| Universi   | ty of Cincinnati  |  |  |  |  |  |  |  |
|--|---|--|--|--|--|--|--|--|
|  | Date: 10/30/2023  |  |  |  |  |  |  |  |
| I. Manish Raj Aryal, hereby submit this degree of Master of Science in Mecha | s original work as part of the requirements for the anical Engineering. |  |  |  |  |  |  |  |
| It is entitled:<br>ImVR : Enabling Immersive Explora<br>Geometries           | tion Environment for Complex Organic                                    |  |  |  |  |  |  |  |
| Student's name: Manish Raj   | Aryal   |  |  |  |  |  |  |  |
|  | This work and its defense approved by:                                  |  |  |  |  |  |  |  |
|  | Committee chair: Sam Anand, Ph.D.                                       |  |  |  |  |  |  |  |
| 1 <i>ā</i> r   | Committee member: Jillian Aurisano, Ph.D.                               |  |  |  |  |  |  |  |
| Cincinnati   | Committee member: Manish Kumar, Ph.D.                                   |  |  |  |  |  |  |  |
|  |   |  |  |  |  |  |  |  |
|  |   |  |  |  |  |  |  |  |
|  | 47801   |  |  |  |  |  |  |  |
|  |   |  |  |  |  |  |  |  |
|  |   |  |  |  |  |  |  |  |
|  |   |  |  |  |  |  |  |  |

# ImVR: Enabling Immersive Exploration Environment for Complex Organic

# Geometries

A thesis submitted to the

Graduate College

Of the University of Cincinnati

in partial fulfillment of the requirements for the degree of

# **MASTERS OF SCIENCE**

in the Department of Mechanical and Material Engineering

of the College of Engineering and Applied Sciences (CEAS)

By

# Manish Raj Aryal

Bachelors of Engineering (B.E.) – Mechanical Engineering

Tribhuvan University, Kathmandu, Nepal - 2020

**Committee Chair: Dr Sam Anand** 

#### Abstract

The assessment and representation of intricate design and simulation results for organic geometries through traditional devices (tablets, computers, phones) possess a set of significant limitations. Consequently, identifying potential manufacturability, design issues, and opportunities for design enhancements remains challenging in the early design stage. The inability to identify these possible areas often leads to time-consuming and costly design revisions and iterations. In contrast, Virtual Reality (VR) offers many benefits over the conventional 3D computer-aided design (CAD) environment, particularly regarding visualization and analysis of complex organic geometries. The advantages of VR may be more visible as designs become more organic or complex. This thesis aims to investigate and develop an immersive environment that enables designers and engineers to explore complex organic geometries. This thesis focuses on generating an Immersive Virtual Reality (ImVR) environment that could allow designers to evaluate and identify critical design areas of a part model based on the design for additive manufacturing (DfAM) parameters. Custom Python scripts and Visualization Toolkit (VTK) algorithms perform surface and volume rendering across the CAD model in the virtual environment. Two case studies of different organic designs with tools and an immersive visualization environment are presented to analyze the effect of immersive design analysis of a CAD model. The ImVR toolset also includes a collaborative multiplayer feature and an artificial intelligence (AI) chatbot powered by ChatGPT. The chatbot can provide real-time guidance to users within the immersive environment, while the collaborative feature enables multi-user interaction and visualization in the immersive environment.

# Acknowledgment

I would like to take this moment to express my heartfelt appreciation to everyone who helped me in my Master's journey.

To begin with, I would like to thank my friends and family, who have supported and guided me throughout this journey.

I sincerely thank my research and academic mentor, Dr. Sam Anand, for providing me with unwavering support and many opportunities throughout my graduate studies. Additionally, I would like to thank Dr. Jilian Aurisano and Dr. Manish Kumar for their roles as members of my thesis defense committee.

# **Table of Contents**

| Abstract   | ii                    |
|--|-----------------------|
| Acknowledgment   | iv                    |
| Table of Contents  | V                     |
| List of Figures  | vii                   |
| 1. Introduction  | 1                     |
| 1.1 Research Motivation  | 4                     |
| 1.2 Research Objective   | 5                     |
| 1.3 Thesis Outline   | 5                     |
| 2. Literature Review   | 7                     |
| 2.1 Immersive Visualization  | 7                     |
| 2.2 AM Specific Critical Parameters                                    | 10                    |
| 2.3 ChatGPT and Voice Translation                                      | 12                    |
| 3. Methodology   | 13                    |
| 3.1 Algorithms for Calculation and Visualization of AM-specific Critic | cal Design Parameters |
|  | 14                    |
| 3.1.1 Slicing Algorithm and Cusp Error Analysis                        | 14                    |
| 3.1.2 Detection of Thin Regions  |                       |
| 3.1.3 Thin-to-thick Transitions  | 20                    |
| 3.1.4 Identification of Sharp Corners                                  | 23                    |
| 3.1.5 Small Opening  |                       |
| 3.1.6 Generation of Support Structures                                 | 27                    |
| 3.2 Thermal Stress Simulation  |                       |
| 3.3 Immersive VR Environment   |                       |
| 3.3.1 System Design and Architecture                                   |                       |
| 3.3.2 Workflow and Process Operation                                   |                       |

| 3.3.3 Collaborative VR Environment          |    |
|---|----|
| 3.4 Voice Recognition and Translation Model |    |
| 3.5 ChatGPT Assistant AI Bot                |    |
| 3.5.1 Training Process                      | 40 |
| 3.5.2 Need and Functionality of AI Bot      | 41 |
| 3.5.3 Operation Process                     | 42 |
| 4. Case Study and Results                   | 44 |
| 4.1 Lattice Wrench Model                    | 46 |
| 4.1.1 Sharp Corners and Vertices            | 46 |
| 4.1.2 Small Opening and Thin Sections       | 47 |
| 4.1.3 Thin-to-thick Transition              | 48 |
| 4.1.4 Cusp Error and Slicing Mechanism      | 49 |
| 4.1.5 Support Structures                    | 50 |
| 4.1.6 Thermal Stress Simulation             | 52 |
| 4.2 Topology-optimized Bracket Model        | 53 |
| 4.2.1 Sharp Corners                         | 53 |
| 4.2.2 Small Opening and Thin Sections       | 54 |
| 4.2.3 Thin-to-thick Transition              | 55 |
| 4.2.4 Cusp Error and Slicing Mechanism      | 56 |
| 4.2.5 Support Structures                    | 57 |
| 4.2.6 Thermal Stress Simulation             | 59 |
| 4.3 Collaborative VR Environment            | 59 |
| 4.4. Assistive ChatGPT Chatbot              | 60 |
| 5. Conclusions and Future Scope             | 63 |
| References                                  | 65 |

# List of Figures

| Figure 1: Visualization of brain data analysis in VR [8]                                       | 2  |
|--|----|
| Figure 2: Application of VR for learning juggling operation [5]                                | 3  |
| Figure 3: Representation of molecular biological structure with UnityMol on a spherica         | ıl |
| display [22]   | 8  |
| Figure 4: Visualization of 3D medical model with image data [7]                                | 9  |
| Figure 5: Conceptualization of collaborative Immersive Virtual Environment presented b         | y  |
| NREL)[23]1   | 0  |
| Figure 6: Variation of visualization through immersive and non-immersive technologies [21      | ]  |
|  | 2  |
| Figure 7: Overall Schematic Representation of the ImVR tool                                    | 4  |
| Figure 8: Sample wavefront and material file generated in the ImVR environment1                | 6  |
| Figure 9: a) Input sample model (taken as reference) b) Visualization of the model showing the | e  |
| slicing mechanism of the part model visualized through spheres in VR1                          | 8  |
| Figure 10: a) Input sample model (taken as reference) b) Visualization of the model showing    | a  |
| thin section of the part model visualized through spheres in VR2                               | 0  |
| Figure 11: a) Input sample model (taken as a reference) b) Visualization of the model with     | h  |
| combined sections for thin, thick, and transition areas in VR2                                 | 3  |
| Figure 12: Variation of curvature across a sample STL file2                                    | 4  |
| Figure 13: a) Input sample model (taken as a reference) b) Visualization of the part mode      | el |
| showing the variation of Gaussian Curvature through color in VR c) Generated part mode         | el |
| showing sharp corners as spheres in VR2  | 6  |
| Figure 14: a) Input sample model (taken as a reference) b) Visualization of the part mode      | el |
| showing small openings in VR   | 7  |
| Figure 15: a) Input sample model (taken as a reference) b) Visualized output file for the part | rt |
| showing part, cube supports, and substrate in VR c) Visualized output file for the part showin | g  |
| part, cylindrical supports, and substrate in VR  | 0  |
| Figure 16: Flowchart for the visualization methodology for stress simulation                   | 1  |
| Figure 17: Comparison between voxelized and faceted part model                                 | 2  |
| Figure 18: Schematic Representation of component for ImVR Virtual Environment                  | 3  |
| Figure 19: Operational workflow for the virtual environment                                    | 5  |
| Figure 20: Default Avatar presence for ImVR environment  | 7  |
| Figure 21: Methodology of collaboration inside the ImVR environment                            | 8  |

| Figure 22: Overview of the operation of Whisper AI and ChatGPT bot in ImVR40                    |
|---|
| Figure 23: Need and application of ImVR bot in different scenarios                              |
| Figure 24: Case study models in ImVR a) Lattice Wrench Model b) Topopt Bracket44                |
| Figure 25: a) Immersive environment for representation of part models b) Generated CAD          |
| models to mimic real environment for ImVR tool c) Researcher analyzing the VR model             |
| through HTC Vive Pro d) Point-of-view (POV) of the menu bar, toolbar, and ImVR                  |
| environment   |
| Figure 26: a) Rendered 3D color plot generated on variation of Gaussian curvature b)            |
| Generated sphere structure across the part model for visualization of sharp corners on points   |
| with higher Gaussian curvature value47  |
| Figure 27: a) Visualization of critical thin points with the ability to show and hide sectional |
| lines across part models for ImVR tools for thin section b) Rendered 3D line plot to show       |
| lines/area of critical region based on small openings   |
| Figure 28: Visualization of thin-to-thick transition sections across the part model49           |
| Figure 29: Visualization of staircase model with section showing cusp error across the part     |
| model   |
| Figure 30: Visualization of support structures across the part model a) Generated cube and      |
| cylindrical support on part model at current orientation b) Real-time generation of cube        |
| supports when the part model is rotated by 40 degrees across the X-axis                         |
| Figure 31: Variation of visualization plots generated across FEA software and ImVR toolkit      |
| for Lattice Wrench Model  |
| Figure 32: Sharp Corner Visualization a) Gaussian curvature across a sharp point (CAD defect)   |
| across the part model b) Presence of sharp corners across those points noted during             |
| visualization   |
| Figure 33: a) Visualization of small opening points across the part model b) Visualization of   |
| small opening sections across the part model  |
| Figure 34: Visualization of thin to thick transitional areas for the part model in the ImVR     |
| environment   |
| Figure 35: Visualization of cusp error across slices for a part model57                         |
| Figure 36: Visualization of support structures for a part model a) Generated support with no    |
| rotation b) Generate support structures with rotation of part model by user                     |
| Figure 37: Comparison of generated results from ANSYS Workbench and VTK rendering               |
| algorithm for Topopt Bracket  |
| Figure 38: Synchronized collaborative immersive VR environment                                  |

| Figure  | 39:    | User   | interaction   | with  | the | voice   | translation | and   | trained  | ChatGPT      | Bot    | for  |
|---------|--------|--------|---------------|-------|-----|---------|-------------|-------|----------|--------------|--------|------|
| underst | tandiı | ng the | DfAM toolk    |       |     |         |             |       |          |              | •••••  | 62   |
| Figure  | 40: 1  | The in | teraction bet | tween | the | user an | d trained C | hatGl | PT Bot f | for help reg | gardin | ng a |
| design  | issue  |        |               |       |     |         |             | ••••• |          |              | •••••  | 62   |

# 1. Introduction

Virtual Reality (VR) is an advanced interface between computers and humans that simulates a physical environment, either real or imaginary. It generally combines software and hardware to allow users a three-dimensional experience, enabling them to explore and interact with objects in the 3D environment [1]. Immersive virtual reality is a subset of VR technology that aims to deliver users with the utmost immersion and realism within their experience in the virtual environment [2]. Immersive virtual reality establishes a profound sense of human presence, allowing users to perceive themselves physically in the virtual environment, bridging the divide between reality and the virtual realm by stimulating the sensory modality of vision, sound, and touch.

