



**HAL**  
open science

## Simplification of continuous chains in 3D CAD models for industrial AR applications

Abhaya Dhathri Arige, Marius Preda, Marius Preda, Titus Zaharia

► **To cite this version:**

Abhaya Dhathri Arige, Marius Preda, Marius Preda, Titus Zaharia. Simplification of continuous chains in 3D CAD models for industrial AR applications. 2023 IEEE 19th International Conference on Intelligent Computer Communication and Processing (ICCP), Oct 2023, Cluj-Napoca, Romania. pp.171-176, 10.1109/ICCP60212.2023.10398677 . hal-04458312

**HAL Id: hal-04458312**

**<https://hal.science/hal-04458312>**

Submitted on 14 Feb 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 CONTINUOUS CHAINS IN 3D CAD MODELS FOR INDUSTRIAL AR APPLICATIONS

AbhayaDhathri Arige

Marius Preda

Titus Zaharia

**Abstract**— 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 paper introduces a novel solution to simplify CAD models by identifying continuous chains within the CAD model, with the help of deep learning techniques. The proposed method makes it possible to suppress such features that do not impede visualization when removed.

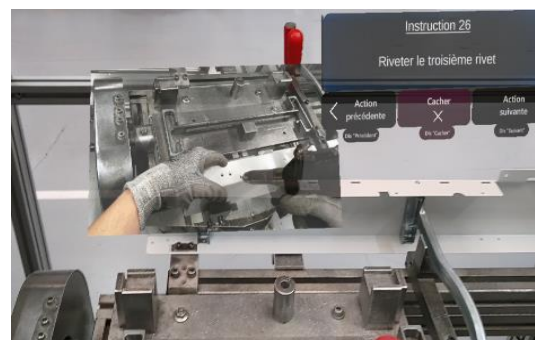
**Keywords**— Augmented reality (AR), Simplified CAD Models, Continuous chains, Deep learning

## I. INTRODUCTION

Industry 4.0 has paved way for the smarter machines improving the efficiency and effectiveness in many areas like production, maintenance and distribution in industries. As a part of industry 4.0, Augmented Reality (AR) techniques now aid assembly line operators in the training process. The research in the field of AR-based task display has started in the early 90s [1]. Horejsi et al. [2] compared classical written instructions against AR solutions. The results reported show that trainees have learnt 43% faster using AR when compared to paper-based instructions. In addition, the error rate is significantly reduced. AR uses existing real-world environments and puts virtual information on top of it to enhance the experience. Various industrial companies have started to implement such powerful tools on their shop floors. AR systems are now considered as promising training platforms for complex and highly demanding tasks. Within this context, an important category of AR applications is related to the processes of machine assembly and maintenance [3][4][5].

The media used in AR technology are mainly divided into two categories of instructions. The first one contains text, images, audio and video, while the second leverages 3D model-based instructions. The limitation of purely 2D audio-visual instructions is related to the occlusions that may appear in the scene. An example is illustrated in Fig. 1. Here, the white color panel is being assembled at the workstation. The instructional image showing how the assembly should be done is occluding the part where the object needs to be assembled (Fig. 1a). On the counterpart, instructions are easier to understand and implement if they are expressed with

the help of 3D models (Fig. 1b) superimposed on the real equipment rather than with text and pictures. This helps operators to understand in a step-by-step manner what tasks need to be done and how to perform them. Such 3D models typically provide accurate information about the object, maximizing the potential for a reliable AR experience.



a) Training using image based instruction



b) Training using 3D model based instruction

Fig. 1. AR instructions

However, preparing 3D CAD models for efficient use in AR industrial applications is a very important step. CAD models are created in various forms, adapted to various industries and applications. The main drawback is related to the huge storage and bandwidth capabilities required. In addition to the inherent complexity of the 3D data representation, they also include various metadata, specifying product structure, constraints, and moving assemblies and parts. Thus, their successful integration into AR devices requires a preprocessing phase aiming at reducing the related complexity of representation. Most of the time, CAD models are tessellated under the form of 3D meshes, which are widely acceptable in AR. Such meshes contain finely

detailed, arbitrary topology surfaces with potentially millions of vertices. Even though visually they have higher degree of closeness to the real-world objects, the computational cost for processing and rendering such meshes is very high.

