Geometry Information Extraction in 3D Viewing Model of Industrial Construction Projects

by

Pengxiang Han

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Construction Engineering and Management

Department of Civil and Environmental Engineering

University of Alberta

Abstract:

In industrial construction projects, cost estimation is one of the most important procedures for all stakeholders. Cost estimation relies on information provided by quantity take-off. The traditional practice of manual quantification based on 2D drawings is time-consuming, tedious, error-prone and expensive. Computer-aided software based on 2D drawings still requires complicated procedures to interact with the computer. It only slightly improves work efficiency. Quantity take-off based on Building Information Modeling (BIM) is gradually applied in industrial construction projects. However, at technical and contractual levels, there are limited applications for BIM technology, due to the lack of data exchange standards, platform compatibilities and the availability of useable BIM models.

This research project proposes an alternative method to improve the efficiency of quantity take-off by automatically generating geometry information of construction components based on 3D viewing models. The 3D viewing model is the digital graphic representation of the object, and it has broadly compatible formats. However, the essential information for quantity take-off, such as component types, material properties and geometry information, is missing in 3D viewing models. The 3D viewing model cannot be directly used for quantity take-off.

This research proposes a method to address the challenges posed by the absence of geometry information in 3D viewing models. The model components will be classified based on their geometry shapes, and the specific algorithm is built to generate corresponding geometry information for each geometry shape. The algorithms will recognize, extract and calculate the geometry information about components in the 3D viewing model. The algorithm calculation results will be stored in the database, which can assist estimators to implement quantity take-off.

The academic contribution of the research is that it proposes an alternative method using 3D viewing models to implement quantity take-off. The algorithms are built to generate geometry information based on computer graphics representations.

The application contribution of the research is that it provides a usable tool which has the potential to improve the efficiency of the quantity take-off process with reliable accuracy. It can decrease the duration and cost in quantity take-off and cost estimation.

Preface

This thesis is an expansion of work published as P. Han, M. Siu, S. AbouRizk, D. Hu and U. Hermann, “3D Model-Based Quantity Take-Off for Construction Estimates,” *Computing in Civil Engineering 2017*, pages 118-124. P. Han was responsible for algorithm development and manuscript composition. M. Siu contributed to the discussion and reviewed the manuscript. S. AbouRizk was the supervisory authority and was involved in concept formation. D. Hu and U. Hermann contributed to the discussion and provided data required for validation of the proposed methodology.

Acknowledgements

I sincerely thank to my supervisor, Dr. Simaan M. AbouRizk for his kind advice, guidance and encouragement. Without his generous help, I am not able to complete this thesis. I also wish to express my gratitude to Dr. Catherine Pretzlaw and Dr. Mostafa Ali and other staff members who rendered their help during the period of last two years.

I also wish to thank Rick Hermann and Dr. Di Hu for providing me the opportunity to do my project work in PCL Industrial Management Inc.

I am grateful to my parents and my girlfriend, who has provided me with moral and emotional support in my life. I am also grateful to my other family members and friends who have supported me along the way.

Contents

List of Tables

List of Figures

1. Introduction

1.1. The Importance of Cost Estimation and Quantity Take-Off

Cost estimation is one of the most fundamental processes in construction projects. It helps the contractors to quantify the workloads, allocate the materials and resources, and ensure profits. Industrial construction projects, such as oil refineries and power stations, are usually large-scale projects with high costs, long durations and a high level of complexity. Cost estimation is important in industrial construction projects because it helps to lower uncertainties and financial risks. During the bidding stage, construction companies need to accurately estimate the cost and duration to make sure their bidding documents are appropriate. During the construction process, the cost and schedule need to be well planned and controlled to maximize profit.

The quantity take-off process generates the most fundamental information to estimate and control the cost and schedule. The process determines the necessary quantities of materials based on the design documents. The quantification result contains information such as unit costs for materials, labor and time constraints for the cost estimation (Sattineni & Bradford, 2011).

Quantity take-off is the foundation for estimating the cost of the project, which has effects on almost all the stakeholders. An efficient and accurate quantity take-off provides estimators with reliable primitive information and enables estimators to better judge the estimates. Reliable estimates can give contractors better control over the construction project's economics (Monteiro & Poças Martins, 2013). Inefficient and inaccurate information given by the quantity take-off can lead to impractical cost estimations. In industrial construction projects, which are large and complex, estimates can vary by as much as 40% from the initial budget (Flyvbjerg, Bruzelius, & Rothengatter, 2003).

Quantity take-off is a time-and-labor-intensive process. In fact, it is the most time-consuming and extremely foundational task in estimating (Hu, Lu, & Abourizk, 2015). The amount of time spent on cost estimation is project specific, and it can last several months for industrial projects (Holm, Schaufelberger, Griffin, & Cole, 2005). Based on estimations from previous projects (Rundell, 2006), 50% to 80% of the time spent on the cost estimation process is for quantity take-off. Improving or fully automating quantity take-off efficiency could significantly reduce the cost estimation period would be and make a company more competitive.

1.2. Current Practices of Quantity Take-Off

At present, there are three main methods to implement quantity take-off in industry construction projects: manually, computer-aided based on 2D drawings and BIM-based quantification.

1.2.1. Traditional Manual Method

Traditionally, quantity take-off is implemented manually by cost estimators based on the Issued for Construction drawings (IFC drawings). The cost estimators need to interpret design plans and specifications to quantities of materials based on the 2D drawings as shown in Fig. 1. For mega construction projects, such as industrial construction, the design drawing can be overwhelmingly complicated. To quantify one single component, the estimators might need to look at drawings from different views. Dealing with the large and complex projects requires patience and experience. Massive detailed information is interpreted by the estimators from the drawings. Quantity take-off in traditional manual method is a time-consuming, tedious, error-prone, and expensive process (Eastman, Eastman, Teicholz, & Sacks, 2011).

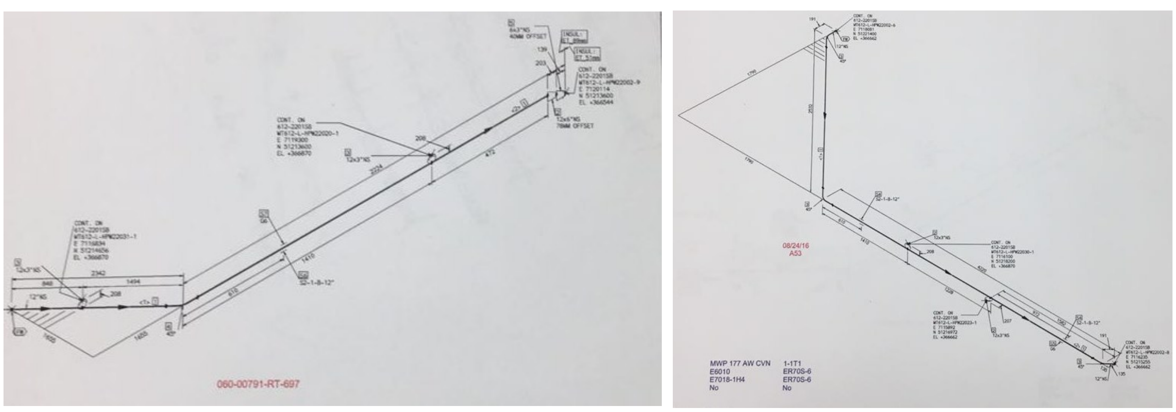
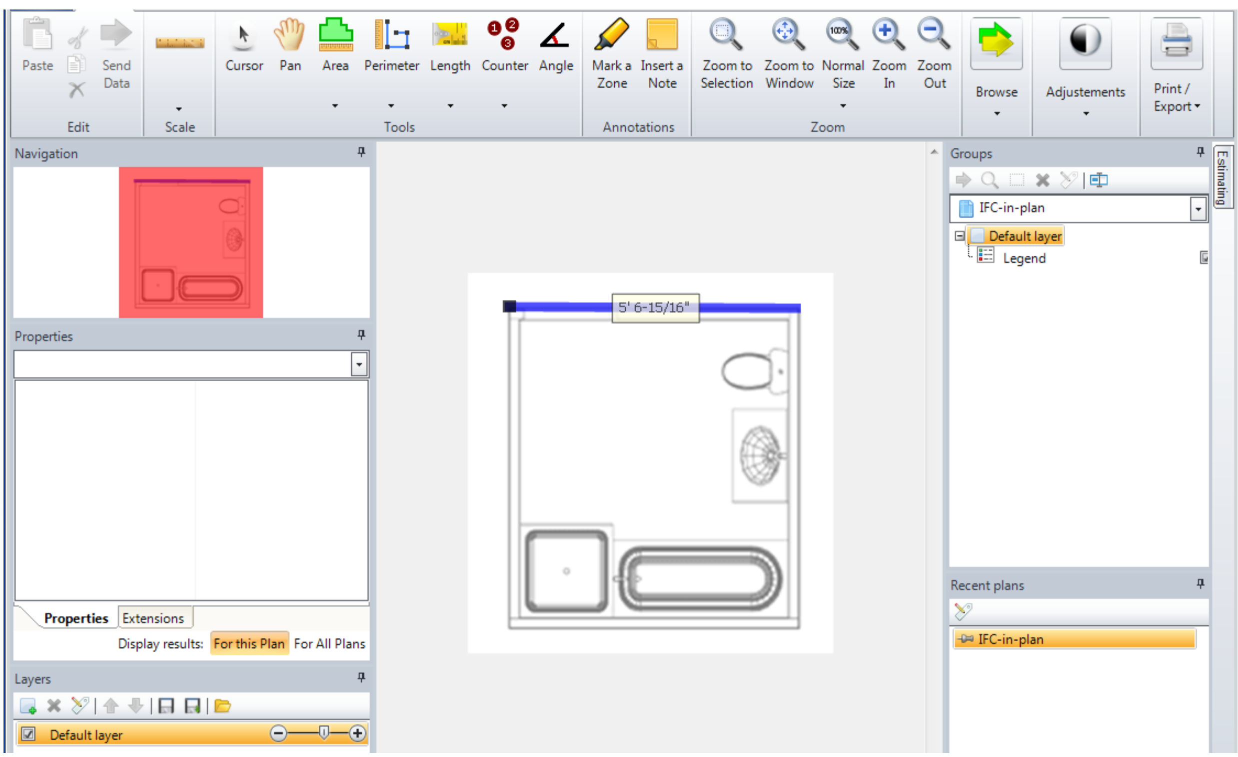


Fig. 1. Example of 2D Drawings for Piping

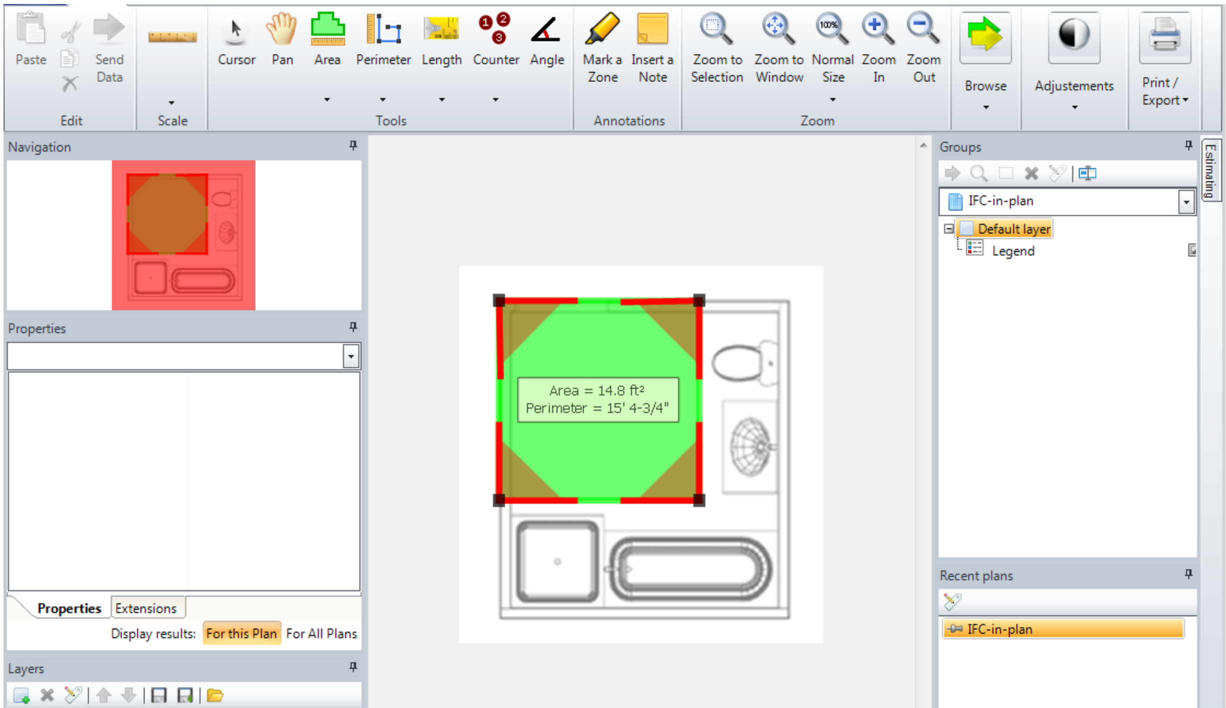
1.2.2. Computer Aided Quantity Take-Off Based on 2D Drawings

Scanned drawings or 2D-based CAD documents are used for the quantification process. Many computer-aided tools have been commercialized to help estimators to manage, view and manipulate 2D-based drawings (Holm et al., 2005). The geometry information can be obtained by selecting, highlighting and marking components on computer screens. The quantification spreadsheets and calculation tables can be generated using software. These kinds of tools have been widely applied to construction companies, since they meet the traditional working habits and do not required complex software skills.

However, the current 2D-based software only slightly increased the efficiency of quantity take-off (Monteiro & Poças Martins, 2013). The tedious interaction with software and interpretation of design drawings by estimators are main limitations in these attempts. As shown in Fig. 2, measuring 2D drawings on a screen still requires enormous manual effort. In industrial construction projects, due to the large scale and high complexity, it is common for estimators to deal with millions of components. Among the disadvantages of 2D-based drawings are that it is difficult to understand the spatial geometry relationship of components and it is difficult to interpret the drawings that have been created from different views. The 2D-based quantity take-off has limited potential for improving efficiency.



(a) Length Measurement



(b) Area Measurement

Fig. 2. Example of Computer-Aided Quantity Take-Off Software Based on 2D Drawings

1.2.3. BIM-Based Quantification

1.2.3.1. The Introduction of BIM

Building Information Modeling (BIM) is a digital representation of physical and functional characteristics of a facility (National BIM Standard, 2017). The BIM platform has been widely applied in industrial construction projects, such as Autodesk Revit, Intergraph SmartPlant and Bentley products. BIM models provide a platform to create, manage and share 3D digital representations for commercial, institutional, industrial, transportation and residential projects. Each object in the BIM models contains accurate geometrical representation, as well as non-geometric properties, such as structural properties, cost of materials, assembly information, life cycle cost and environmental data (CRC Construction Innovation, 2009; Staub–French, Fischer, Kunz, Ishii, & Paulson, 2003).

1.2.3.2. The Current Application of BIM

The information contained in the BIM model makes the model widely applicable for multiple purposes. The BIM models are based on 3D computer graphics modeling and can be used for viewing, demonstration and 3D rendering. (Eastman et al., 2011) It is more convenient to use the BIM model than 2D-based drawings for clash detections and design optimization. Furthermore, the BIM model can contain customized information for multiple purposes, and provide more sophisticated services. The BIM model incorporates 3D model objects with time as a 4D model, which can be used to produce a work schedule (Fu, Kaya, M, & G, 2007). Furthermore, 5D can be used for cost modelling and 6D can be used for facilities management and sharing cost information/data instantly with all the stakeholders in the integrated project team (P. Smith, 2014). Extensive case studies have shown that applying BIM technology can significantly save project costs, reduce time consumption and enhance collaborations among stakeholders (Hartmann, Gao, & Fischer, 2008).

The BIM technique has also been applied in cost estimation and quantity take-off processes. Cost estimation based on the BIM model has the potential to alter the estimation working process. The BIM model contains detailed information about the geometry, type and materials of every component. The components can be assigned with cost values by unit or quantity. The estimators can use the tools provided by the BIM platform to generate readable and accurate quantity take-off results in the early stages of projects. The process is supposed to be much more efficient than traditional manual methods and instantly responds to any changes in BIM models.

The BIM technique makes it possible to perform value engineering and cost control from the beginning of the design process, and it can enable a faster and more cost-effective project delivery process, higher quality buildings, and increased control and predictability for the companies (Forgues, Iordanova, Valdivesio, & Staub-French, 2012).

1.2.3.3. The Limitation of BIM-Based Quantity Take-Off

At present BIM-based quantification and estimation are still not well applied. The limitations mainly exist on the contractual and technical levels(Azhar, 2011; Zhao & Wang, 2014). To implement quantity take-off and cost estimation based on BIM models, there are three major options (Eastman et al., 2011): "1. Export building object quantities to estimating software; 2.

Use a BIM quantity take-off tool; 3. Link the BIM tool directly to the estimating software options."

However, limitations still exist in the following three areas: data exchange standards, platform compatibilities and the availability of useable BIM models.

Firstly, although the BIM model contains information for generating quantity take-off tables, BIM itself is not capable of implementing cost estimates. (Eastman et al., 2011). Estimates have to be processed by the other professional software, such as Innovaya (Sattineni & Bradford, 2011), or the third party format file must be exported from the BIM model, such as Industry Foundation Classes (IFC). IFC was developed by BuildingSMART International as the data model standard to exchange BIM data. BuildingSMART defines the standard to quantify the elements in building construction projects. (BuildingSMART International Ltd., 2017). Even IFC defines a commonly used standard for quantification and estimation (Howard & Björk, 2008), the conflicts still exist between IFC standards and companies' standards in real construction projects. (Zhiliang, Zhenhua, Wu, & Zhe, 2011) The application is limited in real construction projects because information is lost during importing and exporting operations.

Secondly, barriers to data exchange can exist in other areas. The designing of BIM model requires the highly cooperation of the architect and engineer. (Tiwari, Odelson, Watt, & Khanzode, 2009) Different BIM platforms might be used in different disciplines in the same project. If platforms are not compatible, information can be lost during data exchanges (Farah & Guillermo F. Salazar, 2005). It is hard to map elements to quantity take-off tables because different classification systems are used and companies use different measurements (Monteiro & Poças Martins, 2013). Only with the expert application of BIM technology can BIM reduce the duration and expense of cost estimations (Sattineni & Bradford, 2011; Young, Jones, Bernstein, & Gudgel, 2009).

Thirdly, another practical limitation for BIM-based quantity take-off is that many construction companies, as contractors, are not provided with BIM models, especially during the bidding process (Sattineni & Bradford, 2011). It is hard to build an integrated agreement between the owner, architect and general contractor to share and use BIM models. Concerns about ownership of the BIM model and privacy of data is one of the key obstacles for the BIM model sharing

(Simonian, 2010). Even when contractors are provided with BIM models, over 80% of the models lack the necessary information to perform quantity take-offs (Sattineni & Bradford, 2011). Some preliminary BIM models contain too few attributes and are intended more for viewing. Premature BIM models can be thought of as 3D models, which are not capable of the analytical and simulative applications. Contractors need to either build their own BIM models or implement the quantity take-off in a traditional way.

1.3. Quantity Take-Off Based on a 3D Viewing Model

1.3.1. The Current Application of a 3D Viewing Model

The 3D computer model refers to the digital representations of the object in a three-dimensional space. Compared with BIM models, the 3D viewing model contains information only about the surfaces of model components. The 3D viewing model is usually recognized as a 3D shell model, which does not contain BIM information such as material properties and volume.

In construction projects, 3D viewing models are widely applied in visualization and design plan demonstration. The 3D platforms allow users to view a model from different viewpoints. Compared with 2D drawings, the 3D models are better able to demonstrate design plans. Using a computer rendering technique, a photographic picture of design plans can be generated from the 3D model (Rossignac & Borrel, 1993). Furthermore, combining the 3D model with virtual reality (VR) can provide a more realistic and direct viewing experience (Du, Zou, Shi, & Zhao, 2016; Fernandes, Raja, White, & Tsinopoulos, 2006). The user can "walk in the built-up model" when wearing a VR device. It is also possible to interact with the 3D model by measuring the distance and moving the model component with the VR controller device.

The 3D viewing models can simulate the 4D construction process by displaying model components at a scheduled time. Many 3D platforms, such as Navisworks, provide users with the ability to link geometry representations to schedules. The scheduling data can be imported from an external database or software. The components are hidden from the model before the scheduled construction time. The project construction process can be demonstrated by displaying the component at the scheduled time in the model. The simulation can also be recorded to use as a video for demonstration and other purposes.

The 3D viewing models can also be applied in design optimization. For example, the geometrical conflicts caused by components in different floors are hard to find on the 2D drawings. However, 3D viewing platforms provide clash detective tool packages which enable users to detect geometry conflicts in designs (Leite, Akinci, & James, 2009). The conflicting components can be highlighted and commented on. This design optimization application significantly improves the work efficiency of detecting conflicts.

Most applications of the 3D model are based on the surface information about the model's components. The main limitation of a broader application is that the geometry information cannot be obtained from the 3D models (Hughes et al., 2014). Even if some 3D viewing model platforms provide users with the tool for quantity take-off, users have to input geometry information such as length, width and volume or that information must be imported from external data sources. The proposed method will address the limitations of the 3D model in quantity take-off.

1.3.2. The Advantages of the 3D Viewing Model

1.3.2.1. Compatibility

The 3D viewing model has broad cross-platform data formats, such as Virtual Reality Modeling Language (VRML), which is usually represented as a surface in particular polygonal meshes (Tangelder & Veltkamp, 2007). The 3D models of industrial construction projects can be created by different platforms, such as AutoCAD, and can also be exported from the original BIM model, such as Revit and Intergraph SmartPlant. The 3D model from different platforms can be imported and integrated into one platform, such as Navisworks, so that the 3D viewing model is compatible with different BIM platforms.