Immersive visualization offers numerous benefits in comparison to traditional methods of tool visualization, allowing this technology to make it highly effective in the surgical, ergonomic, medicinal, and product design domains [3]–[5]. This technology drives to transform the design experience through multi-sensory engagement, a collaborative multi-user feedback model, and the creation of highly accurate, intricate, lifelike model representation. Furthermore, immersive visualization has demonstrated its effectiveness in time and cost savings [6]–[11]. It significantly diminishes operation errors, enhances knowledge retention compared to traditional teaching methods, and provides valuable training support, enabling industries to employ less-skilled personnel. Figure 1 exemplifies the utilization of virtual reality to visualize brain data and perform data analysis using virtual reality toolkits.



Figure 1: Visualization of brain data analysis in VR [8]

VR is significantly advantageous in product design, dissection, and analysis compared to traditional visualization platforms, arguing that VR can offer a robust design evaluation tool [12]. Since the mid-1990s, researchers have envisioned implementing and integrating VR to enhance the product design process and quality.[13]. In the past, the utilization of VR necessitated a broad range of expertise that hindered the widespread research and adaptation of VR technology. However, in recent years, noteworthy developments in the VR space, especially with the hardware development of head-mounting displays (HMDs), have led to the emergency of high-quality virtual reality systems at an affordable price for companies and industries [14]. As a result of increased affordability and accessibility, industries are increasingly interested in incorporating VR into the part design and process design pipeline [15]. Figure 2 illustrates a study where VR was employed to teach juggling techniques and enhance individuals' juggling skills through immersive VR training. This study observed that immersive VR training enhanced mind-muscle memory and positively impacted individuals' cognitive processes.



Figure 2: Application of VR for learning juggling operation [5]

In recent decades, additive manufacturing (AM) has transcended modern manufacturing by facilitating the production of parts with intricate geometrical features. The AM process provides unprecedented flexibility and freedom to design models because it does not have the constraints imposed by machining and tooling restrictions common in traditional manufacturing processes. Thus, AM has been identified as one of the pillars of Industry 4.0/5.0 for manufacturing complex parts, especially in the aerospace and automobile industries [16]. It is essential to conduct a comprehensive analysis, evaluation, and understanding of the unique design capability and constraints associated with AM to harness the full potential of AM and capitalize on its limitless potential [17].

Critical design considerations such as minimum feature size, thermal stress management, support structure generation, and stress concentration mitigation must be analyzed while designing a part suitable for AM. Moreover, AM introduces heightened responsibility in the design phase because it allows the designer freedom to create intricate structures like lattices and organic geometries. However, designers must ensure that the manufacturability of a part design is considered during the design stage [18].

While AM offers exciting opportunities, it demands adherence to a specific ruleset and presents some limitations. The need to adhere to these design rules highlights the importance of developing a comprehensive understanding and analysis of Design for Additive Manufacturing (DfAM) parameters.

### **1.1 Research Motivation**

The analysis and evaluation of parts intended for AM entail a proactive approach, employing specialized algorithms and traditional visualization tools to facilitate the design evaluation process. The assessment and visualization of complex design parameters through a conventional CAD environment can lead to unforeseen errors across critical areas for the design [18], [19]. Furthermore, the utilization of physical prototypes for analysis necessitates the production of parts through AM, which not only increases the overall cost of the design process but also makes design changes and improvements expensive and time-intensive if revisions and iterations are required.

VR offers an exciting prospect for addressing these challenges, potentially revolutionizing the design of DfAM analysis and reducing evaluation time for part models while highlighting critical design areas that may prevent build failures. VR can also conduct "what-if" scenarios, elevate user interaction, and offer manipulability of part models, potentially enhancing a designer's intuitive grasp of the process. Especially in the realm of complex organic structures, conventional visualization and analysis through CAD have often yielded unexpected errors within critical design areas [20]. Hence, the need for immersive VR is more pronounced while analyzing organic geometrical models. This technology promises to enhance precision and enable a meticulous evaluation of these intricate geometries [21].

The integration of VR with DfAM analysis can lead to the formulation of effective methods for reducing AM-related design errors, enhancing intuitive design thinking, and expediting the decision-making process. This innovative approach promises to streamline the AM design process, saving resources and time across the system while ensuring better overall results.

In addition, the immersive VR toolkit present in this work includes an AI bot powered by an open interface ChatGPT model. The ChatGPT model offers users an interactive framework

within VR to navigate and operate seamlessly, enhancing overall user feedback and productivity during design analysis.

# **1.2 Research Objective**

This thesis aims to develop an immersive VR environment that allows users to analyze design parameters related to the DfAM process. The methodology developed in this research project enables users to visualize and experience real-time visualization based on analysis parameters within the VR environment. Furthermore, the VR environment supports collaboration interactions among multiple users, allowing them to exchange information seamlessly through the immersive headset. This collaborative VR approach enables users to leverage their expertise across various design problems.

The design, rendering, and simulation algorithms can extend to accommodate various types of analysis, such as finite element analysis (FEA) and biomedical analysis, offering versatility in the immersive environment. Furthermore, a customized ChatGPT-based chatbot equipped with voice recognition and translation capabilities specifically trained in additive manufacturing processes and mechanisms has been integrated into the VR tool. This bot aids users in the immersive experiences, assisting them in the AM analysis of a part model. The research methodology has been validated by analyzing two test case geometries within the immersive environment, demonstrating its effectiveness and potential for broader application.

# **1.3 Thesis Outline**

The thesis consists of five chapters, each serving a distinct purpose. Chapter 1 introduces immersive visualization, outlines the research motivation, and explains the thesis's objective. Chapter 2 presents a concise yet comprehensive literature review. Chapter 3 describes the design methodology employed for generating an immersive VR environment. This chapter also elaborates on the design algorithms utilized to create and visualize the immersive environment.

Chapter 4 presents two case study models to demonstrate the methodology described. Chapter 5 describes the conclusion and future scope of the thesis work.

#### 2. Literature Review

This chapter discusses literature for immersive visualization, AM-specific feature recognition methods, and the application of large language modeling (LLM) for assistive systems.

# 2.1 Immersive Visualization

The application of immersive visualization to visualize complex organic geometries has gained significant attraction over the years as VR hardware and software progress. The ability to manipulate three-dimensional data to visualize scientific models and applications is a critical analysis tool.

Doutreligne et al. [22] presented an interactive scientific visualization approach utilizing UnityMol, a *Unity* plugin tailored for biomolecular system visualization. UnityMol is a virtual laboratory platform boasting a graphical representation of molecular data and interactive analytical tools. The research emphasized the potential of dynamic and interactive simulation through the seamless integration of UnityMol with molecular simulation software. Figure 3 presents an intricate molecular structure showing the interaction between different molecules for the *Unity* engine display system.



Figure 3: Representation of molecular biological structure with UnityMol on a spherical display [22]

Expanding on this concept and the need for complex visualization, Wheeler et al. [7] developed a method for exploiting the visualization capabilities of the visualization tool kit (VTK) with *Unity's* robust support for virtual reality (VR) displays. Wheeler's approach utilized the OpenGL context-sharing platform to facilitate seamless information exchange across *Unity* and VTK, enabling the rendering of complex objects through a native *Unity* plugin. The research focused on employing VTK's volume rendering method to generate data frames across the polygonal objects. Figure 4 highlights a volume-rendered 3D model of the skull present in a VR environment through the process suggested by this methodology. Notably, the research also addressed the challenge of real-time rendering virtual objects in the VR environment.



Figure 4: Visualization of 3D medical model with image data [7]

Subsequently, Jaeger et al. [8] presented their research where the researchers showcased the immersive visualization of brain analytical data and dynamic graphical model in their study on immersive visualization of brain model. Their research focused on developing a working prototype for visually analyzing brain data in an immersive environment. This environment facilitated the anatomical mapping of brain data via time series function and formulated valuable evaluation metrics for comprehending brain function.

Brunhart-Lupo et al. [23] demonstrated the necessity of collaboration across a VR exploitation tool to facilitate user communication. Their paper delved into developing a collaborative tool that could allow researchers to immerse and explore statistical parameters related to the city's energy profile. Figure 5 shows the joint exploration developed by Brunhart-Lupo at the National Research Energy Laboratory (NREL) for showing energy variation through spherical structures.



Figure 5: Conceptualization of collaborative Immersive Virtual Environment presented by NREL)[23]

The need for effective collaboration in an immersive environment gained further traction as Hoppe et al. [24] elaborated on the importance of establishing a shared perspective for communication between avatars in VR. Their research centered on human-centered computing and collaborative interaction, aiming to create meaningful shared views and a collaborative environment for seamless user information transfer. Additionally, Hoppe emphasized the crucial role of communication and feedback systems within VR.

# 2.2 AM Specific Critical Parameters

In the past decade, researchers have significantly worked on detecting and reviewing various design techniques for identifying AM-specific critical parameters [25]–[27]. Zhang et al. [28] presented pre-processing computational algorithms for the metal powder AM processes. The paper focuses on the evaluation of different AM parameters and focuses on calculating the producibility index for the part model. Zhang et al. presented algorithms for generating support structures, contact areas, thin sections, and thin-to-thick transitions for part models. These

algorithms visualized a controlled CAD geometry and focused on the production of a part through 3D printing.

Melo et al. [29] explored the scope of Extended Reality (XR) and its utilization in the AM process. The investigation primarily concentrated on how XR technology could serve as a means for location-independent training, aiding individuals in comprehending the intricacies of the AM process and its underlying mechanisms. Student focus groups were created and were subjected to immersive technology to gain insight into AM's Laser Powder Bed Fusion process (LPBF). The study's results found the necessity of immersive visualization tools, particularly while visualizing complex AM geometries.

Ostrander et al. [21] presented a comparative analysis between VR and CAD methodologies to assess their efficacy in the evaluation of parts for additive manufacturing. The research employed an evaluation worksheet as a pivotal tool for discerning the impact of immersive visualization on the analysis of the part model, especially with its added complexity. Ostrander's research revealed that the developed VR toolkit, while less accurate than its CAD counterpart for assessing DfAM features, offered advantages in terms of evaluation efficiency, particularly for intricate structural designs. Figure 6 provides a sneak peek into both immersive and non-immersive technologies utilized by Ostrander for the in-depth analysis of complex structural models.

