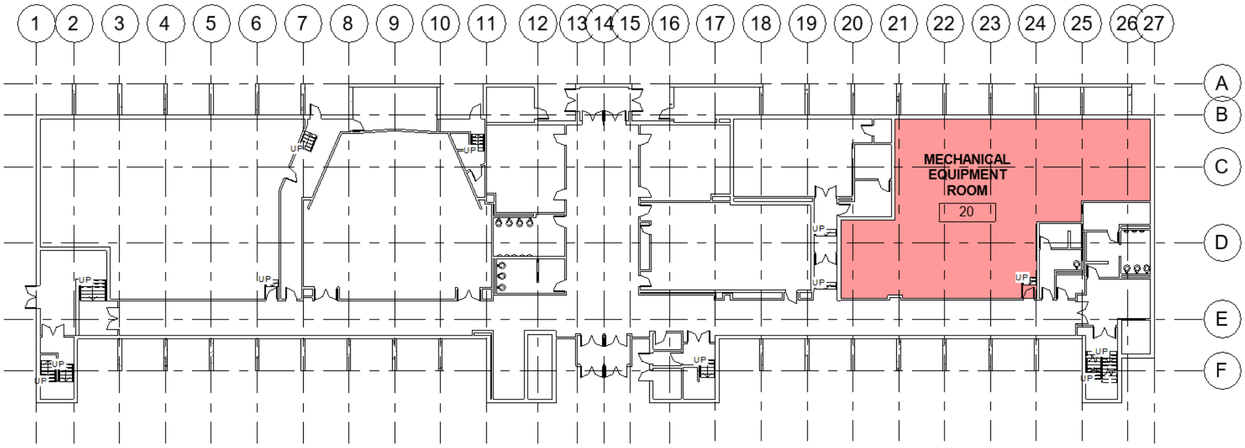


Figure 3 - Illick Hall – Schematic Basement Floor Plan.

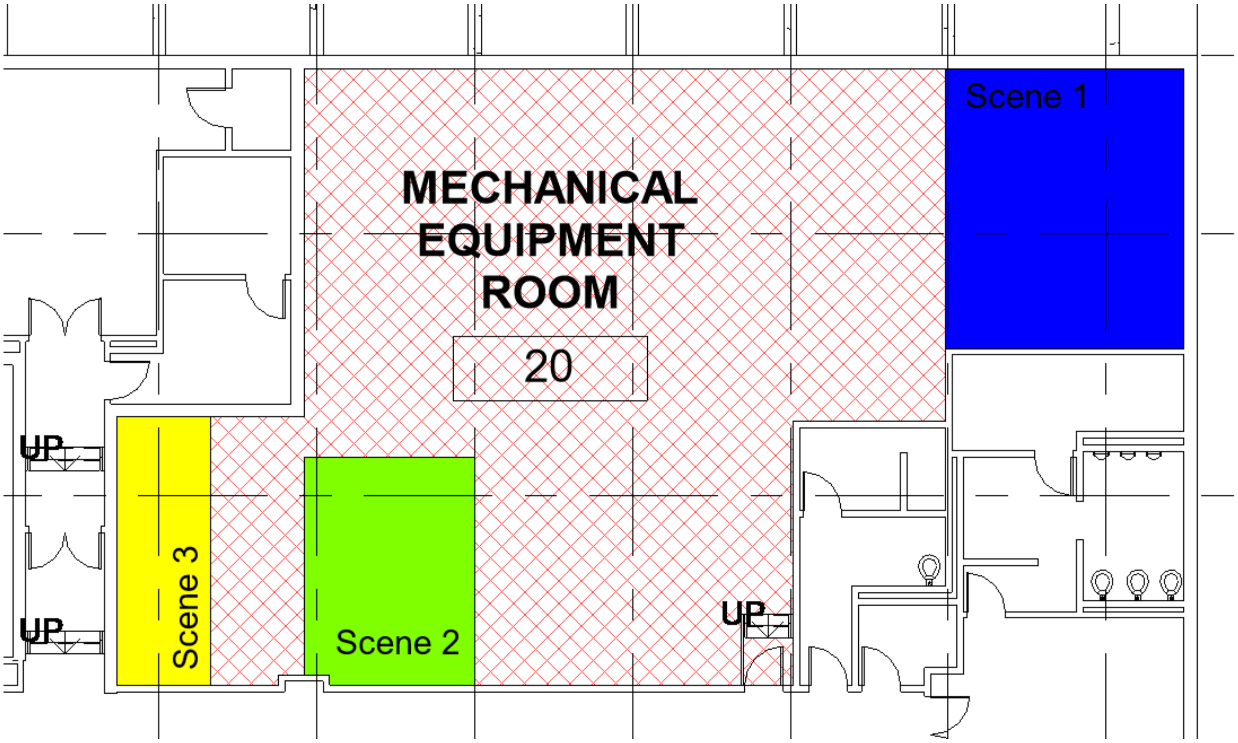


Figure 4 - Location of the three selected scenes in the mechanical room.

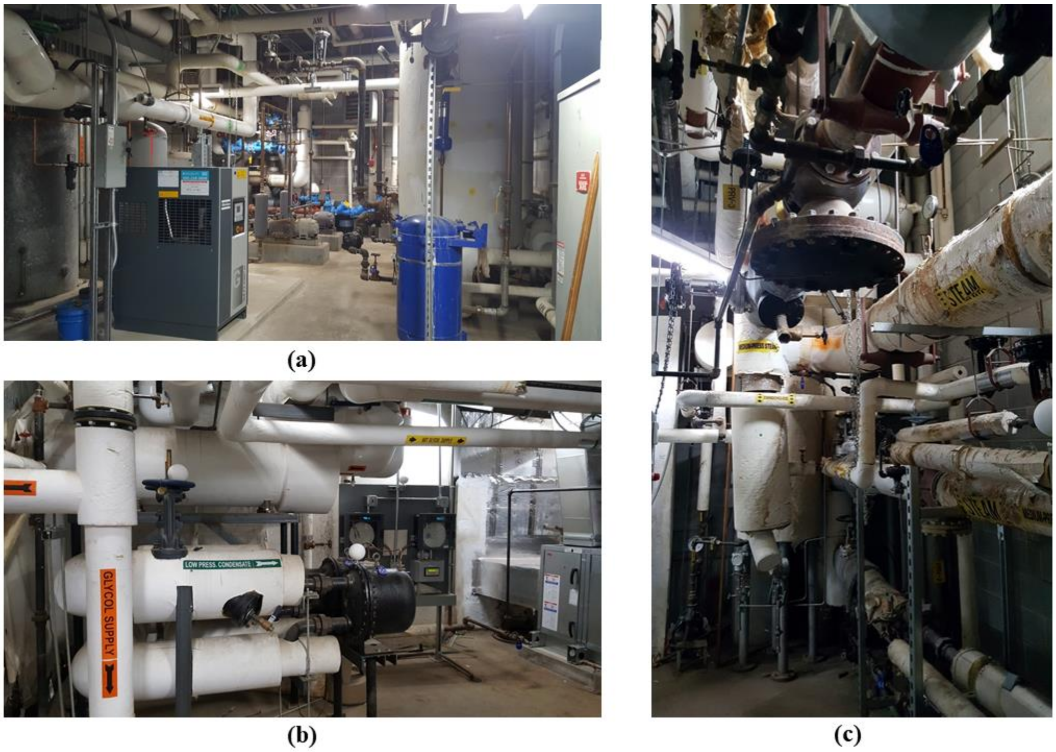


Figure 5 – Perspectives for (a) scene 1; (b) scene 2; (c) scene 3.

MATERIALS

The as-is data related to the three scenes were collected using two different instruments: a Trimble FX laser scanner with no associated sensors, and a FLIR T540, 42º thermal camera with associated RGB camera [128,129].

Laser Scanner – Trimble FX

The Trimble FX (Figure 6) is a laser scanner designed for industrial, shipbuilding and offshore environments that has a relatively high-speed acquisition of data, > 216,000 points per second, and a 360º × 270º field of view [129]. The scanner has a position accuracy of 2 mm at 50 m and a range uncertainty of 1 mm at 15 m and can scan a scene in 5 minutes (single pass), on average.

Different than some of the newer commercially available laser scanners, the Trimble FX scanner has no associated sensors, such as RGB cameras for collecting visible images of the scenes and associating color information to the point cloud, but it can record the received signal intensity for each measurement.

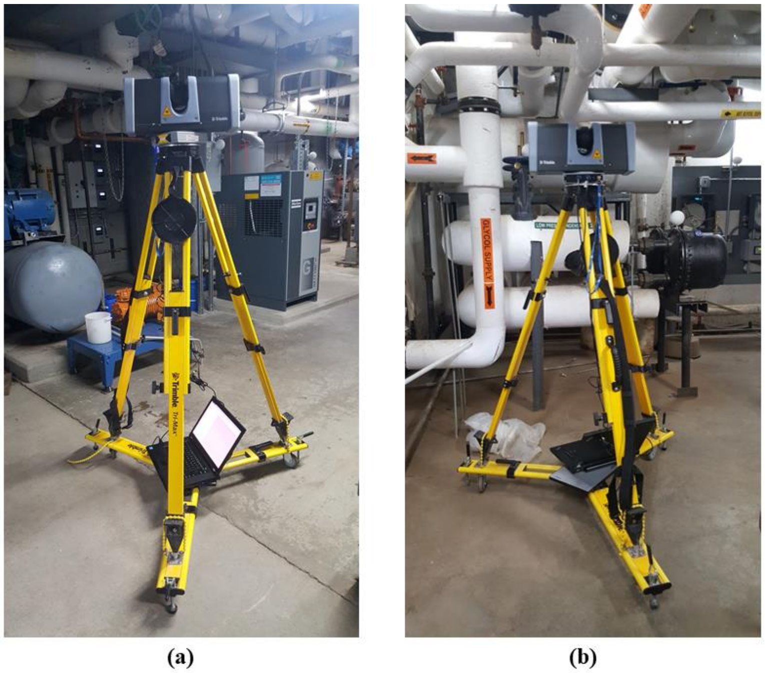


Figure 6 – Trimble FX 3D laser scanner (a) mounted in scene 1; (b) mounted in scene 2

Thermal camera – FLIR T540

A FLIR T540 Professional Thermal Camera (Figure 7), 464 × 348 pixels IR resolution, with a 42º lens was used for collecting thermal information related to the scenes [128]. The camera is suitable for use with objects in the temperature range from -4 ºF to 2732 ºF, separated into three optional ranges: -4 °F to 248 °F; 32 °F to 1202 °F; and 572 °F a 2732 °F. It also has radiometric IR video recording capabilities, and a 1280 × 960 pixels RGB camera that records visible images from the same positions as the thermal images.

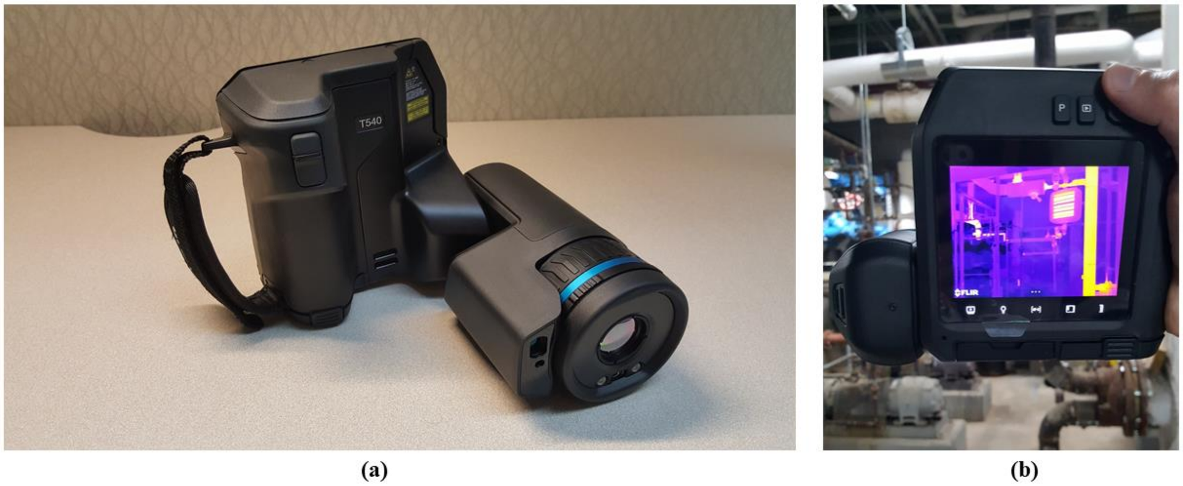


Figure 7 – (a) FLIR T540 professional thermal camera; (b) data collection in scene 1

DATA COLLECTION

Data collection efforts involved two main processes, collecting point clouds from laser scanners, and collecting infrared and visible images from thermal cameras. As the purpose of the final 3D model was to have a single 3D model containing both geometric information from the point cloud from the laser scanner and false-color information from the thermal images, the data collected from the Trimble FX scanner was merged with the data from the FLIR T540 camera.

Data collection was performed on two occasions, February 27th, and February 29th, 2020, during the wintertime in Syracuse, NY. As the data collected refer to enclosed and temperature-controlled environments, the effects of the outdoor low temperatures did not interfere with the operating conditions of the equipment used.

Point Clouds from laser scanners

As presented, a laser scanner was used in the process of collecting geometric information about the scenes. For the selected room, a total of 13 scans were performed in different locations with the help of spherical targets, used for pre-registering the individual scans. To increase the effectiveness of this process, at least three common targets were scanned for each pair of adjacent scans.

Infrared and visible images from thermal cameras

For the case of collecting thermal information about the scenes, the FLIR T540, 42º camera was used. Given that the resulting point cloud from the photogrammetric registration would not be in the correct scale, the geometric point cloud from the Trimble laser scanner was used to scale the photogrammetric point clouds in 3DF Zephyr Aerial.

For facilitating the construction of the thermal point clouds from the infrared thermal maps, some considerations of the photogrammetric practice were assured, and some decisions were made relative to the thermal map output. Regarding the photogrammetric practice, Structure-from-Motion concepts were used, such as the rules of overlapping between different images, consistency between the images, i.e., the scenes were undisturbed, and parallel axis capture [130].

To improve the sharpness of the resulting thermal maps, the FLIR Thermal MSX® (Multi-Spectral Dynamic Imaging) format was used. The advantage of using the Thermal MSX® format is that it adds light visible details to the thermal map (Figure 8). Other decisions involved in taking thermal images were to use a manual thermal scale bar for having consistency between the colors in different images, and the use of the Ironbow Palette, in which light colors show hot objects and dark colors show cool objects.

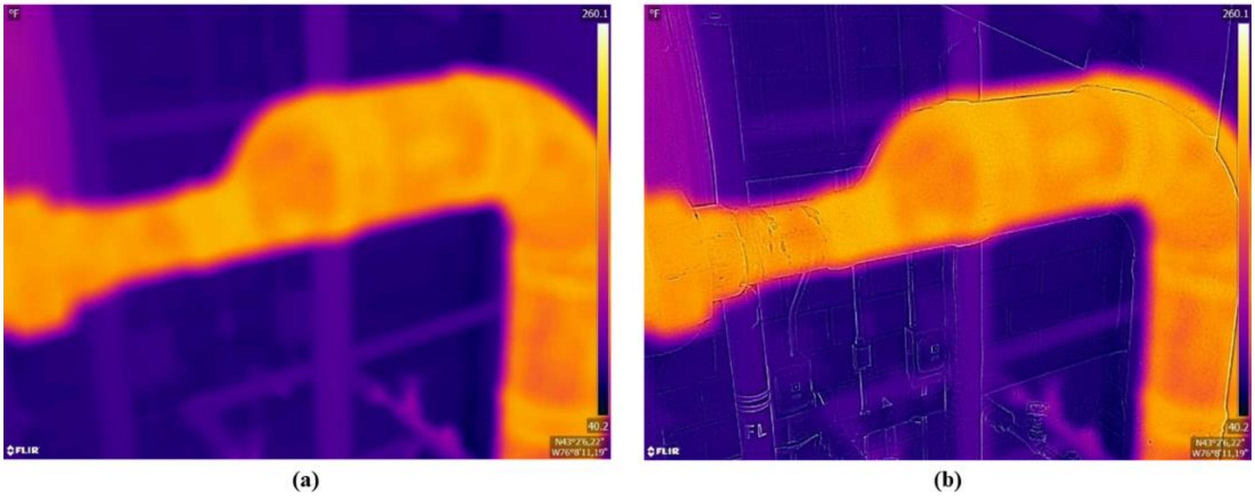


Figure 8 – (a) Thermal image; (b) thermal MSX® image

DATA ANALYSIS AND PREPROCESSING

After generating the point clouds and taking the thermal and visible images for the three scenes, the processing of this data was initiated. For the scans, the first step was to register the scenes using the proprietary software that accompanies each of the scanners. For the processing of thermal images and visible images from the thermal camera, the thermal point clouds were created using third-party software for photogrammetric processing of digital images.

Point Clouds from laser scanners

The generation of point clouds from the laser scanner was a result of the registration process using the proprietary software that accompanies the laser scanner. For the Trimble FX scanner, the software Trimble FX Controller and Trimble RealWorks were used. One such step taken in Trimble RealWorks was to segment the planes that represent walls, floors, and ceilings in the point cloud to facilitate the cleaning of the scene in subsequent steps.

Infrared and Visible Images from Thermal Cameras

The approach used for generating the thermal point clouds was based on the photogrammetric processing of visible images and posterior use of the correspondent thermal images from the FLIR T540 as textures using the software 3DF Zephyr Aerial (Version 4.530, [131]). As the visible images and thermal images presented different resolutions, it was needed to change the resolution of both sets of images before registering them. First, the visible images embedded in the thermal images were extracted from the MSX® images so the two images covered the same areas. Then, a MATLAB R2019a (Version 9.6, [132]) script was developed and used to resize both images. This script is presented in Appendix A.

For the visible images, the set of images was imported into the 3DF Zephyr Aerial workspace, and the New Project tool was used. To produce results with the highest quality, the close-range option with high details was used for both the dense point cloud and mesh. Then, the processed thermal images were used as the textures for the 3D mesh created using the visible images. Next, the texturized (thermal) mesh was converted to a dense point cloud.

The resulting thermal point cloud depicts each point with a false-color that represents the average temperature of the point from all thermal images that contain it [79]. Although these false colors are used to indicate the temperatures in the model, there is still the limitation that the emissivity differences for different materials in the scenes were not corrected [79]. For all images, the emissivity value of 0.90 was used because it is close to most of the materials' emissivities most likely to be found in the scenes.

Point cloud merging

After registration, the scene was scaled using the Registration tools. First, the geometric point cloud from the laser scanner was imported into the same model as the photogrammetric point cloud. Then, for each point cloud, three control points were selected. The three control points were in correspondent locations in both point clouds to allow for the alignment of one point cloud to the other. After that, the photogrammetric point cloud was aligned to the geometric point cloud using the Control Points registration tool having the geometric point cloud as the reference. Next, a fine alignment was performed using the Multi-ICP tool in 3DF Zephyr Aerial. In this process, the photogrammetric point cloud was aligned to the geometric point cloud to minimize the distances between both point clouds through an iterative process.

At the end of this process, both the visible and the thermal point clouds from the photogrammetric process were aligned to the geometric point cloud from the laser scanner. The final step was to project the thermal, false, colors of the thermal photogrammetric point cloud onto the geometric point cloud from the laser scanner. To accomplish this, the geometric point cloud was structured in the 3DF Zephyr tool, a process that projected the colors of the thermal images onto the geometric point cloud.

Point Cloud Preprocessing

In its current state, the proposed application is still not able to differentiate the points in the point clouds that belong to the pipes from the points that do not belong. Thus, it is still needed to clean up the scene, so it only shows the points that belong to the pipes. To accomplish this, the software Autodesk Recap Pro (Version 5.0.1.30, [133]) was used. After manually cleaning up the scene in the geometric point cloud, the set of points representing the pipes only were exported as .pts files and imported into MATLAB R2019a (Version 9.6, [132]). It is important to consider that the use of the software Trimble RealWorks for identifying and segmenting plans such as walls, floors, and ceiling in the point cloud considerably reduced the efforts for creating a point cloud containing only points that represent pipe geometries.

Following the cleanup and file import, the reconstruction process started. In its current the state, all the analysis in the application were performed in MATLAB and the geometry reconstruction, linking the outputs of the MATLAB and the final BIM model in Autodesk Revit 2019 (Version 19.2.20.24, [134]), was done in Dynamo (Version 2.0.2.6833, [135]).

DATA PROCESSING IN THE PROPOSED APPLICATION

Skeletonization

As previously mentioned, the proposed application is based on the estimation of the skeleton of the pipes using an adaptation of the L_1-median locally to a point cloud that results in a *one-dimensional structure* that estimates the center of the 3D objects in the scene as described in Huang et al. [136]. For the proposed application, the algorithm described in Huang et al. [136] was implemented in MATLAB with the appropriate adaptions for handling pipes, and incorporating the same capabilities, including conditional regularization, skeleton estimation via iterative contraction, density-based weighting, and re-centering.

Given a set of points in the 3D space $Q = \{q_j\}_{j \in J} \subset \mathbb{R}^3$, the definition presented in eq. (1) for L1-medial skeletons results in a set of projected points $X = \{x_i\}_{i \in I}$ that approximate the skeleton of the objects in Q [136]:

$$\underset{X}{\operatorname{argmin}} \sum_{i\in I}\sum_{j\in J}\left\|x_i - q_j\right\|\,\theta\left(\left\|x_i - q_j\right\|\right) + R(X) \tag{1}$$

Where the first term defines the local L$_1$-medians of Q weighted by the fast decaying function $\theta(r) = e^{-r^2/\left(\frac{h}{2}\right)^2}$, in which h is the radius of a sphere that contains the neighboring points that support the construction of the skeleton. Additionally, the term $R(X)$ is used to control the local point distribution of X by avoiding point clusters. The indices i and j represent the skeletal points in X and the original point set in Q, respectively [136].

The use of the term $R(X)$ in eq. (1) adds a repulsion force that minimizes the accumulation of skeletal points into clusters in regions of higher point density in the original set Q [136]. $R(X)$ is defined in eq. (2).

$$R(X) = \sum_{i\in I}\gamma_i \sum_{i'\in I\setminus\{i\}} \frac{\theta\left(\left\|x_i - x_{i'}\right\|\right)}{\sigma_i\left\|x_i - x_{i'}\right\|} \tag{2}$$

Where $\{\gamma_i\}_{i\in I}$ are constants, and σ_i is the directionality degree of x_i defined in eq. (3) [136].

$$\sigma_i = \sigma(x_i) = \frac{\lambda_i^2}{\lambda_i^0 + \lambda_i^1 + \lambda_i^2} \tag{3}$$

Where $\lambda_i^0 \leq \lambda_i^1 \leq \lambda_i^2$ are the eigenvalues of a 3 x 3 weighted covariance matrix at each skeletal point in X, as defined in eq. (4). These eigenvalues have associated eigenvectors $\{\mathrm{v}_i^0, \mathrm{v}_i^1, \mathrm{v}_i^2\}$ that define an orthonormal frame that describes the principal components of the skeletal points [136].

$$C_i = \sum_{i'\in I\setminus\{i\}} \theta\left(\left\|x_i - x_{i'}\right\|\right)\left(x_i - x_{i'}\right)^{\mathrm{T}}\left(x_i - x_{i'}\right) \tag{4}$$

When eq. (1) is set to zero, it is possible to set the ratio μ that defines the strength of the conditional repulsion shown in eq. (2) [136]. The authors present that μ should be selected from the interval $[0, 1/2\ )$ to control the penalty applied to the points [136]. For the proposed application presented in this project, $\mu = 0.35$ was used, as in the original paper.

The application developed in Huang et al. [136] uses eq. (1) iteratively to approximate the L_1-medial skeleton of the 3D objects in the scenes. Some of the interesting features of the proposed application are the fact that it is robust to the presence of outliers and it includes a density-based weighting step that alleviates the creation of skeletal point clusters in regions of high point density in the original point cloud [136].

Among the main differences between the skeletonization application developed in Huang et al. and the one algorithm in this project is the fact that while the former requires a structured mesh file, a Polygon File Format - PLY, the latter uses the original set of points that describe the point cloud. The main result of such a difference is the fact that the processing times for estimation in the software developed by Huang et al. are relatively faster than the processing times of the proposed application. Also, the proposed application uses a circle as the fitting geometry instead of an ellipse during the re-centering step. Appendix B presents the method for estimating the skeletons of the pipes in the proposed application.

Skeleton nodes Re-centering

The L1-medial skeleton generated during skeletonization places each skeleton node in the median location of its nearby points. When partial occlusion occurs, the weight of the existing points in the geometric point cloud brings the skeleton node closer to these points, which compromises the ability to estimate the center of the pipe. To solve this issue, a re-centering step is performed.

For the proposed application, instead of using an ellipse as an approximation, as it is the case in Huang et al. [136], a circle was used as the approximation. For each point in the skeleton, a plane passing through the point, and with a normal that is aligned to the direction of the pipe the point belongs, is created. In doing so, directions of variance and Principal Component Analysis (PCA) were used. Then, the 100 nearest points to the skeleton node are projected onto the plane and a circle is used to approximate the distribution of the points in the plane. The proposed circle tries to minimize the distances between the points location and the circumference that describes the circle in the plane. Thus, the new center of the pipe is defined by the center of the circle. Appendix C presents the re-centering algorithm.

