



HAL
open science

Simplification of 3D CAD models with deep learning for augmented reality

Abhaya Dhathri Arige

► **To cite this version:**

Abhaya Dhathri Arige. Simplification of 3D CAD models with deep learning for augmented reality. Artificial Intelligence [cs.AI]. Institut Polytechnique de Paris, 2023. English. NNT : 2023IPPAS018 . tel-04420117

HAL Id: tel-04420117

<https://theses.hal.science/tel-04420117>

Submitted on 26 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Simplification of 3D CAD models with deep learning for augmented reality

Thèse de doctorat de l'Institut Polytechnique de Paris

préparée à Télécom SudParis

École doctorale n°626 Institut Polytechnique de Paris (ED IP Paris)

Thèse présentée et soutenue à Évry, le 14/12/2023, par

Spécialité de doctorat : Informatique

Abhaya Dhathri ARIGE

Composition du Jury:

Valeriu VRABIE

Professeur, Université de Reims Champagne-Ardenne

Président & Rapporteur

Jean-Marc LECAILLEC

Professeur, IMT Atlantique

Rapporteur

Francisco MORAN BURGOS

Professeur, Universidad Politécnica de Madrid

Examineur

Arianne HINDS

Docteur, Tencent America

Examineur

Titus ZAHARIA

Professeur, Télécom SudParis

Directeur de thèse

Marius PREDA

Maitre de Conférence, Télécom SudParis

Co-encadrant de thèse

Emmanuel BRICARD

CEO, SHIFT89

Invité

TABLE OF CONTENTS

Table of contents	iii
List of tables.....	vii
List of figures	ix
List of abbreviations	xiii
Acknowledgments	xv
Abstract	xvii
Chapter 1 General Introduction	1
1.1 Introduction	2
1.2 Background and Motivation	2
1.3 Context.....	4
1.4 Thesis objectives	6
1.5 Thesis contribution	6
1.6 Thesis Outline.....	7
1.7 Publications.....	8
Chapter 2 AR FOR CAD USECASE	9
2.1 Industrial revolution.....	10
2.2 Augmented reality:	11
2.2.1 Applications of augmented reality.....	13
2.2.2 AR in industrial assembly lines.....	15
2.2.3 Media in AR.....	19
2.3 Case study	20
2.4 Visual AR assets for different use cases.....	23
2.4.1 Images and videos.....	25
2.4.2 3D CAD models in AR for workstation setup, creation of instructions and validation.....	29
2.4.3 Solution: 3D model simplification	31
2.5 Summary	32
Chapter 3 State of the art	35
3.1 Introduction	36

3.2	Analysis of simplification techniques.....	38
3.2.1	Feature based simplification.....	39
3.2.2	Dimension reduction-based simplification.....	40
3.2.3	Mesh based simplification.....	41
3.3	Analysis of mesh-based simplification techniques: traditional methods.....	42
3.3.1	Vertex removal.....	44
3.3.2	Edge collapse.....	44
3.3.3	Triangle collapse.....	45
3.3.4	Vertex clustering.....	46
3.4	Mesh based simplification: deep learning techniques.....	49
3.4.1	Simplification in the 3D domain.....	49
3.4.2	MeshCNN: conditions of use.....	50
3.4.3	Difficulties and limitations.....	53
	Chapter 4 Evaluations.....	55
4.1	Objective evaluation.....	56
4.1.1	Evaluation metric.....	57
4.1.2	Test database.....	58
4.1.3	Comparison of mesh simplification techniques.....	60
4.2	Assessment of the level of simplification.....	68
4.3	Framerate evaluation in a real case scenario.....	73
4.4	Subjective user evaluations.....	75
4.4.1	Related work.....	76
4.4.2	Experimental setup.....	77
4.4.3	Instructions.....	78
4.4.4	Evaluation procedure.....	82
4.4.5	Participants and tasks performed.....	83
4.4.6	Evaluation metrics and results.....	85
4.5	Summary.....	89
	Chapter 5 Proposed methodology: Mesh simplification of continuous chains.....	91
5.1	Continuous chains: definition and problem statement.....	92

5.1.1	Continuous chains.....	92
5.1.2	Segmentation of continuous chains: curvature-based analysis	94
5.2	Proposed 2D/3D continuous chain segmentation methodology	95
5.2.1	3D to 2D projection.....	96
5.2.2	2D segmentation of continuous chains	100
5.2.3	2D to 3D mapping	103
5.3	Simplification using graph traversal technique	104
5.4	Experimental results	107
5.4.1	Database pre-processing.....	107
5.4.2	Database creation	110
5.4.3	UNet 2D Segmentation of continuous chains.....	111
5.4.4	Final results of simplification of continuous chains.....	119
	Chapter 6 Conclusion and perspectives	123
	References.....	127
	Appendix	139

LIST OF TABLES

Table 2-1. Effectiveness of visual assets inside AR	25
Table 3-1. Different mesh-based simplification techniques	48
Table 4-1. Simplification rates of each range of vertices.....	70
Table 4-2. Different measurements for the Hydraulic model.....	73
Table 4-3. Framerrates in HoloLens 2	75
Table 4-4. Participants and tasks done	84
Table 4-5. Errors done by participants.....	89

LIST OF FIGURES

Figure 1-1. 3D representation of a scene	3
Figure 1-2. Different Level of details of a 3D model	4
Figure 1-3. Smart training in industries	5
Figure 2-1. Industrial revolution over the years	11
Figure 2-2. Types of realities	13
Figure 2-3. Operator on assembly lines in an industry	16
Figure 2-4. AR Workflow in industries	17
Figure 2-5. Typical assembly line	18
Figure 2-6. Assembly line in a real industry (photo from elm. Leblanc factory).....	18
Figure 2-7. Types of media utilized in AR applications	20
Figure 2-8. Assembled boiler for our case study	22
Figure 2-9. The three main use cases of AR in industrial assembly lines.....	23
Figure 2-10. Various views of a screw.....	26
Figure 2-11. Evaluating usage of images in AR	27
Figure 2-12. AR instruction with image.....	28
Figure 2-13. 3D model workflow in industries.....	30
Figure 2-14. Examples of simple CAD models.....	31
Figure 2-15. Examples of complex CAD models.....	31
Figure 2-16. Complete workflow of 3D model-based instructions in industrial assembly lines.	32
Figure 3-1. plate	37
Figure 3-2. screw	37
Figure 3-3. Assembly process.....	37
Figure 3-4. Examples of simplified models using MeshLab	38
Figure 3-5. Example CAD model - Holder.....	39
Figure 3-6. Feature based simplification.....	40
Figure 3-7. Dimension reduction-based simplification	41
Figure 3-8. Mesh-based simplification.....	42
Figure 3-9. Attributes of a 3D model	43
Figure 3-10. Vertex decimation operation.....	44
Figure 3-11. Edge decimation operation	45
Figure 3-12. Triangle decimation operation	46

Figure 3-13. Vertex clustering operation	47
Figure 3-14. Edge collapse in MeshCNN (Hanocka et al., 2019)	50
Figure 3-15. Feature aggregation during mesh pooling and unpooling (Hanocka et al., 2019) .	52
Figure 3-16. Simplification stages with MeshCNN (Hanocka et al., 2019).....	53
Figure 4-1. Percentage of models in the test dataset with respect to the number of vertices .	59
Figure 4-2. Examples of CAD models in database.....	60
Figure 4-3. Test models.....	61
Figure 4-4. Average error for connector.....	62
Figure 4-5. Average error for pipe	62
Figure 4-6. Average error for mixing unit	63
Figure 4-7. Average error for fan	63
Figure 4-8. Average error for heat exchanger	64
Figure 4-9. Maximum error for connector.....	65
Figure 4-10. Maximum error for pipe	65
Figure 4-11. Maximum error for mixing unit	66
Figure 4-12. Maximum error for fan.....	66
Figure 4-13. Maximum error for heat exchanger	67
Figure 4-14. Examples of simplified models by (Lindstrom & Turk, 1998)– Turbine pipe.....	69
Figure 4-15. Examples of simplified models by (Lindstrom & Turk, 1998)– Bolt.....	69
Figure 4-16. Visual evaluation of the simplified models at the retained simplification rates....	71
Figure 4-17. Hydraulic model.....	72
Figure 4-18. HoloLens2	74
Figure 4-19. Original and simplified models visualized in HoloLens 2.....	75
Figure 4-20. Overview of the evaluations.....	78
Figure 4-21. Assembly of screw	79
Figure 4-22. Assembly of fit	80
Figure 4-23. Assembly with control box	80
Figure 4-24. Assembly with railing.....	81
Figure 4-25. Assembly with cross member.....	82
Figure 4-26. Assembly with bracket.....	82
Figure 4-27. Instructions per participant	84
Figure 4-28. Completion times of the evaluation tasks per participant	87
Figure 4-29. Comprehensiveness score of participants.....	88

Figure 5-1. Example of continuous chains	93
Figure 5-2. Limitations of the curvature-based continuous chain identification	95
Figure 5-3. Continuous chain segmentation workflow.....	96
Figure 5-4. Database preparation	97
Figure 5-5. The three 2D projection images generated for a given model.....	98
Figure 5-6. Camera positioning (Fibonacci sphere algorithm).....	98
Figure 5-7. Visual representation of camera distribution.....	100
Figure 5-8. UNet Architecture (Ronneberger et al., 2015)	103
Figure 5-9. 2D to 3D mapping	104
Figure 5-10. Graph traversal technique	105
Figure 5-11. Continuous chain removal process.....	106
Figure 5-12. 3D CAD models in database.....	110
Figure 5-13. Different views inside database.....	111
Figure 5-14. Evolution of training	112
Figure 5-15. Average IOU score for UNet	113
Figure 5-16. Loss graph	114
Figure 5-17. Evaluation metrics after Optuna	115
Figure 5-18. Visual comparison of before and after optuna.....	116
Figure 5-19. Segmented views of the Fitting model.....	117
Figure 5-20. Segmented views of Ring model.....	118
Figure 5-21. Final simplified models	120
Figure 5-22. Comparison to state of the art	121
Figure 5-23. Visual comparison to state of the art	122

LIST OF ABBREVIATIONS

2D	Two dimension
3D	Three dimension
AR	Augmented reality
VR	Virtual reality
MR	Mixed reality
Fps	Frames per second
CAD	Computer aided design
DL	Deep learning
LOD	Level of details
HMI	Human machine interface

ACKNOWLEDGMENTS

My PhD journey has been a wonderful experience and would not have been possible without the support and guidance I received from many people.

First and foremost, my heartfelt thanks to my advisors Dr. Marius PREDA and Prof. Titus ZAHARIA from Telecom SudParis for providing me with this wonderful opportunity. I sincerely thank them for guiding me throughout my research with great patience and encouragement along with their immense knowledge. Thank you for always being supportive and giving the freedom to explore new subjects without fear of failure.

A special thanks to Emmanuel BRICARD for giving me the great opportunity to work in the project IRONMEN. I owe a deep sense of gratitude to him for the time he has given me during the beginning of the thesis which helped a lot to finish my thesis. I cannot thank him enough for making me part of a wonderful project which helped me to gain professional knowledge about industries and how they operate. I would also like to extend my thanks to the colleagues at Elm LeBlanc for their interest and feedback on my work.

My sincere thanks to all the members of the PhD committee for their valuable suggestions and feedback regarding my thesis.

I am grateful to have had the opportunity to work with some amazing people Traian LAVRIC, Hugo DURCHON, Mathilde DROUOT and Pierre BEGOUT as a part of Project IRONMEN. The brainstorming and technical discussions we had has helped to come up with new ideas which helped greatly in the advancement of the thesis.

I am very thankful to all my colleagues from telecom SudParis for being very kind. I have learnt lot of things both professionally and personally thanks to Antoine DIDIER, Chao CAO, Christian TULVAN, Germain BREGEON, Kaouther OUENNICHE, Lea SAUNIER, Minderis KRIR, Nicolas HOFFMANN, Noureddine KHIATI, Zied LAHIANI. A special thanks to Evelyne TARONI for helping with all the administration works and being very patient in explaining everything to me in detail.

I am deeply indebted to my parents and my brother who made sure I am provided with bestest of opportunities. I want to extend my heartfelt gratitude to my beloved family and friends. Your unwavering moral support has been a source of strength for me throughout these years. Each of you has played a vital role in my journey, and I am truly blessed to have you in my life.

Last but the least, a very special thanks goes to my husband Ramarao GANJI. From the very beginning, you have been my rock, my motivator, and my biggest supporter. Your belief in me and your constant encouragement have been invaluable. I couldn't have come this far without you by my side. Thank you for being my pillar of strength and my greatest source of inspiration.

ABSTRACT

As a part of Industry 4.0 the use of Augmented Reality (AR) devices like HoloLens has gained significant acceptance for training assembly line operators in various industries. When employing Computer-Aided Design (CAD) models to create assembly line instructions for training purposes, preserving all redundant information becomes unnecessary. Utilizing simplified CAD models leads to improved run-time performance of the applications in which they are employed. This specific research project is tasked with developing methods and techniques to streamline complex 3D CAD models, making them suitable for AR applications.

In this research, we explain how 3D models play a significant role in augmented reality (AR) by enriching the virtual experience through the superimposition of computer-aided design (CAD models) onto the real world. The study goes on to offer detailed descriptions of numerous applications of AR in operator training. Furthermore, it elucidates how the integration of 3D CAD models contributes to a deeper understanding of instructions and procedures within these training scenarios.

We conducted an in-depth literature review in the field of CAD model simplification to determine which simplification techniques are most suitable for integration into augmented reality (AR) scenarios. Our research revealed that mesh-based simplification techniques are particularly effective in preserving the essential features of CAD models while offering the advantages of precise control over the level of detail.

Additionally, we have carried out four distinct types of assessments as part of our research. These assessments encompassed objective evaluations that applied mesh-based techniques from existing literature, subjective assessment involving a thorough examination of each simplified model to determine the level of simplification based on vertex ranges, real-world testing conducted with the assistance of the HoloLens2 that demonstrated framerate enhancements when employing simplified CAD models in place of their original versions. To conclude our evaluations, we conducted user assessments, as user experience holds utmost importance in our study. They demonstrated that the simplified models possess a high degree of capability in substituting the original counterparts. However, it was noted that more simplification is required, particularly for intricate CAD models.

An innovative approach centered around segmentation and adaptive simplification through the utilization of deep learning methods is proposed as the main methodology. To illustrate this framework, we employed a specific feature called "continuous chains". We subsequently conducted a comparative analysis against established state-of-the-art techniques, demonstrating that our methodology outperforms existing approaches. In our future research, we intend to expand the scope of our framework to encompass multiple features in CAD model.

Chapter 1

GENERAL INTRODUCTION

1.1 Introduction

3D models are essential tools used in various industries for designing, visualizing, and prototyping objects and structures. They play a crucial role in fields such as engineering, architecture, product design, video games, animation, and more. A 3D model is a virtual, digital representation of an object or a scene encountered in the real life. Unlike 2D images, which are planar representations, 3D models also have depth, allowing them to mimic real-world objects more realistically. Such 3D models can be created using specialized software platforms that enables users to manipulate vertices, edges, and faces to construct complex shapes. The models are used for various purposes, such as visualization, simulation, and 3D printing. They provide a detailed view of the object from different angles, aiding in understanding its form, structure, and proportions.

CAD stands for Computer-Aided Design. CAD models are a specific type of 3D model that is created using computer software designed for engineering and architectural design. CAD software allows designers and engineers to create precise, dimensionally accurate models of physical objects or structures. CAD models are commonly used to create prototypes, perform simulations, and generate technical drawings that can be used for manufacturing. They are particularly valuable in industries like aerospace, automotive, and industrial design, where precision and accuracy are of utmost importance.

Complex 3D CAD models can sometimes be computationally expensive, challenging to work with, or challenging to visualize, especially when dealing with large-scale projects. To address such issues, simplification techniques are used to reduce the complexity of the models while preserving their essential features.

1.2 Background and Motivation

Over the last two decades, 3D models (Fitzgerald et al., 2015; Pollefeys & Gool, 2002) have gained immense popularity and found widespread use in various aspects of everyday life. These versatile models (Figure 1-1) are now integrated into numerous applications, including navigation, entertainment, tourism, healthcare, education, and more. One of the significant contributors to their widespread adoption is the availability of highly performant, dedicated software platforms like Blender, Unity, AutoCAD and Maya, which enables easier 3D model acquisition with highly realistic representations of real-world objects. In the industrial sector, 3D models play a crucial role, enhancing productivity and simplifying work processes, making them an indispensable asset in modern projects.



Figure 1-1. 3D representation of a scene

While complex 3D objects excel in simulating reality, they come with certain drawbacks that hinder their use in practice. Difficulties arise when handling, rendering, or transmitting such complex objects over the internet, as they demand larger storage memory and bandwidth. To address these challenges, the concept of mesh simplification has emerged, and several algorithms (Asmara et al., 2010; Cignoni, Montani, et al., 1998; Gotsman et al., 2002; Ng & Low, 2014) have been developed to achieve this goal. The primary objective of the mesh simplification algorithms is to retain a high level of fidelity in the original object while reducing its complexity, making it more manageable and efficient for various application (Figure 1-2).

Highly detailed 3D models offer a more realistic representation of objects, but their increased complexity can present challenges in terms of interaction. In specific use cases like video games, where interactivity takes a primordial role, simplifying the 3D models becomes essential. By reducing the level of detail and information, rendering and interactivity can be significantly improved, leading to a smoother and more responsive user experience. However, it is important to note that this approach may not be suitable for all applications, as certain scenarios require higher levels of detail to accurately depict the intended content. Striking a balance between complexity and interactivity is crucial in tailoring 3D models for diverse applications.

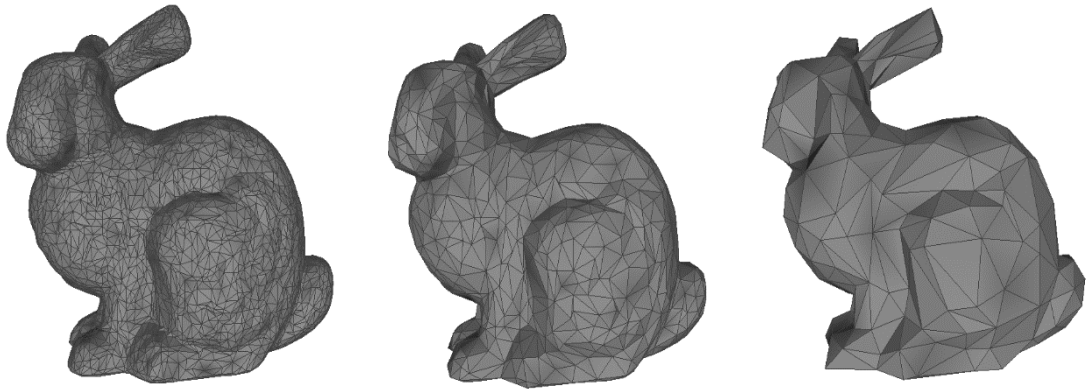


Figure 1-2. Different Level of details of a 3D model

CAD (Computer-Aided Design) (BERNARDINI et al., 1999; de Vries & Harink, 2007; Fuh & Li, 2005; Gevorgian et al., 1995; Jayanti et al., 2006) empowers engineers and designers to construct realistic computer models of parts and assemblies for intricate simulations and digital manufacturing. Widely-used software like AutoCAD, SolidWorks, Catia, Siemens NX, and Autodesk facilitate the creation of such models. However, the 3D models produced by these CAD systems often contain an abundance of features, making them highly detailed and resulting in huge file sizes. Given their usage in industrial production, it becomes necessary to simplify the models before employing them in any application. The development of efficient simplification methods (Thakur et al., 2009; Yoon & Kim, 2016) and technologies, while considering various application constraints, has become a critical challenge in the field.

1.3 Context

Industry 4.0 has paved way for a new era of intelligent machines (Figure 1-3), revolutionizing efficiency and effectiveness across various industrial sectors such as production, maintenance, and distribution. One such groundbreaking technology that has emerged is Augmented Reality (AR), which seamlessly blends virtual information with the real-world environment, enhancing overall experiences. As a key component of Industry 4.0, AR is now playing a crucial role in aiding assembly line operators during the training process. The integration of AR into diverse applications is steadily increasing, with early research on displaying tasks using AR dating back to the 1990s (Feiner et al., 1993; Wiedenmaier et al., 2003). Today, numerous companies have already implemented potent AR tools on their shop floors, and the adoption of this technology is rapidly spreading. AR systems are proving to be highly promising training platforms, particularly for complex and demanding tasks, as they offer immense potential for improving training efficiency and competency.



Figure 1-3. Smart training in industries

In AR, implementing instructions becomes significantly easier and more comprehensible when they are presented as 3D models superimposed onto real equipment, rather than relying solely on text and pictures. This approach enables operators to follow step-by-step procedures with clarity, understanding precisely what tasks are required and how to execute them. By closely replicating the real-world environment, 3D models offer accurate information about the objects, thereby maximizing the potential for a reliable and immersive Augmented Reality (AR) experience. Since they are very close to the real-world environment using 3D models typically provides accurate information about the object, maximizing the potential for reliable AR experience.

Within the context of CAD/AR applications, the most widely used representation is based on 3D meshes. Such models contain finely detailed, arbitrary topology surfaces with potentially billions of vertices. As the complexity of the models increases, the visual approximation to the real-world objects gets better. However, there is a definite trade-off to be established between the cost of processing the models and the visual degree of realism achieved. In this regard, large amounts of storage, RAM, computational power, and bandwidth are required for transmitting and rendering the complex 3D models. Hence, simplification of models plays a very important role for ensuring their applicability in real-life applications.

This research is been carried out within the framework of a CIFRE convention between elm. leblanc, a Bosch subsidiary, and Telecom SudParis. It has contributed to a larger project, so-called "IRONMEN," focused on the training of assembly line operators through the utilization of Augmented Reality (AR) devices.

1.4 Thesis objectives

This thesis focuses on the development of a simplification tool specifically designed for complex 3D CAD models. While the concept of simplifying CAD models is widely explored, recent trends concern the adaptation of the simplification processes to Augmented Reality (AR) applications in industrial assembly lines. CAD models often present an overwhelming number of triangles, which poses challenges for rendering in AR systems using the current data interface. To address this issue, we propose a novel simplification technique that combines feature-based paradigm with mesh-based simplification of CAD models. This innovative approach aims to optimize the CAD models for AR rendering while preserving essential features, and ensuring that they meet the requirements of efficient rendering in Augmented Reality systems.

1.5 Thesis contribution

The thesis encompasses three main contributions. Firstly, it involves an in-depth review of the state-of-the-art, a critical undertaking given the relatively new field of employing simplified CAD models in augmented reality (AR) for generating assembly line instructions. This comprehensive review explores various categories of CAD model simplification techniques to identify the most appropriate approach for the specific application of utilizing simplified CAD models to train assembly line operators within industrial settings.

The second contribution of this research entails a comprehensive investigation, encompassing objective assessments, subjective assessments and evaluations. The objective evaluations presented the best method among the state-of-the-art techniques, while subjective assessments helped identify which simplified model should be selected for use in the specific use case. A crucial aspect of this investigation involves conducting a subjective evaluation with a group of 14 users who assessed the utilization of 3D CAD models within a HoloLens device for operator training. This subjective evaluation serves as a pivotal component of our research, as it offers valuable insights into the user experience and the technology's acceptance in a real-world training context. While the utilization of simplified CAD models is promising, a noteworthy limitation raised when there was a need for incorporating additional models into the scene. This limitation underscores the necessity for an enhanced and innovative simplification strategy to meet the demands of complex training scenarios.

The third contribution concerns the formulation of an adaptive mesh simplification scheme, with adaptability rooted in functional characteristics. The proposed method is characterized by a two-step process: initially, a mesh segmentation is performed in order to identify functional regions, employing an innovative deep learning method. Subsequently, the simplification is applied distinctively to each of these regions.

1.6 Thesis Outline

The remaining sections of the manuscript are structured as follows:

Chapter 2 provides a comprehensive overview of Augmented Reality, its applications in industrial assembly lines, and various use cases within industries. We delve into the utilization of virtual assets for AR-based training in industrial settings and address the challenges associated with 3D assets. Additionally, we propose a potential solution to overcome these challenges.

Chapter 3 extensively examines various existing simplification techniques designed for 3D models, as documented in the literature. An evaluation is undertaken to gauge their appropriateness in the context of CAD modelling and operator training within assembly line scenarios. This evaluation considers the strengths and weaknesses inherent in different categories of CAD model simplification techniques.

Chapter 4 presents objective evaluations performed using real-world industrial CAD models, and expert assessments to determine the optimal level of simplification required. A detailed subjective evaluation of using simplified 3D CAD models in AR for industrial assembly lines is also proposed. The tests are conducted in a laboratory setting, replicating tasks and conditions similar to those found in actual industrial workstations. An unbiased evaluation is carried out, and the results are analyzed and presented with respect to different parameters.

Chapter 5 introduces a deep learning-based framework designed to identify features in CAD models and perform feature-based mesh simplification. The framework incorporates Unity, a deep learning model, and a 2D to 3D mapping strategy for segmenting CAD model features. We focus on specific features, so-called continuous chains and provide detailed insights into the framework.

Chapter 6 concludes the manuscript by summarizing the literature review, methodologies implemented, and the obtained results. We also discuss the research perspectives and the challenges that may arise in future endeavors.

1.7 Publications

[1] Abhayadhathri Arige, Traian Lavric, Marius Preda, Titus Zaharia, Emmanuel Bricard. Analysis of 3D CAD MESH simplification approaches within the framework of AR applications for industrial assembly lines. ISIE 2021: IEEE 30th International Symposium on Industrial Electronics, Jun 2021, Kyoto (online), Japan. pp.1-6.

[2] Abhayadhathri Arige, Traian Lavric, Marius Preda, and Titus Zaharia. 2022. Evaluation of simplified 3D CAD data for conveying industrial assembly instructions via Augmented reality. In Proceedings of the 27th International Conference on 3D Web Technology (Web3D '22). Association for Computing Machinery, New York, NY, USA, Article 7, 1–6. <https://doi.org/10.1145/3564533.3564568>.

[3] Abhayadhathri Arige, Marius Preda and Titus B. Zaharia. “Smart AR workstation configuration in industrial assembly lines.” 2023 IEEE International Conference on Industrial Technology (ICIT) (2023): 1-4.

[4] Arige, Marius Preda and Titus B. Zaharia. “SIMPLIFICATION OF CONTINUOUS CHAINS IN 3D CAD MODELS FOR INDUSTRIAL AR APPLICATIONS.”19th International Conference on Intelligent Computer Communication and Processing (ICCP 2023): pp.1-6.

Chapter 2

AR FOR CAD USECASE

2.1 Industrial revolution

The *industrial revolution* (Figure 2-1), a pivotal epoch in human history, represents a series of profound transformations that reshaped the world's economic, social, and technological landscape. This monumental shift, which unfolded over several centuries, marked the transition from agrarian and craft-based societies to industrial and mechanized ones. It set in motion a cascade of changes that continue to influence the modern world. At its core, the industrial revolution was driven by innovation and technological breakthroughs that catapulted humanity into an era of unprecedented progress.

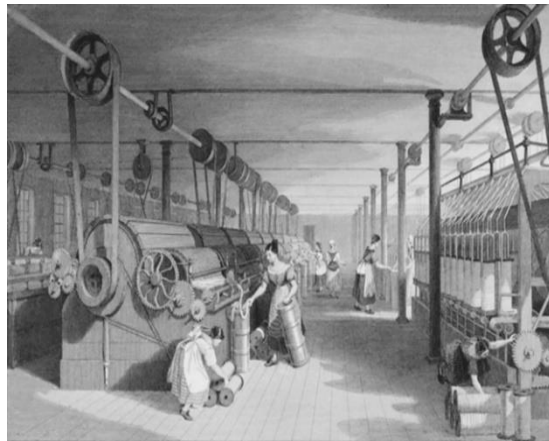
The first industrial revolution began in the 18th century with the mechanization process using water and steam. The introduction of machinery has significantly affected many areas like textile and transportation by aiding human productivity.

The second industrial revolution started in the 19th century and is mainly referred to as the technological revolution. A significant revolution happened with the introduction of electricity, which led to the development of machinery facilitating mass production. Assembly lines became common, optimizing workplace efficiency.

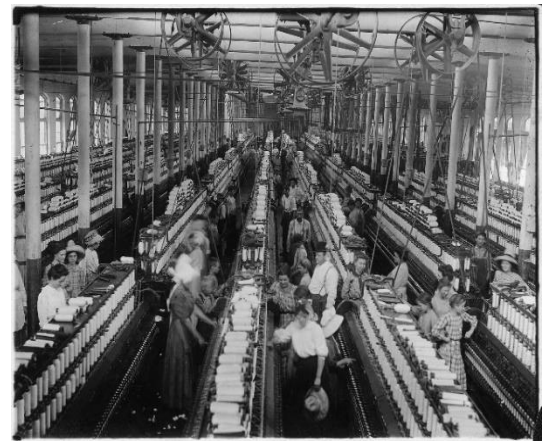
The third revolution started in the 20th century with the invention of computers. The rise of information technology has made it easy to automate whole production lines without human intervention. Robotic assistance in industries has also gained momentum. The third industrial revolution laid the foundation for the digital age and set the stage for the fourth industrial revolution (Industry 4.0) by introducing digital technologies, automation, and global connectivity.

Today, we are currently living the fourth industrial revolution, which is also known as “Industry 4.0” (Cañas et al., 2021; Frank et al., 2019; Lasi et al., 2014; Philbeck & Davis, 2018; Vaidya et al., 2018). The fourth industrial revolution, often abbreviated as 4IR or Industry 4.0, represents the ongoing and transformative era of technological advancement that is reshaping industries, economies, and societies. This revolution builds upon the digital age of the late 20th century and is characterized by the convergence of digital, biological, and physical technologies. Key elements of the Fourth Industrial Revolution include artificial intelligence, quantum mechanics, Internet of things, augmented and virtual reality, robotics and automation, renewable energy, and many more.

The fourth industrial revolution is changing the way we live, work, and interact with the world around us. It is reshaping industries, economies, and societies by enhancing productivity, enabling automation, fostering sustainability, and driving innovation. While it offers prospects for innovation and expansion, the Fourth Industrial Revolution also introduces formidable technical hurdles and complexities.



a) First industrial revolution
(1750 – 1850)



b) Second industrial revolution
(1750 – 1850)



c) Third industrial revolution
(1950 – 2000)



d) Fourth industrial revolution
(2000 - Present)

Figure 2-1. Industrial revolution over the years

2.2 Augmented reality:

The reality-virtuality continuum (Figure 2-2) (Doolani et al., 2020; Milgram et al., 1995) consists of three primary types of realities: Virtual Reality (VR), Augmented Reality (AR), and Mixed Reality (MR). Each of them represents a different level of interaction between the digital and physical worlds. In Virtual Reality (VR), the real world is completely concealed as users immerse themselves in a digitally simulated environment, often in real-time. On the other hand, Augmented Reality (AR) overlays interactive digital objects onto the real world, allowing users

to interact with virtual elements while still being aware of their physical surroundings. Mixed Reality (MR) combines the capabilities of both AR and VR, enabling a seamless integration of digital and physical objects. In MR, these two realms coexist and interact in real time, creating a more immersive and interactive experience.

AR technology has evolved rapidly in recent years, driven by advancements in hardware, software, and computer vision. As a result, AR is becoming increasingly accessible and is finding applications in everyday life, from gaming and social media filters to training, education, and practical solutions for businesses and industries. Its potential for innovation and enhancing the way we interact with the world around us continues to grow. Augmented Reality (AR) offers distinct advantages over Virtual Reality (VR) in industrial applications.

1. **Real-World Integration:** AR enhances the real world by overlaying digital information, making it well-suited for industries where employees need to interact with their physical environment while accessing digital data. In contrast, VR creates entirely virtual environments, which may not be practical for tasks that require interaction with real-world objects.
2. **Maintaining Spatial Awareness:** Industrial settings often require employees to be aware of their surroundings for safety and efficient work. AR allows users to see and interact with real-world objects while accessing digital information, while VR isolates users from their physical environment, potentially leading to safety concerns.
3. **Hands-Free Operation:** Many industrial tasks involve the use of tools and equipment. AR can be implemented in hands-free devices like smart glasses, allowing workers to access information without the need to hold a separate device. VR typically requires users to be tethered to a headset, limiting mobility and dexterity.
4. **Training and Maintenance:** AR is valuable for training and maintenance tasks, as it can provide real-time guidance and information on equipment and processes, improving efficiency and reducing errors. VR is better suited for immersive training but lacks the immediate connection to the real-world equipment.
5. **Wider Range of Use Cases:** AR has a broader range of industrial applications, including remote assistance, real-time data visualization, assembly instructions, quality control, and more. VR is better suited for immersive simulations and training.

While AR offers advantages for many industrial scenarios, it is important to note that the choice between AR and VR depends on the specific use case and requirements. In some situations, a combination of both technologies, known as Mixed Reality (MR), may provide the best solution, allowing users to seamlessly transition between fully virtual, augmented, and real-world experiences as needed.

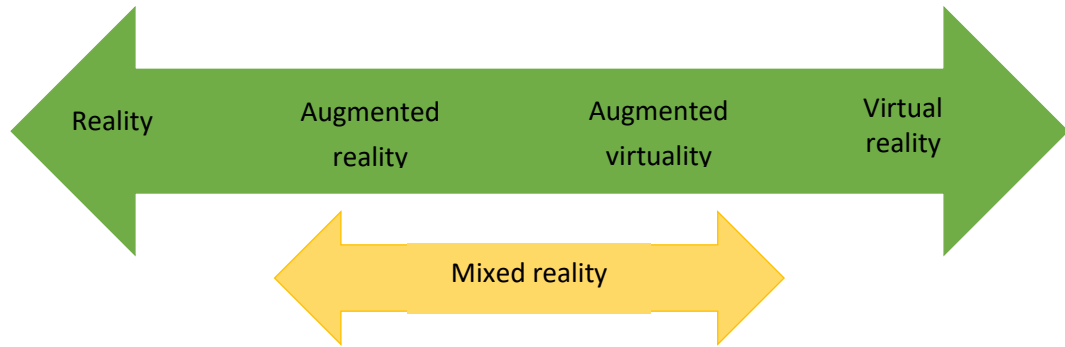


Figure 2-2. Types of realities

2.2.1 Applications of augmented reality

Augmented reality (AR) is used in various sectors (Carmigniani et al., 2011; de Souza Cardoso et al., 2020; Manuri & Sanna, 2016; Mekni & Lemieux, 2014; Sharma, 2019), including navigation, health care, shopping, gaming, sports education, travel, policing, real estate, remote assistance, food services, marketing, and many more. Some examples are given below.

Augmented Reality has proven to be highly beneficial in the medical field (Eckert et al., 2019), enhancing the efficiency of treatment for both doctors and patients. Its applications in healthcare are numerous, with one significant use being in assisting doctors during surgical procedures. In medical treatment, AR enables patients to gain a deeper understanding of their prescribed medications as they can visually see how the drugs work in front of them. This visual representation helps in improving patient compliance and understanding. Moreover, AR facilitates better communication between patients and healthcare professionals. Patients can easily describe their symptoms using interactive AR tools, allowing doctors to gain clearer insights into their conditions and provide more accurate diagnoses. AR has paved the way for interacting and learning with students (Chimakurthi, 2019), helping them quickly understand the context. Overall, the implementation of augmented reality in the medical field has proven to be a valuable asset, streamlining various aspects of healthcare and contributing to more effective and efficient treatments.

Augmented reality has emerged as a powerful tool to enhance the educational process (Chen et al., 2017), offering students a more engaging and safer learning experience. By leveraging this technology, students can conduct virtual experiments, such as exploring various chemical combinations, without any risk of harm. The applications of augmented reality in education extend beyond traditional classrooms, as it also serves as a convenient platform for online learning. Through AR applications, students can access essential learning materials and seamlessly collaborate with peers and teachers in real-time, regardless of their physical locations. Furthermore, AR enables educators to present information in innovative ways,

catering to different learning styles and preferences. Visualizing abstract concepts and simulations in augmented reality aids students in grasping difficult topics more easily, leading to a deeper comprehension of the subject matter. The integration of augmented reality in the educational realm has significantly enriched the learning experience, making it both captivating and secure.

The travel industry (Ahmad et al., 2023; Keller et al., 2015) holds immense potential for leveraging augmented reality to enhance travellers experiences. By incorporating AR technology, travellers perceptions of their physical surroundings can be transformed, offering them supplemental tourism experiences and opportunities for interactive engagement. For passengers with fear of flying, AR can be employed to help ease their anxiety. AR also proven invaluable as a personal tour guide, aiding travellers in navigating unfamiliar destinations. With AR applications, tourists can access real-time information, historical facts, and interactive guides, enriching their exploration and understanding of new places. Furthermore, hotels have embraced AR in their booking applications to enhance customer satisfaction. Travelers can take virtual room tours, providing them with a realistic view of accommodations before making their reservations. Overall, the possibilities for augmented reality in the travel industry are vast, promising to revolutionize how people experience and interact with the world during their journeys.

The entertainment industry (MacIntyre & Hannigan, 2003) is embracing augmented reality with multiple exciting applications. As we look to the future, films and TV shows promise to become more immersive and captivating, offering viewers an enhanced experience. Interactive TV is one area where AR can shine, as it enables the provision of additional information directly overlaid on the screen, enriching the viewing experience with valuable insights. In the field of eSports, augmented reality transforms the way shows are presented, turning them into interactive experiences. Viewers can actively participate, blurring the lines between spectators and participants, leading to a more engaging and enjoyable entertainment spectacle. Moreover, AR technology is proving to be a boon for inclusivity in the entertainment world. Britain's National Theatre is currently testing an AR app designed to assist those who are hard of hearing. With this solution, subtitles are projected directly onto the play, allowing visitors to immerse themselves fully in the story without being distracted by external devices. As AR technology advances, we can anticipate more and more innovative applications and a truly immersive future for entertainment.

In Industry 4.0, AR (Funk et al., 2015; Manca et al., 2012; Marino et al., 2021; Reinhart & Patron, 2003) plays a crucial role in various use cases, primarily focused on supporting technicians in their everyday work environment. By leveraging augmented reality, workers can access invaluable assistance and guidance throughout their tasks. One of the significant applications of augmented reality in Industry 4.0 is providing technicians with step-by-step visualizations of the

procedures they need to perform. This feature allows them to follow instructions seamlessly and accurately, leading to improved efficiency and reduced errors. Furthermore, AR enables technicians to receive real-time visual instructions from remote experts through remote assistance systems. This capability proves particularly useful when workers encounter complex or unfamiliar issues that require expert guidance. With AR, experts can virtually guide technicians through the resolution process, regardless of geographical distance, thereby enhancing problem-solving capabilities. Overall, augmented reality in Industry 4.0 is a powerful tool that empowers technicians with comprehensive support and enhances their productivity by providing valuable visual information and real-time remote assistance. As the technology continues to advance, its potential for revolutionizing industrial processes and improving workforce efficiency only grows.

2.2.2 AR in industrial assembly lines

Written instructions in manual assembly lines are essential documents that guide workers through the step-by-step process of assembling products or components. These instructions are typically printed on paper or displayed on screens within the assembly area. Currently, the most common form of instructions in manual assembly lines are written-based instructions. While written instructions are essential for guiding workers in manual assembly lines, they have several limitations that can impact their effectiveness and efficiency. Some of them include:

- *Limited Comprehension*: Written instructions may not be easily understood by all assembly line workers, especially in the case of those with limited literacy or language proficiency. This can lead to misinterpretation and errors in the assembly process.
- *Lack of Interactivity*: Written instructions are static and do not allow for interactive guidance. Workers cannot seek clarification or ask questions directly from the instructions, which can be a limitation when facing complex assembly tasks.
- *Inadequate detail*: Depending on the complexity of the assembly, written instructions may not provide enough detail to ensure precise and error-free assembly. Important subtleties or nuances of the process may be omitted.
- *Difficulty in updates*: Keeping written instructions up to date can be challenging, notably in the case of industries where products and processes frequently change. Outdated instructions can lead to errors and inefficiencies.
- *Limited feedback*: Written instructions do not offer real-time feedback to workers. If an error is made, it may not be detected until a quality control check, leading to potential rework and increased costs.
- *Visual complexity*: Some assembly tasks involve intricate spatial relationships and orientations that are difficult to convey through written instructions alone. Workers may struggle to visualize the correct assembly configuration.

- *Time-consuming*: Reading and interpreting written instructions can be time-consuming, which may slow down the assembly process, particularly for complex products.

To address such limitations, some industries are exploring the use of augmented reality (AR) and digital work instructions. AR systems can provide interactive, real-time guidance, offer 3D visualizations, and allow for hands-free operation, potentially overcoming many of the limitations associated with traditional written instructions. The AR technology finds diverse applications in the industrial field, where companies and industries across various sectors harness its potential for training and productivity enhancements.