The quality and degree of realism of an AR experience is determined as a sum of the complexity of the animations, the number of polygons, and the materials and textures used. A possible solution to ensure a reliable experience without saturating the fluid rendering is to use simplified versions of the original models, which improves the rendering performance of AR devices like HoloLens. In our previous work [6] we have discussed and analyzed the importance of using simplified models instead of original ones. The experimental evaluations performed demonstrated the pertinence and feasibility of using simplified versions of 3D models. Thus, the related frame rate significantly increased without loss in the visual quality.

In this paper, we focus on eliminating redundant features, specifically referred to as *continuous chains* from the CAD models. By removing such continuous chains, we aim to optimize the rendering process, resulting in improved framerates and thus more efficient AR applications.

The rest of the paper is organized as follows. Section II analyzes the state of the art on both 3D CAD model simplification and continuous chain functionality. Section III presents the novel simplification methodology proposed. Finally, Section IV concludes the paper and opens some perspectives of future work.

## II. RELATED WORK

CAD model simplification techniques play a crucial role in optimizing the efficiency and performance of various applications, including AR-based training for industrial assembly lines. In the literature there are three different types of CAD model simplification techniques [7], which are feature-based, dimensionality reduction-based and mesh-based simplification. Among the existing methods, feature-based simplification [8][9] involves systematically removing features until reaching the desired level of detail. However, this approach faces limitations when used in assembly processes involving CAD models at different workstations. Storing information about which part of the CAD model is used at each workstation and applying simplification accordingly becomes a challenging task. Dimensionality reduction [10] is another simplification technique that eliminates one or more dimensions from a boundary value problem. Although it can be effective in certain scenarios, it is not straightforward, since controlling the level of detail to be removed can be difficult. In addition, such approaches are not universally applicable and thus limited to specific shapes and scenarios.

Both feature-based and dimensionality reduction-based simplification methods are not directly suitable for CAD models used in AR-based training for industrial assembly

lines. The main cause comes from the fact that such methods may not consider the unique requirements and features relevant to assembly instructions. An alternative approach is based on mesh simplification [11][12] which were thoroughly evaluated in our previous work [6] and the method introduced by Lindstrom *et al.* [13] has outperformed the rest of the methods. In this case, the 3D CAD meshes are treated as a whole, without specific information about how the models have been generated or their corresponding features. The primary objective of mesh-based simplification is to progressively reduce the number of vertices without compromising the visual integrity of the model.

A promising way to simplify CAD models used for AR training is to identify redundant features within each model and apply to them mesh simplification strategies. By tailoring the simplification process to each feature, it becomes feasible to achieve more streamlined and visually acceptable simplified models. Among the various features that may be considered, in our work we have focused on the identification of *redundant continuous chains* (Fig. 2). When creating a CAD model, different parts are generated individually and later joined together to form a complete model. To ensure seamless transitions between these parts, continuous chains are introduced during the mesh construction process. These chains are defined as smooth regions of dense mesh triangulations, usually consisting of small triangles, that either round the edges of one component or connect two components together. Once the continuous chains detected, we can effectively remove the entire band of a continuous chain resulting in a significant reduction in the number of vertices. This simplification process helps optimizing the CAD models for improved performance without compromising the visual integrity of the assembly. In our research, we have analyzed a database comprising 730 CAD models, and we have observed that nearly 500 of them contain continuous chains.

The state-of-the-art continuous chain detection technique is based on the segmentation algorithm introduced by Cohen *et al.* [14]. The algorithm involves calculating the minimum and maximum curvature of each vertex, followed by a k-means clustering and subsequent merging of triangles. The segmentation process effectively extracts all vertices that present similar curvature values and are thus considered to lie on the continuous chain.