Segmentation and Node Classification

For the proposed application, the segmentation of the point cloud was based on the segmentation of the estimated skeleton. The purpose was to identify skeleton nodes that were candidates to belonging to a straight pipe segment using the number of connections each node makes when the angle threshold is observed.

Each node in the estimated skeleton was tested against its two nearest neighbors. For each node, the two nearest neighbors were identified and the angle between the vectors formed between the node of interest and its two nearest neighbors was computed. If the absolute value of the $\sin(\alpha)$, where α is the angle between the two vectors, was smaller than the set threshold, $\sin(45°)$, then the three points were set to “connected” in the auxiliary matrix NxN, where N is the number of nodes in the skeleton.

After testing the connectivity of all the nodes in the skeleton and setting the connectivity to be either true or false in the NxN matrix, the number of connections each node makes was computed. This step served the purpose of classifying each node either as belonging to a straight segment or not. If any node made four connections with its neighbors based on the defined rules, the node was considered to belong to a straight segment. If, on the other hand, the node made more than four connections, it was considered as a candidate to a tee connection. Finally, if a node made less than four connections, it was be either considered as a candidate to a connection, elbow or coupling, or was considered as not belonging to the pipes, i.e., noise.

The last step in the segmentation algorithm was to classify the sets of points that represent each straight segment. To accomplish this, the sets of neighboring nodes making four connections between nodes making more or less than four connections received a unique number that represents a unique straight pipe segment. It is also important to mention that, for each unique straight pipe segment, the two nodes that represent the extremities of the segments were identified. These points were crucial to the next steps of the application, where they were used to identify the connections between groups of straight pipes. Appendix D presents the method for segmenting the skeletons.

Parameter Estimation

Parameter estimation can be divided into two main groups, namely diameter estimation, and location descriptors estimation. The first group represents a first estimate of the diameter of the pipes in each straight segment while the second aims to identify the mean values and standard deviations values for the x, y, and z coordinates of each set of nodes in each unique straight segment. At this point, it is important to state that in the current state of the application, only pipes that run parallel to the x, y, and z axes can be correctly estimated.

Diameter Initial Estimate

To determine the initial diameter of the pipes, the distance from each skeleton node in a straight pipe segment was measured during the re-centering step for each re-centered point.

Node and Segment Alignment

Node and segment alignment are two steps that were intended to assess whether the candidates to connections (tees, elbows, and couplings) were connections in the original scene. First, for each unique segment, identified during node classification, the mean x, y, and z coordinates were computed, along with the standard deviation for each mean. If the pipes run parallel to the x, y, or z-axis, then the direction with the highest standard deviation should describe the direction in which the straight segment runs.

The other two directions, on the other hand, describe the mean values of the central axis of each pipe segment. This fact was used to align all the nodes in each straight segment by replacing the values of the two coordinates in each node by the mean values computed for each straight segment. After aligning the nodes within each straight segment, a second alignment was performed, but now with the intent of aligning straight segments to make them either colinear or coplanar whenever their parameters (mean coordinates and standard deviation values) were relatively close and their radius intersect.

If the mean coordinate values for a set of points in any of the aligned straight segment was within a distance of half of its radius to the mean coordinate values of another straight segment, and these mean values did not correspond to the direction with the highest standard deviation for any of the segments, then the segments were considered to be aligned. In order to introduce the minimum error to the location for the straight pipes in this step, the new individual values for the intersecting direction for all the segments in the group were set equal to the mean value of the coordinates' direction for each straight segment in the group.

Connection Identification

The connection identification step used the approach of recognition using context, presented in Tang et al. [24], in which objects more easily identified, in this case, straight segments, were used to facilitate the recognition of more challenging structures, such as tees, elbows, and couplings.

Once the straight segments were either colinear or coplanar, whenever it was the case, the nodes in the skeleton that were not included in any straight segment and that make more than four connections in the NxN matrix were tested as being tee connection nodes. For each candidate, the three nearest extremities from three different straight segments were selected.

Next, each of the candidates was tested. The first test was intended to identify duplicates, i.e., nodes making more than four connections in the NxN matrix that had the same three nearest extremities. Second, the candidates received the mean values of the coordinates of the three segments that connect to the tee and that are different than the direction with the highest standard deviation for each segment. Finally, the triple product of the three vectors formed from the node that was a candidate to the tee connection to each of the three nearest extremities was calculated.

In a tee connection, the three segments are coplanar and, consequently, the triple product of these three vectors is zero. If that was not the case, the node was a false tee candidate.

After testing for the tee connections, the next step was to identify the elbows and couplings. To do so, each extremity of a straight segment not involved in a tee connection was paired to its closest extremity so that each extremity had only one connection. The elbows/couplings coordinates, then, were defined as the ones with the smallest standard deviations when considering the mean values and standard deviations of the segments from which the extremities were candidates to elbows/couplings connections.

Once the candidates were identified and their coordinates were set, the first test was to identify duplicates, in this case, connections involving two pairs of points from the same two straight segments. After that, the algorithm identified false connections from two types, the ones not making a 90º or 180º angle, and the ones that were located too far from the nodes in the skeleton making less than four connections in the NxN matrix, i.e., the candidates to elbows and couplings.

For the first group, two vectors were created, each one from the connection candidate to the two extremities in the straight segments. If the angle between the two vectors was different than 90º (elbows) or 180º (couplings), then the candidate was not a true connection. For the second group, all the skeleton nodes connected to the extremities involved in each possible connection were identified. Next, a threshold equals the sum of the diameters of the two straight segments was set. If the candidate to elbows/coupling was more distant than the threshold from any node with less than four connections in the NxN matrix that were connected to the segments being considered, then the candidate was not a true connection. The connection identification algorithm is presented in Appendix E.

System Classification

System classification was based on the false-color information (thermal colors) for each pipe segment in the thermal point cloud. The first step consisted of determining the temperature ranges for each system in the scene by reading these values on the camera screen. These temperature ranges were assigned to classify the pipes into their systems.

For each point in the skeleton, the average temperature of the 50 closest points in the thermal point cloud was computed and assigned to the point. The conversion of the qualitative information (color) to a quantitative value was done with the help of the auxiliary matrix A. This matrix represents the Ironbow Palette as a set of 120 RGB values. For the temperature range of each scene, each of these 120 values received a temperature value, which was used for determining the average temperature of the 50 closest points to each node in the skeleton. For the colors that do not have a specified temperature value in matrix A, interpolation was used to determine the temperature of the point in the thermal point cloud.

Two important considerations need to be made before classifying the pipes: first, it is important that the fluids inside the pipes in the scene are moving, i.e., that the systems are being used; second, the presence of insulation may make more difficult, or even prohibit, the pipe classification because the actual temperatures of the pipes may not be captured. To solve these two issues, data collection was performed during college hours and the classification of the pipes was made using the segments of the pipes that had no insulation, either because of degradation of the existing insulation or in segments that had no insulation. Thus, to classify the pipes, the system would identify portions of the segments created in the segmentation step that presented a temperature range relatively different than the reference temperature, the value most likely to be found on the surfaces of the insulation. The system classification script is shown in Appendix F.

Pipe Reconstruction in BIM

As MATLAB is not a BIM-based software, it was needed to identify a connection between the output from MATLAB and the input in Revit to solve for interoperability issues. For the proposed application, the chosen connection was Dynamo Revit as it enables the performing of repetitive tasks in Revit. Additionally, as the built-in library in Dynamo has no specific piping tools, the package MEPover (Version 2019.12.1, [137]) was used to facilitate the reconstruction process.

The MEPover package contains custom nodes that can be used to create the routines Dynamo is expected to run and focuses mainly on the MEP side of Revit. These custom nodes, along with Dynamo's library nodes, were used to create the routines to be run to accomplish the two objectives of this research: create the geometry of the pipes and classify them accordingly to the systems they represent.

The inputs for the tools used in the process required the lines that represented the pipes, the diameters of each pipe, and the relationships between sets of pipes to create the connections. Thus, from MATLAB a set of .csv files was exported containing: i) the coordinates of the extremities of each straight segment, which was used to create the lines; ii) the diameters of each straight pipe segment; iii) the relationships between three straight segments forming a tee connection; iv) the relationships between two straight segments forming either an elbow or a coupling; v) and finally the classification of each of the straight pipe segments relative to the mechanical systems they represent.

After running the routine in Dynamo, the geometries of the pipe were reconstructed in Revit. It is important to mention that, other than tees, elbows, and couplings, the software inserts all the necessary reducers whenever the section of the pipes change in a connection. Appendix G presents the MATLAB code used to create the .csv files that were used in the Dynamo routine, and Appendix H presents the Dynamo's logic routine developed to reconstruct the pipes in Revit.

Additionally, it was needed to prepare the template for the Revit file using the systems identified in the scenes. Using the U.S. Metric - Plumbing Default Metric template from the Revit library, the original piping systems families in the template were duplicated and modified to match the names of the systems in the MATLAB application. Finally, for visualization purposes, filters were applied to the piping systems in the 3D view.

PERFORMANCE MEASURES

The set of 18 evaluation measures presented in Tang et al. was used to qualitatively assess the performance of the proposed application [24]. These performance measures are divided into three main groups, namely algorithm performance, environmental conditions, and algorithm performance for a given set of environmental conditions. For this project, these measures aim to assess the performance of the proposed application for modeling as-built BIM models considering the information requirements of the AECOO industry.

LIMITATIONS

In its current state, the application presents some limitations in aspects related to the computational performance of the proposed algorithm and its ability to perform certain tasks that would improve its overall capability. First, the application cannot separate the points that belong to the pipes from points in the remaining objects in the point cloud. This is a current focus, and a greater challenge, of many research projects and, given its complexity and time and resources requirements, it would not be feasible to develop such capability for the current application.

Second, the application can only handle pipes the run parallel to the x, y, and z axes. Pipes that run in other directions cannot be handled due to the need of creating coordinate systems relative to each pipe direction, which was not considered in this current algorithm but can be explored in adaptations of it.

Finally, there is one final limitation in the merging of thermal point clouds and geometric point clouds. Given the need for renting the laser scanner and professional thermal camera, it was not possible to develop an automatic merging framework using the hardware. This step would require calibrating both sensors to work simultaneously and developing computer applications for correcting the distortions created by different sensors. Given the high costs associated with renting both pieces of equipment for the required times for developing such a framework, it was decided to manually align the resulting point clouds in 3DF Zephyr Aerial.

CHAPTER 4: FIELD EXPERIMENTS AND RESULTS

The following sections present the results of the development of the application for automatically reconstructing pipes using point clouds from laser scanners and digital images. All the processing was done using an Alienware 17 R4 with an Intel® Core™ i7-7700HQ CPU @ 2.80GHz with 16.0 GB RAM.

GEOMETRIC POINT CLOUD PREPROCESSING

When using the Trimble FX laser scanner, 13 scans were performed in the entire mechanical room and, after registration, a single geometric point cloud was produced using Trimble RealWorks. Figure 9 presents some perspectives of the mechanical room in Trimble Scan Explorer Viewer.

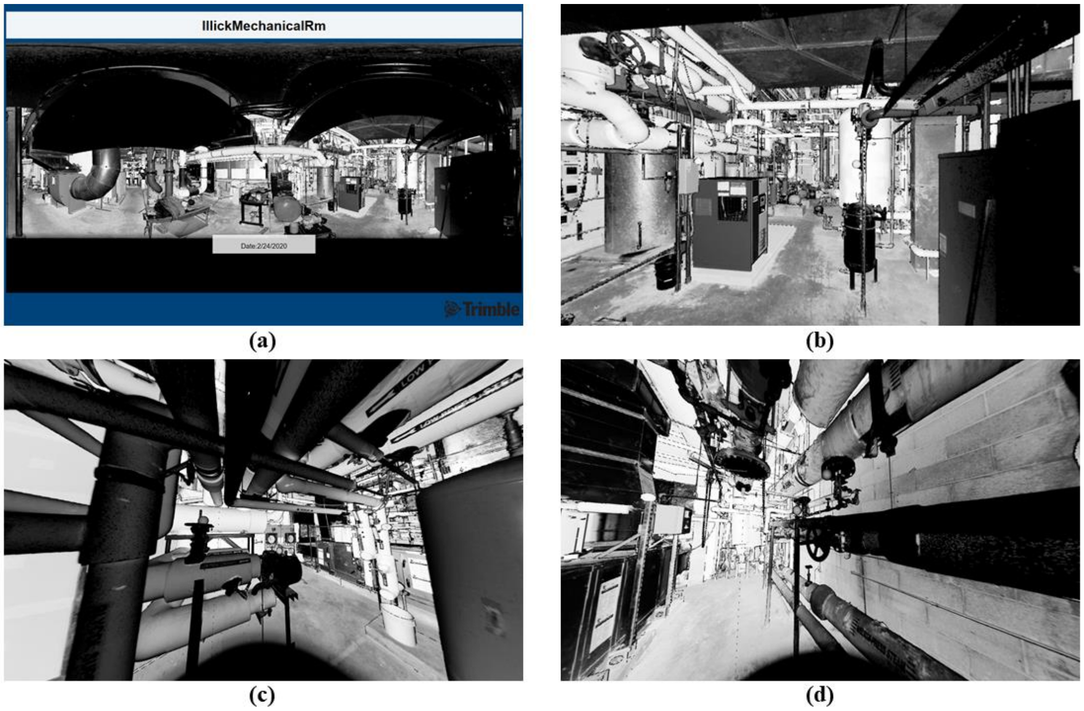


Figure 9 – (a) Trimble Scan Explorer Viewer; Perspectives from (b) scene 1; (c) scene 2; (d) scene 3.

To facilitate the cleaning of the scenes, this single point cloud was sampled to reduce the size of the file and automatically separated into two sets of points in Trimble RealWorks: one containing sets of points that represent planes (wall, floors, and roofs) and other containing the remaining points (equipment, pipes, ducts, conduits, and clutter). This second point cloud was further cleaned, this time manually, to let only points representing pipes in each of the three scenes. This manual cleaning was done using Autodesk ReCap Pro. Figure 10 presents the resulting point clouds for each of the scenes in Autodesk ReCap Pro. Table 1 present the number of points in each of the three scenes after manual cleaning.

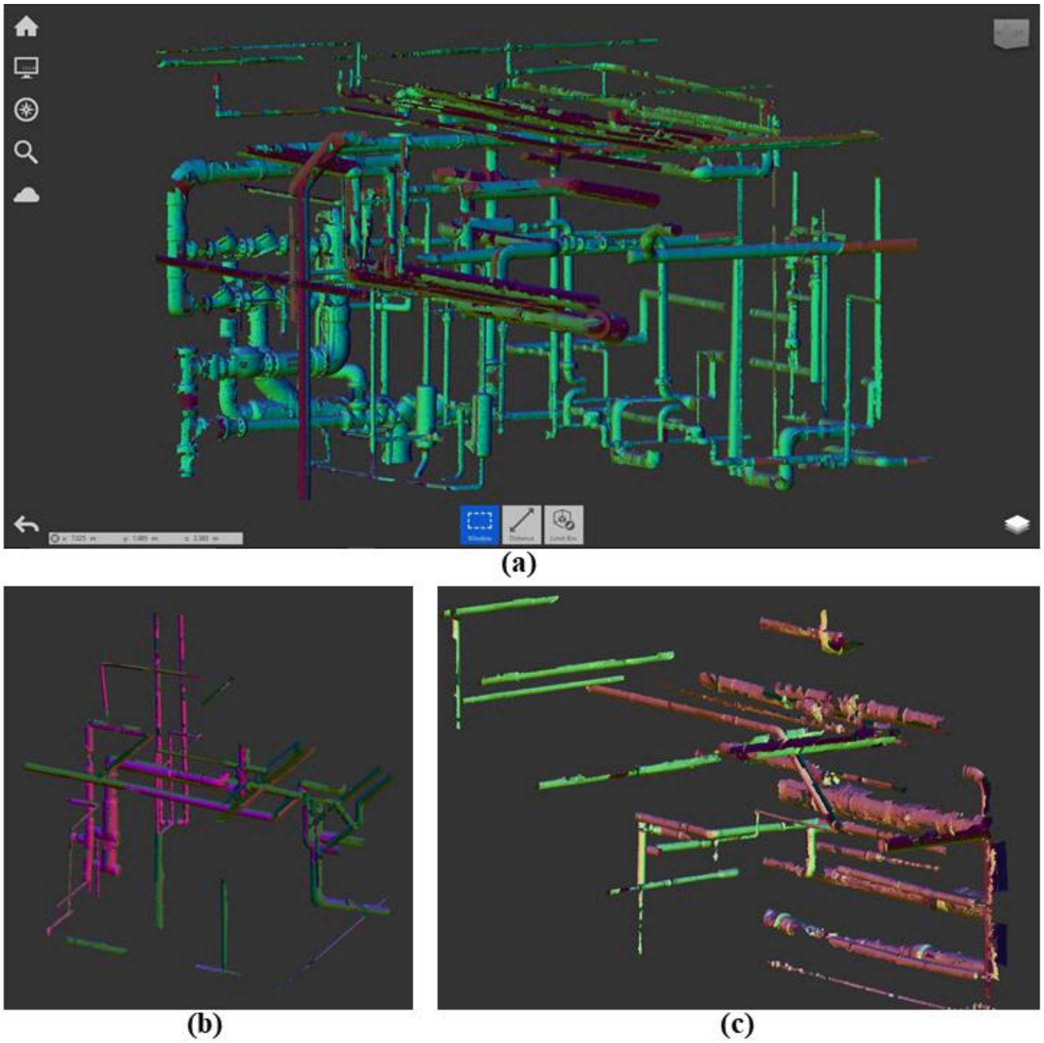


Figure 10 – Cleaned geometric point clouds in Autodesk ReCap Pro from (a) scene 1; (b) scene 2; (c) scene 3

Table 1 – Number of points in point cloud after manual cleaning in ReCap Pro.

Scene	1	2	3
Number of points in Point Cloud after cleaning	27,583,866	5,556,956	7,693,616

THERMAL POINT CLOUD RECONSTRUCTION

As presented, the thermal point cloud reconstruction was done in 3DF Zephyr Aerial by using the sets of visible and thermal images collected with the FLIR T540 camera. Because the resolutions of the visible and thermal camera were different, 1280 × 640 pixels vs. 464 × 348 pixels, respectively, the projections of the thermal images onto the mesh created from the visible images would not match. To solve this issue, the visible images were extracted from the FLIR Thermal MSX® images using the software FLIR tools. The results of the extraction process are shown in Figure 11.

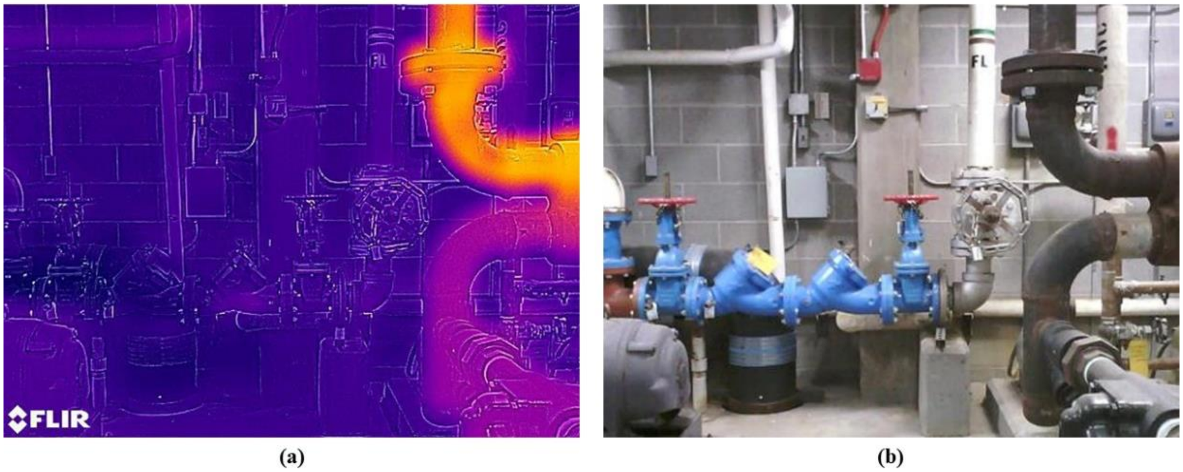


Figure 11 – (a) Thermal image and (b) associated extracted visible image from scene 1

After that, the contents of the images were the same, but with the visible images having a smaller resolution than the thermal images. Thus, before processing the chunk of images in 3DF Zephyr Aerial, both sets of images, RGB and thermal, were resized in MATLAB so they would match when projecting the thermal images onto the mesh produced by using the RGB images.

Table 2 presents the outdoor and indoor temperatures during data collection, and the manual temperature scale used in the camera. Additionally, because the equipment and pipes were losing heat to the environment, a reference temperature was determined using the FLIR camera. This reference temperature was defined as the average temperature of the surfaces in each scene and was used in the pipe classification step.

Table 2 – Room temperature, reference temperature, temperature ranges, and the number of captured images per scene.

Scene	1	2	3
Room Temperature (°F)	70	70	70
Outdoors Temperature (°F)	20	22	19
Reference Temperature (°F)	80	90	100
Temperature range of the images (°F)	40-260	70-240	70-360

Visible Camera Image Registration

Table 3 presents the number of images used to produce each scene, the set of these images which were registered in 3DF Zephyr Aerial, the number of points in the resulting dense point cloud, and the registration times for each scene.

Table 3 – Captured images, registered images, number of points in registered point clouds, and registration times.

Scene	1	2	3
Number of Images	1081	475	347
Registered Images	643	90	129
Number of points in Point Cloud	6,816,051	1,038,156	1,951,381
Registration time (min)	142	25	26

As presented in Table 3, not all visible images used to produce the point clouds were registered in the process. This means that only a set of images were processed in 3DF Zephyr Aerial to produce each of the point clouds. The reason for a relatively large number of images not being registered has many causes. First, by cropping the visible images to match the size of the thermal images in the FLIR Tools software, part of the overlap between adjacent images was lost, which reduced the ability of the software to register some of the images. Second, the size and complexity of each of the scenes may have posed a challenge for the registration process because the size and complexity of complex plant environments is a challenge for the automation of scene description by using image data [23]. Other possible reasons may include the sharpness of the visible images due to the not so high resolution of the visible sensor in the FLIR T540 camera, and the lack parallax in certain sets of images, especially in regions with complex configurations of pipes, which limited the ability to capture images from multiple angles.

Figure 12 presents the registered photogrammetric point clouds in 3DF Zephyr Aerial. As presented, the parameters used in the registration process were set to maximum quality and density, which resulted in relatively complete scenes. The gaps observed in parts of the scenes were a result of the issues that may have occurred during registration, being the areas most affected the areas close to the ceilings because the complexity of pipe configuration in these areas did not allow enough overlap.

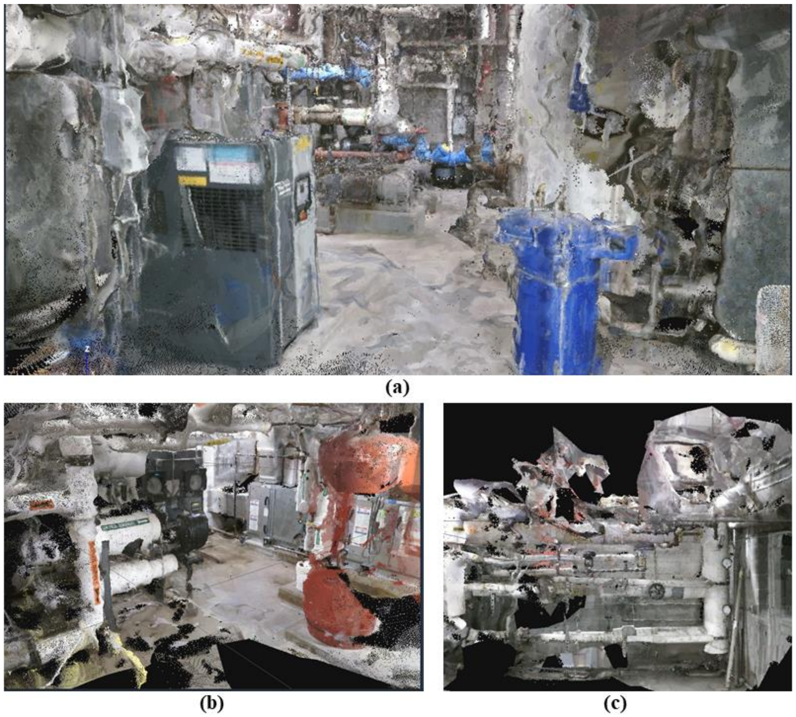


Figure 12 - Photogrammetric point clouds from (a) scene 1; (b) scene 2; scene 3

Thermal Image Projection

After creating the point clouds and meshes from the visible cameras, the thermal images associated with each of the registered visible images were used to produce a textured mesh that was then converted into a mesh, and finally into a dense point cloud that was used for system classification purposes. As it was the case for the visible images, all parameters of the mesh creation process used a high-quality level. Figure 13 presents the final thermal point clouds for scenes 1 to 3. Table 4 presents the processing times for creating the thermal point clouds.