11



Figure 6: Variation of visualization through immersive and non-immersive technologies [21]

# 2.3 ChatGPT and Voice Translation

In the age of artificial intelligence (AI), ChatGPT has enabled users to create customized chatbots, empowering them to tailor their behavior and responses based on specific data prompts.

Jeon et al. [30] delve into the necessity and potential of a speech-recognition-based chatbot designed for learning language via voice recognition. This research examined numerous voice-based chatbots and evaluated the potential application of such systems. Furthermore, the study highlighted the significant promise of incorporating chatbots within an immersive and exploration environment. The adaptation of ChatGPT in the immersive environment offers users a deeper immersion in a virtual environment, amplifying the overall experience.

Shafeeg et al. [31] focused on developing a voice assistant interface seamlessly integrated with ChatGPT. They identified a need for a ChatGPT-based bot equipped with voice translation capabilities, especially to familiarize users with novel game mechanisms. Furthermore, the research highlighted the value of voice-recognition bots, enriched with ChatGPT's extensive knowledge, primarily to address information in an unknown environment.

#### 3. Methodology

In this research, ImVR tools and techniques create an immersive exploration environment for analyzing complex organic geometry structures by breaking them into three crucial components: design parameter computation, parameter visualization, and immersive environment enhancement. These elements, meticulously crafted to ensure a seamless user experience, constitute the cornerstone of our methodology.

The primary emphasis of our immersive visualization tool is to highlight the visualization of critical design features for the AM process that revolve around determining the smallest feasible feature size, generation of support structure, effective thermal stress management system, and reduction of stress concentration points [18]. A comprehensive examination and visualization of these parameters facilitate a thorough design process analysis, enabling users to pinpoint critical design areas according to their specifications. This section centers on developing ImVR tools and techniques for meeting these requirements.

Sections 3.1 and 3.2 delve deep into the intricacies of these design parameters and simulation results for the part model that offer comprehensive insights into the calculation and visualization associated with minimum feature size, thermal stress, stress concentration, and support generation. Section 3.3 expands on the techniques used for the immersive visualization environment. Section 3.4 emphasizes incorporating avatars and creating collaborative VR space for seamless communication with different personnel. These features enhance the tool's utility for teams working on intricate design work, promoting effective communication and shared VR exploration. Lastly, in section 3.5, ImVR focuses on integrating a ChatGPT assistant chatbot equipped with translation and voice recognition capabilities within the ImVR environment. It offers users real-time assistance and facilitates cross-language communication within the immersive environment, justifying the application of ImVR as a cutting-edge tool

for complex organic geometry exploration. Figure 7 illustrates the project's methodology and provides a schematic overview of the ImVR tool and its possibilities.



Figure 7: Overall Schematic Representation of the ImVR tool

# **3.1** Algorithms for Calculation and Visualization of AM-specific Critical Design Parameters

Section 3.1 presents the essential algorithms and techniques for determining and analyzing AM-specific critical design parameters. Each subsection focuses on a specific aspect of the analysis that may lead to optimal design outcomes.

# 3.1.1 Slicing Algorithm and Cusp Error Analysis

All AM modalities commonly utilize a standard CAD format to ensure consistency and ease of use. STL (Stereolithography) is a widely used and standardized file format well-suited to represent 3D geometry in a manner compatible with AM technology. For preparing the data for AM processing, the triangular facets of the 3D STL file are sliced into cross-sections of layers typically parallel to the build platform. These layers are then converted to instructional G-codes that 3D printers utilize for layer-by-layer object creation [32]. The ImVR tool implements the visualization of each sliced contour and the associated cusp error in Python by implementing the Pyvista library. The tabulation of slices and calculation of cusp error is determined by expanding the work of Zha and Anand [33] while focusing on generating and controlling the generated visualization from this method. In their algorithm, the input STL file is sliced along the build direction axis to create a slice for each contour, and for each sliced contour, there is a measurement of deviation ( $E_d$ ), that is, the deviation of the sliced contour from the STL file. The cusp error is calculated by performing a sum of individual variation and dividing it by the number of slices. The technique utilized in this thesis presents a visualization of the cusp error and slicing mechanism by defining a custom Wavefront (.OBJ) file with its dedicated material file (.MTL). For the slicing mechanism, each slice of the part is visualized by its ambient color ( $K_a$ ), diffuse color ( $K_d$ ), and specular color ( $K_s$ ). The ambient color is the dominant RGB color, which the visualized file will contain.

In contrast, the diffuse color emitted from the part during part visualization and the specular color emphasize the color reflection in the VR model. To visualize the part cusp error in each layer, the outer boundary of the STL file is grouped and added to the OBJ file model while setting the dissolve factor /transparency (d) to 0.7 for proper visualization of the staircase effect. Figure 8 shows a sample of the generated visualized file. In addition, Algorithm 1 outlines the detailed steps to create the visualization, where S denotes the input 3D file, and S\* represents the output visualizable file (.OBJ and MTL). Figure 9 shows an example of the rendered S\* file generated by algorithm 1 for VR visualization.



Figure 8: Sample wavefront and material file generated in the ImVR environment

Algorithm 1: Visualization of Cusp Error

**Input:**  $S \leftarrow$  input 3D part file

**Output:**  $S^* \leftarrow$  output visualized part model

# Initialization: Attributes {Build Direction, Number of Slices}

for each build direction of S, do

//generate a slice for each contour of the model

step size ← [max bound – min bound] / number of slices

Generate slice contour for each step size

for each sliced contour, do

//separate the vertex and faces of each slice into a data group

Group vertex and faces into a global data matrix

Define a unique identifier and material for each group

end for

//formulate a material file to store material properties of the part model

Define a material file (MTL) and develop different Ka, Ks, and Kd variables

for each vertex group, do

//define the color spectrum for each slice through the color variables

Set K<sub>a</sub>, K<sub>s</sub>, and K<sub>d</sub> for each slice based on the group number

//reducing part transparency of part model to visualize model for VR

**if** group(vertex) = S(vertex) **do** 

Set dissolve factor,  $d \leftarrow 0.7$ 

end if

# end for

//exporting part model for proper visualization in VR

 $S^* \leftarrow Export OBJ$  and MTL files for the part model

end for



Figure 9: a) Input sample model (taken as reference) b) Visualization of the model showing the slicing mechanism of the part model visualized through spheres in VR

#### 3.1.2 Detection of Thin Regions

In an AM machine, thin regions refer to areas where model feature layers fit below the specified threshold. [34]–[36]. Manufacturing part models with thin features significantly compromises component precision because of the inherent resolution limitations of the machine. The presence of thin regions could lead to bending or deformation due to additional residual thermal stress across the thin region area. Therefore, detecting and correcting such thin regions in the design is crucial to a successful AM build.

ImVR adapts the methodology developed by Chowdhury et al. [34] and Zhang et al. [28] for determining thin sections. The authors discretize the sliced contour and shoot a ray from each slice contour to the part model. Then, the intersection point of the ray is determined, and the approximate distance from the ray's start point to the end point of intersection is calculated. If the value is lower than the pre-set value for the AM process, it is determined as a thin sectional point. Visualizing both section and critical thin-points for a part model is crucial. Algorithm 2 outlines the algorithm to visualize these vital points and areas across the part model. The algorithm highlights critical points through a differential color pattern by generating custom spheres across the points, as shown in Figure 10. In contrast, critical areas are visualized by generating custom line renders across the part model. For the algorithm, S denotes the input

3D file, S\* represents the output visualizable file, and C represents the output comma separated value (CSV) file for thin section opening points.

Algorithm 2: Visualization and Analysis of Thin Section Points and Areas

**Input:**  $S \leftarrow$  input 3D part file

**Output:**  $S^* \leftarrow$  output visualized part model, C  $\leftarrow$  output CSV file

Initialization: Attributes {Build Direction, Number of Slices}

for each build direction of S, do

//generate a slice for each contour of the model

step size ← [max bound – min bound] / number of slices

Generate slice contour for each step size

//formulate a material file to store material properties of the part model

Define a material file (MTL) and Ka, Ks, Kd, and d variables

for each sliced contour, do

Discretize the sliced contour of the part model

//set up the vertex of a thin section with a part color and set up part transparency

**if** slice(point) = thin section(vertex) **do** 

Set point  $K_d$ 's  $\leftarrow$  RGB (0 1 0)

elseif slice(point) = S(vertex) do

Set point dissolve factor,  $d \leftarrow 0.7$ 

end if

# end for

//exporting part model and formulate data file for VR visualization

 $S^* \leftarrow Export OBJ$  and MTL files for the part model

 $C \leftarrow Export thin section(vertex) as CSV$ 

//generate and visualize thin sections through custom line renders in Unity

Set S\* as the parent for visualizing points and lines

for each column in C, do

//separate and generate line renders from data files

Parse data file to represent the location of each sharp point

```
Set up line_renderer as children of parent S* for real-time visualization
line_renderer(start) ← thin_section(start)
line_renderer(end) ← thin_section(end)
```

### end for

end for



Figure 10: a) Input sample model (taken as reference) b) Visualization of the model showing a thin section of the part model visualized through spheres in VR

# 3.1.3 Thin-to-thick Transitions

Thin-to-thick transitions represent regions in the AM process where a build layer with a smaller area transitions to a build layer with a larger area in the build direction of the part. In Direct Metal Laser Sintering (DMLS) process, lasers help to melt and fuse metal powders to generate part geometry. The heat generated by the laser in the current layer is passed onto the lower layer and subsequently to the substrate. The transition area between two consecutive layers must be large enough to transfer the thermal energy and not cause thermal distortion during the build process [37]. The thin-to-thick transitions highlight those possible problem areas of transition in a part model.

Zhang et al. [28] presented the methodology for identifying thin-to-thick transitions across the part model. The algorithm identifies thin-to-thick transitions by calculating area ratios, defined as transitional ratios between the sliced contour of the current slice and one directly below it. If the transitional ratio falls below the specified thermal distortion threshold, the corresponding sliced contours are singled out as a possible DFAM violation. Algorithm 3 shows the adapted methodology to visualize transitional thin-thick regions across the part model. In the algorithm, the sections of each transition are separated by their slice contour, where the ambient color across the thin slice is represented by its unique RGB ambient value.

Similarly, the projected and the thick sections are grouped and represented through their respective ambient RGB values. In addition, for analyzing the effect on intrinsic structures, the value of the shape factor across the boundary layer is changed to accommodate higher transparency. These sections are then attached to a unique identifier, and custom toggle boxes are present in the ImVR environment to visualize this special identifier section. An example of the nature of the visualization technique is highlighted in Figure 11.