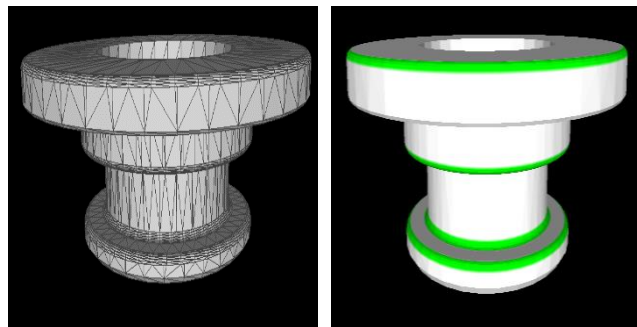




Fig. 2. Continuous chain functionality

Despite the effectiveness of continuous chain curvature-based detection, a significant limitation arises when dealing with CAD models exhibiting substantial deviations in curvature. Two examples of missed and wrong detection are presented in Fig. 3. In both cases, the algorithm fails to accurately identify and distinguish the continuous chains. In Fig 3.a we can see two continuous one at the top and another in the middle, but only the one in the middle is identified. Similarly in Fig. 3b we can see that some of the vertices inside the continuous chain are not detected, while some of them which are not in continuous chains are identified as one. While local optimization may be attempted, it is model-specific and requires manual intervention. To overcome such challenges, we propose a novel approach that leverages deep learning techniques to identify continuous chains with higher accuracy and robustness.

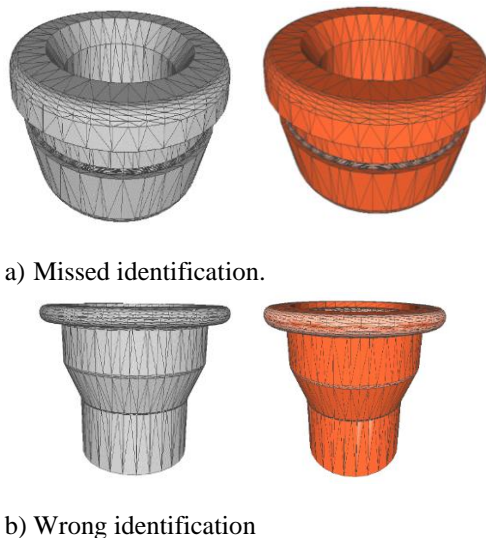


Fig. 3. Limitations of the curvature-based continuous chain identification

### III. PROPOSED METHODOLOGY

The proposed methodology comprises three key phases: database preparation, 2D segmentation using deep learning models, and targeted simplification of the detected continuous chains.

The 3D models are first converted into images by 3D/2D projection. Subsequently, an image segmentation algorithm is applied to identify the corresponding 2D appearances of the continuous chains. The multi-view segmentation results are then mapped back onto the original 3D domain. A dedicated simplification algorithm is finally designed to effectively simplify the identified features while preserving the model's essential characteristics and visual integrity.

Let us now further detail the three steps involved.

#### A. Database creation :

The initial step in our proposed methodology involves creating a comprehensive database of 2D images derived from 3D models captured at various angles of view (Fig. 4).

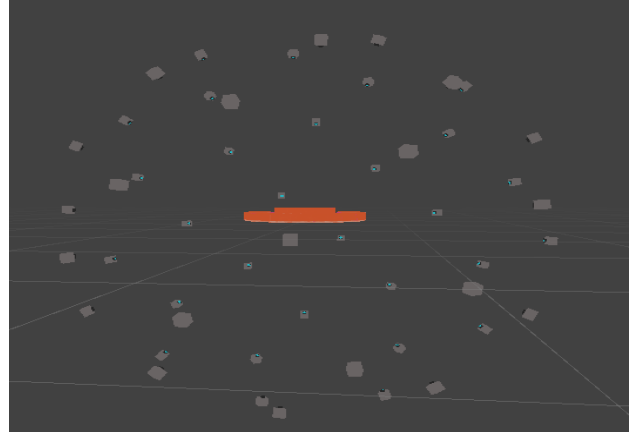


Fig. 4. Camera distribution for capturing a set of images

Our database is derived from a total number of 15 3D mesh models and comprises three sets of images (See Fig. 5):

1. Images of the original 3D model captured from different angles of view and rendered in wireframe visualization mode. We have explicitly defined 50 camera positions which are uniformly spread around a sphere.
2. Images of continuous chain masks extracted from the 3D model, corresponding to the original images. These masks highlight the regions with continuous chains within the model. The masks are obtained by applying the state of art segmentation technique [14], and are further adjusted manually to remove abnormalities and provide a reliable ground truth.
3. Images of the 3D model with unique face colors (randomly selected) corresponding to the original images. These images serve to the purpose of mapping the segmented results from the 2D domain back to the 3D domain.