For the thermal point cloud, the thermal images were not directly registered, as it was the case for visible images, but rather, they were projected onto the point clouds from the angles and positions of their associated visible image. The results present relatively low contrast among the colors in the scenes because the temperature scale of the thermal images was manually set to cover the entire temperature range of the scene, thus not calibrating the temperature range relative to each image, as it is done when the automatic scale is used.

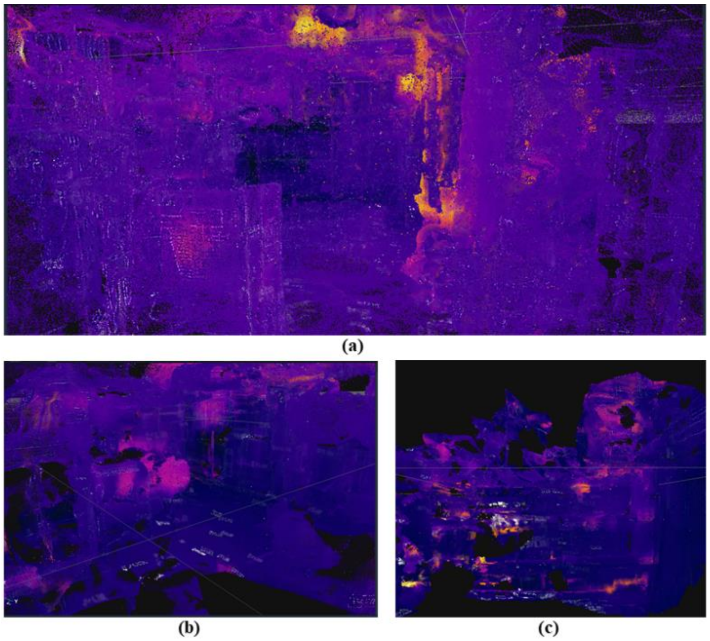


Figure 13 - Thermal point clouds from (a) scene 1; (b) scene 2; (c) scene 3

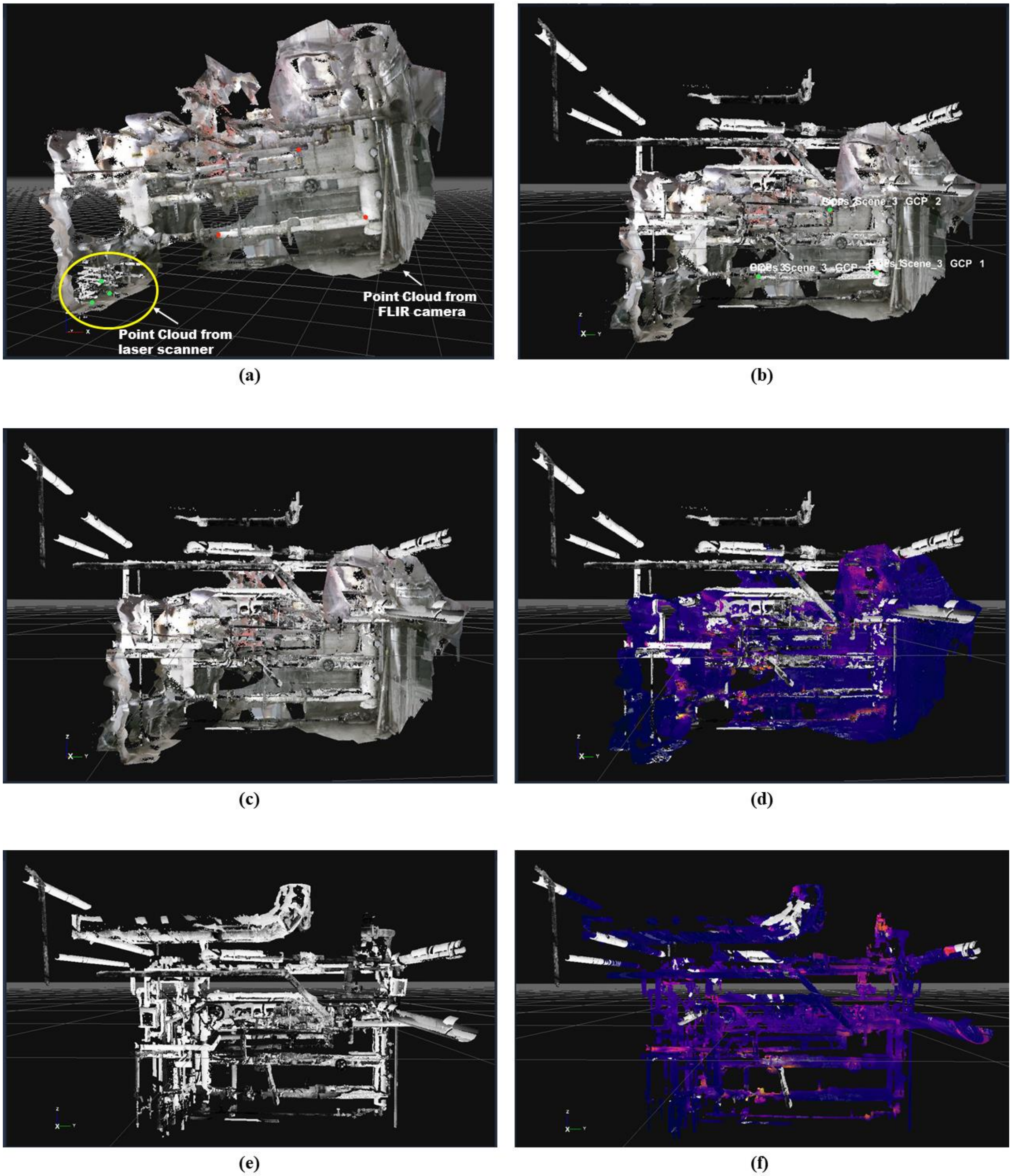
Table 4 – Textured mesh creation time and number of points in the resulting point cloud.

Scene	1	2	3
Textured mesh creation time (min)	27	3	6
Number of points in point cloud	1,999,998	1,038,155	1,951,337

POINT CLOUD MERGING

After both the geometric and thermal point clouds were produced, the geometric point cloud from the Trimble laser scanner was imported into 3DF Zephyr Aerial software so the thermal point cloud could be aligned with it. Because no control points were used in the photogrammetric application, the resulting point cloud was created in a relative coordinate system with an unknown scale. For scaling and aligning the thermal point cloud to the geometric point cloud, the registration tools were used in 3DF Zephyr Aerial software.

Figure 14 presents all the steps involved in the merging process for scene 3. First, (a) three control points were defined for both the geometric point cloud and the photogrammetric point cloud. These control points were located manually using three distinct points that could be identified in both point clouds. Second, (b) a coarse alignment was performed using the Control Points registration tools using the geometric point cloud as the reference. Next, a fine alignment was performed using the Multi-ICP tool in 3DF Zephyr Aerial. In this process, the photogrammetric point cloud was aligned to the geometric point cloud to minimize the distances between both point clouds through an iterative process. As a result, the photogrammetric point cloud and all models associated with it (thermal point cloud and meshes) were aligned, as shown in (c) visible image point cloud and (d) thermal point cloud. Finally, the geometric point cloud (e) was structured to use the projections of the thermal images as the colors of its points (f). In (f), the points in white did not find a proper projection so they did not receive any color. Figure 15 and Figure 16 present the results of scenes 1 and 2, respectively.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 14 - Point cloud merging for scene 3. (a) point clouds from the laser scanner and photogrammetry (unscaled); (b) control points registration results; (c) Multi-ICP alignment results showing visible point cloud; (d) Multi-ICP alignment results showing thermal point cloud; (e) point cloud from the laser scanner; (f) point cloud from the laser scanner with projected thermal, false, colors

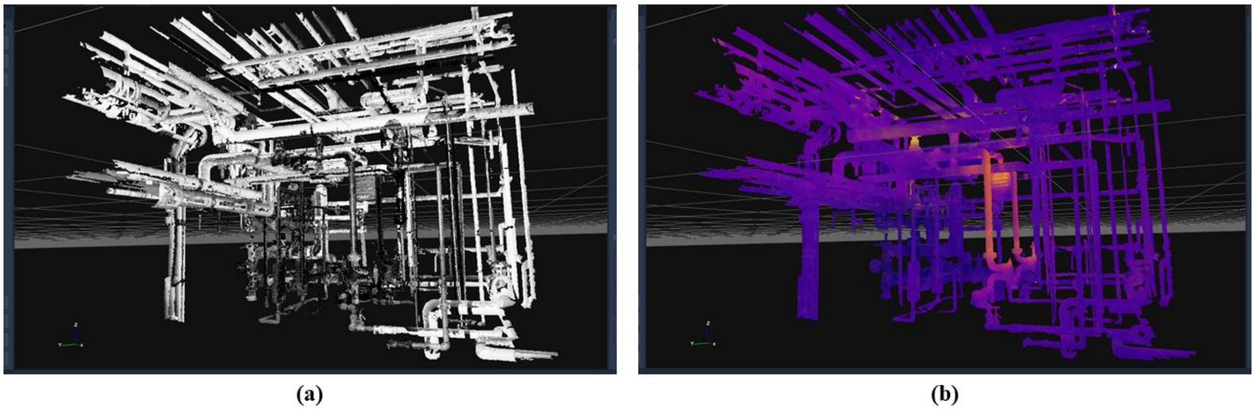

(a) (b)

Figure 15 - Point cloud merging for scene 1. (a) point cloud from the laser scanner; (b) point cloud from the laser scanner with projected thermal, false, colors

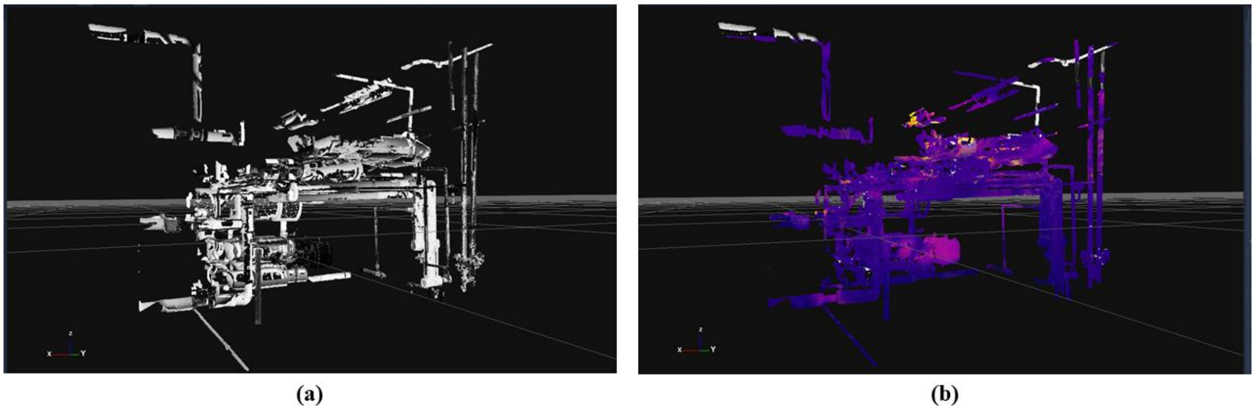

(a) (b)

Figure 16 - Point cloud merging for scene 2. (a) point cloud from the laser scanner; (b) point cloud from the laser scanner with projected thermal, false, colors

POINT CLOUD PROCESSING

As the skeletonization process is computationally intensive, decreasing the size of the original point clouds has a significant impact on the processing times of the application. Thus, after merging both data sets, the original point cloud was resampled by setting the minimum distances between points in the point cloud. Table 5 presents the point cloud sizes before and after resampling, and rules used to resample the sets.

Table 5 – Number of points before and after decimation (resampling).

Scene	1	2	3
Number of points before resampling	27,583,866	5,556,956	7,693,616
Target points distance for decimation (m)	0.015	0.015	0.015
Number of points after resampling	368,480	75,131	109,386

Both point clouds, geometric and thermal, were imported into MATLAB as .pts files containing the geometric, x, y, and z, coordinates of each point, and, for the thermal point cloud, the RGB values representing the false, thermal, colors. From this, the geometric point cloud was processed to produce the final BIM model, and the thermal point cloud was used along with the geometric point cloud during the system classification step to classify the pipes into the different systems in the scene.

Processing Times

Table 6 presents the processing times for all the steps described in the following subsections using the Alienware 17 R4 laptop.

Table 6 – Processing times for the reconstruction application.

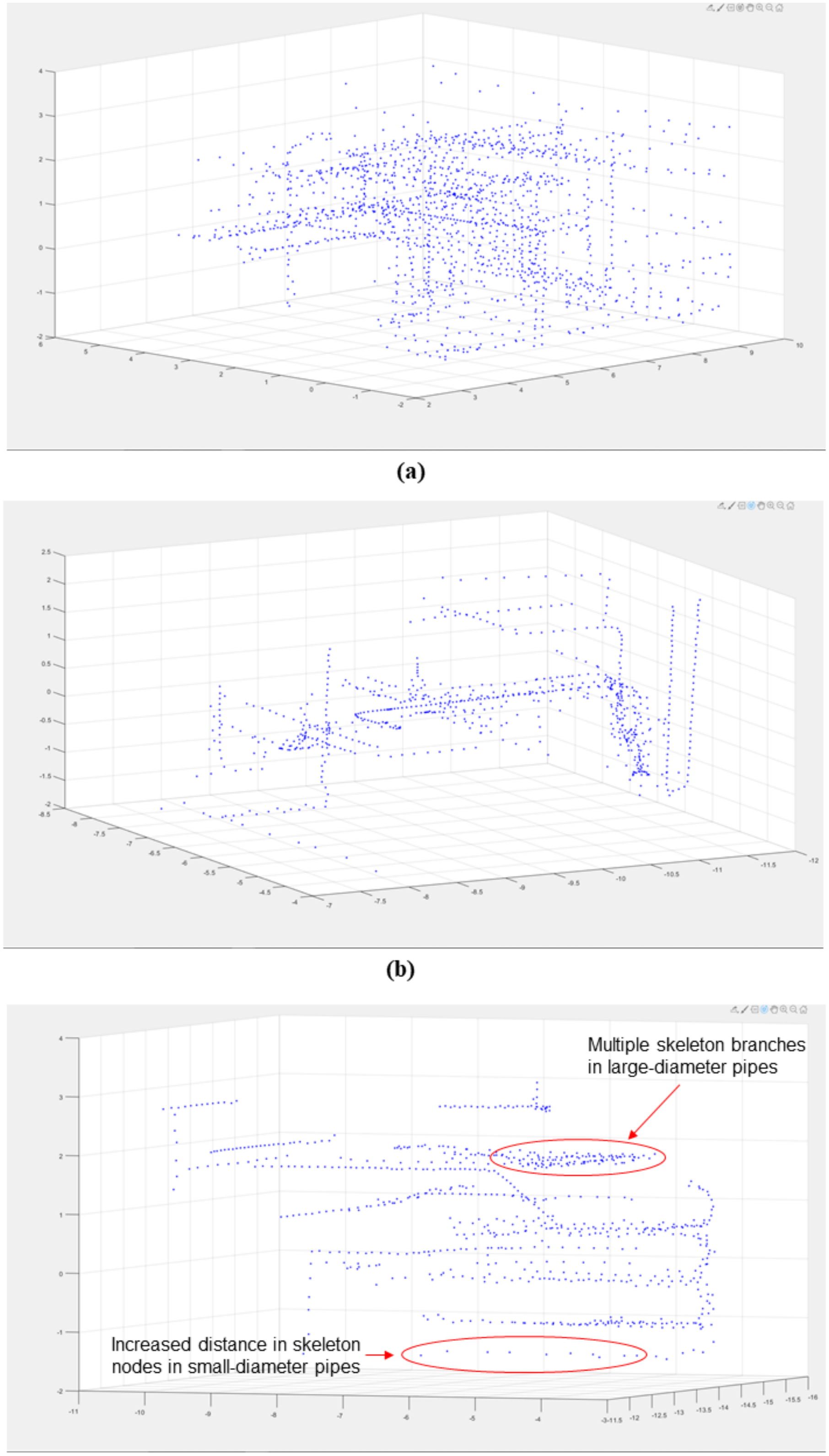
Scene	1	2	3
Number of skeleton nodes	1500	750	750
Process		Duration (s)	
Skeleton First Estimate	407.19	107.22	141.03
Skeleton Iteration	1,751.58	1063.33	882.96
Nodes Re-centering	3.13	1.48	1.75
Point Cloud Segmentation	4.50	1.58	1.67
Connection Identification	4.25	1.02	1.84
System Classification	7.81	6.02	6.34
Export MATLAB to Dynamo Revit	0.08	0.14	0.04
Dynamo Geometry Re-creation	3.95	5.78	5.19
TOTAL	2,182.49	1,186.57	1,040.82

Skeletonization

As presented in Table 6, skeletonization was the most consuming step in the proposed application, accounting for almost the entire processing time. That was expected because although skeleton-based reconstruction processes for pipes provide accurate results, the skeleton extraction is a computationally consuming process [117]. Figure 17 presents the skeletons for scenes 1 to 3.

The quality of the estimated skeletons was a result of the quality of the point clouds used to estimate them. For the three scenes, partial occlusion had a significant impact on the resulting skeletons. For some of the pipes, especially those that run close to walls and ceilings, only a portion of the total circumference was captured (Figure 18 (a)), which may have an impact on the estimation of the pipes' diameters. This situation is discussed in Son et al. [28], who present that problems associated with line-of-sight restrictions which allow each scan to capture information of about 2.5 dimensions of the object.

Another problem associated with partial occlusion is a result of the complex configuration of the pipes in the mechanical room. In situations where there were layers of pipes in different levels, and where these layers run in perpendicular directions, the continuity of the pipes in uppers layers was compromised in the original point cloud (Figure 18 (b)), and consequently in the resulting skeleton. As presented in Huang et al. [136], the skeletonization process is *"an ill-posed problem"* that can provide erroneous results if there is excessive noise or if a large number of points are missing in the original point cloud.



(a)

(b)

(c)

Figure 17 - Resulting skeletons for (a) scene 1; (b) scene 2; (c) scene 3.

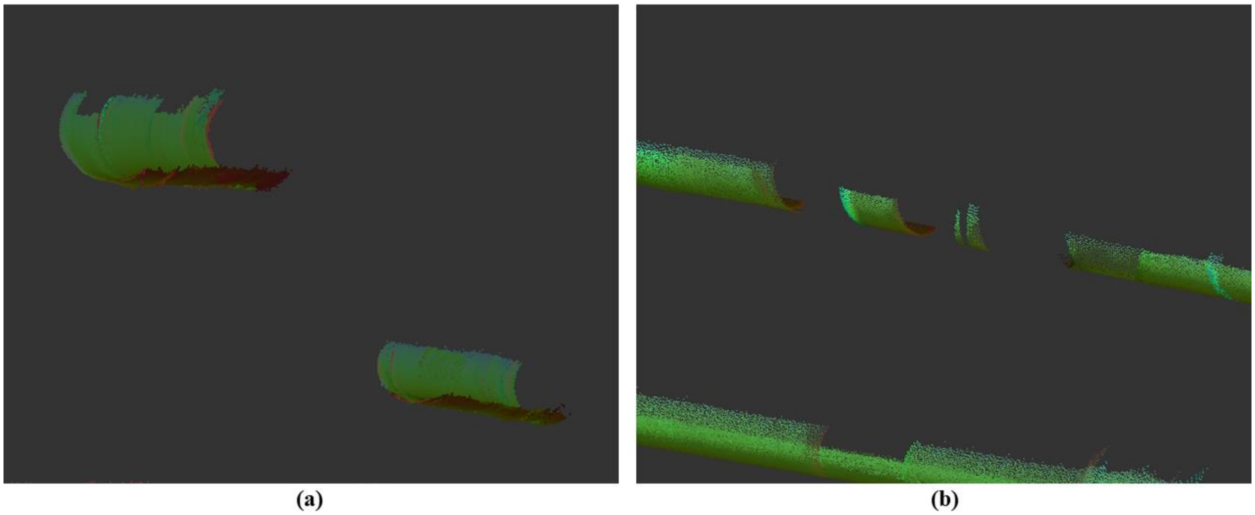


Figure 18 – Partial occlusion problems in the point clouds resulting in (a) only a portion of the complete circumference of a pipe being shown and (b) discontinuity in the pipe run.

Other results of interest in the skeletonization step include the not ideal performance of the application for pipes of larger diameters (>12 in). In some cases, because the diameter was large, two parallel and relatively close skeleton branches would be formed, which impacted the selection and recreation of straight pipe segments in subsequent steps. A possible solution for this issue is the development of an additional step for topological thinning, in which branches running too close would be merged as one single branch. For pipes with smaller diameters (<2in), on the other hand, the resampling step for decreasing the size of the processed point cloud, would decrease the density of points in the pipe run, thus decreasing the number of nodes in their associated skeletons and increasing the distance between them. Examples of both problems are shown in Figure 17 (c).

Nodes Re-centering

One of the parameters of the biggest interest in the application were the pipes' diameters. If the cylinder geometries describing the straight pipe segments were complete in the point clouds, the estimation of the pipes' central axis could be done relatively precisely. Partial and total occlusion and low point density in some parts of the pipes in the point cloud, on the other hand, compromised the quality of the estimated central axis, which was especially true for the pipes that run close to ceilings and walls. In such cases, only a portion of the geometries of the complete cylinders was available and the ability of the proposed application to correctly describe the central axis of the pipes was limited.

The development of the re-centering step aimed to reduce the impact of partial occlusion in the determination of a better estimate for the pipes' central axes. Figure 19 presents an example of re-centered nodes in the skeleton branch.

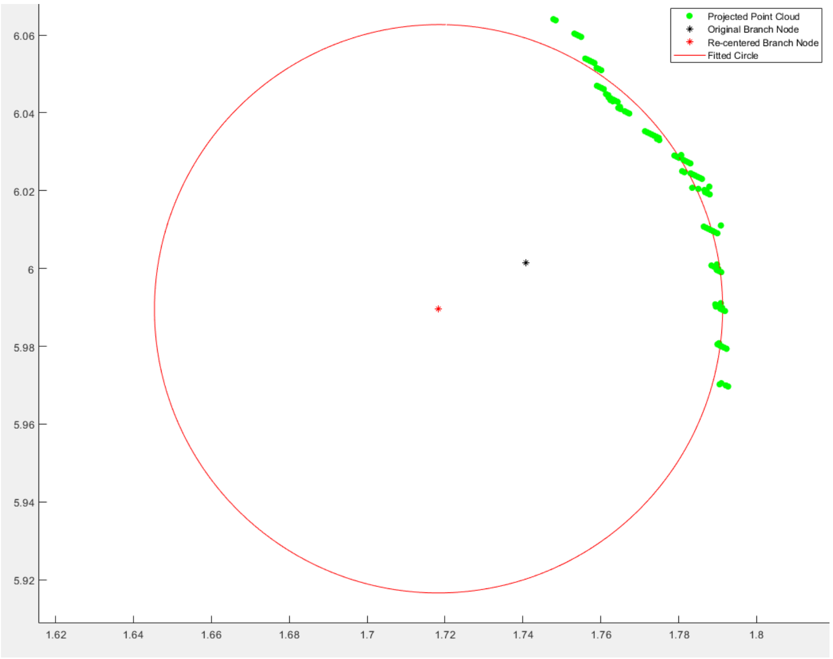


Figure 19 – Re-centering results for a skeleton node.

