

OSLO METROPOLITAN UNIVERSITY
STORBYUNIVERSITETET

Master's Degree in
Structural Engineering and Building Technology
Department of Civil Engineering and Energy Technology

MASTER THESIS

THESIS TITLE AUTOMATED REVERSE MODELING AND EXAMINATION OF AS-BUILT STEEL TRUSS BASED ON POINT CLOUD DATA, MACHINE LEARNING AND BIM	DATE 09.06.2021
	NUMBER OF PAGES 86 (30)
AUTHOR(S) ERAIKUMARAN SELVAKUMARAN	SUPERVISOR(S) Professor Daguang Han

IN COLLABORATION WITH -	CONTACT PERSON -
--------------------------------	-------------------------

<p>SUMMARY</p> <p>Building Information Modeling (BIM) has improved the way structures are designed and constructed over the last decades. Although BIM implementation is an expected part of newer construction projects, the technology is heavily under-represented for use cases other than design. The research presented in this paper has a basis in creating a reliable way of reverse modeling an as-built steel truss to evaluate whether or not the component is suited for use. To do so, a reverse model of the as-built steel truss is created using scanned point cloud data. Results find the proposed method to deliver very accurate results. Digitally examining the reverse modeled steel truss ensures that the component is very similar to the designed model, and further FEA concludes that the component is suited for use. The proposed method lays the foundational ground for using modern technologies such as 3D laser scanning, parametric BIM modeling, machine learning, computer vision, and other BIM software to develop algorithms, scripts, and workflows for reverse modeling and examining as-built steel components digitally. However, making improvements and building on top of the proposed method will open up the possibilities for the technology to be used for other purposes than described in this paper.</p>
--

3 KEYWORDS
Reverse BIM Modeling
Point Cloud
Machine Learning

Problem Statement

Steel structures today are mostly prefabricated as separate components and assembled on the construction site. This process is relatively inefficient as inaccuracies in a component can lead to massive delays and increased costs as construction gets halted. Trial assemblies have been a trusted method used in the AEC-Industry to examine and ensure that components arriving at the construction site are as designed. Such trials are expensive and can cost up to 25% of the total cost [1]. A cheaper and more viable option is to digitize the examination process. However, digitizing the process will need an accurate BIM model representation of the as-built component.

In this thesis, a process of reverse modeling a physical steel component to a digital 3D model based on acquired scan data is created and investigated. The thesis aims to automate as much of the reverse modeling process as possible with the resources at hand by using machine learning and other suitable technologies. The accuracy of the generated model in relation to the acquired scan data is to be analyzed, indicating whether or not the proposed modeling method is of sufficient quality. The model will then undergo a geometric deviation comparison to evaluate the as-built structure's accuracy compared to the designed model. A FEA will then be conducted to ensure that deflections caused by the dead load during assembly are within the ramifications of the standards.

Finally, based on the conducted research, an evaluation is formed on whether or not the steel truss is eligible for use.

Abstract

Building Information Modeling (BIM) has improved the way structures are designed and constructed over the last decades. Although BIM implementation is an expected part of newer construction projects, the technology is heavily under-represented for use cases other than design.

The research presented in this paper has a basis in creating a reliable way of reverse modeling an as-built steel truss to evaluate whether or not the component is suited for use. To do so, a reverse model of the as-built steel truss is created using scanned point cloud data. Results find the proposed method to deliver very accurate results. Digitally examining the reverse modeled steel truss ensures that the component is very similar to the designed model, and further FEA concludes that the component is suited for use.

The proposed method lays the foundational ground for using modern technologies such as 3D laser scanning, parametric BIM modeling, machine learning, computer vision, and other BIM software to develop algorithms, scripts, and workflows for reverse modeling and examining as-built steel components digitally. However, making improvements and building on top of the proposed method will open up the possibilities for the technology to be used for other purposes than described in this paper.

Preface

This master thesis is the concluding work of one student at the Structural Engineering and Building Technology program at Oslo Metropolitan University, written through the spring of 2021.

The intended research was throughout the fall of 2020 planned to be executed in collaboration with South West University and Chongqing Smart City and Sustainable Development Academy (SCSDA) in Chongqing, China. However, Oslo Metropolitan University canceled all forms for exchange programs due to the uncertainties caused by the infectious disease COVID-19. On that basis, the thesis was fully written in Norway with assistance and support from Prof. Daguang Han and his research team in China. Although the large distance and the vast time difference between the cooperating parties led to difficulties, I would like to extend a sincere thank you to everyone who has been supporting us in this period. Especially, I would like to thank my supervisor, Professor Daguang Han, and Kaixin Hu for their impeccable support during this thesis. Their expertise, knowledge, and guidance within this field of research have been vital for completing this thesis.

I am glad to have had the opportunity to acquire extensive knowledge of new and developing software and topics that will characterize the AEC-Industry in the coming years. Machine learning, reverse modeling, and 3D laser scanning is all technologies that will have a future in the AEC-Industry. I am glad to have contributed to their development with the research conducted in this paper.

Contents

Problem Statement	i
Abstract	ii
Preface	iii
1 Introduction	1
1.1 Research background and significance	4
1.2 Research Status	5
1.2.1 BIM	5
1.2.2 Digital Twins	6
1.2.3 3D Laser Scanning	7
1.2.4 Point Cloud Reverse BIM Modeling	8
1.2.5 Machine Learning	10
1.3 Research Content and Technical Route	12
1.3.1 Research Content	12
1.3.2 Technical Route	13
1.4 Applied tools	15
1.4.1 Hardware	15
1.4.2 Software	15
2 Designed BIM model	18
2.1 Designed Intersection Point Extraction	18
2.2 BIM Design	21
3 Steel Truss Point Cloud Data Collection	
Technology	24
3.1 Working Principle of Point Cloud Acquisition	25
3.2 Accuracy of 3D Laser Scanner	26
3.2.1 3D Laser Scanner Reliability	26
3.2.2 Data Volume	27

3.3	Common Point Cloud Collection Methods	29
3.3.1	Airborne Laser Scanning	29
3.3.2	Mobile Laser Scanning	30
3.3.3	Terrestrial Laser Scanning (TLS)	31
4	Steel Truss Point Cloud Collection and Processing	32
4.1	Steel Truss Point Cloud Collection	32
4.1.1	Scanning Scheme Design	32
4.1.2	Steel Truss Scanning	33
4.2	Pretreatment of Steel Truss Point Cloud	34
4.2.1	Steel Truss Point Cloud Merging	34
4.2.2	Steel Truss Point Cloud Noise Reduction	40
5	Intersection Points Extraction	42
5.1	Sectioning & Element Center Point Extraction	43
5.1.1	Mean Shift Algorithm	48
5.1.2	Error Handling	49
5.2	Element Recognition	52
5.2.1	Element Recognition Algorithm	54
5.2.2	Error Handling	56
5.3	Intersection Point Prediction	57
5.3.1	Intersection Point Prediction Algorithm	59
5.3.2	Error Handling	61
6	Reverse BIM Model Update and Accuracy Control	62
6.1	BIM Model Update	62
6.2	Accuracy Control	66
6.2.1	Point Cloud and Updated As-Built Model Accuracy Analysis	66
6.2.2	Geometric Deviation Analysis	66
6.3	Summary	69

7 Steel Truss FEA	70
7.1 Introduction to FEA	70
7.2 Model Generation	72
7.2.1 Designed Steel Truss Model	72
7.2.2 Updated As-Built Steel Truss Model	74
7.3 Finite Element Analysis (FEA)	76
7.3.1 Designed Steel Truss Model	77
7.3.2 Updated As-Built Steel Truss Model	78
7.4 Summary FEA	78
8 Discussion and Prospect	79
9 Conclusion	82
References	83
Appendix	87

List of Figures

1	Year - Height: Home Insurance Building [5], Eiffel Tower [6], Burj Khalifa [7]. . .	2
2	Reverse Engineered BIM model based on Point Cloud Data [8].	3
3	Physical building & Digital Twin [14].	6
4	Point cloud representation of a laser scanned floor [17].	8
5	Revenues from the artificial intelligence for enterprise applications market world-wide, from 2016 to 2025 (in million U.S. dollars) [21].	11
6	Location of Chongqing in China	12
7	Laser Scanner and Scanned Steel Truss.	13
8	Technical Route of this Paper.	14
9	DWG drawing of the steel truss visualized in AutoCAD.	18
10	Intersection point creation process.	19
11	Imported Intersection Points in CATIA.	21
12	Workflow progression in CATIA. Adding lines and Planes.	21
13	Workflow progress. From 2D sketching to creating 3D Element.	22
14	Resulting BIM model design.	22
15	Terrestrial Laser Scanning process with scanned surface in red and un-scanned surface in black.	25
16	2 scans with overlapping points.	26
17	Scan decreasing density with distance.	28
18	Airborne Scan using Drone on Construction Site [34]	29
19	LiDAR for self-driving cars [35].	30
20	3D laser scan of house [36].	31
21	Laser Scanning Workflow.	32
22	Laser Scanner Positions.	33
23	Some Resulting Scans from Different Angles.	33
24	Equipped Spherical Target Balls.	35
25	Manual Point Cloud Initial Merging.	37
26	Deviations from Manual Merging.	37
27	Point Cloud Spherical Element Alignment Control.	38

28	Scans Post Manual and Feature Fitted Merging.	39
29	Scans Post Manual and Feature Fitted Merging.	39
30	Final Point Cloud	41
31	Proposed Method of Extracting Intersection Points.	42
32	Steel Truss Point Cloud Visualized	43
33	Example section along the x-axis.	44
34	Point Elimination Converting 3D Points to 2D.	44
35	Section Projected on to a 2D plane.	45
36	Zoomed in example of one element where the center point is inaccurate.	46
37	Modified center points can be seen as red crosses.	47
38	Resulting center points from using Sectioning and Mean Shift Algorithm.	47
39	Four clusters from a randomly generated data set of 1000points.	48
40	Section where elements are perpendicular to sectioning axis (x-axis).	50
41	Blue » Manually removed; Red » Automatically removed; Black » Remaining .	51
42	Section along y-axis. Element center points visually highlighted to illustrate the desired goal.	52
43	Moving from section [i] to section [i + 1] (next section).	53
44	End results from the Element Recognition Algorithm.	54
45	Matching point to Element when moving from section [i] to section [i + 1].	55
46	Example Point Prediction for Element A.	56
47	Resulting Intersection Points marked in red.	57
48	Cutout of two “intersecting” Elements.	59
49	Zoomed in view of the intersection.	60
50	Random distribution vs post sorting algorithm.	62
51	Aligned Intersection Points	63
52	Updated BIM Model.	65
53	Accuracy analysis between Point Cloud and Updated As-Built Model.	66
54	Geometrical Deviation Analysis in Geomagic Control X.	67
55	Visual Deviation Comparison of Updated As-Built Model vs. Designed.	67
56	Material Change Comparison of Updated As-Built Model vs. Designed.	68
57	Element With Triangular Mesh.	70

58	Designed Intersection Point Import to Midas Civil.	72
59	Material properties. Steel Quality in red.	73
60	Element Profiles	73
61	Designed Model Midas Civil.	74
62	Designed Intersection Points and Intersection Points from As-Built Model (Point Cloud).	75
63	Updated As-Built Model in Midas Civil	75
64	Fixed Truss Supports.	76
65	Maximum allowed deformations from self-weight according to Chinese Standards [41].	76
66	FEA Results for Designed Steel Truss.	77
67	FEA Results for Updated As-Built Steel Truss.	78
68	2D Illustration Bounding Box Around Mean Point of Elements Dataset.	80
69	Example Code for calculating radius for circular profiles.	81

List of Tables

1	Dimensions from DWG drawing.	18
2	Extracted intersection points from DWG drawing.	20
3	Extracted intersection points from DWG drawing.	38
4	Extracted Intersection Points from Point Cloud.	59
5	First Intersection Point from Design and Point Cloud (Post point aligning).	63
6	Intersection Points from Point Cloud post Sorting and Aligning.	65

1 Introduction

Humans have been building constructions since ancient times. Earlier structures served the single purpose of housing, but as humanity has grown, priorities and construction habits have changed. Today, there exists different type of constructions for various purposes. The main categories of structures are roads and bridges for infrastructure and buildings for housing, work, etc.

Typical ancient buildings were either wooden or stone constructions. The material choice was heavily dependent on the global location. In Norway, most houses were constructed using wooden materials due to the cold climate. In contrast, Iraq, where the climate is much hotter, the preferred material was stone [2]. However, in modern times, the most prominent materials for constructions are concrete, steel and a combination of both. For this paper, steel structures will be emphasized.

Human progress in mathematics, physics, and material science has led to a better understanding of how constructions behave. The worlds first steel structure, which still stands today, is the Home Insurance Building located in Chicago, Illinois. The building was built in 1885 and made use of a simple steel frame[3]. Four years later, in 1889, the Eiffel Tower was constructed in Paris, France, using iron. Iron is a predecessor to modern steel. At the time, the Eiffel Tower was the worlds tallest structure and regarded by many as one of the most complex man-made structures [4]. Today, the world's tallest structure is the Burj Khalifa, which was built using concrete and steel.

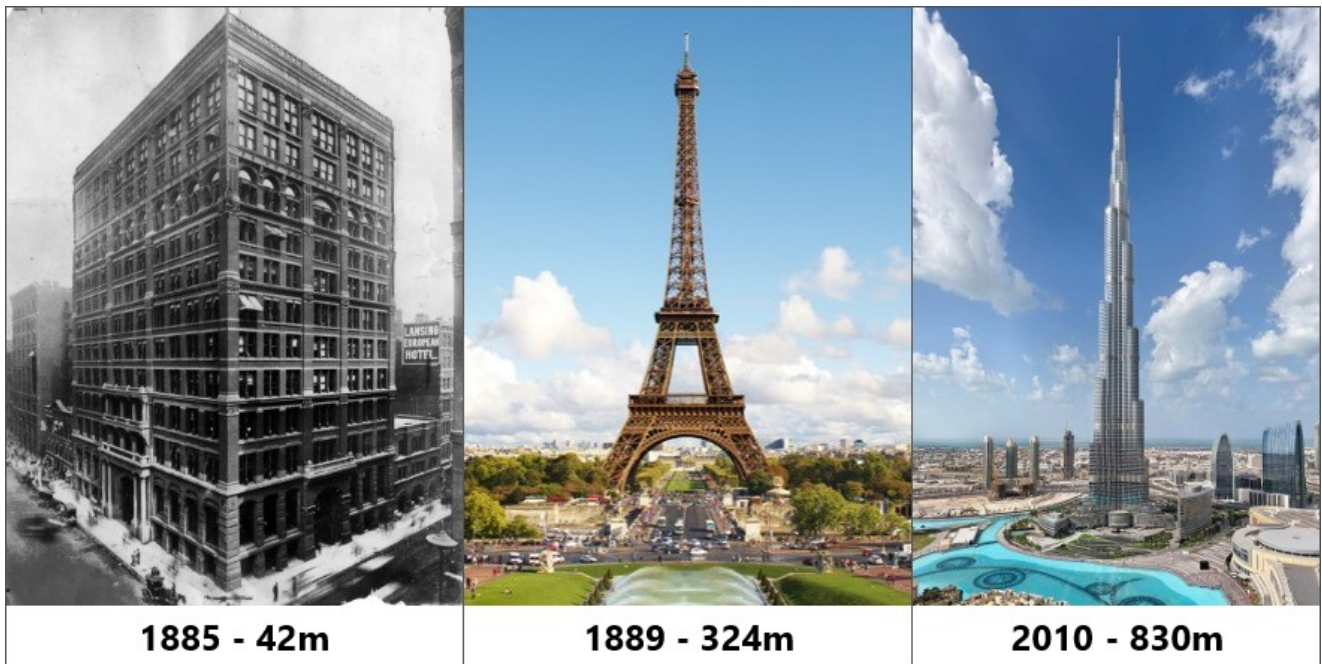


Figure 1: Year - Height: Home Insurance Building [5], Eiffel Tower [6], Burj Khalifa [7].

As seen in Figure 1, constructions are becoming more and more complex with time. This is especially true for steel structures as they usually are built component by component, allowing for more flexible designs. However, the construction process is highly prone to errors as many factors can go wrong. Human errors, manufacturing errors, or even design errors might lead to fatal consequences. Halting construction on-site or dismantling and rebuilding steel structures can be costly and disrupt the logistics at the construction site. As a preventive measure, examinations can be made to ensure that each as-built steel component is in line with the designed structure at the time of assembly. Although this might seem like a viable solution, execution is hard as steel components come in all sorts of shapes and sizes. Manually conducting examinations of all parameters such as individual element length, cross-sectional width & height, deflection, etc., is very inefficient and labor-intensive.

In the last couple of decades, as technology has continued to improve, the AEC-Industry has digitized a big portion of the work previously done manually. The introduction of Building Information Modeling, also known as BIM, has allowed the design process of constructions to go digital. This has led to more collaboration and information exchange between different teams. A Building Information Model, further referred to as a BIM model, is a digital 3D model with integrated information of the structure.

Although BIM is a great tool for the AEC-Industry, it is mainly being used as a tool for design. For steel structures, each component is designed and constructed as individual entities before being assembled into a complete steel structure on the construction site. This implies that each BIM model for each steel component is prone to errors that can lead to problems during assembly. The main enabler of such problems is that the as-built steel component in some way or the other deviate from the designed component. In other terms, the designed BIM model will not be an accurate representation of the as-built component.

To combat this issue, 3D laser scanning technology can be used to create a point cloud representation of the component. The data gathered from the point cloud can in turn be used to update the designed BIM model to be a more accurate representation of the as-built steel component. The concept of using point cloud data to create a reverse BIM models of a complete structure is shown in Figure 2.

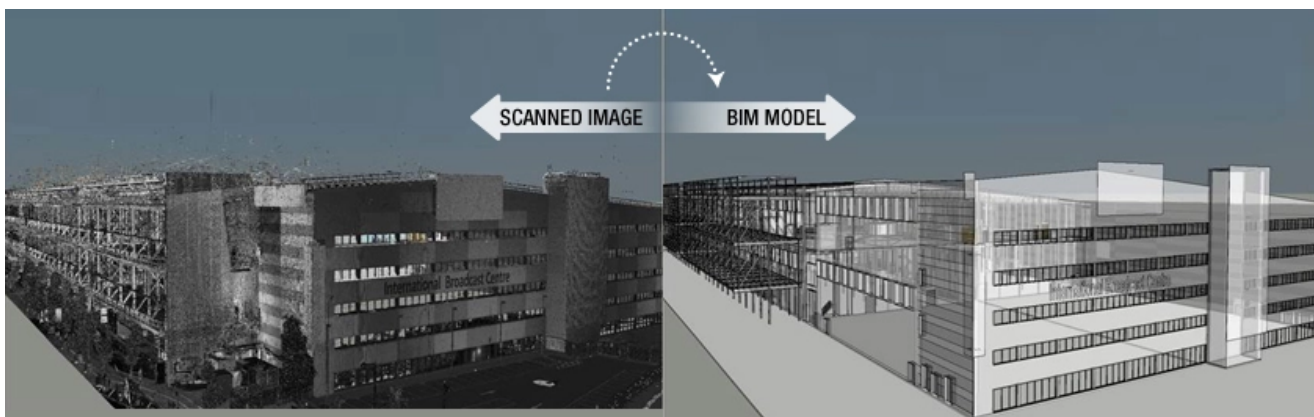


Figure 2: Reverse Engineered BIM model based on Point Cloud Data [8].

The process of reverse modeling and updating existing BIM models is usually a manual task and is considered essential for having a realistic digital representation of the as-built component. The realistic digital representation of the as-built component is used to assess whether the as-built component is fit for assembly. Automating the process of updating the designed model to be in line with the realistic as-built component will have a significant impact on the way steel structures are constructed. Margin of error, cost and safety are all factors which can and must be reduced.

1.1 Research background and significance

In 2018, the AEC-Industry consumed roughly 50% of the global steel supply for buildings and infrastructure. At that time, the global steel industry produced a combined total of 1.7 billion metric tons of steel [9]. This number is expected to grow over time as the need for both iron and steel increases.

The world population is expected to grow by 2 billion people over the coming 30 years. In 2050, the population count is estimated to be around 9.7 billion [10]. Along with the increase in population, the steel consumption within the AEC-Industry is also expected to increase as the need for new infrastructure and buildings emerge. New innovative solutions will therefore be required to ensure an efficient and sustainable construction process.

BIM has revolutionized the way constructions are built. On modern construction sites, every process is preplanned and ideally also simulated digitally. The logistics are often planned to the tiniest detail to save on cost and time. As the cost of material goes down over time, the cost of human labor goes up. For the contractor, it is not ideal to have work halted or prolonged at the construction site as it can bear an immense cost.

When steel components arrive at the construction site, it is important to ensure that the properties of the components are as designed. Inaccuracies can lead to the component having to be scrapped and remade, modified, or even worse, the faulty component can go unnoticed and be used in the construction. Most steel components are built separately and assembled on the construction site. Experience has shown that a trial assembly can be conducted before the steel component is shipped to the construction site. This ensures that the component is fit for use and alleviates any problems on the construction site. However, such trials are expensive and can cost up to 25% of the total cost [1]. A cheaper and more viable option is to digitize the assembly and examination process. To do so, an accurate BIM model representation is needed of the as-built component.

The work done in this paper will heavily emphasize creating new innovative technology to improve and automate the reverse modeling process by finding a sufficient way to update the already existing BIM model to represent the as-built component accurately.

1.2 Research Status

This section will introduce and shed light on the current global development and practices when it comes to the use of BIM, 3D Laser Scanning, Digital Twins, Point Cloud Reverse Modeling and Machine Learning.

1.2.1 BIM

Building Information Modeling (BIM) is a technology that enables Architects, Engineers, and even regular people to digitally construct and assemble a construction before it is done physically. The foundation for BIM was laid in 1975 when an American professor named Charles M. Eastman published an article in the AIA Journal under the title "The Use of Computers Instead of Drawings In Building Design" [11]. In the article, Eastman proposes sketching building designs on the computer instead of on paper. Since then, BIM has evolved from a mere concept, and thus, new companies have emerged.

Autodesk, founded in 1982, has laid the foundation for some of the key components in modern-day BIM. In the beginning, companies created their BIM software tools supported by their file formats. Problems arose when models from different companies could not exchange data with each other due to different software tools being incompatible with different file formats. The solution became standardizing the file format to allow different software tools to exchange data. The solution is called Industry Foundation Classes and are referred to as IFC. The initiative to create such a standardized file format began in 1994 when 12 US companies, including Autodesk, got together and came up with a set of rules which would support integrated application development [12].

Current BIM allows for different stakeholders in a project to participate and work on the same model. The idea is that every team of Architects, Civil Engineers, Electrical Engineers, etc., should work on a model in the same digital environment so that eventual problems that might occur on the construction site can be addressed and fixed before the construction process begins. Everybody on the project can see different changes made by different teams and assess whether those changes are compatible with what they are working on. This ensures that all the teams on the project collaborate preemptively and actively while the construction is being built. It is

easier to digitally address an issue before the construction process begins than it is to address the same issue while the construction is being built.

There is also an increased governmental push for construction companies to digitize their workflow and introduce BIM. In Scandinavian countries such as Norway and Denmark, the government has mandated the use of BIM in every state-sponsored project [13]. There is currently a surge in the use of BIM within the AEC-Industry. However, BIM is mostly used for design purposes only. This is partly due to the technology first being intended to be used for that purpose and the ease of implementation. In the coming years, it is expected that BIM will be more popular and that the technology will keep expanding from design to other fields within the industry.

1.2.2 Digital Twins

Digital twin is a term used to describe a virtual model of an existing or soon to-be existing product. In the AEC industry, the product is usually a full-scale construction. The idea is that the digital twin should serve as a real time virtual counterpart to the physical construction and reflect the condition in which the construction is at. Having a digital twin model opens the possibility to run different simulations to predict what the future outcomes of a change would be without having to try it out on the as-built construction.

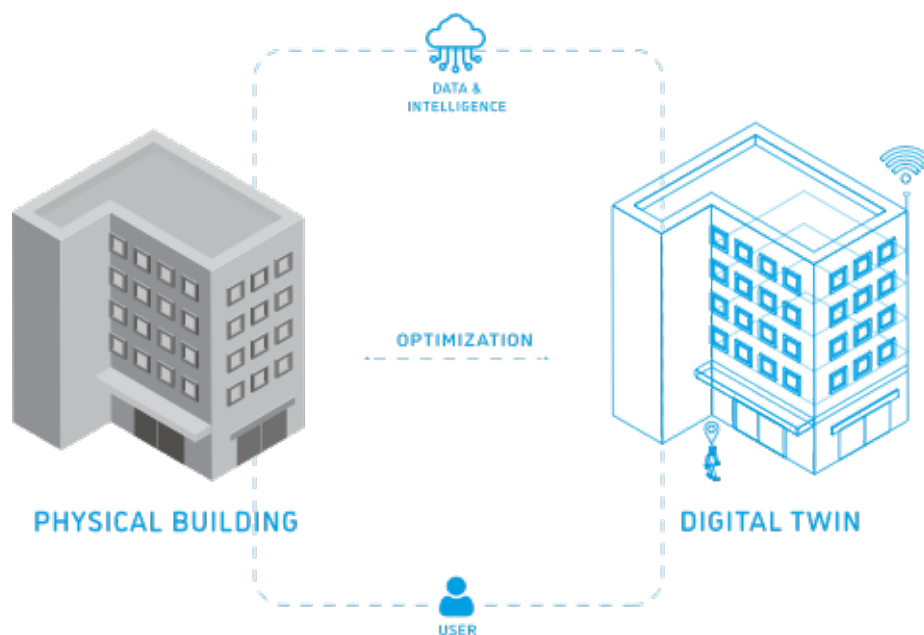


Figure 3: Physical building & Digital Twin [14].

There are many benefits of having a digital twin for larger scale constructions, and even smaller ones. A digital twin model will allow for an increase in operational efficiency and safety, improve asset management, predict maintenance, and save cost for operational expenses [14]. Incorporating sensor data from inside and outside the construction and into the digital twin can enable workers and users to get a feel of what the current state of the construction is. This sensor data could be the room temperature, ventilation airflow, thermal heat loss etc. The possibilities are endless. Data gathered from the sensors can be used in later stages to give live visual predictions to where eventual maintenance might be needed.

Currently, digital twin models are being used for all above mentioned purposes. The use of the technology is however limited to constructions where digital twins exist. There is an ongoing effort by companies and building owners where a digital twin model does not exist to create one. Automating the process of creating or updating designed BIM models to reflect the as-built structure is seen as one of the next directions in which the technology can be developed [15].

1.2.3 3D Laser Scanning

Surveying and mapping technology is constantly improving and has in recent years been put in the spotlight. Improvement in Artificial Intelligence has opened the door for new fields in which the technology can operate. Self-driving cars might be one of the most recognizable and funded fields where the technology is being heavily applied and developed. Laser scanning is regarded as the best current technology to create a 3D map of the surroundings. Improvements in this technology has allowed it to become more and more implemented in the AEC-industry.

3D Laser scanning works by rapidly sending out laser beams, point by point measuring the distance to surrounding solids where the light is reflected and mapping the points within a coordinate system [16]. The collection of these points will then result in a point cloud which mimic the surroundings. Common practice for *Terrestrial Laser Scanning*, described further in section 2, is to have scans taken from different stationary locations in proximity and have their point cloud coordinate systems merged into one. This results in one single point cloud which is vastly more precise.

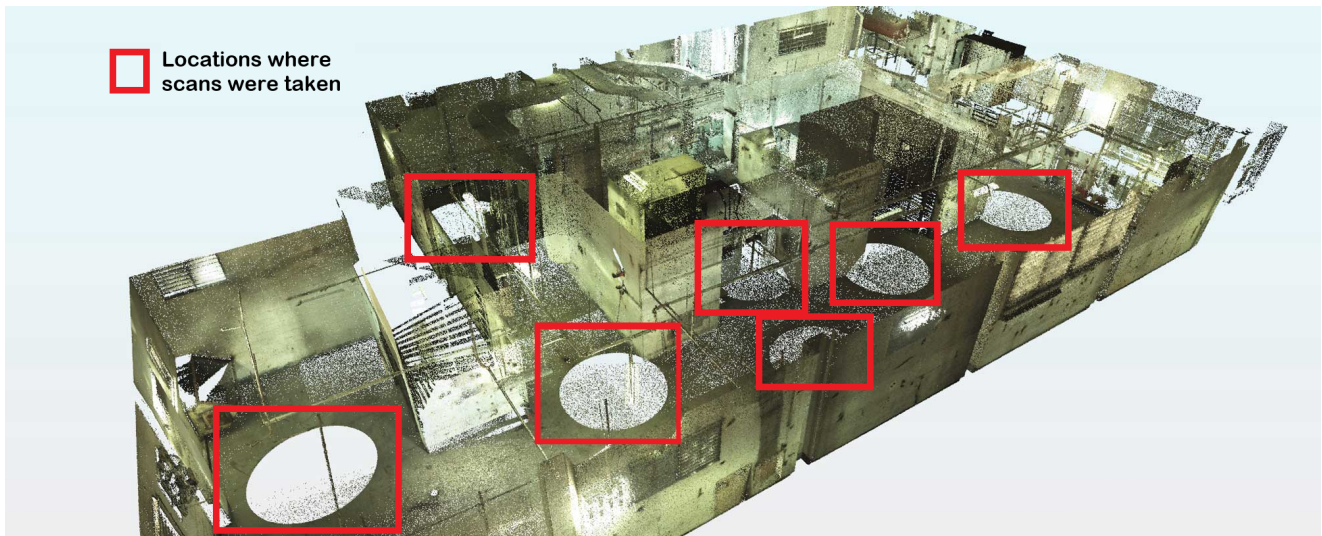


Figure 4: Point cloud representation of a laser scanned floor [17].

The red boxes in Figure 4 represent the locations where each scan was taken.

3D laser scanning is applicable in a variety of fields within the AEC-Industry. The technology is commonly used as a tool to measure precision on site, volume of certain areas, deformation analysis, and to reverse model structures. Although the technology can be beneficial for engineers, there are some limiting factors. Access to the technology can be hard for smaller companies as good quality hardware is expensive. The technology is also still emerging and the possibilities for point cloud processing within CAD software tools are not great. Additional software must be downloaded which can be expensive.

Nevertheless, progression within the field of 3D laser scanning is looking promising and accessibility keeps improving. New companies are currently emerging whose purpose is to provide consultation and assistance within this field. In the coming years, it is expected that this industry is going to grow as the need for digital twins of existing structures rises.

1.2.4 Point Cloud Reverse BIM Modeling

There is currently a growing interest within the AEC-Industry to reverse model existing structures. Reverse modeling is great for old structures where there are no current BIM models. For newer projects where BIM models exist, reverse modeling serves as a tool to improve the accuracy of the existing model so that the digital twin is more aligned with as-built structure on

site.

Around 80% of all buildings in Europe were constructed before 1900. In essence, this implicates that these buildings, along with a lot of newer buildings, do not have accessible BIM models. The surge in demand for digital twins has made BIM models for existing buildings more sought after than ever before. Reverse modeling makes it possible to incorporate the existing building to a BIM methodology and take advantage of benefits such as calculations of design alternatives, cost estimates, material quantification, data management, as-built documentation, constructive state analysis, execution plans, and others [18]. 3D reverse modeling tools are crucial in this process as manual work may provide lower rates of accuracy as well as being more time consuming.

A BIM model created during the design phase will always vary from the corresponding as-built structure on site. Deviations in the structures occur due to different factors such as material deformation, human error, undocumented design changes etc. This will result in the as-built structure being different than the designed BIM model. To have an accurate digital twin, both the BIM model and the as-built structure must be as similar to each other as possible.

Currently, a lot of construction sites are moving over to pre-fabricated components which are assembled on site. This reduces construction time, but the process also elevates the risk of inaccuracies. Inaccuracies can lead to either the component being deemed unfit for use, scrapped or recycled, or lead to changes in the original construction plan. None of these scenarios are ideal. Another problem occurs when the components' inaccuracies get noticed later in the construction phase. This can lead to the structure having to be un-assembled and then re-assembled with more accurate components [19].

Usually, the inaccuracies are manually measured or spotted by workers. Point cloud data could therefore be ideal as it can provide data from millions of datapoints on the structure. The accuracy could be far greater than any manual measurements. The point cloud data gathered from the as-built structure can then be used to create a reverse BIM model. Evaluations can be made on whether a component is fit for the structure or if any additional changes needs to be made.

1.2.5 Machine Learning

The use of the term Artificial Intelligence, also referred to as AI, is becoming more and more prominent in modern society as technology improves. The end-goal with AI is to mimic and exceed the human brains behaviour by using non pre-programmed algorithms. Machine learning, further referred to as ML in this paper, is a branch of AI where the focus is on giving computers the ability to learn without being explicitly programmed. ML explores the possibilities of constructing algorithms which can make predictions on data, based on data [20].

There are no predefined solutions for how ML should work. Due to the lack of knowledge in this field, everything is set to be trial and error where the best solution to a given problem will prevail. However, within the field of ML, *supervised and unsupervised-learning* are the main methods for constructing ML algorithms.

1.2.5.1 Supervised Learning

Algorithms built on the supervised learning method focus on training the ML model using pre-labelled examples. The method is set to mimic some of the learning patterns of children. If a child is wrong and labels a pen as a knife, the parent will step in, supervise, and have the child change the labeling until it is correct. Similarly, a supervised learning algorithm will make predictions based on previous data inputs and outputs, but when wrong predictions are made, the algorithm goes back and adapt for it to make the correct prediction next time around. Supervised learning is commonly used in applications where historical data predicts likely future events [20].

1.2.5.2 Unsupervised Learning

Algorithms built on the unsupervised learning method focus on targeting data that has no historical labels. There are no right or wrong answers in unsupervised learning. The algorithm has to assess what a set of data represents and how it should use it. The method works well on transactional data where different entities' input and output data are similar [20].

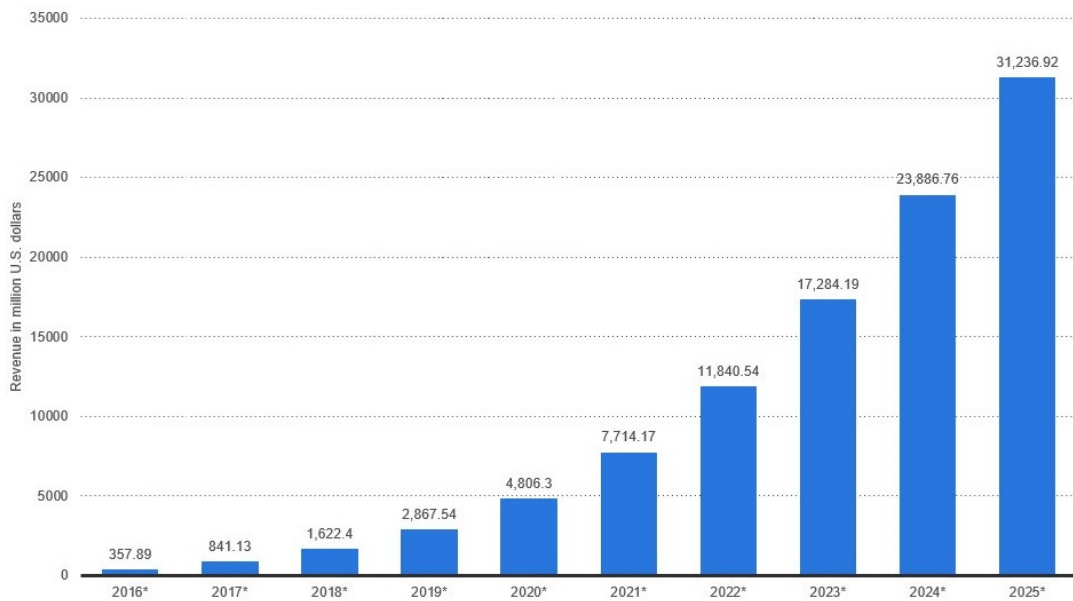


Figure 5: Revenues from the artificial intelligence for enterprise applications market worldwide, from 2016 to 2025 (in million U.S. dollars) [21].

As seen in Figure 5, revenues generated from AI for enterprise applications are expected to be growing into the near future. Current development in the field of ML has been rapid due to several factors such as improved know-how in how the human brain works. However, most important is the improvements made in computing technology. ML requires huge data-sets to be analyzed in order to assert predictions. Improvements in processing units such as the *Central Processing Unit (CPU)* and the *Graphics Processing Unit (GPU)* has allowed more complex computations to take place. According to Moore's Law, which historically has been holding true, transistors and microchips will improve and double over the course of every other year while the price will get cut in half [22]. This implies that the processing technology, over time, will become better and more available. Therefore, the field of AI and ML is expected to keep improving as more and more people gets access and make contributions to the technology.

1.3 Research Content and Technical Route

This section will present the research content as well as the technical route used in this paper. First the research content will be presented with some information about the chosen component as well as the location. Following, a technical route for the conducted research will be presented.

1.3.1 Research Content

As described in the problem statement, the aim of the work conducted in this paper is to reverse model a physical as-built component in order to assess the similarities with its designed model and confirm whether or not the component is eligible for the construction. The chosen component is a steel truss which is at the time of writing, located in a warehouse in Chongqing, China.



Figure 6: Location of Chongqing in China

The steel truss is part of the roofing structure in a new unnamed skyscraper in China. The steel truss has a span of approximately 7.2m and consists of hollow steel tubes as well as spherical elements for consolidation. The laser scanner as well as the steel truss can be seen in Figure 7.

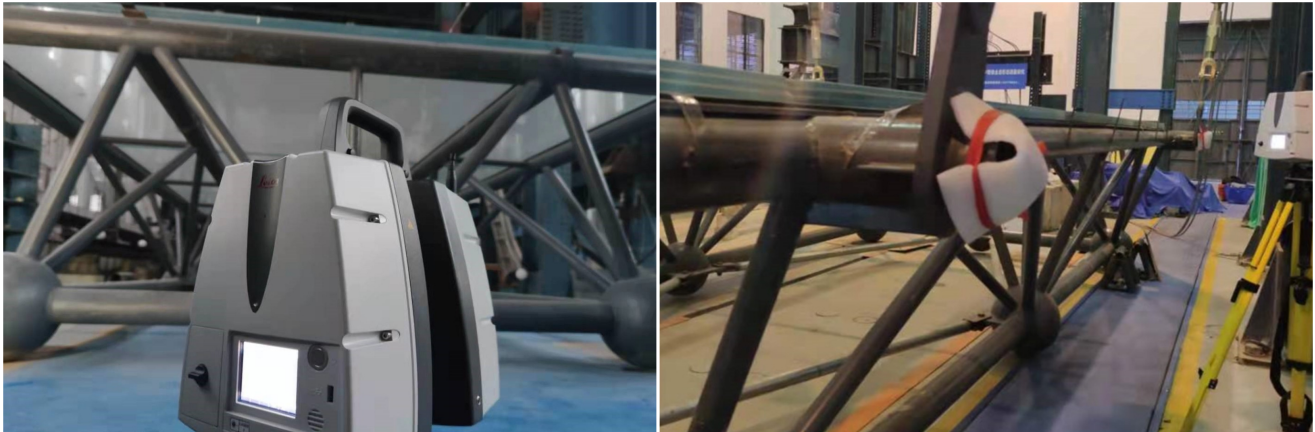


Figure 7: Laser Scanner and Scanned Steel Truss.

1.3.2 Technical Route

A steel component undergoes a design and manufacturing phase pre-assembly. The component should therefore undergo an examination post-manufacturing to ensure quality before reaching the construction site for assembly. The content in this paper is structured to mimic a realistic environment where the technology is applied. On that basis, the content composition seen in Figure 8 is chosen.