Algorithm 3: Visualization and Analysis of Transitional Thin Thick Areas

**Input:**  $S \leftarrow$  input 3D part file

**Output:**  $S^* \leftarrow$  output visualized part model

**Initialization:** Attributes {Build Direction, Number of Slices}

for each build direction of S, do

//generate a slice for each contour of the model

step size ← [max bound – min bound] / number of slices

Generate slice contour for each step size

//formulate a material file to store material properties of the part model

Define a material file (MTL) and  $K_a$ ,  $K_s$ ,  $K_d$ , d variables

for each slice contour, do

//separate vertex and faces for thick and thin slices into a data group

**if** slice(vertex) = thin\_surface(vertex) **do** 

Group vertex and faces into the global data matrix Set up a unique identifier for the thin region

//define a unique color system for each data group Set point  $K_a$ 's  $\leftarrow$  RGB (1 0 0) else if slice(vertex) = thick\_surface(vertex) do

Group vertex and faces into the global data matrix

Set up a unique identifier for thick region

Set point  $K_a$ 's  $\leftarrow$  RGB (0 1 0)

*//generate projection by transferring the contour from the surface* 

projected\_surface ← projection(thick\_surface ← thin\_surface)
for each projected\_surface, do

//formulate a vertex and face contour for the projected surface

Group vertex and faces into the data matrix

Set up a unique identifier for the projected region

Set point  $K_a$ 's  $\leftarrow$  RGB (0 0 1)

# end for

elseif slice(vertex) = S(vertex) do

*//set up part transparency* 

Set point dissolve factor,  $d \leftarrow 0.7$ 

end if

# end for

//exporting part model to visualize in VR

 $S^* \leftarrow Export OBJ$  and MTL files for the part model

//attach toggle system in VR to activate defined identifier

**if** toggle(thin) = TRUE **do** 

Setup S\* and call a unique identifier for the thin region

```
else if toggle(thick) = TRUE do
```

Setup S\* and call a unique identifier for the thick region

```
else if toggle(projected) = TRUE do
```

Setup S\* and call a unique identifier for the projected region

#### end if

#### end for



Figure 11: a) Input sample model (taken as a reference) b) Visualization of the model with combined sections for thin, thick, and transition areas in VR

# **3.1.4 Identification of Sharp Corners**

Sharp corners in AM refer to the sharp angle where two edges meet at each layer, which generally results in stress concentration for the part model [38]. In any AM process, sharp corners may be difficult to manufacture due to the inherent diameter size of the laser. Sharp corners, as defined computationally through Gaussian curvature, refer to the points on the part surface where the curvature of the surface undergoes a sudden and significant transitional change. The Gaussian curvature for a surface measures the curvature value at a specific point. The value of Gaussian curvature helps to analyze whether the curvature of the surface is locally concave or convex [39].

For detecting sharp corners, Gaussian curvature values experience a rapid transition from a positive to a negative value. The abrupt surface change helps indicate the location of sharp corners, as the shift generally happens in an angular or pointed feature. Figure 12 highlights a sharp point for a geometry with a change of Gaussian curvature. These points are critical and are mostly a cause of design defects.



Figure 12: Variation of curvature across a sample STL file

Visualizing sharp corners for the immersive environment is accomplished by generating a GL transitional format through a VTK library. This process involves configuring the Gaussian kernel and interpolator utilizing a VTK mapper to apply color volume rendering to the STL geometry. These kernel and interpolators help define the variation of texture and color across the generated CAD model. In addition, the Gaussian values across the part model are determined through a point mapper and a VTK renderer that relays render to parts. The color-enhanced model overlays onto the STL, resulting in rendering a GL transitional format that is subsequently exported to the visualization environment. Algorithm 4 provides the steps for visualizing color-render plots for sharp corners where S denotes the input 3D file, G\* represents the textured part file, and C represents the output CSV file for sharp corner points. In addition, Figure 13 showcases the part models rendered through the sharp corner algorithm suggested in Algorithm 4.

Algorithm 4: Visualization and Analysis of Sharp Corners

**Input:**  $S \leftarrow$  input 3D part file

**Output:**  $S^* \leftarrow$  output color-render part file, C  $\leftarrow$  output CSV file

#### **Initialization:** Attributes {Gaussian Curvature Threshold}

for each vertex of S, do

# //Calculate Gaussian curvature for all vertex and set up maximum curvature threshold

Determine the Gaussian curvature of each vertex

Establish maximum limit for Gaussian Curvature parameters through cutoff value

if Gaussian curvature < maximum threshold, do

#### //formulate a data matrix for each vertex point and curvature value

sharp\_points ← [vertex\_points; Gaussian Curvature value]

# end if

 $C \leftarrow Export sharp_points as CSV$ 

//set up a VTK visualization framework for the part model

Initialize color\_interpolator, gaussian\_kernel, point\_mapper, VTK\_renderer

//define a mathematical function to assign weight based on nearest neighbor search

gaussian\_kernel = f{[vertex\_points]}

//define the transition spectrum between colors for the rendered model

color\_spec\_min (RGB) = min(gaussian\_curvature\_value)

color\_spec\_max (RGB) = max(gaussian\_curvature\_value)

color\_interpolator = f{[gradient\_interpolation(color\_spec\_min; color\_spec\_max)]}
//set up rendering for the part model for VR

VTK renderer =  $f_{i}$ [actor, light, camera position]}

//render the size, color, and opacity for the 3D model

Point Mapper  $\leftarrow f_{\ell}^{\text{[gaussian_kernel; color_interpolator; VTK_renderer]}$ 

Establish emissivity, color radius, and sharpness for the VTK Renderer

//export the part model into the VR environment

 $G^* \leftarrow \text{Export textured part file}$ 

//exporting sharp corners only as another part model for VR

//formulate a material file to store material properties of the part model

Define a material file (MTL) and Ka, Ks, Kd, d variables

for every sharp point, do

//assigning colored spheres to sharp points for visualization

if S(vertex) = sharp\_point(vertex) do

Set point  $K_d$ 's  $\leftarrow$  RGB (1 0 0)

elseif slice(point) = S(vertex) do

*//set up part transparency* 

Set point dissolve factor,  $d \leftarrow 0.7$ 

end if

end for

//export part geometry with indicated sharp point for part visualization

 $S^* \leftarrow$  Export OBJ and MTL files for the part model

end for



Figure 13: a) Input sample model (taken as a reference) b) Visualization of the part model showing the variation of Gaussian Curvature through color in VR c) Generated part model showing sharp corners as spheres in VR

# 3.1.5 Small Opening

Laser AM machines face limitations in manufacturing a specific feature size due to their inability to produce the minimum laser spot size required for a part model. These limitations may create gaps; if they fall within each layer's manufacturable size, they are considered small openings [40]. These gaps exist based on the laser spot diameter during the powder fusion.

The algorithm for detecting small openings is conceptually similar to that for detecting thin features. This algorithm also focuses on a ray-tracing algorithm for sliced contour curves. But compared to the thin section algorithm, the acting ray is shot across the opposite direction during the ray shooting method [28]. The algorithm results in the detection of small opening
points and formulates lines to visualize the part model. The visualization technique for visualizing small openings resembles that for the thin section, as both export the part file and the CSV file into *Unity*. The exported CSV files aid in generating real-time data points, while the part files facilitate the visualization of these vertices. Figure 14 shows an example of the part model developed through the visualization algorithm to note small openings in a part model. In addition, the line renders emphasize sections of the part file while visualizing the small openings sections.



Figure 14: a) Input sample model (taken as a reference) b) Visualization of the part model showing small openings in VR

#### **3.1.6 Generation of Support Structures**

In an AM process, support structures play a crucial dual role in creating overhang and downward-facing features while enhancing heat transfer between part features and substrate. The determination of support structures varies with the build direction for any part model. The change of build direction affects the quality of overhang and downward-facing features, indicating the need for fewer or more support structures.

Zhang et al. [28] defined a method to detect support structures by shooting a ray from the part model to the support substrate. The method calculates the intersection angles and compares the variable with the upper limit of the overhang angle of the feature for support structures. The comparison of the intersection angle and the overhang angle limit helps to determine the need for a support structure for the part model. Algorithm 5 helps users generate and visualize realtime support structures based on the location of the part in VR. For the visualization process, supports, parts and substrate are generated by rendering cubic/cylindrical support across the part and then assigning varying ambient colors across the part model. Figure 15 provides an example of the developed visualization model for a sample part. In the algorithm, *S* denotes the input 3D file, and *S'* represents the output support structure and substrate for the model.

Algorithm 5: Real-time visualization and generation of support structures

**Input:**  $S \leftarrow$  input 3D part file

**Output:**  $S' \leftarrow$  output support structure and substrate for the model

**Initialization:** Attributes {Upper feature limit, Part Orientation}

for support generation in VR, do

#### //set up the initial location as a base and tabulate rotation based on point tracking

Define a base orientation of the part model

Track rotation of part model in VR

Calculate a rotation matrix by tracking rotation and correlating it with base orientation **for** a particular rotation, **do** 

//use the DfAM support algorithm to determine facets requiring support

Determine the support needing facet for the part model for a particular rotation

//protrude support needing facet sections to the substrate

Project points from each facet to the substrate

Generate a contour boundary for the part model on the substrate

Discretize the boundary into an equidistance section for support

for each section, do

Divide the section into several points

# end for

#### //generate rectangular support from the contour boundary of the model

**if** rectangular\_support = TRUE **do** 

for each point, do

Project each point to the part model

Generate a cube structure from the projection

//set up the grid boundary for the support structures

Discretize the cube structure into vertex and faces

## end for

//generate cylindrical support from the contour boundary of the model

**else if** cylindrical\_support = TRUE **do** 

for each point, do

Project each point to the part model

Generate a cylindrical structure from the projection

//set up the grid boundary for the support structures

Discretize the cylindrical structure into vertex and faces

## end for

# end if

//formulate a material file to store material properties of the part model

Define a material file (MTL) and Ka, Ks, Kd, d variables

for all vertex points, do

//define distinct color render for support, part, and support

**if** vertex = support\_structure(vertex) **do** 

Set point  $K_a$ 's  $\leftarrow$  RGB (1 0 0)

else if vertex = substrate(vertex) do

Set point  $K_a$ 's  $\leftarrow$  RGB (0 0 1)

else if vertex = S(vertex) do

Set point  $K_a$ 's  $\leftarrow$  RGB (0 1 0)

//set up part transparency

Set point dissolve factor,  $d \leftarrow 0.7$ 

end if

# end for

//exporting part model to visualize in VR

 $S' \leftarrow$  Export OBJ and MTL files for the part model

end for

# end for



*Figure 15: a) Input sample model (taken as a reference) b) Visualized output file for the part showing part, cube supports, and substrate in VR c) Visualized output file for the part showing part, cylindrical supports, and substrate in VR* 

## **3.2 Thermal Stress Simulation**

Laser or Electron Beam 3D printing can generate significant residual stress within the object in metal AM processes. The part's poor stress and heat management results in the attainment of these residual stresses. The suggested techniques of residual stress management can cause thermal distortion of part components, creating warping and damaging the layer-by-layer construction process. Additionally, residual thermal stress can lead to improper fabrication of intricate part geometries, affecting the quality of the final 3D model [41], [42].

For the simulation of thermal deformation, an LPBF process is simulated in ANSYS simulation based on the work of Peter et al. [43]. The flowchart in Figure 16: Flowchart for the visualization methodology for stress simulation represents the performed method for the model's simulation and the corresponding visualization methodology for thermal stress.



Figure 16: Flowchart for the visualization methodology for stress simulation

The simulation model consists of a transient and static structural study for the part model, supports, and substrate— and in the final stage, the simulation exports variations in temperature and deformation into text formats. Initially, a part model's respective support and substrate are imported into ANSYS Workbench for processing transient and static studies. Each model undergoes a voxelization process to subdivide these sections into smaller components for easy analysis. The nature of build settings and conditions (hatch spacing, deposition thickness, time between layers) are configured to provide pre-processing information on the simulation. The thermal stress simulation is conducted after defining the mesh and boundary conditions. ANSYS Workbench also exports the deformed model into the VR environment.