The decision to perform segmentation in the 2D domain, rather than directly in 3D, stems from the diverse and non-continuous nature of the CAD model data. Due to the varying structures, shapes and sizes of CAD models, manually labelling classes and distinguishing between different model types/features becomes challenging. By operating in the 2D domain, we can leverage advanced segmentation algorithms tailored for 2D image processing, thereby enhancing the accuracy and efficiency of the segmentation process. Later in the methodology, we efficiently map the segmented results back onto the original 3D models using the unique face-colored images, ensuring a seamless integration of the simplified features into the overall 3D representation. After that simplifying these features becomes quite straight forward.



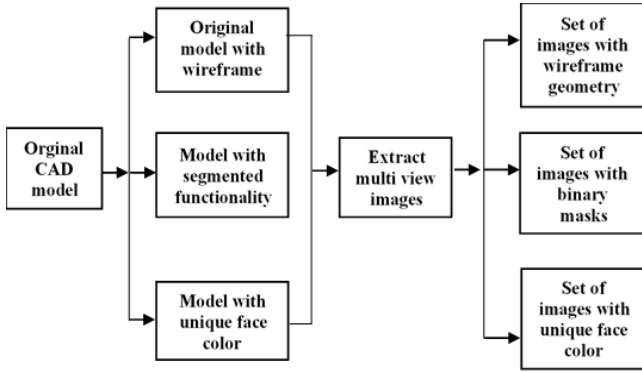


Fig. 5. Database preparation

### B. Training with DL techniques

Various deep learning techniques have been proposed for performing segmentation on 2D images [15][16]. When dealing with a limited amount of training data, UNet has demonstrated superior performance compared to other methods. Given our relatively small training dataset of 15 3D models, we have chosen UNet [16] as our primary model for segmenting the continuous chains in the 2D domain. To accomplish this, we have used a pre-trained UNet model, originally designed for segmenting medical images and identifying abnormal regions.

UNet is a convolutional neural network architecture that incorporates two consecutive  $3 \times 3$  convolutions, each followed by a rectified linear unit (ReLU), and a  $2 \times 2$  max pooling operation for down-sampling. The input images, along with their corresponding segmented masks, are utilized to train the network. The training set consists of 13 models, comprising a total of 650 images, while 2 additional models are reserved for testing. The output of the UNet model is a segmented image, effectively highlighting the regions representing continuous chains within the CAD models. To optimize the default parameters of the UNet model for CAD model segmentation, we leveraged the power of Optuna [18], 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. In our experimentation, we considered the following parameters for our UNet model: Learning Rate: 0.001, Optimizer: SGD, Batch Size: 8, Loss Function: MAE, and Saturation Epoch: 12. 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 in segmenting continuous chains within CAD models. The enhanced model can now be confidently utilized in our CAD model segmentation process. Fig. 6 illustrates the evolution of the training and validation loss 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.

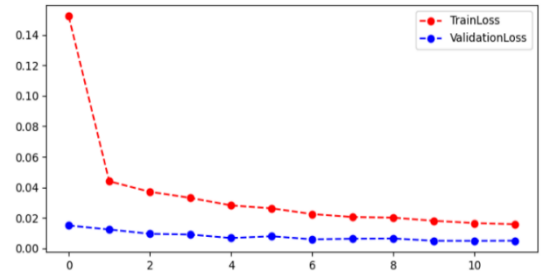


Fig. 6. Loss graph

Fig. 7 presents the final results of our segmentation process, displaying the original CAD model alongside the accurately identified continuous chain functionalities. The network successfully detects the continuous chains in this particular view of the model. It is essential to mention that we have a total number of 50 views captured from all sides of the same model. These multiple views are crucial in mapping the detected functionalities from the 2D domain back onto the 3D domain, ensuring a comprehensive representation of the continuous chains throughout the entire 3D model.