Figure 2-3. Operator on assembly lines in an industry

AR systems in the industry involve using specialized goggles, glasses, or smartphone apps to overlay digital information onto a plant worker's real-life view (Figure 2-3), which can help to increase productivity, efficiency, and safety. With guided AR solutions, work instructions can incorporate quick inspections at any step without hindering cycle time. AR has been applied to manual assembly station planning (Reinhart & Patron, 2003), product assembly guidance (Sääski et al., n.d.; Yuan et al., 2008; Zhang et al., 2011), assembly workplace design (Ong & Wang, 2011), digital virtual prototype augmentation with physical products (Halttunen & Tuikka, 2000), data glove-based virtual assembly (Valentini, 2009), physical manual replacement with augmented virtual contents (Wiedenmaier et al., 2003), and human–robot interaction design for safety and efficiency (Neuhöfer et al., 2010; Odenthal et al., 2012). Many of the research efforts focused on AR-assisted assembly training and guidance. (Hořejší, 2015) have compared classical approaches in assembly lines versus AR-based solutions and suggested an increase in the operator's training speed by 43%. In the same time, the number of manipulation errors was also reduced.

Figure 2-4 presents an overview of how AR can be used in industries. There are different scripting interfaces such as Unity and Visual studio which allows you the create the application according to the use case of interest. With the help of the AR toolkit an open-source library that calculate the real camera position and orientation relative to square physical markers or natural feature markers in real time. Once the real camera position is known a virtual camera can be positioned at the same point and 3D computer graphics models drawn exactly overlaid on the real marker. We can create several actions and the integration of audio and visual assets enhances the immersive nature of AR experience according to the use case.

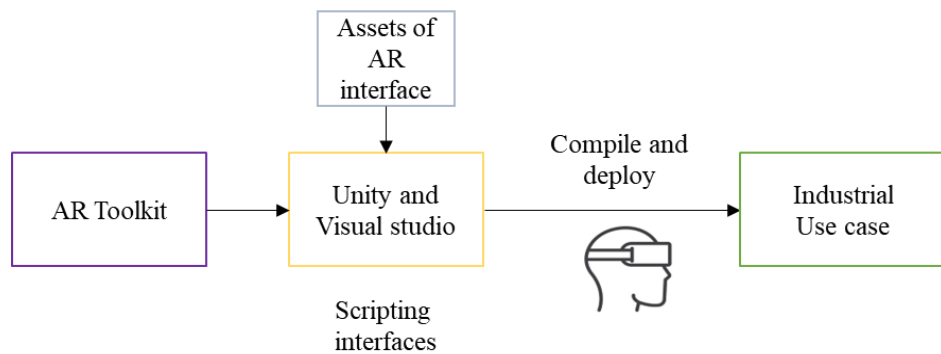


Figure 2-4. AR Workflow in industries

Human-machine interface (HMI) methodologies encompasses the communication and interaction that occurs between a human operator and a machine or system, facilitated through a user interface, which in our context is augmented reality (AR). The HMI approaches play a crucial role in enhancing the overall performance of the applications in which they are deployed. Notably, several industries have already embraced augmented reality for their assembly lines, as evidenced by (Hahn et al., 2015; Syberfeldt et al., 2015).

The design and configuration of an assembly line are intricately tailored to the specific manufacturing processes required for each product component and the final assembled product. It is important to note that this design can vary significantly not only from one industry to another but also among different assembly lines within the same industry. This adaptability underscores the need for flexible and customizable HMI solutions to accommodate the unique demands of each assembly line and industry.

An assembly line is designed to be highly efficient and cost-effective. A classical assembly line (Krüger et al., 2009; Lapierre & Ruiz, 2004) consists of a set of several workstations (Figure 2-5). At each workstation, there are different bins/containers into which components are arranged. These containers serve as repositories for the parts and materials needed to build the final product. The semi-assembled product begins its journey at the initial workstation and proceeds to advance from one workstation to the next in a systematic manner. This progression continues

until the entire assembly process is completed at the final workstation. The allocation of operators at each workstation depends upon the complexity of the assembly being undertaken. Depending on the intricacy of the tasks involved, one or more operators may be assigned to work at each workstation to execute the necessary steps in the assembly process. This dynamic approach ensures that the assembly line can adapt to the specific requirements of the products being manufactured, whether they are relatively simple or highly complex.

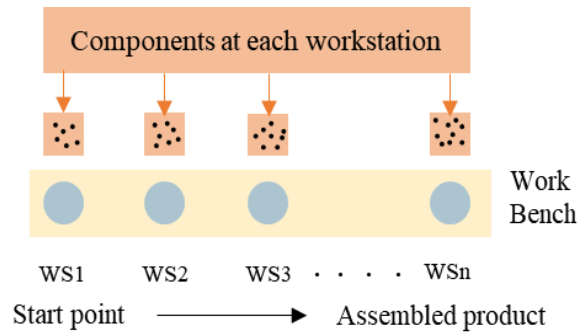


Figure 2-5. Typical assembly line

Figure 2-6 presents the example view of assembly line in a real time industry (elm.leblanc) specialized in the assembly of boilers. We can see containers, assembly carts, screens, work stations and multiple components.



Figure 2-6. Assembly line in a real industry (photo from elm. Leblanc factory)

2.2.3 Media in AR

Media in augmented reality (AR) refers to the various types of digital content that are used to enhance and augment the real-world environment with virtual elements. The media assets are overlaid onto the physical world through AR devices like smartphones, tablets, smart glasses, or headsets, creating a blended experience where the digital and physical realms coexist.

The common types of media used in augmented reality include:

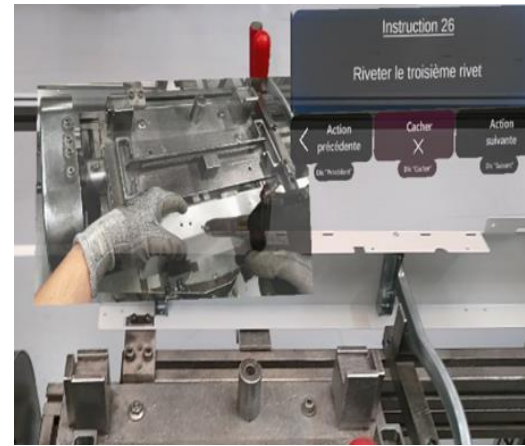
- *Images*: Static images can be super-imposed onto real-world objects, providing visual information or context. For example, AR can display product information when users point their devices at a specific item in a store.
- *Videos*: Augmented reality can incorporate short video clips to provide dynamic and interactive content. For instance, AR advertisements may come to life with animated characters or product demonstrations.
- *3D models*: 3D virtual objects can be placed in the real environment, allowing users to examine them from different angles and interact with them. 3D models are often used in product visualization, architecture, and educational applications.
- *Animations*: AR animations add movement and interactivity to static scenes, making the experience more engaging and immersive. AR games often use animations to bring characters and scenes to life.
- *Audio*: Sound effects, music, or narration can be integrated into AR experiences, enhancing the overall sensory immersion and creating a more realistic and engaging environment.
- *Text*: Textual information can be overlaid onto objects or locations in the real world, providing context or instructions. AR can be used for language translation, guiding users through a museum exhibit, or displaying information about landmarks.
- *Live data*: AR can incorporate real-time data, such as weather updates, stock prices, or live sports scores, to provide users with up-to-date information relevant to their current location or interests.

The combination of such different types of media in augmented reality enables a wide range of applications across industries, including gaming, entertainment, education, marketing, healthcare, and more. As AR technology continues to evolve, the possibilities for creative and practical uses of media in augmented reality are expanding rapidly.

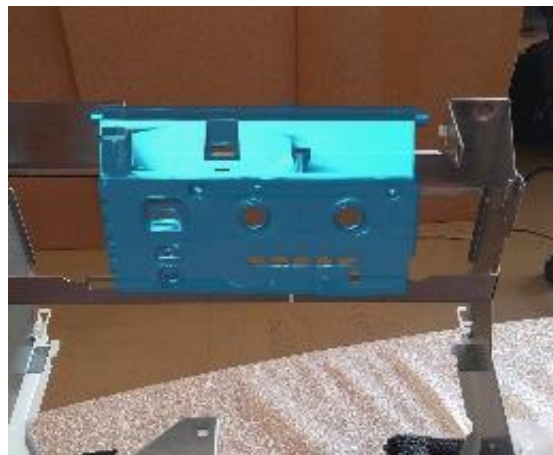
Overall, AR instructions draw upon a mixture of media contents, and the choice of assets depends on the tasks complexity and the desired user experience, with visual assets like images and 3D models often being the preferred choice for training and instructional purposes. Some of the media commonly used in AR are illustrated in Figure 2-7.



(a) Text based instruction



(b) Image based instructions



(c) 3D model based instructions

Figure 2-7. Types of media utilized in AR applications

2.3 Case study

One limitation of research in assembly/training is that instead of using real-world tasks in industries, researchers tend to evaluate the scenario with assembly of LEGO models (Hou et al., 2013; Hou & Wang, 2011) on a general basis, due to which some important scenarios are often missed or go undiscussed. Such studies are unreliable since real-world assembly tasks are much more complex, and the assembly line environment is quite different. Notably, the need for workstation configuration is known only when the research is conducted in a real-time environment with real AR equipment. (Swan & Gabbard, 2005) found that only 8% of published AR research papers included formal evaluations.

For the very same reason, before establishing the use cases in order to have a better understanding of the scenario, we have performed a real-life case study within the industrial use case of elm.leBlanc, which concerns the assembly of boilers.

A complete boiler had been assembled (Figure 2-8) with the help of traditional manual instructions that are available at the assembly line. Note that this is not an evaluation of manual assembly instructions versus augmented reality. Instead, our goal is to understand the various tasks performed in the assembly lines. The key findings of this study are the following:

- Images, on their own, have limitations in providing comprehensive guidance, particularly when dealing with intricate instructions that involve specific product orientations. This means that for complex tasks, images may not effectively convey the precise steps or positioning required for successful execution.
- A single assembly mistake can have far-reaching consequences in the manufacturing process. It has the potential to render the final product defective, making it entirely unusable if the error goes undetected. Even if the mistake is identified, it can still lead to increased production time and costs, as the product often needs to be dismantled and reassembled correctly. This not only impacts efficiency but also affects the overall quality of the manufactured goods, leading to potential delays and increased expenses in rectifying the error. Hence, the prevention and early detection of assembly errors are crucial to maintain production efficiency and product quality in industrial settings.
- Assigning reference numbers to each part of an assembly is a common practice in manufacturing and industrial settings. This system helps in uniquely identifying and cataloguing individual components, making it easier to manage large inventories of parts and ensuring consistency in product assembly. Furthermore, using the same reference number for a part across multiple products with varying viewing scenarios is a practical strategy. For instance, consider a multi-purpose screw that is widely used in various industries. This screw may be utilized in different products, each with its own assembly requirements and orientations.
- When it comes to 3D models and their eventual simplification, it is of crucial importance to ensure the preservation of every feature within a component. This ensures the adaptability and versatility of the simplified model, allowing it to seamlessly integrate into diverse assemblies across various workstations and assembly lines.
- In the process of creating instructional steps for assembly lines, there are essential preparatory and follow-up stages. The initial phase involves configuring the workstation to align with the specific assembly task, ensuring that all the necessary tools, components, and instructions are in place for a seamless workflow. Following this, a post-processing step is applied, often referred to as the validation of assembly. Any errors or discrepancies are addressed, ensuring that the final product meets the required standards and specifications. These two stages, *workstation configuration* and *assembly validation*, are integral to the overall efficiency and quality control of manual assembly processes.



Figure 2-8. Assembled boiler for our case study

Based on these observations, we have identified three main uses cases (Figure 2-9) that concern the utilization of AR inside industrial assembly lines.

1. *Workstation setup/configuration*: Before creating instructions in AR, it is crucial to configure the workstation accurately. This step ensures that the operator has precise information about the layout and positioning of components and tools on the assembly line. A well-configured workstation sets the foundation for effective AR-based training.
2. *Creation of instructions*: Once the workstation is properly set up, instructions can be developed using AR. These instructions are presented to the operator through the AR interface, providing step-by-step guidance on assembling products or performing specific tasks. AR allows for the seamless integration of digital instructions with the physical environment, enhancing the learning experience for trainees.
3. *Validation of the operators tasks*: After the operator receives AR-based training and performs tasks, the system can validate their performance. This step is essential to ensure that the operator follows the instructions accurately and completes tasks without errors. The validation process helps assessing the effectiveness of the training and identifies areas where further improvement or clarification may be needed.

By focusing on these three key areas, companies can maximize the benefits of AR technology for assembly line training. Incorporating AR technology into assembly line training programs aligns with the modernization of industrial processes. It not only improves the skills of operators but also contributes to overall operational efficiency and product quality.

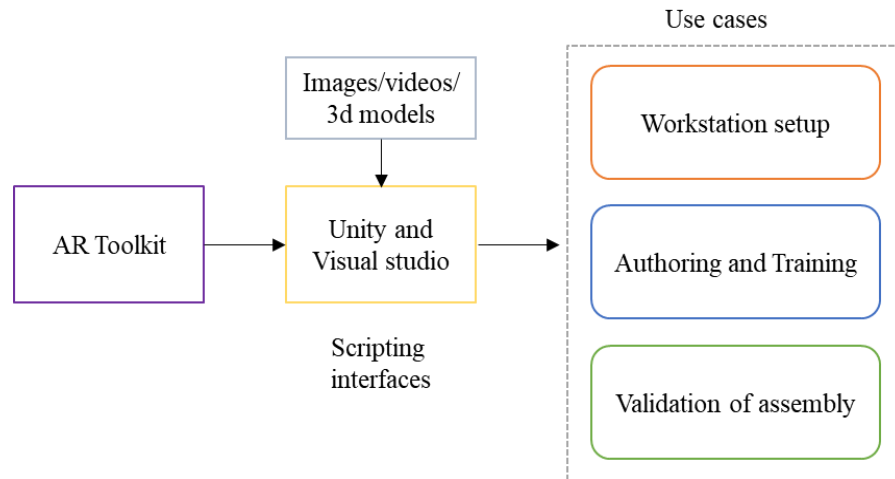


Figure 2-9. The three main use cases of AR in industrial assembly lines

2.4 Visual AR assets for different use cases

Let us now analyse how the various visual assets that can be considered are adapted for these three main uses cases.

For visualization purposes, media plays a vital role in understanding the scenario. Augmented reality assets can be grouped based on their creation difficulty and power of expression. The first assets include text, audio, images, and video, while the second includes 3D models and animations. The difficulty in setting up a workstation using such assets is generally inversely proportional to the power of expression. There is a definite trade-off between difficulty in the creation of AR instructions and the comprehensiveness in industrial applications. In the case of the workstation configuration, the operator has to identify the real-time components and the media that is displayed in AR. For this reason, we eliminate the text and audio assets in the considered scenario and explore only visual assets (Gattullo et al., 2020, 2022).

Irrespectively of considered use cases, let us discuss the general issues of using images, videos, and 3D models in the AR system. The most important factors that affect the effective AR system are occlusion, the level of communication and authoring costs.

1. Occlusion

Occlusion refers to the phenomenon where virtual objects or visual assets in the AR system block or hide parts of the real world. It occurs when digital content is overlaid on the physical environment, and it can pose significant challenges to the user experience. Occlusion can affect the effectiveness and realism of AR instructions in various ways:

-
- *Clarity of instructions:* When virtual objects occlude important real-world elements, it may lead to confusion and difficulty in following instructions. Users may have trouble understanding the precise spatial relationships between virtual and real objects.
 - *Immersion:* Occlusion issues can break the sense of immersion in the AR experience, as the seamless blending of digital and physical elements is disrupted.
 - *User safety:* In certain situations, occlusion can lead to safety concerns. For example, important warning signs or physical hazards may be obscured, potentially endangering users.

2. Level of communication

- The level of communication in an AR system refers to how effectively the instructions and information are conveyed to the users. This depends on the clarity, accuracy, and intuitiveness of the visual assets used in the AR experience. The visual design of images, videos, and 3D models must be clear and straightforward, ensuring that users can quickly grasp the intended message or task.

3. Authoring cost

Authoring cost refers to the effort, time, and resources required to prepare and pre-process visual assets for creating AR instructions. This factor influences the feasibility and scalability of implementing AR in various applications:

- *Content creation:* Developing high-quality images, videos, and 3D models demands skilled personnel and appropriate software tools. This can be time-consuming and resource-intensive, especially for complex scenarios.
- *Asset optimization:* Preparing visual assets for AR often involves optimizing their size, resolution, and format to ensure efficient rendering and minimal impact on the AR system's performance.
- *Iterative processes:* The iterative nature of creating AR content may result in multiple rounds of refinement, further adding to the authoring cost

Table 2-1 illustrates how images, videos, and 3D models align with the different factors that influence the effectiveness of AR systems within industrial assembly lines. While images and videos tend to have lower production costs in comparison to 3D models, it is essential not to underestimate the depth of communication and engagement that 3D models offer. This assessment likely serves as a valuable reference for decision-makers when choosing the most suitable media format for their AR-based training and operational processes. It underscores the trade-offs between cost-effectiveness and the comprehensiveness of communication in the context of industrial assembly line applications. These issues are further detailed in the following sections.

Table 2-1. Effectiveness of visual assets inside AR

	Occlusion	Level of communication	Authoring cost
Images	Yes	<i>Average</i>	<i>Low</i>
Videos	Yes	<i>Average</i>	<i>Low</i>
3D models	No	<i>Very Good</i>	<i>High</i>

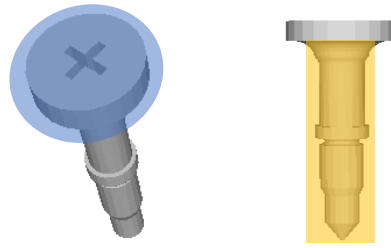
2.4.1 Images and videos

This section discusses how images/videos are used in different use cases.

2.4.1.1 Images and videos for workstation configuration

Workstation configuration is a very important step to perform any kind of tasks inside industrial assembly lines. During the configuration, the assets that need visualization are loaded inside the AR device and are arranged into their bins one by one. The main reason of applying a configuration process is that the operator is directed towards the bin from which he has to pick up the component while creating the instructions and also following the assembly instructions. In order to train the assembly line operators with the help of AR-based instructions, configuring the workstation is the first step to be achieved.

Image represents the most commonly used media in AR for visualization since it is relatively easy to create and provides a good understanding. However, for the workstation configuration, if we use only images, it is difficult to differentiate between different components of the same type. Let us consider one of the most used components in industries which is the screw, a very simple component as it may come in different shapes and sizes (Figure 2-10). In most of the cases, it is impossible to identify the correct component with a single image. In addition, several other factors can impact the understanding of the media, such as the viewing angle and distance to the considered component.



a) head of the screw b) tail of the screw

Figure 2-10. Various views of a screw

This problem is illustrated in Figure 2-11, for different screws. In the case illustrated in Figure 2-11a and b, the two screws have the same type of heads, but differ in the tail sizes. If they are presented to an operator using an image, it is impossible to identify the component only with the help of the head of the screw. Particularly in this case the understanding of the operator can vary based on the distance from which the image is captured. In the case in Figure 2-11c the two screws have almost the same tail size, but because of the angle of view the head type cannot be identified. In all cases, it is impossible to identify the right component.

Addressing the challenge of capturing images from various angles by including multiple pictures corresponding to different views of a component is indeed a potential solution. However, this approach presents several limitations.

The first one relates to the occlusion of the original scene. When multiple images are overlaid in an augmented reality (AR) environment, it is essential that they do not obstruct or interfere significantly with the user's actual surroundings. If the images used for instruction cover too much of the real-world environment, it can be distracting and reduce the user's ability to perform tasks accurately.

The second limitation is associated with information overload during the workstation configuration process. In scenarios where the assembly line or workstation is extensive, including all the components simultaneously in the AR instructions can overwhelm the user. Managing a large number of components and their placement within bins or containers can become complex and challenging for operators, potentially leading to confusion and errors.

These limitations also extend to the use of videos in AR instructions, as they share similar issues related to occlusion and information overload. Therefore, a good balance needs to be identified between providing comprehensive instruction and maintaining a clear, unobstructed view of the user's environment to ensure the effective use of AR in industrial assembly processes.

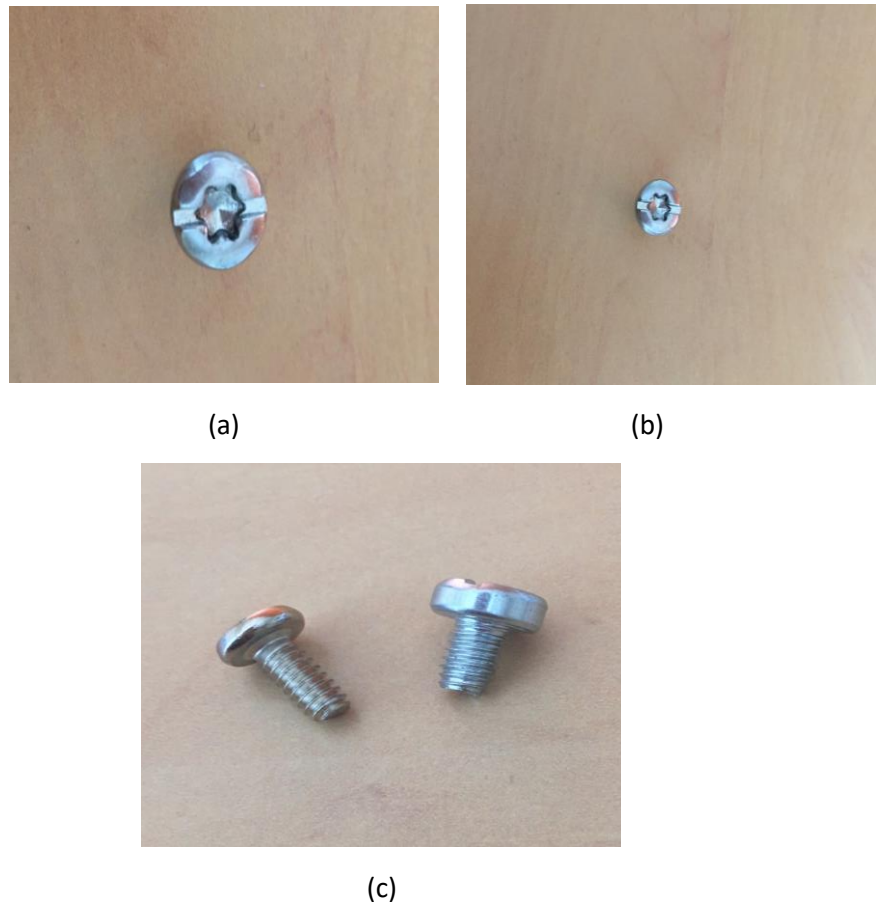


Figure 2-11. Evaluating usage of images in AR

2.4.1.2 Images and videos for instruction creation

Images and videos are undeniably popular choices for creating instructional content within augmented reality (AR) environments due to their ease of production and cost of authoring. However, they come with certain limitations, with one of the primary challenges being occlusion, which can disrupt the seamless integration of digital instructions into the real-world scene.

For instance, in Figure 2-12 , consider an instruction where a rivet needs to be placed on the left side of a white panel. In AR applications like HoloLens, the precise placement of images or videos can be tricky to control. Consequently, this can lead to issues where the digital content interferes with the operator's view of the real-world object, making it challenging to see the original counterpart in its actual context.

Moreover, there are scenarios where assembly tasks occur in concealed or hard-to-reach areas within a structure. In such cases, relying solely on images can prove insufficient for operators,

as these media formats may not adequately guide them through intricate processes that are hidden from plain view.

The limitations discussed underscore the need for a thoughtful selection of the media format when implementing augmented reality (AR)-based instructions in industrial assembly lines. While images and videos have their advantages in terms of simplicity and cost-effectiveness, they may not be sufficient in scenarios where precise spatial alignment and the visibility of real-world objects are of paramount importance.

In situations where the accurate placement of components or alignment with existing structures is crucial, 3D models can offer a significant advantage. These models can provide a more comprehensive and accurate representation of the assembly process, allowing operators to visualize the task from various angles and orientations. This level of detail and spatial awareness can lead to improved precision and reduced errors during assembly.

Moreover, for complex assembly tasks that involve intricate interactions between components, more advanced AR techniques, such as spatial mapping and object recognition, can be employed. Such techniques enable AR systems to understand the physical environment and overlay digital instructions or models seamlessly. This approach ensures that the digital content aligns precisely with real-world objects and provides operators with a clear and contextually relevant guidance.



Figure 2-12. AR instruction with image

2.4.1.3 Images and videos for validation of assembly

The success of any application, especially in industrial assembly lines, hinges on the precision and accuracy with which tasks are executed, as well as the ability to detect and rectify errors

promptly. This verification step, often referred to as quality control, is of utmost importance (Alves et al., 2021). However, in the context of Augmented Reality (AR) within industrial assembly lines, a critical concept is validation, as well as real-time feedback provided step by step or per instruction (Gómez et al., 2016). This real-time validation process is essential for confirming the correctness of workstation configurations, authoring and creating instructions, and training operators.

To illustrate the significance of real-time validation, let us consider the assembly of a boiler on an assembly line. Even a single error during the assembly process can result in significant production delays and associated costs if quality control is only performed at the end of the products assembly. Augmented reality systems offer a solution by enabling the real-time identification of improper assembly and providing immediate feedback to the operator.

In such scenarios, relying solely on images may fall short in providing comprehensive information about the assembly at a particular workstation. Images may not capture the dynamic and real-time aspects of the assembly process, making it challenging to verify the placement of instructions and promptly detect errors. While it is quite hard to perform the visual assessment of assembly using images, 3D model can show the exploded views of products showing assemblies and sub-assemblies with benefits of orientation to validate the assembly done.

2.4.2 3D CAD models in AR for workstation setup, creation of instructions and validation

Bégout *et al.* (Bégout et al., 2022) considered the issue of using AR in assembly lines for workstation configuration. They proposed an AR authoring tool involving 3D CAD models to configure the workstation before creating the instructions. Lavric *et al.* (Lavric et al., 2021b) have explored various assets inside AR in the context of training operators by creating instructions for assembly. The paper discussed different assets for creating instructions and some operator evaluations. Various other works like (Gattullo et al., 2020, 2022) explain the growing use of 3D CAD models as primary visual assets in AR for industrial assembly lines.

When considering the use of 3D models for all three mentioned use cases, two key advantages prominently come to the forefront: the absence of occlusions in the original scene and the heightened level of communication. Their dynamic and non-intrusive nature makes them a powerful tool for improving efficiency, accuracy, and user engagement in industrial assembly lines.

However, preparing 3D models for AR is time-consuming and requires advanced technical expertise. Usually, when the 3D CAD models are created, they are stored in a database with an associated reference number. They are then downloaded through a web service whenever there is a need for the model (Figure 2-13). Since different 3D development platforms can create 3D

CAD models, there are different file formats. The pre-processing of the 3D models includes converting the models into a format supported by the AR device, or the game engine (e.g., Unity) used to create the AR experience. Whatever the format adopted, the 3D CAD models are most often represented as 3D meshes.

The 3D modelling process in industrial assembly lines offers a level of depth and precision that surpasses the capabilities of rough sketches or 2D visual assets. One of its significant advantages lies in the enhanced control over intricate details. Unlike 2D representations, 3D models can capture the full complexity of physical components, allowing for meticulous adjustments in size and proportions. This precision ensures that the 3D model aligns flawlessly with its real-world counterpart, contributing to seamless assembly processes.

Furthermore, 3D modelling facilitates the application of geometric transformations with remarkable ease. Operations like rotation and zoom can be executed seamlessly, providing operators and assembly line workers with intuitive tools to manipulate and examine components. This capability proves to be invaluable across all three primary use cases in industrial assembly lines, where precise adjustments and thorough examinations are essential for efficient and error-free assembly procedures. In essence, 3D modelling empowers industries to achieve a level of accuracy, control, and adaptability that greatly enhances their assembly line operations.

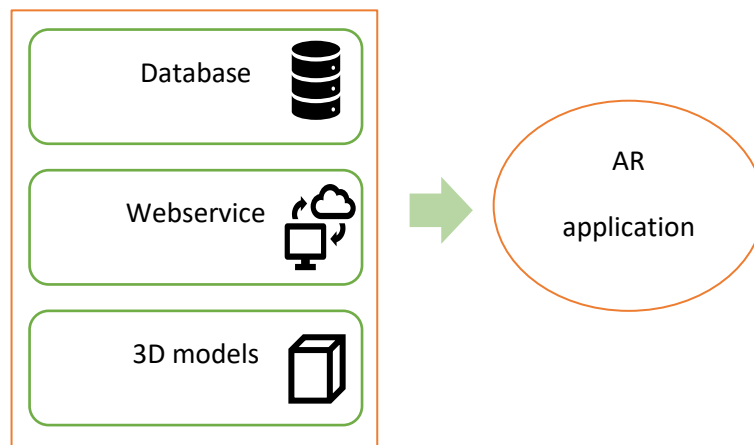


Figure 2-13. 3D model workflow in industries

In industries, the CAD models produced are very detailed, hence very heavy in terms of bandwidth and storage requirements. Naturally, as the complexity of CAD models increases, the visual approximation of real-world objects improves, but the rendering process on AR devices becomes computationally prohibitive. There is a strong need to define an optimal trade-off between the cost of processing these models and the quality of the visual approximation. The crucial challenge is to preserve the quality of the models while ensuring a fluid rendering on the AR devices like HoloLens at the usual rates of 40-60 frames per second(fps) (Yuan et al., 2008).

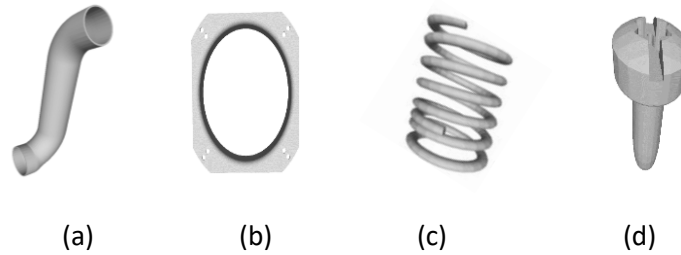


Figure 2-14. Examples of simple CAD models

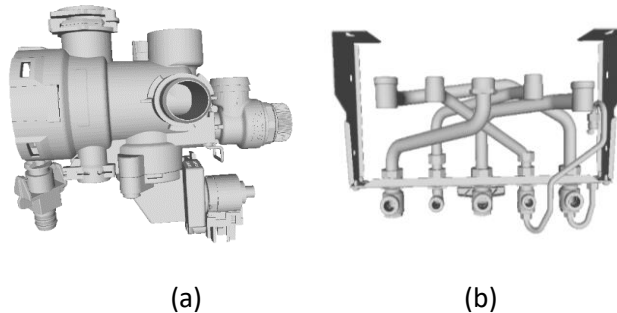


Figure 2-15. Examples of complex CAD models

Figure 2-14 and Figure 2-15 represents different types of 3D CAD models where we can see simple structures like pipe, screw and also complex structures like hydraulic, multi-faucet which have higher level of details. Managing simple 3D models in augmented reality (AR) applications is typically straightforward and does not pose significant challenges. However, when it comes to handling complex 3D models, the task becomes considerably more demanding. Loading and manipulating these intricate models within an AR environment can be accompanied by various difficulties and face performance issues.

The situation becomes even more critical if multiple components have to be simultaneously visualized. Thus, if a workstation has ten complex components to be assembled, rendering them (or even loading them) without latency is almost impossible. Our work mainly deals with the issue 3D models for different use cases inside industrial assembly lines. The solution proposed in based on 3D model simplification techniques.

2.4.3 Solution: 3D model simplification

One possible solution to the issues of 3D model complexity would be the remote rendering of the 3D models in real time, for example in a cloud platform. However, because of connectivity issues, this approach is not always feasible. In addition, most industries do not allow the connection to external networks because of privacy, security and reliability issues.

In our case, we have introduced an alternative solution-based 3D mesh simplification, as illustrated in Figure 2-16. The approach involves the simplification of 3D models subsequent to their acquisition from a CAD model database. This methodology is known as *3D model preparation*, a crucial step in the process of readying the models for seamless integration and utilization within Augmented Reality (AR) systems.

The 3D model preparation phase is pivotal as it ensures that the CAD models are optimized and tailored to meet the specific requirements and constraints of AR systems, thereby facilitating a smoother and more effective user experience in industrial assembly lines.

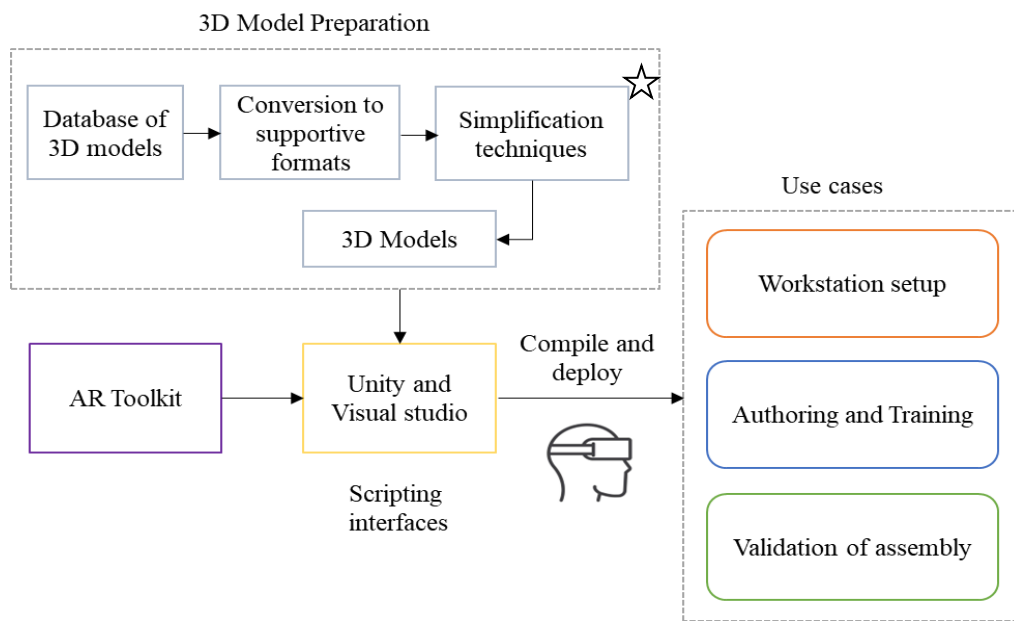


Figure 2-16. Complete workflow of 3D model-based instructions in industrial assembly lines

2.5 Summary

In this chapter, we have focused on the issue of AR application in industrial assembly lines. To illustrate the practical implications, we have conducted a case study in a real-world industry (elm.leBlanc) specialized in boiler assembly. Through this case study, we have established different use cases and explored the application of visual assets in each scenario.

While addressing the use of images and videos, we have pointed out their limitations in certain contexts, prompting us to seek more suitable alternatives. Consequently, we have studied the advantages of using 3D models for various applications. The related challenges concern the presence of highly detailed and complex models. Their utilization in real life scenarios requires the specification of dedicated 3D model simplification solutions.

A key area of focus in recent years has been the utilization of simplified 3D models for different applications (Hasselgren et al., 2021; Li et al., 2013; Liu et al., 2015). CAD model simplification, in particular, has been an active research area due to the intricate functionalities these models possess.

In the following chapters we will extensively delve into the process of CAD model simplification. We will thoroughly examine various techniques both theoretically and empirically, aiming to find effective ways to handle highly detailed CAD models for practical applications in Industry 4.0 and assembly lines.

Chapter 3

STATE OF THE ART

3.1 Introduction

With the evolution of modern acquisition and 3D modelling systems, it becomes possible to generate increasingly complex mesh models. Simplification of 3D models has been a vast research area over the past decade.

In this chapter, we analyse the state-of-the-art simplification techniques that can be used for real world 3D CAD models in different applications. In particular, the suitability of the simplified 3D CAD models for the AR training in the assembly process use case is examined. Irrespective of how 3D CAD models are generated, the simplification strategy should be able to produce results that can be directly exploited in the considered use case.

First, the different families of CAD models simplification techniques are identified. For each technique, we present the underlying principle and main results achieved, analyse related advantages and limitations as well as the suitability with respect to the criteria involved in our use case.

One of the most important application constraints concerns the strict preservation of all the model features during simplification, since we cannot predict a priori which functional part of the model will be used in the assembly process. When it comes to the industries with assembly lines, the same model may be used at different workstations for different purposes. The functional parts of the models that are used during the assembly may differ from one workstation to another. Hence all the functional parts of the CAD model should be fully preserved.

Let us consider a simple assembly task where a screw (Figure 3-2) is being put in the hole of a plate (Figure 3-1). The traditional instructions detail where and how the assembly should be performed (Figure 3-3). The assembly line instructions are as follows:

Instruction 1: Pick the plate with reference number 801

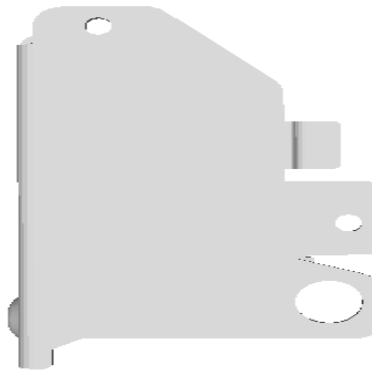


Figure 3-1. plate

Instruction 2: Pick the screw with reference number 802

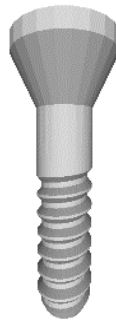


Figure 3-2. screw

Instruction 3: Put the screw in the indicated hole in the plate

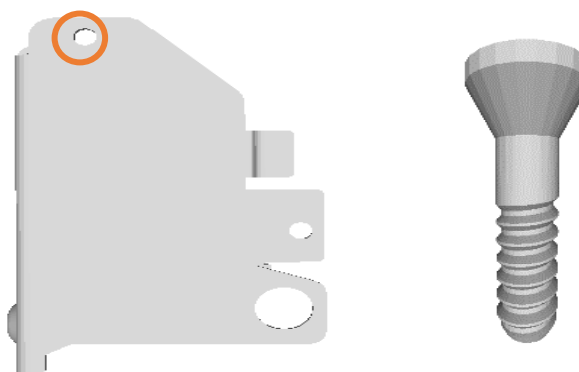


Figure 3-3. Assembly process

We have considered simplified models obtained with the MeshLab platform (quadric edge collapse decimation) (Cignoni et al., 2008), represented in Figure 3-4. After 50% of simplification, the functional part hole where the actual assembly takes place is no longer visible. Creating instructions with such a simplified model becomes impossible. Even though the hole in the reference is a small part and corresponding simplification error may be relatively negligible, since it represents the most important part of the assembly, the simplified model cannot be used any longer in the creation of instructions. If we go on and perform further simplification (Figure 3-4), which distorts the shape of the model too, the operator is no longer sure if the simplified model is the one that needs to be used in reality.

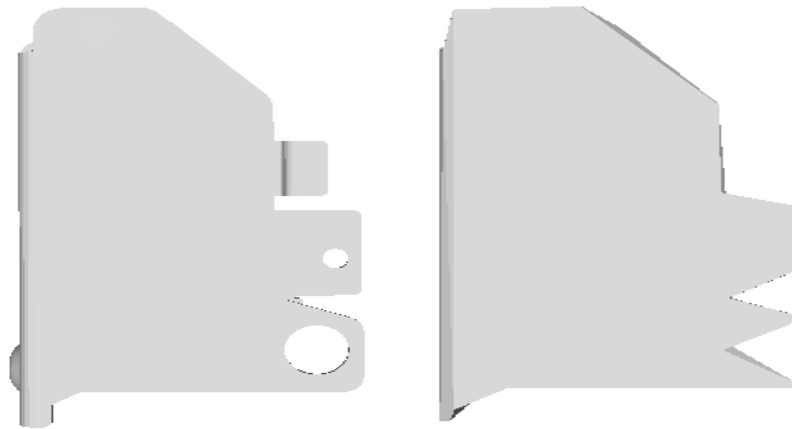


Figure 3-4. Examples of simplified models using MeshLab

Therefore, when employing simplified CAD models in the assembly process, certain critical constraints must be upheld:

- **Visual Fidelity:** The simplified models must closely resemble the original models in visual appearance. This fidelity ensures that operators can readily identify and work with the models in the AR environment.
- **Feature Preservation:** Every feature present in the original model should be retained in the simplified version. This includes intricate details and components. This comprehensive preservation is essential since it is often impossible to predict which parts will be utilized at specific workstations along the assembly line.

3.2 Analysis of simplification techniques

CAD model simplification techniques involve three families of methods, including:

- Feature-based simplification (Elliott et al., 2007; Lee, 2005; Lee & Lee, 2012; Mun & Kim, 2017; Shuming Gao et al., 2008; Yoon & Kim, 2016),

-
- Dimension reduction based (B-rep) (Kwon et al., 2019; Sun et al., 2010; Zhu & Menq, 2002) and
 - Mesh-based simplification (Agarwal & Suri, 1998; DeHaemer & Zyda, 1991; Hua et al., 2015; Pan et al., 2001).

