EXTENDING MODEL USE IN VIRTUAL HERITAGE: USER–CENTRIC IMPLEMENTATION OF A PROTECTED REMOTE RENDERING VISUALIZATION TOOL

Leif Christiansen

Submitted to the faculty of the University Graduate School
in partial fulfillment of the requirements
for the degree
Doctor of Philosophy
in the Department of School of Informatics, Computing, & Engineering,
Indiana University
May 2019

Accepted by the Graduate Faculty, Indiana University, in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Doctoral Committee

Name, Post-Nominal Initials

Name, Post-Nominal Initials

Name, Post-Nominal Initials

Name, Post-Nominal Initials

Date of Defense: 04/22/2019

To all the lovely three-dimensional people in my life.

ACKNOWLEDGEMENTS

Throughout writing this dissertation I have received a bounty of support and assistance. I would first like to express my sincere thanks and gratitude to my adviser Bernard Frischer for his unflagging help and support in formulating and executing this dissertation. I would also like to thank my committee members Mitja Hmeljak, Martin Siegel, and David Crandall for their insight and guidance, with special thanks to Mitja in first introducing me to programming computer graphics. In addition, I would like to thank my fellow students Matthew Brennan, Nicholas Plank, and Matei Tichindelean for their help refining the ideas behind this work and editing the written prose. Finally, a special thanks to my family for raising me right and my partner for keeping me focused and sane.

Leif Christiansen

EXTENDING MODEL USE IN VIRTUAL HERITAGE: USER–CENTRIC IMPLEMENTATION OF A PROTECTED REMOTE RENDERING VISUALIZATION TOOL

Virtual Heritage (VH) is the application of computing technologies, in particular, 3D graphics, to the documentation, study, preservation, and dissemination of cultural heritage. Given recent advances in both software and hardware, there has been a veritable boom in the production of 3D digital models of cultural heritage. These 3D digital models represent significant investments and stand at the intersection of claims to copyright of the digital replica and to ownership of the underlying digitized cultural property. Thus, it is understandable that creators of digital products, cultural heritage institutions, and owners of cultural property may desire to have control over the use, characteristics, and dissemination of VH models. Despite the growing prevalence of 3D digital models in VH, their role in the production of new knowledge remains to be examined in depth.

This dissertation presents the design, development, and implementation of a publicly available tool for the protected visualization of high-resolution 3D VH models in a web browser. First, I survey the historical precedents for the use of 3D digital models in VH and its related fields, with special attention to the epistemological function of models. Second, current practices and needs are documented through surveys and interviews with VH scholars. Third, I present the basic visualization tool and demonstrate its security and usability. Finally, the results of the literature review, surveys, and interviews are used to design and implement a more fully featured visualization tool based on protected remote rendering, one intended to more clearly

address the actual needs and practices of scholars in the new field of VH.

Name, Post-Nominal Initials

Name, Post-Nominal Initials

Name, Post-Nominal Initials

Name, Post-Nominal Initials

TABLE OF CONTENTS

Acknowledgements v

Abstract vi

List of Tables xiv

List of Figures xvi

Chapter 1: Introduction 1

1.1 Motivation 1

1.2 Problem Definition 2

1.3 Key Contributions 4

1.4 Structure of Work 4

Chapter 2: Epistemological Value of Models 6

2.1 Motivating Examples 7

2.1.1 Physical Model: Pumapunku 7

2.1.2 Non-Physical Model: Antonine Itineraries 12

2.1.3 Dictionary Definition 14

2.2 Trends in the Literature 16

2.2.1 Text Bias 17

2.2.2 Material Models 18

2.2.3 "Scientific" Fields 20

2.3 Defining Models 21

2.3.1 Denotation, Demonstration, and Interpretation 21

2.3.2 The Semantic Account 22

2.3.3 Models as Agents 23

2.3.4 The Analogical Property Account 24

2.4 Implications for 3D Model Use 26

2.4.1 Validity 27

2.4.2 Treating the Digital as Material 28

Chapter 3: History and Technologies for 3D Model Visualization 30

3.1 History of Model Use in the Humanities 30

3.1.1 Plaster Casts 31

3.1.2 Architectural Models 35

3.2 3D in the Humanities 37

3.2.1 A Brief Intro to 3D Computer Graphics 37

3.2.2 3D Modeling 39

3.2.3 File Formats 41

3.2.4 Sharing on the Web 45

3.3 Web3D 46

3.3.1 VRML to WebGL 46

3.3.2 Current State 47

3.4 Features for a VH 3D Viewer . . . 49
3.4.1 Technical Features . . . 49
3.4.2 Showing Uncertainty . . . 49
3.4.3 Semantic Annotation . . . 50
3.4.4 Exhibition, Contextualization, and Comparison . . . 51
3.4.5 Computational Analysis . . . 51
3.4.6 Security . . . 52
3.4.7 Accessibility . . . 53
3.5 Current Web Viewers in VH . . . 53
3.5.1 Sketchfab . . . 53
3.5.2 Unity . . . 54
3.5.3 X3DOM . . . 55
3.5.4 3DHOP . . . 56
3.5.5 Custom Viewers . . . 57
3.6 Future Developments for VH Viewers . . . 58
3.6.1 Unity Assets . . . 58
3.6.2 Extending Single Model Viewers . . . 59
3.6.3 Remote Rendering . . . 60
Chapter 4: Protected Remote Rendering Revisited . . . 61
4.1 Previous Work . . . 61
4.1.1 Protected Remote Rendering . . . 61
4.1.2 Remote Rendering . . . 64

4.1.3 Web Clients 70
4.2 System Architecture 72
4.2.1 Client 75
4.2.2 Server 79
4.2.3 Defenses 82
4.3 Security 86
4.3.1 Test Model 86
4.3.2 Shape-from-Silhouette 87
4.3.3 Shape-from-Shading 90
4.3.4 Shape-from-Motion 96
4.4 Performance 102
4.4.1 Render Speed 105
4.4.2 Time to Render 109
4.4.3 Concurrency 112
4.5 Conclusion and Future Work 116

Chapter 5: Characterizing Model Use in VH 118

5.1 Previous Work 118
5.2 Methods 120
5.2.1 Survey 120
5.2.2 Interview 122
5.3 Delineating VH Scholars 123
5.3.1 Demographics 123

5.3.2 Position . . . 125
5.3.3 Technology Adoption . . . 129
5.3.4 Software . . . 133
5.3.5 Sharing and Visualization . . . 138
5.4 Security Concerns . . . 141
5.5 Literature Validation . . . 143
5.5.1 High-resolution Models . . . 144
5.5.2 Accessibility . . . 147
5.5.3 3D Analysis . . . 149
5.6 Personas . . . 150
5.6.1 Template . . . 151
5.6.2 Development . . . 156
5.7 VH Ambassadors . . . 158
5.8 Conclusion . . . 160

Chapter 6: VH Feature Implementation . . . 162
6.1 Snapshot Gallery . . . 162
6.2 Path Render . . . 165
6.3 Navigation Selection . . . 168
6.4 Scale Bar . . . 169
6.5 Distortions GUI . . . 172
6.6 Future Work . . . 174

Chapter 7: Conclusion and Looking Forward . . . 176

Appendix A: Dictionary Statistics for “Model” 180

Appendix B: Mesh Comparison Workflow 182

Appendix C: Survey: Uses of 3D 184

Appendix D: Sample Interview 193

References 194

Curriculum Vitae

LIST OF TABLES

3.1 OFF, OBJ, and PLY files describing a quadrilateral composed of a red triangle and blue triangle. The rendered shape is shown in Figure 3.7. 43

4.1 Sumary of remote rendering systems adapted from Shi and Hsu [154]. Changes are shown in red. 69

4.2 Code for a simple scene in three.js. For simplicity, HTML code is omitted. The resulting render is shown in Figure 4.5. 76

4.3 Mouse and touch input and their corresponding interaction events. 77

4.4 The <body> of a simple HTML page running a Seymour client. 78

4.5 Metrics for the dimensions for the test model shown in Figure 4.11. Triples are reported as x, y, z values. Values were computed using the 'Compute Geometric Measures' tool of Meshlab. 100

4.6 Render times in ms for 15 frames with JPEG quality 40, 512x512 pixels, and no blended high-frequency noise. 106

4.7 Render times in ms for 15 frames with JPEG quality 40, 512x512 pixels, and blended high-frequency noise. 106

4.8 Render times in ms for 15 frames with JPEG quality 90, 512x512 pixels, and no blended high-frequency noise. 107

4.9 Render times in ms for 15 frames with JPEG quality 90, 512x512 pixels, and blended high-frequency noise. 107

4.10 Render times in ms for 15 frames with JPEG quality 90, 1024x1024 pixels, and no blended high-frequency noise. 108

4.11 Render times in ms for 15 frames with JPEG quality 90, 1024x1024 pixels, and blended high-frequency noise. 108

4.12 Average render times for the 6 renderer configurations 109

4.13 Render times for 3DHOP. 110

4.14 Render times for Seymour. 110

5.1 A selection of eligible groups and communities contacted for participation in the survey in interviews. Names are hyperlinked to relevant websites. 121

5.2 Percentage of the 78 respondents that selected each field and a list of the "Other" fields entered. Responses are reproduced exactly as entered. 126

5.3 A reproduction of results from the survey of Koller et al. [78]. 126

5.4 Current positions reported by survey respondents. 127

5.5 Positions supplied by respondents for "Other". Some responses have been altered to maintain anonymity. 128

5.6 The software mentioned by survey correspondents and their number of mentions. . 136

5.7 A rough classification of the software listed by survey respondents. 137

A.1 Python code used to retrieve definitions for "mode" and compute dictionary statistics using WordNet. 181

LIST OF FIGURES

2.1 Images from Tiwanaku 8

2.2 Figures from Vranich [11]. 9

2.3 A simple network with nodes A, B, C, D, and E. 12

2.4 A Netdraw rendering of Graham's network [16, p. 51]. 13

2.5 A reproduction of the Britain portion of the Antonine Itineraries. 13

2.6 Echenique's diagram for the relation between reality, models, and observers [41]. . 25

3.1 *The Plaster Cast Collection in the Royal Academy of Fine Arts* by Julius Exner [43]. 31

3.2 "En billehugger (Christen Christensen) arbejder efter levende model I sit atelier" by Wilhelm Bendz [45]. 32

3.3 Photograph of The West Court of the Cast Courts of the Victorian and Albert Museum. 33

3.4 An example scene with camera, light, and 3D objects (left) and rendered image (right) using the synthetic camera model. 38

3.5 The same scene rendered with an orthographic projection (left) and a perspective projection (right). 39

3.6 An example photogrammetric workflow from the initial photographic input to the final point cloud. 42

3.7 A render of the shape described in Table 3.1. 44

4.1 A schematic of a basic remote rendering architecture. 62

4.2 Sample frames from a remote rendering application. Mouse actions are shown in red. 62

4.3 Sample frames from Seymour. Mouse actions are shown in red. 73

4.4 The system architecture of Seymour. 74

4.5 A rendering of the scene defined in Table 4.2. 77

4.6 Sample Seymour client requests. 80

4.7 A UML Class Diagram for the Seymour rendering module. 81

4.8 Stencil rendering used to blend an original image with 20% of a noise image. . . . 83

4.9 The original image (left) and a highly compressed JPEG of the same image (right). 84

4.10 Images from the original and photogrammetric model of the Antoninus Pius bust. . 88

4.11 The test model of Antoninus Pius without texture (left) and with texture (right). . . 89

4.12 The normal map for the test model presented in Section 4.3.1. 92

4.13 Heatmaps visualizing the difference between the original normal map and reconstructed normal maps. 93

4.14 Histograms showing the distribution of distances calculated between the original normal map and reconstructed normal maps. 95

4.15 From left to right: the original 3D geometry, the reconstructed normal map without server defenses, the reconstructed normal map with all server defenses. 96

4.16 An example distance calculation from CloudCompare [230]. 99

4.17 Histograms showing the distribution of distances calculated using CloudCompare with varying JPEG quaity, geometric distortions (in object space), and blended high-frequency noise. 101

4.18 The reconstructed models using SfM without textures. Server defenses were as follow: **A**: Original model **B**: JPEG: 90; 3 lights **C**: JPEG: 90 **D**: JPEG: 90; Geometric Distortion: 0.001 **E**: JPEG: 90; Noise: 2% **F**: JPEG: 40 **G**: JPEG: 20 **H**: JPEG: 40; Geometric Distortion: 0.001; Noise: 2% 103

4.19 The reconstructed models using SfM with textures. Figures are labeled as with Figure 4.18. 104

4.20 A plot of response times as the number of clients making requests is increased incrementally, up to 30 clients. 113

4.21 A plot of response times as the number of clients making concurrent requests is increased incrementally. 114

4.22 A plot of response times as the number of clients making requests is increased incrementally, up to 100 clients. 115

5.1 Box plots showing age distributions. Koller et al. reported ages as ranges. Thus, for the purpose of visualization, each range was represented by the median value. . . . 124

5.2 A histogram (left) and box plot (right) showing the distributions of the year survey respondents began producing 3D digital models. 130

5.3 A comparison and regression showing the lack of correlation between the year at which a respondent began producing models and the total models they have produced throughout their career. 131

5.4 A comparison and regression showing the lack of correlation between the year at which a respondent began producing models and the total models they have produced throughout their career. 131

5.5 Visualization of *How would you characterize your abilities to perform the following tasks?* 134

5.6 Distribution of software listed per respondent. 135

5.7 Visualization of *How often do you perform the following actions in your research or instruction?*. Note that the legend lists the color correspondences in the same order as the figure. 140

5.8 Persona: The Curious Scholar 153

5.9 Persona: VH Evangelist 154

5.10 Persona: Technological Explorer 155

6.1 The user interface for the "Snapshot Gallery" with annotated UI elements: 1 snapshot button, 2 switch button, 3 snapshot thumbnail, 4 download button, 5 delete button. 164

6.2 Sample frames from the Path Render implementation with three user interface elements, the record button (top), frame slider (middle), and the animator (bottom). Frames are ordered left-to-right and top-to-bottom. 166

6.3 A screenshot of the Navigation Selection page. 169

6.4 Two frames from the scale bar feature, one showing a scale in decimeters (left) and one a scale in centimeters (right). 171

6.5 A screenshot of the Defenses GUI page. 173

CHAPTER 1
INTRODUCTION

1.1 Motivation

Since at least the 1970s, computing technology has been used by art historians, archaeologists, architects, and scholars in the humanities to further their respective studies. In this pursuit, the field of Virtual Heritage (VH) has emerged, a field that utilizes modern computing tools to study, document, and disseminate cultural heritage (CH) through 3D graphics technology. Using technologies such as laser scanning, photogrammetry, and 3D modeling suites, VH scholars may now produce metrically accurate representations of physical objects as well as theoretical reconstructions of damaged or non-extant objects and structures, all in the form of 3D digital models. VH scholars have produced and used 3D digital models in a variety of contexts including structural engineering studies [1], virtual exploration [2], museum exhibits [3], and excavation documentation [4]. In all cases, 3D digital models have proven an effective tool for scholarship, enabling scholars to more efficiently perform their work.

In recent years, 3D digital models have experienced a boom in scholarly publications. Roosevelt et al. describes this phenomenon as a "paradigmatic shift" of the archaeological community's wider adoption of 3D technology as a viable means of scholarship, one driven by the increased performance and cost effectiveness of both the hardware and software components involved in producing and using 3D digital models [5]. Evidence for this shift may be observed not only in publications but also in the repositories of 3D digital models now being made available to scholars and the public. Numerous CH institutions, like museums and universities, have undertaken digitization projects. Some notable projects are the Smithsonian Institute [6], the Uffizi Galleries [7], and the British Museum [8]. Additionally, the EU has funded a series of wide scale, multi-institution 3D digitization projects including 3D-COFORM from 2008 to 2012 and 3D-ICONS from 2012 to

2015 [9]. Between the aforementioned projects, the many more not mentioned, and the work of numerous individuals, there now exist tens of thousands of digitized CH objects. This significant investment in 3D digital models and their production belies scholar's and professional's belief in 3D digital models as useful to the endeavors of those that seek to study and preserve CH.

With the advent of WebGL and native, hardware accelerated 3D graphics on the web, many of these 3D digital models have begun to be shared online. However, the products of these digitization efforts are often highly detailed digital representations, described in digital formats whose files quickly reach unweildly size, making their wider dissemination difficult. Furthermore, the visualization of these high resolution models requires powerful hardware and specialized software. In order to share these 3D digital models more widely, VH practitioners are forced to produced low-resolution or 2D derivatives of the original 3D digital models, thereby limiting the (re)-usability of the 3D content.

The goal of this dissertation is to investigate how individuals involved in VH make use of 3D digital models and identify avenues by which this use may be furthered. To this end, an online visualization tool will be developed, one which allows for the interactive viewing of high-resolution 3D digital models in a web browser. The tool, name Seymour, builds from previous work on remote rendering, which provides demonstrable security for the valuable 3D assets visualized.

1.2 Problem Definition

The primary focus of this dissertation is on the 3D digital models produced and used within VH, and how these models may be described, studied, and disseminated more effectively. As with VH itself, which is a rich intermixing of various theoretical frameworks and technical methods, we may approach such questions from both philosophical and technical directions.

Most often in VH, the 3D digital models are themselves not the final objects of study. Rather, 3D digital models are used as a means to study the original objects, which the models seek to represent. This relation between model and original, the process of producing models, and the ultimate use of models for the production of new knowledge within the field pose epistemological questions. While such questions have not often been addressed in VH, the philosophy of science has long engaged in a discussion surrounding models and their use, and this discussion may be brought to bear on the topic of models in VH.

But regardless of the theoretical value and definition of models more broadly in VH, 3D digital models, in particular, have become a widely adopted tool for work within the field and by those more broadly seeking to preserve, study, and disseminate CH. A multitude of issues present themselves throughout the use and production of these models. However, one of the most pressing issues for the future of the field and the effectiveness with which VH practitioners perform their work, is the wider dissemination and visualization of 3D digital models. 3D digital models of CH are being produced at an accelerating pace, but much of this work remains difficult to access, view, or interact with. While the maturation of 3D technologies for the web have proven an incredibly boon to such endeavors, online dissemination and visualization are still hampered by limits to file size and lacking hardware among those wishing to view and interact with these models.

VH practitioners make use of a variety of software to share and visualize 3D digital models online, among the most popular Sketchfab, Unity, and 3DHOP. Of these three, only 3DHOP was specifically developed for scholarly use in VH. This type of user-centric development represents an important avenue for the production of tools tailored especially for application to VH. However, the development of tools, like 3DHOP, has to date primarily relied on the individual experience of those implementing the tools and review of the literature. The needs and practices of VH practitioners remain to be solicited and codified more generally.

Finally, a key need for applications using 3D digital models of CH remains to be addressed, security. Besides all or nothing access controls, there is currently no means of protecting 3D content while also allowing interactive visualization. At best existing visualization tools make use of obfuscation to prevent the unauthorized reproduction and use of 3D digital models.

The remote rendering architecture, originally proposed by Koller et al. [10] in 2004, provides a useful means for developing a visualization tool for VH which directly addresses the issue of sharing large, interactive 3D digital models on the web, while protecting the valuable 3D digital assets from unauthorized access and reproduction.

1.3 Key Contributions

The main contributions of this dissertation are two-fold, the enumeration and investigation of 3D digital model use in VH and the development of an online visualization tool for VH.

The treatment of 3D digital model use proceeds according to a critical review of the literature and the direct solicitation of opinions and practices from those working in VH. This work contributes a novel synthesis of the philosophy of science and the VH literature on models as well as the documented results of surveys and interviews, serving to more clearly describe and delineate the field of VH.

The development of the online visualization tool revisits and more complete specifies the decade old remote rendering architecture proposed by Koller et al., in so doing expanding the original system to include a web browser-based client and textured models. Furthermore, the security and performance of the system is demonstrated given modern 3D reconstruction techniques, which may be used by malicious users to retrieve the high-resolution geometry.

A key contribution of the visualization tool is its security, an issue not directly addressed by any existing visualization tools utilized in VH. Using remote rendering, the high-resolution 3D assets are never sent to the client machine, and thus never directly exposed to attack. Furthermore, the use of distortions, subtly perturbing both the underlying 3D scene and the rendered 2D image, complicates malicious techniques for the unauthorized reconstruction of the high-resolution geometry.

1.4 Structure of Work

Chapter 2 serves as the theoretical basis framing the discussion of models. This chapter surveys the philosophy of science literature treating models more generally and applies this discussion to the particular case of 3D digital models in VH, presenting a definition of models that serves to better describe the applications in VH and one with the implications for the development of a VH visualization tool. Chapter 3 traces the use of models, in particular, physical models, in art history, archaeology, and architecture, demonstrating the broader history of use in which 3D digital models form the most recent development. This survey continues through to the modern day and the current technologies used in VH for the creation, visualization, and analysis of 3D digital models. Chapter 4 introduces and surveys applications of remote rendering, a client-server architecture that enables the visualization of high-resolution 3D graphics on low-powered devices, presenting a novel remote rendering visualization tool for VH. The security and performance of the tool are tested. In order to more effectively develop a full featured tool and further delineate the field of VH, Chapter 5 reports the findings of interviews and surveys with VH practitioners. Chapter 6, combines the theoretical and empirical study of 3D digital model use from the previous chapters to develop features extending the basic visualization tool presented in Chapter 4, features meant to directly address existing needs and practices within the VH community. Finally, Chapter 7 concludes with an overview of the work accomplished and a discussion of future work on the tool developed here and the outlook for VH more broadly.

CHAPTER 2
EPISTEMOLOGICAL VALUE OF MODELS

3D digital models are but one of a variety of types of models that have been fruitfully deployed in research. As such, we may look to models more generally to better illuminate the beneficial ways in which models may be produced and used in research settings. Furthermore, we may find a theoretically richer definition of 3D models, one which situates models more firmly in the historical and epistemological practices of scholars in VH and its preceding fields.

Models have been a topic of interest across a number of disparate fields. However, models have received the most direct and complete treatment in the philosophy of science. In the philosophy of science, scholars have grappled with defining a model, the role of models in scientific practice, and the connection of models to scientific theory. A rich discourse has emerged in the philosophy of science literature, but one focused closely on a small subset of fields. In developing their theories on models, scholars have looked predominantly to physics, biology, chemistry, economics, and mathematics. This insulation is representative of a larger bias present in the philosophy of science, supported by particular conceptions of knowledge and historical ties between fields. However, models have proven useful in other fields. Of particular interest to use in this study is their use in art history, archaeology, architecture, and VH. Within these fields, models have performed a variety of roles in producing and transmitting knowledge, in many ways similar to functions of models in the fields of interest to philosophers of science. Scholars in art history, archaeology, architecture and VH have not been oblivious to the important epistemological work done by models but these two traditions of thought have not been brought into direct contact with one another. By considering the two in concert, the rich theoretical frameworks developed by philosophers may be brought to bear on new subject material, generating insight into the practices of researchers and the limits of the current philosophy literature. Finally, with a deeper understanding of the creation of models and their role in research, model creators and model users may more effectively and reflectively perform their work.

2.1 Motivating Examples

Before we engage with the theoretical frameworks scholars have proposed to define and explain models, let us ground our discussion in some examples from the field, ones that illustrate the complexities of model use in VH.

2.1.1 Physical Model: Pumapunku

In 2018, Alexei Vranich published an article revisiting the seemingly ineffable ruins of Pumapunku, shown in Figure 2.1b and Figure 2.1c [11]. Pumapunku is an astonishingly geometric temple located among the ruins of Tiwanaku, a pre-Columbian site dating to approximately 500-950 A.D, shown in Figure 2.1a. Most recently, the temple has been the focus of, to put it mildly, imaginative theories as to the extraterrestrial origins of the mysterious masonry [12]. But the Pumapunku has captured the imagination of visitors far before the 21st century, forming a part of mythical and socio-political narratives in the region as far back as the Incan Empire, when the site was repurposed as the birthplace of the Incan people. Correspondingly, each group that has laid claim to the site has intervened to better shape the extant ruins to their narrative. The Spanish conquistadors went about destroying the site as a symbol of the America's native peoples; during the wars for independence from Spain, the ruins were re-erected to symbolize the start of a new order; and finally, the site was heavily reconstructed as part of a Bolivian revival project meant to create an ancient empire to rival other ruins found in American countries like Peru and Mexico. This most recent effort in the mid-twentieth century involved heavy-handed excavation and reconstruction efforts, resulting in what has been considered to be one of the worst reconstructed ancient sites in the Americas. Given this long series of interventions and the wealth of information lost or destroyed, Tiwanaku and Pumapunku represent a difficult challenge for archaeologists.

Vranich sought to revisit, repurpose, and reaffirm historical data gathered about the site documenting its pre-restoration (although not pre-destruction) state. Vranich's essential concept was

(a) A view of the reconstructed site.

(b) Blocks from Pumapunku [13].

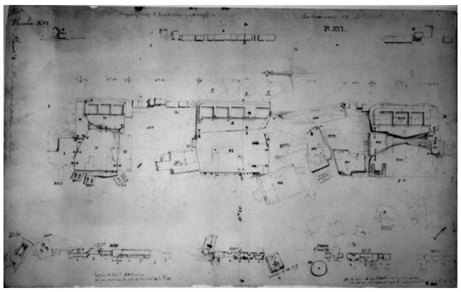
(c) Adorned blocks from Pumapunku [14].

Figure 2.1: Images from Tiwanaku

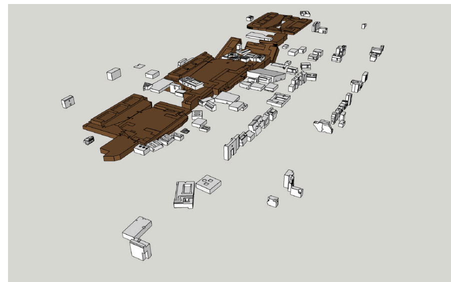
to produce scale models, or maquettes, of the fragmentary blocks for which records exist and to use these to visualize previous historic arrangements of the site and test potential reconstructions. Vranich recognized that methods like photogrammetry and laser scanning could be used to accurately document the current state of the physical objects. But for the purposes of Vranich, these forms of recording were deemed inappropriate. Firstly, the actual recording of the photogrammetric or laser range data was immensely complicated owing to local bureaucracy and the remote nature of the location. Secondly, the non-uniform and highly detailed geometries produced by the aforementioned techniques did not lend themselves to 3D printing. However, the numerous flat surfaces, straight lines, and simple geometric shapes from Pumapunku meant that the original state of the blocks could be modeled accurately with a minimal number of measurements. Thus, based on measurements taken by JP Protzen in the 1990s, Leonce Angrand in 1848, and Max Uhle in 1893, as well as some additional measurements taken by members of the project, Vranich and his team produced 157 3D digital models representing architectural elements from the site [11, pp. 9, 11].

These models were then printed using a powder-based 3D printer [11, p. 12]. Vranich and his team members used the 3D printed blocks as tools to explore and test potential reconstructions, ultimately resulting in a novel proposal for a partial reconstruction of the northern portion of the building and a general layout for the entirety of the structure [11, p. 15].

The research of Vranich demonstrates a complex interrelation between the original objects and the various derivative models they produced, the final of which, the 3D printed scale model, resulted in the generation of new knowledge about the site. But how do the models relate to the original



(a) A site drawing by Angrand from 1848.



(b) A digital reconstruction of the historic state.

(c) 3D printed blocks.

Figure 2.2: Figures from Vranich [11].

things modeled? How exactly do they participate in the generation of new knowledge? And how may we extrapolate this knowledge, gained through the model, back to the original? Let us consider in more detail the process from the original to the models.

First, physical measurements were taken from the site by 19th, 20th, and 21st century archaeologists. The exact procedure of these measuring processes is not described, but Vranich does state that the most recent method used was taught to the team by the 20th century measurer J.P. Protzen. We may also assume that the measuring techniques involved some level of built in error (for example, meter sticks are not always exactly a meter) as well as operator error. Vranich operates on the assumption that, taken in concert, these measurements provide a representation of the site useful for scholarship. And this is an assumption that few scholars in archaeology would find problematic, as site plans and measurements are considered part of the standard toolkit for studying ancient sites. Certainly, the reduction of the site to mere measurements of blocks represents a profound simplification, but one that Vranich has identified as useful to his scholarship.

Second, the measurements taken from the physical blocks of the site were used to produce 3D models of the blocks. This stage involved the use of an additional tool, the 3D modeling software Sketchup, and corresponding technical training. Again, we may assume some level of built-in and operator error. In fact, Vrasich did detect and correct some errors introduced in the 3D modeling process, although the physical prints allowed for their easier detection [11, p. 12]. The observations from the site underlying the 3D models has not changed. In effect, the data are the same as the original measurement's recordings in the paper and pen notebooks of the archaeologists. However, in realizing these measurements in a new medium, a 3D digital model, Vrasich expanded the types of operations and tools, both cognitive and instrumental, that could be applied to the data. We shall return to this point in more detail shortly, but for now it is sufficient to note that the 3D visualation of the measurement allowed for the production of the third stage of modeling, the 3D printing.

Third, the 3D model was transformed from a digital form to a physical form through the use of a 3D printer. Vrasich provides exhaustive description of the metrological limits imposed by 3D printers and different material's potential for distortion during the printing process. Ultimately, Vrasich chose to print using a more accurate, although also more expensive, form of powder printer. Again, we may see the cognizance of Vrasich for the potential negative impacts of the model production and his justification for their utility.

Throughout these stages of transformation and the derivative models produced, no new information about the actual site was added. One could even claim that with each derivative new error was introduced and existing error potentially compounded. Additionally, throughout the process Vrasich made a number of assumptions and simplifications. And yet, the final model was ultimately useful in studying the original site and was able to produce new insight about the site. Vrasich argues that the ultimate benefit of his workflow is that physical reproductions allow for one to bring particular cognitive forces to bear on the subject material, the cognitive process of 3D visualization and manipulation, a skill in which archaeologists are especially well-trained. Vrasich claims that this same cognitive apparatus may be applied to the 3D digital model but that our intuition is often handicapped by the lack of intuitive manipulation techniques for the digital object. Could the digital and physical models be said to belong to the same class of things? Since they both are visualizations of the original measurements, could the measurements be said to be a model of the site? In effect, all three visualize the same information, that is 3D measurements of the original objects. Finally, given the several transformative steps, and the assumptions and error therein, how may we be sure that the results generated from the model apply to the original? In his article, Vrasich compelling argues for each of his simplifying assumption, thus attempting to justify the projection of knowledge from the model back to the original.

While superficially, the use of scale models may seem a common sense and unobjectionable practice, on closer inspection the process actually involves a number of steps requiring explicit explanation and justification, if conclusions concerning the model are to be applied to the original. This process is largely implicit in the research literature but raises important epistemological and ontological questions that deserve more direct attention.

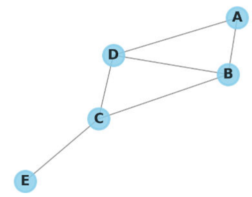


Figure 2.3: A simple network with nodes A, B, C, D, and E.

2.1.2 Non-Physical Model: Antonine Itineraries

In VH, research efforts of a drastically different sort have also been labeled 'modeling', among them network analysis. Scholars in archaeology especially have used techniques developed for the study of complex networks to records of the ancient world. A network may be simply described as a collection of things, called nodes, which are connected through some rule or relation. A simple example of a network is shown in Figure 2.3. Both material and literary evidence from the archaeological record has been used to produce such networks. Topics range from material networks based on the presence or absence of certain grave goods (Mills 9) to spatial networks based on ancient road connections [15, p. 7].

In 2006, Graham produced a network representation of the Antonine Itineraries, Roman texts listing settlement-by-settlement routes throughout the empire. While the exact date of the creation of the itineraries is unknown, it is often ascribed to the 2nd century under the rule of Antoninus Pius. Graham contends that the itineraries, in comparison to pure geographic distances, provide a more accurate representation of Roman conceptions and experiences of space, as these were the actual routes used to traverse the empire at the time. Graham treated settlements as nodes and considered two settlements to be connected if those settlements appeared adjacent in an itinerary. A reproduction of Graham's resulting network and a partial reproduction are shown in Figure 2.4 and Figure 2.5 respectively.

Graham then conducted a series of quantitative tests to study the implications of the resulting network on Roman's conceptions of space and the spatial qualities throughout the Roman empire. Graham argues that the "cohesion" information of the graph indicates that Italy and Iberia were more connected, while Gaul and Britain were less connected [16, pp. 52–53]. In other words, in

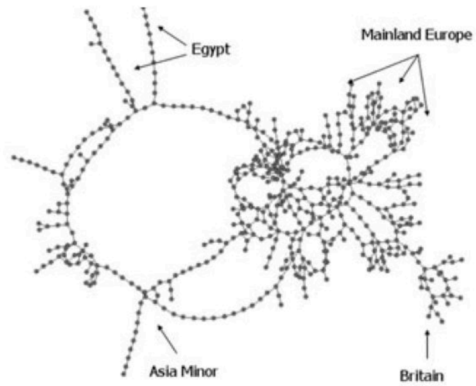


Figure 2.4: A Netdraw rendering of Graham's network [16, p. 51].

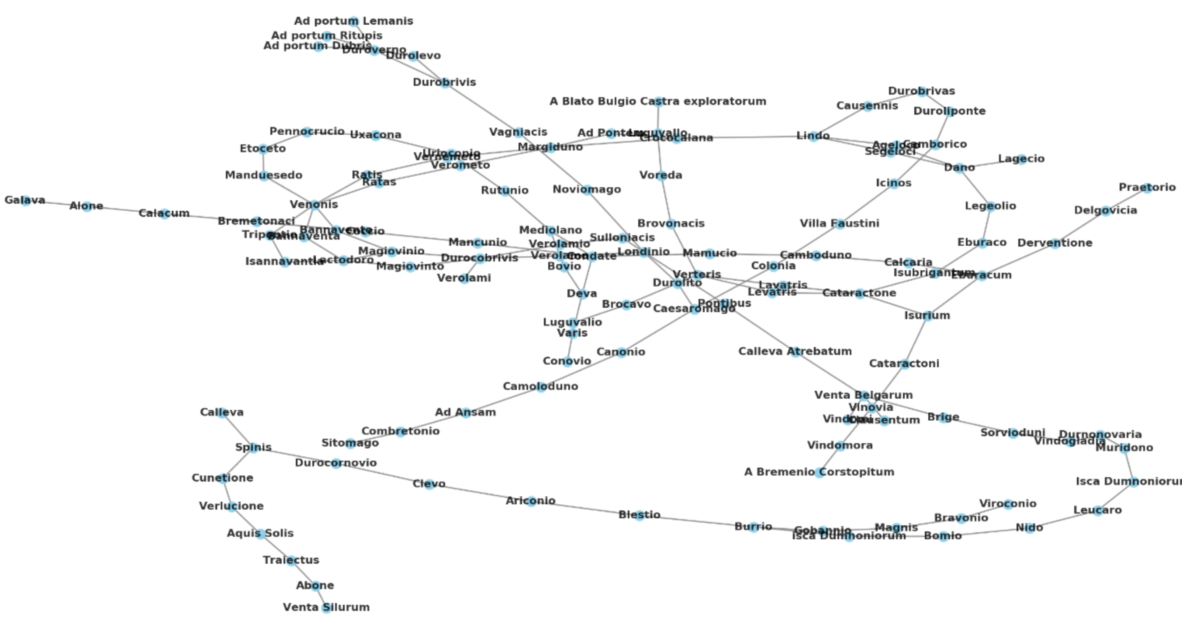


Figure 2.5: A reproduction of the Britain portion of the Antonine Itineraries.

comparison to Gaul and Britain, Italy and Iberia contained more internal connections, information could flow more quickly within them, and they were more resilient to isolation if any one settlement should become unreachable (for instance in the case of a plague).

Graham then 're-animated' the network using agent-based modeling. In agent-based modeling, individual autonomous agents are introduced to the network with given rules for network traversal and inter-agent interaction. In Graham's case, the agents were used to model information transmission throughout the network, therefore meant to simulate how ideas or cultural practices would diffuse through the empire (Graham 55). Finally, Graham compared the observed information diffusion in his model to the diffusion of epigraphic inscriptions evinced in the Corpus Insciptionum Latinorum, arguing that "patterns" in the inscriptional density are "highly reminiscent of those in the model" [16, p. 58].

At first glance, Graham's network and agent-based models may seem a far cry from the scale models of Vrasich. Are these two examples of the same class of thing? Or rather is it merely an unfortunate result of the imprecise English language that we have come to call these both "model"? To the contrary, close inspection yields striking similarities between the two cases. In each case, the scholars began with observations on some real-world system, in the case of Vrasich measurements of stone blocks and in the case of Graham a written record of the routes traveled by 2nd century Romans. Then these observations were transformed into a new medium, a 3D digital model for Vrasich and a network for Graham. In both cases, the scholars made simplifying assumptions with explicit justification. Once realized in the new medium, the scholars were then able to perform new methods of analysis on the original data. Ultimately, these two examples represent the same core process of modeling, and the same final output of a model.

2.1.3 Dictionary Definition

As a precursory attempt to offer a definition, we may turn to the Merriam-Webster Dictionary. Merriam-Webster offers 14 different meanings for 'model' in its noun form. They are as follows:

1. *obsolete* : a set of plans for a building

2. *dialectal British* : COPY, IMAGE [1]
3. : a structural design
4. : a usually miniature representation of something
5. : an example of imitation or emulation
6. : a person or thing that serves as a pattern for an artist
especially : one who poses for an artist
7. : ARCHETYPE (: the original pattern or model of which all things of the same type are representations or copies)
8. : an organism whose appearance a mimic imitates
9. : one who is employed to display clothes or other merchandise
10. **(a)** : a type or design of clothing
(b) : a type or design of product (such as a car)
11. : a description or analogy used to help visualize something (such as an atom) that cannot be directly observed
12. : a system of postulates, data, and inferences presented as a mathematical description of an entity or state of affairs *also* : a computer simulation based on such a system
13. : VARIANT sense 2 (: a form or variant of a type or original)
14. : ANIMAL MODEL (: an animal sufficiently like humans in its anatomy, physiology, or response to a pathogen to be used in medical research in order to obtain results that can be extrapolated to human medicine) [17]

Our previous models discussed do fit among these definitions, specifically 4 and 12 for Puma-punku and the Antonine itineraries respectively.

However, Merriam-Webster does not seem to have narrowed our search by much, as model now appears as an overloaded term with 14 separate meanings. According to a statistical analysis an English dictionary from WordNet [18], the number of meanings for an English noun has a mean of 1.283 and a standard deviation of 0.965. This means that 95% of English nouns have fewer than 4 meanings. This places it among only 55/119034 (0.05%) of English nouns with 14 or greater definitions.[2]

To simplify our definition, we will constrain ourselves to those cases of models derived from something already existing in the world, as opposed to those models used as patterns or reference for the creation of new things. Therefore, 1, 2, 3, 5, 6, 7, 8, 9, and 10 may all be removed. This leaves us with 4, 11, 12, 13, and 14. 13 may also be removed, as it is so vague as to be not useful. The primary difference between the remaining definitions is the means of representation. 4 implies a physical representation, 11 a linguistic, 12 a mathematical, and 14 a biological. But is the way

[1]Capitals indicate a synonym; definitions for synonyms are provided in parenthesis
[2]See Appendix A for a description of this calculation.

in which these models function actually different? In our previous example of Pumapunku, did the miniature model not involve postulates, data, and inferences (definition 12)? Was it not used to obtain results to be extrapolated to the original (definition 14)? In the case of the Antonine itineraries, did the network model not help to visualize something previously opaque (definition 11)? Did it not represent the lengthy list of place names in a more compact form (definition 4)? While useful in the dictionary sense, our remaining definitions actual serve to describe the same essential class of things and point to some of the major trends scholars have used in defining models: representation; analogy; postulates, data, and inferences; and extrapolation of results.

2.2 Trends in the Literature

The philosophy of science offers the most complete treatment of models as they relate to scholarly research. Most importantly, the literature in the philosophy of science will provide a means for reaching a clearer and more complete definition of models, encompassing both the physical and non-physical examples previous discussed, situating models in the process of knowledge production.

Philosophers of science began to rigorously discuss the concept of a 'model' in the 19th century, primarily concerned with the introduction of apparently unobservable entities like atoms and electrons [19, p. 299]. The study of models has maintained and proliferated into the 21st century, shifting towards attempts to explain model's function in science and role of theories therein [20] [21] [22] [23]. Scholars have pursued a variety of questions including: What is a model? How do models function in science? Who is a modeler? How do models function in the production of knowledge? In pursuing these questions, scholars have engaged with a number of central debates in the philosophy of science concerning the nature of theories and data as well as the ontological status of the elements studied by and used in science.

Despite the wealth of studies on models, there remains a distinct lack of consensus and a nebulousness of terms, which even applies to model itself. The models studied range from those that walk down the runway to those that exist only in the mind, from prototype cars used on assembly lines to hydraulic machines used by economists [24][22] [25] [26]. Scholars are certainly aware of this plurality of opinion; Marx Wartofsky coined the phrase "the model muddle" [27, p. 1], R. I. G. Hughes dubbed the concept of model "slippery" [28, p. S325], and Carl Craver prefaced his work by conceding that "the term 'model' is notoriously ambiguous" [29, p. 65]. This is not to say that there are no points of agreement among the many positions. To the contrary, several trends are evident in the literature.

2.2.1 Text Bias

In order to explain and develop their theories on a definitively capricious topic, scholars often use exemplars of models, just as we have done in introducing this chapter. A survey of the exemplars chosen by scholars belies a preference for models more theoretical than physical. Physics models have proven especially popular. Hesse traces the use of models back to the thinkers of ancient Greece, with their models of the atom and solar system [19, p. 299]; Giere uses the mathematical equations describing a pendulum as his "canonical example" of a model [22, p. 745]; and Hughes devotes ample time to Galileo's model of falling bodies [28]. Other fields well represented in the literature include biology [30] [31], chemistry [21], and economics [24] [26]. What emerges is a prevalent focus on models that are primarily theoretical, most often composed of interrelated mathematical equations.

A number of scholars have explicitly stated their interest in theoretical models. Giere claims that "the models of most interest are theoretical models" [32, p. 5]. Often, scholar's conceptions of models and choice of exemplars are influenced by the scholar's respective fields. This is certainly the case for the physicists and historians of physics like Giere and Hesse, for as Hacking writes "a model in physics is something you hold in your head rather than your hands" [33, p. 216]. The result of these trends is that the discussion of models has focused on theoretical, abstract models from only a few specific fields.

The theoretical focus of past scholars and their dismissal of material models is a manifestation of a broader trend in the philosophy of science, one that privileges a semantic account of knowledge. In his 2004 book *Thing Knowledge*, Davis Baird labels this trend as "text bias" and identifies it with David Gooding's account of semantic ascent. In semantic ascent, "observers ascend from the world to talk, thought and argument about the world" [34, p. 8] [35, p. 3]. According to Baird, this epistemological bias may be traced back to the works of Plato and his description of knowledge acquisition as the use of reason to move from the physical world to the world of ideas [Baird 2004, 5]. However, this focus on semantic ascent has caused philosophers to ignore significant epistemological work that takes place in the material realm and is not readily explicable in terms of theory. Baird seeks to fill this lacuna in the literature by providing a "material epistemology for instrumentation" [34, p. xvii].

2.2.2 Material Models

Models are included among the instrumentation that Baird deems knowledge producing and are devoted a chapter length treatment. In chapter 2 of *Thing Knowledge*, Baird explores how models function epistemologically on a material level. Baird discusses three examples: 18th century orreries, John Smeaton's 18th century model waterwheel, and Watson and Crick's model of DNA from the twentieth century.

Baird argues that material models perform similar epistemological work to theories, as both represent their object [34, p. 25]. However, material models have an important difference, they may be manipulated materially. According to Baird, this offers "a different entry point for our cognitive apparatus" and thus allows one to investigate areas for which theory, language, or mental abstraction is inadequate [34, p. 40]. Material models are not necessarily a unique group separate from theoretical models. But as shown by Baird, material models do exhibit particular benefits and behaviors worthy of direct consideration.

There are certainly other, more material, types of models mentioned by scholars. One particularly popular example is the ball and stick model of DNA produced by Watson and Crick, a decidedly material object situated well within the theoretical discussion of unobservable entities framed in the literature. Stefanov and Giere make use maps in explicating their ontology of models, a similarly material model, although not necessarily as scientific. Clearly, material models are not absent from the literature on models. However, they are deemed ancillary.

Surprisingly, the theoretically inclined Giere has material models well represented in his brief enumeration of the types of models: “the things that are commonly called models seem to form a quite heterogeneous class including physical models, scale models, analogue models, and mathematical models” [22, pp. 746–7]. Giere attempts to understand models “in a way that usefully encompasses much of this heterogeneity” [22, p. 767]. Yet as we have seen previously, in focusing on theoretical models, Giere chooses to ignore a goodly portion of models. Stefanov most clearly demonstrates scholar’s relegation of material models. Stefanov chooses to forgo any lengthier discussion on material models as he deems the way in which they function to be “commonsense” [36, p. 69]. Despite scholar’s recognition that material models form a distinct part of the class of models, material models are often deemed of less scholarly interest. At best material models are treated as a trivial extension of theoretical models and at worst they are ignored.

The skewed representation of models in the literature led a group of authors to publish *Models: The Third Dimension of Science* in 2004, a treatment focused “not on abstract, mental entities, but precisely on objects that people grasped with their hands” [26, p. 2]. Throughout the chapters, scholars cover a variety of material models including wax medical models, casts of skin conditions, scale models of ships, models of electrical storms, mechanical hydraulic flow models, and 3-dimensional computer simulations. The treatment of these models is largely descriptive, documenting the people, places, and historical and cultural contexts of the models and their development. However, a focus on material models does not mean one must forgo a theoretical examination. Two concluding chapters investigate the ways in which the models function on a more theoretical, epistemological level. In line with Baird, the authors argue that 3-dimensional, material models challenge the prevailing conception of knowledge as comprehending the world through linguistic interpretation [26, p. 434].

3D digital models have been similarly relegated. *Models* represents one of the few instances in which 3D models have been mentioned, and in this case they are subsumed under the broader category of computer simulation. The dearth of 3D digital models does not stem from a lack of attention to the digital. To the contrary, computer simulations are well treated within the discussion of models. Rather, it is 3D digital model's proximity to 3D physical models that has relegated them, if not to oblivion, then at least to a minor role in discussions of models. Not only is the production of 3D digital models similar to that of scale models, i.e., the reproduction of form and color, but the two are interacted with in similar ways, leveraging human's intuitive understanding of manipulating objects in 3D space. While this has led some to dismiss both 3D digital and physical models, other scholars like Baird, Vrasich and De Chadarevian argue that it is the very materiality of these models which lends them their epistemological force. This fact is made exceedingly clear in the case of Vrasich, it was the very materiality of the 3D printed blocks which allowed for the generation of insight and the creation of new knowledge.

2.2.3 "Scientific" Fields

Models: The Third Dimension of Science also has a distinct methodological focus, limiting the examples of models discussed. *Models* is a study of 'scientific' models. What exactly makes a model scientific is not made explicit. Rather, *Models* seems to operate on the same implicit assumption of what constitutes science as much of the modern philosophy of science literature. *Models* discusses the use of models in western cultures beginning in the Enlightenment through to modern day. The fields of biology, physics, economics, and mathematics are the focus. These fields are representative of the disciplinary interests of the founders of the philosophy of science. Thus, even though compelling examples of theoretical models exist outside of these fields, like our example of the Antonine itineraries, these models have not received direct attention.

The foundation for the field of philosophy of science may be traced back arbitrarily far in the western tradition of philosophical thought. However, the field first explicitly began to take shape in 20th century Europe with the logical positivists. With the Vienna Circle especially, philosophers began to pursue a specific program bringing together epistemology and science with the dual interests of making philosophy more scientific and science more philosophical [37]. The majority of these scholars were not only philosophers, they were also physicists, economists, biologists, etc., with appointed positions at universities. And it was these fields that were of most interest to them. Later, philosophers of science began to more critically examine the distinction of science but the core of the 'hard sciences' has remained.

The traditional focus of philosophy of science on these fields is not problematic in general. Scholars have attempted to delineate 'proper' science but that is outside of the scope of this paper.[3] However, there do exist other fields in which models have been used to transmit and explain ideas between individuals and investigate, discover, and predict new things, fields that lie outside of the typical purview of the philosophy of science. These models may provide valuable insight on the topic and be entered into the current philosophy of science discussion in a constructive way.

2.3 Defining Models

Scholars in the philosophy of science have offered a number of definitions that seek to explicate the ways in which models may be identified and describe how they participate in the production of knowledge. While these definitions may bring some insight to the new instances of models just considered, the majority offer unsatisfactory roles for the materiality of objects.

2.3.1 Denotation, Demonstration, and Interpretation

R.I.G. Hughes presents an account of model design and use divided into three actions: denotation, demonstration, and interpretation (DDI). This is the account that Baird adopts in describing the function of models. However, Baird adds the addendum that materiality plays an important role, as materiality is offered no clear place in Hughes' original account. Hughes claims his account extends to material models but they are considered outliers, they are "the exception, not the rule" [28, p. S329]. 'Denotation' is a fuzzy concept but one that Hughes claims forms the core of representation and is independent of resemblance, wherein a model may be considered a symbol or some kind of referent to a real world system [28, p. S330]. Models 'demonstrate' through a dynamic that allows an individual to study the model and draw hypothetical conclusions [28, p. S331]. Finally,

[3] For examples see [38], [39], [33].

’interpretation’ is the numerous attendant theories, assumptions, and background knowledge utilized in generalizing the results of a model back to the real world system(s) under consideration [28, p. S333]. These three steps may be nested arbitrarily, allowing for complex hierarchies of models.

Hughes presents a very general account of models that has been widely cited. However, this generality leads to a lack of clarity. Denotation is based upon representation, itself a complicated concept whose issues we will return to shortly. Here it seems the issue of explicating the process of moving from a real world system to model has merely been provided an additional layer of definitions, without much additional insight into the process. Demonstration is based on an internal dynamic, but this too is unsatisfactorily vague. What is interacting? Are there static models as well? How might materiality fit into this scheme?

One of Hughes interesting propositions is the way in which these models may be nested in a hierarchically. In this description, an individual relies on a series of interrelated models ultimately traced back to a real world phenomenon. A similar structure is the focus of another definition of models, the semantic account.

2.3.2 The Semantic Account

The semantic account of theories (SCT) describes scientific theories as the set of models for which the theory is true. Since all models in the set satisfy the theory, they are logically equivalent in terms of ’reality’ or ’truth’. Ronald N. Giere has presented a notable interpretation of SCT. In Giere’s account, models are “artful specifications” of theories designed by scientists so that “elements of the model can be identified with features of the real world” [22, p. 747]. Giere proposes a mechanism of similarity that is used to create the representation relation between the model and the real world. Giere does not provide an objective measure for his similarity measure but relies on a naturalistic description, similarity is constrained by the biological and social factors of the observer.

The lack of materiality in Giere has been remarked upon previously. This is a shortcoming in our particular case, but not necessarily a serious flaw in Giere’s account. For Giere is self-avowedly interested in explaining scientific theory and how said theory is conceptualized. Baird would certainly argue that Giere is missing an important part of how scientists interact with, use, and discover theory but it remains to be seen if the material dimension is necessary for Giere's purpose. However, Giere's account of models is representative of the prevailing sentiment in the literature, one focused on theoretical models to the detriment of materiality.