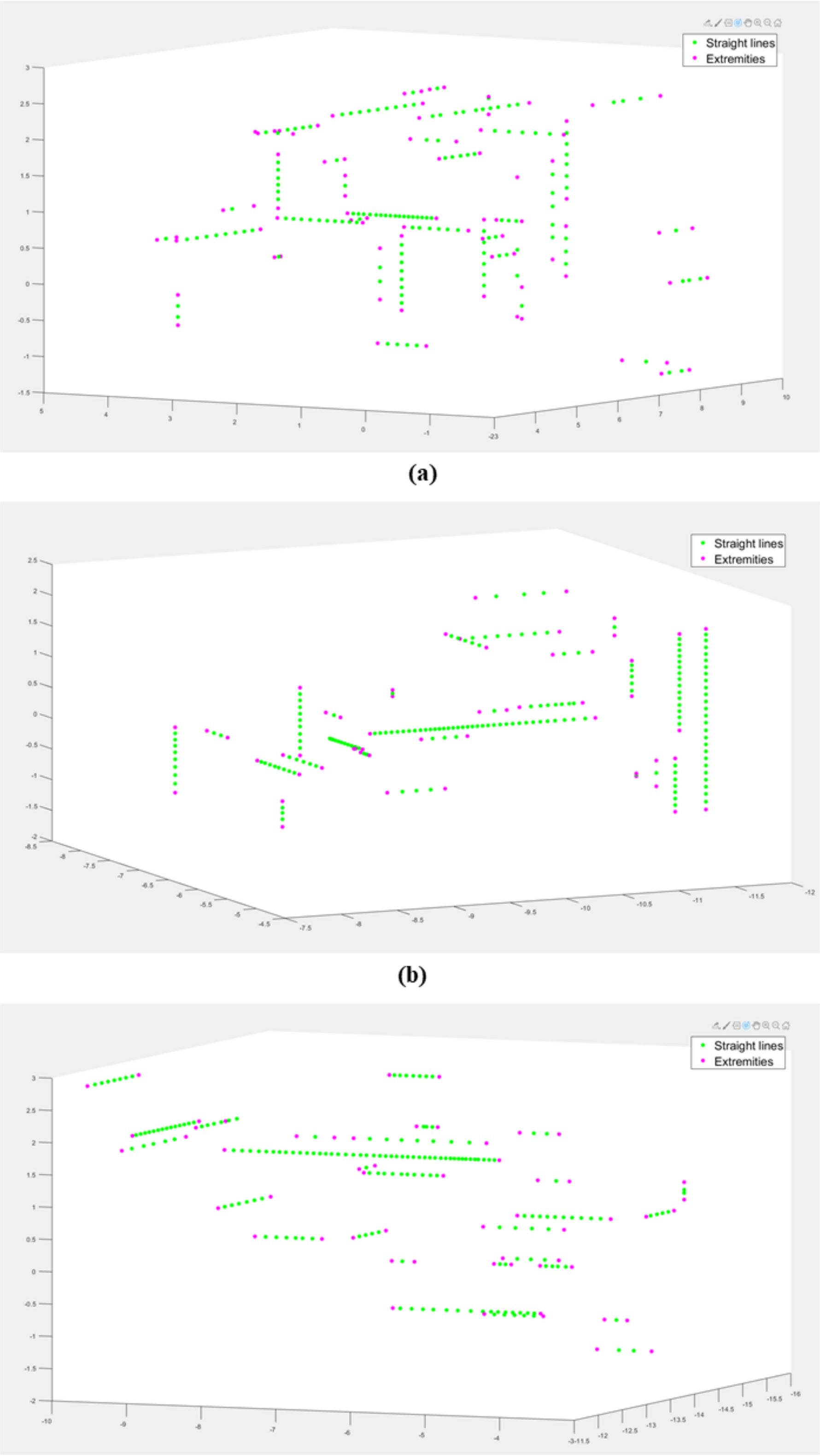
Based on the original branch node (black node) location, the 100 nearest neighbors to this point were selected and projected (green points) onto a plane that is perpendicular to the eigenvector parallel to the direction with the highest variance from the original branch node to its 10 nearest neighbors in the skeleton. From that, a circle was fitted to the points projected on the plane, and the center of the circle was used as the updated branch node, to be used in subsequent steps. The example shown in Figure 19 indicates that the original branch node was closer to the region with a higher point density given the weight of these points in the determination of the medial location of the original node. By re-centering this point, the effect of partial occlusion was reduced, and the new branch node (red node) represents the axis of the pipe better.

Segmentation and Node Classification

The re-centered skeleton nodes were, then, processed to identify candidates for straight pipe segments. Using the rules presented in the segmentation section of the methods, the straight pipe segments shown in Figure 20 were estimated from the skeleton candidates of Figure 17. The nodes in green represent branch nodes and the nodes in magenta represent the extremities of each straight pipe segment. These extremities were used in the connection identification step.

In the segmentation step, the estimated skeleton nodes were segmented into individual straight pipe runs, and their nodes were aligned considering the mean coordinates of the points in each segment and their associated standard deviations. Because the rules in the application work only for pipes that run parallel to the x, y, and z directions, some segments in other directions could not be properly aligned, as it is the case of the highlighted segment in Figure 21 (c). Figure 21 presents the match between the estimated skeletons and the aligned segments. Again, as it is shown in Figure 21 (c), the inclined segment was straightened out, which resulted in a mismatch between its adjacent segment. For all other straight segments, no issues were identified.

To solve this issue, there is a need to loosen up the rules that require that the segments run parallel to the x, y, and z axes. A possible solution is to align these segments relative to the largest direction of variance as defined by the orthonormal frame resulting from the eigenvectors associated with the variance matrix for the points in each straight segment.

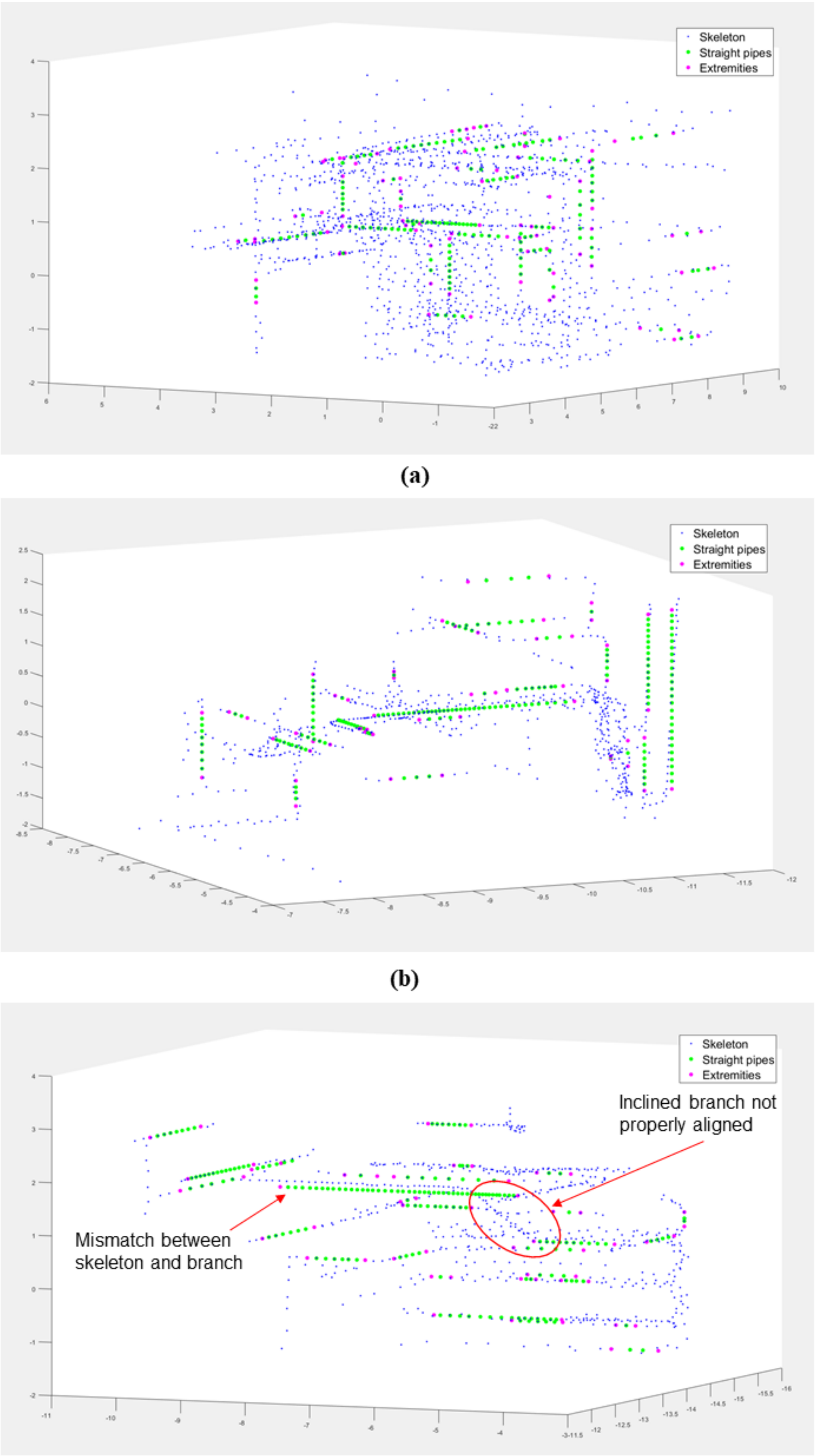


(a)

(b)

(c)

Figure 20 – Segmented and aligned straight segments for (a) scene 1; (b) scene 2; (c) scene 3. Points in green are branch nodes and points in magenta are extremities.



(a)

(b)

(c)

Figure 21 – Skeletons and aligned straight segments for (a) scene 1; (b) scene 2; (c) scene 3. Points in blue are skeleton nodes, points in green are branch nodes and points in magenta are extremities.

Parameter Estimation

The main issue with parameter estimation refers to the determination of the correct diameter of the pipe runs as the diameters determined during node re-centering refer to both pipe and its associated insulation. In such cases, the estimated diameters refer to the pipe plus the insulation, and further manual input is needed to correctly determine the pipes' diameters only. In the absence of insulation, no major concerns are expected, and the estimated diameters from MATLAB are expected to represent the actual diameter of the pipe.

Connection Identification

As presented, connection identification was based on the relationships between the straight pipe segments and the non-branch nodes in the skeletons. Three types of connections were estimated, namely tees, elbows, and couplings. As it will be presented in the next two sections, the MATLAB application estimated more connections than the number of connections that were modeled, which was a result of improper estimations in MATLAB, especially in complex configuration regions.

System Classification

System classification was the most challenging step in the application. First, the presence of insulation made it difficult to differentiate the temperatures of the systems. Thus, the application had to rely on sections of pipes with degraded or no insulation. To do so, the color of the 50 nearest points in the thermal point cloud to each node in a straight segment was collected and converted into a temperature value in the temperature range of the scene. For each point, the highest and lowest temperatures were stored, and, for each segment, the highest and lowest temperatures among the points in the segment were selected.

To determine which temperature would be used as the reference value for each pipe run, the temperature that provided the biggest difference relative to the reference temperature of the scene was selected. Then, this temperature was compared to a reference table that contained the possible systems in each scene and the ranges of temperature in the exposed sections of the pipes in each system as measured in the thermal images, which is shown in Table 7. To classify the pipes, the temperature of the pipe should be in the interval of the system in the reference table. As presented in Table 7, some systems were not being used at the time of data collection, and other systems could not have their temperature ranges determined as no exposure could be found in the pipes in those systems.

Table 7 – Mechanical systems and associated temperature ranges.

System	Temperature range (°F)		
	Scene 1	Scene 2	Scene 3
Medium Pressure Steam	[-]	[NA]	212-360
Low Pressure Steam	180-260*	212-240	[NA]
Condensate	[NA]	[NA]	120-212
Low Pressure Condensate	[-]	110-135	[NA]
Hot Water Condensate	105-125	[NA]	[NA]
Glycol Supply	[NA]	135-160	[NA]
Glycol Return	[NA]	[NR]	[NA]
Domestic Hot Water	[-]	[NA]	[NA]
Domestic Cold Water	40-60	[NA]	[NA]
Chilled Water Supply	[NA]	[NR]	[NA]
Chilled Water Return	[NA]	[NR]	[NA]

* Some sections of the low-pressure steam pipes as measured with the thermal camera presented temperatures inferior to 212 °F, even though it is steam.

Legend: [NA] – Not applicable; [NR] – Not Running; [-] no exposed area was found to measure temperature.

Systems with temperature ranges relatively close were also challenging to classify. In such cases, small color variations can classify the pipes in the wrong system. More challenging were the cases in which two different systems had overlapping temperature intervals. When that was the case, the overlapping intervals were not used for any of the systems. Figure 22 and Table 9 present the results of the system classification step.

Pipe Reconstruction in BIM

The final step of the application was the reconstruction of the pipes in Revit with the use of Dynamo Revit. The MATLAB results of straight pipes, connections, and diameters were exported as .csv tables into Dynamo and, from there, the Revit models were created. Figure 22 presents the three reconstructed scenes in Revit.

Although the positions of the straight pipe segments are relatively close to the locations of the pipes in the point clouds, many of the connections identified in MATLAB and presented in Table 8 were not reconstructed in Revit. For Figure 22 (a), scene 1, only one elbow was reconstructed. For Figure 22 (b), scene 2, four elbows were reconstructed, and finally, for Figure 22 (c), scene 3, one tee, two elbows, one coupling, and three reducers were modeled. These numbers, however, are smaller than the number of identified connections in MATLAB.

The results of the reconstructed models in Revit, although good estimates of the pipes in the original point clouds, should be taken with care. Among the main sources of modeling error are data editing, operator errors, and errors due to the choice of fitting algorithms, and thus the uncertainty of the recreated models is higher than the uncertainty of the point clouds [69]. In scene 2, Figure 22 (b), for example, there are two pipes classified as belonging to different systems, LPS and GS, respectively, but they are connected in the model. The cause of this error may be due to varying levels of exposure in each segment during data collection, or due to problems in the classification step when the 50 nearest neighbors in the thermal point cloud, relative to each skeleton node, were selected to determine the temperature of the segment.

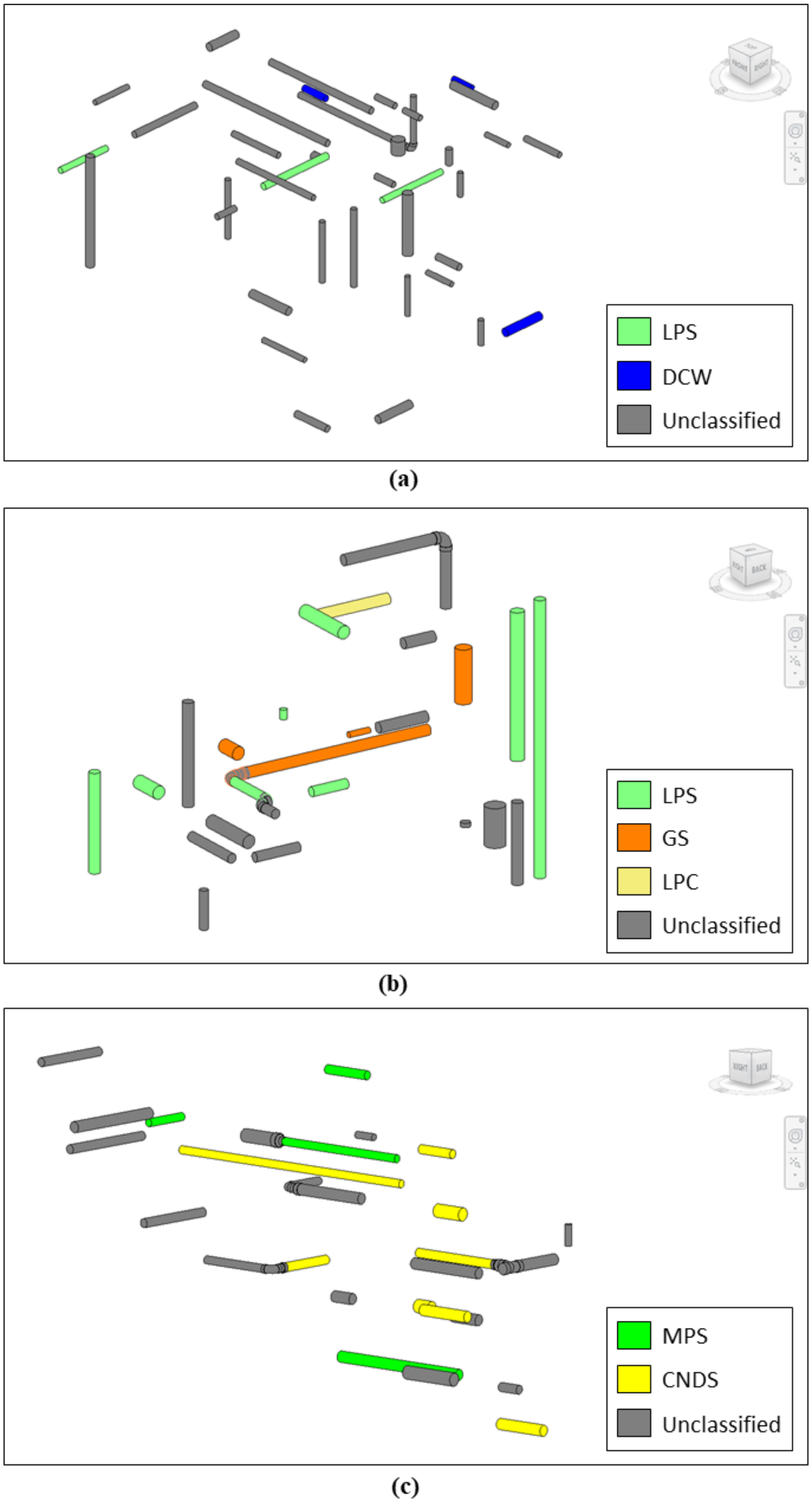


Figure 22 – Final Revit models for (a) scene 1; (b) scene 2; (c) scene 3.
Legend: CNDS – Condensate; DCW – Domestic Cold Water; GS – Glycol Supply; LPC – Low-Pressure Condensate; LPS – Low-Pressure Steam; MPS – Medium Pressure Steam.

Table 8 presents a summary of the results of the estimated elements in MATLAB in comparison to the number of elements in the point clouds.

Table 8 – Summary of results for identified instances in MATLAB.

Element	Identified instances (MATLAB) / Total instances (point cloud) (%)		
	Scene 1	Scene 2	Scene 3
Straight Pipes	40/149 (26.8%)	27/50 (54.0%)	28/35 (80.0%)
Tees	10/29 (34.5%)	7/9 (77.8%)	4/4 (100%)
Elbows and Couplings	1/84 (1.1%)	4/8 (50.0%)	3/19 (15.8%)

From Table 8, the smaller the number of straight pipe segments in the point clouds, the higher the percentage of identified straight pipe segments in the scenes by the MATLAB application. For scene 1, with 149 straight pipe segments in the point cloud, only 40 (26.8%) of these segments were identified, while for scene 3, with 35 straight pipe segments in the point cloud, 28 (80%) of these segments were identified. The same tendency is observed for the tee connections, the smaller the number of elements in the point cloud, the higher the percentage of identified elements. For the elbows, however, this tendency was not observed.

Based on the limitations of the application, some causes can be used to explain the non-identification of some of the straight pipe segments. First, straight pipe segments following any direction other than the x, y, and z directions were not modeled. Second, for large diameter pipes (> 12 in), the existence of multiple skeletons for a single pipe provided regions with excessive noise, which impaired the ability of the algorithm to properly identify the segment. For small diameter pipes (< 2 in), because the nodes were relatively far from each other, the application could not properly identify them as belonging to a straight segment. Finally, for regions with relatively high complexity of pipe configurations, surface points from close pipes may have been included in the iterations of the medial location of the nodes, which interfered with the positioning of the estimated skeleton nodes. For this last issue, the re-centering step aimed to reduce this bias, however, some inconsistencies may have remained even after re-centering.

Focusing now on the connections, because the application identifies the connections based on the relationships of straight pipe segments, the fact that some straight pipe segments were not identified reduced the ability of the application to identify the connections that are associated with them. Based on the number of identified instances in MATLAB presented in Table 8, Table 9 presents the summary of results for modeled instances in Revit, and compare them with the number of identified connections. It also presents the number of pipes that were classified considering the systems presented in Table 7.

Table 9 - Summary of results for modeled instances in Revit.

Element	Modeled instances (Revit) / Identified instances (MATLAB) (%)		
	Scene 1	Scene 2	Scene 3
Straight Pipes	40/40 (100%)	27/27 (100%)	28/28 (100%)
Tees	0/10 (0%)	0/7 (0%)	1/4 (25%)
Elbows and Couplings	1/1 (100%)	4/4 (100%)	3/3 (100%)
Reducers	0	0	3
Pipes Classified*	6/40 (15.0%)	13/27 (48.1%)	12/28 (42.9%)

* For the Pipes Classified row, exclusively, the ratios mean the number of pipes classified in Revit / total number of pipes modeled in Revit. These pipes classified in Revit refer only to pipes classified in categories different than “Other” in the MATLAB application.

As presented in Table 9, all the straight pipe segments and all the elbows/couplings that were identified in MATLAB could be successfully modeled. Alternatively, only one tee could be identified in Scene 3, and none could be identified in Scenes 1 and 2. Two main reasons can explain this fact. The first one is that some of these tees were wrongfully identified in the MATLAB application due to the complexity of the scenes, or even a bug in the code. The second option is that there was an issue with the geometry modeling in Dynamo/Revit. Modeling tees in Dynamo using the MEPover package can be done by two main approaches, one using two pipes (one main and one branch pipes), and the other using three pipes (two mains and one branch pipe). The routine presented in Appendix H uses the second approach and, due to the complexity of the configurations of the pipes in the scenes, problems with the inputs in some of the nodes may have caused the tees not to be correctly modeled. Finally, different than Table 8, Table 9 includes the modeling of reducers, which were not identified in MATLAB, but modeled in Revit whenever a change in diameter happened in a connection.

Finally, Table 9 presents the results of the classification step in MATLAB and Revit. For the three scenes, the percentages of classified pipes are lower than 50%, which shows that the application could not classify most of the pipes in the scenes. The main reasons for these findings are related to the existence of insulation around the pipes and the fact that some systems were not running during data collection, which did not allow for the collection of temperature values within the ranges presented in Table 7 for the systems in each scene. These findings help to answer both research questions presented in chapter one, which are presented in the next section again to facilitate the reading.

RESEARCH QUESTIONS REVISITED

- To which extent can thermal information be used in the classification of mechanical systems comprised of round pipes in automatic reconstruction applications for BIM?

As presented in Table 9, the application was able to identify and classify some of the straight pipe segments using the rules defined in the algorithm, which shows that thermal information can be used in the classification of mechanical systems for BIM applications. However, it should be considered that some limitations still exist. For this project, these limitations will be classified into three main groups: environmental constraints, data collection constraints, and data processing constraints.

Environmental constraints are related to the configuration of the pipes, the existence of insulation around them, and the need for the systems to be running during data collection. In the case of the configuration of the pipes, areas with a complex configuration of pipes may inhibit the reconstruction of the thermal point clouds. The existence of insulation around pipes, although necessary for energy conservation and reduced condensation-related problems, also limits the ability of thermal images to provide useful results, especially when no exposed areas are found in the scenes. Finally, if the system is not running, no useful information can be collected, as it was the case for chilled water supply and return in scene 2. Other environmental constraints not considered in this project include limitations associated with accessing spaces to collect data and eventual risks the operator may be exposed to during data collection.

Data collection constraints are associated with the time and resource requirements associated with taking the pictures. Due to the need for overlapping images, many pictures are required for each scene, which can be prohibitive in some cases. Just to illustrate, 1,903 images were taken for the three scenes in this project, which took over 12 hours for just a portion of a single mechanical room with 2912.6 sq. ft. Resource requirements may also limit the use of thermal cameras for these applications as thermal cameras with higher resolutions can be expensive for most users.

Finally, data processing constraints refer to the registration of the thermal images to create thermal 3D models. First, as the resolutions of the thermal images and their associated RGB images are relatively low when compared to the resolution of most of the modern digital cameras, the processing of these images may have been impaired in the registration software and, as presented in Table 3, many images were not registered in the photogrammetric point clouds. Added to that, the fact that visible images were extracted from the thermal images limited even more the ability of the software to register some images, especially because the overlap between images decreased after image extraction.

- What are the effects of insulation around the pipes on the ability of the proposed application to correctly classify the mechanical systems?

As presented, the existence of insulation around the pipes was one of the main limitations of the proposed application when considering the classification of the systems using temperature data. Table 9 shows that less than 50% of all pipes in the three scenes were not classified under any system presented in Table 7 and, for many of these systems, the fact that no exposed area of the pipes was found was the reason for the pipe not being classified.

To illustrate this fact, it is interesting to compare scenes 1 and 3. For scene 1, three out of the six systems in place could not have their temperatures measured because no exposed areas were found in the pipes, and thus these systems were not considered in the classification step. Additionally, the insulation around the pipes in scene 1 seemed relatively new when compared to the insulation around the pipes in scene 3 (Figure 23). Scene 3, on the other hand, presented the most exposure of the pipes because many portions of the insulation were deteriorated. As a result, only 6 out of 40 pipes were classified in scene 1 versus 12 out of 28 pipes being classified in scene 3. It is also interesting to note that, in some cases, the pipes were classified even when just a small portion of the pipe was exposed.

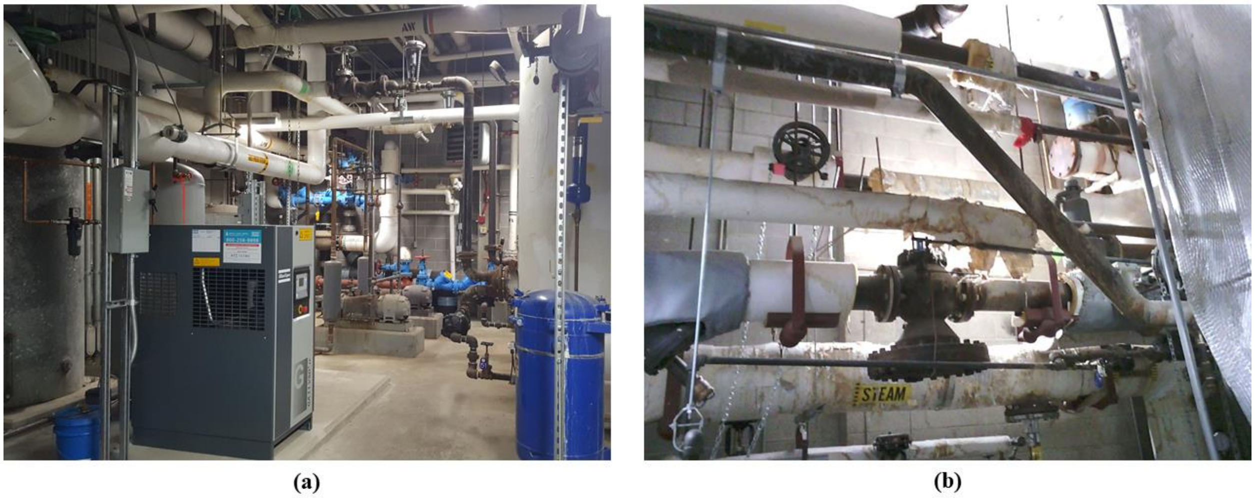


Figure 23 - Insulation around pipes in (a) scene 1; (b) scene 3.