Let us consider the example in Figure 3-5 as a reference test model and analyse the effects of each simplification technique.



Figure 3-5. Example CAD model - Holder

3.2.1 Feature based simplification

In feature-based modelling, the fundamental building blocks correspond to features, which represent shapes with specific engineering meanings. Such features can be either additive or subtractive, and the final shape is constructed by incrementally applying the features in a systematic manner. Essentially, a model is assembled by stacking features on top of each other. Our test model in Figure 3-5, is composed of four subcomponents. Among them, two are additive features, while the remaining two are subtractive (Figure 3-6).

Feature-based methods are widely used in CAD modelling, and when it comes to simplifying such models, one straightforward approach is to remove features. However, the criteria for removing features can vary, including factors such as importance, order, and size. When importance is considered as a metric, features are reordered based on their significance within the model. Simplification is then achieved by systematically removing one feature at a time, starting with the least important feature. Another metric involves using the order in which features are originally arranged or assembled to create the final model. Features are simplified by preserving this original order and removing them accordingly. The third metric focuses on the size of subcomponents within the features. Features are rearranged based on the sizes of their constituting elements, typically in ascending order. After this rearrangement, the simplification process involves suppressing features one by one until the model achieves the desired Level of Detail (LOD).

The different feature-based model simplification approaches provide flexibility in tailoring the simplification process to specific modelling requirements and objectives. Depending on the

context and goals of the CAD modelling project, one of them can be used to achieve the desired level of simplification while retaining essential design intent and functionality.

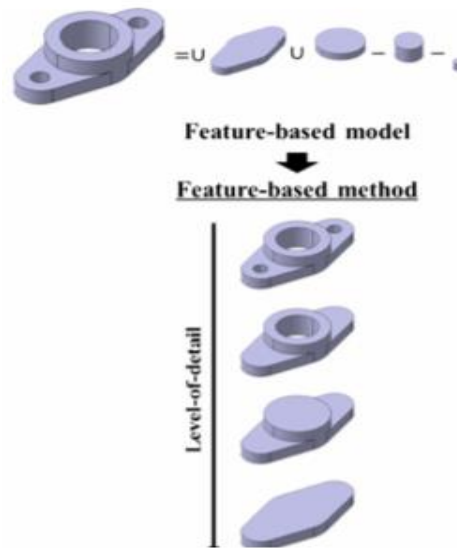


Figure 3-6. Feature based simplification

While feature-based model simplification methods offer advantages in terms of organization and systematic reduction of complexity, they do not always align with the criteria of preserving functional components essential for assembly processes. Such methods primarily focus on simplifying based on criteria like importance, order, or size, which might not inherently consider the functional significance of components within the model. However, preserving functional components is often a critical requirement in assembly line scenarios.

3.2.2 Dimension reduction-based simplification

Dimensional reduction in CAD modeling is based on a simplification strategy that entails the removal of one or more dimensions from a boundary value problem (Figure 3-7) (Herrera, 1981). However, it is important to note that the methods are not universally applicable and are typically well-suited solely for specific shapes and geometries. This limited applicability restricts their broad use in simplification techniques.



Figure 3-7. Dimension reduction-based simplification

Furthermore, controlling the level of detail in a simplified model can be a complex task, often yielding non-intuitive or challenging results. One significant advantage of employing dimensionality reduction methods is the substantial compression achieved. By transforming each region within the model into a lower-dimensional space, the model's size can be significantly reduced. However, this compression comes at a trade-off: the loss of functional details. While dimensionality reduction may excel in terms of compactness, it may not be suitable for scenarios where retaining functional details is critical.

As illustrated in Figure 3-7, a notable issue arising from the CAD model simplification process is the misfit between the simplified model and its intended position. Specifically, the simplified CAD model no longer aligns precisely with the dimensions and shape of the original model, resulting in a misfit that can disrupt the assembly process and overall functionality of the appliance. Additionally, the simplification process can lead to the loss of crucial features situated along the rim of the model. Such features, which may be essential for proper assembly, functionality, or aesthetic appeal, are compromised or omitted during simplification.

3.2.3 Mesh based simplification

Mesh simplification is a class of algorithms that transform a given polygonal mesh into another mesh with a reduced number of faces, edges, and vertices (Figure 3-8). The mesh simplification techniques are iteratively reducing the number of vertices/faces/edges of the model across successive simplification levels. The user has full control over the simplification of the level of details. As the simplification level increases, the features and the boundaries are distorted, yet they are not eliminated. The simplification process is usually controlled by a set of user-defined quality criteria that can preserve specific properties of the original mesh as much as possible. Being used in a wide variety of industrial applications, different metrics and criteria can be considered.

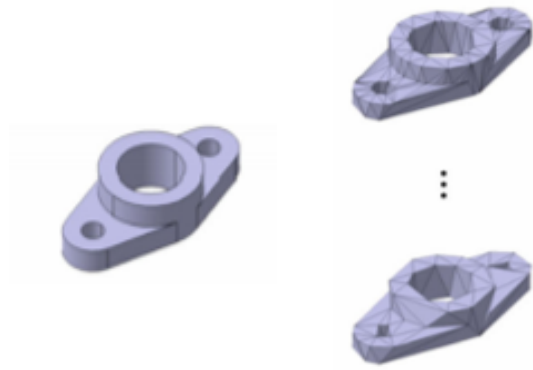


Figure 3-8. Mesh-based simplification

Mesh simplification methods are well suited for our use-case as they generally preserve the functional details and offer a good approximation of the overall shape, ensuring a good understanding of the components in AR. In addition, while using mesh-based simplification, a 3D CAD model is considered as one whole model and does not require any other details like assemblies and structures to be retained in order to perform simplification. For all these reasons, in our work, we have retained a mesh-based simplification approach.

Let us further analyse the state of the art, traditional 3D mesh simplification methods.

3.3 Analysis of mesh-based simplification techniques: traditional methods

Before going into details of the mesh-based simplification let us recall the important attributes of the mesh (Figure 3-9). It consists of three key components: vertices, faces, and edges:

Vertex: A vertex (plural: vertices) is a single point in 3D space. It is the most basic building block of a 3D mesh. Vertices define the mesh's geometry by specifying its 3D coordinates (X, Y, Z) in a given cartesian coordinate system. The coordinates determine the position of each vertex in the 3D world. In a mesh, vertices are connected by edges, which define the adjacency properties. Vertices can also store additional information, such as colors or texture coordinates (UV coordinates), which are used for shading and texture mapping.

Edge: An edge is a line segment that connects two vertices in a mesh. It represents the boundary or connection between two vertices. Edges define the shape and structure of a mesh by outlining its surfaces. While edges themselves do not typically have visual properties, they play a crucial role in defining the mesh's topology. Edges are used to determine which vertices are connected and which faces are formed by those connections.

Face: A face is a flat, polygonal surface defined by a closed loop of edges in a mesh. Faces are typically triangular (composed of three vertices) or quadrilateral (composed of four vertices), but they can have more sides for more complex shapes. Faces are responsible for the mesh visual appearance. They can have material properties, texture maps, and normals that affect how they reflect light and appear in a rendered scene. The arrangement and connectivity of vertices and edges within a face determine the shape of the face and the overall shape of the mesh.

In summary, vertices are points in 3D space, edges are the lines connecting those points, and faces are the flat, polygonal surfaces formed by closed loops of edges. Together, these components define the geometric and topological properties of a 3D mesh, making it possible to represent a wide range of 3D objects and shapes in computer graphics.

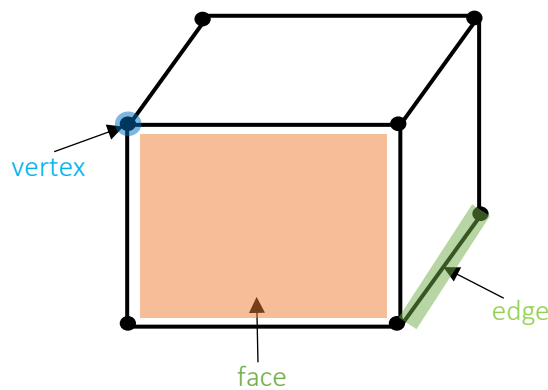


Figure 3-9. Attributes of a 3D model

Mesh-based simplification is generally based on elementary operators, which include either clustering or decimation of vertices, edges or triangles.

The clustering techniques aim to group vertices that are close together, simplifying the mesh by merging nearby vertices to reduce redundancy. On the other hand, decimation techniques iteratively remove vertices, edges, or triangles based on various criteria. This process reduces the complexity of the mesh while attempting to retain its overall shape and related features.

Overall, the choice of an appropriate simplification technique depends on the specific requirements of the application, as each approach comes with its trade-offs in terms of speed, accuracy, and LOD (level of detail) preservation. Evaluating the effectiveness of mesh simplification techniques involves considering multiple factors that need to be taken into account for selecting the most suitable approach in a given scenario.

Let us first review the main simplification operators involved.

3.3.1 Vertex removal

Vertex removal (or decimation) is a fundamental operation used to simplify 3D models, which plays a crucial role in various mesh-based simplification techniques (Franc & Skala, 2002; W. J. Schroeder et al., 1992). The process involves systematically reducing the number of vertices in a 3D model while preserving its overall shape and features (Figure 3-10). The main aspect that distinguishes vertex decimation algorithms concerns the selection strategy of the vertices to be removed. Among the various criteria that can be applied let us mention:

- The geometric error: Vertices with the least impact on the overall geometric error of the model are selected for removal. This ensures that areas with less detail are simplified first, while preserving critical features.
- The vertex attributes: Algorithms may take into account additional attributes associated with vertices, such as vertex normals curvature or color information, to prioritize the removal of less significant vertices while preserving important ones.

In all cases, a simplification threshold is needed in order to control the level of detail in the final simplified model. Vertices that exceed this threshold are targeted for removal.

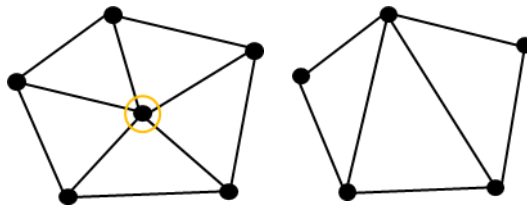


Figure 3-10. Vertex decimation operation

3.3.2 Edge collapse

Edge collapse, also referred to as edge decimation or edge contraction, is an alternative approach for simplifying a mesh. This technique involves iteratively collapsing edges to reduce the number of edges (Figure 3-11) while maintaining its overall shape and features (Botsch & Kobbelt, 2004; Cohen et al., 1998; Garland & Heckbert, 1997; Hoppe, 1996; Lindstrom & Turk, 1998).

Edge decimation methods vary primarily in two aspects:

- *Edge processing order*: Different algorithms adopt varying strategies for determining the order in which edges should be collapsed. Some common approaches include:
 - Geometric Error*: Like vertex removal, edges are selected for collapse based on their contribution to the overall geometric error of the model. This prioritizes the removal of edges with a minimum impact on the visual fidelity, ensuring that essential features are preserved.

b. Edge Attributes: Algorithms may consider additional attributes associated with the vertices connected to the edges, such as vertex normals or color information, to influence the selection of edges for collapse.

c. Simplification Threshold: A predefined simplification threshold guides the edge selection process, ensuring that only edges exceeding this threshold are targeted for collapse.

- *Placement of the new vertex:*

After an edge collapse operation, a new vertex needs to be placed to preserve the mesh's structure. Different strategies are employed for determining the position of the new vertex:

a. Midpoint Position: The new vertex is placed at the midpoint of the collapsed edge. This straightforward approach ensures the overall shape of the model is retained but may result in loss of fine details.

b. Quadric Error Metrics: More advanced algorithms use quadric error metrics to optimize the position of the new vertex, considering the geometric errors introduced by the edge collapse. This leads to a smoother and visually appealing simplified model.

c. Surface Area Minimization: Some methods aim to minimize the surface area distortion during edge collapse, which can result in improved preservation of critical features.

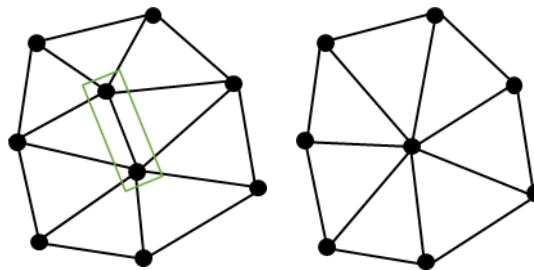


Figure 3-11. Edge decimation operation

3.3.3 Triangle collapse

Triangle decimation, as described in (Gieng et al., 1997; Hamann, 1994; Rossignac & Borrel, 1993), can be viewed as a variation of edge decimation, wherein the removal of a triangle is equivalent to three simultaneous edge collapse operations (Figure 3-12). This approach simplifies the mesh by collapsing all three edges of a triangle, resulting in a single vertex.

The key features of triangle decimation are as follows:

- *Simultaneous Edge Collapse:*

In triangle decimation, the operation involves the collapse of all three edges of a triangle simultaneously. This means that the three edges converge to a single vertex, reducing the number of vertices in the mesh.

- *Weighted Triangle Removal:*

Each triangle is assigned a weight based on various factors, such as its importance in preserving the overall shape and features of the model. The weighting process ensures that critical areas or significant details are retained while less essential triangles are prioritized for removal.

- *Surface Triangulation:*

Once a triangle is removed, the surface needs to be retriangulated to maintain a valid mesh structure. The triangulation process ensures that the newly created vertices are connected appropriately to form triangles that conform to the overall shape of the simplified model.

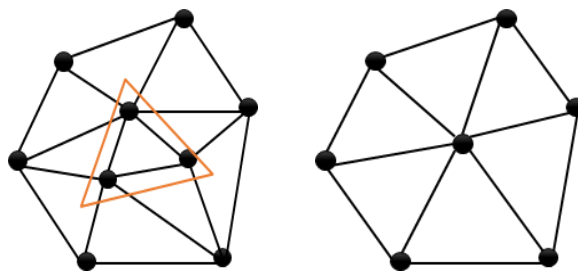


Figure 3-12. Triangle decimation operation

3.3.4 Vertex clustering

Vertex clustering algorithms (Low & Tan, 1997; Rossignac & Borrel, 1993) employ a technique of grouping adjacent mesh vertices into clusters, subsequently replacing each cluster by a single representative vertex (Figure 3-13). To achieve this, a 3D grid of cells is overlaid onto the vertices. Vertices falling within the same cell are clustered together. As a result, numerous faces of the mesh are reduced to points or edges.

Key aspects of vertex clustering algorithms are as follows:

- *Grouping Vertices into Clusters:*

In the vertex clustering process, nearby vertices in the 3D mesh are grouped together into clusters based on their spatial proximity. Vertices that are close to each other and lie within the same cell of the 3D grid are combined into a single cluster.

- **Representative Vertex Replacement:**

Each cluster is then represented by a single vertex, which serves as a proxy for the entire group. This replacement effectively simplifies the mesh by reducing the number of vertices, streamlining subsequent rendering and processing operations.

- **Degeneration of Faces:**

Due to the merging of vertices within each cluster, several faces of the mesh undergo degeneration. These faces may reduce to a single point if they become degenerate triangles or to edges if they become degenerate quadrilaterals.

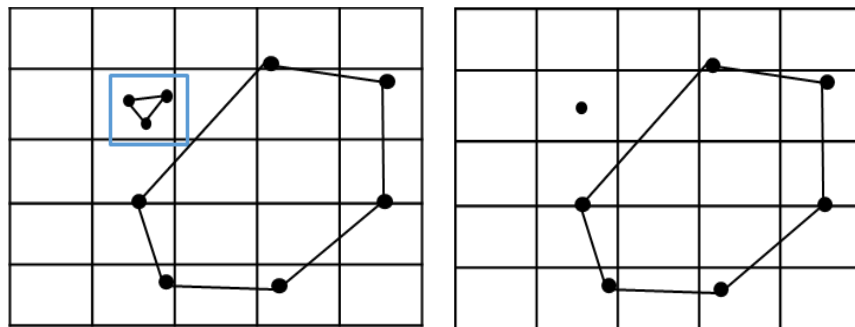


Figure 3-13. Vertex clustering operation

The mesh simplification techniques listed in Table 3-1 differ based on the simplification operator utilized, which determines the number of simplified vertices involved in each operation. Additionally, they vary based on the metric considered for guiding the simplification process. These metrics can be broadly categorized into three types: direct geometry, local geometry, and global geometry.

a) Direct Geometry:

Algorithms falling under direct geometry features focus solely on the properties of individual vertices or edges without considering their neighboring elements. The simplification decisions are made based on the direct attributes of each vertex or edge before removal or clustering.

b) Local Geometry:

Local geometry algorithms take into account the first-order neighborhood of a vertex or edge within the model. They consider the nearby elements when determining whether to remove or cluster a specific vertex or edge.

c) Global Geometry:

Global geometry algorithms assess the overall distortion caused by removing or clustering a vertex or edge across the entire model. These algorithms consider the impact on the entire mesh to make simplification decisions.

The error metric used to determine vertex or edge removal or clustering varies significantly among different algorithms. Each algorithm employs its unique approach to measure and quantify the distortion or error associated with the simplification operation. The techniques listed are widely used and serve as fundamental building blocks for subsequent algorithm development. They have been extensively implemented and incorporated into various libraries for simplifying 3D meshes. As foundational methods in the literature, these algorithms have paved the way for advancements in mesh simplification research, leading to the development of more sophisticated and efficient techniques.

Table 3-1. Different mesh-based simplification techniques

No. of simplified vertices per operation	Simplification operator	Direct geometry	Local geometry	Global geometry
1	<i>Vertex removal</i>	(W. J. Schroeder et al., 1992)	(Hoppe, 1996)	
2	<i>Edge collapse</i>	(Botsch & Kobbelt, 2004)	(Garland & Heckbert, 1997) (Hoppe, 1996)	(Lindstrom & Turk, 1998) (Cohen et al., 1998)
3	<i>Triangle collapse</i>	(Gieng et al., 1997)	(Hamann, 1994) (Rossignac & Borrel, 1993)	
n	<i>Vertex Clustering</i>	(Rossignac & Borrel, 1993)	(Low & Tan, 1997)	

3.4 Mesh based simplification: deep learning techniques

3.4.1 Simplification in the 3D domain

Deep learning-based 3D mesh simplification methods (Hanocka et al., 2019; Potamias et al., 2021, 2022; Rakotosaona et al., 2021; Sharp & Ovsjanikov, 2020) aim at overcoming the limitations of the traditional techniques. Whatever the network architecture retained, the typical steps that need to be achieved are the following:

Data Preparation: A dataset of 3D models is collected, where each model presents both a highly-detailed and a simplified version (usually obtained through traditional techniques). The high-detail models serve as input, and the corresponding simplified models serve as the target output. Another possible way is to provide a database of only original models and specify a visual quality metric and make sure that the output is the result which minimizes the specified metric.

Model Architecture Selection: A suitable deep learning architecture is chosen for the task. Convolutional neural networks (CNNs), (Hanocka et al., 2019) autoencoders (Schönhof et al., 2022), and generative adversarial networks (GANs) (Ramasinghe et al., 2020) are commonly used architectures for 3D model simplification.

Training: The neural network is trained using the prepared dataset. During training, the network learns to capture the relevant features of the high-detail models and generate corresponding simplified versions. The objective is to minimize a given loss function that measures the difference between the output of the network and the target simplified model. The loss guides the training process by helping the network learn to generate accurate simplifications.

Validation and Tuning: The trained model is evaluated on a validation dataset to ensure it generalizes well to unseen data. Hyperparameters and the network architecture may be fine-tuned to improve performance.

Inference: Once the model is trained and validated, it can be used to simplify new 3D models.

Post-Processing: Depending on the specific application, post-processing techniques may be applied to further refine the simplified model. These could involve smoothing, detail restoration, or texture mapping.

It is important to note that while deep learning approaches have shown promising results in 3D model simplification, they also present some challenges. Over-simplification or loss of critical details can occur, and the training process requires a diverse and representative dataset. Furthermore, fine-tuning the model to strike the right balance between simplification and preservation of key features can be complex. This issue is notably challenging in the case of CAD models where the type and amount of simplification varies drastically from one model to another.

The choice of the best algorithm or architecture for 3D mesh simplification depends on the specific task, the size of the dataset, the availability of labelled data, and the desired balance between accuracy and computational efficiency.

To understand more about simplification in 3D domain we have analysed one of the most popular networks called MeshCNN (Hanocka et al., 2019). MeshCNN is a general method for applying CNNs directly on triangular mesh, which supports various many applications, including 3D mesh simplification. The primary goal of MeshCNN is to simplify complex 3D mesh models while preserving their essential features and details. Traditional methods for 3D mesh simplification often involve iterative processes that can be computationally expensive and challenging to optimize. MeshCNN presents a neural network-based approach that leverages convolutional neural networks (CNNs) to directly process 3D mesh data and perform simplification. The underlying simplification operator is the edge collapse (Figure 3-14).

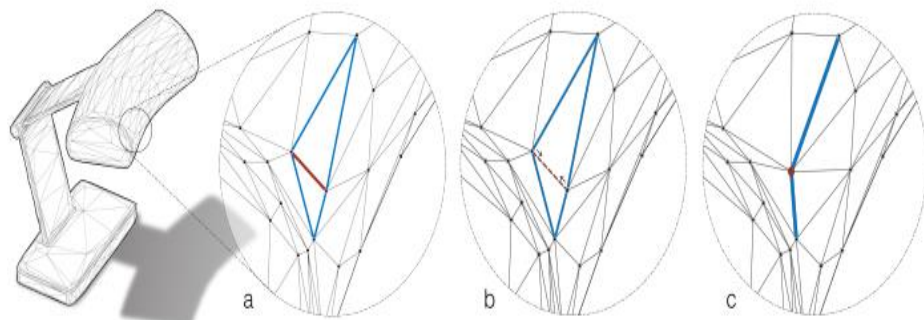


Figure 3-14. Edge collapse in MeshCNN (Hanocka et al., 2019)

3.4.2 MeshCNN: conditions of use

Applying CNNs to meshes requires a combination of data pre-processing, appropriate network architectures, and sometimes specialized layers to handle the characteristics of 3D mesh data. Some of the important steps/components in the MeshCNN are detailed here below.

Input features

The input edge feature is a 5-dimensional vector associated to each edge and including the following elements: the dihedral angle, two inner angles and two edge-length ratios for each face. Let us observe that the edge features are all relative, making them invariant to translation, rotation and uniform scale.

Global ordering

The global ordering of the edges is the order in which the edge data (input features) of a particular shape enters the network. This ordering has no influence during the convolutional

stage since convolution is performed within local neighborhoods. This layer renders the initial ordering inconsequential and thus guarantees invariance to transformations.

Mesh pooling

Mesh pooling is accomplished by an edge collapse operation, illustrated in Figure 3-15. The dashed edge is collapsing to a point, and, subsequently, the four incident edges (blue) merge into the two (blue) edges. Note that in this edge collapse operation, five edges are transformed into two. The operator is prioritized by the edge features with the smallest norm, thereby allowing the network to select which parts of the mesh to simplify. An edge collapse yielding a non-manifold face is not allowed, in order to be able to define consistently the convolution operator, on a fixed number of 4 edges. The desired resolution of the mesh after each pooling operator, by adding a hyper-parameter which defines the number of target edges in the pooled mesh.

Mesh Convolution

An edge-based convolution operator is defined, with a spatial support corresponding to the four edges appearing in the two incident triangles that share the current edge. This operation enables the network to extract hierarchical features from the mesh while taking into account its irregular structure. Following the convolution operation, a new batched-feature tensor is generated, where the new number of features is equal to the number of convolution kernels. Note that after each pooling phase, the new connectivity will define the new convolutional neighbours for the next convolution.

Mesh Unpooling

Mesh unpooling represents the inverse of the pooling operation. The unpooling process is typically combined with convolutions that need to be learnt in order to recover the original resolution lost in the pooling operation. Each mesh unpooling layer is paired with a mesh pooling layer, to upsample the mesh topology and the edge features. For unpooled edge feature computation, a graph structure stores the adjacencies from the original edges (prior to pooling) to the new edges (after pooling). Each unpooled edge feature is then a weighted combination of the pooled edge features. The case of average unpooling is illustrated in Figure 3-15.

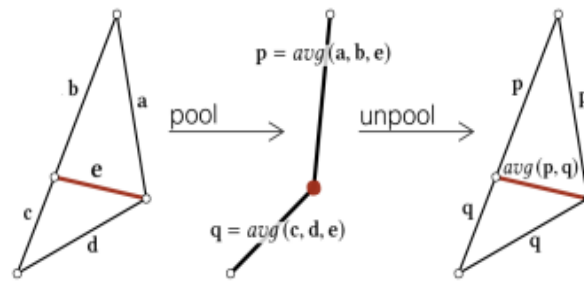


Figure 3-15. Feature aggregation during mesh pooling and unpooling (Hanocka et al., 2019)

Figure 3-16 illustrates some simplification examples for various meshes with different complexities and connectivities. The input meshes are all simplified to roughly 750 edges (500 faces), and are sequentially pooled to 600, 450, 300 and 150 edges.

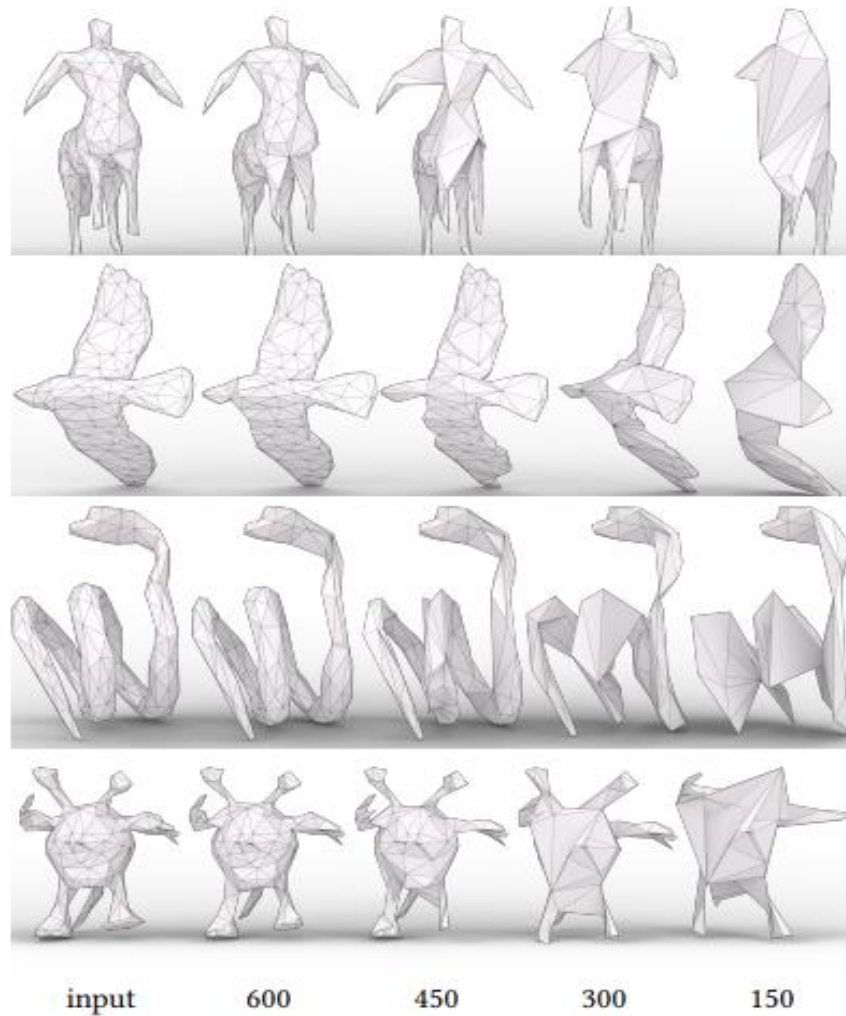


Figure 3-16. Simplification stages with MeshCNN (Hanocka et al., 2019)

3.4.3 Difficulties and limitations

The MeshCNN approach also presents some limitations.

Pool resolution

An essential challenge when employing MeshCNN lies in determining the appropriate pool resolution. During the simplification process, a specific percentage of vertices is typically reduced during the pooling process. Our situation involves a database of CAD models, with vertex counts spanning a wide range from 180,000 to 180 vertices. This broad spectrum of model complexities presents a notable difficulty and achieving a one-size-fits-all solution can be difficult.

Lack of Explainability:

Deep learning models are often considered "black boxes" because it can be challenging to interpret why they make specific simplification decisions. This lack of explainability can be a limitation when precise control over the simplification process is needed. In particular, it is not always possible to preserve fine geometric details or textures in 3D models, which can be critical for certain applications.

Chapter 4

EVALUATIONS

In this chapter we propose an extensive evaluation of existing state of the art 3D mesh simplification approaches. Four different types of evaluations are considered, which include objective evaluations, subjective assessment of simplification level, framerate evaluation in a real case scenario and subjective user evaluation.

- *Objective evaluation*

This evaluation objectively assesses the difference between the original model and simplified model, helping to determine which of the techniques is performing well on a wide range of CAD models.

- *Subjective assessment of the simplification level*

This evaluation aims at establishing, for the most performant evaluation techniques identified in the previous step, the appropriate level of simplification that can be achieved without losing the functional details.

- *Framerate evaluation in a real case scenario*

The most performant state of the art technique identified is further tested in a real AR scenario in the industry, which concerns the creation of assembly line instructions. The advantages of using simplified models in place of original models is evaluated with the help of a frame rate assessment.

- *Subjective user evaluation*

We have conducted a comprehensive evaluation of the use of simplified 3D CAD models, focusing on user experience. A total number of 14 participants performed a set of real time assembly tasks in order to evaluate the performance of simplified models using various metrics. Limitations and possible axes of future improvement are also discussed.

4.1 Objective evaluation

The objective evaluation typically refers to quantitative assessments using various metrics and criteria. Such metrics could include measures of model fidelity, computational efficiency, file size reduction, and other relevant factors. The objective evaluations are designed to provide data-driven insights into the performance of different simplification techniques and help to determining which one is most suitable based on predefined criteria.

To perform the evaluations at least one method from each family has been retained. Hence, one vertex removal (W. J. Schroeder et al., 1992), three edge collapse methods (Garland & Heckbert, 1997; Hoppe, 1996; Lindstrom & Turk, 1998), and one vertex clustering (Low & Tan, 1997) techniques are considered. Note that edge collapse is the most widely used simplification operator, with various metrics involved.

The various simplification algorithms retained for comparison are the following:

- 1) Schroeder *et al.* (W. J. Schroeder et al., 1992), implemented in Visualization toolkit (W. Schroeder et al., 2006)
The vertices are divided into simple, complex, or boundary depending on the angles, and number of edges sharing the vertex. Simple vertices of relatively low importance are deleted first and complex ones never participate in deletion process.
- 2) Hoppe *et al.* (Hoppe, 1996), implemented in the Mesh processing library (Mesh Processing Library, 2023)
The approach is based on the concept of progressive meshes that explicitly models mesh complexity and fidelity as an energy function to be minimized. The goal of this optimization procedure is to preserve not just the geometry of the original mesh, but more importantly its overall appearance as defined by its discrete and scalar appearance attributes such as material identifiers, color values, normal, and texture coordinates.
- 3) Garland *et al.* (Garland & Heckbert, 1997), implemented in VCGLIB (VCG Library, 2023)
The technique is based on an iterative fusion of vertex pairs, which are not necessarily linked by an edge and. The quadratic error metric is here used in order to prioritize the vertex merging process.
- 4) Lindstrom *et al.* (Lindstrom & Turk, 1998), implemented in CGAL (CGAL, 2020)
Lindstrom *et al.* use both volume and boundary optimization in order to reduce the error between the simplified and original models. The local volume preservation ensures that the global mesh is preserved while the boundary preservation makes it possible to conserve as much as possible the shape of surface boundaries in models that are not closed. When both boundary and volume preservation optimization fail triangle shape optimization is used.
- 5) Low *et al.* (Low & Tan, 1997), implemented in VCGLIB (VCG Library, 2023)
This vertex clustering algorithm involves three different steps named grading, triangulation and clustering. A weight is assigned to each vertex depending on the angle between all the pairs of its incident edges. Based on the geometric proximity vertices are formed into clusters. The vertices that fall into the same cluster and are represented by a unique prototype vertex.

4.1.1 Evaluation metric

The evaluation metric is of utmost importance as it serves as a quantitative measure for assessing the performance. There are a few tools developed for measuring the geometric approximation error between two meshes with different connectivity. Among them, the most popular and commonly used one is the Metro tool (Cignoni, Rocchini, et al., 1998).

The Metro method has widespread recognition within the computer graphics community and has been employed numerous times in academic literature to gauge dissimilarities between 3D models. Its fundamental purpose is to compare two triangle meshes that represent the same

surface but at different Levels of Detail (LOD). This comparison is accomplished through the evaluation of the Hausdorff distance between the meshes. The Hausdorff distance is fundamentally a maximum distance among a set of desirable distances between 2 meshes. These desirable distances are minimum distances of all possible vectors resulting from points taken from the first mesh to the second mesh.

One notable advantage of the Metro method is its genericity, since it does not necessitate any prior knowledge of the specific simplification algorithms employed or the characteristics of the 3D models under examination. The Metro method initiates by selecting sampling points on the surface of the first input mesh and then computes the distances between these points and the second mesh, a process referred to as forward distance calculation. Subsequently, this procedure is reversed, and the second mesh is sampled, with distances being computed accordingly, termed as backward distance calculation. Let us note that even when comparing an original mesh with its simplified counterpart, the two meshes should exhibit similar shapes. If the shapes diverge significantly, resulting in the loss of essential features, the accuracy of error computation may be compromised in localized areas. The method provides both the mean and maximum numerical errors resulting from the distance calculations between the two meshes, serving as valuable metrics for assessing the degree of dissimilarity.

4.1.2 Test database

Over the years, the computer graphics community has developed several standard benchmark models (the Stanford bunny is one of the most famous) to assess the performance of simplification techniques. Concerning the 3D CAD models, there is a need of a solid benchmark set of 3D meshes.

To conduct comprehensive testing, we have assembled a database comprising 732 3D models, categorized based on their vertex counts. The models are distributed across five categories: those with less than 1,000 vertices, models with vertex counts ranging from 1,000 to 10,000, those between 10,000 and 50,000 vertices, models with vertex counts between 50,000 to 100,000, and finally, models with more than 100,000 vertices. This diverse selection ensures a wide range of testing scenarios.

All the meshes in the database are real-life models used in the assembly of boilers by elm. leblanc. They are collected from three distinct boilers models.

The characteristics of the models are summarized in Figure 4-1. The blue bars indicate the number of models per boiler while the orange ones indicate the number of models in the database, which represents the whole collection of three boilers (the entire database).

By encompassing a significant number of models with varying vertex counts from real-world boiler assembly processes, our database provides an extensive and representative set of 3D CAD

models to evaluate the performance and effectiveness of the various simplification techniques retained. This comprehensive approach ensures that our analysis is well-grounded and can cater to a broad range of real-life scenarios.

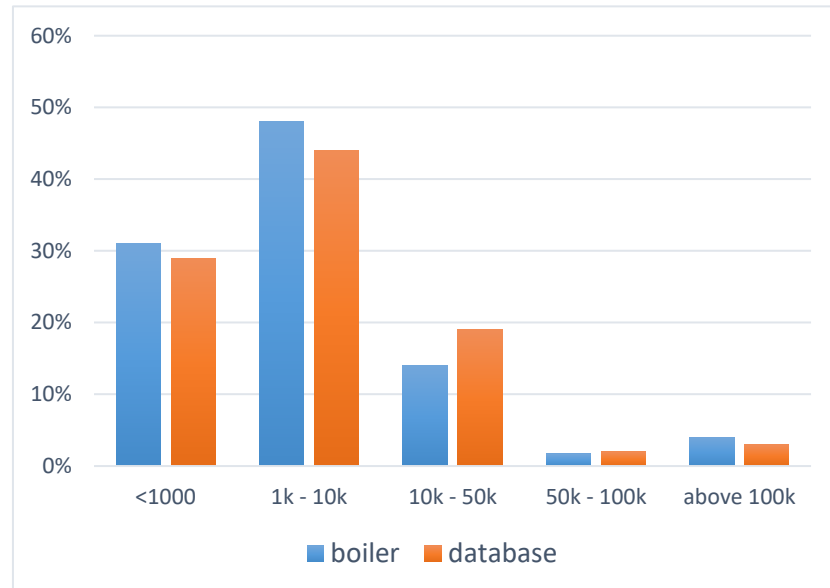


Figure 4-1. Percentage of models in the test dataset with respect to the number of vertices

The models in the database encompass a diverse range of real-life 3D CAD models, each with its own shape, size, functionality, and structure. These models have been carefully selected to represent various aspects of boiler assembly and are crucial for assessing the effectiveness of our testing. Figure 4-2 illustrates some examples, which showcase the variety and complexity of the models, highlighting the differences in their designs, components, and intricacies.

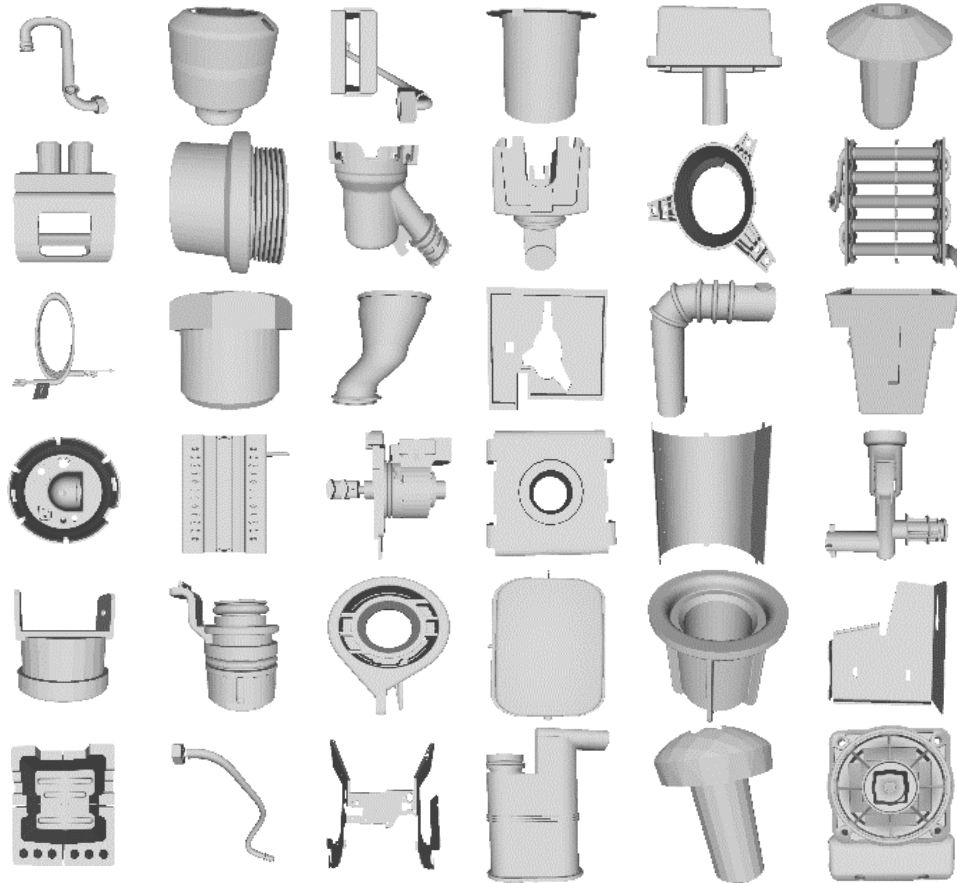


Figure 4-2. Examples of CAD models in database

4.1.3 Comparison of mesh simplification techniques

In our extensive testing of the database, we have evaluated numerous models. To present the results in a focused and detailed manner, we have carefully chosen one representative model from each category of vertices. Specifically, we selected the most commonly used model within each category for a more in-depth and objective evaluation (Figure 4-3). The selected models have been chosen deliberately to encompass a wide variety of sizes, shapes, structures, and features. By including models with diverse characteristics, we can thoroughly examine the effectiveness of our augmented reality-based instructions using simplified 3D CAD models in different scenarios. In Figure 4-3, V denotes the number of vertices.

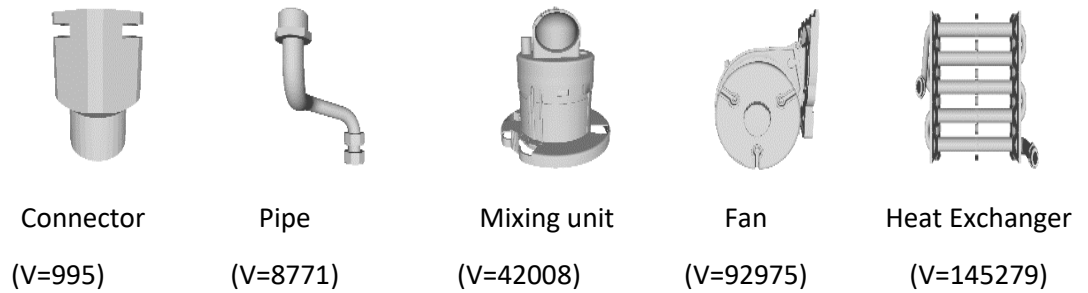


Figure 4-3. Test models

Both the mean and the maximum Hausdorff errors between initial and simplified meshes are provided. The results obtained are presented in Figure 4-4 to Figure 4-13. The X-axis corresponds to the number of vertices in the models, while the Y-axis represents the error rate. To assess the impact of mesh simplification, the evaluation has been conducted at various stages of simplification, which are of 50%, 60%, 70%, 80%, 90%, and 95%.