2.3.3 Models as Agents

Knuutila critiques accounts like Giere's and Hughes' in that they place too great an emphasis on representation and do not properly encompass the various roles of models. Knuutila argues for a more "material and practical approach" [40, p. 1261]. Knuutila, convinced by Baird of the importance of the material dimension, sees the approach of Morgan and Morrison as the most fruitful avenue of investigation.

Morgan and Morrison presented their theory of models in their 1999 work, Models as Mediators: Perspectives on natural and social science. Morgan and Morrison describe models as autonomous agents, independent from theory and experiment yet able to intervene in both [23, p. 64]. Morgan and Morrison stress the function of models as tools, mediating between data and theory. In such an account models are afforded additional agency as independent entities, thus accounting for the behavior of models and their interactions outside of the constraining linguistic framework of the semantic account. However, Morgan and Morrison still rely on representation as an integral element of their account. A model is said to be explanatory and useful for prediction only insofar as it represents the real system [23, p. 64]. In this case, representation is considered a sharing of "certain kinds of structural dependencies" between the model and real world. "The model shows us how particular bits of the system are integrated and fit together in such a way that the system's behavior can be explained" [23, p. 63]. Knuutila claims that even here, representation is afforded too important a role in describing models.

Knuutila offers her own account of models focusing on models as agents with some independence from individuals. According to Knuutila, models may be most effectively described as artifacts, "intentionally constructed things that are materialized in some medium and used in our epistemic endeavors in a multitude of ways" [40, p. 1269]. Models are described with two components, a material sign-vehicle and an intentional representation relation of the sign-vehicle to that represented. But as with other authors, in describing models Knuutila resorts to vague language. What constitutes a sign-vehicle or how the representation relation is specified remain unclear. Knuutila's description of models ultimately appears quite similar to the approach of Baird. For Baird, Hughes provided the theoretical framework describing how models related to the things modeled. Then, Baird added the extra category of materiality and espoused its importance. Knuutila certainly recognizes the importance of materiality and attempts to incorporate it into her general account of models. However, we are left with a fragmentary and ill-defined result.

2.3.4 The Analogical Property Account

Perhaps the most promising account of models, at least as regards our effort of effectively incorporating materiality, stems from the work of Mary Hesse. Hesse proposed her analogical conception of models in 1965 and it has been developed by her and others throughout the last 50 years. Not altogether different than Giere, Hesse describes models as analogies, related to a real world system through a "relation of similarity and/or difference between a model and some theoretical description of the world" [19, p. 299]. These analogies may be positive or negative. Hesse does not explicitly describe the system through which such analogies are established but does hint at its elements. Hesse describes positive analogy as a sharing of properties between a system and a model.

Echenique explicates Hesse's account through a distinct ontology, wherein elements of reality are described using properties. An individual observes properties of the intrinsic reality resulting in a conception or extrinsic reality. The observer then translates this extrinsic reality to a model. A model is comprised of properties corresponding to properties of the real world system (positive analogies) and properties of the model not corresponding to properties of the real world system (negative analogies). Reference Figure 2.6 for Echenique's diagram of this relation.

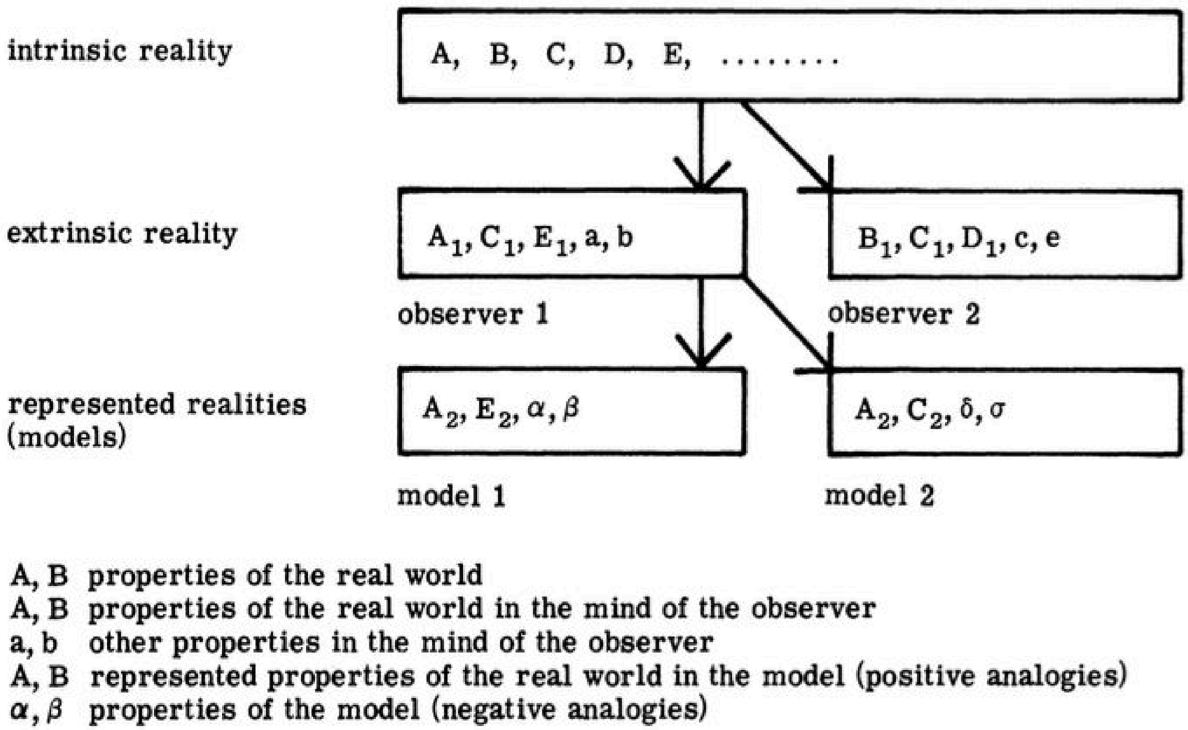


Figure 2.6: Echenique's diagram for the relation between reality, models, and observers [41].

Echenique's diagram shows clearly the relation between reality, model, and observer. However, the operative elements of this relation remain to be explained. How does one go from a property of reality, A, to a property of a model, A_2? What are the implications of representing these entities as properties? How do these properties relate to one another, both within an entity and across entities? Paul Humphreys offers just such an explanation.

Humphreys also builds from Hesse's definition of models and like Echenique identifies properties as a key component of the account. Unlike Echenique, Humphreys offers an explanation of how properties may be observed and interrelated, property cluster realism. According to Humphreys' property cluster realism, "properties are primary, both metaphysically and epistemologically" [42, p. 23]. The components of reality are then defined as clusters of properties. It is these properties that we observe and through which we classify entities. Properties themselves described causally according to the property localization principle. According to this principle, "each instance of a property possesses the entire causal force of that property" [42, p. 41]. Given this principle, a model is ensured to behave similarly to the thing modeled since the properties will interact with each other and external properties in a consistent way.

Finally, we have secured a role for materiality that does not present it as an outlier. The materiality of a model is (in most cases) a negative analogy and thus a property present in the model that is not present in the thing modeled. However, this property is fundamentally no different than the properties making up the positive analogies. Nonetheless, the material properties possess distinct causal force that leads to new behavior within the model not represented in the original system. In many cases, this new behavior serves as a benefit to the producers and users of the model. We may call this definition the analogical property account of models (APAM).

The APAM is not necessarily at odds with the previous definitions of models that we have covered. Firstly, Figure 2.6 begins to show a recursive structure not altogether different than Hughes. Certainly, instead of only drawing from 'intrinsic reality,' the modeler could use other models as referents for the new model created. Secondly, the APAM makes explicit the role of the modeler emphasized in Knuutila's conception of a model as an "intentionally constructed" artifact [40, p. 1269]. The modeler serves as an intermediary, observing properties of reality and then embodying those properties in the model. However, once embodied, these properties interact independent of the modeler according to their causal force. Finally, APAM provides a distinct representation of the potential separation between the model and the model, the result of which is an increased agency for the model itself.

2.4 Implications for 3D Model Use

APAM has a number of implications for the use of models and their description. Furthermore, these implications may serve to inform the development of tools and workflows for conducting research using models, thus seeking to ensure both a model's effective use and the applicability of findings back to the original. Contained in our new definition of APAM are our initial concepts from the dictionary definition of model whence we began, specifically representation and analogy. Postulates and data are also present, as they may be considered properties built into the model, data representing properties based on observation of the original and postulates representing properties added by the observer for the purpose of experimentation. Yet there remain some points in need of explication, namely the missing concepts from the dictionary definition and the potential of models for reuse.

2.4.1 Validity

Inferences and extrapolation can be roughly equated to Hughes step of 'interpretation', wherein the results of the model are generalized back to the original. In other words, how may we ensure that our findings reached using a model are valid in regard to the original? In APAM, the validity of the model rests on the reproduction of the causal force of the properties taken from the original and embodied in the model. Given property cluster realism, interactions observed between properties in a model will hold for those same properties if present in the original. Therefore, when providing justification for the validity of a model, one must be explicit in the mappings from properties of the original to properties of the model. Not only must one identify the corresponding pairs of properties but one must also make explicit the process through which these properties are first observed in the original and then embodied in the model. For example, in the case of the 3D prints of Pumapunku, Vrasich observed the three-dimensional form of the site through measurements then reproduced these measurements first digitally and then physically. Vrasich was clear in his justification for the accuracy of the embodiment of these measurements, describing in the detail the digital modeling process and physical printing process used. However, Vrasich does not provide description of the methods through which the initial observations were made. While this does not invalidate his findings, it does pose a significant issue in fully evaluating the validity of his model.

In the specific case of 3D digital models, this means that in an ideal case, the recording process, modeling process, and all aspects of the 3D geometry should be documented and inspectable. For the recording process, the means and sources used to measure the original model must be made explicit. As demonstrated in Vrasich, measurement may involve both direct processes such as laser scanning or the use of rulers as well as indirect processes using documents representing historical recordings and accounts of the original. Both cases involve an interpretive step which may bias or introduce error into the data, although in the case of laser scanning this error may be more accurately quantified. The modeling process, involves the use of specialized software and numerous technical operations, again both the tools and techniques used must be documented. Finally, the model itself must be inspectable including the vertices, texture coordinates, normals, textures, animations, etc. That said, full transparency throughout the model pipeline is rarely possible and, as shall be shown later, in some cases is only realized through compromises to security or other aspects of the visualization task.

2.4.2 Treating the Digital as Material

The greatest point of agreement between the varied definitions of models discussed is the heterogeneity of the category. APAM certainly encompasses this wide variety, providing the greatest flexibility for describing models comprised of myriad components. However, for our purposes here we are particularly interested in a certain type of model, 3D digital models. 3D digital models may be considered closely related to physical 3D models. In the parlance of APAM, both may be said to embody the three-dimensional properties of the object. The main difference between them is the negative analogy of their medium, in other words, their material. Unlike some previous definitions of models, APAM does not relegate materiality to a secondary status. But as argued by Baird, the epistemological function of material models is distinct.

Baird argues for two distinct aspects of material models. 1. Material models embody knowledge, transferring the knowledge from the modeler to the material medium, wherein the knowledge may interact in a way independent of the modeler. Baird terms the latter “working knowledge”. 2. Material models provide an entry point for a different sort of non-semantic “cognitive apparatus”. Given APAM, the first point no longer appears limited to material models, at least in the limited sense discussed by Baird of physical models. In effect, under APAM all models represent an external embodiment of concepts from the mind of the modeler, some based on observation of the original and some not. Thus, Baird’s concept of working knowledge may be extended to all instances of models in APAM. The similarities between Baird's physical models and 3D digital models are further demonstrated in regard to the second point of Baird. As Vrasich clearly showed the similarity between the cognitive similarities between 3D digital models and physical models. Namely, both allow for the application of human's documented ability to manipulate and visualize in three-dimensions, and the potential for this ability in generating insight. .

Thus, 3D digital models may be considered an extension of 3D physical models, differing only in their medium. As we shall see in the next chapter, 3D digital models and physical 3D models share not only functional similarities but also a history of use which may be traced back in the art historical tradition.

CHAPTER 3

HISTORY AND TECHNOLOGIES FOR 3D MODEL VISUALIZATION

3D models have formed a major part of the fields upon which VH is based and are quintessential to VH itself. 3D digital models in VH represent the most recent development in a long history of model use in art history, archaeology, and architecture. Enabled by recent developments in 3D visualization and the world-wide web, scholars in VH have extended the possibilities of 3D model use. As a scholarly field, VH is certainly reflective upon how to best make use of these 3D digital models, and questions about the veracity and usability of 3D digital models at times overlap previous methodological questions posed about physical 3D models. However, existing solutions for the visualization of 3D digital models in VH fail to address the entirety of these needs and issues. This is not to say that existing solutions have proven inadequate for scholarly research. To the contrary, scholars in VH have used current 3D visualization tools to produce numerous projects of academic value. Thus, the solution to the outstanding issues in visualization of 3D digital models in VH is not to consolidate and replace current tools but rather to fill the gaps through the development of complementary visualization tools, tailored to specific use cases of VH.

3.1 History of Model Use in the Humanities

Scholars in art history, archaeology, and architecture have made use of models throughout the modern history of the fields, 3D models especially. These fields are primarily concerned with physical objects and have engaged in a variety of ways of documenting and studying these objects, often involving the use of three-dimensional reproductions or recordings. Some of these methods, such as plaster reproductions through molding and hand modeled, scale reproductions, produce models as defined in Chapter 2. Furthermore, the use of these models by scholars represent the practical and methodological foundation for current work with 3D digital models.

Figure 3.1: *The Plaster Cast Collection in the Royal Academy of Fine Arts* by Julius Exner [43].

3.1.1 Plaster Casts

Artists and academics have used plaster casts of ancient statuary since the 16th century. A number of methods have been used to produce plaster casts but the end result is generally the same, a three-dimensional reproduction of an object's surface features in plaster. For an example of such casts, see Figure 3.1.

The Italian sculptor Leone Leoni provides an early documented use of plaster casts from the 16th century. On a visit to Leoni's home in Milan, Giorgio Vasari noted an impressive plaster cast of the Marcus Aurelius equestrian statue from the Campidoglio predominantly displayed on the façade [44, p. 235]. This cast was ostentatiously displayed as a testament to the skill and standing of Leoni. While it is likely that Leoni used plaster casts to further his work, no such documented occurrences exist.

The modern European sculptural tradition was enormously inspired by the extant Greek and Roman works. Sculptors were expected to study and emulate these pieces in their schooling and training. However, not every school was lucky nor wealthy enough to have ancient pieces in their collection, or even easy access to such pieces. This led the notable sculptor Bernini, during his stay

Figure 3.2: “En billehugger (Christen Christensen) arbejder efter levende model I sit atelier” by Wilhelm Bendz [45].

in Paris in 1665, to recommend the Royal Academy of Art in Paris to make casts of every ancient piece possible to further its teaching of sculptors. Plaster casts were not only used by sculptors in training, but also formed a part of established sculptors practice. The enthusiasm for casts, both as collectors items and didactic tools, grew into the 18th century.

In the 18th century, Étienne Falconet and Johannes Winckelmann offered their opinions on the reason for the value of casts among sculptors. According to the two, clean white plaster was one of the best mediums for displaying the form and beauty of a statue, qualities essential to the study of sculptors throughout their careers. Falconet even went so far as to claim that the plaster offered a more effective means of studying the aesthetic qualities of a statue than the original itself, as it makes the strengths and weaknesses of the artist clearer.[1] The aesthetic appreciation of casts was also present in the artists studio. Wilhelm Bendz illustrated just this process in his 1827 painting, shown in Figure 3.2. In the painting, the sculptor Christen Christensen is shown sculpting from

[1]See [46] for complete citations.

Figure 3.3: Photograph of The West Court of the Cast Courts of the Victorian and Albert Museum.

a male model in a studio filled with plaster casts of ancient works. Typically, a sculptor such as Christensen would “correct” his model according to the proportions of the ancients [46].

Casts found additional uses outside of the schools and studios of sculptors. Plaster casts offered museums and scholars the opportunity to create more complete collections and exhibits and perform invasive study that would otherwise be prohibited by difficult to access and fragile originals. As early as the 16th century, private collectors began assembling cast collections. Casts were made of objects of all sizes ranging from small statuettes to triumphal columns. During the 17th century, museums began amassing large collections, especially the Royal Academy of the Arts in Berlin. Through casts, museums were able to acquire seminal works that otherwise they would never have been able to purchase. This allowed museums to assemble large collections representing a great extent of the extant works. According to Borbein, such collections facilitated the creation of a new Art History, founded upon comparative study; these cast collections were largely responsible for the creation of the field of Art History as we now know it today [46, p. 9].

Cast collections went through capricious shifts in popularity, resulting in many collections and casts being lost or destroyed. However, casts have recently seen a resurgence in popularity. Cast collections have been recomposed and revived, reflecting much of the previous glory experienced by the collections in the 19th century. One such example is the Albert Museum shown in Figure 3.3. The Albert Museum exhibits ambitious casts of triumphal columns at a large scale. Casts offer additional benefits to modern scholars, so much so that Borbein was led to call them a type of "scientific preparation" [46, p. 17]. Casts allow scholars to more easily perform photographic studies and test reconstructions of fragments, statue groups, and polychromy. Without casts, such studies would be prohibitively expensive and/or destructive.

Frischer identified the major benefits of cast usage as *representation*, casts accurately illustrate the form of the original at a 1-to-1 scale; *reproducibility*, a cast may serve as the basis for additional reproductions; *portability*, casts may be more easily transported thus allowing for study in geographically disparate locations as well as the direct comparison and exhibition with other statues; and *experimentability*, potentially invasive and/or destructive procedures may be performed on the cast with no danger of damaging the original [47, p. 2]. Furthermore, in removing potentially distracting surface properties such as color and reflection, casts (at least according to Winckelmann) allow for easier comprehension of the form and shape of the original. However, casts are not free from drawbacks. Namely, plaster is a sensitive material and subject to degradation over time. On top of this, cast collections have been notoriously poorly treated and stored, at least in the 20th century. Finally, depending on the casting process, varying levels of accuracy may be achieved [48, p. 116]. Thus, one must be careful to ensure the formal accuracy of cast reproduction. That said, given access to the object casts are fairly cheap and quick to produce, given access to the original or another cast in proper condition, so the health of casts may be maintained through healthy reproduction.

Frischer is quick to draw a comparison between casts and 3D digital models, arguing that 3D digital models benefit art historical and archaeological study in just the same ways as casts, i.e. representation, reproducability, portability, and experimentability [47, p. 2]. Furthermore, the digital form greatly mitigates the physical degradation suffered by casts This is not to say that digital media is free from physical degradation. Ultimately, digital information is stored digitally, for example on magnetic tape or semiconductors, and these media are themselves subject to degradation and data loss. Additionally, environmental elements such as heat, humidity, and even solar radiation [49].

3.1.2 Architectural Models

Architects and archaeologists have made extensive use of another form of physical models in three-dimensions, architectural models. Historically, architectural models have been produced both at 1-to-1 and reduced scale. 1-to-1 architectural models fit within the art historical tradition of cast making previously discussed, used as a means of reproducing, documenting, and sharing architectural elements.

Beginning at the end of the 18th century, architects and scholars in Europe began systematically collecting casts of architectural elements, both individual details like leaves and volutes and large elements like column capitals and bases. Originally, such casts were primarily used for the training of architects in the reproduction and proper inclusion of such elements in their own designs. As with casts of sculpture, casts of architectural elements were collected by museums and schools in order to provide representative selections of ancient architecture and facilitate comparative study. The interest in architecture casts peaked during the 19th century, culminating with full scale plaster reproductions of Roman temples in Paris. But as with the casts more generally, architectural casts soon fell into disrepute [50].

Writers have remarked on the usefulness of scale models as early as 1929. Briggs published a duet of pieces in The Burlington Magazine for Connoisseurs documenting the use of scale models by European architects. It was common practice among European architects in the 15th to 17th centuries to produce small scale models of their commissioned works in plaster, wax, wood, or even brick and mortar [51]. Models survive from architects including Brunelleschi, Antonio da Sangallo the Younger, Michelangelo, and Wren [52, p. 174]. The earliest is a small wooden model by Wren of the chapel of Pembroke college from 1663 [53, p. 246]. These models were used by architects to coordinate subordinates and communicate plans to investors and patrons. The prevalence of such models is testament to their usefulness. Sir John Soane had a large number of models produced in the 18th century, which at the time would have cost an exorbitant amount [53, p. 252]. Briggs posits that this cost was justified by the increased effectiveness that these models brought when communicating plans to workmen and investors.

This usage of scale models would fall outside of our definition of models from in Chapter 2, as representations of already existing things. Nonetheless, these scale models and their use form a part of the didactic tradition treated here utilizing material mediums, These architectural models demonstrate of how information and understanding is transmitted through a material medium, in a way providing increased effectiveness over linguistic methods. In the case of an aging Michelangelo and St. Peter's, the models were used even after the artist's death [52, p. 183]. Thus, these models may communicate in a way at least partially autonomous from the individual producing them.

There was a tradition of model making more in line with Chapter 2, and whose practices largely reflect the later applications of 3D models described in Section 3.2. Beginning in the 18th century, scholars have been producing scale models based on existing Greek and Romans ruins. These models, produced in plaster, cork and wax, were used to present both the "picturesque" current state of ruins as well as novel reconstructions reflecting their classical ideal [50, p. 424]. According to the standards of the time, these scale models were meant as exact reproductions and reconstructions based in scholarship. With these scale models, monumental architecture was reduced in scale, "rendering it accessible for everyone", as long as one had access to the museums and private collections housing these objects [50, p. 422] Most often, these models were held at teaching institutions and used to demonstrate the architecture as a whole and the relations between its elements [50, p. 427].

Scale models were also used in a less academic setting to show fortifications of cities. When planning the siege of Florence in 1529, the pope supposedly used a scale model of Florence and its fortifications produced by Tribolo [52, p. 180]. The pope was able to use this model to direct the siege from afar. Louis XIV also made extensive use of models of city fortifications, commissioning some 50 models during his reign. The models were produced with exceptional attention to detail and included a large area surrounding the towns [54].

3.2 3D in the Humanities

The use of 3D digital models in the humanities parallels the historic use of scale models and plaster casts. While 3D digital models have opened new avenues of research, they have also been incorporated into the previous practices of scholars. As argued in Chapter 2, it is primarily the material medium in which these models are conveyed that prescribe their usefulness, but in sharing material embodiment they also share the potential for 3D manipulation.

Hstorians and archaeologists began using 3D digital visualization as early as 1992, when Japanese researchers produced a 3D digital visualization of a mummy using CT scans [55]. However, many of the earliest 3D visualizations required bulky and expensive hardware, in some cases costing over a quarter of a million dollars [56, p. 28]. Also during the 90s, the world-wide web was identified as a means of making these inaccessible 3D visualizations accessible. With the earliest attempt to bring 3D graphics to the web, VRML, archaeologists were making use of 3D visualization on the web [56]. In recent years, 3D models have experienced a boom in scholarly publications. Roosevelt et al. describes this phenomenon as a "paradigmatic shift" of the archaeological community's wider adoption of 3D technology as a viable means of scholarship, driven by the increased performance and cost effectiveness of both hardware and software components involved in producing 3D models [5].

3.2.1 A Brief Intro to 3D Computer Graphics

In their 2014 survey of 3D computer graphics for the web, Evans et al. provide the following definition:

> "We define 3D graphics to be the use of 3D geometric data (usually through Cartesian coordinates) to perform certain calculations (for example, changes of form, animation, collision detection, etc.) and to create 2D images suitable for display on a standard computer screen or monitor."

Evans et al. describe the central process of computer graphics as producing images, albeit with the added limitations of those for "standard" displays.

While the tasks involved in 3D computer graphics are varied and may be accomplished through

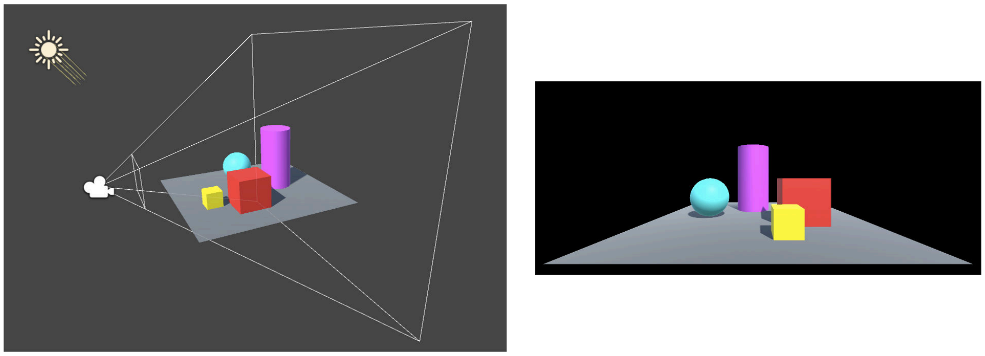


Figure 3.4: An example scene with camera, light, and 3D objects (left) and rendered image (right) using the synthetic camera model.

equally varied means, certain conventions prevail throughout the field. First, the vast majority of graphics applications are conceptually based on the "synthetic-camera model". This model seeks to describe synthetic image formation in terms analogous to physical image formation and the human visual system. In the synthetic-camera model, image formation is considered as a projection of a three-dimensional scene onto a camera (see Figure 3.4). The process may be partitioned into three steps: the mathematical description of individual objects or models, the placement of models relative to one another in space to specify a three-dimensional scene, and the projection of the said scene to a camera thus creating a rendered image.

In the modeling step, models may be defined explicitly or implicitly. Explicit definition is most often done as a surface, using vertices (points in space) and edges connecting vertices or as a volume, using voxels. Implicit definition is most often done using parametric equations such as NURBS. In this work, we will primarily concerned with the surface definition. The vertices of a surface may be complemented with normals, used for the computation of lighting equations and texture coordinates, used to map a 2D texture to a 3D surface.

In the scene definition step, models are positioned in space relative to one another, typically using matrix operations to scale, rotate, and translate the models. The relationships models have to one another are often described hierarchically in a structure called a scene graph.

Lastly in the rendering step, a projection of the scene is calculated, again using matrix algebra. The most common projections are perspective, which mimics the foreshortening we in real imagery,

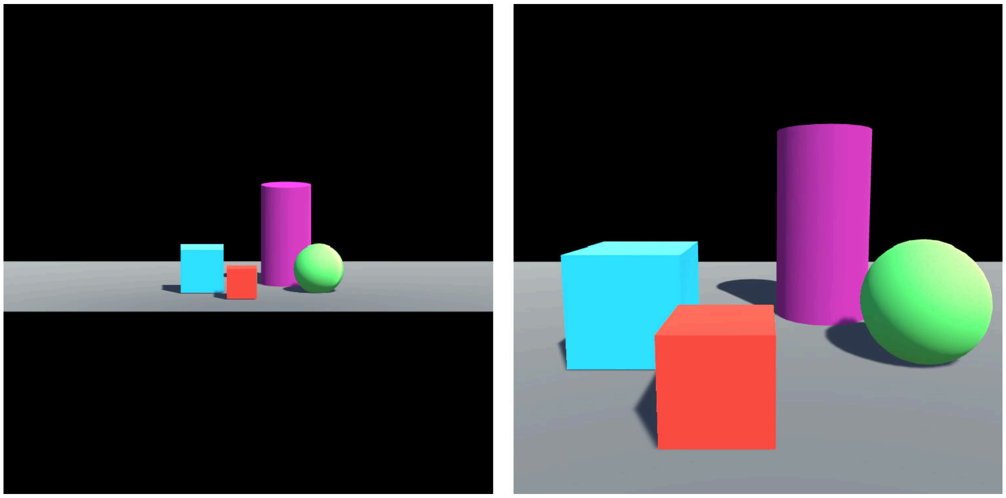


Figure 3.5: The same scene rendered with an orthographic projection (left) and a perspective projection (right).

and orthographic, which preserves relative distances and angles of the rendered objects. See Figure 3.5 for examples.

3D computer graphics involves large amounts of matrix operations and geometric calculations. To assist these calculations, most modern machines have dedicated graphics hardware called Graphics Processing Units (GPU). Programs may make use of the GPU to provide hardware acceleration to graphical applications through the use of programming APIs such as OpenGL or DirectX.

3.2.2 3D Modeling

While there exist efforts to image the internal structure and material properties of objects using technologies such as CT [55] and X-ray [57], the majority of 3D visualizations in VH focus on surface reproduction of objects. This task is accomplished either automatically using scanning or manually using 3D modeling software.

3D Scanning

A wide variety of 3D scanning technologies exist, reproducing 3D structure based of off various input. The most commonly used 3D scanning techniques in VH are laser scanning and photogrammetry.

A wide range of laser scanners exist, varying in both cost and accuracy. At the low end are devices like the Micorosft kinect, which make use of projected infrared patterns, calculating the distortion of the pattern when projected onto a 3D surface. The low price of the Kinect led to its wide use in publications. However, the accuracy can range from several millimeters to several centimeters, making the Kinect a poor choice for applications requiring strict metrical accuracy [58]. High end scanners, like those produce by Artec [59] or Leica [60] often use time-of-flight to record depth information, measuring the time from the light emission by the scanner to detection by its receiver. Such devices may easily cost tens of thousands of dollars but also provide sub-millimeter accuracy. Laser scanners may only directly measure depth. However, these scanners may be supplemented with photographs used to generate textures for the models.

Photogrammetry reconstructs a 3D object using correspondences between photographs. First, easily identifiable features within photographs are identified. Next, corresponding features between photographs are found. Once a significant number of correspondences have been detected, the position of the camera and the three-dimensional location of the feature relative to the camera may be calculated. The final result of a photogrammetric algorithm is a three-dimensional point cloud, where each point has associated color information. The point cloud may then be converted to a watertight model through the use of triangulation algorithm. An example of a photogrammetric workflow is show in Figure 3.6 Originally, photogrammetry was an incredibly time intensive process, requiring the manual input of correspondences between photographs. Initial attempts at producing an automatic software workflow partially simplified the process but still required extensive training and technical knowledge to operate [61]. It has not been until the last five or so years that photogrammetry has become popular more broadly in the field, largely due to the release of user friendly commercial software such as Agisofts Photoscan[2] [62] and Reality Capture [63]. Using photogrammetric software, a consumer level camera, and a computer. one may now easily produce detailed 3D digital models with rich textures. Similar to laser scanning, photogrammetry

[2]Photoscan was rebranded to Metashape in 2019.

may produce models with sub-millimeter accuracy. However, this accuracy is largely dependent on the quality of the original photographs, properties of the material, and the environment in which the object was photographed [64] [65]. However, the high fidelity textures produced by photogrammetry make for visually detailed models, even when the geometry is reduced. For these reason, photogrammetry has proven especially popular for public facing projects, given the visually striking models one may produce.

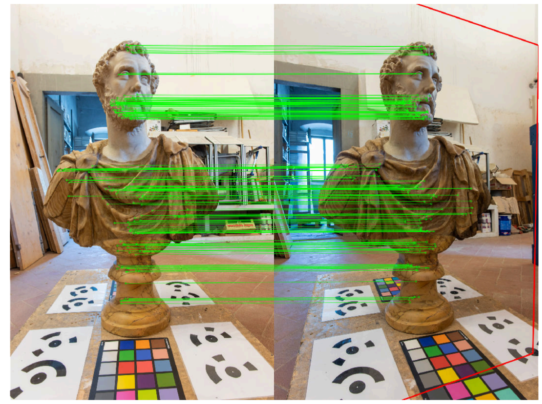
Hand Modeling

In contrast to recording physical objects, scholars may choose to produce models digitally using 3D modeling software. Open source and freemium software may be used such as Blender [66] and Sketchup [67], or more expensive and powerful proprietary software like Zbrush [68] and 3D Studio Max [69]. Hand modeling may be desirable in cases such as Vrasich when representing geometrically simple objects. Additionally, hand modeling may be used to create potential reconstruction of partially extant or non-extant objects.

3.2.3 File Formats

The digital formats used to work with and share 3D models are as numerous as their applications. Digital formats may be divided between open source solutions, which are documented publicly, may be easily incorporated into new software solutions, and serve as exchange formats facilitating the transfer of models between different software, and closed source solutions, which require specialized software to work with. Closed source formats are used by proprietary software as efficient representations for manipulation within the program. Open source is greatly preferred in academic contexts, as it is more readily shareable and its preservation does not hinge on the continued access to proprietary software. However, digital conservators have had success using emulation to continually support out of date software [70]. So while using open source file formats is best practice, the dangers of proprietary formats have been significantly lessened. This plus the power and robustness of proprietary 3D modeling software packages has led to the continued

(a) Input images.



(b) Corresponding features (green) and orientation (red) between two images.

(c) Two views of the final model rendered with and without texture.

Figure 3.6: An example photogrammetric workflow from the initial photographic input to the final point cloud.

quad.off	quad.obj	quad.ply
OFF 4 2 0 1.0 1.0 0.0 1.0 -1.0 0.0 -1.0 -1.0 0.0 -1.0 1.0 0.0 3 0 2 1 255 0 0 3 2 0 3 0 0 255	mtllib quad.mtl v 1.0 1.0 0.0 v 1.0 -1.0 0.0 v -1.0 -1.0 0.0 v -1.0 1.0 0.0 usemtl red f 1 3 2 **quad.mtl** usemtl blue f 3 1 4	ply format ascii 1.0 element vertex 4 property float x property float y property float z element face 2 property list uchar int vertex_index property uchar red property uchar green property uchar blue end_header 1.0 1.0 0.0 1.0 -1.0 0.0 -1.0 -1.0 0.0 -1.0 1.0 0.0 3 0 2 1 255 0 0 3 2 0 3 0 0 255

Table 3.1: OFF, OBJ, and PLY files describing a quadrilateral composed of a red triangle and blue triangle. The rendered shape is shown in Figure 3.7.

prevalence of proprietary formats among academic projects, at least as intermediaries for processing. In nearly all cases, at some point in the lifecycle of a model it will be converted to one of the open formats.

[71] presents a comprehensive overview of the available 3D file formats and software as of 2008. Certainly, 3D software has advanced considerably in the last 10 years. However, the vast majority of 3D file formats and software remain relatively unchanged. glTF is the most notable development, a new exchange format developed with the purpose of becoming a standard. A selection of the most common file formats are discussed below. See Table 3.1 for simple examples of the discussed formats. For explanation of the 3D graphics terminology used in this section, reference Section 3.2.1.

OFF

.OFF is one of the simplest 3D file formats storing polygonal faces in the form of vertices and edges and vertex color [72]. .OFF files begin with a header specifying the file format and a triple

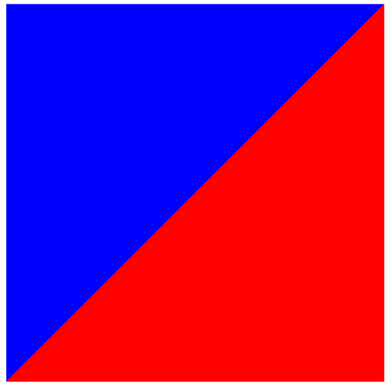


Figure 3.7: A render of the shape described in Table 3.1.

specifying the number of vertices, the number of faces, and the number of edges (although this value is always zero), followed by a list of vertices and a list of faces. Vertices are specified with x y z coordinates and faces are specified using vertex indices, beginning at 0.

OBJ

The OBJ file format, originally developed by the Hollywood animation company Wavefront, has become a near ubiquitous feature within 3D graphics. The prevalence of .OBJ is largely due to the simplicity of the format. OBJ stores vertices, faces, uv coordinates, and per vertex color in plaintext. A key-value pair is stored on each line of the file, with *v* indicating a vertex, *f* a face, *n* a normal, etc.[3] .OBJ represents polygons with an arbitrary amount of edges using vertex indices. OBJ files may not directly store materials. Instead, OBJ files store references to materials specified in an additional sidecar file of the MTL format.[4] While the plaintext, key-value storage of .OBJ files make them human-readable, it does result in one of the more verbose storage schemes for geometry. Additionally, the separation between the geometry and material files can also be an inconvenience for users.

[3]see [73] for the full OBJ specification.
[4]See [74] for the full MTL specification.

PLY

The Polygon File Format (PLY) was developed by Stanford to store models generated using 3D scanners. Two versions of the format exist, a plaintext and a binary. PLY files begin with a header identifying the file as PLY and specifying the format (plaintext or binary). PLY files are meant to be easily extensible. Each .PLY file defines a set of 'element' data structures. Each 'element' has a list of 'property' values. Elements, their properties, and the number of each type in the file are also defined in the header.

glTF

glTF is a recently developed file format used to specify 3D scenes using JSON [75]. In addition to geometric information, glTF may encode cameras, lights, and animations. Furthermore, unlike OFF and OBJ, glTF may encode complex material and rendering properties. glTF is developed by the Khronos Group and is intended as a new standard for the transmission of 3D scenes.

COLLADA

COLLADA files (COLLaborative Design Activity), stored with the DAE extension, are another file format meant specifically for the interchange of 3D graphics scenes stored using an XML schema. COLLADA is a widely supported ISO standard, which like glTF supports far more than just 3D geometry such as animation, rendering effects, data validation, and physics [76].

3.2.4 Sharing on the Web

In order to fully realize the portability and reproducability of 3D digital models, scholars have argued for the creation of online repositories to facilitate the sharing and access of scholarly 3D content. Online collections free the physical object from its geological constraints, allowing for ease of access across the globe as well as the comparison between objects that may never be physically in the same location. In VH, online collections are seen as a necessity in accomplishing the goals of academic study and wide access sought by 3D digitization projects [77] [78].

3.3 Web3D

In order to enable the transparent sharing of 3D content on the web, repositories of scholarly 3D digital models may make use of technologies for the visualization of interactive 3D content on the web. Web3D refers to interactive 3D content displayed on the web. Web3D features distinct challenges for application development that have shaped the trajectory of research. To support interactivity with Web3D content, an application must be able to respond to user input and render the scene at a sufficiently high frame rate, typically 30 frames per second (fps) or higher. This requirement imposes limits on the calculations and rendering techniques able to be employed. Running such applications on a network further complicates the issue. Web applications are subject to potentially poor connections and, given the cross-platform implementations of web browsers, are often required to be run on a heterogeneous collection of devices with differing processing power. These constraints must be dealt with in such a way as to maintain the necessary fps for interaction. Despite these limitations (or perhaps in spite of them) significant energies have been devoted towards bringing 3D to the web.

3.3.1 VRML to WebGL

The first concerted efforts to create Web3D were began in 1994 by David Raggett and Tim Berners-Lee. The two sought to actualize a distinct vision of 3D on the web largely reminiscent of the "cyberspace" presented in the science fiction novels of Gibson and Stephenson, a virtual reality ecosystem that would allow users represented as virtual avatars to "walk" between web pages [79]. The file format that was proposed to accomplish this, VRML, was implemented and demonstrated shortly thereafter in 1995 [80]. In 1996, a more refined version, VRML97, was released and quickly accepted as an ISO standard in 1997 (ISO/IEC 14772-1:1997) [81]. Using VRML97, one may define a scene composed of 3D objects described using sets of points in Cartesian space with accompanying materials, lights, animation, and user interaction. Plugins were developed to render these files within a web browser including Cosmo Player, WorldView, and VRMLView [82].

The properties of VRML97, i.e. the use of a scene graph, highly abstracted syntax, and the requirement for plugins, would define Web3D for the better part of the following decade. Various rival technologies to VRML were introduced such as Macromedia Shockwave, Flash Player, Java3D, and O3D. In 2001, VRML itself was superseded by the new X3D format, which among other improvements added support for parametric model descriptions and a clearer syntax.

While the languages and software involved in producing 3D content varied from technology to technology, all used plugins to render. Plugins allowed direct access to the graphics hardware of the client machine and freed graphics programmers from the constraints of web-based programming languages. However, plugins often required lengthy downloads and administrator privileges for installation. These drawbacks were recognized at the time and were attributed by some as culprits for Web3D's lack of popularity [83].

The next significant change came with WebGL, version 1.0 was officially released in 2011. WebGL, a JavaScript graphics API based on OpenGL ES, allowed for hardware accelerated 3D graphics to run natively in the web browser. WebGL offered significant performance gains over previous methods and freed Web3D from the lengthy downloads, sporadic support, and security concerns of plugins. However, the low-level approach of WebGL required significant programming skills. This drawback was partially ameliorated by the contemporaneous release of a number of high level JavaScript libraries for WebGL including SpiderGL and Three.js [84].

3.3.2 Current State

As Web3D has developed, a rich and complex ecosystem of technologies has emerged to support the process. As of 2010, Ortiz described Web3D as in high demand but not fully realized. Ortiz attributed this disappointing development to five factors:

1. The need for plugins
2. A lack of standardization
3. Long authoring times of 3D content
4. Lack of processing power and network bandwidth
5. Lack of online 3D technologies targeted at the average users [85]

WebGL provided a solution to several of these hurdles that would lead to large increases in 3D content on the web. The benefit of WebGL was quickly realized throughout the Web3D community. Over the four years following its release, WebGL and its component libraries were used to develop complex graphical applications for the web. Additionally, WebGL was combined with previous techniques such as X3D. The result, X3DOM, used WebGL to provide native support for X3D in the browser. The proliferation of Web3D content led Evans et al. in 2014 to describe the community as "vibrant and exciting, both in the academic and wider developer communities" [86, p. 59]. All but one of the hurdles identified by Ortiz seemed to have been addressed, the only outstanding being standardization. Per Evans et al.'s analysis, Web3D applications seemed largely split between X3DOM and WebGL JavaScript libraries, particularly Three.js.

Over the last four years, the prevalence of native 3D graphics on the web has only continued to increase. Evans et al. considered the reluctance of Safari, the default browser for all iOS devices, to adopt WebGL one of the major remaining stumbling blocks for Web3D [86, p. 58]. However, the same year that Evans et al. made this observation, Safari announced that it would add support for WebGL [87]. Now, WebGL is supported in all major browsers, including those for mobile [88]. Furthermore, support for 3D graphics on the web has expanded to capture the heterogeneity of modern internet enabled devices [89]. Stereoscopic displays and virtual reality now run on the web.

The increasing prevalence of Web3D content may be seen in the growing number of high-profile websites that make use of such content. The New York Times has emerged as a forerunner in pushing the boundaries of web content. The digital publication now features weekly, 360 videos and often incorporates three-dimensional content into their online articles [90]. Other large sites have made intermittent use of 3D content. YouTube [91], the digital payment platform Stripe [92], and Playdoh [93] have all featured interactive 3D content on their websites. Additionally, websites centered entirely around 3D content are becoming increasingly popular. Sketchfab, a site devoted to the viewing and sharing of 3D models on the web, recently surpassed one million users [94]. The vibrant community remarked upon by Evans et al. has only continued to develop over the intervening years, in part thanks to the increasing support of WebGL and plugin-free graphics.

While the popularity of Web3D may or may not have reached the levels expected of it, the usefulness of 3D content has been definitively shown. The prevailing wisdom, that three-dimensional content allows for improved engagement and easier reasoning, has been largely corroborated through scholarly study [95] [96] [97]. Scholarly support for Web3D initiatives has remained strong through well attended technical conferences on both the web more generally and Web3D specifically as well as enthusiastic publishing on the topics [98] [99].

3.4 Features for a VH 3D Viewer

Using Web3D technology, it is possible to provide in-browser viewing of the contents of 3D repositories or individual objects. But before addressing existing viewers, let us discuss more generally the features that scholars in VH desire and need in a 3D model viewer. The features identified here are based on a survey of the scholarship pertaining to 3D models in VH and their viewing. These features will be further investigated and compared to practices and opinions solicited from VH researchers in Chapter 5.

3.4.1 Technical Features

VH scholars tend towards certain file formats and model resolutions. Typically, VH scholars have used STL, PLY, and OBJ file formats [100]. Additionally, in the pursuit of scholarly rigor and metrical accuracy, models in VH are often large and high resolution, featuring millions of polygons [100] [101]. While scholars have found ways to reduce the size of their models, the loss of resolution often forgoes the potential for metrical analysis. Ideally, a viewer would be able to support models of this kind, thus maximizing convenience for scholars working in the field and the potential for integrating past work.

3.4.2 Showing Uncertainty

As objects of scholarly study and often times the subjects of peer-review, 3D models in VH must document uncertainty inherent in the model [100, p. 44]. Uncertainty may be introduced at any point in the modeling processes, from 3D data acquisition to the use of reference drawings to features of a final virtual reconstruction model. This issue has been identified as integral to the academic value of 3D models in VH and has been codified in international charters meant to inform 3D work in VH [102] [103] [78, p. 12]. Specifically, both the Ename Charter and London Charter make explicit that sources for 3D modeling should be evaluated in a rigorous and documented way [104] [105]. Richards-Rissetto and von Schwerin identify archaeological data sources, other data sources, and paradata as essential to accurately assess a model's uncertainty [100, p. 44]. Various methods for conveying uncertainty have been developed including color coding [106], hyperlinks to relevant documents [107], and text annotations in 3D space [108]. At minimum, a VH viewer should incorporate at least one mechanism for conveying uncertainty as well as a means of linking relevant sources and documentation of the modeling task.

3.4.3 Semantic Annotation

As can be seen in the previous section, 3D models represent interrelated constellations of data, metadata, and media. The information contained in these sources and the relations between them may be leveraged according to the principles of the Semantic Web. All of these components, but most importantly the 3D model itself, may be semantically annotated by converting the vocabularies and ontologies already used in the field to describe them into machine readable representations. In this way, objects may be more effectively indexed and searched, more fully capturing the semantic interrelations already present in scholarly research. Semantic annotation may be performed manually or automatically through the use of algorithms that may identify features of 3D digital models [109] or measure similarities between models [110]. The major part of the work required for semantic annotation falls on the back-end of the system in the form of model processing, search algorithms, etc. However, the benefits of semantic annotation may be more fully realized through its incorporation into model viewing, both in the visualization of semantic annotations (e.g. the components of the model or sources and their corresponding semantic values) and the ability of manual input. While automatic techniques for semantic markup are being developed with increasing sophistication, they may still benefit from the unrivaled erudition and domain knowledge of professionals in the applicable field.

3.4.4 Exhibition, Contextualization, and Comparison

VH is at heart a branch of the more traditional fields of archaeology, history, museum studies, and art history. As such, it adopts many of the methodologies of these fields. Comparative study, contextualization, and exhibition of cultural heritage artifacts are all important methods of study. A viewer should be able to provide tools that enable these traditional methods, allowing for the viewing of multiple objects, positioning objects in space relative to one another, metrical analysis, etc. In other words, viewing should not be limited to simple inspection. A viewer should offer support for complex extension to further the work of scholars.

3.4.5 Computational Analysis

Along with more traditional examination methods meant to parallel physical techniques already employed by scholars, a VH viewer may incorporate more computationally intensive analysis techniques not possible with purely physical media. Such techniques may be applied to individual models, collections of models, or features extracted from models.

Photogrammetric models of structures have been shown to be useful in structural analysis, determining the current conservation state of the structure as well as points that may be prone to failure. Most simply, 3D models may be used as tools for locating and measuring cracks and structural displacements indicative of shifting architectural elements and underlying stresses [111]. This work may be taken further by using algorithmic techniques like mechanical modeling tools and finite element analysis, performing simulations to analyze the forces acting upon the structure and the potential for structural failure [112].

Computational techniques may also be used to derive mathematical representations of objects and perform search and classification algorithms on these representations. Ceramics have been a popular area of application for such techniques. Both Biasotti et al. and Smith et al. use a variety of mathematical functions to describe the surface curvature and profiles of ceramics [113] [109].

These mathematical representations of the ceramics may then be used for classification and search algorithms, producing results that the authors argue align with common archaeological sense.

Agus et al. apply feature extraction techniques to a data set comprised of 3D models of protohistoric Mediterranean sculptures. Agus et al. argue that while automatic feature extraction, such as curvature-based saliency, are lacking in the case of archaeological objects, especially physically degraded objects [3, p. 3]. Rather than rely on automatic techniques, Agus et al. analyze user interaction with the models, focusing on the areas of the models that users spend the most time examining and thus, one may assume, they find the most interesting. Finally, Agus et al. argue that these features of interest, derived from human interaction, are more reliable and meaningful than the typical automatic methods [3, p. 10].

Inter-object comparison ultimately requires extensive organizational and computational backend support. However, the results of such computation are often made most meaningful when presented for inspection by individuals with domain specific knowledge. Thus, not only should the ability to perform such methods be present in a viewer, but so too should methods for display and analysis of the results of such methods.

3.4.6 Security

Content owners and producers may desire control of the use and dissemination of their 3D models for a number of reasons: 3D models may be expensive to produce, objects modeled may be copyrighted, or the location or cultural value of objects may need to be protected. Multiple scholars have identified the protection and security of 3D data as a key component of any 3D repository or viewer [78]. In some cases, simply limiting the viewing of particular objects to certain user groups may be sufficient. But often, this is an unnecessarily limiting method, discouraging the dissemination of valuable artifacts. Ideally, it would be possible to provide partial and/or protected viewing to certain objects. Remote rendering has been shown as a promising solution for provide secure interactive 3D graphics and may be complemented by additional security technologies like data encryption and watermarking.

3.4.7 Accessibility

Accessibility is important in two regards. Firstly, in providing access regardless of geographic location, online collections represent a democratization of knowledge [77]. As such, the viewer itself should be equally widely supported. With WebGL and native Web3D, this point is no longer a pressing issue, but still worth mention. Secondly, as will be elaborated Chapter 5, scholars within VH come from a diverse range of backgrounds with an equally diverse range of skills. Thus, a 3D viewer for VH must provide both easy use for those with limited technical skill as well as more sophisticated access and control to the technical functions of the system [114].

3.5 Current Web Viewers in VH

Scholars in VH have historically made use of several 3D viewers, among them are Sketchfab, Unity, X3DOM, and 3DHOP.

3.5.1 Sketchfab

Thanks to its simplicity, infrastructure, and interoperability, Sketchfab has become one of the most widely used platforms for 3D digital models of cultural heritage objects on the web. Sketchfab allows users to upload 3D digital models in over 50 different formats along with accompanying texture and material files [67]. These models are then viewable in a WebGL based viewer that may be easily embedded directly into source code of web pages or on supported platforms such as Tumblr and Facebook [115] [116]. Sketchfab supports basic object interaction of rotation, zooming, and translation; the application of camera filters for post processing; model animation; 3D annotation; and manipulation of rendering parameters like geometric primitives and materials. Additionally, Sketchfab associates Markdown descriptions to each model. These features have made Sketchfab advantageous for wide dissemination of 3D models and the annotations and descriptions have been used to provide contextual information, supporting documents, and label uncertainty on the model. For these reasons, Sketchfab is now used by over 500 cultural institutions including The British

Museum, The Metropolitan Museum of Art, and The RMN – Grand Palais [117]. The British Museum is an especially prolific user with 243 models uploaded to the service. Sketchfab stores and hosts these models on their own servers, offering the ability to search and sort uploads to the service [8]. However, Sketchfab's suitability for scholarly applications has been questioned. Sketchfab offers limited support in terms of model analysis and little customizability in terms of interaction. Furthermore, the storage of data on Sketchfab's servers can pose issues for data privacy and security [114]. Regardless, Sketchfab has become a standard for cultural heritage institutions seeking to display and disseminate 3D models and its simplicity and interoperability may serve as examples for the development of future viewers.

3.5.2 Unity

Unity is a game engine and graphical development environment which may be used to develop 3D and 2D applications for over 25 different platforms [118]. Unity allows for the easy development of life-like and immersive environments supporting three-dimensional sound, physics, complex rendering, and lighting. While the more advanced features of Unity may require extensive coding and application specific knowledge, a vibrant developer community has emerged around the software with numerous resources like tutorials and forums [119]. Along with mobile, desktop, and VR platforms, Unity may also build directly for the web with WebGL. Unity is one of the most widely used technologies for visualizing 3D VH models, especially when producing navigable environments like virtual exhibitions or architectural visualizations. Primarily, these applications have been deployed as standalone desktop applications or VR experiences.