The main tasks performed and the hardware/software tool used within each section are presented in Figure 8, except for sections one, three, eight, and nine. These sections were excluded from the illustration as they serve as literary chapters for introduction, discussion, and result to the conducted research.

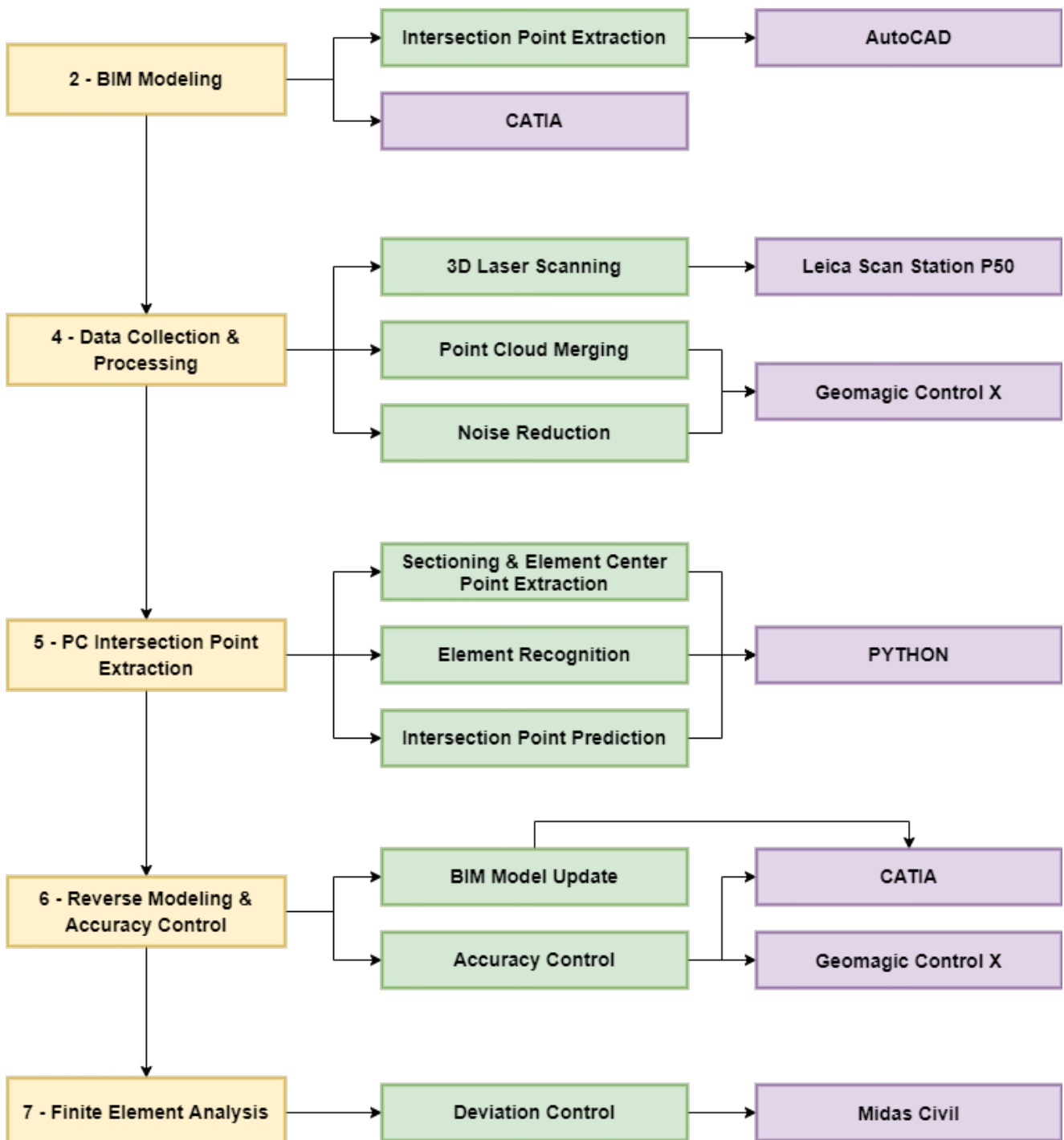


Figure 8: Technical Route of this Paper.

1.4 Applied tools

A combination of hardware and software was used in order to conduct the research described in this paper. This section will briefly introduce the tools used as well as describe what purpose each served.

1.4.1 Hardware

Although a majority of the tools used are categorized as software, a 3D Laser Scanner was used to acquire the point cloud data of the steel truss. Different computers were also used to run programs etc. but they will not be taken into account.

Leica ScanStation P50

Leica ScanStation P50 is a long range 3D Terrestrial laser scanner. The laser scanner is a professional hardware tool which allow users to create an accurate 3D point cloud mapping of the desired surroundings. The tool can be used to map out large infrastructures such as bridges, factories, tall buildings etc. The scanning device has a range of up to 1km and provide high quality 3D point data overlapped with HDR imaging as needed [23]. For this paper, the Leica ScanStation P50 was only used in order to collect the point cloud data of the steel truss.

1.4.2 Software

The research conducted in this paper mostly consist of work done to improve or supplement software. A combined total of 5 software tools were used in this thesis. This section will briefly introduce these software tools.

1.4.2.1 Geomagic Control X

Geomagic Control X is a software tool created and published by 3D Systems. The software is used to inspect 3D models as well as point clouds. The built-in features of the software allows for easy and seamless treatment of point cloud data. In addition, the software can also be used to run complex analysis using multiple data-points [24]. For this paper, the software is used for multiple accuracy control purposes. First, the software is used to conduct a accuracy analysis

between the 3D model and the scanned point cloud. Then the software is used to conduct a complete deviation analysis between the designed and the updated as-built model.

1.4.2.2 AutoCAD

AutoCAD is a commercial computer-aided drafting and design software tool developed and published by Autodesk. The software can be used to draft both 2D and 3D drawings while allowing for tasks such as drawing comparison, dimension measurements, exporting key data and much more [25]. In this paper, AutoCAD is used to process and extract data from the designed component. The designed steel truss is received by the architect as a 3D DWG file. Intersection points are marked and exported using built-in features in AutoCAD.

1.4.2.3 CATIA

CATIA is a multi-purpose software tool used for *Computer Aided Design (CAD)*, *Computer Aided Engineering (CAE)* as well as *Computer Aided Manufacturing (CAM)*. The name CATIA is an acronym for *Computer Aided Three-Dimensional Interactive Application*. CATIA is a world leading software solution for product design and is powered by Dassault Systèmes [26]. The application is well adopted among engineers in all fields ranging from aviation to civil. For this paper, CATIA was used for CAD and some CAE purposes. The software is used to parametrically design the BIM model and then to conduct a visual deviation comparison between the designed and updated as-built model.

1.4.2.4 Python

Python is a well-known general purpose programming language, initially developed by Guido Rossum. The programming language is developed and maintained under an OSI-approved (Open Source Initiative) open source license [27]. The programming language is free and is widely adopted amongst engineers and programmers. The ease of use as well as the large global community has attracted many engineers to adopt the programming language for day-to-day use in their field of work. As a result, a collection of packages containing codes and algorithms are available for free. In this paper, Python is used to write custom scripts for sectioning, element detection, and extracting the intersection points used for updating the designed BIM model.

1.4.2.5 Midas Civil

Midas Civil is a Finite Element Analysis (FEA) software tool developed by MIDAS IT. The software is mainly used for bridge analysis and design, although it is fit for other FEA purposes. The software is intuitive and allow for parametric design based on input nodes. The software allow for choosing between a variety of materials and their properties [28]. In this thesis, the software is used to conduct an FEA to assess whether the as-built steel truss fulfills the requirements in China for dead-load deflection during assembly.

2 Designed BIM model

The BIM model of the steel truss is created in accordance with the structural drawings received by the architect. The following sections will describe the design process from extracting the critical intersection points from the DWG drawing in AutoCAD, throughout the BIM model creation in CATIA.

2.1 Designed Intersection Point Extraction

The drawing of steel truss with all its corresponding dimensions is received as a DWG file from the architect. AutoCAD is then used to process the DWG drawing and extract the critical intersection points needed to create the BIM model.

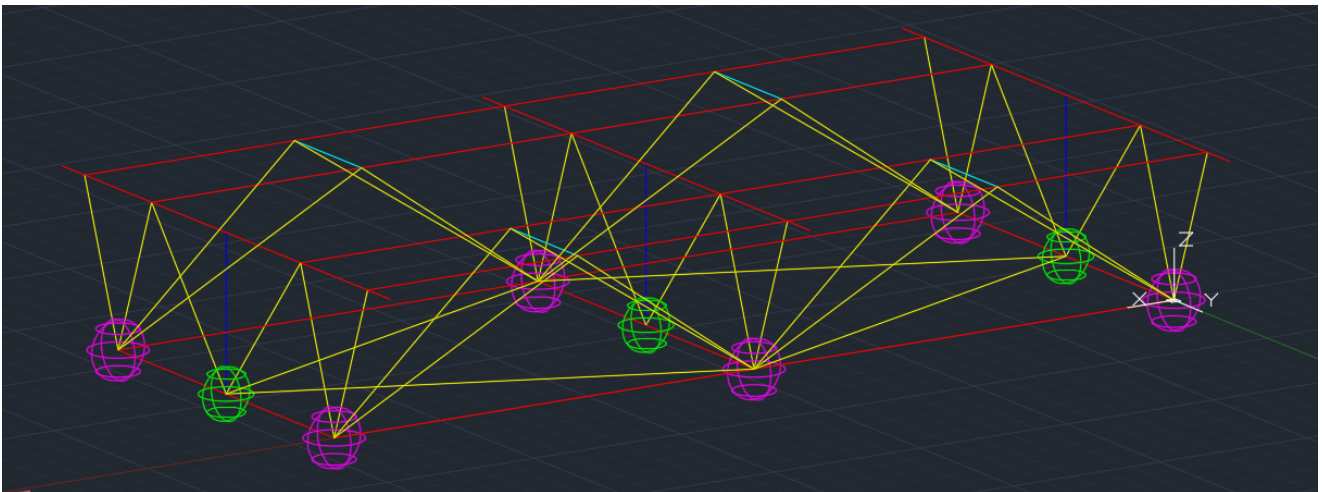
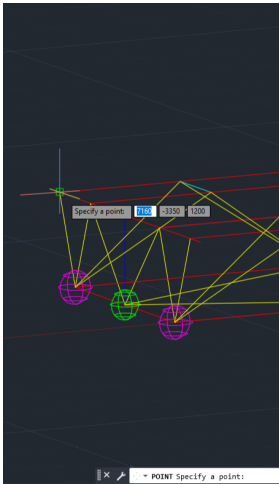


Figure 9: DWG drawing of the steel truss visualized in AutoCAD.

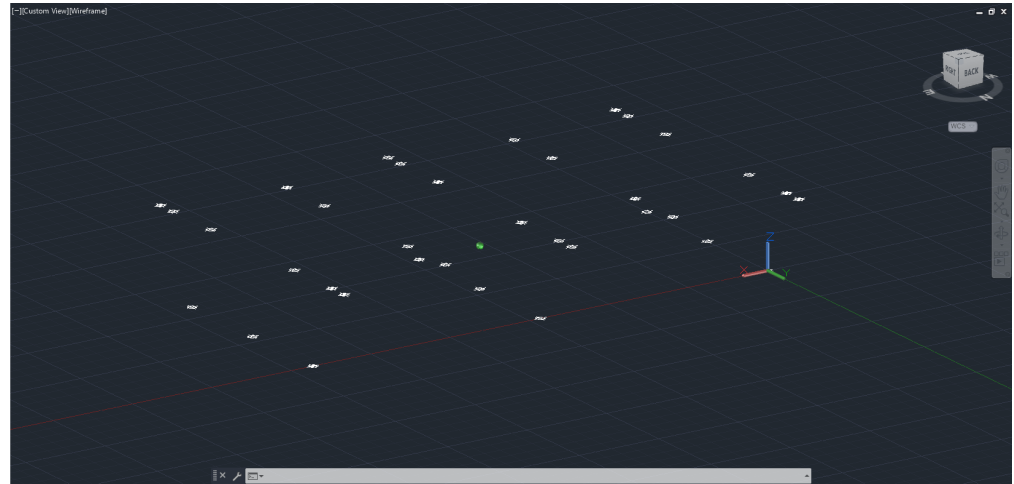
ELEMENT	DIMENSIONS (DIAMETER)
Yellow Tube	75mm
Red Tube	150mm
Blue Tube	50mm
Dark Blue Tube	60mm
Green Sphere	400mm
Purple Sphere	450mm

Table 1: Dimensions from DWG drawing.

From the DWG drawing, AutoCAD can be used to create intersection points where the tubes/spheres connect by applying the POINT command as shown in Figure 10. These intersection points can then be exported to excel using AutoCAD data extraction.



1. Select Points



2. All intersection Points

Figure 10: Intersection point creation process.

Position X	Position Y	Position Z
0.0000	-3650.0000	1200.0000
0.0000	-3350.0000	1200.0000
0.0000	-2900.0000	0.0000
0.0000	-2450.0000	1200.0000
0.0000	-1450.0000	0.0000
0.0000	-1450.0000	1200.0000
0.0000	750.0000	1200.0000
0.0000	450.0000	1200.0000
0.0000	-450.0000	1200.0000
0.0000	0.0000	0.0000
1790.0000	-450.0000	1200.0000
1790.0000	450.0000	1200.0000
1790.0000	-3350.0000	1200.0000
1790.0000	-2450.0000	1200.0000
3580.0000	-3350.0000	1200.0000

3580.0000	-2450.0000	1200.0000
3580.0000	-450.0000	1200.0000
3580.0000	-1450.0000	1200.0000
3580.0000	-3650.0000	1200.0000
3580.0000	-2900.0000	0.0000
3580.0000	-1450.0000	0.0000
3580.0000	450.0000	1200.0000
3580.0000	750.0000	1200.0000
3580.0000	0.0000	0.0000
5370.0000	-2450.0000	1200.0000
5370.0000	-3350.0000	1200.0000
5370.0000	450.0000	1200.0000
5370.0000	-450.0000	1200.0000
7160.0000	-2900.0000	0.0000
7160.0000	-3650.0000	1200.0000
7160.0000	-3350.0000	1200.0000
7160.0000	0.0000	0.0000
7160.0000	-1450.0000	1200.0000
7160.0000	750.0000	1200.0000
7160.0000	450.0000	1200.0000
7160.0000	-450.0000	1200.0000
7160.0000	-2450.0000	1200.0000
7160.0000	-1450.0000	0.0000

Table 2: Extracted intersection points from DWG drawing.

2.2 BIM Design

Intersection points extracted from the DWG file can be seen in Table 2. These points from the excel file will then be imported into CATIA using CATIA's own excel import feature-file. Figure 11 show the points once they are imported into CATIA.

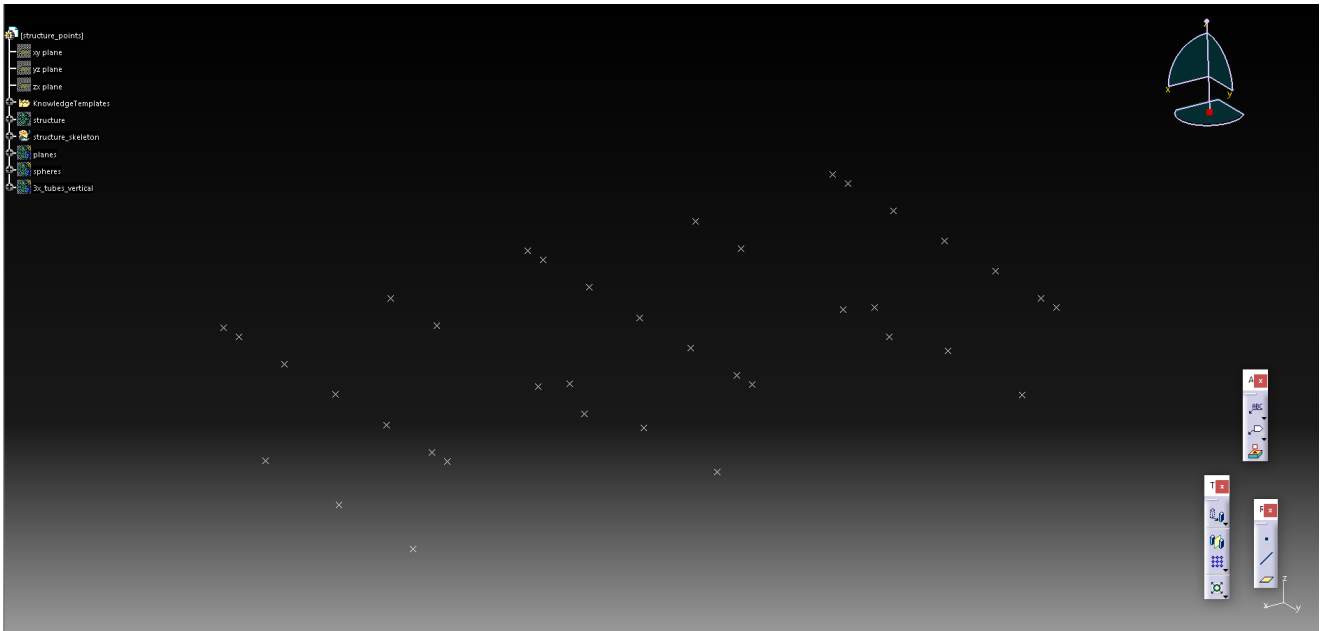
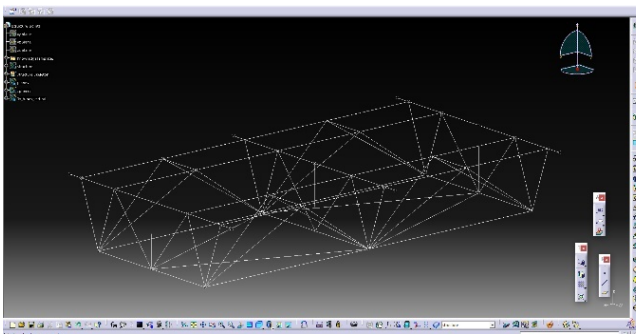
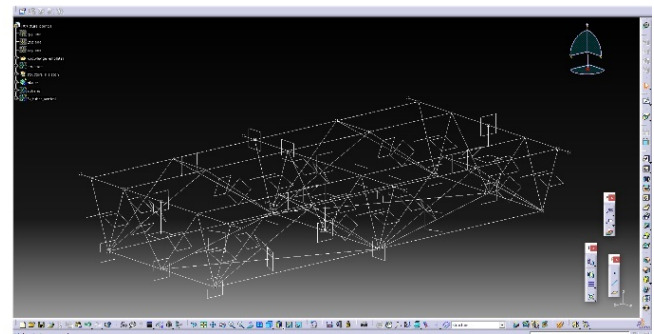


Figure 11: Imported Intersection Points in CATIA.

Lines, which represent the different elements, are then drawn between the intersection points. Corresponding planes are created in the center of each element to enable sketching of the element profiles. See Figure 12.



1. Lines Between Points



2. Added Planes (For sketching purposes)

Figure 12: Workflow progression in CATIA. Adding lines and Planes.

The planes are perpendicular to the direction of the lines on which they are placed upon. They act like a 2D plane where sketching can take place. See Figure 13.

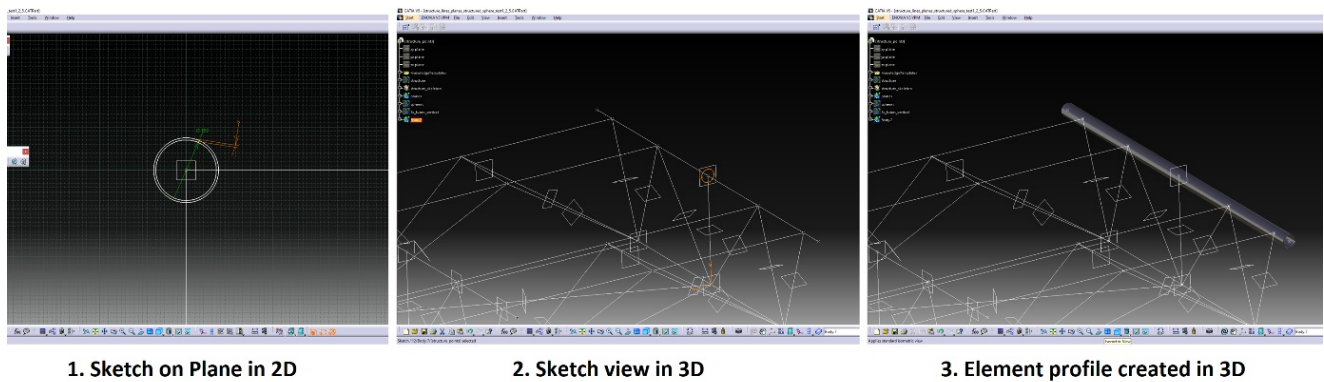


Figure 13: Workflow progress. From 2D sketching to creating 3D Element.

CATIA allow the user to design the element profile one time and then, create a user-profile. The user-profile is stored as template for the designed element profile. Using the line, plane, start and end-point, the user profile can be instantiated. A user profile is therefore created for each unique element and then instantiated where the respective element is located. Once the elements are added to the model, some minor refinements are done to Finalize the BIM model. These refinements consisted of removing areas where elements were intersecting. The finalized results can be seen in Figure 14.

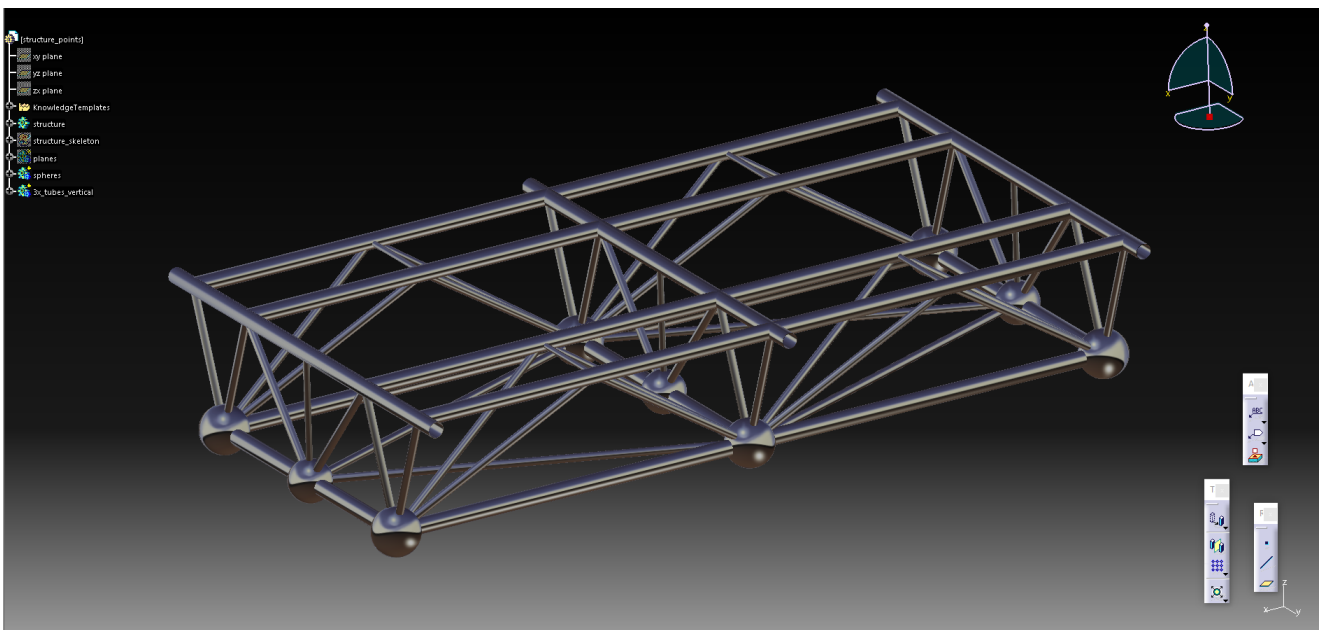


Figure 14: Resulting BIM model design.

The finalized result from this design is in accordance with the DWG drawings, and is the base model from which the physical steel truss will be built. A design table is also created parallel to the BIM model to enable parametric altering of the intersection points. The design table is used to update the model once the physical steel truss is constructed and sufficient data is extracted. Updating the model ensures that the BIM model is as close to the physical as-built steel truss as possible. The process for updating the BIM model is discussed further in section 5 and 6.

3 Steel Truss Point Cloud Data Collection Technology

To create an accurate digital twin of the as-built steel truss, reliable measurement data needs to be collected. Collecting the measurements manually is a slow, inaccurate, and time-consuming process. The difficulty of doing such a job manually grows in proportion with the size of the steel truss. Doing point by point, manual measurements will require a lot of assumptions when predicting where the center-point of a particular element or intersection might be as it can be hard to get into an already built element. In addition, transferring the manually measured data into a digital tool can be troublesome and very ineffective. Doing such work manually can take days and is therefore not desirable as any inaccuracies or problems in the steel truss needs to be fixed immediately. These fixes can take days and therefore, the construction process will be halted by having the data collection and analysis process take longer than necessary. This is where 3D laser scanning technology can be implemented.

Light Detection and Ranging, also known as LiDAR, is a technology initially developed in the 1960s. The first prototype was built by Hughes Aircraft Company who also built the worlds' first laser one year earlier. The technology was first used for terrain mapping of aeronautics and aerospace. One of the first beneficiaries of the technology was the United States' space program where the technology was used to map the moon during the 1971 Apollo 15 mission. [29].

Since then, the technology has slowly been evolving and hence, implemented into the AEC-Industry for different tasks such as quality control, terrain mapping, documenting historical sites, site modeling, quality survey, and freeway design [30]. The technology is becoming more and more adopted during recent years as the technology improves, becomes cheaper and more available. Implementing this kind of technology for measurement data collection can drastically improve efficiency on site and reduce the time it takes to perform analysis and controls of as-built components.

The following sections will describe the working principle of point cloud acquisition, accuracy as well as common methods in which the technology is currently being used. The scanning technology in focus will be Terrestrial Laser Scanning, further referred to as TLS.

3.1 Working Principle of Point Cloud Acquisition

Once a 3D laser scan is performed, the result will be a set of data-points corresponding to the distance to every solid within a range. This set of data-points are referred to as a point cloud. A point cloud will give a 3D mapping of the surroundings from the perspective of the scanning devices' location at the point of measurement. See Figure 15.

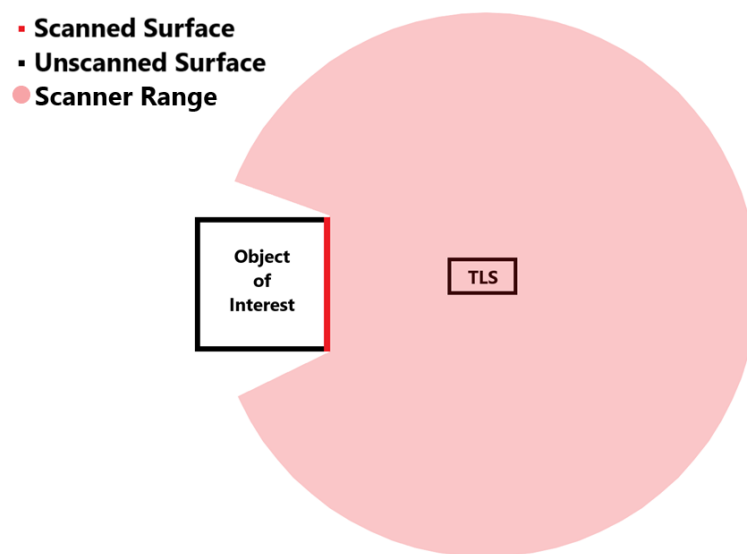


Figure 15: Terrestrial Laser Scanning process with scanned surface in red and un-scanned surface in black.

To attain a detailed 3D model of a specific object, several measurements need to be made. In other words, point clouds need to be generated from different target angles and combined to achieve the best results with the highest accuracy. This is due to all surfaces of the desired object not being visible from one angle alone. The non-visible regions are illustrated as black lines in Figure 15 and 16. The location of each scan should carefully be planned to have the resulting point cloud cover all surfaces of the desired object. As more scans are taken, overlapping points will be expected. These overlapping points will serve as target points which will be used to merge point clouds together. See Figure 16.

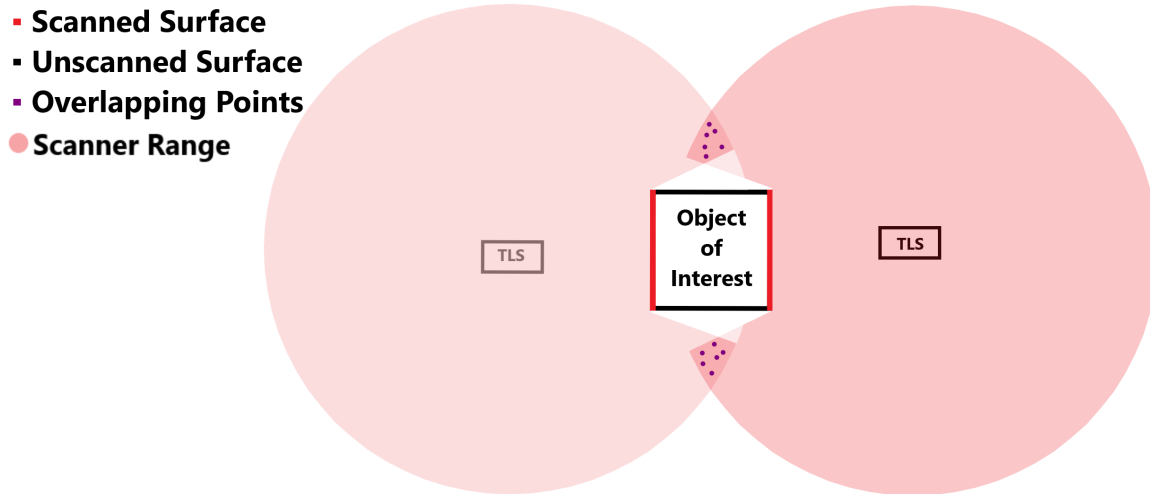


Figure 16: 2 scans with overlapping points.

Overlapping points from both scans are illustrated as purple dots in Figure 16. Point Cloud processing software such as Geomagic Control X or Autodesk ReCap Pro are both capable of merging point clouds together by using the overlapping points. The merging process is further described in section 4. By acquiring more scans from different angles will eventually result in all surfaces of the object being scanned. However, performing the scans can be a time-consuming process. It is therefore crucial to plan out the scans to optimize the scanning efficiency.

3.2 Accuracy of 3D Laser Scanner

The accuracy of the point cloud which is used to reverse model the steel truss is dependent on the precision of the laser scanning hardware. The accuracy of a laser scan expresses the degree in which the measured distance correlates to the true distance. On the other hand, precision of a laser scanner reflects the variation in repeated measurements. In other words, the reliability of the hardware.

3.2.1 3D Laser Scanner Reliability

Just like any other hardware, TLS is prone to a variety of factors which can affect the accuracy and the precision of the scan. These factors can be instrumental errors, errors caused by the objects material or even errors caused by the internal algorithms in the hardware.

In order to attain the most accurate results possible, the laser scanner needs to be calibrated and given the settings most suitable for its use case. Most laser scanners have settings which allow users to change the vertical and horizontal scan sizes, point density in addition to the number of points usually referred to as MPts (Million points). Changes in all of the mentioned settings will affect the scan duration.

Owen Brown and Chris Hugenholtz both published an article in 2013 called “*Quantifying the effects of terrestrial laser scanner settings and survey configuration on land surface roughness measurement*” [31]. The article highlights how user-defined changes in the TLS settings affects the accuracy and precision of a laser scan. Some key takeaways from the article to keep in mind when calibrating the TLS are:

- 1. Point density of the scan should be smaller than the smallest element.
- 2. TLS should be performed from at least three different locations around the object to reduce occlusions.

3D laser scans are used for different purposes and therefore, the quality of the point cloud model should reflect the purpose it should serve. For use cases where the point cloud is used to reverse model an element, settings should be calibrated to optimize the geometrical accuracy of the scan. According to Wolfgang Boehler and Andreas Marbs article, “*Investigating Laser Scanner Accuracy*”, TLS can provide sufficient geometrical accuracy if calibrated and performed right [32].

3.2.2 Data Volume

The volume of data in a point cloud can make up for inaccuracies in the hardware and vice versa. Inaccuracies within the point cloud usually stems from the laser scanning hardware itself. Laser scanners use the time-of-flight from the sensor to the target. See equation 1.

$$d = \frac{t \times c}{2} \quad (1)$$

Where d is the distance from the sensor to the surface point, t is the required time it takes for the emitted laser beam to return to the target and c is the speed of light [31]. A minor time delay will result in an inaccurate distance which in turn will result in a point being positioned further away

than it should be. Such inaccuracies can be made up for by tweaking the point cloud processing algorithms.

Due to the laser scanners limitations with range, accuracy can be affected by the distance in which the scanner is located away from the object of interest. Larger distances can result in more inaccurate points [32]. The number of points in a point cloud is inversely proportionate to the scanning distance. Light from a laser beam will diverge as the travelled distance of the beam gets greater. With this effect, an object with 2000 individual points when a scan is performed with a 5m distance can have the individual points decreased to 50 points by simply increasing the scan distance to 30m [33].

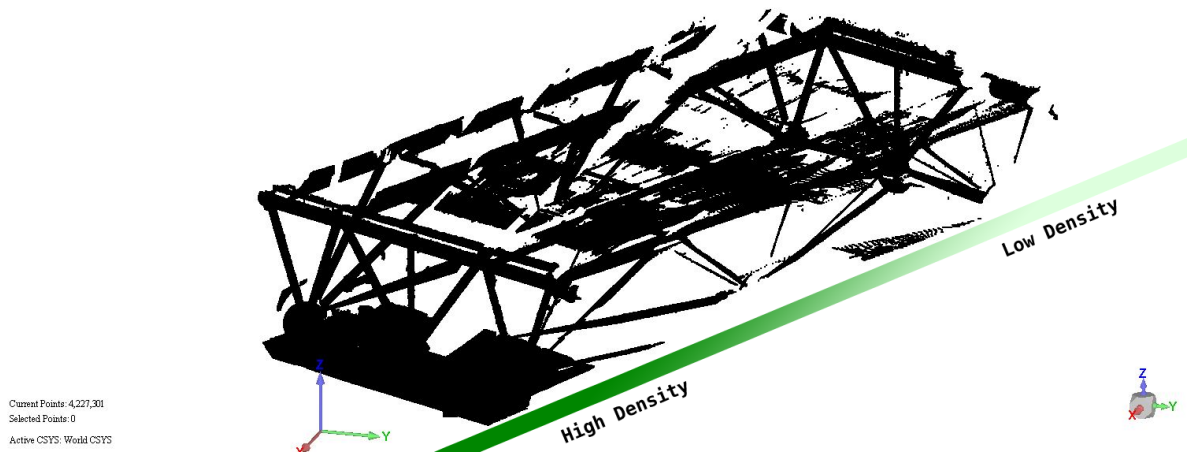


Figure 17: Scan decreasing density with distance.

Other factors such as the objects material reflectiveness also play a role in the accuracy. A non-reflective material will not reflect the emitted laser beam back, resulting in a void within the point cloud. More reflective materials result in more accurate point clouds.

3.3 Common Point Cloud Collection Methods

There are a variety of ways 3D laser scanning can be implemented in order to collect point cloud data. This section will briefly introduce some of the more common methods used for 3D laser scanning and highlight what purpose they serve.

3.3.1 Airborne Laser Scanning

Laser scanners can be mounted on airplanes, helicopters, or drones to scan and generate a 3D point cloud map of the terrain beneath. This method is becoming more and more common as the technology gets cheaper and more portable. Companies such as VEIDEKKE has been using drones with laser scanning capabilities to map out terrains for their projects. The point clouds are primarily used to generate a 3D model of the construction site, but also for other use cases as they are discovered. For instance, the technology can be beneficial for companies as 3D models of the terrain can be used to accurately calculate volume of soil which has been dug out. This data can then be used to optimize transportation cost for the company.

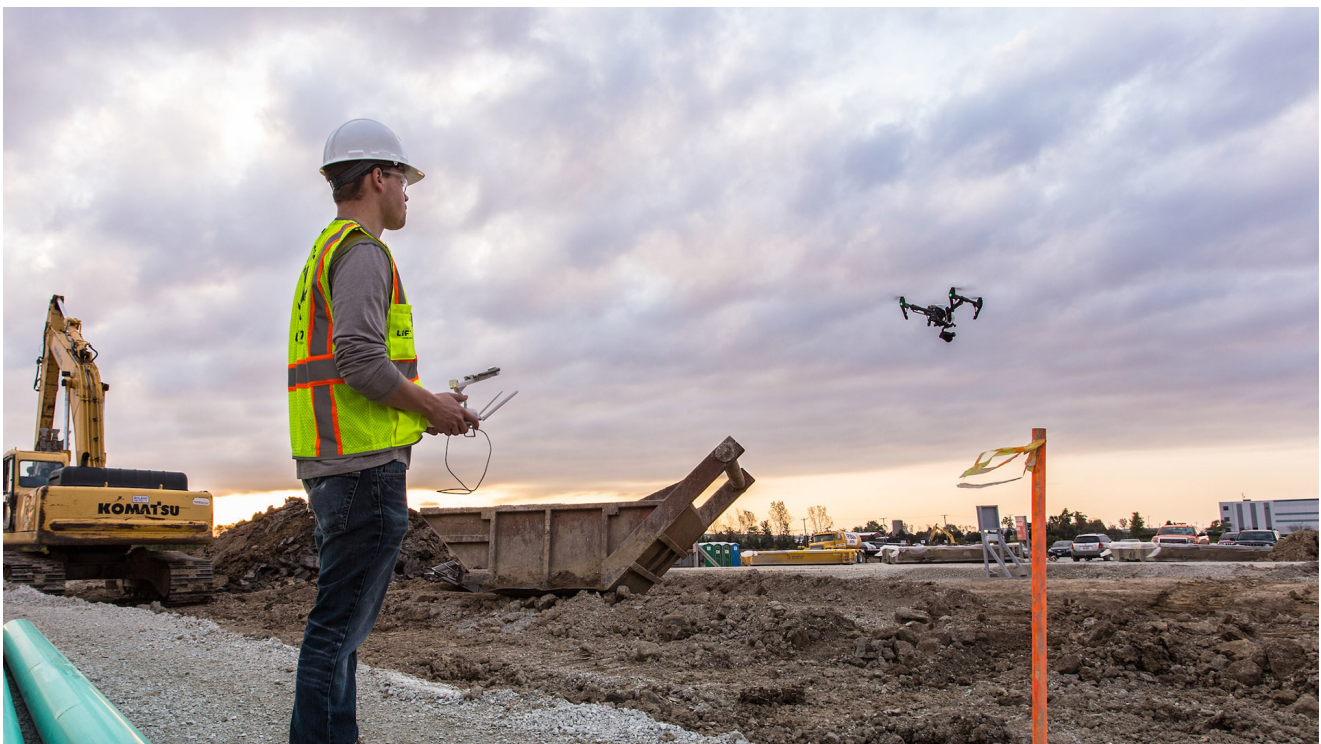


Figure 18: Airborne Scan using Drone on Construction Site [34]

Although the airborne laser scanning technology provides users with a lot of options, the point clouds are not that accurate. This is due to the scan being performed from a high altitude, which in turn result in a lower resolution scan. However, there are workarounds for this. Drones are becoming more and more sophisticated and can therefore carry more sensitive scanners when flying. Another alternative currently being used is that the drones fly lower to scan with greater accuracy as needed.

3.3.2 Mobile Laser Scanning

Mobile laser scanners can be placed on moving object such as cars, bikes or even walking humans. Smaller mobile laser scanners usually use scanning devices with lower sensitivity due to the weight constrains. The method can serve a variety of purposes such as road quality control.