1.3.2.2. Low Risk for Sharing

Privacy and ownership concerns can be eliminated using a 3D model. The 3D viewing model is compiled only for viewing and visualization (Ali, Mohamed, Taghaddos, & Hermann, 2015). Exporting the BIM model to the 3D viewing model is a one-way operation that uses a non-propriety exchange format (Volk, Stengel, & Schultmann, 2014). Concerns about copyright and ownership of the sensitive information (Azhar, 2011) are eliminated, since most attributes of model items are missing during the exporting and importing process between the BIM and 3D viewing platforms. It is easier to share the 3D viewing models between the owner, architect and general contractors.

1.3.3. The Challenges of Using The 3D Viewing Model for Quantity Take-Off

When implementing the quantity take-off, it is necessary to obtain the geometry information of each component which needs to be quantified, such as the length and diameter of the tube. However, the information related to geometry is the surface of the model component. The geometrical features, such as length, width and volume, cannot be directly obtained from the surface information. The absence of key information, such as that related to geometrical information, component type and component materials type, is the main challenge to implementing quantity take-off with the 3D viewing model.

1.3.4. Previous Work Related to the 3D Viewing Model in Quantity Take-Off

To address the limitations of using the 3D viewing model in quantity take-off, the absent information needs to be extracted from the surface information of model components. In the quantity take-off process, the basic information required includes the type and the geometry of each model component.

Information about types of components can be obtained by different methods. As shown in Fig. 3, the 3D viewing model retains the properties and descriptions from the original BIM model. The properties and descriptions, which can be accessed as text data, contain the information that can be used to determine the component type. However, different companies and disciplines have their own naming rules and internal standards. It is hard to group model components using semantic information due to the lack of standards and conventions between model designers and model users(Ali et al., 2015; Anumba, Pan, Issa, & Mutis, 2008). For example, the value of "TYPE" property could be "E," "ELBO" and "ELBOW," and all these values can indicate that the component type is "Elbow." On the other hand, even for two components that both have the value of "TYPE" as "Column," it is not clear that the components are I-shaped steel columns or round concrete columns. A grouping recognition method has been built to obtain information about types of components. The method set a series of rules as filters to group model components based on the extracted properties of components, which are stored in a separated database as text data. However, the filter need to be custom-built for each project and it is not capable of mapping all model components with the component types in the quantity take-off table.

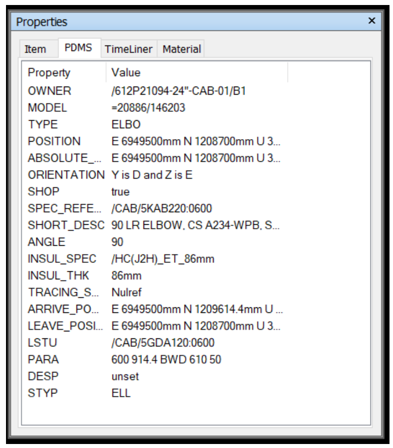


Fig. 3. The Properties of Model Component in Navisworks

Other attempts to obtain information about types of components have been based on the shape of the model component. Components of the same type usually have the same geometrical shape. For components of different types that have the same geometrical shape, the geometrical features required for quantity take-off are the same. In other words, the shape of components is an important property for quantity take-off. Many general algorithms have been built to recognize and retrieve the 3D model shape (Iyer, Jayanti, Lou, Kalyanaraman, & Ramani, 2005). Computer vision and object recognition are also used to quantify the elements in a 3D model, especially for models based on cloud point data or feature point vector data. The data can be converted to solid models (Hinks, Carr, Truong-hong, & Laefer, 2013) and recognized for quantity take-off. Specifically for industrial construction areas, the components' shape in the 3D model can be grouped and automatically recognized based on 3D mesh data (Ali, Mohamed, & Hermann, 2016).

In the shape recognition algorithms proposed by Ali (Ali et al., 2016), the shape descriptor is used to represent different shapes. The shape descriptor is defined as the histogram, which is constructed from different shape functions. In the triangle-mesh-based 3D model, the shape functions are formed based on the vertex points of the triangle mesh. By comparing the shape descriptors of each component using defined dissimilarity measures, it is possible to recognize the shape of the model component.

Combining the aforementioned methods with manual checks, the type of components in the 3D viewing model can be recognized. The shape of the model element needs to be clarified to process the quantity take-off, so that the geometrical features required can be specified. The components belong to different type required different geometrical features, such as length, radius and volume. The detailed geometry information needs to be calculated from the 3D model mesh representation. Some general algorithms have been built to extract the 3D objects' geometrical features (Zhang & Chen, 2001), but the algorithms have not been specifically applied to the industrial construction models. The method is limited to finding the principal axes of the 3D mesh model and calculating the volume of the component. The detailed geometrical features for quantity take-off, such as the length of steels and the radius of tubes, are not included. Other 3D model software packages already provide libraries to measure geometrical features of model components. These libraries are software dependent or not free to use (Abanda, Kamsu-Foguem, & Tah, 2015). The libraries are not applicable in the construction models.

In this research project, a novel method is proposed to obtain geometry information required for the quantity take-off process based on the 3D viewing model. The proposed algorithm is based on the perspective of computer graphics, using the primitive triangle mesh data of 3D model elements as data inputting. The geometry information for each component, such as length, diameter and angle, will be recognized, extracted and calculated by the proposed algorithm. The research project improves the efficiency of quantity take-off by automating the process to extract geometry information from 3D models and addressing the limitations of software interaction and data inputting.

1.4. Research Problem Statement

Quantity take-off is one of the most important processes in cost estimation of industrial construction projects. This research proposes a novel method to improve the efficiency of quantity take-off by automatically providing geometry information about components based on a 3D viewing model.

The 3D viewing model cannot be directly used to implement quantity take-off. The necessary information for quantity take-off, such as geometry information and component type, is absent in the 3D viewing models. For contractors, the geometry information of each model component is the most important information to quantify the materials required for the project. The estimators need to calculate the lengths, widths, radiuses, volumes, weights and other information for each component, and then they can use the information to generate calculation spreadsheets and estimate the expected cost for the entire project. It is the essential and primal task to extract and calculate geometry information for each component in the 3D viewing model.

The objective of the research is to automatically extract and calculate the geometry information about each component in the 3D viewing models of industrial construction projects. Accurately and efficiently obtaining the geometry information from 3D models helps to enhance the efficiency of the quantity take-off process.

2. Methodology

2.1. Methodology framework

2.1.1. Framework Introduction

This research proposes an alternative method to recognize, extract and calculate geometry information about 3D model components for quantity take-off. The method is built based on 3D viewing platforms. The framework of the proposed method is shown in Fig. 4.

The inputs are the 3D viewing model and the type of components to be quantified, such as tubes, columns or beams. In this research, the component type of each model item is provided by the preceding preparation work. The component type information can be obtained from the database.

The processing includes three major parts: 1. Isolate the components of selected types; 2. Extract the primitive data of each model component; 3. Apply the specific algorithm to the component. The components of selected types will be isolated from the 3D viewing model. Then the primitive triangle mesh data of the model component will be extracted using a built-in application program interface (API) provided by the 3D platform. The specific algorithm will be applied to each single model component based on its component type, after which geometry information about the model component can be obtained.

The outputs of the method are the geometry information and the checklist of components. Calculation results will be saved in the database, which can provide information for quantity take-off.

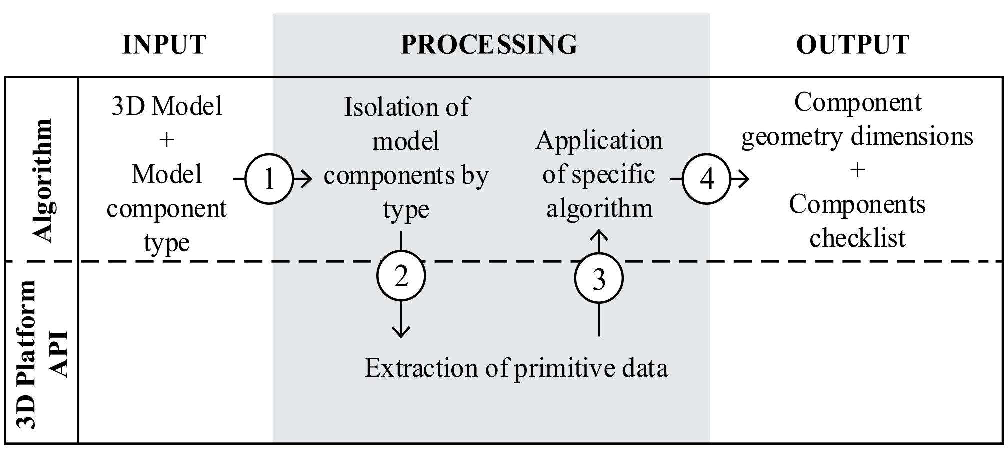


Fig. 4. The Framework of Proposed Method

2.1.2. The 3D Viewing Model Platform Navisworks

In this research project, the original design models are converted and imported into Autodesk Navisworks. Then the Navisworks files are delivered to the general contractors. Navisworks is one of the most widely used 3D viewing platforms in the construction industry (Abanda et al., 2015). It allows users to integrate models from different sources; the source files of the 3D viewing model can be from different design platforms, such as SmartPlant, Autodesk Revit and Tekla. Table 1 shows the frequently used files formats and applications which are supported by Autodesk Navisworks. Tool packages are provided for users to view, comment, highlight, hide, and measure the Navisworks model. The tool packages also provide plug-ins to implement clashes detection, 4D time construction simulation and visualization rendering. Unlike Building Information Modeling (BIM) models, the Navisworks model intended only for viewing; the model components cannot be edited, deleted or duplicated. This limitation reduces the risk of source model providers in the intelligence property ownerships.

Table 1. The Frequently Used File Formats and Applications Supporting Navisworks

Format	**Extension**	**File Format Version**
Navisworks	.nwd .nwf .nwc	All versions
AutoCAD	.dwg, .dxf	Up to AutoCAD 2017
MicroStation (SE, J, V8 & XM)	.dgn .prp .prw	v7, v8
3D Studio	.3ds .prj	Up to Autodesk 3ds Max 2017
ACIS SAT	.sat .sab	All ASM SAT. Up to ACIS SAT v7
Catia	.model .session .exp .dlv3 .CATPart .CATProduct .cgr	V4, v5
DWF/DWFx	.dwf .dwfx	All previous versions
FBX	.fbx	FBX SDK 2017.0
IFC	.ifc	IFC2X_PLATFORM, IFC2X_FINAL, IFC2X2_FINAL, IFC2X3, IFC4

Inventor	.ipt .iam .ipj	Up to Inventor 2017
Pro/ENGINEER	.prt .asm .g .neu	Wildfire 5.0, Creo Parametric 1.0-3.0
RVM	.rvm	Up to 12.0 SP5
Revit	.rvt	Up to 2017
SketchUp	.skp	v5 up to 2016
Solidworks	.prt .sldprt .asm .sldasm	2001 Plus-2015
STL	.stl	Binary only
VRML	.wrl .wrz	VRML1, VRML2
PDF	.pdf	All versions
Rhino	.3dm	Up to 5.0

(Adapted from Autodesk Knowledge Network, 2016)

2.1.3. API of Autodesk Navisworks

2.1.3.1. Introduction of Navisworks API

The application programming interface (API) is a package of routines, protocols and tools used to build the application software. For the user's convenience, some information and functions are hidden from the user interactive interface. By using API, the hidden information and functions can be accessed. The tools provided by API can be used to develop customized programs for other purposes, which help to extend the software's applications and functions.

Navisworks provides API such as .NET API, COM API and NwCreate API. The COM API is used to manipulate documents and models. It has a plug-in and automation and also provides ActiveX controls. COM API updates are no longer available and new functions are only supported in .NET API. The Navisworks API supports Visual Basic and C# language. In this research, the method was mainly developed using .NET and COM API based on C# language.

2.1.3.2. The Bounding Box of Model Component

Navisworks provides API to access geometry information about components. However, it does not directly give the geometry dimensions of the components. For each model component, the Navisworks API can return a 3D rectangular box, which is called a bounding box. The edges of the bounding box are parallel to any coordinate axis. The Max and Min are two pairs of coordinates, which are defined by the maximum X, Y and Z coordinates' values and minimum X, Y and Z values. The Max and Min correspond to two diagonal vertex points of the bounding box.

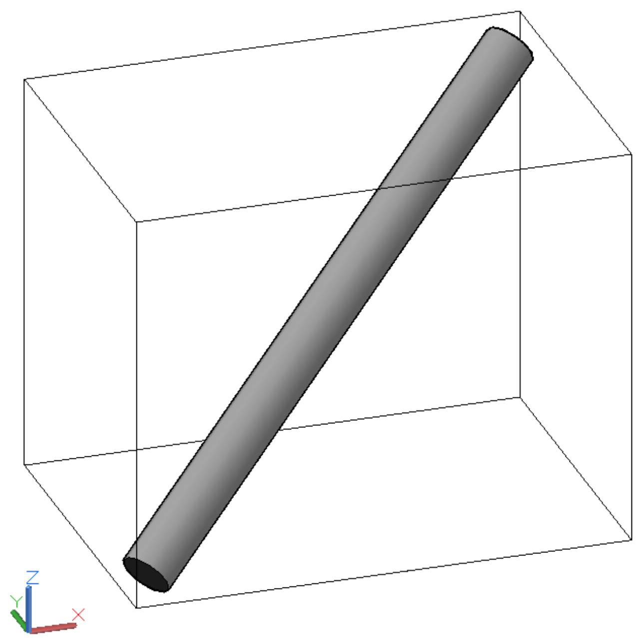


Fig. 5. The Bounding Box of the Model Component

Since the bounding box is a 3D rectangle with edges parallel to the three coordinates axes, it can only indicate the rough location and scale of the component. It cannot accurately indicate the geometry information such as length and volume. The information provided by the bounding is not sufficient for quantity take-off.

2.1.3.3. Triangle Mesh Data of 3D Viewing Model

The Navisworks model contains surface information, which is represented as triangle mesh data. Triangle mesh is the most widely used representation of a shape in graphics (Hughes et al., 2014). An example of a triangle mesh data structure is shown in Fig. 6. Triangle mesh representations are made up of a series of triangles with shared edges and vertexes.

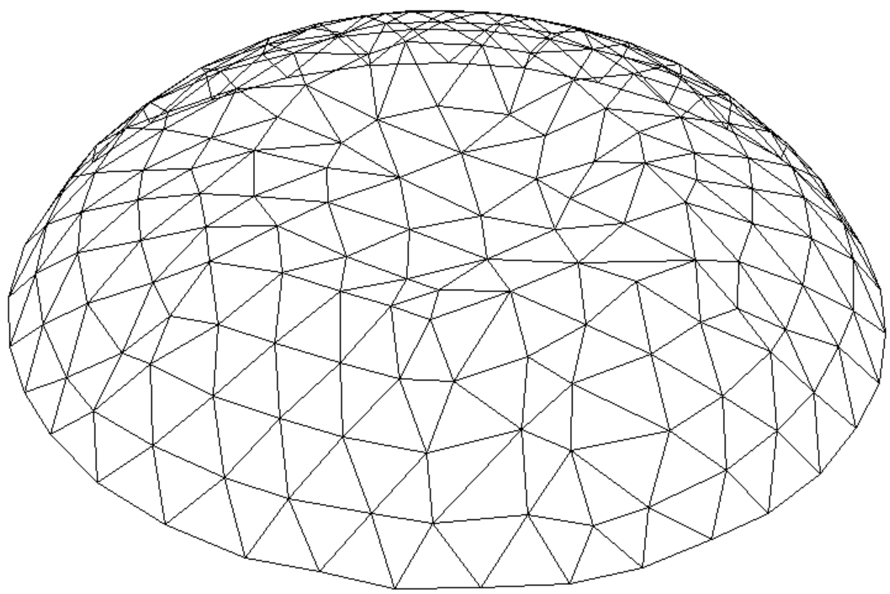


Fig. 6. The Triangle Mesh Representation of a Shell

The data structure of triangle mesh can be implemented based on a particular problem. The easiest representation is face-vertex meshes (Hoppe, Derose, Mcdonald, & Stuetzle, 1993; Hughes et al., 2014). Table 2 shows an example of Face-vertex mesh implementation. The table on the left lists each triangle separately. Each triangle contains three vertex points. The table on the right shows the coordinates of all the vertex points and their 3D coordinates.

Table 2. The Face-Vertex Implementation of Triangle Mesh

Faces / Triangles			
t1	v1	v2	v3
t2	v2	v3	v4
t3	v3	v4	v5
t4	v5	v4	v7
…	…	…	…

Vertex Points Coordinates			
	X	Y	Z
v1	x1	y1	z1
v2	x2	y2	z2
v3	x3	y3	z3
v4	x4	y4	z4
…	…	…	…

The face-vertex implementation is not efficient to manipulate the triangle mesh data (Hughes et al., 2014), but it works in the case of this research, which will extract the geometry information without changing the triangle mesh data and editing the model itself.

2.1.4. Access Model Components in Navisworks

In the Navisworks file, the model of the whole project is a combination of imported sub-models from different sources. Each sub-model can be subdivided to the next lower level model parts. The relationships of the model parts are listed as a tree structure, which can be found in Navisworks as the "Selection Tree." The lowest level of model components in the selection tree will be selected as the basic components for the quantity take-off. In this research, to improve the accuracy and reduce the complexity of the algorithm, the proposed method will be applied to each basic component. The geometry information of each basic component will be recognized, extracted and calculated.

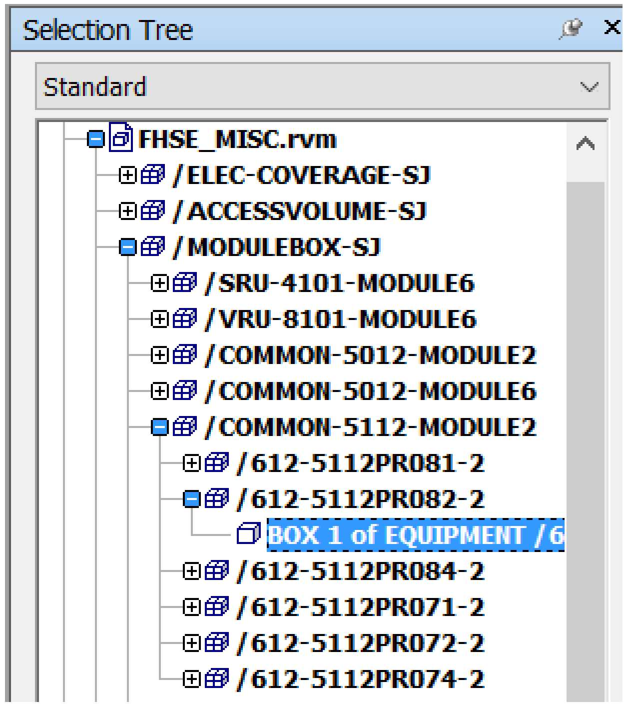


Fig. 7. Screenshot of Selection Tree and Lowest Level of Model Components

The components for quantification need to be found from the selection tree. In the construction project models, millions of components exist in the selection tree. The tree traversal algorithm based on a Breadth-first search (BFS) (Cormen, Leiserson, Rivest, & Stein, 2009) is applied in the program. The top node of the selection tree is defined as the root. A lower level node directly connected to the top node is defined as the child of the node and the converse notion of child node is defined as parent node. The program will start from the root of the selection tree. After every node is visited in the current level, the program will continue to visit next lower level. The algorithm will visit the nodes as broadly as possible on each level before going to the next level. By the end of the traversal algorithm, all the components in the selection tree will be visited.

2.1.5. External Database to Save Extracted Information

In Navisworks, the non-geometry information about each component appears as the properties of the component. A database is built to store the non-geometry and the bounding box information.

This database can be used for other purposes, such as hydraulic tests. The Entity Relationship Diagram (ERD) is shown in Fig. 8.

