Virtual Materiality: A Virtual Reality Framework for the Analysis and Visualization of Cultural Heritage 3D Models

Matthew Brennan
School of Informatics and Computing, Indiana University
Bloomington, Indiana, USA
matthewrbrennan@gmail.com

Leif Christiansen
School of Informatics and Computing, Indiana University
Bloomington, Indiana, USA
leifchristiansen92@gmail.com

***Abstract*— Relatively recent advances in software and hardware now allow almost anyone to create digital facsimiles of real-world physical objects, and to view, interact with, and analyze these models in a variety of environments. Traditionally, art-historical, archaeological, and anthropological analysis of material objects required physical access to, and interaction with, the artifacts in question. Virtual reality environments provide, as near as is currently possible, a physical, "material" interaction with an otherwise purely digital object or objects, simulating traditional methods of inquiry, while offering the opportunity for physically or logistically impractical interactions with, and actions upon, the objects. The purpose of this study is the development and testing of a real-time, virtual reality tool for collaborative analysis of metrically accurate 3-dimensional, digital reproductions of physical objects.**

Keywords*— *virtual reality, 3D models, visualization, cultural heritage

I. Introduction

Relatively recent advances in software and hardware now allow almost anyone to create digital facsimiles of real-world physical objects, and to view, interact with, and analyze these models in a variety of environments. Traditionally, art-historical, archaeological, and anthropological analysis of material objects required physical access to, and interaction with, the artifacts in question. Virtual reality environments provide, as near as is currently possible, a physical, "material" interaction with an otherwise purely digital object or objects, simulating traditional methods of inquiry, while offering the opportunity for physically or logistically impractical interactions with, and actions upon, the objects. The purpose of this study is the development and testing of a real-time, virtual reality tool for collaborative analysis of metrically accurate 3D, digital reproductions of physical objects. This framework provides a suite of didactic, comparative, and analytic tools aimed at replicating and augmenting the traditional physical approaches to material object study. Our framework is targeted specifically for reuse, utilizing the existing infrastructure and technologies offered by Unity for collaboration, dissemination, and sharing of applications.

II. Digital 3D Models for Cultural Heritage

3D models have proven useful to academic study, generating insights on the function and use of ancient monuments [1], the structural integrity of buildings [2], and even ancient Roman practices of statue copying [3]. Accordingly, cultural and government institutions have increasingly invested in projects to digitize cultural objects. Notably, the Uffizi museum in partnership with Indiana University has produced hundreds of models which have been made publicly available through the 3D model sharing site sketchfab.com [4]. Additionally, the EU funded 3D-ICONS project has contributed an equally large number of 3D models to the Europeana digital platform [5]. Thus, there now exists a wealth of publicly and privately available 3D models of cultural heritage objects with potential scholarly benefit.

A. Virtual Reality and Digital Materiality

While 3D models may be incredibly detailed and data rich representations of physical objects, interaction with these models is always a highly mediated experience, requiring computers and input hardware. Typical interaction with 3D is done either using standard computer monitors with a keyboard and mouse or on touchscreen devices like smartphones and tablets. However, viewing 3D models on 2D displays limits their full potential as it lacks proper visual cues for a realistic sense of scale and intuitive input methods.

Virtual reality (VR) as a medium, offers an unparalleled level of immersion in a virtual environment, which may address the tunnel vision forced upon a user by a computer screen, where they are effectively looking through a window into virtual space. VR headsets afford the user a feeling of 'presence' in a virtual environment, as well as a sense of real-world scale when standing before a digital object, that are impossible to achieve with screen-based applications [6]. Sophisticated tracking of the user's head and hands allows even minute movements to be replicated in the virtual environment. This kinetic quality not only heightens the sense of immersion in the virtual space, but allows the user to interact with digital models using physical gestures. Digital 3D models can be ‘picked up’ using tracked hand controllers and held just as if one were picking up a scale model, or actual object, in reality.

B. The Unity real-time Engine

Previous work has sought to bring the benefit of 3D models to cultural heritage institutions and scholars. In order to speed development times and lower costs, scholars have used freely available game engines such as Unity3D, Unreal Engine, and CryEngine (for a summary and comparison of their features see [7]). Unity3D has proven popular in VR applications for museums. [8] developed several applications for the HTC Vive to display ancient monuments and archaeological sites with ‘hotspots’ providing additional contextual information. Similarly, [9] exhibited a digitized model of the Selimiye mosque in the city of Edirne, Turkey using the HTC Vive, which offered multiplayer support. [7] created a fully virtual museum housing works from the The B & M Theocharakis Foundation for the Fine Arts and Music, a virtual structure modeled after the original building that allowed users to navigate a traditional museum space using a desktop computer.