Recently, Unity has been used to create VR tours of historic monuments [120], complete virtual galleries [121] [122], and installations for physical galleries [123][124]. A portion of scholars have also deployed applications to the web. Kiourt et al. developed DynaMus, a framework for the development of 3D virtual museums on the web [125] [126]. Curators may build interactive 3D exhibits in virtual galleries, using multimedia drawn from external sources like Europeana or Google Images. 2D images may be loaded automatically into the virtual gallery, while 3D objects must be added manually when creating the space. Kiourt et al. chose Unity for its “low cost, rich functionalities, user-friendliness, fast development, cross-platform delivery and powerful coding and database connectivity capabilities” [125] Unity has also been used to present architectural models on the web. Wardijono et al. created a model of the Indonesian National Monument in Jakarta, hosted on the official web page for the monument [127]. More interestingly, Frischer et al. used Unity as a testbed for astronomical simulations pertaining to the symbolic relationship between the Montecitorio Obelisk and the Ara Pacis of Augustus in Rome [128]. These interactive simulations were embedded within the online publication using Unity for the web. The user is able to control the astronomical parameters, view the scene from multiple angels, and navigate the environment in a first-person view or an orbit view.

Unity’s quick development and game engine features make it an effective tool for producing interactive environments, especially when one desires to deploy to multiple platforms. Nonetheless, Unity has inefficiencies and drawbacks that have led authors to question its use as single model viewer. Given the numerous physical calculations Unity computes as a game engine, it quickly slows with larger geometries. This issue is further compounded when applications are built for the web. Additionally, Unity WebGL builds feature a number of bugs resulting from unsupported C# code, like threads and try/catch blocks, often requiring ad hoc fixes [129]. Further development is needed for Unity to be a truly useful tool when deployed on the web due to code refactoring. For these reasons, Unity has been identified as complementary to single model viewers, useful for the exploration of modeled scenes or immersive environments [114, p. 131].

3.5.3 X3DOM

X3DOM is a JavaScript library and open-source framework that, through WebGL, provides native support for X3D in the web browser [130]. X3DOM offers declarative 3D graphics fully integrated with the HTML document. It is widely used among the research community but has failed to gain traction more widely among online developers of Web3D [86]. X3DOM may succinctly add single 3D models to webpages with basic interaction and easily configurable lighting and viewing parameters. Additionally, X3DOM has a built-in solution for large geometries [131]. Scholars have argued that X3DOM offers a desirable means of representing 3D cultural heritage on the web with primitive exploration, dynamic walkthroughs, and on-site AR experiences [132]. Similar to the work of Kiourt et al., Wilkosinka et al. use X3DOM to develop a framework for the implementation of online virtual museums [133]. In this endeavor, X3DOM provides a distinct advantage in that it may be more easily connected with other elements and media on the web. However, it does not provide the extensive built in physics and rendering support available in Unity. X3DOM is not easily extensible, lacks comprehensive documentation for developing more complex applications, and has been shown to perform slower than alternative techniques for rendering large geometries [114]. Thus, it is an imperfect solution as a VH viewer.

3.5.4 3DHOP

3DHOP is a declarative style, 3D model viewer specifically developed for cultural heritage applications. Using progressive mesh streaming and level-of-detail techniques it is able to provide an initial rendering of enormous models (over 100 million polygons) in seconds. The rendering is iteratively improved and this iteration may result in slower times to download the entire geometry. 3DHOP has been incorporated into existing online databases of 3D models like the Archaeological Data Service [134] and Europeana [135]. 3DHOP was developed to leverage the benefits of the declarative approach and, as with X3DOM, incorporate fully with the HTML document. 3DHOP requires meshes to be preprocessed into the Nexus format [136]. Furthermore, at least basic coding knowledge is required to add 3DHOP to a web page and even more advanced technical skills if one desires to add interaction or model analysis tools. Out of the box, 3DHOP is able to support model rotation, changes of lighting and rendering parameters, and measurement. Some VH scholars with sufficient technical experience have been able to implement and extend 3DHOP [137][138]. Yet despite the authors attempts for simplicity, the technical requirements of 3DHOP have been a barrier [139] and development of an automated workflow for the viewer is under research with ARIADNE [135]. 3DHOP is able to support the high-resolution models used by VH scholars and provides some basic tools for their scholarly study. Additionally, 3DHOP claims that the file format used provides basic security and encryption for the 3D models. However, this claim is not substantiated and ultimately rests on the format's obfuscation of geometry, a fundamentally insecure scheme. 3DHOP represents a promising development in 3D viewers for VH.

3.5.5 Custom Viewers

The lack of a satisfactory solution for 3D model viewing on the web is evinced by the continued development of custom viewers for VH projects. Many of these projects have utilized JavaScript libraries for 3D content like Three.js and SpiderGL. However, unlike 3DHOP, these projects do not intend to develop reusable viewers but instead are focused on providing a front end for a particular application. The motivations for the development of these custom viewers reflect functionality lacking in the existing, generic solutions.

Fisseler et al. implement a custom viewer for 3D scans of cuneiform tablets using Three.js and the Nexus format, essentially duplicating the work of 3DHOP [140]. Their motivation for using the Nexus format is identical to 3DHOP, to provide fast and interactive visualization of large scanned models with detailed textures. However, in order to support the typical work of scholars studying cuneiform tablets, which requires detailed and magnified examination of often highly damaged tablets, Fisseler et al. implement a number of advanced rendering methods not easily supported by 3DHOP: ambient occlusion, ambience scaling, lit sphere shading and autography mode, a stylized line-drawing like rendering meant to mimic typical illustrations of cuneiform tablets. Additionally, the viewer includes a measurement tool and a visible ruler that scales proportionally with the view, allowing measurements to be taken from screenshots. Fisseler et al. develop a viewing solution well-tailored to their specific application. Resources permitting, this is an excellent option. However, such development required significant technical expertise. Ideally, the extensions required by Fisseler et al. could have been added to an existing viewer, thereby avoiding the duplicated efforts with other viewers like 3DHOP.

Pierdicca et al. 2016 used Three.js and Tween.js to develop an AR enabled browser viewer for 3D models of architectural elements from the Chan Chan site in Peru [141]. Both Unity and Sketchfab support stereoscopic rendering. But neither were usable in this particular case as Unity suffers harsh performance issues when built for the web and Sketchfab did not easily support the side by side and annotated comparisons desired by Pierdicca et al.

3.6 Future Developments for VH Viewers

Several crucial requirements for 3D viewers in VH remain to be addressed. Most notably, besides all-or-nothing access controls like in the case of Fisseler et al, no satisfactory solutions have been presented for securely viewing 3D models. Scholars are faced with a harsh choice between easy access and dissemination without rigorous analysis tools or security (Sketchfab), virtual spaces and immersion but hard limits on performance (Unity), or efficient single model manipulation with added complexity (3DHOP, X3DOM). If none of these platforms are deemed sufficient, projects must then devote significant resources towards the development of custom viewers, which in many cases is not possible given lacking funds or technical skills.

3.6.1 Unity Assets

Regardless of its drawbacks, Unity remains an incredibly powerful tool for creating interactive scenes. Numerous scholars have made use of the game engine and the results have been promising. However, virtually all of this work has focused on the production of singular exhibitions and applications. As such, Unity provides two avenues for future development that could bring wider benefit to the VH.

First, Unity itself offers support for the distribution of code and model libraries through the Unity Asset Store [142]. Popular free assets have made it possible for Unity users to quickly and easily incorporate virtual reality [143], professional grade post-processing filters [144], and particle simulations [145]. Scholars in VH could develop Unity Store assets in order to streamline and standardize the development of VH Unity applications. For example, assets could be produced for visualizing uncertainty of structures, adding annotations to 3D scenes, or examining 3D models according to art historical proclivities.

Second, Unity may be used to build more complete VH tools that provide additional, graphical instruction. Using the Unity interface and incorporating downloaded assets requires a non-trivial amount of technical knowledge and can prove cumbersome to those without the appropriate technical experience. However, with Unity one may produce pre-built graphical software that more visually and simply provides access to tools useful to VH scholars. Much of the functionality of Unity may be built into applications and executed upon user input and interaction. This could include the loading and orienting of 3D architectural models or sculptural models, visual examination and comparison between models, and the creation of virtual walkthroughs. Essentially, an additional, graphical layer may be added to the Unity workflows already demonstrated in the literature in order to increase ease of use for scholars with less technical experience.

3.6.2 Extending Single Model Viewers

Both Sketchfab and 3DHOP offer distinct benefits for the viewing of single models. Potentially, with added development efforts the drawbacks of these platforms could be minimized.

Not only has Sketchfab proven a popular and efficacious means for the dissemination of VH 3D models but Sketchfab has payed particular attention to the VH community. Sketchfab encourages use by scholars and academic institutions by awarding such users complementary paid subscriptions. Furthermore, cultural heritage models have been a featured element of Sketchfab's platform, codified as a category of model. Sketchfab has encouraged the production of such models through promotional efforts. Yet despite the pointed encouragement and promoting, Sketchfab has done little to modify the viewer so as to facilitate academic use. Occasional efforts have been made, such as the implementation of a rudimentary measurement tool [146]. This lack of development is unsurprising. For while Sketchfab clearly desires to further scholarly efforts, it is the graphic artist and gaming communities that ultimately finance the site. But given Sketchfab's willingness to facilitate use for cultural heritage, a beneficial partnership could be possible. Potentially, Sketchfab could be extended to better support the significant use by VH scholars. Using the Sketchfab Viewer

API [147] it is possible to link custom elements on the HTML page to the Sketchfab viewer as well as load models programmatically. Coupled with the Sketchfab Data API [148], used to query the Sketchfab database, one may build more complex and custom sites around the Sketchfab viewer.[5] Nonetheless, this does not address the more fundamental issues of Sketchfab's external hosting and lack of analysis tools.

As an already open-source and academic project, extending 3DHOP is certainly feasible. As already noted, with ARIADNE Ponchio et al. have already put in efforts to make deploying models to 3DHOP simpler and easier. Additional work is needed to add analysis tools and interaction tools. But as a result of the underlying file format and the partitioned and obfuscated geometry, some drawbacks of the platform may not be solved. Specifically, there may be only limited improvements in security of the models and the computational analysis of the models.

3.6.3 Remote Rendering

With Koller et al. 2004, remote rendering, where computationally intensive rendering tasks are offloaded to a server and the results transmitted back to a client viewer, was shown as an effective means for the protected viewing of high-resolution interactive 3D digital models of cultural heritage objects. Yet since its publication, there has been only one other instance of the systems use and the majority of the areas for future research on the system identified by Koller et al. remain to be addressed. With the current, advanced state of WebGL, Koller et al.'s remote rendering system may be further improved by moving the client from stand-alone software to a web browser. Such a system could be bolstered through the incorporation of the numerous technical advances in the protection, analysis, search, and visualization of 3D content. Current remote rendering development efforts will be surveyed and extended in Chapter 4.

[5] For an example of such a site see http://www.digitalsculpture.org/florence, which uses the Sketchfab Data API to present a collection of Sketchfab models in a custom, search-able web page.

CHAPTER 4

PROTECTED REMOTE RENDERING REVISITED

The protected remote rendering system of Koller et al. [10] represents a promising avenue for the development of a complementary 3D viewer for VH, one that has seen only minimal development since its publication in 2004. In this chapter, Seymour[1] is presented, a modern implementation of protected remote rendering following the original proposal by Koller et al. [10]. The code for the project is available at https://github.iu.edu/leifchri/Seymour.

4.1 Previous Work

Seymour builds directly from the previous work of Koller et al. [10] However, the development of the system has also been informed by other works on remote rendering and 3D visualization on the web. In the following section, we present a selection of important previous works on these topics.

4.1.1 Protected Remote Rendering

In 2004, Koller et al. [10] proposed a remote rendering system, named ScanView, for the visualization of high-resolution 3D digital models of CH objects. Remote rendering is a client-server architecture in which resource and/or time intensive computations are performed by a server and transmitted to the client for display. A schematic of a remote rendering architecture is shown in Figure 4.1. In the case of Koller et al., the original 3D data is never transmitted to the client viewer, thereby not directly exposed to unauthorized access.

Koller et al. used remote rendering to protect 3D models of statues by Michelangelo produced as part of the Digital Michelangelo Project [101]. In Koller et al.'s system, the user interacts with a low-resolution reference model on the client. When the interaction event is determined to be finished

[1]For those confused by the name, it is a pun on the benefit offered by the software.

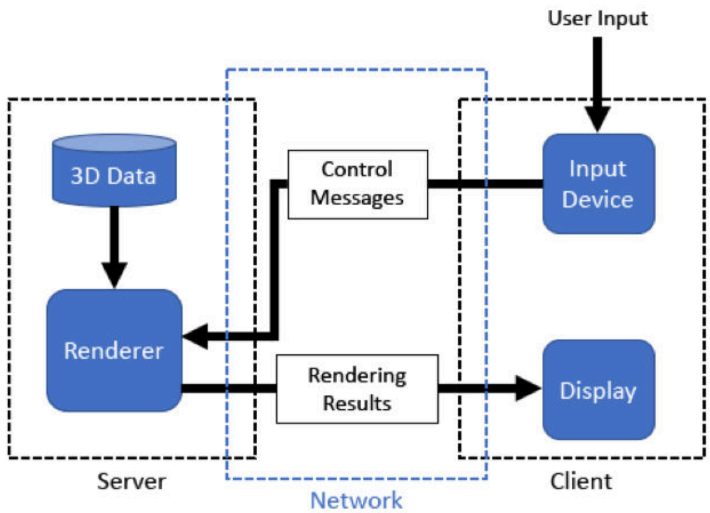


Figure 4.1: A schematic of a basic remote rendering architecture.

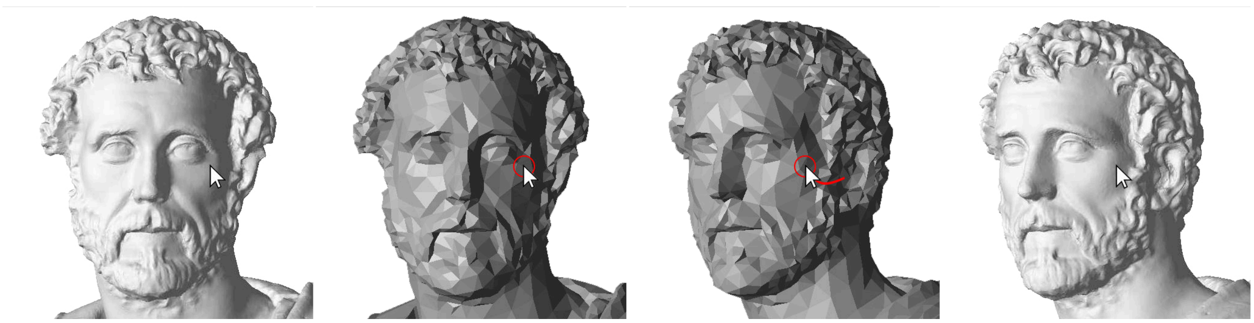


Figure 4.2: Sample frames from a remote rendering application. Mouse actions are shown in red.

(via a mouse up event), the client sends a request to the server for a high-resolution rendering of the corresponding view. This rendering is then superimposed upon the client's viewing window, giving the appearance that the low-resolution model has been replaced with the high-resolution one. An example series of frames from such an application are shown in Figure 4.2.

While the 3D data is not directly exposed, there still exist malicious techniques for the piracy of the original 3D model, namely image-based reconstruction attacks. In order to thwart this potential malicious activity, Koller et al. implemented a number of defenses including client request monitoring, limited viewing areas, and perturbations to the rendered image [10, p. 4]. Koller et al. empirically demonstrated the security of their system against several reconstruction attacks. Scanview proved popular, in the four years following publication the standalone software was downloaded over 50,000 times [149].

In the original publication and those following, Koller et al. identified a number of areas for future work and improvement. These included support for textured models [10, p. 699], porting the client to run in web browsers, the incorporation of analysis tools like measurement [149, p. 330], the addition of complementary protection techniques like watermarking, and measurement of the perceptual impact of the defensive distortions [78].

While well cited, the system architecture of Koller et al. has not seen wide use. Partly this may be attributed to the vagaries of the original publication. Koller et al. do document the major technologies used in building the application: Apache web server, FastCGI Apache module, and OpenGL. Yet no comprehensive architecture of the system is provided and only sparse implementation details are mentioned. Thus, publications seeking to extend this work must re-engineer the system nearly from scratch. Likely, the lack of details was an effort on the part of Koller et al. to prevent compromise of the security by malicious users. However, this security through obscurity approach runs counter to security best practices and Shannon's Maxim that, "one ought to design systems under the assumption that the enemy will immediately gain full familiarity with them" [150]. Regardless, two publications have directly addressed these issues for future work identified by Koller et al.

In 2006, Su et al. [151] presented a similar remote rendering system named Protected-3DMPS. Su et al. made several modifications to the original architecture proposed by Koller et al. Most notably, Su et al. included support for textured models. However, the authors did not rigorously test the effect of textures on adversarial reconstruction techniques. Additionally, rather than interact with a triangular mesh, in Protected-3DMPS the user manipulates a colored point cloud as a reference model. The motivation behind this change goes unstated. While rendering a point cloud does marginally lighten the rendering load on the client, this may be a trivial improvement since the reference model is already heavily decimated. Furthermore, the point cloud rendering limits the users ability to effectively modify lighting parameters using the reference model. Su et al. partition the server into three modules, one for rendering, one for image readback and compression, and one for communication.

In 2008 Zhu et al. performed a usability study to test the perceptual impact of distortions included in ScanView [152]. Zhu et al. found that low levels of Perlin noise introduced into the rendered image did not have an impact on user interaction nor were perceptually identifiable by users. Zhu et al. asked users to complete a simple 3D manipulation task, rotating a 3D model such that a point on the model aligned with a cross hairs drawn on the screen. Perlin noise was used to perturb the rotation input of the user. However, Zhu et al. did not satisfactorily demonstrate the efficacy of their chosen distortions against reconstruction attacks. While the authors did test reconstruction attacks, even the undefended reconstructions were visibly degraded.

Certainly, the major part of the future research directions identified by Koller et al. remain to be actualized. Furthermore, since the security of the system is demonstrated empirically, it must be re-tested given modern image-based reconstruction algorithms, hardware, and software. That said, there have been broader developments within remote rendering applicable to the particular case of VH.

4.1.2 Remote Rendering

While research specifically on Koller et al.'s system has been minimal, remote rendering more generally has proven a popular topic in the scholarship. Remote rendering has received intense attention over the past decade as a method for delivering interactive graphics to heterogeneous devices as remote rendering facilitates the visualization of large or unwieldly data in real-time. Scholars have developed many remote rendering systems, in so doing discussing issues of implementation and data representation. This has included multiple surveys, which attempt to distill meaningful trends and identify fruitful avenues for future research. Among these surveys, authors have used differing criteria to examine the field. As such, when put into dialogue with one another, these surveys may offer additional insight on the field and inform the development of new remote rendering applications.

Technically, any rendering task using two networked devices may be considered remote rendering. In the most trivial and common case, a client requests a 3D scene from a server, downloads all of the 3D information, and then renders the scene locally on the client. Unsurprisingly, far more complex remote rendering systems exist; the rendering processes may be split at different points between the client and server, data may be represented in a variety of ways, compression may be added, various messaging protocols and transmission schemes may be used, etc.

Remote rendering has been an active area of research since at least the early 1990s. Given the state of graphics rendering technology at the time, for many applications it was desirable to have dedicated machines for part or the entirety of graphics operations. For example, Ohazama investigated the sharing of dedicated graphics workstations [153]. General systems for the control and display of remote systems continued to be researched through the mid 2000s but never featured strongly in the scholarship [154]. Now, robust software solutions exist to solve this issue such as TeamViewer [155] and some are even built into operating systems [156]. More problem-specific remote rendering systems were also represented among the earliest research including architectural walkthroughs [157], parallel rendering of physics data sets [158], and collaborative virtual reality environments [159]. In recent years, new application areas for remote rendering have emerged, most notably cloud gaming and mobile applications. Scholars have continued to research the benefits of remote rendering in these areas and remote rendering has begun to be adopted in industry [86] [160].

Remote rendering has been used in a variety of situations, with equally various client and server architectures. However, the problems addressed by remote rendering in most cases fall into one of two categories: limitations of computing resources or limitations of data. Computing limitations most often stem from underpowered clients but may also be a result of computationally intensive rendering techniques. In this first case, remote rendering may be used to bring complex graphical applications to low-powered clients like mobile devices, providing cross platform support for heterogeneous clients, and making efficient use of computing resources through server architectures like parallelism or dedicated hardware. In the second case, data may be exceedingly complex or large, prohibiting transmission or loading the entire data set in memory. While useful for introduction, scholars have found such simplistic partitions insufficient to represent the great diversity of remote rendering systems.

Several classification schemes have been introduced to organize the dizzying amount of publications on remote rendering. The earliest of these was presented by Schmalstieg and Gervautz in 1996 [161]. Schmalstieg and Gervautz classified remote rendering approaches into three categories based upon the data sent by the server to the client: images, graphics commands, or geometry. Graphics commands are the low-level API calls used to interact with the hardware and software responsible for rendering content to the screen.

Martin presented another classification scheme in 2000 [162]. Unlike Schmalstieg and Gervautz who focused on networked virtual environments, Martin was interested in the rendering of complex 3D models, i.e. standalone representations of individual objects. Martin classified remote rendering systems according to where the rendering occurred: client-side, server-side, or a hybrid of the two. Martin developed a system meant to optimize performance relative to the capabilities of the client by the adaptive selection of 3D representation, e.g. a polygon mesh, 2D image, depth image, or basic shapes. Martin considered this process as an extension of transcoding, the direct conversion between digital encoding schemes, to 3D data. This type of 3D geometry transformation represented a novel tool for remote rendering systems and would become a category viable for distinction in later classification schemes.

In 2014, Evans et al. [86] used Schmalstieg and Gervautz as the basis for a similar classification scheme, also based on the type of data transmitted from the server. Evans et al. added an addition category, “Primitives or Vectors”, reminiscent of the transcoding process of Martin. Evans et al. defined Vectors as new representations of data based on the selection of cogent features of the 3D geometry. For example, Quillet et al. [163] developed a system to transmit line drawings of 3D city models, better representing the cogent building features of doors, windows, and architectural details.

In 2015, Shi and Hsu [154] provided the most comprehensive survey of remote rendering systems along with two classification schemes. The first and coarser of the two schemes classifies remote rendering techniques according to two dimensions, whether the 3D model data is static or dynamic and whether the user interactions are restricted or unrestricted. Restricted user interaction is defined as when "a user can only interact with the 3D data in a few limited ways," for example only viewing the model from predefined viewpoints or being only able to change the viewing within a limited range [154, p. 6]. According to Shi and Hsu, static models and restricted interaction encompasses the majority of remote rendering systems.

While Shi and Hsu initially present the difference between restricted and unrestricted interaction as one of viewing, this does not seem to be the case. Among their examples of static and restricted applications are what may be considered full featured viewers [164] [165]. Furthermore, Shi and Hsu's numerous examples of VR walkthroughs, presented as unrestricted interaction, may be considered restricted. A first-person perspective is but one of many modes through which a 3D scene may be examined. Interaction with 3D data is a lively area of research and one with numerous gray areas. All interaction techniques involve some level of tradeoffs and some amount of restriction. The most typical display format for 3D models, 2D displays, itself imposes certain restrictions on interaction techniques. Even further restrictions derive from the typical interaction hardware, i.e. keyboard and mouse. Arguably, a 3D trackball controller would allow greater interaction capabilities [166]. While in some cases, restricted user interaction is clear, such as Koller et al. [10] who greatly restrict the viewing controls, this is often not an easy distinction to make.

In their first classification scheme, Shi and Hsu identify two components of key importance when designing and evaluating remote rendering systems: data representation and user interaction. Making a priori assumptions about either of these categories allows remote rendering systems to be better developed to suit a particular problem. For example, systems dealing with static data may make use of time-saving preprocessing and constraints on interaction may allow for optimization of rendering techniques by excluding certain renderings as potential output. As previously mentioned, the boundaries of Shi and Hsu's first classification scheme appear quite muddled. However, we may rephrase this scheme as one of a priori assumptions, assumptions made concerning the data representation and concerning user interaction. In this way, we may avoid the issue of defining boundaries between restricted and unrestricted and instead treat the dimensions as a continuum.

Shi and Hsu's second classification scheme is fine-grained and based upon the type of data transmitted from the server. While similar to the approaches of Schmalstieg and Gervautz and Evans et al., Shi and Hsu provide far more detail and corresponding categories (see Table 4.1). The classification scheme of Shi and Hsu encompasses all the same categories as Evans et al. except one, what Evans et al., and Schmalstieg and Gervautz before them, labeled "Graphics Commands". Shi and Hsu focus their survey on problem-specific remote rendering systems. Graphics Commands, at least in Shi and Hsu's analysis, are utilized in general purpose systems and thus are not included in the classification. Unlike Evans et al., Shi and Hsu do not consider feature extraction and the computation of Primitives or Vectors as a unique category. Shi and Hsu place feature extraction under "Simplified Model". According to Shi and Hsu, this approach has the con of loss of quality. However, feature extraction does not always result in loss of quality and is unique enough to deserve addressal as its own category.

While feature extraction may be used as a means of simplifying models, as is seen in Shi and Hsu's cited example of Li et al. [167],[2] it may also be used to generate additional models from data sets. Quilllet et al. [163] developed a remote rendering system for the visualization of large city models wherein high-resolution textures of building facades are replaced by lines representing their distinct features such as windows and doors. Part of the benefit of this system is the reduced size of the transmitted textures. However, the derived models are not necessarily of lower quality than the originals. To the contrary, Quillet et al. argue that the line renderings allow for the cogent features of the model to be more easily observable.

Similar to Quillet et al., Mindek et al. [171] developed a remote rendering system that renders complex scientific visualizations based on extracted features from 3D datasets. Mindek et al. developed their system, named Marion, to solve issues with biological illustrations, time consuming endeavors that are often out of date by the time they are published due to the frequent updates of the underlying data. Marion extracts representations from surface, volumetric, and parametric models, combining these representations to produce an interactive 3D model. The models output by Marion

[2] Li et al. present a collaborative CAD system that, when a new element is instantiated, sends only minimal information to other clients, who then recompute the result of the operations.

Data Type	Description
Original Model	**Server**: transmits all 3D data to client **Client**: perform 3D graphics rendering when all data is received **Pros**: general purpose **Cons**: “fat“ client; excessive bandwidth for complex models **Examples**: Karlton et al. (2005) [168] and Eisert and Fechteler (2008) [169]
Progressive Model	**Server**: transmit all 3D data progressively to client, based on viewpoint or multi resolution representation **Client**: perform 3D graphics rendering upon the arrival of 3D data **Pros**: reduce the rendering “start“ time; bandwidth control **Cons**: “fat“ client; preprocessing to generate progressive models **Examples**: Potenziani et al. (2015) [114] and Abderrahim and Bouhlel (2016) [170]
Transcoded Model	**Server**: transmits transcoded 3D model(s) to client **Client**: perform 3D graphics rendering on the 3D model(s) received **Pros**: bandwidth control; reduce rendering workloads of client **Cons**: additional processing requirements on server **Examples**: Mindek et al. (2018) [171] and Martin (2000) [162]
Simplified Model	**Server**: transmit simplified 3D models to client **Client**: perform 3D graphics rendering on the simplified models received **Pros**: reduce bandwidth usage; reduce rendering workloads of client **Cons**: quality loss due to model simplification; preprocessing to generate simplified models **Examples**: Duguet and Drettakis (2004) [165], Li et al. (2011) [167], and Moraes et al. (2017) [172]
Model + Image	**Server**: transmit reformed (usually simplified) 3D models and the difference image to client **Client**: perform 3D graphics rendering and apply the difference image **Pros**: maintain rendering quality with low bandwidth and light weight computation on client **Cons**: preprocessing to generate simplified models and render both original and simplified models for the difference image **Examples**: Spini et al. (2016) [173]
Image	**Server**: perform 3D graphics rendering, transmit result images to client **Client**: display the images received **Pros**: high rendering quality; low bandwidth usage; no rendering workloads on client; source secure **Cons**: interaction latency **Examples**: Koller et al. (2004) [10] and Raji et al. (2017) [174]
Environment Map	**Server**: perform 3D graphics rendering of the whole environment, generate an environment map (i.e., panorama) and transmit the environment map to client **Client**: project the received environment map to the correct viewpoint **Pros**: pros of Image; no latency for some types of user interaction (e.g., pan, tilt) **Cons**: extra workloads on server to generate environment maps; interaction latency for other unsupported user interactions **Examples**: Lai et al. (2017) [175]
Image + Depth	**Server**: perform 3D graphics rendering one or multiple times, extract depth maps together with result images, and send all result images and depth maps to client **Client**: display the result images received; if necessary, run IBR algorithms to synthesize images at new viewpoints **Pros**: pros of Image; reduce latency for most user interactions that only change rendering viewpoint **Cons**: extra workloads on server to generate multiple depth images; extra bandwidth needed to transmit all depth images; IBR artifacts; interaction latency for other unsupported user interactions **Examples**: Chen et al. (2017) [176]

Table 4.1: Sumary of remote rendering systems adapted from Shi and Hsu [154]. Changes are shown in red.

are inspired by scientific visualizations from biology and utilize common visualization techniques for the field such as the ball-and-stick model of molecules and pseudo cylinders for mitochondria. In this case, representing the final models as simplifications of the original data is especially problematic. The final models are themselves carefully constructed scientific visualizations of arbitrary complexity. Given these remote rendering applications, we may add an additional category to Shi and Hsu's classification, that of transcoded models. An updated version of Shi and Hsu's classification is shown in Table 4.1.

The initial research surveyed and undertaken by Schmalstieg and Gervautz used remote rendering with expensive client devices like CAVEs. In contrast, both Evans et al. and Shi and Hsu discuss thin clients, devices with limited computational capacity and equally constrained graphics acceleration. The focus on thin clients has continued to characterize remote rendering research in the following years. The wide support for WebGL, enabling hardware accelerated graphics to run natively in web browsers, has led to a proliferation of remote rendering systems using web browsers. Since this support extends to mobile, these applications often make use of mobile devices for interaction. Thus, both Evan et al.'s intimation that remote rendering would prove a valuable tool for 3D graphics on the web and Shi and Hsu's identification of mobile and cloud gaming as the next key areas of remote rendering research, have held true. Remote rendering systems using browser-based clients have been developed for collaborative CAD design [177], virtual walkthroughs of architectural spaces [173], scientific visualization [171], viewing models of CH objects [114], and creating 3D artistic suites [178]. Mobile platforms have also received attention as gaming platforms [179], untethered VR headsets [175], and tools for scientific visualization [180]. The wide variety observed by the previous authors has remained, with scholars presenting remote rendering systems falling throughout the categories of Table 4.1.

4.1.3 Web Clients

Recent research on remote rendering reflects the identification of web clients as a valuable avenue of research [86] [10] [154]. Given the proliferation of relatively low powered devices like smart phones, remote rendering has been identified as a key technology in enabling high-fidelity gaming experiences [154]. To this end, browser-based clients have proven popular. As discussed in Section 3.3, web browsers now provide cross platform and native hardware accelerated graphics support without the need for additional downloads. In their 2014 survey of 3D graphics on the web, Evans et al. [86] identified remote rendering as a key technology, precisely because web browsers are so often used as the client.

ParaViewWeb [181] is a well received web framework for the interactive remote visualization of large scientific datasets. ParaViewWeb allows a web browser to function as a remote client to a ParaView instance on a webserver, forwarding the user interaction from the browser to the server and displaying the rendering results in the browser. However, while ParaViewWeb does perform rendering tasks remotely, the result of the server-side computation, whether it be an image, 3D geometry, etc., is transmitted in its entirety to the client for visualization. ParaViewWeb demonstrates how remote rendering may provide thin clients access to computationally intensive analysis algorithms. However, ParaViewWeb only minimally extends the real time graphics capabilities of the client, as any 3D results must be transmitted in full to the client.

Du et al. [177] present a system for collaborative CAD design, meant as a proof of concept for realtime Web3D design, which may be classified as Original Model according to Table 4.1. Du et al. used the Three.js JavaScript library [182] for in browser 3D rendering and WebSocket [183] for inter-browser communication. Du et al. represent an edge-case within the remote rendering literature. While technically a client-server architecture, the system of Du et al. was designed to function more as a peer-to-peer architecture. The initial client is deemed the "Host" and it is the state of the Host that is given priority. The server is used to route control messages between the clients. In order to avoid desynchonization, Du et al. utilize a locking scheme so that no two users may edit the same geometry at the same time. Du et al. are able to achieve synchronization speeds of 50ms. However, no statistics are reported on rendering speeds.

Similar to Du et al., Spini et al. [173] present a system for the online authoring of 3D digital models. Their system is composed of three components: an authoring tool, a rendering service, and the Web VR Explorer [184]. The interactive 3D components were implemented using Three.js. In the case of Spini et al., the remote rendering is not interactive. Rather, the user sets up a 3D scene, requests a baked rendering of the scene, and after a processing period is able to load and explore the rendered scene.

Raji et al. [174] propose a remote rendering architecture largely similar to Koller et al., although one built for volumetric, as opposed to surface, rendering. Raji et al. remove all 3D requirements from the client, transmitting only an image stream. The user may interact with the graphical element as with typical 3D views, clicking and dragging to rotate. The input is then sent to the server and requested renderings are streamed back to the client. Raji et al. make use of a similar interaction based level-of-detail technique as Koller et al. During interaction, images streamed are of a lower resolution, so that interactive frame rates may be achieved. Once input has ended, a high resolution rendering is sent.

The remote rendering systems using web browser cleints surveyed here, represent a critical advancment in remote rendering that may be applied to Koller et al. In fact, Koller et al. themselves identified such a development as important for future work on the system over a decade ago [149].

4.2 System Architecture

The original system of Koller et al. may be improved through the use of modern web technologies for 3D, thereby transitioning from a standalone client application to one running natively in the web browser. Furthermore, the system may be extended to support textured models. The system implemented for this work is called Seymour.[3] This section presents only the basic protected remote rendering system. The visualization tool is expanded to more fully meet the needs and uses of VH scholars in Chapter 6.

As with Koller et al., Seymour is divided into two primary components, a server and a client. The specifics of each are discussed in Section 4.2.2 and Section 4.2.1 respectively. In Seymour, the user interacts with a low resolution model on the client. Once interaction ceases, i.e. when

[3]See the project page at https://github.iu.edu/leifchri/Seymour [185].

Figure 4.3: Sample frames from Seymour. Mouse actions are shown in red.

the input device (mouse or touch) is released, the server transmits a high resolution rendering in the JPEG format corresponding to the current view of the client. Rather than a component of the rendering, the low resolution client may be considered an interaction element. Thus, Seymour may be classified as Image within Table 4.1. A series of sample frames is shown in Figure 4.3.

Per Section 4.1.2, the design of Seymour is based on two crucial assumptions as to the data represented and the user interaction.

1. The visualized geometry is static, i.e. there are no dynamic elements such as animations and the geometry is not modified by user input.
2. Interaction is primarily limited to art historical inspection; users are interested in high fidelity views of detailed objects.

Assumption 1 implies that the underlying geometry visualized is stateless. Therefore, there is no requirement for persistent server-side rendering contexts. The "stateful" parameters, such as viewing angle, light positions, etc., may be retained by the client and sent per rendering request. Given this, the server may be decoupled from the client. Assumption 2 describes the typical interaction of users as relatively slow, focusing on high fidelity images examined in detail, as opposed to quickly navigating an environment or interacting in a fast-paced or time sensitive way as is the case for racing games. This allows for the use of what Raji et al. describe as interaction dependent level-of-detail rendering [174, p. 2]. The visual quality of the rendering may be reduced during interaction, with high-quality renderings provided once user interaction has ceased.

The basic implementation of Seymour includes support for the viewing of a scene comprised of

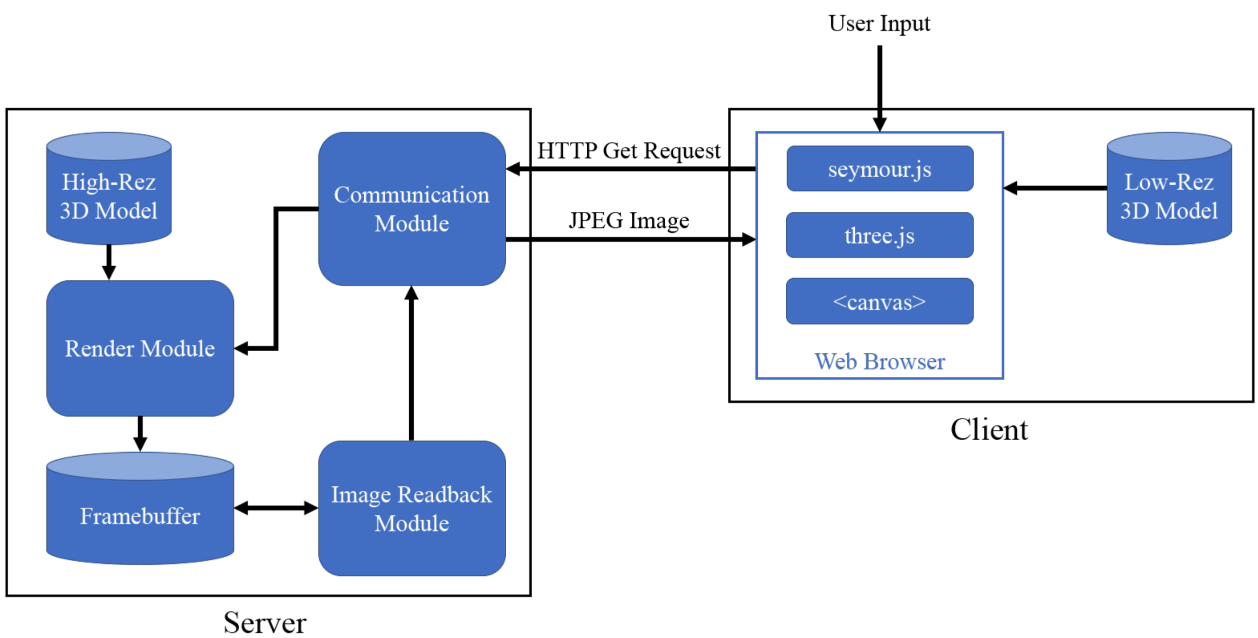


Figure 4.4: The system architecture of Seymour.

a single 3D digital model with up to three colored point lights. The user may interact with the model by rotating the model about its center of mass, panning the camera, and modifying the lighting parameters.

Seymour requires a model pre-processing step to properly format the model for display and generate the low-resolution reference model. First, to simplify the server-side rendering and defensive distortions, the model is normalized such that the extents of model fit within the unit cube. Second, using the MeshLab command line tool, a reference model is generated by decimating the original model to 10,000 polygons. While software is included to automate this step for the sake of ease-of-use, its use is not required. The reference model may be generated manually and need not even correspond to the high resolution model on the server.

A diagram of the complete system architecture is shown in Figure 4.4. Let us examine the server and client of Seymour in more detail individually, their component tools and technologies, design decisions, and deployment details.

4.2.1 Client

The web client, running natively in a web browser, is implemented using HTML, CSS, JavaScript, and WebGL. The client is responsible for detecting and responding to user input events, rendering the reference model, requesting frames from the server, and displaying the returned frames.

In order to further facilitate the use of Seymour by underpowered devices, the initial download of the web client was kept as light as possible. Therefore, no styling, layout, or JavaScript frameworks were included.[4] Two additional JavaScript libraries were used: Three.js, to support rendering the reference model, and Hammer.js, for touch input.

Three.js [182], previously mentioned in the discussion of Web3D in Section 3.3, is an open-source JavaScript library for cross-browser interactive 3D computer graphics released under the MIT license. Three.js leverages native hardware acceleration using WebGL and the HTML5 <canvas> element. Three.js is a robust community effort, initially began by Ricardo Cabello in 2010 [186]. Since then, Three.js has grown into one of the most popular 3D graphics libraries on the web and has expanded to include a host of 3D algorithms and design patterns like scene graphs, physically based rendering, depth peeling, level-of-detail rendering, and custom shaders. Three.js has been incorporated into curriculum as well, and has been featured in both textbooks [187] and online courses [188]. Support for Three.js is available through online communities include StackOverflow [189], Github [182], and Freenode [190]. Three.js has been used to develop full games within the browser [191], in-browser VR applications [192], and was even featured on the website for the recent Hollywood film Swiss Army Man [193]. A simple Three.js scene is shown in Table 4.2. Additional JavaScript files are necessary for the loading of 3D files. Three.js extension libraries are available to load glTF, FBX, OBJ, PLY, and many more formats and may be included on an as needed basis. The binary PLY file format was chosen for the first implementation given its compactness and easy conversion to a human readable format. The Seymour client exposes

[4]Whether this is a necessary optimization is debatable. Many of the most common JavaScript libraries are available via Content Delivery Networks (CDN), like Google. Using a CDN, web pages from different servers may use the same distribution of JavaScript libraries. Thus, when visiting the page, the web browser may reuse the previously cached JavaScript files and avoid additional downloads.

```
// create the camera
var camera = new THREE.PerspectiveCamera( 70, window.innerWidth /
    window.innerHeight, 1, 1000 );
camera.position.z = 4;
// create the root of the scene graph
var scene = new THREE.Scene();

// create and position a white point light
var pointlight = new THREE.PointLight( 0xffffff );
pointlight.position.x = 5;
pointlight.position.z = 5;
scene.add( pointlight );

// create a purplish sphere with specular shading
var geometry = new THREE.SphereGeometry( 1, 40, 40 );
var material = new THREE.MeshPhongMaterial( {color: 0x7777ff} );
var mesh = new THREE.Mesh( geometry, material );
scene.add( mesh );

// create the renderer and add it to the html document
var renderer = new THREE.WebGLRenderer();
renderer.setSize( window.innerWidth, window.innerHeight );
document.body.appendChild( renderer.domElement );

renderer.render( scene, camera );
```

Table 4.2: Code for a simple scene in three.js. For simplicity, HTML code is omitted. The resulting render is shown in Figure 4.5.

the Three.js objects used in rendering the client scene, and thus may be manipulated and extended according to the well documented usage of the Three.js library.

Hammer.js [194], also an open-source JavaScript library released under the MIT license, provides support for event-driven multi-touch gestures. Hammer.js is an exceptionally small library, requiring less than 10kb, but offers powerful cross-platform support for touch screen devices. Further description and demo code may be found on the Hammer.js official site [194]. Hammer.js serves as the foundation for the supported gestures discussed shown in Table 4.3.

As a starting point, Seymour supports basic user interaction based on Colin Ware's "world-in-hand" spatial navigation metaphor. In his seminal work Information Visualization [195], Ware

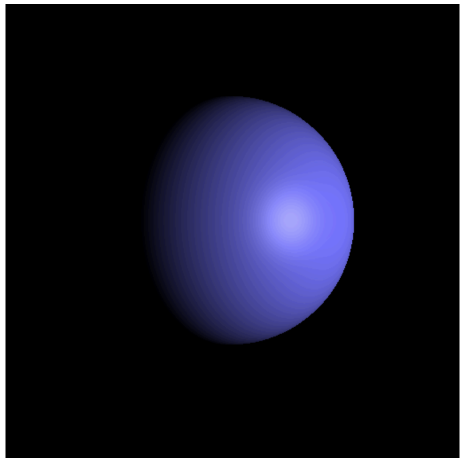


Figure 4.5: A rendering of the scene defined in Table 4.2.

Mouse	**Touch**	**Interaction**
left click + drag	single finger drag	3D rotation
right click + drag	two finger drag	XY-axis translation
right click + drag + shift	three finger drag	Z-axis translation
mouse wheel	pinch	zoom

Table 4.3: Mouse and touch input and their corresponding interaction events.

presents four metaphors for interaction with 3D content: world-in-hand, eyeball-in-hand, walking, and flying. Walking and flying reproduce the typical methods of terrestrial and aerial locomotion. Given the compactness of the data visualized here, i.e. single artifacts or groups of artifacts, these forms of navigation are ill-suited. For such use cases, Ware recommend the use of world-in-hand, wherein the user interacts with the 3D scene as if one were to grab a portion of it in hand and manipulate it. However, as previously discussed, the interaction and design elements of the client are decoupled from the server. The server only requires positions of the elements in the scene, without any knowledge of how the particular arrangement was produced. Therefore, switching between the navigation metaphors as they become appropriate, e.g. if one were to use Seymour to visualize a larger architectural space or terrain, is a relatively easy task, one which will be demonstrated in Section 6.3. Table 4.3 lists the interaction techniques and their corresponding input.

Typical world-in-hand interaction with an object includes translation, uniform scaling/zooming, and rotation. Translation and scaling/zooming are more easily implemented, generally using a

```html
<div id="seymour-container"></div>

<script src="js/three.min.js"></script>
<script src="js/hammer.min.js"></script>
<script src="js/loaders/PLYLoader.js"></script>
<script src="js/seymour.js"></script>

<script type="text/javascript">
  var element = document.getElementById( 'seymour-container' );
  var options = {
    width: 512,
    height: 512,
    backend: 'http://localhost'
  };

  var seymour = new Seymour( element, options );
  seymour.loadModel( 'models/ply/VC_0001_Antoninus_Pious-5k.ply' );
</script>
```

Table 4.4: The <body> of a simple HTML page running a Seymour client.

press/click and drag gesture. The 2D input coordinates may then be mapped to one or two axes in the case of translation or to all three axes or camera parameters in the case of scaling/zooming. Most often, touch screen enabled devices use a pinch gesture for scaling/zooming. Rotation is less easily accomplished. Physical controls for 3D manipulation do exist but for most applications this is an unreasonable requirement. Instead, mappings of 2D interactions with a screen to 3D rotations have been shown to at least partially capture user intuitions [196]. Such techniques have been widely incorporated into world-in-hand model viewers, such as Sketchfab and 3DHOP. However, as pointed out by Potenziani et al., it is difficult to develop a rotation scheme that works for all applications [114]. For this reason, Potenziani et al. include three choices of rotation, Full-Sphere, TurnTable, and Pan-Tilt. There are additional useful rotation techniques that remain to be implemented in remote rendering systems, such as Hanson's Rolling Ball [197]. As an initial implementation, Hanson's Rolling Ball was chosen for rotation, as it most accurately describes the physical movements of a trackball.

Ultimately, the setup of Seymour was abstracted as much as possible to allow for a quick and easy addition to web pages. A sample webpage, with the most basic Seymour settings, is shown in Table 4.4.[5]

4.2.2 Server

The server is implemented using C++ and OpenGL with a number of libraries for graphics, networking, and image processing. A summary and description of the libraries used can be found on the project Github page [185]. The server is responsible for rendering requested frames and transmitted the frames to the client. Nginx and FastCGI are used for network communication and requests.

Nginx is a free and open-source web server software that supports reverse proxy, load balancing, and HTTP caching [198] [199]. According to netcraft, as of February 2019 nginx accounted for 25.91% of the top million busiest sites on the web, is the second most used server among "active" sites at 19.60%, and is one of the fastest growing web server softwares [200]. Nginx uses an event-driven architecture. Nginx was chosen for its wide use, extensive documentation, built in support for the FastCGI protocol, and scalability.

FastCGI is an extension of the CGI protocol that allows for persistent, long-lived CGI processes, i.e. application servers [201]. FastCGI reduces overhead and increases scalability by removing the one process per request limit of CGI. Technically, as of 2017 support for FastCGI was dropped [202]. However, it remains a commonly used protocol throughout the internet and has modules implemented for both Nginx and Apache. The old specification continues to be available online [203] [204] [205].

Following Su et al. [151], the server is broken into modular components. The server software consists of modules for communication, rendering, and image readback.

[5] Code may be subject to change, see the project Github [185] for up to date examples and instructions.

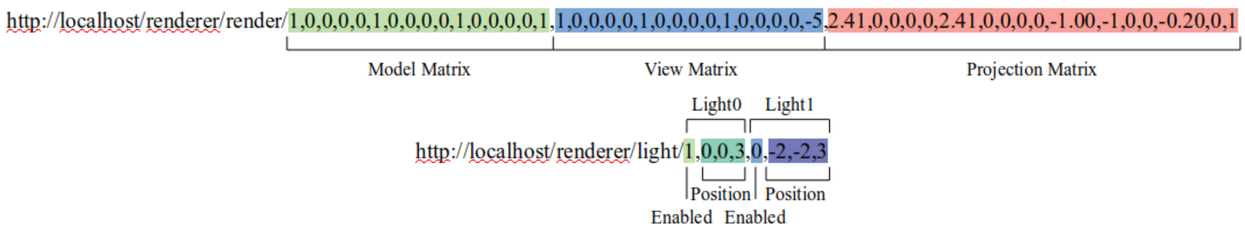


Figure 4.6: Sample Seymour client requests.

Communication Module

The communication module serves as middleware between the client and server, routing requests from the client to the rendering module and the output of the image-readback module to the client. Client requests are made in the form of HTTP Get requests, with a Uniform Resource Identifier (URI) used to specify the desired server-side action. Two actions are supported by default, modifying geometric elements within the scene, e.g. 3D models and the camera, and modifying lights. Example requests are shown in Figure 4.6. Each request is comprised of a route, e.g. “renderer” or “lighting”, followed by a string of values and results in a rendered frame returned from the server. Additional URIs and corresponding server-side functionality may be added. An example of this is shown in Section 6.5, which extends Seymour to include a webpage GUI for the modification of server side defenses.

Rendering Module

The rendering module is responsible for rendering frames with the requested viewing parameters and included server-side defenses. The module is architected based upon Three.js. First, objects representing the Camera, Scene, Renderer, Lights, and Meshes are instantiated. See Figure 4.7 for a diagram of the component classes. Then, the lights and meshes are added to the scene graph. Finally, the Renderer uses a Camera and Scene to render a frame. The rendering module stores the 3D geometry and vertices, normals, and texture coordinates, in the graphics card memory, thereby utilized hardware graphics acceleration available to the server.