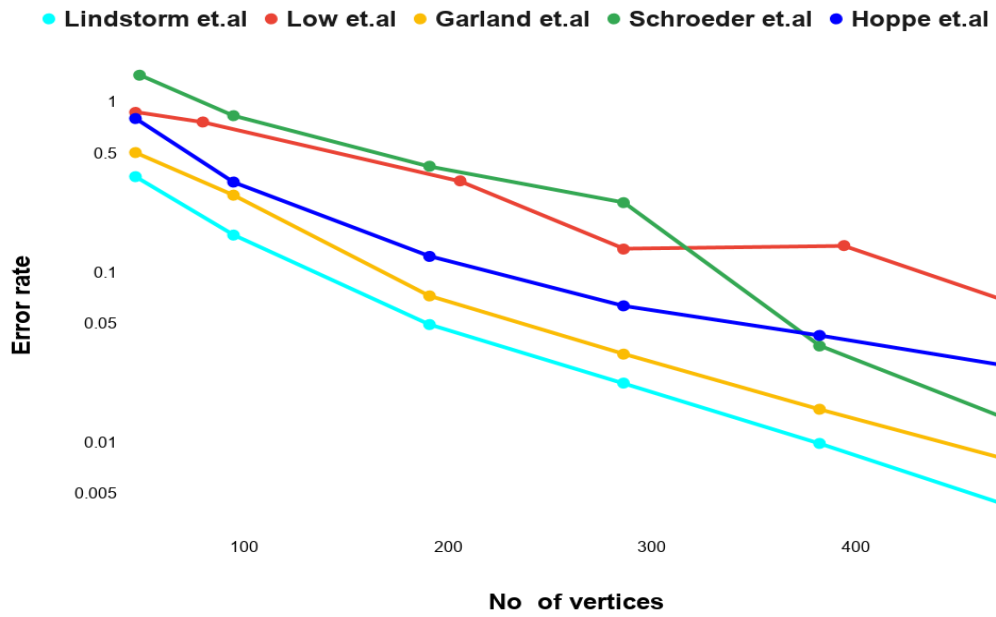


Figure 4-4. Average error for connector

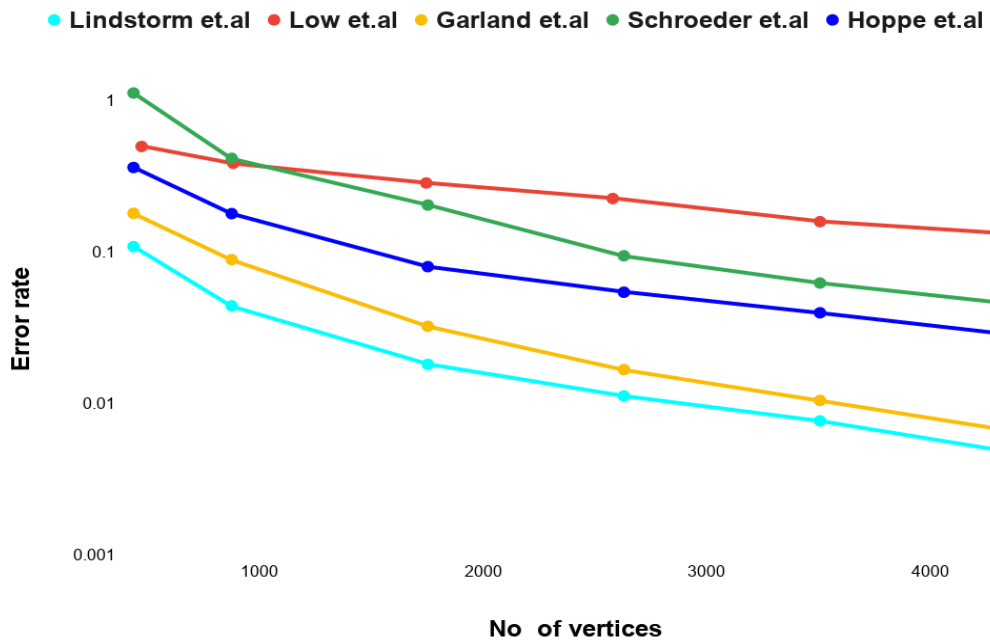


Figure 4-5. Average error for pipe

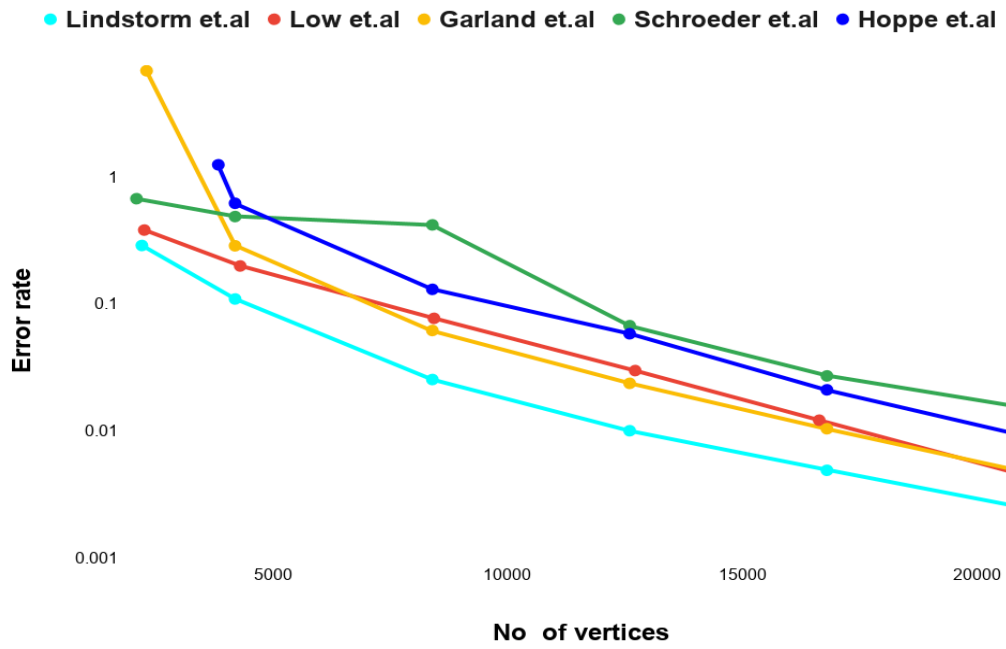


Figure 4-6. Average error for mixing unit

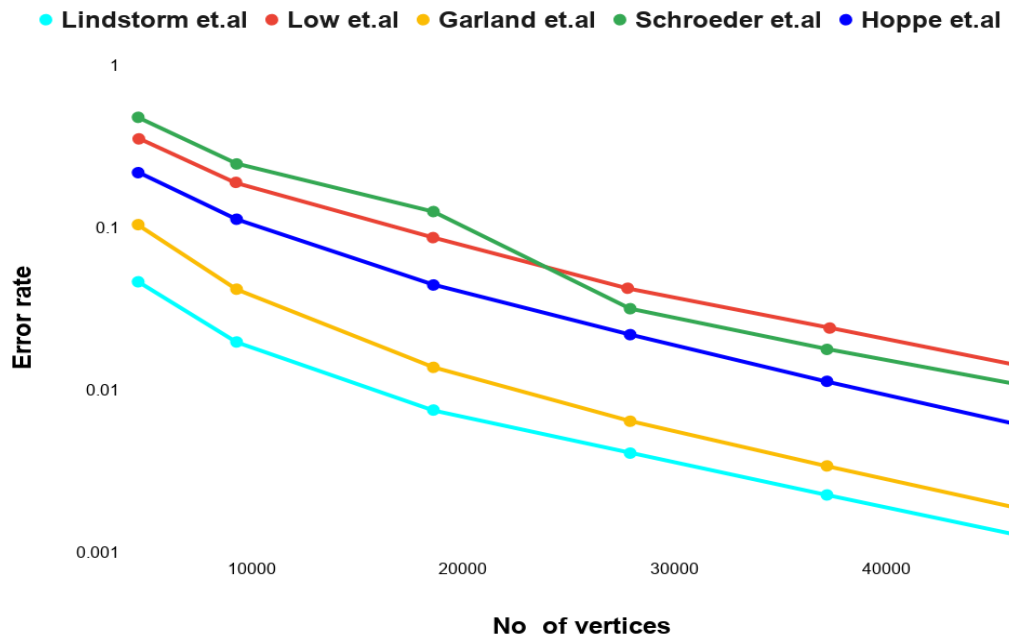


Figure 4-7. Average error for fan

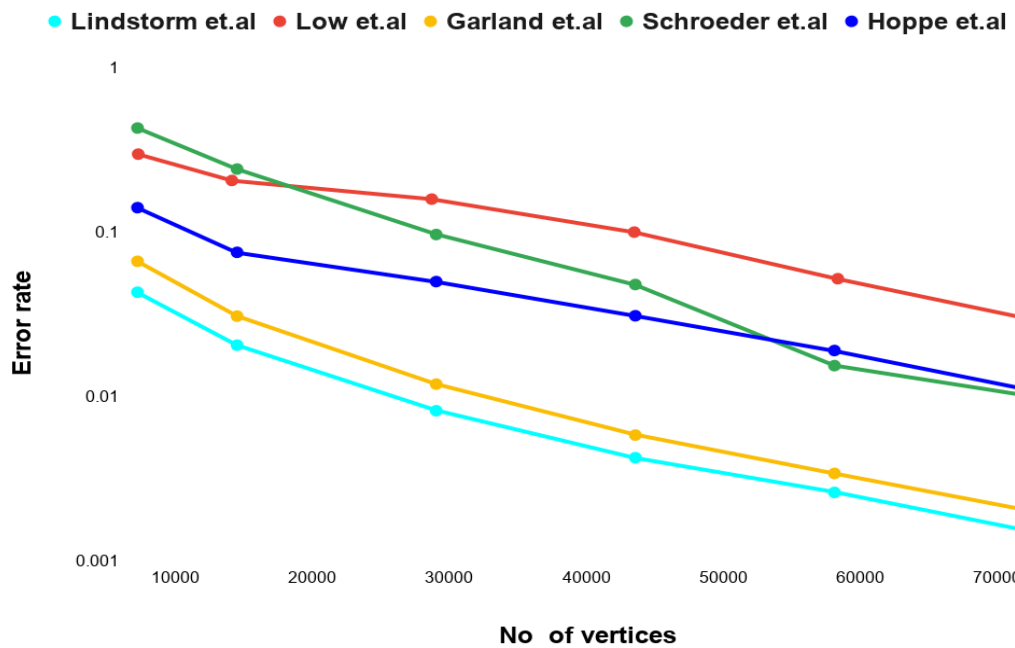


Figure 4-8. Average error for heat exchanger

Let us mention that if after some amount of simplification there are no vertices/edges to be removed or clustered the simplification process stops for some algorithms.

In a general manner, we can observe that the approach proposed in Lindstrom *et al.* (Lindstrom & Turk, 1998) outperforms the rest of the algorithms at all simplification levels in terms of average error. The performance of other methods varies with different percentages of simplification and test models. As we go to higher percentages of simplification, The Schroeder *et al.* (W. J. Schroeder et al., 1992) method, which is based on vertex removal, becomes the least performant method. For example, for the test model connector Lindstrom *et al.* (Lindstrom & Turk, 1998) has an error rate 10 times less than Schroeder *et al.* (W. J. Schroeder et al., 1992).

The maximum errors are presented from Figure 4-9 to Figure 4-13.

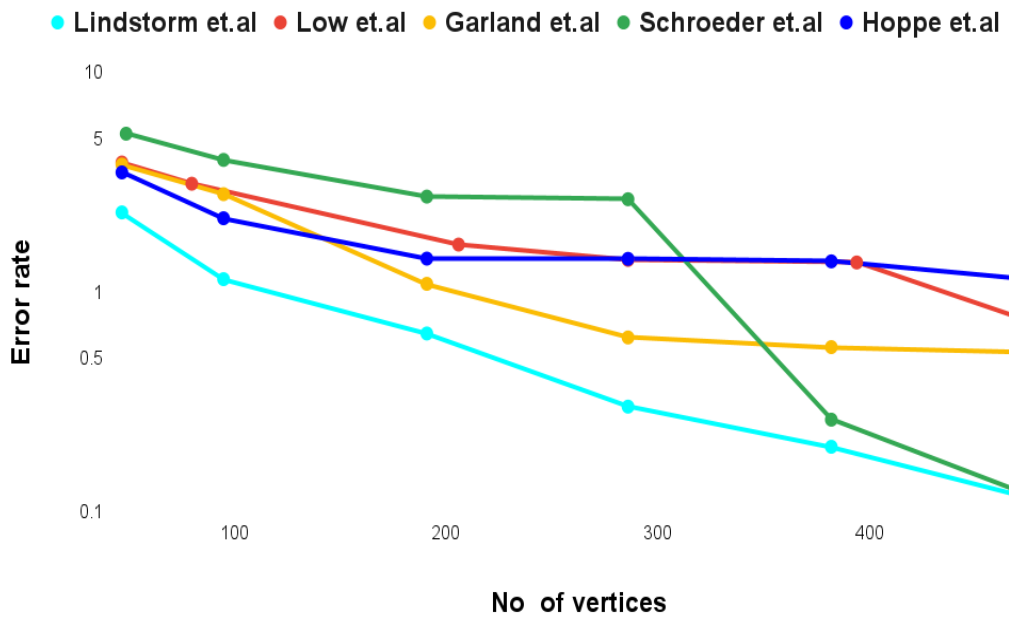


Figure 4-9. Maximum error for connector

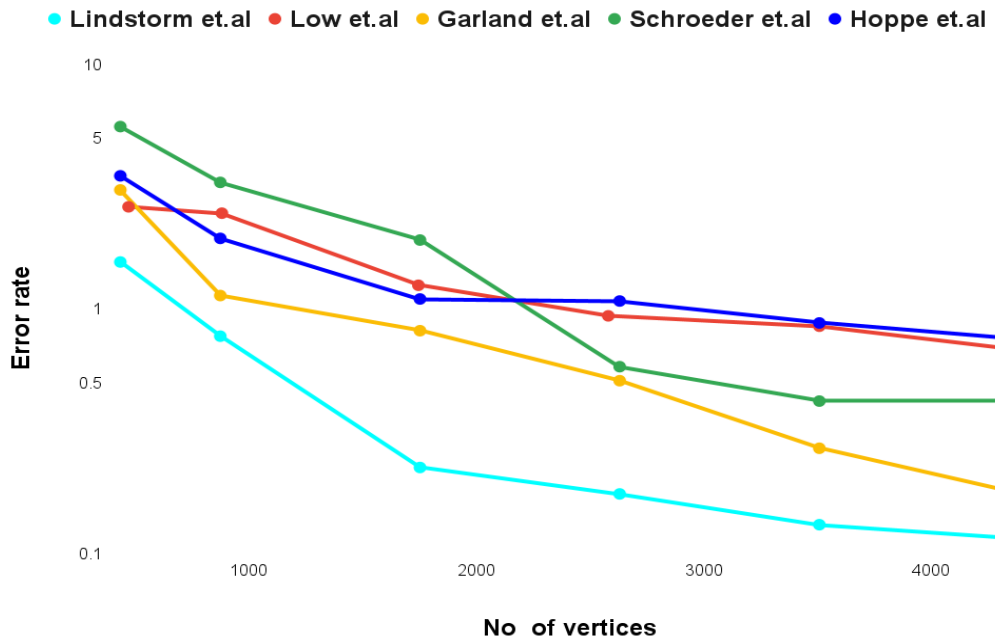


Figure 4-10. Maximum error for pipe

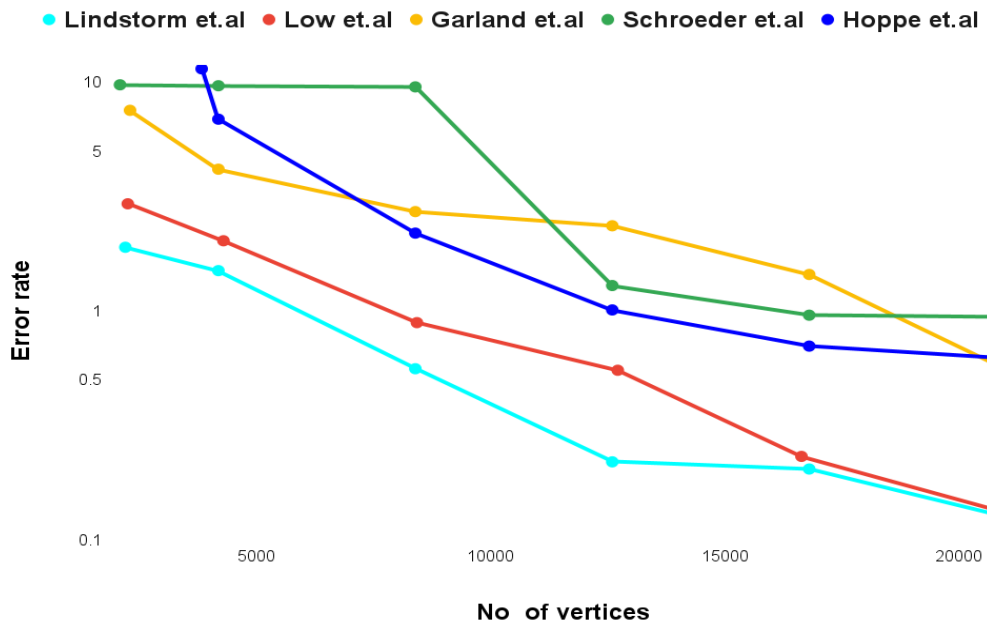


Figure 4-11. Maximum error for mixing unit

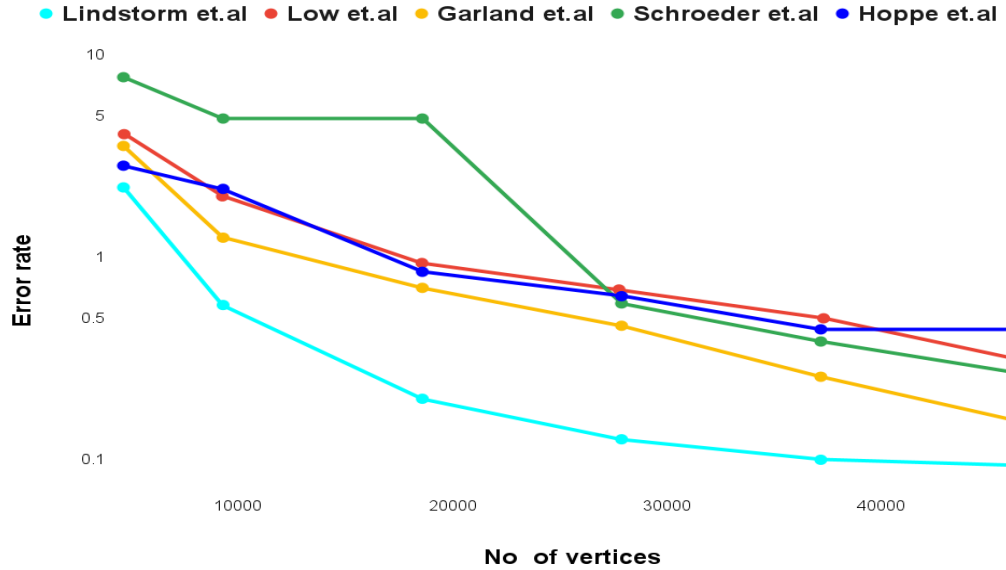


Figure 4-12. Maximum error for fan

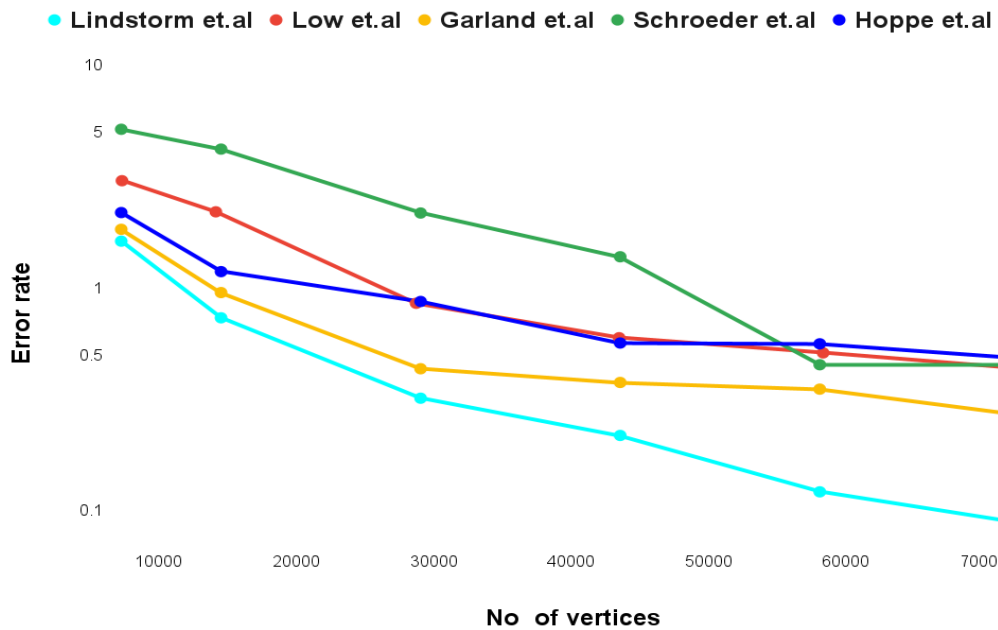


Figure 4-13. Maximum error for heat exchanger

The results obtained for the maximum error have shown the same trend as for the average error. The method of Lindstrom *et al.* (Lindstrom & Turk, 1998), which uses a global error metric, outperforms the other techniques. Although the performance of the other algorithms kept on changing from one simplification level to another and from one model to another, Lindstrom *et al.* (Lindstrom & Turk, 1998) has constantly shown better results in terms of both maximum and average errors.

One of the challenges we faced is determining the optimal simplification rate for the CAD models. While Lindstrom *et al.* (Lindstrom & Turk, 1998) approach has proven to be effective, selecting the most suitable level of simplification from the available options is not a straightforward task. The decision becomes more complex as the number of vertices increases.

For example, with the largest model considered in our database, which is the heat exchanger, we can still achieve visually satisfactory results even with a high level of simplification, such as 95%. The simplified model at this percentage remains usable on the HoloLens and maintains a good level of visual quality. However, this is not the case for the connector model, which only has 995 vertices. For smaller models like the connector, a 95% simplification using Lindstrom's approach may result in a significant loss of critical features and visual fidelity, making it unsuitable for effective use on the HoloLens.

The process of determining the optimal simplification rate for each model is a critical and delicate task. Several factors come into play to strike the right balance between simplification

and retaining essential features, ensuring that the augmented reality-based instructions remain effective and engaging for users.

The factors that affect the usability of simplified models for AR varies from one model to another and include:

- *Model Size*: The number of vertices in a model directly impacts its complexity and computational requirements. Larger models with a higher number of vertices may necessitate a higher simplification rate to achieve real-time rendering performance.
- *Model Complexity*: The intricate design and details present in a model influence the degree to which simplification can be applied without sacrificing crucial information. Complex models with intricate features may require more cautious handling to ensure that vital structural elements are preserved.
- *Desired Level of Visual Detail*: The level of visual detail required for a specific application or use case also affects the simplification rate. Some scenarios may necessitate higher levels of detail, while others can tolerate greater simplification for efficient rendering.

4.2 Assessment of the level of simplification

We have conducted a subjective analysis to identify the level of simplification desired for each model. This evaluation is paramount because ordinary users may not possess the discernment to detect certain disparities. If during the simplification process, crucial features vital for the assembly are omitted, it can arise usability issues and impair the models suitability for its intended purpose in the AR use cases.

In the example of the turbine pipe illustrated in Figure 4-14, where the original CAD model consists of approximately 60,000 vertices, a simplification of 90% can indeed be considered reasonable in many cases. This level of simplification would result in a substantial reduction in vertex count, potentially bringing it down to around 6,000 vertices, while still retaining the essential structural and functional aspects of the model. Hence, a simplification rate of 90 percent can be accepted for this specific model. However, it is important to note that simplifying a model by 90% will inevitably introduce some level of distortion or loss of detail, especially at the feature level. Therefore, the decision to simplify to this extent should be made in consideration of the specific use case and the level of detail required for the task at hand.

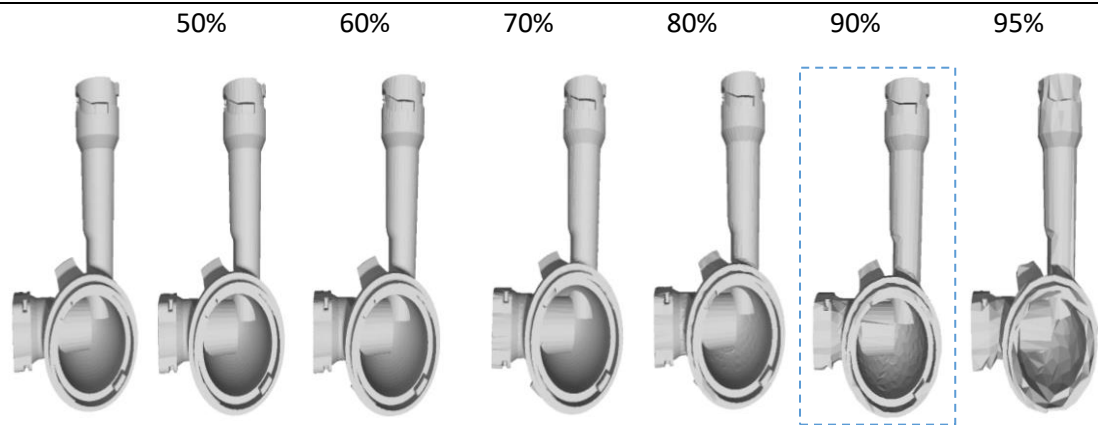


Figure 4-14. Examples of simplified models by (Lindstrom & Turk, 1998)– Turbine pipe

In the case of the bolt model illustrated in Figure 4-15, which comprises approximately 7,000 vertices, a simplification rate of 80% has been determined as a suitable level of simplification by experts. This observation highlights that the optimal simplification rate for CAD models can indeed vary based on their individual characteristics, including size and complexity. The choice of an 80% simplification rate suggests that while reducing the models complexity by a significant margin, essential details and features are still preserved at an acceptable level.

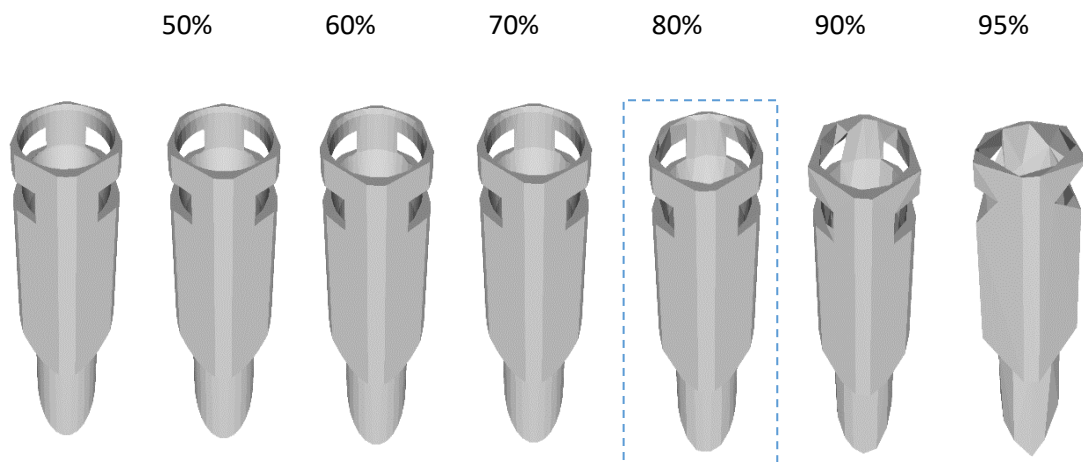


Figure 4-15. Examples of simplified models by (Lindstrom & Turk, 1998)– Bolt

To address this variation in optimal simplification rates, a subjective evaluation has been made with expert visual inspection in order to identify suitable level of simplification for CAD models in different ranges. All the simplification levels from 50% to 95% have been considered for each model, with a step of 5%. Several test models have been selected from each range of vertices considered. Thus, 10 test models are taken from below 1k range whereas 100 meshes are taken in the range 1k to 10k since in this range we encounter the highest number of models in the

database. Globally, more than 150 models have been manually assessed. The acceptable simplification rates as well as the number of models considered in each range are reported in Table 4-1.

Table 4-1. Simplification rates of each range of vertices

Range of vertices	No of models assessed	Simplification rate
Less than 1k	<i>10</i>	<i>70%</i>
1k to 10k	<i>25</i>	<i>80%</i>
10k to 50k	<i>10</i>	<i>80%</i>
50k to 100k	<i>5</i>	<i>90%</i>
Above 100k	<i>5</i>	<i>95%</i>

Figure 4-16 shows the results of the subjective evaluations, confirming that Lindstrom *et al.* (Lindstrom & Turk, 1998) simplification technique yields visually superior models compared to the other simplified models at the proposed simplification rates. The models highlighted in orange demonstrate that, at these specific rates, there is no noticeable visual difference when compared to the original models. This indicates that these specific rates strike an optimal balance between simplification and preservation of crucial visual details.


























Model	Lindstrom et al.	Low et al.	Garland et al.	Schroeder et al.	Hoppe et al.
Connector (75%)					
Pipe (80%)					
Mixing unit (80%)					
Fan (90%)					
Heat exchanger (95%)					

Figure 4-16. Visual evaluation of the simplified models at the retained simplification rates

Let us now consider the case of a specific model called "hydraulic" (Figure 4-17), which contains approximately 222,000 vertices. When applying Lindstrom *et al.* (Lindstrom & Turk, 1998) simplification algorithm to the hydraulic model, we have observed that the frame rate inside AR device (HoloLens2) of the model significantly improved. At around 80% simplification, the frame rate increased by 6 frames per second (6 FPS) from 35.77 fps of original model to 40.52fps of simplified model. This indicates that the simplification process successfully reduced the complexity of the model, leading to smoother and more efficient rendering.

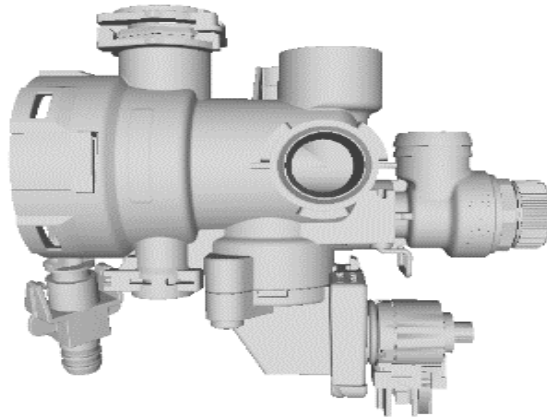


Figure 4-17. Hydraulic model

One notable benefit of the simplification process was the reduction in file size. By removing unnecessary details from the model, the file size was drastically reduced, leading to faster loading times and better resource management.

Table 4-2 presents different measurements of the model Hydraulic at various percentages of simplification.

Table 4-2. Different measurements for the Hydraulic model

CAD model	No of vertices	Percentage of simplification	Time taken to simplify (MS)	Frame rate (Hololens2)	File size (KB)
Hydraulic	222462	<i>original</i>	-	35.77	12601
	111231	50	41752268	38.43	7096
	88984	60	50279508	39.32	5738
	66738	70	59569531	40.35	4355
	44492	80	70132116	41.11	2929
	24828	90	88334245	40.52	1543

4.3 Framerate evaluation in a real case scenario

The framerate evaluation has been performed in the Microsoft's HoloLens2 (Figure 4-18) device (Ungureanu et al., 2020). The evaluation procedure has been conducted as follows: models are loaded in the scene and rendered for one minute while rotating around their centre of mass. The framerate obtained from MRTK's Diagnostics module is used to obtain the average framerate. The testing application has been developed in Unity 2019.4.10f1. MRTK 2.4.0 was used for creating the interface of the application and to measure the framerate. Let us note that the absolute framerate value is not important for our evaluation, as changing quality parameters in Unity will increase or decrease the framerate drastically. The quality settings we are referring to are the rendering parameters: texture quality, anisotropic textures, anti-aliasing and soft particles. By changing the mesh renderer and the material properties of the CAD models, the value of the framerate will again be affected. We do not try to evaluate combinations and find the best set-up of these properties because each AR use-case has different requirements when it comes to rendering CAD models. Our objective is to identify the framerate difference between the original model and its simplified versions by using the same rendering properties. The framerate values presented in the Table 4-3 have been obtained by setting the quality settings to "Ultra".



Figure 4-18. HoloLens2

In our evaluation, we have selected a set of 12 complex CAD models for assembly at a workstation. When combined, the original models contained a total number of 1,401,956 vertices. To improve performance and optimize the rendering process, we used the simplification technique developed by Lindstrom et.al to create corresponding simplified models. The simplified models, after applying the Lindstrom *et al.* (Lindstrom & Turk, 1998) algorithm, reduced the total vertex count to 127,619. This significant reduction in vertices was achieved while preserving the essential structural features of the original models, with the help of the expert evaluation procedure described in Section 4.2.

In Figure 4-19a, we present a visual representation of all the original models loaded simultaneously into the HoloLens2 device. In Figure 4-19b, we showcase the same scenario, but with all the simplified models loaded instead. The Figure 4-19 illustrates the successful rendering of the simplified models with reduced computational demands and improved efficiency.

The obtained framerates for original and simplified models are reported in Table 4-3.

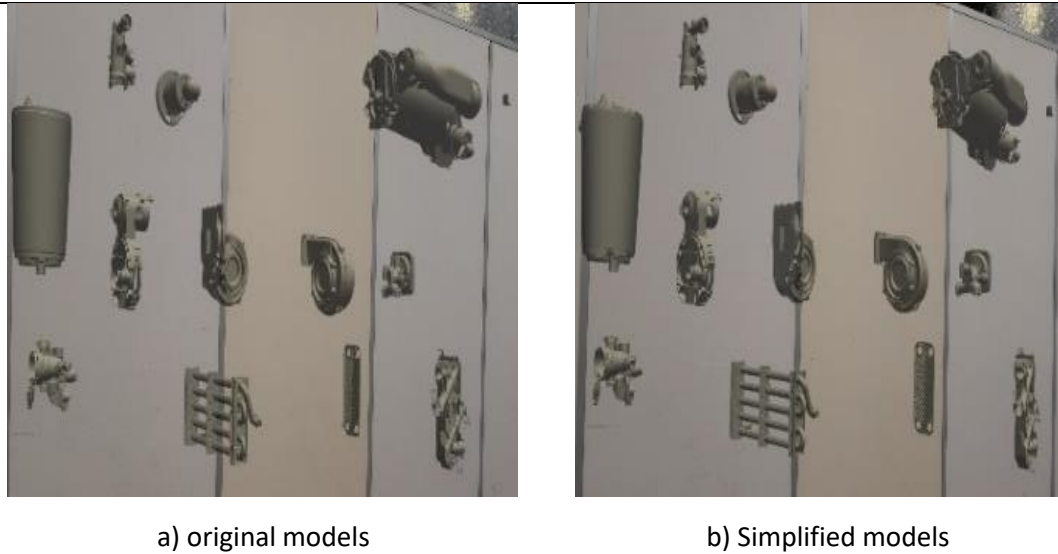


Figure 4-19. Original and simplified models visualized in HoloLens 2

When the original models are loaded it is impossible to create instructions or just to interact with the models since they are very heavy. When simplified models are used instead of original models the frame rate increases significantly, making it possible a fluid instruction creation.

The simplification process greatly helps to improving the framerate. However, let us recall that when we have to load a whole boiler composed of hundreds of models, which accounts to millions of vertices, the framerate remains very low despite the use of simplified models. Therefore, there is a still a need to improve simplification algorithms.

Table 4-3. Framerates in HoloLens 2

Type of models	Total number of vertices	Framerate(fps)
Original (12)	<i>1401956</i>	<i>16.9</i>
Simplified (12)	<i>127619</i>	<i>57.47</i>

A final subjective evaluation, focusing on the end-users, is described in the following section.

4.4 Subjective user evaluations

While subjective assessments are invaluable for assessing complex 3D models, it is important to combine their insights with both objective evaluations and user testing whenever possible. This multi-faceted approach ensures a well-rounded assessment of the models suitability for the intended use case, taking into account both technical and practical considerations.

User evaluations play a crucial role in assessing the performance and quality of emerging technologies, especially in the context of AR applications (Gabbard & Swan, 2008). As such, user-based experiments are essential for driving design activities and improving usability. AR user evaluations play a vital role in the iterative design process, allowing developers and designers to identify strengths and weaknesses in the AR application and iteratively refine it based on user feedback. These evaluations are essential for creating AR experiences that are user-centric, efficient, and impactful across a wide range of industries and use cases.

One common limitation in AR evaluations is the reliance on monitors and screens instead of actual augmented reality devices (Werrlich et al., 2017). Such informal evaluations may not accurately represent the complexities and challenges faced in real-world industrial settings. To address this, we have conducted subjective evaluations in a real-world industrial environment, ensuring that our findings are more reliable and applicable to practical scenarios.

In the literature, various types of AR evaluations are reported (Dünser et al., 2008), each tailored to the specific applications and use cases. For our research, we have retained two types of user evaluations:

- *Objective Measurements:* We collected objective data, including the time taken by users to complete tasks, the accuracy of their performance, and the number of errors made during assembly or other activities. These quantitative measurements provided valuable insights into the efficiency and effectiveness of our augmented reality-based instructions using simplified 3D CAD models.
- *Subjective Measurements:* In addition to objective data, we gathered subjective feedback from users through ratings, judgments, and descriptive analysis. User opinions and insights offer a deeper understanding of their experiences, preferences, and challenges while interacting with the augmented reality instructions.

By combining both objective and subjective measurements, our evaluation approach provides a holistic view of the effectiveness of the state of the art solution (Lindstrom & Turk, 1998).

Overall, the user evaluations are instrumental in validating the practicality and usability of our augmented reality-based instructions. The insights gained from these evaluations enable us to fine-tune our solution and ensure that it meets the real-world needs of industrial assembly tasks, paving the way for more efficient and accurate operations.

4.4.1 Related work

Previous research by (Lavric et al., 2021a; Vincenzi et al., 2003) has focused on comparing various assets and media within AR devices. Lavric *et al.* (Lavric et al., 2021a) compared different visual assets such as images, videos, and 3D models, while Vincenzi *et al.* (Vincenzi et al., 2003) compared different learning methodologies, including printed material, videotapes, and text annotations. However, these works primarily involved comparing 3D model-based instructions

against other types of instructions, and they did not specifically address the comparison between original 3D models and their simplified counterparts to improve application performance.

One of the main reasons for the lack of such comparisons could be the difficulty in producing simplified models that retain all the functionalities of the original models while enhancing the AR applications performance. Achieving a balance between simplification and retaining critical features can be challenging.

In our work, we have addressed this challenge and conducted an in-depth analysis of different available simplification techniques. To the best of our knowledge, our research is among the first to evaluate the usability of simplified 3D CAD models in comparison to their original counterparts specifically within the context of AR instructions for manual assembly operations. The evaluation has been carried out in a real-world industrial setup.

4.4.2 Experimental setup

We have designed six tasks categorized into three levels of difficulty: simple, medium, and complex. Each task involves assembling components in an AR environment. The simple tasks are focused on placing an assembly component accurately. The medium tasks require to considering alignment while placing the components. The complex tasks involved both alignment and orientation considerations during the assembly process.

To perform the assembly tasks, we utilized 3D models (boilers and related components) from the elm.leblanc CAD model database. The models were carefully selected to represent various components and complexities encountered in real-world boiler assembly operations.

The assembly tasks are organized into three sets, each containing two similar tasks. This design allowed us to ensure a non-biased evaluation by providing participants with diverse scenarios to assemble. The range of assembly actions participants performed included common industrial actions such as "screw, fit, install, place, push, slide." This diverse set of actions allowed us to assess the versatility and applicability of the proposed AR instructions.

Figure 4-20 provides an overview of the evaluations performed. By employing a varied range of tasks and 3D models, we ensured a comprehensive evaluation that covered different levels of complexity. The participants' actions varied from one AR instruction to another, simulating real-world assembly scenarios.