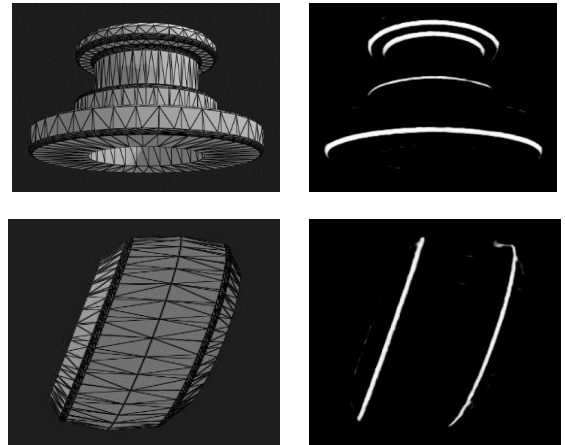


Fig. 7. Segmentation results

### C. 2D to 3D mapping

Segmentation in the 2D domain yields multiple images, each capturing the continuous chain functionality at a specific angle. To facilitate mapping between the 2D and 3D domains, we employ a unique color identity scheme (See Fig. 8). More precisely, each face of the 3D CAD model is assigned a distinct color. When the 3D model is converted into a 2D image, each face color corresponds to multiple pixels in the image. In the final segmented images, we identify the pixel colors that represent the segmented region. These pixel colors are traced back to a specific face number in the 3D domain. Consequently, for each view, we can map the segmented region from the 2D domain back to the corresponding region in the 3D model. By performing this mapping for different views, we aggregate the results to achieve a comprehensive

identification of the segmented continuous chain in the 3D domain.

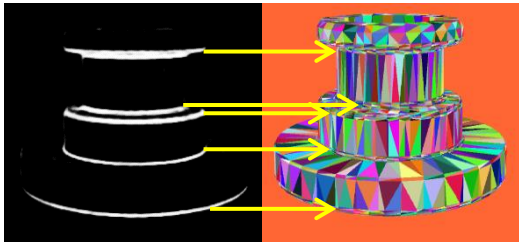


Fig 8. 2D to 3D mapping

#### D. Simplification of continuous chains

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, 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 the corresponding vertices.

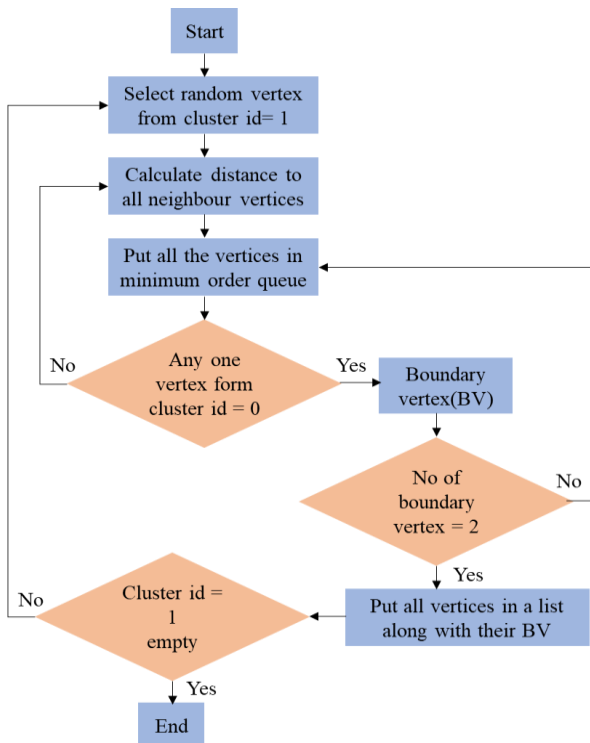


Fig. 9. Identifying vertices for simplification

The flowchart in Fig. 9 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, one is from 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 continuous chain and traverse through the vertical plane of that particular vertex with the help of shortest distance until we reach the boundary vertex on either side. Once we find a vertical plane of vertices they are separated into another list. We continue the process until all the vertices from cluster id 1 that is all the vertices from continuous chain are listed along with their respective vertical plane vertices and boundary vertices.

The two boundary vertices are collapsed at their centroid creating a single point (See Fig. 10b) representing the simplified shape and edges while preserving the overall shape and structure of the model.