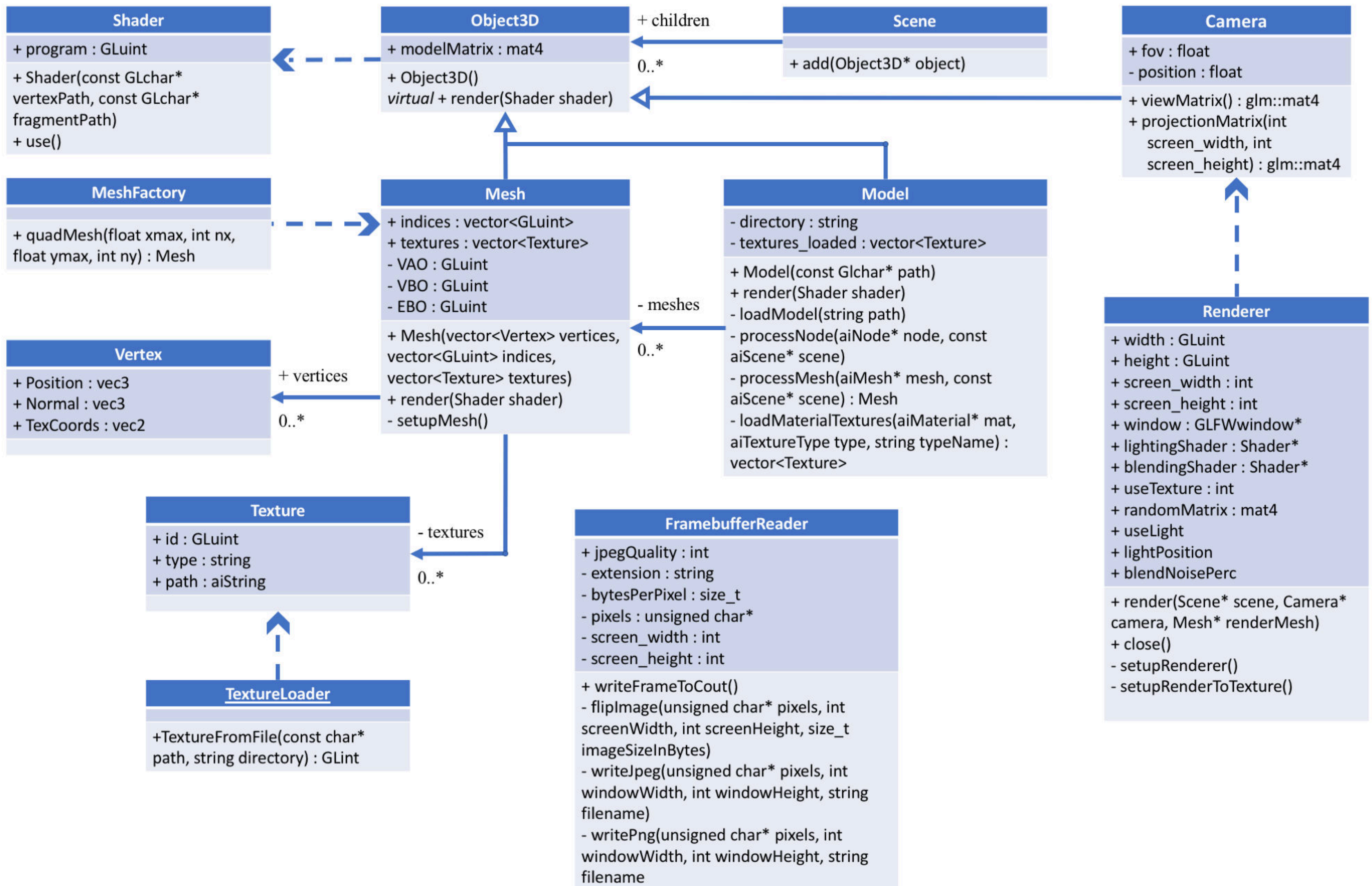


Figure 4.7: A UML Class Diagram for the Seymour rendering module.

Image Readback Module

The image readback module is responsible for retrieving the rendered frame from the server framebuffer and applying JPEG compression. As described in Section 4.2.3, the server makes use of two-pass rendering in order to introduce high-frequency noise. The final result is rendered to the an onscreen framebuffer, a portion of memory used to store the color values of the video display. The values of the pixels, corresponding to the final rendered frame, are read into memory using the OpenGL glReadPixels command. These values are then compressed according using the JPEG compression scheme.

4.2.3 Defenses

While the original high-resolution geometry is never sent to the client, and thus never directly exposed, it may still be retrieved through the use of image-based reconstruction attacks, algorithms that use the rendered 2D frames to reconstruct the 3D geometry. In order to prevent such attacks a number of server-side defenses may be implemented. However, the use of these defenses is a trade-off between security and image quality. The defenses result in either degradation of the image quality (high-frequency noise, JPEG compression) or potentially distracting changes between the requested and rendered views (light perturbation, geometric distortion). Thus, the levels of each defense should be chosen on a per-application basis, depending on the goals of the project.

High-Frequency Noise

Using two pass rendering, high-frequency noise is blended with the rendered frames. Blending in noise greatly increases the entropy of the image, and thus can lead to significantly larger image sizes due to poor compression. To limit this factor, a stencil buffer is used to only blend high-frequency noise with the rendered object and not the background, see Figure 4.8 for an example. 400 1024x1024 pixel noise textures were pre-generated using ImageMagick [206]. At each rendering request, a noise texture is selected and blended with the rendered frame. This method was chosen for its runtime efficiency, as the noise textures do not need to be generated dynamically. However, this

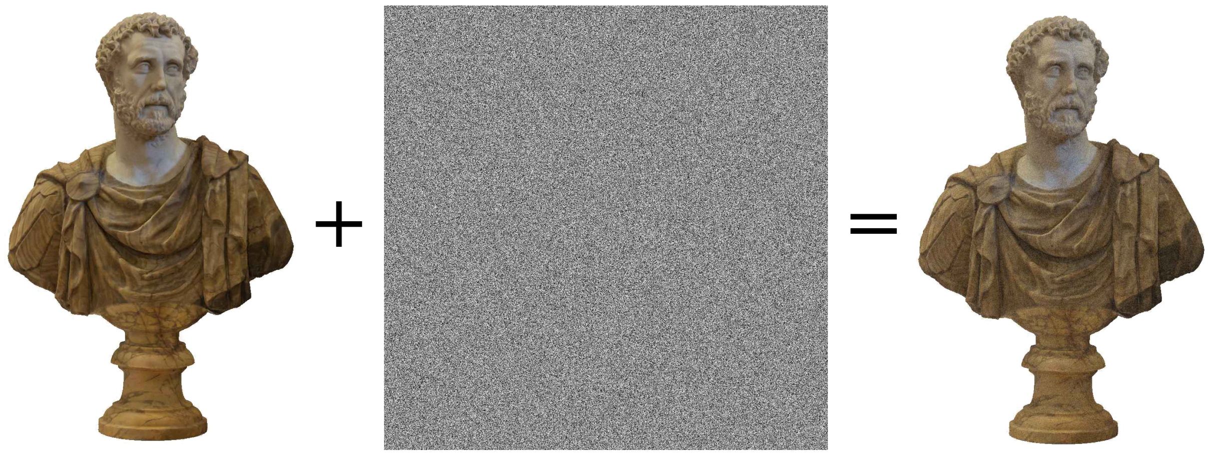


Figure 4.8: Stencil rendering used to blend an original image with 20% of a noise image.

does require a non-trivial amount of additional storage space on the server, although it is minimal compared to the file sizes of the high-resolution models. Two other potential implementations are the generation of the noise textures at runtime, either using an external tool like ImageMagick or generating the noise values directly in code on the CPU, or generation of the noise rgb values in the fragment shader. The former may result in significantly increased runtimes and the latter requires the use of less-sophisticated psuedorandom algorithms given the lack of well-developed psuedorandom libraries for shader languages.

Geometric Distortion

In the original publication of Koller et al., geometric distortions were introduced using a random viewing matrix meant to scale, rotate, translate, and shear the model. Koller et al. offer no information as to how this matrix is actually generated or how they ensure that the distortions are strictly bounded by M object units and N pixels, whichever is greater. When testing security, Koller et al. instead perturbed the viewing angle of the camera by a random amount up to a set number of degrees . A simpler approach with more complex results is to introduce noise into the vertex coordinates themselves. We may speculate as to why Koller et al. chose to use a transformation matrix instead. Potentially, this could be a result of the graphical technologies available to Koller et al. at the time. In 2008, many applications still made use of the programmable graphics pipeline.

Figure 4.9: The original image (left) and a highly compressed JPEG of the same image (right).

Additionally, it may been in order to also transform normals appropriately.

In Seymour, geometric distortions are introduced into the rendered image in two ways, per-vertex and per-model. In both cases, the distortions are constrained such that no rendered vertex is more than M object units or N pixels away from its original position.[6]

In the per-vertex case, a psuedorandom displacement value is added to each vertex. The displacement value is calculated using the RGBA values of a randomly selected noise texture, wherein each channel varies independently. A pixel is chosen on the noise texture using the vertex's pre-existing texture coordinates. Then, the RGB values are normalized and used as a direction. The vertex is displaced $A * N$ units in direction norm(RGB).

In the per-model case, a psuedorandom transformation matrix generated as the composition of non-uniform scaling, translation, and rotation, in a random order, applied to the 3D digital model.

JPEG Compression

Jpeg compression is a lossy compression algorithm that, depending on the compression level, may introduce artifacts into the image. The quantization step in the compression algorithm results in the noise around high contrast areas such as edges and corners and the introduction of "blocking" artifacts, shown in Figure 4.9.

[6]Koller et al. also bounded the transformations by an additional object space constraint. However, to determine the global constraint, the object coordinates constraint must be converted to pixels and the min of the two values chosen. Thus, it is extraneous.

While studies have shown that forms are identifiable to experts even with high JPEG compression ratios, there are stricter limits in our application case [207]. Art historical inspection of ancient artifacts requires detailed examination. Therefore, even subtle noise introduced may call into question the accuracy of observations about the surface of the object. However, JPEG compression serves to greatly reduce the image size, which as is discussed in Section 4.4 represents a potential bottleneck for rendering speeds. Furthermore, as shown in Section 4.3, JPEG compression is a significant deterrent to SfM reconstruction attacks. Once again, a balance must be achieved, on a per application basis, between security and fidelity.

Light Perturbation

The lighting parameters for the scene also pseudorandomly perturbed. First, the position of each light is shifted pseudorandomly. As with the geometric distortion, this shift is constrained by a constant, M, in world space coordinates. Second, an additional random light is added to scene with a random position outside of the object bounds.

Session Monitoring

By monitoring the activity of clients, potentially malicious activity may be detected and the offending clients proactively forbidden from making further requests. This may be accomplished through the automatic analysis of server logs. Image-based reconstruction techniques often require large numbers of images as input as well as certain geometric arrangements of images. Requests may be monitored per session in order to match such patterns. However, not all reconstruction techniques fit this criteria and such monitoring may be circumvented by distributed requests over time and/or across different sessions. Thus, the security of the system must not rely on such monitoring. Session monitoring offers another distinct benefit, in that it may be used to protect the server from being overloaded with rendering requests, if a user were to misuse the system. Given that the automatic server logging is ancillary to the visualization task, and only a minor security degree any at best, it will not be treated in detail here.

4.3 Security

The security of the system was tested against to three of the most robust reconstruction attacks identified by Koller et al.: shape-from-silhouette, shape-from-shading, and structure-from-motion. Only a subset of the server defenses discussed in Section 4.2.3 were included in the tests: high-resolution noise, JPEG compression, and geometric distortion. The automatic frame gathering used could theoretically be detected and stopped by the session monitoring defense, as the frames were requested in quick succession by a single client. However, given a distributed attack, i.e. through the use of multiple clients gather frames in a coordinated manner, this defense may be defeated. Therefore, these attacks may be considered feasible.

For each reconstruction attack, the initial images used were gathered manually. Since each rendered frame is displayed in an HTML <img>element, these images may be easily downloaded from the client. The render requests for the images were recorded and used to automatically generate the corresponding images given changes to the server-side defense parameters.

4.3.1 Test Model

The model used for testing is a photogrammetric reconstruction of a bust of Antoninus Pius held at the Uffizi Galleries in Florence, Italy. The physical bust is 71 centimeters in height and consists of an ancient head and bust, likely dating to the 2nd century CE, measuring 34 centimeters in height affixed to a modern base measuring 37 centimeters in height. Some of the locks of hair are missing as well as part of the right eyebrow. The tip of the nose and a large portion of the base have been restored. Additionally, the surface has been cleaned with acids and scraped, especially the face [208, p. 47]. The bust is of exceptional quality, depicting an an aged and contemplative Antoninus Pius.

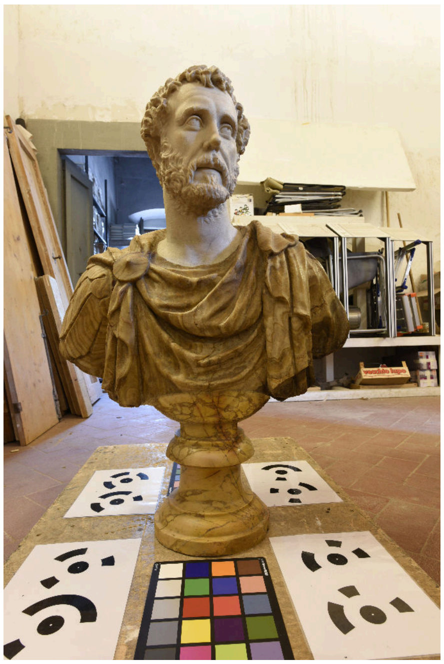
The model was produced using a set of 203 photos captured with a Nikon D810 camera with a Nikkor 24-85 mm lens and processed using RealityCapture. The initial model was 20 million triangles but was simplified to 12 million using RealityCapture's simplification process. The model is textured with a 4096x4096 pixel JPEG file. This model was chosen as an ideal candidate for reconstruction attacks given its rich detail both in terms of surface geometry and texture. The original bust and resulting photogrammetric model are shown in Figure 4.10.

The physical bust itself is currently in storage, in a remote facility owned by the Uffizi. Given the weight and location of the object, shooting conditions were not ideal. The object was photographed placed on the floor. As such, several of the bottom-most portions of the bust were not properly photographed and the bottom of the base was not photographed at all. As a result, several areas of erroneous texturing and geometry are visible on the bottom of the base and downward facing portions of the bust. Using Meshlab, the photogrammetric model was reduced to approximately 7 million triangles, the floor removed, and the resulting hole in the base was filled. The final model used for testing is shown in Figure 4.11. A reduced version of the model with accompanying metadata is also available online [209].

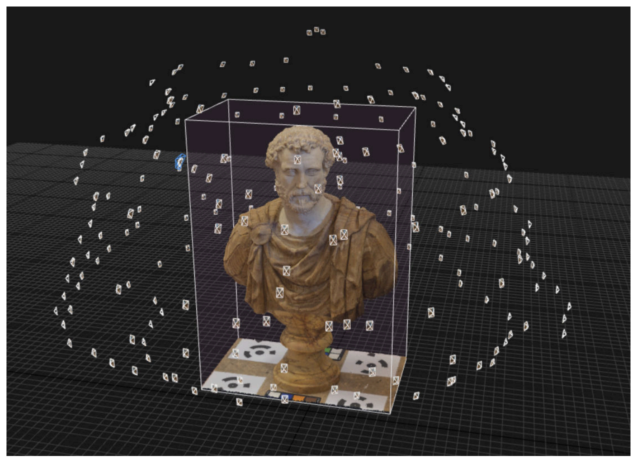
4.3.2 Shape-from-Silhouette

Shape-from-silhouette (SfS) is a shape reconstruction technique that uses silhouette images of an object to estimate 3D shape. SfS algorithms proceed by first segmenting the object of interest from the background to derive silhouettes, then projecting the silhouettes into 3D space and finding their intersection, and last computing the 3D reconstruction of the object. Typically, the final step is accomplished through the use of volumetric refinement, with the resulting volumetric model often converted to a polyhedral model [210, p. 284]. SfS was first proposed by Baumgart in his PhD thesis in 1974 [211], and since has proven popular for human pose estimation, motion capture [212], and autonomous navigation [213].

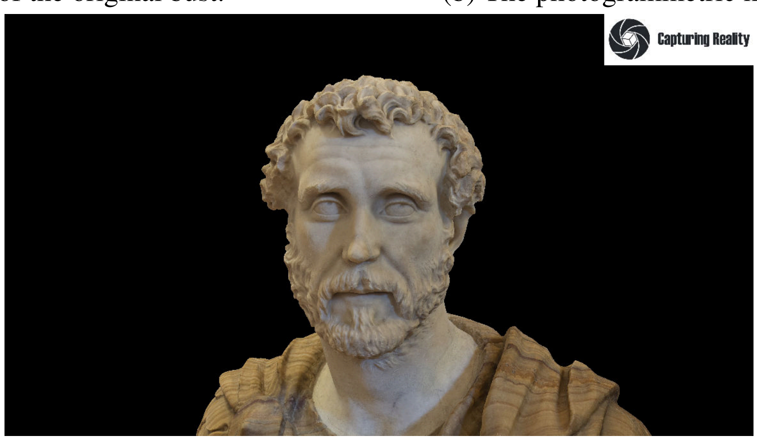
While SfS is effective for the fast and robust estimation of 3D shape from limited photo sets, it is ill-suited for accurate 3D reconstruction. Using the silhouettes, one may only reconstruct the convex regions of an object, what scholars have termed the “visual hull” [214]. In the case of 3D digital models of CH objects, very often the objects feature numerous concavities and thus are poor targets for SfS reconstruction. However, the coarse models obtained through SfS may be used as the basis for further refinement by more accurate algorithms like shape-from-shading [10, p. 7] and



(a) An image of the original bust.



(b) The photogrammetric model.



(c) Detail from the photogrammetric model .

Figure 4.10: Images from the original and photogrammetric model of the Antoninus Pius bust.

Figure 4.11: The test model of Antoninus Pius without texture (left) and with texture (right).

stereo correspondence [215]. It was for this reason that Koller et al. included SfS among their tested reconstruction attacks.

Koller et al. implemented the SfS algorithm of Tarini et al. [210] and demonstrated that perturbations to the viewing parameters were the most effective [10, p. 7]. This is due to the linear relation between error in viewing parameter estimation and 3D reconstruction in SfS [216]. In order to accurately project the silhouette, the camera position and orientation must be known. Likely, in Koller et al.'s implementation the camera position was derived from the viewing parameters sent to the server and was assumed to be unperturbed. Thus, any perturbations to the viewing parameters would directly increase the error in its estimation. However, the algorithm of Tarini et al. could be supplemented to include automatic camera pose estimation, similar to that discussed in Section 4.3.4.

Given the simplicity of the object-background segmentation, SfS is an exceptionally robust reconstruction technique. Especially with the addition of automatic pose estimation, one would expect such algorithms to perform well regardless of the server defenses outlined here. Yet even in the best case, SfS cannot fully reconstruct the types of detailed geometries typically found among digitized CH objects. Thus, the primary value of an SfS reconstruction to attackers is its use as a coarse model for additional refinement. However, there exists an additional attack vector for the acquisition of such a model, extraction of the coarse reference model from the client. Given the fact that the client of Seymour runs in a web browser, all files and code for the client are directly accessible to an attacker with minimal effort. While obfuscation may be used to impede the efforts of attackers to properly render and manipulate 3D files, similar efforts have had only limited success [217] [218]. Thus, one must all but assume that the attacker will be able to access directly the low-resolution reference model, whether directly or through the use of SfS.

4.3.3 Shape-from-Shading

Photometric stereo is a technique for the reconstruction of 3D surface information based on images with varied lighting conditions taken from the same view point. In a simple lighting model, the brightness of a surface is based on the surface normal (Section 3.2.1 for a description of surface normals) and the position of the light and observer. Using an equation assumed to accurately represent the lighting model, a known light position, and a known camera position, photometric stereo solves for the unknown surface normal. Photometric stereo performs best for surfaces whose material properties may be described most simply such as matte surfaces like rubber.

Photometric stereo in real-world settings is an exceptionally difficult task, as light position may often only be approximately determined, materials have complex properties, and reflected light complicates light source isolation. Photometric stereo has proven most popular in astronomy thanks to the dull surface of heavenly bodies, like mars and the moon, and the simple single lighting setup, the sun [219]. To facilitate such work, NASA offers extensive open source tools [220]. Other successful efforts have made use of laboratory settings augmented with reference objects (spheres with the same material as the objects imaged) [221] and custom lighting and camera setups [222].

Koller et al. discuss photometric stereo as a special case of shape-from-shading and implement an algorithm that uses the computed normal to refine a low-resolution mesh with the computation of a displacement map. Perturbation of viewing parameters and the complication of the lighting parameters were most successful against the photometric stereo as implemented by Koller et al [10, p. 8]. This is to be expected, as both directly effect the system of equations solved in computing photometric stereo.

Since the basic photometric steroe algorithm implemented here requires precise recordings for pixel colors, a lossless image format (PNG) was used for the initial "defenseless" tests. Realistically, PNG images are typically too large for attaining the transmission speeds required for an interactive Seymour session.

Error Test

Photometric stereo directly reconstructs surface normals, which may then be used to generate displacement maps used for the refinement and/or reconstruction of 3D geometry. However, this second steps involves additional assumptions and approximations, in essence a derivative of the

Figure 4.12: The normal map for the test model presented in Section 4.3.1.

information directly sampled from the original and its rendered images. Therefore, the error was tested in reference to the reconstructed normals, so as to forgo any additional error introduced in the 3D approximation.

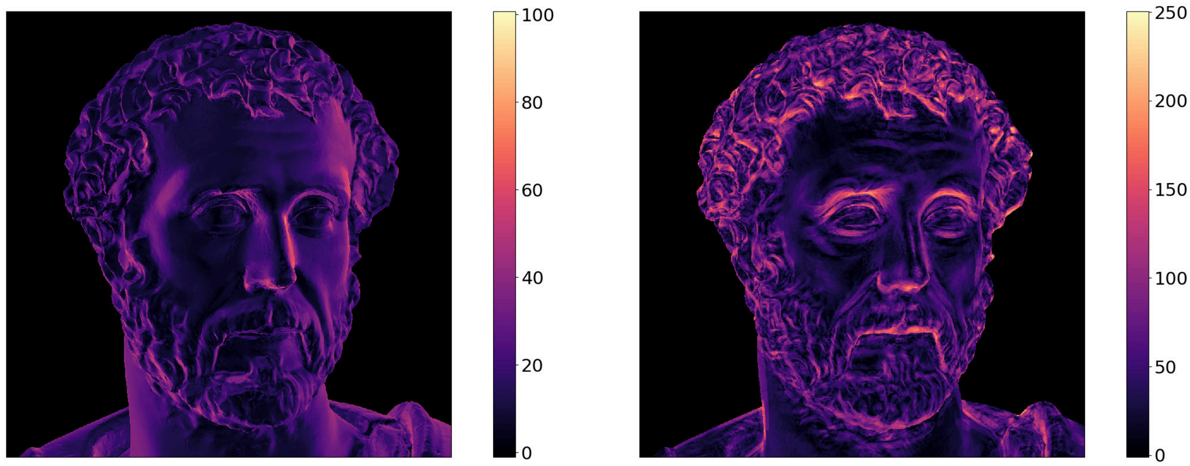
In computer graphics, normals may be represented independently from a 3D model in the form of a normal map. A normal map for a rendered view of our test model is shown in Figure 4.12. A normal map is produced by mapping the xyz coordinates of the normals to rgb colors. Color channels for the PNG files used in reconstruction and display of the normal maps are specified on a scale of 0 to 255, 255 being the maximum color for a channel. Thus, the normal range [-1, 1] is mapped to the color space range of [0, 255]. Normal maps were produced in this way for the original model and the subsequent reconstructions.

As a measure of error, the Euclidean distance was computed between normals in color space, i.e. 0 to 255 rgb values. In this case, the maximal error possible between normals is that between two normals with the opposite direction:

$$(-1, 0, 0)_{xyz} \rightarrow (0, 127, 127)_{rgb}$$

$$(1, 0, 0)_{xyz} \rightarrow (255, 127, 127)_{rgb}$$

$$\sqrt{(255)^2 + 0 + 0} = 255$$



(a) No server defenses.

(b) JPEG compression quality of 40, a random added light, random perturbation to the existing light, and geometric distortions up to 0.05 object units.

Figure 4.13: Heatmaps visualizing the difference between the original normal map and reconstructed normal maps.

Reconstruction

A basic photoetric stereo algorithm was implemented based on a Lambertian reflection model. If the camera position remains constant, then for a given light position L, normal n, and albedo k, the intensity I may be calculated as:

$$I = k(L \cdot n)$$

Given sufficient observed light positions (greater than 3), we may isolate the unknown k and n values as follows:

$$L^T I = L^T k(L \cdot n)$$
$$(L^T L)^{-1} L^T I = kn$$

The final equation may be solved using least-squares. The algorithm was implemented in C++ using image and matrix maths libraries. All code for the photometric stereo tests is available at https://github.iu.edu/leifchri/photometric-stereo.

Reconstructions were produced varying the JPEG quality, light perturbation, and geometric perturbation per-model server defenses. The measured errors are shown in Figure 4.14. As can be seen in the figure, even the undefended server results in some error. This is likely a result of incomplete coverage of the lighting positions sampled. Furthermore, as can be seen in Figure 4.13, the closer a normal is to orthographic to the camera, the more difficult it is to reconstruct. Therefore, error is concentrated around the edges of the model and pronounced ridges and valleys.

While the perceptually subtle defenses do contribute to an increased error, the visual impact on the reproduced model is minimal. Figure 4.15 shows a comparison of the original 3D geometry and renderings using the reconstructed normal map to generate new lighting conditions. Even the undefended reconstruction results in a perceivable softening of edges. The most impactful defense is geometric distortions. However, perceptually low-impact geometric distortions, in this case 0.01 object units, result in only minor degradation to the reconstructed model, as the magnitude of the perturbations in the reconstruction correspond directly to the magnitude of the distortions. Thus, while the geometric distortions of 0.05 object units had a profound impact on the reconstruction, this distortion may also be distracting and disorienting for the user.

The use of geometric distortions to defend against shading based reconstructions pose an additional issue. Namely, when perturbing the geometry based on lighting parameters, the transformation and placement of the camera remain constant. Therefore, statistical techniques may be used to average out the distortion, thereby circumventing the defense for 3D based reconstruction techniques such as SfM, discussed in Section 4.3.4.

Given more sophisticated photometric stereo techniques, one would expect even better performance against the defenses tested here. For example, the effectiveness of many of the lighting based defenses rely on the attackers assumption that the lighting parameters are unperturbed. If an attacker were able to add a model to the scene for which the normals were known, for example a sphere, the light position(s) could be computed based on the illumination of the known object, as in [221]. Furthermore, one could make use of feature matching techniques, similar to those in Section 4.3.4, to correct for changes in viewpoints.

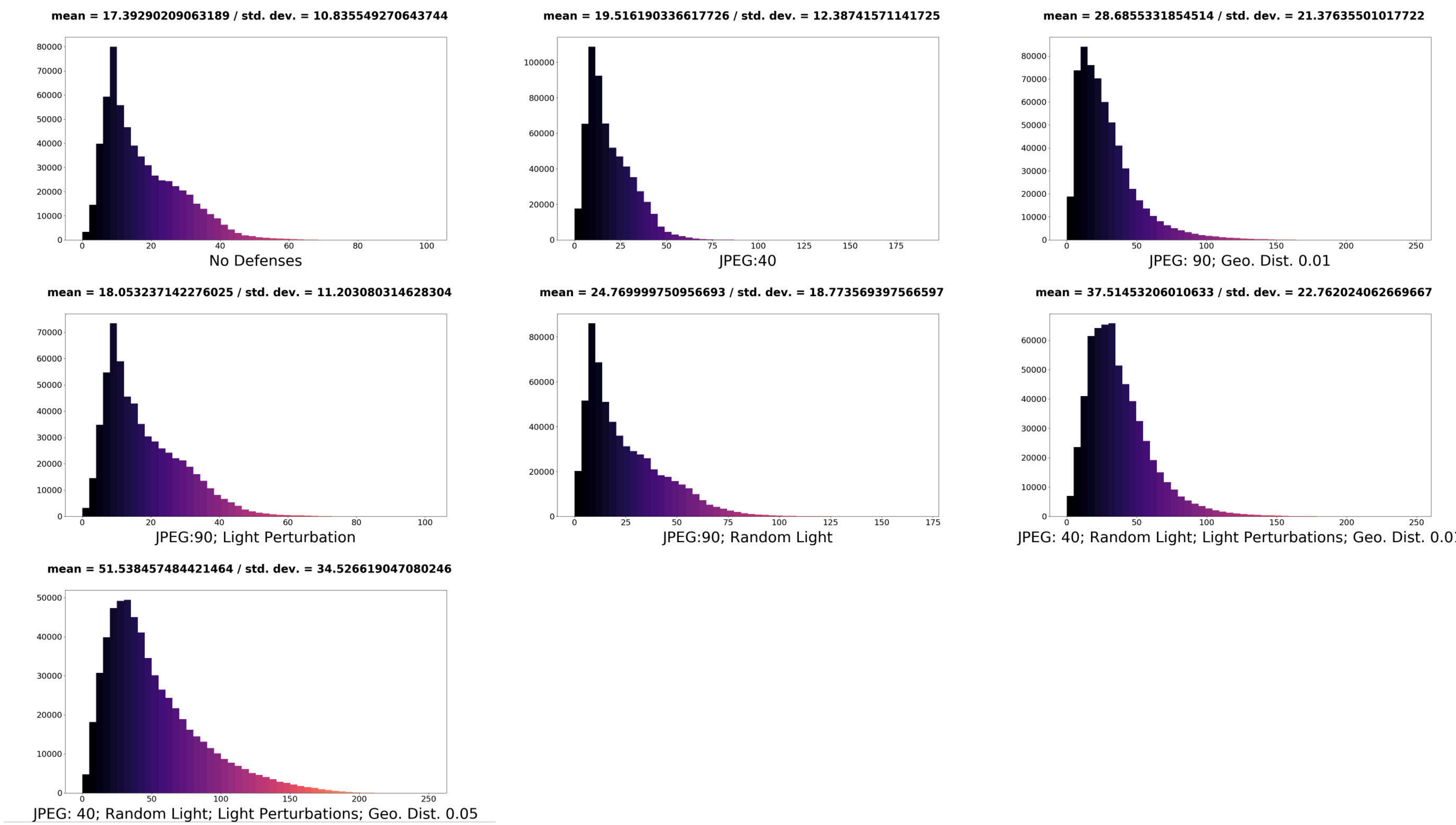


Figure 4.14: Histograms showing the distribution of distances calculated between the original normal map and reconstructed normal maps.

Figure 4.15: From left to right: the original 3D geometry, the reconstructed normal map without server defenses, the reconstructed normal map with all server defenses.

4.3.4 Shape-from-Motion

As discussed in Section 3.2.2, structure from Motion (SfM) is a photogrammetric technique used to derive 3D geometric information from a series of two-dimensional images. Defined in its broadest sense, photogrammetry is the art and science of taking measurements from two-dimensional images. While numerous methods fall under the purview of photogrammetry, SfM is by far the most prevalent method utilized in VH and scholars in the field often use photogrammetry as synonymous with SfM.

SfM proceeds by finding easily identifiable portions of the scene, called “features”. Features are identified using algorithms called “feature detectors”, which look for high contrast areas of an image, such as corners or edges, and seek to describe them in a position and scale invariant form. Next, occurrences of matching features are identified within the set of images, these matches are referred to as “correspondences”. Feature matching is an approximate process, and heuristics for determining matches are often employed. Given a sufficient number of correspondences, the motion of features between images may be used to solve for the exterior orientation of the camera (i.e. rotation and distance from the photographed point), called extrinsic parameters, and the internal projective distortions of the camera (i.e. the effect of the lens), called intrinsic parameters. Once the extrinsic and intrinsic parameters have been found, points in the images may be re-projected into

3D space, resulting in a cloud of points. The point cloud may then be converted to a surface model and a texture from the aligned cameras applied to the surface.

Numerous software programs, both proprietary and open-source, exist to perform some or all of the aforementioned steps of SfM. SfM has proven especially popular among those working with CH, as end-to-end solutions such as Agisoft's Metashape [223] and Capturing Reality's RealityCapture [224] allow for the creation of highly accurate 3D models of physical objects with minimal technical knowledge as to the underlying algorithms. Given SfM's reliance on the identification of features and correspondences, for the method to be effective input images must have sufficient detail and overlap. For example, highly uniform or specular surfaces often produce poor SfM results. In contrast, matte, detailed, and rough surfaces often produce good SfM results. Other causes that lead to a lack of sharpness and clarity in the image or a lack of sufficient resolution, can further degrade results. Finally, changes to intrinsic camera parameters, such as focal length, or the inclusion of multiple cameras can complicate finding solutions for re-projection. Ultimately, many of the major factors that may negatively impact SfM results are a product of human error in camera operation.

In our case of reconstruction attacks, SfM will be applied not to physically produced images but rather synthetically produced images. Synthetic images present a simpler problem in terms of SfM. Firstly, synthetic images do not suffer from any of the operator errors, a fact that led Scharstein and Szeliski to remark in their survey of stereo correspondence algorithms, another photogrammetric technique, that current synthetic image data sets were inadequate for benchmarking [225, p. 25]. Secondly, for images generated synthetically, the intrinsic parameters, i.e. the projective transformations from camera space to image space, may be known exactly. In the case of physical cameras, manufactures do release profiles of camera model's intrinsic parameters. However, as a physical device, no two cameras are exactly identical, and some refinement and approximation is necessary in the determination of intrinsic parameters.

Koller et al. do not provide experimental results for SfM reconstructions given the lack of support for textured models in ScanView. Thus, if one were to desire more assured defense against SfM reconstruction attacks, the texture support for Seymour could be removed. But as previously remarked, the detailed textures of photogrammetric models contribute significantly to their visual quality and usefulness. Luckily, the security measures of Seymour are able to provide protection from malicious SfM reconstructions of textured models.

Error Test

Following Koller et al., error was computed through the comparison of the original mesh to the reconstructed mesh and the statistical distributions of the measured errors reported.

Once a reconstruction model was created, the error from the original was calculated using the free and open-source CloudCompare software [226]. CloudCompare, originally developed for use with point clouds, supports a range of editing, analysis, and comparison algorithms for both polygonal meshes and point clouds. In our particular case, CloudCompare was used to align the original and reconstruction models with maximum theoretical overlap and then compute distances between the two.

Image-based reconstruction techniques do not record a real-world scale or intrinsic orientation. Thus, there is no guarantee, and likely one would not expect, that the reconstructed model will be oriented nor scaled as the original. The alignment step was accomplished through Iterative Closest Point (ICP) matching. ICP is a widely used algorithm for the alignment of 3D digital models, which iteratively refines the orientation of a model in reference to another to minimize some distance function between the two. CloudCompare implements Zinser et al. [227] who extend the standard ICP algorithm to include scale refinements, useful for SfM comparisons that have no set scale. For a complete step-by-step of the mesh comparison workflow, see Appendix B

The distance measure is computed using CloudCompare's "Cloud-to-Mesh Distance" tool [228]. This tool finds the nearest neighbor point, using Euclidean distance, on the "Reference" surface to each vertex from the "Compared" model [229]. See Figure 4.16 for an example. Thus, the number of comparisons is equal to the number of vertices in the Compared model. The results may be inaccurate given a Reference model with sparse vertices. In our case, the Reference model is constant and of high resolution, so all values are consistent with one another. Vertex counts and

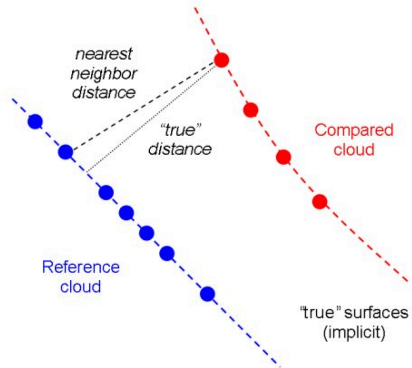


Figure 4.16: An example distance calculation from CloudCompare [230].

distributions are given for all distance calculations.

Finally, as described in Section 4.3.1, the model used for testing (the Reference model in CloudCompare parlance) is normalized such that it's maximum dimension, its height, is constrained to [-1, 1]. Metrics for the dimensions of the model are shown in Table 4.5. All error measures are reported in object units. Given the 71 cm height of the object, 1 mm corresponds to approximately 0.0028 object units.

Direct comparison to the error measurements of Koller et al. is problematic. Koller et al. report error as the mean surface distance in mm from the 5 m tall original model. First, Koller et al. do not report an alignment step. Second, Koller et al. do not give full documentation for their error calculation. Thus, it is unclear whether distance calculations are performed vertex to vertex or whether one of the two models is subsampled, as is done in CloudCompare. Finally, in solely reporting the mean, Koller et al. have the potential to misrepresent data due to outliers.

As a sanity check, an initial comparison was run between the original model and a copy of the original model that had been scaled and rotated. The resulting mean computed distance was 0.000027 object units. Being a heuristic algorithm, ICP does not guarantee complete overlap and this is evinced by the minuscule error evident in the identical comparison. Therefore, we may conclude that all error measures reported in the following section are accurate to within ±0.000027 object units or 10 micrometers.

Mesh Bounding Box Size 1.403444 1.988369 0.694079
Mesh Bounding Box Diag 2.530813
Mesh Bounding Box min -0.710596 -0.991671 -0.412465
Mesh Bounding Box max 0.692848 0.996698 0.281614
Mesh Surface Area is 5.317668
Mesh Total Len of 10500789 Edges is 13838.146484 Avg Len 0.001318
Mesh Total Len of 10500789 Edges is 13838.146484 Avg Len 0.001318 (including faux edges))
Thin shell (faces) barycenter: -0.001537 -0.023923 -0.064629
Vertices barycenter -0.013317 0.095963 -0.045724

Table 4.5: Metrics for the dimensions for the test model shown in Figure 4.11. Triples are reported as x, y, z values. Values were computed using the 'Compute Geometric Measures' tool of Meshlab.

Reconstruction

SfM tests were conducted using the commercial RealityCapture photogrammetric software. While proprietary, closed-source solutions do pose issues in terms of reproducability, in this case it was deemed a necessary compromise as the results of RealityCapture were far superior to other non-commercial and commercial software packages. Results of the photogrammetric reconstructions are shown in Figure 4.17. Tests were primarily conducted using an unlit rendering, as this produced significantly improved textures. Input images were 4096x4096 pixels.

In order to speed processing time and more accurately represent a malicious SfM reconstruction attack, JPEG images were used even in the "defenseless server" case. Theoretically, a lossless image format like PNG could yield improved results. However, processing such an image set would take an inordinate amount of time given the numerous SfM reconstructions run here. Furthermore, as previously discussed, PNG images would typically not be used in a realistic deployment of Seymour.

Meant to act in a scene in which the object, lighting, and projection are fixed, modern SfM techniques are fairly robust given changes in lighting. Thus, the distortions that we may expect to be most impactful are those that directly muddle the detection of features, such as the smoothing of edges by JPEG compression and the introduction of image noise, as well as those that directly affect the geometry.

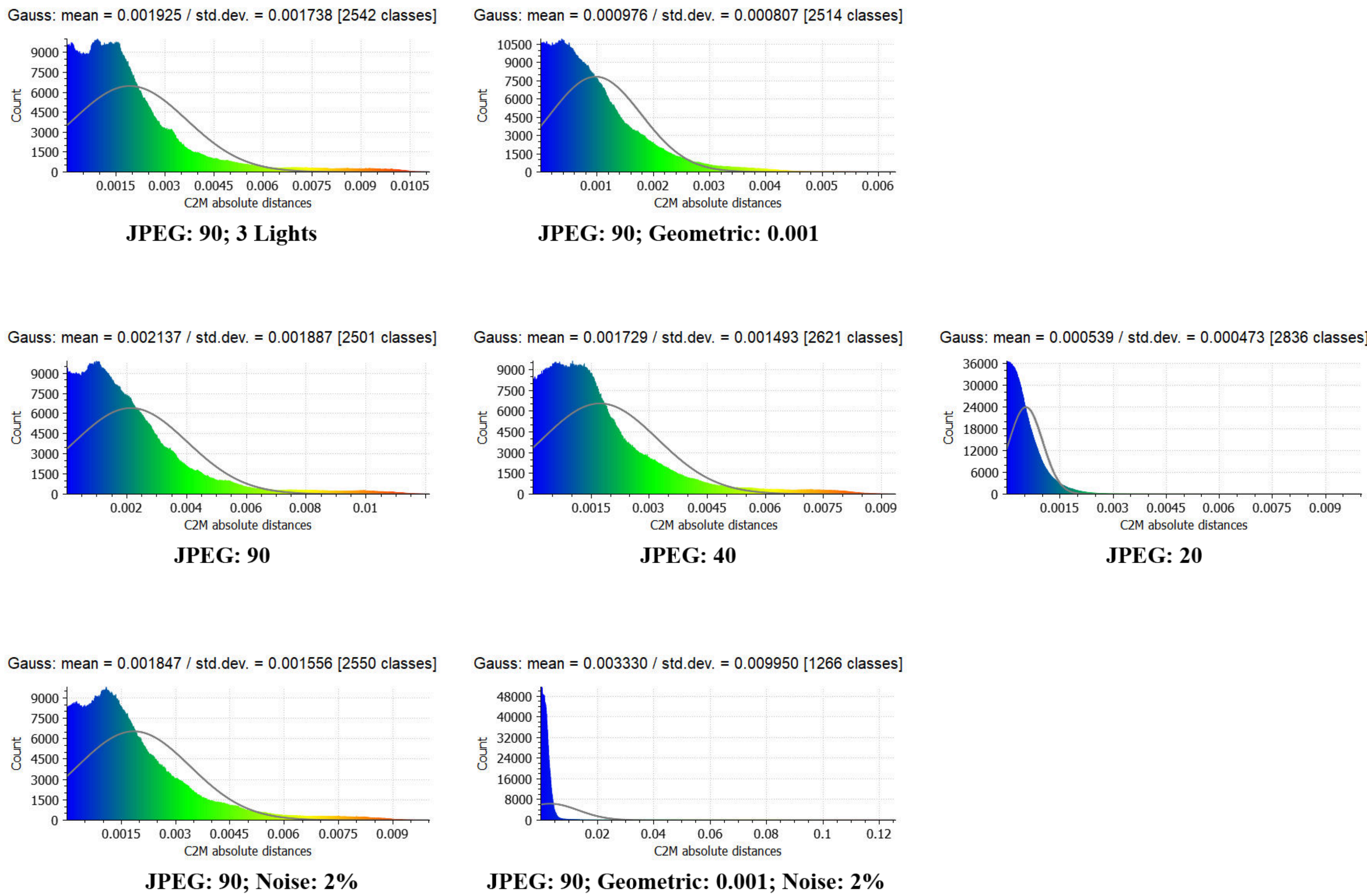


Figure 4.17: Histograms showing the distribution of distances calculated using CloudCompare with varying JPEG quaity, geometric distortions (in object space), and blended high-frequency noise.

Surprisingly, the inclusion of additional server defenses does not increase the error, at least as measured by CloudCompare. To the contrary, the overall error measure is lessened with increased defenses. However, a visual comparison of the models, shown in Figure 4.18 and Figure 4.19, does show degradation, especially in the case of high JPEG compression. It appears that while the undefended server reconstruction has significantly more detail, the smoothing effect introduced by the server distortions results in a model, on average, closer to the original. With distortions kept within limits unobtrusive to the observer, JPEG compression appears to have the most pronounced effect on the SfM reconstruction. However, when defenses are combined, it leads to a significantly degraded reconstruction.

Unfortunately, none of the defenses tested here had a significant impact on the texture quality of the reconstruction. In all cases, the textured models looks quite close to the original, although some noticeable artifacts are evident in the JPEG 20 reconstruction. While increasing levels of high-frequency noise will directly degrade the texture, they will be equally observable in the rendered frames by the user. More development is needed to effectively protect reconstruction of the model texture. One potential avenue for future development is the use of adversarial machine learning techniques. Using machine learning, scholars have been able to generate distortions to both 2D images and 3D printed objects that are able to fool algorithms meant to classify such objects [231]. Potentially, similar techniques could be leveraged to generate distortions to both the rendered images and underlying 3D geometry that could serve to disrupt the feature detection and matching algorithms underlying SfM reconstruction techniques.

4.4 Performance

The performance of Seymour was tested both in regard to speed and support for concurrent users. All tests were run using Nginx/1.14.0 server running on an MSI laptop with Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHZ, 32 GB RAM, NVIDIA GP104M GeForce GRX 1070 , and Ubuntu 18.04.2 LTS functioning as both client and server.

Figure 4.18: The reconstructed models using SfM without textures. Server defenses were as follow:

A: Original model

B: JPEG: 90; 3 lights

C: JPEG: 90

D: JPEG: 90; Geometric Distortion: 0.001

E: JPEG: 90; Noise: 2%

F: JPEG: 40

G: JPEG: 20

H: JPEG: 40; Geometric Distortion: 0.001; Noise: 2%

Figure 4.19: The reconstructed models using SfM with textures. Figures are labeled as with Figure 4.18.

4.4.1 Render Speed

The speed of a Seymour rendering instance was timed in a granular way, recording measurements at each step of the rendering process. These recordings are shown in Table 4.6 – Table 4.9. The renderer setup was varied according to three factors: size of the rendered image, use of high-frequency distortion, and JPEG quality. This factors were identified as having, potentially, the greatest impact on render time. However, experimental tests showed that changes in JPEG quality only had a minor impact on the total time. Increasing the size of the rendered image did have an impact on render time, as it directly effects the most time intensive step in the rendering process, reading of the pixels from the framebuffer. Excluding the high-frequency noise texture, Seymour is able to achieve render speeds that could theoretically support close to 30 frames per second. Including the noise textures the speed drops below this threshold. However, given the particular implementation of Seymour, and its selective transmission of frames, the render speeds are more than adequate. The averages for the tests are shown in Table 4.12.

Time was measured at 8 intervals beginning with parsing of the request from the client until the final cleanup following the transmission of a rendered image. As can be seen in the tables, in general, the actual rendering itself only accounts for a small percentage of the overall time spent processing a client request. The most time intensive operations are in loading the high-resolution noise texture for the server-side defense described in Section 4.2.3 and image readback. The former is only a necessary operation depending on the security requirements of the Seymour deployment. Furthermore, this stage could be made more efficient using lower resolution noise textures, for instance 512x512 or even 256x256 pixels. The image readback is comprised of 4 steps: reading the pixels from the framebuffer; flipping the pixels, since screen space does not match with the pixel order used for the JPEG codec; compressing the pixels to a JPEG image; and streaming the image back to the client. Reading pixels back from the framebuffer is an operation with little room for improvement, as it is physically limited given the transmission speeds from the GPU to the CPU.

Parse URI	Random Matrix	Render	Read Pixels	Flip Pixels	JPEG Write	Stream Image	Cleanup	Full Render
0.057	0.001	0.93	22.196	2.187	1.346	0.014	0.046	26.782
0.028	0	0.253	14.022	2.772	1.763	0.023	0.034	18.9
0.085	0.002	0.572	11.619	3.533	1.625	0.019	0.033	17.498
0.084	0.001	0.677	11.777	7.388	2.084	0.027	0.042	22.089
0.082	0.002	0.483	12.166	4.652	1.988	0.018	0.034	19.433
0.022	0	0.234	12.643	2.238	1.177	0.016	0.067	16.402
0.059	0	0.253	13.512	2.354	2.894	0.019	0.028	19.125
0.047	0.001	7.707	17.144	3.789	1.703	0.026	0.035	30.46
0.022	0	0.173	14.304	2.857	2.746	0.019	0.063	20.19
0.087	0.001	0.655	11.284	10.937	2.613	0.025	0.04	25.652
0.088	0.001	0.643	21.591	4.271	1.676	0.02	0.037	28.337
0.066	0.001	0.483	17.608	5.958	2.766	0.025	0.042	26.956
0.081	0.001	0.456	18.096	3.749	2.5	0.018	0.033	24.941
0.082	0.001	0.506	11.711	7.459	2.679	0.036	0.053	22.535
							Average	22.807

Table 4.6: Render times in ms for 15 frames with JPEG quality 40, 512x512 pixels, and no blended high-frequency noise.

Parse URI & Load Noise	Random Matrix	Render	Read Pixels	Flip Pixels	JPEG Write	Stream Image	Cleanup	Full Render
51.784	0.004	0.807	23.522	2.082	1.405	0.107	0.008	79.724
53.665	0.002	0.237	13.93	2.099	1.56	0.1	0.022	71.621
46.725	0.002	0.155	12.512	2.039	1.211	0.017	0.047	62.714
46.268	0.001	0.149	20.894	2.115	1.1	0.012	0.04	70.585
26.436	0.002	0.192	21.968	2.22	1.1	0.013	0.045	51.983
26.8	0.001	0.161	20.339	2.321	1.509	0.03	0.026	51.194
57.733	0.002	0.159	23.623	2.039	1.399	0.105	0.023	85.088
38.281	0.001	0.156	18.418	2.123	1.246	0.057	0.007	60.296
43.088	0.002	0.165	13.711	2.148	1.234	0.07	0.025	60.448
58.6	0.002	0.156	18.101	2.427	1.392	0.057	0.009	80.75
26.117	0.002	0.172	18.988	2.257	1.236	0.039	0.029	48.846
54.809	0.002	0.167	16.646	2.159	1.261	0.09	0.026	75.166
28.787	0.002	0.156	26.249	2.176	1.4	0.074	0.008	58.856
37.663	0.002	0.205	35.016	2.312	1.545	0.075	0.008	76.832
							Average	66.722

Table 4.7: Render times in ms for 15 frames with JPEG quality 40, 512x512 pixels, and blended high-frequency noise.

Parse URI	Random Matrix	Render	Read Pixels	Flip Pixels	JPEG Write	Stream Image	Cleanup	Full Render
0.125	0.005	2.42	26.018	4.191	1.916	0.073	0.021	34.777
0.062	0.001	0.673	11.552	3.992	1.826	0.067	0.012	18.193
0.061	0.001	0.422	11.478	3.406	1.397	0.047	0.009	16.826
0.074	0.001	0.475	20.624	4.533	1.852	0.065	0.012	27.643
0.085	0.001	0.611	18.331	3.407	1.662	0.049	0.01	24.163
0.08	0.001	0.485	11.633	3.482	1.711	0.051	0.01	17.461
0.053	0	0.218	13.717	2.214	1.131	0.04	0.005	17.384
0.022	0	4.729	11.971	2.414	1.103	0.037	0.006	20.289
0.051	0	0.326	12.185	3.106	1.303	0.046	0.009	17.033
0.088	0.002	0.781	11.476	4.558	1.681	0.054	0.011	18.66
0.086	0.002	0.503	11.976	6.192	2.737	0.082	0.019	21.604
0.076	0.001	0.46	11.705	8.197	2.965	0.087	0.019	23.52
0.064	0.001	0.482	21.602	4.113	1.828	0.06	0.012	28.169
0.059	0.001	0.506	15.236	5.085	2.194	0.071	0.014	23.172
							Average	22.063

Table 4.8: Render times in ms for 15 frames with JPEG quality 90, 512x512 pixels, and no blended high-frequency noise.

Parse URI & Load Noise	Random Matrix	Render	Read Pixels	Flip Pixels	JPEG Write	Stream Image	Cleanup	Full Render
27.001	0.004	0.723	26.684	2.22	2.096	0.115	0.011	58.861
26.433	0.002	0.232	14.945	2.081	1.281	0.057	0.007	45.046
48.915	0.001	0.157	77.132	2.283	1.419	0.044	0.026	129.982
46.906	0.002	0.172	29.147	2.278	1.34	0.113	0.022	79.987
48.065	0.002	0.157	31.063	2.027	1.883	0.077	0.008	83.287
43.744	0.002	0.221	20.832	2.242	1.347	0.057	0.026	68.477
46.848	0.002	0.195	25.598	2.257	1.353	0.102	0.025	76.386
26.74	0.002	0.156	19.904	2.153	2.785	0.045	0.008	51.8
26.922	0.002	0.189	18.242	2.835	1.23	0.093	0.008	49.527
27.627	0.001	0.164	19.679	2.752	1.181	0.066	0.021	51.497
29.336	0.002	0.279	33.56	2.256	1.694	0.088	0.009	67.23
49.143	0.002	0.16	32.423	2.064	1.185	0.072	0.007	85.062
52.865	0.002	0.161	18.763	2.284	1.3	0.057	0.026	75.464
26.865	0.002	0.161	17.657	2.557	1.956	0.055	0.007	49.265
							Average	69.419

Table 4.9: Render times in ms for 15 frames with JPEG quality 90, 512x512 pixels, and blended high-frequency noise.