One of the more interesting use cases of this technology now is for self-driving cars. Innovations in the field of AI combined with the greater processing capabilities of recent computers has allowed laser scanning technology to play a major role in the development of self-driving technology [35].

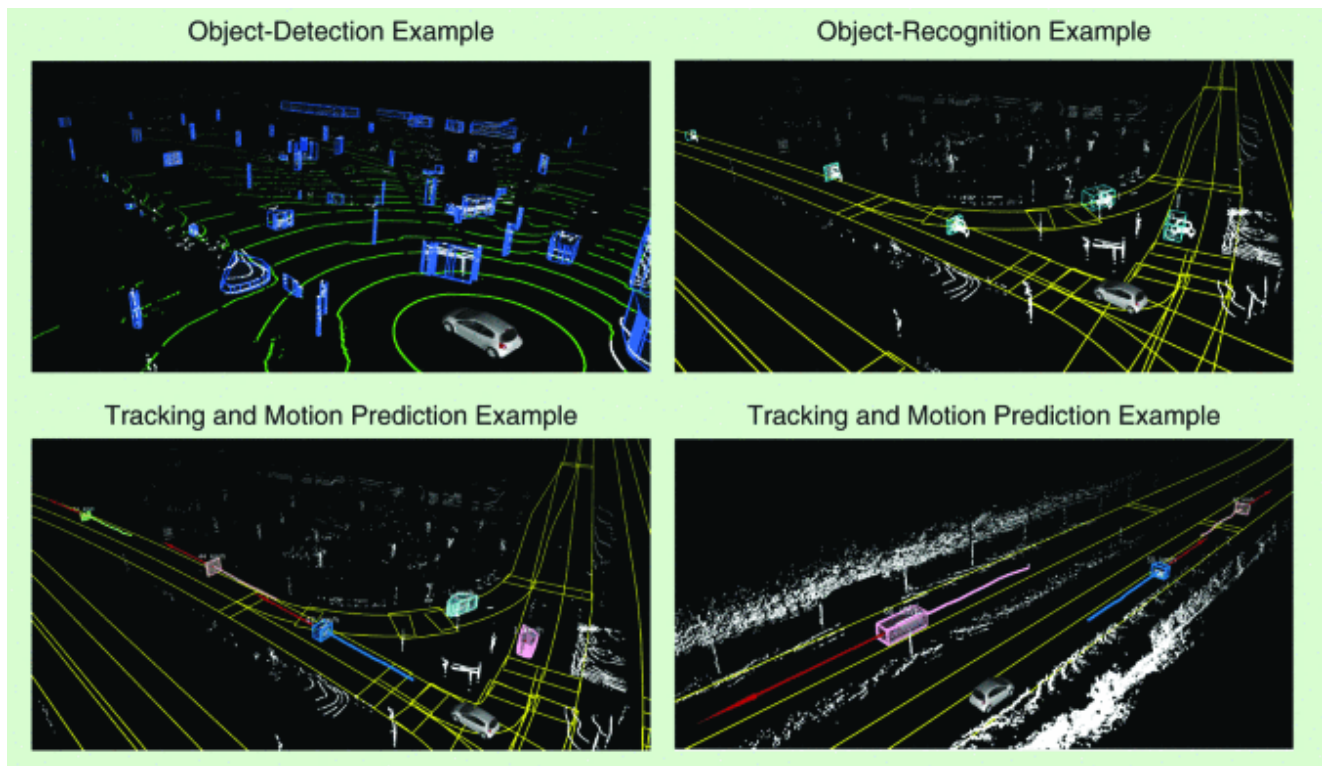


Figure 19: LiDAR for self-driving cars [35].

3.3.3 Terrestrial Laser Scanning (TLS)

TLS refers to laser scanning performed from a stationary location. Usually, TLS is used for precision scanning of already existing as-built structures or terrains within short proximity of the scanning devices' location. Since TLS is performed from a stationary location, occlusions will occur. This is due to the laser scanner not being able to get a complete 3D view of the 3D object from one single stationary location. Therefore, when performing TLS, scans need to be performed from different angles to avoid occlusions.

The advantages of using TLS for generating point clouds of as-built structures is the level of accuracy. The resulting point cloud will be of higher quality due to the scans being taken from shorter distances where the hardware usually is more sensitive. Depending on the use case of the point cloud, image overlays can be added to give the scan a sense of reality. Overlays could for instance be beneficial for house touring where users can use virtual reality headgear to virtually visit pre-scanned properties. See figure 20.



Figure 20: 3D laser scan of house [36].

Each point can also be assigned a color parameter as the fourth dimension. For instance, a scanned point can be given with more than three parameters such as $p = [x, y, z, c]$ where the first three parameters define the location of the point whilst the last parameter, c define the color of the point in either RGB, HEX or another chosen format.

4 Steel Truss Point Cloud Collection and Processing

As described in section 3, 3D laser scans result in a 3D point cloud mapping of the surrounding region as far as the laser scanners range allows. Removing undesired points and inaccuracies within the point cloud is essential in attaining good results. This section will focus on the steel trusses point cloud collection and pretreatment process.

4.1 Steel Truss Point Cloud Collection

The point cloud representation needs to be as accurate as possible to the as-built steel truss. When reverse modeling the steel truss, the accuracy of the point cloud will directly reflect the accuracy in which the designed BIM model can be updated. This section will present the planning process with a workflow chart highlighting the required steps needed to achieve the best results possible.

4.1.1 Scanning Scheme Design

There exists no standard method or blueprint on how to perform a laser scan of a structure. Each structure has unique characteristics which need to be taken into consideration when planning and scanning. However, the workflow for scans requiring this level of accuracy should be as shown in Figure 21.



Figure 21: Laser Scanning Workflow.

For the steel truss, a total of four scans were necessary. Nevertheless, six scans were performed. The steel truss was at the time of the scan located within the factory, waiting to get shipped out. Scanning locations can be seen in Figure 22, where the scans marked in red represents the additional scans.

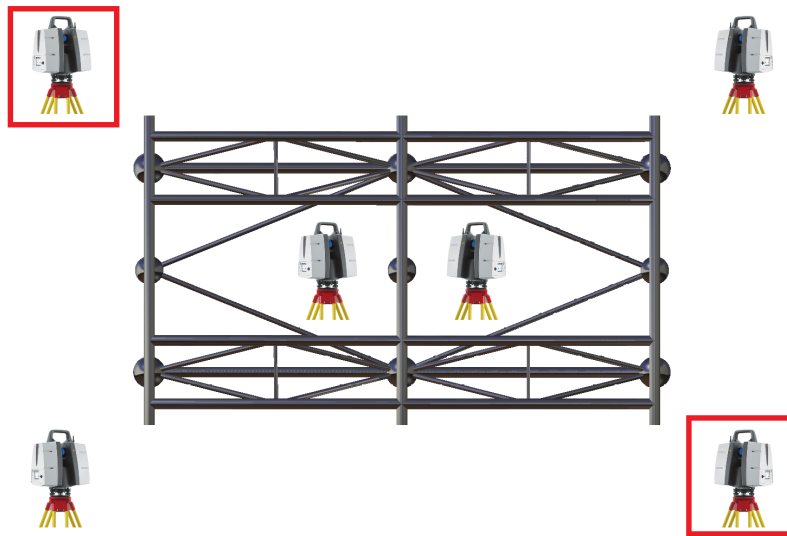


Figure 22: Laser Scanner Positions.

In theory, only four scans are needed as these scans provide sufficient data for a point cloud of the steel truss. Due to the purpose of these scans being constrained to research purposes only, additional scans were taken to have as much data as possible in case some unexpected problems were to occur. However, in a realistic environment where the end goal is results and efficiency, four scans would have been sufficient.

4.1.2 Steel Truss Scanning

The scanned point clouds had a combined total of over 40 million points. Some of the resulting images can be seen in Figure 23. The point cloud is dense around the steel truss, although noise can be spotted.

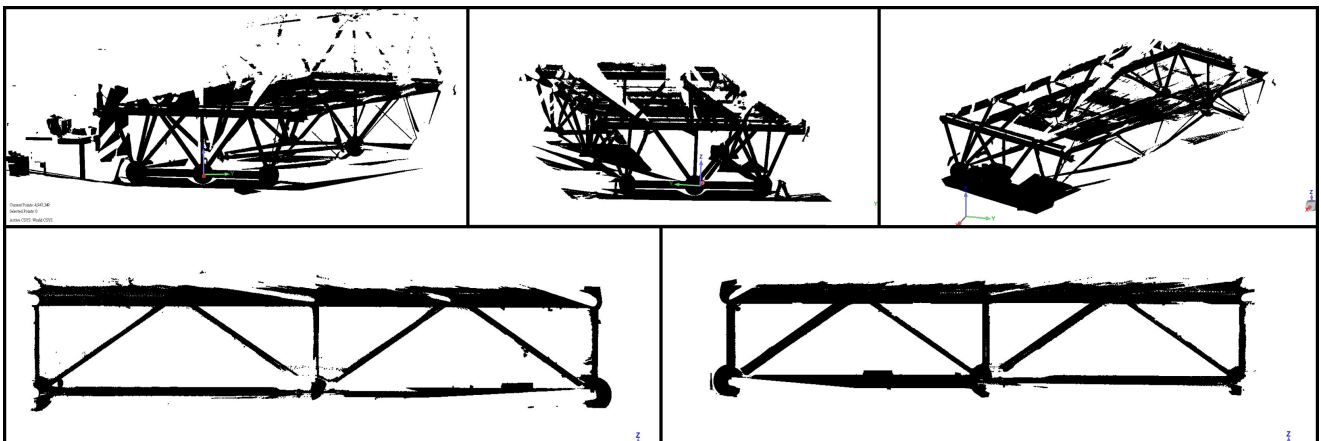


Figure 23: Some Resulting Scans from Different Angles.

4.2 Pretreatment of Steel Truss Point Cloud

The raw point cloud data must undergo pretreatment before being fed to the algorithms for intersection point extraction. The pretreatment process will result in a merged point cloud with reduced noise. Current algorithms alone are not effective enough to separate and remove surrounding noise from a given element of interest. A combination of algorithms and human work is therefore needed for this step.

4.2.1 Steel Truss Point Cloud Merging

As described in section 4.1, the steel truss is scanned from six stationary locations. This corresponds to six point clouds with their individual coordinate systems. To get a complete 3D view of the steel truss, all six point clouds need to be merged into one unified coordinate system.

The scans taken from different locations will have some overlapping points by design. This is due to the same area being in proximity of two or more scans, hence, being included. When point clouds merge, a transformation relationship is found between two clouds with overlapping points. The transformation relationship can be described using a rotation and translation matrix as seen in Equation 2.

$$P_t = R \times P_s + T \quad (2)$$

Where P_t is the transformed point, P_s is the original point from the stationary location and T is the translation matrix. The rotation matrix R is a multiple of three rotational matrices targeting rotation in x , y , and z direction. See Equation 3.

$$R = R(\alpha) \times R(\beta) \times R(\gamma) \quad (3)$$

Where

$$R(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{bmatrix}, \quad R(\beta) = \begin{bmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{bmatrix}, \quad R(\gamma) = \begin{bmatrix} \cos \gamma & \sin \gamma & 0 \\ -\sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The critical part when merging point clouds is to find matrix R and T . For this paper, the feature-based point cloud merging is used.

4.2.1.1 Introduction to Feature-based Merging

The feature-based point cloud merging principle is to obtain the rotation and translation matrix required for point cloud merging through several sets of paired features. These paired features are also referred to as overlapping points and can be seen in Figure 16. What is most important when conducting feature-based merging is the selection of features. These features exist in the form of points, lines, and surfaces. Among them, point features are the most commonly used. There are three general methods in which the feature points are attained.

- **Sticking labels:** sticking labels such as reflective sheets and checkerboard paper on the surface of the object. When the data accuracy is high, the coordinates of these points can also be obtained and used for merging purposes.
- **Placing the target:** Most scanners of today's brands are equipped with special target balls. The surface of the target balls is coated with a special material to make them have unique laser reflectivity, which can feedback a large number of spherical sampling points with better quality. When merging, the sphere center coordinates can be extracted through the fitting of the spherical points and used as overlapping points. See Figure 24.



Figure 24: Equipped Spherical Target Balls.

- **Manual selection:** Manual selection is preferred when no preemptive measures have been taken to obtain feature points. Manual selection is based on picking feature points manually within the point cloud. This method is the least accurate one of the three described methods.

For the merging process in this paper, no target balls were placed due to the steel truss having built-in spherical elements. These spherical elements are used to extract center coordinates on which merging will occur.

4.2.1.2 Point Cloud Merging

A manual method is chosen to make an initial rough merging of the point clouds. This method is chosen because the point cloud scans are taken from distances somewhat far from each other. Geomagic Control X is used to manage and move the point clouds to the locations they are intended to be. This process can be seen in Figure 25. (A) show two scans, one in pink and one in blue, in their initial location. (B) show the same scans once they have been manually moved to their intended location.

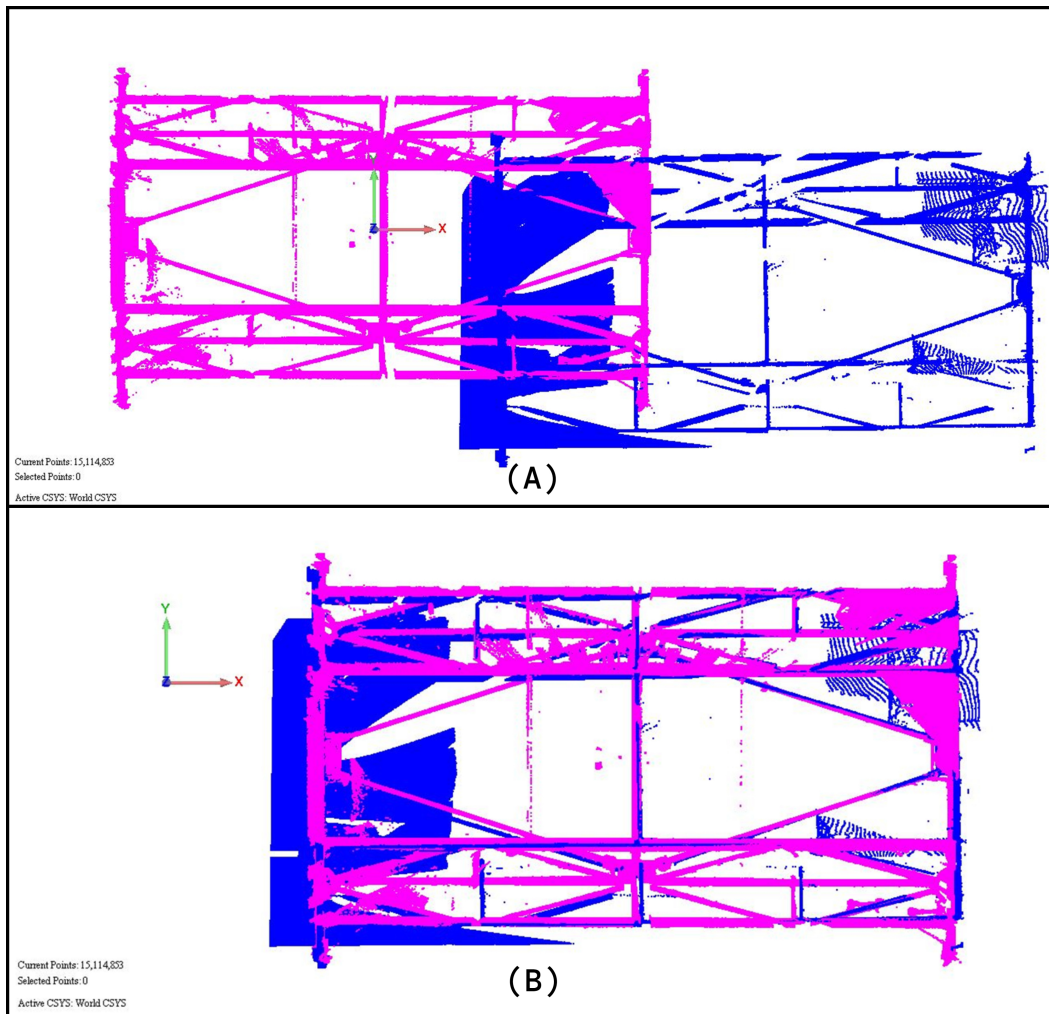


Figure 25: Manual Point Cloud Initial Merging.

Although manual merging can be a somewhat acceptable solution, the precision and accuracy are not ideal. The deviations in two of the scans can be seen in Figure 26.

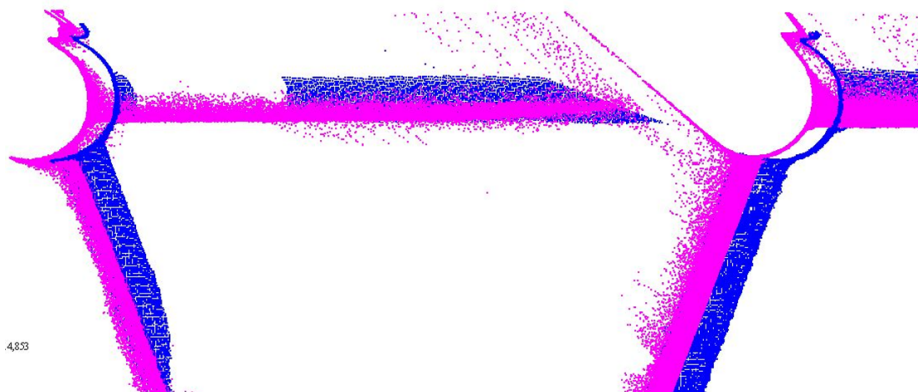


Figure 26: Deviations from Manual Merging.

A feature-based merging is therefore conducted to achieve the desired accuracy. The spherical parts of the steel truss are used as the main features on which the merging is conducted. In essence, one of the scans is chosen to have a fixed location. The other five scans will be transformed slightly to have their spherical elements align with the fixed scan, one after the other. This method of controlling the spherical elements when aligning the point clouds can be seen in Figure 27.

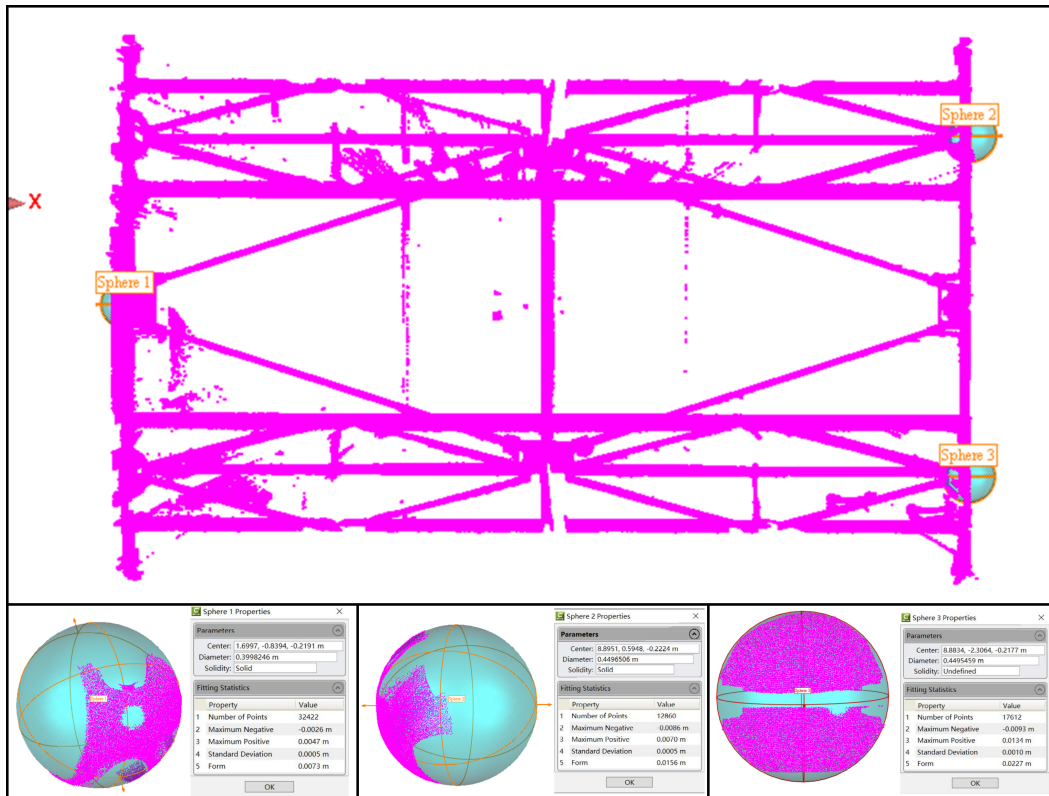


Figure 27: Point Cloud Spherical Element Alignment Control.

The spherical elements in Figure 27, Sphere 1-3, are control elements positioned in Geomagic Control X at a fixed location in accordance with the fixed scan. The elements' positions and their diameters can be seen in Table 3.

	x	y	z	D
Sphere 1	1.6997	-0.8394	-0.2191	0.3998
Sphere 2	8.8951	0.5948	-0.2224	0.4497
Sphere 3	8.8834	-2.3064	-0.2177	0.4495

Table 3: Extracted intersection points from DWG drawing.

Using a combination of manual and feature-fitted merging results in one point cloud with sufficient accuracy. Both clouds merged in Figure 26 can be seen to have much more accuracy after conducting a feature-fitted merging. See Figure 28.

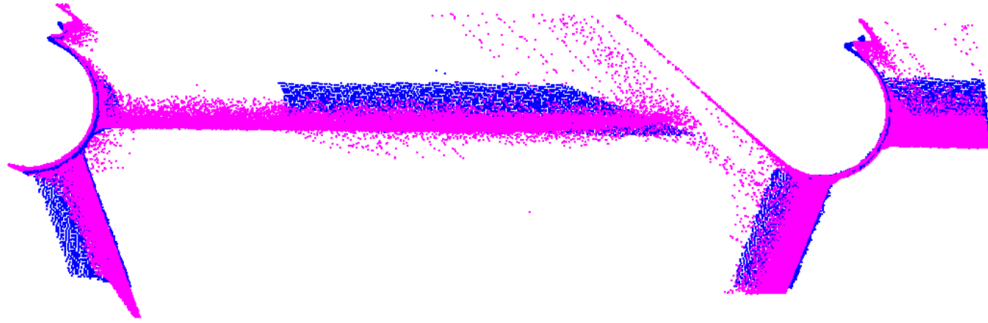


Figure 28: Scans Post Manual and Feature Fitted Merging.

The results from the merging process can be seen in Figure 29. Although all six scans have merged successfully, unwanted noise still exists in the point cloud. These are points created by the floor being scanned, instrument errors, and solids in the background. The noise reduction process is further described in section 4.2.3.

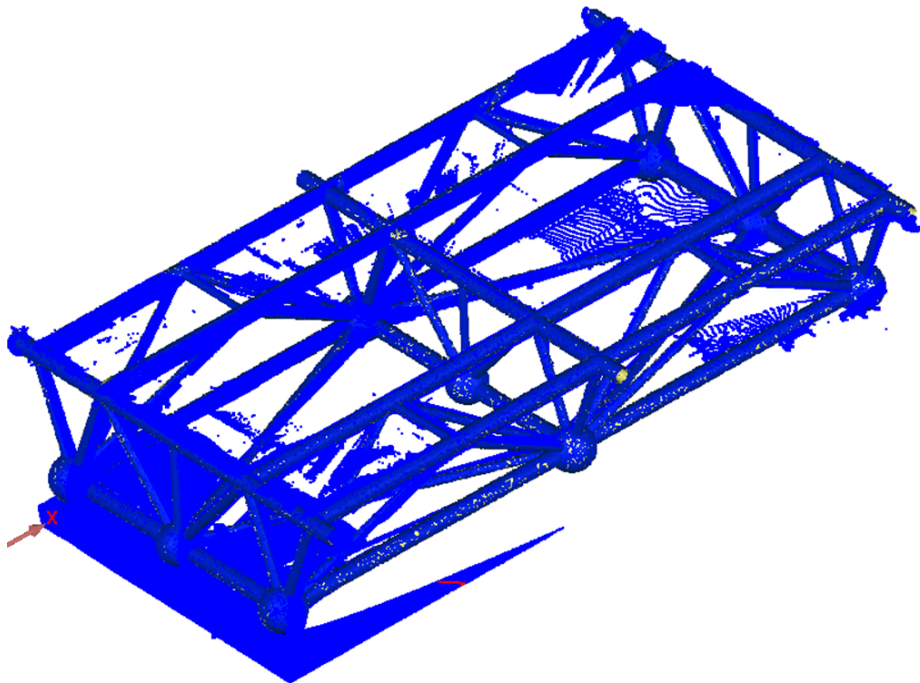


Figure 29: Scans Post Manual and Feature Fitted Merging.

4.2.2 Steel Truss Point Cloud Noise Reduction

During the 3D scanning process, a large number of noise points are gathered. As described earlier in sections 1 and 3, there can be various reasons for why these noise points are generated. However, the noise points can be categorized into four main categories as follows:

- **Drifting points** - Object surface points positioned far away from the object itself. Drifting points are usually generated due to inconsistencies with the material reflectiveness or problems in the scanning devices timer.
- **Isolated points** - Points in dense clusters far away from the object of interest. Isolated points are mostly generated due to solids in the background.
- **Redundant points** - Useless points located outside the scanning area. These points are usually generated due to instrumental errors with the timer.
- **Confusion points** - Points which does not belong to the object of interest but are located on or in proximity. Confusion points can be caused by different factors such as the ground, small inaccuracies with the scanning device, tiny dust particles in the air, etc.

Of the noise categories mentioned above, the fourth one is the hardest to detect and delete. Removing confusion points located on the object of interest can result in removing critical points which actually belong to the object itself. For the steel truss, noise cancellation commands within Geomagic Control X are used to remove noise. Final results can be seen in Figure 30.

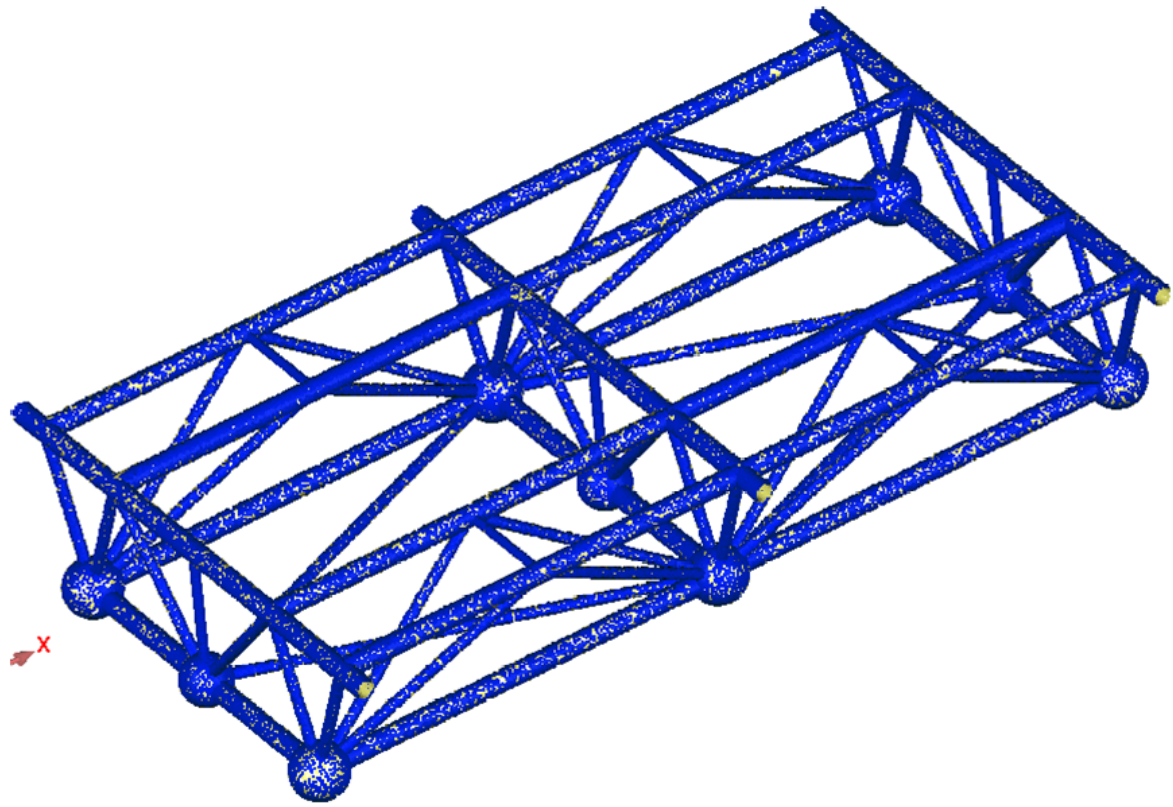


Figure 30: Final Point Cloud

5 Intersection Points Extraction

In order to reverse model an accurate digital twin of the as-built steel truss, intersection points must be extracted from the point cloud. Retrieving intersection points of a steel truss from a point cloud is traditionally a time-consuming job, as each element needs to be identified and extracted manually. This is typically done using point cloud processing software tools such as Geomagic Control X or AutoDesk ReCap Pro.

Elements are first separated as individual point clouds. A center-line will then be generated for each of these elements. Once the center-line is generated for all elements, the intersection points can be predicted and extracted. Although this process works, it is not ideal. The whole element separation, center-line fitting, and intersection point extraction process is very repetitive. If a steel truss is made of 30 elements, all 30 elements need to be separated, then a center-line needs to be fitted, etc.

The proposed method will research and develop unsupervised ML algorithms to automate most of the process required to extract the intersection points from the point cloud. The intent is to reduce the manual, repetitive work and automate as much of the workflow as currently possible. The proposed method of automation can be seen in Figure 31.



Figure 31: Proposed Method of Extracting Intersection Points.

The finalized point cloud from section 4.2 consists of 299 394 exclusive points. Research conducted in this paper required many repetitive trials of running the code and debugging. The number of points was significantly reduced due to the limited processing power of the laptop used. Algorithms were also sectioned into three separate entities. The complete python codes can be seen in appendix A1, A2, and A3.

5.1 Sectioning & Element Center Point Extraction

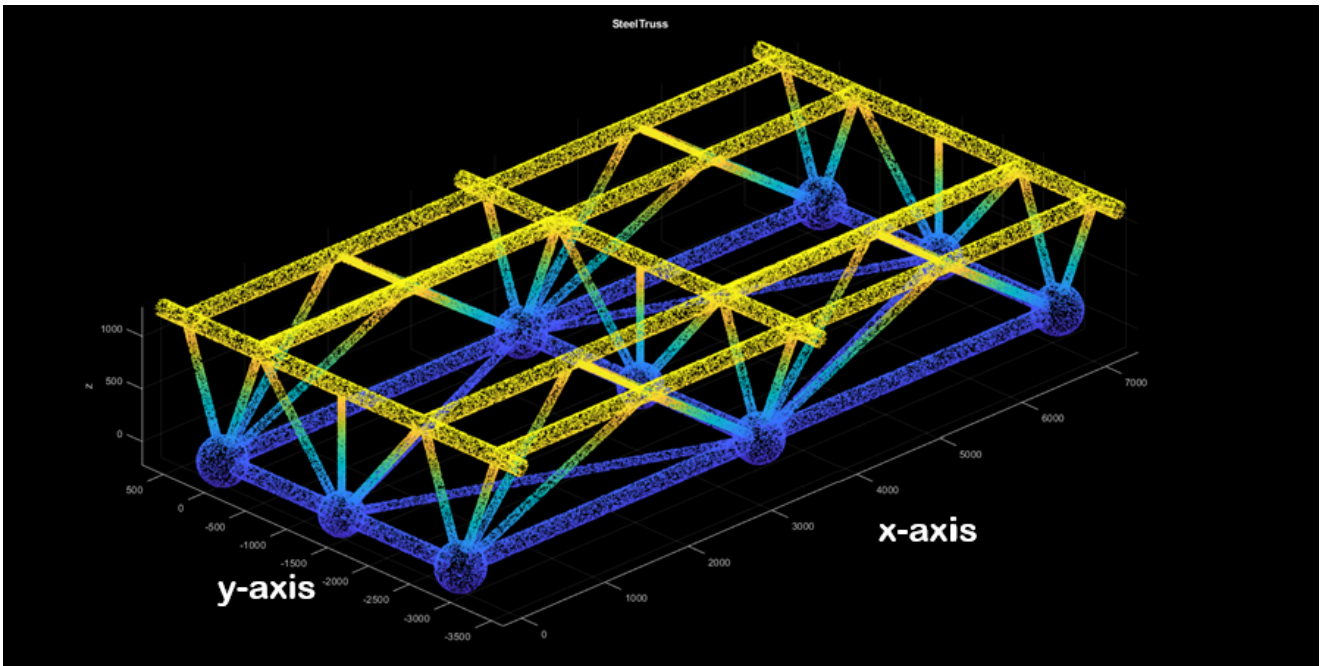


Figure 32: Steel Truss Point Cloud Visualized

The steel truss is sectioned along the x and y – axis. The component is approximately 4200mm wide along the y – axis and 7500mm wide along the x – axis. Therefore, due to differences in lengths, the width of each section along the x – axis is chosen as 200mm while 70mm is chosen for sections along the y – axis. The widths are chosen based upon the lengths of the steel truss along the respective axis that is being sectioned along. Figure 33 displays a section along the x -axis.

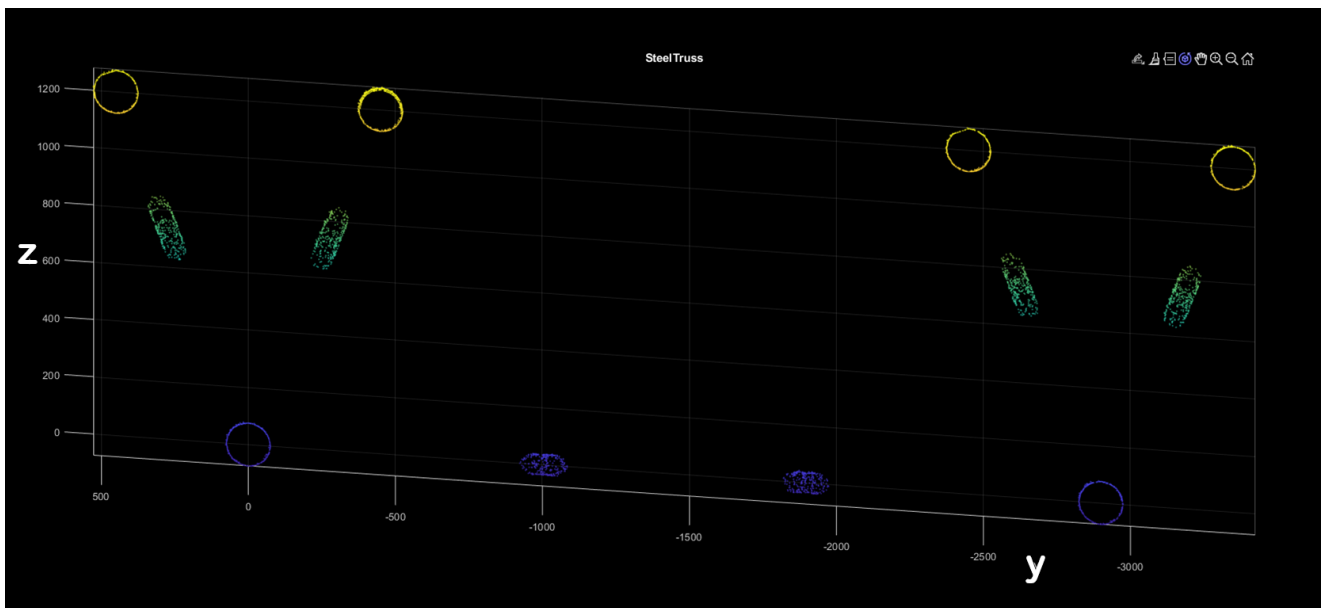


Figure 33: Example section along the x-axis.

The cutout section, in Figure 33, is 3-Dimensional, although it might seem to be 2-Dimensional. This is due to the sectioning width being almost 20 times smaller than the horizontal axis (y) and six times smaller than the vertical axis (z). The next step is to project all these 3D points onto a 2D plane. This is done by eliminating the dimension which is being sectioned along. See Figure 34.

```
SECTION ALONG X-AXIS
point_3D = [x, y, z] ==> point_2D = [y, z]

SECTION ALONG Y-AXIS
point_3D = [x, y, z] ==> point_2D = [x, z]
```

Figure 34: Point Elimination Converting 3D Points to 2D.

By doing this, each element will appear as individual clusters of points once they have been projected onto a 2D plane. The Mean Shift Algorithm is then run on sections to identify each set of clusters as individual elements.

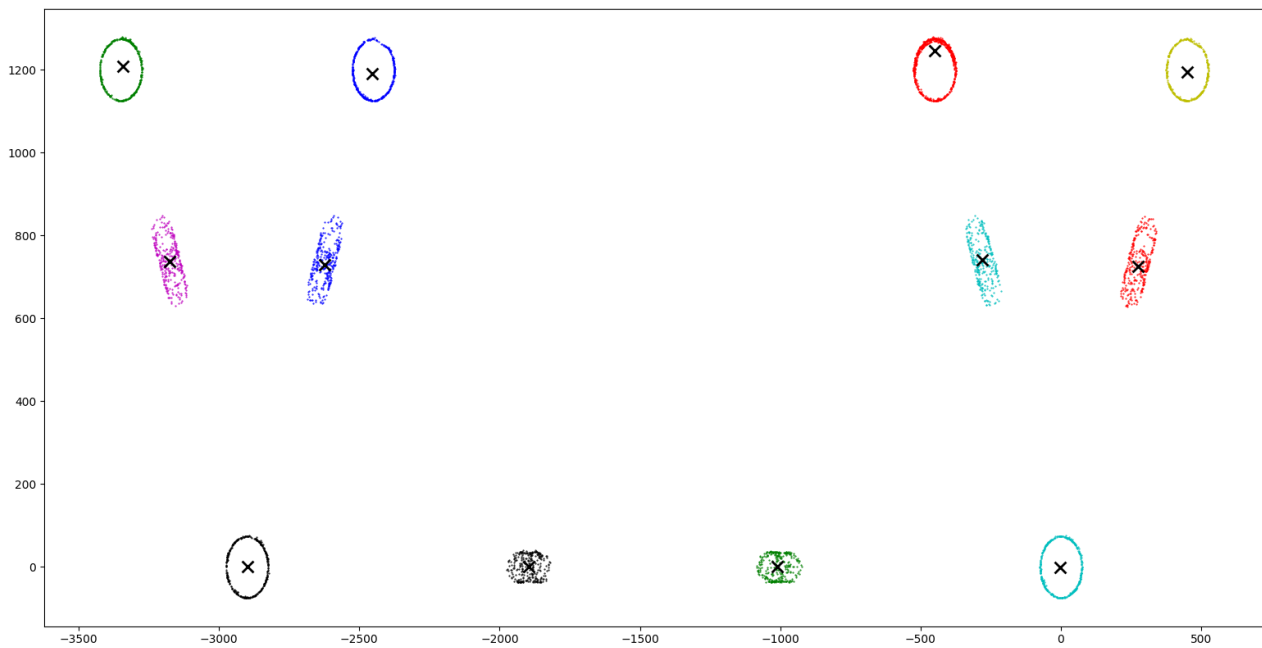


Figure 35: Section Projected on to a 2D plane.

The result of running the Mean Shift algorithm on a section can be seen in Figure 35. Each element/cluster is represented with a color, and the algorithm's predicted mean-point/center point is represented as a black cross for each element. Due to inaccuracies in the point cloud, some areas will have more points than others. This causes the predicted center points to be inaccurate. See Figure 36.