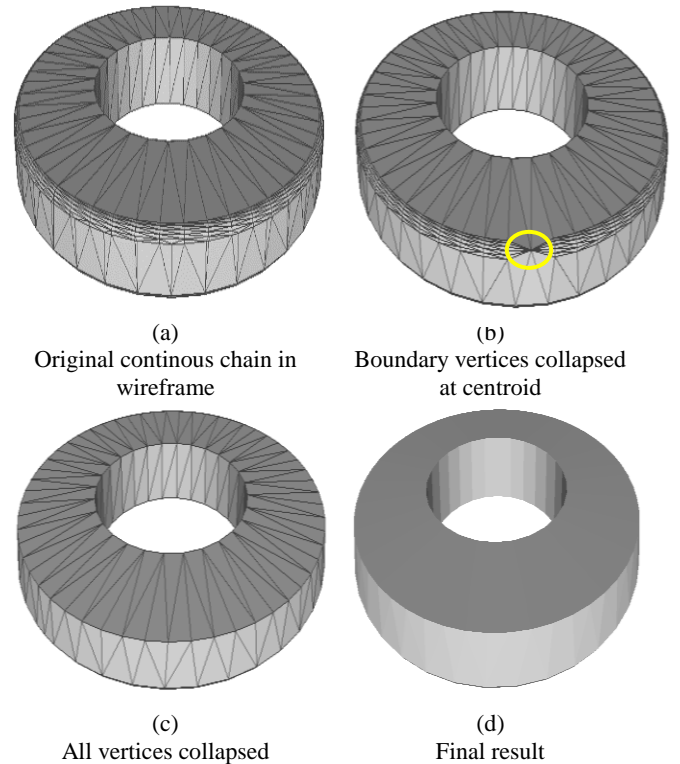


Fig 10. Merging continuous chain

The final results are illustrated in Fig. 11. The first model (Fig. 11a) 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 (Fig. 11b), 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 in turn increasing the framerate of the application they are being used in ensuring fluid rendering. 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. Also, we can compare it against the state-of-the-

art method [13] which had previously performed better. In Fig. 11e&f we can notice the abnormalities in the edges of the model where as our method is more visually acceptable. 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.

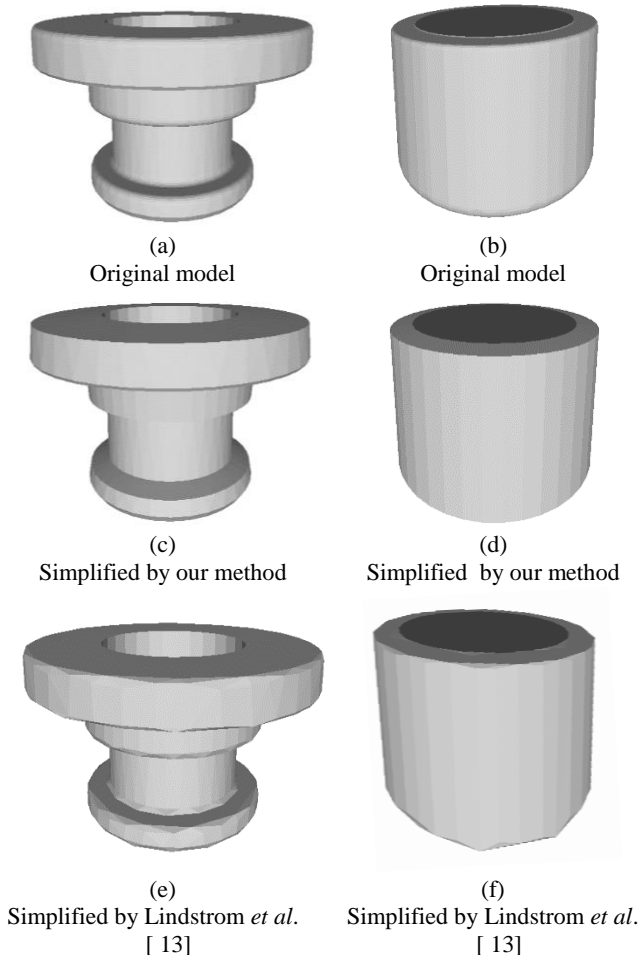


Fig. 11. Final simplified models

#### IV. CONCLUSION AND PERSPECTIVES

Over the past few years, CAD model simplification has been a prominent area of research. Within this context, the trend of simplifying CAD models specifically for AR applications has emerged more recently. This paper proposes a novel approach for 3D CAD mesh simplification by combining mesh-based and feature-based simplification. A deep learning method is employed to identify redundant continuous chains, within the 2D domain. Furthermore, a dedicated simplification strategy is developed, focusing on collapsing these continuous chains into single chain, thus reducing model complexity.