Parse URI	Random Matrix	Render	Read Pixels	Flip Pixels	JPEG Write	Stream Image	Cleanup	Full Render
0.068	0.004	1.921	24.898	10.155	4.514	0.127	0.01	41.705
0.085	0.002	0.604	19.78	10.909	4.513	0.109	0.007	36.018
0.028	0	0.188	21.134	9.08	4.089	0.138	0.005	34.666
0.088	0.001	0.528	20.242	11.282	5.145	0.2	0.01	37.505
0.025	0	0.192	68.86	10.669	4.413	0.079	0.008	84.251
0.03	0	0.283	15.145	8.791	4.85	0.1	0.024	29.228
0.062	0	0.262	14.145	9.028	4.888	0.168	0.007	28.565
0.035	0	0.182	14.489	9.697	4.962	0.071	0.007	29.448
0.059	0.001	0.467	22.657	9.568	4.807	0.066	0.007	37.639
0.082	0.002	13.844	15.402	9.983	4.42	0.061	0.012	43.816
0.045	0	0.208	15.45	9.684	4.339	0.06	0.006	29.797
0.024	0	0.195	13.92	9.8	5.129	0.124	0.007	29.204
0.029	0	0.209	16.085	8.857	4.383	0.093	0.005	29.667
0.09	0.002	0.614	13.403	14.234	4.659	0.069	0.023	33.102
							Average	37.472

Table 4.10: Render times in ms for 15 frames with JPEG quality 90, 1024x1024 pixels, and no blended high-frequency noise.

Parse URI & Load Noise	Random Matrix	Render	Read Pixels	Flip Pixels	JPEG Write	Stream Image	Cleanup	Full Render
37.002	0.004	0.764	15.135	9.044	4.502	0.086	0.009	66.552
35.771	0.002	0.193	24.6	9.747	4.365	0.115	0.021	74.819
49.126	0.002	0.192	28.44	9.064	4.929	0.103	0.007	91.87
29.756	0.001	0.161	18.749	9.244	4.225	0.068	0.007	62.218
38.787	0.002	0.15	17.041	9.131	4.462	0.135	0.007	69.72
43.06	0.001	0.162	14.365	9.226	4.689	0.072	0.007	71.588
42.237	0.002	0.162	15.534	9.133	4.729	0.14	0.008	71.952
27.877	0.002	0.154	20.693	8.329	4.065	0.078	0.007	61.211
47.921	0.002	0.15	26.737	9.02	4.428	0.079	0.007	88.35
27.461	0.002	0.167	19.088	8.88	5.067	0.086	0.019	60.776
26.438	0.002	0.156	22.188	9.042	5.017	0.068	0.007	62.925
41.518	0.001	0.151	21.51	9.018	5.637	0.119	0.006	77.968
46.332	0.002	0.164	22.814	8.721	4.452	0.075	0.029	82.595
47.146	0.002	0.161	13.793	10.15	5.053	0.068	0.006	76.384
							Average	72.781

Table 4.11: Render times in ms for 15 frames with JPEG quality 90, 1024x1024 pixels, and blended high-frequency noise.

Pixels	Noise Blend	JPEG Quality	Full Render (ms)
512x512	No	40	22.807
512x512	Yes	40	66.722
512x512	No	90	22.063
512x512	Yes	90	69.419
1024x1024	No	90	37.472
1024x1024	Yes	90	72.781

Table 4.12: Average render times for the 6 renderer configurations

4.4.2 Time to Render

The speed with which the server renders the frame is but a component of the larger interaction process. Ultimately, the usability of the system rests on the latency between client requests and rendered frames on the client. In addition to the time spent rendering by the server, this also includes transmission time to and from the server and the time to display the received frame.

The speed of the system may be compared to similar techniques for the online rendering of high resolution models. 3DHOP [114] is one of the best performing progressive streaming systems for viewing high-resolution 3D digital models of CH objects. 3DHOP uses the Nexus [232] format to stream meshes, which typically results in a first rendering in less than a second. In Potenziani et al. [114], the authors demonstrated that 3DHOP outperforms similar progressive streaming solutions of Google's WebGL-loader and X3DOM's POP buffer.

As noted by Potenziani et al. [114], solutions that fully download the mesh such as Sketchfab and Unity may out-perform progressive techniques for small models and high network speeds. However, since our application is targeted towards larger meshes, this case need not be tested.

The rendering speed of 3DHOP was tested using the simplest setup from the project website, loaded using a 512x512 pixel Firefox 66.0 window [233]. The same 7 million polygon test model described in Section 4.3.1 was used, converted into the Nexus format. The Firefox "Network Monitor" [234] was used to measure the time and amount of data transferred. The page was loaded ten times, five using the full 60 MB/s connection and five using a throttled 4 MB/s connection with a minimum latency of 20 ms. The experiment was repeated several times given 3DHOP's adaptive

Network Speed	First Frame (ms)	First Transferred (KB)	Final Frame (ms)	Total Transfer (KB)
60 MB/s	709	736.6	5700	167300
	818	739.6	5510	167300
	679	739.6	5430	167300
	705	378.99	5440	16.7300
	775	739.6	5210	167300
Average	**737**	**666.8**	**5458**	**167300**
4 MB/s 20 ms latency	3680	1870	36870	181600
	3790	1910	36980	181700
	3720	1690	36870	181700
	3820	1910	36920	181700
	3810	1910	36960	181700
Average	**3764**	**1858**	**36920**	**181700**

Table 4.13: Render times for 3DHOP.

Network Speed	First Frame (ms)	First Transferred (KB)	Final Frame (ms)	Total Transfer (KB)
60 MB/s	193	15.14	263	340
	197	15.14	255	340
	137	15.14	209	340
	197	15.14	269	340
	130	15.14	200	340
Average	**170**	**15.14**	**239**	**340**
4 MB/s 20 ms latency	3260	1320	3340	1640
	3180	1310	3290	1630
	3170	1320	3310	1640
	3140	1340	3280	1640
	3110	1340	3310	1640
Average	**3172**	**1326**	**3306**	**1638**

Table 4.14: Render times for Seymour.

progressive streaming scheme, which modifies transmission parameters based off network and client parameters. The Network Monitor was used to simulate a typical 4G connection of 4 MB/s [235]. The elapsed time and amount of data transfered were measured at two points, at the first rendering of the geometry and after data transfer ceased. In both the full speed and throttled cases, the initial view will not represent the entriety of the refinements loaded by 3DHOP. Given 3DHOP's level-of-detail refinements based on resolution and viewing distance, zooming into the model results in the loading of additional patches. Finally, the download values reported do not include the initial downloads of the 3DHOP JavaScript library and its dependencies, which are around 500 KB in total, as they were already in the cache. The results of the test are shown in Table 4.13

With the fast, 60 MB/s connection, the time to the first rendered frame was close to the 500 ms render time reported by Potenziani et al. [114]. The higher render time may be a result of the high-resolution texture applied to the model, which is a large 19.3 MB, or possibly due to the fact that the test model used here is seven times the size of the one used in Potenziani et al. However, the progressive streaming scheme is meant to mitigate such factors. Whatever the issue, it is exacerbated given the throttled connection, resulting in an initial render time of over 3 seconds.

The measurement of the final frame is somewhat of a misleading metric, but is meant to demonstrate an upper bound on the amount of data transferred to the client. The entire rendering of the initial view of the test model with 3DHOP may take over 30 seconds with the throttled 4G connection. However, after only a handful of seconds, 3DHOP will have rendered a sufficiently refined mesh such that future refinements are all but visually unnoticeable. Furthermore, since 3DHOP transfers data based on the graphics capabalities of the client, and the client machine used in testing was a high-powered machine, patches were transfered up to a correspondingly high-resolution. Thus, the model is fully ready for interaction long before the final patch is transferred.

The same test, using the exact same server and client setup, was repeated with Seymour. In the throttled case, caching was disabled. The download size is contingent on the level of decimation of the reference model and the compression of the texture. In our test, the reference model was decimated to 5 thousand polygons and the texture reduced to a 512x512 pixel JPEG at quality 60.

Seymour and its dependencies account for aproximately 600 KB, 500 of which is three.js. However, as with 3DHOP, after one visit these libraries will be cached and no longer need to be downloaded. The results are shown in Table 4.14.

Seymour compares favorably to 3DHOP in both the time to first render and the amount of data sent. Given a heavily decimated reference model and cached JavaScript files, the full download for Seymour is exceptionally light. Seymour has the additional advantage that the initial rendering is at the complete resolution, unlike in the case of 3DHOP. However, once 3DHOP has loaded the model on the client, renderings are near instantaneous, whereas Seymour must make a new request for each frame. This may also result in larger amounts of data transferred over long sessions. In this case, a good compression ratio for the transmitted frames is crucial.

4.4.3 Concurrency

The concurrency of Seymour was tested by measuring the response time of a server with a single renderer instance as the number of simulated clients was increased. The clients requested 512x512 pixel frames of the 7 million polygon Antoninus Pius model without defensive distortions. Clients were simulated using Python 3.6.5 and the requests [236] and _thread [237] libraries. A single client would randomly request one of seven different frames, waiting a random amount of time between requests ranging from 0.1 to 2.0 seconds. The 0.1 to 2.0 second interval was chosen to simulate an extremely active user. More typical user interaction would likely involve longer periods of inactivity. This test was repeated twice, adding a client every 20 seconds up to a maximumum 30 cleints, shown in Figure 4.20, and adding a client every 5 seconds up to a maximum of 100 clients, shown in Figure 4.22. The initial maximum of 30 clients was chosen to repeat similar tests performed by Su et al. [151] for their remote rendering system, Protected-3 DMPS.

As can be seen in Figure 4.20, up to 30 clients the average time to receive a rendered frame stays fairly constant as additional clients were added, hovering between 20 to 40 milleseconds. While in this case the average time does not appear to increase as the number of clients is increased, there is an upward trend among the scattered requests with the highest wait times. Such scattered high

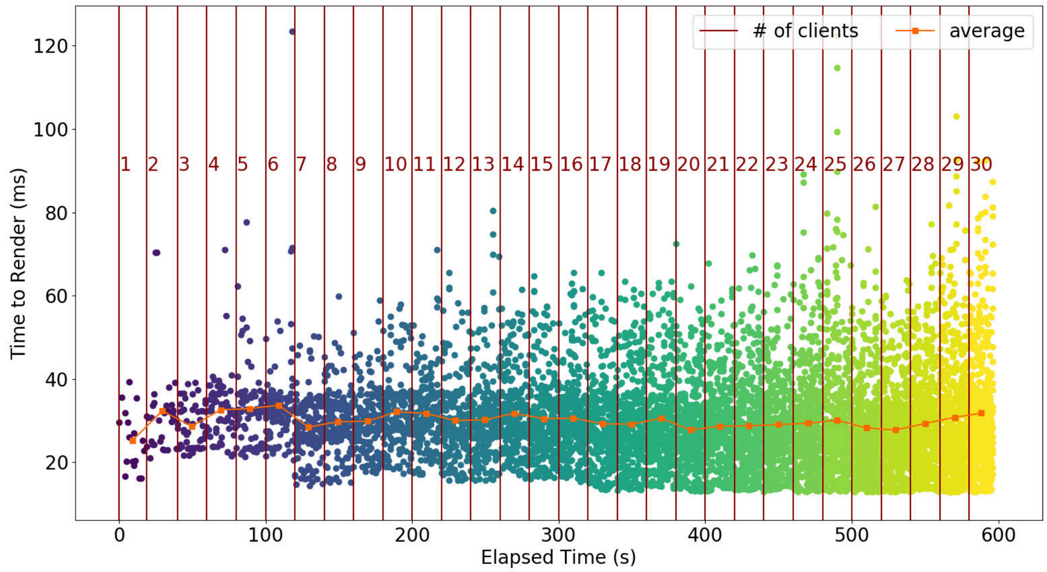


Figure 4.20: A plot of response times as the number of clients making requests is increased incrementally, up to 30 clients.

wait times are evident even with only a few clients. These points are likely a result of temporally overlapping requests for a rendered frame. Since the server renderer only processes one request at a time, concurrent requests result in a linear increase in processing time based on the number of requests, as later requests must wait for all previous requests to be served. These collisions become increasing likely as the number of active clients increases, thus increasing the average overall wait time. However, as can be seen in Figure 4.20, even 30 extremely active users result in relatively few collisions.

Su et al. [151] ran similar tests for the concurrency of their remote rendering system. According to their graph, the server of Su et al. rendered an 800x600 pixel frame of an approximately 900,000 polygon model in approximately 18 milliseconds. As the number of clients was increased, the "average total time of handling a request" increased linearly [151, p. 5]. Likely, the upward linear trend was a result of Su et al.'s simulated clients requesting rendered frames exactly concurrently. Whether this is the case is unclear from the article, as Su et al. only say that "every client continuously requests frames from the server" [151, p. 5]. However, experimental results with

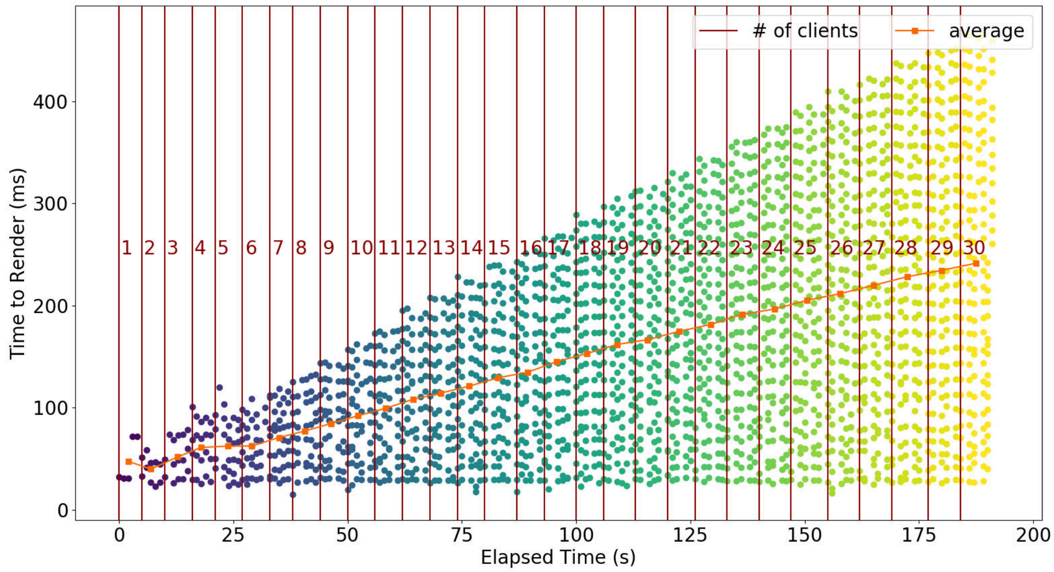


Figure 4.21: A plot of response times as the number of clients making concurrent requests is increased incrementally.

Seymour for concurrent requests seem to replicate the trend observed by Su et al.

Figure 4.21 shows a plot of response times again up to a maximum of 30 clients with clients added every 5 seconds. Once per second, every client requests a randomly selected frame concurrently. Concurrent requests were accomplished using the concurrent.futures [238] Python module to execute code in parallel. Figure 4.21 demonstrates a nearly perfect linear upward trend, just as the results of Su et al. The vertical stacks of requests represent concurrent requests, and the fairly consistent vertical spacing between requests in a stack represent the render plus transmission time of the server.

Thus, 30 concurrent clients seem to represent only a light load for a single Seymour rendering instance. Figure 4.22 shows the results of another concurrency test, this time with random wait intervals between requests, a maximmum of 100 clients, and clients added every 5 seconds. Based on the figure, it appears that a single instance is capable of serving approximately 70 concurrent clients without a significant dip in performance. Over 70 clients, the time to render begins to sharply increase, reaching wait times of over half a second and approaching a full second. This portion of

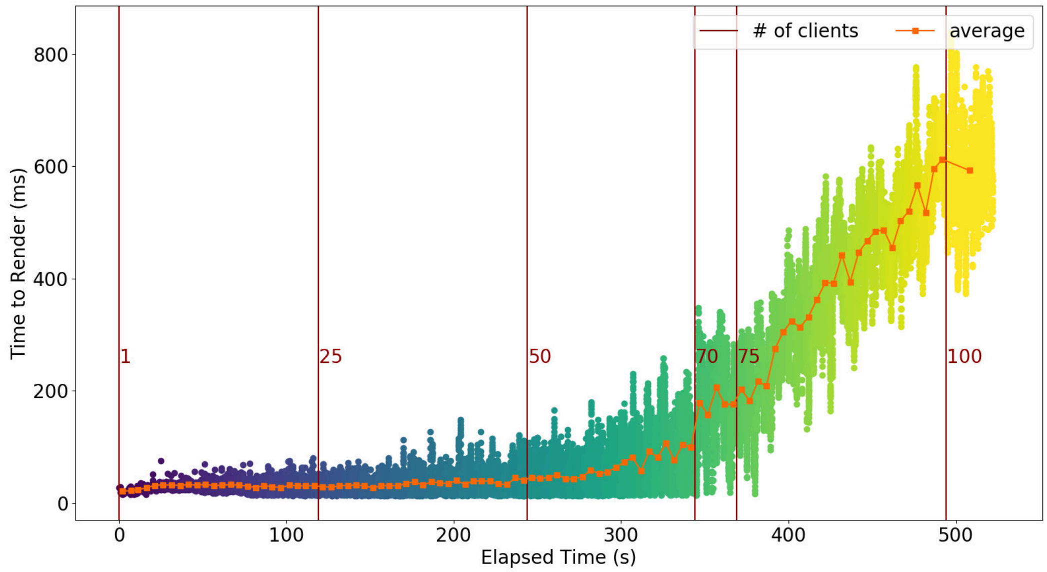


Figure 4.22: A plot of response times as the number of clients making requests is increased incrementally, up to 100 clients.

the graph shows an average with much higher variance than the previous portion. Likely, longer gathering intervals between adding clients would smooth the averages, but one would expect the upward trend to remain. Essentially, at around 70 client or so, the server reaches a point of saturation, wherein it is unable to serve all client requests before clients repeat there requests. Therefore, even the fastest rendering times begin to exhibit an upward trend.

Given the short server-side rendering time, and the natural intervals between client responses, A single Seymour rendering instance is able to serve a goodly number of concurrent users. Ultimately, this type of monitoring may be used to automatically scale server-side resources, creating and destroying rendering instances as demand fluctuates. However, in this case, one would expect actual client wait time between requests and received frames to be a far more accurate measure of quality of service, rather than the simple number of clients. In reality, clients would not exhibit such consistent patterns of request intervals, more likely showing periods of intense activity and inactivity. Furthermore, if server-side operations are expanded, as is done in Chapter 6, individual clients may demand more server resources than the tests run here.

4.5 Conclusion and Future Work

With Seymour, advances in remote rendering have been applied to the original proposal by Koller et al. for the protected viewing of 3D digital models of CH. Seymour allows for the delivery of high-resolution 3D graphics to low powered devices, all while providing interaction with demonstrable security against image-based reconstruction techniques. Furthermore, a single renderer instance of Seymour has the potential to serve nearly 100 active users concurrently. Such a system may be useful for those wishing to disseminate culturally sensitive or in-copyright works, as CH objects so often are, as well as deliver high-resolution 3D graphical content to a wider group of users without access to high-powered hardware. The work presented here represents a validation of the architecture presented by Koller et al. and its wider application to visualization in CH, which may hopefully serve as the foundation for the future development of visualization tools for VH. Chapter 6 demonstrates how Seymour may be usefully extended to better meet the needs of VH practitioners.

While Seymour addresses some of the most impactful areas for future work on protected remote rendering, there still remains ample room for future development. First, given the decoupling of the server and client, there is no intrinsic need for the client to have any 3D capabilities. Thus, Seymour could be extended as with Tapestry developed by Raji et al. [174] such that the client interacts with an image stream, as opposed to a low-resolution reference model. Second, as implemented here, Seymour only supports manually instantiated renderer instances and does not dynamically respond to increases or decreases in user activity. Therefore, load balancing functionality should be added to the communications module, instantiating and destroying instances of the renderer to meet demand. Third, while not treated in detail here, client session monitoring represents a useful defense against reconstruction attacks, and one for which there already exist well developed tools and methods. Techniques for anomaly detection and pattern recognition of potentially malicious activity in server log files deserves further attention. Finally, as discussed in Section 4.3.4, adversarial machine learning techniques pose a promising avenue for the future development of more finely tuned defensive distortions.

Ultimately, Seymour offers another visualization tool to VH scholars, one complementary to existing approaches such as 3DHOP, Unity, and Sketchfab. The novelty of Seymour lies in its explicit attention to security and the client-server architecture. Such an application is not always applicable, and in no way diminishes the usefulness of existing tools. Rather, Seymour expands the potential for VH applications and the ability to disseminate the valuable 3D assets of CH VH practioners produce. Most importantly, Seymour is one of a growing number of applications specifically targeted at visualization in VH, an area of study that deserves increased and concerted attention.

CHAPTER 5

CHARACTERIZING MODEL USE IN VH

Through our previous discussion of model use in VH, we have only just begun to document the wide variety of applications and scholars evident in the field. VH lies at the intersection of fields both old and new, and scholars approach research within VH from perspectives ranging across the entirety of this spectrum. In this chapter, we document the attitudes, practices, and experiences of a selection of these scholars through a survey and interviews, and distill these observations into a more actionable form. In so doing, developing instruments that may be useful in the further design and refinement of the visualization tool presented in Chapter 4. Thus, the goal of this chapter is to characterize model use in VH in order to elaborate the needs and practices of target users, for whom a more fully featured visualization tool will be developed in Chapter 6.

5.1 Previous Work

A number of previous publications have discussed the nature of VH and prescribed criteria for the scholarly use of 3D models therein. See Section 3.4 for a more complete discussion of these works. Largely, these publications have represented the experience of scholars with long-term engagement in VH. Some, like Fisseler et al. [140] and their research of cuneiform tablets, constrain their recommendations to a particular field or application. Others like Maschner [77], Richards-Rissetto and von Schwerin[100], and Frischer [47] make broader calls for the future of the field as a whole. Additional works, like the ICOMOS Charter [104] and London Charter [105] represent the combined and thoughtful codification of best practices for the use of 3D digital models of CH, as determined by a panel of selected scholars. While these works are certainly useful tools for practicing VH scholars, they represent VH at an highly abstract level. This is of course a purposeful characterization, as such works are meant to be broadly applicable across the field. Yet one is left to wonder as to the day-to-day practices of VH scholars.

The aforementioned works may be said to represent the culmination of published practice and knowledge within VH. However, this forms only a piece of the broader research practices underlying the field. Publications are the concise, curated, and clean output of a long messy process of research. The day-to-day lives and work of scholars does not follow the neat logical lines of a 10 page article. To more fully understand the field, one may look to describe in more detail the quotidian lives and practices of VH practitioners.

A handful of publications on VH have engaged more directly with practicing VH scholars. For four years beginning in 2006, Koller et al. [78] conducted a survey of VH scholars attending conferences and other professional events in the United States and Europe . Koller et al. were interested in gauging the perceived need of VH scholars for scholarly curated repositories of 3D digital models. The results of Koller et al. were certainly motivating, of the 54 individuals surveyed, 90.7% agreed there was a need for a central repository.

A decade later, Copy Culture: Sharing in the Age of Digital Reproduction [239] was published as the result of the Reproductions of Art and CH (ReACH) initiative. Through essays, interviews, and project profiles, Copy Culture attempts to better understand the opportunities and challenges for the use of digital media in the "everyday life" of professionals working with CH [240]. Copy Culture focuses on museums and archival institutions dealing with digitized content, such as images and 3D scans, and born digital content. Besides the exceedingly brief project profiles, 3D digital models are discussed in three sections: an interview with the founders of Cultural Heritage Imaging (CHI) about their metadata recording tool the Digital Lab Notebook, an essay by Adam Lowe from Factum Arte about the applications of 3D scanning to CH, and another essay by Diane Zorich from the Smithsonian Institute detailing their large scale scanning efforts.

Online community groups provide another important means for conglomerating, documenting, and refining the work of scholars and professionals. Some more general groups, like the MUSEUM-L listserv for museum professionals, have frequent postings related to 3D. Other groups, like ImageMuse and IIF 3D, are specifically dedicated to the discussion, development, and implementation of community standards for 3D. While the discussions of these groups certainly echo more broadly in the field, through the practice of members, various publications, and some in person events, their impact is still growing and there are few means, other than direct involvement with the groups, to review the findings of these groups comprehensively.

5.2 Methods

The VH community was explored based on two procedures: survey and interview. A subject was eligible to participate in the study if they were working in academia, either producing published works or working in a supporting role to scholars producing published works; they made use of digital 3D models in their work; and they were working on topics involving CH objects, for example architecture, art history, or museum studies. Subjects were recruited through direct emails, through posts in online forums and communities, and at professional events. In the case of the survey, snowball sampling was also be used; subjects were asked to share the survey with others that the original respondent believes may meet the selection criteria. Subjects were identified by their belonging to relevant online communities, for example a digital humanities listserv or a 3D modeling forum; by their attendance at digital humanities conferences; and through their online credentials like CVs and university pages. A selection and description of eligible groups and communities contacted for survey and interviews is shown in Table 5.1.

5.2.1 Survey

The survey was meant to, in a broad and coarse way, assess the state of 3D model use among VH scholars. More specifically, scholars were asked about the frequency of their 3D model production and use, the nature of said use, the technologies utilized when working with 3D models, and their perceptions of these technologies. The survey consisted of multiple choice and short response questions and took between 10-15 minutes. See Appendix C for the survey in its entirety. The survey was authored and administered through Qualtrics, a cloud-based survey tool [241].

The survey questions can be divided into roughly three topics: five demographic questions, four questions on 3D modeling experience, and eight questions concerning 3D modeling software

Name	Description
3DOM 3D optical Metrology	A research unit focused on 3D reconstruction and metrology.
AD&D	Spain-based association for the applying cutting-edge technology to preserve CH.
Archaeovision	Private group offering 3D scanning and web sharing for CH.
CAA International	International conference on Computer Applications & Quantitiative Methods in Archaeology.
Code4Lib	Volunteer-driven collective of those who are interested in new technological applicaations in libraries, museums, archives, and more.
Consiglio Nazionale delle Ricerche	Multidisciplinary and interdisciplinary institute for research on CH.
Cyark	Non-profit with the goal of digitizing the wolrds most important CH landmarks.
Digital Archaeology Lab	UCLA lab for education and research in digital archaeology.
Digital Heritage	Conference on the application of digital technologies and CH.
IDIA Lab	Ball State University lab exploring the intersection between art, science, and technology.
IIIF 3D	Community group to facilitate the development and implementation of standards for 3D.
ImageMuse	Discussion group for CH imaging and publishing.
Kunst Histrisches Museum Wien	Austrian museum with modest but high quality 3D model collection on Sketchfab.
Museum-L	Listserv for museum professionals.
Santa Cruz Museum of Art & History	Mid-sized nonprofit institution that uploaded the results of a one-time digitization project to Sketchfab.
Sketchfab Forum: Cultural Heritage	Online Sketchfab forum for CH users of the 3D model sharing site.
The British Museum	Public museum located in London with an extensive online collection, including 3D models on Sketchfab.
The CH Engineering Initiative	UCSD institute for engineering applications to CH.
The Discovery Programme	Group dedicated to the preserervation of Irish CH through the use of modern tools, including 3D scanning and photogrammetry.
The Mel Fisher Maritime Museum	Not-for-profit maritime museum that performs frequent underwater photogrammetry.

Table 5.1: A selection of eligible groups and communities contacted for participation in the survey in interviews. Names are hyperlinked to relevant websites.

and their use. The final question asks for contact information, should the respondent be willing to discuss their survey responses in more detail in the form of an interview.

Ultimately, the survey served as a diagnostic for identifying the areas of scholar's practice most in need of addressal, while documenting the benefits and successes brought to VH by 3D models and their attendant technologies. Furthermore, the survey allowed for a broader enumeration of the demographics and characteristics of the target population, i.e. VH scholars. Nonetheless, in general surveys suffer from rigidity and vagueness that make them a poor tool for in-depth information gathering. There were two additional issues for the case of this specific survey. First, in spreading the survey broadly among online communities of which 3D CH users comprised only a subsection and with the use of snowball sampling, it was not guaranteed that survey respondents would fit the subject selection criteria. However, the expertise and fields represented in the survey results do seem to indicate that the majority of subjects were eligible as previously defined. Second, through the course of the survey process, an academic bias of the survey itself was uncovered. Fortunately, this was not a fatal flaw and actually served as a spark for discussion, as is discussed. Both of these issues are discussed in more detail in Section 5.3. Regardless, the surveys were ultimately successful in their purpose as an exploratory tool.

5.2.2 Interview

The interviews provided more detailed data on VH scholar's uses of 3D models. Semi-structured interviews lasting between 30 to 90 minutes were conducted either in person, via video conferencing, or via phone. In all cases, the audio was recorded for the purposes of transcription. An outline of an interview and a list of questions may be seen in Appendix D. The thrust of the interview questions was essentially the same as the survey, focusing on the practices and technologies surrounding scholarly 3D model use. However, the interviews served as a more open-ended tool to explore in detail scholar's practices. It was meant to facilitate a space to discover unanticipated aspects of 3D model use. For this reason, semi-structured interviews were chosen over structured interviews, as they allow for a more organic conversation to develop.

11 individuals were interviewed, reached initially through contact information from the surveys. These individuals were numbered arbitrarily for reference throughout this work. A basic description of those interviewed is shown below:

I1: University professor in archaeology
I2: University professor archaeology and VH lab head
I3: University museum curator
I4: VH professional and educator
I5: Maritime archaeologist
I6: Web3D professional
I7: Manager and developer of online 3D repository
I8: VH professional
I9: VH graduate student
I10: Library research support staff
I11: Library research support staff

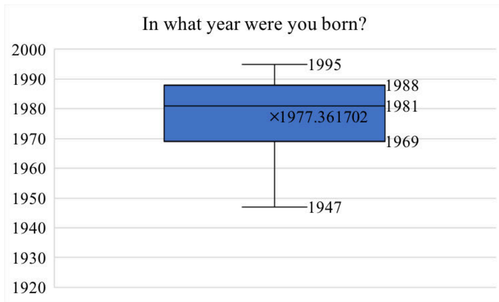
5.3 Delineating VH Scholars

VH is an emerging field, and thus not as clearly defined as more mature fields such as chemistry or sociology. This fact was made exceedingly clear through the course of the surveys and interviews. To view the survey responses discussed in the following section in full, contact the author. For purposes of anonymity, the full interviews may not be released.

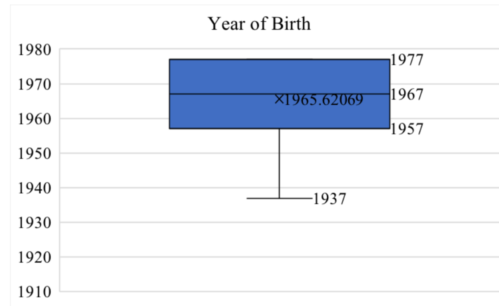
5.3.1 Demographics

Survey respondents may be described according to their age and gender. Based off of those surveyed, the average respondent is a male between 29 to 53 years old, who began producing 3D digital models of CH within the last five years.

49 of the 109 respondents provided a birth year. Survey respondents ranged in age from 24 to 72 years with a mean of approximately 42 years of age and a standard deviation of approximately 12 years. However, as can be seen in Figure 5.1a, the data is skewed towards older ages and 50% of respondents fall within a much more narrow range of 31 to 50 years of age and 68% in the range 29 to 53. The 2010 survey of Koller et al. saw a similar distribution, with 50% of respondents between 33 to 53 years of age and a skew towards older ages, see Figure 5.1b. Given the relatively small



(a) A box plot showing the distributions of the age of survey respondents.



(b) A box plot showing the distributions of the age of survey respondents from [78].

Figure 5.1: Box plots showing age distributions. Koller et al. reported ages as ranges. Thus, for the purpose of visualization, each range was represented by the median value.

sample sizes of our more recent survey and the survey by Koller et al., it is difficult to make any concrete statements. However, it does seem that VH has retained its dynamism throughout the last decade, drawing in fresh young scholars as well as old.

The majority of respondents, 77.6%, identified as male, 20.4% identified as female, and 2% declined to answer. This distribution is approximately equal to that observed by Koller et al., whose 31 respondents were split 80.6% to 19.4% between male and female. Whether these distributions are representative of the field of VH as a whole is debatable. Let us consider the primary fields through which scholars may become involved with VH, the arts and humanities and computer science. While women accounted for a slim majority of Doctorate and Masters degrees in the arts and humanities earned in 2016, 54.0% and 57.4% respectively, mathematics and computer science were far more skewed with 74.2% of Doctorate 67.3% of Masters degrees conferred upon men [242]. Thus, the gender disparity could indicate a prevalence of computer scientists among the sample, although this seems doubtful given that 4.85% of respondents identified themselves as working in Computer Science, see Table 5.2. On the other hand, the gender disparity may also be attributed to a trend observed in archaeology wherein publications are largely biased towards men while the degrees awarded are largely equivalent across gender [243].

Scholars were asked to list the fields in which they work, allowing for the selection of more than one response. The results of the question are shown in Table 5.2. All of the fields listed on the survey were well represented, with “Archaeology” the maximum by far, selected by 53.85% of respondents. However, 33.33% chose to also enter a user defined field with “Other”. “Other” was selected more often than over half of the possible fields. Arguably, some of the other responses may be included with existing selections, for example “Cultural Heritage Preservation” and “preservation”, could conceivably be grouped with “Conservation”. Furthermore, one could argue that the various museum related positions may fall under Museum Studies. History is the most frequently listed among Other, with five total mentions. Regardless, the frequent selection of other and the distribution across the provided 10 fields is yet another indicator of the variety contained in VH. Additionally, several responses have minimal relations to academic fields, i.e. “Theater Scenic Design”, “3D Modeling”, “Documentaries”, “Game Development”, and “Film”. This is representative of the expansion of VH into non-academic areas and highlights a methodological issue within the survey, a topic which will be discussed in more detail in Section 5.3.2.

5.3.2 Position

The variety of VH as a field was clearly evident in regards to the positions held by those surveyed. The positions listed on the survey were adapted from Koller et al.’s earlier 2010 survey [78]. The results of Koller et al. are reproduced in Table 5.3. These positions were a reflection of the arenas which served to define VH as a field at that time, namely academic events like conferences and publications. Since then, VH, and the intermarriage of technology and the humanities which it represents, has expanded beyond a purely academic interest. Digital technologies applied to CH have become a fixture among CH institutions world-wide, such as museums, libraries, and historic sites.

The formulation of the survey was such that it did not initially encompass the full variety of positions held by scholars and professionals working with 3D digital models of CH. This was first identified in a concerned email from one of the survey respondents. The respondent found the listed positions to be limited, as they had too prevalent a focus on academic positions.

> “I am giving you my feedback because you may not realize how your survey appears to someone outside of academia.”

Answer	Count	%
Archaeology	42	53.85%
Digital Humanities	31	39.74%
Museum Studies	27	34.62%
Art and Architectural History	26	33.33%
Other	26	33.33%
Conservation	21	26.92%
Anthropology	13	16.67%
Architectural Design	13	16.67%
Computer Science	11	14.10%
Engineering	11	14.10%
Paleontology	6	7.69%
Total	227	

Other
3D Modeling
Cartography
Collection management in a museum
CH Preservation
Curator of Collections at a museum
Digital Heritage
Documentaries
Film
Game Development
Geography
History
History
History
History
informatics
Information Technology
Libraries - Special Collections
Library and Information Science
Library
museum administration
Interdisciplinary studies
preservation
Public History
Space and Earth Sciences
Theater Scenic Design
Zoology

Table 5.2: Percentage of the 78 respondents that selected each field and a list of the “Other” fields entered. Responses are reproduced exactly as entered.

Answer	Count	%
Technician	1	2.22%
Cultural ministry	2	4.44%
Indep. Scholar	4	8.89%
Assist. Prof.	11	24.44%
Assoc. Prof.	5	11.11%
Full Prof.	11	24.44%
Other	11	24.44%
Total	45	

Table 5.3: A reproduction of results from the survey of Koller et al. [78].

Answer	Count	%
Technician	5	6.49%
Cultural minister	0	0.00%
Undergraduate student	0	0.00%
Graduate student	13	16.88%
Independent scholar	6	7.79%
Assistant professor	5	6.49%
Associate professor	4	5.19%
Full professor	5	6.49%
Other	39	50.65%
Total	77	

Table 5.4: Current positions reported by survey respondents.

> "I have worked in the imaging and heritage field for more than 16 years, and I just don't feel that your survey is setup to get good results outside of academia, and academia is only one part of the imaging work that is going on."

This was borne out by the survey respondents more generally. 39 of the 77 respondents, or a little over 50%, selected "Other" for their position and chose to supply their own title. The results of the survey are shown in Table 5.4. This clearly indicates that the supplied options were insufficient for the majority of survey respondents. The original survey of Koller et al. saw a similar, although far less extreme, prevalence of "Other" selections, with "Other" selected by 24.44% of respondents, tied for the most of any category.

What positions then might more fully capture the roles held by survey respondents? As can be see in the Table 5.5, responses for "Other" positions can be grouped into four areas of practice: Industry, Museum, Library, and University. All of the supplied positions, excepting Technician and Cultural minister, would fall into University. Yet even these five options were woefully inadequate to describe this single area, as 11 of the respondents entered unlisted academic positions. "Cultural minister" may be said to belong to an additional area of "Government" not represented in our sample. "Technician" represents a gross simplification of the varied positions that may be held throughout any of the areas of practice since technical staff may be employed throughout the listed areas.

Asking respondents for their position was problematic for the generation of any proper insight,

Industry
3D Modeler
Architect
CEO of Design Firm/Lecturer
CH Professional
Digital Imaging Coordinator
Digital Production Coordinator
Digital Team Lead - Systems Administrator
Director of Imaging
Imaging Studio Manager
Photographer
Photographer
Project Lead
Software Engineer

Museum
Curator of Collections
Director of Conservation
Director of Technology and Digital Initiatives at Museum
Museum Curator
Museum Director
Museum director
Museum Photographer
museum registrar
Museum Researcher

Library
Librarian
Librarian

University
Adjunct faculty
Director (Professional Staff at a university)
Lab Head
Lecturer (UK)
Lecturer/ Associate Researcher
PhD student
Postdoc
Research Assistant
Research Manager
Senior Researcher
Staff supporting research

Table 5.5: Positions supplied by respondents for "Other". Some responses have been altered to maintain anonymity.

and only presented a muddled view of the different areas in which VH practitioners work. While the position of individuals working within VH is certainly interesting information, the variety and particularities of job titles make a clean partitioning difficult. Certain academic distinctions are certainly clear, such as students and professorial appointments. But this only covers a portion of those individuals involved in the production and use of 3D models of CH. Possibly, a more informative question would ask individuals about the sector and groups within which they work, while also offering an open response for position titles titles.

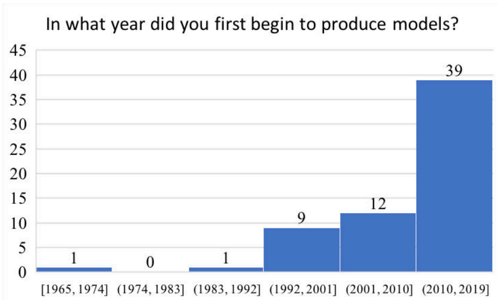
5.3.3 Technology Adoption

Survey respondents were questioned as to their experience producing, using, and sharing 3D digital models, both in regards to volume and individual technologies.

There was a wide range in the year in which respondents began producing models, ranging between 1965 to 2018 with a mean of 2009 and a median 2013. However, as can be seen from Figure 5.2, the data is heavily skewed towards the lower end, with the overwhelming majority of respondents, 75%, starting to produce models in the last 12 years, and 50% in the last 5 years. Clearly, the production of 3D digital models of CH is not a new activity. Yet it appears, at least among the survey respondents, that there has been a recent influx of new 3D digital model producers. To those familiar with the field, this would not be surprising, as 3D digitization projects have recently seen significant increases in prevalence, ease of execution, and visibility.

Possibly, the recent increase in model producers could be a result of recent improvements to the photogrammetric workflow, which have lowered the barrier to entry for those without significant technical training. *I4* observed such a trend in their role as an educator in scanning techniques for the CH community.

> "We within our team had been applying photogrammetry since 2001/2002 but we didn't feel like the technique and software was really ready for this audience at that time...The digital photogrammety in the earl days was a real pain in the butt...the software was either extremely difficult to use or very expensive. It wasn't until probably 6 or 7 years ago that we felt like, hey things have shifted enough that we can start thinking about doing a training and we actually did the first class in 2014."



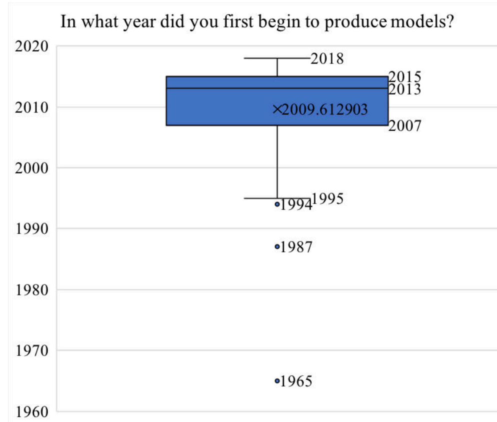


Figure 5.2: A histogram (left) and box plot (right) showing the distributions of the year survey respondents began producing 3D digital models.

These dates correspond well with the first major releases of what have become prevalent photogrammetric software used in the VH community, Agisoft's Photoscan version 1.0 in 2014 [223] and Capturing Reality's RealityCapture in 2015 [244].

While new 3D scanning technologies have likely encouraged new practitioners to enter the field, it is unclear whether these technologies have made the average 3D model producer any more efficient. There appears to be no observable relationship between the year in which scholars began producing models and their rate of model production (see Figure 5.3) nor between the starting year and the proportion of total models respondents have produced in the last 12 months (see Figure 5.4). Therefore, among those surveyed, longer experience producing 3D digital models did not indicate increased model production, nor did an introduction to 3D model production with recent technologies. Of course, our current discussion of 3D digital models completely ignores issues of quality. Whether experience producing 3D digital models or the use of new technologies result in more visually appealing or metrically accurate models remains to be seen.

Regardless, those survey represent a significant volume of 3D models produced. The number of models produced over a respondents entire career ranged wildly, with a minimum of 0, a maximum of 10,000, and a mean of 452. However, 91.8% of the reported values are less than 1000 and 75.3% fall between 0 and 400. In total, the 73 survey respondents who reported their total models produced 32,964 models. This number is certainly skewed by the 10,000 model producer, but even 20,000

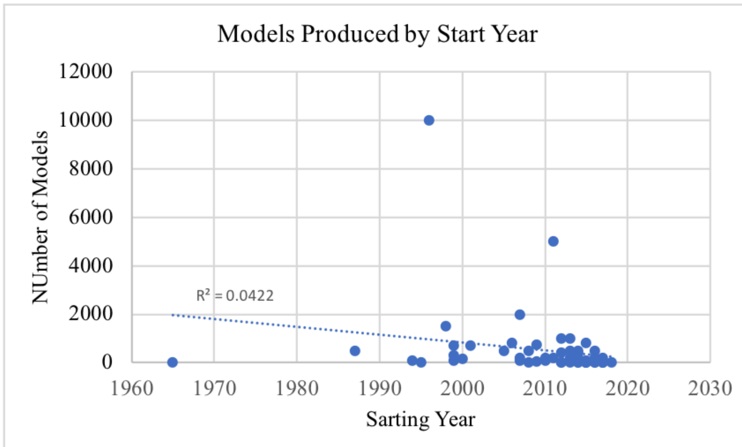


Figure 5.3: A comparison and regression showing the lack of correlation between the year at which a respondent began producing models and the total models they have produced throughout their career.

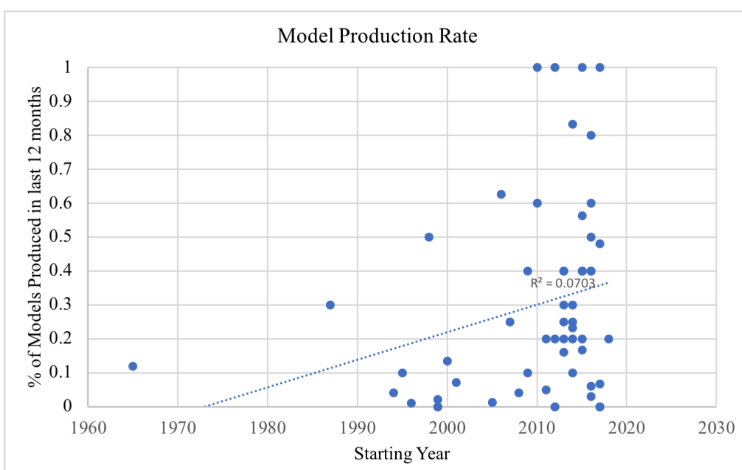


Figure 5.4: A comparison and regression showing the lack of correlation between the year at which a respondent began producing models and the total models they have produced throughout their career.

models is striking considering the number of models available online. Europeana and Sketchfab represent some of the largest online collections of 3D digital models of CH objects. Europeana lists its total 3D collection at 11,052 models [245]. Thomas Flynn, the Cultural Heritage Lead at Sketchfab, reports that there are over 70,000 models uploaded to their service labeled as cultural heritage, spread among over 500 CH institutions and an unknown, but likely equally large, number of individual contributors [117]. If we extrapolate the number of models produced in our sample of 73 to just the 500 CH institutions on Sketchfab, we have 10,000,000 models. This is likely a high number, one skewed by exceptionally productive individuals among our sample and the conflation between models made and those fit to share more broadly.

Whatever the number may be, this discrepancy implies that there remains a significant number of 3D models inaccessible or even undocumented among the wider VH community. The exact number and nature of these missing models and the reasons for their inaccessibility certainly deserve attention.

Respondents were asked to rank their abilities to perform various tasks related to 3D model production, visualization according to following scale:

Learner: I am not sure how to do this task.
Basic: I have done this before, but I may need help.
Proficient: I can perform this task without assistance.
Advanced: I could train others to do this.

The results are shown in Figure 5.5. The most common "Learner" categories were "Create a model by hand (eg in Blender)" with 31, "Create a model using laser or structured light scanning" with 23, and "Create a model using photogrammetry" a distant third with 18. Overall, the majority of selections were in the "Advanced" category, reflecting the status of those surveyed as experts in the field. Only two categories did not receive their highest respective values in "Advanced", "Create a model by hand" and "Create a model using laser or structured light scanning". By far the most picked selection was "Share a 3D model with others" at 45 with "Display a 3D model on the web" second most at 36. This coupled with the previous discussion of the apparent lack of sharing, at least online, indicates that it is not a lack of technical experience limiting the availability online. Finally, in line with the experience of *I4*, photogrammetry appears the technique of 3D model production with which survey respondents were most comfortable.

5.3.4 Software

When asked to list the software they used to work with models, respondents listed a large variety of software, both across the entire sample and within individual responses. All of the software listed and their relative mentions are shown in Table 5.6. 41 respondents listed 45 unique software with a total of 184 distinct mentions. On average, each respondent listed approximately five unique software, see Figure 5.6. Given the diversity of tasks involved in the production and use of 3D models, such a number is not altogether surprising.

Respondents listed software ranging throughout the lifecycle of 3D model production and use. The listed software may be roughly grouped into five categories: File Management, Image Processing, Scanning, Modeling, and Visualization. Modeling is a heterogeneous grouping which includes software for both the creation and editing of 3D geometry and also analysis tools, as such features often go hand-in-hand. However, these categories are only approximate, as software may range in use between the latter three. Meshlab for instance is capable of processing scan data, editing 3D geometry, and is often used as a flexible tool for the visualization of high-resolution geometry on the desktop.

File Management and Image Processing are the two most underrepresented categories. File Management can be crucial in maintaining a rigorously documented workfow and ensuring the highest standard of metadata as to the modeling process (for the importance of such documentation see Section 3.4). Few tools exist to facilitate such work. The Digital Lab Notebook released by CHI is a notable exception, an open source pipeline to maintain richly documented workflow information [246]. While metadata documentation is an issue with which VH is currently grappling, a more complete treatment falls outside the scope of this work.

Image Processing is used in the case of photogrammetry. Not all photogrammetry workflows make use of image processing, as certain image processing techniques such as sharpening may

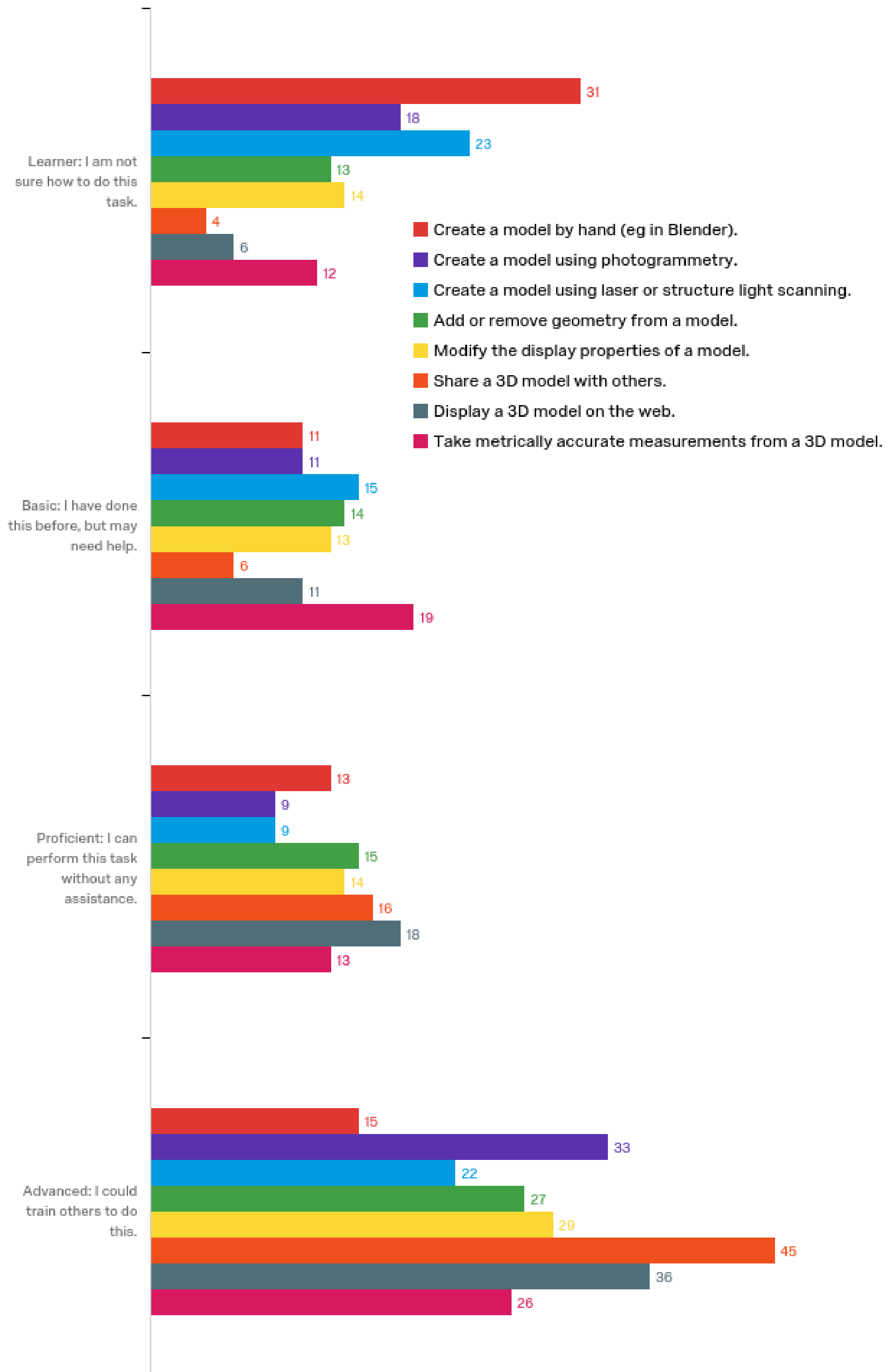


Figure 5.5: Visualization of *How would you characterize your abilities to perform the following tasks?*

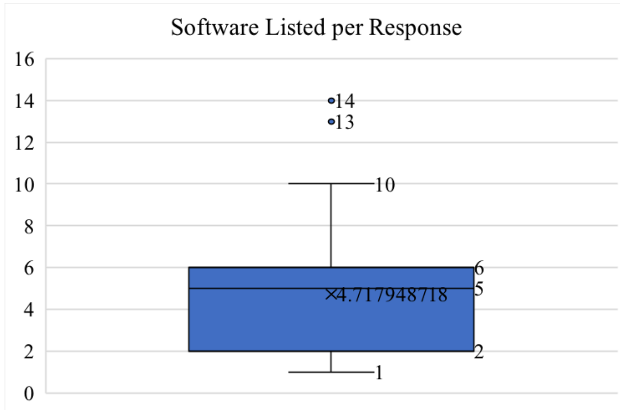


Figure 5.6: Distribution of software listed per respondent.

result in erroneous matches between images. Regardless, Photoshop appears the de facto software for such operations.