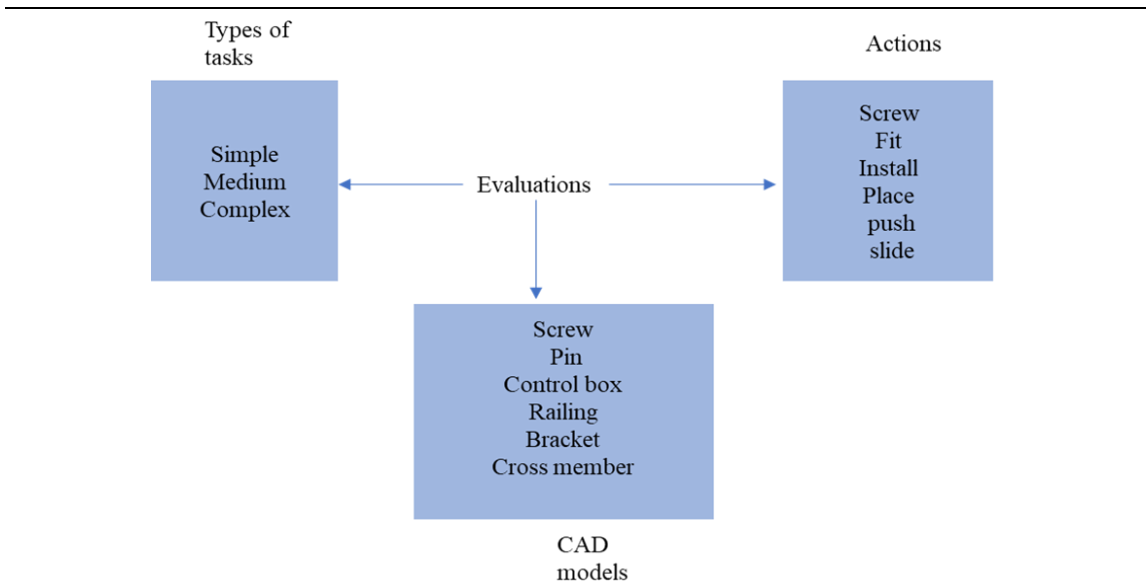


Figure 4-20. Overview of the evaluations

4.4.3 Instructions

The testing application for evaluating instructions on industrial assembly lines has been developed using Unity 2019.4.10f1 in conjunction with HoloLens2. To create the instructions, we utilized a tool developed by Lavric *et al.* (LAVRIC et al., 2020), which generates step-by-step guidance for users. To create a realistic evaluation environment, a scene resembling industrial settings was generated in the laboratory. We have also included CAD models commonly used in the industry to replicate the assembly tasks accurately.

To ensure a comprehensive evaluation, we have carefully studied multiple instructions performed on real industrial assembly lines. From this analysis, they selected similar tasks that could be replicated in the testing application for evaluation purposes. These selected tasks likely cover various aspects of industrial assembly processes, enabling a comprehensive assessment of the effectiveness of the AR-based instructions.

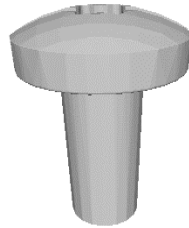
By utilizing Unity, HoloLens2, and the tool developed by Traian *et al.* (LAVRIC et al., 2020), we aimed to provide an immersive and interactive experience for the evaluation of AR instructions in an industrial context. This setup allows them to measure the performance, usability, and effectiveness of the simplified CAD models in AR instructions, potentially leading to insights that can improve the design and implementation of AR-based training and guidance in industrial settings.

Figure 4-21 to Figure 4-26 illustrate the 3D models used in the tasks (AR views), with both original and simplified models. The first two tasks were relatively straightforward, serving as

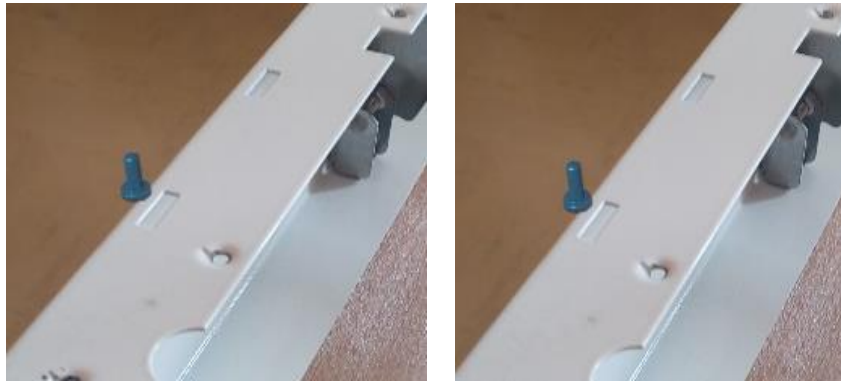
Evaluations

introductory tasks. The subsequent two tasks presented medium difficulty, adding alignment considerations. The final two tasks were part of the complex set, requiring participants to align and orient the components for successful assembly.

Instruction 1: Screw



Screw

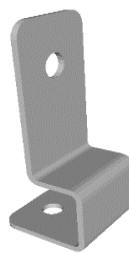


Instruction with: Original model

Simplified model

Figure 4-21. Assembly of screw

Instruction 2: Fit



Pin

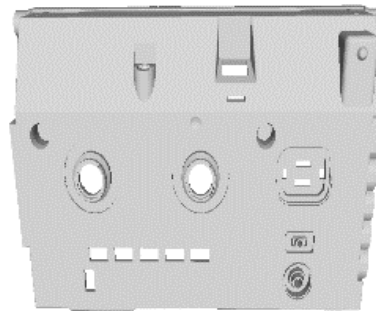


Instruction with: Original model

Simplified model

Figure 4-22. Assembly of fit

Instruction 3: Install



Control box

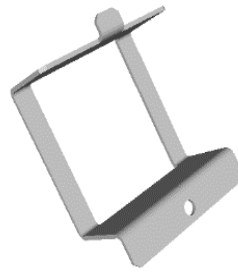


Instruction with: Original model

Simplified model

Figure 4-23. Assembly with control box

Instruction 4: Push



Railing



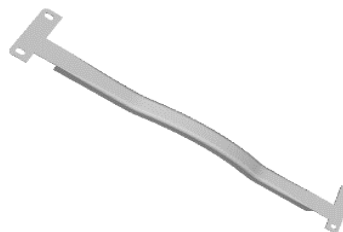
Instruction with: Original model



Simplified model

Figure 4-24. Assembly with railing

Instruction 5: Place



Cross member



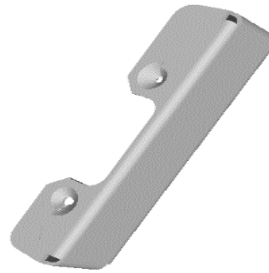
Instruction with: Original model



Simplified model

Figure 4-25. Assembly with cross member

Instruction 6: Slide



Bracket



Instruction with: Original model



Simplified model

Figure 4-26. Assembly with bracket

4.4.4 Evaluation procedure

The step-by-step procedure involved in performing the evaluations includes:

- At the start of the evaluations, participants receive an explanation tailored to their level of experience with AR and assembly. For those with no prior AR and assembly experience, the average time taken to explain the basic concepts of AR, assembly work, and the experiment was approximately 6-8 minutes. On the other hand, participants with some essential knowledge of AR and assembly required an average explanation time of about 2-3 minutes. This customized approach ensured that all participants had a clear understanding of the evaluation process, setting the stage for a meaningful and informed assessment of the AR application.
- During the experiment, participants underwent eye calibration with HoloLens 2 to ensure accurate visualization of the 3D CAD models. The eye calibration process typically took approximately 1 minute for each participant. This calibration step was essential to optimize the AR experience and ensure that the participants can view the CAD models

with precision and alignment to their gaze, enhancing the overall usability and effectiveness of the evaluation.

- During the evaluation, a timer is set for each instruction to record the time taken by participants to complete the task. At the end of each instruction, participants are asked to stop and provide feedback, including ratings and their perception of the task's complexity. After answering the questions, participants proceed to the next instruction. Throughout the evaluation, various parameters are noted and recorded for each instruction, enabling a comprehensive analysis of user performance, task difficulty, and user feedback. This structured approach ensures that the evaluation process is systematic and allows for a detailed understanding of participants' experiences.
- After completing all six tasks, the participants are provided with an explanation regarding the purpose of rating the 3D models. They are informed that both the original and simplified models were used in creating the instructions. This explanation is given after completing the tasks to eliminate any potential bias that may arise from the knowledge of the existence of simplified models within the instructions. By providing this clarification after the tasks are performed, the evaluation ensures that participants' ratings and feedback are based on their actual experiences and interactions with the AR application, without any preconceived notions or influences. This approach enhances the reliability of the evaluation results.
- Subsequently, general feedback is collected from the participants regarding various aspects of the evaluation process. This feedback encompasses their thoughts on the setup, experimentation procedures, the clarity of explanations provided, and the overall implementation of the evaluations. Participants have the opportunity to share their insights, suggestions, and any potential areas for improvement.

4.4.5 Participants and tasks performed

A total number of 14 participants participated in the evaluation. The participants have diverse backgrounds, including students, researchers, engineers, and entry-level operators. Some have AR experience without assembly knowledge, some have AR experience with assembly knowledge, and others have neither AR experience nor assembly knowledge. However, we have observed that this difference in experiences did not have a detrimental effect on the evaluations.

After instructions are set, we alternate between original and simplified models and between users to perform the evaluations. Each set consists of two instructions: one with the original model and one with a simplified model. The tasks performed by each participant are presented in Figure 4-27. To record the completion time, the number of errors, and other statistics, we cannot show the original model and simplified model of the same instruction since the user has already seen the same instruction before and when he/she repeats it with the original/simplified model, the data will be better but biased.

Evaluations

To eliminate such a bias, we have chosen to perform in an alternating method. In a set of similar tasks, one participant will perform the first instruction with the original model and the simplified model in the second task. When it comes next set of similar tasks, the same participant will do the first instruction with the simplified model and the original model in the second task. In this way, we keep alternating between the original and simplified models and the tasks to have a non-biased evaluation. When it comes to the second participant, he/she does the exact instructions as the first participant with exactly opposite 3D models(original/simplified). Likewise, all the participants who do the first set of instructions are put into one group while the rest into another group. Ultimately, we compare different parameters between original and simplified models between the two sets of participants and analyse how the participants have performed the tasks, especially with simplified models.

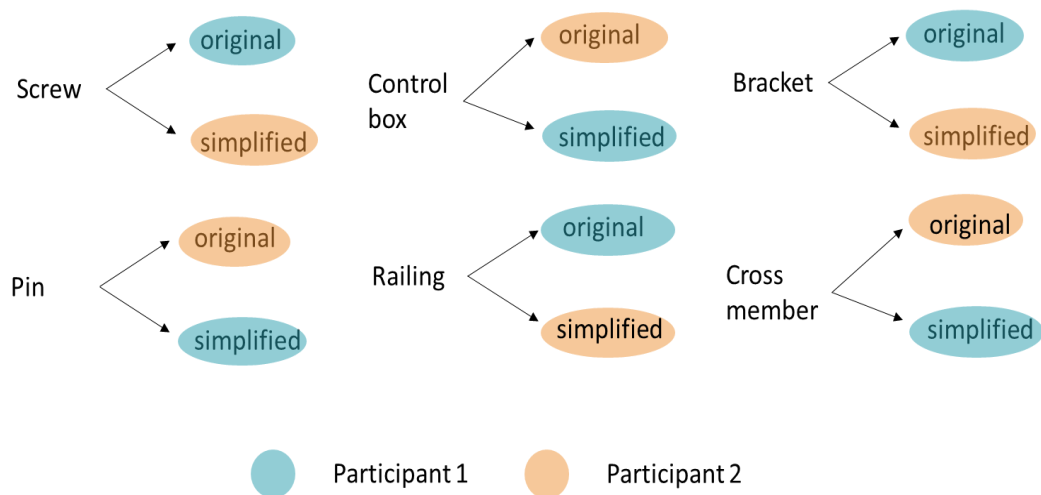


Figure 4-27. Instructions per participant

Table 4-4 gives a detailed overview of the total number of participants and tasks done by them respectively.

Table 4-4. Participants and tasks done

Participants	3D models in tasks done
P1, P3, P5, P7, P9, P11, P13	SET 1: Screw (O), Pin (S) Control box (S), Railing (O) Bracket (O), Cross member (S)
P2, P4, P6, P8, P10, P12, P14	SET 2: Screw (S), Pin (O) Control box (O), Railing (S) Bracket (S), Cross member (O)

*O-original model, S- simplified models

4.4.6 Evaluation metrics and results

The evaluation encompassed various metrics, including the time taken to complete the assembly, model comprehensiveness, error rate, completion rate, and task complexity. To ensure a diverse range of tasks in the evaluations, participants were asked to rate the complexity of each task from simple to hard. While the number of errors committed and successful task completions are closely related, instances where participants rectified their errors independently by carefully examining the instructions were considered separately to account for individual problem-solving abilities.

Both the complexity of the task and the rate of successful completion were measured to identify general trends and gather more valuable feedback from participants. Graphs and tables were utilized to present the results of the three remaining parameters.

Additionally, participants were questioned about their prior experience with augmented reality and assembly, ranging from none, primary, or intermediate to expert level, at the beginning of the evaluations. This inquiry aimed to explore any potential influence of experience on the evaluation data.

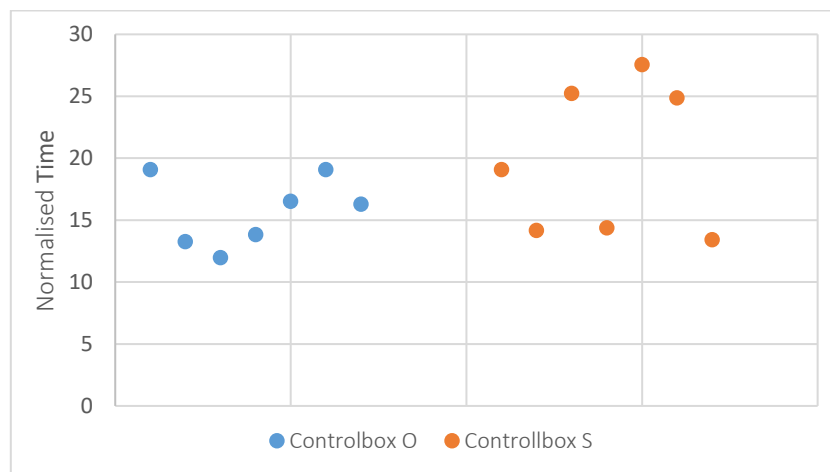
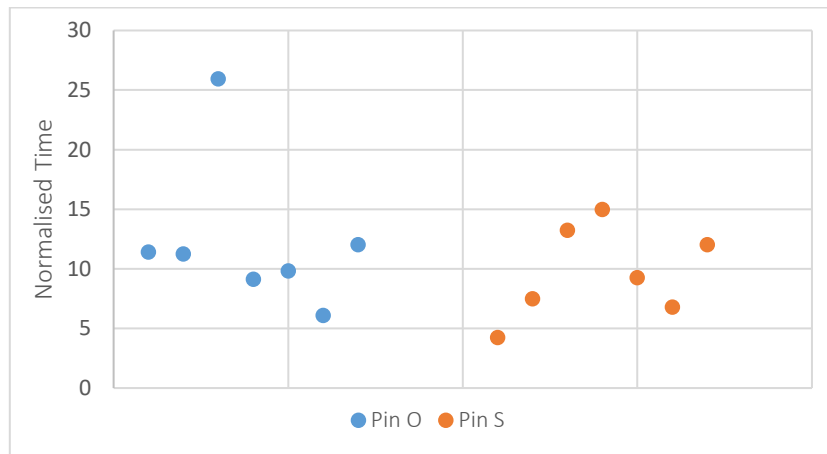
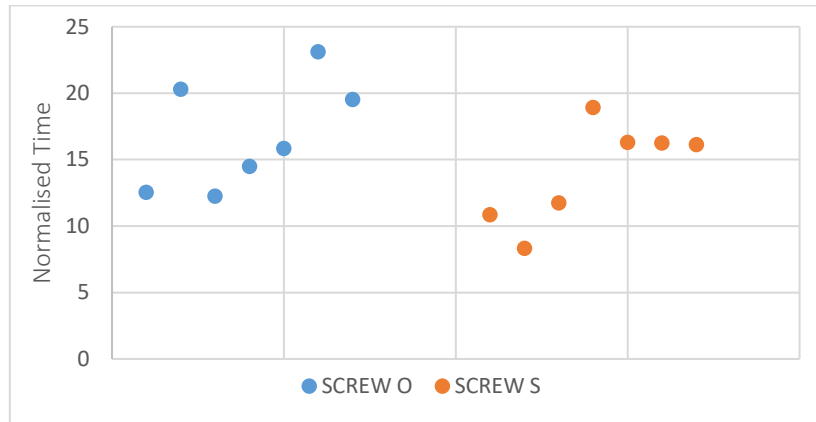
4.4.6.1 Completion time

A timer is set for each instruction, and the time taken to complete each task has been recorded. A cumulative score of the total time taken to complete all six instructions by a single participant has been calculated. The average time taken to finish all the tasks by the first set of participants is about 167sec while the average time taken by the second set of participants is 139.5 sec. This difference is due to a single participant in the first set who has been relatively slow compared to the rest. Since the person is too slow in completing all the tasks, it would Favor the opposite group in getting better results in all the tasks. We cannot directly use the values in the evaluations since the extremities affect the overall comparison. To avoid this, the completion time is normalized, i.e., each task is divided by the total time the participant takes to finish all the tasks which bring an equalization in comparisons.

Figure 4-28 shows the completion time comparisons between the original(O) and simplified(S) models of all the components used in the evaluations. The vertical axis represents the normalized time in percentage, while the horizontal is just a visualization axis. The reported times of the two AR instruction sets suggest that the simplified models do not affect the completion time. A closer look at the graphs reveals that irrespective of the original or simplified model used in the instruction, the normalized time cluster is around the same band most of the time. The only time we see more differences is for the control box, where the finish time with the simplified model is negligible over the normal range. However, when cross-checked with another parameter, the comprehensiveness of the model given by the participant was between 8.5 and 9.5, close to the ratings of the original models, indicating that simplified control box

Evaluations

models did not affect the finish times. Whenever simplified models are used in the evaluations, the participants understood the tasks as fast as when the original models were used, which did not affect the finish times.



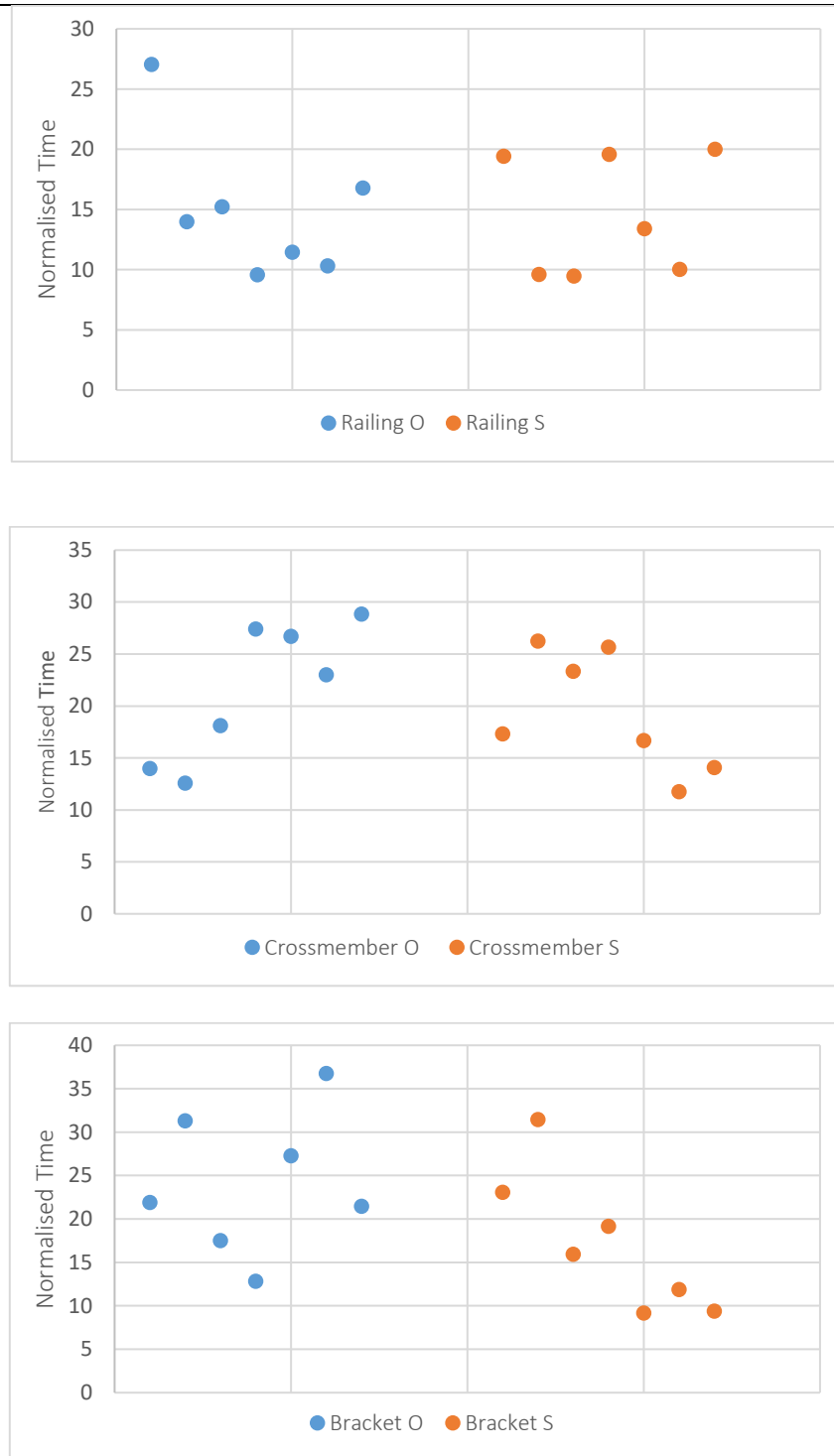


Figure 4-28. Completion times of the evaluation tasks per participant

4.4.6.2 Comprehensiveness of the model

The primary objective of these evaluations is to assess the usability of simplified 3D models within AR devices like HoloLens 2. The goal is to determine whether these simplified models are

effective in accurately identifying physical objects and conveying task instructions without causing confusion or loss of quality. Participants were asked to rate all the models used in the instructions on a scale of 1 to 10, with a focus on the model's quality in terms of shape, features, and size compared to the physical components.

Remarkably, all the ratings provided by the participants for both the original and simplified models fell between 7.5 to 10. This indicates that all 3D models used in the instructions were considered clear and suitable for performing the tasks. The quality of the models is crucial as it significantly impacts users' ability to understand the task at hand and correctly identify the physical components.

To analyse the average ratings further, a comparison has been made between the average rating of original models and their corresponding simplified versions for all components used in the instructions. The results presented in Figure 4-29 show that in 3 out of 6 cases, both the original and simplified models received the same average score. In the other three cases, the simplified models outperformed the original models in two instances, though the margin of difference was not significant.

Overall, the results suggest that the simplified 3D models were equally effective to the original models. This indicates the potential usability and suitability of these simplified models for creating effective instructions within AR environments.

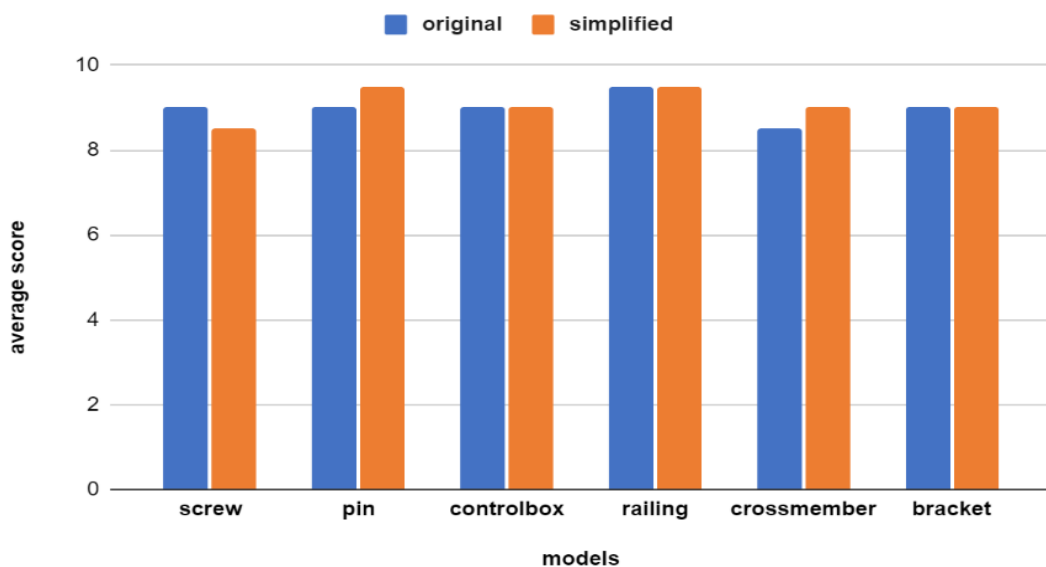


Figure 4-29. Comprehensiveness score of participants

4.4.6.3 Number of errors

The primary objective in completing any task, especially assembly tasks, is to avoid errors, as even a simple mistake can lead to defective products with significant functional issues. Table 4-5

presents the number of errors committed by participants during the evaluations. Interestingly, regardless of whether the participants used the original or simplified model, the only instruction where errors occurred was the one involving the cross-member model.

The difficulty in this specific instruction is attributed to its complexity, where the 3D model needs to be placed inside out, leading to challenges in proper identification of its placement. This issue is related to the overall depth perception of the model rather than being specific to the original or simplified model alone. The complexity of this particular task seems to be causing difficulties for participants, irrespective of the quality of the 3D model used. Identifying and addressing challenges like this in the instruction design and 3D models can help improve the overall user experience and reduce errors during assembly tasks.

Table 4-5. Errors done by participants

Models	screw	pin	control box	railing	crossmember	Bracket
Original	x	x	x	x	2	x
Simplified	x	x	x	x	3	x
*x- No errors done						

4.5 Summary

This chapter involved a comprehensive investigation into CAD model simplification techniques, encompassing both theoretical and practical aspects. The main highlights of the study are to assess the performance and efficiency of mesh-based simplification methods. Various numerical evaluations have also been conducted. These evaluations provided quantitative insights into the effectiveness of each technique. An expert visualization has been performed to determine the optimal simplification rate for CAD models depending on their vertices. To demonstrate the practical applicability of the simplification techniques, we have conducted evaluations in real-case scenarios that confirmed the fact that simplified models increase the framerate of the application without compromising the visibility and usability. In addition, we have also performed a non-biased user evaluation which highlighted the usage of simplified models at selected simplification rates in place of their original counterparts.

However, the necessity of improved simplification algorithms remains apparent, particularly when dealing with complex models that could encompass billions of vertices. This demand becomes even more pronounced in scenarios like workstation configuration, where the simultaneous display of multiple models for assembly is absolutely essential. The same challenge arises when attempting to load an entire boiler, potentially containing billions of vertices, into the AR environment during the final stage of assembly validation.

Evaluations

Addressing this limitation becomes crucial for ensuring seamless and efficient operations within these AR-based industrial workflows. We propose a novel mesh simplification algorithm for CAD models targeting redundant features in the next chapter.

Chapter 5

PROPOSED METHODOLOGY: MESH SIMPLIFICATION OF CONTINUOUS CHAINS

The proposed methodology combines a feature-based paradigm with a specialized mesh simplification algorithm. This approach centers on a specific feature known, called *continuous chains*, which is often present in complex CAD models. Continuous chains represent redundant features that can be eliminated without affecting the 3D models structure and visual presentation.

Within this context, two fundamental problems need to be solved. The first one concerns the automatic segmentation/identification of continuous chains within 3D models. The second relates to the dedicated simplification strategies that make possible their simplification/removal without altering the overall models structure and appearance.

5.1 Continuous chains: definition and problem statement

5.1.1 Continuous chains

Preparing 3D CAD models for use in augmented reality (AR) within industrial contexts is a critical preliminary step. 3D CAD models are created in various forms across different industries, and they are typically stored in commonly used CAD model file formats such as JT, STEP, IGS, DWG, and DAE. Whatever the platform used to generate CAD models, multiple phases need to be performed. They notably include:

- *Conceptualization*: The initial step involves conceptualizing the design idea. This might include sketching, brainstorming, and defining the basic shape and features of the object or system.
- *Geometry creation*: In a CAD software, the designer starts creating the geometry of the object by using fundamental shapes like lines, curves, arcs, circles, and polygons. These basic elements are manipulated to form the desired 2D or 3D representation.
- *Parametric modelling*: Many CAD tools support parametric modelling, which allows defining dimensions and constraints that control the shape and behaviour of the model. Changes made to one aspect of the design can automatically update related features.
- *3D Modelling*: The 3D model is created using techniques such as extrusion (pushing/pulling a 2D shape into a 3D form), revolve (rotating a 2D shape around an axis), lofting (blending between profiles), and more.
- *Detailing*: Adding intricate features, such as fillets (rounded edges), chamfers (bevelled edges), holes, threads, and intricate curves, to enhance the realism and functionality of the design.
- *Assembly*: If the design includes multiple parts that interact with each other, an assembly is created. This involves placing individual parts in relation to each other and defining how they fit and move together.
- *Materials and Textures*: Applying realistic materials and textures to the model can help visualize its appearance accurately. This is especially important in architecture, industrial design, and visualization.

In our research, we are focusing on identifying and addressing redundant parts known as *continuous chains* that are generated during the assembly step. The continuous chains emerge as the result of individual parts constructed separately and then interconnected to create a complete CAD model (Figure 5-1). Their role is to ensure smooth transitions between parts. Our objective is to automatically detect and optimize, in terms of number of vertices, such continuous chains.

More precisely, the continuous chains are defined as smooth regions of dense mesh triangulations, usually consisting of relatively small triangles that either round the edges of one component or connect two components together.

Figure 5-1 presents some examples of continuous chains both in wireframe and solid geometry rendering modes. In wireframe geometry we can observe the dense triangulations at the edges where the same is represented in green color in solid geometry representation. The continuous chains can be drastically simplified or even removed, without compromising the visual integrity of the assembly since they are redundant information and do not involve any assembly operations. This can lead to a significant reduction of the number of vertices.

In our work, we have analysed a database comprising 730 CAD models and have observed that nearly 500 of them contain continuous chains. The number of continuous chains in each model depends on the complexity of the model. Therefore, the identification/segmentation of such continuous chain structures that can be removed/simplified without altering the functional properties of the considered 3D models becomes of utmost importance.

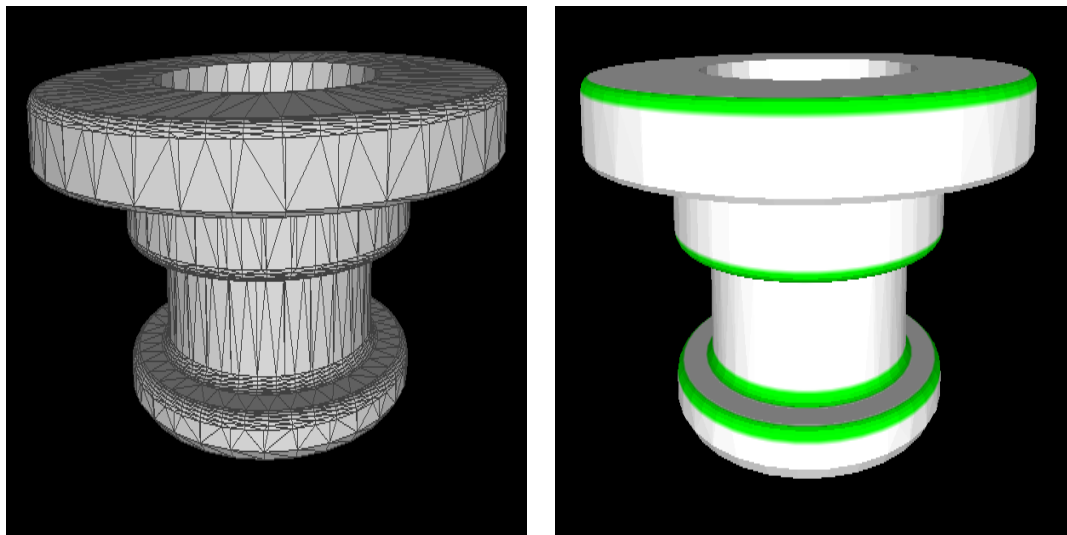


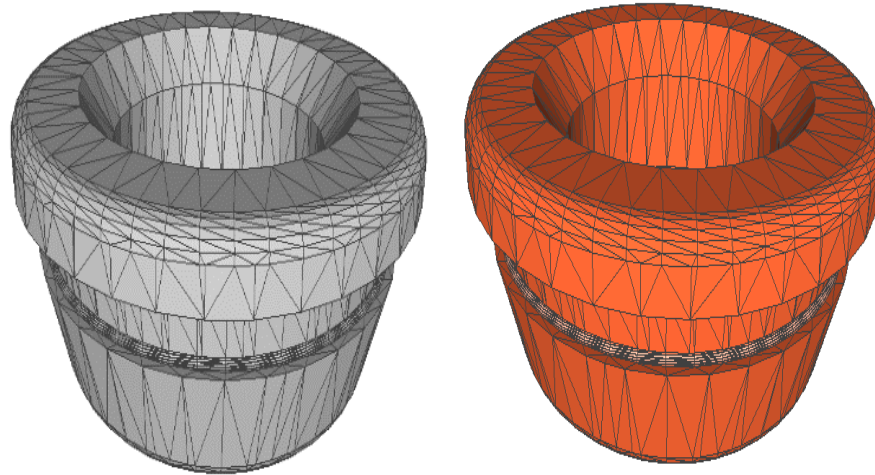
Figure 5-1. Example of continuous chains

5.1.2 Segmentation of continuous chains: curvature-based analysis

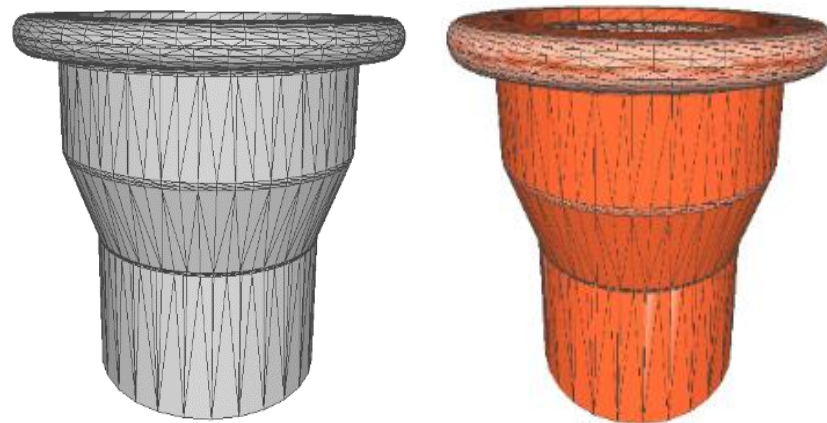
Since the continuous chains are generally curved surfaces, we have first considered the state-of-the-art curvature-based mesh segmentation technique introduced by Lavoué *et al.* in (Lavoué et al., 2005). The segmentation algorithm is based on the analysis of the curvature of each vertex. For each vertex, the curvature tensor (Ahmed et al., 2020) is calculated and the principal curvature values K_{min} , K_{max} and corresponding directions d_{min} , d_{max} are determined. They correspond, respectively, to the eigenvalues and the eigenvectors of the curvature tensor. A set of curvature clusters, defined as groups of similar vertices in the curvature space, is finally determined.

A significant limitation of this approach is related to its inability to distinguish continuous chains when a substantial deviation in curvature is present. While local optimization techniques can be employed, they are often model-specific and require manual intervention. In a more general manner, when using curvature-based methods, the algorithms performance is contingent upon the uniformity of curvature within the data.

Two examples of missed and wrong detection are presented in Figure 5-2. In both cases, the algorithm fails to accurately identify the continuous chains. In Figure 5-2a, we can see two continuous chains, one at the top and the second in the middle, but only the one in the middle is identified. Similarly, in Figure 5-2b, some of the vertices inside the continuous chain are missed (false negatives), while some other which are not in the continuous chains are identified as one (false positives).



(a) Missed identification



(b) Wrong identification

Figure 5-2. Limitations of the curvature-based continuous chain identification

To overcome such limitations, we propose a novel 2D/3D segmentation approach that leverages deep learning techniques to identify continuous chains with higher accuracy and robustness. The proposed framework is described in details in the following section.

5.2 Proposed 2D/3D continuous chain segmentation methodology

Figure 5-3 presents an overview of the proposed segmentation scheme, which includes the following three stages: 3D to 2D projection, 2D segmentation and 2D/3D mapping.

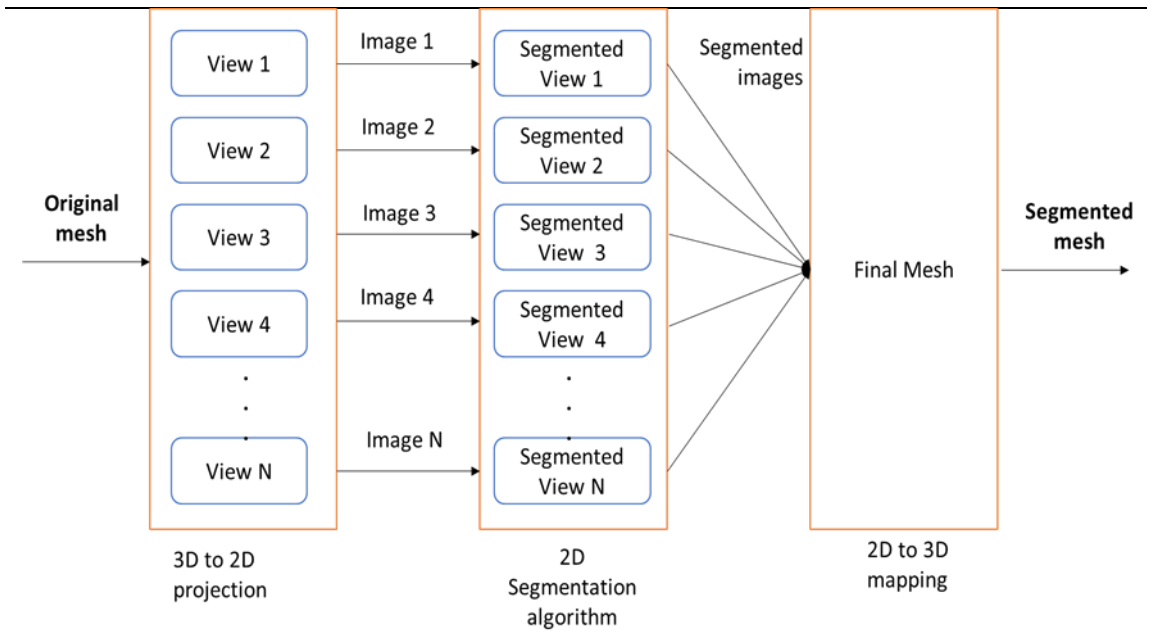


Figure 5-3. Continuous chain segmentation workflow

The principle consists in converting the 3D model into a set of 2D images corresponding to different angles of views. Each view is then segmented using 2D segmentation algorithms. After segmenting the intended features, a back-propagation process is performed in order to map the 2D segmentation masks back onto the 3D model.

The advantage of the approach comes from the 3D/2D conversion process, since in the 2D domain we can utilize advanced, and notably powerful deep learning-based image segmentation algorithms. Let us now detail the various stages involved, by starting with the 3D to 2D projection process.

5.2.1 3D to 2D projection

The primary stage of our proposed approach revolves around the establishment of an extensive repository containing 2D images corresponding to views extracted from 3D models. This foundational step entails the development of a systematic procedure that seamlessly translates the multi-dimensional nature of 3D models into informative 2D visualizations, captured from diverse viewpoints. To accomplish this, we have employed a dedicated Unity capture framework.

The Unity framework takes an .obj file as its input, accompanied by its corresponding .mtl and texture files. This approach ensures that the original format and characteristics of the model are preserved throughout the image acquisition process. To achieve this, we employ PyMeshlab, an integral component derived from the MeshLab framework(Cignoni et al., 2008). The resulting

database comprises three distinct sets of images (Figure 5-4), each serving to a specific purpose in our methodology.