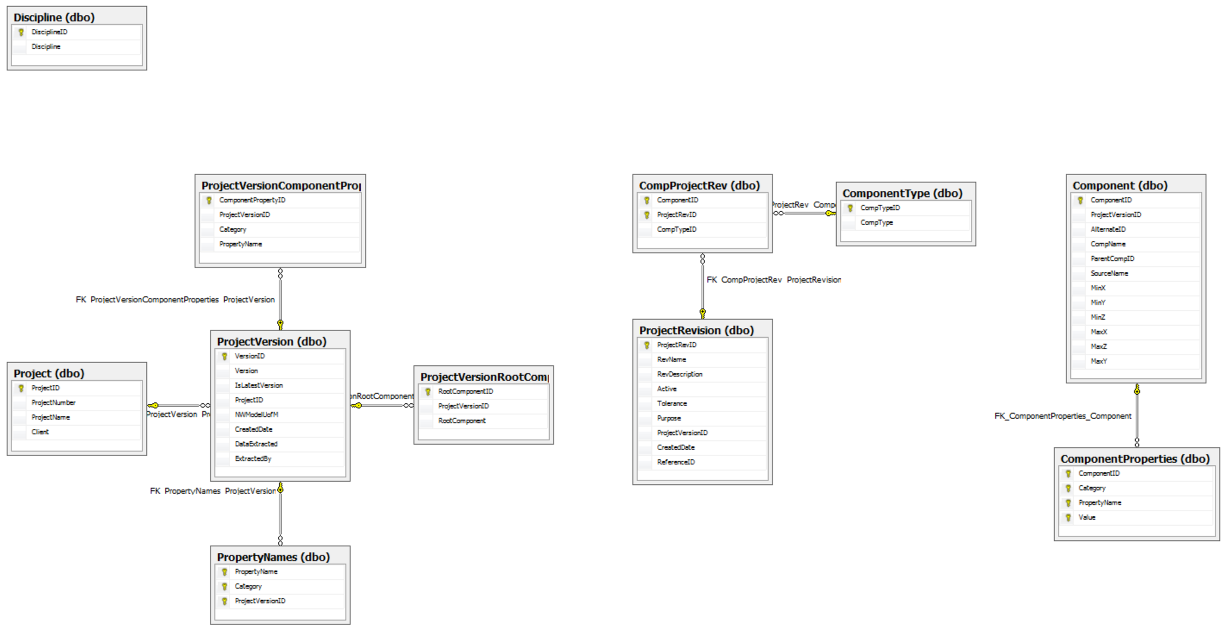
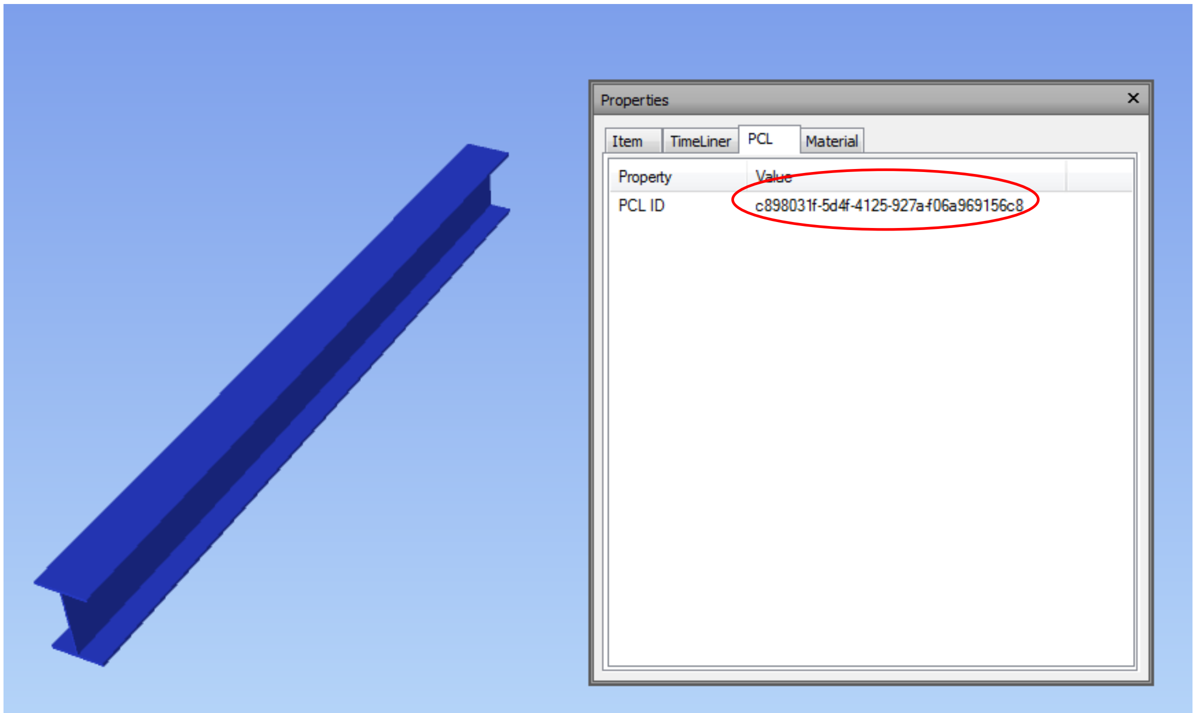


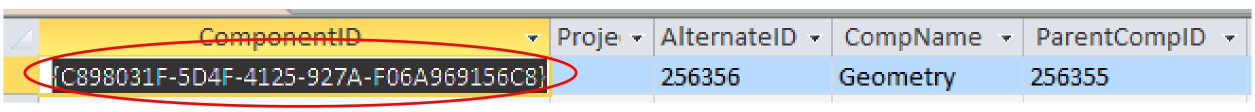
Fig. 8. The ERD Diagram of the Database

To build the database, the information about every model component is set as one record. The unique key value for each record is necessary in the database. However, Navisworks does not have unique key values for each component; the corresponding relationship between Navisworks model components and database records need to be built.

To uniquely identify each single component in the model, the Globally Unique Identifier (GUID) is introduced in the component's properties. The GUID is a 128-bit number that can be generated by standard algorithms (Leach, Mealling, & Salz, 2005). The generating algorithm can assign every component a GUID number without duplicates and write the GUID in the component's property data. Fig. 9. Example of Component's GUIDshows an example. The same GUID will be the key value of the record in the database.



(a) GUID of the Component in 3D Model



(b) GUID of the Component in the Database

Fig. 9. Example of Component's GUID

In the current database, all the existing non-geometry information has been extracted. The MAX and MIN values of the bounding box are also stored in the database. However, the current information is not capable of implementing the quantity take-off. The key information missing is the basic geometrical features of each component. The proposed method will address this missing information using the primitive data of model component.

2.2. Detailed Steps of the Proposed Method

The process is summarized as workflow shown in Fig. 10 and in the Data Flow diagram in Fig. 11. After opening the model, an add-on is provided in Navisworks to implement the proposed method. Users need to specify the component type to execute the program. The program will build a filter based on the user selection and traverse the model. All the components matched with the selected type will be found. The program will obtain the primitive triangle mesh data for each found component, adapt the data to the algorithm and apply the algorithm to obtain its geometry information. The program will implement the process for all the components with the selected component type. Then the results of the components' geometry information will be stored into the built database, which can be used to generate the final quantity take-off result.

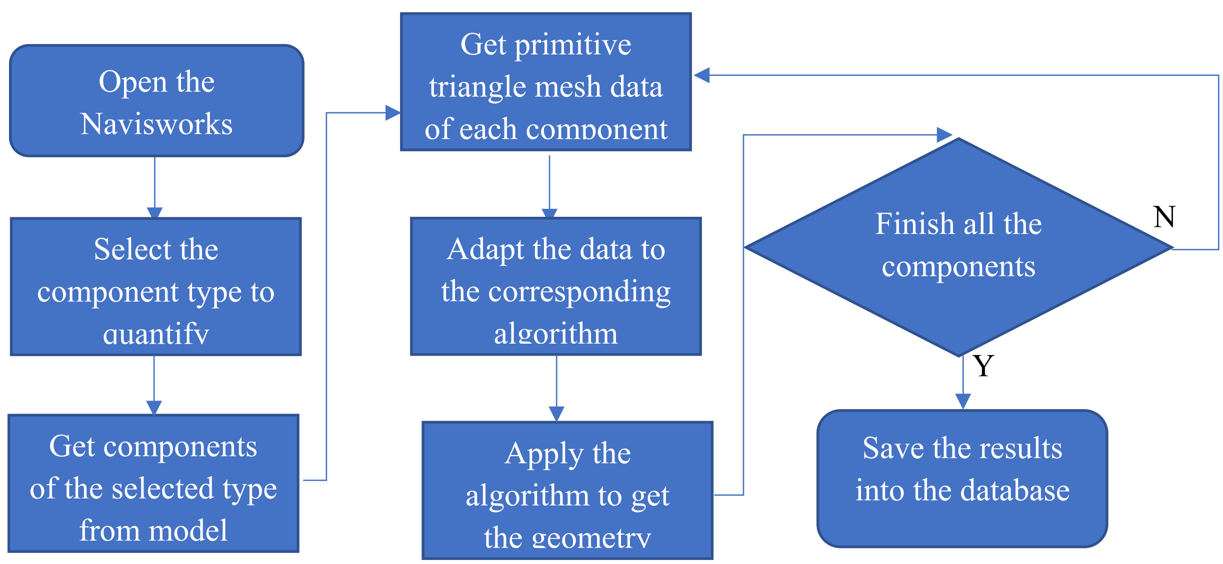


Fig. 10. The Workflow of the Proposed Algorithm

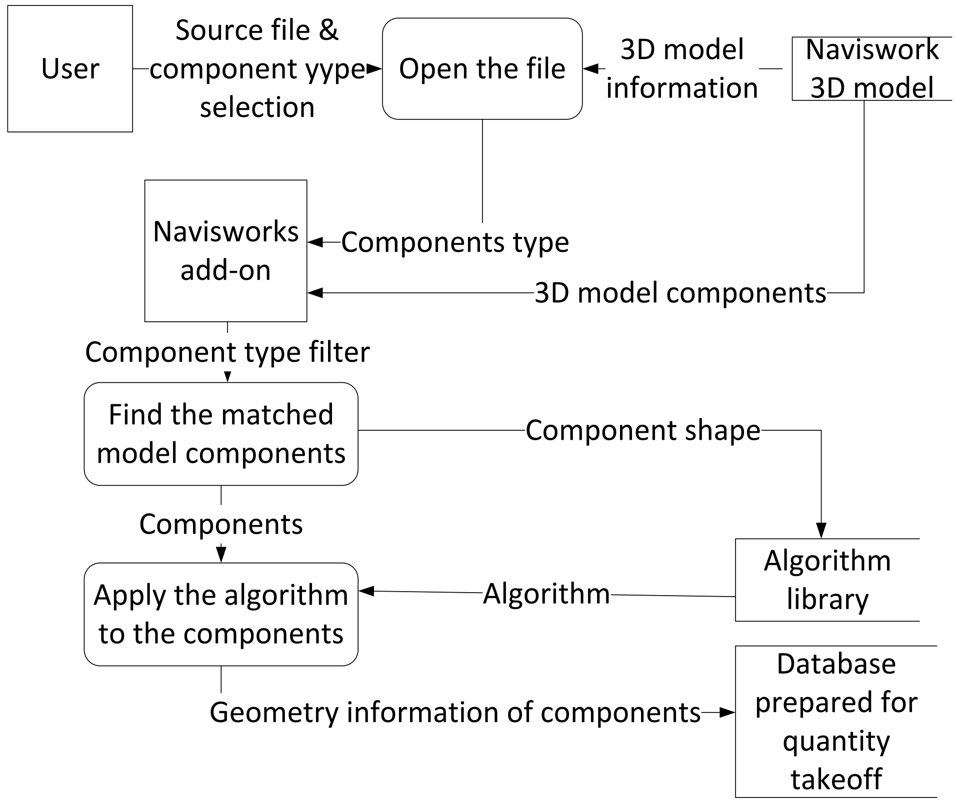


Fig. 11. Data Flow Diagram for Using 3D Model to Extract Geometry Information

2.2.1. Select Model Components by Type from Selection Tree

In this research, the type information of each component has been provided in the database. An industry construction project comprises many disciplines and types of components. Even if information about the type of components is lost in Navisworks, the related research (Ali & Mohamed, 2017) has been done to retrieve the information. It is also possible to manually check and label the component type and store the information in the database.

However, different geometrical features need to be measured for different component types. It is not practical to design algorithms for each type of component. In most cases, the same type of components have similar geometry shapes. Many types of components can be assumed to have the same shape. For example, a round column, round beam and round pile can all be assumed to be cylindrical. For certain shapes, the geometrical features that need to be quantified are the same, so a library of algorithms can be built for each geometry shape. The specific algorithm can be applied to each component based on its shape, so that the geometry information can be obtained. The detailed processes of algorithms will be introduced in Section 2.4.

In this research, algorithms are built for several basic and common geometry shapes. Table 3 lists the relationships between geometry shapes and component types. For each shape, the geometrical features are specified to implement the quantity take-off.

Table 3. List of Shapes for Data Extraction

Shape	Component Type	Geometrical Features
Cylinder	Round Column, Round Beam, Tube	Length, Radius/Diameter, Volume
Pipe	Pipe	Length, Inner Radius, External Radius, Thickness of the Pipe Wall and the Volume
Irregular Prism	I-Shaped Steel, T-Shaped Steel, Channel Steel and Square Steel	Length, Height, Width and Volume
Elbow	Elbow	Arc Length, Radius of the section, Angle of the elbow, Volume
Others	Concrete	Volume
	Others	

When using the proposed program, it is necessary to select the component type in the provided user interface. The program can find all the components with the selected type and the shape of components can be matched based on the component type. The corresponding algorithm will be applied to each component to obtain its geometry information.

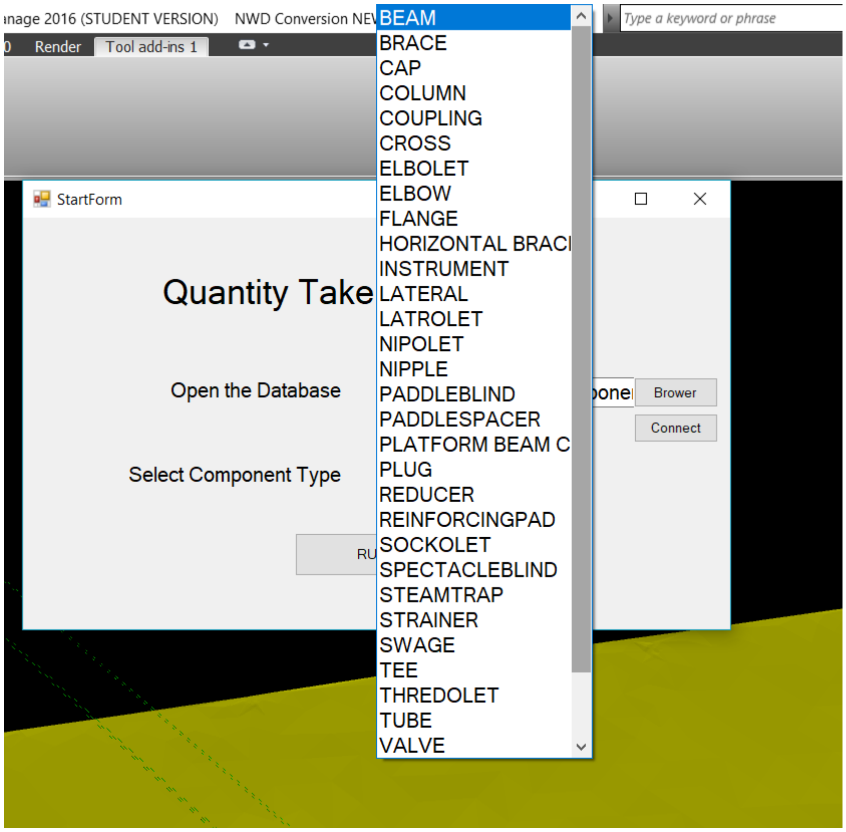


Fig. 12. User Interface of the Add-On

2.2.2. Access to The Primitive Data of Each Component

The primitive triangle mesh data of each component can be accessed via the Navisworks COM API. In Navisworks, all the components, no matter the type, will return to the same primitive data format, which is the only way that geometry information can be accessed.

Fig. 13 shows the data obtained by the method. One single component can consist of tens to hundreds of triangles. Every triangle has three vertex points. In Fig. 13, each line of data shows the coordinates of three vertex points belonging to one triangle.

Point 1			Point 2			Point 3		
-0.074	-0.374	0.000	-0.111	-0.365	0.000	-0.074	-0.374	-4.694
-0.074	-0.374	-4.694	-0.111	-0.365	0.000	-0.111	-0.365	-4.694
-0.111	-0.365	0.000	-0.146	-0.352	0.000	-0.111	-0.365	-4.694
-0.111	-0.365	-4.694	-0.146	-0.352	0.000	-0.146	-0.352	-4.694
-0.146	-0.352	0.000	-0.180	-0.336	0.000	-0.146	-0.352	-4.694
-0.146	-0.352	-4.694	-0.180	-0.336	0.000	-0.180	-0.336	-4.694
-0.180	-0.336	0.000	-0.212	-0.317	0.000	-0.180	-0.336	-4.694
-0.180	-0.336	-4.694	-0.212	-0.317	0.000	-0.212	-0.317	-4.694
-0.212	-0.317	0.000	-0.242	-0.295	0.000	-0.212	-0.317	-4.694
-0.212	-0.317	-4.694	-0.242	-0.295	0.000	-0.242	-0.295	-4.694
-0.242	-0.295	0.000	-0.269	-0.269	0.000	-0.242	-0.295	-4.694
-0.242	-0.295	-4.694	-0.269	-0.269	0.000	-0.269	-0.269	-4.694
-0.269	-0.269	0.000	-0.295	-0.242	0.000	-0.269	-0.269	-4.694
-0.269	-0.269	-4.694	-0.295	-0.242	0.000	-0.295	-0.242	-4.694
-0.295	-0.242	0.000	-0.317	-0.212	0.000	-0.295	-0.242	-4.694

Fig. 13. The Primitive Data of a Model Component Obtained by Navisworks API

2.2.3. Adapting the Data to Proposed Algorithms

Before applying the algorithm to obtain the geometrical features of the component, the primitive data will be adapted to the algorithm for further calculations. In the primitive data, duplicated vertex points exist because the same vertex point could be shared by several adjacent triangles. The program needs to find and eliminate the duplicated points. After that, a point set can be formed by the vertex points in the triangle mesh. The triangles' information can be recorded by the points since every triangle can be indicated by the three vertex points. The detailed data adaption of the algorithm will be introduced in Section 2.3.

2.2.4. Applying Algorithm Based on the Geometry Shape

The triangle mesh of the 3D model component is generated by Navisworks. The components with same geometry shape will have distribution patterns similar to those of the triangles in the triangle mesh. The algorithm can be designed to deal with each type of model component.

Aiming to each geometry shape, the program will find the specific algorithm to obtain the component's geometrical features using the adapted data. At present, the proposed algorithms support several basic geometry shapes, such as cylinder, pipe, elbow, and I/C/H-shaped prism. The program also contains the general algorithm to calculate the volume of the 3D model component with an arbitrary shape. The detailed steps of algorithms will be elaborated in Chapter 2.4.

2.2.5. Getting Geometry Information and Storing the Calculation Results in the Database

The calculation results will be saved in the database (based on SQL Server 2008 or Microsoft Access). Fig. 14 shows the schema of the proposed database for storing the generated geometry information. The Navisworks model of the whole project is imported from several source models, which are distinguished as sub-models. After the user selects the sub-model and component type, the program will find all the matched components and generate geometry information for each component. The program will save the result to the corresponding table based on the component type.

Every component will be given a unique ID (a serial number) which can be used as the key value to identify the component in the model and database. Each record in the database will contain the ID, component type and geometry information of one component. Cost estimators can easily generate calculation spreadsheets or quantity take-off results based on the database.

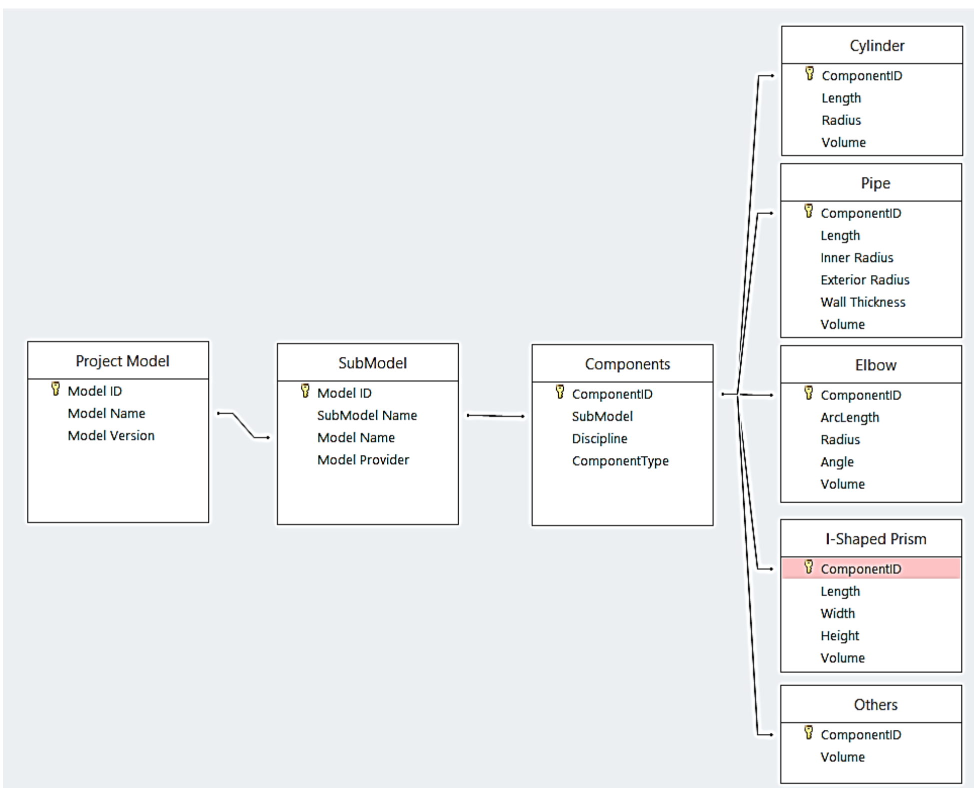


Fig. 14. The Schema of the Database for Components' Geometry Information

2.3. Basic Definitions in The Algorithms

Several shared concepts among different geometry shapes will be defined in the programs. The program will extract primitive data using Navisworks API and adapt the data to the geometry concepts. The definitions of geometry concepts in the algorithm will be introduced below.

2.3.1. Point

The Point refers to a vertex point in the triangle mesh. It is the most foundational definition in the algorithm. Each point has a serial number for identification. The parameters X, Y and Z are used to record the 3D coordinates' values, which come directly from the primitive data of the model component. In the triangle mesh, the two end points of an edge of a triangle can be recognized as two connected points. For each point, the connected points will also be recorded. The basic definition of Class Point in C# is given below.

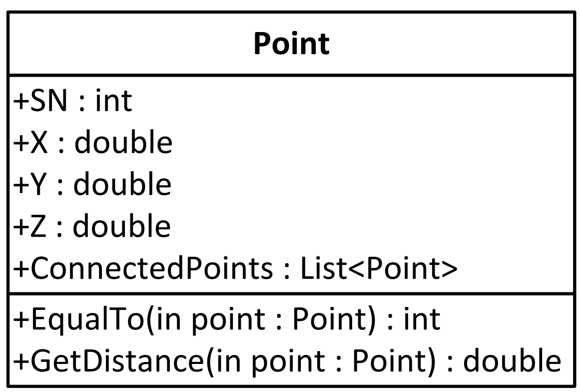


Fig. 15. The UML Diagram of Point

The method EqualTo( ) is used to judge if the two points are duplicates. If the coordinates of two points have the exact same value, this means that the two points are duplicates. If not, the two points are not equal.

The method GetDistance( ) is used to calculate the Euclidean distance of two points. The general calculation formula for calculating the Euclidean distance of Point A(x_1, y_1, z_1) and Point B(x_2, y_2, z_2) is

$$D = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

2.3.2. Triangles

Triangle refers to the single triangle in the primitive triangle mesh. Every triangle contains three vertex points, and the vertex points will be recorded to represent the triangle. The normal vector of the plane where the triangle belongs can be obtained using Navisworks API.