Modeling is certainly a broad category, but even so the variety of software evident is telling. In part, this variety may be attributed to the differing roles of the software listed. Some are used for decimation and retopology algorithms, like Meshlab and ZBrush; others for creating to-scale models for printing and study purposes, like 3ds Max and Autocad; others for computational comparison and analysis, CloudCompare and MorphoJ; and still others for the procedural generation of 3D geometry, like CityEngine. The breadth of Modeling tools is also representative of the breadth of research actions and the diversity of practitioners in VH. Some of the software require expensive monthly or annual licenses, and cannot be afforded by smaller institutions or less well funded departments, thus the evident intermixing of free and proprietary software.

Given the wide variety of tasks and applications contained in the use of 3D digital models, it is questionable whether such diversity of software tools is in any way avoidable. But this variety is also present individually among scholars. The variety of software used by individual scholars is most clearly shown in the extreme case of a respondent who listed 14 different software:

> "Capture - Autodesk Recap Photo, Artec Studio, Faro Scene, Trimble Realworks, Agisoft Photoscan, Autodesk Recap
> Meshing/Modelling - Geomagic Studio, Autodesk Mudbox, Autodesk REVIT, Autodesk 3DS Max, Autodesk Inventor, Substance Painter,
> Desktop Vis:Meshlab, Unity, Unreal
> Online Vis: Sketchfab"

Software	Mentions	Software	Mentions
Meshlab	25	Box.com	1
Photoscan	24	Instant Meshes	1
Sketchfab	23	Autocad	1
3ds Max	10	Sketchup	1
Blender	10	CityEngine	1
Unity	7	AGMT3-D	1
Rhinoceros 3D	7	MorphoJ	1
CloudCompare	6	MS Office 365	1
ZBrush	5	ArcScene	1
ReCap	5	Digital Lab Notebook	1
Meshmixer	5	Ashlar-Vellum Argon	1
Artec Studio	5		
RealityCapture	5		
Maya	4		
Revit	4		
ContextCapture	3		
Faro Scene	3		
Photoshop	2		
ArcGIS	2		
Geomagic Wrap	2		
Bentley	2		
Unreal	2		
Scan Studio	2		
3DHOP	2		
NextEngine 3D	2		
Youtube	2		
Adobe Bridge	1		
Polyworks	1		
Adobe Dimension	1		
Geomagic Pro	1		
Trimble RealWorks	1		
Autodesk Mudbox	1		
Autodesk Inventor	1		
Substance Painter	1		

Table 5.6: The software mentioned by survey correspondents and their number of mentions.

File Management	**Image Processing**	**Scanning**	**Modeling**	**Analysis**	**Visualization**
Adobe Dimension	Photoshop	Photoscan	Meshlab	CloudCompare	Sketchfab
Digital Lab Notebook		Autodesk ReCap	3ds Max	Trimble Realworks	3DHOP
Adobe Bridge		Artec Studio	Blender	MorphoJ	Youtube
		RealityCapture	Rhinocers 3D	AGMT3-D	Box.com
		ContextCapture	ZBrush		MS Office 365
		Faro Scene	Meshmixer		Unity
		Scan Studio	Maya		Unreal
		NextEngine	Revit		
		ArcScene	Geomagic Wrap		
			Bentley		
			Polyworks		
			Geomagic Pro		
			Autocad		
			Sketchup		
			Autodesk Mudbox		
			Autodesk Inventor		
			Substance Painter		
			Instans Meshes		
			CityEngine		
			ArcGIS		
			Ashlar-Vellum Argon		

Table 5.7: A rough classification of the software listed by survey respondents.

This response also demonstrates the relative distribution between the software categories observable more generally across respondents, Scanning and Modeling include the majority of the software, with Visualization a third.

Importantly, only four of the seven software listed are able to visualize interactive 3D content: Sketchfab, 3DHOP, Unity, and Unreal. Unreal, like Unity, is a game engine and accompanying development environment [247]. Unlike Unity, Unreal does not allow for deployment online, although it does support numerous devices ranging from AR to mobile to desktop. The former three were previously discussed in Chapter 3, and while each have distinct benefits they do not fully address the needs of VH scholars. The use of additional tools like Youtube, Box.com, and MS Office 365 demonstrate the unmet needs of VH scholars for the sharing and visualization. The importance, and existing impediments, to visualization and the sharing it facilitates were a prevalent theme throughout the responses.

5.3.5 Sharing and Visualization

Respondents were asked about the frequency of 10 actions involving 3D digital models selecting between six levels of frequency, see Figure 5.7 for the complete list. The top four most chosen responses were, in order, “View a model in a web browser” “At least once per week”, “View a model offline (eg in Meshlab)” “At least once per week”, “Share models with other scholars” “At least once per month”, and “Share models with the public” “At least once per month”. Considered cumulatively, the distinction of sharing and visualization remain. Over 70% of respondents share and view models at least once per quarter or more frequently, with almost 90% viewing a model in a web browser at least once per quarter or more frequently. Undoubtedly, sharing and visualization go hand-in-hand and are fundamental to any work with 3D digital models. Thus, one would expect their frequencies to be higher than other actions. Furthermore, The frequency of viewing models on the web demonstrates the maturity of Web3D technologies and their adoption within VH. An alternative explanation for the disparity between sharing and viewing and other actions could be the type of individuals represented in the sample. Possibly, survey respondents could be more properly characterized as model consumers rather than model producers. However, the high number of models produced among the respondents seems to belie this claim (see Section 5.3.3).

As a follow up, respondents were asked to elaborate on which actions were the most useful and how. The sharing and online display of models were by far the most frequently mentioned with 25 of the 40 responses doing so. Other mentions were the importance of editing models so as to facilitate visualization and reuse, mentioned four times, and measurement, mentioned three times. One action not listed in the survey but mentioned four times by scholars was the importance of the creation and enforcement of community wide standards for model production and use.

In discussing the sharing and visualization of models, respondents commented on the potential issues in sharing high-resolution 3D models via the web.

> "Viewing models in a web browser is useful for sharing models with others, but viewing offline allows a higher resolution mesh for applications that require it."
> "I have found that sharing models with others, whether the general public, students, or colleagues is best when I make videos of my models. It works well in lectures, conference papers, and social media. Providing manipulable models such as those on sketchfab are also useful, but I must assume that the user will know how to explore and inspect the model. I have had people lose interest in viewing my models because their web browser did not load models properly. Videos just seems to be the safest way to ensure that people will view the models."

Alternatives for sharing 3D models were present among the software listed.

Box.com, MS Office 365, and Sketchfab (in certain cases) are solutions that may facilitate the direct sharing of 3D model files for download. However, among the survey respondents there was a distinct gap between the frequency of sharing and the frequency of download. Besides "Include a model in a publication", "Download a model from the internet" was the most infrequently performed action with 50% of respondents doing so "At least once per semester" or less frequently. The frequency of model sharing has already been remarked upon. This implies one of two scenarios: 1. scholars are sharing models physically, i.e. using physical memory devices such as USB drives of external hard drives or 2. Scholars are sharing visualizations of models without the ability to download. Two factors of particular interest to our purposes here, and which play a role in both the aforementioned cases, were repeatedly mentioned among survey responses: the importance (and difficulty) of sharing high-resolution models and concerns about security. These issues will be

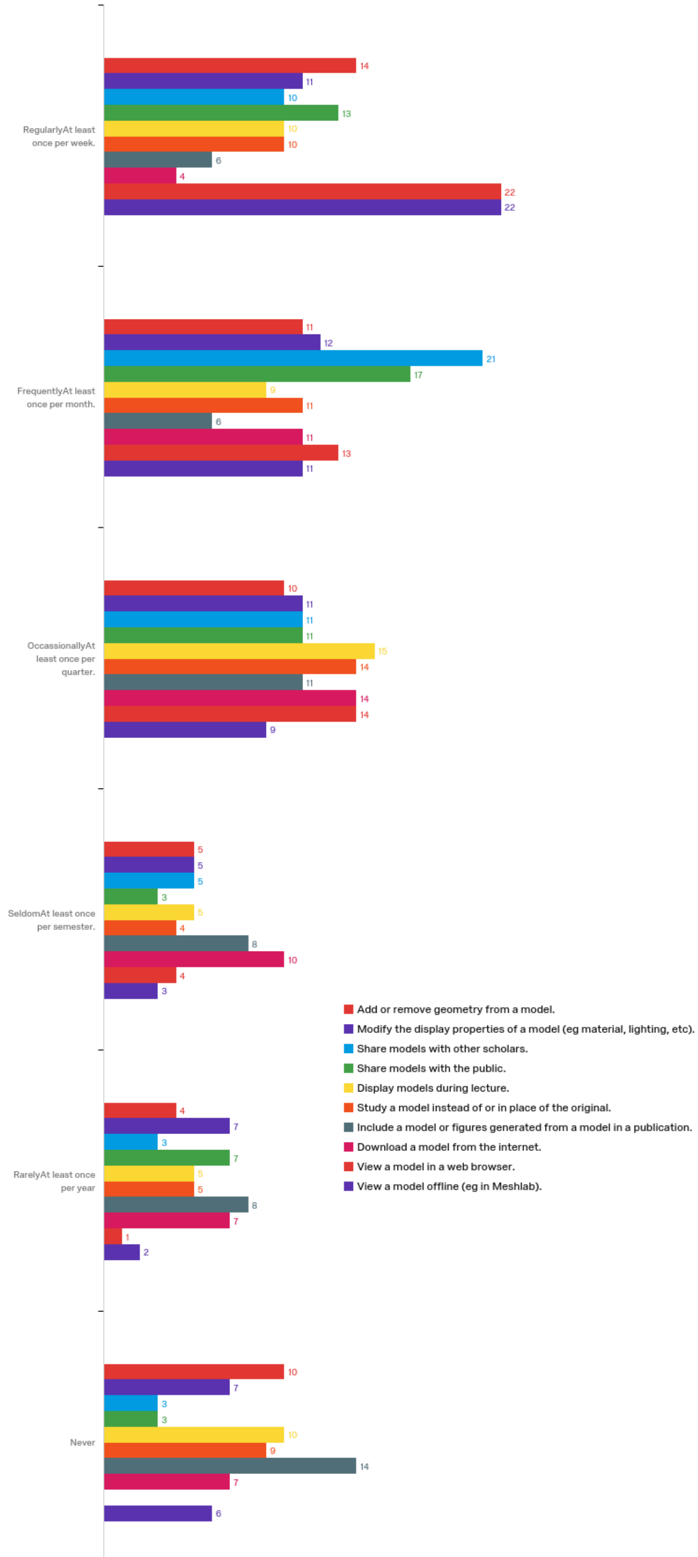


Figure 5.7: Visualization of *How often do you perform the following actions in your research or instruction?*. Note that the legend lists the color correspondences in the same order as the figure.

discussed in Section 5.5.1 and Section 5.4 respectively.

5.4 Security Concerns

A wide range of views on the topic of security were uncovered through the course of the surveys and interviews.

I1 described a similar stance by the Greek government towards the wider sharing of digital assets as the traditional view expressed in Koller et al., that culturally valuable objects should be protected so as to discourage unauthorized reproduction in a physical form through milling. In the experience of *I1*, even the use of photographic reproductions involved lengthy legal processes. Despite receiving an initial permit to photograph an object, *I1* was required to go through an additional 10 month permitting process before publishing the work, since the original permit did not include the word "publish". Based on this experience, *I1* expects a similarly fraught permissions process for 3D reproductions. Per the analysis of *I1*, governments and large institutions are slow to adapt to changing technologies, Greece representing a particularly conservative case.

> "This leads into 3D models and how people publish these because of course archaeologists are using these and I think its an issue that hasn't been entirely resolved. The technology is moving faster than...the various [government organizations] that are in charge of protecting the antiquities can write the laws or implement them...It's this gray area, in the meantime, I've been told, they just say no to things...and I think primarily it's because they don't want information out there that they can't necessarily control, i.e. the 3D model and all its data that can be downloaded by anybody and 3D printed."

Thus, given the slow maturation of 3D technologies, it is not altogether surprising that the views of such organizations have changed little in the intervening time between the publication of Koller et al. and now.

Similar views were expressed by museum professions in the survey when asked what features were most important in visualization tools.

> "high-resolution and security. As a lot of museum are hesitant to have their models displayed on a platform that is readily available to the public, they tend to not give permission (or it is hard/impossible to obtain permission) to publish 3D models"
> "Since I work in a museum (and an academic institution), security and the attachment of proper information and credit to the models is very important, but so is accessibility."

A desire for security, albeit in a more modest form, was also expressed by those in charge of virtual and physical collections. *I3*, who serves as a curator for a university museum, discussed the dissemination of digital models in relation to the efforts of the museum. According to *I3*, the museum is a steward of the information, and part of that responsibility is in the respectful and properly attributed sharing of derivatives from the objects the museum holds.

> "With so many platforms I think it important from a museum point of view, to have it on a single platform that we can access and control...In the past...museums had in-house photographers with complete control...we share far and wide but we can control all the information so that wherever these are going out we know where they're going...they can do what they want with it and we can know what they're doing...we want to be able to do the same things with models...Just as a librarian controls the books...its part of our job to make sure these things are properly cataloged."

Thus, *I3* did not want security so as to discourage sharing but rather to ensure the proper attribution in sharing.

I2 also discussed the importance of control over dissemination. *I2* digitized a Greek theater and made the theater available on Sketchfab. This model was added to a collection of Roman theaters. Thus, someone searching Sketchfab for Greek theaters could be shown the Roman theater of *I2* and thereby be misinformed as to the nature of the 3D model. While this may seem a minor offense, to *I2* it represented a subversion of the good practices of scholarship which *I2* sought to embody in their work.

In contrast to *I2*, *I5* saw no harm in such contextual misinformation, as Sketchfab still provides linked descriptions of the objects that be used to clearly explain the classification, provenance, etc., of the object. However, *I5* did not go so far as to make their models available for download. For while *I5* recognized that once something has been shared on the internet, it takes on a life of its own, they was not willing to explicitly release his work for reproduction and modification.

On the opposite end of the spectrum were individuals who explicitly encourage the sharing and download of their work. Individuals such as *I6* claim that there has been no evidence for the malicious use of 3D digital models of CH content. And rather than exposing content to misuse and improper reproduction, open sharing encourages the continued use, reuse, remixing, and visibility of CH content. This has certainly been the approach of some larger institutions such as the British

Museum, who has made much of their content available for download on Sketchfab.

I7, in their work developing an online repository of 3D models, commented on the wide diversity of control requirements desired by the producers of 3D models. To accommodate this, their platform offers fine grained permissions controls.

> "We recognize is that digital data representing physical objects often times unavoidably carries sharing limitations that derive from the physical object itself so we base tools that allow different levels of access."

Such issues are especially complicated in the cases of modern, in copyright works

Given opinions to the contrary, one may question whether the reluctance of larger institutions to share 3D digital models more broadly is warranted. However, the worries of governments and larger institutions that these 3D models will be used for reproduction are not unfounded. Walking the streets of Rome one may see numerous neon colored miniature 3D prints of famous sculptures, Michelangelo's David especially. Similar miniatures are for sale on numerous websites. Close inspection of these statuettes reveals that they are not exact copies but rather artists imitations. Whether or not a more accurate reproduction of these works would have any effect on the already existing souvenir market as it is is unclear. That said, there certainly are valid, and currently undecided, copyright claims surrounding such digitized works, both in regards to its relation to the original object and as a new work itself.

At the very least there currently exists a pragmatic need for the secure and protected viewing of 3D digital models. The work of scholars like *I1* rest upon access to the objects they study and thus the continued goodwill of the governments and institutions in control of these objects. Methods for the protected sharing of 3D digital models could serve to further convince such groups to allow for broader sharing of the 3D digital models.

5.5 Literature Validation

The results of the surveys and interviews may be used to corroborate some key claims made in the literature as to the requirements for VH visualization tools and characterizations of the field. For a more complete discussion of the literature discussing the production, use, and dissemination of 3D digital models of CH, refer to Section 3.4.

5.5.1 High-resolution Models

The 3D digital models used by VH scholars have been described as, more often than not, high-resolution [100] [101] [78] [114] [77]. This results in large file sizes and potentially unwieldy 3D digital models, ones unable to be visualized in software without explicit means for handling such large files. While this certainly was an issue for numerous individuals in both the surveys and interviews, there were several notable opinions to the contrary.

Issues with handling large data sets was prevalent throughout the survey, in particular high-resolution models.

> "Visualization of high-resolution models is extremely important. Often, models have 100s of millions, and sometimes billions of points (for 3D point clouds)."
> "FILE SIZE. Sharing is great, but you need a lot of space to store files, and many people who want to see the scans either cannot figure out how to view them (unless uploaded to third party sites like sketchfab) or do not have the email storage to receive them."
> "The large model size can be difficult to be managed on certain software packages, so the visualization and any operation on the model risk to be painfully slow and in the end not sustainable in a real workflow."
> "the large file sizes make models difficult to share outside of an online gallery."

For the most part, the high-resolution models were described as a result of photogrammetric workflows for the digitization of real-world objects.

> "...models based on real objects are often quite large (or unedited, and therefore large), so support for high-resolution meshes is quite useful."
> "We do a lot of photogrammetry in our lab. Being able to handle very dense data and build high-resolution models is important."
> "Photogrammetry in general. The biggest hurdle right now is preservation of large sets of data."

The difficulty of working with the high-resolution photogrammetric models is well remarked upon in the literature. Our earlier discussion of Vranich [11] in Section 2.1.1 represents an illustrative case. Without sophisticated software and sufficient computing power, such models can be impractical.

But photogrammetric models pose additional data issues besides the size of the 3D geometry itself. The photographic data sets, used to generate the models, themselves represent a work of scholarly importance. Not only are the data sets useful in assessing the quality of the produced model through reference to the underlying data, but they may also serve as the basis for additional reconstructions given future improvements in software and hardware.

> "Photogrammetry in general. The biggest hurdle right now is preservation of large sets of data."
> "Processing power. We are used to capturing extremely high-resolution still photographs and most of our 3D use cases still expect sub-millimeter precision on large (3m wide) works... that is a lot of data for most systems."

Additionally, the high-resolution textures accompanying photogrammetric models may also prove a constraining factor.

> "The optimization of large 3D scanned scenes for real-time use. Large environmental scans often have a very high-resolution (polygons) and require many textures to maintain visual fidelity when run in a real-time engine or VR. The optimization of the mesh and UV-spaces often requires a substantial amount of manual work, or a very powerful computer to "brute force" display the un-optimized model."
> "I shy away from making small improvements on large datasets with millions of polys and/or multiple high res texture files because the interface and editing tools get really bogged down."

Ultimately, to effectively disseminate models more broadly, scholars are required to heavily reduce, i.e. decimate, the results of their work.

> "Being able to present models that are not heavily decimated to the global community. Current viewers, like Sketchfab, prevent high density models from being put online. There are no viewers that easily allow for the dissemination of big data to the larger public."

However, as we will see shortly, the requirement for decimation is not always seen as a hindrance.

I2 felt acutely the negative impact of the requirement for decimation in sharing both online and offline. *I2* has produced numerous 3D scans of large interior spaces and objects, which result in millions, if not billions, of points and gigabytes of data. Even when sharing such scans with partners physically, *I2* was inhibited by the devices of their partners.

> "Not always our partners have computers as powerful as we have...so you open a point cloud on your computer and it looks great and you assume that when you give to your partner the point cloud they will be open it instead their laptop will explode. And this causes also issues because you cannot truly show the deliverables to your partners"

In practice, *I2* has found no effective means of circumventing the issue of disseminating high-resolution models. *I2* may bring partners to the offices of *I2*, where *I2* can use their own powerful machines for visualization. While this removes any doubts the partners may have about the overall quality of the work, this merely postpones the issue of dissemination. When sharing with partners, *I2* may provide the decimated, interactive version as well as the high-resolution version, with the hopes that in the future the high-resolution version may be useful.

When sharing models with the public, i.e. online, *I2* has found no realistic solutions for sharing high-resolution data, with 500 MB to be a practical limit on file size.

> "I think that half a gig is probably a reasonable size limit, a lot is lost especially when it comes to the point cloud because I have a lot of point cloud models...when I decimate it and put it on [online] it looks like crap. And I feel bad because people will think that I produce poor data when instead I have outstanding data."

Thus, rather than share interactive 3D digital models, *I2* has sought alternative methods to demonstrate the quality of their work.

> "I have realized that I need to be smarter like my colleagues...my colleagues they don't disseminate their point cloud data they disseminate animations. They create beautiful animations of these point cloud models and then they put these videos on their social media accounts."

I2 was aware of efforts to visualize high-resolution point clouds online, namely 3DHOP [114] and Potree [248]. And while they plans on exploring these options, the investment of time to learn the required technical information, or money to hire someone to implement them, have proven prohibitive. Thus, the issue at hand is not entirely a lack of available technologies, a point we shall return to in Section 5.5.2.

Not all scholars have encountered the same dissemination issues as *I2* and the various survey respondents. For example, *I5* with their models of maritime artifacts both underwater and in the museum, experienced only the most minor issues in uploaded their photogrammetric models to the web, which ranged in size from 100 thousand to 2 million polygons.

> "I haven't had any issue with them being too big, I just haven't...I've had a few that have been kind of big and are a little slow to load but they've its been generally very manageable."

Similarly, *I8* has found Sketchfab, and its limit of 500mb uploads, to be more than sufficient for the dissemination of high quality models. *I8* sees no recourse for the type of dissemination/scientific model dichotomy that requires the publicly shared and visualized models to be reduced. Rather, using techniques from gaming, *I8* produces a visually accurate reduction of the high-resolution model.

> "Because what we try and specify is that there is the original data, which is the high-quality scientific data, but you can never give that out over Sketchfab because it is just too dense. So we try and make people understand that Sketchfab is a child of that data and with Sketchfab we kind of follow two criteria one is that the objects should look as identical to the high resolution data as possible while using methods and techniques in gaming that allow us to reduce the file size."

Using techniques such as 4k textures, algorithms for retopology, and high-resolution normal maps, *I8* is able to publicly display models that meet their high standard of visual quality.

There is no question that the dissemination and visualization of large 3D digital models on the web presents a technical hurdle. Furthermore, such models are routinely produced in the process of digitization projects in VH. But that is not to say that there are no means of sharing such models nor if, as argued by *I8*, visualizing such high-resolution models more broadly (i.e. on the web) is a requirement for the effective work of VH scholars. Rather, additional training and time are required in order to learn and apply appropriate methods to the 3D digital models in order to either prepare them for specialized, big data visualization tools (3DHOP and Potree) or reduce them in a way that preserves the visual quality. However, the prevalence of concerns over the sharing of high-resolution 3D digital models evident in the survey and interviews indicate that, whether it be for lack of training on the part of scholars, lack of visibility of the relevant tools, or some unforseen technical or social factors, as a whole VH scholars continue to grapple with this issue.

5.5.2 Accessibility

Accessibility has been identified as a key tenant of VH digitization initiatives [77] [103]. As products of scholarly efforts contributing to the sum of human knowledge, and an effectively nondepletable resource, authors have argued that 3D digital models should be shared as widely as possible. While survey and interview participants did not comment on accessibility as central goal of the field, issues preventing the wider access to and use of 3D models were frequently discussed. These issues effect both the model consumers, those seeking to access the work of others, as well as the model producers, those seeking to make their work more widely available.

> "I think most 3D repositories are poorly accessible. In my experience they are hidden deep in library websites or hard drives, and despite all the scholarship coming out about 3D repositories

> of ceramics there are not many that are accessible or displayed in a way that could facilitate research."

As discussed previously, the large file sizes of high-resolution 3D models can prove a hurdle for both model producers and consumers, both in terms of software and hardware. Hardware is an especially hard constraint.

> "Funding for better CPU processors and GPU video cards slows me down but my University Library does a pretty good job of trying to keep up. CPU and GPU are the main limiting factors in processing the data loads I acquire."
> "Limitations of hardware to display point clouds for robust exploration by visitors."

Largely, the factor of hardware may be reduced to an issue of resources. As shown in Section 5.3, scholars and individuals working with these 3D digital models are often from archaeology departments or other groups without significant technology budgets, and thus without access to expensive, top-of-the-line hardware.

Software does not pose the same issues. While the production of 3D digital models often requires expensive software licenses for scanning and processing 3D geometry, some of the most powerful and effective visualization tools are free to use, such as 3DHOP, Potree, Meshlab, and Sketchfab. Rather the primary barrier is ease of use, i.e. the ability of VH scholars to set up and operate the software tools. *I2* described the skills of digital archaeologists as "super users":

> "[Visualizing models online] is not very hard but of course it depends on the skills we have to face the reality and admit that the average digital archaeologist is a super user. Super user is someone who is able to use a lot of hardware, a lot of software better than many others but who write HTML code or simply modify existing HTML goes beyond the skills of the super user."

This exact scenario was demonstrated by *I3*, who, as a scholar of the ancient world, felt only partially equipped to properly visualize and work with the models produced by their collaborators. *I3* described their experience creating and using 3D digital models thusly:

> "I sort of half know how to do it...what I'm still missing is the software side of it and its still sort of a magic black hole in my mind...What is still don't understand is where it is, and how you get to it, and how you use it...I can manipulate [on Sketchfab] it I can use it but I don't know how to get it off Sketchfab and physically use it."

I3 attributed their lack of understanding to a lack of time for pursuing the proper instruction, since there were numerous resources for training on such topics.

As previously discussed in Section 5.5.1 and Section 5.3.5, one may circumvent these access issues at the sacrifice of interactivity, through the use of derivatives such as rendered video and images, formats with more fully developed platforms for dissemination and more widely accepted standards of interaction.

The heterogeneity of hardware and software available to VH and the general public, and user's varying levels of competence in operation, clearly represent a significant hurdle to the wider accessibility of 3D digital models of CH. Projects such as 3DHOP, with its support for high-resolution models and design principle of "ease-of-use", address core issues for accessibility. The developers 3DHOP recognized the diversity of VH users, and thus sought to provide acceptable defaults and "out-of-the-box" functionality. However, even this effort may be too much of a technical commitment for some VH users, as discussed by *I1*. Likely, a one-size-fits all tool for VH is unrealizable. Instead, VH visualization tools should address a clearly defined subset of VH users in total, thus designing to ensure the highest levels of ease-of-use and accessibility within the target group.

5.5.3 3D Analysis

As discussed in Section 3.4.5, 3D digital models allow for the application of a host of analysis techniques, ranging from inter-mesh comparisons to structural load bearing simulations to morphological analysis. Maschner goes so far as to claim that the key to the success of online 3D repositories, at least in the case of zooarchaeological collections, rests upon their support for analysis tools, in particular, measurement [77, p. 2].

Within the surveys, most mentions of analysis tools fell within discussion of existing software and the benefits of 3D digital models. Both more advanced methods as well as straightforward measurement techniques were discussed.

> "The primary reason that our institution generates 3D models is for geometric morphometric analysis."
> "Being able to take measurements and analyze an artifact or site that are more accurate than in person. They make accessible to researchers and the public a site or artifact that might otherwise be inaccessible through dissemination."

> "Being able to take measurements in places where measuring would be difficult are useful (a work with hundreds of objects carefully arranged on the table and measuring by hand risks bumping the placement of the objects)."

Various offline tools exist to perform such operations, see Section 5.3.4 for a list of those mentioned. There are fewer online tools, but measurement is currently available in 3DHOP. Nonetheless, when asked about potential improvements in existing visualization solutions, three respondents requested analytic methods.

> "It would be nice if the functionality of Cloud Compare and that of Sketchfab could be combined."
> "Analytical features - ability to isolate features, to measure the divergence between like models in order to conduct deviation analysis etc."
> "What I'm interested in in a 3D viewer is its ability to display annotations that could be pertinent to research and the ability to take measurements in the viewer. An ability to measure surfaces or volume would be even better."

Thus, the use and benefit of analysis tools, measurement especially, are well represented among those surveyed. Outstanding issues appear to be in the simplification and combination of existing tools and the continued support, for measurement especially, in visualization tools.

5.6 Personas

Personas are an additional method for the generation of insight from information about users, and one well documented in the design literature. Personas are fictitious representations of target users, meant to represent an archetypal user based on aggregated information about users. First proposed by Cooper in 1999, personas have proven a popular method for user-centric design both in industry and academia [249]. While initially meant as a thought experiment on the part of the designer, personas have evolved to be largely based on empirical data [250, p. 1]. According to Pruitt and Grudin, the main benefits of personas are that they make explicit our assumptions about the target audience and provide a means to exercise the documented human psychological capacity to predict the behavior and understanding of others [250, p. 5–6].

Personas may be used in our particular case to distill the findings of the surveys and interviews into a more clear, compact, and actionable form. By consolidating the 75 plus participants into a handful of personas we may more readily design for their needs and desires. In effect, the personas are a method to embed the various numerical and statistical insights of the surveys and interviews into a story, thereby creating a more memorable means of presenting and organizes this data. Of course, some caution is always necessary in processes involving reduction and conglomeration. Thus, what follows is a detailed description of both the general selection process for the descriptive sections of the personas as a group and specific justification for the development of the personas in particular.

See Figure 5.8, Figure 5.9, and Figure 5.10 for the final personas developed.

5.6.1 Template

The persona is meant to accomplish two complimentary goals, the creation of a believable and "real enough" persona and the sufficient characterization of practices pertinent to our specific design task, i.e. a visualization tool for VH. In regard to the former, we must include personal and biographical details to activate our empathy and understanding, a la Pruitt and Grudin. In regard to the latter, we must identify broader trends through which we may characterize the day-to-day practices of those working in VH. Finally, we must provide enough detail as to the proclivities and motivations of the persona such that, rather than a static description, the persona becomes a fruitful tool for the prediction of future action. The persona template may be described according to two types of information, static and dynamic, each meant to meaningfully contribute to the twin goals of persona realization and VH description. Interspersed among these sections are quotes taken directly from survey responses and interviews which demonstrate the sentiment of the persona.

Static information includes both personal information, including name, age, work, and location, as well as information meant to characterize the experience of the persona in VH, including a selection of representative software and experience ratings in four categories.

"IT and Internet" was selected as a general measure of technical competence. Any software for working with 3D digital models requires some level of competence and familiarity with computing technologies. We may consider the most basic computer literacy as the bottom of this sliding scale, and a requirement for the use of any of software solutions so far discussed. Similarly, if one is to make use of online visualizations, a basic knowledge of internet technologies is required. Beyond basic use, those with more experience in “IT and Internet technologies” may make fuller use of the more advanced functionality built into such software or even leverage available APIs and open-source code bases.

VH incorporates at least two disciplines, a domain area of application (e.g. art history, archaeology, architecture, etc.) and computer imaging technologies. As such, a VH scholar will often have two disparate skill sets, the area of application and the technical. “Area of Application” represents the personas knowledge on the topic, e.g. Roman bath complexes, Renaissance sculptors, etc., to which they seek to apply the computing and imaging technologies.

“3D Model Production” and “3D Model Analysis” represent complementary skills in the production and use of 3D digital models. 113D Model Production” includes technologies such as photogrammetry, laser scanning, and 3D modeling software. “3D Model Analysis” includes techniques for both the qualitative and quantitative inspection of the 3D digital model such as the ability to inspect geometry and textures separately, knowledge of common artifacts produced in various scanning processes, testing if a model is watertight, etc.

Rather than a description of general characteristics, “Personality” in the case of these personas is meant specifically in relation to VH practices. Three dimensions were chosen, “Leadership”, “3D Application”, and “Sharing”. “Leadership” represents the typical role in which the persona works, whether that be in a more active leadership role directing the project and its component tasks, or a more passive role executing previously defined tasks. “3D Application” describes the types of ways a persona uses 3D models, whether that be more often for illustration, i.e. merely for showing aspects of the original objects as in typical art historical publications, or more for experimentation, altering aspects of the 3D digital model, performing analytic tests and comparisons, etc. “Sharing” encapsulates issues surrounding the broader use of the 3D digital models used and produced by

Curious Scholar

Sandra Ainsworth
Age: 49
Work: University Faculty
Location: Michigan, USA

"[3D models] are a legitimate and revealing research tool."

Software

Quicktime
Sketchfab
MS Word
MS Powerpoint
Firefox

Bio

The Curious Scholar is a domain expert who would like to bring new techniques to bear on old questions. She has been exposed to computer imaging technologies only recently and while she is enthusiastic about the benefit of these novel technologies, and may even have specific applications in mind, the Curious Scholar has only limited working experience with these technologies.

"I sort of half know how to [make 3D models] … what I'm still missing is the software side of it and it's still sort of a magic black hole in my mind."

Goals

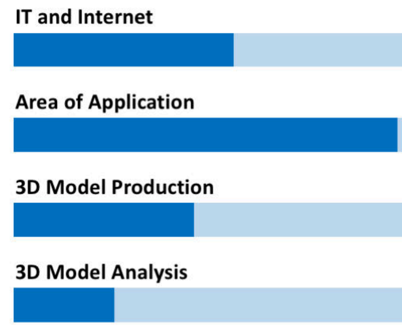
- Quality and novel research in their chosen area
- Effectively educating university students in a topic about which the Curious Scholar is passionate

Pain Points

- Needs help to perform 3D actions desired
- Lacks adequate hardware

"In short, I feel like I'm relatively new to using 3D models and how they're useful … but I'm a convert in terms of thinking about the possibilities and the usefulness of [3D technologies]."

Experience



Personality

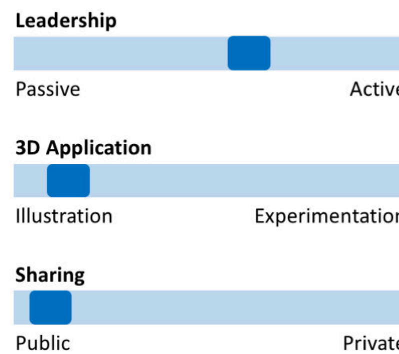


Figure 5.8: Persona: The Curious Scholar

VH Evangelist

Robert Ricci
Age: 54
Work: Lab Director & University Dean
Location: California, USA

"If we don't find a way to solve this [dissemination] issue, there is no future for this emerging discipline of ours."

Software

Meshlab
Sketchfab
RealityCapture
MS Powerpoint
Chrome

Bio

The VH Evangelist is an individual devoted to defining and refining the field of Virtual Heritage. The VH Evangelist seeks to push the boundaries of the field, both in terms of methodology and public visibility. He has a deep knowledge of how to produce high quality VH content but more often serves in leadership roles, proposing and organizing large scale projects.

"[Some] curators have a much more suspicious attitude towards this sort of global dissemination … they keep thinking that a traditional paper publication has much more value."

Goals

- Producing high profile projects
- Increasing number of scholars in VH
- Generating public excitement about VH

Pain Points

- Keeping projects going without his direct involvement
- Disseminating high quality models widely
- Convincing CH holders to allow digitization and dissemination of valuable objects

"I teach a lot of courses at the undergraduate and graduate levels in digital archaeology, virtual museums, so I train students in this field and then usually former students of mine join my lab … so we grow as an institute."

Experience

IT and Internet

Area of Application

3D Model Production

3D Model Analysis

Personality

Leadership

Passive — Active

3D Application

Illustration — Experimentation

Sharing

Public — Private

Figure 5.9: Persona: VH Evangelist

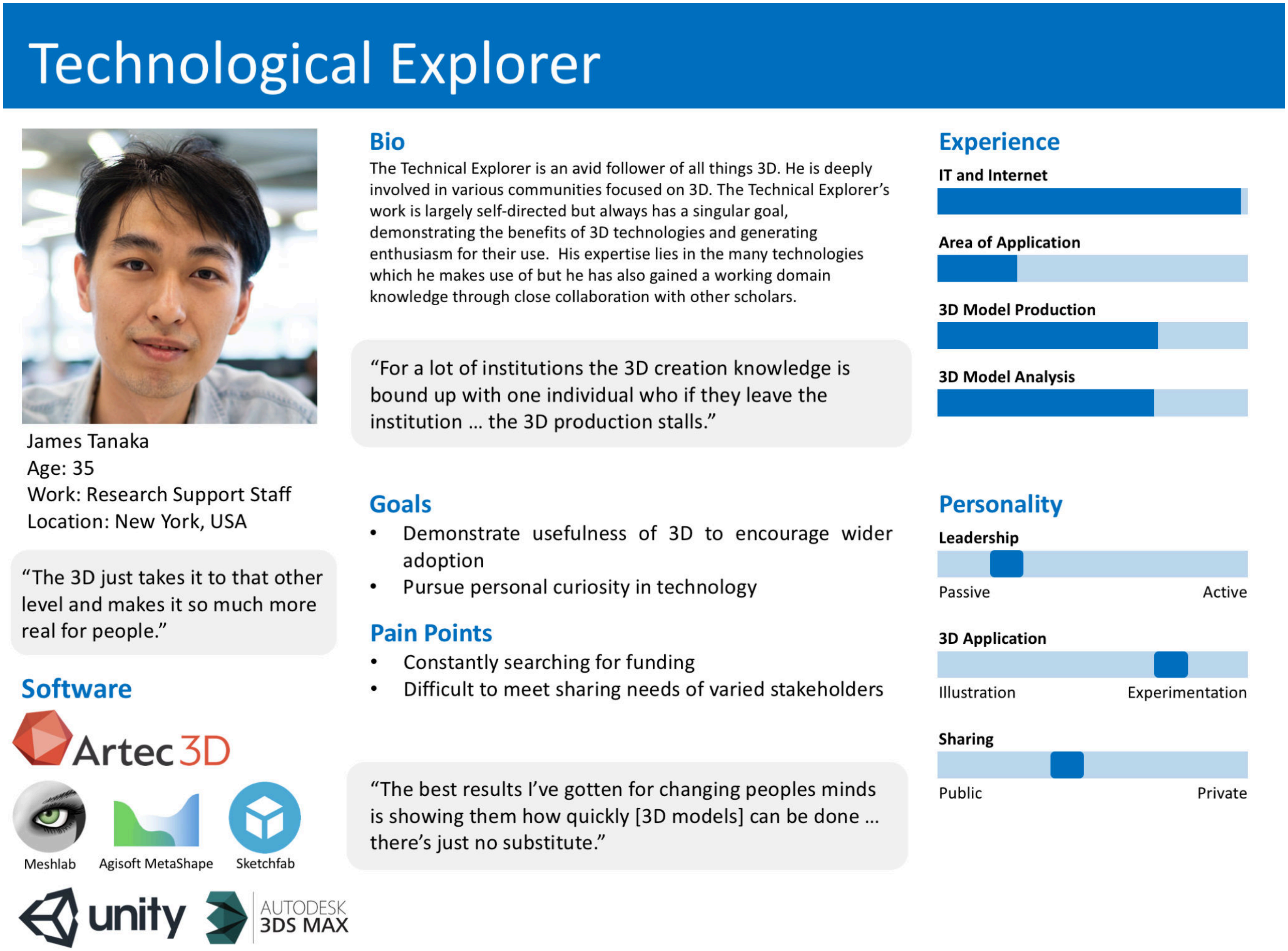


Figure 5.10: Persona: Technological Explorer

the persona. As previously discussed factors surrounding the copyright status and qualities of the original object as well as the digital derivative may limit the abilities of scholars to share work more broadly. Furthermore, some projects, such as those for private individuals or industry groups, may by nature be meant as private.

The dynamic sections are “Bio”, “Goals”, and “Pain Points”. As opposed to the static sections, Motivation is meant as a more open ended illustration of the underlying motivations of the persona, and thus as a means of predicting the persona’s perception and interaction with newly designed tools. The “Bio” is long form and narrative description of the persona as a whole, meant to contextualize the various metrics. “Goals” and “Pain Points” describe respectively primary motivators behind the persona’s use of 3D digital models and issues complicating or impeding their work.

5.6.2 Development

Three personas were developed from the data gathered from the surveys and interviews: Sandra Ainsworth the Curious Scholar, Robert Ricci the VH Evangelist, and James Tanaka the Technological Explorer. Taken together, these personas are meant to provide a rough bounds on the design space for our new VH visualization tool. As previously discussed, it is nigh impossible to design a tool for a field as varied as VH, and it would be similarity difficult to develop sufficient personas to describe the field completely.

Curios Scholar

The “Curios Scholar” was developed largely from *I1* and *I3*, individuals with deep knowledge about their respective studies but only newly introduced to the applications and uses of 3D. Two quotes were taken from *I1*:

> “[3D models] are a legitimate and revealing research tool.”
> “In short, I feel like I’m relatively new to using 3D models and how they’re useful. I work on stone working, but I”m a convert in terms of thinking about the possibilities and the usefulness of [3D technologies].”

And one quote was taken from *I3*:

> "I sort of half know how to [make 3D models]...what I'm still missing is the software side of it and it's still sort of a magic black hole in my mind."

The applications *I1* and *I3* saw for 3D digital models were primarily based upon their previous experience using the physical objects, and thus were in their use as novel tools for illustration and inspection of the physical objects. *I3* went so far as to describe her conception and use of 3D digital models as "moving 3D photographs". Both worked closely with other researchers and university staff to realize their desired 3D application. Their collaborators brought technical experience and worked largely at the direction of *I1* and *I3*. As such, the Curious Scholar makes only limited use of true 3D software, in this case solely Sketchfab, and instead more often deals with the derivatives of the 3D models put into written documents and slide decks. The Curious Scholar also represents those struggling with under-powered hardware evident both in the survey and *I2*, in their discussion of the difficulties of sharing the final products of their digitization projects with their partners.

VH Evangelist

The VH Evangelist was developed based on *I2*, *I8*, *I10*, as well as trends in the literature. All three of the quotes were taken from the *I2*:

> "If we don't find a way to solve this [dissemination] issue, there is no future for this emerging discipline of ours."
> "[Some] curators have a much more suspicious attitude towards this sort of global dissemination...they keep thinking that a traditional paper publication has much more value."
> "I teach a lot of courses at the undergraduate and graduate levels in digital archaeology, virtual museums, so i train students in this field and then usually former students of mine join my lab...so we grow as an institute."

The VH Evangelist is one who is deeply involved with the field, participating broadly in events, engaging with the public, and educating budding new scholars. The VH Evangelist is meant to represent the current observed in publications of those reflectively interested in VH, its characterization, and betterment, such as Maschner [77] and Koller et al. [78].This involves both engagement within the field between scholars as well as public engagement. As such, the issue of dissemination, and therefore visualization, is central to the work of the VH Evangelist and thus the previous issues of high-resolution sharing and accessibility discussed in Section 5.5.1 and Section 5.5.2 are especially impactful. While such individuals are often engaged in groundbreaking work to extend the use cases of 3D, this work is tempered by more illustrative cases meant for wider public display, thus the more central value of “3D Application”.

Technological Explorer

The Technological Explorer was developed from *I5*, *I6*, *I9*, *I10*, and *I11*. With quotes taken from *I5*, *I6*, and *I10*:

> *I5*: “The 3D just takes it to that other level and makes it so much more real for people.”
> *I6*: “For a lot of institutions the 3D creation knowledge is bound up with one individual who if they leave the institution...the 3D production stalls.”
> *I10*: “The best results I’ve gotten for changing people minds is showing them how quickly [3D models] can be done...there’s just no substitute.”

The Technological Explorer is meant to represent those with significant experience in the technologies facilitating VH but less so in domain knowledge like the humanities. Granted, the majority of the individuals interviewed and surveyed also had extensive domain knowledge. For example, while *I10* served in a position supporting university students and faculty with technology, they had actually formally studied the humanities and history. It seems nigh inevitable to be engaged in and enthusiastic for VH work without a deep interest in the non-technical objects and topics studied. However, in order to more fully capture the variety of VH as a field, the Technological Explorer was given a lower level of “Area of Application” experience. Even so, considered relatively to the general population, the Technological Explorer may be considered to have significant experience on the topics studied. The Technological Explorer works with a wide group of other scholars and professionals, aiding them in equally varied projects. As such, they are more often confronted by issues of copyright and ownership that complicate the public sharing of the projects and their 3D derivatives. Thus, the Technological Explorer has the “Sharing” value closest to “Private”.

5.7 VH Ambassadors

The use of 3D models of CH is certainly a wide spread and (fairly) well publicized practice. A large number of the leading academic institutions in the United States and Europe fund groups dedicated to the investigation and execution of projects utilizing 3D to work with CH. However, the role of these groups within their institutions is almost always one of research and development, meant to explore the possibilities of 3D, without any clear avenue for the more holistic or long-term inclusion of these practices into the larger functioning of the institution. There is no lack of individuals at these institutions convinced of the vast benefits which 3D has to offer the study, preservation, and dissemination of CH. And this fact is evinced by the existence of these many 3D groups. However, on the institutional level, 3D is not nearly at the same level of adoption. The institutional efforts to incorporate 3D tools into their official workflows are few and far between, the Smithsonian and British Museum being notable exemptions.

But the question then becomes, how may those involved in the production and use of 3D models of CH, and convinced of their usefulness and benefit, further the levels of adoption at a broader scale and encourage the growth of the field?

I3, *I6*, and *I10* all spoke to the importance of demonstration of and enthusiasm for 3D when convincing others of the potential benefit for 3D, all from differing perspectives. *I3*, as a museum curator, had been convinced of the value of 3D, and the useful role it could play at their institution, through the "positive energy" and "enthusiasm" of a handful of individuals that *I3* described as "ambassadors". According to *I3*, the most important method for convincing people such as themselves was "show-and-tell", i.e. the in person demonstration of 3D applications. Within their institution, *I3* had become a proponent for 3D, and had found their colleagues receptive to the idea. But within only limited technical experience, her ambassadorship was inhibited and often relied on the presence of additional help to provide the hardware and operational expertise.

I6 had an identical experience, albeit from the opposite position. In *I6*'s position as the primary individual responsible for 3D efforts at their institution, *I6* was no stranger to convincing others for funding. In the experience of *I6*, "there's absolutely no substitution for an in-person demonstration." This lesson was learned through working directly with museums and museum staff. In one case, *I6* went to Italy to scan a small collection of objects in a museum collection for a graduate student. Initially, *I6* was only allowed one day to scan, the day that the museum was closed. But upon seeing the results of the first days work, the curators quickly requested that *I6* spend their remaining days going about the museum scanning objects, while the museum was open. Only the briefest of demonstrations had completely convinced, and enthused, the staff on 3D scanning in the museum.

I10 had seen the effect of 3D ambassadors from a third-party perspective. In dealing with institutions currently involved in or considering 3D digitization projects, *I10* found that "there is often a champion for 3D who...pushes the use of 3D and then they get more people interested by explaining it to them in the right terms." However, the work of these ambassadors is not quite as straight forward as simple demonstration. For *I10* had observed that the success of such ambassadors was also dependent on the social and organizational factors of their institution.

Thus, an important means of increasing the adoption of 3D technologies and techniques and furthering the field of VH is in the demonstration of 3D applications by VH ambassadors, whether it be physically using 3D prints or digitally using visualization tools. However, as has been discussed throughout this and preceding chapters, the visualization of high quality, high-resolution 3D digital models is hampered by lacking hardware and software, especially among those who would benefit most from demonstrations, like museum professionals. *I11* described this very same issue as the "dissemination bottleneck". VH scholars are able to produce stunning and compelling 3D digital models of CH objects but are stifled by a lack of tools for disseminating these models. Therefore, to further the work of VH ambassadors and the success of VH as a field, VH visualization tools must continue to be developed and refined, with particular attention to those that facilitate the wide sharing to potentially under-powered devices.

5.8 Conclusion

The surveys and interviews demonstrate the importance of sharing and visualization to VH scholars and factors hindering their wider use an adoption. While the survey was an imperfect tool for the full enumeration of VH, given its academic bias, it still served as a launching point for fruitful discussions as to the nature of those involved in VH, showing the diversity evident in the field. Ultimately the results of the survey and interviews present a fuller picture of the practices and peoples comprising VH than just the previous literature review, and motivated the need for an online and protected visualization tool to address dissemination issues and security concerns.

CHAPTER 6
VH FEATURE IMPLEMENTATION

Seymour, as developed in Chapter 4, represents a novel tool for the protected visualization of high-resolution 3D digital models of CH. However, as shown in Chapter 5, practitioners in VH have far more diverse and complex needs than simple visualization. Thus, the goal of this chapter is to demonstrate the extensibility of Seymour and develop a more useful VH tool through the implementation of features based off of the practices and needs of VH practitioners as discovered through the course of the literature review, survey, and interviews. Furthermore, in the process of extending Seymour, methods of interacting with the software may be explored leading to a more complete Application Programming Interface (API), useful for developers seeking to work more closely with the Seymour software.. Code for all of the tools discussed here is available on the project Github page [185].

6.1 Snapshot Gallery

Motivation

For the first feature, let us consider a means to extend Seymour solely from the client-side, one based on traditional means of object inspection in the art historical tradition. The ultimate purpose of Seymour, and the 3D digital models for which it is intended, is the study of CH objects. As argued in Chapter 3, these 3D digital models form the latest development in a long history of mediums used to reproduce and study CH. Therefore, one of the uses for 3D digital models is in the traditional sense of its predecessors, namely static views of the original. As Seymour stands in its basic form, one may manipulate the 3D model to find the desired view, take a screenshot, and then use the resulting image as a reference figure. This method requires additional software (although software typically bundled with operating systems) and technical skills. But what if the user desires to make a minor change to the view? Or compare multiple views and then make minor changes between them?

As a 3D application, Seymour may be used to generate a near infinite number of views of the object under examination. However, depending on the interaction methods used, reproducing a view exactly across multiple sessions can be a non-trivial task, especially given the various levels of ability in operating such 3D interfaces observed in Section 5.3. Thus, reproducing a view to make minor changes or for the purposes of presentation can be a potentially difficult process. The reproduction and manipulation of these views becomes even more time consuming when one begins to compare views between one another, as is often done in art historical examination and research. Furthermore, these manipulation issues extend beyond the initial user generating the views, as users may want to share the views with other scholars who themselves may wish to make minor adjustments.

The basic architecture of Seymour allows for a simplification of this process. Since each user generated view is rendered on the server as a 2D image, these images may themselves be presented, organized, and manipulated. Furthermore, by linking the rendered frames with their 3D parameters, the rendered images may be used as control input to recover the 3D view in the client. We call this functionality a “Snapshot Gallery”.