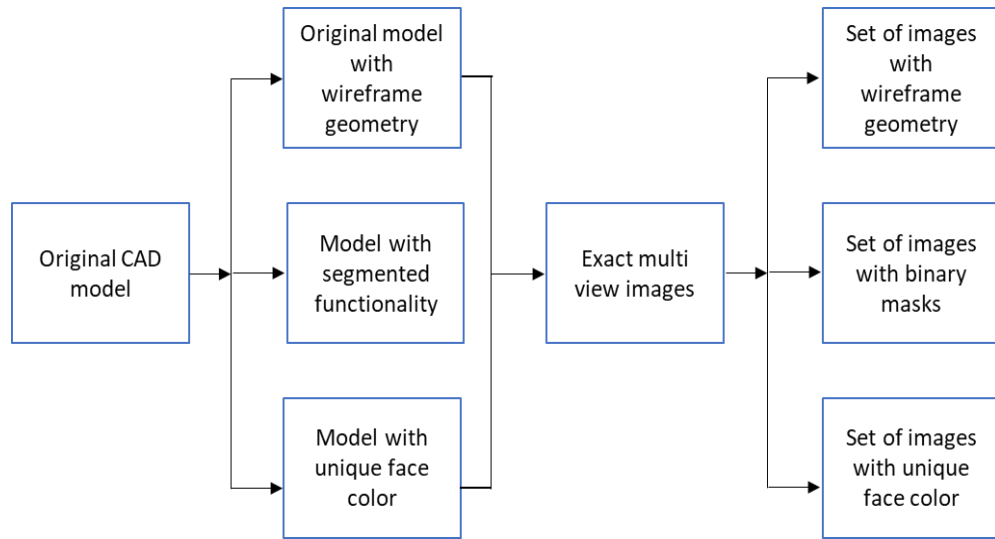
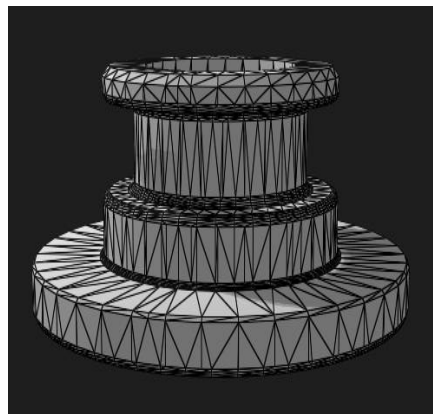


Figure 5-4. Database preparation

The first set of images correspond to wireframe-rendered 3D model projections. The second one represents the 2D projections of the continuous chains (segmentation masks) previously determined in the 3D space. Finally, in the third set of images, a unique, randomly determined color, is assigned to each face of the mesh. The color images notably serve to facilitate the 2D/3D mapping process. The three different projection images generated for a given model (and for the same angle of view) are illustrated in Figure 5-5.



(a)

Wireframe-rendered model



(b)

Continuous chain mask



(c)

Unique color per each face

Figure 5-5. The three 2D projection images generated for a given model

Prior to projection, the models' gravity centre is placed into the origin of the Cartesian coordinate system (as defined in the Unity environment) and its axes of inertia are aligned with the x , y and z directions. The projection process is guided by strategic camera placements, designed to capture the model from a diverse array of viewpoints. To this purpose, we have employed the Fibonacci sphere algorithm (Figure 5-6), which offers a solution for positioning the cameras around a sphere in a manner that ensures a uniform distribution of N camera points. In this way, we obtain a balanced camera distribution that optimally represents the model dimensions. For experimentation purposes, a set of 50 camera positions has been explicitly established. Of course, a higher number of viewpoints can also be considered.

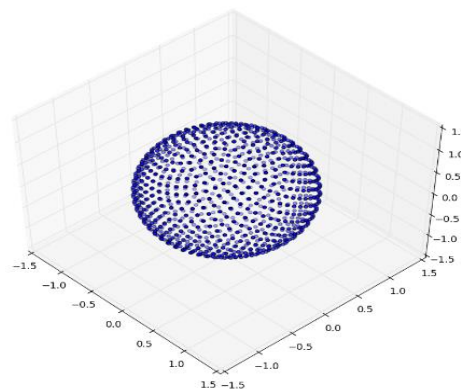
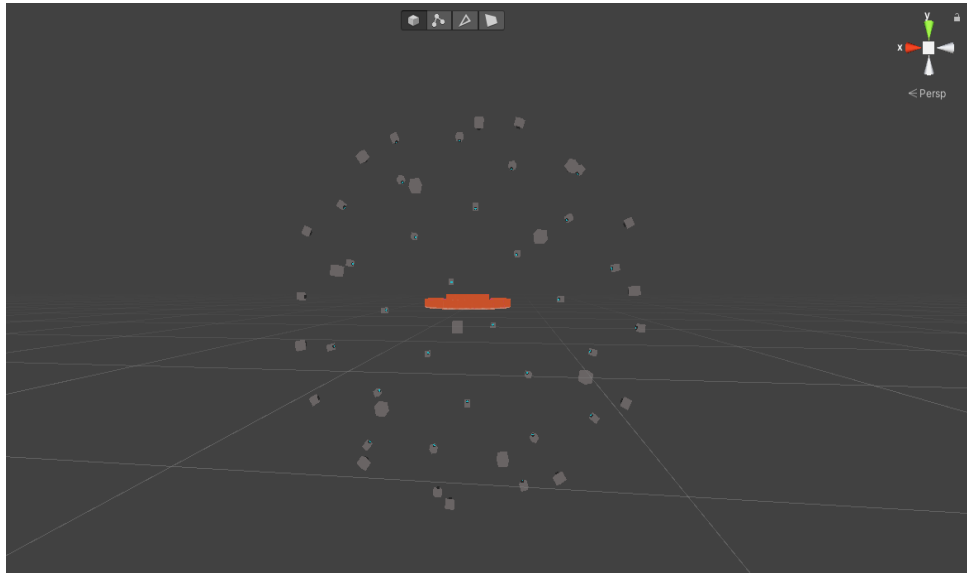


Figure 5-6. Camera positioning (Fibonacci sphere algorithm)

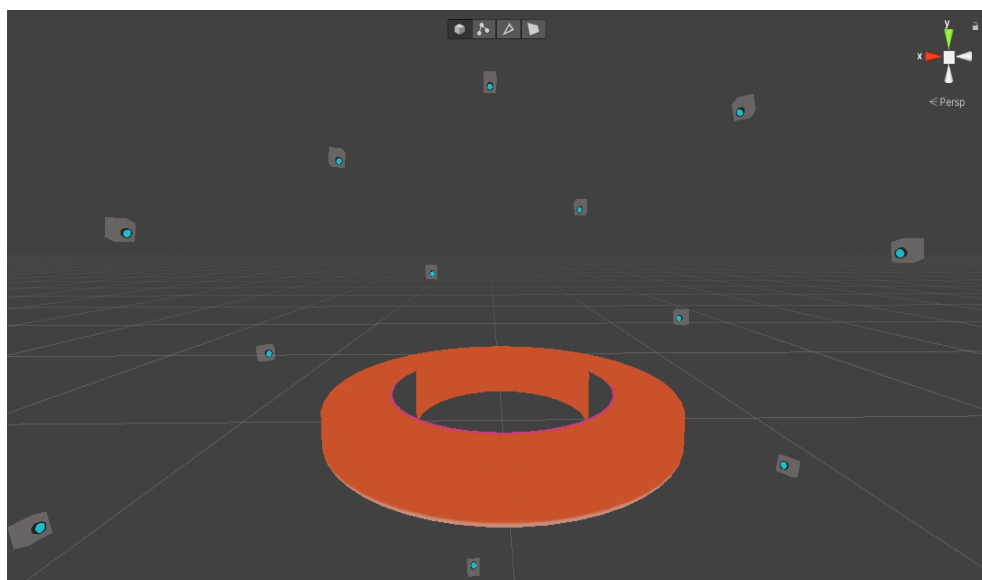
Proposed methodology: Mesh simplification of continuous chains

Concerning the implementation details, the illumination conditions, including light intensity and direction, are left at default settings. Some other additional parameters include *shaders*, which indicate the rendering techniques employed, the appearance of the background which, and the *material type*, which specifies the manner in which surface properties like color and texture are defined. These parameters are maintained at their default settings within the Unity environment to ensure consistency and comparability across the dataset.



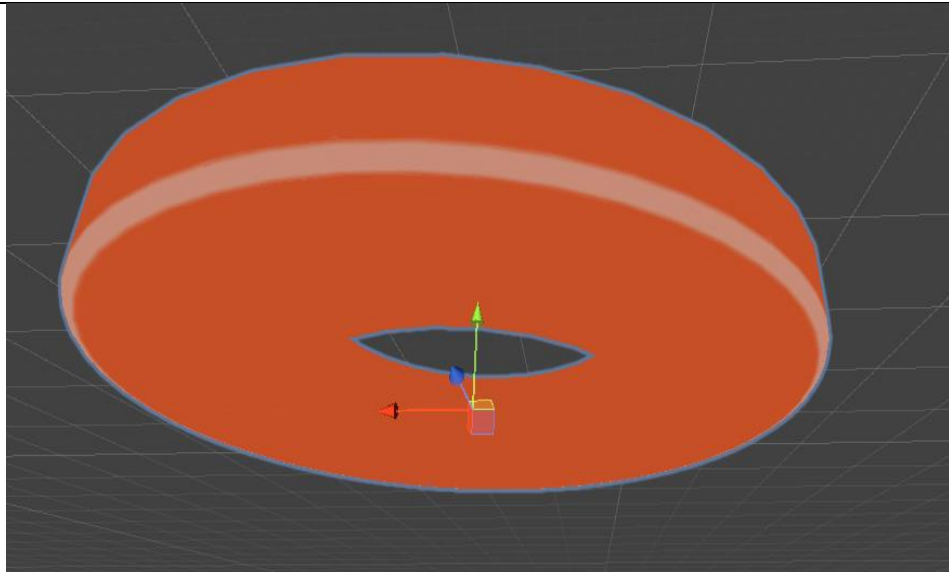
(a)

Camera distribution for capturing a set of images



(b)

Camera distribution for capturing a set of images (zoom in)



(c)

Object rendered from one camera position

Figure 5-7. Visual representation of camera distribution

The obtained 2D wireframe-rendered and ground truth continuous chain masks are further used for automatic segmentation purposes.

5.2.2 2D segmentation of continuous chains

5.2.2.1 Deep learning-based segmentation technique: state of the art

Segmentation of 2D images using deep learning techniques has become a powerful tool in computer vision and image processing. Deep learning-based segmentation methods have proven to be highly effective in a wide range of applications. Among some of the most popular segmentation models developed in recent years, let us mention:

- *FCN (Fully Convolutional Network)* (Long et al., 2014): FCN was one of the pioneering architectures for semantic segmentation. Unlike traditional CNNs, which produce spatially reduced feature maps, FCN replaces fully connected layers with convolutional layers, enabling it to produce pixel-wise predictions for segmentation. It uses up-sampling layers to recover the spatial resolution of the input image, making it capable of producing dense segmentation maps. FCN laid the foundation for subsequent architectures like U-Net and SegNet, which improved upon its design for various segmentation tasks.

-
- *SegNet* (Badrinarayanan et al., 2015): SegNet is another CNN architecture designed for semantic segmentation. It was initially developed for road scene understanding in autonomous driving applications. SegNet's key innovation lies in its use of an encoder-decoder architecture. The encoder processes the input image and extracts features, while the decoder reconstructs the segmented output. Importantly, SegNet uses the max-pooling indices from the encoder to perform efficient up sampling in the decoder, which helps retain spatial information. It is known for its memory-efficient design and has been used in tasks like object detection, scene parsing, and medical image analysis.
 - *Mask R-CNN* (He et al., 2017): Mask R-CNN is an extension of the Faster R-CNN architecture (Ren et al., 2015), designed for instance segmentation. It simultaneously performs object detection and pixel-wise instance segmentation in images. Mask R-CNN extends Faster R-CNN by adding a branch for predicting segmentation masks for each detected object. It employs a Region Proposal Network (RPN) to generate region proposals, and then a mask head network to predict segmentation masks for these regions. This architecture is widely used in applications that require precise object instance segmentation, such as object tracking, human pose estimation, and interactive image editing.
 - *DeepLabv3+* (Zeng et al., 2020): DeepLabv3+ is an evolution of the DeepLab family of CNN architectures for semantic segmentation. It excels in capturing fine details and handling objects at different scales. DeepLabv3+ uses dilated convolutions to increase the receptive field of the network without down sampling the input excessively. This allows it to maintain high-resolution feature maps and capture fine details. It also employs a feature pyramid network (FPN) to combine multi-scale features, making it robust in handling objects of various sizes and achieving state-of-the-art results in tasks like object segmentation and boundary detection.
 - *U-Net* (Ronneberger et al., 2015): U-Net is a convolutional neural network (CNN) architecture designed for semantic segmentation tasks, particularly exploited in the field of medical image analysis. It is known for its U-shaped, encoder-decoder architecture.

The UNet model proved to be highly effective in applications where a limited amount of training data is available. For this reason, in our work we have adopted the UNet Model, recalled in the following section.

5.2.2.2 Adopted UNet approach

When confronted with the challenge of limited training data, UNet has consistently exhibited remarkable performance when compared to alternative methods. We have adopted the UNet (Ronneberger et al., 2015) as our primary model for the segmentation task, specifically

targeting the identification of continuous chains in the 2D domain. To achieve this, we leverage a pre-trained UNet model (Ronneberger et al., 2015) that has been originally designed for the segmentation of medical images and the detection of abnormal regions.

UNet is a convolutional neural network (CNN) architecture commonly used for image segmentation tasks. It was originally developed for biomedical image segmentation but has found applications in various fields, including computer vision and remote sensing. The name "UNet" is derived from its U-shaped architecture, which resembles the letter "U". Let us briefly recall the UNet architecture (Figure 5-8), which includes three main components, which are the encoder, the bottleneck layer and the decoder:

Contracting Path (Encoder): The top part of the U-shape consists of a series of convolutional layers followed by max-pooling layers. This portion is responsible for capturing the coarse-grained features of the input image while reducing its spatial dimensions.

Bottleneck: At the center of the network, there is a bottleneck or central layer with a high number of feature maps. This layer captures important contextual information.

Expansive Path (Decoder): The bottom part of the U consists of a series of up sampling layers and convolutional layers. This portion gradually increases the spatial dimensions of the feature maps, allowing the network to generate a segmented output that is of the same size as the input.

In addition, UNet includes a set of so-called *skip connections*, which connect corresponding layers in encoder and decoder. These connections help in preserving fine-grained details and spatial information during the up sampling process.

The final layer of the network typically consists of a convolutional layer with a sigmoid activation function, which produces pixel-wise predictions indicating the presence or absence of the object of interest in the input image.

The UNet architecture is particularly well-suited for tasks where precise localization of objects within an image is essential. Various researches in the field customize UNet for specific applications by adjusting the number of layers, filters, and other architectural elements to achieve the optimal performance for a given task. Additionally, pre-training on a larger dataset, as mentioned in the original passage, can be beneficial when dealing with limited training data.

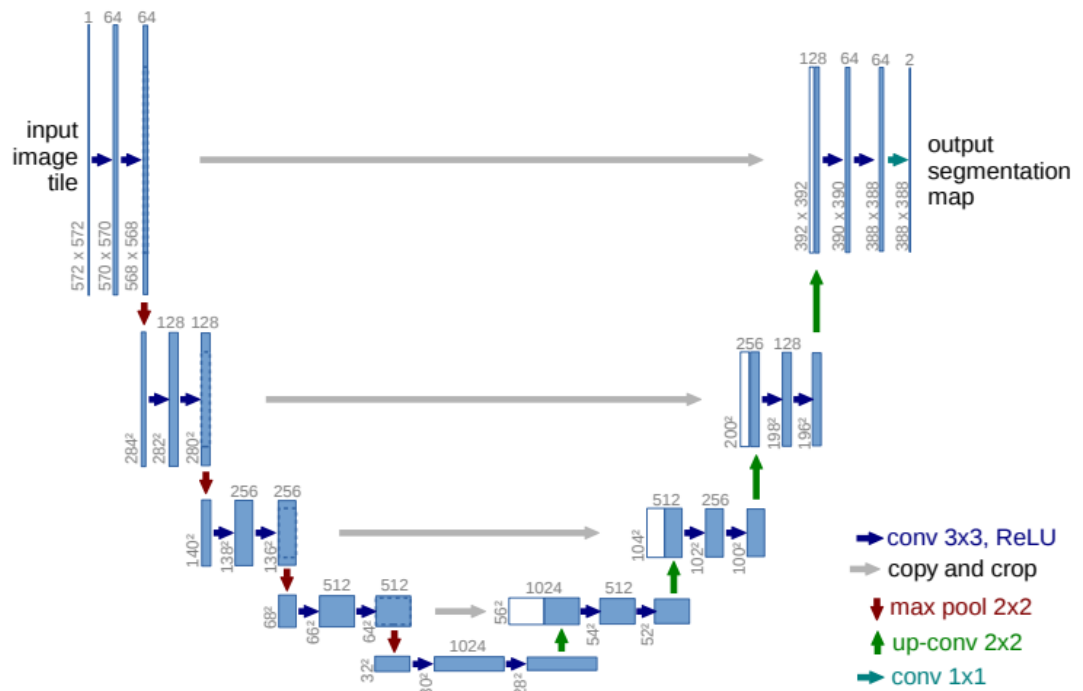


Figure 5-8. UNet Architecture (Ronneberger et al., 2015)

The wireframe-rendered images are fed as input to the UNet model, while the output of the network corresponds to the detected continuous chain segmentation mask.

5.2.3 2D to 3D mapping

The process of segmentation within the 2D domain yields a set of binary masks, each capturing the continuous chain functionality from a specific angle. To bridge the gap between the 2D and 3D, we deploy a distinctive color identity system. This system involves assigning a unique color to each face of the 3D CAD model. Consequently, during the conversion of the 3D model into a 2D image, every face color translates to multiple pixels within the image.

In the resulting segmented images, we pinpoint the pixel colors that correspond to the segmented region. These specific pixel colors are then traced back to their originating face numbers within the 3D domain (Figure 5-9). This process enables us to establish a direct link between the segmented region in the 2D representation and its corresponding region within the intricate 3D model. All the pixels that are classified as continuous chains are stored separately and are traced back to the face color with the help of pixel number.

By executing this mapping for a range of distinct views to achieve an all-encompassing identification of the segmented continuous chain within the expansive 3D domain. This orchestrated amalgamation of results effectively bridges the gap between 2D segmentation and

3D model interpretation, enhancing our capacity to comprehend and analyze the continuous chain's behavior from a multi-dimensional perspective.

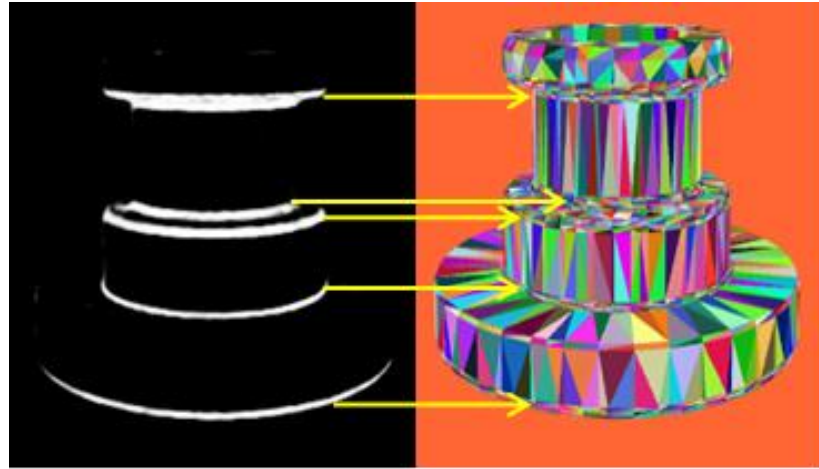


Figure 5-9. 2D to 3D mapping

5.3 Simplification using graph traversal technique

For this particular case of continuous chains, since they include redundant information, they can be completely removed. However, this process needs to be done very carefully, in order to avoid disrupting the original structure of the model.

The key novelty of our algorithm lies in the approach used to suppress the continuous chains. Instead of relying solely on segmentation techniques to identify these chains, we employ a method that focuses on extracting the vertices situated on the same vertical plane and subsequently merging them. This novel process is achieved through an edge collapse operation. To implement this edge collapse, we employ a graph traversal technique (Figure 5-10), which enables us to identify the vertices lying on the same vertical plane and their corresponding boundary vertices. The metric used for the graph traversal algorithm is the shortest Euclidian distance between corresponding vertices.

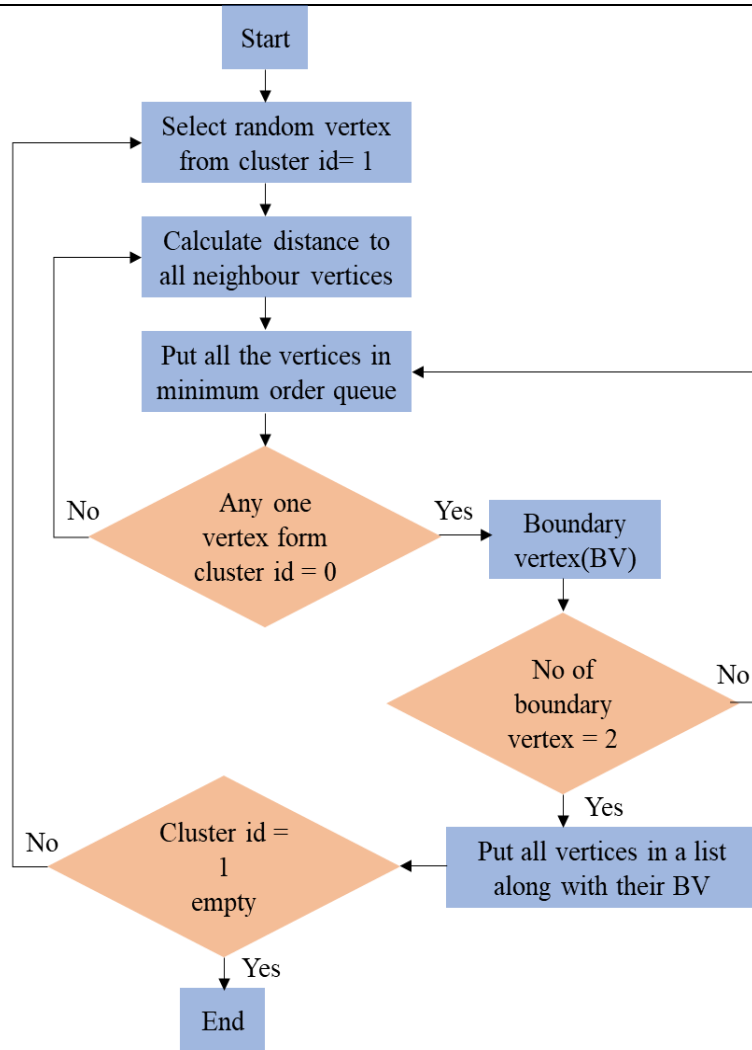


Figure 5-10. Graph traversal technique

The flowchart in Figure 5-10 provides a deeper understanding of how the vertices on the vertical planes are identified along with their respective boundary vertices. There are two types of vertices, cluster id 1 indicating the vertex belongs to a continuous chain and cluster id 0 indicating it is a non-continuous chain vertex. At first, we select a random vertex from the continuous chain and traverse through the vertices with the help of minimum distance and merge them together except for boundary vertices. The two boundary vertices are collapsed at their centroid creating a single point (Figure 5-11). We continue the process until all the vertices from cluster id 1 are empty. We can observe that the overall shape and structure of the model is preserved. The band of continuous chains is simplified into a single chain uniformly along the model.

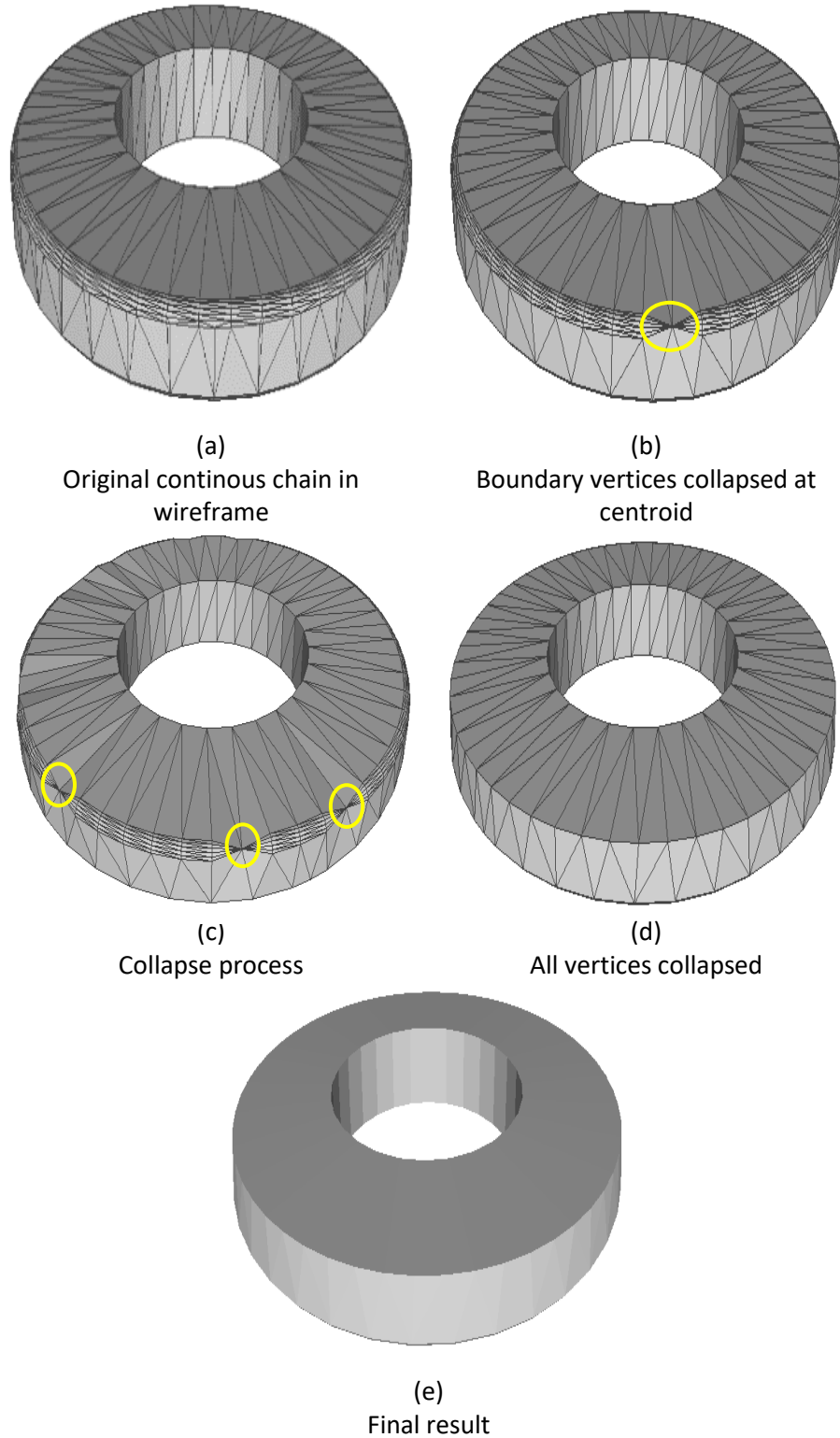


Figure 5-11. Continuous chain removal process

5.4 Experimental results

5.4.1 Database pre-processing

Preparing accurate, high-quality training data is crucial for any machine learning models. Within the context of segmentation, it is important to dispose of correct and reliable ground truth data that can serve to an effective learning process. To this purpose, we have considered a semi-automatic continuous chain segmentation procedure, whose role is to provide the necessary ground truth database. To overcome the limitations by Lavoué et al. (Lavoué et al., 2005) we have done a series of operations on the results of segmentation to have perfect ground truth.

K means clustering

Certainly, it seems that in the Lavoué *et al.* (Lavoué et al., 2005) algorithm, continuous chains are detected primarily based on curvature. However, there are situations where the algorithm might fail to correctly identify some vertices that are actually part of continuous chains. In these cases, it is possible that neighbouring vertices are correctly identified, but certain vertices within the chain are left undetected. This scenario represents a case of incorrect detection, where some vertices belonging to continuous chains are wrongly categorized, and this issue can potentially be resolved using k-means clustering. By considering the characteristics of neighbouring vertices, clustering can help group the wrongly identified vertices with their correct counterparts, ultimately leading to the identification of complete continuous chains.

K-means clustering (Ahmed et al., 2020), when applied in this context, can aid in refining the detection process by taking into account the spatial relationships between vertices. This approach helps to correct misclassifications and ensure that the entire continuous chain is accurately identified, thus improving the overall accuracy and completeness of the chain detection process within the algorithm.

Manual adjustments

In the manual adjustment process, we aim to address two specific issues related to the detection of continuous chains. The first issue, as demonstrated in previous examples, arises when there are multiple continuous chains with significant curvature variations. This scenario can lead to missed detections. The second issue involves instances of incorrect detection, where a vertex is erroneously identified as belonging to a continuous chain when it does not. To tackle both of these issues, we have chosen to utilize a tool called Blender.

Blender is a versatile and powerful 3D computer graphics software application. One of its strengths is its ability to interactively and precisely manipulate 3D models and their components. By leveraging Blender, we can manually address and correct these issues. This involves a detailed

process of working with each vertex, inspecting their placement and characteristics, and making the necessary adjustments to rectify both missed detections and incorrect identifications.

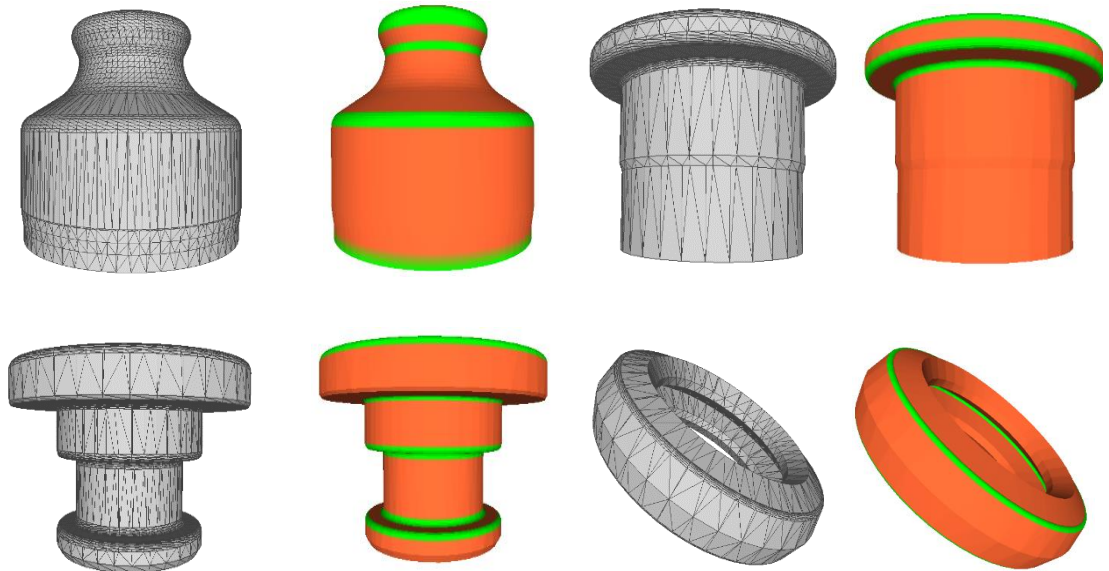
Segmentation into parts

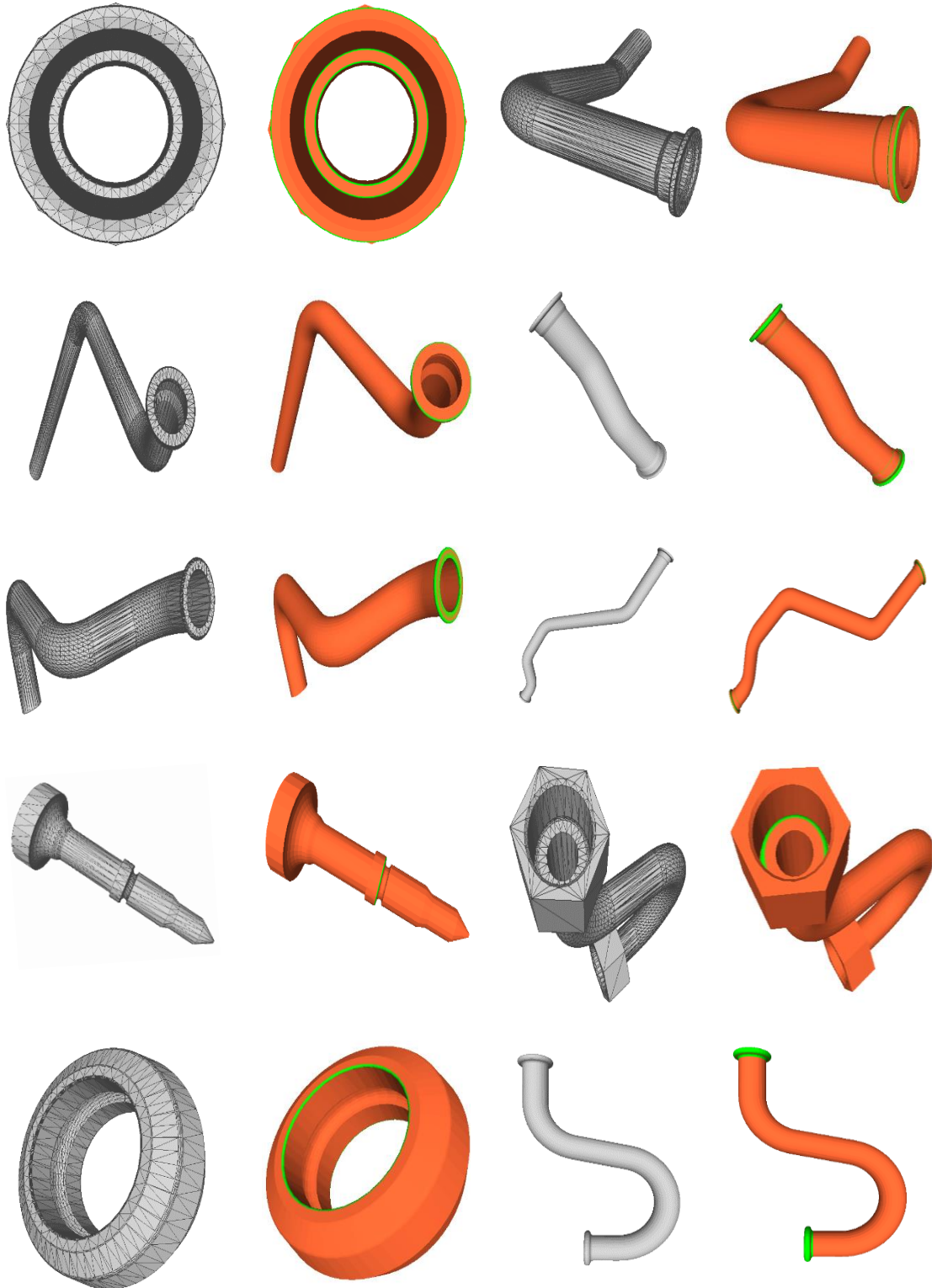
The segmentation process described here involves the use of the CGAL (Computational Geometry Algorithms Library) (CGAL, 2020), and is distinct from identification. Segmentation, in this context, refers to the act of dividing or cutting a CAD model into distinct pieces or segments. This segmentation step serves multiple purposes.

Enhancing Visibility of Continuous Chains: By dividing the CAD model into segments, some continuous chains may become more visible and distinguishable. This can be particularly useful in scenarios where continuous chains are complex or overlapping within the original model. Segmenting the model can expose these chains, making them more accessible for further processing.

Improving Training Data: Segmentation can also serve to improve the quality and diversity of our training data. Each segment can represent a different part or component of the CAD model. This can lead to a more comprehensive dataset, which is valuable for training machine learning models, especially when you want to capture a wide range of variations in your data.

The main reason that we have considered solely 15 3D CAD models from the database is the manual adjustments that are needed to have the correct mask for training. Figure 5-12 shows the original wireframe models along with their rectified continuous chains (represented in green color).





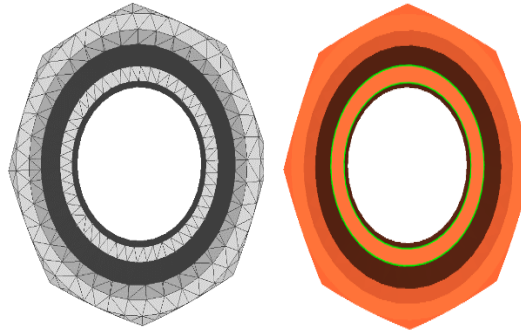


Figure 5-12. 3D CAD models in database

5.4.2 Database creation

As discussed in section 5.2.1 we use a Unity framework (3D to 2D projection) to obtain the database for segmentation. There are 50 views of each model indicating we have a total of 750 original images for our 15 CAD models. In this step we generate two sets of images required for our segmentation algorithm. The first set of images are original 3D models along with their wireframe geometry. For the second set of images the continuous chains, we adjusted manually the 3D models to have the correct mask for training purpose.

Figure 5-13 shows example views of the pipe model along with its corresponding continuous chain mask. Note that the camera views can be increased in case of necessity.

After the acquisition of three sets of images they are cropped according to the size of the model inside the image. Every image cropping is different and a padding of at least 2 rows is left for each image. Firstly, the wireframe original images are cropped and then their corresponding masks, unique face color images follow according to the saved parameters of their originals. The database is now ready for further implementations.

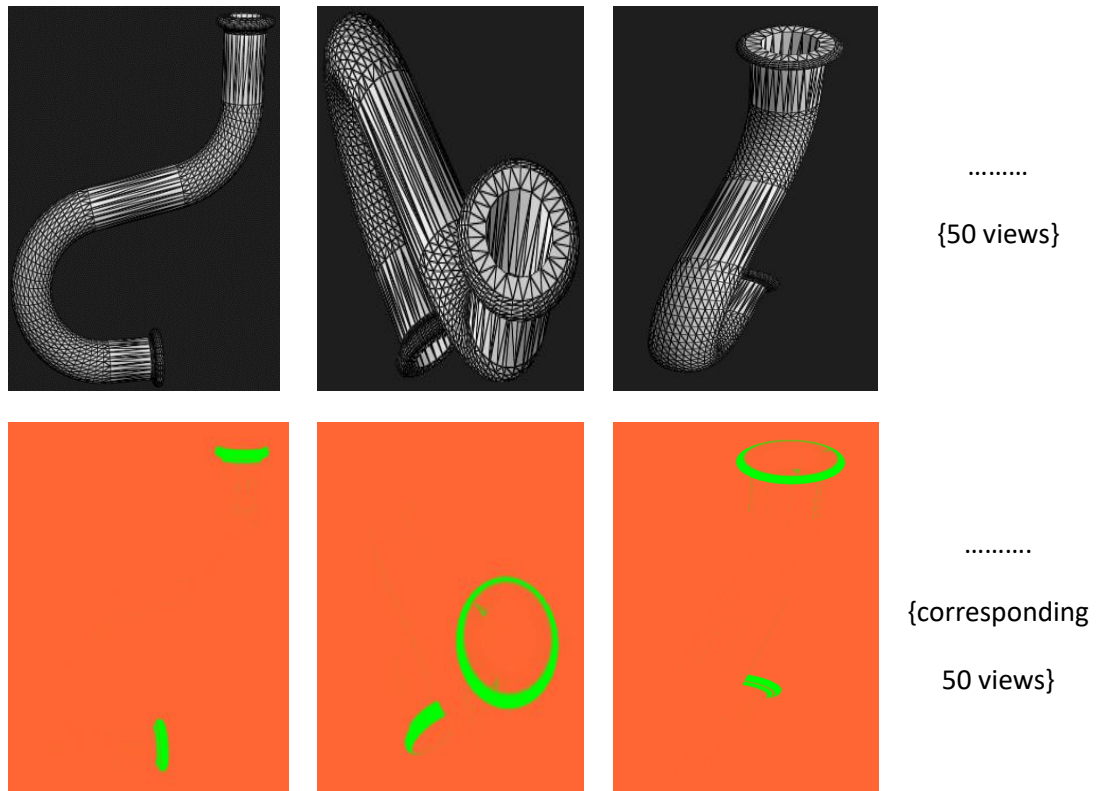
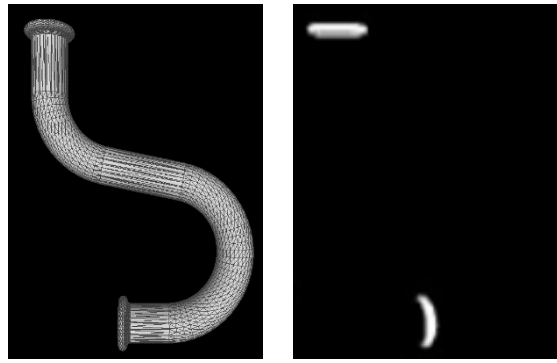


Figure 5-13. Different views inside database

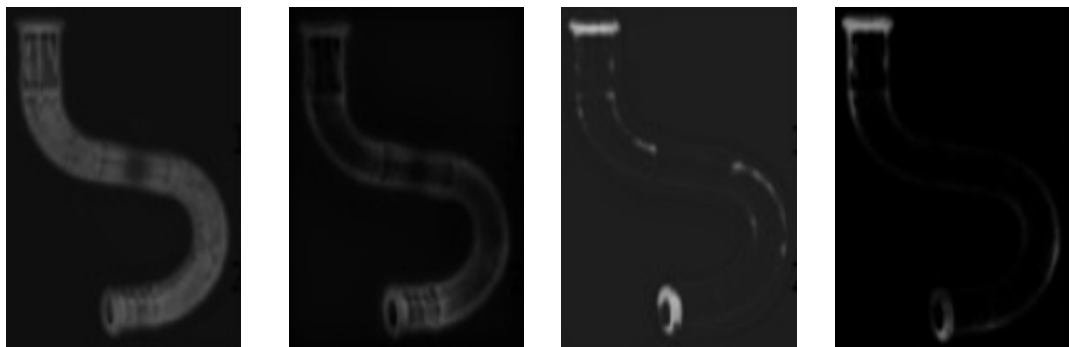
5.4.3 UNet 2D Segmentation of continuous chains

5.4.3.1 Training and validation conditions

From the entire database of 15 3D models, we have selected a training set consists of 13 models, comprising a total number of 650 images, while 2 additional models have been reserved for testing. In our experimentation, we considered the following default parameters for our UNet model: Learning Rate: 0.001, Optimizer: SGD, Batch Size: 8, Saturation Epoch: 12 and Loss Function: MAE. Figure 5-14 illustrated the evolution of training for the pipe model.