In the visualization process, the algorithm generates a rendering of the deformed structure model based on data from the simulation result. The technique for visualizing simulation data is similar to the algorithm for visualizing sharp corners. However, the data for the deformed model is present as voxelized nodes, while the rendering is generated based on part vertices and facets. The closest neighbor algorithm is implemented before the visualization algorithm to correlate the information from voxelized nodes to part facets. The nearest neighbor algorithm determines the facets relative to the mesh nodes and assigns the facet values present for those mesh nodes, as shown in Figure 17.



Figure 17: Comparison between voxelized and faceted part model

After converting voxelized simulation data into faceted part model data, the geometry of the deformed part model gets passed through the VTK rendering framework. The rendering color spectrum parameters are determined through a point mapper, which renders the part. The render overlay occurs on the deformed model, exporting a GL transitional 3D model for the immersive virtual environment.

## **3.3 Immersive VR Environment**

Generating a virtual environment provides users with a space to navigate and utilize ImVR tools and visualization techniques. This generation process comprises three key elements: establishing a system architecture, formulating operational procedures, and creating a collaborative workspace. Figure 18 illustrates the schematic representation of the ImVR virtual environment generation process.



Figure 18: Schematic Representation of component for ImVR Virtual Environment

## 3.3.1 System Design and Architecture

In crafting an immersive environment, it is essential to establish a clear requirement for precise coordination among hardware, software, networking, scene management, and VR audio and visual performance. Lee and Yoo [44] and He et al. [45] stated the significance of meticulous architectural planning and design throughout the VR environment development process.

# 3.3.1.1 Hardware Selection

For the hardware front for the immersive visualization, the HTC Vive Pro is chosen as the HMD system for the testing environment. The hardware can capture user movements, track necessary positional data, and provide sensory feedback throughout the visual environment. In addition to the HMD, two VR hand controllers play a pivotal role in determining the orientation and position of the user's hands, interpreting hand gestures, and processing input information through the buttons in the controller. These controllers and the HMD functions enable the operation of the visualization tools for the VR model, allowing visualization of the immersive environment.

### 3.3.1.2 Software Selection

The selected software game engine for the immersive environment is *Unity*. *Unity* Game Engine, known for its user-friendly and helpful features, plays a vital role in crafting the architecture of the immersive visualization environment. *Unity* offers extensive tools, libraries, and plugins tailored for VR development, empowering different immersive tools to generate a rich, immersive experience. Furthermore, the engine supports C# scripting, allowing users to precisely control various aspects of the VR environment. The immersive system ensures a smooth transition between different VR environments through a robust scene management system. In addition, the immersive environment focuses on integrating audio spatialization techniques to transfer sound that corresponds to facilitate communication across the VR system.

## **3.3.2 Workflow and Process Operation**

Theokisto and Fairen [46] emphasized the need for three key operation interfaces for proper process flow in a virtual environment. The operational workflow consists of three essential primary interfaces: Virtual Object Generation, Graphics Processing and Rendering, and User Interaction (UI) model, as showcased in Figure 19.



*Figure 19: Operational workflow for the virtual environment* 

# 3.3.2.1 Graphics Processing and Rendering

The ImVR environment utilizes *Unity's* robust rendering pipeline, which facilitates the generation of realistic lighting, textures, and special effects across the VR model. The data frame for the immersive visualization tool is optimized at 120 frames per second (fps) to maintain consistency and prevent motion sickness during VR operation and interaction. The transitional effect is added to the canvas for changing the menu bar and action button. As the immersive visualization tool renders geometries in different file formats, libraries, namely Parabox-stl, GLTF master, OBJ-import, and Text Mesh Pro, are integrated to visualize these part models. These toolkits are essential to render and visualize parts properly.

## 3.3.2.2 UI System

Users perform actions and operations in the immersive environment by interacting through a UI system. The UI system presents different dropdown menus in the HMD through the head user displays (HUDs). In addition, The immersive environment's HUD system comprises an action button, toggle boxes, sliders, text output fields, and images.

In terms of interaction with the UI, the XR Origin Toolkit, XR Interaction Toolkit, and the Vive Unity Plugin are integrated to facilitate the operation of the VR environment effortlessly. The immersive visualization system introduces a gravity and physics-based collision tool to enhance the platform's realism. The immersive visualization model uses mesh colliders, object interactors, and a character movement system inside the *Unity* Game Engine to provide authenticity. In the immersive environment, a locomotion system enables users to move within the virtual realm using a stick pad button. In addition to the stick pad button, users can use the action-based system that moves the character based on head movements and rotation of the HMD. These systems allow users to immerse themselves deeper into the part model through the to-and-fro motion of the HMD. The controllers are programmed to deliver operational functions to interact with the UI. The controllers emit a beam of light to project the direction of hand movement. The beam is subsequently employed through a virtual ray line intersection algorithm to identify the project of the beam on the UI. The controllers have trigger and grab buttons for interacting with the UI.

#### 3.3.2.3 Virtual Object Generation

In the immersive environment, the integration of virtual objects plays a pivotal role in creating a lifelike 3D model. These virtual objects are static models, allowing higher realism in the VR space. These stationary models are generated and rendered through *Siemens NX* software. While users can explore these models closely, they cannot directly interact with or manipulate them.

In the virtual environment, the user experience is further enhanced by adding avatars representing their unique selves. These avatars default to only having a head and hands to minimize latency. As shown in Figure 20, the avatars possess mouths in their heads for effective communication and movement during collaborative VR activities. Normcore plugin

acts as a default source for these avatars in a virtual environment, providing multi-user avatar functionality required for collaborative VR. Users can only maneuver the part model in the VR space and not each other, as the main focus of the toolkit centers on the development of intricate interaction between part models and users. Thus, the primary emphasis is visualizing part models, their geometry, and intricate details.



Figure 20: Default Avatar presence for ImVR environment

## 3.3.3 Collaborative VR Environment

Collaborative VR enables multiple users from different locations to interact in the same virtual environment, fostering real-time virtual interaction and shared VR experiences [45]–[47]. In the immersive virtual environment, having multiple collaborators work on a particular part model is valuable as each collaborator may analyze a design part differently. The collaborative virtual VR tool also allows users to engage and cooperate as if physically in the same space.

Steam-VR plugin and Normcore networking system have been added to the virtual reality system to facilitate the multiplayer system in VR. *Unity* allows direct installation of the normcore package through its internal game engine. Normcore enables the users to host the immersive visualization environment in their internal cloud, facilitating synchronous and remote networking systems. Normcore also facilitates a back-and-forth audio feed between users to communicate through a voice-over-internet protocol (VoIP). It reduces the latency between multiple users in the virtual space. The process reduces latency between various users in the virtual room by generating simplistic avatar models for the collaborative VR model,

representing each user in the virtual environment. Figure 21 shows the methodology communication and collaborative space inside the ImVR environment.



Figure 21: Methodology of collaboration inside the ImVR environment

#### **3.4 Voice Recognition and Translation Model**

In the ImVR toolkit's immersive environment, it is essential to establish effective communication channels between users and the chatbot to provide user assistance when required. To activate and pass information to the AI bot, it is necessary to predefine specific trigger keywords for operating and initiating the chatbot system. The activation of the chatbot should occur distinctly through an environmental change, indicating the bot's readiness to communicate with the user. The initial recognition algorithm leverages the built-in *Unity* Engine's speech detection module, where the keyword detection code is formulated in C# within *Unity*.

While the voice recognition module initiates the ChatGPT bot, there is a need to guarantee a seamless word-by-word translation between the user and the immersive environment. In addition, the voice bot must accurately capture all words and preserve semantic information; thus, relying solely on *Unity's* voice recognition capabilities is insufficient to operate the system. The voice translation algorithm used in the immersive visualization platform is a

modification of the algorithm by Radford et al. [48]. The algorithm uses the Whisper AI model and translates the data through a Python wrapper for facilitating incorporation into the *Unity* game engine. The information from the adapted Whisper model is transferred into the chatbot for further processing. In this process, the adapted whisper-based model converts voice recognition into input text for the chatbot.

Algorithm 6 mentions the voice recognition and translation model that activates the artificial intelligence (AI) assistant bot to provide support in the immersive environment. Here, A denotes the spatial audio input into the ImVR environment, and O represents the output text from the voice translation system.

Algorithm 6: Real-time voice recognition and translation system
Input: A ← spatial audio input from the user
Output: O ← output text from voice translation system
Initialization: Attributes {Keyword = "Hey Immersive Bot"}
for each audio input in the ImVR environment, do

//assign keywords to mean certain actions in VR

dictionary  $\leftarrow$  [keyword; action]

for each argument in the audio input, do

//perform predefined actions based on defined arguments

if argument = dictionary, do

Perform assigned action and activate the Whisper model

## end if

//converting voice to text for passing information to the ChatGPT assistant

Record input data from the user and pass it to the Whisper translator model

 $O \leftarrow$  export translated model into text and pass it to AI assistant

end for

# end for

#### 3.5 ChatGPT Assistant AI Bot

An integrated assistive AI bot in the *Unity* environment actively assists users with the tools present in the ImVR toolkit and environment. The AI bot also would provide users with a broader reference of AM, especially with DfAM, the impacts of different visualized parameters on design, and offer preliminary design solutions to address these issues. The assistive AI bot uses the broad knowledge of ChatGPT through the OpenAI API framework. However, without proper training and prompt engineering, the chatbot does not have enough information to guide users in the immersive environment. The immersive chatbot must be trained, especially with relevant information in the field of AM, to assist users in the ImVR environment. It is then integrated with the voice interface to allow users to operate the Chatbot in VR. Figure 22 presents a schematic flowchart of the training and operation method used to manage the ImVR environment's assistive Chatbot.



Figure 22: Overview of the operation of Whisper AI and ChatGPT bot in ImVR

# **3.5.1 Training Process**

Many documents related to AM and DfAM are fed into the training data model to train ChatGPT on AM-specific parameters and datasets. ChatGPT 3.5-Turbo has primarily been used to train these data sets as it can retrieve data quickly and is cost-effective for preparing large data sets. The training process includes splitting the large data sets into smaller data information packets. The data requires normalization and standardization with proper vector representation to allow ChatGPT's API to train data. Thus, the data sets are embedded through OpenAI embeddings to represent data in a standard format during training. This data set acts as an additional knowledge base for ChatGPT and helps the assistive AI during the query search process. It assists in providing better context and also helps ChatGPT answer the query better.

The training process for the prompt uses a few-shot training approach to enhance ChatGPT's ability to respond effectively to user queries. It involves presenting ChatGPT with a few example inputs and corresponding desired outputs, allowing the model to learn and generalize from these examples to improve its prompt responses. These prompts are in the form of questions and their respective answers. Good, bad, and even intermediate responses fit the training process for allowing ChatGPT to fine-tune outputs in a better method.

## 3.5.2 Need and Functionality of AI Bot

The users can use the trained ChatGPT Bot for operational help and further assistance in the ImVR environment. The application of an assistive AI bot helps the users understand more about the immersive environment. The bot allows for aid regarding the toolkit and helps users formulate a better framework of the ImVR environment. Figure 23 showcases scenarios where ChatGPT would help users in the ImVR environment. The assistance AI bot's primary functionality is to help users in the virtual environment with design-related issues. Users could use the AI bot to find preliminary solutions and also reflect on results present through immersive visualization.