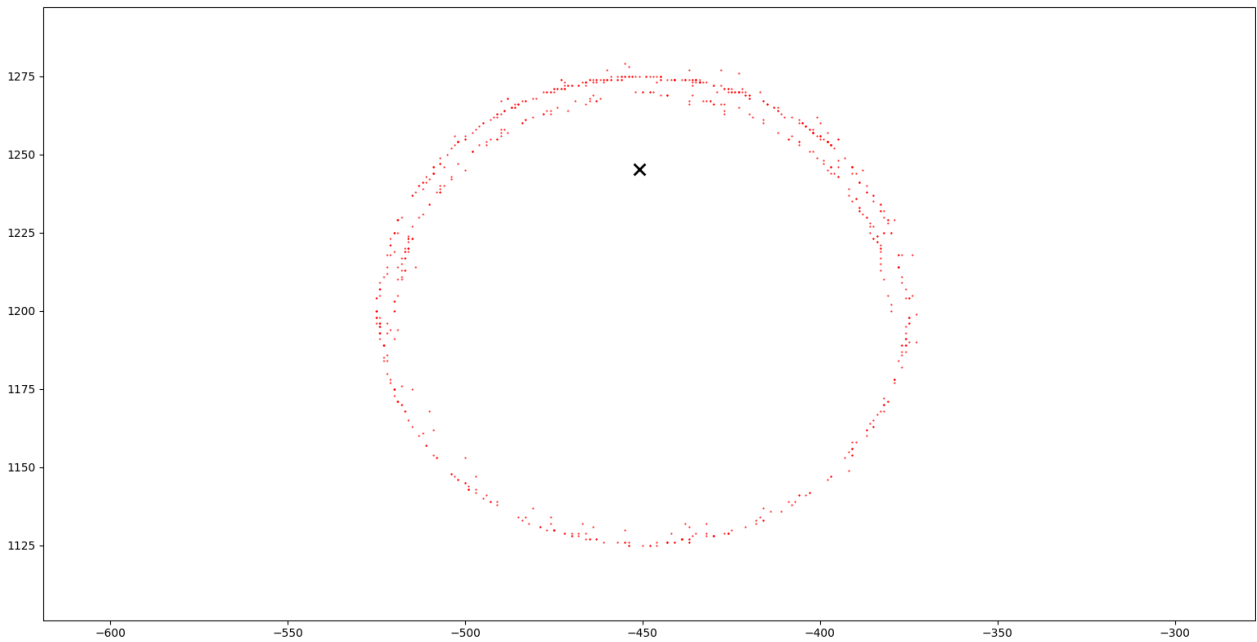


Figure 36: Zoomed in example of one element where the center point is inaccurate.

Figure 36 is displaying an example where the center point is predicted wrong. Ideally, the center point should be in the middle of the element. This problem is fixed by building on top of the existing Mean Shift algorithm. The center point is set to be the average of the ten highest and lowest values along the horizontal and vertical axis. This results in an even distribution where the center points are more aligned with the “true” center. The results can be seen in Figure 37.

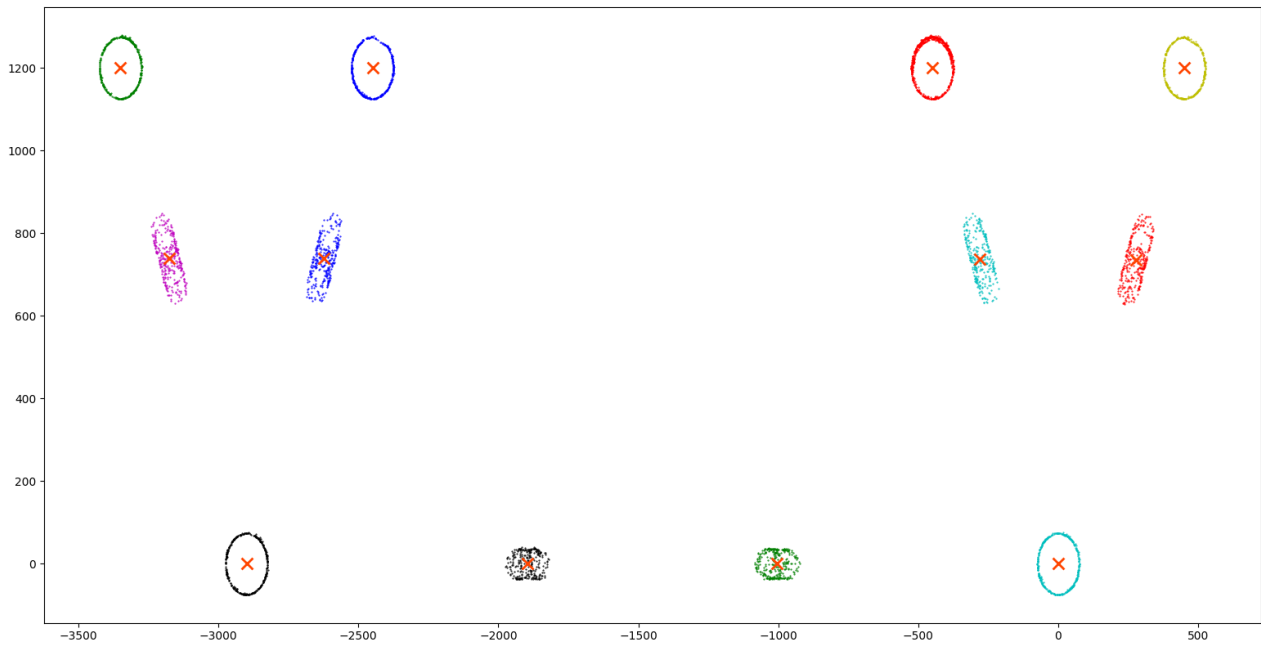


Figure 37: Modified center points can be seen as red crosses.

A loop is then created to make a section, run the Mean Shift Algorithm, and predict center points. The resulting center points extracted from all sections can be seen in Figure 38.

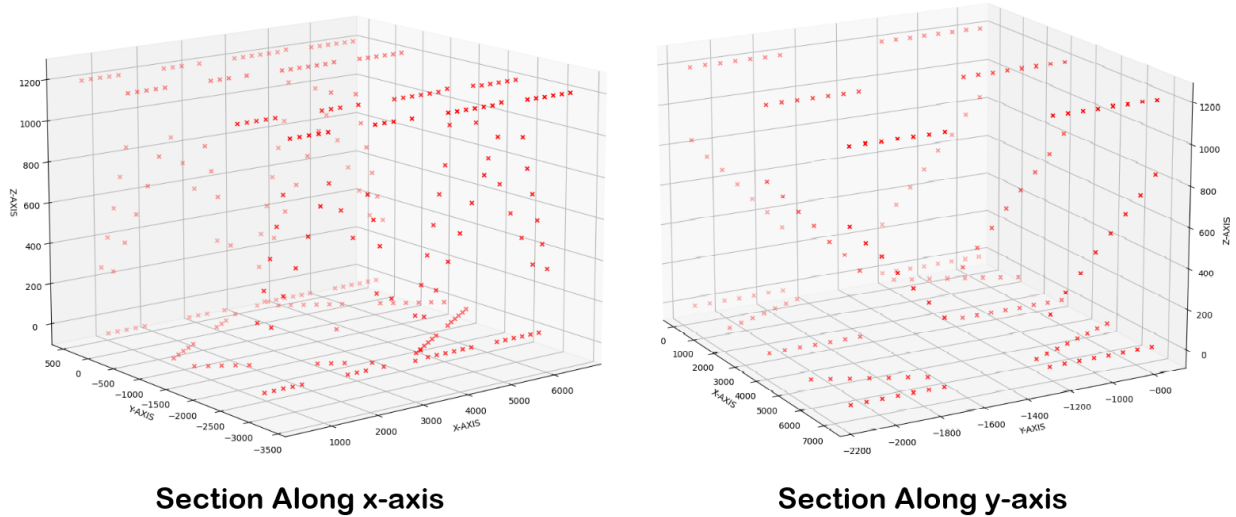


Figure 38: Resulting center points from using Sectioning and Mean Shift Algorithm.

5.1.1 Mean Shift Algorithm

Mean shift is a clustering algorithm heavily used for unsupervised ML to identify and label clusters within a set of data points. The algorithm can detect points within proximity of each other and cluster them together into different sets of clusters. This is great for situations where the points of interest are in their respective regions of the graph. Figure 39 shows the clustering algorithm in action where each cluster is marked in its color.

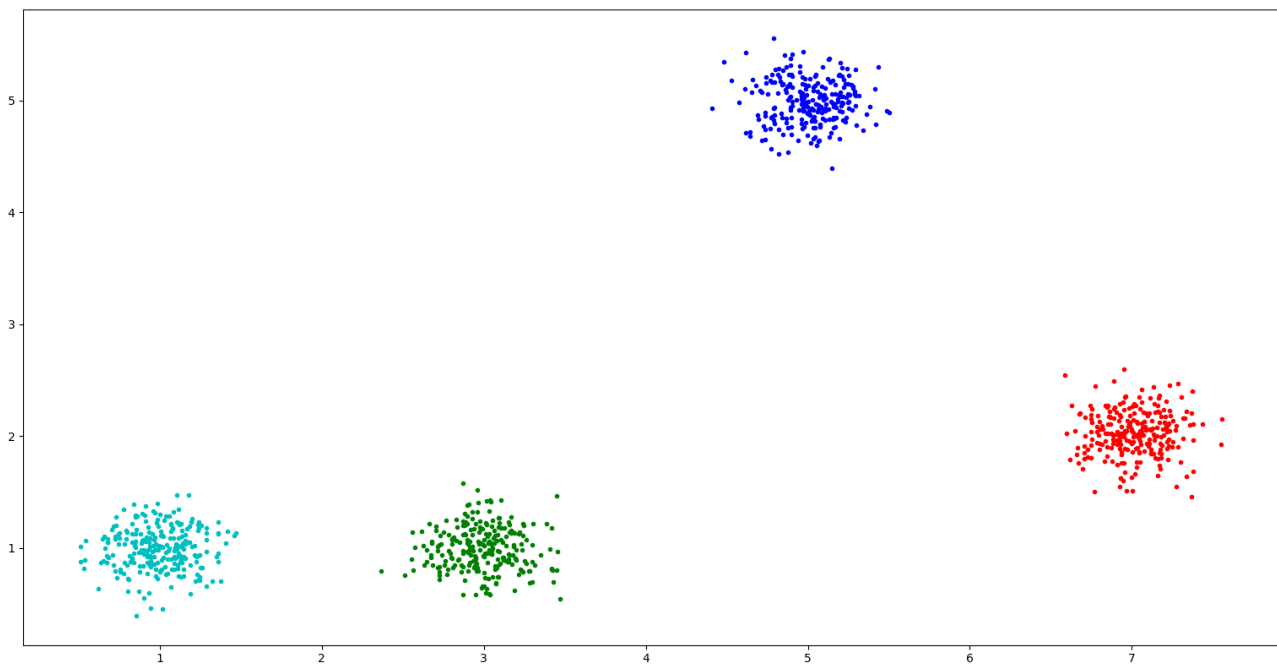


Figure 39: Four clusters from a randomly generated data set of 1000points.

The algorithm works by assigning several points to the plot. These points can either be assigned randomly or predefined within the graph. The points serve as references. A mean value will be calculated from all the points within a given region from the reference points. The reference points are then updated, and the process will continue until convergence occurs [37]. For each iteration, the reference point will move closer to the predicted cluster of points.

5.1.1.1 Mathematical Formulation

$$x_i^{t+1} = m(x_i^t) \quad (4)$$

In Equation 4, x corresponds to the i 'th center point while t is the number of iterations done. $m(x_i^t)$ is the formula used to calculate the weighted mean point, also referred to as the mean shift vector, from all the points in the specified region surrounding x_i^t [37].

$$m(x_i^t) = \frac{\sum_{x_j \in N(x_i)} K(x_j - x_i) x_j}{\sum_{x_j \in N(x_i)} K(x_j - x_i)} \quad (5)$$

$N(x_i)$ in Equation 5 refers to the all the points given within a given region of x_i . K is the weighting kernel function. A Gaussian kernel is typically used to calculate the mean shift vector although other kernel functions can be used. Two of the more popular kernel functions can be seen in Equation 6 and Equation 7 [38].

$$K(u) = \frac{1}{2} \begin{cases} 1, & -1 \leq |u| \leq 1 \\ 0, & \text{else} \end{cases} \quad (6)$$

Union Kernel [38]

$$K(u) = \frac{1}{(2\pi)^{\frac{d}{2}}} \times e^{-\frac{|u|^2}{2}} \quad (7)$$

Gaussian Kernel [38]

In Equation 7, d refers to the dimension of the space. A Gaussian kernel is used in this paper. The kernel function will return a higher value for points closer to the reference point while assigning higher values to points further away.

5.1.2 Error Handling

The Mean Shift Algorithm will have trouble predicting the elements for sections where the elements are placed perpendicular to the axis that is being sectioned along. See Figure 40.

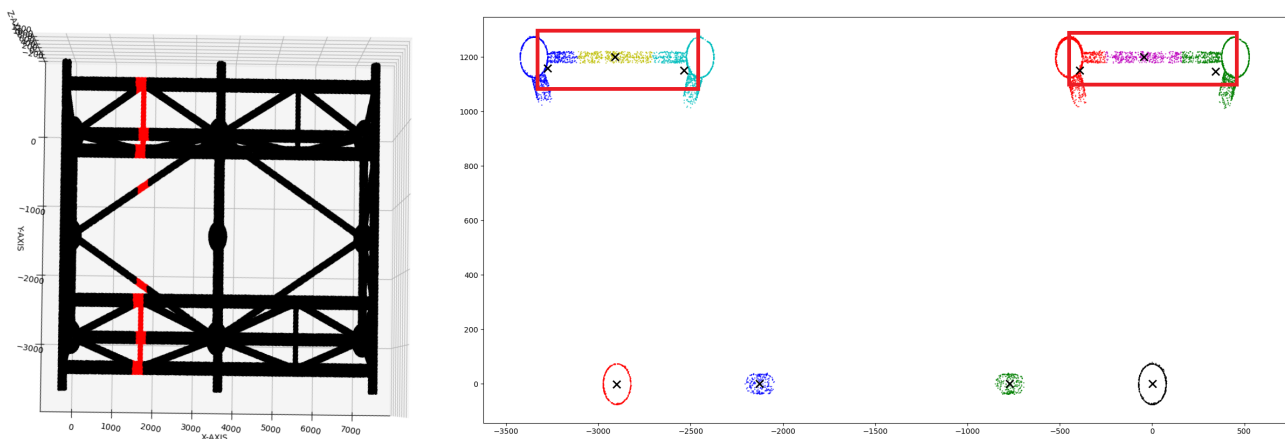
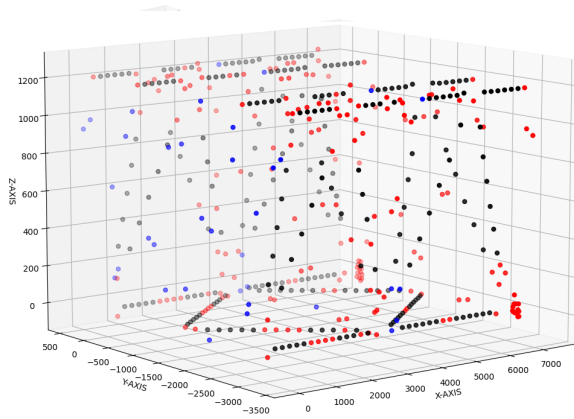


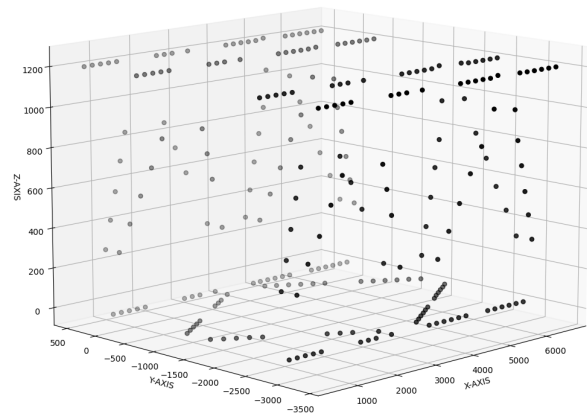
Figure 40: Section where elements are perpendicular to sectioning axis (x-axis).

The Mean Shift Algorithm will in cases like the one in Figure 40 be unable to predict the desired elements correctly. Ideally, the perpendicular element should be neglected. To work around this problem, the whole section is discarded. This was be done automatically by counting the most frequent number of elements occurring in each section and removing the sections which return an element-count higher or lower than the most frequent value. The remaining sections with minor inaccuracies are manually removed. See Figure 41.

Section along x-axis

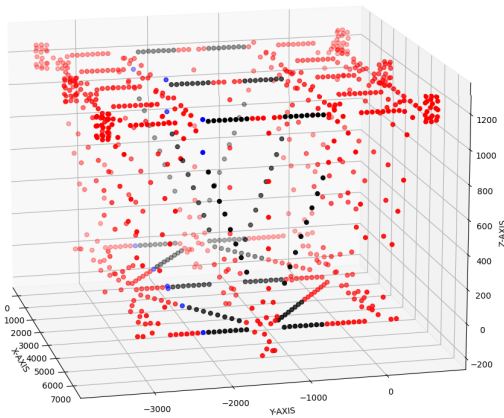


RAW OUTPUT

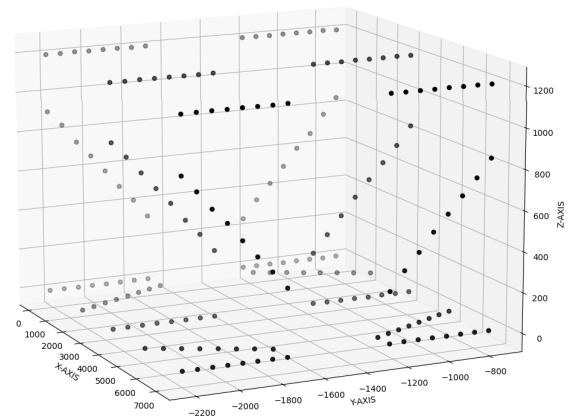


OUTPUT WITH SECTIONS DISCARDED

Section along y-axis



RAW OUTPUT



OUTPUT WITH SECTIONS DISCARDED

Figure 41: **Blue** » Manually removed; **Red** » Automatically removed; **Black** » Remaining

Removing the sections will not pose as a major problem as computer vision will be used to predict the location of some of the removed points. This process is described further in Section 5.2.

5.2 Element Recognition

The center points for the elements are returned as a set of points from the algorithm described in Section 5.1. These points must be grouped together as separate elements in order to extract a best-fit line through them. See Figure 42.

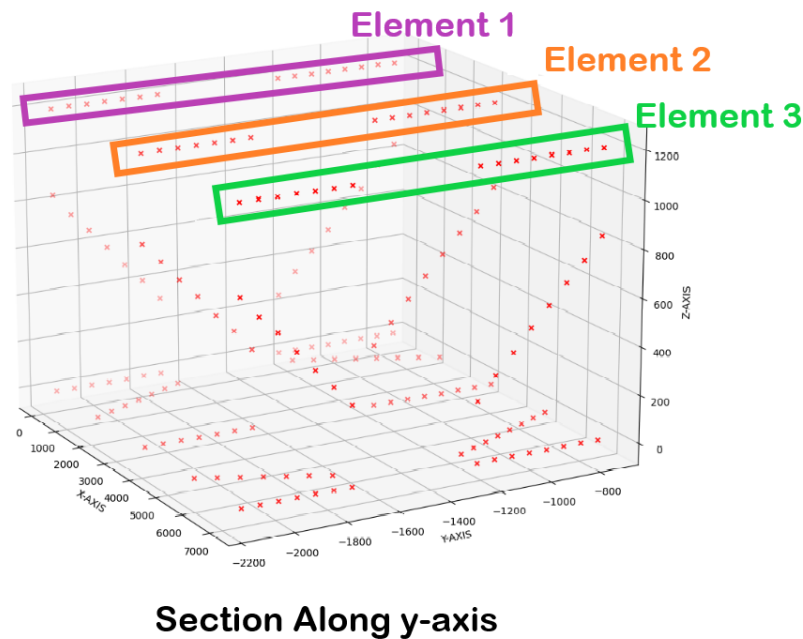


Figure 42: Section along y-axis. Element center points visually highlighted to illustrate the desired goal.

Three elements are highlighted with separate boxes in different colors. It is easy for humans to see that these sets of points might correspond to separate elements as the points follow the same pattern, but it is hard task for the computer. However, inspiration can be taken from computer vision to solve this problem.

Moving from one section to the other create an illusion that resemble moving objects. This is because one elements center point in one section will be moved to another location in the next, unless the element lies parallel to the direction of sectioning. In that case, the point will stay stationary. See Figure 43.

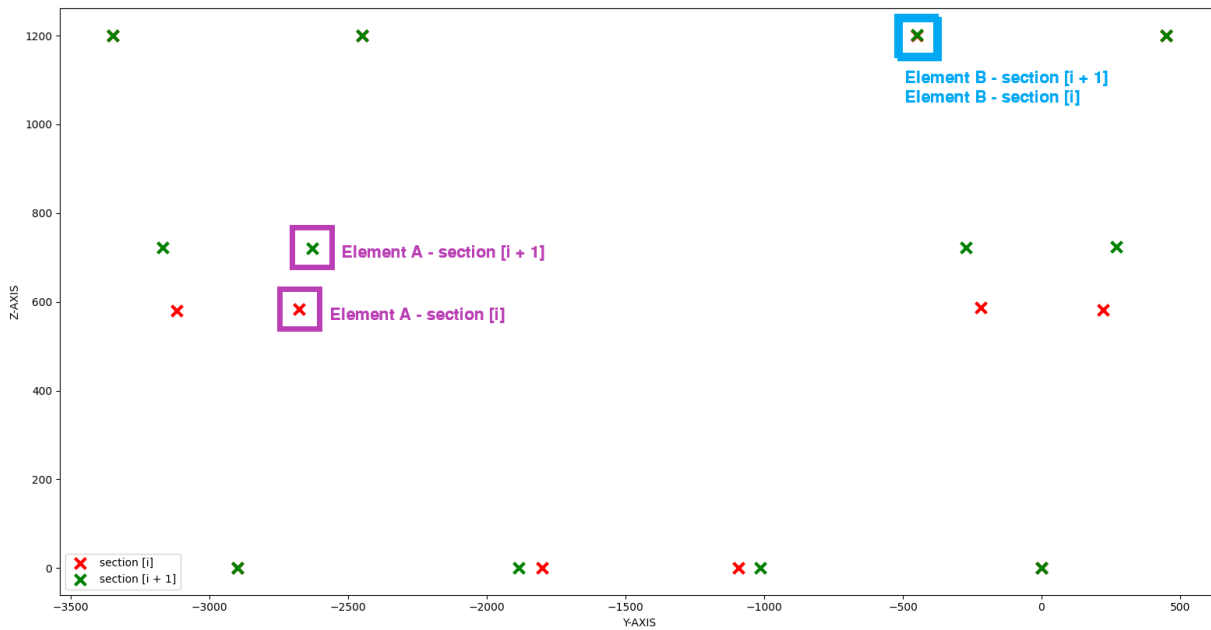


Figure 43: Moving from section $[i]$ to section $[i + 1]$ (next section).

From Figure 43, the center point for Element A is changing position from section $[i]$ to section $[i + 1]$. Element A is therefore not parallel to the sectioned axis. Center point for Element B is seen to not be moving from one section to the next and is therefore regarded a parallel to the sectioned axis.

Looping through all the sections along one axis and detecting the closest point will make it possible to classify the closest points as individual elements. In areas where the sections are removed, or no point is found, the algorithm will predict where the next point should lie, based on the data from the previous points in the element. These are the main principles behind the Element Recognition Algorithm described in this section.

The results from running the Element Recognition Algorithm can be seen in Figure 44. The algorithm is pretty accurate when it comes to recognizing the center points for each element along the $x - axis$. Some flaws were detected when sectioning along the $y - axis$, but these elements were neglected as they were non-significant for the end results. Overlapping Elements were also removed.

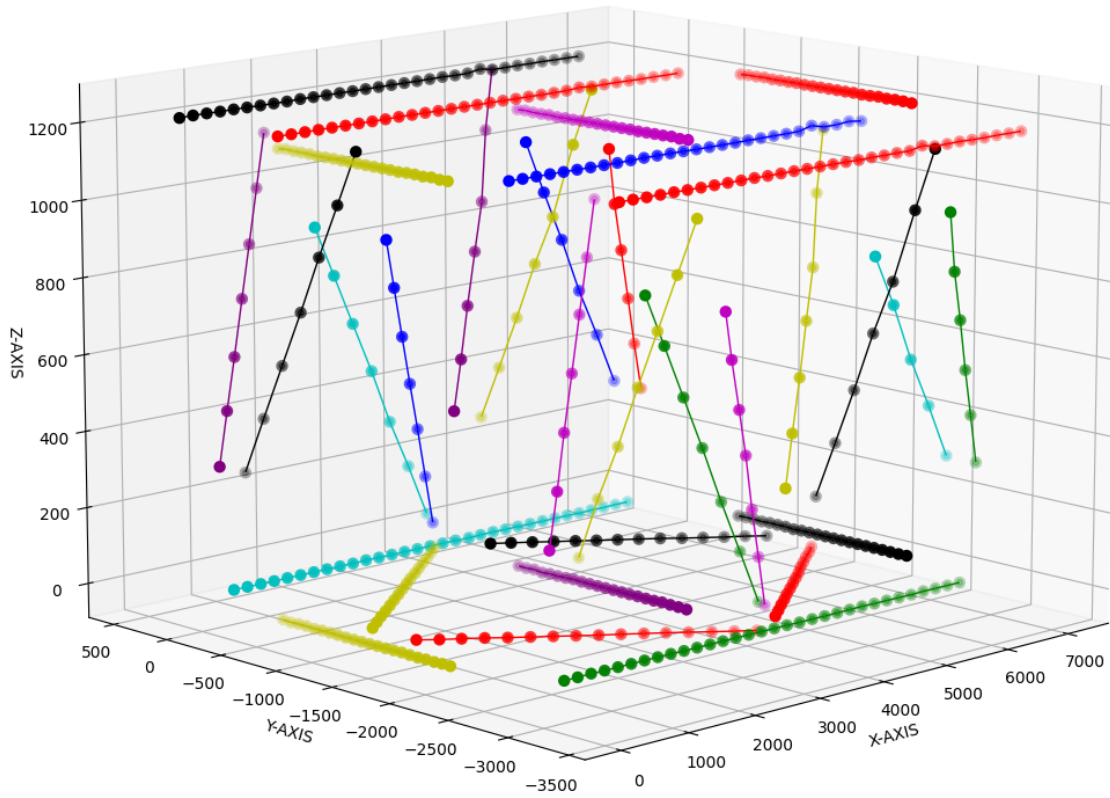


Figure 44: End results from the Element Recognition Algorithm.

5.2.1 Element Recognition Algorithm

A loop is created to loop through all the sections. Each section will be instantiated as separate objects where an axis value will be stored. The axis value corresponds to the mean value along the sectioning axis. See Equation 8.

$$AV = \frac{2 * section_{start} + section_{size}}{2} \quad (8)$$

$section_{start}$ is the value along the sectioning axis where the current section starts.

$section_{size}$ is the size of the section.

AV is the Axis Value

Each point within a section is instantiated as separate objects where the default Element is set as None. The points will then be looped through and if a point has no elements assigned, a new one will be instantiated. Then, within the same loop, a new loop is created to measure the distance

between Point A (current point) and all the points within the next section. The closest point will then have its element overwritten from None to the same as Point A.

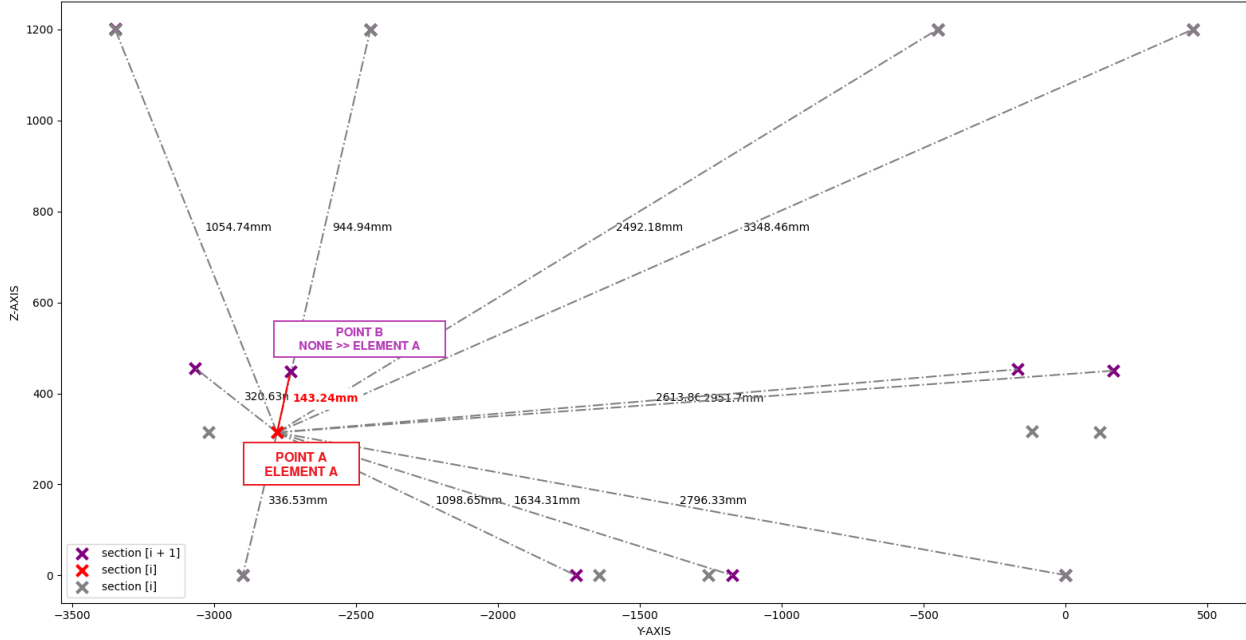


Figure 45: Matching point to Element when moving from section [i] to section [i + 1].

For sections where no points exist, the algorithm will use regression to predict where the next point should be. The proposed method relies on using Gauss-Newton Least Squares Method to find a best-fit line through all the previous points within the element.

$$El_k = [(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)]$$

$$(x_{n+1}, y_{n+1}) = (H(X), V(x_{n+1}))$$

Where x is the set of all the horizontal values (x_1, \dots, x_n) in the 2D projection of El_k

$$H(X) = \begin{cases} x_n + \frac{x_{max} - x_{min}}{n-1}, & x_1 < x_n \\ x_n - \frac{x_{max} - x_{min}}{n-1}, & x_1 < x_n \end{cases} \quad (9)$$

$$V(x) = ax + b \quad (10)$$

Where

$$a = \frac{n \sum xy - (\sum x)(\sum y)}{n \sum x^2 - (\sum x)^2}$$

$$b = \frac{\sum y - a \sum x}{n}$$

Gauss-Newton Least Squares Method

All x and y values are from El_k (Element k)

Equation 9 and 10 formulates the mathematical basis for the algorithms point prediction part. (x_{n+1}, y_{n+1}) is the predicted point for El_k . The 3D spatial points can be created by appending the missing Axis Value into the 2D point. See Figure 46.

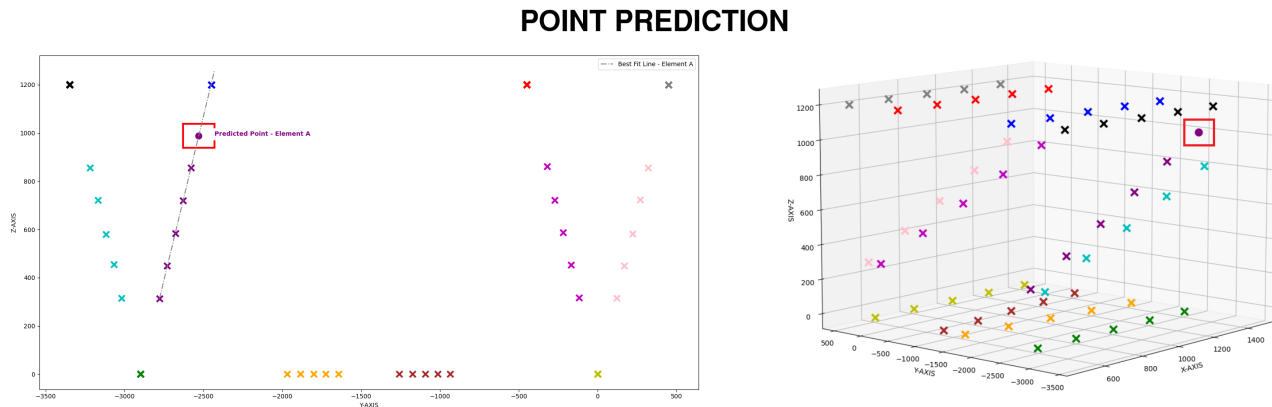


Figure 46: Example Point Prediction for Element A.

5.2.2 Error Handling

Some boundary conditions are set to prevent predicting points outside the area of interest. For instance, in Figure 46, no points should be predicted to be above or below a certain value. In the case of Figure 46, the threshold vertical values were set to between -5 and 1220. If a point is predicted with a vertical value outside this region, the point is neglected.

5.3 Intersection Point Prediction

Once center points for individual elements are identified and extracted, a best fit line can be parametrized through the points. Then, a set of elements are chosen to be the "main" elements where the intersections occurs. The *Intersection Point Prediction Algorithm* is used to predict where the intersections occurs.

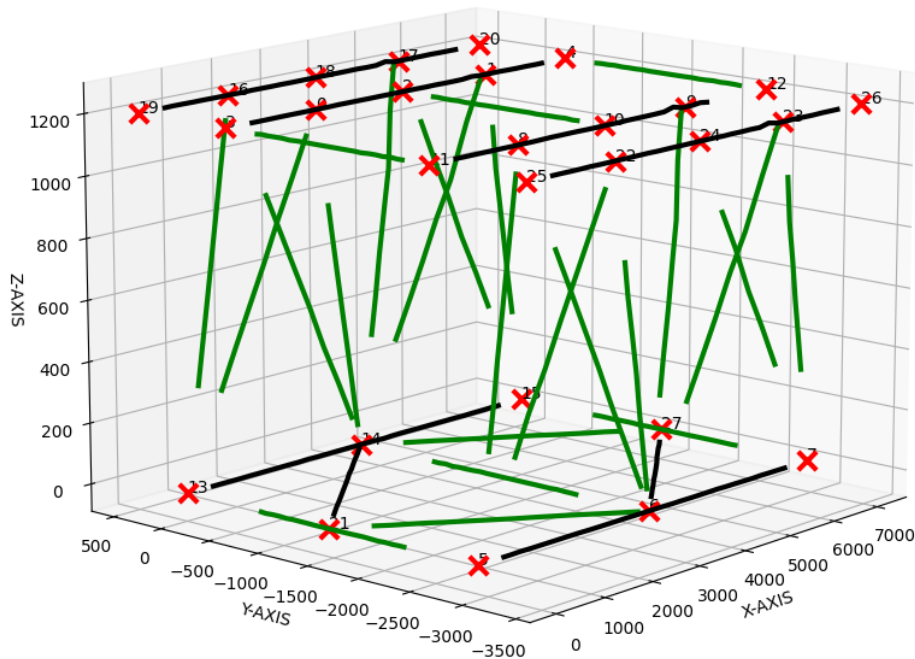


Figure 47: Resulting Intersection Points marked in red.

Intersection points can be seen as marked in red on Figure 47. The main elements are marked in black while the remaining elements are marked in green. Some intersection points were unable to be predicted. These were added manually. The resulting intersection points can be seen in Table 4.

Position X	Position Y	Position Z
1787	-450	1201
5377	-450	1201
3581	-450	1201
2	-450	1201

7160	-450	1201
2	-2899	1
3581	-2899	1
7160	-2899	1
1787	-2449	1201
5350	-2449	1201
3581	-2449	1201
2	-2449	1201
7160	-2449	1201
2	0	1
3581	0	1
7160	0	1
1787	450	1201
5350	450	1201
3581	450	1201
2	450	1201
7160	450	1201
2	-1450	1
1787	-3350	1201
5377	-3350	1201
3581	-3350	1201
2	-3350	1201
7160	-3350	1201
7160	-1450	1
3581	-1450	1
7160	-1450	1201
3581	-1450	1201
2	-1450	1201
7160	750	1201

3581	750	1201
2	750	1201
7160	-3650	1201
3581	-3650	1201
2	-3650	1201

Table 4: Extracted Intersection Points from Point Cloud.

5.3.1 Intersection Point Prediction Algorithm

Each element is returned as a set of points. Due to minor inaccuracies from the scan, sectioning algorithm or the finished product, these points will not lie in a perfect line. To work around this, the mean point of all the points in each element is found. This results in a point, for each element, that will average out all the minor errors. Another point within the element is then chosen as the initial point in order to have a line parametrized. See Figure 48.

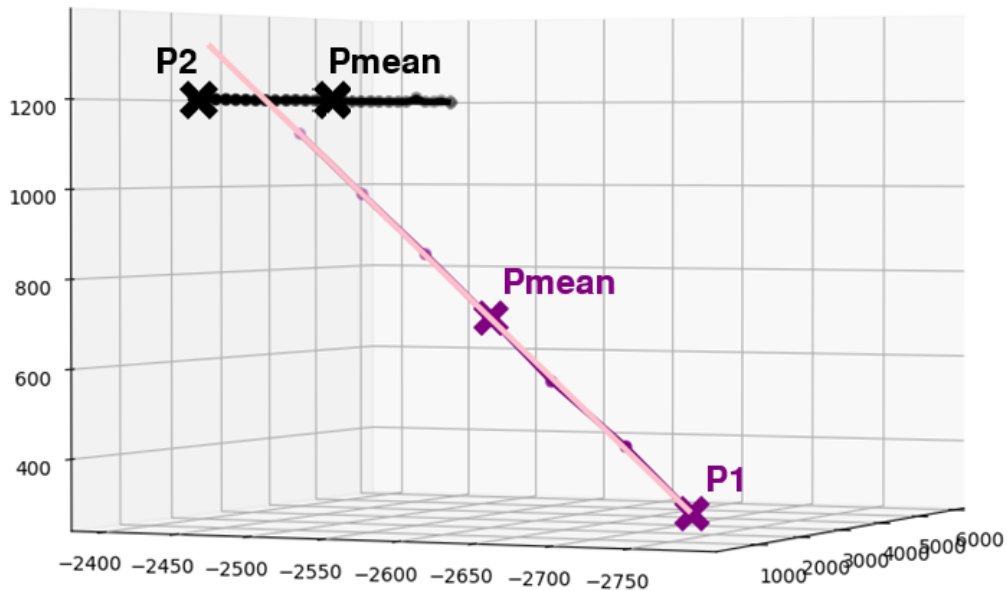


Figure 48: Cutout of two “intersecting” Elements.

P_1 and P_2 are the initial points within the Elements while P_{mean} is the mean point from all points within the respective Elements. A line is parametrized through the initial and the mean point.

As shown in Figure 49, the parametrized lines does not intersect, but rather, barely miss each other.