For future research, advanced deep learning techniques can be investigated to achieve a better feature identification and segmentation results. Lastly, extending the application of

the proposed method to various other features in complex CAD models will even simplify the models enhancing the performance of AR-based training in industrial assembly lines and beyond.

#### REFERENCES

- [1] Feiner, Steven, Blair MacIntyre, and Dorée Seligmann. Knowledge-based Augmented Reality. *Communications of the ACM* 36, 7 (July 1993), 52-62.
- [2] Hořejší, Petr (2015): Augmented Reality System for Virtual Training of Parts Assembly. In: *Procedia Engineering*, S. 699–706. DOI:10.1016/j.proeng.2015.01.422.
- [3] Jürgen Hahn, Bernd Ludwig, Christian Wolff, Augmented reality-based training of the PCB assembly process, *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia* November 2015 Pages 395–399 <https://doi.org/10.1145/2836041.2841215>
- [4] G. Reinhart, C. Patron, Integrating Augmented Reality in the Assembly Domain - Fundamentals, Benefits and Applications, *CIRP Annals*, Volume 52, Issue 1, 2003, Pages 5-8, ISSN 0007-8506, [https://doi.org/10.1016/S0007-8506\(07\)60517-4](https://doi.org/10.1016/S0007-8506(07)60517-4).
- [5] Davide Manca, Roberto Totaro, Salman Nazir, Sara Brambilla, Simone Colombo, Virtual and Augmented Reality as Viable Tools to Train Industrial Operators, Elsevier, Volume 31, 2012, Pages 825-829, ISSN 1570-7946, ISBN 9780444595058, <https://doi.org/10.1016/B978-0-444-59507-2.50157-8>.
- [6] Abhayadhatri 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.
- [7] Mun, D., Kim, B.C. Extended progressive simplification of feature-based CAD models. *Int J Adv Manuf Technol* 93, 915–932 (2017). <https://doi.org/10.1007/s00170-017-0491-y>
- [8] Lee SH (2005) Feature-based multiresolution modeling of solids. *ACM Trans Graph* 24(4):1417–14413.
- [9] Lee SH, Lee K (2012) Simultaneous and incremental feature-based multiresolution modeling with feature operations in part design. *Comput Aided Des* 44(5):457–483
- [10] Rui Sun, Shuming Gao, Wei Zhao, An approach to B-rep model simplification based on region suppression. *Computers & Graphics*, Volume 34, Issue 5, 2010, Pages 556-564, ISSN 0097-8493, <https://doi.org/10.1016/j.cag.2010.06.007>
- [11] P. Agarwal and S. Suri. Surface approximation and geometric partitions. In *Proceedings Fifth Symposium on Discrete Algorithms*, pages 24–33, 1994.
- [12] M. J. DeHaemer, Jr. and M. J. Zyda. Simplification of objects rendered by polygonal approximations. *Computers & Graphics*, 15(2):175–184, 1991.
- [13] Lindstrom, Peter & Turk, Greg. (1998). Fast and Memory Efficient Polygonal Simplification. *IEEE Visual*. 10.1109/VISUAL.1998.745314.
- [14] David Cohen-Steiner and Jean-Marie Morvan. 2003. Restricted delaunay triangulations and normal cycle. In *Proceedings of the nineteenth annual symposium on Computational geometry (SCG '03)*. Association for Computing Machinery, New York, NY, USA, 312–321. DOI: <https://doi.org/10.1145/777792.777839>
- [15] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation (2014), arXiv:1411.4038 [cs.CV]
- [16] Seyedhosseini, M., Sajjadi, M., Tasdizen, T.: Image segmentation with cascaded hierarchical models and logistic disjunctive normal networks. In: *Computer Vision (ICCV)*, 2013 IEEE International Conference on. pp. 2168–2175 (2013)
- [17] Ronneberger, O., Fischer, P. and Brox, T., 2015, October. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (pp. 234-241). Springer, Cham.
- [18] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19)*. Association for Computing Machinery, New York, NY, USA, 2623–2631. <https://doi.org/10.1145/3292500.3330701>