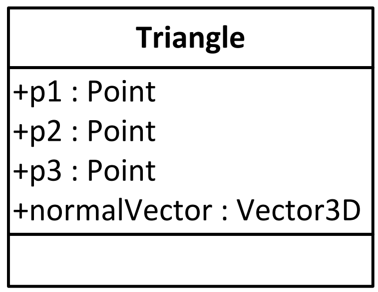


Fig. 16. The UML Diagram of Triangle

When normal vectors are considered on a closed surface, they can point to two different directions: inward and outward. As shown in Fig. 17, the inward-pointing normal vector is directed towards the interior of the surface and the outward-pointing normal vector is directed towards the exterior of the surface. All the normal vectors of triangles identified by a single model component in Navisworks API point in the same direction: either all inward-pointing or all outward-pointing.

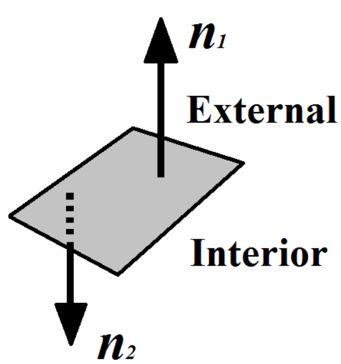


Fig. 17. The Inward-Pointing and Outward-Pointing Normal Vectors of The Surface

2.3.3. Plane

The Plane is defined as a flat two-dimensional surface. In a three-dimensional Euclidean space, a plane can be described as function:

$$ax + by + cz + d = 0,$$

Where a, b, c and d are constants and a, b, and c are not all zero. The equation for the plane is called the general form of the equation of the plane. The plane has a normal vector $\boldsymbol{n} = (a, b, c)$, which refers to a vector that is perpendicular to the plane (Anton, 2010). Based on the equation, any three points not on a single line (i.e., three non-collinear points) can determine a unique plane.

On the other hand, for the point A (x_1, y_1, z_1) and a given plane P (a, b, c, d), if the point A is located on the plane P, the equation $ax_1 + by_1 + cz_1 + d = 0$ will be satisfied; if the point A is not located on the plane P, the equation is $ax_1 + by_1 + cz_1 + d \neq 0$. In the algorithm, the points located on the plane are defined as the InnerPoints of the plane. The basic definition of the Plane class in C# is given below.

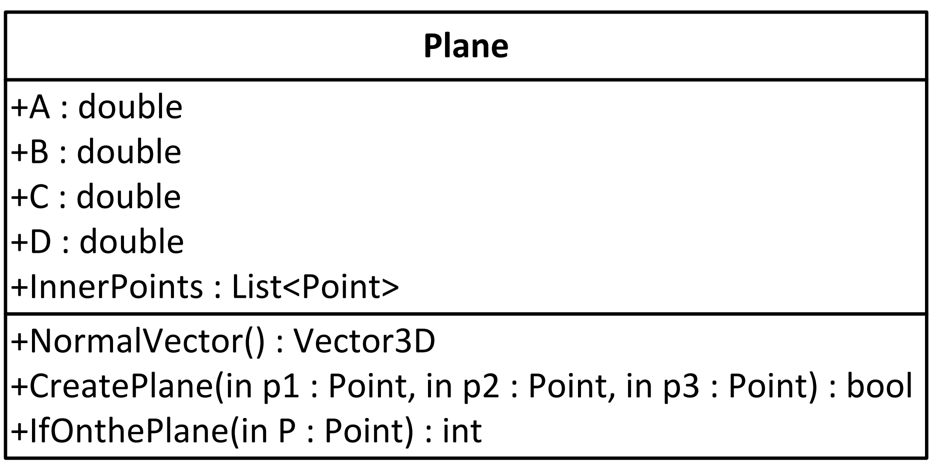


Fig. 18. The UML Diagram of Plane

The method CreatePlane() uses three non-collinear points to create the plane and calculate the general function. Assume the general function of the plane is $ax + by + cz + d = 0$ and three non-collinear points are located on the plane, $P_1 = (x_1, y_1, z_1)$, $P_2 = (x_2, y_2, z_2)$, and $P_3 =$

(x_3, y_3, z_3). Since the normal vector of the plane $\boldsymbol{n} = (a, b, c)$, the values of parameters a, b, c can be determined by forming one normal vector. Two vectors can be formed from the given points $\overrightarrow{P_1P_2}$ and $\overrightarrow{P_2P_3}$(Dawkins, 2007):

$$\overrightarrow{P_1P_2} = (x_2 - x_1, y_2 - y_1, z_2 - z_1)$$

$$\overrightarrow{P_1P_3} = (x_3 - x_1, y_3 - x_1, z_3 - z_1)$$

The cross product of any two vectors will be orthogonal to both of those vectors. Those two vectors $\overrightarrow{P_1P_2}$ and $\overrightarrow{P_2P_3}$ will lie completely in the plane since they are formed from points that were in the plane. It can be concluded that the cross product of $\overrightarrow{P_1P_2}$ and $\overrightarrow{P_2P_3}$ will be orthogonal to the plane. Therefore, the cross product will be a normal vector of the plane. It can be formed by the following steps:

$$\boldsymbol{n} = \overrightarrow{P_1P_2} \times \overrightarrow{P_1P_3} = \begin{vmatrix} \boldsymbol{i} & \boldsymbol{j} & \boldsymbol{k} \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - x_1 & z_3 - z_1 \end{vmatrix}$$

$$= a\boldsymbol{i} + b\boldsymbol{j} + c\boldsymbol{k} = (a, b, c),$$

where $\boldsymbol{i}$, $\boldsymbol{j}$, and $\boldsymbol{k}$ are three unit vectors,

$$\boldsymbol{i} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{j} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \boldsymbol{k} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

The value of parameters a, b, and c can be determined by solving the equation above.

$$a = (y_2 - y_1)(z_3 - z_1) - (y_3 - x_1)(z_2 - z_1)$$

$$b = (z_2 - z_1)(x_3 - x_1) - (z_3 - z_1)(x_2 - x_1)$$

$$c = (x_2 - x_1)(y_3 - x_1) - (x_3 - x_1)(y_2 - y_1)$$

The parameter d can be determined by substituting $P_1 = (x_1, y_1, z_1)$ into the general function:

$$ax_1 + by_1 + cz_1 + d = 0$$

$$d = -(ax_1 + by_1 + cz_1)$$

The method NormalVector( ) returns the normal vector of the plane. One of the normal vectors can be directly obtained from the general equation of the plane as $\boldsymbol{n} = (a, b, c)$.

The method IfOnthePlane( ) is used to judge whether a point is on the plane or not. If the point is on the plane, the method will return the integer 1; if the point is out of the plane, the method will return 0.

2.3.4. Circles

In the algorithm, Circles refers to a set of points. The centre of the circle is the given point. The distance between the given point and the other points is constant. The constant distance between a given point and other points is the radius of the Circle and the given point is the central point of the Circle.

Each Circle will be given a unique serial number. If the distance between a point and the central point is equal to the radius, the point will be recognized as "on the Circle". All the points located on the Circle will be recorded. In some cases, the Circles are arranged in a specific order, and the neighboring Circles are connected. The Circles connected to the current Circle will be recorded in the algorithm. The central point of the circle $P_{central}(x_c, y_c, z_c)$ can be calculated by the following equation:

$$x_c = \frac{\sum_{i=1}^{n} x_i}{n},\ y_c = \frac{\sum_{i=1}^{n} y_i}{n},\ z_c = \frac{\sum_{i=1}^{n} z_i}{n}$$

Where x_i, y_i and $z_i$are coordinates of points uniformly located on the circles.

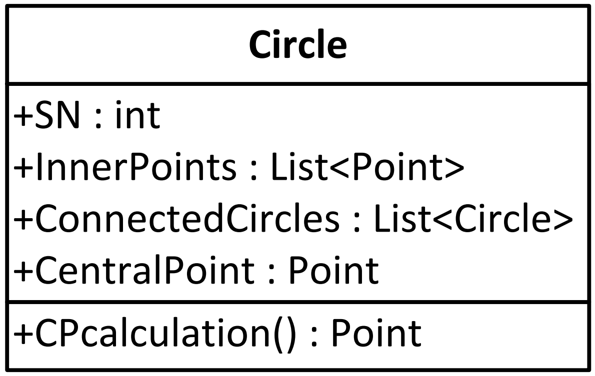


Fig. 19. The UML Diagram of Circle

2.4. The Library of Algorithms for Each Geometry Shape

The algorithm will deal with a single component to identify, extract and calculate the geometrical features based on the geometry shape. The inputting data is the primitive triangle mesh data of the component, and the algorithm will adapt the data to the geometry concepts. Then the corresponding algorithm will be applied based on the geometry shape. The final calculation results will be stored into the database, which can be further used by estimators for quantity take-off. The working process of the algorithm is shown below:

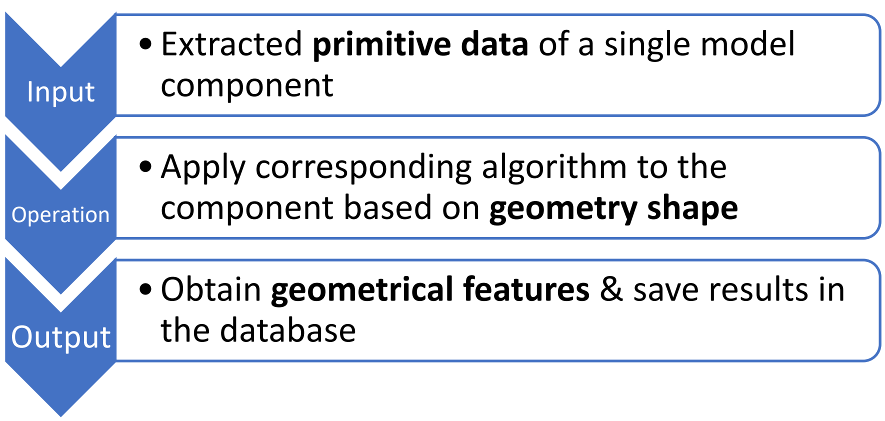


Fig. 20. Method to Obtain Geometry Information of a Single Model Component

The following sections will separately introduce the algorithms for each basic geometry shape.

2.4.1. Algorithm for Cylinder Shape

The Cylinder is one of the most basic and common geometry shapes in industrial construction, is used to model many components including round beam, round column, and tubes. Since Navisworks only keeps graphics representations of a model, some components are simplified as cylinders. For example, the screw pile is a kind of pipe with helix flanges. However, in some models, the screw piles are modeled with cylinders and the helix flanges are omitted in the model. The algorithm can still be applied to this case and give an approximate result for reference.

The cylinder shape has two circular end planes perpendicular to the axis. The distance between the two end planes is the value of the cylinder's length. For cylindrical-shaped components, the required geometrical features are the radius of the cross-section, the length of the cylinder, and the volume.

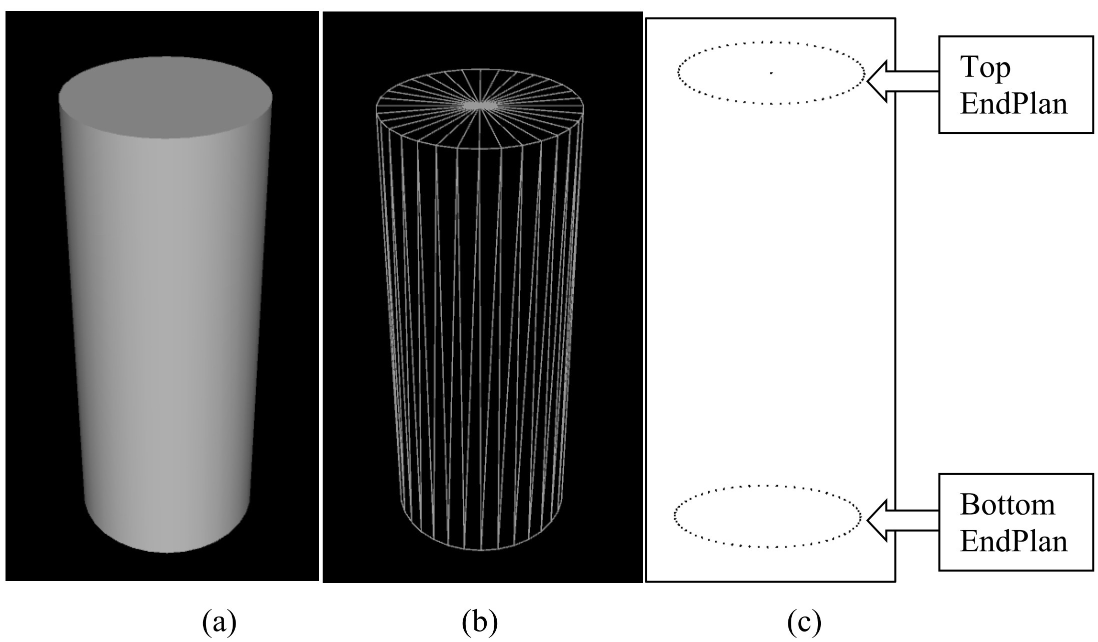


Fig. 21. Extracted Primitive Data of Cylindrical-Shaped Component
(a) Component in the Original 3D Model, (b) Triangle Mesh Representation of the Model, (c) Extracted Vertex Points of the Triangle Mesh

Fig. 21 shows different representation forms of model components. Fig. 21(a) shows a round column component in a 3D model. Fig. 21(b) shows the column in triangle mesh and Fig. 21(c) shows the vertex points of the triangle mesh. It is easy to find the vertex points on the centre and edge of the two end planes. No point is located between the two end planes.

The end plane can be determined by the number of points on the plane. The vertex points consist of a 3D point set. In the point set, every three points can be used to define a plane, but only two of the planes can be found have more than four points. This is because at most, four intersection points exist if the plane has intersections with both the Top and Bottom EndPlanes. After two EndPlanes are found, all the points on the planes form two Circles. The central point of each Circle can be found. It is easy to calculate the Circle's radius, which is the Radius of the Cylinder. The distance between two central points is equal to the value of the Cylinder's Length. The volume can be calculated using the Radius and the Length.

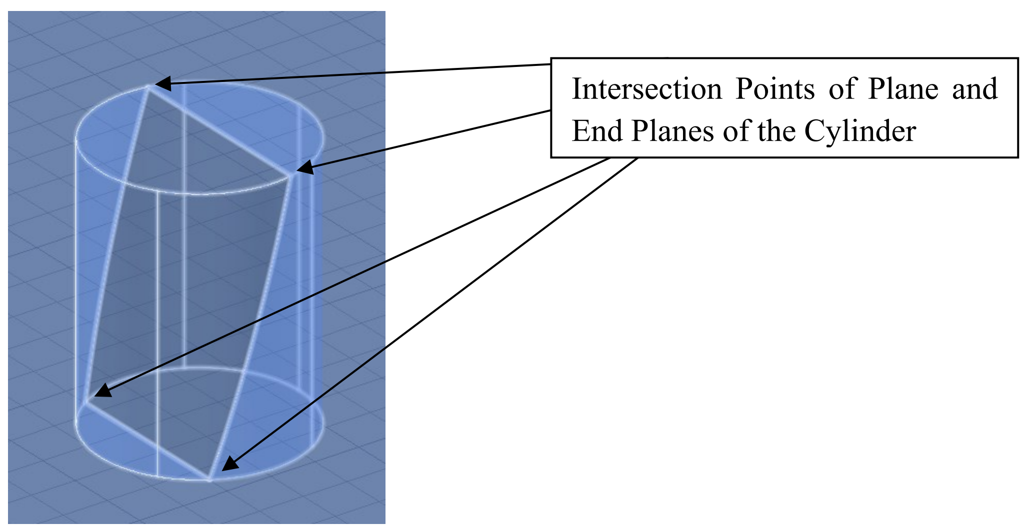


Fig. 22. The Intersection Points of the Plane and End Planes of the Cylinder

The steps and pseudo code of the algorithmic process for quantifying cylindrical shapes are detailed in Fig. 23 and List 1, respectively.

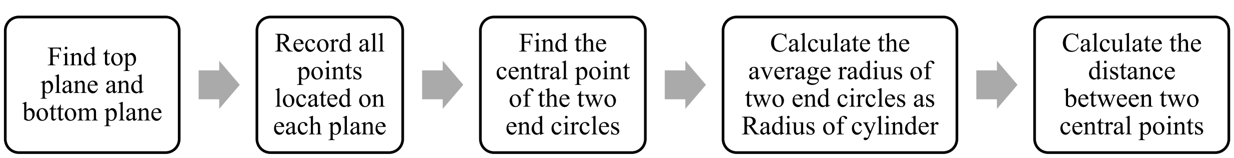


Fig. 23. The Algorithm Process Designed to Quantify the Cylinder

List. 1. Pseudo code for Cylinder:

Algorithm 1. Cylinder

Input: Coordinates of Vertex Points in Triangle Mesh
Output: Length, Radius and Volume

1. N = number of Points
2. PointSection[3] = {Point[0], Point[1], Point[2]}
3. Create newPlane using PointSection //newPlane: Ax + By + Cz + D = 0
4. **for i** = 1 to N
5.   **if** Point[i] is not used && Point[i] is on newPlane
6.   Mark Point[i] on newPlane
7. **if** the number of Points on newPlane is more than **five** //num of intersection points of plane and cylinder's EndPlanes is four at most
8.   Mark all Points on newPlane as used
9.   Save the newPlane as one EndPlane
10. **else** discard the newPlane
11. Build new PointSection with other Points
12. repeat steps 3 to 11 until two EndPlanes are found
13. **Length** = distance of two EndPlanes
14. **Radius** = Average radius of point sets on two Endplanes
15. **Volume** = $\pi Radius^2 \times$ Length

2.4.2. Algorithm for Pipe Shape

The Pipe shape is defined as a hollow cylinder. The wall of the pipe has a thickness attribute. In some cases, including steel tubes and round braces, the components are modeled as pipe shapes. The geometrical features required are length, inner radius, external radius and thickness of the pipe wall.

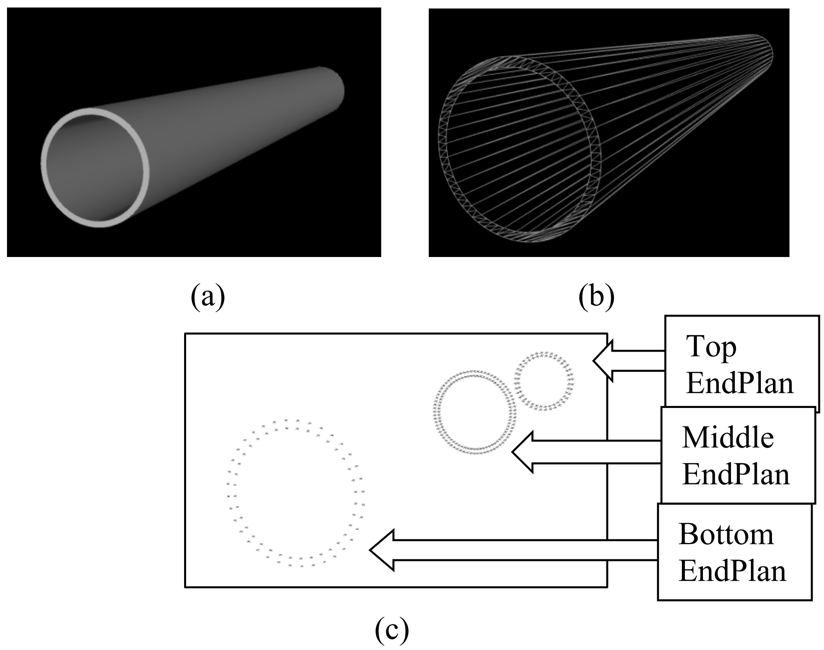


Fig. 24. Extracted Primitive Data of Pipe-Shaped Component
(a) Component in The Original 3D Model, (b) Triangle Mesh Representation of The Model, (c) Extracted Vertex Points of The Triangle Mesh

As shown in Fig. 24, the shape and triangle mesh representation of the pipe resembles the Cylinder. A similar algorithm can be used to find the Top, Middle and Bottom planes. The longest distance between the three planes will be the length of the Pipe.

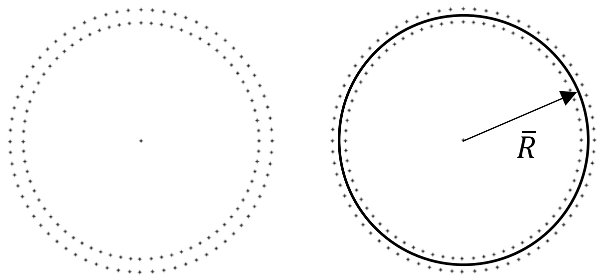


Fig. 25. Points on Bottom Plane

The radius can be obtained from the single planes. Take the Bottom plane as an example. The average radius $\bar{R}$ of the point set can be obtained by calculating the average distance of all the points to the center of a circle. Then for each point, if the distance of the point to the center is larger than $\bar{R}$, the distance is the External Radius; if the distance is less than $\bar{R}$, the distance is the Inner Radius. The Thickness is the difference between the External Radius and the Inner Radius. The volume can be calculated using the Length, External Radius and Inner Radius. The pseudo code, detailed in List 2, describes the algorithmic process for quantifying pipe shapes.

List. 2. Pseudo code for Pipe:

Algorithm 2. Pipe

Input: Coordinates of Vertex Points in Triangle Mesh
Output: Length, Inner Radius, External Radius and Thickness of the Pipe Wall

1. N = number of Points
2. PointSection[3] = {Point[0], Point[1], Point[2]}
3. Create newPlane using PointSection //newPlane: Ax + By + Cz + D = 0
4. **for i** = 1 to N
5.   **if** Point[i] is not used && Point[i] is on newPlane
6.   Mark Point[i] on newPlane
7. **if** the number of Points on newPlane is more than **nine** //num of intersection points of plane and pipe's EndPlanes is eight at most
8.   Mark all Points on newPlane as used
9.   Save the newPlane as one EndPlane
10. **else** discard the newPlane
11. Build new PointSection with other Points
12. Repeat steps 3 to 11 until two EndPlanes are found
13. Find the central point CP of one EndPlane
14. Calculate the average radius $\bar{R}$ of points set Points[] one the EndPlane

15. **for** Point in Points[]
    d = distance between Point to central point CP
    if $d < \overline{\mathrm{R}}$
        Inner Radius = d
    if $d > \overline{\mathrm{R}}$
        External Radius = d
16. **Length** = distance of two EndPlanes
17. **Thickness** = **External Radius - Inner Radius**

2.4.3. Algorithm for Irregular Prism Shape

The prism shape has a uniform cross-section shape along its length. Prisms can be categorized by the shape of the cross section; e.g., a triangular prism has a triangle-shaped cross section.