PERFORMANCE MEASURES

Based on the work of Tang et al. [24], this section assesses the performance of the proposed application in terms of the 18 measures proposed by the authors. These measures can be further divided into three main groups, namely algorithm performance, environmental conditions, and algorithm performance for a given set of environmental conditions.

Algorithm Performance

Degree of automation: in its current state, the application is not fully automatic, thus requiring some level of user input. First, because the application is not able to differentiate between points that belong to pipes' surfaces and points that belong to other surfaces, the user is still required to clean the scenes to let only pipes in the scenes. To solve this issue, it is possible to develop an initial segmentation step based on curvature computation applied to the original point cloud to identify points that are candidates to belong to pipes.

Second, because it was needed to process and align two point clouds, a geometric one and a thermal one, the user is still required to align both sets by registering them together. For this, it is possible to automatically merge both data sets by integrating the two sensors and developing an algorithm for projecting thermal, false, colors onto the geometric point clouds on the fly. Some commercial laser scanners with this capability were found, such as the Leica BLK360, but the field of view, FoV, of the thermal sensor would not be suitable for the application as it does not collect a full-dome view of the scene.

Input and output assumption and data types: the application was developed in a way that it requires point clouds as simple .txt files in which each row represents a point in space defined by a set of x, y, and z, coordinates, and, in the case of thermal point clouds, a set of three values representing the false-color information in the Ironbow Palette. The output of the application is a single Revit model representing the 3D geometry of the pipes in the point cloud, and the identified connections, i.e., tees, elbows, and couplings. Additionally, the output includes the classification of the pipes into the different systems they represent.

For future developments, it is possible to use polygonal meshes as the input in the .txt file, instead of points, as to decrease the processing times of the skeletonization algorithm. For the outputs, it is possible to adapt the application to produce IFC models instead of proprietary Revit models. By doing so, it is possible to use the produced models in other BIM-based software.

Finally, it is possible to develop a full application, software, or even a plugin for some of the most common BIM-based software, as to avoid the use of MATLAB in part of the application. If that is the case, it will be possible to reduce the costs associated with acquiring

MATLAB and to provide a simpler and more intuitive environment for users not familiar with programming languages and concepts.

Computational complexity: as presented, the use of a skeletonization-based algorithm would provide relatively accurate results, but at the cost of a computationally consuming process. To process the original point clouds and estimate its skeleton, the algorithm needs to process every single point and verify its relationships with groups of neighboring points in a computationally-intensive process. Some tests performed on the proposed application show that increasing the size of the original point clouds has a significant impact on the initial estimation of the point clouds. One possible way to improve the performance of the proposed application is to use meshes instead of point sets, as it is the case in the application developed in Huang et al. [136].

To some extent, the user can control the computational complexity, and consequently the processing times, of the current application by setting less restrictive rules before running the scripts. Examples of parameters that the user can control to reduce the processing times include decreasing the density of the original point cloud, reducing the number of nodes in the estimated skeleton, reducing the number of iterations in the estimation of the pipes' central axes, and reducing the radius of the sphere that identify neighboring points in many steps of the application, h_0.

Extensibility to new environments: it is expected that future adaptations of the proposed application will include the processing of other cylindrical elements such as electrical conduits and round ducts, rectangular elements such as rectangular air ducts, and structural elements with defined sections such as steel beams and columns. In any case, the application would be using elements defined by a section that is extruded along a given path.

In the case of the conduits, it is possible that reducing the radius of the cylindrical elements may pose a challenge to the estimation of the radius of the real-world elements. For the other two applications, it will be needed to incorporate a set of geometric elements that represent the non-circular sections associated with ducts and structural elements to be fitted to the original point clouds, which will require a higher level of development.

Learning capabilities: at this point, the proposed application does not use learning to improve its performance, although such capabilities would benefit the application overall. Examples of learning capabilities that can be used in the application include the ability to differentiate the pipes from the insulation and the ability to recognize and differentiate the materials in the pipes.

Confidence-level and uncertainty modeling: Other limitations associated with the current state of the proposed application is the fact that it does not provide confidence-levels and uncertainty in geometric modeling. By including such capabilities may help the final user to assess to which extent the algorithm was able to estimate the correct diameters of the pipes and the correctness of the estimated connections.

Environmental Conditions

Types of objects present: for the three scenes used to test the proposed application, many elements were present in the scene. Common architectural elements included planar elements such as walls, floors, and ceilings. As the scenes were produced from a mechanical room, many pieces of mechanical equipment were present, including air handling units, pumps and expansion tanks. From the mechanical equipment, pipes and ducts connected in different sizes and in complex configurations, which created much occlusion in the produced scenes. There were also many conduits, switches, and power outlets.

Other than the architectural and mechanical elements in the scenes, clutter was identified. Part of the clutter was represented by furniture, including tables, chairs, computers. The rest of the clutter was represented by cardboard boxes and unused pipes, conduits, buckets, and other smaller elements.

Level of sensor noise: No measure for noise was used in this project, and the robustness of the application for noise was not determined.

Level of occlusion: No measure for the level of occlusion was used in this project, and the robustness of the application for the level of occlusion was not determined.

Level of clutter: No measure for the level of clutter was used in this project, and the robustness of the application for the level of clutter was not determined.

Presence of moving objects: no moving objects were present in the scenes such as people, animals, or vehicles, but the natural movement of the pipes due to the movement of the fluids inside them could have introduced noise in the resulting point clouds and interfered with the integrity of data.

Presence of specular surfaces: none of the scenes contained shiny surfaces, such as polished metals, which could have interfered with the laser scanner data. For the camera point cloud, there were no transparent materials, which could have interfered with the resulting point cloud.

Presence of dark (low-reflectance) surfaces: especially for scenes 2 and 3, some metal pipes have dark-colored surfaces that interfered with the resulting point cloud from the laser scanner, which led to missing data in some regions.

Sparseness of data: because the skeletonization process is computationally intensive, it was important to reduce the density of points in the processed scene to a level that increased performance, but without compromising the quality of the recreated models. The sparseness of the thermal point clouds was not computed as the skeletonization was only applied to the geometric point cloud.

Algorithm Performance for a Given Set of Environmental Conditions

Geometric modeling accuracy: the accuracy of the reconstructed models was measured by comparing the reconstructed model with the original point cloud in terms of positioning of objects, orientation, and parameters (diameters and system classification) of the modeled pipes. The accuracy of the reconstruction process is presented quantitatively in Table 8.

Recognition/labeling accuracy: as all the elements in the processed point clouds were pipes, the labeling of the pipes was restricted to classifying the reconstructed pipes in terms of the systems they represent based on temperature differences. The classification accuracy of the algorithm is presented quantitatively in Table 8.

Relationship modeling accuracy: relationship modeling was a critical step in the proposed application, as the connections between adjacent straight pipe segments were modeled based on the observance of a set of rules between the straight segments. The accuracy of the relationship modeling is presented quantitatively in Table 8.

Level of detail: at the current state of the application, the level of detail of the resulting BIM model is not too high. As the algorithm cannot differentiate the actual pipes from the insulation, the resulting BIM model is only of a generic pipe with relatively accurate dimensions, but no differentiation between pipes and insulation. By incorporating the ability of the application to separate pipes from insulation, it will be possible to produce more detailed outputs.

CHAPTER 5: CONCLUSIONS

Existing buildings account for a considerable portion of the total global energy consumption. As many of these buildings were built before the development of stricter energy codes and regulations, there are many opportunities to improve their energy performances and reduce their environmental impacts. In most of the cases, the planning of the renovation efforts will require the development of digital 3D models based on the as-is conditions of the building and its systems. Many commercial applications exist, but not many of them can provide semantic-rich BIM models, focusing solely on providing accurate geometric models.

This research project presented the development of an automatic application for reconstructing the geometric BIM models of pipes in existing buildings and associating semantic information to the models by classifying the reconstructed pipes based on thermal information collected with a thermal camera. The developed application, although providing more limited results when compared to current commercially available applications, presented a potential workflow for creating thermal 3D models using thermal images from an infrared camera, and using this model to automatically classify the pipes for BIM-based applications.

As for the geometric reconstruction process, the use of the skeleton-based method provided satisfactory results when identifying the skeleton nodes for most of the pipes, although it is still needed to develop more flexible rules for selecting the straight pipe segments and identifying the connections between these segments. Additionally, it is still needed to reduce the processing times of the skeletonization step by developing the process in a way the convergence of the skeleton nodes to the central axis occurs faster.

The proposed pipe classification feature, on the other hand, was more challenging and, due to the lack of consolidated workflows, is still limited. The limitations for the classification step can be divided into three main areas, including: (i) environmental constraints, as the existence of complex configurations complicated the thermal point cloud reconstruction, the existence of insulation limited temperature measurements, and the fact that some systems were not running during data collection impaired temperature data collection; (ii) data collection constraints, as the time requirements for collecting thermal images were long, and the costs associated with the thermal cameras may be high; and (iii) data processing constraints, as the registration of the thermal point clouds was limited due to the low resolution of the thermal camera and the need for extracting visible images from thermal images in this project. For the last two constraints, the use of laser scanners with associated thermal sensors may help to expedite the collection and registration of the thermal point clouds.

FUTURE RECOMMENDATIONS

The proposed application, however, is not complete, and some improvements can be done. As mentioned, it is still necessary to improve the computer code in aspects such as giving it the ability to differentiate the points that belong to the pipes to the points that represent noise in the application, including walls, ceilings, roof, floors, furniture, and other elements. Also, a better method for determining the diameter of the pipe segments is needed as in its current state the application cannot separate the pipes from the insulation. As shown, most of the errors in the application were due to partial occlusion and low point density in some regions of the scene, which led to the wrong estimate of the center of the pipes. Yet, it is still needed to develop less strict rules for pipe alignment, which would allow for the reconstruction of pipes in directions other than the x, y, and z axes. Finally, it is possible to adapt the application as a plugin for some of the BIM-based software, or even as standalone software, which would make the application available and more intuitive for users who do not have access to the software used in this project.

REFERENCES

[1] R.J. Scherer, P. Katranuschkov, BIMification: How to create and use BIM for retrofitting, Adv. Eng. Informatics. 38 (2018) 54–66. https://doi.org/10.1016/j.aei.2018.05.007.

[2] NIBS, National Building Information Modeling Standard. Version 1 - Part 1: Overview, Principles, and Methodologies, 2007.

[3] UN Environment and International Energy Agency, Towards a zero-emission, efficient, and resilient buildings and construction sector. Global Status Report 2017., 2017.

[4] Architecture 2030, Existing buildings: operational emissions, (n.d.). https://architecture2030.org/existing-buildings-operation/ (accessed March 31, 2020).

[5] W.T. Grondzik, A.G. Kwok, Mechanical and electrical equipment for buildings, 12th ed., John Wiley & Sons, Inc, Hoboken, NJ, 2014.

[6] M.A. Brown, F. Southworth, T.K. Stovall, Towards a Climate-Friendly Built Environment, Arlington, VA, 2005. https://www.c2es.org/document/towards-a-climate-friendly-built-environment/.

[7] C. Eastman, P. Teicholz, R. Sacks, K. Liston, BIM handbook: A guide to Building Information Modeling for owners, managers, designers, engineers, and contractors, 2nd ed., Wiley, Hoboken, NJ, 2008.

[8] R. Sacks, C. Eastman, G. Lee, P. Teicholz, BIM Handbook: A Guide to Building Information Modeling for Owners, Designers, Engineers, Contractors, and Facility Managers, 3rd ed., John Wiley & Sons, Hoboken, NJ, 2018.

[9] R. Liu, R.R.A. Issa, Survey: Common knowledge in BIM for facility maintenance, J. Perform. Constr. Facil. 30 (2016). https://doi.org/10.1061/(ASCE)CF.1943-5509.0000778.

[10] S.N. Naghshbandi, BIM for Facility Management: Challenges and Research Gaps, Civ. Eng. J. 2 (2016) 679–684. https://doi.org/10.28991/cej-2016-00000067.

[11] B. Becerik-Gerber, F. Jazizadeh, N. Li, G. Calis, Application areas and data requirements for BIM-enabled facilities management, J. Constr. Eng. Manag. 138 (2012) 431–442. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000433.

[12] M. Khaddaj, I. Srour, Using BIM to Retrofit Existing Buildings, Procedia Eng. 145 (2016) 1526–1533. https://doi.org/10.1016/j.proeng.2016.04.192.

[13] Energiesprong, Transition Zero. Energiesprong, 2015. https://energiesprong.org/wp-content/uploads/2017/04/EnergieSprong_UK-Transition_Zero_document.pdf.

[14] U.S. Green Building Council, LEED rating system, U.S. Green Build. Counc. (2020). https://www.usgbc.org/leed (accessed February 5, 2020).

[15] J.J. McArthur, A Building Information Management (BIM) Framework and Supporting Case

Study for Existing Building Operations, Maintenance and Sustainability, Procedia Eng. 118 (2015) 1104–1111. https://doi.org/10.1016/j.proeng.2015.08.450.

[16] F. Bosché, A. Guillemet, Y. Turkan, C.T. Haas, R. Haas, Tracking the built status of MEP works: Assessing the value of a Scan-vs-BIM system, J. Comput. Civ. Eng. 28 (2014) 1–13. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000343.

[17] L. Klein, N. Li, B. Becerik-Gerber, Imaged-based verification of as-built documentation of operational buildings, Autom. Constr. 21 (2012) 161–171. https://doi.org/10.1016/j.autcon.2011.05.023.

[18] B. Gu, S. Ergan, B. Akinci, Challenges Associated with Generating Accurate As-is Building Information Models for Existing Buildings by Leveraging Heterogeneous Data Sources, in: Comput. Civ. Build. Eng., 2014: pp. 560–568.

[19] Y. Arayici, Towards building information modelling for existing structures, Struct. Surv. 26 (2008) 210–222. https://doi.org/10.1108/02630800810887108.

[20] Y. Arayici, The use of the 3D laser scanner in the built environment, in: Built Environ. Des. Manag. Appl., 2010: pp. 89–120.

[21] J. Lee, H. Son, C. Kim, C. Kim, Skeleton-based 3D reconstruction of as-built pipelines from laser-scan data, Autom. Constr. 35 (2013) 199–207. https://doi.org/10.1016/j.autcon.2013.05.009.

[22] H. Veldhuis, G. Vosselman, The 3D reconstruction of straight and curved pipes using digital line photogrammetry, ISPRS J. Photogramm. Remote Sens. 53 (1998) 6–16. https://doi.org/10.1016/S0924-2716(97)00031-2.

[23] D. Chapman, A. Deacon, J.-L. Brown, an Omnidirectional Imaging System for the Reverse Engineering of Industrial Facilities, Arch. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. Vol. 34, P (2004).

[24] P. Tang, D. Huber, B. Akinci, R. Lipman, A. Lytle, Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques, Autom. Constr. 19 (2010) 829–843. https://doi.org/10.1016/j.autcon.2010.06.007.

[25] Leica, Laser Scanners, Leica Geosystems AG. (n.d.). https://leica-geosystems.com/products/laser-scanners (accessed February 7, 2020).

[26] S. Ochmann, R. Vock, R. Wessel, R. Klein, Automatic reconstruction of parametric building models from indoor point clouds, Comput. Graph. 54 (2016) 94–103. https://doi.org/10.1016/j.cag.2015.07.008.

[27] K. Kawashima, S. Kanai, H. Date, Automatic Recognition of Piping System From Large-Scale Terrestrial Laser Scan Data, ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XXXVIII-5/ (2011) 283–288. https://doi.org/10.5194/isprsarchives-XXXVIII-5-W12-283-2011.

[28] H. Son, C. Kim, C. Kim, Fully Automated As-Built 3D Pipeline Extraction Method from Laser-Scanned Data Based on Curvature Computation, J. Comput. Civ. Eng. 29 (2014) B4014003. https://doi.org/10.1061/(asce)cp.1943-5487.0000401.

[29] ClearEdge3D, EdgeWise, Clear. Inc. (n.d.). https://www.clearedge3d.com/products/edgewise/ (accessed February 7, 2020).

[30] L. López-Fernández, S. Lagüela, D. González-Aguilera, H. Lorenzo, Thermographic and mobile indoor mapping for the computation of energy losses in buildings, Indoor Built Environ. 26 (2017) 771–784. https://doi.org/10.1177/1420326X16638912.

[31] M. Golparvar-Fard, Y. Ham, Automated diagnostics and visualization of potential energy performance problems in existing buildings using energy performance augmented reality models, J. Comput. Civ. Eng. 28 (2014) 17–29. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000311.

[32] D. González-Aguilera, P. Rodriguez-Gonzalvez, J. Armesto, S. Lagüela, Novel approach to 3D thermography and energy efficiency evaluation, Energy Build. 54 (2012) 436–443. https://doi.org/10.1016/j.enbuild.2012.07.023.

[33] S. Lagüela, J. Armesto, H. González-Jorge, P. Arias, J. Herráez, Automation of thermographic 3D modelling through image fusion and image matching techniques, Autom. Constr. 27 (2012) 24–31. https://doi.org/10.1016/j.autcon.2012.05.011.

[34] S. Lagüela, L. Díaz-Vilariño, J. Armesto, P. Arias, Non-destructive approach for the generation and thermal characterization of an as-built BIM, Constr. Build. Mater. 51 (2014) 55–61. https://doi.org/10.1016/j.conbuildmat.2013.11.021.

[35] Y.K. Cho, Y. Ham, M. Golpavar-Fard, 3D as-is building energy modeling and diagnostics: A review of the state-of-the-art, Adv. Eng. Informatics. 29 (2015) 184–195. https://doi.org/10.1016/j.aei.2015.03.004.

[36] S. Gunnarson, D. Ben-Yehoshua, T. Stefansson, K. V Matthíasdóttir, Y. Guðmundsson, UAV-Based Thermal Imagery of Hot Water and Steam Pipelines for Internal Leakage, (2020) 1–6.

[37] R.G. Dulce, A.D. Baltazar, F.E.B. Bayon, An Evaluation on the Applicability of Infrared Thermography in Identifying Silica Deposition in Geothermal Brine Lines, World Geotherm. Congr. 2015. (2015) 7.

[38] R. Vanlande, C. Nicolle, C. Cruz, IFC and building lifecycle management, Autom. Constr. 18 (2008) 70–78. https://doi.org/10.1016/j.autcon.2008.05.001.

[39] International Organization for Standardization, ISO 29481-1:2016 Building information models — Information delivery manual — Part 1: Methodology and format, ISO. (2016) 29.

[40] R. Volk, J. Stengel, F. Schultmann, Building Information Modeling (BIM) for existing buildings - Literature review and future needs, Autom. Constr. 38 (2014) 109–127. https://doi.org/10.1016/j.autcon.2013.10.023.

[41] L. Sabol, BIM Technology for FM, in: P. Teicholz (Ed.), BIM Facil. Manag., John Wiley & Sons, Inc., Hoboken, New Jersey, 2013: pp. 17–45.

[42] R. Liu, R.R.A. Issa, BIM for Facility Management: design for maintainability with BIM tools, in: ISARC 2013 - 30th Int. Symp. Autom. Robot. Constr. Mining, Held Conjunction with 23rd World Min. Congr., 2013: pp. 321–328. https://doi.org/10.22260/ISARC2013/0035.

[43] P. Teicholz, Owner BIM for FM Guidelines, in: P. Teicholz (Ed.), BIM Facil. Manag., John Wiley & Sons, Inc., Hoboken, New Jersey, 2013: pp. 47–84.

[44] K.A. Aguilar, H.W. Ashcraft, Legal Issues When Considering BIM for Facilities Management, in: P. Teicholz (Ed.), BIM Facil. Manag., John Wiley & Sons, Hoboken, NJ, 2013: pp. 85–106.

[45] A. Golabchi, M. Akula, V.R. Kamat, Leveraging bim for automated fault detection in operational buildings, ISARC 2013 - 30th Int. Symp. Autom. Robot. Constr. Mining, Held Conjunction with 23rd World Min. Congr. 48109 (2013) 187–197. https://doi.org/10.22260/isarc2013/0020.

[46] N.D. Aziz, A.H. Nawawi, N.R.M. Ariff, Building Information Modelling (BIM) in Facilities Management: Opportunities to be Considered by Facility Managers, Procedia - Soc. Behav. Sci. 234 (2016) 353–362. https://doi.org/10.1016/j.sbspro.2016.10.252.

[47] A. Akcamete, B. Akinci, J.H. Garrett, Potential utilization of building information models for planning maintenance activities, Proc. Int. Conf. Comput. Civ. Build. Eng. (2010) 151–157.

[48] A. Akbarnezhad, K.C.G. Ong, L.R. Chandra, Economic and environmental assessment of deconstruction strategies using building information modeling, Autom. Constr. 37 (2014) 131–144. https://doi.org/10.1016/j.autcon.2013.10.017.

[49] N. Hichri, C. Stefani, L. De Luca, P. Veron, G. Hamon, From Point Cloud To Bim: a Survey of Existing Approaches, ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XL-5/W2 (2013) 343–348. https://doi.org/10.5194/isprsarchives-xl-5-w2-343-2013.

[50] X. Xiong, A. Adan, B. Akinci, D. Huber, Automatic creation of semantically rich 3D building models from laser scanner data, Autom. Constr. 31 (2013) 325–337. https://doi.org/10.1016/j.autcon.2012.10.006.

[51] M.P. Gallaher, A.C. O'Connor, J.L. Dettbarn, L.T. Gilday, Cost Analysis of Inadequate Interoperability in the U.S. Capital Facilities Industry. National Institute of Standards and Technology (NIST), Gaithersburg, MD, 2004. http://www.bentleyuser.dk/sites/default/files/nist_report.pdf.

[52] P. Pishdad-Bozorgi, X. Gao, C. Eastman, A.P. Self, Planning and developing facility management-enabled building information model (FM-enabled BIM), Autom. Constr. 87 (2018) 22–38. https://doi.org/10.1016/j.autcon.2017.12.004.

[53] B. East, Using COBie, in: P. Teicholz (Ed.), BIM Facil. Manag., John Wiley & Sons, Inc., Hoboken, New Jersey, 2013: pp. 107–143.

[54] Y. Ham, Vision-based building energy diagnostics and retrofit analysis using 3D thermography and Building Information Modeling (Doctoral dissertation), The University of Illinois at Urbana-Champaign, 2015.

[55] X. Yang, S. Ergan, Leveraging BIM to Provide Automated Support for Efficient Troubleshooting of HVAC-Related Problems, J. Comput. Civ. Eng. 30 (2016) 1–13. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000492.

[56] T. Gerrish, K. Ruikar, M. Cook, M. Johnson, M. Phillip, C. Lowry, BIM application to building energy performance visualisation and management: Challenges and potential, Energy Build. 144

(2017) 218–228. https://doi.org/10.1016/j.enbuild.2017.03.032.

[57] A. Khalil, S. Stravoravdis, H-BIM and the domains of data investigations of heritage buildings current state of the art, ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. 42 (2019) 661–667. https://doi.org/10.5194/isprs-Archives-XLII-2-W11-661-2019.

[58] Y. Arayici, An approach for real world data modelling with the 3D terrestrial laser scanner for built environment, Autom. Constr. 16 (2007) 816–829. https://doi.org/10.1016/j.autcon.2007.02.008.

[59] M. Der Yang, T.C. Su, H.Y. Lin, Fusion of infrared thermal image and visible image for 3D thermal model reconstruction using smartphone sensors, Sensors (Switzerland). 18 (2018). https://doi.org/10.3390/s18072003.

[60] A. Bhatla, S.Y. Choe, O. Fierro, F. Leite, Evaluation of accuracy of as-built 3D modeling from photos taken by handheld digital cameras, Autom. Constr. 28 (2012) 116–127. https://doi.org/10.1016/j.autcon.2012.06.003.

[61] M. Kedzierski, A. Fryskowska, Terrestrial and aerial laser scanning data integration using wavelet analysis for the purpose of 3D building modeling, Sensors (Switzerland). 14 (2014) 12070–12092. https://doi.org/10.3390/s140712070.

[62] L. Bartolini, M. Ferri de Collibus, G. Fornetti, M. Guarneri, E. Paglia, C. Poggi, R. Ricci, Amplitude-modulated laser rangefinder for 3D imaging with multisensor data integration capabilities, in: Proc. SPIE 5850, Adv. Laser Technol. 2004, Rome and Frascati, Italy, 2005: pp. 152–159. https://doi.org/10.1117/12.633683.

[63] T. Mill, A. Alt, R. Liias, Combined 3D building surveying techniques-Terrestrial laser scanning (TLS) and total station surveying for BIM data management purposes, J. Civ. Eng. Manag. 19 (2013) 23–32. https://doi.org/10.3846/13923730.2013.795187.

[64] S. Lagüela, J. Martínez, J. Armesto, P. Arias, Energy efficiency studies through 3D laser scanning and thermographic technologies, Energy Build. 43 (2011) 1216–1221. https://doi.org/10.1016/j.enbuild.2010.12.031.

[65] L. Barazzetti, F. Banfi, R. Brumana, M. Previtali, F. Roncoroni, Bim From Laser Scans… Not Just for Buildings: Nurbs-Based Parametric Modeling of a Medieval Bridge, ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. III–5 (2016) 51–56. https://doi.org/10.5194/isprsannals-iii-5-51-2016.