Two avenues for extending the current work exist. First, the majority of VR applications using 3D models of cultural heritage objects are educational, displaying current knowledge and scholarship associated with the modeled object or space. However, the question remains, how may these VR applications be used to further the work of scholars and generate new insights on the modeled objects? Second, scholars rarely make the technical products of their studies available, i.e. the code developed. This may in part be attributed to how closely most of these applications are tied to individual objects. Museums and scholars are not always willing or at liberty to share their 3D models. This practice seems out of keeping with the spirit of scholarship and hinders the work of future scholars, as it requires duplicated efforts.

III. A VR Framework for Analysis and Visualization of Cultural Heritage 3D Models

What follows will be a description and discussion of the various tools and methods implemented in our VR application for the analysis and visualization of cultural heritage 3D models.

A. Technologies Used

The software foundation for the application was the Unity game engine. Unity was chosen for its low cost (free for personal, educational, and non-commercial use), robust documentation, and native support for VR. In addition, Unity has a rich ecosystem of reusable free and paid packages available on the Unity Asset store that may be quickly and easily incorporated into existing Unity projects (Unity asset store). Assets include VR support (VRTK), multiplayer (Photon), and image effects (Post Processing Stack).

The VR hardware was chosen based on cost (at writing, $399 for Oculus Rift and Touch), although the application will run on both of the main currently available VR headsets, HTC Vive [10] and Oculus Rift [11]. Cheaper headsets are available, such as the Samsung GearVR [12], Oculus Go [13], and Google Cardboard [14]. However, these options lack the full spatial tracking that allows users to move both headset and controllers with six degrees of freedom (3-axis rotational tracking and 3-axis positional tracking), and the computational power, at present, to run complex 3D scenes.

The 3D models given the most consideration are those created with photogrammetry and which have a scale bar present. This decision was made due to the fact that photogrammetric models are increasingly prevalent in the archaeological, anthropological, and art historical fields, and, by their nature, are based on physical objects in reality.

Finally, in order to ensure access and reuse of the final application, all the code for the project has been made available on GitHub [15]. GitHub is a code-hosting and version-control repository based on Git. GitHub allows for easy access and collaboration on code-based projects.

B. Tools and Methods

1) *Designing for Virtual Reality:* When designing for VR, there are a number of unique considerations and constraints concerning locomotion, interaction, the use of text, and menu systems. To avoid cybersickness, the application uses “blink teleportation” movement, whereby when the user indicates a point on the ground in the virtual environment, their view briefly fades to black (about 0.2 seconds) and then is “teleported” to the selected location. The application was designed for use with tracked ‘hand controllers’, which are currently standard components for both the Rift and Vive headsets, and which accurately track the position of the user’s hands and render them as controllers in the virtual environment.

   The menu system was designed in such a way that it ‘follows’ the user as they teleport, and can be repositioned using the hand controllers in 3D space. The size and appearance of the menu is reminiscent of a clipboard or digital tablet, and the user interacts with it by using the hand controllers. To avoid having to stretch or strain to physically “reach” the menu, the hand controllers project a virtual “stylus” or selection laser, when aimed at the menu.

   The menu system is divided into three columns: Models, Modify, and Lights. Models and lights are each virtual objects in the environment that can be selected, and once selected, be modified via the options in the Modify column, which are specific to the chosen object.

2) *Loading a 3D Model and Setting its Scale:* The foundational function of the application is the ability to load one or more 3D models from the user’s Unity “project folder” into the virtual environment. A model is inserted into the scene at a default scale, where its largest X-Y-Z dimension is set to 1 meter. Often, models created using photogrammetry include targets or scale bars for defining scale, and if this is the case, the user can simply select two points, such as the beginning

and end of a scale bar, to set the scale of the virtual object. Once points are selected, the user may then enter the scale of the selected segment in meters and the model is rescaled appropriately (see Fig. 1).

Properly scaling models is important for a number of reasons. In a VR environment, objects appear 'at scale' relative to the user; this is not the case in screen-based model viewers, such as Sketchfab.com. Additionally, if the user wishes to make measurements on a model, it must have a properly defined scale, otherwise those measurements will be arbitrary or incorrect. Finally, when comparing multiple models in the virtual environment, it is important that the models are correctly scaled not only relative to user, but to each other.

3) *Manipulating and Transforming Models in VR:* Models that have been loaded into the VR environment may be transformed in two ways: 'grab' and move/rotate using the hand controller or by using translate, rotate, and scale gizmos that are a staple tool of 3D modeling and editing software. These two options respectively give the user a more physical, kinetic experience via the 'grab' mechanic, or, when detailed interactions are desired, fine-grain control over movement, rotation, and scale via the gizmos.

Intuitive 3D rotation is a particularly challenging problem, especially using 2D displays. A number of schemes have been introduced for mapping 2D input to 3D space including rolling ball [16] and various implementations of the virtual trackball [17]. Fully tracked hand controllers ameliorate the problem of 2D input. Using the grab mechanic movements of the controller are applied to a 3D object, as if one was holding an object grasped in their hand. However, in the case of life-sized and over-life-sized objects, the grab mechanic may seem counterintuitive and preclude fine-grained control. For this reason, we have also included the rotation gizmo, which implements fixed-axis rotation akin to a turntable as well as arbitrary axis rotation using virtual trackball rotation.