Implementation

The Snapshot Gallery was implemented entirely client-side, with no changes made to the server-side code. The image gallery feature required the implementation of three functionalities: displaying the selected snapshots, comparing images, and storing the 3D view information with the image. The former two also required the addition of new interface elements. The final developed tool is shown in Figure 6.1.

Numerous rendered images are generated over the course of a user session. So as to avoid the snapshot gallery becoming over-full with intermediary views and duplicates, only user selected images are saved to the snapshot gallery. Upon clicking the snapshot button the current image and

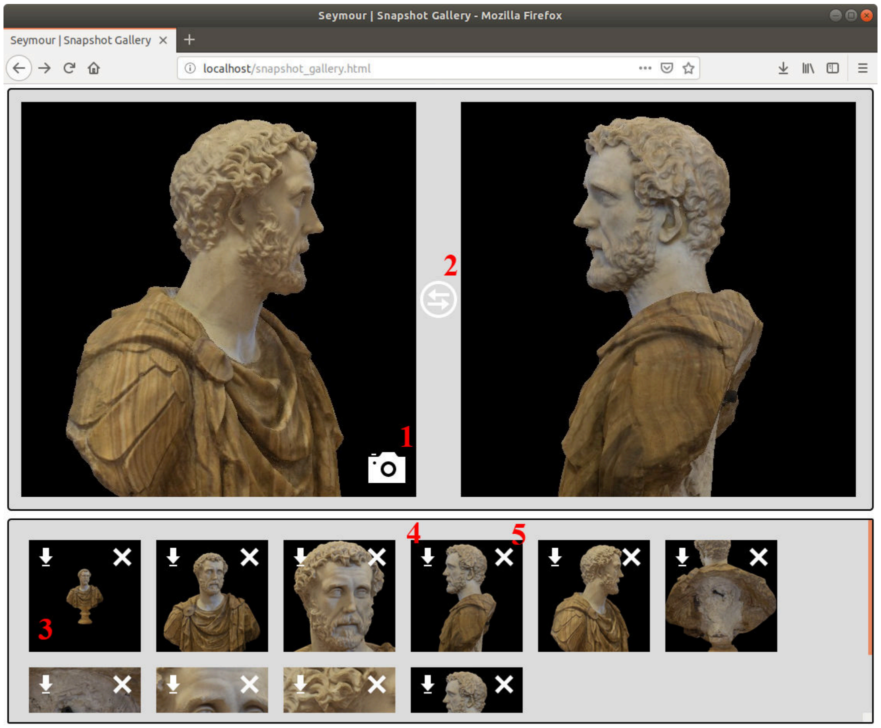


Figure 6.1: The user interface for the “Snapshot Gallery” with annotated UI elements: 1 snapshot button, 2 switch button, 3 snapshot thumbnail, 4 download button, 5 delete button.

relevant 3D viewing parameters are stored in the snapshot gallery. Selecting a snapshot from the snapshot gallery reorients the 3D view to the view of the snapshot. Snapshots may be removed by clicking a delete button and downloaded by clicking a download icon.

To facilitate the direct comparison between snapshots, an additional snapshot viewing area was placed alongside the 3D view. At any point, the user may move an image to the 2D view by clicking the transfer button, both saving the snapshot and displaying the saved view in the large adjoining area.

The majority of the implementation for the Snapshot Gallery was straightforward, and required no interfacing with the Seymour client or server. The saving, downloading, and switching of the snapshots was accomplished through vanilla JavaScript and CSS. However, in order to accomplish the reorientation of the client 3D view, basic code was required to communicate between the page and the Seymour client library. This was accomplished through the use of basic API calls for getting and setting the client viewing parameters, documented in more detail on the project Github page

[185].

Comparison, not just between views of the same object but between differing objects, is a fundamental action in art historical investigation. Thus, the concept of side-by-side comparison that forms the basis of the snapshot gallery feature could be extended to include inter-object comparison as well as 3D comparison through the use of side-by-side viewing instances of Seymour. These instances could even be connected such that interaction with one mirrors the interaction with another.

6.2 Path Render

Motivation

As discussed in Section 5.3, VH practicioners make use of dynamic renderings of 3D, e.g. videos, in much the same way as static renderings like images. Both allow for sharing derivatives of 3D models that do not rely on any 3D capabilities of the client machine nor knowledge of the user in the operation of 3D interfaces. In its basic form, Seymour may only display single, disjointed images, as interactive manipulation of the model is only shown with the low resolution reference model.

Seymour may be extended to support video-like rendering through client side modification to rendering requests and display. As video is only a series of still frames displayed consecutively and at speed, series of rendered frames may be requested and displayed sequentially. Thus, we may allow for the generation of more dynamic content through the specification and rendering of user defined images sequences, a feature we call the “Path Render”.

Implementation

The implementation of the Path Render requires two components: user specification of a path and display and interaction with the rendered frames. The former may be accomplished through the simple recording of user interaction. As a basic implementation we record user rotation and translation. To begin the recording, the user selects the record button, see Figure 6.2. While recording, frame requests are registered each frame, not only at the end of the interaction. Requested

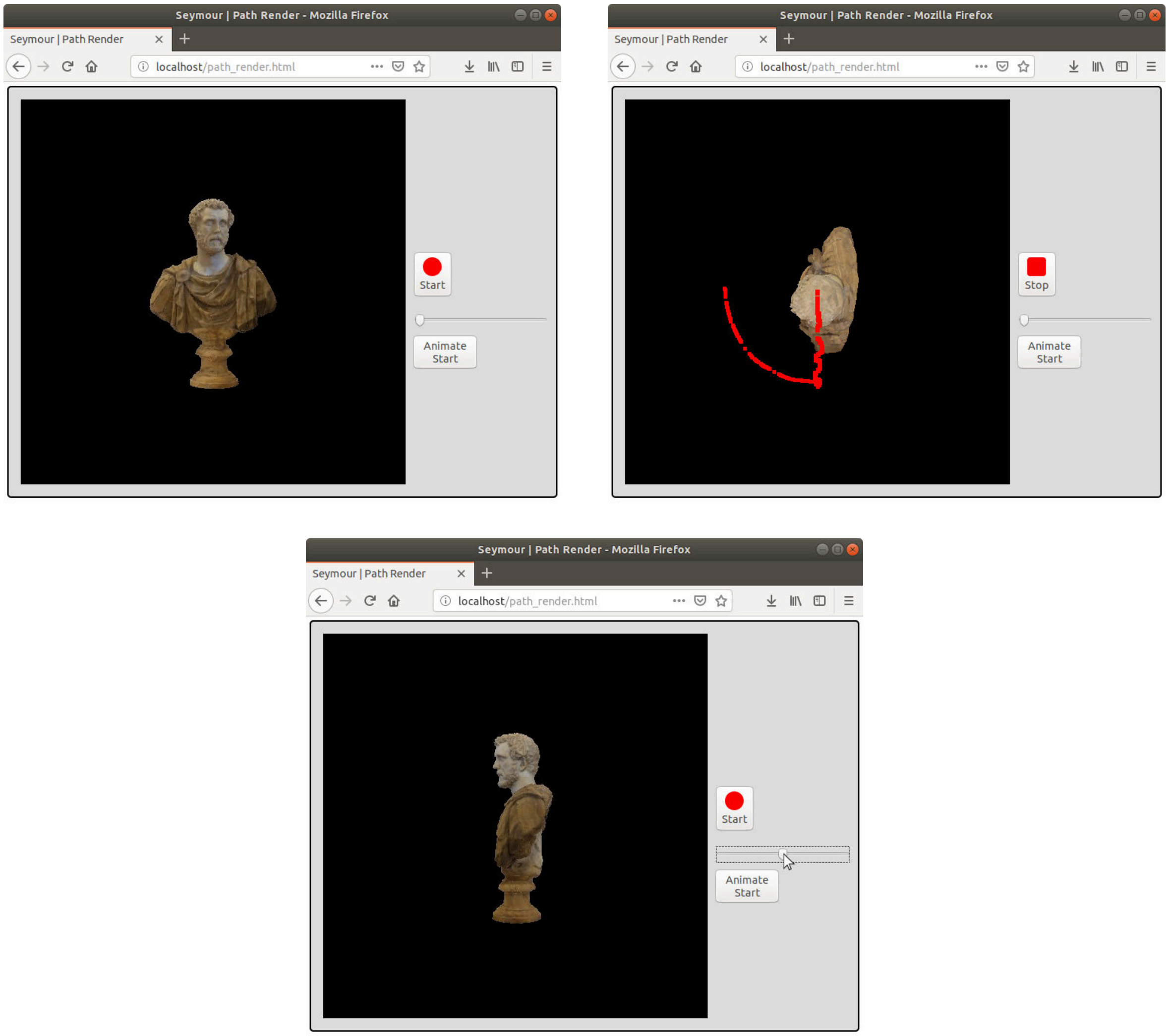


Figure 6.2: Sample frames from the Path Render implementation with three user interface elements, the record button (top), frame slider (middle), and the animator (bottom). Frames are ordered left-to-right and top-to-bottom.

frames are loaded in hidden <img> elements. By loading the rendered frames hidden on the page, they may be stored in the browser cache, and therefore when switching between frames using the frame slider new render requests do not need to be sent to the server. In order to provide visual feedback to the user, the recorded path may be visualized interactively. At each frame during recorded user interaction, a red dot is visualized directly in front of the camera, in this way showing the viewpoints which will be rendered along the path.

The display and interaction with the rendered frames requires only minimal additional development. The display for the frames is an HTML <img> element. This element may either be overlayed on the 3D view as with the normal frame requests, or placed alongside the 3D view so as to facilitate similar comparison as in Section 6.1. The frame slider allows the user to manually step through the rendered frames, already loaded into the browser cache. The animator below, animates the frame slider using a linear animation curve, although this could be simply changed in the JavaScript code.

The implementation of the Path Render exposed three issues with Seymour, two architectural and one API related.

The per-frame requests used to render the path pose both a security and performance issue. Such a series of requests could be used for reconstruction of the 3D geometry, and would likely trigger client session monitoring protocols to block the client, as the activity would appear malicious. While the session monitoring tools are not the ultimate means of Seymour's security, deactivating them would be a compromise to the overall protection of the system, but one which could be justified in certain use cases.

The series of requests also presents a performance issue, as the quick series of rendered frames may take up an inordinate amount of server resources. On a low-traffic system, this load would not be an issue, but it would quickly become one at scale. To deal with such a volume of rendering requests, which are not time sensitive, one could implement a priority queue for frame requests. Thus, the Path Render frames would be given lower priority than regular render requests, thereby ensuring that regular users did not experience irregular latency and server resources were more effectively distributed.

Finally, in implementing the Path Render, an additional functionality of the client-side code needed to be exposed. For simplicity, the user recorded path was only visualized on the client. But this means that any rendered frames sent by the server will result in a jarring change to the scene. Therefore, during the recording of user interaction it was necessary to disable the automatic high-resolution renderings of the client and instead manually request the rendered frames with the hidden <img> elements. This was accomplished through API calls to start and stop the client-side rendering as well as retrieve the rendering request URI from the client instance.

6.3 Navigation Selection

Motivation

Seymour implements what Colin Ware described as "world-in-hand" navigation, wherein a user manipulates individual objects as if they were to physically grab them. However, as discussed in Section 4.2.1, this is but one of the possible interaction paradigms, and is primarily suited for interaction with individual objects. If one were to visualize a larger architectural space, terrain, or arrangement of statues, world-in-hand navigation would feel encumbering and artificial. In cases such as these, flying or walking navigation is more appropriate, as they more closely imitate the ways in which we navigate through spaces, as opposed to inspect individual objects. World-in-hand itself is a largely abstract metaphor, and can be implemented in any number of ways. As previously discussed in Section 4.2.1, the rotation schemes subsumed under world-in-hand also have varying cases of application. Conveniently, Seymour makes no commitments to any navigation scheme, the client is able to manipulate the objects and camera in any manner, with only the final result sent to the server for rendering. Thus, Seymour may be simply extended to support alternative means of navigation.

Implementation

Flying navigation was implemented on the client alongside the default rolling ball control and a turntbale style rotation, with the ability to switch between the three. Given that the test model used thus far, the bust of Antoninus Pius, does not lend itself to space-centric navigation, an architectural model was used to demonstrate this new feature. A photogrammetric model of fountains from outside the Chapel of Saint Nicodemus in Pluméliau France was used, downloaded from Sketchfab [251]. The resulting webpage, with the new model and interaction selection is shown in Figure 6.3.

The user specification and modification of interaction schemes required the ability to disable the default model interaction on the client. The rolling ball default was not removed entirely, as one would expect most applications to desire sensible defaults and use out of the box. Once

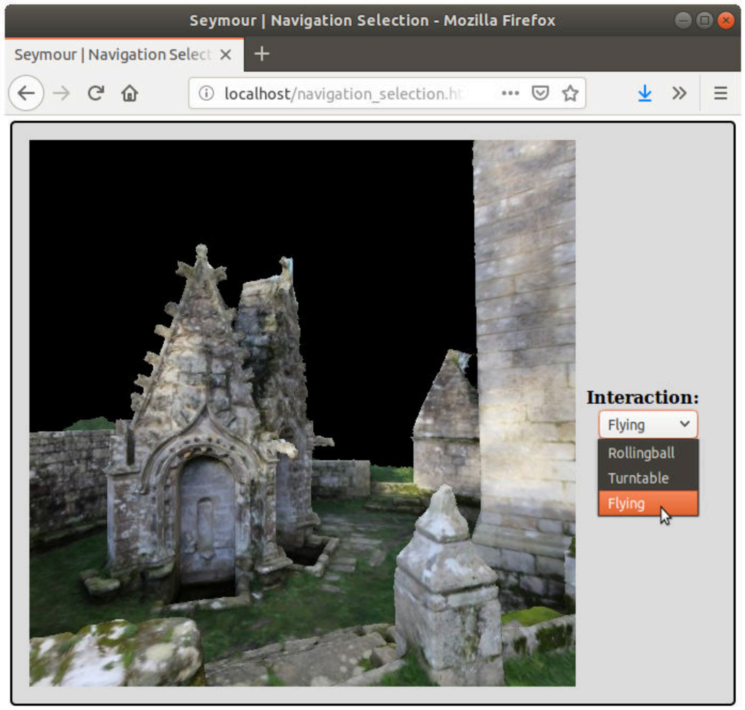


Figure 6.3: A screenshot of the Navigation Selection page.

disabled, flying navigation as well as the alternative world-in-hand rotation were implemented using functionality from the Three.js library and modification of the scene's camera and models, exposed on the client. In flying navigation, the WASD or arrow keys serve to move the camera forward, backward, and side-to-side. The user then clicks and drags on the screen to change the pitch and yaw of the camera. In turtable navigation, meant to simulate as if the object was on a turtable that a user may spin, user movement along the x-axis controls the y-axis rotation of the model and vice versa.

6.4 Scale Bar

Motivation

Measurement is an integral method for art historical and archaeological investigation and as shown in Section 5.5.3 it is a method of analysis that VH practitioners have come to expect in visualization tools. With 3D models, one may perform three-dimensional measurements. However, such methods involve a certain level of coordination manipulating the 3D digital model. Maschner [77] proposed an interesting alternative measurement tool.

Rather than provide measurement in the visualization tool, the viewer for zooarchaeological specimens discussed by Maschner allows for the generation of figures that may be used for simpler 2D measurement. The 3D model is rendered with an orthographic projection,[1] as one would with an architectural drawing, thereby preserving relative distances between points in the image. Then, a scale bar is superimposed on the rendered image. This rendering method allows for one to take direct measurements from the image. However, while this method is certainly simple to use and results in images useful for more traditional means of sharing and publication, it fails to leverage the full capabilities of the 3D models.

Various forms of measurement may be performed in 3D space, depending on the number of points used. With a single point, one may measure the x, y, and z coordinates at a single point on the model. With two points, one may measure a straight line, or euclidean distance, between them, as would be done with calipers or a laser rangefinder, or the distance across the surface, the geodesic distance. With three points, and thus a defined area, one may measure area, both surface and direct. Finally, four or more points allow for volumetric measurement.

Implementation

As a preliminary, Maschner's scale bar figure generation was implemented. The additional measurement schemes discussed may be the work of future implementation.

The object space to real world scale conversion was hard coded. As detailed in Section 4.3.1, the 71 cm tall bust of Antoninus Pius was scaled to 2 object units in height. Therefore, 1 object unit translates to 35.5 cm. A vector of length 1 was placed in the scene and projected to screen coordinates, giving a pixel to real-world scale ration. This was used to scale the scalebar appropriately on the screen. The scalebar is rendered below the 3D viewing window with five labeled metric units. As the user scales the model scales the model up and down, the scale bar changes appropriately.

[1] See Section 3.2.1 for a discussion of orthographic versus perspective projections.

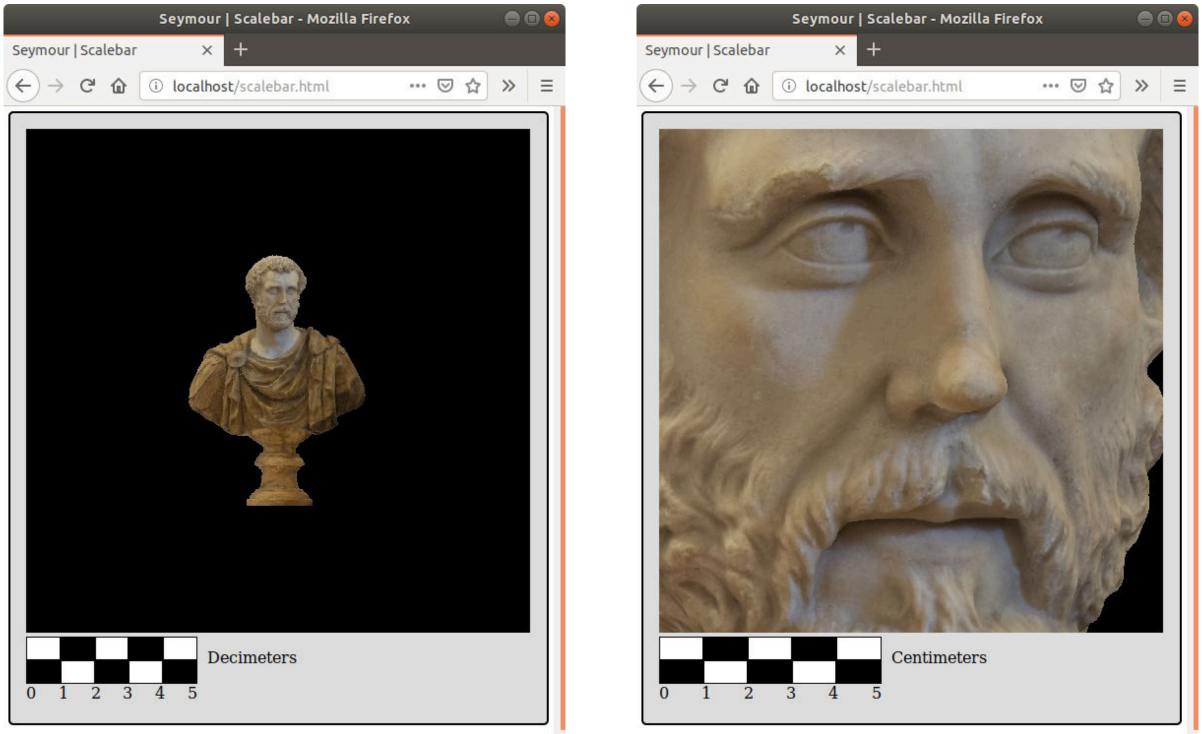


Figure 6.4: Two frames from the scale bar feature, one showing a scale in decimeters (left) and one a scale in centimeters (right).

When the scale bar exceeds the width of the viewing window or is less than 10% of the length, the scale bar switches to the next metric unit by a power of 10. Two views of the scale bar feature are shown in Figure 6.4.

While measurement is certainly an important tool for the analysis of 3D digital models of CH, it poses a significant issue to the security of the system discussed here. If no restraints are placed on user specified measurements, an attacker could use such measurements to fully reconstruct the 3D geometry. The exact reconstruction algorithm will depend on the types of measurement available. In the simplest case, an attacker could directly record measurements from the model, in 3D space, using single point measurement. An attacker may also reconstruct the 3D geometry using direct, point-to-point measurement. Supposing an attacker could automate an exact process to repeatedly select the same point, an entirely feasible task accomplished through automated mouse movements using exact pixel coordinates, an attacker could retrieve a series of distances between N unknown points. With a sufficient number of distances, may arbitrarily assume the position of a single point in the set and then solve for the positions of the rest of the points in the set.

6.5 Distortions GUI

Motivation

As shown in Section 5.3, both the surveys and interviews evinced the wide range of technical experience evident among VH practitioners. While highly technical users are represented, there is also a distinct group with only minimal ability in using 3D software and technologies. The museum curator interviewed (*I3* in Chapter 5) represented the latter group. However, she was still able to access, manipulate, and upload models on Sketchfab. Likely, it is Sketchfab's visual workflow, clear textual elements, and lack of any code manipulation that makes it a tool usable by users such as the museum curator.

3DHOP codified this variety in technical skills as a design principle for their system, aiming to ensure "ease-of-use" (see Section 5.5.2). However, the author's recognized that, with a group as hetereogeneous as VH, designing for all users represented in the field would be exceptionally difficult. Nonetheless, to create a 3DHOP page, one must manipulate and edit some basic HTML code, a task, in the estimation of the VH lab head (*I2* in Chapter 5), beyond the basic expertise expected of a VH practioner, at least those that have entered the field from the humanities. To address this issue, Ponchio et al. [135] developed an easy web publishing service as part of ARIADNE, a larger repository project meant to address the sharing, discovery, access, and (re)-use of 3D digital models of CH.

While the initial setup of Seymour and its server-side components is a technical task not easily simplified, the fine-tuning of the Seymour server instance may be presented in a more user friendly way. As argued in Section 4.2.3, settings for the defensive distortions should be determined on a per-application basis so as to achieve a proper balance between security and the quality of user experience. However, the parties responsible for the application of Seymour instances may not themselves be overly technical nor the one responsible for setting up the server-side system. Therefore, to facilitate the broader use of Seymour by users with limited technical experience, a

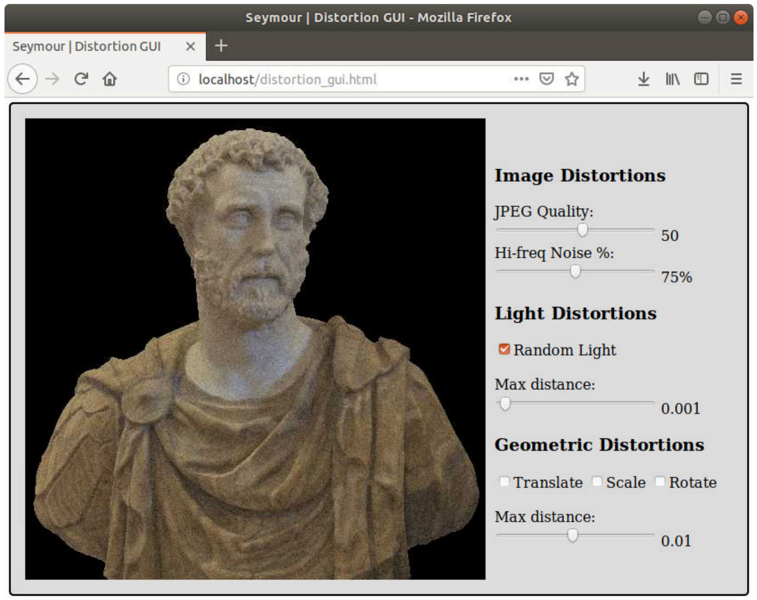


Figure 6.5: A screenshot of the Defenses GUI page.

graphical user interface (GUI) for manipulating the server-side defensive parameters may be created, one which provides immediate visual feedback as to the effects of the distortions.

Implementation

Besides the basic interface elements on the web page, the implementation of the Distortions GUI required two additional developments, both on the server: the creation of an URI and user authentication.

A new URI path was added to the Seymour communications module, one which controls the defensive settings of the server, as opposed to the lighting or geometry. This taks required modifications to the Seymour server code itself. Ideally, the communications module would allow for the more flexible addition of custom URIs, such as can be seen in common libraries for the creation of APIs like Slim PHP [252] and Codeigniter [253].

In order to prevent unauthorized users from modifying or completely disabling the server defenses, basic HTTP authentication was added to the newly created distortion URI. HTTP authentication may be easily added to paths on the Nginx server through minor modifications of the Nginx configuration files and the addition of an .htpasswd file. With the new configuration, the user is prompted for a username and password, which are then cached for future requests.

The resulting GUI is shown in Figure 6.5. Using sliders and checkboxes, the user may modify image, light, and geometric distortions. With each modification an updated rendering of the current view is shown in the viewing window. In this way, the perceptual impact of the distortions may be directly assessed, without the need of any direct modification of server-side code or management of Seymour instances.

6.6 Future Work

The development of additional features for Seymour uncovered a variety of useful modifications to the software and its client and server libraries, not all of which were implemented. What follows is a brief list of the afformentioned improvements, among others, specifically targeted at Seymour's potential for usability, extensibility, and reuse:

- **Priority Queue**: A priority queue is an extension of a more typical queue, a first-in-first-out data structure, that assigns each element a priority, with higher priority elements taking precedence over lower priority elements. A priority queue could be used to implement a more sophisticated order of request processing by the Seymour server, one that would support more varied use cases, such as the path renderer discussed earlier in the chapter.
- **Custom Routes**: At a fundamental level, Seymour is an architecture to provide access to server-side computing resources to client machines. This is accomplished through the use of URI routes, sending client requests to the appropriate functions. However, Seymour need not only provide rendered images. The system could be extended to provide any other number of services. In so doing, one would add new URI routes, directed to the new services. In its current state, such routes may not be easily added to Seymour, a point in need of improvement.
- **Side-by-side Rendering**:The Snapshot Gallery discussed in Section 6.1 only begins to address the many ways in which scholars using 3D digital models of CH use comparative study. Perhaps most importantly in this regard, Seymour
- **Model Loading**: To further the use of model comparison, users could be given the ability to upload models, not just on the client, but replicated on the server. This would could potentially pose security issues, especially since adding known 3D geometry to the scene would allow an attaacker to deduce exactly the distortions to both geometry and light parameters. However, given the varying security needs of applications in VH, such a feature could prove useful.
- **Advanced Measurement**: Alongside comparison, measurement is an essential method of art historical interaction with physical objects, and thus one that should be reproduced for digital

media. 3D measurement in various forms such as direct, geodesic, volumentric, and surface area should be added as basic features.

This list is but a minor subset of the myriad applications for visualization more generally, and Seymour in particular, to problems in VH and its attendent fields. As VH continues to develop, VH practioners will no doubt uncover new and exciting ways of studying and interacting with the 3D digital models of CH they produce and use. Visualization tools for VH, like Seymour, must remain flexibile so as to have the potential to change and adapt with the dynamic field for which they are intended.

CHAPTER 7

CONCLUSION AND LOOKING FORWARD

We began with a theoretical investigation of models and their function in the production of new knowledge through a review of the literature and applications for 3D digital models of CH, presenting a definition of models with particular attention to the important role of materiality. 3D digital models were described as collections of properties interpreted and embodied by the actions of the modeler, properties defined by their causal force. Not only does did this description provide a frame through which to discuss and investigate the usefulness of models more broadly, it also demonstrated the unity of 3D digital models with earlier forms of representation in art history and archaeology. Thus, a history of material models in art history and archaeology was presented, culminating with recent developments in Web3D. This discussion served to identify existing methods for the effective use of 3D digital models of CH and current features required and desired in VH, especially those met and unmet by current visualization solutions.

To the end of addressing some of these unmet needs, security especially, we surveyed remote rendering, a study most recently focused on heterogeneous clients like smart phones and web browsers. We applied the most recent developments in the field, e.g. web browser clients, along with the handful of directly relevant publications, to an architecture for a remote rendering system based on that presented by Koller et al. [10]. The defensive distortions included in the system were shown to be effective in preventing both structure-from-motion and shape-from-shading reconstruction attacks, although the use of more sophisticated hybrid algorithms may still pose an issue. Additionally, we demonstrated the scalablity of the system and the ability of a single server instance to support over 50 concurrent users.

Through the use of surveys and interviews, we investigated and more fully described the target users for the tool began in Chapter 4. These results provided support for major trends observed in the literature and were ultimately distilled into the more actionable form of personas. In the process, we demonstrated the variety of individuals and practices evident in the field of VH, a variety which also applied to the perceptions of indivudals on the need, or lack thereof, for security in the visualization of 3D digital models of CH.

Finally, we applied the insights gained through the literature review and empirical study of target users to design and implement a selection of additional features for the remote rendering visualization tool. In the implementation of these features, the extensibility of the visualization tool was demonstrated and improved, adding useful API functions for interacting with both the client and survey libraries.

Future work includes the development of more advanced defensive measures based on adversarial machine learning; improvements to the scalability of the system through the implementation of load balancing and more nuanced handling of render requests through priority queues; and the implementation of numerous useful features for VH such as advanced measurement, different rendering techniques, geometric analysis (like CloudCompare), etc.

On the scale of an individual project, the visualization tool developed here, Seymour, certain has numerous directions for future work. However, Seymour also represents a step forward towards loftier goals for the field of VH and 3D digital models of CH more generally. In particular, the use of cloud services in VH and the inroads of 3D digital models oof scholarly quality into public media and entertainment.

Cloud services, computing resources delivered on demand over the internet, is an industry experiencing massive growth, largely thanks to the efforts of cloud computing providers like Amazon AWS and Microsoft Azure [254] [255]. The client-server architecture of remote rendering allows 3D graphics applications to take advantage of the computing resources offered by the cloud. Thus, as cloud resources lower in cost and increase in processing power, the use of such services may present a valuable benefit for potentially underfunded groups and institutions working with CH to host and run high quality graphical applications.

The use of 3D technologies in entertainment are now well established, with many of the highest grossing films making heavy use of 3D computer graphics and the gaming industry valued at over

$100 billion [256]. 3D digital models of CH are well represented among both the film and gaming industries. Films making use of CH are even present among the top 250 highest grossing of all time with titles such as Indiana Jones, Night at the Museum, and Gladiator [257]. AAA game studies also produce content based on CH, such as the Assassins Creed titles that are set in ancient locals like Greece and Egypt. However, the one commonality between these titles is there imaginative, anachronistic, and often outright fanciful interpretation and representation of the ancient world. Perhaps, as scholars continue to produce higher and higher quality 3D content of CH, enabled by improved tools, workflows, and training, this content may make inroads into more public arenas, thereby exposing the public to the important work done by those in VH.

Ultimately, the visualization tool and its development presented here provide a novel 3D software, specifically with in regards security, and a template for the user-centric development of future tools for VH. There will never be a single, ultimate software to address the variety of VH research actions. Rather, it is through the development of novel tools and the continued support and improvement of existing tools, that the toolkit of the VH practitioner may be expanded, and the field of VH thereby furthered.

Appendices

APPENDIX A

DICTIONARY STATISTICS FOR "MODEL"

WordNet was used to retrieve dictionary definitions of "model" and compute statistics for the number of meanings per noun. WordNet is a lexical database for the English language that groups words into sets of synonyms called "sysnets". These sysnets are indexed according to a number of parameters, but the one pertinent to our task here was "part of speech", abbreviated to pos. Table A.1 shows the Python code used to look up the 14 definitions of "model" in WordNet, retrieve all synsets of nouns, and count the number of meaning for each synset. From this, a mean and standard deviation were computed and "model" was located in the distribution.

```python
from nltk.corpus import wordnet as wn
from collections import defaultdict
import numpy as np

# -- Definitions of 'model' --
for synset in wn.synsets('model', pos='n'):
  print(synset.definition())

# -- Statistics for English nouns --
words_by_pos = defaultdict(set)

for synset in wn.all_synsets():
    pos = synset.pos()
    for lemma in synset.lemmas():
        words_by_pos[pos].add(lemma)

lemma_name = []
num_meanings_per_noun = []

for word in words_by_pos['n']:
  lemma_name.append(word.name())
  # No. of meaning for a word given a POS.
  num_meaning = len(wn.synsets(word.name(), pos='n'))
  num_meanings_per_noun.append(num_meaning)

arr_meanings = np.array(num_meanings_per_noun)
mean = np.mean(arr_meanings)
std = np.std(arr_meanings)
print("Mean:  " + str(mean))
print("Std:  " + str(std))

arr_names = np.array(lemma_name)
for i in range(1,max(num_meanings_per_noun)+1):
  count = num_meanings_per_noun.count(i)
  print(str(i) + ":  " + str(count))
  if count<30 and count!=0:
    print(arr_names[arr_meanings==i])
```

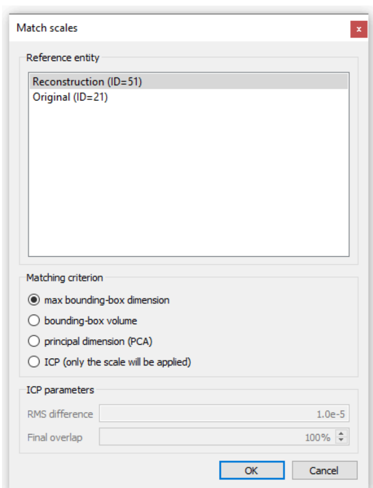
Table A.1: Python code used to retrieve definitions for “mode” and compute dictionary statistics using WordNet.

APPENDIX B

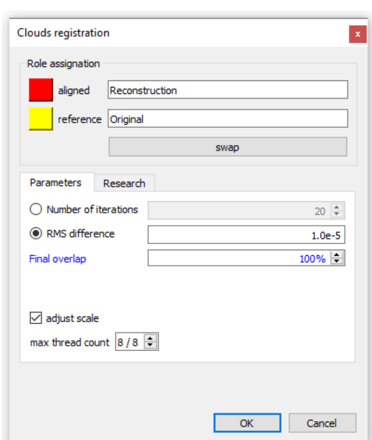
MESH COMPARISON WORKFLOW

The workflow to compare the photogrammetric reconstructions to the original model using CloudCompare was as follows:

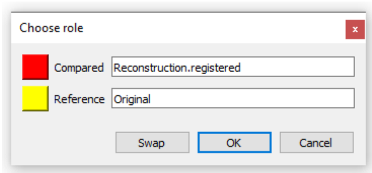
1. Tools -> Registration -> Match Bounding Box Centers
2. Edit -> Translate/rotate
3. Tools -> Registration -> Match Scales

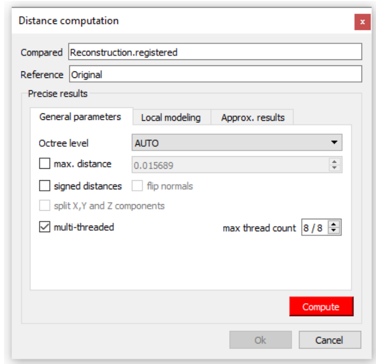


4. Tools -> Registration –> Fine Registration (ICP)



5. Tools -> Distance -> Cloud/Mesh Distance





APPENDIX C

SURVEY: USES OF 3D

What follows is a reproduction of the online survey used in Chapter 5. In order to print, the survey has been slightly modified from its original form. However, the textual content of the questions and the available options have been reproduced exactly.

Welcome to Leif Christiansen's *Uses of 3D* survey!
Thank you for agreeing to take part in this survey on your use of 3D models. The results of this survey will form an integral part of my dissertation. This survey should take only 10-15 minutes to complete. Be assured that all your responses will be kept strictly confidential and you may choose to leave any question blank.

For more information on the study see http://pages.iu.edu/~leifchri/1810783157-3D_Model_Use_in_VH.pdf

Feel free to contact me at leifchri@iu.edu with any questions or concerns.

Note: Throughout the survey "3D digital models" will be shortened to "models".

Select all fields in which you primarily work.

- ☐ Anthropology
- ☐ Archaeology
- ☐ Art and Architectural History
- ☐ Architectural Design
- ☐ Computer Science
- ☐ Conservation
- ☐ Digital Humanities
- ☐ Engineering
- ☐ Museum Studies
- ☐ Paleontology
- ☐ Other ____
- ☐ Other ____
- ☐ Other ____

What is your current position?

- ◯ Technician
- ◯ Cultural minister
- ◯ Undergraduate student
- ◯ Graduate student
- ◯ Independent scholar
- ◯ Assistant professor
- ◯ Associate professor
- ◯ Full professor
- ◯ Other ______________________________

Approximately how many models have you produced in the last 12 months?

____

Approximately how many models have you produced throughout your career?

____

In what year did you first begin to produce models?

____

How would you characterize your abilities to perform the following tasks?

	Learner I am not sure how to do this task.	**Basic** I have done this before, but may need help.	**Proficient** I can perform this task without any assistance.	**Advanced** I could train others to do this.
Create a model by hand (eg in Blender).	○	○	○	○
Create a model using photogrammetry.	○	○	○	○
Create a model using laser or structure light scanning.	○	○	○	○
Add or remove geometry from a model.	○	○	○	○
Modify the display properties of a model (eg material, lighting, etc).	○	○	○	○
Share a 3D model with others.	○	○	○	○
Display a 3D model on the web.	○	○	○	○
Take metrically accurate measurements from a 3D model.	○	○	○	○

How often do you perform the following actions in your research or instruction?

	Regularly At least once per week.	**Frequently** At least once per month.	**Occassionally** At least once per quarter.	**Seldom** At least once per semester.	**Rarely** At least once per year	**Never**
Add or remove geometry from a model.	○	○	○	○	○	○
Modify the display properties of a model (eg material, lighting, etc).	○	○	○	○	○	○
Share models with other scholars.	○	○	○	○	○	○
Share models with the public.	○	○	○	○	○	○
Display models during lecture.	○	○	○	○	○	○
Study a model instead of or in place of the original.	○	○	○	○	○	○
Include a model or figures generated from a model in a publication.	○	○	○	○	○	○
Download a model from the internet.	○	○	○	○	○	○
View a model in a web browser.	○	○	○	○	○	○
View a model offline (eg in Meshlab).	○	○	○	○	○	○

In regard to the actions above, or actions not listed, which are most useful to you in your work? How are they useful?

In regard to the actions above, or actions not listed, which take up the most of your time? How could they be made quicker and easier?

Is there some action that you would like to perform more often? What is it and what prevents you from doing so?

List the software that you utilize when working with models (eg Meshlab, Autodesk, 3DHOP, Sketchfab, etc) and briefly explain how you incorporate each of them into your work.

Please rank the importance of the following features of software for working with models.

	Most Important	**Important**	**Neutral**	**Unimportant**	**Not important at all**
Visualization of uncertainty	○	○	○	○	○
Semantic annotation	○	○	○	○	○
Security (ie prevention of unauthorized access to model viewing and/or download)	○	○	○	○	○
Multiple file format support	○	○	○	○	○
High resolution (over 10 million polygons)	○	○	○	○	○
Measurement (eg point-to-point, surface, or volumetric)	○	○	○	○	○

In regard to the features above, or features not listed, which are particularly useful to you? How are they useful?

What is the top factor that hinders your use of 3D models in your work? How does it hinder you?

In which country do you currently reside?

▼ Afghanistan ... Zimbabwe

In what year were you born?

To what gender do you most identify?

- ○ Male
- ○ Female
- ○ Other ____
- ○ Decline to answer

If you are willing to discuss your use of models in more detail, please enter your email address below.

____

APPENDIX D

SAMPLE INTERVIEW

Introduction

- Goal of the project: To learn more about the day-to-day practices of scholars using 3D models, with specific attention as to what makes 3D models useful and how this usefulness may be furthered

General

- Where do you work? What is your position?
- How many models have you made in the last 12 months, all time? In what year did you begin producing 3d models?

Benefits of 3D Models

- How do you use 3D models in your work?
- What about 3D models makes them useful to you? Workflow
- What technologies do you use when working with 3D models?
    - Could you describe your typical workflow?
    - What are the greatest benefits of these technologies?
- What technologies do you use to view and visualize 3d models?
    - What are the benefits? Drawbacks? What features are excellent? Which are missing?
- How does your use of 3D models relate, if at all, to your use of the physical objects modeled?

Hindrances and Adoption

- What parts of your workflow are particularly time intensive? Do you think they could be made easier?
- What is a primary factor that hinders you in your work with 3D? How does it hinder you?
- How does your use of 3D compare to others you work with?
    - Do you think they could make greater use of 3D? What reasons could possibly be preventing them?

Recap

- Reiterate to subject what was covered and insights gained from session

REFERENCES

[1] Olsen Michael J. et al. "Terrestrial Laser Scanning-Based Structural Damage Assessment". In: *Journal of Computing in Civil Engineering* 24.3 (May 1, 2010), pp. 264–272. DOI: 10.1061/(ASCE)CP.1943-5487.0000028. URL: https://ascelibrary.org/doi/abs/10.1061/(ASCE)CP.1943-5487.0000028 (visited on 04/04/2018).

[2] Arlen F. Chase et al. "Airborne LiDAR, archaeology, and the ancient Maya landscape at Caracol, Belize". In: *Journal of Archaeological Science* 38.2 (Feb. 1, 2011), pp. 387–398. DOI: 10.1016/j.jas.2010.09.018. URL: http://www.sciencedirect.com/science/article/pii/S0305440310003286 (visited on 03/05/2019).

[3] Marco Agus et al. "Data-Driven Analysis of Virtual 3D Exploration of a Large Sculpture Collection in Real-World Museum Exhibitions". In: *J. Comput. Cult. Herit.* 11.1 (Dec. 2017), 2:1–2:20. DOI: 10.1145/3099618. URL: http://doi.acm.org/10.1145/3099618 (visited on 02/23/2018).

[4] M. Doneus et al. "From Deposit to Point Cloud – a Study of Low-Cost Computer Vision Approaches for the Straightforward Documentation of Archaeological Excavations". In: *Geoinformatics FCE CTU* 6.0 (Dec. 21, 2011), pp. 81–88. DOI: 10.14311/gi.6.11. URL: https://ojs.cvut.cz/ojs/index.php/gi/article/view/gi.6.11 (visited on 03/05/2019).

[5] Christopher H. Roosevelt et al. "Excavation is Destruction Digitization: Advances in Archaeological Practice". In: *Journal of Field Archaeology* 40.3 (June 1, 2015), pp. 325–346. DOI: 10.1179/2042458215Y.0000000004. URL: https://doi.org/10.1179/2042458215Y.0000000004 (visited on 03/20/2018).

[6] *Smithsonian Digitization | 3D*. URL: https://3d.si.edu/ (visited on 10/04/2018).

[7] *The Uffizi Digitization Project*. The Uffizi Digitization Project. URL: http://www.digitalsculpture.org/florence/ (visited on 01/08/2019).

[8] *The British Museum on Sketchfab*. Sketchfab. URL: https://sketchfab.com/britishmuseum (visited on 04/13/2018).

[9] *Work Package Deliverables*. Cultural Informatics Research Group. URL: http://www.culturalinformatics.org.uk/work-package-deliverables (visited on 10/17/2017).

[10] David Koller et al. "Protected interactive 3D graphics via remote rendering". In: *ACM Transactions on Graphics (TOG)*. Vol. 23. ACM, 2004, pp. 695–703. ISBN: 0730-0301.

[11] Alexei Vranich. "Reconstructing ancient architecture at Tiwanaku, Bolivia: the potential and promise of 3D printing". In: *Heritage Science* 6.1 (Dec. 13, 2018), p. 65. DOI: 10.1186/s40494-018-0231-0. URL: https://doi.org/10.1186/s40494-018-0231-0 (visited on 02/25/2019).

[12] *The Ancient Aliens - Puma Punku*. TheAncientAliens.com. URL: https://www.theancientaliens.com/puma-punku (visited on 02/26/2019).

[13] Janikorpi. *English: Pile of rocksSuomi: Kivikasa Pumapunkussa*. Aug. 10, 2011. URL: https://commons.wikimedia.org/wiki/File:Puma_Punku_pile_of_rocks.JPG (visited on 04/07/2019).

[14] Brattarb. *English: Puma_Punku*. URL: https://commons.wikimedia.org/wiki/File:7_Puma_Punku.jpg (visited on 04/07/2019).

[15] Barbara J. Mills. "Social Network Analysis in Archaeology". In: *Annual Review of Anthropology* 46.1 (2017), pp. 379–397. DOI: 10.1146/annurev-anthro-102116-041423. URL: https://doi.org/10.1146/annurev-anthro-102116-041423 (visited on 02/27/2019).

[16] Shawn Graham. "Networks, Agent-Based Models and the Antonine Itineraries: Implications for Roman Archaeology". In: *Journal of Mediterranean Archaeology* 19.1 (June 2006), pp. 45–64. DOI: 10.1558/jmea.2006.19.1.45. URL: http://www.equinoxjournals.com/ojs/index.php/JMA/article/view/2436 (visited on 02/27/2019).

[17] *Definition of MODEL*. URL: https://www.merriam-webster.com/dictionary/model (visited on 03/03/2019).

[18] *WordNet | A Lexical Database for English*. URL: https://wordnet.princeton.edu/ (visited on 03/15/2019).

[19] Mary Hesse. “Models and analogies”. In: *A Companion to the Philosophy of Science: Malden, MA, Blackwell Publication* (2000), pp. 299–307.

[20] Mary B. Hesse. “Models and analogies in science”. In: (1965).

[21] Bas C. Van Fraassen. *The scientific image*. Oxford University Press, 1980. ISBN: 0-19-824427-4.

[22] Ronald N. Giere. “How models are used to represent reality”. In: *Philosophy of science* 71.5 (2004), pp. 742–752.

[23] Mary S. Morgan, Margaret Morrison, and Quentin Skinner. *Models as mediators: Perspectives on natural and social science*. Vol. 52. Cambridge University Press, 1999. ISBN: 0-521-65571-4.

[24] Caroline Evans. “The ontology of the fashion model”. In: *AA Files* 63 (2011), pp. 56–69.

[25] Gabriele Guidi. *Il modello fisico come forma di rappresentazione*. Poli. Design, 2005. ISBN: 88-87981-77-9.

[26] Soraya De Chadarevian and Nick Hopwood. *Models: the third dimension of science*. Stanford University Press, 2004. ISBN: 0-8047-3972-2.

[27] Marx W. Wartofsky. “The Model Muddle: Proposals for an Immodest Realism”. In: *Models: Representation and the Scientific Understanding*. Ed. by Marx W. Wartofsky. Boston Studies in the Philosophy of Science. Dordrecht: Springer Netherlands, 1979, pp. 1–11. ISBN: 978-94-009-9357-0. DOI: 10.1007/978-94-009-9357-0_1. URL: https://doi.org/10.1007/978-94-009-9357-0_1 (visited on 03/04/2019).

[28] Richard IG Hughes. “Models and representation”. In: *Philosophy of science* 64 (1997), S325–S336.

[29] Carl F. Craver. "Structures of scientific theories". In: *The Blackwell guide to the philosophy of science* (2002), pp. 55–79.

[30] David Ludwig. "Mediating Objects: Scientific and Public Functions of Models in Nineteenth-Century Biology". In: *History and philosophy of the life sciences* (2013), pp. 139–166.

[31] Nina A. Atanasova. "Validating animal models". In: *THEORIA. Revista de Teoría, Historia y Fundamentos de la Ciencia* 30.2 (2015), pp. 163–181.

[32] Ronald N. Giere. "Using Models to Represent Reality". In: *Model-Based Reasoning in Scientific Discovery*. Ed. by Lorenzo Magnani, Nancy J. Nersessian, and Paul Thagard. Boston, MA: Springer US, 1999, pp. 41–57. ISBN: 978-1-4615-4813-3. DOI: 10.1007/978-1-4615-4813-3_3. URL: https://doi.org/10.1007/978-1-4615-4813-3_3 (visited on 03/04/2019).

[33] Ian Hacking. *Representing and intervening: Introductory topics in the philosophy of natural science*. Cambridge University Press, 1983. ISBN: 0-521-28246-2.

[34] Davis Baird. *Thing knowledge: A philosophy of scientific instruments*. Univ of California Press, 2004. ISBN: 0-520-23249-6.

[35] David Gooding. *Experiment and the Making of Meaning: Human Agency in Scientific Observation*. Springer, Dec. 31, 1990.

[36] Anguel Stefanov. "Theoretical models as representations". In: *Journal for general philosophy of science* 43.1 (2012), pp. 67–76.

[37] Mauro Murzi. *Vienna Circle*. Internet Encyclopedia of Philosophy. URL: https://www.iep.utm.edu/viennacr/ (visited on 03/04/2019).

[38] Thomas S. Kuhn. *The Structure of Scientific Revolutions: 50th Anniversary Edition*. Google-Books-ID: 3eP5Y_OOuzwC. University of Chicago Press, Apr. 18, 2012. 264 pp. ISBN: 978-0-226-45814-4.

[39] Karl Popper. *The Logic of Scientific Discovery*. Routledge, Nov. 4, 2005. ISBN: 978-1-134-47002-0. DOI: 10.4324/9780203994627. URL: https://www.taylorfrancis.com/books/9781134470020 (visited on 03/04/2019).

[40] Tarja Knuuttila. “Models, representation, and mediation”. In: *Philosophy of Science* 72.5 (2005), pp. 1260–1271.

[41] Marcial Echenique. “Models: A discussion”. In: *Architectural Research and Teaching* (1970), pp. 25–30.

[42] Paul Humphreys. *Extending Ourselves: Computational Science, Empiricism, and Scientific Method*. Google-Books-ID: ZIot7QGz7eEC. Oxford University Press, July 22, 2004. 183 pp. ISBN: 978-0-19-803600-5.

[43] Julius Exner. *The Plaster Cast Collection in the Royal Academy of Fine Arts*. 1843. URL: https://commons.wikimedia.org/wiki/File:Julius_Exner_-_The_Plaster_Cast_Collection_in_the_Royal_Academy_of_Fine_Arts_-_KMS3110_-_Statens_Museum_for_Kunst.jpg (visited on 04/07/2019).

[44] Michael P. Mezzatesta. “The Façade of Leone Leoni’s House in Milan, the Casa degli Omenoni: The Artist and the Public”. In: *Journal of the Society of Architectural Historians* 44.3 (1985), pp. 233–249.

[45] Wilhelm Bendz. *Dansk: En billedhugger (Christen Christensen) arbejder efter levende model i sit atelier*. 1827. URL: https://commons.wikimedia.org/wiki/File:Der_Bildhauer_Christen_Christensen.jpg (visited on 04/07/2019).

[46] Adolf H. Borbein. “On the History of the Appraisal and Use of Plaster Casts of Ancient Sculpture (especially in Germany and in Berlin)”. In: *Les moulages de sculptures antiques et l’histoire de l’archéologie. Actes du colloque International Paris*. Vol. 24. 1997, pp. 29–43.

[47] Bernard Frischer. “Three-Dimensional Scanning and Modeling”. In: *The Oxford Handbook of Roman Sculpture* (Mar. 1, 2015). DOI: 10.1093/oxfordhb/9780199921829.013.0006.

URL: http://www.oxfordhandbooks.com/view/10.1093/oxfordhb/9780199921829.001.0001/oxfordhb-9780199921829-e-6 (visited on 03/07/2019).

[48] Marden Fitzpatrick Nichols. "Plaster cast sculpture: A history of touch". In: (), p. 17.

[49] *Soft error*. In: *Wikipedia*. Page Version ID: 884770698. Feb. 23, 2019. URL: https://en.wikipedia.org/w/index.php?title=Soft_error&oldid=884770698 (visited on 03/07/2019).