# Assistive AI ChatGPT Bot in VR Environment

| Acts as a<br>communicative<br>chatbot   | Help users understand<br>about DfAM tool and<br>boarder design<br>framework  | Familiarize users with<br>immersive virtual<br>environment  | Assist in providing<br>technical context and<br>feedback  | Provide alternative<br>design changes and<br>solutions   |
|---|--|---|---|--|
| <u>Scenario 1</u> :   | <u>Scenario 2</u> :  | <u>Scenario 3</u> :   | <u>Scenario 4</u> :   | <u>Scenario 5</u> :  |
| User – Hi! Immersive bot<br>could you help me today?<br>AI Assistant – Sure, tell me<br>what do you want to know<br>about this toolkit or<br>environment. | User – What is a thin-to-thick<br>transition?<br>Al Assistant – "Thin-to-thick<br>transition" refers to a<br>situation where the part that<br>is 3D printed transitions from<br>a relatively thin section to a<br>thicker section. It can present<br>challenges in the printing<br>process and may require<br>special considerations to<br>ensure successful production. | User – What would trigger<br>button do in this ImVR<br>environment?<br>Al Assistant – A "trigger<br>button" typically refers to a<br>physical button or a digital<br>control input on a VR<br>controller. Here, the trigger<br>button is used to pick up,<br>grab, or interact with virtual<br>objects. | User – My part model has<br>large number of small<br>openings, is it bad?<br>Al Assistant – In AM process<br>having large number of small<br>opening may cause design to<br>lose structural integrity and<br>may have trouble during<br>printing due to laser size. It<br>could also effect the heat flow<br>if your designing a HX model.<br>Tell me more. | User – What should I do with<br>the high small opening part?<br>AI Assistant – One method is<br>that you could adjust the<br>orientation of the part during<br>printing to minimize the<br>impact of the openings. This<br>can help reduce the need for<br>support structures and<br>improve the overall print<br>quality. |

Figure 23: Need and application of ImVR bot in different scenarios

# **3.5.3 Operation Process**

The AI assistant receives information sent by the Whisper model in the virtual environment and stores the information in data sets. The query data are embedded into certain standardized and normalized data sets. The embedded input data set passes itself into a semantic data search process that allows the data set to find similar words related to trained data prompts. The data passes into the trained ChatGPT, where a contextual prediction technique enables the bot to predict the result better. The model then results in an answer to the query and feeds it into the UI canvas text of the ImVR environment.

Algorithm 7 presents the AI assistant model's training process and operational method. The algorithm specifies the applied prompt training methodology, communication protocol, and semantic data search process. In the algorithm, *O* denotes the output from the Whisper model that users record through the virtual environment, and *O*\* presents the response of the AI result as UI text in the virtual environment.

## Algorithm 7: Prompt training and operation of chatbot

**Input:** *O* ← output from the Whisper Model

**Output:**  $O^* \leftarrow$  response from AI model

Initialization: Attributes {UI, Input Training Data, Prompt Information, AM Documents}

if whisper model = "true" do

*//activate custom training for the ChatGPT bot once the voice command is activated* 

API key ← Open AI API access key

//inputting information from text documents and books

Prompt reader ← read (AM Documents)

//dividing and embedding information into small chunks for training

Split prompt reader into text information sections based on ChatGPT's token size

for each text information section, do

Embed text information using OpenAI embeddings

## end for

#### *//perform prompt training by providing a few-shot training methodology*

Establish prompt training methodology as role: {"AI assistant":: "Case 1: Prompt 1": "Expected Output 1" "Case 1: Prompt 2": "Expected Output 2"::::}

Develop a knowledge base by providing semantic information alongside text

//activate and pass query from user to ChatGPT bot

Initiate two-way communication between the ImVR environment and ChatGPT

Insert query from the user to the AI assistant model

//make sure the token limit doesn't exceed the max token size to prevent overfitting

Formulate a lower token limit and use the GPT Turbo model

if the user requests information, do

#### //searching keywords in trained prompt for better prompt response

Embed user information into query data and perform semantic data search

Correlate user request to trained ChatGPT

//emphasize trained model output over the untrained model output

Perform semantic prediction based on semantic data search

# end if

## //export result of trained ChatGPT model to VR environment

 $O^* \leftarrow$  text from trained ChatGPT model

## end if

#### 4. Case Study and Results

Emphasizing the practical application of the ImVR toolkit involves conducting a comprehensive demonstration of the tool to analyze distinct case studies. The primary objective of these case studies is to showcase the advantages and capabilities of the ImVR toolkit in the context of visualizing intricate organic geometries within the ImVR environment. A lattice-design equipped wrench model and a topology-optimized organic bracket model are selected to showcase the power of these algorithms and are presented in Figure 24. The integration demonstrates the benefits and abilities of the ImVR toolkit to navigate the VR environment, handle visualization of complex geometries, and help analyze the part geometry better. Figure 25 presents images of the immersive visualization tool and environment developed for the ImVR toolkit.



Figure 24: Case study models in ImVR a) Lattice Wrench Model b) Topopt Bracket



Figure 25: a) Immersive environment for representation of part models b) Generated CAD models to mimic real environment for ImVR tool c) Researcher analyzing the VR model through HTC Vive Pro d) Point-of-view (POV) of the menu bar, toolbar, and ImVR environment

#### 4.1 Lattice Wrench Model

#### 4.1.1 Sharp Corners and Vertices

The complex Lattice Wrench design is typical of a part model that needs to be fabricated using metal 3D printing. Observing, understanding, and evaluating the DfAM parameters is critical to the quality of the 3D printed. Sharp corners in the part design present a problem in 3D printing that must be addressed. The parameters related to Gaussian curvature and its value across the part model must be determined and visualized to visualize sharp corners efficiently. These values help create a 3D rendering of a part model, with each vertex showing an associated Gaussian curvature value through color. The model depicted in Figure 26 showcases diverging color plots, highlighting the varying Gaussian curvature across the part model. The sudden change of color across the model in Figure 26 shows the sharp change of Gaussian curvature, thus creating a sharp corner point for the part model. These points contribute to higher thermal and structural stress and buildability problems. Figure 26 also includes an immersive perspective, featuring the generation of spheres to highlight the sharp point vertices.



Figure 26: a) Rendered 3D color plot generated on variation of Gaussian curvature b) Generated sphere structure across the part model for visualization of sharp corners on points with higher Gaussian curvature value

# 4.1.2 Small Opening and Thin Sections

Visualizing small openings and thin sections in the Lattice Wrench model ensures accurate design and printing of intricate lattice structures. A clear visualization allows users to design and analyze lattice geometry that helps to reduce printing errors and material usage. The critical points and areas on small openings and thin sections are visualized for the Lattice Wrench model in Figure 27 by generating 3D spheres and line contours. The spheres represent small openings/thin sections across those vertices, and the generated lines represent the start and end of those features. Furthermore, a sectional plane is overlayed to enhance the clarity of

visualized sections and openings. Also, the ImVR toolkit allows users to show/hide the generated points, lines, and planes for the part model.



Figure 27: a) Visualization of critical thin points with the ability to show and hide sectional lines across part models for ImVR tools for thin section b) Rendered 3D line plot to show lines/area of critical region based on small openings

# 4.1.3 Thin-to-thick Transition

Visualizing a thin-to-thick transitional area is crucial for a lattice structure as it helps identify regions where the slice contour shifts from a thinner area to a thicker area. In the design process for a Lattice Wrench model, it's crucial to ensure a uniform distribution of lattice structures when incorporating them into a CAD design, as improper distribution creates many thin-tothick transitional areas in the part model. In the visualization process, sectional contours for a part model help emphasize these transitional sections. Three contours (thick transitional area, thin transitional area, and sectional transitional area) are presented through a toggle box to visualize these transitional areas. As in Figure 28, these boxes allow users to hide part models and capture contour changes across different sections for showing thin-to-thick transitions.



Figure 28: Visualization of thin-to-thick transition sections across the part model

## 4.1.4 Cusp Error and Slicing Mechanism

The part model's slicing mechanism is formulated by generating different colored contours across the part model perpendicular to the build direction. These contours represent the sliced object and allow users to visualize the 3D printing process slice-by-slice. In addition to displaying the sliced contour, the ImVR tool enables users to visualize and observe the impact of cusp errors or stair casing effects during the AM process. These lead to immersion for visualization of staircase error for the part model, contributing to cusp error. Figure 29 represents the generated visualization model for visualizing cusp error and slicing mechanism.



Figure 29: Visualization of staircase model with section showing cusp error across the part model

# 4.1.5 Support Structures

Visualizing the support structure for a Lattice Wrench model helps users assess the need and placement of support within the model's intricate lattice framework. A proper support structural design ensures that the model can be successfully 3D printed with minimal failures or distortions, maintaining its structural integrity. The visualization is particularly vital for lattice structures, which rely on their complex lattice structures for lightweight and structural performance, making precise support placement integral for achieving desired mechanical properties. The ImVR toolkit allows users to visualize and generate support structures in real-time while changing the part model orientation in VR. Users can freely rotate the part along the x-axis, y-axis, and z-axis and visualize the change of support structures across each axis after the rotation of the part model. Figure 30 illustrates the same phenomenon, showcasing users' generation of cube and cylindrical supports before and after rotating the part model. These buttons and slider bars allow users to visualize the change of support structures in real-time.



Generated cube supports across no rotation



Generated cylindrical supports across no rotation



Go Back

enerate Cube Suppo Current Orientation

der (Current)

(a)

Rotate Part Model

Visualize Support Structures

Rotate about X

Rotate about \

Rotate about Z

Generated real-time cube supports across part model for desired rotation

Figure 30: Visualization of support structures across the part model a) Generated cube and cylindrical support on the part model at current orientation b) Real-time generation of cube supports when the part model is rotated by 40 degrees across

the X-axis

### 4.1.6 Thermal Stress Simulation

Visualizing thermal stress in a Lattice Wrench helps users identify potential heat accumulation and dissipation points, especially across lattice points within the structure. The stress simulation data lets users visualize vulnerabilities throughout the lattice model and showcase possible failure points under high-temperature conditions. The toolkit also showcases a voxelized model that serves as a mesh for the part model, enabling FEA analysis on the part model. Figure 31 represents the variation of the generated voxelized and rendered models for ImVR. In addition, ANSYS Workbench plots and renders the thermal stress to visualize the stress change across the part model. The simulation results in CSV data files where nodes have values across the part model. Figure 31 shows the generated render plots from ANSYS and their corresponding correlated value. In addition, the figure also has developed render plots for visualization of these values for the ImVR environment. These FEA model renderings are generated using VTK algorithms, and they present a diverging color palette emphasizing areas of higher stresses and associated issues in the part model.



Figure 31: Variation of visualization plots generated across FEA software and ImVR toolkit for Lattice Wrench Model

#### 4.2 Topology-optimized Bracket Model

# 4.2.1 Sharp Corners

In recent years, the generation of organic geometries through AM has emerged in the limelight with topology optimization. A sample bracket model depicts these organic geometries as a sample case study. The model must be printed through the AM process, and the evaluation of these geometries help evaluate possible design weakness in the part model. These design defect may lead to structural weakness during 3D printing and thus needs to be remedied. The design defect may be in a sharp corner, a thin section, or a thin-to-thick transition area. To evaluate the presence of sharp corners, Gaussian curvature values and the sharp corners' vertices are displayed in the same manner as presented for the Lattice Wrench part model. Figure 32 represents the rendered 3D plots and sharp corner points for the Topopt Bracket model. In this case study, the change of color contour can be seen to coincide with the design defect where there is a small abnormal cut across the part geometry. Visualizing sharp corners helped identify the design defect that resulted in the sharp points across those areas.