[66] A. Bucksch, R. Lindenbergh, M. Menenti, Robust skeleton extraction from imperfect point clouds, Vis. Comput. 26 (2010) 1283–1300. https://doi.org/10.1007/s00371-010-0520-4.

[67] J. Li, H. Bao, X. Han, F. Pan, W. Pan, F. Zhang, D. Wang, Real-time self-driving car navigation and obstacle avoidance using mobile 3D laser scanner and GNSS, Multimed. Tools Appl. 76 (2017) 23017–23039. https://doi.org/10.1007/s11042-016-4211-7.

[68] W. Böhler, A. Marbs, 3D scanning instruments, in: Proc. CIPA WG 6 Int. Work. Scanning Cult. Herit. Rec., 2002: pp. 9–18. http://www.i3mainz.fh-mainz.de/publicat/korfu/p05_Boehler.pdf.

[69] GSA, GSA Building Information Modeling Guide Series: 03 – GSA BIM Guide for 3D Imaging, U.S. Gen. Serv. Adm. (2009). https://www.gsa.gov/cdnstatic/GSA_BIM_Guide_Series_03.pdf.

[70] N. Pfeifer, P. Dorninger, A. Haring, H. Fan, Investigating terrestrial laser scanning intensity data: Quality and functional relations, 8th Conf. Opt. 3-D Meas. Tech. (2007) 328–337.

[71] Leica Geosystems, Leica BLK360 Imaging Laser Scanner, Leica Geosystems AG. (2017). https://leica-geosystems.com/products/laser-scanners/scanners/blk360 (accessed February 15, 2020).

[72] G. Antova, Registration Process of Laser Scan Data in the Field of Deformation Monitoring, Procedia Earth Planet. Sci. 15 (2015) 549–552. https://doi.org/10.1016/j.proeps.2015.08.096.

[73] T. Rabbani, F. Van Den Heuvel, 3D industrial reconstruction by fitting CSG models to a combination of images and point clouds, Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. 35 (2004) 2.

[74] A. Costanzo, M. Minasi, G. Casula, M. Musacchio, M.F. Buongiorno, Combined use of terrestrial laser scanning and IR Thermography applied to a historical building, Sensors (Switzerland). 15 (2014) 194–213. https://doi.org/10.3390/s150100194.

[75] L. Barazzetti, F. Banfi, R. Brumana, D. Oreni, M. Previtali, F. Roncoroni, HBIM and augmented information: Towards a wider user community of image and range-based reconstructions, Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch. 40 (2015) 35–42. https://doi.org/10.5194/isprsarchives-XL-5-W7-35-2015.

[76] F. Remondino, S. El-Hakim, Image-based 3D modelling: a review, Photogramm. Rec. 21 (2006) 269–291. https://doi.org/https://doi.org/10.1111/j.1477-9730.2006.00383.x.

[77] J.L. Schonberger, J.M. Frahm, Structure-from-Motion Revisited, Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 2016-Decem (2016) 4104–4113. https://doi.org/10.1109/CVPR.2016.445.

[78] M.J. Westoby, J. Brasington, N.F. Glasser, M.J. Hambrey, J.M. Reynolds, "Structure-from-Motion" photogrammetry: A low-cost, effective tool for geoscience applications, Geomorphology. 179 (2012) 300–314. https://doi.org/10.1016/j.geomorph.2012.08.021.

[79] G.J. Edelman, M.C. Aalders, Photogrammetry using visible, infrared, hyperspectral and thermal imaging of crime scenes, Forensic Sci. Int. 292 (2018) 181–189. https://doi.org/10.1016/j.forsciint.2018.09.025.

[80] S. Tuttas, A. Braun, A. Borrmann, U. Stilla, Comparision of photogrammetric point clouds with BIM building elements for construction progress monitoring, Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch. 40 (2014) 341–345. https://doi.org/10.5194/isprsarchives-XL-3-341-2014.

[81] H. Mahami, F. Nasirzadeh, A.H. Ahmadabadian, S. Nahavandi, Automated progress controlling and monitoring using daily site images and building information modelling, Buildings. 9 (2019). https://doi.org/10.3390/buildings9030070.

[82] A. Murtiyoso, M. Koehl, P. Grussenmeyer, T. Freville, Acquisition and processing protocols for UAV images: 3D modeling of historical buildings using photogrammetry, ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. 4 (2017) 163–170. https://doi.org/10.5194/isprs-annals-IV-2-W2-163-2017.

[83] A. Ley, R. Hänsch, O. Hellwich, Automatic building abstraction from aerial photogrammetry, ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. 4 (2017) 243–250. https://doi.org/10.5194/isprs-annals-IV-2-W4-243-2017.

[84] M. Lo Brutto, D. Ebolese, G. Dardanelli, 3D modelling of a historical building using close-range photogrammetry and remotely piloted aircraft system (RPAS), Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch. 42 (2018) 599–606. https://doi.org/10.5194/isprs-archives-XLII-2-599-2018.

[85] P. Arias, J.C. Caamaño, H. Lorenzo, J. Armesto, 3D modeling and section properties of ancient irregular timber structures by means of digital photogrammetry, Comput. Civ. Infrastruct. Eng. 22 (2007) 597–611. https://doi.org/10.1111/j.1467-8667.2007.00512.x.

[86] J. Armesto, I. Lubowiecka, C. Ordóñez, F.I. Rial, FEM modeling of structures based on close range digital photogrammetry, Autom. Constr. 18 (2009) 559–569. https://doi.org/10.1016/j.autcon.2008.11.006.

[87] Y. Furukawa, B. Curless, S.M. Seitz, R. Szeliski, Reconstructing building interiors from images, 2009 IEEE 12th Int. Conf. Comput. Vis. (2010) 80–87. https://doi.org/10.1109/iccv.2009.5459145.

[88] C. Palestini, A. Basso, L. Graziani, Integrated photogrammetric survey and BIM modelling for the protection of school heritage, applications on a case study, Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch. 42 (2018) 821–828. https://doi.org/10.5194/isprs-archives-XLII-2-821-2018.

[89] C. Meola, Origin and Theory of Infrared Thermography, in: C. Meola (Ed.), Infrared Thermogr. Recent Adv. Futur. Trends, Bentham Science Publishers, Sharjah, U.A.E, 2012: pp. 3–28. https://doi.org/10.2174/97816080514341120101003.

[90] R. Rinaldi, Infrared Devices: Short History and New Trends, in: C. Meola (Ed.), Infrared Thermogr. Recent Adv. Futur. Trends, Bentham Science Publishers, Sharjah, U.A.E, 2012: pp. 29–59. https://doi.org/10.2174/97816080514341120101003.

[91] E. Grinzato, State of the Art and Perspective of Infrared Thermography Applied to Building Science, in: C. Meola (Ed.), Infrared Thermogr. Recent Adv. Futur. Trends, Bentham Science Publishers, Sharjah, U.A.E, 2012: pp. 200–229. https://doi.org/10.2174/97816080514341120101003.

[92] Y. Ham, M. Golparvar-Fard, BIM and Thermographic Sensing: Reflecting the As-is Building Condition in Energy Analysis, J. Constr. Eng. Proj. Manag. 5 (2015) 16–22. https://doi.org/10.6106/jcepm.2015.5.4.016.

[93] D. González-Aguilera, S. Lagüela, P. Rodríguez-Gonzálvez, D. Hernández-López, Image-based thermographic modeling for assessing energy efficiency of buildings façades, Energy Build. 65 (2013) 29–36. https://doi.org/10.1016/j.enbuild.2013.05.040.

[94] C.N. Grey, E.C. Wartman, Infrared Technology application in Building Envelopes: Applications, ASTM Standards, and Limitations, in: ITC User Conf., Coventry, UK, 2017: pp. 1–10. http://www.sgh.com/sites/default/files/downloads/Knowledge-Sharing/11_grey.pdf.

[95] D. Borrmann, J. Elseberg, N. Andreas, Thermal 3D Mapping of Building Facades Thermal 3D Mapping of Building Facades, Intell. Auton. Syst. (2012). https://doi.org/10.1007/978-3-642-33926-4.

[96] G. Clarkson, S. Luo, R. Fuentes, Thermal 3D modelling, ISARC 2017 - Proc. 34th Int. Symp. Autom. Robot. Constr. (2017) 493–499. https://doi.org/10.22260/isarc2017/0068.

[97] S. Lagüela, L. Díaz-Vilariño, J. Armesto, P. Arias, Thermographic 3D models as the foundation for Building Information Models, in: 11th Int. Conf. Quant. InfraRed Thermogr., Naples, Italy, 2012. https://doi.org/10.21611/qirt.2012.180.

[98] W. Natephra, A. Motamedi, N. Yabuki, T. Fukuda, T. Michikawa, Building Envelope Thermal Performance Analysis using BIM-Based 4D Thermal Information Visualization, Icccbe 2016. (2016) 1539–1546.

[99] H. Youngjib, M. Golparvar-Fard, Updating R-values of BIM Elements using 3D Thermography for Accurate Building Energy Performance Simulation, in: R. Issa, I. Flood (Eds.), Comput. Civ. Build. Eng., ASCE, Orlando, Florida, 2014: pp. 113–120. https://doi.org/https://doi.org/10.1061/9780784413616.015.

[100] L.G. Eads, R.A. Epperly, J.R.S. Jr, Practical Guide - Thermography, ASHRAE J. 42 (2000) 51–55.

[101] J. Rangel, S. Soldan, A. Kroll, 3D Thermal Imaging: Fusion of Thermography and Depth Cameras, in: 12th Int. Conf. Quant. Infrared Thermogr., Bordeaux, FR, 2014: pp. 1–10. https://doi.org/http://dx.doi.org/10.21611/qirt.2014.035 3D.

[102] Y. Ham, M. Golparvar-Fard, An automated vision-based method for rapid 3D energy performance modeling of existing buildings using thermal and digital imagery, Adv. Eng. Informatics. 27 (2013) 395–409. https://doi.org/10.1016/j.aei.2013.03.005.

[103] D. Lin, M. Jarzabek-Rychard, X. Tong, H.G. Maas, Fusion of thermal imagery with point clouds for building façade thermal attribute mapping, ISPRS J. Photogramm. Remote Sens. 151 (2019) 162–175. https://doi.org/10.1016/j.isprsjprs.2019.03.010.

[104] S. Lagüela, J. Armesto, H. González-Jorge, P. Arias, J. Herráez, Automation of thermographic 3D modelling through image fusion and image matching techniques, Proc. 28th Int. Symp. Autom. Robot. Constr. ISARC 2011. (2011) 818–822.

[105] Y. Ham, M. Golparvar-Fard, Rapid 3D Energy Performance Modeling of Existing Buildings using Thermal and Digital Imagery, in: H. Cai, A. Kandil, M. Hastak, P.S. Dunston (Eds.), Constr. Res. Congr., ASCE, West Lafayette, Indiana, 2012: pp. 991–1000. https://doi.org/9780784412329.

[106] M. Scaioni, E. Rosina, A. L’erario, L. Dìaz-Vilariño, Integration of infrared thermography & photogrammetric surveying of built landscape, Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch. 42 (2017) 153–160. https://doi.org/10.5194/isprs-Archives-XLII-5-W1-153-2017.

[107] S. Vidas, P. Moghadam, M. Bosse, 3D thermal mapping of building interiors using an RGB-D and thermal camera, Proc. - IEEE Int. Conf. Robot. Autom. (2013) 2311–2318. https://doi.org/10.1109/ICRA.2013.6630890.

[108] F. Javadnejad, D.T. Gillins, C.E. Parrish, R.K. Slocum, A photogrammetric approach to fusing natural colour and thermal infrared UAS imagery in 3D point cloud generation, Int. J. Remote Sens. 00 (2019) 1–27. https://doi.org/10.1080/01431161.2019.1641241.

[109] A. Adan, T. Prado, S.A. Prieto, B. Quintana, Fusion of thermal imagery and LiDAR data for generating TBIM models, Proc. IEEE Sensors. 2017-Decem (2017) 1–3. https://doi.org/10.1109/ICSENS.2017.8234261.

[110] D. Borrmann, A. Nüchter, M. Dakulović, I. Maurović, I. Petrović, D. Osmanković, J. Velagić, A mobile robot based system for fully automated thermal 3D mapping, Adv. Eng. Informatics. 28 (2014) 425–440. https://doi.org/10.1016/j.aei.2014.06.002.

[111] H. Macher, T. Landes, P. Grussenmeyer, From point clouds to building information models: 3D semi-automatic reconstruction of indoors of existing buildings, Appl. Sci. 7 (2017) 1–30. https://doi.org/10.3390/app7101030.

[112] Deep Design Studio, OUR SERVICES, Deep Des. Stud. (2018). https://deepdesign.studio/our-services/ (accessed March 15, 2020).

[113] ENGworks, Scan to BIM/REVIT point cloud modeling services, ENGworks. (n.d.). https://engworks.com/3d-laser-scanning-scan-to-bim/scan-to-bim/ (accessed March 15, 2020).

[114] Scan2bim, About Us, Scan2bim. (n.d.). https://scan2bim.com/ (accessed March 15, 2020).

[115] Scan To BIM, Get started with Scan To BIM wall, pipe, round duct, conduit extraction, Scan To BIM. (n.d.). https://scantobim.xyz/ (accessed March 15, 2020).

[116] Trimble, Trimble Edgewise, Trimble. (n.d.). https://geospatial.trimble.com/products-and-solutions/trimble-edgewise (accessed March 15, 2020).

[117] V. Patraucean, I. Armeni, M. Nahangi, J. Yeung, I. Brilakis, C. Haas, State of research in automatic as-built modelling, Adv. Eng. Informatics. 29 (2015) 162–171. https://doi.org/10.1016/j.aei.2015.01.001.

[118] J. Jung, S. Hong, S. Jeong, S. Kim, H. Cho, S. Hong, J. Heo, Productive modeling for development of as-built BIM of existing indoor structures, Autom. Constr. 42 (2014) 68–77. https://doi.org/10.1016/j.autcon.2014.02.021.

[119] D. Huber, B. Akinci, A. Adan, E. Anil, B. Okorn, X. Xiong, Methods for Automatically Modeling and Representing As-built Building Information Models, in: Proc. 2011 NSF Eng. Res. Innov. Conf., Atlanta, Georgia, 2011: p. 8.

[120] I. Brilakis, M. Lourakis, R. Sacks, S. Savarese, S. Christodoulou, J. Teizer, A. Makhmalbaf, Toward automated generation of parametric BIMs based on hybrid video and laser scanning data, Adv. Eng. Informatics. 24 (2010) 456–465. https://doi.org/10.1016/j.aei.2010.06.006.

[121] F. Goulette, Automatic CAD modeling of industrial pipes from range images, Proc. Int. Conf. Recent Adv. 3-D Digit. Imaging Model. (1997) 229–233. https://doi.org/10.1109/im.1997.603870.

[122] A.E. Johnson, R. Hoffman, J. Osborn, M. Hebert, System for semi-automatic modeling of complex environments, Proc. Int. Conf. Recent Adv. 3-D Digit. Imaging Model. (1997) 213–220.

https://doi.org/10.1109/im.1997.603868.

[123] F. Bosché, M. Ahmed, Y. Turkan, C.T. Haas, R. Haas, 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, Autom. Constr. 49 (2015) 201–213. https://doi.org/10.1016/j.autcon.2014.05.014.

[124] M. Nahangi, C.T. Haas, Automated 3D compliance checking in pipe spool fabrication, Adv. Eng. Informatics. 28 (2014) 360–369. https://doi.org/10.1016/j.aei.2014.04.001.

[125] U. Bauer, K. Polthier, Generating parametric models of tubes from laser scans, CAD Comput. Aided Des. 41 (2009) 719–729. https://doi.org/10.1016/j.cad.2009.01.002.

[126] A.K. Patil, P. Holi, S.K. Lee, Y.H. Chai, An adaptive approach for the reconstruction and modeling of as-built 3D pipelines from point clouds, Autom. Constr. 75 (2017) 65–78. https://doi.org/10.1016/j.autcon.2016.12.002.

[127] SUNY ESF, Illick Hall, SUNY Coll. Enviromental Sci. For. (n.d.). https://www.esf.edu/welcome/campus/illick.htm (accessed February 11, 2020).

[128] FLIR Systems Inc., FLIR T540, FLIR Syst. Inc. (2018). https://www.flir.com/products/t540/?model=79303-0201 (accessed February 12, 2020).

[129] Trimble Navigation Limited, Datasheet Trimble FX scanner, Triblem Navig. Ltd. (2010). https://www.solitech.bg/uploads/assets/Trimble-FX-Scanner.pdf (accessed February 8, 2020).

[130] 3Dflow, 3DFlow 3DF Zephyr: User Manual - Version 4.5, (2019) 109. http://3dflow.net/zephyr-doc/3DF Zephyr Manual 4.500 English.pdf.

[131] 3DFLOW, 3DF ZEPHYR, 3DFLOW. (2019). https://www.3dflow.net/3df-zephyr-pro-3d-models-from-photos/ (accessed February 25, 2020).

[132] MathWorks, MATLAB, MatWorks. (2019). https://www.mathworks.com/products/matlab/whatsnew.html (accessed March 2, 2020).

[133] Autodesk, RECAP, Autodesk Inc. (2020). https://www.autodesk.com/products/recap/overview (accessed March 2, 2020).

[134] Autodesk, Revit, Autodesk Inc. (2020). https://www.autodesk.com/products/revit/overview (accessed March 2, 2020).

[135] Dynamo, Dynamo, Dynamo. (2020). https://dynamobim.org.

[136] H. Huang, S. Wu, D. Cohen-Or, M. Gong, H. Zhang, G. Li, B. Chen, L1-Medial Skeleton of Point Cloud, ACM Trans. Graph. 32 (2013) 65–72.

[137] T_Pover, MEPover package published, (2016). https://forum.dynamobim.com/t/mepover-package-published/8305 (accessed March 2, 2020).

APPENDICES

Appendix A - Image resizing Script in MATLAB

```
% resizes RGB images extracted from thermal images in FLIR tools to match
% the size of the thermal images.

basename = "img (%d).jpg";       % input filenames
filename = "image (%d).jpg";     % output filenames

n = 1081;    % sets the number of images in folder

for i = 1:n
    I = imread(sprintf(basename,i));
    J = imresize(I,[960,1280]);
    imwrite(J,sprintf(filename,i));
end
```

Appendix B – Skeletonization Script in MATLAB

```
% Reads a point cloud (.pts file) containing pipes only and estimates
% the L1-Medial Skeleton of the Point Cloud

%% Creates a random sample of defined size for the skeleton
Q = dlmread('Resampled_pc_scene_3.pts');    % reads point cloud
Q(1,:) = [];    % deletes header
J = size(Q,1);  % stores size of point cloud Q
indQ = randperm(J);    % returns a random permutation of the indices of Q
maxSize = 750;        % stores maximum possible size defined by user
I = (min(fix(J/20),maxSize));  % stores the size of X (skeleton)
X = Q(indQ(1:I),1:3);  % stores the x,y,z coordinates of the skeleton nodes

% Parameter Definition

% boundBox stores the minimum and maximum values of x,y,z in Q
boundBox =
[min(Q(:,1)),min(Q(:,2)),min(Q(:,3));max(Q(:,1)),max(Q(:,2)),max(Q(:,3))];

h0 = 2*pdist(boundBox)/(J^(1/3)); % initial neighborhood size

% mu controls the global level of penalty applied on accumulated points;
mu = 0.35;      % Should be in the interval [0,1/2)

deltaH = h0/2;  % increment of h0 in successive iterations
hD = h0/2;      % supporting neighborhood parameter

% Density-based weighting
ptCloudQ = pointCloud(Q(:,1:3));    % stores a point cloud object for Q
dJ = zeros(J,1);    % creates an arrays of zeros for dJ

% for each qj in Q, calculates dJ
% there is no need to add "1" because the distance from the point to
% itfelf, "0", is stored in the dists array
for i = 1:J
    [indices,dists] = findNeighborsInRadius(ptCloudQ,Q(i,1:3),h0);
    thetaJ = exp(-(dists.^2)/((hD/2)^2));
    S = sum(thetaJ,'all');
    dJ(i,1) = S;
end

% iterate skeleton
newX = X;   % step needed for first iteration when X = newX

% it is possible to increase the radius of the sphere for each iteration.
% it is 0:0 here because of the complexity of the scenes. For the three
% scenes, if n was incremented, pipes that are relatively close would
% interfere with the estimation of the skeleton of one another.
% If it was n = 0:3, for example, the radius of the sphere would increase
% at each increment of n.
% The user is free to change this number, but n > 3 may not be advisable.
for n = 0:0
    error = ones(I,1);
    count = 0;
    while min(error)>=0.00005
        count = count + 1;
        if count>50
            break
        else
            X = newX;
            
            % calculates the directionality degree of xi within a local
neighborhood
            sigmaI = zeros(I,1);
            
            % for each xi in X, calculates sigma i
            % Ci stores a 3 × 3 weighted covariance matrix
            for i = 1:I
                Ci = zeros(3,3);
                for j = 1:I
                    if i~=j
                        difX = X(i,:) - X(j,:);
                        thetaI = exp(-((norm(difX))^2)/((h0/2)^2));
                        Cij = thetaI*(difX)'*difX;  % increment of Ci in each
iteration
                        Ci = Ci + Cij;  % Ci increases at each iteration
                    end
                end
                e = eig(Ci);    % stores the eigenvalues of the square matrix
Ci
                % sigmaI is the directionality degree of xi within a local
neighborhood
                sigmaI(i,1) = max(e)/(e(1,1)+e(2,1)+e(3,1));
            end
            
            % Calculates newX
            alphaTerm = zeros(I,3);
            betaTerm = zeros(I,3);
            
            % for each i in X, calculates the new coordinates of xi
            for i = 1:I
                sumQjAlphaij = [0,0,0];     % stores the sum of the product
of qj by alpha ij
                sumAlphaij = 0;             % stores the sum of alph ij
                [indices,dists] =
findNeighborsInRadius(ptCloudQ,X(i,:),2*h0);
                sumXiBetaii = [0,0,0];      % stores the sum of the product
of the difference of xi by beta ii
                sumBetaii = 0;              % stores the sum of beta ii
                for j = 1:size(indices,1)
                    difXiQj = X(i,:) - Q(indices(j,1),1:3);
                    if difXiQj~=0
                        aij = (exp(-
((norm(difXiQj))^2)/((h0/2)^2)))/(norm(difXiQj));
                        sumQjAlphaij = sumQjAlphaij +
aij*Q(indices(j,1),1:3)/dJ(indices(j,1));
                        sumAlphaij = sumAlphaij + aij/dJ(indices(j,1));
                    end
                end
                alphaTerm(i,:) = sumQjAlphaij/sumAlphaij;   % alpha term in
iteration
```

```
                for k = 1:I
                    if i~=k
                        difX = X(i,:) - X(k,:);
                        betaii = (exp(-
((norm(difX))^2)/((h0/2)^2)))/((norm(difX))^2);
                        sumXiBetaii = sumXiBetaii + betaii*difX;
                        sumBetaii = sumBetaii + betaii;
                    end
                end
                betaTerm(i,:) = sumXiBetaii/sumBetaii;    % beta term in
iteration
            end
            newX = alphaTerm + mu*sigmaI(:,1).*betaTerm;  % updates X
            for i = 1:I
                error(i,1) = norm(X(i,:) - newX(i,:));    % defined as the
displacement of the point
            end
        end
    end
end
```

Appendix C – Re-centering Script in MATLAB

```
% Recenters the skeleton nodes due to partial occlusion of pipes by
% changing the node's coordinates to the center of a circle that fits the
% 100 nearest points in the original point cloud relative to each node

skeleton = newX;        % the last newX is the estimated skeleton

% Creates a point cloud object with the points in skeleton
ptCloudX = pointCloud(skeleton);

% stores the updated coordinates of the skeleton
updatedSkeleton = skeleton;

% used to project the coordinates of the 100 points in the original point
% cloud to the reference plane used in the recentering process
projections = zeros(100,3);

% creates an array to store the radius of the pipe around each point in the
% skeleton
pointRadius = zeros(1,I);

%% Find Principal components direction and project points onto a plane

% for each point in the skeleton, finds 10 nearest neighbors in the
% skeleton and calculates principal component directions using the
% orthonormal frame formed by the eigenvectors of Ci.
for i = 1:I
    [indices,~] = findNearestNeighbors(ptCloudX,skeleton(i,:),11);
    Ci = zeros(3,3);
    for j = 1:size(indices,1)
        difX = skeleton(i,:) - skeleton(indices(j,1),:);
        Cij = (difX)'*difX;
        Ci = Ci + Cij;
    end
    [V,~] = eig(Ci);    % matrix V whose columns are the eigenvectors of Ci

    % for each point in the skeleton, find 100 nearest neighboors in the
    % original point cloud. Then project those points to a plane that is
    % orthogonal to the largest direction of variance.
    [indices,dists] = findNearestNeighbors(ptCloudQ,skeleton(i,:),100);
    for k = 1:size(indices,1)
        vec = ptCloudQ.Location(indices(k,1),:) - skeleton(i,:);
        projections(k,:) = ptCloudQ.Location(indices(k,1),:) -
dot(vec,(V(:,3))')*(V(:,3))';
    end

    % Recenters skeleton nodes by fitting a circle to the projected points
    % previously calculated.
    % saves new skeleton coordinates to updatedSkeleton.
    % saves the radius of the pipe around each skeleton node in pointRadius

    % The following piece of code was adapted from:
    % Sam Murthy (2020). Best fit 3D circle to a set of points
    %(https://www.mathworks.com/matlabcentral/fileexchange/55304-best-fit-3d-
circle-to-a-set-of-points),
    % MATLAB Central File Exchange. Retrieved May 11, 2020.
    % Copyright (c) 2016, Sam
    % Copyright (c) 1981, Izhak Bucher
    % All rights reserved.

    % Make sure to include circFit.m and RodriguesRotation.m functions to
    % your current folder.
    meanLoc = mean(projections);
    numCurPts = length(projections);
    movedToOrigin = projections - ones(numCurPts, 1)*meanLoc;
    [U, s, V] = svd(movedToOrigin);
    circleNormal = V(:, 3);
    circleLocsXY = RodriguesRotation(movedToOrigin, circleNormal, [0, 0, 1]);
    [xc, yc, radius] = circFit(circleLocsXY(:, 1), circleLocsXY(:, 2));
    centerLoc = RodriguesRotation([xc, yc, 0], [0, 0, 1], circleNormal) +
meanLoc;
    updatedSkeleton(i,:) = centerLoc;
    pointRadius(1,i) = radius;
end
```