[50] Valentin Kockel. "Plaster Models and Plaster Casts of Classical Architecture and its Decoration". In: *Plaster Casts*. Ed. by Rune Frederiksen and Eckart Marchand. Berlin, New York: De Gruyter, Jan. 17, 2010. ISBN: 978-3-11-021687-5. DOI: 10.1515/9783110216875.417. URL: https://www.degruyter.com/view/books/9783110216875/9783110216875.417/9783110216875.417.xml (visited on 03/07/2019).

[51] Rolf Janke. *Architectural models/Architekturmodelle.* Tech. rep. Academy Editions, 1978.

[52] Martin S. Briggs. "Architectural Models-I". In: *The Burlington Magazine for Connoisseurs* 54.313 (1929), pp. 174–183.

[53] Martin S. Briggs. "Architectural Models-II". In: *The Burlington Magazine for Connoisseurs* 54.314 (1929), pp. 245–252.

[54] George A. Rothrock. "The Musee des Plans-Reliefs". In: *French Historical Studies* 6.2 (1969), pp. 253–256.

[55] T. Yasuda et al. "3D visualization of an ancient Egyptian mummy". In: *IEEE Computer Graphics and Applications* 12.3 (May 1992), pp. 13–17. DOI: 10.1109/38.135909.

[56] Matthew Pendlebury. "3d virtual reality reconstruction on the internet using vrml". PhD thesis. Verlag nicht ermittelbar, 1996.

[57] R. Pedrazzani et al. "Study of sulphation of Candoglia marble by means of micro X-ray diffraction experiments". In: *Applied Physics A* 83.4 (June 2006), pp. 689–694. DOI: 10.1007/s00339-006-3539-7. URL: http://link.springer.com/10.1007/s00339-006-3539-7 (visited on 03/06/2019).

[58] Kourosh Khoshelham and Sander Oude Elberink. “Accuracy and resolution of kinect depth data for indoor mapping applications”. In: *Sensors* 12.2 (2012), pp. 1437–1454.

[59] *Professional 3D Scanners | Artec 3D | Best 3D Scanning Solutions*. URL: https://www.artec3d.com/ (visited on 03/07/2019).

[60] *Laser Scanners*. URL: https://leica-geosystems.com/en-US/products/laser-scanners (visited on 03/07/2019).

[61] The Center for Photogrammetric Training. *History of Photogrammetry*. URL: http://ibis.geog.ubc.ca/courses/geob373/lectures/Handouts/History_of_Photogrammetry.pdf (visited on 03/06/2019).

[62] *Agisoft Metashape 1.5.0 pre-release*. URL: https://www.agisoft.com/forum/index.php?topic=9793.0 (visited on 03/06/2019).

[63] *Home - CapturingReality.com*. URL: https://www.capturingreality.com/ (visited on 03/06/2019).

[64] L. M. Galantucci, M. Pesce, and F. Lavecchia. “A stereo photogrammetry scanning methodology, for precise and accurate 3D digitization of small parts with sub-millimeter sized features”. In: *CIRP Annals* 64.1 (Jan. 1, 2015), pp. 507–510. DOI: 10.1016/j.cirp.2015.04.016. URL: http://www.sciencedirect.com/science/article/pii/S0007850615000244 (visited on 03/07/2019).

[65] G. Guidi et al. “Fusion of range camera and photogrammetry: a systematic procedure for improving 3-D models metric accuracy”. In: *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics: a publication of the IEEE Systems, Man, and Cybernetics Society* 33.4 (2003), pp. 667–676. DOI: 10.1109/TSMCB.2003.814282.

[66] Blender Foundation. *blender.org - Home of the Blender project - Free and Open 3D Creation Software*. blender.org. URL: https://www.blender.org/ (visited on 03/07/2019).

[67] *3D File Formats*. Sketchfab Help Center. URL: http://help.sketchfab.com/hc/en-us/articles/202508396-3D-File-Formats (visited on 04/27/2018).

[68] *Pixologic : ZBrush - The all-in-one-digital sculpting solution*. Pixologic - Home of ZBrush. URL: http://pixologic.com (visited on 03/07/2019).

[69] *3ds Max | 3D Modeling, Animation & Rendering Software | Autodesk*. Autodesk. URL: https://www.autodesk.com/products/3ds-max/overview (visited on 03/07/2019).

[70] Sarah Mason. *Oxford LibGuides: Introduction to Digital Preservation: Emulation*. URL: //ox.libguides.com/digitalpreservation/emulation (visited on 03/08/2019).

[71] Kenton McHenry and Peter Bajcsy. "An overview of 3d data content, file formats and viewers". In: *National Center for Supercomputing Applications* 1205 (2008), p. 22.

[72] *Object File Format (.off)*. URL: http://segeval.cs.princeton.edu/public/off_format.html (visited on 03/10/2019).

[73] *B1. Object Files (.obj)*. URL: http://www.martinreddy.net/gfx/3d/OBJ.spec (visited on 03/10/2019).

[74] *MTL OBJ materials file*. URL: http://paulbourke.net/dataformats/mtl/ (visited on 03/10/2019).

[75] *glTF - Runtime 3D Asset Delivery*. The Khronos Group. Apr. 3, 2013. URL: https://www.khronos.org/gltf/ (visited on 03/10/2019).

[76] *COLLADA - 3D Asset Exchange Schema*. The Khronos Group. July 19, 2011. URL: https://www.khronos.org/collada/ (visited on 03/11/2019).

[77] H. Maschner et al. "The Virtual Repository of Arctic Archaeology and the Democratization of Science". In: *2015 Digital Heritage*. 2015 Digital Heritage. Vol. 2. Sept. 2015, pp. 573–580. DOI: 10.1109/DigitalHeritage.2015.7419575.

[78] David Koller, Bernard Frischer, and Greg Humphreys. "Research Challenges for Digital Archives of 3D Cultural Heritage Models". In: *J. Comput. Cult. Herit.* 2.3 (Jan. 2010), 7:1–7:17. DOI: 10.1145/1658346.1658347. URL: http://doi.acm.org/10.1145/1658346.1658347 (visited on 04/06/2018).

[79] David Raggett. *Extending WWW to support Platform Independent Virtual Reality*. W3. URL: https://www.w3.org/People/Raggett/vrml/vrml.html (visited on 04/27/2018).

[80] *VRML 1.0C Specification*. URL: http://paulbourke.net/dataformats/vrml1/#History (visited on 03/22/2018).

[81] *VRML 2.0, Part 1: Introduction*. URL: http://gun.teipir.gr/VRML-amgem/spec/part1/introduction.html (visited on 04/27/2018).

[82] *VRML 2.0 Specification*. URL: http://gun.teipir.gr/VRML-amgem/spec/index.html (visited on 03/14/2018).

[83] *Web Virtual Reality and 3D - in VRML or XML? - The Web Developer's Journal*. Aug. 16, 2000. URL: https://web.archive.org/web/20000816124309/http:/webdevelopersjournal.com:80/articles/virtual_reality.html (visited on 04/27/2018).

[84] *Khronos Releases Final WebGL 1.0 Specification*. The Khronos Group. Mar. 3, 2011. URL: https://www.khronos.org/news/press/khronos-releases-final-webgl-1.0-specification (visited on 03/16/2018).

[85] S. Ortiz. "Is 3D Finally Ready for the Web?" In: *Computer* 43.1 (Jan. 2010), pp. 14–16. DOI: 10.1109/MC.2010.15.

[86] Alun Evans et al. "3D graphics on the web: A survey". In: *Computers & Graphics* 41 (June 1, 2014), pp. 43–61. DOI: 10.1016/j.cag.2014.02.002. URL: http://www.sciencedirect.com/science/article/pii/S0097849314000260 (visited on 02/23/2018).

[87] Mark Pesce. *iOS 8 release: WebGL now runs everywhere. Hurrah for 3D graphics!* The Register. Sept. 17, 2014. URL: https://www.theregister.co.uk/2014/09/17/after_20_years_apple_finally_enters_the_third_dimension/ (visited on 05/01/2018).

[88] *Can I use... Support tables for HTML5, CSS3, etc*. URL: https://caniuse.com/#feat=webgl (visited on 04/27/2018).

[89] *WebVR - Bringing Virtual Reality to the Web*. URL: https://webvr.info/ (visited on 04/27/2018).

[90] *360 Video Channel - NYTimes.com*. URL: https://www.nytimes.com/video/360-video (visited on 04/27/2018).

[91] *Rewind 2017*. Rewind 2017. URL: https://rewind2017.withyoutube.com/ (visited on 04/27/2018).

[92] Tom May. *9 brilliant uses of 3D in web design*. Creative Bloq. URL: https://www.creativebloq.com/inspiration/9-brilliant-uses-of-3d-in-web-design (visited on 04/27/2018).

[93] *Play-Doh presents: the Gallery of Emerging Species*. Play-Doh presents: the Gallery of Emerging Species. URL: http://playdoh-lagaleriedesespeces.com/en/ (visited on 04/27/2018).

[94] *Sketchfab Passes One Million Members*. Sketchfab. July 10, 2017. URL: https://blog.sketchfab.com/sketchfab-passes-one-million-members/ (visited on 04/27/2018).

[95] So-Yeon Yoon, James Laffey, and Hyunjoo Oh. "Understanding Usability and User Experience of Web-Based 3D Graphics Technology". In: *International Journal of Human–Computer Interaction* 24.3 (Mar. 19, 2008), pp. 288–306. DOI: 10.1080/10447310801920516. URL: https://doi.org/10.1080/10447310801920516 (visited on 03/01/2018).

[96] Romanowska. "So You Think You Can Model? A Guide to Building and Evaluating Archaeological Simulation Models of Dispersals". In: *Human Biology* 87.3 (2015), p. 169. DOI: 10.13110/humanbiology.87.3.0169. URL: http://www.jstor.org/stable/10.13110/humanbiology.87.3.0169 (visited on 02/25/2019).

[97] KIRSTEN Risden et al. "An initial examination of ease of use for 2D and 3D information visualizations of web content". In: *International Journal of Human-Computer Studies* 53.5 (Nov. 1, 2000), pp. 695–714. DOI: 10.1006/ijhc.2000.0413. URL: http://www.sciencedirect.com/science/article/pii/S1071581900904130 (visited on 03/05/2019).

[98] *Web3D Conferences | Web3D Consortium*. URL: http://www.web3d.org/conferences (visited on 03/05/2019).

[99] iw3c2admin. *IW3C2 - Past and Future Conferences*. URL: http://www.iw3c2.org/conferences/ (visited on 03/05/2019).

[100] Heather Richards-Rissetto and Jennifer von Schwerin. "A catch 22 of 3D data sustainability: Lessons in 3D archaeological data management & accessibility". In: *Digital Applications in Archaeology and Cultural Heritage*. The Past now showing in 3D: case studies in 3D archaeology 6 (Sept. 1, 2017), pp. 38–48. DOI: 10.1016/j.daach.2017.04.005. URL: http://www.sciencedirect.com/science/article/pii/S2212054816300273 (visited on 02/23/2018).

[101] Marc Levoy et al. "The Digital Michelangelo Project: 3D Scanning of Large Statues". In: *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '00. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 2000, pp. 131–144. ISBN: 978-1-58113-208-3. DOI: 10.1145/344779.344849. URL: http://dx.doi.org/10.1145/344779.344849 (visited on 03/22/2018).

[102] Ename Charter. *The ICOMOS Charter for the Interpretation and Presentation of Cultural Heritage Sites*. 2008.

[103] Hugh Denard. "The London Charter for the computer-based visualisation of cultural heritage". In: *no. February* (2009), pp. 1–13.

[104] Neil A. Silberman. "ICOMOS charter for the interpretation and presentation of cultural heritage sites". In: (2008).

[105] Hugh Denard. "A new introduction to the London Charter". In: *Paradata and transparency in virtual heritage*. Routledge, 2016, pp. 83–98.

[106] Fabrizio I. Apollonio. "Classification Schemes for Visualization of Uncertainty in Digital Hypothetical Reconstruction". In: *3D Research Challenges in Cultural Heritage II*. Lecture Notes in Computer Science. Springer, Cham, 2016, pp. 173–197. ISBN: 978-3-319-47646-9

978-3-319-47647-6. DOI: 10.1007/978-3-319-47647-6_9. URL: https://link.springer.com/chapter/10.1007/978-3-319-47647-6_9 (visited on 04/02/2018).

[107] *The Digital Hadrian's Villa Project*. URL: http://vwhl.soic.indiana.edu/villa/ (visited on 04/28/2018).

[108] *Maya Arch 3D | Virtual Tour*. URL: http://3dom.fbk.eu/repository/files/vt_copan/index.html (visited on 04/28/2018).

[109] Neil G. Smith et al. "Pottery Informatics Query Database: A New Method for Mathematic and Quantitative Analyses of Large Regional Ceramic Datasets". In: *Journal of Archaeological Method and Theory* 21.1 (Mar. 1, 2014), pp. 212–250. DOI: 10.1007/s10816-012-9148-1. URL: https://link.springer.com/article/10.1007/s10816-012-9148-1 (visited on 04/04/2018).

[110] Karina Rodriguez Echavarria, Dean Few, and Ran Song. "Web-based Exploration of Semantically Rich 3D Decorative Ornament". In: (2016), pp. 1–4.

[111] P. Arias et al. "Control of structural problems in cultural heritage monuments using close-range photogrammetry and computer methods". In: *Computers & Structures* 83.21 (Aug. 1, 2005), pp. 1754–1766. DOI: 10.1016/j.compstruc.2005.02.018. URL: http://www.sciencedirect.com/science/article/pii/S0045794905001094 (visited on 04/04/2018).

[112] B. Riveiro et al. "Photogrammetric 3D modelling and mechanical analysis of masonry arches: An approach based on a discontinuous model of voussoirs". In: *Automation in Construction* 20.4 (July 1, 2011), pp. 380–388. DOI: 10.1016/j.autcon.2010.11.008. URL: http://www.sciencedirect.com/science/article/pii/S0926580510001871 (visited on 04/12/2018).

[113] Silvia Biasotti et al. "3D artifacts similarity based on the concurrent evaluation of heterogeneous properties". In: *Journal on Computing and Cultural Heritage (JOCCH)* 8.4 (2015), p. 19.

[114] Marco Potenziani et al. "3DHOP: 3D Heritage Online Presenter". In: *Computer & Graphics* 52 (Nov. 2015), pp. 129–141. URL: http://vcg.isti.cnr.it/Publications/2015/PCDCPS15.

[115] *Tumblr*. Sketchfab Help Center. URL: http://help.sketchfab.com/hc/en-us/articles/207685976-Tumblr (visited on 04/28/2018).

[116] *Facebook now supports Sketchfab 3D embeds!* Sketchfab Blog. Feb. 26, 2015. URL: https://blog.sketchfab.com/facebook-now-supports-sketchfab-3d-embeds/ (visited on 04/28/2018).

[117] *Over 500 Cultural Institutions have joined Sketchfab - Sketchfab Blog*. URL: https://blog.sketchfab.com/500-cultural-institutions-joined-sketchfab/?utm_source=newsletter&utm_medium=email&utm_content=2017-04-500-museums&utm_campaign=newsletter (visited on 04/28/2018).

[118] *Unity - Unity - Multiplatform*. Unity. URL: https://unity3d.com/unity/features/multiplatform (visited on 04/16/2018).

[119] *Unity - Community*. Unity. URL: https://unity3d.com/community (visited on 04/16/2018).

[120] T. P. Kersten, F. Tschirschwitz, and S. Deggim. "DEVELOPMENT OF A VIRTUAL MUSEUM INCLUDING A 4D PRESENTATION OF BUILDING HISTORY IN VIRTUAL REALITY". In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLII-2-W3 (Feb. 23, 2017), pp. 361–367. DOI: https://doi.org/10.5194/isprs-archives-XLII-2-W3-361-2017. URL: https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-2-W3/361/2017/ (visited on 04/16/2018).

[121] Ferimar R. Priolo et al. "Virtual Heritage Tour: A 3D Interactive Virtual Tour Musealisation Application". In: *Proceedings of the 3rd International Conference on Communication and Information Processing*. ICCIP '17. New York, NY, USA: ACM, 2017, pp. 190–195. ISBN: 978-1-4503-5365-6. DOI: 10.1145/3162957.3162987. URL: http://doi.acm.org/10.1145/3162957.3162987 (visited on 04/16/2018).

[122] S. Kazanis et al. "DEVELOPING A VIRTUAL MUSEUM FOR THE ANCIENT WINE TRADE IN EASTERN MEDITERRANEAN". In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLII-2/W5 (Aug. 18,

2017), pp. 399–405. DOI: 10.5194/isprs-archives-XLII-2-W5-399-2017. URL: https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-2-W5/399/2017/ (visited on 04/16/2018).

[123] E. Pietroni, M. Forlani, and C. Rufa. "Livia's Villa Reloaded: An example of re-use and update of a pre-existing Virtual Museum, following a novel approach in storytelling inside virtual reality environments". In: *2015 Digital Heritage*. 2015 Digital Heritage. Vol. 2. Sept. 2015, pp. 511–518. DOI: 10.1109/DigitalHeritage.2015.7419567.

[124] Guy Peter Schofield et al. *Viking VR: Designing a Virtual Reality Experience for a Museum*. ACM DIS Conference on Designing Interactive Systems. 2018. URL: http://eprints.whiterose.ac.uk/129158/ (visited on 04/16/2018).

[125] C. Kiourt et al. "A dynamic web-based 3D virtual museum framework based on open data". In: *2015 Digital Heritage*. 2015 Digital Heritage. Vol. 2. Sept. 2015, pp. 647–650. DOI: 10.1109/DigitalHeritage.2015.7419589.

[126] Chairi Kiourt, Anestis Koutsoudis, and George Pavlidis. "DynaMus: A fully dynamic 3D virtual museum framework". In: *Journal of Cultural Heritage* 22 (Nov. 1, 2016), pp. 984–991. DOI: 10.1016/j.culher.2016.06.007. URL: http://www.sciencedirect.com/science/article/pii/S1296207416301029 (visited on 04/16/2018).

[127] B. A. Wardijono, F. Hendajani, and S. A. Sudiro. "Virtual Reality Website of Indonesia National Monument and Its Environment". In: *Journal of Physics: Conference Series* 812.1 (2017), p. 012035. DOI: 10.1088/1742-6596/812/1/012035. URL: http://stacks.iop.org/1742-6596/812/i=1/a=012035 (visited on 04/16/2018).

[128] Bernard Frischer et al. "New Light on the Relationship between the Montecitorio Obelisk and Ara Pacis of Augustus". In: *Studies in Digital Heritage* 1.1 (May 24, 2017), pp. 18–119. DOI: 10.14434/sdh.v1i1.23331. URL: https://scholarworks.iu.edu/journals/index.php/sdh/article/view/23331 (visited on 04/16/2018).

[129] *(11) Everything you need to know about Unity WebGL Building | LinkedIn*. URL: https://www.linkedin.com/pulse/everything-you-need-know-unity-webgl-building-hulusi-onder/ (visited on 04/16/2018).

[130] *x3dom: Instant 3D the HTML way!* URL: https://www.x3dom.org/ (visited on 04/16/2018).

[131] Limper M. et al. "The POP Buffer: Rapid Progressive Clustering by Geometry Quantization". In: *Computer Graphics Forum* 32.7 (Nov. 25, 2013), pp. 197–206. DOI: 10.1111/cgf.12227. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12227 (visited on 04/16/2018).

[132] Y. Jung, J. Behr, and H. Graf. "X3DOM AS CARRIER OF THE VIRTUAL HERITAGE". In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XXXVIII-5/W16 (Sept. 10, 2012), pp. 475–482. DOI: 10.5194/isprsarchives-XXXVIII-5-W16-475-2011. URL: http://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XXXVIII-5-W16/475/2011/ (visited on 04/16/2018).

[133] Katarzyna Wilkosinska et al. "Towards a Common Implementation Framework for Online Virtual Museums". In: *Design, User Experience, and Usability. Health, Learning, Playing, Cultural, and Cross-Cultural User Experience*. International Conference of Design, User Experience, and Usability. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, July 21, 2013, pp. 321–330. ISBN: 978-3-642-39240-5 978-3-642-39241-2. DOI: 10.1007/978-3-642-39241-2_36. URL: https://link.springer.com/chapter/10.1007/978-3-642-39241-2_36 (visited on 03/18/2018).

[134] *Archaeology Data Service*. URL: http://archaeologydataservice.ac.uk/research/3DViewerObject.xhtml (visited on 04/16/2018).

[135] Federico Ponchio et al. "ARIADNE Visual Media Service: Easy Web Publishing of Advanced Visual Media". In: *CAA2015*. Vol. 1. 2016, pp. 433–442. URL: http://vcg.isti.cnr.it/Publications/2016/PPDCS16.

[136] Paolo Cignoni et al. "Batched multi triangulation". In: *Visualization, 2005. VIS 05. IEEE*. IEEE, 2005, pp. 207–214. ISBN: 0-7803-9462-3.

[137] Francesco Gabellone. “Integrated Technologies for Museum Communication and Interactive Apps in the PON DiCet Project”. In: *Augmented and Virtual Reality*. International Conference on Augmented and Virtual Reality. Lecture Notes in Computer Science. Springer, Cham, Aug. 31, 2015, pp. 3–16. ISBN: 978-3-319-22887-7 978-3-319-22888-4. DOI: 10.1007/978-3-319-22888-4_1. URL: https://link.springer.com/chapter/10.1007/978-3-319-22888-4_1 (visited on 04/16/2018).

[138] Eva Savina Malinverni et al. “Web Tool as a Virtual Museum of Ancient Archaeological Ruins in Peru”. In: *Augmented Reality, Virtual Reality, and Computer Graphics*. International Conference on Augmented Reality, Virtual Reality and Computer Graphics. Lecture Notes in Computer Science. Springer, Cham, June 12, 2017, pp. 282–296. ISBN: 978-3-319-60927-0 978-3-319-60928-7. DOI: 10.1007/978-3-319-60928-7_25. URL: https://link.springer.com/chapter/10.1007/978-3-319-60928-7_25 (visited on 04/16/2018).

[139] S. Rizvic, V. Okanovic, and A. Sadzak. “Visualization and multimedia presentation of cultural heritage”. In: *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). May 2015, pp. 348–351. DOI: 10.1109/MIPRO.2015.7160292.

[140] Denis Fisseler, Gerfrid G. W. Müller, and Frank Weichert. “Web-Based Scientific Exploration and Analysis of 3D Scanned Cuneiform Datasets for Collaborative Research”. In: *Informatics* 4.4 (Dec. 12, 2017), p. 44. DOI: 10.3390/informatics4040044. URL: http://www.mdpi.com/2227-9709/4/4/44 (visited on 02/23/2018).

[141] Roberto Pierdicca et al. “3D visualization tools to explore ancient architectures in South America”. In: *Virtual Archaeology Review* 7.15 (Nov. 15, 2016), pp. 44–53. DOI: 10.4995/var.2016.5904. URL: https://polipapers.upv.es/index.php/var/article/view/5904 (visited on 03/18/2018).

[142] *Asset Store*. URL: https://assetstore.unity.com/ (visited on 04/28/2018).

[143] *SteamVR Plugin - Asset Store*. URL: https://assetstore.unity.com/packages/templates/systems/steamvr-plugin-32647 (visited on 04/28/2018).

[144] *Post Processing Stack - Asset Store*. URL: https://assetstore.unity.com/packages/essentials/post-processing-stack-83912 (visited on 04/28/2018).

[145] *Unity Particle Pack - Asset Store*. URL: https://assetstore.unity.com/packages/essentials/asset-packs/unity-particle-pack-73777 (visited on 04/28/2018).

[146] *Measure This!* Sketchfab Blog. Jan. 20, 2017. URL: https://blog.sketchfab.com/measure-this/ (visited on 04/28/2018).

[147] *Viewer API - Introduction*. Sketchfab. URL: https://sketchfab.com/developers/viewer (visited on 03/07/2019).

[148] *Data API v3 - Developers*. Sketchfab. URL: https://sketchfab.com/developers/data-api/v3 (visited on 03/07/2019).

[149] David Koller. "Protected Sharing of 3D Models of Cultural Heritage and Archaeological Artifacts". In: *Proceedings of the 36th CAA Conference*. Vol. 2. 2008, pp. 326–331.

[150] *Shannon's Maxim*. URL: https://lti.umuc.edu/contentadaptor/topics/byid/aac12aff-bb83-48cc-8ce3-7788f8f60fc0 (visited on 03/16/2019).

[151] Cai Su et al. "Protected-3DMPS: remote-rendering based 3D model publishing system in digital museum". In: *J. Comput. Inf. Syst* 2 (2006), pp. 277–283.

[152] Jiajun Zhu et al. "Quantifying usability in secure graphics: assessing the user costs of protecting 3D content". In: *Proceedings of the 5th symposium on Applied perception in graphics and visualization*. ACM, 2008, pp. 91–96. ISBN: 1-59593-981-4.

[153] Chikai Ohazama. "Opengl vizserver". In: *White Paper, Silicon Graphics Inc* (1999).

[154] Shu Shi and Cheng-Hsin Hsu. "A Survey of Interactive Remote Rendering Systems". In: *ACM Comput. Surv.* 47.4 (May 2015), 57:1–57:29. DOI: 10.1145/2719921. URL: http://doi.acm.org/10.1145/2719921 (visited on 02/23/2018).

[155] *The #1 Choice in Remote Desktop Access and Support | TeamViewer*. URL: https://www.teamviewer.us/ (visited on 04/16/2018).

[156] *Connect to another computer using Remote Desktop Connection - Windows Help*. Aug. 31, 2016. URL: https://support.microsoft.com/en-us/help/17463/windows-7-connect-to-another-computer-remote-desktop-connection (visited on 04/16/2018).

[157] Thomas A. Funkhouser. "RING: A Client-server System for Multi-user Virtual Environments". In: *Proceedings of the 1995 Symposium on Interactive 3D Graphics*. I3D '95. New York, NY, USA: ACM, 1995, 85–ff. ISBN: 978-0-89791-736-0. DOI: 10.1145/199404.199418. URL: http://doi.acm.org/10.1145/199404.199418 (visited on 03/21/2018).

[158] P. S. McCormick, Ji Qiang, and R. D. Ryne. "Visualizing high-resolution accelerator physics". In: *IEEE Computer Graphics and Applications* 19.5 (Sept. 1999), pp. 11–13. DOI: 10.1109/38.788792.

[159] Brian Blau et al. "Networked Virtual Environments". In: *Proceedings of the 1992 Symposium on Interactive 3D Graphics*. I3D '92. New York, NY, USA: ACM, 1992, pp. 157–160. ISBN: 978-0-89791-467-3. DOI: 10.1145/147156.147187. URL: http://doi.acm.org/10.1145/147156.147187 (visited on 03/21/2018).

[160] *PlayStation Now – PS Now Subscription for 600+ PS4 & PS3 Games*. URL: https://www.playstation.com/en-us/explore/playstationnow/ (visited on 04/16/2018).

[161] Dieter Schmalstieg and Michael Gervautz. "Demand-Driven Geometry Transmission for Distributed Virtual Environments". In: *Computer Graphics Forum*. 1996, pp. 421–433.

[162] Ioana M. Martin. *Adaptive rendering of 3D models over networks using multiple modalities*. IBM TJ Watson Research Center, 2000.

[163] Jean-Charles Quillet et al. "Using expressive rendering for remote visualization of large city models". In: ACM Press, 2006, p. 27. ISBN: 978-1-59593-336-2. DOI: 10.1145/1122591.1122595. URL: http://portal.acm.org/citation.cfm?doid=1122591.1122595 (visited on 03/22/2018).

[164] Ilmi Yoon and Ulrich Neumann. “Web-Based Remote Rendering with IBRAC (Image-Based Rendering Acceleration and Compression)”. In: *Computer Graphics Forum*. Vol. 19. Wiley Online Library, 2000, pp. 321–330. ISBN: 1467-8659.

[165] Florent Duguet and George Drettakis. “Flexible point-based rendering on mobile devices”. In: *IEEE Computer Graphics and Applications* 24.4 (2004), pp. 57–63.

[166] *SpaceMouse Enterprise*. 3Dconnexion. URL: https://www.3dconnexion.com/products/spacemouse/spacemouse-enterprise.html/?source=box (visited on 03/16/2019).

[167] W. D. Li et al. “Feature-based design in a distributed and collaborative environment”. In: *Computer-Aided Design*. Distributed CAD for supporting internet collaborative design 36.9 (Aug. 1, 2004), pp. 775–797. DOI: 10.1016/j.cad.2003.09.005. URL: http://www.sciencedirect.com/science/article/pii/S0010448503001817 (visited on 03/01/2018).

[168] Phil Karlton and Paula Womack. *OpenGL Graphics with the X Window System*. Ver, 2005.

[169] Peter Eisert and Philipp Fechteler. “REMOTE RENDERING OF COMPUTER GAMES”. In: (), p. 6.

[170] Zeineb Abderrahim and Mohamed Salim Bouhlel. “Scalable and Adaptive Streaming of 3D Mesh to Heterogeneous Devices”. In: *3D Research* 7.4 (Dec. 1, 2016), p. 32. DOI: 10.1007/s13319-016-0108-5. URL: https://link.springer.com/article/10.1007/s13319-016-0108-5 (visited on 02/23/2018).

[171] P. Mindek et al. “Visualization Multi-Pipeline for Communicating Biology”. In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (Jan. 2018), pp. 883–892. DOI: 10.1109/TVCG.2017.2744518.

[172] Thiago F. de Moraes et al. “Out-of-Core Progressive Web-Based Rendering of Triangle Meshes”. In: *VipIMAGE 2017*. European Congress on Computational Methods in Applied Sciences and Engineering. Lecture Notes in Computational Vision and Biomechanics. Springer, Cham, Oct. 18, 2017, pp. 456–466. ISBN: 978-3-319-68194-8 978-3-319-68195-5.

DOI: 10.1007/978-3-319-68195-5_50. URL: https://link.springer.com/chapter/10.1007/978-3-319-68195-5_50 (visited on 03/19/2018).

[173] Federico Spini et al. "Web 3D Indoor Authoring and VR Exploration via Texture Baking Service". In: *Proceedings of the 21st International Conference on Web3D Technology*. Web3D '16. New York, NY, USA: ACM, 2016, pp. 151–154. ISBN: 978-1-4503-4428-9. DOI: 10.1145/2945292.2945309. URL: http://doi.acm.org/10.1145/2945292.2945309 (visited on 02/23/2018).

[174] Mohammad Raji, Alok Hota, and Jian Huang. "Scalable web-embedded volume rendering". In: IEEE, Oct. 2017, pp. 45–54. ISBN: 978-1-5386-0617-9. DOI: 10.1109/LDAV.2017.8231850. URL: http://ieeexplore.ieee.org/document/8231850/ (visited on 03/19/2018).

[175] Zeqi Lai et al. "Furion: Engineering High-Quality Immersive Virtual Reality on Today's Mobile Devices". In: *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. MobiCom '17. New York, NY, USA: ACM, 2017, pp. 409–421. ISBN: 978-1-4503-4916-1. DOI: 10.1145/3117811.3117815. URL: http://doi.acm.org/10.1145/3117811.3117815 (visited on 02/23/2018).

[176] Yu-Jung Chen, Chung-Yao Hung, and Shao-Yi Chien. "Distributed rendering: Interaction delay reduction in remote rendering with client-end GPU-accelerated scene warping technique". In: *2017 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*. 2017 IEEE International Conference on Multimedia Expo Workshops (ICMEW). July 2017, pp. 67–72. DOI: 10.1109/ICMEW.2017.8026271.

[177] P. Du, Y. Song, and L. Deng. "A real-time collaborative framework for 3D design based on HTML5". In: *2016 IEEE 20th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. 2016 IEEE 20th International Conference on Computer Supported Cooperative Work in Design (CSCWD). May 2016, pp. 215–220. DOI: 10.1109/CSCWD.2016.7565991.

[178] Alun Evans, Javi Agenjo, and Josep Blat. “A pipeline for the creation of progressively rendered web 3D scenes”. In: *Multimedia Tools and Applications* (Dec. 8, 2017), pp. 1–29. DOI: 10.1007/s11042-017-5463-6. URL: https://link.springer.com/article/10.1007/s11042-017-5463-6 (visited on 02/23/2018).

[179] Kevin Boos, David Chu, and Eduardo Cuervo. “FlashBack: Immersive Virtual Reality on Mobile Devices via Rendering Memoization”. In: *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*. MobiSys ’16. New York, NY, USA: ACM, 2016, pp. 291–304. ISBN: 978-1-4503-4269-8. DOI: 10.1145/2906388.2906418. URL: http://doi.acm.org/10.1145/2906388.2906418 (visited on 03/27/2018).

[180] Michael Krone, Christoph Müller, and Thomas Ertl. “Remote Rendering and User Interaction on Mobile Devices for Scientific Visualization”. In: *Proceedings of the 8th International Symposium on Visual Information Communication and Interaction*. VINCI ’15. New York, NY, USA: ACM, 2015, pp. 21–26. ISBN: 978-1-4503-3482-2. DOI: 10.1145/2801040.2801057. URL: http://doi.acm.org/10.1145/2801040.2801057 (visited on 03/24/2018).

[181] *Web | ParaView*. URL: https://www.paraview.org/web/ (visited on 03/13/2019).

[182] Mr.doob. *JavaScript 3D library. Contribute to mrdoob/three.js development by creating an account on GitHub*. original-date: 2010-03-23T18:58:01Z. Mar. 11, 2019. URL: https://github.com/mrdoob/three.js (visited on 03/11/2019).

[183] *websocket.org - Powered by Kaazing*. URL: https://www.websocket.org/ (visited on 04/20/2018).

[184] *Web VR Explorer*. URL: http://cvdlab.github.io/bak3d-demo/ (visited on 03/13/2019).

[185] Leif Christiansen. *leifchri/Seymour*. GitHub Enterprise. URL: https://github.iu.edu/leifchri/Seymour (visited on 03/16/2019).

[186] *First public version. Still a lot to do · mrdoob/three.js@a90c4e1*. GitHub. URL: https://github.com/mrdoob/three.js/commit/a90c4e107ff6e3b148458c96965e876f9441b147 (visited on 03/11/2019).

[187] Tony Parisi. *WebGL: up and running*. " O'Reilly Media, Inc.", 2012. ISBN: 1-4493-2357-X.

[188] *Interactive 3D Graphics | Udacity*. URL: https://www.udacity.com/course/interactive-3d-graphics--cs291 (visited on 03/11/2019).

[189] *Newest 'three.js' Questions*. Stack Overflow. URL: https://stackoverflow.com/questions/tagged/three.js (visited on 03/11/2019).

[190] *Connect - freenode Web IRC*. URL: http://webchat.freenode.net/?channels=three.js (visited on 03/11/2019).

[191] Odin et al. *Heraclos, and the quest he never ever asked for*. URL: http://heraclosgame.com (visited on 03/11/2019).

[192] *WebVR Homepage • WITHIN*. WebVR Homepage • WITHIN. URL: https://vr.with.in/ (visited on 03/11/2019).

[193] *Swiss Army Man - Official Movie Site*. URL: http://swissarmyman.com/ (visited on 03/11/2019).

[194] *Hammer.JS - Hammer.js*. URL: https://hammerjs.github.io/ (visited on 03/11/2019).

[195] Colin Ware. *Information visualization: perception for design*. Elsevier, 2012. ISBN: 0-12-381465-0.

[196] Michael Chen, S. Joy Mountford, and Abigail Sellen. "A Study in Interactive 3-D Rotation Using 2-D Control Devices". In: *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '88. New York, NY, USA: ACM, 1988, pp. 121–129. ISBN: 978-0-89791-275-4. DOI: 10.1145/54852.378497. URL: http://doi.acm.org/10.1145/54852.378497 (visited on 03/30/2018).

[197] Andrew J. Hanson. "The rolling ball". In: *Graphics Gems III (IBM Version)*. Elsevier, 1992, pp. 51–60.

[198] *Welcome to NGINX Wiki! | NGINX*. URL: https://www.nginx.com/resources/wiki/ (visited on 03/12/2019).

[199] *nginx*. URL: https://nginx.org/en/ (visited on 03/12/2019).

[200] *February 2019 Web Server Survey | Netcraft*. URL: https://news.netcraft.com/archives/2019/02/28/february-2019-web-server-survey.html (visited on 03/12/2019).

[201] fbbdev. *FastCGI Specification*. URL: https://fast-cgi.github.io/spec (visited on 03/12/2019).

[202] *r/linuxadmin - what happened to fastcgi.com?* reddit. URL: https://www.reddit.com/r/linuxadmin/comments/4bywjn/what_happened_to_fastcgicom/ (visited on 03/12/2019).

[203] *mod_fcgid - FastCGI interface module for Apache 2 - The Apache HTTP Server Project*. URL: https://httpd.apache.org/mod_fcgid/ (visited on 03/12/2019).

[204] *Module ngx_http_fastcgi_module*. URL: http://nginx.org/en/docs/http/ngx_http_fastcgi_module.html (visited on 03/12/2019).

[205] fbbdev. *Unofficial FastCGI Specification website*. URL: https://fast-cgi.github.io/ (visited on 03/12/2019).

[206] ImageMagick Studio LLC. *ImageMagick*. ImageMagick. URL: https://imagemagick.org/ (visited on 03/28/2019).

[207] Richard F. Chuang Haines. *The effects of video compression on acceptability of images for monitoring life sciences experiments*. July 1, 1992. URL: https://ntrs.nasa.gov/search.jsp?R=19920024689 (visited on 03/12/2019).

[208] Guido Achille Mansuelli. *Galleria degli Uffizi; le sculture*. Vol. 2. Istituto poligrafico dello Stato, Libreria dello Stato, 1958.

[209] VWHL. *Antoninus Pius Inv. 1914 n. 1*. The Uffizi Digitization Project. URL: http://www.digitalsculpture.org/florence/main/model/deed29a71c48484ebc46e9cf16946e02 (visited on 03/15/2019).

[210] Marco Tarini et al. "Marching Intersections: An Efficient Approach to Shape-from-Silhouette." In: *VMV*. 2002, pp. 283–290.

[211] Bruce Guenther Baumgart. *Geometric modeling for computer vision*. STANFORD UNIV CA DEPT OF COMPUTER SCIENCE, 1974.

[212] K. M. G. Cheung, S. Baker, and T. Kanade. "Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture". In: *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.* 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings. Vol. 1. June 2003, pp. I–I. DOI: 10.1109/CVPR.2003.1211340.

[213] Kong-man (German) Cheung, Simon Baker, and Takeo Kanade. "Shape-From-Silhouette Across Time Part I: Theory and Algorithms". In: *International Journal of Computer Vision* 62.3 (May 2005), pp. 221–247. DOI: 10.1007/s11263-005-4881-5. URL: http://link.springer.com/10.1007/s11263-005-4881-5 (visited on 03/14/2019).

[214] Aldo Laurentini. "The visual hull concept for silhouette-based image understanding". In: *IEEE Transactions on pattern analysis and machine intelligence* 16.2 (1994), pp. 150–162.

[215] Y. Matsumoto, K. Fujimura, and T. Kitamura. "Shape-from-Silhouette/Stereo and Its Application to 3-D Digitizer". In: *Discrete Geometry for Computer Imagery*. Ed. by Gilles Bertrand, Michel Couprie, and Laurent Perroton. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1999, pp. 177–188. ISBN: 978-3-540-49126-2.

[216] Wolfgang Niem. "Error analysis for silhouette-based 3D shape estimation from multiple views". In: *Proc. Int. Workshop on Synthetic-Natural Hybrid Coding and Three-Dimensional Imaging*. Citeseer, 1997, pp. 143–146.

[217] *Model Security*. Sketchfab Help Center. URL: http://help.sketchfab.com/hc/en-us/articles/202484576-Model-Security (visited on 02/14/2019).

[218] *Models getting stolen from sketchfab?* Sketchfab Forum. Nov. 4, 2017. URL: https://forum.sketchfab.com/t/models-getting-stolen-from-sketchfab/17437 (visited on 02/14/2019).

[219] Oleg Alexandrov and Ross A. Beyer. "Multiview Shape-From-Shading for Planetary Images". In: *Earth and Space Science* 5.10 (2018), pp. 652–666. DOI: 10.1029/2018EA000390. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018EA000390 (visited on 02/15/2019).

[220] *The NASA Ames Stereo Pipeline is a suite of automated geodesy & stereogrammetry tools designed for processing planetary imagery captured from orbiting and landed robotic explorers on other plan..* original-date: 2010-06-11T02:11:55Z. Feb. 13, 2019. URL: https://github.com/NeoGeographyToolkit/StereoPipeline (visited on 02/15/2019).

[221] A. Hertzmann and S.M. Seitz. "Example-based photometric stereo: shape reconstruction with general, varying BRDFs". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.8 (Aug. 2005), pp. 1254–1264. DOI: 10.1109/TPAMI.2005.158. URL: http://ieeexplore.ieee.org/document/1453513/ (visited on 03/14/2019).

[222] Tom Malzbender et al. "Surface Enhancement Using Real-time Photometric Stereo and Reflectance Transformation". In: (), p. 6.

[223] *Agisoft PhotoScan Change Log*. Agisoft Metashape. Dec. 23, 2018. URL: https://www.agisoft.com/pdf/photoscan_changelog.pdf (visited on 03/25/2019).

[224] *Home - CapturingReality.com*. CapturingReality. URL: https://www.capturingreality.com/ (visited on 02/05/2019).

[225] Daniel Scharstein and Richard Szeliski. "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms". In: *International Journal of Computer Vision* 47 (2001), pp. 7–42.

[226] *CloudCompare main repository. Contribute to CloudCompare/CloudCompare development by creating an account on GitHub*. original-date: 2012-11-06T20:15:51Z. Mar. 14, 2019. URL: https://github.com/CloudCompare/CloudCompare (visited on 03/15/2019).

[227] Timo Zinßer, Jochen Schmidt, and Heinrich Niemann. "Point Set Registration with Integrated Scale Estimation". In: (), p. 4.

[228] *Cloud-to-Mesh Distance - CloudCompareWiki*. URL: https://www.cloudcompare.org/doc/wiki/index.php?title=Cloud-to-Mesh_Distance (visited on 03/15/2019).

[229] *Distances Computation - CloudCompareWiki*. URL: https://www.cloudcompare.org/doc/wiki/index.php?title=Distances_Computation (visited on 03/15/2019).

[230] *File:NNDistancePrinciple.jpg - CloudCompareWiki*. CloudCompare. URL: https://www.cloudcompare.org/doc/wiki/index.php?title=File:NNDistancePrinciple.jpg (visited on 04/07/2019).

[231] Anish Athalye et al. "Synthesizing Robust Adversarial Examples". In: *arXiv:1707.07397 [cs]* (July 24, 2017). arXiv: 1707.07397. URL: http://arxiv.org/abs/1707.07397 (visited on 04/04/2019).

[232] *Nexus: adaptive 3D*. Visual Computing Lab. URL: http://vcg.isti.cnr.it/nexus/ (visited on 03/30/2019).

[233] *3DHOP - Home*. 3DHOP 3D Heritage Online Presenter. URL: http://vcg.isti.cnr.it/3dhop/ (visited on 03/30/2019).

[234] *Network Monitor*. MDN Web Docs. URL: https://developer.mozilla.org/en-US/docs/Tools/Network_Monitor (visited on 03/30/2019).

[235] *Throttling*. MDN Web Docs. URL: https://developer.mozilla.org/en-US/docs/Tools/Network_Monitor/Throttling (visited on 03/30/2019).

[236] *Requests: HTTP for Humans™ — Requests 2.21.0 documentation*. Requests: URL: http://docs.python-requests.org/en/master/ (visited on 03/27/2019).

[237] *_thread — Low-level threading API — Python 3.7.3 documentation*. Python. URL: https://docs.python.org/3/library/_thread.html (visited on 03/27/2019).

[238] *concurrent.futures — Launching parallel tasks — Python 3.7.3 documentation*. Python. URL: https://docs.python.org/3/library/concurrent.futures.html (visited on 03/30/2019).

[239] Brendan Cormier, ed. *Copy Culture: Sharing in the Age of Digital Reproduction*. Victoria and Albert Museum, South Kensington, London: V&A Publishing, 2018.

[240] *Copy Culture*. URL: https://www.vam.ac.uk/shop/copy-culture.html (visited on 03/18/2019).

[241] *The Leading Research & Experience Software*. Qualtrics. URL: https://www.qualtrics.com/ (visited on 03/24/2019).

[242] *Gender Distribution of Advanced Degrees in the Humanities*. URL: http://www.humanitiesIndicators.org/content/indicatorDoc.aspx?i=47 (visited on 03/18/2019).

[243] Dana N. Bardolph. "A critical evaluation of recent gendered publishing trends in American archaeology". In: *American Antiquity* 79.3 (2014), pp. 522–540.

[244] Jim Thacker. *Capturing Reality releases RealityCapture*. CG Channel. Dec. 28, 2015. URL: http://www.cgchannel.com/2016/02/capturing-reality-releases-realitycapture-in-open-beta/ (visited on 03/25/2019).

[245] *Search Results*. Europeana Collections. URL: https://www.europeana.eu/portal/en/search?f%5BMEDIA%5D%5B%5D=true&f%5BTYPE%5D%5B%5D=3D (visited on 03/19/2019).

[246] *Cultural Heritage Imaging | Digital Lab Notebook*. chi. URL: http://culturalheritageimaging.org/Technologies/Digital_Lab_Notebook/ (visited on 03/20/2019).

[247] *What is Unreal Engine 4*. Unreal Engine. URL: https://www.unrealengine.com/en-US/what-is-unreal-engine-4 (visited on 03/20/2019).

[248] *Potree*. URL: http://www.potree.org/index.html (visited on 03/22/2019).

[249] Tomasz Miaskiewicz and Kenneth A. Kozar. “Personas and user-centered design: How can personas benefit product design processes?” In: *Design Studies* 32.5 (Sept. 1, 2011), pp. 417–430. DOI: 10.1016/j.destud.2011.03.003. URL: http://www.sciencedirect.com/science/article/pii/S0142694X11000275 (visited on 10/18/2018).

[250] John Pruitt and Jonathan Grudin. “Personas: practice and theory”. In: *Proceedings of the 2003 conference on Designing for user experiences - DUX ’03*. the 2003 conference. San Francisco, California: ACM Press, 2003, p. 1. ISBN: 978-1-58113-728-6. DOI: 10.1145/997078.997089. URL: http://portal.acm.org/citation.cfm?doid=997078.997089 (visited on 10/18/2018).

[251] *Saint-Nicodème, Pluméliau (56) - Download Free 3D model by Archéomatique (@archeomatique)*. URL: https://sketchfab.com/models/8b6cca104a2742af93244b60fa03b402/embed?autostart=1 (visited on 04/05/2019).

[252] *Router*. URL: http://www.slimframework.com/docs/v3/objects/router.html (visited on 04/05/2019).

[253] *URI Routing — CodeIgniter 3.1.10 documentation*. URL: https://www.codeigniter.com/userguide3/general/routing.html (visited on 04/05/2019).

[254] Louis Columbus. *Roundup Of Cloud Computing Forecasts And Market Estimates, 2018*. Forbes. URL: https://www.forbes.com/sites/louiscolumbus/2018/09/23/roundup-of-cloud-computing-forecasts-and-market-estimates-2018/ (visited on 04/08/2019).

[255] *5 Cloud Computing Trends to Expect in 2019*. N2WS. Nov. 23, 2018. URL: https://n2ws.com/blog/aws-cloud/cloud-trends-2019 (visited on 04/08/2019).

[256] *GamesIndustry.biz presents... The Year In Numbers 2018*. GamesIndustry.biz. URL: https://www.gamesindustry.biz/articles/2018-12-17-gamesindustry-biz-presents-the-year-in-numbers-2018 (visited on 04/08/2019).

[257] *All Time Box Office Adjusted for Ticket Price Inflation*. URL: https://www.boxofficemojo.com/alltime/adjusted.htm (visited on 04/08/2019).

Leif Christiansen

Education

Indiana University – Bloomington IN **Graduated May 2019**
PhD in Informatics with Virtual Heritage track
Indiana University – Bloomington IN **Graduated May 2018**
Master of Science in Informatics; GPA: 4.0
Lewis & Clark College – Portland OR **Graduated May 2015**
Bachelor of the Arts in Mathematics and Computer Science; GPA: 3.9
Minors in Classical Studies and Music
Honors: *summa cum laude*, Phi Beta Kappa, Pi Mu Epsilon

Experience

Graduate Assistant Indiana University **Jan 2018-May 2019**
- Develop new technological apps to increase museum patron interest, engagement, and enjoyment.
- Applications include: phone tours, interactive 3D tablet apps, app for Microsoft HoloLens.

Software Developer Flyover Productions **June 2018-May 2019**
- Implemented multiplayer support for existing Virtual Reality app.
- Designed and built Drupal website as educational component to Virtual Reality app to offer forums for community engagement, quizzes, connection to social media, and achievements.

Tutor Bloomington Tutors **Jan 2016-May 2019**
- Work with students to teach concepts in finite math, propositional logic, predicate calculus, and python.

iOS App Developer – 4Dice Indiana University **Sep 2016-May 2019**
- Port mobile app for loading and viewing 4D objects from Linux to iOS.
- Updated app visualizing 4D cube to 2016 iOS standards, added 4D interaction, and posted to app store.

Assistant Instructor Indiana University **Sep 2015-Dec 2017**
- Assisted professors in teaching the courses Interactive Computer Graphics and Programming Virtual Reality.
- Wrote and graded programming assignments, taught weekly labs, led lectures, and held review sessions.

Languages and Technologies

- GLSL (OpenGL, OpenGL ES, and WebGL), C#.NET, C/C++, Python, Javascript, HTML/CSS, R, jQuery
- Unity, Zbrush, 3D Studio Max, Blender, Xcode, Drupal
- Conversational Italian (3 years' experience), Conversational Spanish (4 years' experience)

Projects

Uffizi Digitization Project **May 2016-May 2019**
- Use photogrammetry to create 3D models of ancient sculpture for preservation and study.
- Manage photo, 3D, and metadata files for project manually and with scripts to ensure security and integrity.
- See the results of the project at https://digitalsculpture-uffizi.org

Data Mining Latin Text **April 2017**
- Used Naïve Bayes to test authorship of Latin texts from http://www.thelatinlibrary.com/.
- Found quantitative support for author comparisons made in Classics literature.

VR Temple Reconstruction **April 2016-May2016**

- Created HTC Vive application of Roman temple in order to test reconstructions of temple and cult statue.
- Allowed user to display reconstructions, alter fragment placements, view info, and change lighting settings.

Publications

Christiansen, L. (2019). Extending Model Use in Virtual Heritage: User-Centric Implementation of a Protected Remote Rendering Visualization Tool. [Bloomington, Indiana]: Indiana University.

Brennan, M. & Christiansen, L. (Forthcoming) Virtual Materiality: A Virtual Reality Framework for the Analysis and Visualization of Cultural Heritage 3D Models. *Digital Heritage*.

Donaldson, D. R., McClanahan, A., Christiansen, L., Bell, L., Narlock, M., Martin, S., & Suby, H. (2018). Media Digitization and Preservation Initiative: A Case Study.

Patay-Horváth, A., & Christiansen, L. (2017). From Reconstruction to Analysis. Re-use and re-purposing of 3D scan datasets obtained from ancient Greek marble sculpture. *Studies in Digital Heritage*, *1*(2), 491-500. https://doi.org/https://doi.org/10.14434/sdh.v1i2.23236.

Kamburugamuve, Supun, Christiansen, Leif, Fox, Geoffrey. “A Framework for Real-Time Processing of Sensor Data in the Cloud.” *Journal of Sensors* (2015): http://dx.doi.org/10.1155/2015/468047.