Figure 32: Sharp Corner Visualization a) Gaussian curvature across a sharp point (CAD defect) across the part model b) Presence of sharp corners across those points noted during visualization

# 4.2.2 Small Opening and Thin Sections

In topology optimization, geometries may be over-constrained, leading to sections that possess excessive issues during 3D printing. These defects may occur as a small opening or a thin section. A precise design analysis helps users visually inspect before 3D printing, reducing potential errors. For this, in Figure 33, critical points for the part model for small openings and thin sections are visualized through spheres. These rendered texture OBJ models allow users to generate lines signifying the start and end points. In addition, the toolkit will enable users to show/hide the points and lines for the part model, improving users' ability to visualize objects.



Figure 33: a) Visualization of small opening points across the part model b) Visualization of small opening sections across the part model

# 4.2.3 Thin-to-thick Transition

In an organic topology optimized-based design, numerous thin to thicker transitions might be present in the part geometries. These transitions are current areas of potential structural and thermal weakness. By clearly visualizing these transition areas in ImVR, designers can refine the design and optimize the design to ensure proper distribution of material across the part. Thus, Figure 34 shows the sectional contours across the part model to visualize the thin-to-thick transition across the Topopt Bracket model. The toolkit facilitates visualization through three toggle boxes, and users can hide and unhide respective thin and thick sections. Additionally, users can observe how the transition occurs by embedding themselves in the part design.



Visualization of current and project slice hiding thick regions



Zoomed in Representation

Figure 34: Visualization of thin to thick transitional areas for the part model in the ImVR environment

# 4.2.4 Cusp Error and Slicing Mechanism

An organically optimized topology may encounter a significant staircase effect in 3D printing, as the generated organic contour might not be a regular polygonal shape. It is essential to visually inspect these regions to ensure they fall within the specified threshold. The visualization of the slicing mechanism and cusp error across the Topopt Bracket model can be seen in Figure 35. The figure represents the variation of the slicing mechanism across the part model, and users can hide/unhide sections of the part model to enable better visualization of the cusp error. In addition, users can see a distinct effect of the staircase model effect where slice thickness is reduced.





Immersed view on top view with hidden bottom contours Cusp Error across the sliced model

Figure 35: Visualization of cusp error across slices for a part model

# 4.2.5 Support Structures

A Topopt Bracket designed through a topological optimization process may necessitate a substantial quantity of support structures. Visualizing these structures enables users to assess whether the resultant intrinsic features will be generated in the final 3D print. The generation of support structures for the Topopt Bracket model can be seen in Figure 36, where users can create cube and cylindrical supports for the Topopt Bracket model. In addition, the user the rotation of the part model across all three axes and the associated change in the generation of supports can be seen in the figure. The sliders and buttons allow users to instantly change and render the part model and support.



Generated cube supports across no rotation



Generated cylindrical supports across no rotation



Generated real-time cylindrical supports across pa model for set rotation

Figure 36: Visualization of support structures for a part model a) Generated support with no rotation b) Generate support structures with rotation of part model by user



#### 4.2.6 Thermal Stress Simulation

A Topopt Bracket may possess an area where excessive heat would build up during the 3D printing. By pinpointing areas of extreme heat built up in the part model, users can refine the design, ensuring that the AM process produces parts with minimal heat stress-related defects. The voxelized model and the 3D rendered stress simulation model of the Topopt Bracket part are displayed in Figure 37. The figure shows the variation of the voxelized model and the rendered part geometry through FEA software and the 3D visualization algorithm. The visualized renderings generated demonstrate the ability of the developed VTK algorithm to import FEA data into the ImVR environment directly.



Figure 37: Comparison of generated results from ANSYS Workbench and VTK rendering algorithm for Topopt Bracket

#### 4.3 Collaborative VR Environment

Users can invite and talk to another collaborator through the ImVR toolkit. Each user has an inbuilt custom avatar based on Normcore's default avatar model, allowing users to see, communicate, and interact with each other and the ImVR environment. Figure 38 shows a

synchronized scene where users can interact and talk to each other through the built-in microphone system of the headset. The invite collaborator button sends a mail to the possible collaborator, and the collaborator can join the present scheme through the shared application key interface. In addition, both users are granted equal ability to control and manipulate part models in the ImVR environment.



Figure 38: Synchronized collaborative immersive VR environment

# 4.4. Assistive ChatGPT Chatbot

The ImVR tool and environment have an assistive AI chatbot in the virtual system. Users can perform numerous VR operations through the assistive chatbot. For example, the command *"Take me home"* defaults the user position back to the start point, while the *"Invite Collaborator"* invites collaborators to analyze a part in the immersive environment. In addition, users can activate the chatbot through the phrase *"Hey, immersive bot"* or by clicking the activate button. The trigger button changes the application scene, where users can start and stop recording their voice. The recorded voice is translated into text through whisper AI built-in translation. The UI displays the translated text in the canvas after the user has communicated

with the chatbot. Users can transfer the information to the AI assistant through the button, which passes information to the custom-trained ChatGPT model.

The custom-trained ChatGPT model can assist the designers in different scenarios, as presented in the methodology section of the assistive AI bot. The ability of assistive AI to help users is shown through two possible scenarios in Figure 39 and Figure 40. Figure 37 shows a scenario where the user asks the assistant bot for help regarding additive manufacturing parameters. It illustrates a situation in which the user sought assistance from the chatbot to comprehend the parameter displayed by the ImVR toolkit related to the thin-to-thick transitional threshold. In this, the information presented in the output canvas helps to form a descriptive communication source between ChatGPT and users, helping them understand these complex terminologies. The second scenario shown in Figure 40 presents a problem where the user has informed ChatGPT that the analyzed part model has more sharp corners and is asking for recommendations to solve these issues. The assistant AI considered the user input scenario and presented a solution for the design problem specific to that part model. It shows how ChatGPTbased AI assistant could guide users and help users solve design process issues.



Figure 39: User interaction with the voice translation and trained ChatGPT Bot for understanding the DfAM toolkit



Figure 40: The interaction between the user and the trained ChatGPT Bot for help regarding a design issue
## **5.** Conclusions and Future Scope

This thesis presents a comprehensive toolkit and methodology for visualization of design parameters and simulations. This work aims to generate an immersive VR toolkit to help analyze critical DfAM-specific geometrical elements and features related to powder bed fusion additive manufacturing processes. The toolkit evaluates DfAM features such as thin sections, narrow openings, sharp corners, thin-to-thick transition, and thermal stress simulation effects based on VTK-rendered part model data. Custom algorithms identify these features and render those algorithms to formulate visual part models. The adopted visualization methodology enables users to understand the impact of each DfAM parameter and the necessary changes users need to mitigate the effects at the design stage. These changes help users evaluate designs better and allow users to save design iteration costs in the long run.

Currently, the immersive virtual environment cannot render 3D color plots representing dynamic simulation of part models. The ImVR tool presented can only display the static final results from the simulation, offering no real-time representation of how a part model would deteriorate or change with time during the simulation process. The ImVR tool could also be seamlessly integrated with GAN and topology optimization algorithms to allow users to observe and track the progression of part model evolution over time. Visualizing the change of simulation parameters over time facilitates users to understand part models critically, leading to better design choices. A pathway for future development for the ImVR tool would involve allowing users to generate VR renders in real-time to give users control and will enable them to set limits on input variables and data structures. This enhanced flexibility with the immersive environment for 3D model visualization would facilitate collaborative efforts, enabling multiple users to generate various visualizations and test what-if scenarios. It would allow appropriate extraction of part model/process information from the immersive environment.

Further, it would help provide multiple collaborators the ability to create unique visualizations and extrapolate information from the immersive environment much more quickly.

Further, the potential of the ImVR toolkit can be extended beyond the visualization of the additive manufacturing process, as it can be seamlessly implemented with traditional subtractive manufacturing processes such as machining, casting, metal forming and joining processes. Incorporating Design for Manufacturing (DfM) and Design for Manufacturing Assembly (DfMA) tools in the ImVR toolkit would help assess and optimize geometrical parameters critical to traditional manufacturing processes. The extension of this framework would help manufacturers visualize and analyze subtractive manufacturing design rules and associated parameters such as thread clearance, internal planar grooves, and tube stock. It would facilitate informed decision-making and further enhance the efficiency of the subtractive manufacturing process.

We could also enhance the ImVR tool's capability inside the VR environment by implementing a custom keywords-based Natural Language Processing (NLP) System. The ChatGPT integration helps users familiarize themselves with the VR environment and the DfAM tools, but users must perform all the UI actions themselves. We could adapt a custom NLP to enable users to interact inside the VR environment more intuitively. In such a scenario, users will be able to issue verbal commands instead of clicking UI buttons to help manipulate, navigate, and query the visualization data with ease inside VR environments. This addition to the ImVR toolkit would significantly improve the user's overall experience and streamline the analysis of simulation parameters, enabling a more dynamic, user-friendly interaction inside the ImVR toolkit.

## References

- J. M. Zheng, K. W. Chan, and I. Gibson, "Virtual Reality: A real world review on a somewhat touchy subject," *IEEE*, pp. 20–23, 1998.
- [2] T. Mazuryk and M. Gervautz, *Virtual Reality History, Applications, Technology and Future*. 1999.
- M. T. Schultheis and A. A. Rizzo, "The application of virtual reality technology in rehabilitation," *Rehabil. Psychol.*, vol. 46, no. 3, pp. 296–311, 2001, doi: 10.1037/0090-5550.46.3.296.
- [4] J. T. Bell, H. S. Fogler, and A. Arbor, "The Investigation and Application of Virtual Reality as an Educational Tool," *Proc. Am. Soc. Eng. Educ.*, no. 2513, pp. 1–11, 1995, [Online]. Available: https://www.researchgate.net/profile/Hs\_Fogler/publication/247920944\_The\_Investiga tion\_and\_Application\_of\_Virtual\_Reality\_as\_an\_Educational\_Tool/links/55f721fb08a e07629dbfcfee.pdf
- [5] J. Adolf, P. Kán, B. Outram, H. Kaufmann, J. Doležal, and L. Lhotská, "Juggling in VR: Advantages of immersive virtual reality in juggling learning," *Proc. ACM Symp. Virtual Real. Softw. Technol. VRST*, no. April 2020, 2019, doi: 10.1145/3359996.3364246.
- [6] M. Fiorentino, A. E. Uva, M. Gattullo, S. Debernardis, and G. Monno, "Augmented reality on large screen for interactive maintenance instructions," *Comput. Ind.*, vol. 65, no. 2, pp. 270–278, 2014, doi: 10.1016/j.compind.2013.11.004.
- [7] G. Wheeler *et al.*, "Virtual interaction and visualisation of 3D medical imaging data with VTK and unity," *Healthc. Technol. Lett.*, vol. 5, no. 5, pp. 148–153, 2018, doi: 10.1049/htl.2018.5064.