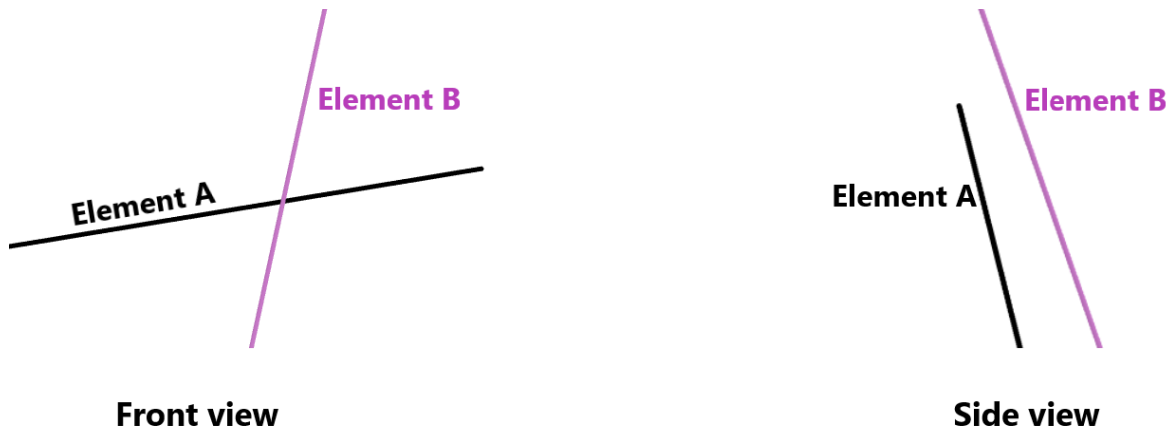


Figure 49: Zoomed in view of the intersection.

Since no intersection occur, no intersection point can be calculated. The intersection points are therefore predicted using the following equations.

$$P_1 = (a, b, c), \quad P_2 = (d, e, f)$$

$$\vec{V}_1 = (v_a, v_b, v_c), \quad \vec{V}_2 = (v_d, v_e, v_f)$$

P_1 and P_2 are the initial points in each element while V_1 and V_2 are the vectors spanning between the initial points and the mean points. Parametrizing from the initial points through their vectors will give us the following equations for Point A and Point B.

$$\begin{aligned} x &= a + v_a t & x &= d + v_d s \\ A : y &= b + v_b t & B : y &= e + v_e s \\ z &= c + v_c t & z &= f + v_f s \end{aligned} \tag{11}$$

Point A and B will be closest to each other when the vector spanning between both points are perpendicular to the parametrized lines/vector. This gives the following equations.

$$\vec{AB} \times \vec{V}_1 = 0, \quad \vec{AB} \times \vec{V}_2 = 0 \quad (12)$$

The closest points between the lines can be found by solving for variable s and t . Using Equation 13 we get the following solution.

$$\begin{bmatrix} s \\ t \end{bmatrix} = M^{-1} \times N \quad (13)$$

Where:

$$M = \begin{bmatrix} v_d v_a + v_e v_b + v_f v_c & -(v_a)^2 - (v_b)^2 - (v_c)^2 \\ (v_d)^2 + (v_e)^2 + (v_f)^2 & -v_a v_d - v_b v_e - v_c v_f \end{bmatrix}$$

$$N = \begin{bmatrix} a v_a + b v_b + c v_c - d v_a - e v_b - f v_c \\ a v_d + b v_e + c v_f - d v_d - e v_e - f v_f \end{bmatrix}$$

Some main Elements are chosen where the intersection points should be placed. In essence, intersection points will be predicted on-to the lines corresponding to the main elements. The main elements (Black in Figure 47) are separated from the secondary elements (Green in Figure 47). A loop is created through all the main Elements. Within the loop, another loop is created through all the secondary elements. The equations above are applied to find the closest points between the main Elements and the secondary Elements. If the distance, $|\vec{AB}|$ is lower than 10mm, the points will be appended to the intersection points.

5.3.2 Error Handling

Running the algorithm in section 3.3.1 will result in some intersection points being generated within very close proximity of each other. Points are therefore grouped together if the distance between them are less than 100mm. The mean point from these group of points are then chosen to the intersection points.

6 Reverse BIM Model Update and Accuracy Control

The resulting work in this section is dependent on the extracted intersection points and the designed BIM model, described in section 5 and 2. The designed BIM model is updated with the true intersection points from the physical as-built steel truss. The accuracy of the updated model is examined against the scanned 3D point cloud. Finally, a geometric deviation analysis is conducted between the designed and updated as-built model to assess where and discuss why deviations occur.

6.1 BIM Model Update

The intersection points extracted from the point cloud are randomly distributed. In order to update the BIM model, the order needs to be similar to those extracted from the DWG drawing. This is done by running a simple sorting algorithm which sorts the extracted intersection points from the point cloud by the shortest distance to original set of points. See Figure 50.

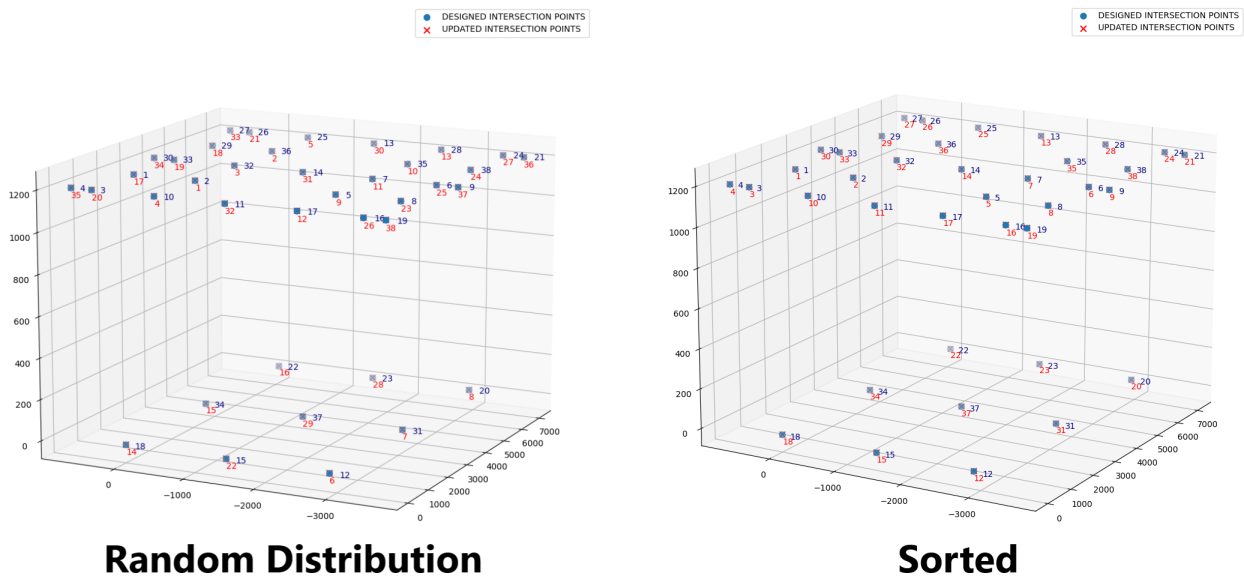


Figure 50: Random distribution vs post sorting algorithm.

Then, the points are translated and aligned so that the first intersection point from design is equal to the first intersection point extracted from the point cloud, post ordering. See Table 5.

First Intersection Point from Design	First Intersection Point from Point Cloud
[1790, 450, 1200]	[1787, 450, 1201]

Table 5: First Intersection Point from Design and Point Cloud (Post point aligning).

The intersection points from the point cloud needs to translate the *x-values* by +3, *y-values* by 0 and the *z-values* by -1 in order for both first points to be aligned. The results can be seen in Figure 51 and Table 6.

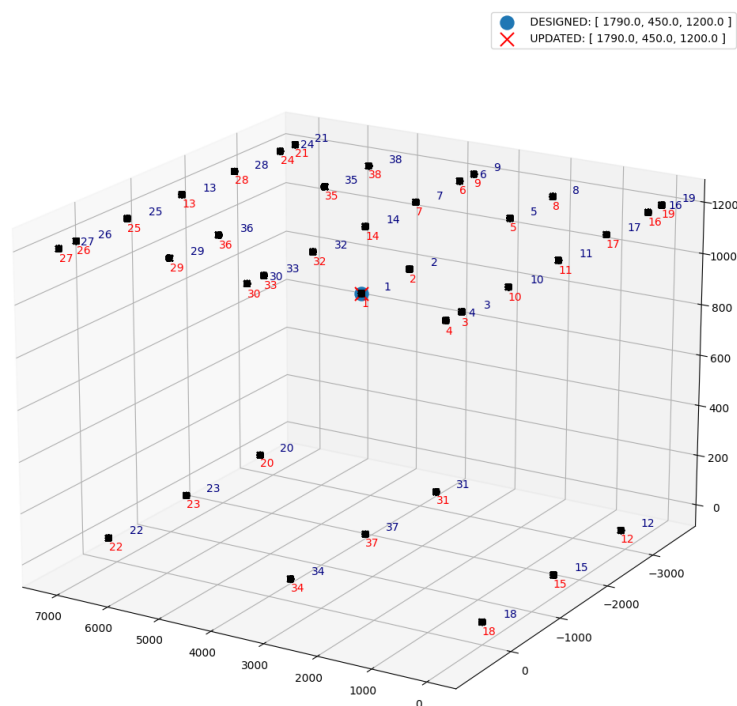


Figure 51: Aligned Intersection Points

Position X	Position Y	Position Z
1790.0	450.0	1200.0
1790.0	-450.0	1200.0
5.0	450.0	1200.0
5.0	750.0	1200.0
1790.0	-2449.0	1200.0
3584.0	-3350.0	1200.0
3584.0	-2449.0	1200.0

1790.0	-3350.0	1200.0
3584.0	-3650.0	1200.0
5.0	-450.0	1200.0
5.0	-1450.0	1200.0
5.0	-2899.0	0.0
7163.0	-1450.0	1200.0
3584.0	-1450.0	1200.0
5.0	-1450.0	0.0
5.0	-3350.0	1200.0
5.0	-2449.0	1200.0
5.0	0.0	0.0
5.0	-3650.0	1200.0
7163.0	-2899.0	0.0
7163.0	-3650.0	1200.0
7163.0	0.0	0.0
7163.0	-1450.0	0.0
7163.0	-3350.0	1200.0
7163.0	-450.0	1200.0
7163.0	450.0	1200.0
7163.0	750.0	1200.0
7163.0	-2449.0	1200.0
5353.0	450.0	1200.0
3584.0	750.0	1200.0
3584.0	-2899.0	0.0
3584.0	-450.0	1200.0
3584.0	450.0	1200.0
3584.0	0.0	0.0
5353.0	-2449.0	1200.0
5380.0	-450.0	1200.0

3584.0	-1450.0	0.0
5380.0	-3350.0	1200.0

Table 6: Intersection Points from Point Cloud post Sorting and Aligning.

These points are used to update the BIM model in CATIA. This is done by updating the Design table. The coordinates of the intersection points (nodes) are be updated to make parametric changes. The updated as-built model can be seen in Figure 52.

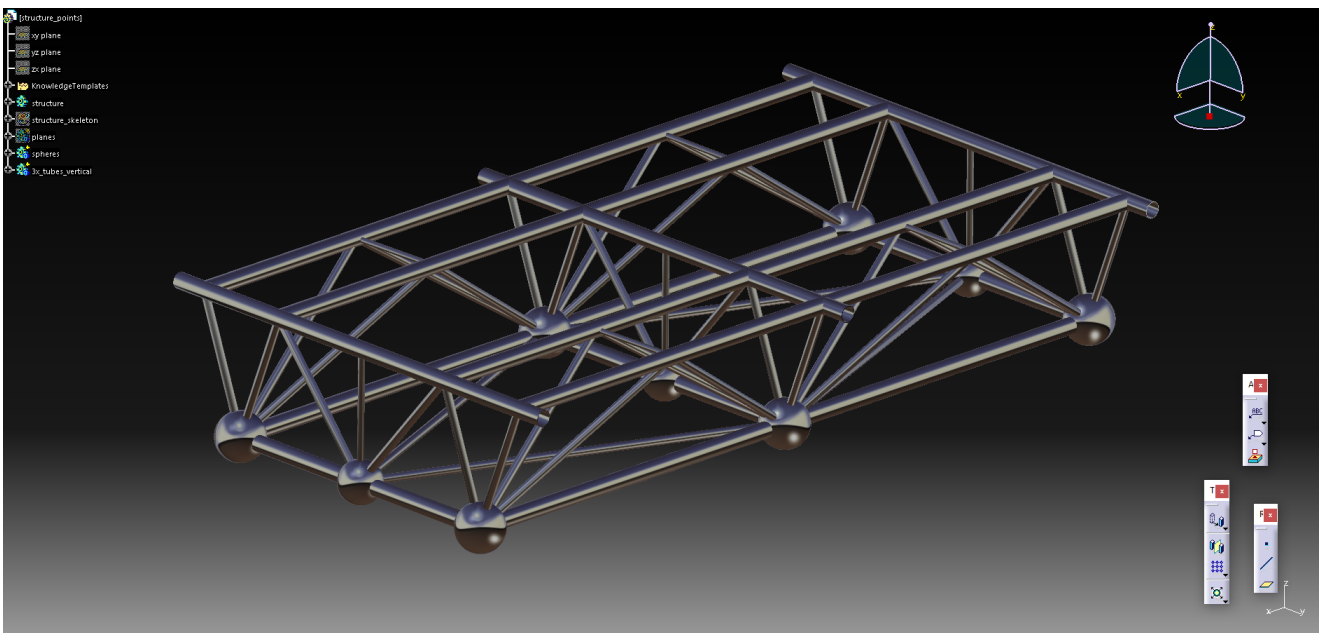


Figure 52: Updated BIM Model.

6.2 Accuracy Control

Although the updated and the designed BIM models are strikingly similar, some minor inaccuracies still occur. In this section, an accuracy control is conducted on the updated as-built model. The updated as-built model undergoes an accuracy analysis compared to the point cloud before a geometric deviation analysis is performed against the designed model.

6.2.1 Point Cloud and Updated As-Built Model Accuracy Analysis

The updated as-built BIM model is exported into Geomagic Control X to conduct an accuracy analysis between the model and the scanned 3D point cloud. The results from the analysis can be seen in Figure 53.

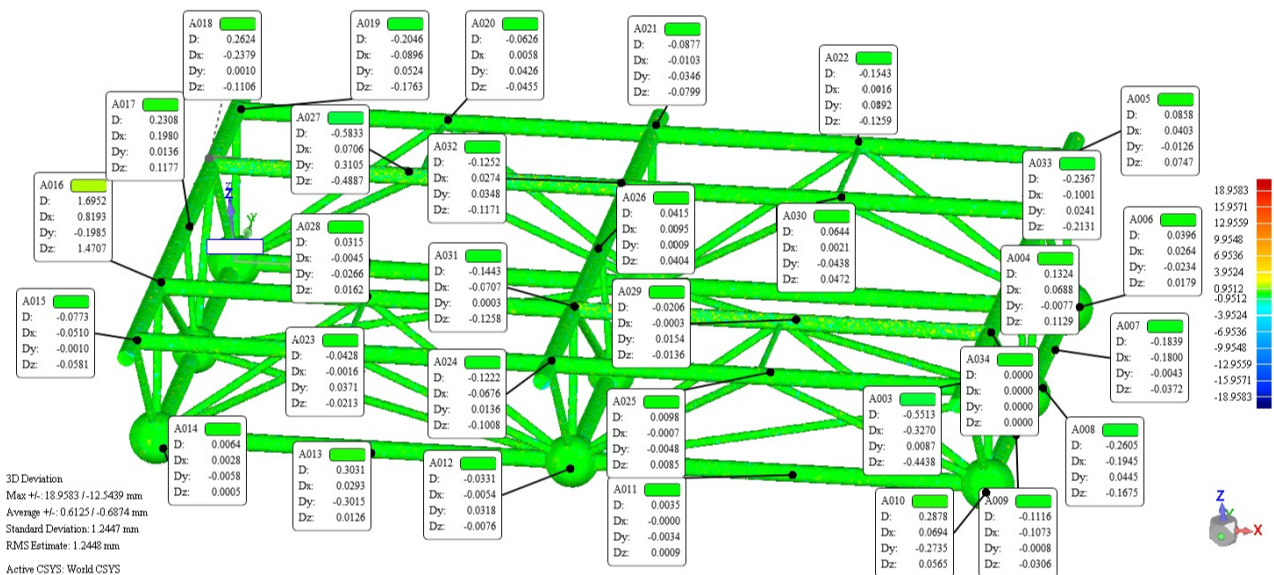


Figure 53: Accuracy analysis between Point Cloud and Updated As-Built Model.

According to the conducted accuracy analysis, the updated as-built model is very accurate in comparison to the scanned 3D point cloud. A standard deviation of 1.2mm is considered acceptable for steel truss components such as the one scanned [15].

6.2.2 Geometric Deviation Analysis

A geometric deviation analysis is conducted between the designed and updated as-built model to assess if and where inaccuracies occur. The results from the geometric deviation analysis can be

seen in Figure 54.

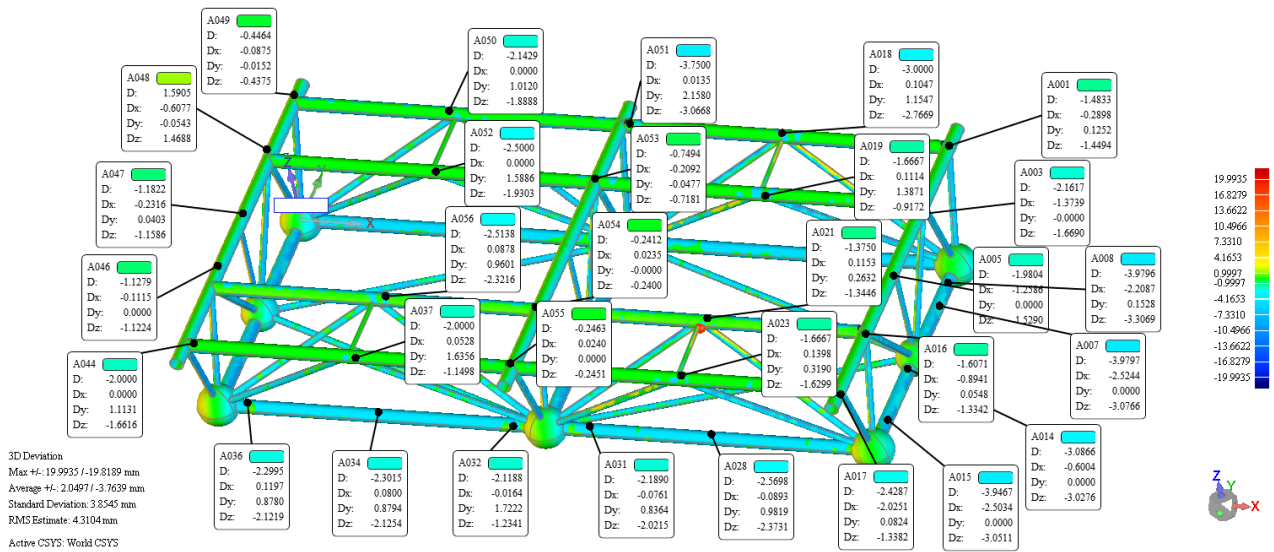


Figure 54: Geometrical Deviation Analysis in Geomagic Control X.

According to the results seen in Figure 54, there seems to be occurrences of some elements being slightly displaced. The standard deviation comes out to 3.85mm, which for steel trusses, are not that bad [15]. The models are visually analysed in CATIA to get a better feel of why these inaccuracies occurs. The results from the visual analysis in CATIA can be seen in Figure 55.

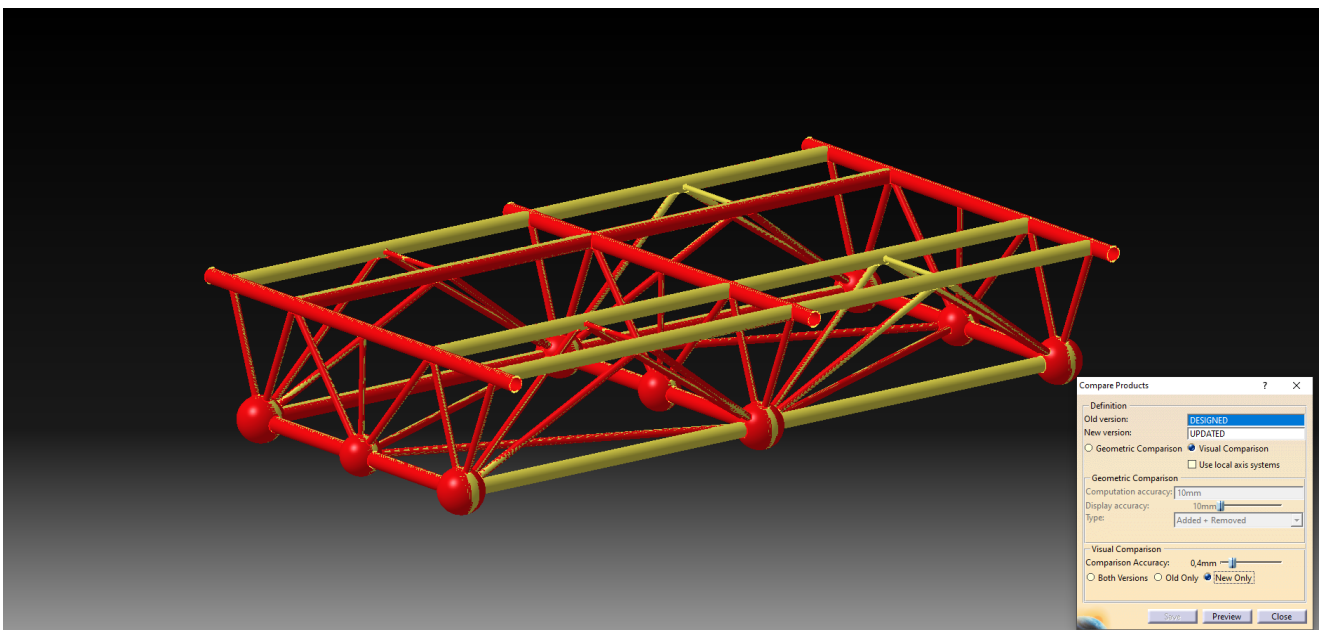


Figure 55: Visual Deviation Comparison of Updated As-Built Model vs. Designed.

The visual deviation comparison in Figure 55 shows that both the designed and updated as-built models are relatively similar. The areas marked in yellow are similar for both models while the areas marked in red are where the deviations occur. By just looking at the results in Figure 55, there seems to be only minor changes in the positioning and lengths of the elements. By conducting a more in-depth geometrical analysis, added and removed materials can be visualized in CATIA. See Figure 56.

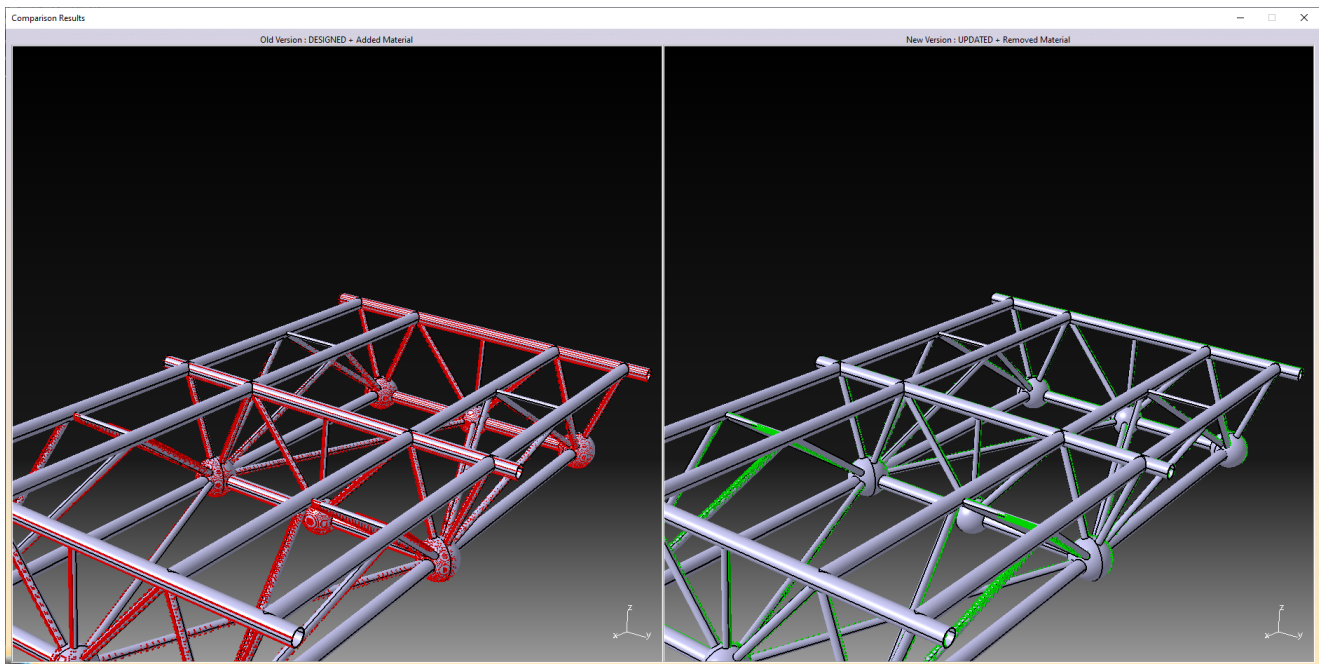


Figure 56: Material Change Comparison of Updated As-Built Model vs. Designed.

Figure 56 highlights the added and removed material to the designed and updated as-built model in red and green. It can clearly be seen that where there has been removed material in the designed model, there has been added material in the updated model. This indicates that these elements have shifted slightly from where they were intended to in the design. Most likely is this an occurrence of some elements being shorter or longer than they were supposed to. This does not pose as a major problem as these deviations are relatively small and is more or less not visible to the naked eye.

6.3 Summary

From the accuracy analysis comparing the 3D point cloud of the as-built structure to the updated as-built model, it can be seen that the updated as-built model has a high level of accuracy with a standard deviations of 1.2mm. A geometric deviation analysis between the designed and updated as-built model show that both models are strikingly similar although some inaccuracies occurs. However, these deviations can be explained by some elements being shorter or longer than they were intended to. Such inaccuracies should be expected as the elements are manufactured and assembled onto the steel truss manually.

7 Steel Truss FEA

To control if the as-built steel truss will have acceptable levels of deflection during assembly, a Finite Element Analysis, referred to as FEA in this paper, will be conducted. The essence of the FEA is to assess whether the deflections caused by lifting the steel truss during assembly is within the ramifications of the set standards. CATIA and ABAQUS are both software tools compatible with each other and would therefore allow the CATIA model to be cross used in Abaqus for a FEA. However, Midas Civil will be used due to some difficulties with attaining a software license for ABAQUS.

7.1 Introduction to FEA

The foundation for FEA dates back to the famous mathematician Euler in the 16th century. Much of the mathematical models used in present day FEA software tools were discovered during that time. However, the first mention of FEA dates back a little over a century, back to the works of Schellbach in 1851 [39].

The essence of a FEA is to conduct numerical analysis by dividing complex structures into a finite number of smaller pieces. This process is referred to as meshing. See Figure 57.

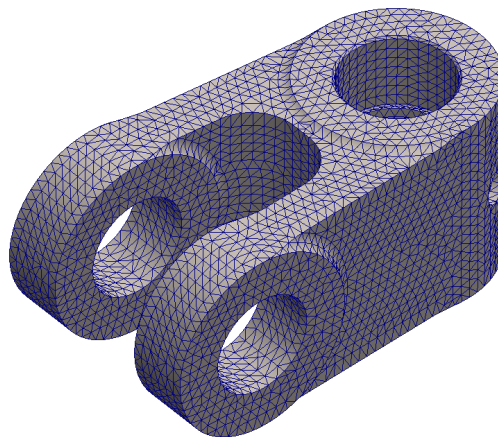


Figure 57: Element With Triangular Mesh.

In most cases, relations can be drawn between each of the sectioned piece due to the stress and strain relations being linear. Hooke's Law can be applied to all the pieces as described in Equation 14 [40].

$$[f] = [k] \times [u] \quad (14)$$

Where f is the Force Matrix, k is the Stiffness Matrix and u is the Displacement Matrix.

In order for an FEA to be conducted, a model needs to be generated. This process is usually done in the FEA software itself, or in another compatible software. For the FEA to be conducted properly, the geometry needs to be modeled in a non over-complicated way, material properties need to be assigned as well as the loads and constraints. The geometry needs to not be overly complicated due to the fact that the whole component will undergo meshing, where it will be divided into smaller pieces. Complex geometry can halt this process and cause errors due to unsuccessful meshing.

Usually, the material stiffness k and the applied forces f are known. These are then used to solve Equation 14 for the displacement u . Due to the FEA being conducted on a component that has been meshed into a finite number of pieces, large matrices are needed to be solved to calculate displacements. The process of conducting FEA is therefore left for the computer to do as it can be a time consuming and error prone task for humans.

7.2 Model Generation

Like CATIA, Midas Civil allow for the parametric design and update of nodes. A model is first created with the designed dimensions of the steel truss. The intersection points from the DWG drawing, described in Section 2, are used as a baseline to generate the model. Then the model is updated with the intersection points from the as-built model. This section will highlight the process of creating the models.

7.2.1 Designed Steel Truss Model

To create the model in Midas Civil, the intersection points (nodes) are imported into the program.

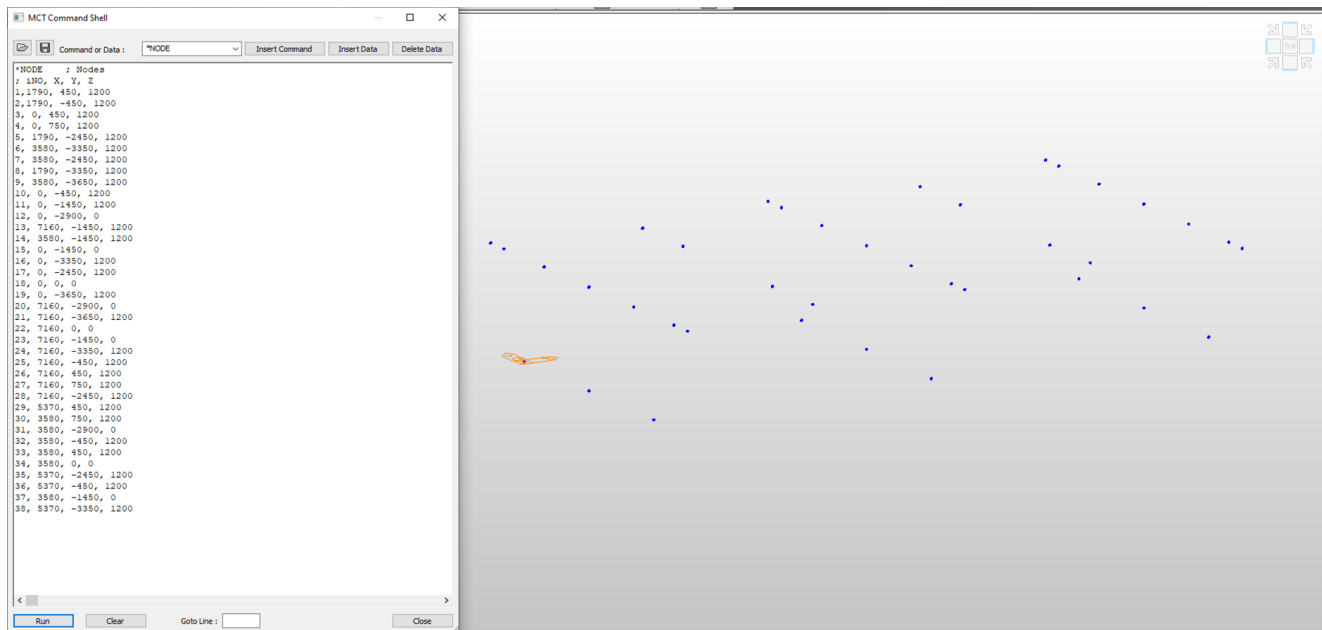


Figure 58: Designed Intersection Point Import to Midas Civil.

Elements are then created. To run a FEA, the material needs to be chosen as well as the material properties. The steel truss has steel tube profiles with a quality of S235.

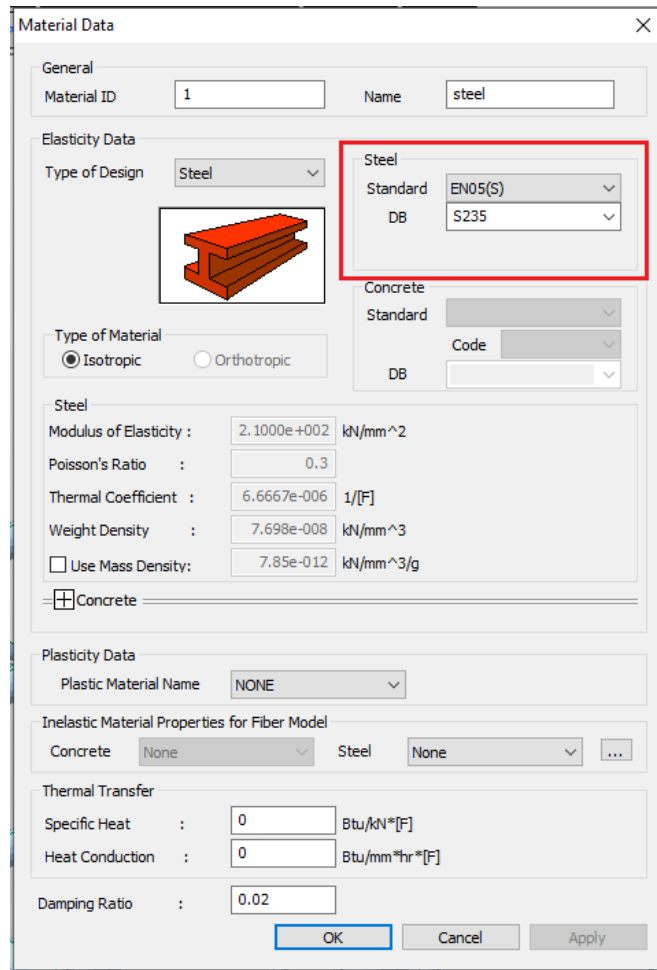


Figure 59: Material properties. Steel Quality in red.

ELEMENT PROFILES

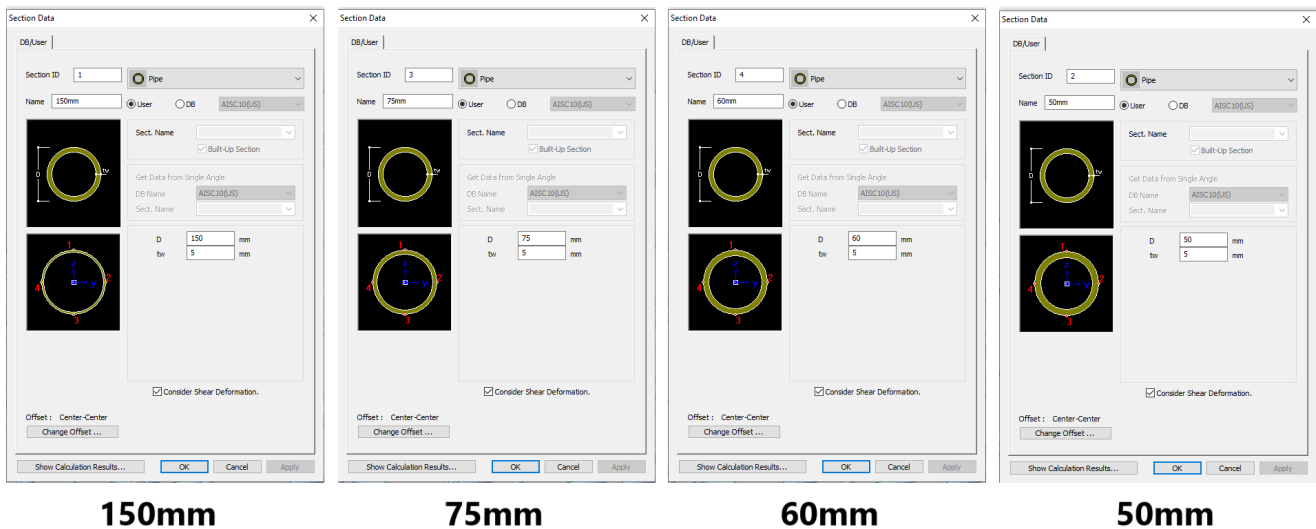


Figure 60: Element Profiles

Elements are then drawn from one intersection point to the other. The spherical elements will be neglected for the FEM analysis as they are only used for consolidation. The final model can be seen in Figure 61.

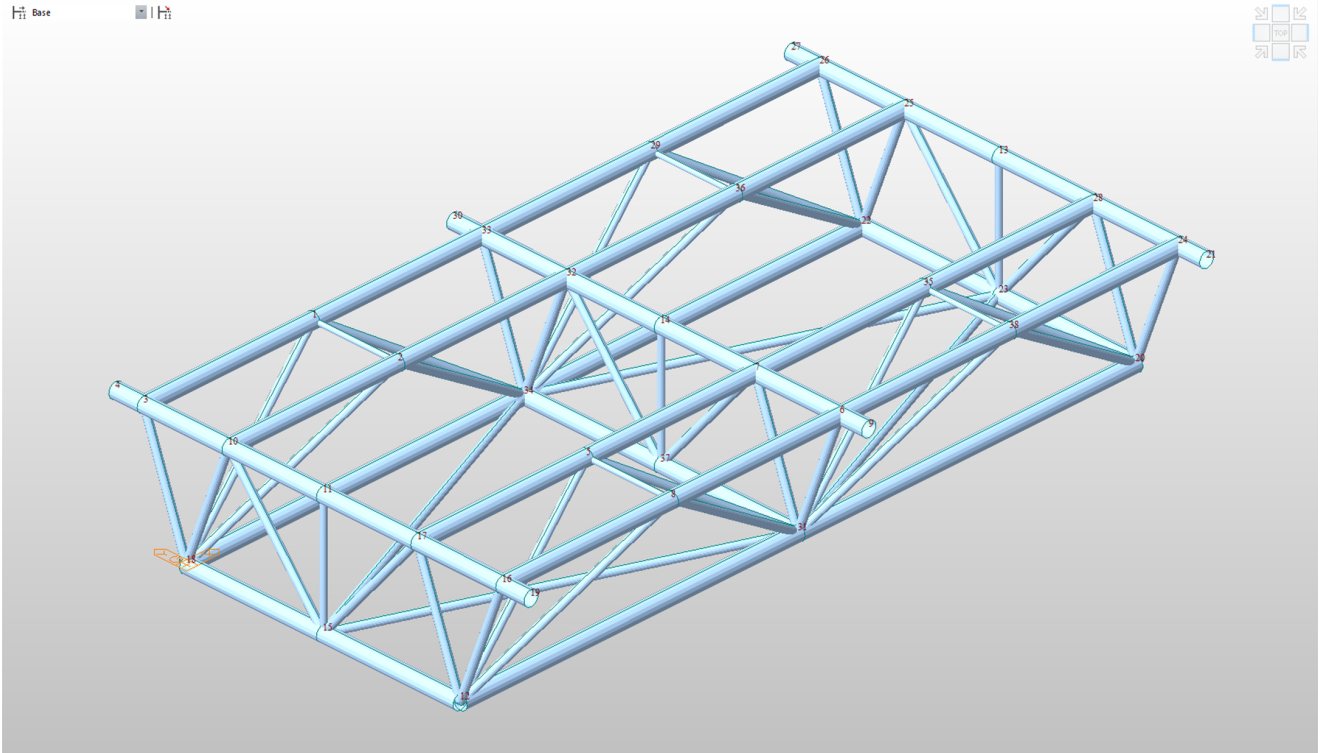
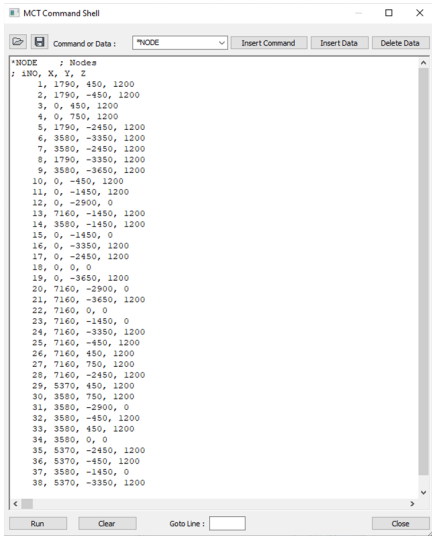


Figure 61: Designed Model Midas Civil.