The proposed algorithm focuses on a prism shape that has an axisymmetric cross section shape. Most steel structures in 3D models can be assumed to be irregularly shaped prisms, such as I-beams, T-bars, channel steel and square steel, with axisymmetric cross sections. Fig. 26 shows some examples of steel model components.

Steel structure designs that incorporate cross sections as I-shapes, T-shapes, C-shapes and squares take full advantage of a material's mechanical properties and also reduce the amount of materials needed.

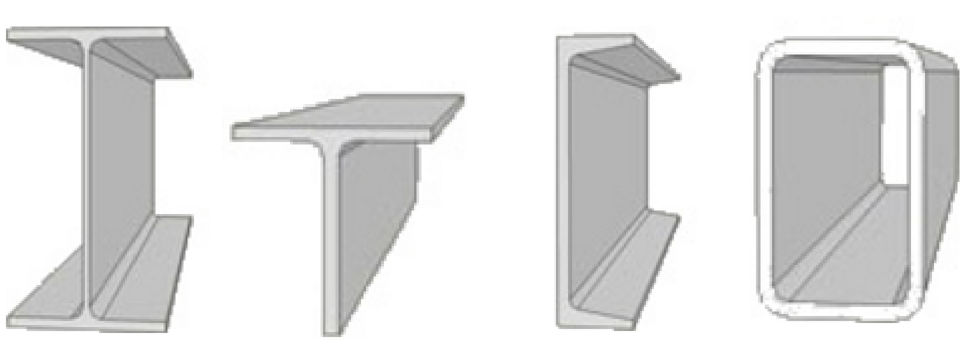
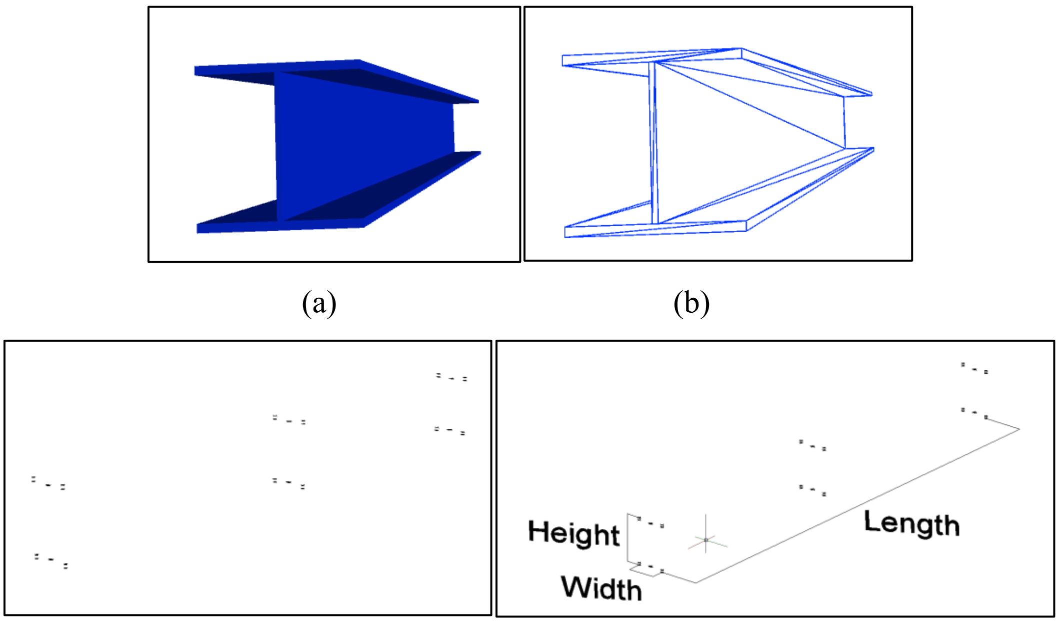


Fig. 26. Examples of Steels with Different Cross Sections

Geometry information required for irregular prisms includes the length of the prism, width, height and other details about the cross section. However, the current algorithm only extracts the approximate geometrical features of the component. When the height and width are obtained, the algorithm has the potential to compare the result with the provided structural steel tables and determine more detailed geometry information. Fig. 27 shows the primitive triangle mesh representation of the model component.



(a) (b)

(c)

Fig. 27. Extracted Primitive Data of I-Shaped Prism Component

(a) Component in the Original 3D Model, (b) Triangle Mesh Representation of the Model, (c) Extracted Vertex Points of The Triangle Mesh

The steps of the algorithm are listed below.

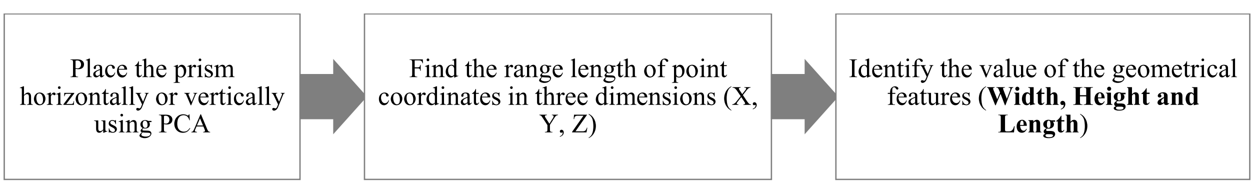


Fig. 28. The Algorithm Process Designed to Quantify the Prism

In construction projects, the steel structures, such as the braces and support beams, can be placed at any angle. To obtain the width and height of the cross-section, the prism must be positioned horizontally or vertically in the 3D coordinate system. The method of principal component analysis (PCA) is used to align the components when they are positioned in different directions (Zhang & Chen, 2001).

PCA is a statistical procedure that uses orthogonal transformation to transform an original set of variables into a set of linearly uncorrelated variables that represents most of the information in the original set of variables. The initial idea was proposed by Pearson (Pearson, 1901) and independently developed by Hotelling (Hotelling, 1933). The method has been widely applied to statistical analysis, data calculation and other areas. Its primary goal is to reduce the dimensionality of the original data set (Dunteman, 1989).

In the PCA method, the first principal component provides a projection axis. The axis produces the smallest total projection distances when all the data points are vertically projected. For symmetric geometry, the first principal component coincides with the symmetry axis. In this case, the orthogonal principal components can be used to rotate the 3D model aligned with the coordinate axes of the three-dimensional space.

In this research, the proposed algorithm used the mathematical computing process introduced by Smith (L. I. Smith, 2002). Fig. 29 shows the process that aligns a point set with the coordinate axes of the three-dimensional space. Fig. 29(a) is the original status of the point set. The coordinates of the points can form a $n \times 3$ matrix, where three columns represent the X, Y and Z coordinate values. Since the points are three-dimensional, the covariance matrix can be calculated as a 3×3 matrix. Three eigenvectors and eigenvalues of the covariance matrix can be calculated. These eigenvectors are considered to be the principal components, as shown in Fig. 29(b). In this research, the data points are defined in the three-dimensional linear space. All three eigenvectors are used to form a transformation matrix, which can rotate the point set without changing the relative position of the points. Fig. 29(c) shows the result after the point set is rotated to align with the coordinate axes.

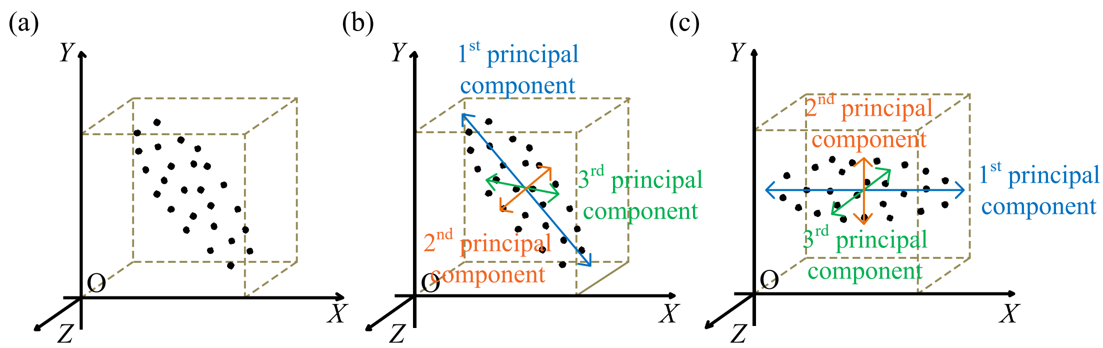


Fig. 29. PCA Process

(a) Original 3D Data Points, (b) Determine the Three Principal Components Based on Data Points, (c) Data Points after Transformation.

Using the PCA process, the component is rotated horizontally or vertically without changing its shape or dimensions. An example is given in Fig. 30.

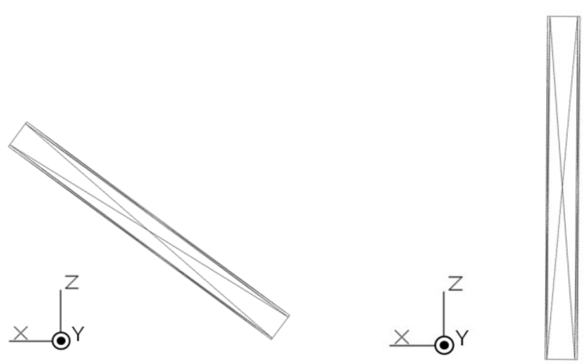


Fig. 30. I-Shaped Steel Component Before and After PCA

After the PCA process, the range lengths (as shown in Fig. 27(c)) of coordinates will be found on each direction, axis-X, axis-Y and axis-Z. Among these three lengths, the maximum one will be recognized as the Length of the component, the minimum will be the Width and the middle value will be the Height. In case of exceptions, such as the steel being very short, the obtained dimensions will be verified by this criterion: Length > 2 × (Width + Height). By comparing the height and width with the handbook of steel designs, the specific type of steel can be confirmed. The algorithm is also applicable to steel structures with axisymmetric cross-sections, such as C-shaped and T-shaped structures. However, the algorithm only gives the value of width and height (depth). The thickness of the web and flange cannot be given by the algorithm.

The pseudo code, detailed in List 3, describes the algorithmic process for quantifying irregular prisms.

List. 3. Pseudo code for Irregular Prism Shape:

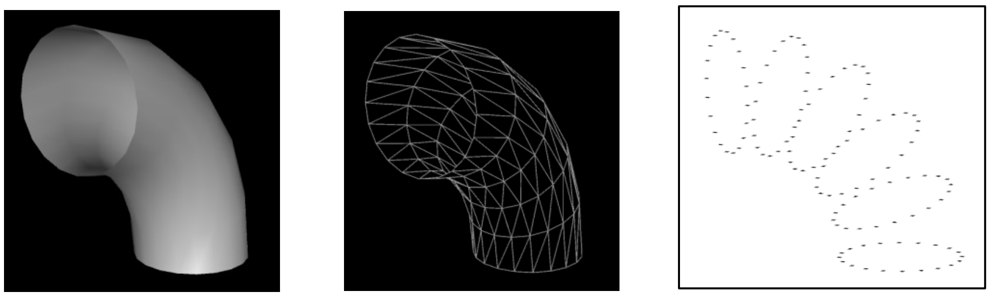
Algorithm 3. Irregular Prism Shape

Input: Coordinates of Vertex Points in Triangle Mesh
Output: Length, Width and Height

1. N = number of Points
2. Applying PCA from Step 3 to Step 7 to rotate the prism horizontally or vertically
3. Transfer coordinates of Points to a $N \times 3$ matrix M
4. Subtract the mean value of each column, get the adjusted matrix M_Adj
5. Calculate the 3×3 covariance matrix of M_Adj
6. Calculate the eigenvectors and eigenvalues of the covariance matrix to form the transformation matrix M_Trans
7. Get the coordinates of Points after transformation: M_Final = M_Adj × M_Trans
8. int range[] = new int[3];
9. Find the ranges of coordinates in three dimensions X, Y, Z, save the data in range[]
10. Sort range in ascending order
11. **Length** = range[2] (maximum range value)
12. **Height** = range[1]
13. **Width** = range[0] (minimum range value)

2.4.4. Algorithm for Elbow Shape

An elbow is defined in this research as an item that connects two pipes of different orientations. The elbow shape can be considered as a "bended cylinder." For elbow-shaped components, the geometrical features required are the radiuses of two end cross-sections, the length of the arc and the bended angle of the elbow.



(a) (b) (c)

Fig. 31. Extracted Primitive Data of Elbow-Shaped Component

(a) Component in the Original 3D Model, (b) Triangle Mesh Representation of The Model, (c) Extracted Vertex Points of the Triangle Mesh

Fig. 31 shows the primitive triangle mesh representation of the elbow component. The elbow is formed by several cross-section planes arranged along the central axis. The points are located on the cross-section planes of the elbow. The geometrical features of the elbow can be obtained by finding the cross-section planes. Steps of the algorithm are listed below.

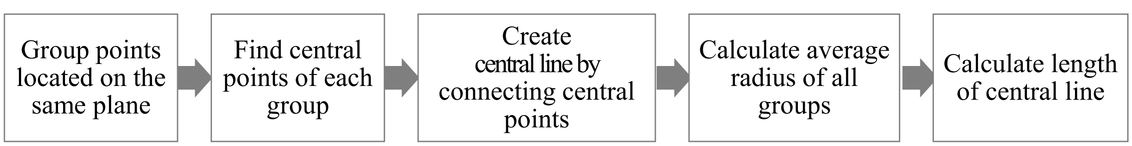


Fig. 32. The Algorithm Process Designed to Quantify the Elbow

Points located on the same plane will be classified as one group. The algorithm will find two end cross-section planes and calculate the radius of each circle formed by the points. The angle of the elbow can be obtained by calculating the angle of normal vectors for two end cross-section planes. For each cross-section plane, the central point of the points can be found. The Central line will be created by connecting the central points one by one. The length of the central line is an approximate representation of the elbow's length. The pseudo code is shown in List. 4 to describe the algorithm for quantifying the Elbow shape component.

List. 4. Pseudo code for Elbow:

```
Algorithm 4. Elbow
---------------------------------------------------------------
Input: Coordinates of Vertex Points in Triangle Mesh
Output: Radius, Length of the Arc, Bended Angle of the Elbow

1.  Select a random Point from the PointsSet
2.  Calculate the minimum distances MinD between the selected Point to other points
3.  double Density =  MinD × 1.01
4.  List<Circle> circleList = new List<Circle>( );
5.  while PointsSet contains non-used Point
        Select a non-used random Point as StartPoint
        Create a new Circle and add to the circleList
        for Point in Points Set
            if distance between current Point and StartPoint d <= Density
                StartPoint and current Point belong to the same Circle
                Marked the current Point as used
                StartPoint = current Point
6.  for Circle in circleList
        if all the Points in the current Circle locate on the same plane
            Check next Circle
        else
            Create a new Circle which contains most of the Points belongs to the old Circle
            Abandon the old Circle and add the new Circle to the circleList
7.  Calculate the Central Point of each Circle
8.  Form the Central Line of the Elbow by connecting the Central Points
9.  Length = length of the Central Line
10. Radius = average radius of Circles
11. Angle = angle of two planes where two end Circles located on
```

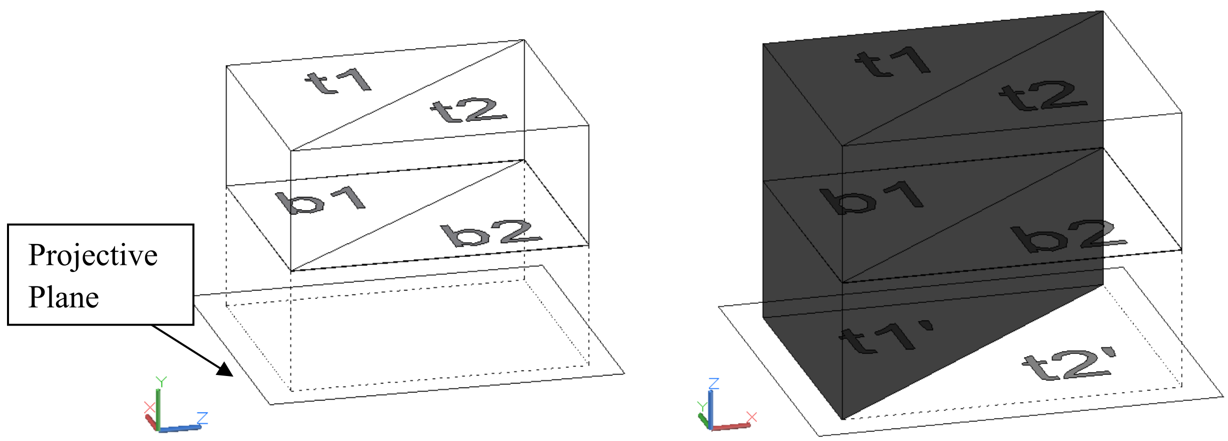
2.4.5. Volume-Calculating Algorithm for Other Shapes

For components with regular shapes, once the dimensions are obtained by the algorithm, the volume can be easily calculated. However, for components such as those made of concrete, which has irregular shapes, it is hard to generalize the geometrical features required for quantity take-off. The volume of irregular shapes cannot be calculated directly from the geometry dimensions.

Based on the triangle mesh data, the volume of an irregularly shape component can be calculated by the **projection method**. The method is applicable to any closed geometry shape represented by triangle mesh data. A plane which does not intersect with the component will be set as the projective plane. For each triangle of the triangle mesh data, such as triangle t, the projection on the projective plane will be labeled as t'. As shown in Fig. 33, a component with a cuboid shape is used to explain the volume calculation process. In Fig. 33(a), t1, t2, b1 and b2 are the triangles forming the surface of the cuboid. In Fig. 33(b), the shaded section shows the volume between t1 and the projected t1'. The volume of the component can be calculated by the formula:

$$V = V_{t1} + V_{t2} - V_{b1} - V_{b2}$$

V_{t1}, V_{t2}, V_{b1} and V_{b2} separately stand for the volume between t1-t1', t2-t2', b1-b1' and b2-b2'.



(a) (b)

Fig. 33. The Volume Calculation by Projection Method

When applying the projection method to calculate the volume of the model component, for the convenience of calculation, the plane $Z = Z_{min}$ is set as the projective plane. Z_{min} is the minimum Z coordinates value of the point set of the component's triangle mesh data. The volume between the original triangle t and its projection t' on the plane $Z = Z_{min}$ is defined as V_t.

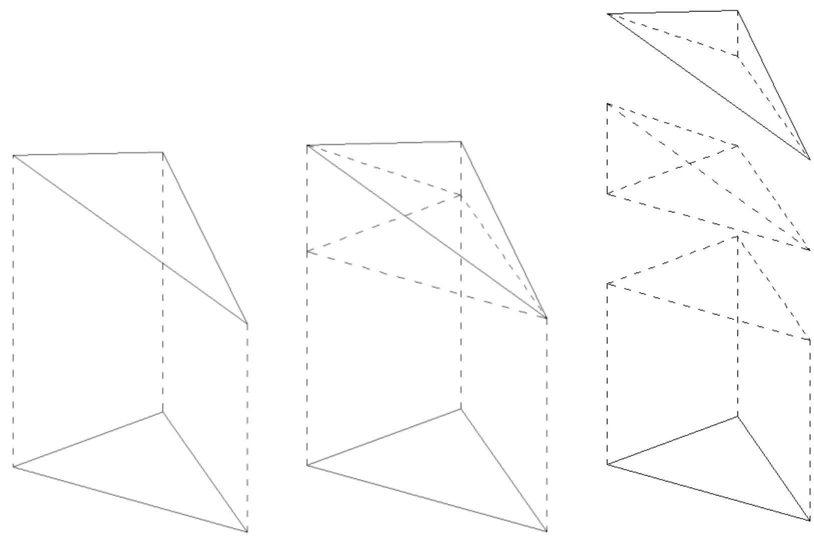


Fig. 34. The Process of Calculating the Volume Between the Triangle and Its Projection

For the single triangle t, the projective volume V_t can be calculated as shown in Fig. 34. V_t can be divided into three parts, which are one triangular prism and two tetrahedrons. The volume of the triangular prism can be calculated by the product of the triangle's area and prism's height. For the tetrahedron, the coordinates of four vertexes are represented as (x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3) and (x_4, y_4, z_4). The volume of the tetrahedron can be calculated by the formula:

$$V_{((x_1,y_1,z_1),(x_2,y_2,z_2),(x_3,y_3,z_3),(x_4,y_4,z_4))} =$$

$$\frac{1}{6} \times \begin{vmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{vmatrix} = \frac{1}{6} \times \begin{vmatrix} x_2 - x_1 & x_3 - x_1 & x_4 - x_1 \\ y_2 - y_1 & y_3 - y_1 & y_4 - y_1 \\ z_2 - z_1 & z_3 - z_1 & z_4 - z_1 \end{vmatrix}$$

Since the triangle mesh representation of the component is formed by a set of triangles, the projective volume of each triangle can be calculated. The volume of the whole geometry shape is the algebraic sum of each triangle's projective volume:

$$V = \sum_{i=1}^{n} kV_{t_i}$$

Where n is the total number of triangles in the triangle mesh. The value of parameter k can be -1, 0 or 1, which is used to decide whether to add or subtract the triangle's volume. The value of k is decided by the direction of the triangle's normal vector. All the normal vectors are assumed to be outward-pointing. Assume the normal vector of triangle t as $\vec{v}_t$ and a basis vector as $\vec{\boldsymbol{b}} = (0,0,1)$, which are pointed along the positive direction of the Z-axis. The angle between $\vec{\boldsymbol{b}}$ and $\vec{v}_t$ can be calculated with the following equation:

$$\cos\alpha = \frac{\vec{\boldsymbol{b}} \cdot \vec{v}_t}{|\vec{\boldsymbol{b}}||\vec{v}_t|}$$

If $\cos\alpha > 0$, $\alpha \in [0^{\circ}, 90^{\circ})$, then k = 1 and the triangle's projective volume is positive;

If $\cos\alpha < 0$, $\alpha \in (90^{\circ}, 180^{\circ}]$, then k = -1 and the triangle's projective volume is negative;

If $\cos\alpha = 0$, $\alpha = 90^{\circ}$, then k = 0 and the triangle's projective volume is 0.