(a) Ground truth



1st epoch

2nd epoch

4th epoch

8th epoch

(b) Number of epochs

Figure 5-14. Evolution of training

For an objective evaluation of the segmentation results, we have considered the various dedicated metrics usually considered in the state of the art.

5.4.3.2 Objective evaluation

For quantitatively assessing the performance of segmentation algorithms, various metrics are available in the state of the art.

IOU score

Among them, let us first mention the popular IOU score. IOU stands for *Intersection over Union*, and is a commonly used metric in deep learning and computer vision for evaluating the performance of object detection and image segmentation models. It measures the overlap between the predicted bounding boxes or segmentation masks and the ground truth annotations. IOU is also known as the Jaccard index.

The IOU score is calculated as described in the following equation:

$$IOU = \frac{\text{Intersection area}}{\text{Union area}} = \frac{\text{Area of overlap between prediction and ground truth}}{\text{Area of prediction} + \text{Area of ground truth} - \text{Intersection area}} \quad [1]$$

The IOU score typically ranges from 0 to 1, with 0 indicating no overlap between the prediction and ground truth and 1 indicating a perfect match. In the context of deep learning and computer vision, IOU is often used as a loss function or evaluation metric during training and testing of object detection and segmentation models. It helps measure how well the model is able to accurately localize and segment objects within an image.

In the case of a database of images, in order to obtain a global evaluation of the segmentation performances, an average IOU score, often referred to as the mean IOU or mIOU, model is retained.

The mIOU scores obtained over successive epochs on our database are reported in Figure 5-15. Initially, the training IOU score commenced at a modest level of 0.15, but a remarkable increase in performance has appeared over the subsequent epochs. This increase is true for both training and validation datasets.

In addition, the gap between the training and validation IOU scores, which amounted to a mere 0.05 demonstrates the effective generalization capability of the model and suggests that the model does not overfit.

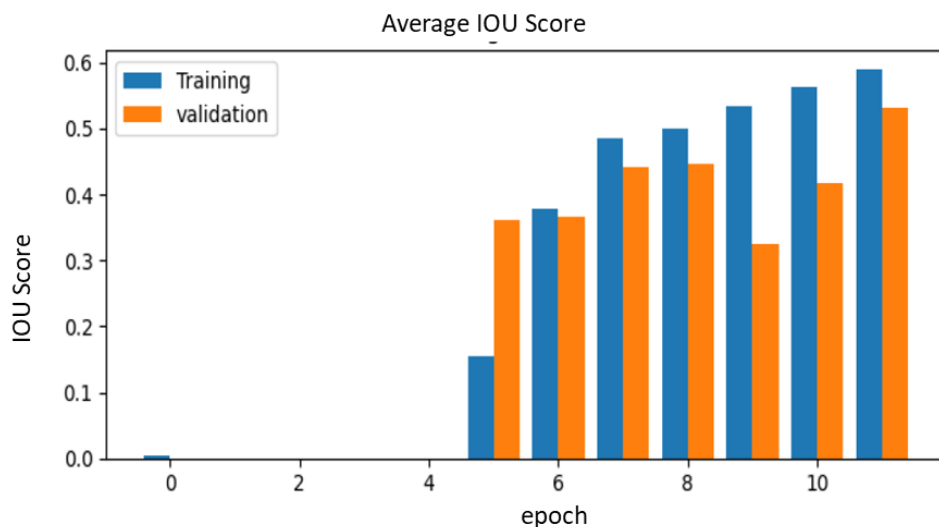


Figure 5-15. Average IOU score for UNet

Loss graph

The loss function measures the error or mismatch between the model's predictions and the actual target values. The goal during training is to minimize this loss.

Figure 5-16 plots the training and validation losses during training. The important difference in the training and validation graphs might be due to the default parameters like learning rate which needed to be adjusted.

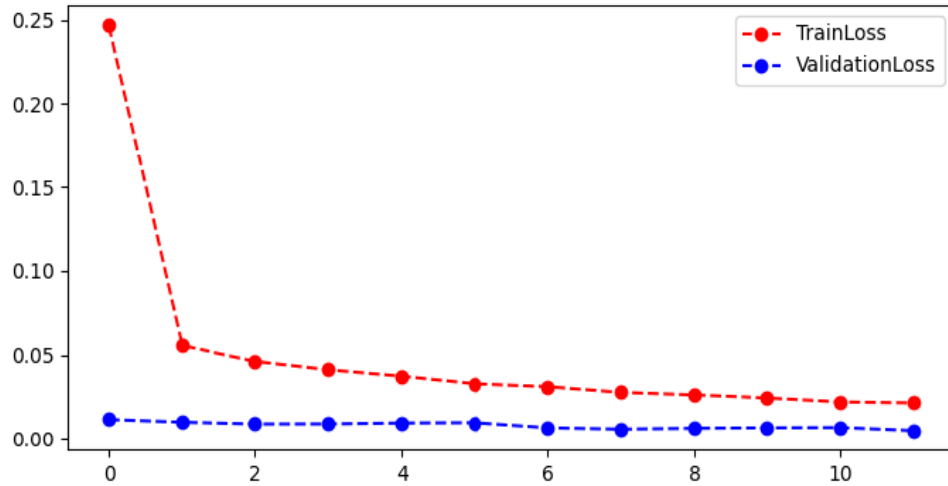
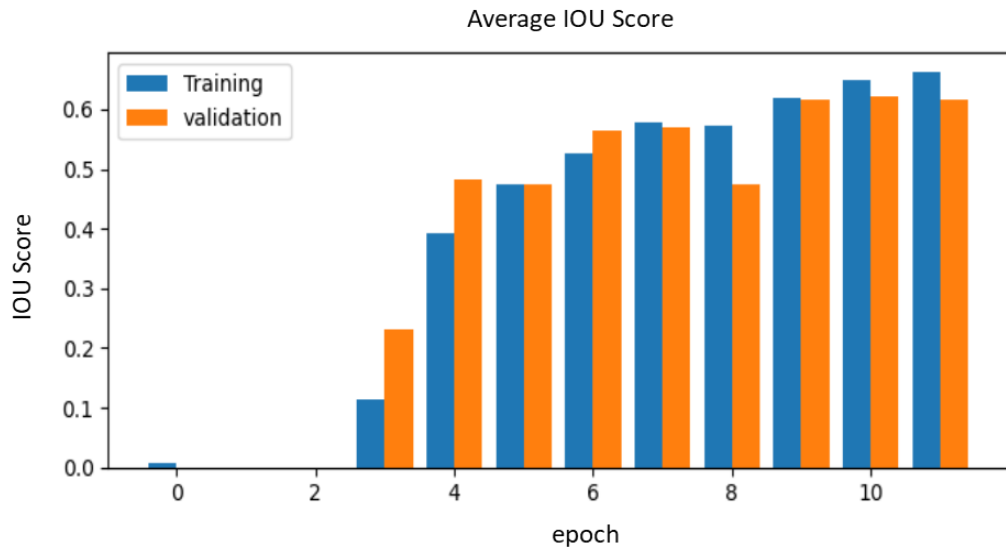


Figure 5-16. Loss graph

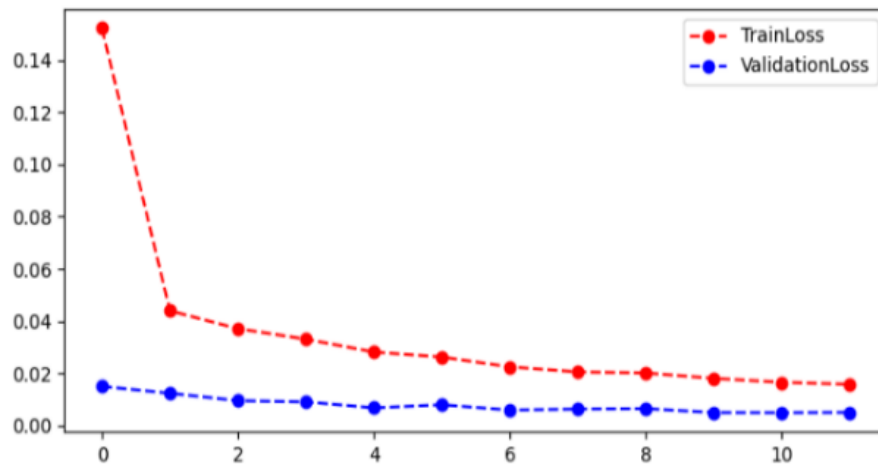
5.4.3.3 Optuna optimization

To optimize the default parameters of the UNet model for CAD model segmentation, we leveraged the power of Optuna (Akiba et al., 2019), an automatic hyperparameter optimization software framework specifically designed for machine learning tasks. Optuna dynamically constructs search spaces for hyperparameters, allowing us to efficiently determine the best configuration for our model.

By employing Optuna for hyperparameter tuning, we have achieved an optimized configuration that significantly improved the performance of our UNet model with respect to the original UNet parameters. The enhanced model can now be confidently utilized in our CAD model segmentation process. Figure 5-17 illustrates the evolution of the training and validation losses during the optimization process. We observe a strong correlation between the training loss and the validation loss, indicating that the model is well-trained and performs well on both training and validation datasets. It also illustrates the significant improvement in model performance achieved after incorporating Optuna for hyperparameter tuning. The average IOU has improved notably where we can see that the training and validation score close to each other when compared to the original score. Specifically, in terms of loss, Optuna initiated the process with a loss of 0.15, following hyperparameter tuning, while the default UNet model had a loss of 0.25. Although, by the end, the loss gap between the default model and the Optuna-tuned model narrowed to just 0.005, the model's performance has been substantially enhanced, as evidenced by the accuracy graph.



(a)



(b)

Figure 5-17. Evaluation metrics after Optuna

To illustrate this improvement, let us consider the example presented in Figure 5-18. Here, two continuous chains are present: one at the top of the pipe and another at the bottom. In the default configuration, at the 10th epoch, only the top continuous chain is successfully segmented, while the bottom chain remains undetected. However, when the Optuna-tuned model is used, a notable transformation becomes apparent at the level of 10th epoch. Here, not only the top continuous chain is accurately detected, but the model has also begun to identify the bottom continuous chain. This visual representation underscores how Optuna's hyperparameter optimization has substantially improved the model's ability to identify and segment both continuous chains simultaneously, marking a significant enhancement in its performance.

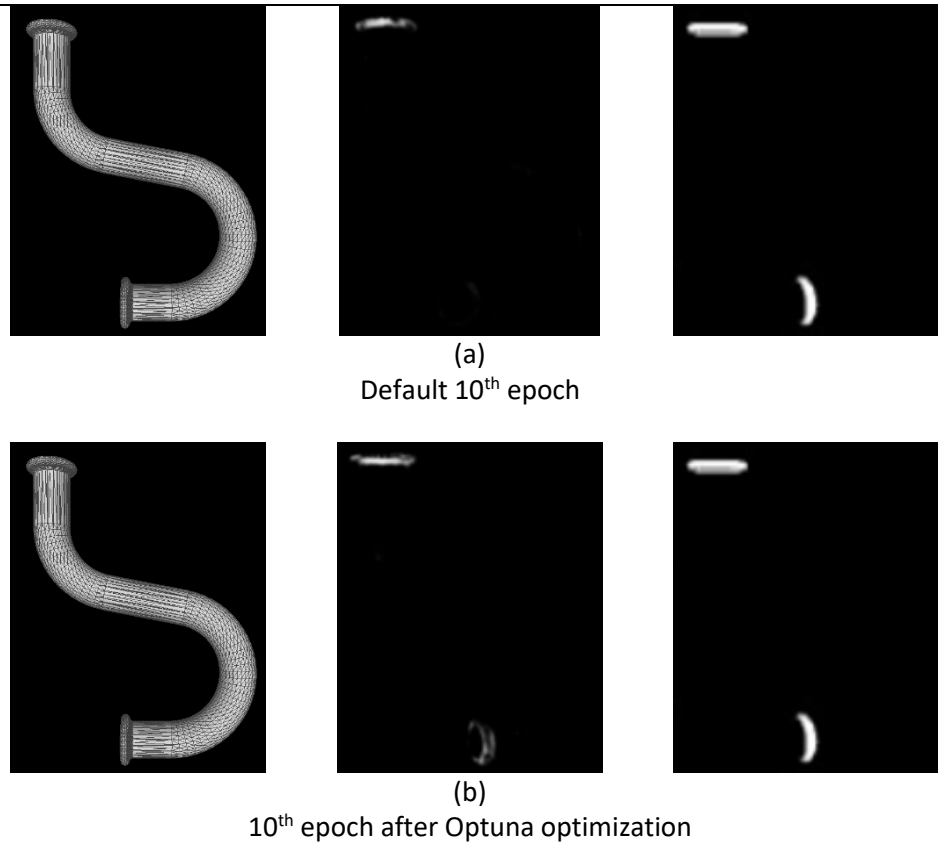


Figure 5-18. Visual comparison of before and after optuna

Figure 5-19 and Figure 5-20 presents the final results of our segmentation process on the two validation models, displaying the original views alongside the accurately identified continuous chain functionalities. The network successfully detects the continuous chains in this particular view of the model. Note that we have 50 different views of each CAD model which will be later used to detect all the faces involving continuous chains.

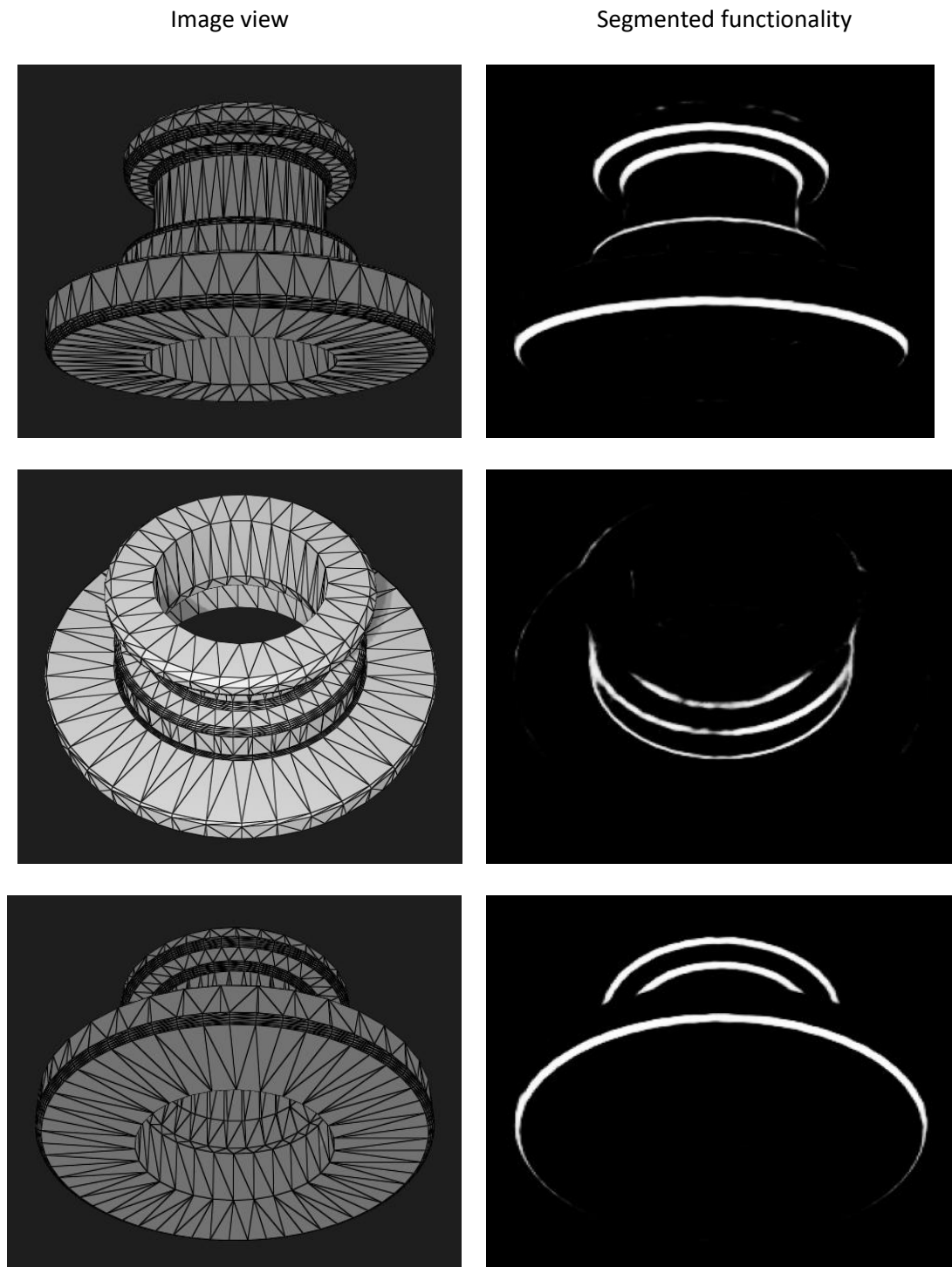


Figure 5-19. Segmented views of the Fitting model

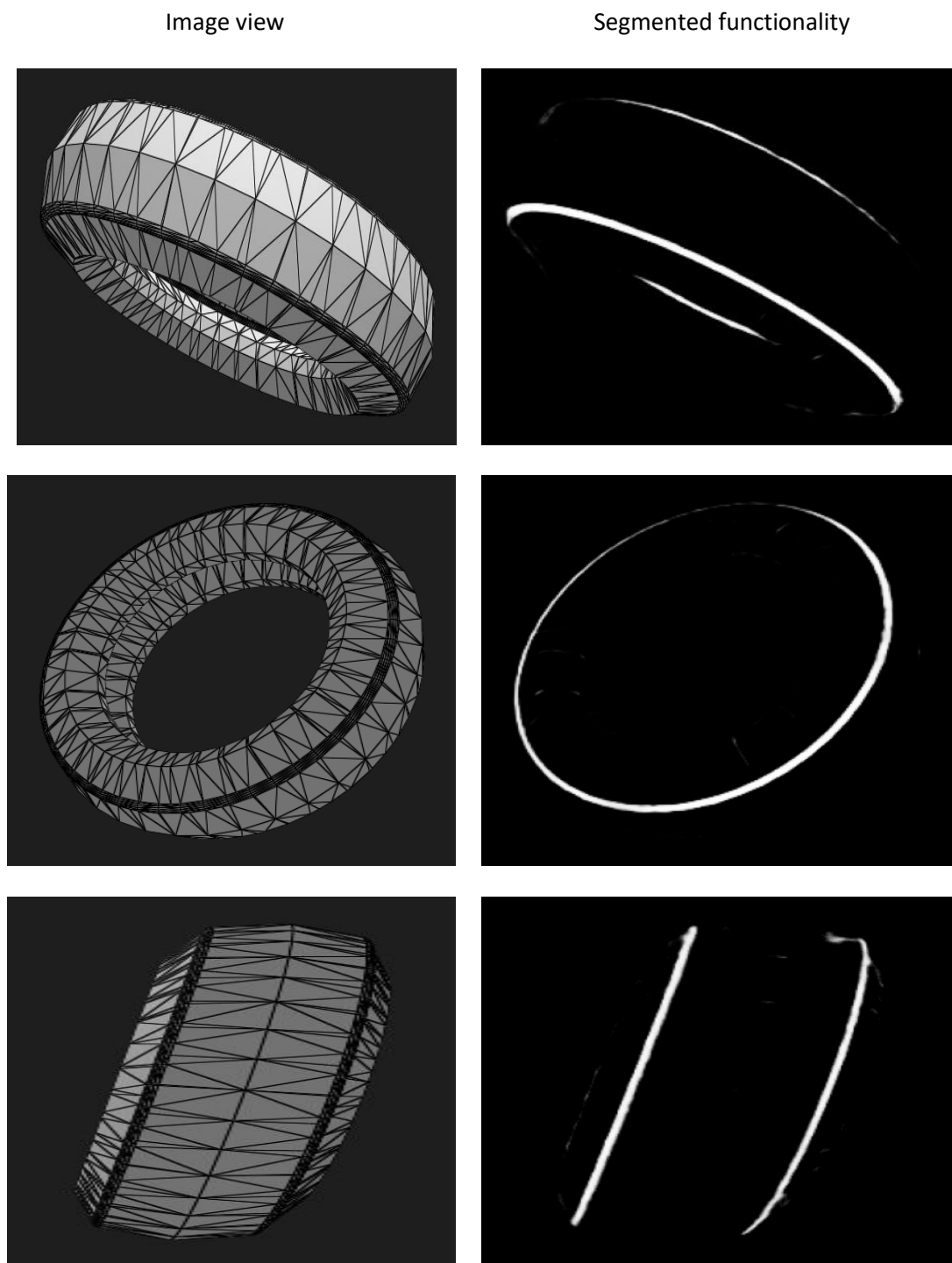


Figure 5-20. Segmented views of Ring model

5.4.4 Final results of simplification of continuous chains

After segmentation, the 2D images are back propagated to 3D domain, as illustrated in Figure 5-3 and section 5.2.3. We apply the simplification technique detailed in section 5.3 on the segmented meshes.

Some examples of simplification are illustrated in Figure 5-21. The first model, called Fitting, initially contains 2070 vertices. After the removal of continuous chains using the proposed approach, the simplified version now consists of only 566 vertices. This remarkable reduction in vertices amounts to a simplification of approximately 72.7%. Similarly, the second model, known as Hood, started with 828 vertices. Through the application of our simplification technique, the model has been reduced to only 157 vertices. This represents a remarkable simplification of approximately 81%. There is a significant reduction in complexity, which in turn increases the framerate of the application and ensures a fluid rendering. Let us note that that percentage of simplification depends on the continuous chains present in the particular CAD model. Complex models tend to have more continuous chains due to higher number of transitions from one feature to another.

These results demonstrate the effectiveness and efficiency of our framework in simplifying CAD models, specifically by targeting and removing continuous chains. The substantial reduction in vertex count leads to improvement in run-time performance, making the simplified models more suitable for AR-based training in industrial assembly lines.

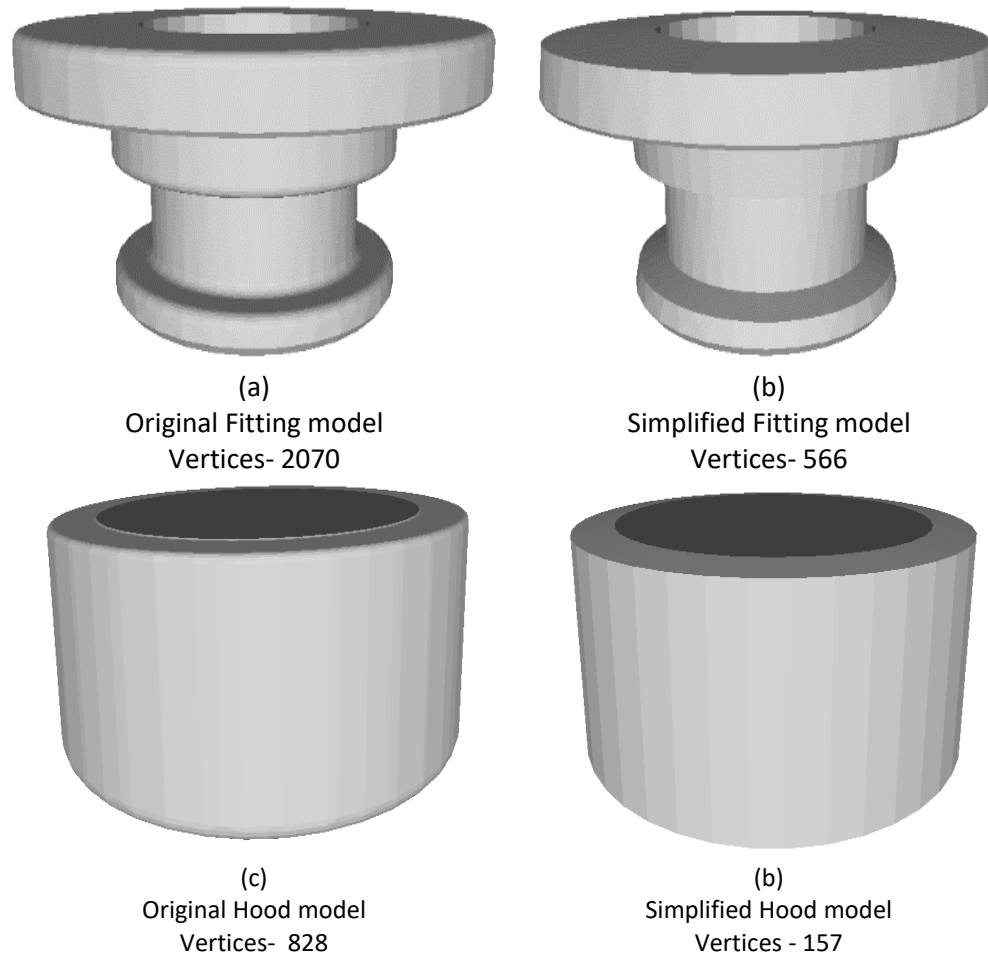
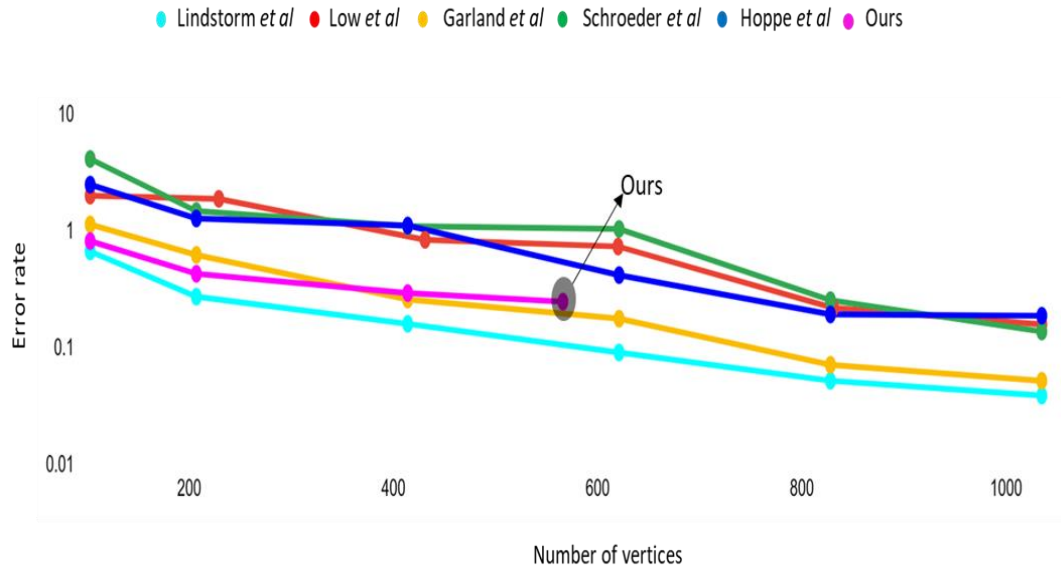
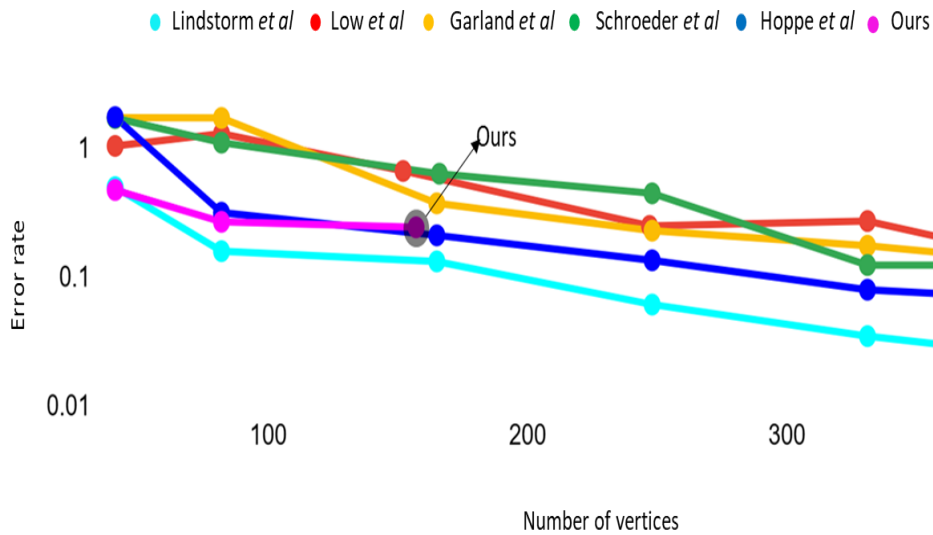


Figure 5-21. Final simplified models

To compare our methodology against the state-of-the-art techniques (Garland & Heckbert, 1997; Hoppe, 1996; Lindstrom & Turk, 1998; Low & Tan, 1997; W. J. Schroeder et al., 1992), we have performed the same objective evaluation considered in section 4.1.3. After removing continuous chains, we calculate the Hausdorff error between the original and simplified models. In our case, we have first eliminated the continuous chains (this corresponds to the last point on the curve) (Figure 5-22) and then we continued the simplification by applying the best state of the art technique (Lindstrom & Turk, 1998) determined in Chapter 4 section 4.1.3.



a) Fitting



b) Hood model

Figure 5-22. Comparison to state of the art

Figure 5-22 presents the objective evaluations of the proposed methodology against the state-of-the-art techniques. The error rate our methodology is higher when compared to state-of-the-art techniques. This is mainly because the objective evaluation metric available in the state of the art mainly focuses on the mesh-based simplification where the metric tries to find the correspondent vertex for each vertex in the simplified model. Since we have employed a feature aware mesh-based simplification and removed an entire band of vertices obviously the metric

cannot find some of the correspondent vertices of continuous chains in the simplified model. For this reason, we cannot solely rely on the objective evaluation.

Figure 5-23 presents the visual comparison of the test models simplified by our methodology against the best state of the art technique. We can clearly observe that our models preserve the structure of the whole model intact while we can notice the abnormalities in the edges of the models simplified by Lindstrom *et al.* (Lindstrom & Turk, 1998) for the same range of simplification. Overall, our method has better visual approximations, which is of outmost importance when using simplified CAD models in AR for industrial assembly lines when compared to the state of the art techniques.

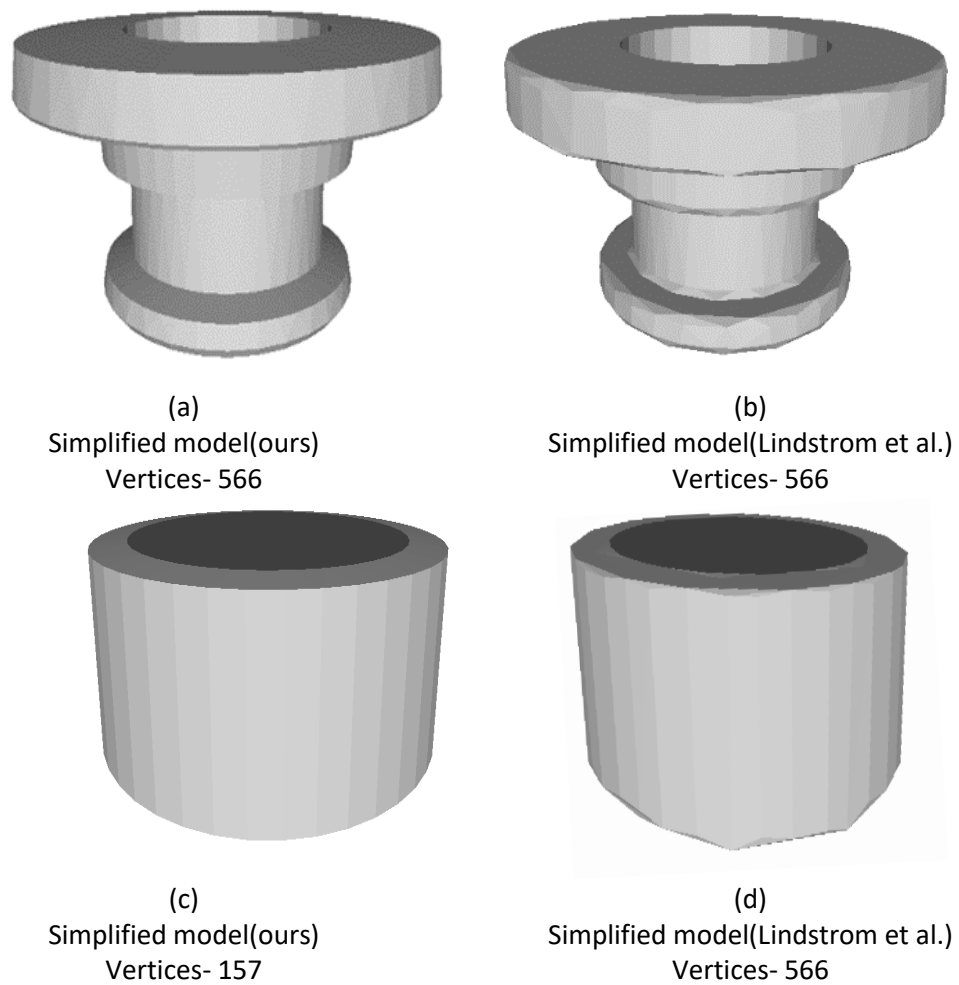


Figure 5-23. Visual comparison to state of the art

Chapter 6

CONCLUSION AND PERSPECTIVES

The main aim of the thesis is to simplify the 3D CAD models for Augmented reality for operator training in industrial assembly lines.

To pinpoint the challenges associated with integrating 3D CAD models into augmented reality (AR) for assembly training, we took a hands-on approach by physically assembling an entire boiler. Our practical approach of physically assembling a boiler revealed the challenges of integrating 3D CAD models into augmented reality (AR) for assembly training. While this hands-on experience helped us understand the limitations of simplification techniques, it also showcased the specific scenarios where 3D CAD models excel on assembly lines. We have learned that 2D images often obscure crucial details, emphasizing the unique advantages of 3D CAD models despite the higher costs. In summary, our exploration underscores the significance of investing in 3D CAD models for improved training and comprehension in assembly line scenarios, emphasizing their superiority over 2D images.

Our research focused on two critical areas to determine the optimal CAD model simplification approach. First, we conducted a thorough theoretical analysis of various simplification techniques, including B-Rep based, dimensionality-based, and mesh-based methods. After careful evaluation, we found that mesh-based techniques were the most suitable for assembly line applications. Following this, we delved into a comprehensive analysis of traditional mesh-based approaches and those based on deep learning. We provided insights into their strengths and limitations, with particular attention to the challenges of deep learning when dealing with data lacking continuity and model completeness.

Expanding upon the theoretical analysis, our research ventured into the practical assessment of mesh-based simplification techniques. We put into practice five distinct algorithms falling under the mesh-based simplification. These algorithms were deliberately chosen to offer a spectrum of approaches, varying in their simplification operators and how they handle geometric aspects. We conducted thorough numerical assessments to quantify the disparities between the original CAD models and their simplified versions. Ultimately, this evaluation led us to identify the most effective method among the tested algorithms. This best-performing approach emerged as the preferred choice for achieving optimal results in CAD model simplification within the context of our research objectives.

In conjunction with our objective evaluations, we have carried out a subjective assessment, which was instrumental in determining the appropriate simplification rates for each model, considering the intricacies and complexity of the specific model. To conclude our research and validate our findings, we designed and executed a real-world augmented reality (AR) scenario. In this scenario, we seamlessly integrated the simplified CAD models into the HoloLens2, a popular AR device, and deployed them within actual workstation setups in industrial assembly

lines. The results were compelling, as we observed significant improvements in the applications frame rate compared to when using the original, more complex CAD models.

In the training of assembly line operators, user experience (UX) is of utmost importance. To complement our numerical assessment of simplified CAD models in augmented reality (AR) training, we have conducted a dedicated user evaluation. The evaluation aimed to measure the degree of acceptance of simplified CAD models in the training process under real-world conditions. To ensure the authenticity of the assessments, we have designed assembly line instructions based on boiler assembly, replicating the actual working environment in a laboratory setting. Our careful selection of CAD models encompassed a wide range of complexities and structures, enabling us to assess the performance and user acceptance of simplified CAD models across diverse instructional scenarios. Participants engaged in unbiased evaluations, experiencing tasks from simple to complex, closely mirroring industrial assembly line operations for practical and relevant assessments.

Our comprehensive investigation thoroughly addressed both theoretical and practical facets of CAD model simplification, resulting in an informed and efficient solution to improve assembly line workflows with augmented reality. While the existing simplified models meet perceptual standards, a notable challenge arises when multiple complex models need to be assembled or configured at a workstation. In these scenarios, the framerate experiences a noticeable decline, which impacts the fluid rendering and overall user experience. This framerate reduction can be particularly problematic, especially in industrial settings where efficiency and precision are paramount.

Our focus was specifically on identifying and addressing redundant features within CAD models, which could be streamlined or eliminated without compromising the models visual integrity when viewed in AR environments. In the course of our research, we pinpointed a redundant feature so-called *continuous chains* that frequently appears within CAD models. We have developed a robust solution to simplify the continuous chains feature by firstly segmenting them in 2D domain.

For segmentation purposes, we have notably harnessed the UNet models capabilities and fine-tuned hyperparameters using Optuna to ensure precise and reliable results. Once successful 2D segmentation achieved, the results are mapped back onto the original 3D CAD model, bridging the gap between 2D analysis and the 3D models representation. To eliminate the identified continuous chains, we have implemented a graph traversal technique, effectively consolidating these chains into a single entity. The results obtained from the framework are notably impressive both visually and functionally. We have also compared the final results with the state of the techniques which showed our proposed methodology has superior visual quality.

Conclusion and perspectives

Future research in this domain holds promising avenues for expansion and enhancement. Building upon our successful simplification of continuous chains, there is an opportunity to extend this approach to address multiple features within CAD models. This multi-feature simplification can lead to further optimization in industrial assembly lines. Large-scale evaluations involving operators within these industrial settings can provide valuable insights into the real-world applicability and benefits of this extended approach.

The limited availability of CAD model databases corresponding to original industrial models presents a notable challenge. Future work could involve the creation and curation of comprehensive databases that align with the models used in industries. Such databases would be instrumental when applying deep learning techniques for simplification. Access to diverse and representative datasets can significantly enhance the accuracy and effectiveness of simplification methods.

While our work primarily focused on the 2D domain due to data constraints, advancements in data availability and computational capabilities make it worthwhile to explore simplification directly in the 3D domain. This transition can further enhance the applicability and robustness of simplification techniques, aligning with the evolving landscape of CAD modeling and augmented reality technologies. The future research directions hold the potential to advance the field of CAD model simplification and its applications in industrial assembly processes.