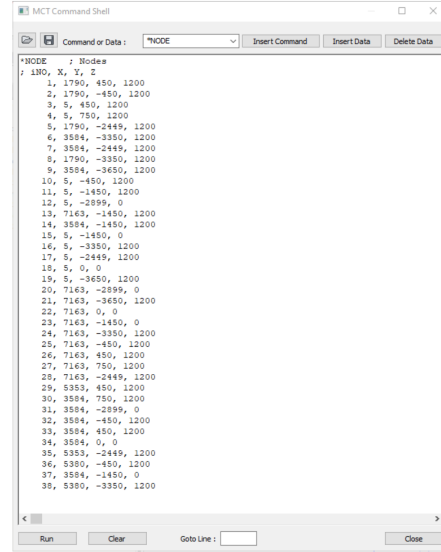
7.2.2 Updated As-Built Steel Truss Model

The model is updated using the same technique as when updating the designed model in CATIA. Extracted intersection points from the point cloud are sorted using the shortest distance to the original set of points, as described in Section 6.1. The resulting model can be seen in Figure 63.

INTERSECTION POINTS



Designed Intersection Points



Intersection Points from as-built model

Figure 62: Designed Intersection Points and Intersection Points from As-Built Model (Point Cloud).

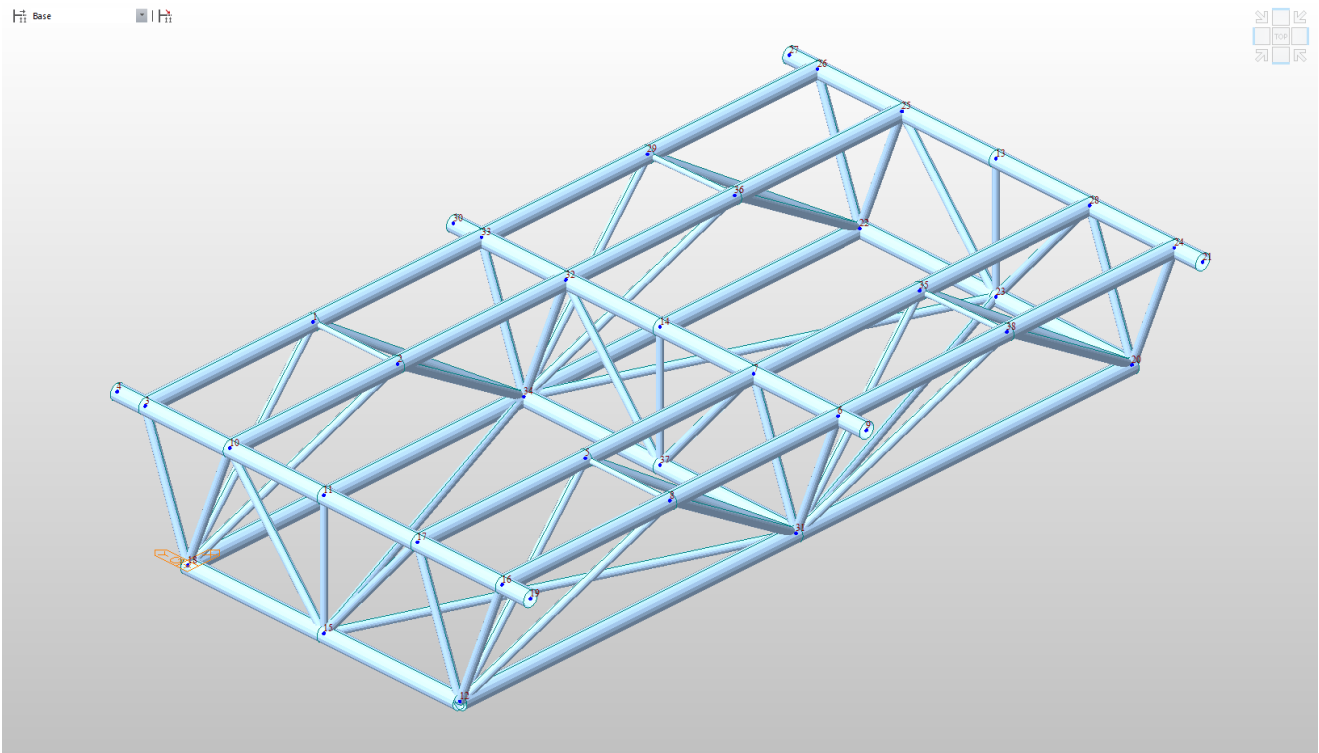


Figure 63: Updated As-Built Model in Midas Civil

7.3 Finite Element Analysis (FEA)

A FEA is conducted in this section to compare the displacements caused by the self-weight of the steel truss when lifting. The FEA is conducted for both the designed and the updated as-built model. To perform the FEA, some boundary conditions are set.

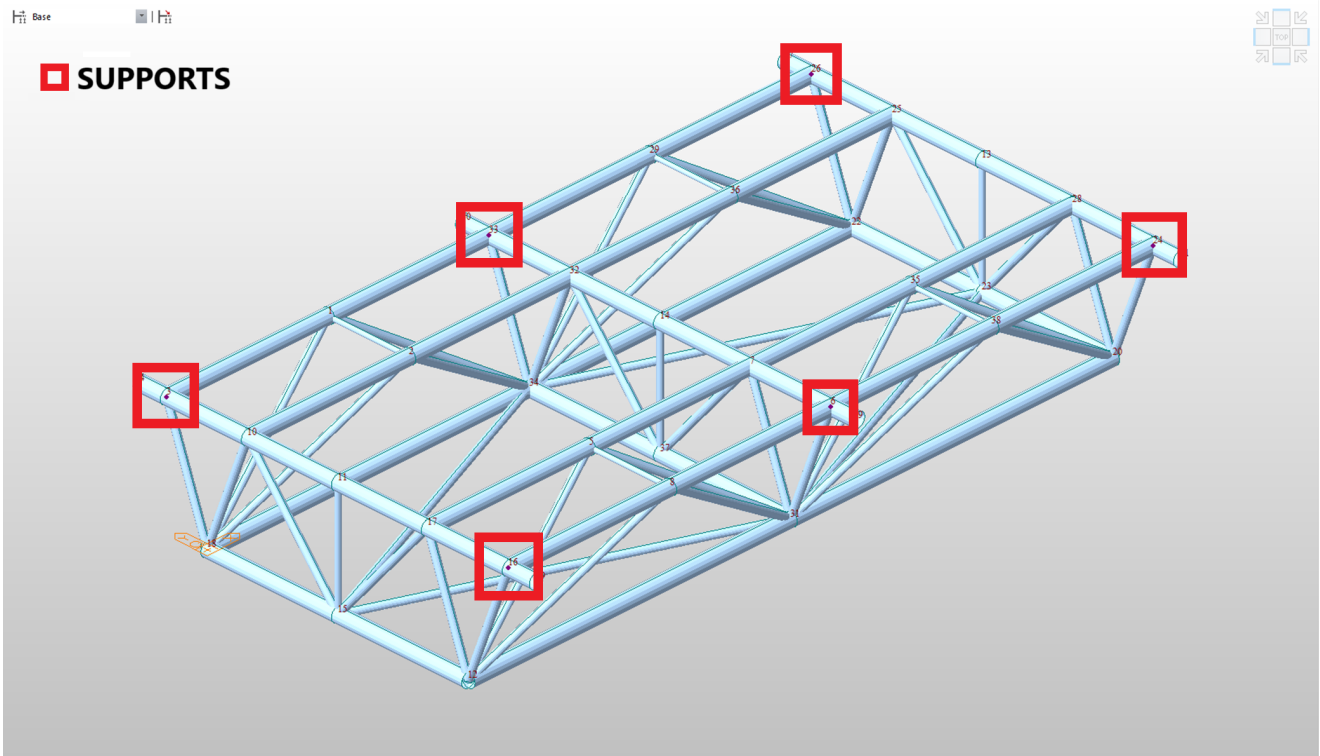


Figure 64: Fixed Truss Supports.

There are six fixed supports on which the steel truss will be mounted while lifting under the assembly process. These supports are marked with a red bounding boxes in Figure 64. In addition to the supports, a load factor of 1.04 is added to the steel truss. The support factor of 1.04 is the self-weight design standard in China, where the steel truss is located. For steel trusses in other locations/countries, load factors need to be chosen with respect to the respective standards at the location.

	$l \leq 30\text{m}$	$l/1000$, and should not be greater than 10.0
Lateral bending vector height	$30\text{m} < l \leq 60\text{m}$	$l/1000$, and should not be greater than 30.0
	$l > 60\text{m}$	$l/1000$, and should not be greater than 50.0

Figure 65: Maximum allowed deformations from self-weight according to Chinese Standards [41].

Figure 65 show a cutout of maximum allowed deformations according to Chinese Standards [41]. The steel truss has a maximum lateral span of 7160mm or 7.16m. It will therefore have to satisfy the first condition where $l \leq 30m$ and not have larger deformations than 10mm.

7.3.1 Designed Steel Truss Model

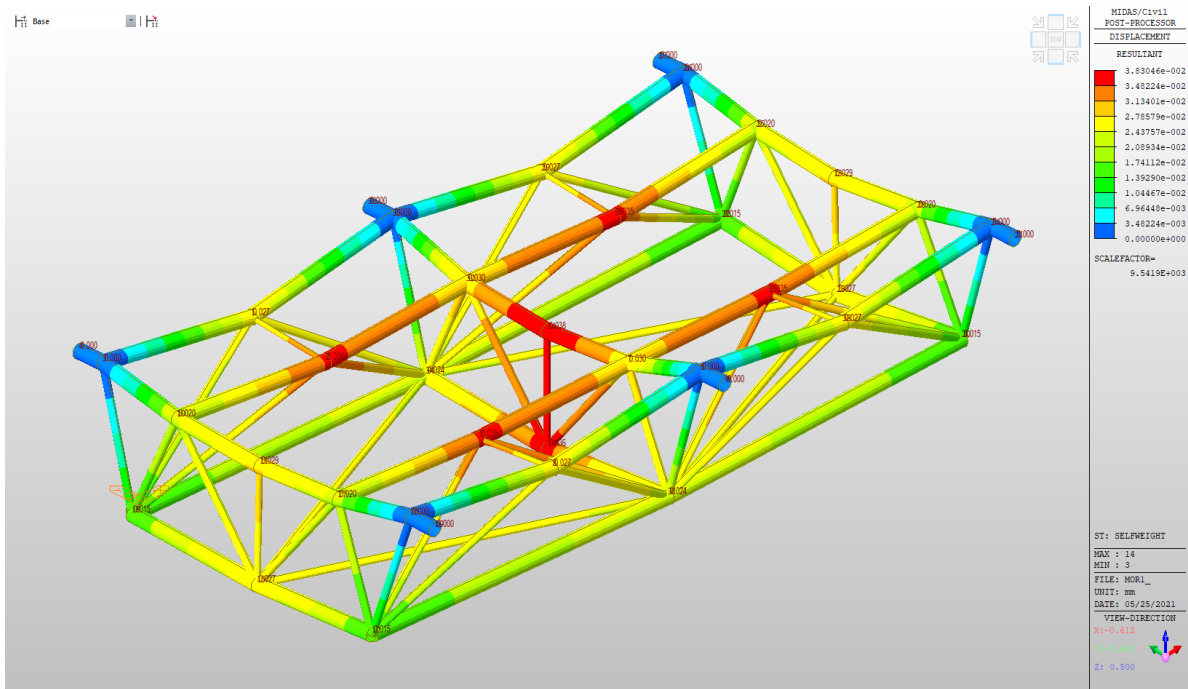


Figure 66: FEA Results for Designed Steel Truss.

Figure 66 visualizes the deformations in the steel truss when no loads are applied. A maximum deformation of 0.038mm is seen in the center of the steel truss. Chinese standards allow for a maximum deformation of 10mm from self-weight. The designed steel-truss does therefore satisfy the requirements for deformations caused by the self-weight.

7.3.2 Updated As-Built Steel Truss Model

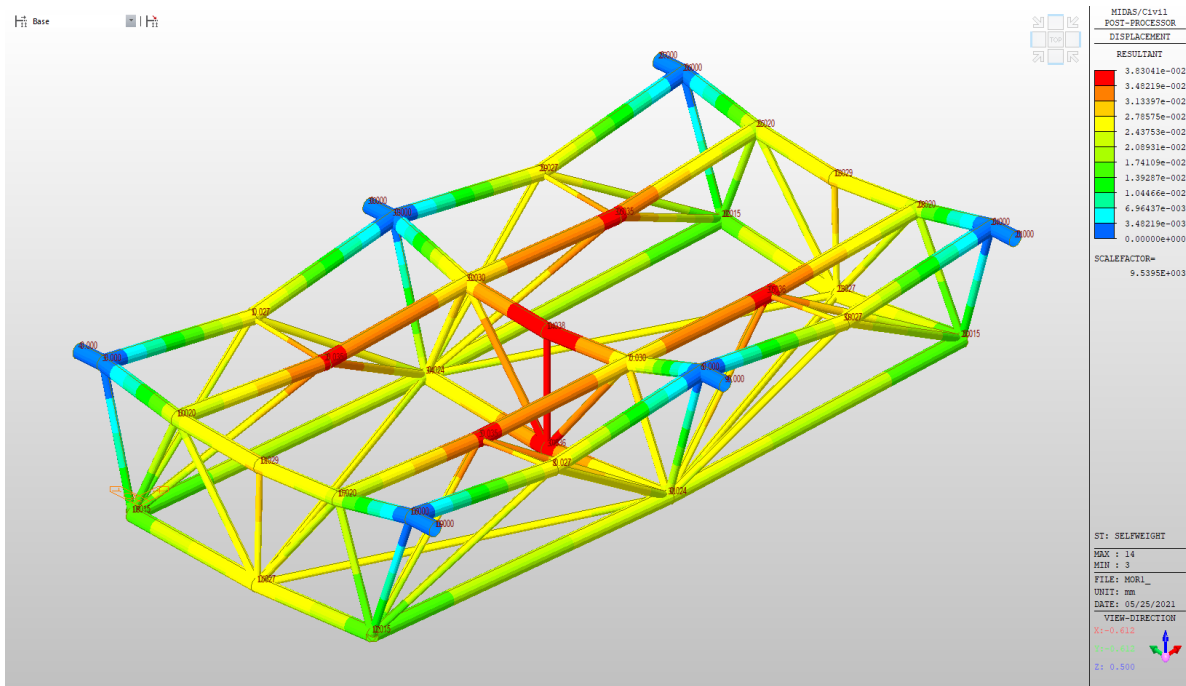


Figure 67: FEA Results for Updated As-Built Steel Truss.

Figure 67 visualizes the displacements in the as-built steel truss when no loads are applied. A maximum deformation of 0.038mm is seen in the center of the steel truss. Chinese standards allow for a maximum deformation of 10mm from self-weight. The as-built steel-truss does therefore satisfy the requirements for deformations caused by the self-weight, as predicted in Section 6.

7.4 Summary FEA

Both the designed and as-built steel truss displayed similar maximum deformation caused by self-weight. As seen in section 6, the as-built structure is pretty accurate when compared to the designed steel truss. The geometric deviation comparison displayed some minor differences between the designed and the as-built structure. However, these differences are not big and would not have been visible to the naked eye without the advanced visual analysis tool that CATIA and Geomagic Control X provided. Altogether, the FEA concludes that the as-built structure is sufficiently similar to the designed steel truss and is eligible for use.

8 Discussion and Prospect

Although the method proposed in this paper allows for reverse modeling steel trusses using point cloud data and machine learning, the scope of this method can be much greater with more computational power, data, and development. This paper focuses on using machine learning to find the intersection points for the original BIM model to be updated. The Mean Shift Algorithm described in section 5.1.1, and the Element Recognition Algorithm described in section 5.2.1 serve as the backbone for automatically identifying and extracting center points for elements within steel trusses. Once the elements are identified, and the center points are extracted, the data can be implemented in whatever way needed. Following, two additional use cases and updates for the algorithms in Section 5 will be proposed.

Deflection using Point Cloud Data

Equation 11, 12 and 13 in Section 5.3.1 propose vectorizing a whole dataset of center points within a given element to estimate the intersection points with other elements. This is an acceptable method to target the problem in this paper. Still, for other use cases where the goal might be to find the deflection within elements using point cloud data, this method becomes unusable as it assumes everything to be linear. However, the algorithm can be updated by editing these equations. Instead of vectorizing the whole dataset, vectors can be created from one point to another within the elements dataset, creating a wireframe between all points. The wireframe will pose as a true representation of the elements condition. Nevertheless, for this to work, each center point needs to be placed as accurately as possible. Therefore, the point cloud used in this paper would have needed to be a lot denser for this method to work.

Element Dimension Control using Point Cloud Data

This paper assumes that the elements have the same diameter as they were intended to in the design. However, this might not always be the case. In some cases, the designed element properties such as the span or cross-sectional width/height might differ from the as-built element. Using point cloud data can be ideal for controlling such properties. Once elements have been identified

using the algorithms in Section 5.1.1 and 5.2.1, an additional algorithm can be created to measure the element's dimensions. A bounding box with a proposed width and height can be created around the element center point datasets mean point. See Figure 68.

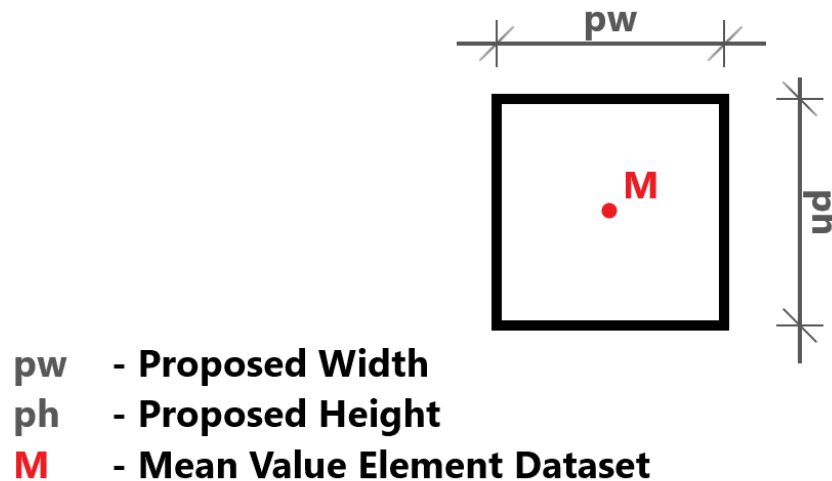


Figure 68: 2D Illustration Bounding Box Around Mean Point of Elements Dataset.

The points within the bounding box can then be filtered out. These points can then be used to find a maximum and minimum value in the sectioning axis. The difference between the max and the min value will indicate the elements length. See Equation 15.

$$Length_E = |Max_{value} - Min_{value}| \quad (15)$$

The elements' width/height or diameter can be calculated by adding on to the Sectioning & Element Center Point Extraction algorithm described in Section 5.1.1. Once a center point is predicted, the max and min value in both horizontal and vertical directions can be found within the elements cluster of points. Equation 15 can be applied respectively to find the width and height. For circular profiles, such as those described in this paper, a loop can be created to find the average distance to each point within the cluster with the center point as the base. The mean value of all these lengths will indicate what the radius is.

```
radiuses = []
for point in cluster_points:
    distance = distance(center_point, point) # Calculate Distance using
                                              Pythagoras Formula
    radiuses.append(distance)

radius = mean(cp)
```

Figure 69: Example Code for calculating radius for circular profiles.

Figure 69, displays an example code in python that can be used to calculate an element's radius from the center point. The radius can be stored within the point as the fourth variable, $p = [x, y, z, r]$, where r is the radius.

9 Conclusion

Using machine learning in combination with point cloud data for reverse modeling turns out to be a promising method for comparing an as-built steel component to the designed model. The accuracy control conducted in section 6.2 concludes that the updated as-built model has high accuracy compared to the scanned point cloud data with a standard deviation of 1.2mm. Compared to the designed model, inaccuracies in the manufactured elements result in a standard deviation of 3.85mm. These inaccuracies are small and therefore also negligible. The final FEA confirms that the deformations caused by the steel trusses self-load during assembly are within the ramifications of the set standards, which allow 10mm of deformation. The dead-load caused a maximum deformation of 0.038mm on the steel truss, which is well within range.

The research conducted in this paper confirms that the proposed method enhances the way steel components are examined before assembly. For the steel truss described and analyzed in this paper, no major deformations or inaccuracies were found, thus it is concluded that the component is eligible for use.

References

- [1] C. Ying, Y. Zhou, D. Han, G. Qin, K. Hu, J. Guo, and T. Guo, “Applying bim and 3d laser scanning technology on virtual pre-assembly for complex steel structure in construction,” Sep. 2019.
- [2] E. Al-Taie, N. Al-Ansari, and S. Knutsson, “Progress of building materials and foundation engineering in ancient iraq,” *Advanced Materials Research*, vol. 446-449, Jan. 2012. DOI: 10.4028/scientific5/AMR.446-449.220.
- [3] *Home Insurance Building*, Accessed on: Jun. 08, 2021., Aug. 2018. [Online]. Available: <https://www.history.com/topics/landmarks/home-insurance-building>.
- [4] B. Roland, “The eiffel tower,” *AA Files*, no. 64, pp. 112–131, 2012, ISSN: 02616823. [Online]. Available: <http://www.jstor.org/stable/41762314>.
- [5] B. Kamin, *Column: The same people who demoted Willis Tower could strip Chicago of another skyscraper title*, Accessed on: Jun. 08, 2021., Nov. 2019. [Online]. Available: <https://www.chicagotribune.com/columns/blair-kamin/ct-biz-first-skyscraper-kamin-20191107-kl6e5af6vgvc1lgfy4ikhe52dm-story.html>.
- [6] N. P. Walsh, *Shortlist Announced for Competition to Redesign the Eiffel Tower Visitor Experience*, Accessed on: Jun. 08, 2021., May 2018. [Online]. Available: <https://www.archdaily.com/893829/shortlist-announced-for-competition-to-redesign-the-eiffel-tower-visitor-experience>.
- [7] *Burj Khalifa: The tallest building in the world*, Accessed on: Jun. 08, 2021. [Online]. Available: <https://www.guinnessworldrecords.com/records/hall-of-fame/burj-khalifa-tallest-building-in-the-world>.
- [8] *Point Cloud to BIM*, Accessed on: Jun. 08, 2021. [Online]. Available: <https://www.hitechcaddservices.com/bim/support/point-cloud-to-bim/>.
- [9] A. Petroff, *The global steel industry by the numbers*, Accessed on: Jun. 08, 2021., Mar. 2018. [Online]. Available: <https://money.cnn.com/2018/03/02/news/economy/steel-industry-statistics-us-china-canada/index.html>.
- [10] *Steel In Buildings And Infrastructure*, Accessed on: Jun. 08, 2021. [Online]. Available: <https://www.worldsteel.org/steel-by-topic/steel-markets/buildings-and-infrastructure.html>.

-
- [11] C. Eastman, "The use of computers instead of drawings in building design," *AIA Journal*, vol. 63, no. 3, pp. 46–50, 1975.
- [12] *History and versions of IFC*. Accessed on: Jun. 08, 2021. [Online]. Available: <https://app.bimsupporters.com/courses/ifc/lessons/history-and-versions-of-ifc/>.
- [13] I. Singh, *BIM adoption and implementation around the world: Initiatives by major nations*, Accessed on: Jun. 08, 2021., May 2017. [Online]. Available: <https://www.geospatialworld.net/blogs/bim-adoption-around-the-world/>.
- [14] *5.0 Modelling and Simulation - Digital Twin*. Accessed on: Jun. 08, 2021. [Online]. Available: <https://scottbrownrigg.b-cdn.net/media/4726/covid-19-the-potential-impact-on-airport-terminal-design.pdf>.
- [15] K. Hu, *Lecture and conversation*, Topic: BIM, Point Clouds, Digital Twins, Work in Field etc., 2021.
- [16] J. Virtanen, H. Hyypä, M. Kurkela, M. Vaaja, P. Alho, and J. Hyypä, "Rapid Prototyping A Tool for Presenting 3-Dimensional Digital Models Produced by Terrestrial Laser Scanning," *ISPRS International Journal of Geo-Information*, vol. 3, no. 3, pp. 871–890, 2014. DOI: 10.3390/ijgi3030871.
- [17] bimengus, *Our Experience While Working With Point Clouds and Scan To BIM*, Accessed on: Jun. 08, 2021., Nov. 2017. [Online]. Available: <https://blog.bimengus.com/point-cloud-scan-bim/>.
- [18] G. Rocha, L. Mateus, J. Fernandez, and V. Ferreira, "A Scan-to-BIM Methodology Applied to Heritage Buildings," *Heritage*, vol. 3, no. 1, pp. 47–67, 2020. DOI: 10.3390/heritage3010004.
- [19] D. Han, *Class lecture*, Topic: BIM and construction sites, Nov. 2020.
- [20] P. Ongsulee, "Artificial intelligence, machine learning and deep learning," in *2017 15th International Conference on ICT and Knowledge Engineering (ICT&KE)*, IEEE, 2017, pp. 1–6.
- [21] *Revenues from the artificial intelligence for enterprise applications market worldwide, from 2016 to 2025*, Accessed on: Jun. 08, 2021. [Online]. Available: <https://www.statista.com/statistics/607612/worldwide-artificial-intelligence-for-enterprise-applications/>.

-
- [22] R. R. Schaller, “Moore’s law: Past, present and future,” *IEEE spectrum*, vol. 34, no. 6, pp. 52–59, 1997.
- [23] *Leica ScanStation P50 Long Range 3D Terrestrial Laser Scanner*, Accessed on: Jun. 08, 2021. [Online]. Available: <https://leica-geosystems.com/products/laser-scanners/scanners/leica-scanstation-p50>.
- [24] *What is Geomagic Control X?* Accessed on: Jun. 08, 2021. [Online]. Available: <https://www.3dsystems.com/software/geomagic-control-x>.
- [25] *What is AutoCAD?* Accessed on: Jun. 08, 2021. [Online]. Available: <https://www.autodesk.com/products/autocad/overview?term=1-YEAR>.
- [26] *Discover CATIA*. Accessed on: Jun. 08, 2021. [Online]. Available: <https://www.3ds.com/products-services/catia/>.
- [27] *About Python*. Accessed on: Jun. 08, 2021. [Online]. Available: <https://www.python.org/about/>.
- [28] *About MIDASoft*, Accessed on: Jun. 08, 2021. [Online]. Available: <https://www.midasoft.com/aboutus>.
- [29] E. Blakemore, *Lasers are driving a revolution in archaeology*, Accessed on: Jun. 08, 2021., Jun. 2019. [Online]. Available: <https://www.nationalgeographic.com/culture/article/lasers-lidar-driving-revolution-archaeology>.
- [30] M. Edl, M. Mizerák, and J. Trojan, “3D LASER SCANNERS: HISTORY AND APPLICATIONS,” *Acta Simulatio*, vol. 4, no. 4, pp. 1–5, 2018. DOI: 10.22306/asim.v4i4.54.
- [31] O. Brown and C. Hugenholtz, “Quantifying the effects of terrestrial laser scanner settings and survey configuration on land surface roughness measurement,” *Geosphere*, vol. 9, no. 2, pp. 367–377, 2013. DOI: 10.1130/ges00809.1.
- [32] W. Boehler, M. Vicent, and A. Marbs, “Investigating laser scanner accuracy,” *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 34, no. 5, pp. 696–701, 2003.
- [33] L. Rui, “Research on Applied Talents Training of Urban and Rural Planning Major Based on 3D Laser Scanning Technology,” *Journal of Beijing City University*, vol. 34, p. 02, 2017.
- [34] *Drones are gathering crucial real-time data on construction sites to create 3D aerial maps*, Accessed on: Jun. 08, 2021., Jun. 2018. [Online]. Available: <https://dronedj.com/2018/07/11/drones-crucial-data-construction-sites/>.

-
- [35] Y. Li and J. Ibanez-Guzman, “Lidar for Autonomous Driving: The Principles, Challenges, and Trends for Automotive Lidar and Perception Systems,” *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 50–61, 2020. DOI: 10.1109/msp.2020.2973615.
- [36] *134 S Granados Ave, Solana Beach, CA, 92075*. Accessed on: Jun. 08, 2021. [Online]. Available: <https://my.matterport.com/show/?m=Dzs3kQQuFgx&brand=0>.
- [37] *Mean Shift*. Accessed on: Jun. 08, 2021. [Online]. Available: <https://scikit-learn.org/stable/modules/clustering.html#mean-shift>.
- [38] GeeksforGeeks, *ML / Mean-Shift Clustering*, Accessed on: Jun. 08, 2021., May 2019. [Online]. Available: <https://www.geeksforgeeks.org/ml-mean-shift-clustering/>.
- [39] E. Stein, “History of the finite element method—mathematics meets mechanics—part i: Engineering developments,” in *The History of Theoretical, Material and Computational Mechanics—Mathematics Meets Mechanics and Engineering*, Springer, 2014, pp. 399–442.
- [40] G. Nikishkov, “Introduction to the finite element method,” *University of Aizu*, pp. 1–70, 2004.
- [41] K. Lu and Z. Hou, “Gb502052001 code for construction quality acceptance of steel structure engineering,” Ph.D. dissertation, 2002.

Appendix

Python Code

A1	Sectioning & Element Center Point Extraction	1
A2	Element Recognition	13
A3	Intersection Point Prediction	26

A1: Sectioning & Element Center Point Extraction

CODE - Sectioning & Element Center Point Extraction Algorithm:

```
import numpy as np
from sklearn.cluster import MeanShift, estimate_bandwidth
import matplotlib.pyplot as plt
from math import ceil as round_up
import time

ptCloud = np.loadtxt('steel_truss_raw.txt', delimiter=' ') # LOAD PRE PROCESSED PT CLOUD
x = ptCloud[:, 0]
y = ptCloud[:, 1]
z = ptCloud[:, 2]

def name_of(variable): # Return name of the variable. F.I, x = 'Hello'. Na-
    me_of(x) >> 'x'
    for name in globals():
        if globals()[name] is variable:
            return name

def create_section(axis, section_start, section_size): # RETURNS AS NUMPY ARRAY
    section = []
    names_axis = ['x', 'y', 'z']
    names_axis.remove(name_of(axis))

    for i in range(len(ptCloud)):
        if axis[i] > section_start and axis[i] < section_start + section_size:
            section.append([eval(names_axis[0])[i], eval(names_axis[1])[i]])
    section = np.array(section)
    return section

def iterations(axis, section_size):
    return round_up((max(axis) - min(axis))/section_size)

def create_clusters(points):
    bandwidth = estimate_bandwidth(points, quantile=0.05)
    ms = MeanShift(bandwidth=bandwidth, bin_seeding=True)
    ms.fit(points)
```

```

return ms

def visualize_plot(points, clusters):
    colors = 10*['r.', 'g.', 'b.', 'c.', 'k.', 'y.', 'm.']
    labels = clusters.labels_
    for i in range(len(points)):
        plt.plot(points[i][0], points[i][1], colors[labels[i]], markersize=1)
    plt.show()

def find_center_point(section, cluster_length, labels, point_count = 10):
    # Point count; give the amount of points to take the
    # mean value from. Default 10.
    cluster_centers = []
    for i in range(cluster_length):
        points_h = []
        points_v = []
        for idx, lbl in enumerate(labels):
            if lbl == i:
                points_h.append(section[:,0][idx])
                points_v.append(section[:,1][idx])

        points_h.sort()
        points_v.sort()

        center_h = np.mean( points_h[:point_count] + points_h[-point_count:] )
        center_v = np.mean( points_v[:point_count] + points_v[-point_count:] )
        cluster_centers.append([center_h, center_v])

    cluster_centers = np.array(cluster_centers)
    return cluster_centers

def most_frequent_value(list_):
    n = np.array(list_)
    return np.bincount(n).argmax()

from sectioning_functions import *

##### SECTION X-AXIS #####
x_section = []
cluster_lengths = []

```

```

x_section_size = 200
for i in range(iterations(x, x_section_size)):
    x_section_child = []
    x_start         = min(x) + i*x_section_size
    x_point         = x_start + x_section_size/2
    X               = create_section(x, x_start, x_section_size)
    clusters        = create_clusters(X)
    cluster_centers = find_center_point(X, len(np.unique(clusters.labels_)), clusters.labels_)

    for k in range(len(cluster_centers)):
        x_section_child.append([ x_point, cluster_centers[:, 0][k], cluster_centers[:, 1][k] ])
    x_section.append(x_section_child)
    cluster_lengths.append(len(np.unique(clusters.labels_)))

    print(str(round_up((i+1)*100/iterations(x, x_section_size))+'%')          # SHOW PERCENT DONE

x_val_to_remove = []
for i in range(len(cluster_lengths)):
    if cluster_lengths[i] != most_frequent_value(cluster_lengths):
        x_val_to_remove.append(i)

for value in sorted(x_val_to_remove, reverse = True):
    del x_section[value]

print(' ')
print(len(x_section))
# ##### END SECTION X-AXIS #####

##### SECTION Y-AXIS #####
y_section = []
cluster_lengths = []
y_section_size = 70
for i in range(iterations(y, y_section_size)):
    y_section_child = []
    y_start         = min(y) + i*y_section_size
    y_point         = y_start + y_section_size/2
    Y               = create_section(y, y_start, y_section_size)

```

```

clusters      = create_clusters(Y)
cluster_centers = find_center_point(Y, len(np.unique(clusters.labels_)), clusters.labels_)

for k in range(len(cluster_centers)):
    y_section_child.append([ cluster_centers[:, 0][k], y_point, cluster_centers[:, 1][k] ])
y_section.append(y_section_child)
cluster_lengths.append(len(np.unique(clusters.labels_)))

print(str(round_up((i+1)*100/iterations(y, y_section_size)))+'%')           # SHOW PERCENT DONE

y_val_to_remove = []
for i in range(len(cluster_lengths)):
    if cluster_lengths[i] != most_frequent_value(cluster_lengths):
        y_val_to_remove.append(i)

for value in sorted(y_val_to_remove, reverse = True):
    del y_section[value]

# ##### END SECTION Y-AXIS #####

# STORE CENTER POINTS AS VARIABLE IN SEPARATE FILE "section_var.py" #
f = open('section_var.py', 'w')
f.close()

f = open('section_var.py', 'a')
f.write( 'x_section = ' + str(x_section) + '\n')
f.write( 'y_section = ' + str(y_section) + '\n')
f.write( 'section_size_x = ' + str(x_section_size) + '\n')
f.write( 'section_size_y = ' + str(y_section_size) + '\n')
f.close()

print(' ')
print('_____')
print(str(time.perf_counter()) + ' sec')
print('*****')

```

Execution time >> 124.7750629 sec

RESULTS STORED IN SEPARATE FILE - "section_var.py":

```
x_section = [[[-124.40375689999999, -1254.0243507249997, 195.55355967999998], [-124.40375689999999, -243.944537325, 187.59875096000002], [-124.40375689999999, -2795.09877032, 186.70237385000001], [-124.40375689999999, -2537.943212225, 937.9742734700001], [-124.40375689999999, -492.29373801, 944.6425677849999], [-124.40375689999999, -1334.5928661200003, 934.0057979050001], [-124.40375689999999, 369.55043796, 936.595588715], [-124.40375689999999, -3271.6746362800004, 940.34626071], [-124.40375689999999, -2091.3503637400004, 0.15857448499999763], [-124.40375689999999, -1951.34723317, 1199.74181209], [-124.40375689999999, -929.955413555, 442.238964685], [-124.40375689999999, -1960.1105959700003, 610.663497855]], [[75.59624310000001, -1501.391904225, 212.48640950499998], [75.59624310000001, -484.88717437, 1068.88483078], [75.59624310000001, -2412.9311995850003, 1088.695187885], [75.59624310000001, -3270.9482381300004, 976.14884979], [75.59624310000001, 374.78000164499997, 981.40794809], [75.59624310000001, -1443.0154913350002, 952.70903858], [75.59624310000001, -168.24750171, 223.98808898000001], [75.59624310000001, -2688.4665869900005, 219.532148345], [75.59624310000001, -2831.3464536099996, 685.7239579550001], [75.59624310000001, -895.605951325, 401.42877822500003], [75.59624310000001, -62.61319995500001, 651.24665265], [75.59624310000001, -2002.4885228299997, 543.538311945]], [[475.59624310000004, -450.1154963299999, 1200.7095528650002], [475.59624310000004, -2900.047414405, 0.11599522500000177], [475.59624310000004, -2449.6841219899998, 1200.4407518300002], [475.59624310000004, 0.5815553050000049, 0.44979917499999617], [475.59624310000004, -2778.626737025, 314.22804523], [475.59624310000004, -3020.86955062, 314.994959515], [475.59624310000004, 450.557353485, 1199.50751447], [475.59624310000004, 119.484658685, 314.8230536749999], [475.59624310000004, -1257.7435830050001, 0.37879012999999945], [475.59624310000004, -1642.04242438, 0.630338470000001], [475.59624310000004, -119.03224072500002, 316.85270916999997], [475.59624310000004, -3349.57342986, 1200.725028375]], [[675.5962431, -449.4167767, 1200.657215615], [675.5962431, -3349.751587645, 1200.964253545], [675.5962431, 0.05055082000000084, 0.5509323399999978], [675.5962431, -2449.617212795, 1200.0383059649998], [675.5962431, -2899.5724453699995, 0.1792574000000009], [675.5962431, -2730.976328215, 449.30935809500005], [675.5962431, 450.37652425, 1200.64294239], [675.5962431, -168.45638171, 453.108263365], [675.5962431, 169.94304372999997, 450.08241444999993], [675.5962431, -1174.7410118049997, 0.3265863650000143], [675.5962431, -3066.57585048, 455.251089775], [675.5962431, -1725.85189566, 0.08178844500000046]], [[875.5962431, -449.51900313999994, 1200.7240803], [875.5962431, -2449.6078522550006, 1200.0514930049999], [875.5962431, 0.09888918999999916, 0.25936175499999975], [875.5962431, -2899.4666651, 1.1340033650000023], [875.5962431, 449.7910970049999, 1200.5528404200002], [875.5962431, -3348.9491995099997, 1200.6277347900002], [875.5962431, -1093.4029436749997, 0.7816977850000008], [875.5962431, -2676.297189625, 584.088313935], [875.5962431, 220.68092701999998, 581.263234365], [875.5962431, -218.97985170999996, 586.9678600700001], [875.5962431, -3119.251670715, 580.66308822], [875.5962431, -1801.6644661950002, 0.14608664999999982]], [[1075.5962431, -449.66571265999994, 1201.03569277], [1075.5962431, 0.19593498499999988, 0.06985569499999685], [1075.5962431, -2449.669083465, 1200.12103119], [1075.5962431, -2899.6589619650003, 0.3063644399999973], [1075.5962431, 450.51643931999996, 1199.974902335], [1075.5962431, -
```