The steps of the volume calculation algorithm are summarized in List 5.

List. 5. Pseudo code for Volume Calculation:

Algorithm 5. Volume Calculation

Input: Coordinates of Vertex Points and Triangles in Triangle Mesh
Output: Volume

1. Find the minimum Z coordinates' value of vertex points as BaseZvalue
2. **double Volume** = 0
3. **foreach** Triangle **in** the mesh
    Find the projections of three vertex Points on the plane z = BaseZvalue
    Divide the volume between current Triangle and its projection to three parts: 2 tetrahedrons and 1 prism
    Calculate the volume of each part separately
    Get the sum of three parts as **triangleVolume**
    Get the normal vector of the current Triangle as Normal
    Vector Positive = (0,0,1)
    if the dot product of Normal and Positive > 0
        Volume = Volume + triangleVolume
    else Volume = Volume - triangleVolume
4. **return Volume**

3. Validation

3.1. Introducing the Testing Model

In this research, a testing model is used to validate the result of the algorithm. The models represent a section of the steel structure and piping system. As shown in Fig. 35, each component can be found on the selection tree. The geometry information of each component will be manually measured with “Measure” tools, which is provided by Navisworks. The program will go through the testing model and obtain geometrical features of each component. The results from the manual measurement and algorithm calculation will be compared for validation.

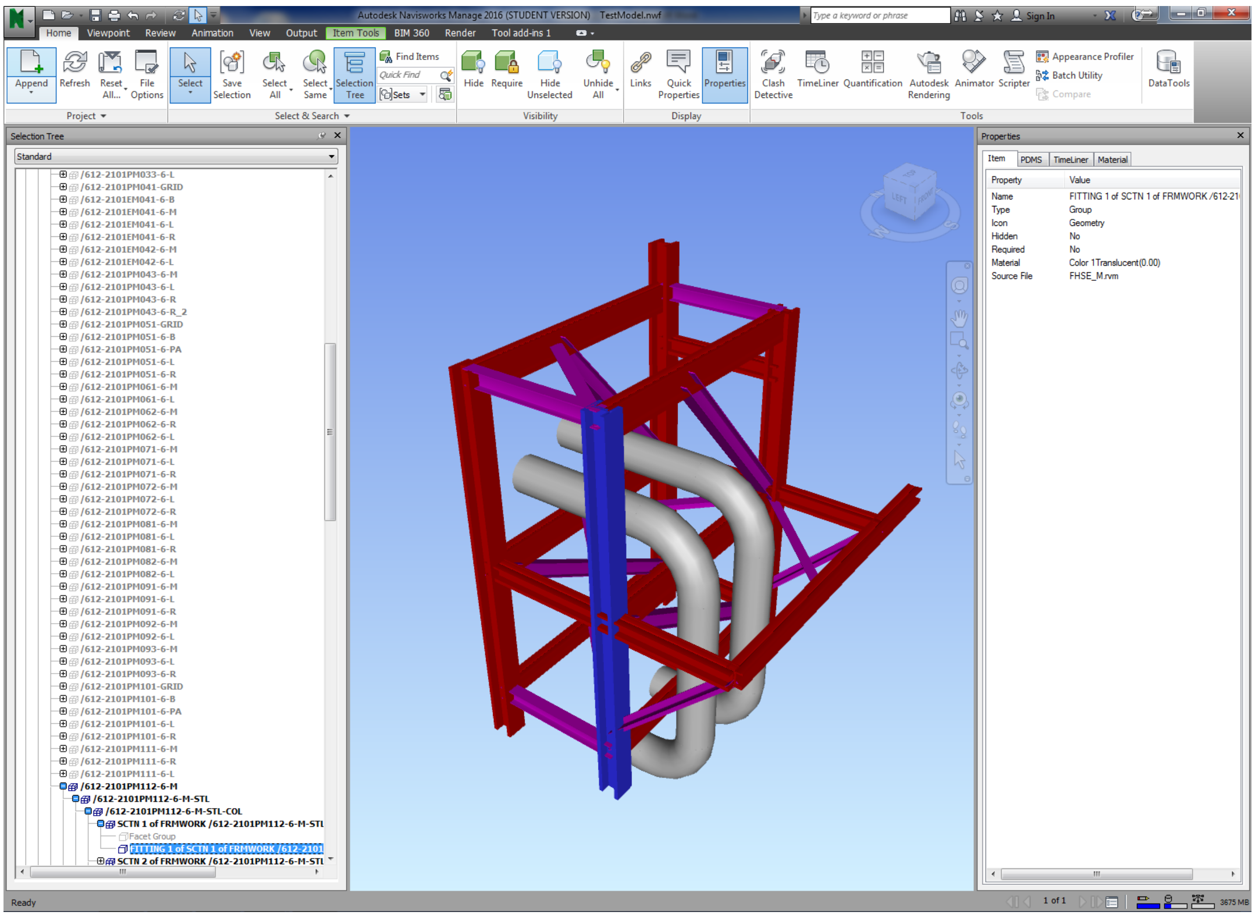


Fig. 35. Components and Selection Tree in The Testing Model

3.2. The Manually Measured Geometrical features of Model Components

As shown in Fig. 36, the “Measure” tool is provided in Navisworks. It enables users to measure the components’ geometrical features, such as length, angle and area. The tool is manually operated by using a cursor to select a point on the screen. Fig. 37 shows an example of measuring the length of structure steel. The geometrical features of each component in the testing model will be measured. The result of the manual measurement is shown in the Appendix.

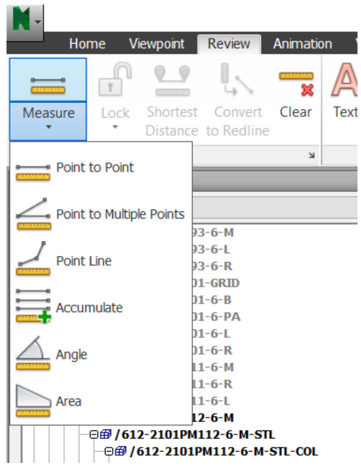


Fig. 36. The “Measure” Tool in Navisworks

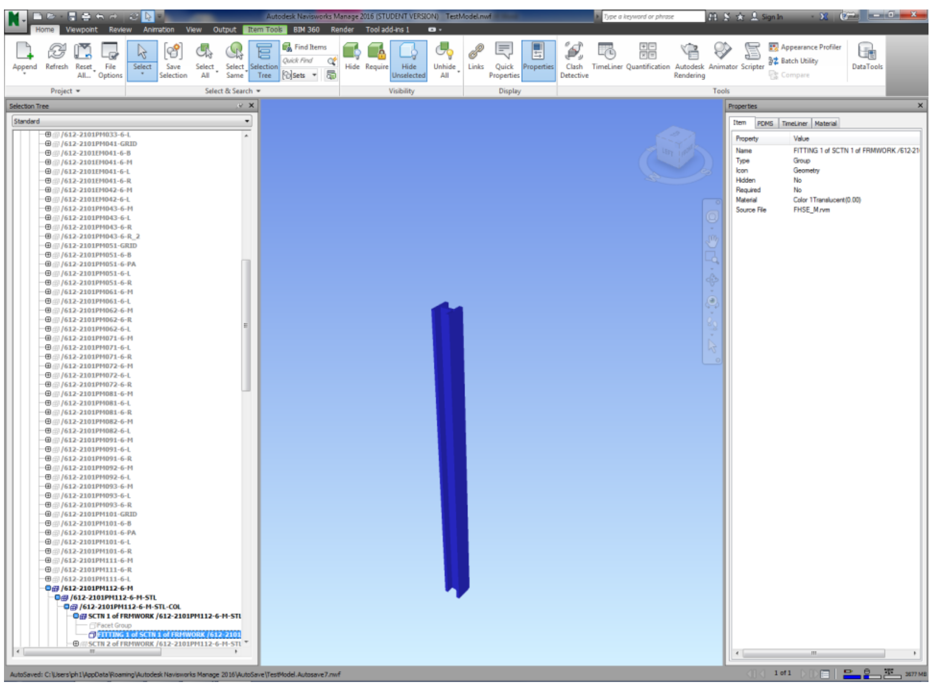


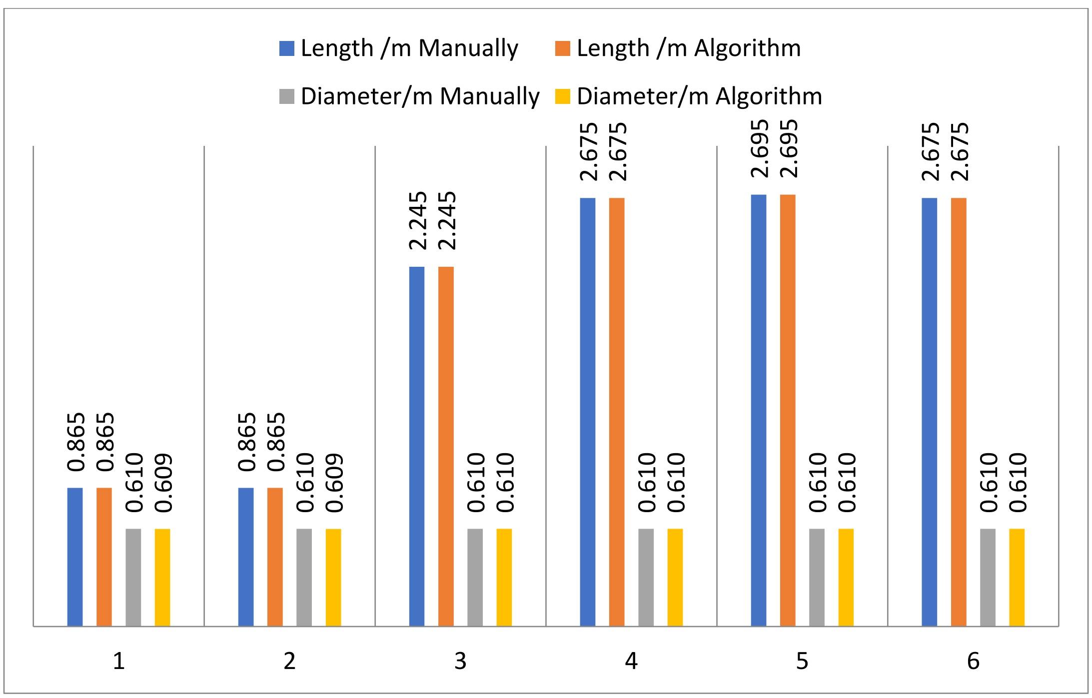
Fig. 37. Measuring the Length of Structure Steel Using Navisworks “Measure” Tool

3.3. The Calculation Result of the Proposed Algorithm

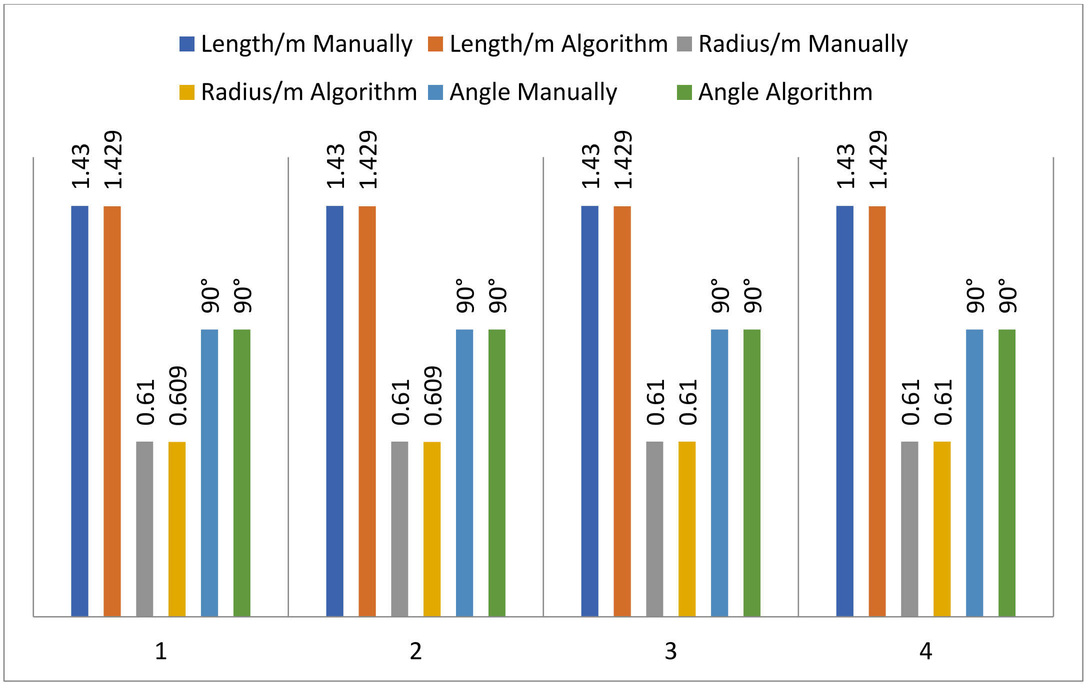
To proposed algorithm is applied to each component in the testing model to obtain its geometrical features. The algorithm will extract and calculate the corresponding geometry information for each component type. The results will be compared with the manual measurements.

3.4. Comparison and Analysis of Results

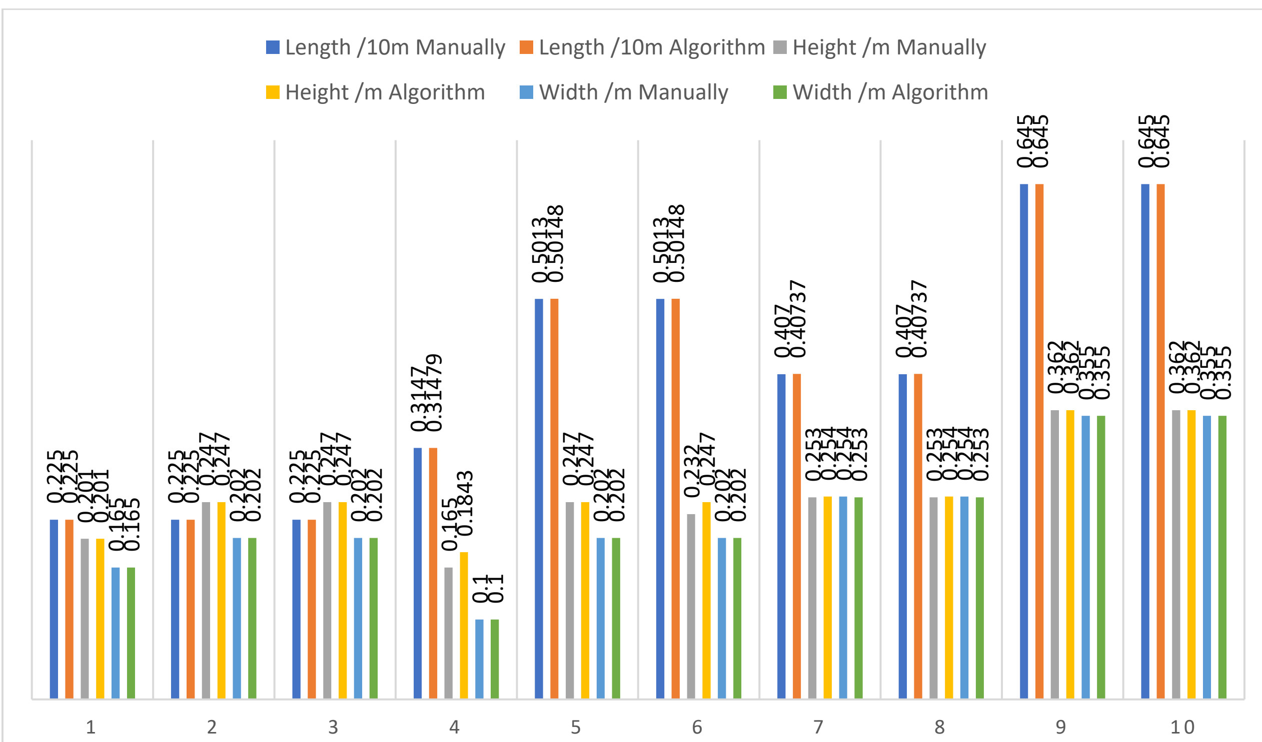
The testing model contains three different component types: tubes, elbows and steel structures. These three components have different geometry shapes. The comparison between the algorithm calculation results and manual measurements is shown in Fig. 38.



(a) Comparison of Tube Results



(b) Comparison of Elbow Results



(c) Comparison of the First Ten Steel Structures Results

Fig. 38.The Comparison between Algorithm Calculation Results and Manual Measurements

By comparing manual measurements to algorithm calculation results, the algorithm can accurately extract the geometrical features of the components in the testing model.

The testing model contains tubes which are placed in different directions. The proposed algorithm accurately extracted the length and radius of each component. The average difference of lengths and radiuses between manual measurements and algorithm calculations is less than 0.1m.

For the elbow components in the testing model, the elbows have same geometry dimensions but are placed in different directions. The average difference of length and radius is less than 0.001m and the angle can also be accurately extracted.

For the structure steels, all the cross sections are I-Shaped. The length, height and width are quantified. The average differences of the three dimensions are less than 0.1m.

The proposed algorithm shows the potential to generate accurate geometry information from 3D viewing models. Since the average difference between the calculation results and real value is less than 0.1%, it can be neglected in the quantity take-off process. The extracted geometry information is acceptable for this testing model.

4. Case Study

4.1. Case Study Description

As shown in Fig. 39, an energy infrastructure construction project is used as a case study in this thesis. The contractor was provided with an early stage 3D model in the Navisworks platform and Issued for Construction (IFC) drawings. The model is comprised of millions of components in different disciplines. In current practice, the estimating team needs to spend months of work implementing the quantity take-off process based on IFC drawings.

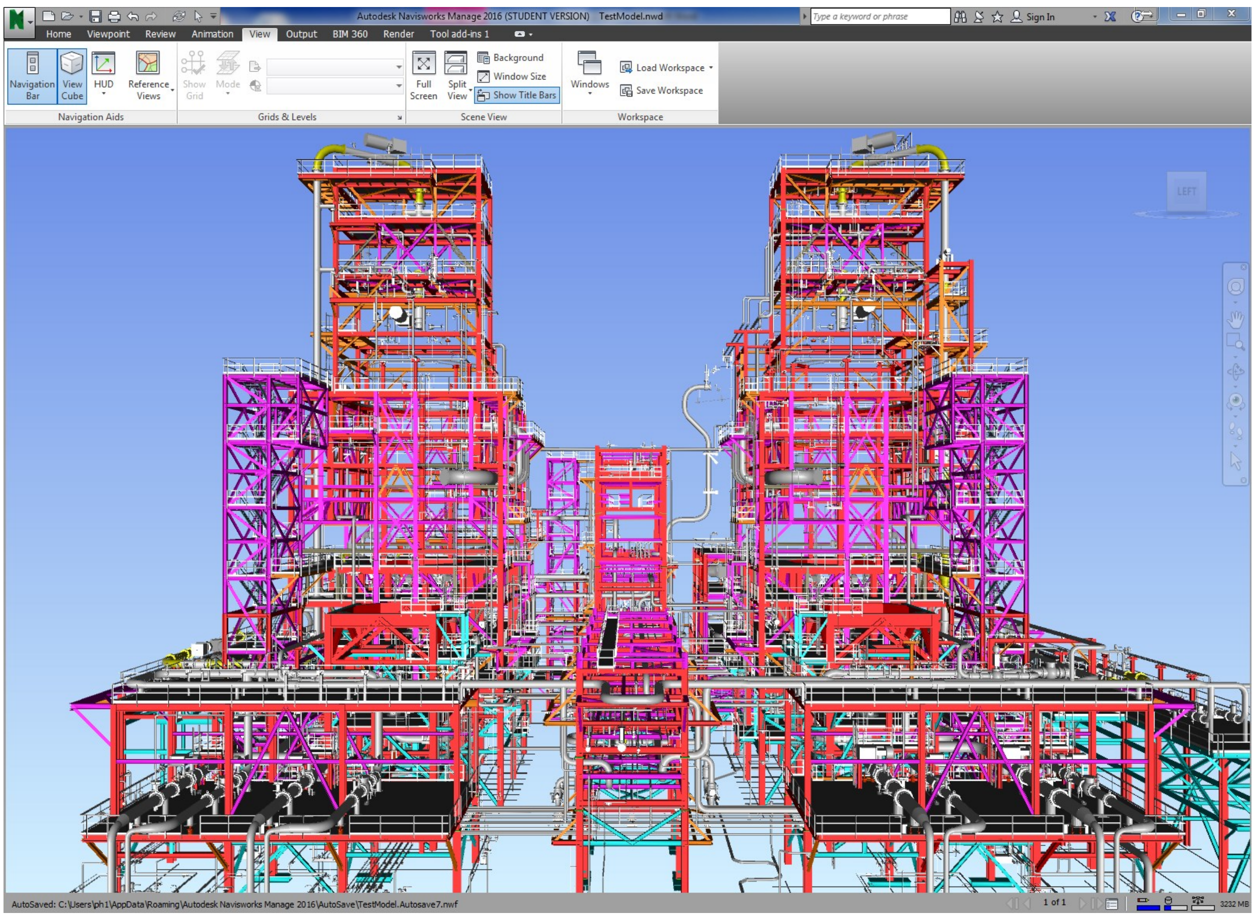


Fig. 39. The Screenshots of Model Part for Case Study

In this research, part of the components in the Piping and Steel Structure disciplines are quantified by the proposed algorithm. Each type of model component will be matched with the specific basic geometry shape. The component types and shape-matching relationship are listed in Table 4.