REFERENCES

References

- Agarwal, P. K., & Suri, S. (1998). Surface Approximation and Geometric Partitions. *SIAM Journal on Computing*, 27(4), 1016–1035. <https://doi.org/10.1137/S0097539794269801>
- Ahmad, H., Butt, A., & Muzaffar, A. (2023). Travel before you actually travel with augmented reality – role of augmented reality in future destination. *Current Issues in Tourism*, 26(17), 2845–2862. <https://doi.org/10.1080/13683500.2022.2101436>
- Ahmed, M., Seraj, R., & Islam, S. M. S. (2020). The k-means Algorithm: A Comprehensive Survey and Performance Evaluation. *Electronics*, 9(8), 1295. <https://doi.org/10.3390/electronics9081295>
- Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2623–2631. <https://doi.org/10.1145/3292500.3330701>
- Alves, J. B., Marques, B., Dias, P., & Santos, B. S. (2021). Using augmented reality for industrial quality assurance: a shop floor user study. *The International Journal of Advanced Manufacturing Technology*, 115(1–2), 105–116. <https://doi.org/10.1007/s00170-021-07049-8>
- Asmara, A., Nienhuis, U., & Hekkenberg, R. (2010). Approximate orthogonal simplification of 3D model. *IEEE Congress on Evolutionary Computation*, 1–4. <https://doi.org/10.1109/CEC.2010.5586162>
- Badrinarayanan, V., Kendall, A., & Cipolla, R. (2015). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *CoRR*, abs/1511.00561. <http://arxiv.org/abs/1511.00561>
- Begout, P., Kubicki, S., Bricard, E., & Duval, T. (2022). Augmented Reality Authoring of Digital Twins: Design, Implementation and Evaluation in an Industry 4.0 Context. *Frontiers in Virtual Reality*, 3. <https://doi.org/10.3389/frvir.2022.918685>
- BERNARDINI, F., BAJAJ, C. L., CHEN, J., & SCHIKORE, D. R. (1999). AUTOMATIC RECONSTRUCTION OF 3D CAD MODELS FROM DIGITAL SCANS. *International Journal of Computational Geometry & Applications*, 09(04n05), 327–369. <https://doi.org/10.1142/S0218195999000236>
- Botsch, M., & Kobbelt, L. (2004). A remeshing approach to multiresolution modeling. *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, 185–192. <https://doi.org/10.1145/1057432.1057457>
- Cañas, H., Mula, J., Díaz-Madroñero, M., & Campuzano-Bolarín, F. (2021). Implementing Industry 4.0 principles. *Computers & Industrial Engineering*, 158, 107379. <https://doi.org/10.1016/j.cie.2021.107379>

References

- Carmigniani, J., Furht, B., Anisetti, M., Ceravolo, P., Damiani, E., & Ivkovic, M. (2011). Augmented reality technologies, systems and applications. *Multimedia Tools and Applications*, 51(1), 341–377. <https://doi.org/10.1007/s11042-010-0660-6>
- CGAL. (2020). *Computational Geometry Algorithms Library* (5.1). CGAL Editorial Board. <https://www.cgal.org/>
- Chen, P., Liu, X., Cheng, W., & Huang, R. (2017). A review of using Augmented Reality in Education from 2011 to 2016 (pp. 13–18). https://doi.org/10.1007/978-981-10-2419-1_2
- Chimakurthi, V. N. S. S. (2019). Efficacy of Augmented Reality in Medical Education. *Malaysian Journal of Medical and Biological Research*, 6(2), 135–xxx. <https://doi.org/10.18034/mjmbr.v6i2.609>
- Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., & Ranzuglia, G. (2008). MeshLab: an Open-Source Mesh Processing Tool. In V. Scarano, R. De Chiara, & U. Erra (Eds.), *Eurographics Italian Chapter Conference*. The Eurographics Association. <https://doi.org/10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136>
- Cignoni, P., Montani, C., & Scopigno, R. (1998). A comparison of mesh simplification algorithms. *Computers & Graphics*, 22(1), 37–54. [https://doi.org/10.1016/S0097-8493\(97\)00082-4](https://doi.org/10.1016/S0097-8493(97)00082-4)
- Cignoni, P., Rocchini, C., & Scopigno, R. (1998). Metro: Measuring Error on Simplified Surfaces. *Computer Graphics Forum*, 17(2), 167–174. <https://doi.org/10.1111/1467-8659.00236>
- Cohen, J., Olano, M., & Manocha, D. (1998). Appearance-preserving simplification. *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '98*, 115–122. <https://doi.org/10.1145/280814.280832>
- de Souza Cardoso, L. F., Mariano, F. C. M. Q., & Zorzal, E. R. (2020). A survey of industrial augmented reality. *Computers & Industrial Engineering*, 139, 106159. <https://doi.org/10.1016/j.cie.2019.106159>
- de Vries, B., & Harink, J. M. J. (2007). Generation of a construction planning from a 3D CAD model. *Automation in Construction*, 16(1), 13–18. <https://doi.org/10.1016/j.autcon.2005.10.010>
- DeHaemer, M. J., & Zyda, M. J. (1991). Simplification of objects rendered by polygonal approximations. *Computers & Graphics*, 15(2), 175–184. [https://doi.org/10.1016/0097-8493\(91\)90071-O](https://doi.org/10.1016/0097-8493(91)90071-O)
- Doolani, S., Wessels, C., Kanal, V., Sevastopoulos, C., Jaiswal, A., Nambiappan, H., & Makedon, F. (2020). A Review of Extended Reality (XR) Technologies for Manufacturing Training. *Technologies*, 8(4), 77. <https://doi.org/10.3390/technologies8040077>

References

- Dünser, A., Grasset, R., & Billinghurst, M. (2008). A survey of evaluation techniques used in augmented reality studies. *ACM SIGGRAPH ASIA 2008 Courses on - SIGGRAPH Asia '08*, 1–27. <https://doi.org/10.1145/1508044.1508049>
- Eckert, M., Volmerg, J. S., & Friedrich, C. M. (2019). Augmented Reality in Medicine: Systematic and Bibliographic Review. *JMIR MHealth and UHealth*, 7(4), e10967. <https://doi.org/10.2196/10967>
- Elliott, J. H., Berglund, C. L., & Greg Jensen, C. (2007). An Automated Approach to Feature-Based Design for Reusable Parameter-Rich Surface Models. *Computer-Aided Design and Applications*, 4(1–4), 497–507. <https://doi.org/10.1080/16864360.2007.10738569>
- Feiner, S., Macintyre, B., & Seligmann, D. (1993). Knowledge-based augmented reality. *Communications of the ACM*, 36(7), 53–62. <https://doi.org/10.1145/159544.159587>
- Fitzgerald, K. A., Malhotra, M., Curtin, C. M., O' Brien, F. J., & O' Driscoll, C. M. (2015). Life in 3D is never flat: 3D models to optimise drug delivery. *Journal of Controlled Release*, 215, 39–54. <https://doi.org/10.1016/j.jconrel.2015.07.020>
- Franc, M., & Skala, V. (2002). *Fast Algorithm for Triangular Mesh Simplification Based on Vertex Decimation* (pp. 42–51). https://doi.org/10.1007/3-540-46080-2_5
- Frank, A. G., Dalenogare, L. S., & Ayala, N. F. (2019). Industry 4.0 technologies: Implementation patterns in manufacturing companies. *International Journal of Production Economics*, 210, 15–26. <https://doi.org/10.1016/j.ijpe.2019.01.004>
- Fuh, J. Y. H., & Li, W. D. (2005). Advances in collaborative CAD: the-state-of-the art. *Computer-Aided Design*, 37(5), 571–581. <https://doi.org/10.1016/j.cad.2004.08.005>
- Funk, M., Kosch, T., Greenwald, S. W., & Schmidt, A. (2015). A benchmark for interactive augmented reality instructions for assembly tasks. *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia*, 253–257. <https://doi.org/10.1145/2836041.2836067>
- Gabbard, J. L., & Swan, J. E. (2008). Usability Engineering for Augmented Reality: Employing User-Based Studies to Inform Design. *IEEE Transactions on Visualization and Computer Graphics*, 14(3), 513–525. <https://doi.org/10.1109/TVCG.2008.24>
- Garland, M., & Heckbert, P. S. (1997). Surface simplification using quadric error metrics. *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '97*, 209–216. <https://doi.org/10.1145/258734.258849>
- Gattullo, M., Evangelista, A., Uva, A. E., Fiorentino, M., & Gabbard, J. L. (2022). What, How, and Why are Visual Assets Used in Industrial Augmented Reality? A Systematic Review and

References

- Classification in Maintenance, Assembly, and Training (From 1997 to 2019). *IEEE Transactions on Visualization and Computer Graphics*, 28(2), 1443–1456. <https://doi.org/10.1109/TVCG.2020.3014614>
- Gattullo, M., Scurati, G. W., Evangelista, A., Ferrise, F., Fiorentino, M., & Uva, A. E. (2020). *Informing the Use of Visual Assets in Industrial Augmented Reality* (pp. 106–117). https://doi.org/10.1007/978-3-030-31154-4_10
- Gevorgian, S., Linner, L. J. P., & Kollberg, E. L. (1995). CAD models for shielded multilayered CPW. *IEEE Transactions on Microwave Theory and Techniques*, 43(4), 772–779. <https://doi.org/10.1109/22.375223>
- Gieng, T. S., Hamann, B., Joy, K. I., Schussman, G. L., & Trotts, I. J. (1997). Smooth Hierarchical Surface Triangulations. *Proceedings of the 8th Conference on Visualization '97*, 379–386.
- Gómez, G. O., Viganò, R., & Arbeláez, J. C. (2016). An augmented reality tool to validate the assembly sequence of a discrete product. *International Journal of Computer Aided Engineering and Technology*, 8(1/2), 164. <https://doi.org/10.1504/IJCAET.2016.073263>
- Gotsman, C., Gumhold, S., & Kobbelt, L. (2002). *Simplification and Compression of 3D Meshes* (pp. 319–361). https://doi.org/10.1007/978-3-662-04388-2_12
- Hahn, J., Ludwig, B., & Wolff, C. (2015). Augmented reality-based training of the PCB assembly process. *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia*, 395–399. <https://doi.org/10.1145/2836041.2841215>
- Halttunen, V., & Tuikka, T. (2000). Augmenting virtual prototyping with physical objects. *Proceedings of the Working Conference on Advanced Visual Interfaces*, 305–306. <https://doi.org/10.1145/345513.345363>
- Hamann, B. (1994). A data reduction scheme for triangulated surfaces. *Computer Aided Geometric Design*, 11(2), 197–214. [https://doi.org/10.1016/0167-8396\(94\)90032-9](https://doi.org/10.1016/0167-8396(94)90032-9)
- Hanocka, R., Hertz, A., Fish, N., Giryas, R., Fleishman, S., & Cohen-Or, D. (2019). MeshCNN. *ACM Transactions on Graphics*, 38(4), 1–12. <https://doi.org/10.1145/3306346.3322959>
- Hasselgren, J., Munkberg, J., Lehtinen, J., Aittala, M., & Laine, S. (2021). Appearance-Driven Automatic 3D Model Simplification. *Eurographics Symposium on Rendering*. <https://api.semanticscholar.org/CorpusID:233204317>
- He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2017). Mask R-CNN. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2980–2988. <https://doi.org/10.1109/ICCV.2017.322>
- Herrera, I. (1981). An Algebraic Theory of Boundary Value Problems. *Kinam*, 3.

References

- Hoppe, H. (1996). Progressive meshes. *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, 99–108. <https://doi.org/10.1145/237170.237216>
- Hořejší, P. (2015). Augmented Reality System for Virtual Training of Parts Assembly. *Procedia Engineering*, 100, 699–706. <https://doi.org/10.1016/j.proeng.2015.01.422>
- Hou, L., & Wang, X. (2011, June 29). *Experimental Framework for Evaluating Cognitive Workload of Using AR System in General Assembly Task*. <https://doi.org/10.22260/ISARC2011/0117>
- Hou, L., Wang, X., Bernold, L., & Love, P. E. D. (2013). Using Animated Augmented Reality to Cognitively Guide Assembly. *Journal of Computing in Civil Engineering*, 27(5), 439–451. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000184](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000184)
- Hua, Z., Huang, Z., & Li, J. (2015). Mesh Simplification Using Vertex Clustering Based on Principal Curvature. *International Journal of Multimedia and Ubiquitous Engineering*, 10(9), 99–110. <https://doi.org/10.14257/ijmue.2015.10.9.11>
- Jayanti, S., Kalyanaraman, Y., Iyer, N., & Ramani, K. (2006). Developing an engineering shape benchmark for CAD models. *Computer-Aided Design*, 38(9), 939–953. <https://doi.org/10.1016/j.cad.2006.06.007>
- Keller, B., Schmidt, R., Möhring, M., & Bayer, C. (2015, October). *Augmented Reality in the travel industry: A perspective how modern technology can fit consumer's needs in the service industry*.
- Krüger, J., Lien, T. K., & Verl, A. (2009). Cooperation of human and machines in assembly lines. *CIRP Annals*, 58(2), 628–646. <https://doi.org/10.1016/j.cirp.2009.09.009>
- Kwon, S., Mun, D., Kim, B. C., Han, S., & Suh, H.-W. (2019). B-rep model simplification using selective and iterative volume decomposition to obtain finer multi-resolution models. *Computer-Aided Design*, 112, 23–34. <https://doi.org/10.1016/j.cad.2019.03.003>
- Lapierre, S. D., & Ruiz, A. B. (2004). Balancing assembly lines: an industrial case study. *Journal of the Operational Research Society*, 55(6), 589–597. <https://doi.org/10.1057/palgrave.jors.2601708>
- Lasi, H., Fettke, P., Kemper, H.-G., Feld, T., & Hoffmann, M. (2014). Industry 4.0. *Business & Information Systems Engineering*, 6(4), 239–242. <https://doi.org/10.1007/s12599-014-0334-4>
- Lavoué, G., Dupont, F., & Baskurt, A. (2005). A new CAD mesh segmentation method, based on curvature tensor analysis. *Computer-Aided Design*, 37(10), 975–987. <https://doi.org/10.1016/j.cad.2004.09.001>

References

- LAVRIC, T., BRICARD, E., PREDA, M., & ZAHARIA, T. (2020). An AR Work Instructions Authoring Tool for Human-Operated Industrial Assembly Lines. *2020 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, 174–183. <https://doi.org/10.1109/AIVR50618.2020.00037>
- Lavric, T., Bricard, E., Preda, M., & Zaharia, T. (2021a). *An Industry-Adapted AR Training Method for Manual Assembly Operations* (pp. 282–304). https://doi.org/10.1007/978-3-030-90963-5_22
- Lavric, T., Bricard, E., Preda, M., & Zaharia, T. (2021b). Exploring Low-Cost Visual Assets for Conveying Assembly Instructions in AR. *2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, 1–6. <https://doi.org/10.1109/INISTA52262.2021.9548570>
- Lee, S. H. (2005). Feature-based multiresolution modeling of solids. *ACM Transactions on Graphics*, 24(4), 1417–1441. <https://doi.org/10.1145/1095878.1095887>
- Lee, S. H., & Lee, K. (2012). Simultaneous and incremental feature-based multiresolution modeling with feature operations in part design. *Computer-Aided Design*, 44(5), 457–483. <https://doi.org/10.1016/j.cad.2011.12.005>
- Li, Q., Sun, X., Yang, B., & Jiang, S. (2013). Geometric structure simplification of 3D building models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 84, 100–113. <https://doi.org/10.1016/j.isprsjprs.2013.07.006>
- Lindstrom, P., & Turk, G. (1998). Fast and memory efficient polygonal simplification. *Proceedings Visualization '98 (Cat. No.98CB36276)*, 279-286,. <https://doi.org/10.1109/VISUAL.1998.745314>
- Liu, S., Li, W., Ogunbona, P., & Chow, Y.-W. (2015). Creating Simplified 3D Models with High Quality Textures. *2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 1–8. <https://doi.org/10.1109/DICTA.2015.7371249>
- Long, J., Shelhamer, E., & Darrell, T. (2014). Fully Convolutional Networks for Semantic Segmentation. *CoRR*, abs/1411.4038. <http://arxiv.org/abs/1411.4038>
- Low, K.-L., & Tan, T.-S. (1997). Model simplification using vertex-clustering. *Proceedings of the 1997 Symposium on Interactive 3D Graphics - SI3D '97*, 75-ff. <https://doi.org/10.1145/253284.253310>
- MacIntyre, B., & Hannigan, B. (2003). *Using Augmented Reality for Entertainment* (pp. 471–477). https://doi.org/10.1007/978-0-387-35660-0_57

References

- Manca, D., Totaro, R., Nazir, S., Brambilla, S., & Colombo, S. (2012). *Virtual and Augmented Reality as Viable Tools to Train Industrial Operators* (pp. 825–829). <https://doi.org/10.1016/B978-0-444-59507-2.50157-8>
- Manuri, F., & Sanna, A. (2016). A Survey on Applications of Augmented Reality. *Advances in Computer Science : An International Journal*, 5, 18–27. <https://api.semanticscholar.org/CorpusID:12840083>
- Marino, E., Barbieri, L., Colacino, B., Fleri, A. K., & Bruno, F. (2021). An Augmented Reality inspection tool to support workers in Industry 4.0 environments. *Computers in Industry*, 127, 103412. <https://doi.org/10.1016/j.compind.2021.103412>
- Mekni, M., & Lemieux, A. (2014). *Augmented Reality : Applications , Challenges and Future Trends*. <https://api.semanticscholar.org/CorpusID:15436826>
- Mesh Processing Library. (2023). *Mesh Processing Library source code*. <https://github.com/Hhoppe/Mesh-Processing-Library>. <https://github.com/hhoppe/mesh-processing-library>
- Milgram, P., Takemura, H., Utsumi, A., & Kishino, F. (1995). *Augmented reality: a class of displays on the reality-virtuality continuum* (H. Das, Ed.; pp. 282–292). <https://doi.org/10.1117/12.197321>
- Mun, D., & Kim, B. C. (2017). Extended progressive simplification of feature-based CAD models. *The International Journal of Advanced Manufacturing Technology*, 93(1–4), 915–932. <https://doi.org/10.1007/s00170-017-0491-y>
- Neuhöfer, J. A., Odenthal, B., Mayer, M. P., Jochems, N., & Schlick, C. (2010). Analysis and Modeling of Aimed Movements in Augmented and Virtual Reality Training Systems for Dynamic Human-Robot Cooperation. *11. IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Human-Machine Systems*, 6 S. <https://publications.rwth-aachen.de/record/119003>
- Ng, K.-W., & Low, Z.-W. (2014). Simplification of 3D Triangular Mesh for Level of Detail Computation. *2014 11th International Conference on Computer Graphics, Imaging and Visualization*, 11–16. <https://doi.org/10.1109/CGiV.2014.9>
- Odenthal, B., Mayer, M. Ph., Kabuss, W., & Schlick, C. M. (2012). Design and evaluation of an Augmented Vision System for human-robot cooperation in cognitively automated assembly cells. *International Multi-Conference on Systems, Signals & Devices*, 1–6. <https://doi.org/10.1109/SSD.2012.6197931>
- Ong, S. K., & Wang, Z. B. (2011). Augmented assembly technologies based on 3D bare-hand interaction. *CIRP Annals*, 60(1), 1–4. <https://doi.org/10.1016/j.cirp.2011.03.001>

References

- Pan, Z., Zhou, K., & Shi, J. (2001). A new mesh simplification algorithm based on triangle collapses. *Journal of Computer Science and Technology*, 16(1), 57–63. <https://doi.org/10.1007/BF02948853>
- Philbeck, T. D., & Davis, N. R. (2018). The Fourth Industrial Revolution: Shaping a New Era. *Journal of International Affairs*, 72, 17. <https://api.semanticscholar.org/CorpusID:198627085>
- Pollefeys, M., & Gool, L. Van. (2002). From images to 3D models. *Communications of the ACM*, 45(7), 50–55. <https://doi.org/10.1145/514236.514263>
- Potamias, R. A., Bouritsas, G., & Zafeiriou, S. (2021). Revisiting Point Cloud Simplification: A Learnable Feature Preserving Approach. *CoRR*, abs/2109.14982. <https://arxiv.org/abs/2109.14982>
- Potamias, R. A., Ploumpis, S., & Zafeiriou, S. (2022). Neural Mesh Simplification. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 18562–18571. <https://doi.org/10.1109/CVPR52688.2022.01803>
- Rakotosaona, M.-J., Aigerman, N., Mitra, N. J., Ovsjanikov, M., & Guerrero, P. (2021). Differentiable Surface Triangulation. *CoRR*, abs/2109.10695. <https://arxiv.org/abs/2109.10695>
- Ramasinghe, S., Khan, S., Barnes, N., & Gould, S. (2020). Spectral-GANs for High-Resolution 3D Point-cloud Generation. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 8169–8176. <https://doi.org/10.1109/IROS45743.2020.9341265>
- Reinhart, G., & Patron, C. (2003). Integrating Augmented Reality in the Assembly Domain - Fundamentals, Benefits and Applications. *CIRP Annals*, 52(1), 5–8. [https://doi.org/10.1016/S0007-8506\(07\)60517-4](https://doi.org/10.1016/S0007-8506(07)60517-4)
- Ren, S., He, K., Girshick, R. B., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *CoRR*, abs/1506.01497. <http://arxiv.org/abs/1506.01497>
- Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation* (pp. 234–241). https://doi.org/10.1007/978-3-319-24574-4_28
- Rossignac, J., & Borrel, P. (1993). Multi-resolution 3D approximations for rendering complex scenes. In *Modeling in Computer Graphics* (pp. 455–465). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-78114-8_29
- Sääski, J., Salonen, T., Hakkarainen, M., Siltanen, S., Woodward, C., & Lempiäinen, J. (n.d.). Integration of Design and Assembly Using Augmented Reality. In *Micro-Assembly*

References

- Technologies and Applications* (pp. 395–404). Springer US. https://doi.org/10.1007/978-0-387-77405-3_39
- Schönhof, R., Elstner, J., Manea, R., Tauber, S., Awad, R., & Huber, M. F. (2022). Simplified Learning of CAD Features Leveraging a Deep Residual Autoencoder. *Procedia CIRP*, 109, 84–88. <https://doi.org/10.1016/j.procir.2022.05.218>
- Schroeder, W. J., Zarge, J. A., & Lorensen, W. E. (1992). Decimation of triangle meshes. *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*, 65–70. <https://doi.org/10.1145/133994.134010>
- Schroeder, W., Martin, K., & Lorensen, B. (2006). *The Visualization Toolkit (4th ed.)*. Kitware.
- Sharma, L. (2019). *Recent Advancements of Augmented Reality in Real Time Applications*. 8, 537–542. <https://doi.org/10.35940/ijrte.B10100.07825719>
- Sharp, N., & Ovsjanikov, M. (2020). PointTriNet: Learned Triangulation of 3D Point Sets. *CoRR*, *abs/2005.02138*. <https://arxiv.org/abs/2005.02138>
- Shuming Gao, Wei Zhao, Fanqin Yang, & Xiang Chen. (2008). Feature suppression based CAD mesh model simplification. *2008 IEEE International Conference on Shape Modeling and Applications*, 271–272. <https://doi.org/10.1109/SMI.2008.4548002>
- Sun, R., Gao, S., & Zhao, W. (2010). An approach to B-rep model simplification based on region suppression. *Computers & Graphics*, 34(5), 556–564. <https://doi.org/10.1016/j.cag.2010.06.007>
- Swan, J. E., & Gabbard, J. L. (2005). *Survey of User-Based Experimentation in Augmented Reality*. <https://api.semanticscholar.org/CorpusID:13985675>
- Syberfeldt, A., Danielsson, O., Holm, M., & Wang, L. (2015). Visual Assembling Guidance Using Augmented Reality. *Procedia Manufacturing*, 1, 98–109. <https://doi.org/10.1016/j.promfg.2015.09.068>
- Thakur, A., Banerjee, A. G., & Gupta, S. K. (2009). A survey of CAD model simplification techniques for physics-based simulation applications. *Computer-Aided Design*, 41(2), 65–80. <https://doi.org/10.1016/j.cad.2008.11.009>
- Ungureanu, D., Bogo, F., Galliani, S., Sama, P., Duan, X., Meekhof, C., Stühmer, J., Cashman, T. J., Tekin, B., Schönberger, J. L., Olszta, P., & Pollefeys, M. (2020). HoloLens 2 Research Mode as a Tool for Computer Vision Research. *CoRR*, *abs/2008.11239*. <https://arxiv.org/abs/2008.11239>
- Vaidya, S., Ambad, P., & Bhosle, S. (2018). Industry 4.0 – A Glimpse. *Procedia Manufacturing*, 20, 233–238. <https://doi.org/10.1016/j.promfg.2018.02.034>

References

- Valentini, P. P. (2009). Interactive virtual assembling in augmented reality. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 3(2), 109–119. <https://doi.org/10.1007/s12008-009-0064-x>
- VCG Library. (2023). VCG Library source code. In <https://github.com/cnr-isti-vclab/vcglib>. <https://github.com/cnr-isti-vclab/vcglib>
- Vincenzi, D. A., Valimont, B., Macchiarella, N., Opalenik, C., Gangadharan, S. N., & Majoros, A. E. (2003). The Effectiveness of Cognitive Elaboration Using Augmented Reality as a Training and Learning Paradigm. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 47(19), 2054–2058. <https://doi.org/10.1177/154193120304701909>
- Werrlich, S., Eichstetter, E., Nitsche, K., & Notni, G. (2017). An Overview of Evaluations Using Augmented Reality for Assembly Training Tasks. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 11, 1080–1086. <https://api.semanticscholar.org/CorpusID:721164>
- Wiedenmaier, S., Oehme, O., Schmidt, L., & Luczak, H. (2003). Augmented Reality (AR) for Assembly Processes Design and Experimental Evaluation. *International Journal of Human-Computer Interaction*, 16(3), 497–514. https://doi.org/10.1207/S15327590IJHC1603_7
- Yoon, Y., & Kim, B. C. (2016). CAD model simplification using feature simplifications. *Journal of Advanced Mechanical Design Systems and Manufacturing*, 10, 99. <https://api.semanticscholar.org/CorpusID:54912455>
- Yuan, M. L., Ong, S. K., & Nee, A. Y. C. (2008). Augmented reality for assembly guidance using a virtual interactive tool. *International Journal of Production Research*, 46(7), 1745–1767. <https://doi.org/10.1080/00207540600972935>
- Zeng, H., Peng, S., & Li, D. (2020). Deeplabv3+ semantic segmentation model based on feature cross attention mechanism. *Journal of Physics: Conference Series*, 1678(1), 012106. <https://doi.org/10.1088/1742-6596/1678/1/012106>
- Zhang, J., Ong, S. K., & Nee, A. Y. C. (2011). RFID-assisted assembly guidance system in an augmented reality environment. *International Journal of Production Research*, 49(13), 3919–3938. <https://doi.org/10.1080/00207543.2010.492802>
- Zhu, H., & Menq, C. H. (2002). B-Rep model simplification by automatic fillet/round suppressing for efficient automatic feature recognition. *Computer-Aided Design*, 34(2), 109–123. [https://doi.org/10.1016/S0010-4485\(01\)00056-2](https://doi.org/10.1016/S0010-4485(01)00056-2)

References

APPENDIX

Appendix 1: Questionnaire for user evaluations

1. Personal information :

- Name of the participant :
- Gender : Male Female
- Age :
- Profession/Education :

2. Prevoius expertise :

- Have you ever tried any kind of AR/VR/XR device before?
 Yes No
- How well do you know about Augmented reality?
 No idea Basic Medium Expert
- How well do you know about assembly operations inside industries?
 No idea Basic Medium Expert
- Any additional comments?

.....
.....

3. About the assembly

- How complex was the task?
 Low Medium High
- How well did you understand the task on a scale of 1 to 10?



- Rate the 3D model on a scale of 1 to 10?



4. General feedback

- Are there any areas of assembly/3D models that were difficult to understand?

.....
.....

- Do you have any suggestions or feedback?

.....
.....

- How was the overall experience on a scale of 1 to 10?



Appendix 2: Project IRONMEN

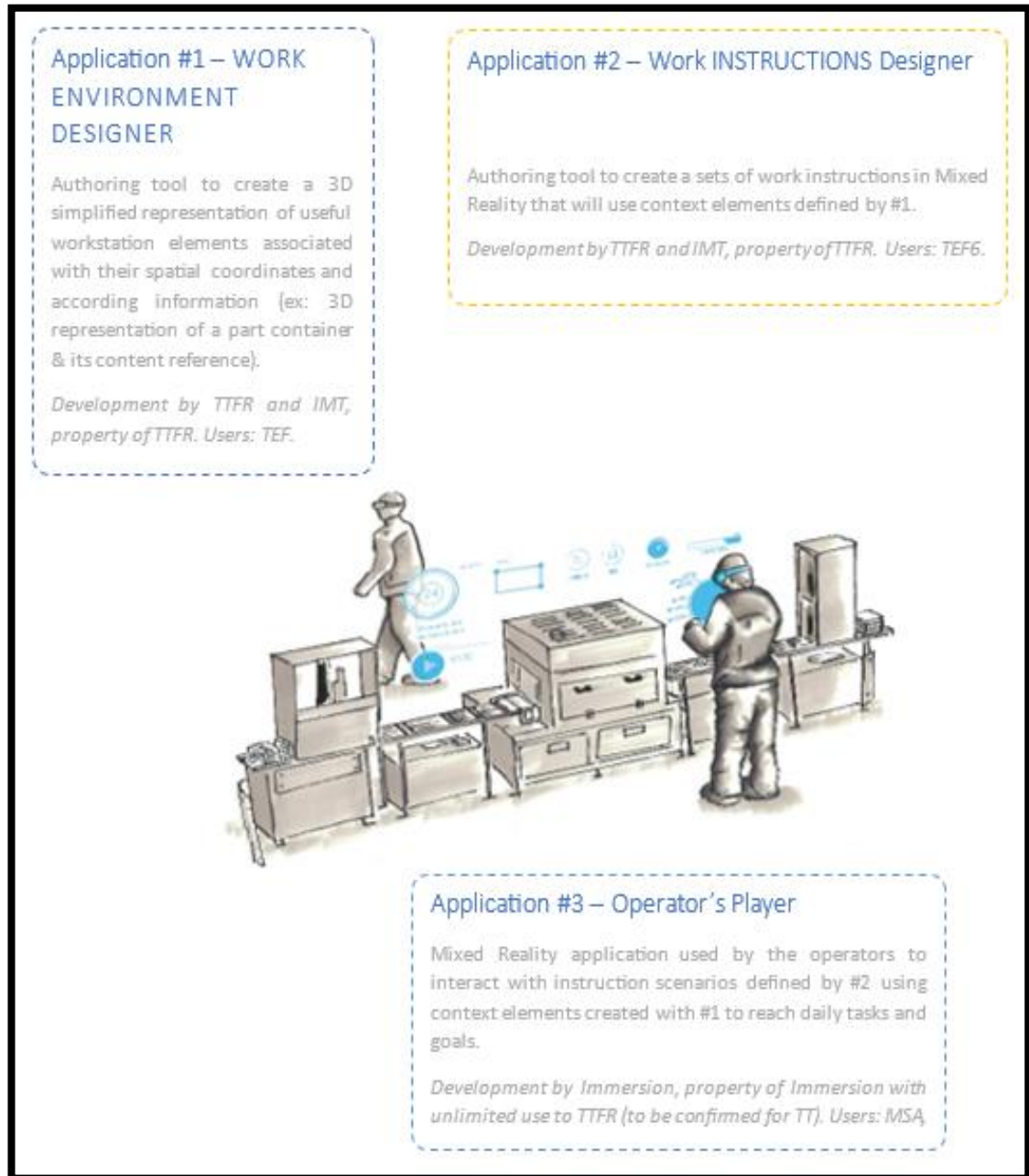
Industry 4.0 and Augmented Reality are complementary technologies that are driving significant improvements in industrial processes. They enhance efficiency, training, maintenance, and decision-making, ultimately leading to more agile and competitive industries. The integration of these technologies is likely to continue shaping the future of manufacturing and other industrial sectors.



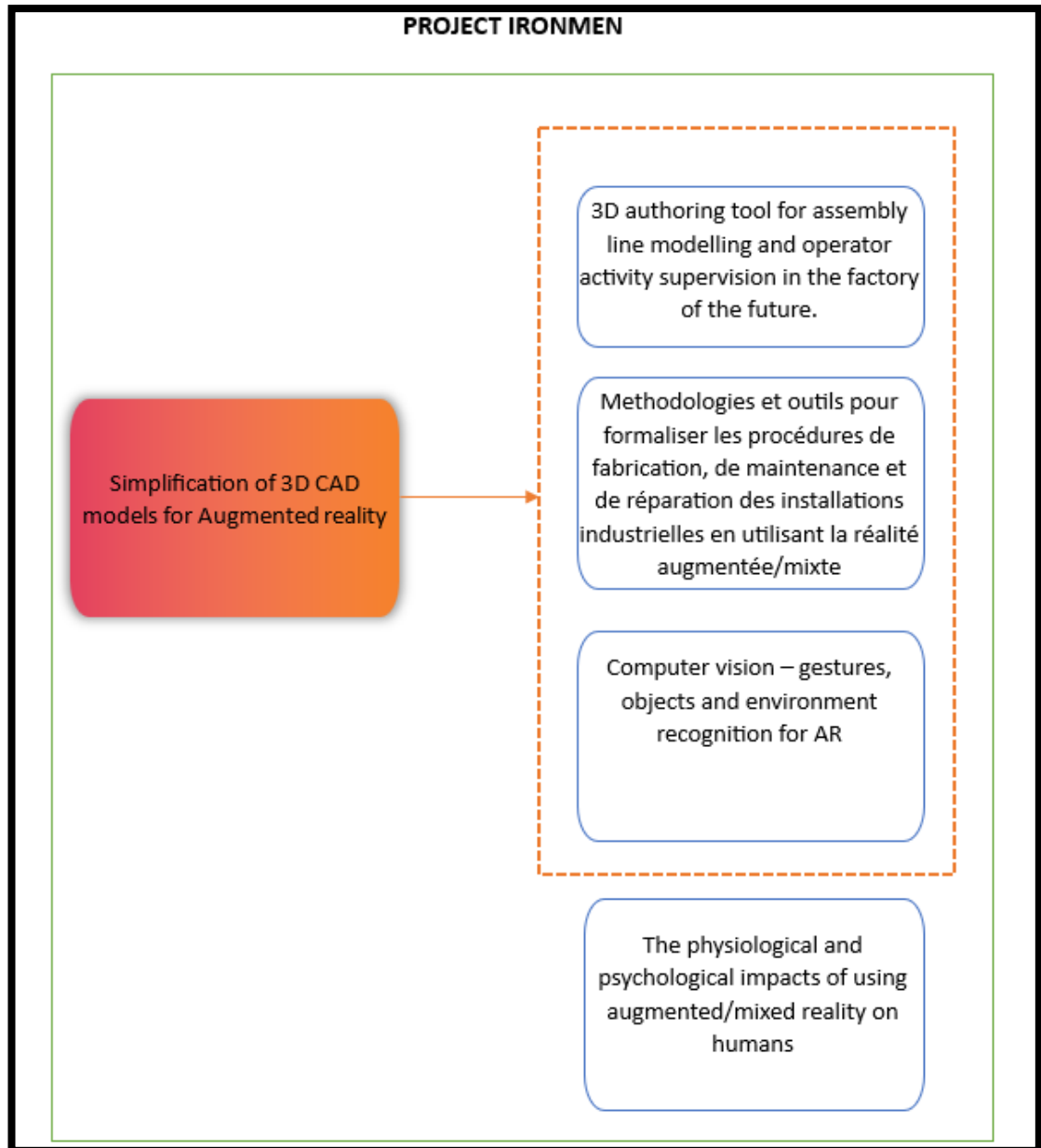
Elm.Leblanc, in collaboration with its partners Immersion Technologies, Institut Mines Télécom, and Adecam Industries, presents the IRON-MEN project, which has been chosen by BPI France as part of the Grand Défis du Numérique Projects, Augmented Reality category. This consortium is committed to a three-year endeavor to create an Augmented Reality solution tailored for production operators. The primary objective of IRON-MEN is to enhance the capabilities of individuals at the core of industrial processes. It will achieve this by fostering versatility, empowering operators to acquire new skills, and elevating the standards of the industry professions involved.



The solutions ambition is to offer an industrial and professional AR response tailored to specific needs to effectively assist users in their manual operations. It follows an original approach that combines digital transformation tools with respect for the individual within the production unit. It should be quickly adaptable to various business challenges that validate similar needs. This solution will accompany its user by guiding them in their daily tasks, thereby enhancing their versatility and meeting an increasingly personalized customer demand.



In this collaborative endeavor, each research project likely contributes its unique expertise and objectives towards a common goal. Our research focuses on simplifying 3D CAD models for augmented reality (AR) applications. This work is part of a collaborative effort that involves four other research projects within the same overarching initiative.



Titre : Simplification des modèles CAO 3D avec l'apprentissage profond pour la réalité augmentée

Mots clés : 3D, CAO Modèle, Simplification, l'apprentissage profond, Réalité augmentée

Résumé : Dans l'Industrie 4.0, l'utilisation d'appareils de Réalité Augmentée (RA) tels que HoloLens a acquis une acceptation significative pour la formation des opérateurs de ligne d'assemblage dans diverses industries. La simplification des modèles CAO 3D pour la formation en RA est essentielle pour une meilleure performance des applications. Notre recherche se concentre sur le développement de méthodes et de techniques visant à rationaliser des modèles CAO 3D complexes, les rendant adaptés aux applications de RA.

Notre recherche met en avant le rôle des modèles 3D en RA, améliorant l'expérience virtuelle en superposant des modèles CAO sur le monde réel. Nous détaillons les applications de la RA dans la formation des opérateurs et comment l'intégration de modèles CAO 3D améliore la compréhension des instructions et des procédures.

Nous avons réalisé une revue approfondie de la littérature sur la simplification des modèles CAO pour leur intégration dans des scénarios de réalité augmentée (RA). Nos conclusions indiquent que les techniques de simplification basées sur le maillage excellent dans la préservation des éléments essentiels des modèles CAO, offrant un contrôle précis sur les niveaux de détail.

De plus, nous avons effectué quatre types distincts d'évaluations dans notre recherche. Ces évaluations comprenaient des évaluations objectives utilisant des techniques basées sur le maillage issu de la littérature existante, des avis d'experts impliquant un examen approfondi de chaque modèle simplifié pour déterminer le niveau de simplification en fonction des plages de sommets, des tests en conditions réelles assistés par HoloLens2, qui ont révélé des améliorations du taux de rafraîchissement lors de l'utilisation de modèles CAO au lieu de leurs versions originales.

Pour conclure nos évaluations, nous avons également réalisé des évaluations par les utilisateurs, en donnant la priorité à l'expérience utilisateur dans notre étude. Ces évaluations ont confirmé que les modèles simplifiés sont hautement capables de remplacer les versions originales. Cependant, il a été observé qu'une simplification supplémentaire est nécessaire, en particulier pour les modèles CAO complexes.

La méthodologie principale propose une approche innovante axée sur la segmentation du maillage et la simplification adaptative grâce à l'utilisation de méthodes d'apprentissage profond. Pour réduire la complexité associée à la segmentation et à la simplification 3D, nous avons projeté les données dans le domaine 2D pour effectuer la segmentation et avons ensuite cartographié les résultats dans le domaine 3D. Nous avons illustré ce cadre à l'aide d'une fonction spécifique appelée "chaînes continues" pour expliquer le processus de simplification. Par la suite, nous avons réalisé une analyse comparative par rapport à des techniques de pointe établies, démontrant la performance supérieure de notre méthodologie. Dans nos futures recherches, nous visons à élargir la portée de notre cadre pour englober plusieurs caractéristiques et les régions fonctionnelles à l'intérieur des modèles CAO.

Title: Simplification of 3D CAD models with deep learning for augmented reality

Key words: 3D, CAD Model, Simplification, Deep Learning, Augmented reality

Resume: In Industry 4.0 the use of Augmented Reality (AR) devices like HoloLens has gained significant acceptance for training assembly line operators in various industries. Simplifying 3D CAD models for AR training is crucial for better application performance. Our research focuses on developing methods and techniques to streamline complex 3D CAD models, making them suitable for AR applications.

Our research highlights the role of 3D models in AR, enhancing the virtual experience by overlaying CAD models onto the real world. We detail AR applications in operator training and how integrating 3D CAD models improves understanding of instructions and procedures.

We performed an extensive literature review in CAD model simplification for integration into augmented reality (AR) scenarios. Our findings indicated that mesh-based simplification techniques excel in preserving CAD model essentials, providing precise control over the level of details.

Furthermore, we conducted four distinct types of assessments in our research. These assessments included objective evaluations utilizing mesh-based techniques from existing literature, expert review involving a comprehensive examination of each simplified model to determine the level of simplification based on vertex ranges, real-world testing aided by HoloLens2, which revealed improvements in framerate when using CAD models instead of their original versions. To conclude our evaluations, we also conducted user assessments, prioritizing user experience in our study. These assessments confirmed that the simplified models are highly capable of replacing the original counterparts.

To conclude our evaluations, we also conducted user assessments, prioritizing user experience in our study. These assessments confirmed that the simplified models are highly capable of replacing the original counterparts. However, it was observed that further simplification is necessary, especially for intricate CAD models.

The main methodology proposes an innovative approach, focusing on mesh segmentation and adaptive simplification using deep learning methods. To reduce the complexity associated with 3D segmentation and simplification we projected the data into the 2D domain to perform segmentation and mapped the results back to the 3D domain. We illustrated this framework using a specific feature called "continuous chains" to explain the simplification process. Subsequently, we conducted a comparative analysis against established state-of-the-art techniques, demonstrating the superior performance of our methodology. In our future research, we aim to broaden our framework's scope to encompass multiple features and functional regions within CAD models.