1014.4711170900002, 0.5855555750000008], [1075.5962431, -2628.8495596350003, 720.569379915], [1075.5962431, -
3169.853417845, 721.87155102], [1075.5962431, -
3349.556639645, 1200.860055495], [1075.5962431, 269.46060639499996, 723.593010065], [1075.5962431, -
1885.72167639, 0.7136560849999999], [1075.5962431, -271.32247670499993, 721.7940709600001]], [[1275.5962431, -
449.7269940950001, 1201.04419521], [1275.5962431, -3349.3375988300004, 1201.3668414299998], [1275.5962431, -
2449.4090157500004, 1199.8241997050002], [1275.5962431, -
2899.7269712999996, 0.6577587999999984], [1275.5962431, 450.5607253499999, 1200.22469823], [1275.5962431, 0.382352975000002, 0.3800998649999994],
[1275.5962431, -319.39417906, 860.64462892], [1275.5962431, -
3219.78597832, 856.080404675], [1275.5962431, 321.43438897, 856.73636882], [1275.5962431, -2579.7242670850005, 856.432586545], [1275.5962431, -
934.4001825899999, 0.922195180000001], [1275.5962431, -1968.30792817, -0.02371491499999969]], [[2275.5962431, -
449.77906978499993, 1201.633296105], [2275.5962431, -2449.8971697650004, 1200.086319975], [2275.5962431, -
2899.3003022549997, 0.2512086399999987], [2275.5962431, 450.42630780999997, 1200.49823893], [2275.5962431, 0.4843996950000033, -
0.043952435000000636], [2275.5962431, -2571.34805768, 871.474498865], [2275.5962431, -
327.770874285, 876.453241265], [2275.5962431, 328.0602993049999, 870.305239815], [2275.5962431, -
3349.5687449800002, 1200.9875382349996], [2275.5962431, -3229.54847698, 872.4778077950001], [2275.5962431, -
2372.303162485, 0.28819601500000047], [2275.5962431, -525.19991527, 0.44874509999999856]], [[2475.5962431, -
449.92771727499996, 1201.2190279000001], [2475.5962431, -
3349.13968065, 1201.2659250299998], [2475.5962431, 0.21125004500000344, 0.9938427549999972], [2475.5962431, 450.57397198000007, 1200.210926975],
[2475.5962431, -2899.4561447249994, 0.2587605299999979], [2475.5962431, -
2449.67094486, 1200.6011390950002], [2475.5962431, 275.8624595950001, 739.689636935], [2475.5962431, -
275.961407605, 741.880489605], [2475.5962431, -2621.1421370549997, 741.75772277], [2475.5962431, -
449.55553539500005, 0.7481598300000009], [2475.5962431, -3173.959596555, 743.11350237], [2475.5962431, -
2454.98517854, 0.7142095399999985]], [[2675.5962431, -449.60360297, 1201.204961], [2675.5962431, -
3350.0789410049997, 1200.7324936], [2675.5962431, 450.25646939, 1200.1641795299997], [2675.5962431, 0.38494053000000006, 0.322382485], [2675.596
2431, -2449.83999315, 1200.4225641950002], [2675.5962431, -
2899.689381925, 0.257971334999997], [2675.5962431, 226.40427074, 609.1308035500001], [2675.5962431, -
227.00171607000001, 603.9327935149998], [2675.5962431, -3129.51338914, 608.991443045], [2675.5962431, -
2531.692152355, 0.5161686649999997], [2675.5962431, -2671.70899202, 609.0056755150001], [2675.5962431, -
366.633185775, 0.4333685049999996]], [[3075.5962431, -449.0876370950001, 1200.912269175], [3075.5962431, -
3349.9495980449997, 1201.0077739049998], [3075.5962431, 0.4156140199999996, 0.6089545449999981], [3075.5962431, 450.32930055500003, 1200.1269145
85], [3075.5962431, -2899.773660605, 0.5883635649999988], [3075.5962431, -3026.5015977300004, 336.83426322], [3075.5962431, -
2449.2442221349997, 1200.10536463], [3075.5962431, -
2772.597043965, 337.723150695], [3075.5962431, 129.65479561499998, 340.75444361499996], [3075.5962431, -
126.18100577499999, 338.59951565], [3075.5962431, -2698.9050726749997, 0.1951112250000005], [3075.5962431, -
207.41624145, 0.44002777500000007]], [[3875.5962431, -105.09692666000001, 19.100654499999997], [3875.5962431, -2833.26189861, -
15.512915240000003], [3875.5962431, -2922.0765636899996, 44.330607910000005], [3875.5962431, -
1.4198385649999978, 186.16029234500002], [3875.5962431, -2449.4060742600004, 1199.9199033200002], [3875.5962431, -

2824.524187965, 201.51961821], [3875.5962431, 450.621113555, 1200.782704115], [3875.5962431, -
2975.405051155, 210.744705785], [3875.5962431, 36.849634374999994, -14.269436235], [3875.5962431, -
450.1729751749999, 1200.666261195], [3875.5962431, -3350.1183413999997, 1200.82123717], [3875.5962431, -
1446.3898247599998, 5.5292859199999995]], [[4075.5962431, -2449.4133761750004, 1200.1274201], [4075.5962431, -
3349.8659353049998, 1200.122082565], [4075.5962431, 450.4439628849999, 1200.5302470349998], [4075.5962431, -
449.607285985, 1200.3426923749998], [4075.5962431, -2899.4091995549998, -0.03260557500000374], [4075.5962431, -
125.16755100999998, 335.7555840250001], [4075.5962431, -
3024.9867652600005, 332.80624565999995], [4075.5962431, 124.03637316499999, 334.64519882], [4075.5962431, -
2775.5655760299996, 335.210331185], [4075.5962431, -2697.6283350149997, -0.0808331949999996], [4075.5962431, -206.30682407999998, -
0.05176653000000116], [4075.5962431, -7.8336916599999995, 0.31041080500000007]], [[4275.5962431, -
2448.8755293350005, 1199.998211695], [4275.5962431, 450.297294865, 1200.7192129500002], [4275.5962431, 0.23332189500000383, 0.4029980249999987],
[4275.5962431, -449.69332731, 1200.82598098], [4275.5962431, -2899.676675225, 0.3033770499999967], [4275.5962431, -
3349.709192665, 1200.8007800649998], [4275.5962431, -2724.1618202, 465.3353043849999], [4275.5962431, -
3076.27010219, 465.40019941], [4275.5962431, -
175.63480046499998, 465.5607705199999], [4275.5962431, 173.89167377500002, 464.62992411000005], [4275.5962431, -279.993281335, -
0.050083800000000525], [4275.5962431, -2618.346153995, 0.476039875]], [[4475.5962431, -
2449.4631629650003, 1200.0810329150002], [4475.5962431, 450.219957325, 1201.2233271999999], [4475.5962431, -
449.8384015749999, 1200.2752404199998], [4475.5962431, -3349.3197412350005, 1200.34003924], [4475.5962431, -
2899.42913198, 0.06719040999999777], [4475.5962431, 228.39153650000003, 599.247380255], [4475.5962431, 0.1409665700000012, 0.7978229299999995],
[4475.5962431, -3123.65734729, 596.16539163], [4475.5962431, -360.435359465, 0.23741310499999885], [4475.5962431, -
223.189546515, 595.8804124499999], [4475.5962431, -2536.1809740099993, 0.7116273700000004], [4475.5962431, -
2677.7091096800004, 598.3826516]], [[4675.5962431, -2449.634897705, 1200.214252205], [4675.5962431, -
3348.9373918650003, 1200.909516995], [4675.5962431, -2899.2857301800004, 0.15285708499999515], [4675.5962431, -
449.48541165999995, 1200.1835273749998], [4675.5962431, 0.20969768999999588, 0.38404878499999595], [4675.5962431, 450.35546962499996, 1200.18558
1485], [4675.5962431, -274.24414388499997, 736.5062701749999], [4675.5962431, -
2624.977582265, 733.663933535], [4675.5962431, 274.84219033500005, 736.43119757], [4675.5962431, -
3172.2201726900003, 736.5374585699999], [4675.5962431, -2457.6048720900003, 0.42168395500000067], [4675.5962431, -
441.64403499500014, 0.08679285499999949]], [[4875.5962431, -
2448.8731742299997, 1200.840044735], [4875.5962431, 450.33959686500003, 1200.61973762], [4875.5962431, -
449.4465926, 1200.4863201849998], [4875.5962431, 0.05966564000000076, 0.3694490899999948], [4875.5962431, -
2899.75486815, 0.8297536699999967], [4875.5962431, -3349.7885417650004, 1200.2984959149999], [4875.5962431, -
324.25485655, 858.931165755], [4875.5962431, -2573.059609715, 861.0124323800001], [4875.5962431, -
3226.6439475099996, 864.63559727], [4875.5962431, 324.35933039, 862.1777552249999], [4875.5962431, -
525.440243405, 0.5679785700000014], [4875.5962431, -2374.4680930950003, 0.24186345500000073]], [[5075.5962431, -
2449.7739578, 1209.8953070750001], [5075.5962431, 402.37634699, 1045.22433199], [5075.5962431, -3298.56083448, 1043.15695465], [5075.5962431, -
2496.235953785, 1040.564351195], [5075.5962431, -
395.27073226, 1049.3059976149998], [5075.5962431, 450.627351595, 1207.1239715049999], [5075.5962431, -

604.055132065, 0.15760119499999875], [5075.5962431, -2296.9533062950004, 0.8053911450000001], [5075.5962431, -
3349.536112475, 1207.9657086050001], [5075.5962431, -2899.4323572499998, 1.1896734300000014], [5075.5962431, -
450.07506273999996, 1203.8126896750002], [5075.5962431, 0.4339956299999999, 0.2321531849999964]], [[5675.5962431, -
2448.7850994600003, 1204.49951301], [5675.5962431, 450.36282076, 1201.084498585], [5675.5962431, -
449.819146785, 1201.583200565], [5675.5962431, -3349.4622179049998, 1201.8935091950002], [5675.5962431, -
2899.83332402, 0.33687850999999397], [5675.5962431, -382.37191207000006, 1020.6920190050001], [5675.5962431, -
2504.2016509699997, 1031.905950045], [5675.5962431, -
3283.6374613450002, 1019.6251778100001], [5675.5962431, 374.60551976500005, 997.64140984], [5675.5962431, -
2054.748980105, 0.5229168500000003], [5675.5962431, -
846.3331186599999, 0.8232848449999992], [5675.5962431, 0.3061876749999996, 0.2847518099999952]], [[5875.5962431, -
2449.3771288999997, 1200.021588555], [5875.5962431, 450.57378140500003, 1200.9683927849999], [5875.5962431, -
449.6148110349999, 1200.6530689799997], [5875.5962431, -3349.5331165149996, 1200.82150602], [5875.5962431, 0.24691166500000108, -
0.14295992499999954], [5875.5962431, -2900.313388865, 0.04564662499999841], [5875.5962431, 323.205559055, 858.44890522], [5875.5962431, -
3222.960719065, 858.62350341], [5875.5962431, -2574.973435865, 858.40199737], [5875.5962431, -321.90918035, 863.69042125], [5875.5962431, -
930.636721545, 0.2611920999999981], [5875.5962431, -1968.083420775, 0.8191167350000018]], [[6075.5962431, -
2449.4790361349997, 1200.0258760049999], [6075.5962431, -2899.7877965, 1.005944074999998], [6075.5962431, -
3349.4524578650003, 1200.34191509], [6075.5962431, -
449.64176783000005, 1200.2383140299999], [6075.5962431, 0.2512742050000007, 0.42970046499999626], [6075.5962431, -
271.848481235, 730.540347595], [6075.5962431, -
2628.471222335, 729.677500825], [6075.5962431, 271.56786810999995, 726.33765731], [6075.5962431, -
3174.7239574249998, 725.1066933449999], [6075.5962431, 450.254387725, 1201.466429385], [6075.5962431, -
1887.2260896100001, 0.26709170000000027], [6075.5962431, -1010.7651992349998, 0.21505394500000108]], [[6275.5962431, -
2448.878692675, 1200.1920247199996], [6275.5962431, 450.35438278000004, 1200.88759791], [6275.5962431, -
449.89628839, 1200.49749185], [6275.5962431, -2899.30393268, 0.4674997749999939], [6275.5962431, -
3349.3269545099997, 1200.903370085], [6275.5962431, 0.5958493049999987, 0.3314350549999993], [6275.5962431, -
2675.4759493950005, 584.251270865], [6275.5962431, -
3121.97615297, 586.5315381], [6275.5962431, 224.45447140999994, 583.4917694650002], [6275.5962431, -
1805.5665758849998, 0.4015517799999991], [6275.5962431, -223.775096975, 590.01069192], [6275.5962431, -
1091.1722235000002, 0.031572014999997934]], [[6475.5962431, -2449.4716659400005, 1200.4537421], [6475.5962431, -
3349.993331575, 1200.67005528], [6475.5962431, 450.603197525, 1200.8121726250004], [6475.5962431, 0.14164005000000018, 1.0677745199999997], [647
5.5962431, -449.92497291499996, 1199.8989682449999], [6475.5962431, -
2899.0175332999997, 0.45941373499999827], [6475.5962431, 172.23416337999998, 459.14123623999996], [6475.5962431, -
170.109438065, 457.0499207749999], [6475.5962431, -3071.5604859849996, 459.45667529], [6475.5962431, -
2726.1512859550003, 460.6065804599999], [6475.5962431, -1729.2122008449999, 0.027092505000000246], [6475.5962431, -
1169.93262387, 0.8378910799999989]], [[6675.5962431, -
2449.4206631700004, 1200.358414455], [6675.5962431, 450.67820162500004, 1200.4941501950002], [6675.5962431, -
3350.390618, 1200.5303842950002], [6675.5962431, -

122.29351659500003, 324.45756139], [6675.5962431, 0.0967874750000143, 1.0624335850000002], [6675.5962431, -
450.59725233499995, 1200.3484348349998], [6675.5962431, -2900.3304470099997, 0.5438086699999992], [6675.5962431, -
2778.0431787099997, 325.940162835], [6675.5962431, -
3021.2157763349996, 325.55842979], [6675.5962431, 123.75298292000002, 328.446730635], [6675.5962431, -
1645.07755377, 0.49168135999999973], [6675.5962431, -1254.1539547399998, 0.42048835500000126]]
y_section = [[[3589.56544942, -2285.0, 1045.347820965], [0.9498447400000017, -2285.0, 1095.7416935750002], [7160.338325065, -
2285.0, 1210.515621525], [7159.118581090001, -2285.0, 1026.4175042600002], [5094.310177519999, -
2285.0, 0.5776583499999994], [7160.404436080001, -2285.0, 0.8563894799999957], [2068.88553574, -
2285.0, 0.4405273049999998], [0.10353494999999668, -2285.0, -0.485393245000003], [3580.280002935, -
2285.0, 1231.7067636099998], [3545.1452958899995, -2285.0, 3.5843898199999993], [3624.95190288, -2285.0, -
5.190855444999999], [[5270.79499193, -2215.0, 0.8227232400000002], [0.43860020000000083, -2215.0, 912.7746589000001], [1885.9816280549999, -
2215.0, 1.0982661650000012], [3579.9085967799997, -2215.0, 1201.968432965], [-0.5601482699999998, -
2215.0, 0.14392689999999816], [3580.18667991, -2215.0, 911.9729995199999], [7161.146947589999, -
2215.0, 914.5464894300001], [7159.456410795001, -2215.0, 0.34300008499999846], [3579.5579778449996, -2215.0, -1.069911870000002], [-
0.6732986799999992, -2215.0, 1200.31311551], [7160.6836469499995, -2215.0, 1199.137076085], [[3579.7678108200007, -2145.0, 1200.99953248], [-
0.005545284999998757, -2145.0, 1199.88222444], [5450.416602845001, -2145.0, 0.9285916099999998], [1709.1218033449998, -
2145.0, 0.4603623650000003], [3579.65124037, -2145.0, 835.213925], [7160.34579773, -2145.0, -0.37728619500000066], [7160.083667135002, -
2145.0, 833.7333008299998], [0.18993909500000292, -2145.0, 838.753752785], [7159.439967830001, -2145.0, 1200.50930116], [-0.2773339949999986, -
2145.0, 0.4137984750000001], [3579.82182751, -2145.0, -0.20592255500000006], [[5618.185134115, -2075.0, 0.8553813500000004], [7160.55573132, -
2075.0, -0.03594335000000086], [1541.5028769650003, -2075.0, 1.4142624699999993], [-0.2686351999999982, -
2075.0, 745.317976405], [3579.9134937549998, -2075.0, 0.15594701499999958], [0.5658307449999995, -2075.0, 1201.901162985], [3580.18871804, -
2075.0, 751.17534233], [3580.368736275, -2075.0, 1203.03304994], [2.0936866800000002, -2075.0, -0.78583655000000041], [7160.19954361, -
2075.0, 745.369086435], [7160.36636973, -2075.0, 1200.9578195850002], [[0.5272507199999996, -2005.0, 667.507169235], [7160.366246595001, -
2005.0, 671.2068980199999], [3580.5340003750002, -2005.0, 666.90795926], [-0.1169425500000024, -2005.0, 1199.8792640349998], [7161.243523075, -
2005.0, 0.20747421499999633], [7160.341719339999, -2005.0, 1200.1898328900002], [1370.9087273199998, -
2005.0, 0.2909388999999997], [5786.346197950001, -2005.0, 0.042539920000000106], [-0.36889965499999844, -2005.0, -
0.024147990000003006], [3579.8731479000003, -2005.0, 1.2230819249999982], [3579.4549611500006, -2005.0, 1201.971755755], [[7160.836442935001, -
1935.0, 581.4553533350002], [3580.323529115, -1935.0, 586.7269880199999], [7160.195450789999, -
1935.0, 0.30258303499999994], [0.6586137500000007, -1935.0, 583.3828277000001], [1206.837072995, -
1935.0, 0.41683857999999924], [0.40115018999999846, -1935.0, -1.1703766250000052], [5963.15747059, -
1935.0, 0.9055562249999991], [3580.5007958399997, -1935.0, -0.03268527999999975], [3579.6232544249992, -1935.0, 1201.9464641050001], [-
0.2688334549999979, -1935.0, 1201.23548038], [7161.3215249800005, -1935.0, 1201.4765731899997], [[3580.5312814599993, -
1865.0, 495.4790673549999], [7160.090257000001, -1865.0, 497.019633925], [6131.696048669999, -1865.0, 0.09722569500000092], [1025.933746335, -
1865.0, 0.6378976099999999], [-0.5295799549999998, -1865.0, 496.27133733], [-0.43341488500000336, -1865.0, -
0.16026278500000046], [7160.008509695001, -1865.0, 0.30132501999999517], [7159.230240410002, -
1865.0, 1200.2265174150002], [3580.3883498000005, -1865.0, 1201.457109605], [3580.9651738000002, -
1865.0, 0.07763862000000046], [0.37382459999999895, -1865.0, 1201.0034522499998], [[7160.497959764999, -

1795.0, 416.92723815499994], [6296.574287354999, -1795.0, 0.49419525499999895], [3580.3902552150003, -
1795.0, 0.4339620899999964], [7160.427252740001, -1795.0, 0.021128839999995108], [852.2510788650001, -
1795.0, 0.05058692000000029], [3580.3413156, -1795.0, 417.6915245], [3579.953170755, -1795.0, 1200.97134187], [0.39405813999999995, -
1795.0, 417.26880996], [0.07302447500000042, -1795.0, 0.6552850599999978], [1.2757426350000038, -
1795.0, 1199.3870595399999], [7160.840121855001, -1795.0, 1199.6396149050001]], [[0.4739507600000003, -
1725.0, 0.21280866499999718], [3581.3291416, -1725.0, 0.13639514999999847], [7160.586376159999, -1725.0, -
0.20699794999999738], [6484.211799305, -1725.0, 0.1677848750000024], [682.529063875, -1725.0, -0.25647317500000033], [0.2120519100000003, -
1725.0, 327.34640425500004], [3580.357367810001, -1725.0, 328.96141854499996], [7159.187392670001, -
1725.0, 1200.906371545], [0.283930644999996, -1725.0, 1200.5912812], [7160.466549394999, -1725.0, 331.57366326], [3579.857456750001, -
1725.0, 1202.0571272149998]], [[0.11879090500000374, -1235.0, -0.049296889999996554], [7160.79292663, -1235.0, -
1.4504288500000038], [3581.62689284, -1235.0, 1.6738421600000017], [0.3617070700000003, -1235.0, 256.79730508], [3580.1772053700006, -
1235.0, 256.75855919], [7160.254923545001, -1235.0, 253.400119715], [533.8452949650001, -1235.0, 0.6576849199999991], [6631.987673590001, -
1235.0, 0.4920972000000023], [0.4361809749999971, -1235.0, 1200.761694130003], [3581.1651406100004, -
1235.0, 1200.73274922], [7160.585828960002, -1235.0, 1201.25942068]], [[-0.7959362049999996, -1165.0, 0.1568548399999976], [7159.851009000001, -
1165.0, 0.37709123499999964], [3582.1985065300005, -1165.0, 0.14593772499999957], [709.766842485, -
1165.0, 0.2805960750000004], [7160.29255934, -1165.0, 345.66446058500003], [0.1637250649999981, -1165.0, 349.120867075], [3580.368461795, -
1165.0, 342.57560271], [0.09254889999999989, -1165.0, 1200.499397315], [7161.136310065, -1165.0, 1200.6726838199997], [3579.9797137749993, -
1165.0, 1201.23433694], [6460.642087335, -1165.0, 0.8325778599999982]], [[7160.450188140001, -1095.0, 426.299479435], [877.1979693199999, -
1095.0, 0.8129752200000006], [-0.002203390000002514, -1095.0, 0.44205945999999513], [3580.8152543749993, -
1095.0, 430.109003785], [7160.321547754999, -1095.0, 0.31669823999999663], [6287.015577275, -
1095.0, 0.031572014999997934], [1.1076517649999995, -1095.0, 428.21824129999993], [3579.9439745849995, -1095.0, -
0.12157506000000211], [0.3166857349999965, -1095.0, 1199.2899288599997], [3580.378341145, -1095.0, 1200.2718608849998], [7160.669506644999, -
1095.0, 1200.9059005400002]], [[1056.133464855, -1025.0, 0.5607084350000008], [3579.388134865, -
1025.0, 1201.9853619250002], [7160.037357159999, -1025.0, 516.0521877100001], [-0.06987291000000084, -
1025.0, 510.35963064500004], [6115.43261546, -1025.0, 0.3904762099999971], [7159.829139125, -1025.0, 1200.449968795], [3580.0591326099993, -
1025.0, 0.2151371349999991], [3580.078886785, -1025.0, 516.27181923], [0.14719013499999747, -1025.0, 1200.9843697699998], [7160.408380830001, -
1025.0, 0.2988221099999997], [1.2006980200000021, -1025.0, -1.035496525000002]], [[-0.08785198500000072, -
955.0, 598.23849154], [7160.728855474999, -955.0, 595.1017398249999], [3581.5102476599995, -955.0, -1.817065955000001], [-0.7342714300000012, -
955.0, 0.5836218949999961], [7160.185570510001, -955.0, -0.9046843800000012], [1219.7781153800001, -
955.0, 0.5650326750000015], [7160.6626610250005, -955.0, 1201.8939380999996], [3579.94398641, -955.0, 596.6877566400001], [3581.0965710899995, -
955.0, 1200.85966605], [5940.509578054999, -955.0, 0.2795620149999998], [-1.394411950000005, -
955.0, 1201.0267106799997]], [[0.3229601599999999, -885.0, 668.996690115], [5768.639818564999, -
885.0, 0.17158004500000068], [3580.7409679949997, -885.0, 675.407662035], [1397.2929419850002, -
885.0, 0.8608655300000005], [7158.994450724999, -885.0, 675.707753935], [0.3920664550000012, -885.0, 1200.445726745], [7159.33423404, -885.0, -
0.017393345000003536], [7160.399305839999, -885.0, 1200.68445744], [3581.3123114350005, -885.0, 0.9313024799999994], [-0.31694948500000136, -
885.0, 0.4238379099999989], [3578.0916259000005, -885.0, 1201.1503272349998]], [[1566.0935903949999, -815.0, 0.3786695650000006], [-
0.3484194849999998, -815.0, 763.1979753149999], [3580.15499019, -815.0, -0.7495191699999999], [5591.295395110001, -

```
815.0, 0.40603677999999943], [3579.286587469999, -815.0, 1202.3783775900004], [3579.928152415, -815.0, 754.9773782700001], [0.872486784999996, -  
815.0, -1.2854291800000026], [7160.237325874999, -815.0, -0.009998749999999746], [7160.695321474998, -  
815.0, 755.0402747200001], [7160.2020259599985, -815.0, 1200.95963276], [0.0467807800000017, -815.0, 1201.58946349]], [[5423.74675835, -745.0, -  
0.20233648500000073], [3580.178421899999, -745.0, 846.7638404900001], [1739.90958862, -745.0, 0.61470823999999988], [7160.899468844999, -  
745.0, 1199.492078135], [7160.285170654999, -745.0, 843.230925065], [0.30813022500000037, -745.0, 844.008813025], [3580.79054945, -745.0, -  
0.12772305000000017], [1.1926324250000042, -745.0, 1198.874328985], [3581.481864235, -745.0, 1201.0905445700002], [-0.30357406000000253, -  
745.0, 0.8567077150000003], [7161.8980673450005, -745.0, -0.9689969900000008]]]  
section_size_x = 200  
section_size_y = 70
```

Visualized results, see Figure 38.

A2: Element Recognition

```

import time
import numpy as np
import matplotlib.pyplot as plt
from section_var import x_section, y_section, section_size_x, section_size_y

class Element():
    # NAME, SLOPE, POINTS (NOT OBJECT)
    def __init__(self, name, slope, points):
        self.name = name
        self.slope = slope
        self.points = points

class Sections():
    # NAME, AXIS_VAL (3D), POINTS (LIST OF OBJECTS)
    def __init__(self, name, axis_val, points):
        self.name = name
        self.axis_val = axis_val
        self.points = points

class Point():
    # POINT, ELEMENT (OBJECT)
    def __init__(self, point, element):
        self.point = point
        self.element = element

def clean_section(array, skip_sections):
    # SKIP THE UNNECESSARY / UNWANTED SECTIONS
    section = []
    for i in range(len(array)):
        x = array[i][:, 0]
        y = array[i][:, 1]
        z = array[i][:, 2]
        if i not in skip_sections:
            section.append(array[i])
    return section

def distance(point_a, point_b):
    return np.sqrt( (point_a[0] - point_b[0])**2 + (point_a[1] - point_b[1])**2 )

def predict_point(points, lbh, ubv):
    # PREDICTION ALGORITHM TO PREDICT WHERE THE NEXT POINT SHOULD BE; lbh (lower boundary horizontal) = list with boundary hor in lower part, ubv (upper boundary vertical) = list with boundary vertical
    points = np.array(points)
    hor_val = points[:, 0]
    ver_val = points[:, 1]

```



```

avg_step_hor = (max(hor_val) - min(hor_val))/(len(hor_val) - 1)

if hor_val[0] > hor_val[-1]:
    avg_step_hor = -avg_step_hor

a, b = np.polyfit( hor_val, ver_val, 1 )

pred_ver_val = a*(hor_val[-1] + avg_step_hor) + b

point_pred = [ hor_val[-1] + avg_step_hor,  pred_ver_val]

if pred_ver_val > ubv[1] or pred_ver_val < ubv[0]:                # BOUNDARY
    return None

if pred_ver_val < 5 and ( point_pred[0] > lbh[1] or point_pred[0] < lbh[0] ):    # BOUNDARY FOR LOWER PART
    return None

return point_pred
def create_section_objects(section_a, section_size_a, axis, skip_sections=None):    # axis; x = 0, y = 1, z = 2
    sections    = []
    section     = np.array(section_a)
    if skip_sections != None:                # REMOVE SECTIONS IF THERE ARE ANY SECTIONS THAT NEEDS TO BE REMOVED
        section = np.array(clean_section(section, skip_sections))

    ax = [0, 1, 2]                          # 0 = x, 1 = y, 2 = z
    ax.remove(axis)

    count = 0
    for i in range(len(section)):            # STORE EACH SECTION AS INDIVIDUAL OBJECTS
        points = []
        count  = count + 1

        for k in range(len(section[i])):
            points.append( Point([section[i][:, ax[0]][k], section[i][:, ax[1]][k]], None) )    # APPEND POINTS TO A LIST

    new_section = Sections('SECTION_' + str(count), section[i][:, axis][0], points)        # CREATE SECTION OBJECT WITH POINT LIST
    sections.append(new_section)

```

```

# print( section[i][:, axis][0] )

if i+1 != len(section) and section[i + 1][:, axis][0] != section[i][:, axis][0] + section_size_a:
    axis_val = section[i][:, axis][0]
    while section[i + 1][:, axis][0] != axis_val + section_size_a:

        # print('GAP: ' + str( axis_val + section_size_a ))
        count = count + 1
        new_section = Sections('SECTION_' + str(count), axis_val + section_size_a, None)
        sections.append(new_section)

        axis_val = axis_val + section_size_a

return sections

def create_elements(sections, lbh, ubv):
    elements = []
    for i, section in enumerate(sections):
        L = []
        for point_a in section.points:
            section_b = sections[i + 1]
            if section_b.points == None:
                predicted_point = predict_point(point_a.element.points, lbh, ubv)

                if predicted_point != None:
                    predicted_point.append(section_b.axis_val)
                    point_a.element.points.append(predicted_point)

                    new_point = Point(predicted_point, point_a.element)
                    L.append(new_point)

            else:
                i_b = 0
                point_b = section_b.points[i_b]
                dist = distance(point_a.point, point_b.point)
                for k, point_b_ in enumerate(section_b.points):

```

```

        dist_control = distance(point_a.point, point_b.point) # CALCULATE DISTANCE => IF SENTENCE BELOW TO FIGURE OUT IF SMALL-
ER DISTANCE EXIST.
        if dist > dist_control:
            dist = dist_control
            point_b = point_b_

    if point_a.element == None:
        p_1 = point_a.point
        p_2 = point_b.point

        p_1.append(section.axis_val) # APPEND X VALUE LAST

        if 3 > len(p_2):
            p_2.append(sections[i + 1].axis_val)

        new_element = Element('ELEMENT_' + str(len(elements) + 1), None, [p_1, p_2])
        point_a.element = new_element
        point_b.element = new_element
        elements.append(new_element)
    else:
        p_2 = point_b.point

        if 3 > len(p_2): # IF LEN LARGER => 3, IT INDICATES THAT AXIS_VAL HAS BEEN APPEND-
ED EARLIER
            p_2.append(sections[i + 1].axis_val)

        point_a.element.points.append(point_b.point)
        point_b.element = point_a.element

    if section_b.points == None:
        section_b.points = L

    if i == len(sections) - 2:
        break

return elements

```

```

def remove_elements(elements, elem_del):                                     # REMOVE UNWANTED ELEMENTS (CREATED DUE TO INACCURACIES)
    elem_del.sort(reverse=True)
    for i in elem_del:
        del elements[i]
    return elements
def clean_elements(elements, axis):
    point_set = []
    order = [0, 1]
    order.insert(axis, 2)
    for element in elements:
        points = []
        for point in element.points:
            point = [point[i] for i in order]
            points.append(point)
        point_set.append(points)

    return point_set

x_sections = create_section_objects(x_section, section_size_x, 0, [0, 1, 11]) # X-SECTIONS AS OBJECT (GAPS FILLED)
y_sections = create_section_objects(y_section, section_size_y, 1, [0])       # Y-SECTIONS AS OBJECT (GAPS FILLED)

x_elements = create_elements(x_sections, [-2920, 29], [-5, 1220])           # RAW ELEMENTS (SECTIONED, X-AXIS)
y_elements = create_elements(y_sections, [-20, 7200], [-5, 1220])           # RAW ELEMENTS (SECTIONED, Y-AXIS)

elem_del_x = [23]                                                          # X_ELEMENTS TO DELETE (DUE INACCURACY)
elem_del_y = [0, 1, 2, 5, 6, 11, 12]                                       # Y_ELEMENTS TO DELETE (DUE INACCURACY)

x_elements = remove_elements(x_elements, elem_del_x)                       # X ELEMENTS TO USE
y_elements = remove_elements(y_elements, elem_del_y)                       # Y ELEMENTS TO USE

x_elements = clean_elements(x_elements, 0)                                  # CLEAN UP POINT ORDER X; [y, z, x] >> [x, y, z]
y_elements = clean_elements(y_elements, 1)                                  # CLEAN UP POINT ORDER Y; [x, z, y] >> [x, y, z]

elements = x_elements + y_elements                                          # ALL ELEMENTS TOGETHER

```

Execution time >> 0.7045238 sec

RESULTING ELEMENTS:

elements = [[[475.59624310000004, -450.1154963299999, 1200.7095528650002], [675.5962431, -449.4167767, 1200.657215615], [875.5962431, -449.51900313999994, 1200.7240803], [1075.5962431, -449.66571265999994, 1201.03569277], [1275.5962431, -449.7269940950001, 1201.04419521], [1475.5962431, -449.55231418750014, 1200.8260484778268], [1675.5962431, -449.41257026150015, 1200.8177560643455], [1875.5962431, -449.2954159167502, 1200.8108041180567], [2075.5962431, -449.1782615720002, 1200.8038521717672], [2275.5962431, -449.77906978499993, 1201.633296105], [2475.5962431, -449.92771727499996, 1201.2190279000001], [2675.5962431, -449.60360297, 1201.204961], [2875.5962431, -449.51839981018185, 1200.924563644633], [3075.5962431, -449.0876370950001, 1200.912269175], [3275.5962431, -449.0085710000013, 1200.7714746178854], [3475.5962431, -448.92950490500016, 1200.7455001025653], [3675.5962431, -448.8504388100002, 1200.7195255872443], [3875.5962431, -448.7713727150002, 1200.6935510719245], [4075.5962431, -449.607285985, 1200.3426923749998], [4275.5962431, -449.69332731, 1200.82598098], [4475.5962431, -449.8384015749999, 1200.2752404199998], [4675.5962431, -449.48541165999995, 1200.1835273749998], [4875.5962431, -449.4465926, 1200.4863201849998], [5075.5962431, -450.07506273999996, 1203.8126896750002], [5275.5962431, -450.0166225828261, 1201.3141811675819], [5475.5962431, -449.9606174322011, 1201.2735236626593], [5675.5962431, -449.819146785, 1201.583200565], [5875.5962431, -449.6148110349999, 1200.6530689799997], [6075.5962431, -449.64176783000005, 1200.2383140299999], [6275.5962431, -449.89628839, 1200.49749185], [6475.5962431, -449.92497291499996, 1199.8989682449999], [6675.5962431, -450.59725233499995, 1200.3484348349998]], [[[475.59624310000004, -2900.047414405, 0.11599522500000177], [675.5962431, -2899.5724453699995, 0.1792574000000009], [875.5962431, -2899.4666651, 1.1340033650000023], [1075.5962431, -2899.6589619650003, 0.3063644399999973], [1275.5962431, -2899.7269712999996, 0.6577587999999984], [1475.5962431, -2899.5817839737497, 0.6118365456386528], [1675.5962431, -2899.4656341127497, 0.749064108899347], [1875.5962431, -2899.368670730708, 0.8636234250170673], [2075.5962431, -2899.271707348666, 0.9781827411325139], [2275.5962431, -2899.3003022549997, 0.2512086399999987], [2475.5962431, -2899.4561447249994, 0.2587605299999979], [2675.5962431, -2899.689381925, 0.257971334999997], [2875.5962431, -2899.6188631016967, 0.47508878439657565], [3075.5962431, -2899.773660605, 0.5883635649999988], [3275.5962431, -2899.7139908314357, 0.42761927813444345], [3475.5962431, -2899.6585831845546, 0.46758986191161966], [3675.5962431, -2899.606869380799, 0.5048957401031657], [3875.5962431, -2899.558387689778, 0.539870009059212], [4075.5962431, -2899.4091995549998, -0.03260557500000374], [4275.5962431, -2899.676675225, 0.3033770499999967], [4475.5962431, -2899.42913198, 0.06719040999999777], [4675.5962431, -2899.2857301800004, 0.15285708499999515], [4875.5962431, -2899.75486815, 0.8297536699999967], [5075.5962431, -2899.4323572499998, 1.1896734300000014], [5275.5962431, -2899.398630856246, 0.5425628476309612], [5475.5962431, -2899.366309728899, 0.5519049635627198], [5675.5962431, -2899.83332402, 0.33687850999999397], [5875.5962431, -2900.313388865, 0.04564662499999841], [6075.5962431, -2899.7877965, 1.0059440749999998], [6275.5962431, -2899.30393268, 0.4674997749999939], [6475.5962431, -2899.0175332999997, 0.45941373499999827], [6675.5962431, -2900.3304470099997, 0.5438086699999992]], [[[475.59624310000004, -2449.6841219899998, 1200.4407518300002], [675.5962431, -2449.617212795, 1200.0383059649998], [875.5962431, -2449.6078522550006, 1200.0514930049999], [1075.5962431, -2449.669083465, 1200.12103119], [1275.5962431, -2449.4090157500004, 1199.8241997050002], [1475.5962431, -2449.340239190001, 1199.665380828535], [1675.5962431, -2449.271462630001, 1199.5504647877779], [1875.5962431, -2449.271462630001, 1199.5504647877779], [2075.5962431, -2449.271462630001, 1199.5504647877779], [2275.5962431, -2449.271462630001, 1199.5504647877779], [2475.5962431, -2449.271462630001, 1199.5504647877779], [2675.5962431, -2449.271462630001, 1199.5504647877779], [2875.5962431, -2449.271462630001, 1199.5504647877779], [3075.5962431, -2449.271462630001, 1199.5504647877779], [3275.5962431, -2449.271462630001, 1199.5504647877779], [3475.5962431, -2449.271462630001, 1199.5504647877779], [3675.5962431, -2449.271462630001, 1199.5504647877779], [3875.5962431, -2449.271462630001, 1199.5504647877779], [4075.5962431, -2449.271462630001, 1199.5504647877779], [4275.5962431, -2449.271462630001, 1199.5504647877779], [4475.5962431, -2449.271462630001, 1199.5504647877779], [4675.5962431, -2449.271462630001, 1199.5504647877779], [4875.5962431, -2449.271462630001, 1199.5504647877779], [5075.5962431, -2449.271462630001, 1199.5504647877779], [5275.5962431, -2449.271462630001, 1199.5504647877779], [5475.5962431, -2449.271462630001, 1199.5504647877779], [5675.5962431, -2449.271462630001, 1199.5504647877779], [5875.5962431, -2449.271462630001, 1199.5504647877779], [6075.5962431, -2449.271462630001, 1199.5504647877779], [6275.5962431, -2449.271462630001, 1199.5504647877779], [6475.5962431, -2449.271462630001, 1199.5504647877779], [6675.5962431, -2449.271462630001, 1199.5504647877779], [6875.5962431, -2449.271462630001, 1199.5504647877779], [7075.5962431, -2449.271462630001, 1199.5504647877779], [7275.5962431, -2449.271462630001, 1199.5504647877779], [7475.5962431, -2449.271462630001, 1199.5504647877779], [7675.5962431, -2449.271462630001, 1199.5504647877779], [7875.5962431, -2449.271462630001, 1199.5504647877779], [8075.5962431, -2449.271462630001, 1199.5504647877779], [8275.5962431, -2449.271462630001, 1199.5504647877779], [8475.5962431, -2449.271462630001, 1199.5504647877779], [8675.5962431, -2449.271462630001, 1199.5504647877779], [8875.5962431, -2449.271462630001, 1199.5504647877779], [9075.5962431, -2449.271462630001, 1199.5504647877779], [9275.5962431, -2449.271462630001, 1199.5504647877779], [9475.5962431, -2449.271462630001, 1199.5504647877779], [9675.5962431, -2449.271462630001, 1199.5504647877779], [9875.5962431, -2449.271462630001, 1199.5504647877779], [10075.5962431, -2449.271462630001, 1199.5504647877779], [10275.5962431, -2449.271462630001, 1199.5504647877779], [10475.5962431, -2449.271462630001, 1199.5504647877779], [10675.5962431, -2449.271462630001, 1199.5504647877779], [10875.5962431, -2449.271462630001, 1199.5504647877779], [11075.5962431, -2449.271462630001, 1199.5504647877779], [11275.5962431, -2449.271462630001, 1199.5504647877779], [11475.5962431, -2449.271462630001, 1199.5504647877779], [11675.5962431, -2449.271462630001, 1199.5504647877779], [11875.5962431, -2449.271462630001, 1199.5504647877779], [12075.5962431, -2449.271462630001, 1199.5504647877779], [12275.5962431, -2449.271462630001, 1199.5504647877779], [12475.5962431, -2449.271462630001, 1199.5504647877779], [12675.5962431, -2449.271462630001, 1199.5504647877779], [12875.5962431, -2449.271462630001, 1199.5504647877779], [13075.5962431, -2449.271462630001, 1199.5504647877779], [13275.5962431, -2449.271462630001, 1199.5504647877779], [13475.5962431, -2449.271462630001, 1199.5504647877779], [13675.5962431, -2449.271462630001, 1199.5504647877779], [13875.5962431, -2449.271462630001, 1199.5504647877779], [14075.5962431, -2449.271462630001, 1199.5504647877779], [14275.5962431, -2449.271462630001, 1199.5504647877779], [14475.5962431, -2449.271462630001, 1199.5504647877779], [14675.5962431, -2449.271462630001, 1199.5504647877779], [14875.5962431, -2449.271462630001, 1199.5504647877779], [15075.5962431, -2449.271462630001, 1199.5504647877779], [15275.5962431, -2449.271462630001, 1199.5504647877779], [15475.5962431, -2449.271462630001, 1199.5504647877779], [15675.5962431, -2449.271462630001, 1199.5504647877779], [15875.5962431, -2449.271462630001, 1199.5504647877779], [16075.5962431, -2449.271462630001, 1199.5504647877779], [16275.5962431, -2449.271462630001, 1199.5504647877779], [16475.5962431, -2449.271462630001, 1199.5504647877779], [16675.5962431, -2449.271462630001, 1199.5504647877779], [16875.5962431, -2449.271462630001, 1199.5504647877779], [17075.5962431, -2449.271462630001, 1199.5504647877779], [17275.5962431, -2449.271462630001, 1199.5504647877779], [17475.5962431, -2449.271462630001, 1199.5504647877779], [17675.5962431, -2449.271462630001, 1199.5504647877779], [17875.5962431, -2449.271462630001, 1199.5504647877779], [18075.5962431, -2449.271462630001, 1199.5504647877779], [18275.5962431, -2449.271462630001, 1199.5504647877779], [18475.5962431, -2449.271462630001, 1199.5504647877779], [18675.5962431, -2449.271462630001, 1199.5504647877779], [18875.5962431, -2449.271462630001, 1199.5504647877779], [19075.5962431, -2449.271462630001, 1199.5504647877779], [19275.5962431, -2449.271462630001, 1199.5504647877779], [19475.5962431, -2449.271462630001, 1199.5504647877779], [19675.5962431, -2449.271462630001, 1199.5504647877779], [19875.5962431, -2449.271462630001, 1199.5504647877779], [20075.5962431, -2449.271462630001, 1199.5504647877779], [20275.5962431, -2449.271462630001, 1199.5504647877779], [20475.5962431, -2449.271462630001, 1199.5504647877779], [20675.5962431, -2449.271462630001, 1199.5504647877779], [20875.5962431, -2449.271462630001, 1199.5504647877779], [21075.5962431, -2449.271462630001, 1199.5504647877779], [21275.5962431, -2449.271462630001, 1199.5504647877779], [21475.5962431, -2449.271462630001, 1199.5504647877779], [21675.5962431, -2449.271462630001, 1199.5504647877779], [21875.5962431, -2449.271462630001, 1199.5504647877779], [22075.5962431, -2449.271462630001, 1199.5504647877779], [22275.5962431, -2449.271462630001, 1199.5504647877779], [22475.5962431, -2449.271462630001, 1199.5504647877779], [22675.5962431, -2449.271462630001, 1199.5504647877779], [22875.5962431, -2449.271462630001, 1199.5504647877779], [23075.5962431, -2449.271462630001, 1199.5504647877779], [23275.5962431, -2449.271462630001, 1199.5504647877779], [23475.5962431, -2449.271462630001, 1199.5504647877779], [23675.5962431, -2449.271462630001, 1199.5504647877779], [23875.5962431, -2449.271462630001, 1199.5504647877779], [24075.5962431, -2449.271462630001, 1199.5504647877779], [24275.5962431, -2449.271462630001, 1199.5504647877779], [24475.5962431, -2449.271462630001, 1199.5504647877779], [24675.5962431, -2449.271462630001, 1199.5504647877779], [24875.5962431, -2449.271462630001, 1199.5504647877779], [25075.5962431, -2449.271462630001, 1199.5504647877779], [25275.5962431, -2449.271462630001, 1199.5504647877779], [25475.5962431, -2449.271462630001, 1199.5504647877779], [25675.5962431, -2449.271462630001, 1199.5504647877779], [25875.5962431, -2449.271462630001, 1199.5504647877779], [26075.5962431, -2449.271462630001, 1199.5504647877779], [26275.5962431, -2449.271462630001, 1199.5504647877779], [26475.5962431, -2449.271462630001, 1199.5504647877779], [26675.5962431, -2449.271462630001, 1199.5504647877779], [26875.5962431, -2449.271462630001, 1199.5504647877779], [27075.5962431, -2449.271462630001, 1199.5504647877779], [27275.5962431, -2449.271462630001, 1199.5504647877779], [27475.5962431, -2449.271462630001, 1199.5504647877779], [27675.5962431, -2449.271462630001, 1199.5504647877779], [27875.5962431, -2449.271462630001, 1199.5504647877779], [28075.5962431, -2449.271462630001, 1199.5504647877779], [28275.5962431, -2449.271462630001, 1199.5504647877779], [28475.5962431, -2449.271462630001, 1199.5504647877779], [28675.5962431, -2449.271462630001, 1199.5504647877779], [28875.5962431, -2449.271462630001, 1199.5504647877779], [29075.5962431, -2449.271462630001, 1199.5504647877779], [29275.5962431, -2449.271462630001, 1199.5504647877779], [29475.5962431, -2449.271462630001, 1199.5504647877779], [29675.5962431, -2449.271462630001, 1199.5504647877779], [29875.5962431, -2449.271462630001, 1199.5504647877779], [30075.5962431, -2449.271462630001, 1199.5504647877779], [30275.5962431, -2449.271462630001, 1199.5504647877779], [30475.5962431, -2449.271462630001, 1199.5504647877779], [30675.5962431, -2449.271462630001, 1199.5504647877779], [30875.5962431, -2449.271462630001, 1199.5504647877779], [31075.5962431, -2449.271462630001, 1199.5504647877779], [31275.5962431, -2449.271462630001, 1199.5504647877779], [31475.5962431, -2449.271462630001, 1199.5504647877779], [31675.5962431, -2449.271462630001, 1199.5504647877779], [31875.5962431, -2449.271462630001, 1199.5504647877779], [32075.5962431, -2449.271462630001, 1199.5504647877779], [32275.5962431, -2449.271462630001, 1199.5504647877779], [32475.5962431, -2449.271462630001, 1199.5504647877779], [32675.5962431, -2449.271462630001, 1199.5504647877779], [32875.5962431, -2449.271462630001, 1199.5504647877779], [33075.5962431, -2449.271462630001, 1199.5504647877779], [33275.5962431, -2449.271462630001, 1199.5504647877779], [33475.5962431, -2449.271462630001, 1199.5504647877779], [33675.5962431, -2449.271462630001, 1199.5504647877779], [33875.5962431, -2449.271462630001,