Table 4. Component Types in Piping and Steel Structure Disciplines and Shape-Matching Relationship

Disciplines	Component Type	Shape
Piping	Tube	Pipe
Steel Structure	Beam	I-Shaped Prism
	Column	I-Shaped Prism
	Brace	Cylinder

The geometry information of each component will be extracted from the provided 3D model and the geometry information of the same components will also be obtained from the IFC drawing quantification results. The results will be used as the real values. The algorithm calculation results and real values will be compared to verify the accuracy of the proposed method.

4.2. Calculation Results and Analysis

To verify the algorithm, one zone of the construction project is selected. In this zone, all the components of the selected type will be used for testing. The geometry information about the model components in the selected zone has been manually measured. The algorithm calculation results will be compared with the manual measurement values. Fig. 40 uses a flowchart to show the case study steps.

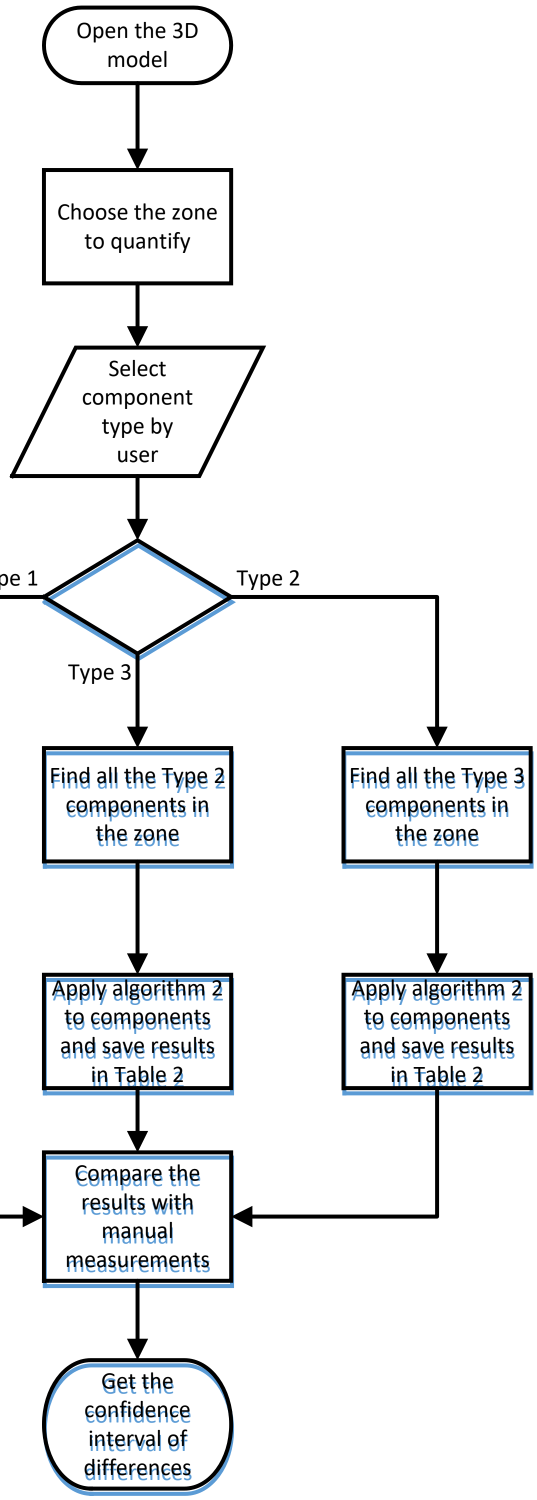
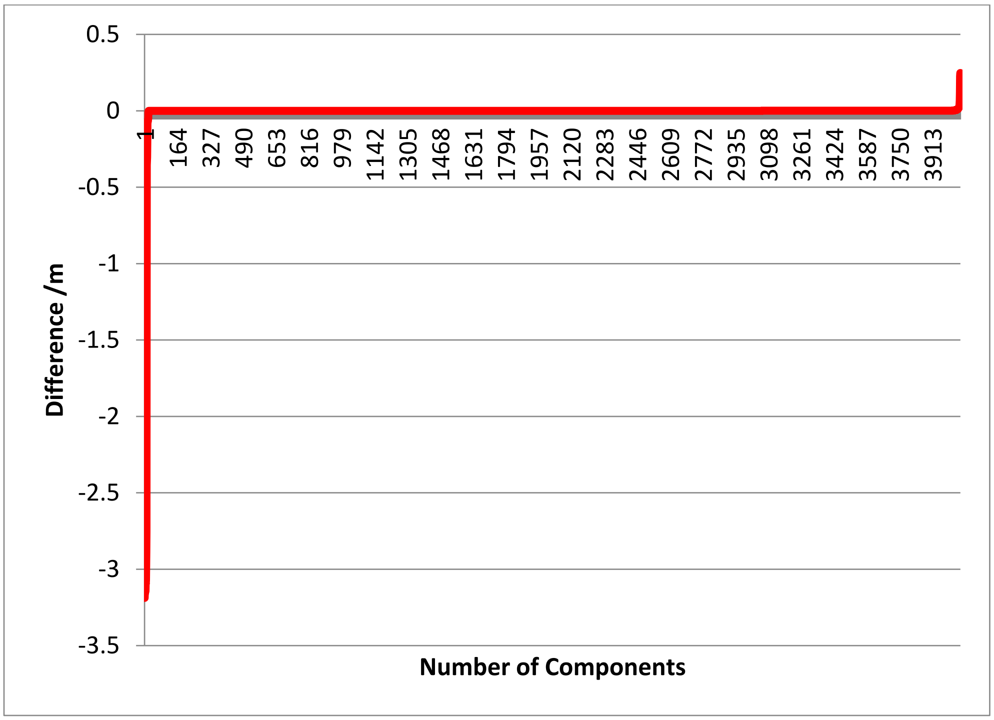


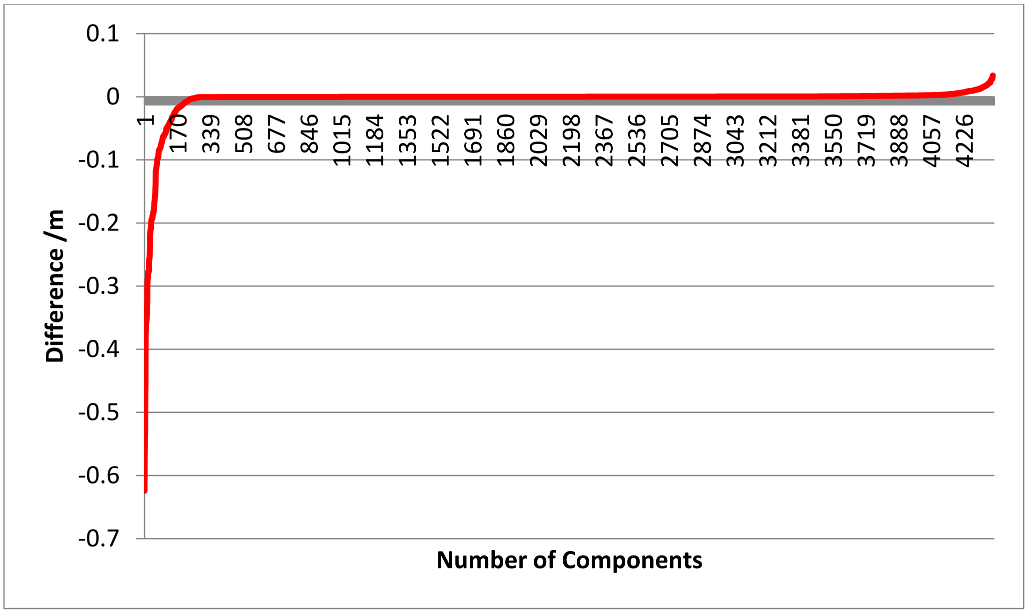
Fig. 40. The Flowchart for Case Study Steps

In this case study, the only dimension for comparison between the algorithm calculation and manual measurements is the length of every component. In the selected zone of the testing model, 4052 Tubes, 4377 Beams, 975 Columns, and 729 Braces are used for testing. The comparison results are shown below.

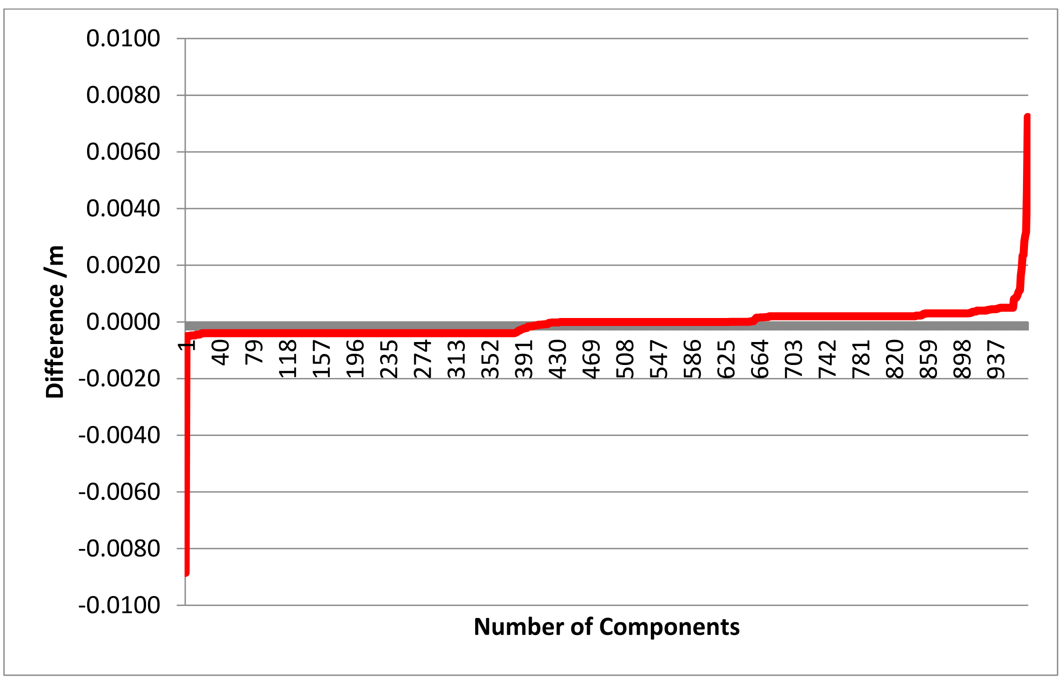
Fig. 41 lists the differences between algorithm calculation results and manual measurements of each component in ascending order. Fig. 42 shows the percentages of absolute differences that are less than 0.01 m.



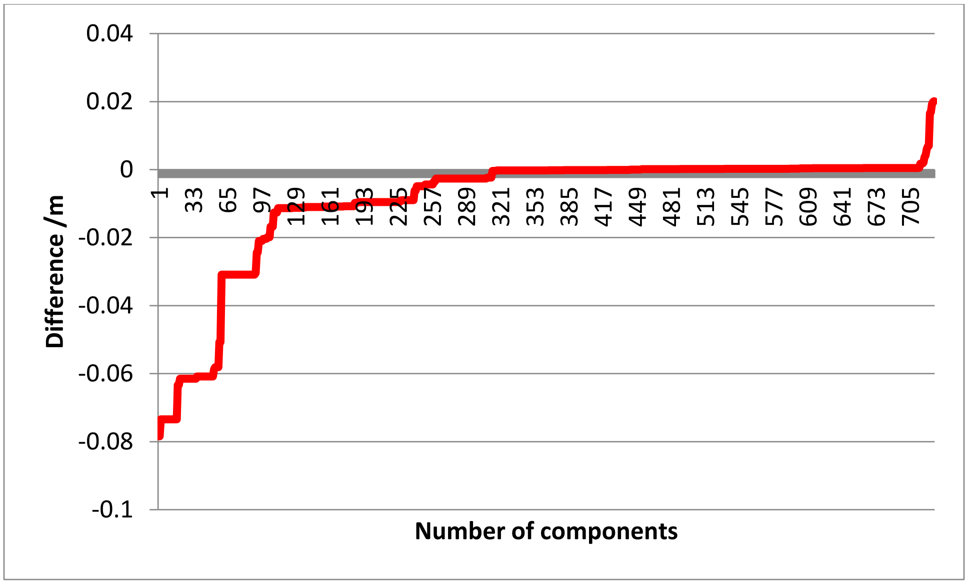
(a) Differences in The Length of Tube Components



(b) Differences in the Length of Beam Components



(c) Differences in the Length of Column Components



(d) Differences in the Length of Brace Components

Fig. 41. Differences between Calculation Results and Manually Measured Values

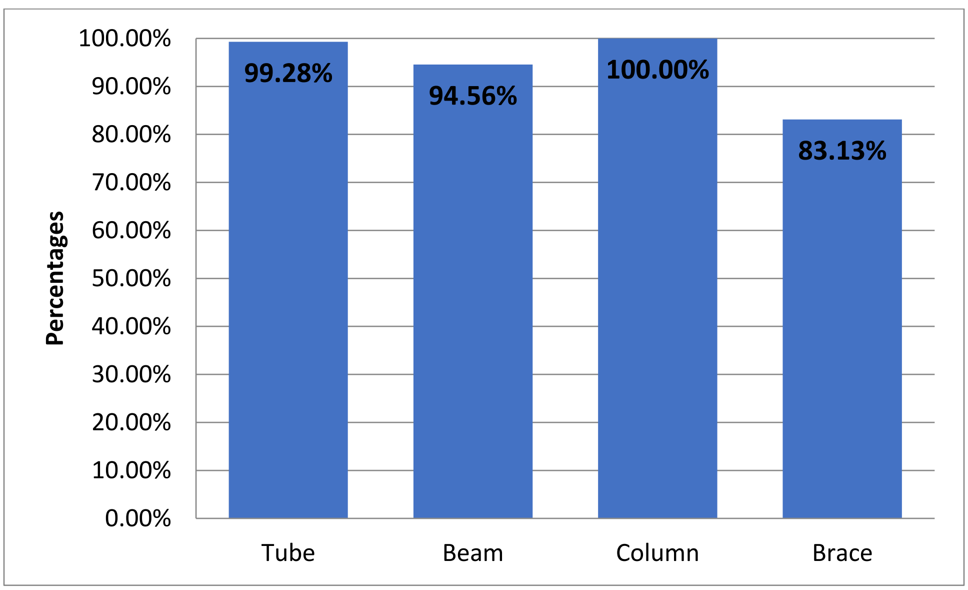


Fig. 42. The Percentages of Absolute Differences Less Than 0.01 m

The mean absolute error (MAE) and root mean square error (RMSE) are frequently used to measure the errors between predicted value and observation value. In this research, MAE is used to compare the differences between calculation results and manual measurements. RMSE is used to represent the sample standard deviation of differences between two results. The MAE and RMSE for each type of component are shown in Table 5.

Table 5. MAE and RMSE Between Calculation Results and Manual Measurements

Component Type	MAE	RMSE
Tube	0.0087	0.1543
Beam	0.0061	0.0348
Column	0.0003	0.0005
Brace	0.0096	0.0204

The differences between algorithm calculation results and manual measurements are set as d_i ($i = 1, 2, 3, \dots n$, where n is the total number of the selected components). The average of d_i is set as $\bar{d}$, and the variance is σ. The confidence interval can be calculated by:

$$\left[\bar{d} - \frac{\sigma}{\sqrt{n}} z_{\frac{\alpha}{2}}, \bar{d} + \frac{\sigma}{\sqrt{n}} z_{\frac{\alpha}{2}}\right].$$

Set confidence level as 95%, so $1 - \alpha = 0.95$ and $\alpha = 0.5$. The value of $z_{\frac{\alpha}{2}}$ can be found in the z value table as 1.96. The confidence interval of each component type is shown in Table 6.

Table 6. The Confidence Interval of Errors for Each Component Type

Component Type	Mean	Variance	Number of Components	Confidence Interval
Tube	-0.0080	0.0238	4052	[-0.0087, -0.0073]
Beam	-0.0043	0.0012	4377	[-0.0044, -0.0043]
Column	-0.0001	0.0000003	975	[-0.0001, 0.0000]
Brace	-0.0090	0.0003	728	[-0.0091, -0.0090]

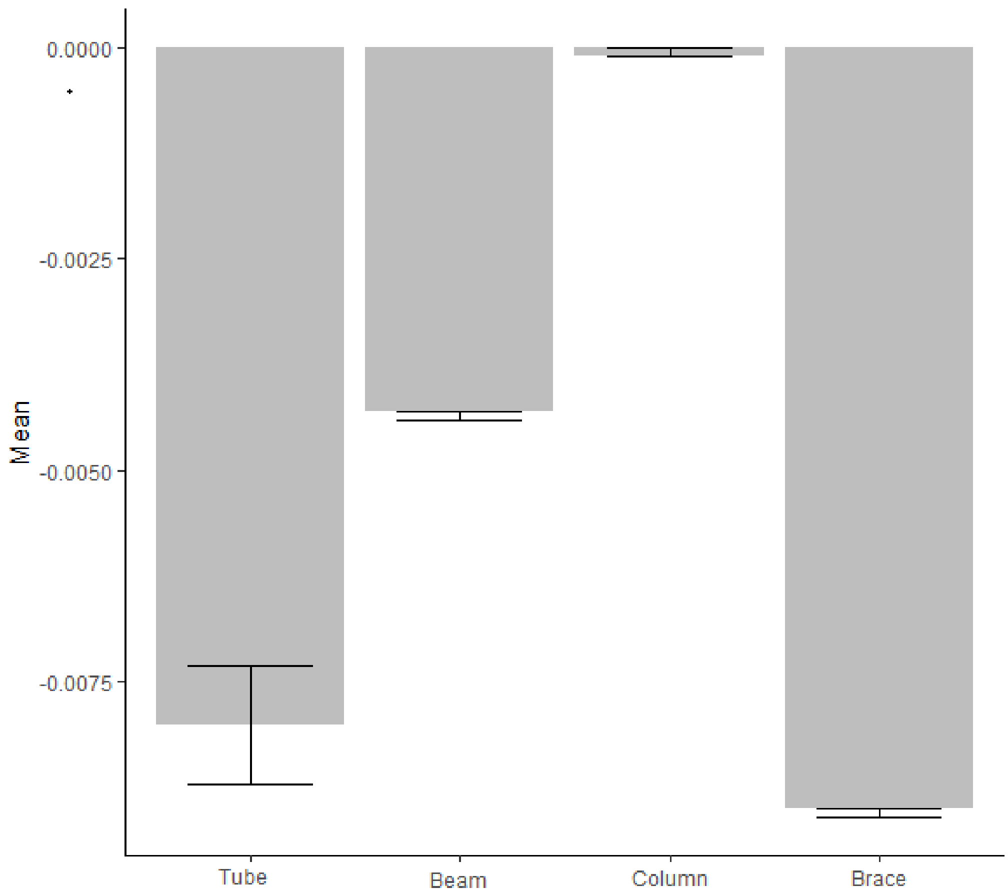


Fig. 43. The Confidence Interval of the Errors for Each Component Type

In this case study, four different component types are selected to verify the algorithms. Each algorithm shows high accuracy and stability. The differences between algorithm calculation results and manual measurements are stable; most are less than 0.01 m. The algorithm used in the case study has the potential to provide geometry information which can be used in quantity take-off.

5. Discussion and Conclusion

5.1. Overview

In industrial construction projects, Building Information Modeling (BIM)-based design, calculation and construction have been widely applied. However, BIM technologies still have limitations such as the lack of standards, inability to model-share between stakeholders, and difficulty using provided models.

This research project proposed an alternative method to assist the quantity take-off process using a 3D viewing model, which was converted from BIM and other 3D models. The method is based on the primitive triangle mesh data, which is widely used in 3D platforms. The algorithm can be easily applied to different software and used to extract geometry data.

5.2. Academic Contribution

This research combined original algorithms with existing algorithms to extract geometry information of 3D viewing model components. The research project also applied the Principal Component Analysis (PCA) algorithm to rotate model components to normal (i.e., horizontal or vertical) positions. The codes were independently built in C# languages. The work is summarized in Table 7.

Table 7. The Summary of Work in Algorithms and Codes

Geometrical Shapes	Algorithms	Codes
Cylinder	Original	Independently Built
Pipe	Original	Independently Built
Irregular shaped prism	**PCA** + Original	MathNet (Open Source) + Independently Built
Elbow	Original	Independently Built
Others	Projection Method	Independently Built

5.3. Industrial Contribution

This research project provides a tool for estimators to extract geometry information. An add-on including a user interface is built in the Navisworks platform, allowing calculation results to be automatically stored in the database. The primary functions of the add-on are listed in Table 8.

Table 8. Main Functions of the Navisworks Add-On

Functions	User Interaction Involved
Find the directory of the database	Yes
Connect to the database	No
Find all the component types in the current 3D model	No
Select the component type to quantify	Yes
Create the corresponding table in the database for the selected component type	No
Go through the selection tree to find all the components by selected type	No
Extract the geometry information of each component	No
Save the results to the created table in the database	No

5.4. Advantages and Expected Applications

The proposed method is highly accurate and efficient for geometry measurements. As the verification results show, the average absolute error is less than 0.1 m. More than 91% of components in the testing model have relative errors that are less than 1%. For the company involved in this study, the current quantity take-off process can involve months of effort using manual measurements. The algorithm can extract the geometry information for millions of components in hours. It has the potential to intensely increase the efficiency of the quantity take-off process in industrial construction projects, especially in the bidding process, where bidders usually have a tight timeline for preparing bidding documents. The proposed method provides an efficient and comparative accurate quantification tool to improve a company's ability to be competitive in the bidding process.

The method can also integrate models from different platforms. For highly complex projects, such as oil refineries and power stations, the design models can be finished on different platforms. It is not practical to separately implement quantity take-off on each platform. A 3D model platform, such as Navisworks, can be used to integrate all the models and generate information for quantification.

The proposed algorithm has potential to be applied to other 3D viewing platforms based on triangle mesh data, such as 3D max and SketchUp. At present, the algorithm is based on a Naviswork platform and is in C# language. The inputting information of the algorithm is the triangle mesh-based data, which is widely applied in 3D viewing platforms. The algorithm will be encapsulated as a standard library, so it can be directly imported and used in other programs and languages. The range of applications will be expanded to multiplatform.