Appendix D – Skeleton Segmentation Script in MATLAB

```
% Organizes the skeleton nodes to allow for the segmentation of skeleton
% nodes in subsequent steps

BoundBoxCenter = mean(boundBox);    % determines the center of bound box

% finds indices and distances from center of bound box to points in the
% skeleton
[indices,dists] = findNearestNeighbors(ptCloudX,BoundBoxCenter,I);
orderedSkeleton = zeros(I,3);    % creates the ordered skeleton array

% stores the most distant point from the center of the bound box to the
% skeleton point cloud in the last cell of the ordered skeleton array
orderedSkeleton(I,:) = updatedSkeleton(indices(I,:),:);

% cleans the most distant point from the center of the bound box to the
% skeleton point cloud in the updated skeleton array
updatedSkeleton(indices(I,:),:) = [];

% creates an auxiliary point cloud using the new updated skeleton array
ptCloudAux = pointCloud(updatedSkeleton);

% for each point in the skeleton, gets the closest point to the last point
% in the ordered skeleton array, adds it to the ordered skeleton array, and
% removes it from the updated skeleton array. Then, creates a new auxiliary
% point cloud using the new updated skeleton array
for i = I-1:-1:2
    [indices,dists] =
findNearestNeighbors(ptCloudAux,orderedSkeleton(i+1,:),1);
    orderedSkeleton(i,:) = updatedSkeleton(indices,:);
    updatedSkeleton(indices,:) = [];
    ptCloudAux = pointCloud(updatedSkeleton);
end

% saves the last point in updatedSkeleton to orderedSkeleton
orderedSkeleton(1,:) = updatedSkeleton;

% creates a new point cloud using the ordered skeleton nodes
ptCloudSkel = pointCloud(orderedSkeleton);

% Segment Point Cloud
% creates a matrix to store whether two points in the point cloud are
% connected or not
Connected = zeros(I);

% for each point in the skeleton, find the two nearest neighbors, and
% creates two vectors from the point to each of the two nearest neighbors.
% Then, calculates the angle between the two vectors.
% If the angle is smaller than the set threshold, set connected to true.
for i = 1:I
    [indices,dists] =
findNearestNeighbors(ptCloudSkel,ptCloudSkel.Location(i,:),3);
    v1 = ptCloudSkel.Location(i,:) - ptCloudSkel.Location(indices(2),:);
    v2 = ptCloudSkel.Location(i,:) - ptCloudSkel.Location(indices(3),:);
    angle = atan2(norm(cross(v1,v2)),dot(v1,v2));
    if sin(angle) < sin(pi/4)   % threshold angle is set to PI/4
        for j = 1:3
            for k = 1:3
                if j~=k
                    Connected(indices(j),indices(k)) = 1;
                end
            end
        end
    end
end

pointConnections = sum(Connected,2);   % stores the number of connections
each point makes

% Segment skeleton
% separates the points into segments using the number of connections each
% point makes and their relative position in the pointConnections array
segments = zeros(I,1);
for i = 1:I
    for j = i:I
        if pointConnections(j,1)==4 && segments(j,1)==0
            segments(j,1) = i;
            for k = j:I
                if pointConnections(k,1)==4 && segments(k,1)==0 &&
Connected(j,k)==1
                    segments(k,1) = segments(j,1);
                end
            end
        end
    end
end

% identifies each segment by a number by updating the segments array
segNumber = 0;
for i = 1:I-1
    if segments(i,1)<segments(i+1,1)
        segNumber = segNumber + 1;
    elseif segments(i,1)~=0
        segments(i,1) = segNumber;
    end
end

% sets all the segments that have less than three nodes to 0. It is a way
% to remove noise
for i = 1:max(segments)
    if sum(segments==i)<3
        segments(segments==i) = 0;
    end
end
```

Appendix E – Connection Identification Script in MATLAB

```matlab
% calculates the mean and standard deviation of the segments
segMean = zeros(max(segments),3);   % mean coordinates of each segment
segStdv = zeros(max(segments),3);   % sandard deviation of the coordinates of
each segment

% calculates the mean coordinates of each segment and the standard
% deviation of the coordinates of each segment
for i = 1:max(segments)
    if nnz(segments==i)
        indices = find(segments==i);
        segMean(i,1) = mean(ptCloudSkel.Location(indices,1));
        segStdv(i,1) = std(ptCloudSkel.Location(indices,1));
        segMean(i,2) = mean(ptCloudSkel.Location(indices,2));
        segStdv(i,2) = std(ptCloudSkel.Location(indices,2));
        segMean(i,3) = mean(ptCloudSkel.Location(indices,3));
        segStdv(i,3) = std(ptCloudSkel.Location(indices,3));
    end
end

% finds the extremities of each segment
extremities = zeros(max(segments),2);
for i = 1:max(segments)
    indices = find(segments==i);
    distanceMatrix = squareform(pdist(ptCloudSkel.Location(indices,:)));
    maximumDist = max(max(distanceMatrix));
    [row, column] = find(distanceMatrix == maximumDist);
    if nnz(segments==i)
        extremities(i,1) = indices(column(1,1),1);
        extremities(i,2) = indices(row(1,1),1);
    end
end

% replaces the two coordinates with the smallest standard deviation in each
% point by the mean value of these coordinates within each segment
alignedSkeleton = orderedSkeleton;
for i = 1:max(segments)
    if nnz(segments==i)
        indices = find(segments==i);
        if max(segStdv(i,:))==segStdv(i,1)
            for j = 1:size(indices)
                alignedSkeleton(indices(j),2) = segMean(i,2);
                alignedSkeleton(indices(j),3) = segMean(i,3);
            end
        elseif max(segStdv(i,:))==segStdv(i,2)
            for j = 1:size(indices)
                alignedSkeleton(indices(j),1) = segMean(i,1);
                alignedSkeleton(indices(j),3) = segMean(i,3);
            end
        else
            for j = 1:size(indices)
                alignedSkeleton(indices(j),1) = segMean(i,1);
                alignedSkeleton(indices(j),2) = segMean(i,2);
            end
        end
    end
end

% stores the mean diameter of each segment based on the radius of the pipe
% determined during recentering
segDiameter = extremities;
for i = 1:size(extremities,1)
    if nnz(segments==i)
        indices = find(segments==i);
        segDiameter(i,3) = 2*mean(pointRadius(indices));
    end
end

% calculates the new segment means by determining which segments are
% coplanar. Due to variations in the mean values of the coordinates, it
% assumes that two segments are coplanar if the differences between the two
% coordinates with the smallest standard deviation in both segments are
% less than the sum of the radius of the two segments.
newSegMean = segMean;
for i = 1:3
    usedSegment = zeros(size(extremities,1),1);
    for j = 1:size(extremities,1)-1
        samePlane = zeros(size(extremities,1),1);
        if usedSegment(j,1)==1
            continue
        end
        usedSegment(j,1) = 1;
        if segStdv(j,i)==max(segStdv(j,:))
            continue
        end
        if segDiameter(j,:)==0
            continue
        end
        samePlane(j,1) = segMean(j,i);
        for k = j+1:size(extremities,1)
            if segDiameter(k,:)==0
                continue
            end
            if segStdv(k,i)==max(segStdv(k,:))
                continue
            end
            difSegMean = abs(segMean(j,i)-segMean(k,i));
            sumRadii = (segDiameter(j,3)+segDiameter(k,3))/2; %divide by 2!!!
            if difSegMean<sumRadii && usedSegment(k,1)==0
                usedSegment(k,1) = 1;
                samePlane(k,1) = segMean(k,i);
            end
        end
        meanCoord = mean(nonzeros(samePlane));
        for h = 1:size(extremities,1)
            if samePlane(h,1)~=0
                newSegMean(h,i) = meanCoord;
            end
        end
    end
end
```

```
% aligns the segments using the new mean coordinates of each segment
for i = 1:3
    for j = 1:size(newSegMean,1)
        if newSegMean(j,i)~=segMean(j,i)
            indices = find(segments==j);
            for k = 1:size(indices,1)
                alignedSkeleton(indices(k,1),i) = newSegMean(j,i);
            end
        end
    end
end


%% Connection identification

% Tees
% identifies candidates to Tee connections by identifiying the skeleton
% nodes that make more than 4 connections
indices = find(pointConnections>4); % indices of nodes that make more than 4
connections
teeCandidates = zeros(size(indices,3));
uniqueExt = unique(extremities);    % return unique indices of extremities in
the skeleton
uniqueExt = setdiff(uniqueExt,0);   % removes zeros from uniqueExt
indicesTeesExt = [indices;uniqueExt];   % array of indices of tee candidates
and extremities
distanceMatrix = squareform(pdist(ptCloudSkel.Location(indicesTeesExt,:)));

% sets distances between tee candidates to 0
for i = 1:size(indices)
    for j = 1:size(indices)
        distanceMatrix(i,j) = 0;
    end
end

% gets the three nearest extremities relative to each tee candidate
for i = 1:size(indices)
    rowDistMat = distanceMatrix(i,:);   % stores row of distanceMatrix
    uniqueRowDistMat = unique(rowDistMat);  % unique elements of rowDistMat

    % gets the three nearest extremities relative to each tee candidate
    for j = 2:4
        minimumDist = uniqueRowDistMat(1,j);
        [row, column] = find(distanceMatrix == minimumDist);
        teeCandidates(i,j-1) = indicesTeesExt(row(1,1),1);
    end
end

% tests each of the tee candidates to find duplicates. If more than one tee
% candidate have the same three extremities, sets all tee candidates but
% one to [0,0,0]
for i = 1:size(teeCandidates,1)-1
    vector1 = teeCandidates(i,:);
    for j = i+1:size(teeCandidates,1)
        vector2 = teeCandidates(j,:);
        if vector1==vector2
            teeCandidates(j,:) = [0,0,0];
        end
    end
end

% deletes all tee candidates that have indices 0
[row,~] = find(teeCandidates(:,1)==0);
for i = size(row,1):-1:1
    teeCandidates(row(i,1),:) = [];
end

% sets the tee coordinates based on the three extremities it is linked to.
% It considers that the tee coordinates are the same as the ones of the
segments
% that have the smallest standard deviations.
teesCoord = zeros(size(teeCandidates,1),3);
vector1 = zeros(1,3);
for i = 1:size(teeCandidates,1)
    for j = 1:3
        [row,column] = find(extremities==teeCandidates(i,j));
        vector1(1,j) = row(1,1);
    end
    for k = 1:3
        for h = 1:3
            if segStdv(vector1(1,k),h)~=max(segStdv(vector1(1,k),:))
                teesCoord(i,h) = newSegMean(vector1(1,k),h);
            end
        end
    end
end

% Deletes false connections
% calculates three vectors, from the tee candidate to each potential
% extremity, and calculates the triple product of them. If the triple
% product is different than zero, i.e. non-coplanar points, sets the
% indices of the extremities in the tee candidates array to zero.
for i = 1:size(teeCandidates)
    vector1 = teesCoord(i,:)-alignedSkeleton(teeCandidates(i,1),:);
    vector2 = teesCoord(i,:)-alignedSkeleton(teeCandidates(i,2),:);
    vector3 = teesCoord(i,:)-alignedSkeleton(teeCandidates(i,3),:);
    tripleProduct = dot(vector1,cross(vector2,vector3));
    if tripleProduct~=0
        teeCandidates(i,:) = [0,0,0];
    end
end

% if the indices in tee candidates are zero, set tee coordinates to zero
for i = 1:size(teeCandidates)
    if teeCandidates(i,1)==0
        teesCoord(i,:) = [0,0,0];
    end
end

% deletes the rows containing zeros in tee candidates and tee coordinates
[row,~] = find(teeCandidates(:,1)==0);
for i = size(row,1):-1:1
    teeCandidates(row(i,1),:) = [];
    teesCoord(row(i,1),:) = [];
```

```
end

% Elbows
% gets extremities not involved in tee connections
remainingExt = setdiff(uniqueExt,teeCandidates);

% calculates distances between pais of extremities in remaining extremities
distanceMatrix = squareform(pdist(ptCloudSkel.Location(remainingExt,:)));

% temporarily stores pais of extremities to test elbow connections
testConnections = [0,0];

% if a given pair of extremities belongs to a single segment, sets
% distances between extremities to zero in distanceMatrix
for i = 1:size(remainingExt,1)
    testConnections(1,1) = remainingExt(i);
    for j = 1:size(remainingExt,1)
        testConnections(1,2) = remainingExt(j);
        for k = 1:size(extremities,1)
            if testConnections==extremities(k,:)
                distanceMatrix(i,j) = 0;
                distanceMatrix(j,i) = 0;
            end
        end
    end
end

% creates the elbow candidates using the remaining extremities by setting
% the pair of extremities in an elbow as the pair of extremities with the
% minimum distance between them. Also, tests if a pair of extremities is in
% a single segment. If that is the case, set them to zero.
difElbowExt = remainingExt;
elbowCandidates = zeros(fix(size(remainingExt,1)/2),2);
for i = 1:size(remainingExt,1)
    if size(distanceMatrix,1)<2
        break
    elseif size(distanceMatrix,1)>=2 & max(distanceMatrix)>0
        minimumDist = min(min(distanceMatrix(distanceMatrix>0)));
        [row,column] = find(distanceMatrix==minimumDist);
        elbowCandidates(i,:) =
[difElbowExt(column(1,1),1),difElbowExt(row(1,1),1)];
        uniqueElbowCandidates = unique(elbowCandidates(elbowCandidates~=0));
        difElbowExt = setdiff(remainingExt,uniqueElbowCandidates);
        distanceMatrix =
squareform(pdist(ptCloudSkel.Location(difElbowExt,:)));
        for j = 1:size(difElbowExt,1)
            testConnections(1,1) = difElbowExt(j);
            for k = 1:size(difElbowExt,1)
                testConnections(1,2) = difElbowExt(k);
                for h = 1:size(extremities,1)
                    if testConnections==extremities(h,:)
                        distanceMatrix(j,k)=0;
                        distanceMatrix(k,j)=0;
                    end
                end
            end
        end
    end
end

% cleans elbow candidates that have index 0
for i = size(elbowCandidates):-1:1
    if elbowCandidates(i,1)==0 && elbowCandidates(i,2)==0
        elbowCandidates(i,:) = [];
    end
end

% sets the coordinates of the elbows using the mean values and standard
% deviations of the of the segments involved in the the connection.
elbowsCoord = zeros(size(elbowCandidates,1),3);
vector1 = zeros(1,2);
for i = 1:size(elbowCandidates,1)
    for j = 1:2
        [row,column] = find(extremities==elbowCandidates(i,j));
        vector1(1,j) = row(1,1);
    end
    for k = 1:3
        if segStdv(vector1(1,1),k)~=max(segStdv(vector1(1,1),:))
            elbowsCoord(i,k) = newSegMean(vector1(1,1),k);
        end
    end
    for h = 1:3
        if elbowsCoord(i,h)==0
            elbowsCoord(i,h) = newSegMean(vector1(1,2),h);
        end
    end
end

% deletes elbows that have the same location. It is used in cases when the
% two extremities of one segment are matched to the two extremities of
% another segment. In this case, the coordinates of the two potential
% elbows are the same but the distances from the connection to its
% respective extremities are different. It deletes the one with the longest
% distance.
for i = 1:size(elbowCandidates,1)-1
    vector1 = elbowCandidates(i,:);
    ec1 = elbowsCoord(i,:);     % elbow coordinates number 1
    for j = i+1:size(elbowCandidates)
        vector2 = elbowCandidates(j,:);
        ec2 = elbowsCoord(j,:); % elbow coordinates number 2
        if ec1==ec2
            v1 = ptCloudSkel.Location(vector1(1,1),:);  % vector number 1
            v2 = ptCloudSkel.Location(vector1(1,2),:);  % vector number 2
            v3 = ptCloudSkel.Location(vector2(1,1),:);  % vector number 3
            v4 = ptCloudSkel.Location(vector2(1,2),:);  % vector number 4
            dist12 = pdist2(v1,v2);     % distance from v1 to v2
            dist34 = pdist2(v3,v4);     % distance from v3 to v4
            if dist12>dist34
                elbowsCoord(i,:) = [0,0,0];
            elseif dist12<dist34
                elbowsCoord(j,:) = [0,0,0];
            end
        end
    end
```

```
end

% delete false connections. In this case, if the angles formed between the
% vectors from the cooenection to each of the extremities is not 0° or 180°,
% for couplings, or 90°, for elbows, the elbow candidate is set to 0.
for i = 1:size(elbowsCoord,1)
    vector1 = elbowsCoord(i,:) - alignedSkeleton(elbowCandidates(i,1),:);
    vector2 = elbowsCoord(i,:) - alignedSkeleton(elbowCandidates(i,2),:);
    cosTheta = dot(vector1,vector2)/(norm(vector1)*norm(vector2));
    if cosTheta~=-1 && cosTheta~=0 && cosTheta~=1
        elbowsCoord(i,:) = [0,0,0];
    end
end

% delete false connections. A threshold is set to the maximum distance
% between the elbow to each of its extremities. If the distance is greater
% than the threshold, sets the coordinates to zero.
for i = 1:size(elbowCandidates,1)
    if elbowsCoord(i,:)~=0
        [~,column] = find(Connected(elbowCandidates(i,1),:)==1);
        for j = 1:size(column,2)
            if pointConnections(column(1,j))==4
                column(1,j) = 0;
            end
        end
        vector1 = column(column>0);
        [~,column] = find(Connected(elbowCandidates(i,2),:)==1);
        for j = 1:size(column,2)
            if pointConnections(column(1,j))==4
                column(1,j) = 0;
            end
        end
        vector2 = column(column>0);
        [row,~] = find(extremities==elbowCandidates(i,1));
        diam1 = segDiameter(row(1,1),3);
        [row,~] = find(extremities==elbowCandidates(i,2));
        diam2 = segDiameter(row(1,1),3);
        thres = 2*(diam1 + diam2);      % threshold equals 2 times the sum of
the diameters
        vector3 = [vector1,vector2];
        testElbCoord = zeros(1 + size(vector1,2) + size(vector2,2),3);
        testElbCoord(1,:) = elbowsCoord(i,:);
        for j = 2:size(testElbCoord,1)
            testElbCoord(j,:) = alignedSkeleton(vector3(j-1),:);
        end
        if size(testElbCoord,1)>1
            distanceMatrix = squareform(pdist(testElbCoord));
            if thres<max(distanceMatrix(1,:))
                elbowsCoord(i,:) = [0,0,0];
            end
        end
    end
end
```

Appendix F – System Classification Script in MATLAB

```
% classifies the segments into their different
ironPalette = dlmread('Ironbow_Palette.txt'); % reads ironbow palette file
thermal = dlmread('Thermal_pc_scene_3.pts');  % reads thermal pt cloud file
thermal(1,:) = [];  % deletes header of thermal point cloud file
ptCloudIron = pointCloud(ironPalette);  % creates point cloud of iron palette
ptCloudThermal = pointCloud(thermal(:,1:3)); %create pt cloud of thermal file
temperatureVector = zeros(size(ironPalette,1),1); % stores conversion of RGB
to temperature values

% The user needs to set these values based on site's conditions
%Scene 1
% rangeTempMin = 40;      % minimum temperature on camera
% rangeTempMax = 260;     % maximum temperature on camera
% referenceTemp = 80;     % reference temperature based on site's conditions
% % stores array of pipe system names
% pipeSystems = ["Low Pressure Steam";"Domestic Cold Water";"Hot Water
Condensate"];
% tempMin = [180;40;105];    % minimum system's temperature (measured)
% tempMax = [260;60;125];    % maximum system's temperature (measured)

% Scene 2
% rangeTempMin=70;
% rangeTempMax=240;
% referenceTemp=90;
% pipeSystems=["Low Pressure Steam";"Glycol Supply";"Low Pressure
Condensate"];
% tempMin=[212;135;110];
% tempMax=[240;160;135];

%Scene 3
rangeTempMin=70;
rangeTempMax=360;
referenceTemp=100;
pipeSystems=["Medium Pressure Steam";"Condensate";];
tempMin=[212;120];
tempMax=[360;212];

% stores system name and associated temperatures
pipeSystemMatrix = [pipeSystems,tempMin,tempMax];

% calculates temperature increment for each color in Ironbow Palette
tempIncrement = (rangeTempMax - rangeTempMin)/(size(ironPalette,1) - 1);

% fills temperature vector using temperature ranges and temperature
% increment
for i = 0:size(ironPalette,1)-1
    temperatureVector(i+1,1) = rangeTempMin + i*tempIncrement;
end

% classifies pipes into systems
segmentID = setdiff(unique(segments),0);    % segment IDs
```

```
% stores minimum segment temperature
% set a value higher than the temp range
minSegTemp = zeros(size(segmentID,1),1) + 1000;

% stores maximum segment temperature
maxSegTemp = zeros(size(segmentID,1),1);

% stores names of classified pipes
nameClassPipe = strings(size(segmentID,1),1);

% stores segment ID and temperature range
segTempRange = [segmentID,minSegTemp,maxSegTemp];

% fills segTempRange by finding the temperatures of the nearest 50 points
% in the thermal point cloud.
for i = 1:size(alignedSkeleton,1)
    segAux = segments(i,1);
    row = find(segmentID==segAux);
    if segAux>0
        [indices1,~] =
findNearestNeighbors(ptCloudThermal,alignedSkeleton(i,:),50);
        for j = 1:50
            [indices2,~] =
findNearestNeighbors(ptCloudIron,thermal(indices1(j,1),5:7),1);
            pointTemp = temperatureVector(indices2,1);
            if pointTemp<segTempRange(row,2)
                segTempRange(row,2) = pointTemp;
            end
            if pointTemp>segTempRange(row,3)
                segTempRange(row,3) = pointTemp;
            end
        end
    end
end

% stores the set temperature of the pipe to classify it
setTemp = zeros(size(segmentID,1),1);

% verifies which temperature (max or min) is more distant from the
% reference temperature to determine the set temperature for the pipe
for i = 1:size(segmentID,1)
    if abs(segTempRange(i,2) - referenceTemp)<abs(segTempRange(i,3) -
referenceTemp)
        setTemp(i,1) = segTempRange(i,3);
    else
        setTemp(i,1) = segTempRange(i,2);
    end
end

% if set temperature is whithin the range of one of the systems, classify
% the pipe as belonging to that system
for i = 1:size(segTempRange,1)
    for j = 1:size(pipeSystemMatrix,1)
        if setTemp(i,1)>tempMin(j,1) && setTemp(i,1)<tempMax(j,1)
            nameClassPipe(i,1) = pipeSystems(j,1);
        end
    end
```

```
end

% if set temperature is not in any range, classify the system as "Other"
for i = 1:size(segTempRange,1)
    if nameClassPipe(i,1)==""
        nameClassPipe(i,1) = 'Other';
    end
end
```