2449.2026860700016, 1199.4355487470161], [2075.5962431, -2449.133909510002, 1199.3206327062585], [2275.5962431, -
2449.8971697650004, 1200.086319975], [2475.5962431, -2449.67094486, 1200.6011390950002], [2675.5962431, -
2449.83999315, 1200.4225641950002], [2875.5962431, -2449.909380445909, 1200.5122782257313], [3075.5962431, -
2449.2442221349997, 1200.10536463], [3275.5962431, -2449.1845705245455, 1199.5681452222825], [3475.5962431, -
2449.129179743409, 1199.4980070153767], [3675.5962431, -2449.0771663632427, 1199.4321454199082], [3875.5962431, -
2449.025152983076, 1199.3662838244372], [4075.5962431, -2449.4133761750004, 1200.1274201], [4275.5962431, -
2448.8755293350005, 1199.998211695], [4475.5962431, -2449.4631629650003, 1200.0810329150002], [4675.5962431, -
2449.634897705, 1200.214252205], [4875.5962431, -2448.8731742299997, 1200.840044735], [5075.5962431, -
2449.7739578, 1209.8953070750001], [5275.5962431, -2449.81901024417, 1201.2717960912414], [5475.5962431, -
2449.862185503166, 1201.3666369379553], [5675.5962431, -2448.7850994600003, 1204.49951301], [5875.5962431, -
2449.3771288999997, 1200.021588555], [[475.59624310000004, 0.5815553050000049, 0.44979917499999617], [675.5962431, 0.05055082000000084, 0.55093
23999999978], [875.5962431, 0.09888918999999916, 0.25936175499999975], [1075.5962431, 0.19593498499999988, 0.069855694999999685], [1275.5962431, 0
.382352975000002, 0.3800998649999994], [1475.5962431, 0.249601853750001, 0.3403327317830246], [1675.5962431, 0.1434009567500002, 0.32579944523607
63], [1875.5962431, 0.05490020924999954, 0.3136883731136195], [2075.5962431, -
0.020957574321429598, 0.30330745415151356], [2275.5962431, 0.4843996950000033, -
0.043952435000000636], [2475.5962431, 0.21125004500000344, 0.9938427549999972], [2675.5962431, 0.38494053000000006, 0.322382485], [2875.5962431,
0.33016663187986967, 0.33776536539231006], [3075.5962431, 0.4156140199999996, 0.6089545449999981], [3275.5962431, 0.3692668754368123, 0.3635407
1947780103], [3475.5962431, 0.326230241199567, 0.3668237439596354], [3675.5962431, 0.28606271591147137, 0.36988790014268064], [3875.5962431, 0.2
4840566095388172, 0.37276054656428553], [4075.5962431, -
7.8336916599999995, 0.31041080500000007], [4275.5962431, 0.23332189500000383, 0.4029980249999987], [4475.5962431, 0.1409665700000012, 0.79782292
99999995], [4675.5962431, 0.209697689999999588, 0.384048784999999595], [4875.5962431, 0.05966564000000076, 0.36944908999999948], [5075.5962431, 0.4
33995629999999, 0.23215318499999964], [5275.5962431, 0.06811532717391183, 0.38371390108081976], [5475.5962431, -
0.28251996303442173, 0.3810761951298106], [5675.5962431, 0.3061876749999996, 0.28475180999999952], [5875.5962431, 0.24691166500000108, -
0.14295992499999954], [6075.5962431, 0.25127420500000007, 0.429700464999999626], [6275.5962431, 0.59584930499999987, 0.3314350549999993], [6475.596
2431, 0.14164005000000018, 1.0677745199999997], [6675.5962431, 0.09678747500000143, 1.0624335850000002], [[475.59624310000004, -
2778.626737025, 314.22804523], [675.5962431, -2730.976328215, 449.30935809500005], [875.5962431, -
2676.297189625, 584.088313935], [1075.5962431, -2628.8495596350003, 720.569379915], [1275.5962431, -
2579.7242670850005, 856.432586545], [1475.5962431, -2529.9986496000006, 988.4838715114856], [1675.5962431, -
2480.2730321150007, 1123.256900461738], [[475.59624310000004, -3020.86955062, 314.994959515], [675.5962431, -
3066.57585048, 455.251089775], [875.5962431, -3119.251670715, 580.66308822], [1075.5962431, -3169.853417845, 721.87155102], [1275.5962431, -
3219.78597832, 856.080404675], [1475.5962431, -3269.515085245, 989.9406598824498], [1675.5962431, -
3319.24419217, 1123.7125803187982], [[475.59624310000004, 450.557353485, 1199.50751447], [675.5962431, 450.37652425, 1200.64294239], [875.59624
31, 449.7910970049999, 1200.5528404200002], [1075.5962431, 450.51643931999996, 1199.974902335], [1275.5962431, 450.5607253499999, 1200.22469823]
, [1475.5962431, 450.7531324362499, 1199.8497171875645], [1675.5962431, 450.9455395224999, 1199.6876099085098], [1875.5962431, 451.1379466087499
, 1199.5255026294558], [2075.5962431, 451.3303536949999, 1199.363395350401], [2275.5962431, 450.42630780999997, 1200.49823893], [2475.5962431, 4
50.57397198000007, 1200.210926975], [2675.5962431, 450.25646939, 1200.1641795299997], [2875.5962431, 450.11653696363635, 1200.435468035982], [30
75.5962431, 450.32930055500003, 1200.126914585], [3275.5962431, 450.2108961942308, 1200.3407954331803], [3475.5962431, 450.1009492878023, 1200.4

3413395759], [3675.5962431, 449.9983321751356, 1200.5212499137056], [3875.5962431, 449.9021286320106, 1200.6029211225637], [4075.5962431, 450.44
39628849999, 1200.5302470349998], [4275.5962431, 450.297294865, 1200.7192129500002], [4475.5962431, 450.219957325, 1201.2233271999999], [4675.59
62431, 450.35546962499996, 1200.185581485], [4875.5962431, 450.33959686500003, 1200.61973762], [5075.5962431, 450.627351595, 1207.1239715049999]
, [5275.5962431, 450.69427579891305, 1200.4212771941734], [5475.5962431, 450.7584114943297, 1200.392093872587], [5675.5962431, 450.36282076, 120
1.084498585], [5875.5962431, 450.57378140500003, 1200.9683927849999], [6075.5962431, 450.254387725, 1201.466429385], [6275.5962431, 450.35438278
000004, 1200.88759791], [6475.5962431, 450.603197525, 1200.8121726250004], [6675.5962431, 450.67820162500004, 1200.4941501950002]], [[475.596243
10000004, 119.484658685, 314.8230536749999], [675.5962431, 169.94304372999997, 450.08241444999993], [875.5962431, 220.68092701999998, 581.263234
365], [1075.5962431, 269.46060639499996, 723.593010065], [1275.5962431, 321.43438897, 856.73636882], [1475.5962431, 371.92182154125, 994.3214995
183273], [1675.5962431, 422.4092541125, 1130.42956369485]], [[475.59624310000004, -1257.7435830050001, 0.37879012999999945], [675.5962431, -
1174.7410118049997, 0.32658636500000143], [875.5962431, -1093.4029436749997, 0.7816977850000008], [1075.5962431, -
1014.4711170900002, 0.5855555750000008], [1275.5962431, -934.4001825899999, 0.922195180000001], [1475.5962431, -
853.5643324862498, 1.0017768675474374], [1675.5962431, -772.7284823824997, 1.1366705464973366], [1875.5962431, -
691.8926322787496, 1.2715642254472357], [2075.5962431, -611.0567821749995, 1.4064579043971346], [2275.5962431, -
525.19991527, 0.44874509999999856], [2475.5962431, -449.55553539500005, 0.7481598300000009], [2675.5962431, -
366.633185775, 0.4333685049999996], [2875.5962431, -285.62314966318183, 0.9520340217563454], [3075.5962431, -
207.41624145, 0.44002777500000007], [3275.5962431, -126.62183056115383, 0.8492954403649345], [3475.5962431, -
45.827419672307656, 0.8593784307720851]], [[475.59624310000004, -1642.04242438, 0.630338470000001], [675.5962431, -
1725.85189566, 0.08178844500000046], [875.5962431, -1801.6644661950002, 0.14608664999999982], [1075.5962431, -
1885.72167639, 0.7136560849999999], [1275.5962431, -1968.30792817, -0.02371491499999969], [1475.5962431, -
2049.8743041175, 0.10349853757178673], [1675.5962431, -2131.440680065, 0.03491595883605392], [1875.5962431, -2213.0070560125, -
0.03366661989968045], [2075.5962431, -2294.57343196, -0.10224919863541326], [2275.5962431, -
2372.303162485, 0.28819601500000047], [2475.5962431, -2454.98517854, 0.7142095399999985], [2675.5962431, -
2531.692152355, 0.5161686649999997], [2875.5962431, -2612.5694003527274, 0.28204955287648115], [3075.5962431, -
2698.9050726749997, 0.1951112250000005], [3275.5962431, -2780.2021994669226, 0.2642596991277378], [3475.5962431, -
2861.4993262588455, 0.26571856669835736]], [[475.59624310000004, -119.03224072500002, 316.85270916999997], [675.5962431, -
168.45638171, 453.108263365], [875.5962431, -218.97985170999996, 586.9678600700001], [1075.5962431, -
271.32247670499993, 721.7940709600001], [1275.5962431, -319.39417906, 860.64462892], [1475.5962431, -
369.48466364374997, 991.9022203793656], [1675.5962431, -419.57514822749994, 1126.7793459342452]], [[475.59624310000004, -
3349.57342986, 1200.725028375], [675.5962431, -3349.751587645, 1200.964253545], [875.5962431, -
3348.9491995099997, 1200.6277347900002], [1075.5962431, -3349.556639645, 1200.860055495], [1275.5962431, -
3349.3375988300004, 1201.3668414299998], [1475.5962431, -3349.1370017962504, 1200.847740194678], [1675.5962431, -
3348.9765241692503, 1200.814722621495], [1875.5962431, -3348.842792813417, 1200.787207977176], [2075.5962431, -
3348.7129649803337, 1200.76049646567], [2275.5962431, -3349.5687449800002, 1200.9875382349996], [2475.5962431, -
3349.13968065, 1201.2659250299998], [2675.5962431, -3350.0789410049997, 1200.7324936], [2875.5962431, -
3350.203120643606, 1200.9376959130482], [3075.5962431, -3349.9495980449997, 1201.0077739049998], [3275.5962431, -
3350.064225403713, 1200.9468179346973], [3475.5962431, -3350.170665093947, 1200.9534687853234], [3675.5962431, -
3350.270008804832, 1200.959676245907], [3875.5962431, -3350.367324043863, 1200.9657569580634], [4075.5962431, -

3349.8659353049998, 1200.122082565], [4275.5962431, -3349.709192665, 1200.8007800649998], [4475.5962431, -
3349.3197412350005, 1200.34003924], [4675.5962431, -3348.9373918650003, 1200.909516995], [4875.5962431, -
3349.7885417650004, 1200.2984959149999], [5075.5962431, -3349.536112475, 1207.9657086050001], [5275.5962431, -
3349.464183820064, 1201.126602204346], [5475.5962431, -3349.395252192417, 1201.128944309674], [5675.5962431, -
3349.4622179049998, 1201.8935091950002], [5875.5962431, -3349.5331165149996, 1200.82150602], [6075.5962431, -
3349.4524578650003, 1200.34191509], [6275.5962431, -3349.3269545099997, 1200.903370085], [6475.5962431, -
3349.993331575, 1200.67005528], [6675.5962431, -3350.390618, 1200.5303842950002], [[2275.5962431, -
2571.34805768, 871.474498865], [2475.5962431, -2621.1421370549997, 741.75772277], [2675.5962431, -
2671.70899202, 609.0056755150001], [2875.5962431, -2721.8894591900003, 477.937702375908], [3075.5962431, -
2772.597043965, 337.723150695], [3275.5962431, -2822.90929053625, 207.64733002466528], [3475.5962431, -
2873.2215371075, 74.54412161041728], [[2275.5962431, -327.770874285, 876.453241265], [2475.5962431, -
275.961407605, 741.880489605], [2675.5962431, -227.00171607000001, 603.9327935149998], [2875.5962431, -
176.6171369625, 469.6234966149212], [3075.5962431, -126.18100577499999, 338.59951565], [3275.5962431, -
75.7835386475, 201.29597854411216], [3475.5962431, -
25.386071519999994, 66.1210427354787], [[2275.5962431, 328.0602993049999, 870.305239815], [2475.5962431, 275.8624595950001, 739.689636935], [26
75.5962431, 226.40427074, 609.1308035500001], [2875.5962431, 175.57625645750005, 479.7696659774713], [3075.5962431, 129.65479561499998, 340.7544
4361499996], [3275.5962431, 80.0534196925, 217.98865594463027], [3475.5962431, 30.452043770000024, 86.46504454115268], [[2275.5962431, -
3229.54847698, 872.4778077950001], [2475.5962431, -3173.959596555, 743.11350237], [2675.5962431, -
3129.51338914, 608.991443045], [2875.5962431, -3079.49584522, 484.1704739174111], [3075.5962431, -
3026.5015977300004, 336.83426322], [3275.5962431, -2975.7398779175005, 205.3356091636042], [3475.5962431, -
2924.9781581050006, 70.54580910149798], [[4075.5962431, -125.16755100999998, 335.7555840250001], [4275.5962431, -
175.63480046499998, 465.5607705199999], [4475.5962431, -223.189546515, 595.8804124499999], [4675.5962431, -
274.24414388499997, 736.5062701749999], [4875.5962431, -324.25485655, 858.931165755], [5075.5962431, -
395.27073226, 1049.3059976149998], [5275.5962431, -449.29136851, 1193.068430951123], [[4075.5962431, -
3024.9867652600005, 332.80624565999995], [4275.5962431, -3076.27010219, 465.40019941], [4475.5962431, -
3123.65734729, 596.16539163], [4675.5962431, -3172.2201726900003, 736.5374585699999], [4875.5962431, -
3226.6439475099996, 864.63559727], [5075.5962431, -3298.56083448, 1043.15695465], [5275.5962431, -
3353.2756483239996, 1194.548192360984], [[4075.5962431, 124.03637316499999, 334.64519882], [4275.5962431, 173.89167377500002, 464.6299241100000
5], [4475.5962431, 228.39153650000003, 599.247380255], [4675.5962431, 274.84219033500005, 736.43119757], [4875.5962431, 324.35933039, 862.177755
2249999], [5075.5962431, 402.37634699, 1045.22433199], [5275.5962431, 458.044341755, 1198.0919438974474], [[4075.5962431, -
2775.5655760299996, 335.210331185], [4275.5962431, -2724.1618202, 465.3353043849999], [4475.5962431, -
2677.7091096800004, 598.3826516], [4675.5962431, -2624.977582265, 733.663933535], [4875.5962431, -
2573.059609715, 861.0124323800001], [5075.5962431, -2496.235953785, 1040.564351195], [5275.5962431, -
2440.3700293360002, 1193.2128447677733], [[4075.5962431, -2697.6283350149997, -0.0808331949999996], [4275.5962431, -
2618.346153995, 0.476039875], [4475.5962431, -2536.1809740099993, 0.7116273700000004], [4675.5962431, -
2457.6048720900003, 0.42168395500000067], [4875.5962431, -2374.4680930950003, 0.24186345500000073], [5075.5962431, -
2296.9533062950004, 0.8053911450000001], [5275.5962431, -2216.8183005510004, 0.769972213744218], [5475.5962431, -

2136.6832948070005, 0.8674569190998387], [5675.5962431, -2054.748980105, 0.5229168500000003], [5875.5962431, -
1968.083420775, 0.8191167350000018], [6075.5962431, -1887.2260896100001, 0.26709170000000027], [6275.5962431, -
1805.5665758849998, 0.4015517799999991], [6475.5962431, -1729.2122008449999, 0.027092505000000246], [6675.5962431, -
1645.07755377, 0.49168135999999973]], [[4075.5962431, -206.30682407999998, -0.05176653000000116], [4275.5962431, -279.993281335, -
0.050083800000000525], [4475.5962431, -360.435359465, 0.237413104999999885], [4675.5962431, -
441.64403499500014, 0.08679285499999949], [4875.5962431, -525.440243405, 0.5679785700000014], [5075.5962431, -
604.055132065, 0.157601194999999875], [5275.5962431, -683.6047936619999, 0.4343817007573888], [5475.5962431, -
763.1544552589999, 0.5127313722591326], [5675.5962431, -846.3331186599999, 0.8232848449999992], [5875.5962431, -
930.636721545, 0.26119209999999981], [6075.5962431, -1010.7651992349998, 0.21505394500000108], [6275.5962431, -
1091.1722235000002, 0.031572014999997934], [6475.5962431, -1169.93262387, 0.83789107999999989], [6675.5962431, -
1254.1539547399998, 0.42048835500000126]], [[5675.5962431, -382.371912070000006, 1020.6920190050001], [5875.5962431, -
321.90918035, 863.69042125], [6075.5962431, -271.848481235, 730.540347595], [6275.5962431, -223.775096975, 590.01069192], [6475.5962431, -
170.109438065, 457.0499207749999], [6675.5962431, -122.29351659500003, 324.45756139]], [[5675.5962431, -
3283.6374613450002, 1019.6251778100001], [5875.5962431, -3222.960719065, 858.62350341], [6075.5962431, -
3174.7239574249998, 725.1066933449999], [6275.5962431, -3121.97615297, 586.5315381], [6475.5962431, -
3071.5604859849996, 459.45667529], [6675.5962431, -
3021.2157763349996, 325.55842979]], [[5675.5962431, 374.60551976500005, 997.64140984], [5875.5962431, 323.205559055, 858.44890522], [6075.5962431,
1, 271.56786810999995, 726.33765731], [6275.5962431, 224.45447140999994, 583.4917694650002], [6475.5962431, 172.23416337999998, 459.141236239999
96], [6675.5962431, 123.75298292000002, 328.446730635]], [[5875.5962431, -2574.973435865, 858.40199737], [6075.5962431, -
2628.471222335, 729.677500825], [6275.5962431, -2675.4759493950005, 584.251270865], [6475.5962431, -
2726.1512859550003, 460.6065804599999], [6675.5962431, -2778.0431787099997, 325.940162835]], [[3579.9085967799997, -
2215.0, 1201.968432965], [3579.7678108200007, -2145.0, 1200.99953248], [3580.368736275, -2075.0, 1203.03304994], [3579.4549611500006, -
2005.0, 1201.971755755], [3579.6232544249992, -1935.0, 1201.9464641050001], [3580.3883498000005, -1865.0, 1201.457109605], [3579.953170755, -
1795.0, 1200.97134187], [3579.857456750001, -1725.0, 1202.0571272149998], [3579.7241155142865, -
1655.0, 1201.7128413663627], [3579.6074419330366, -1585.0, 1201.6592818911477], [3579.5037320830365, -
1515.0, 1201.6116734687362], [3579.4103932180365, -1445.0, 1201.568825888563], [3579.3214880742216, -
1375.0, 1201.5280136258773], [3579.232582930407, -1305.0, 1201.4872013631893], [3581.1651406100004, -
1235.0, 1200.73274922], [3579.9797137749993, -1165.0, 1201.23433694], [3580.378341145, -1095.0, 1200.2718608849998], [3579.388134865, -
1025.0, 1201.9853619250002], [3581.0965710899995, -955.0, 1200.85966605], [3578.0916259000005, -
885.0, 1201.1503272349998], [3579.286587469999, -815.0, 1202.3783775900004], [3581.481864235, -745.0, 1201.0905445700002]], [[-
0.5601482699999998, -2215.0, 0.14392689999999816], [-0.27733399499999986, -2145.0, 0.4137984750000001], [2.0936866800000002, -2075.0, -
0.7858365500000041], [-0.368899654999999844, -2005.0, -0.024147990000003006], [0.401150189999999846, -1935.0, -1.1703766250000052], [-
0.43341488500000336, -1865.0, -0.16026278500000046], [0.07302447500000042, -1795.0, 0.65528505999999978], [0.4739507600000003, -
1725.0, 0.21280866499999718], [0.8530700385714289, -1655.0, -0.3424435590195221], [1.184799407321429, -1585.0, -
0.46630918348123623], [1.479669957321429, -1515.0, -0.5764119607805376], [1.745053452321429, -1445.0, -
0.6755044603499091], [1.9863111750487017, -1375.0, -0.765588550867519], [2.2074640875487015, -1305.0, -
0.8481656338419954], [0.11879090500000374, -1235.0, -0.0492968899999996554], [-0.7959362049999996, -1165.0, 0.15685483999999976], [-

0.00220339000002514, -1095.0, 0.4420594599999513], [1.2006980200000021, -1025.0, -1.035496525000002], [-0.7342714300000012, -955.0, 0.5836218949999961], [-0.31694948500000136, -885.0, 0.4238379099999989], [0.8724867849999996, -815.0, -1.2854291800000026], [-0.30357406000000253, -745.0, 0.8567077150000003]], [[7159.456410795001, -2215.0, 0.34300008499999846], [7160.34579773, -2145.0, -0.37728619500000066], [7160.55573132, -2075.0, -0.03594335000000086], [7161.243523075, -2005.0, 0.20747421499999633], [7160.195450789999, -1935.0, 0.30258303499999994], [7160.008509695001, -1865.0, 0.30132501999999517], [7160.427252740001, -1795.0, 0.021128839999995108], [7160.586376159999, -1725.0, -0.20699794999999738], [7160.841677914285, -1655.0, -0.0169984972912971], [7161.065066949285, -1585.0, -0.05644864465330102], [7161.263634980396, -1515.0, -0.09151544230780928], [7161.444357398936, -1445.0, -0.12343073323995668], [7161.625079817475, -1375.0, -0.15534602417164933], [7161.805802236015, -1305.0, -0.18726131510356936], [7160.79292663, -1235.0, -1.4504288500000038], [7159.851009000001, -1165.0, 0.37709123499999964], [7160.321547754999, -1095.0, 0.31669823999999663], [7160.408380830001, -1025.0, 0.2988221099999997], [7160.185570510001, -955.0, -0.9046843800000012], [7159.33423404, -885.0, -0.017393345000003536], [7160.237325874999, -815.0, -0.009998749999999746], [7161.8980673450005, -745.0, -0.9689969900000008]], [[3579.5579778449996, -2215.0, -1.0699118700000002], [3579.82182751, -2145.0, -0.20592255500000006], [3579.9134937549998, -2075.0, 0.15594701499999958], [3579.8731479000003, -2005.0, 1.2230819249999982], [3580.5007958399997, -1935.0, -0.03268527999999975], [3580.9651738000002, -1865.0, 0.07763862000000046], [3580.3902552150003, -1795.0, 0.4339620899999964], [3581.3291416, -1725.0, 0.13639514999999847], [3581.5821649935715, -1655.0, 0.35076282573390927], [3581.835188387143, -1585.0, 0.4020180461793643], [3582.088211780714, -1515.0, 0.4532732666235688], [3582.3412351742854, -1445.0, 0.5045284870676596], [3582.5942585678567, -1375.0, 0.5557837075136831], [3582.847281961428, -1305.0, 0.6070389279576602], [3581.62689284, -1235.0, 1.6738421600000017], [3582.1985065300005, -1165.0, 0.14593772499999957], [3579.9439745849995, -1095.0, -0.12157506000000211], [3580.0591326099993, -1025.0, 0.2151371349999991], [3581.5102476599995, -955.0, -1.817065955000001], [3581.3123114350005, -885.0, 0.9313024799999994], [3580.15499019, -815.0, -0.7495191699999999], [3580.79054945, -745.0, -0.12772305000000017]], [[-0.6732986799999992, -2215.0, 1200.31311551], [-0.005545284999998757, -2145.0, 1199.88222444], [0.5658307449999995, -2075.0, 1201.901162985], [-0.1169425500000024, -2005.0, 1199.8792640349998], [-0.2688334549999979, -1935.0, 1201.23548038], [0.37382459999999895, -1865.0, 1201.0034522499998], [1.2757426350000038, -1795.0, 1199.3870595399999], [0.283930644999996, -1725.0, 1200.5912812], [0.5623651185714249, -1655.0, 1200.454814722044], [0.8059952829464253, -1585.0, 1200.410725602105], [1.0225554290575367, -1515.0, 1200.3715352732697], [1.2174595605575371, -1445.0, 1200.3362639773181], [1.3946451346484465, -1375.0, 1200.3041991628163], [1.5669737858691504, -1305.0, 1200.2730132930692], [0.4361809749999971, -1235.0, 1200.7616941300003], [0.09254889999999989, -1165.0, 1200.499397315], [0.3166857349999965, -1095.0, 1199.2899288599997], [0.14719013499999747, -1025.0, 1200.9843697699998], [-1.394411950000005, -955.0, 1201.0267106799997], [0.3920664550000012, -885.0, 1200.445726745], [0.0467807800000017, -815.0, 1201.58946349], [1.1926324250000042, -745.0, 1198.874328985]], [[7160.6836469499995, -2215.0, 1199.137076085], [7159.439967830001, -2145.0, 1200.50930116], [7160.36636973, -2075.0, 1200.9578195850002], [7160.341719339999, -2005.0, 1200.1898328900002], [7161.3215249800005, -1935.0, 1201.4765731899997], [7159.230240410002, -1865.0, 1200.2265174150002], [7160.840121855001, -1795.0, 1199.6396149050001], [7159.187392670001, -1725.0, 1200.906371545], [7158.882516625716, -1655.0, 1200.454885889437], [7158.57764058143, -1585.0, 1200.4724400144976], [7158.272764537145, -1515.0, 1200.489994139553], [7157.96788849286, -1445.0, 1200.5075482646105], [7157.663012448575, -1375.0, 1200.5251023896699], [7157.358136404289, -

```
1305.0, 1200.5426565147268], [7160.585828960002, -1235.0, 1201.25942068], [7161.136310065, -1165.0, 1200.6726838199997], [7160.669506644999, -  
1095.0, 1200.9059005400002], [7159.829139125, -1025.0, 1200.449968795], [7160.6626610250005, -955.0, 1201.8939380999996], [7160.399305839999, -  
885.0, 1200.68445744], [7160.2020259599985, -815.0, 1200.95963276], [7160.899468844999, -745.0, 1199.492078135]]]
```

Visualized results, see Figure 44.

A3: Intersection Point Prediction

```

from point_recognition import elements
import matplotlib.pyplot as plt # FOR VISUALS ONLY
import time
import numpy as np

def mean_point(element): # FIND MEAN POINT
    el = np.array(element)
    x = el[:, 0]
    y = el[:, 1]
    z = el[:, 2]

    n_point = [ np.mean(x), np.mean(y), np.mean(z) ]
    return n_point

def closest_point(element_a, element_b): # FIND CLOSEST INTERSECTION POINT
    el_1, el_2 = element_a[0], element_b[0]
    a, b, c = el_1[0], el_1[1], el_1[2]
    d, e, f = el_2[0], el_2[1], el_2[2]

    el_a_mean, el_b_mean = mean_point(element_a), mean_point(element_b)

    v_1 = -(np.array(el_a_mean) - np.array(el_1))
    v_2 = -(np.array(el_b_mean) - np.array(el_2))

    v_a, v_b, v_c = v_1[0], v_1[1], v_1[2]
    v_d, v_e, v_f = v_2[0], v_2[1], v_2[2]

    M_1 = v_d*v_a + v_e*v_b + v_f*v_c
    M_2 = -v_a**2 - v_b**2 - v_c**2
    M_3 = v_d**2 + v_e**2 + v_f**2
    M_4 = -v_a*v_d - v_b*v_e - v_c*v_f

    N_1 = a*v_a + b*v_b + c*v_c - d*v_a - e*v_b - f*v_c
    N_2 = a*v_d + b*v_e + c*v_f - d*v_d - e*v_e - f*v_f

    M = np.array( [ [M_1, M_2], [M_3, M_4] ] )
    N = np.array( [ [N_1], [N_2] ] )

    s = np.linalg.inv(M).dot(N)[0][0]

```

```

t = np.linalg.inv(M).dot(N)[1][0]

X, Y, Z = d + v_d*s, e + v_e*s, f + v_f*s
K, J, L = a + v_a*t, b + v_b*t, c + v_c*t

dist = np.sqrt( (X - K)**2 + (Y - J)**2 + (Z - L)**2 )

return [K, J, L], dist
def distance(point_a, point_b):
x = point_a[0] - point_b[0]
y = point_a[1] - point_b[1]
z = point_a[2] - point_b[2]
return np.sqrt( x**2 + y**2 + z**2 )
def clean_points(points):                                     # RETURNS CLEANED OUT VERSION INTERSECTION POINTS (MEAN POINT OF TH-
OSE WITHIN 25mm)
is_points_clean = []
for point_a in points:
set_points = [point_a]
for point_b in points:
if point_b != point_a and distance(point_a, point_b) < 100:
set_points.append(point_b)
mean_p = mean_point( set_points )

if mean_p not in is_points_clean:
is_points_clean.append(mean_p)

for point_a in is_points_clean:                             # REMOVE SOME OVERLAPPING POINTS DUE TO INACCURACY IN COMPUTATION
for point_b in is_points_clean:
if point_a != point_b and distance(point_a, point_b) < 1:
is_points_clean.remove(point_b)
return is_points_clean

intersection_points_raw = []
no_lst = [0, 1, 2, 3, 6, 8, 11, 20]                       # MAIN ELEMENTS BY INDEX NR, WHERE LINES INTERSECT
for i in no_lst:                                           # GET THE INTERSECTION POINTS OF ALL ELEMENTS GOING THROUGH MAIN ELE
MENTS (DISTANCE < 10mm)

```

```
el_a = elements[i]
for k, element in enumerate(elements):
    if k not in no_lst:
        cp, dist = closest_point(el_a, element)[0], closest_point(el_a, element)[1]
        if dist < 10 and ( -200 < cp[0] < 8000 ) and ( -4000 < cp[1] < 500 ):
            intersection_points_raw.append(cp)

points_clean = clean_points(intersection_points_raw)
```

Execution time >> 0.762063 sec

RESULTING INTERSECTION POINTS:

```
points_clean = [  
[ 1.78965048e+03, -4.49894022e+02, 1.20077951e+03]  
[ 5.37837645e+03, -4.49289167e+02, 1.20097057e+03]  
[ 3.57979952e+03, -4.49592305e+02, 1.20087482e+03]  
[ 1.95119859e+00, -4.50195326e+02, 1.20068434e+03]  
[ 7.15844230e+03, -4.48989150e+02, 1.20106534e+03]  
[ 4.34002835e+00, -2.90011697e+03, 5.84206758e-02]  
[ 3.58380470e+03, -2.89813551e+03, 1.00809022e-01]  
[ 7.14047661e+03, -2.89906371e+03, 9.30260312e-01]  
[ 1.78451298e+03, -2.44955661e+03, 1.20051026e+03]  
[ 5.34864765e+03, -2.44920942e+03, 1.20069954e+03]  
[ 3.57992316e+03, -2.44938172e+03, 1.20060561e+03]  
[-1.02238198e+00, -2.44973055e+03, 1.20041544e+03]  
[ 7.16098057e+03, -2.44903287e+03, 1.20079579e+03]  
[ 6.72003033e+00, 6.72064499e-01, 4.56480579e-01]  
[ 3.58323354e+03, -9.48487844e-02, 8.11750555e-01]  
[ 7.13979539e+03, -7.04863751e-01, 3.54835498e-01]  
[ 1.77781462e+03, 4.50512876e+02, 1.19997159e+03]  
[ 5.35141238e+03, 4.50390818e+02, 1.20124512e+03]  
[ 3.57974383e+03, 4.50451330e+02, 1.20061375e+03]  
[ 3.29128336e+00, 4.50573485e+02, 1.19933920e+03]  
[ 7.15730166e+03, 4.50329137e+02, 1.20188869e+03]  
[ 1.15035438e+00, -1.44975616e+03, 2.50656893e-01]  
[ 1.78615293e+03, -3.34957971e+03, 1.20087251e+03]  
[ 5.37597919e+03, -3.34959691e+03, 1.20127648e+03]  
[ 3.57997887e+03, -3.34958831e+03, 1.20107437e+03]  
[-2.36062060e+00, -3.34957114e+03, 1.20067124e+03]  
[ 7.16212317e+03, -3.34960547e+03, 1.20147748e+03]  
[ 7.16865826e+03, -1.44983298e+03, 1.25739494e+00]  
]
```

Visualized results, see Figure 47.