In some cases, BIM model owners are unwilling to share their models with general contractors or sub-contractors, mainly because of concerns about privacy of information. However, 3D models can not be edited and most sensitive information is lost during the importing. The model used in this method can promote the model-sharing mode. That means the BIM model owner can share the converted 3D model with other stakeholders with less risk.

Even if a well-designed BIM model is provided to a construction company, the proposed method may still be used to verify quantified results. BIM-based quantity take-off requires a high level of collaboration between designers and constructions. Misunderstandings and conflicts can arise when data exchange standards are not well defined or adopted between two companies, and certain model components may be missed in quantity take-off results. The geometry information generated by BIM models could also be inaccurate if designs are not well modeled. The method proposed in this project generates geometry information directly from primitive data and is not, therefore, limited by such constraints. The method is also capable of identifying and selecting all components required for quantification. Accordingly, the results can be used to verify the BIM-based quantification results.

5.5. Conclusion

To improve the material quantification process in industrial construction projects, the research proposed an alternative method to implement quantity take-off based on 3D viewing models. The geometry information for quantification is absent in 3D models; the proposed algorithm is applied to recognize, extract and calculate the geometrical features of the components in the model. The algorithm built in this project is based on primitive triangle-mesh data, which is widely used in 3D model platforms. The verification results show that the proposed algorithm can accurately and efficiently extract the geometry information from the 3D model. The results generated by the algorithm can be used to generate quantity take-off results.

Even though BIM-based quantity take-off has been applied in industrial construction projects, there are still limits to the technical and contract aspects. The 3D model used in this research is exported from the BIM model or other platforms, which have higher compatibility and fewer contract risks. The algorithm proposed in this project cannot fully automate the material quantification process of a construction project. However, it shows the potential to apply 3D viewing models in quantity take-off and cost estimation. It is an alternative method to efficiently implement quantity take-off without using the original BIM model.

The proposed method can improve the efficiency of the quantity take-off process in the following three ways:

First, the proposed algorithm automatically provides geometry information to estimators based on 3D viewing models. It relieves the workload, as estimators no longer have to go through 2D drawings, interpret them, and input geometry data to generate the calculation spreadsheet. The geometry information required for quantity take-off is generated from 3D viewing model and stored in the database. The estimators use the database to quantify the construction components.

Second, the proposed algorithm can be used to validate the quantity take-off results. In industrial construction projects, estimators usually implement the quantity take-off several times to ensure the accuracy of the results. Even though the proposed method is not capable of fully automatically implementing the quantity take-off, the calculation result can be used as intermediate step to validate the result of current practices.

Third, the proposed method can be used to generate a checklist of model components. In most cases, there are millions of components in an industrial construction project. It is inevitable that some will be missed. Especially for some complicated structures, the same component will be represented in several separate 2D drawings in different views. The proposed method will traverse the whole 3D model to find every model component. It can generate a checklist of components to assist the estimators in quantifying every component without repetition or omission.

This research proposed an alternative method to assist the quantity take-off process based on 3D viewing models. Compared with the current practices, the proposed method has advantages in efficiency, usability and platform compatibility. However, there are still limitations: the range of applications is limited, and the more detailed information needs to be extracted. The limitations can be addressed by extending the geometry shapes that the algorithm matches and the geometry details that can be extracted. The proposed method has the potential to improve the efficiency and accuracy of the quantity take-off process.

5.6. Limitations

For the current proposed approach, the following limitations exist:

The algorithm cannot be applied to undefined geometry shapes. The component type and geometry shape have to be matched before the algorithm is calculated. For some components with complicated shapes, such as a valve, it is hard to automatically measure the geometrical features. The unmatched components still need to be measured manually.

The algorithm cannot automatically recognize the component type and its geometry shape, although the shapes can be partially recognized based on the geometry information (Ali et al., 2016) , or from the semantic and ontology aspect (Forgues et al., 2012; Lee, Kim, & Yu, 2014). However, the algorithm has not been adapted into this research project and therefore some components still need to be manually recognized.

The algorithm cannot directly provide accurate information about materials, which is necessary for quantity take-off and cost estimation. In the 3D viewing models, the material information about the components is lost during the export and import process. In some cases, the material information can be concluded based on the discipline and component type, such as steel structures. However, in other cases, it is hard to determine the component's material . For example, the foundation piles could be concrete or steel. Identifying the material requires other approaches, such as obtaining data from 2D drawings and design descriptions.

5.7. Future Work

The algorithm will be extended to match more shapes to extract geometry information. For example, other common component types exist in industrial construction projects, such as the reducer and bend in the piping system, and the steel plate in steel structures. In most cases, each component type has a certain geometry shape. A method similar to the one proposed in this thesis can be applied to extract geometry information about those other types of components. By increasing the range of matched geometry shapes, more component types will be automatically quantified based on the 3D viewing model.

The proposed algorithm will also go deeper into extracting detailed geometry information for some component types. At present, the algorithm only extracts the rough height and width of the cross section for steel structures. The flange thickness and web thickness of steels cannot be extracted. The algorithm will extract detailed cross-section geometry information based on different cross-section shapes.

For components such as tubes, 3D graphic representations are usually modeled as a cylinder. The thickness of the component is not represented in the graphic models. Consequently, the current proposed method cannot obtain the thickness of tubes from the 3D graphic representation. However, the thickness can be obtained by other methods. For example, some 3D viewing models retain the extra properties of each model component. The thickness can be obtained by using application program interface (API) to access the properties.

References

Autodesk Inc. (2016). Supported file formats and applications for Autodesk Navisworks. Retrieved July 15, 2017, from https://knowledge.autodesk.com/support/navisworks-products/learn-explore/caas/sfdcarticles/sfdcarticles/Supported-file-formats-and-applications-for-Autodesk-Navisworks-2009-to-2012.html

Abanda, F. H., Kamsu-Foguem, B., & Tah, J. . H. M. (2015). Towards an intelligent ontology construction cost estimation system: Using BIM and New Rules of Measurement techniques. *World Acad.Sci.Eng.Technol.Int.J.Comput.Electr.Autom.Control Inf.Eng*, *9*, 294–299.

Ali, M., & Mohamed, Y. (2017). A method for clustering unlabeled BIM objects using entropy and TF-IDF with RDF encoding. *Advanced Engineering Informatics*, *33*, 154–163. https://doi.org/10.1016/j.aei.2017.06.005

Ali, M., Mohamed, Y., & Hermann, R. (2016). Automated Recognition of Unlabeled Items in BIM Models. *Construction Research Congress 2016*, *2*, 2039–2049. https://doi.org/10.1061/9780784479827.203

Ali, M., Mohamed, Y., Taghaddos, H., & Hermann, R. (2015). BIM Obstacles in Industrial Projects : a Contractor Perspective. *Proceedings of ICSC15: The Canadian Society for Civil Engineering 5th International/11th Construction Specialty Conference*, 1–10. https://doi.org/10.14288/1.0076376

Anton, H. (2010). *Elementary Linear Algebra*. John Wiley & Sons.

Anumba, C. J., Pan, J., Issa, R. R. A., & Mutis, I. (2008). Collaborative project information management in a semantic web environment. *Engineering, Construction and Architectural Management*, *15*(1), 78–94.

Azhar, S. (2011). Building Information Modeling (BIM): Trends, Benefits, Risks, and Challenges for the AEC Industry. *Leadership & Management in Engineering*. https://doi.org/10.1061/(ASCE)LM.1943-5630.0000127

BuildingSMART International Ltd. (2017). IFC Overview summary. Retrieved July 15, 2017, from http://www.buildingsmart-tech.org/specifications/ifc-overview

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms*. MIT press.

CRC Construction Innovation. (2009). *National Guidelines for Digital Modelling*. *Icon. Net Pty Ltd., Brisbane*. Retrieved from http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:National+Guidelines+for+Digital+Modelling#0

Dawkins, P. (2007). Equations of Planes, Calculus;III. Retrieved from http://tutorial.math.lamar.edu/Classes/CalcIII/EqnsOfPlanes.aspx

Du, J., Zou, Z., Shi, Y., & Zhao, D. (2017). Simultaneous Data Exchange between BIM and VR for Collaborative Decision Making. In *Computing in Civil Engineering 2017* (pp. 1–8).

Dunteman, G. H. (1989). *Principal components analysis*. Sage.

Eastman, C. M., Eastman, C., Teicholz, P., & Sacks, R. (2011). *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, And Contractor*. Hoboken, N.J.: John Wiley & Sons. Retrieved from http://site.ebrary.com/lib/ualberta/Doc?id=10460861

Farah, T., & Guillermo F. Salazar. (2005). Review of Current Estimating Capabilities of the 3D Building Information Model Software to Support Design for Production/Construction, 1–129.

Fernandes, K. J., Raja, V., White, A., & Tsinopoulos, C.-D. (2006). Adoption of virtual reality within construction processes: a factor analysis approach. *Technovation*, *26*(1), 111–120.

Flyvbjerg, B., Bruzelius, N., & Rothengatter, W. (2003). *Megaprojects and risk : an anatomy of ambition*. United Kingdom ;New York: Cambridge University Press. Retrieved from http://www.loc.gov/catdir/samples/cam034/2002074193.html

Forgues, D., Iordanova, I., Valdivesio, F., & Staub-French, S. (2012). Rethinking the Cost Estimating Process through 5D BIM: a Case Study Daniel. In *Construction Research Congress 2012* (pp. 1450–1459). https://doi.org/10.1061/9780784412329.146

Fu, C., Kaya, S., M, K., & G, A. (2007). The development of an IFC-based lifecycle costing prototype tool for building construction and maintenance: integrating lifecycle costing to nD modelling. *CONSTRUCTION INNOVATION -LONDON-*, *7*(1), 85–98. Retrieved from http://login.ezproxy.library.ualberta.ca/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edsbl&AN=RN202864786&site=eds-live&scope=site

Hartmann, T., Gao, J., & Fischer, M. (2008). Areas of Application for 3D and 4D Models on Construction Projects. *Journal of Construction Engineering and Management*, *134*(10), 776–785. https://doi.org/10.1061/(ASCE)0733-9364(2008)134:10(776)

Hinks, T., Carr, H., Truong-hong, L., & Laefer, D. F. (2013). Point Cloud Data Conversion into Solid Models via Point-Based Voxelization. *Journal of Surveying Engineering*, *139*(2), 72–83. https://doi.org/10.1061/(ASCE)SU.1943-5428.0000097.

Holm, L., Schaufelberger, J. E., Griffin, D., & Cole, T. (2005). *Construction cost estimating: process and practices*. Prentice Hall.

Hoppe, H., Derose, T., Mcdonald, J., & Stuetzle, W. (1993). Mesh Optimization, *d*, 19–26.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, *24*(6), 417–441. https://doi.org/10.1037/h0071325

Howard, R., & Björk, B. C. (2008). Building information modelling - Experts' views on standardisation and industry deployment. *Advanced Engineering Informatics*, *22*(2), 271–280. https://doi.org/10.1016/j.aei.2007.03.001

Hu, X., Lu, M., & Abourizk, S. (2015). BIM-based data mining approach to estimating job man-hour requirements in structural steel fabrication. *Proceedings - Winter Simulation Conference*, *2015–Janua*(1983), 3399–3410. https://doi.org/10.1109/WSC.2014.7020173

Hughes, J. F., Dam, A. van, McGuire, M., Sklar, D. F., Foley, J. D., Feiner, S. K., & Akeley, K. (2014). *Computer graphics: Principles and practice*. *Addison-Wesley Professional* (Vol.

22). Retrieved from http://doi.wiley.com/10.1002/%28SICI%291520-6378%28199702%2922%3A1%3C65%3A%3AAID-COL11%3E3.0.CO%3B2-7

Iyer, N., Jayanti, S., Lou, K., Kalyanaraman, Y., & Ramani, K. (2005). Three-dimensional shape searching: state-of-the-art review and future trends. *Computer-Aided Design*, *37*(5), 509–530. https://doi.org/10.1016/j.cad.2004.07.002

Leach, P. J., Mealling, M., & Salz, R. (2005). A Universally Unique Identifier (UUID) URN Namespace.

Lee, S.-K., Kim, K.-R., & Yu, J.-H. (2014). BIM and ontology-based approach for building cost estimation. *Automation in Construction*, *41*, 96–105. https://doi.org/10.1016/j.autcon.2013.10.020

Leite, F., Akinci, B., & James, J. G. (2009). Identification of data items needed for automatic clash detection in MEP design coordination. In *Construction Research Congress 2009: Building a Sustainable Future* (pp. 416–425).

Monteiro, A., & Poças Martins, J. (2013). A survey on modeling guidelines for quantity takeoff-oriented BIM-based design. *Automation in Construction*, *35*, 238–253. https://doi.org/10.1016/j.autcon.2013.05.005

National BIM Standard. (2017). Frequently Asked Questions About the National BIM Standard-United States. Retrieved July 15, 2017, from https://www.nationalbimstandard.org/faqs

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *2*(11), 559–572.

Rossignac, J., & Borrel, P. (1993). Multi-Resolution 3D Approximations for Rendering Complex Scenes. *Modeling in Computer Graphics: Methods and Applications*, *465*.

Rundell, R. (2006). 1-2-3 Revit: BIM and cost estimating. Retrieved July 27, 2017, from http://www.cadalyst.com/cad/building-design/1-2-3-revit-bim-and-cost-estimating-part-1-3350

Sattineni, A., & Bradford, R. (2011). Estimating with bim: a survey of US construction companies. *BIM-Cost Estimating*, 564–569. Retrieved from http://www.iaarc.org/publications/fulltext/S16-6.pdf

Simonian, L. (2010). Legal considerations associated with building information modeling. *Cal Poly San Luis Obispo College of Architecture & Environmental Design*.

Smith, L. I. (2002). A tutorial on Principal Components Analysis. *Statistics*, *51*, 52. https://doi.org/10.1080/03610928808829796

Smith, P. (2014). BIM & the 5D Project Cost Manager. *Procedia - Social and Behavioral Sciences*, *119*, 475–484. https://doi.org/http://dx.doi.org/10.1016/j.sbspro.2014.03.053

Staub–French, S., Fischer, M., Kunz, J., Ishii, K., & Paulson, B. (2003). A feature ontology to support construction cost estimating. *Ai Edam*, *17*(May 2016), 133–154. https://doi.org/10.1017/S0890060403172034

Tangelder, J. W. H., & Veltkamp, R. C. (2007). A survey of content based 3D shape retrieval methods. *Multimedia Tools and Applications*, *39*(3), 441–471. https://doi.org/10.1007/s11042-007-0181-0

Tiwari, S., Odelson, J., Watt, A., & Khanzode, A. (2009). Model Based Estimating to Inform Target Value Design. *AECBytes" Building the Future*.

Volk, R., Stengel, J., & Schultmann, F. (2014). Building Information Modeling (BIM) for existing buildings — Literature review and future needs. *Automation in Construction*. https://doi.org///dx.doi.org/10.1016/j.autcon.2013.10.023

Young, J. N. W. ., Jones, S. A., Bernstein, H. M., & Gudgel, J. E. (2009). *THE BUSINESS VALUE OF BIM Getting Building Information Modeling to the Bottom Line*. *McGraw Hill Construction - SmartMarket Report Construction*. Retrieved from http://bim.construction.com/research/pdfs/2009_BIM_SmartMarket_Report.pdf

Zhang, C., & Chen, T. (2001). Efficient feature extraction for 2D/3D objects in mesh representation. *Proceedings 2001 International Conference on Image Processing (Cat. No.01CH37205)*, *3*, 935–938. https://doi.org/10.1109/ICIP.2001.958278

Zhao, P. A., & Wang, C. C. (2014). A Comparison of Using Traditional Cost Estimating Software and BIM for Construction Cost Control. *Iccrem 2014*, 256–264. https://doi.org/10.1061/9780784413777.031

Zhiliang, M., Zhenhua, W., Wu, S., & Zhe, L. (2011). Application and extension of the IFC standard in construction cost estimating for tendering in China. *Automation in Construction*, *20*(2), 196–204. https://doi.org/10.1016/j.autcon.2010.09.017

Appendix I:

The Manually Measuring and Algorithm Calculation Results for Tubes

	Length /m		Diameter/m	
	Manually	Algorithm	Manually	Algorithm
TUBE 1 of BRANCH /612P21029-42"-CAB-01/B4	0.865	0.865	0.610	0.609
TUBE 1 of BRANCH /612P21029-42"-CAB-01/B5	0.865	0.865	0.610	0.609
TUBE 1 of BRANCH /612P21093-24"-CAB-01/B1	2.245	2.245	0.610	0.610
TUBE 2 of BRANCH /612P21093-24"-CAB-01/B1	2.675	2.675	0.610	0.610
TUBE 1 of BRANCH /612P21094-24"-CAB-01/B1	2.695	2.695	0.610	0.610
TUBE 2 of BRANCH /612P21094-24"-CAB-01/B1	2.675	2.675	0.610	0.610

The Manually Measuring and Algorithm Calculation Results for Elbows

	Length/m		Radius/m		Angle	
	Manually	Algorithm	Manually	Algorithm	Manually	Algorithm
ELBOW 1 of BRANCH /612P21093-24"-CAB-01/B1	1.43	1.429	0.61	0.609	90	90
ELBOW 2 of BRANCH /612P21093-24"-CAB-01/B1	1.43	1.429	0.61	0.609	90	90
ELBOW 1 of BRANCH /612P21094-24"-CAB-01/B1	1.43	1.429	0.61	0.61	90	90
ELBOW 2 of BRANCH /612P21094-24"-CAB-01/B1	1.43	1.429	0.61	0.61	90	90

The Manually Measuring and Algorithm Calculation Results for Steel Structures

	Length /10m		Height /m		Width /m	
	Manually	Algorithm	Manually	Algorithm	Manually	Algorithm
SCTN 2 of FRMWORK /612-2101PM112(TOS-380550)-6-M-STL-BEAM	0.225	0.225	0.201	0.201	0.165	0.165

SCTN 1 of FRMWORK /612-2101PM112(TOS-383600)-6-M-STL-BEAM	0.225	0.225	0.247	0.247	0.202	0.202
SCTN 4 of FRMWORK /612-2101PM112(TOS-383600)-6-M-STL-BEAM	0.225	0.225	0.247	0.247	0.202	0.202
SCTN 3 of FRMWORK /612-2101PM112(TOS-378100)-6-M-STL-BRACE	0.3147	0.31479	0.165	0.1843	0.1	0.1
SCTN 1 of FRMWORK /612-2101PM112-6-M-STL-BRACE	0.5013	0.50148	0.247	0.247	0.202	0.202
SCTN 2 of FRMWORK /612-2101PM112-6-M-STL-BRACE	0.5013	0.50148	0.232	0.247	0.202	0.202
SCTN 5 of FRMWORK /612-2101PM112-6-M-STL-BRACE	0.407	0.40737	0.253	0.254	0.254	0.253
SCTN 6 of FRMWORK /612-2101PM112-6-M-STL-BRACE	0.407	0.40737	0.253	0.254	0.254	0.253
FITTING 1 of SCTN 1 of FRMWORK /612-2101PM112-6-M-STL-COL	0.645	0.645	0.362	0.362	0.355	0.355
FITTING 1 of SCTN 2 of FRMWORK /612-2101PM112-6-M-STL-COL	0.645	0.645	0.362	0.362	0.355	0.355
FITTING 1 of SCTN 4 of FRMWORK /612-2101PM112-6-M-STL-COL	0.645	0.645	0.362	0.362	0.355	0.355
FITTING 1 of SCTN 5 of FRMWORK /612-2101PM112-6-M-STL-COL	0.71	0.71	0.362	0.362	0.355	0.355
FITTING 1 of SCTN 1 of FRMWORK /612-2101PM112(TOS-378100)-6-M-STL-BEAM	0.225	0.225	0.271	0.271	0.25	0.25
FITTING 1 of SCTN 2 of FRMWORK /612-2101PM112(TOS-378100)-6-M-STL-BEAM	0.225	0.225	0.225	0.225	0.213	0.213
FITTING 1 of SCTN 7 of FRMWORK /612-2101PM112(TOS-378100)-6-M-STL-BEAM	0.44	0.44	0.271	0.271	0.25	0.25
FITTING 1 of SCTN 9 of FRMWORK /612-2101PM112(TOS-378100)-6-M-STL-BEAM	0.44	0.44	0.271	0.271	0.25	0.25
FITTING 1 of SCTN 1 of FRMWORK /612-2101PM112(TOS-380550)-6-M-STL-BEAM	0.225	0.225	0.271	0.271	0.25	0.25
FITTING 1 of SCTN 3 of FRMWORK /612-2101PM112(TOS-380550)-6-M-STL-BEAM	0.225	0.225	0.271	0.271	0.25	0.25
FITTING 1 of SCTN 8 of FRMWORK /612-2101PM112(TOS-380550)-6-M-STL-BEAM	0.44	0.44	0.434	0.434	0.227	0.227
FITTING 1 of SCTN 10 of FRMWORK /612-2101PM112(TOS-380550)-6-M-STL-BEAM	0.44	0.44	0.434	0.434	0.227	0.227
FITTING 1 of SCTN 11 of FRMWORK /612-2101PM112(TOS-380550)-6-M-STL-BEAM	0.2	0.2	0.271	0.271	0.25	0.25
FITTING 1 of SCTN 13 of FRMWORK /612-2101PM112(TOS-380550)-6-M-STL-BEAM	0.2	0.2	0.271	0.271	0.25	0.25
FITTING 1 of SCTN 14 of FRMWORK /612-2101PM112(TOS-380550)-6-M-STL-BEAM	0.575	0.575	0.271	0.271	0.25	0.25
FITTING 1 of SCTN 2 of FRMWORK /612-2101PM112(TOS-383600)-6-M-STL-BEAM	0.225	0.225	0.271	0.271	0.25	0.25
FITTING 1 of SCTN 9 of FRMWORK /612-2101PM112(TOS-383600)-6-M-STL-BEAM	0.44	0.44	0.434	0.434	0.227	0.227
FITTING 1 of SCTN 11 of FRMWORK /612-2101PM112(TOS-383600)-6-M-STL-BEAM	0.44	0.44	0.434	0.434	0.227	0.227