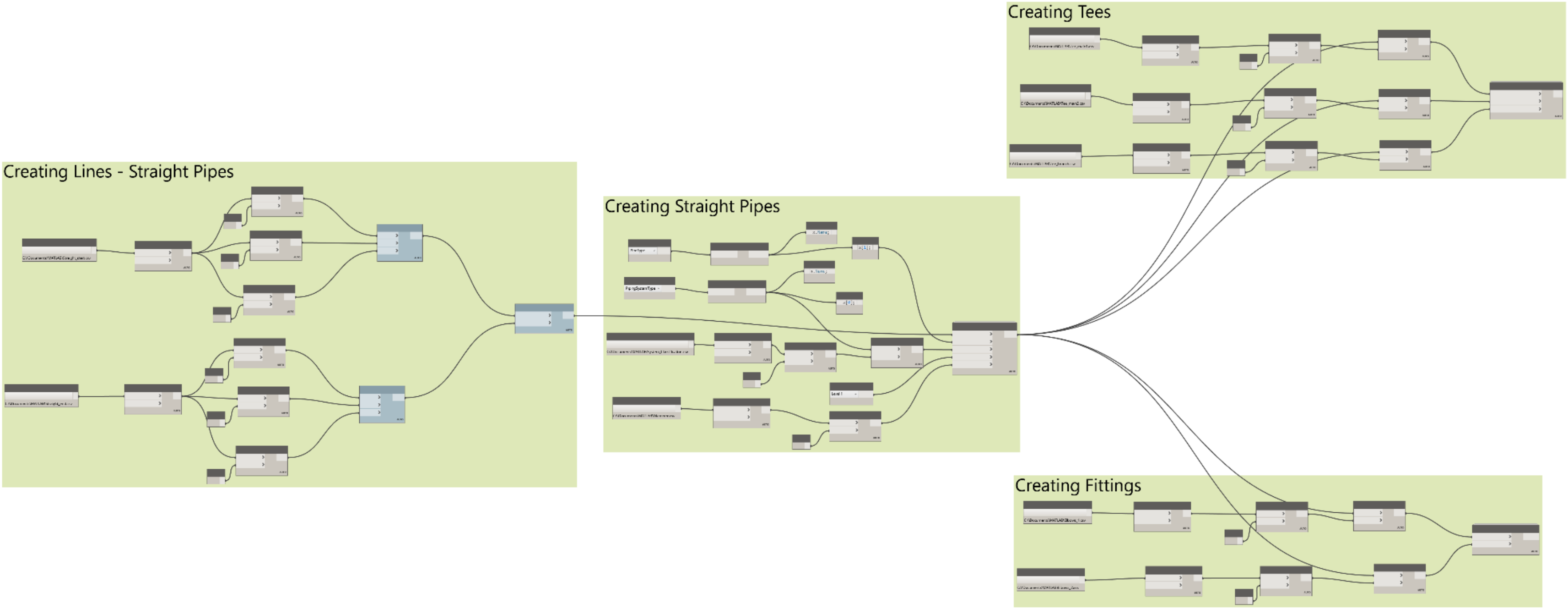
Appendix G - MATLAB to Dynamo Export Script in MATLAB

```
% Clean zeros in extremities
[row,~] = find(extremities(:,1)==0);
for i = size(row,1):-1:1
    extremities(row(i,1),:)=[];
end

% Straight Pipes
% exports csv files containing the extremities (start and end) of each
straight pipe
straightStart = zeros(size(extremities,1),3);
straightEnd = zeros(size(extremities,1),3);
for i = 1:size(extremities,1)
    if extremities(i,1)~=0 && extremities(i,2)~=0
        straightStart(i,:) = alignedSkeleton(extremities(i,1),:);
        straightEnd(i,:) = alignedSkeleton(extremities(i,2),:);
    end
end

csvwrite('Straight_start.csv',straightStart);
csvwrite('Straight_end.csv',straightEnd);

% Tees
% exports csv files containing the extremities of each tee as tee main 1,
% tee main 2, and tee branch
teeMain1 = zeros(size(teeCandidates,1),1);
teeMain2 = zeros(size(teeCandidates,1),1);
teeBranch = zeros(size(teeCandidates,1),1);

% defines which segments are tee main 1, tee main 2, and tee branch based
% on the angle between the vectors from the tee to each extremity
cosTheta = zeros(1,3);
for i = 1:size(teeCandidates,1)
    vector1 = alignedSkeleton(teeCandidates(i,1),:) - teesCoord(i,:);
    vector2 = alignedSkeleton(teeCandidates(i,2),:) - teesCoord(i,:);
    vector3 = alignedSkeleton(teeCandidates(i,3),:) - teesCoord(i,:);
    cosTheta(1,1) = dot(vector1,vector2)/(norm(vector1)*norm(vector2));
    cosTheta(1,2) = dot(vector1,vector3)/(norm(vector1)*norm(vector3));
    cosTheta(1,3) = dot(vector2,vector3)/(norm(vector2)*norm(vector3));
    if cosTheta(1,1)~=0
        [row,~] = find(extremities==teeCandidates(i,3));
        teeBranch(i,1) = row(1,1)-1;
        [row,~] = find(extremities==teeCandidates(i,1));
        teeMain1(i,1) = row(1,1)-1;
        [row,~] = find(extremities==teeCandidates(i,2));
        teeMain2(i,1) = row(1,1)-1;
    elseif cosTheta(1,2)~=0
        [row,~] = find(extremities==teeCandidates(i,2));
        teeBranch(i,1) = row(1,1)-1;
        [row,~] = find(extremities==teeCandidates(i,1));
        teeMain1(i,1) = row(1,1)-1;
        [row,~] = find(extremities==teeCandidates(i,3));
        teeMain2(i,1) = row(1,1)-1;
    else
```

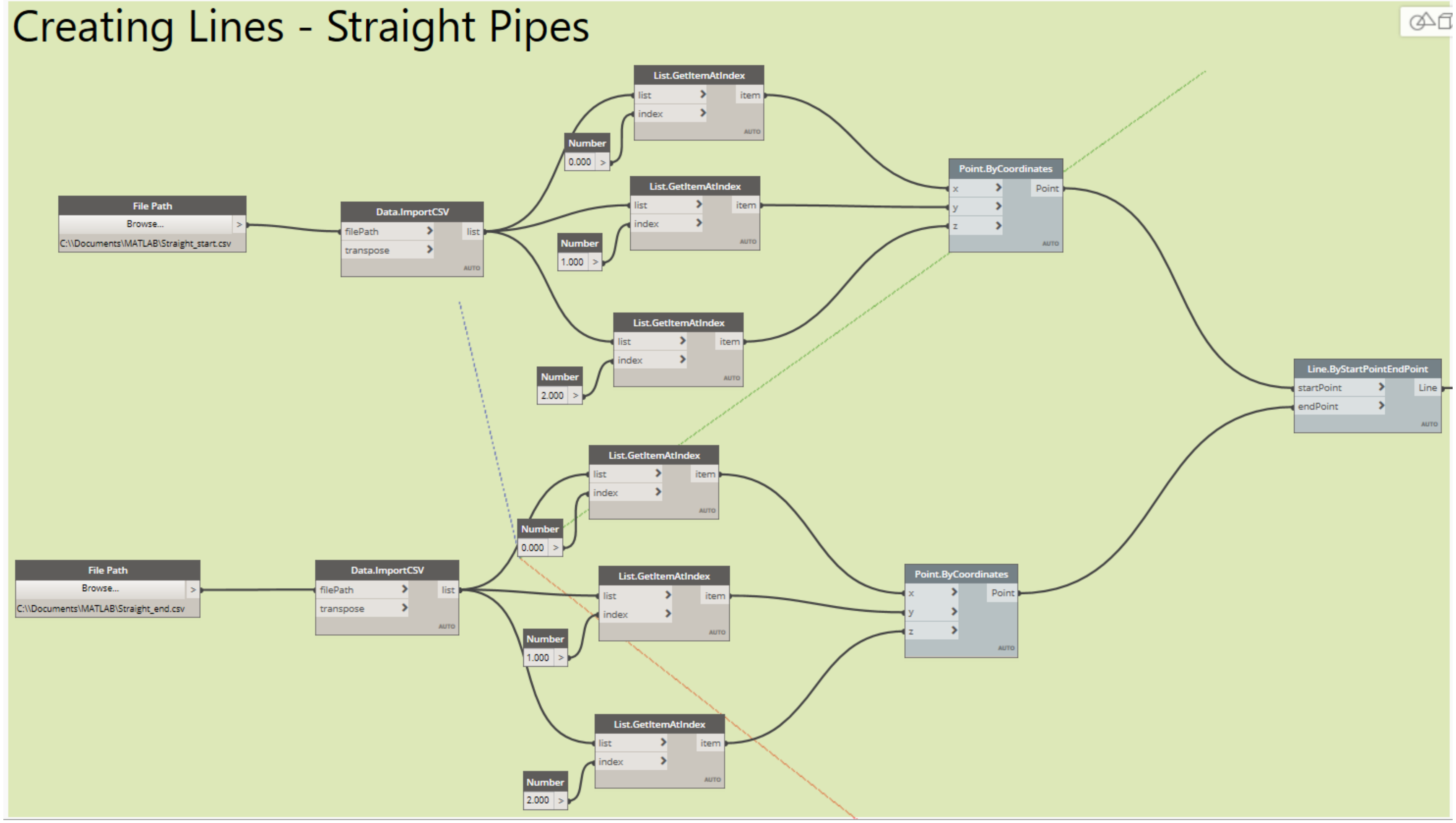
```
        [row,~] = find(extremities==teeCandidates(i,1));
        teeBranch(i,1) = row(1,1)-1;
        [row,~] = find(extremities==teeCandidates(i,2));
        teeMain1(i,1) = row(1,1)-1;
        [row,~] = find(extremities==teeCandidates(i,3));
        teeMain2(i,1) = row(1,1)-1;
    end
end

csvwrite('Tee_main1.csv',teeMain1);
csvwrite('Tee_main2.csv',teeMain2);
csvwrite('Tee_branch.csv',teeBranch);

% Elbows
% exports csv files containing the extremities of each elbow
elbow1 = zeros(size(elbowCandidates,1),1);
elbow2 = zeros(size(elbowCandidates,1),1);
for i = 1:size(elbowCandidates,1)
    [row,~] = find(extremities==elbowCandidates(i,1));
    elbow1(i,1) = row(1,1)-1;
    [row,~] = find(extremities==elbowCandidates(i,2));
    elbow2(i,1) = row(1,1)-1;
end
% deletes empty elbows
for i = size(elbowCandidates,1):-1:1
    if elbowsCoord(i,:)==0
        elbow1(i,:) = [];
        elbow2(i,:) = [];
    end
end

csvwrite('Elbows_1.csv',elbow1);
csvwrite('Elbows_2.csv',elbow2);

% Diameter
% exports csv files containing the diameter of each straight segment
pvcDiam = dlmread('pvc_diameter.txt'); % reads pvc diameter file
diameter = 1000*segDiameter(:,3);   % converts diameter from m to mm
[row,~] = find(diameter(:,1)==0);
% deletes diameter rows equals zero
for i = size(row,1):-1:1
    diameter(row(i,1),:) = [];
end
% sets the segment diameter to the closest commercial diameter in pvcDiam
for i = 1:size(diameter,1)
    absoluteDif = abs(diameter(i,1) - pvcDiam(:,1));
    minimum = min(absoluteDif);
    [row,~] = find(absoluteDif==minimum);
    diameter(i,1) = pvcDiam(row(1,1));
end

csvwrite('Diameter.csv',diameter);

% Pipe Classification
% exports csv files containing the pipes' classifications
revitClassName = zeros(size(nameClassPipe,1),1);
```

```
systemMatrix = ["Hydronic Supply"; "Hydronic Return"; "Sanitary"; "Domestic Hot Water"; "Domestic Cold Water";...
    "Fire Protection Wet"; "Fire Protection Dry"; "Fire Protection Pre-Action"; "Fire Protection Other"; "Other";...
    "Vent"; "Medium Pressure Steam"; "Low Pressure Steam"; "Condensate"; "Low Pressure Condensate";...
    "Hot Water Condensate"; "Glycol Supply"; "Glycol Return"];
for i = 1:size(nameClassPipe,1)
    index = find(systemMatrix==nameClassPipe(i,1));
    revitClassName(i,1) = index-1;
end

csvwrite('System_Classification.csv',revitClassName);
```

Appendix H - Revit Dynamo Script – Point to Pipe



The following four images are just zoomed views for each of the four modules presented in this picture.



Creating Lines - Straight Pipes

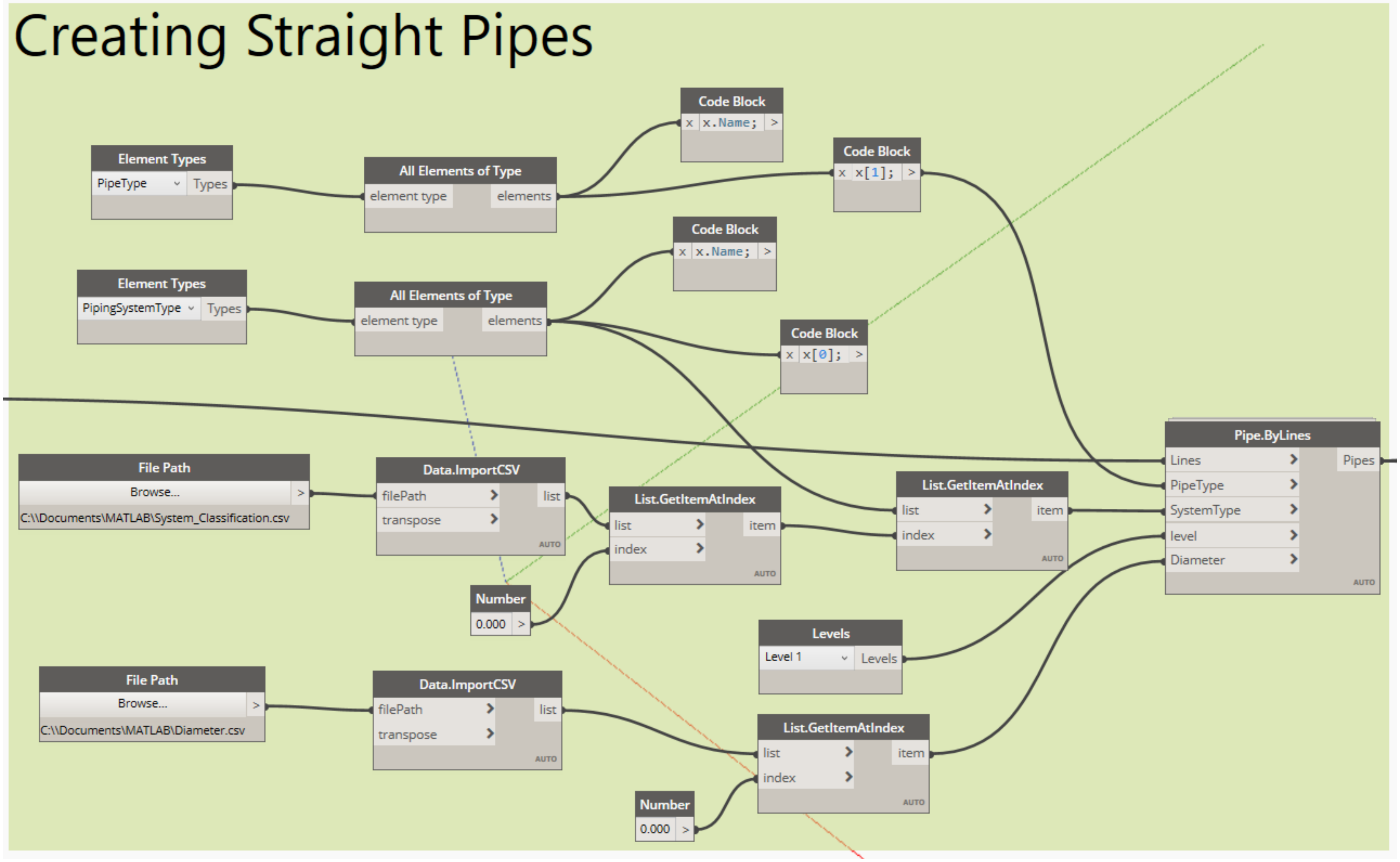
File Path: C:\\Documents\MATLAB\Straight_start.csv

File Path: C:\\Documents\MATLAB\Straight_end.csv

Number: 0.000

Number: 1.000

Number: 2.000



Creating Straight Pipes

Element Types: PipeType

Element Types: PipingSystemType

Code Block:
```
x.Name;
```

Code Block:
```
x[1];
```

Code Block:
```
x.Name;
```

Code Block:
```
x[0];
```

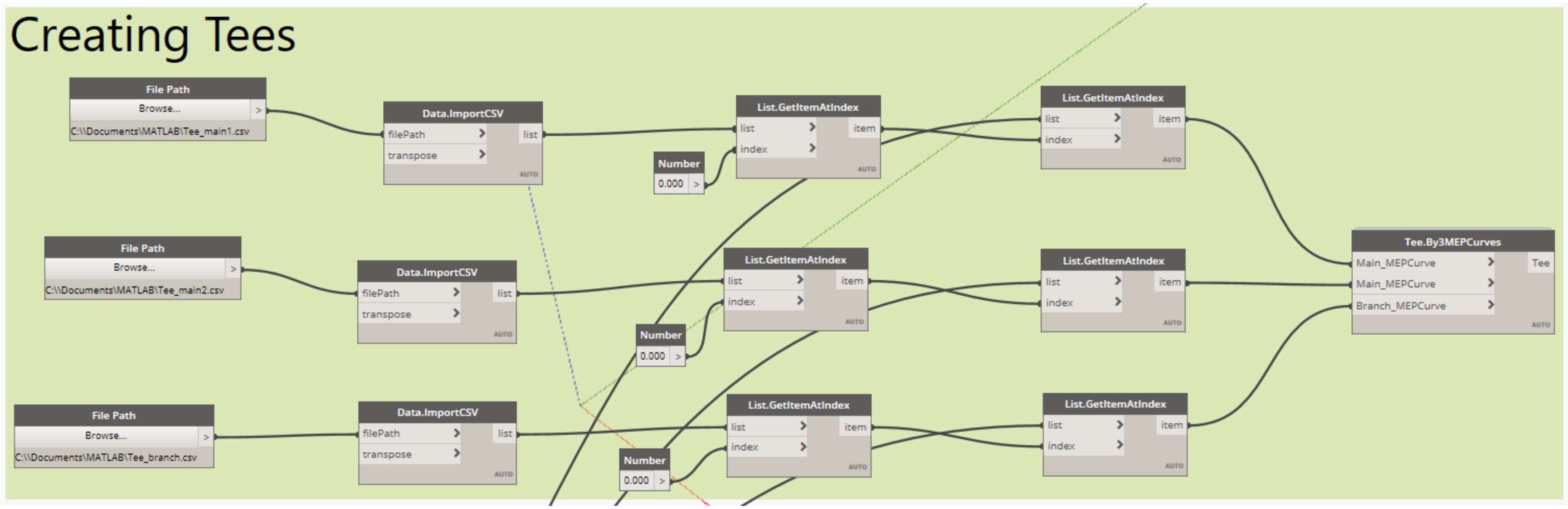
File Path: C:\\Documents\MATLAB\System_Classification.csv

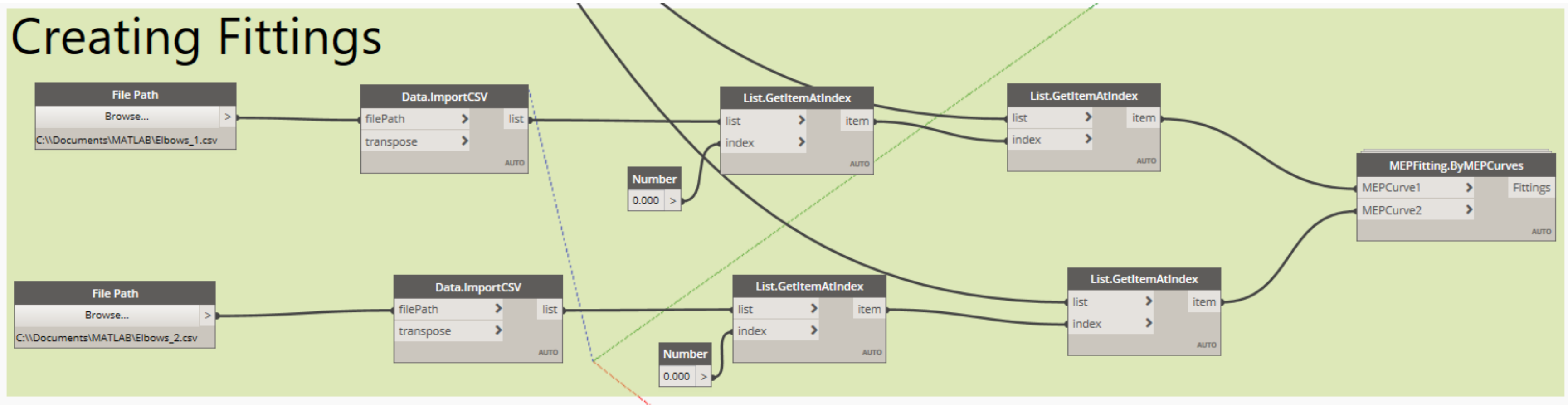
File Path: C:\\Documents\MATLAB\Diameter.csv

Number: 0.000

Number: 0.000

Levels: Level 1


Creating Tees
File Path
Browse...
C:\\Documents\MATLAB\Tee_main1.csv
File Path
Browse...
C:\\Documents\MATLAB\Tee_main2.csv
File Path
Browse...
C:\\Documents\MATLAB\Tee_branch.csv
Data.ImportCSV
filePath
transpose
list
AUTO
Number
0.000
List.GetItemAtIndex
list
index
item
AUTO
Tee.By3MEPCurves
Main_MEPCurve
Main_MEPCurve
Branch_MEPCurve
Tee
AUTO


Creating Fittings
File Path
Browse...
C:\\Documents\MATLAB\Elbows_1.csv
File Path
Browse...
C:\\Documents\MATLAB\Elbows_2.csv
Data.ImportCSV
filePath
transpose
list
AUTO
Number
0.000
List.GetItemAtIndex
list
index
item
AUTO
MEPFitting.ByMEPCurves
MEPCurve1
MEPCurve2
Fittings
AUTO

RESUME

PATRICK BORGES RODRIGUES

LEED Green Associate, 700 Jamesville Avenue, Syracuse, NY 13210 | (315) 880-1989 | pborgesr@esf.edu

EDUCATION

State University of New York College of Environmental Science and Forestry (SUNY ESF), Syracuse, NY Jan 2019 - Present
Department of Forest and Natural Resources Management
Master of Science in Construction Management (Expected Graduation: Spring 2020)
2019 C. Eugene Farnsworth Fellow
Member of Sigma Lambda Chi the International Honor Society for Construction
Cumulative GPA: 4.00/4.00

State University of Santa Cruz, Department of Exact and Technological Sciences, Ilheus, Brazil Mar 2012 – Dec 2018
Bachelor's Degree in Civil Engineering
Cumulative GPA: 9.13/10.00

Syracuse University, College of Engineering & Computer Science, Syracuse, NY Jun 2015 – May 2016
Non-matriculated undergraduate, Civil Engineering
Science Without Borders Scholar, Brazilian Federal Government
Cumulative GPA: 4.00/4.00

WORK EXPERIENCE

State University of Santa Cruz, Engineering Intern at the Physical Plant, Ilheus, Brazil Jul 2017 – Dec 2019

- Responsible for implementing BIM at the Physical Plant at the State University of Santa Cruz.
- Developed projects for the Physical Plant using Revit, Navisworks, Infraworks, and AutoCAD, including the conceptual design for the Exact Science Laboratory Complex (CLCE).
- Provided training in BIM for the employees of the Physical Plant three times.

BIM Instructor, freelancer, Itabuna, Brazil Jun 2017 – Dec 2018

- Taught Autodesk Revit for more than 100 undergraduate students and architecture/engineering professionals.

LIFE Jr. Innovation Lab (Junior Enterprise), Financial and Administrative Manager, Ilheus, Brazil Sep 2016 - Aug 2017

- Developed the company's cash flow.
- Restructured the contracts for the innovation projects.

LIFE Jr. Innovation Lab (Junior Enterprise), Project Manager, Ilheus, Brazil Jun 2014 – May 2015

- Developed, Managed and Executed internal and external projects.
- Designed the project of adapting a home for a blind wheelchair user.

Bioconsultoria - Environmental Management and Licensing, Intern, Caetite, Brazil Jun 2014 – Jul 2014

- Monitored Vegetation Removal to allow for the construction of wind turbines and towers for electrical power transmission in the Brazilian Northeast.
- Developed maps of impacted areas using AutoCAD.

RESEARCH AND TEACHING EXPERIENCE

Graduate Assistant, SUNY ESF, Syracuse, NY Jan 2019 - Present

- Teaching Assistant for Building Information Modeling for Construction Management; Sustainable Energy Systems for Buildings; Light Construction; Engineering Materials for Sustainable Construction; and Applied Structures.

Research Assistant, SUNY ESF, Syracuse, NY May 2019 – Jul 2019
LEED O+M certification of Baker Lab SUNY ESF,

- Collected data, performed analyses, and documented results for online submission.
- Created the digital model of the building using Autodesk Revit.

Senior Research Aide, The Research Foundation for SUNY, Syracuse, NY May 2019 – Jun 2019

- Assembled Cross Laminated Timber - CLT beams using black locust (Robinia Pseudoacacia) and tested them for vibration, deflection and bending stress.

Research Assistant, State University of Santa Cruz, Ilheus, Brazil Dec 2016 – Jul 2017
Applying Active Learning Methods in Civil Engineering

- Helped students and professors on implementing active learning methods in civil engineering.
- Published five papers.

CAMPUS INVOLVEMENT

A Tiny Home for Good, Syracuse, NY	Volunteer	April 2019
Civil Engineering Student Association, UESC, Brazil	President	Jan 2017 - Dec 2017
Alpha Phi Omega Community Service Fraternity, Syracuse, NY	Brother	Sep 2015 - Mar 2016

SKILLS

Software: Microsoft Office, Microsoft Project, Autodesk Revit, Navisworks, Infraworks, AutoCAD, Dynamo, ArcGIS, MATLAB, RStudio, Ftool, Lumion 8, Trimble Sketchup.

Languages: Portuguese (Native speaker)