- [8] S. Jaeger *et al.*, "Challenges for brain data analysis in VR environments," *IEEE Pacific Vis. Symp.*, vol. 2019-April, pp. 42–46, 2019, doi: 10.1109/PacificVis.2019.00013.
- [9] C. Ware and G. Franck, "Viewing a graph in a virtual reality display is three times as good as a 2D diagram," *IEEE Symp. Vis. Lang. Proc.*, pp. 182–183, 1994, doi: 10.1109/vl.1994.363621.
- [10] S. Mandal, "Brief Introduction of Virtual Reality & its Challenges," Int. J. Sci. Eng. Res., vol. 4, no. 4, pp. 304–309, 2013, [Online]. Available: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Brief+Introduction+ of+Virtual+Reality+&+its+Challenges#0
- [11] M. Kraus *et al.*, "Assessing 2D and 3D Heatmaps for Comparative Analysis: An Empirical Study," *Conf. Hum. Factors Comput. Syst. Proc.*, pp. 1–14, 2020, doi: 10.1145/3313831.3376675.
- [12] L. P. Berg and J. M. Vance, "An Industry Case Study: Investigating Early Design Decision Making in Virtual Reality," J. Comput. Inf. Sci. Eng., vol. 17, no. 1, Mar. 2017, doi: 10.1115/1.4034267.
- S. Jayaram, H. I. Connacher, and K. W. Lyons, "Virtual assembly using virtual reality techniques," *CAD Comput. Aided Des.*, vol. 29, no. 8, pp. 575–584, 1997, doi: 10.1016/S0010-4485(96)00094-2.
- [14] C. Kwon, "Verification of the possibility and effectiveness of experiential learning using HMD-based immersive VR technologies," *Virtual Real.*, vol. 23, no. 1, pp. 101–118, 2019, doi: 10.1007/s10055-018-0364-1.
- [15] A. Syamimi, Y. Gong, and R. Liew, "VR industrial applications—A singapore perspective," *Virtual Real. Intell. Hardw.*, vol. 2, no. 5, pp. 409–420, 2020, doi:

10.1016/j.vrih.2020.06.001.

- [16] U. M. Dilberoglu, B. Gharehpapagh, U. Yaman, and M. Dolen, "The Role of Additive Manufacturing in the Era of Industry 4.0," *Procedia Manuf.*, vol. 11, no. June, pp. 545– 554, 2017, doi: 10.1016/j.promfg.2017.07.148.
- [17] B. Vayre, F. Vignat, and F. Villeneuve, "Designing for Additive Manufacturing," *Procedia CIRP*, vol. 3, pp. 632–637, 2012, doi: 10.1016/j.procir.2012.07.108.
- [18] S. S. Babu, A. H. I. Mourad, K. H. Harib, and S. Vijayavenkataraman, "Recent developments in the application of machine-learning towards accelerated predictive multiscale design and additive manufacturing," *Virtual Phys. Prototyp.*, vol. 18, no. 1, 2023, doi: 10.1080/17452759.2022.2141653.
- [19] S. Bryson, "Virtual reality in scientific visualization," *Comput. Graph.*, vol. 17, no. 6, pp. 679–685, Nov. 1993, doi: 10.1016/0097-8493(93)90117-R.
- [20] R. Daniil, G. Wohlgenannt, D. Pavlov, Y. Emelyanov, and D. Mouromtsev, "A New Tool for Linked Data Visualization and Exploration in 3D/VR Space," 2019, pp. 167–171. doi: 10.1007/978-3-030-32327-1\_33.
- [21] J. K. Ostrander, L. Ryan, S. Dhengre, C. McComb, T. W. Simpson, and N. A. Meisel, "A comparative study of virtual reality and computer-aided design to evaluate parts for additive manufacturing," *Proc. ASME Des. Eng. Tech. Conf.*, vol. 2A-2019, pp. 1–11, 2019, doi: 10.1115/DETC2019-97480.
- M. El Beheiry, S. Doutreligne, C. Caporal, C. Ostertag, M. Dahan, and J.-B. Masson,
  "Virtual Reality: Beyond Visualization," *J. Mol. Biol.*, vol. 431, no. 7, pp. 1315–1321,
  Mar. 2019, doi: 10.1016/j.jmb.2019.01.033.
- [23] N. Brunhart-Lupo, B. Bush, K. Gruchalla, K. Potter, and S. Smith, "Collaborative

Exploration of Scientific Datasets using Immersive and Statistical Visualization: Preprint," *Osti.Gov*, no. September, 2020, [Online]. Available: www.nrel.gov/publications.

- [24] A. H. Hoppe, F. van de Camp, and R. Stiefelhagen, "ShiSha," Proc. ACM Human-Computer Interact., vol. 4, no. CSCW3, pp. 1–22, 2021, doi: 10.1145/3432950.
- [25] A. Wiberg, J. Persson, and J. Ölvander, "Design for additive manufacturing a review of available design methods and software," *Rapid Prototyp. J.*, vol. 25, no. 6, pp. 1080– 1094, Jul. 2019, doi: 10.1108/RPJ-10-2018-0262.
- [26] M. McMillan, M. Leary, and M. Brandt, "Computationally efficient finite difference method for metal additive manufacturing: A reduced-order DFAM tool applied to SLM," *Mater. Des.*, vol. 132, pp. 226–243, Oct. 2017, doi: 10.1016/j.matdes.2017.06.058.
- [27] M. Kumke, H. Watschke, P. Hartogh, A. K. Bavendiek, and T. Vietor, "Methods and tools for identifying and leveraging additive manufacturing design potentials," *Int. J. Interact. Des. Manuf.*, vol. 12, no. 2, pp. 481–493, 2018, doi: 10.1007/s12008-017-0399-7.
- [28] B. Zhang, A. Goel, O. Ghalsasi, and S. Anand, "CAD-based design and pre-processing tools for additive manufacturing," *J. Manuf. Syst.*, vol. 52, no. November 2018, pp. 227–241, 2019, doi: 10.1016/j.jmsy.2019.03.005.
- [29] G. Melo, A. Ercan, M. Kolter, and J. H. Schleifenbaum, "Investigating the Use of Immersive Technologies for Additive Manufacturing," 2022, pp. 469–476. doi: 10.1007/978-3-031-16411-8\_54.
- [30] J. Jeon, S. Lee, and H. Choe, "Beyond ChatGPT: A conceptual framework and

systematic review of speech-recognition chatbots for language learning," *Comput. Educ.*, vol. 206, p. 104898, Dec. 2023, doi: 10.1016/j.compedu.2023.104898.

- [31] A. Shafeeg, I. Shazhaev, D. Mihaylov, A. Tularov, and I. Shazhaev, "Voice Assistant Integrated with Chat GPT," *Indones. J. Comput. Sci.*, vol. 12, no. 1, Feb. 2023, doi: 10.33022/ijcs.v12i1.3146.
- [32] I. Stroud and P. C. Xirouchakis, "STL and extensions," *Adv. Eng. Softw.*, vol. 31, no. 2, pp. 83–95, 2000, doi: 10.1016/S0965-9978(99)00046-0.
- [33] W. Zha and S. Anand, "Geometric approaches to input file modification for part quality improvement in additive manufacturing," *J. Manuf. Process.*, vol. 20, pp. 465–477, 2015, doi: 10.1016/j.jmapro.2015.06.021.
- [34] S. Chowdhury, K. Mhapsekar, and S. Anand, "Part Build Orientation Optimization and Neural Network-Based Geometry Compensation for Additive Manufacturing Process," *J. Manuf. Sci. Eng.*, vol. 140, no. 3, Mar. 2018, doi: 10.1115/1.4038293.
- [35] A. Seharing, A. H. Azman, and S. Abdullah, "A review on integration of lightweight gradient lattice structures in additive manufacturing parts," *Adv. Mech. Eng.*, vol. 12, no. 6, p. 168781402091695, Jun. 2020, doi: 10.1177/1687814020916951.
- [36] C. Klahn, B. Leutenecker, and M. Meboldt, "Design for Additive Manufacturing Supporting the Substitution of Components in Series Products," *Procedia CIRP*, vol. 21, pp. 138–143, 2014, doi: 10.1016/j.procir.2014.03.145.
- [37] G. A. O. Adam and D. Zimmer, "Design for Additive Manufacturing—Element transitions and aggregated structures," *CIRP J. Manuf. Sci. Technol.*, vol. 7, no. 1, pp. 20–28, Jan. 2014, doi: 10.1016/j.cirpj.2013.10.001.
- [38] K. Meena and S. Singamneni, "A new auxetic structure with significantly reduced stress

concentration effects," *Mater. Des.*, vol. 173, p. 107779, Jul. 2019, doi: 10.1016/j.matdes.2019.107779.

- [39] Richard J. Lisle, "Detection of Zones of Abnormal Strains in Structures Using Gaussian Curvature Analysis," *Am. Assoc. Pet. Geol. Bull.*, vol. 78, 1994, doi: 10.1306/A25FF305-171B-11D7-8645000102C1865D.
- [40] C. C. Seepersad, T. Govett, K. Kim, M. Lundin, and D. Pinero, "A designer's guide for dimensioning and tolerancing SLS parts," 23rd Annu. Int. Solid Free. Fabr. Symp. - An Addit. Manuf. Conf. SFF 2012, pp. 921–931, 2012.
- [41] A. J. Dunbar, E. R. Denlinger, M. F. Gouge, and P. Michaleris, "Experimental validation of finite element modeling for laser powder bed fusion deformation," *Addit. Manuf.*, vol. 12, pp. 108–120, Oct. 2016, doi: 10.1016/j.addma.2016.08.003.
- [42] E. R. Denlinger, M. Gouge, J. Irwin, and P. Michaleris, "Thermomechanical model development and in situ experimental validation of the Laser Powder-Bed Fusion process," *Addit. Manuf.*, vol. 16, pp. 73–80, Aug. 2017, doi: 10.1016/j.addma.2017.05.001.
- [43] N. Peter, Z. Pitts, S. Thompson, and A. Saharan, "Benchmarking build simulation software for laser powder bed fusion of metals," *Addit. Manuf.*, vol. 36, p. 101531, Dec. 2020, doi: 10.1016/j.addma.2020.101531.
- [44] Y. Lee and B. Yoo, "XR collaboration beyond virtual reality: Work in the real world,"
  *J. Comput. Des. Eng.*, vol. 8, no. 2, pp. 756–772, 2021, doi: 10.1093/jcde/qwab012.
- [45] Z. He, R. Du, and K. Perlin, "CollaboVR: A Reconfigurable Framework for Creative Collaboration in Virtual Reality," *Proc. - 2020 IEEE Int. Symp. Mix. Augment. Reality, ISMAR 2020*, pp. 542–554, 2020, doi: 10.1109/ISMAR50242.2020.00082.

- [46] V. Theoktisto and M. Fairén, "On extending collaboration in virtual reality environments," *Brazilian Symp. Comput. Graph. Image Process.*, pp. 324–331, 2004, doi: 10.1109/SIBGRA.2004.1352977.
- [47] C. Pidel and P. Ackermann, "Collaboration in virtual and augmented reality: A systematic overview," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12242 LNCS, pp. 141–156, 2020, doi: 10.1007/978-3-030-58465-8\_10.
- [48] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust Speech Recognition via Large-Scale Weak Supervision," Dec. 2022, [Online]. Available: http://arxiv.org/abs/2212.04356