When the scale tool is active, the model can only be scaled uniformly. Such a constraint was implemented to reduce accidental 'unrealistic' distortion of models based on real-world objects through non-uniform scaling. When the rotate tool is active, the model can be freely rotated, or constrained to a certain axis. The user can transform a model to their liking and then set those values as a default state, or revert the transforms to the originally loaded values (see Fig. 2).

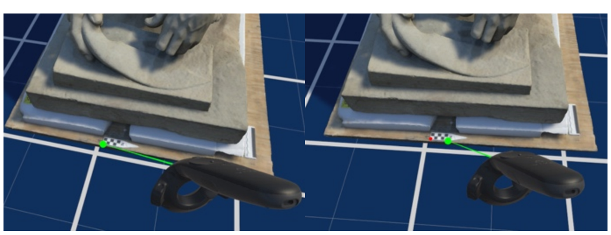

Fig. 1. A user selecting two points on a scale.

A particularly useful feature when comparing two or more models is the ability to mirror the transformations from one model to the other(s). The reason for implementing this is the limits of visual working memory. The capacity of visual working memory "is limited to a small number of simple visual objects and patterns, perhaps three to five…" [18]. By quickly mirroring transforms performed on one object to the other, the cognitive effort and time of switching attention is minimized.

4) *Visualization and Lighting:* Digital 3D models, like their physical counterparts, can come in a variety of materials and textures, and can have surfaces that reflect or absorb light in highly realistic ways. These properties can either be used to simulate the surface and material qualities of the original piece on which the model is based or derived from, or they can be digitally manipulated and modified in order to make visualization of certain aspects of a model easier. One of the great benefits of 3D digital models and environments is the ease with which lighting, material, and texture can be modified and varied.

In visualizations of 3D digital models and environments, lighting can play two roles: it will either be dynamic and modifiable by the user, for analytic and investigative purposes, or it will be as accurate a simulation of natural ambient or situational light as possible. In the former case, a user should be able to modify the direction, intensity, color, position, and number of lights in the virtual environment. For example, in the analysis of a sculpture, form and surface texture are essential to an understanding of the object not only as a 3D volume, but also as an art historical object. Strategically changing the direction and intensity of lights in a virtual scene can serve to

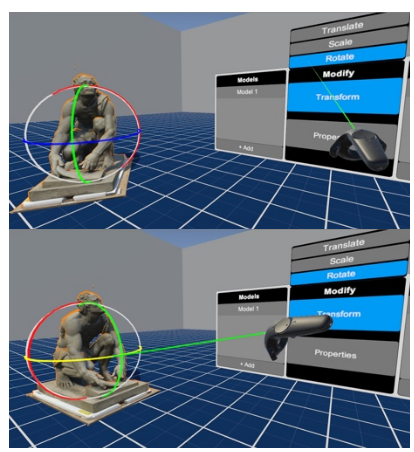

Fig. 2. A user selecting the Rotate tool and rotating an object.

serve to highlight and accentuate subtle surface variation. In conservation and archaeology, the technique and usefulness of applying 'raking light' to a surface for visual inspection is well known.

Another important consideration in lighting parameters for visualization of 3D models is shading and shadows. Shading can dramatically affect the perception of a surface and its texture, and reveal fine details. However, shadows can also obscure important information and, if the light source is moving, be visually distracting. Certain visualizations may benefit from and even require the rendering of strong shadows and shading, while in others, it may make analysis of the 3D model extremely difficult.

In the present application, a user is able to create multiple lights, and place them within the scene. The light types correspond to Unity's classes of "Point Light", "Spot Light", and "Directional Light". The color, intensity, rotation, and position of the lights can be modified, as well as whether they cast shadows (see Fig. 3).

5) *Material and Texture:* Differences in material and texture are what cue our visual system to changes in surface and the boundaries of objects in space. In computer graphics, texture most often means an image map of some kind that is applied to a surface mesh. Material refers to additional surface and physical properties such as reflection, scattering, and density.

In the fields of art history, archaeology, and virtual heritage, it is often desired for 3D digital models to look as much like the real object on which they are based as possible. Because many of these models are created using photogrammetry, they often have a texture that is created using the same photographs from which the model itself was reconstructed. However, photogrammetry models do not inherently contain any built-in information about materiality (i.e. stone vs. wood vs. bronze, etc…). Material definitions must be added in using the visualization medium, whether it is a webGL viewer such as Sketchfab that supports physically based rendering (PBR) or a real-time game engine that uses metalness or specular shaders to approximate surface materials. Materiality properties do not have a major impact on analytical visualization, but they can certainly help in the "believability" of a 3D digital model as being based on a real world object.

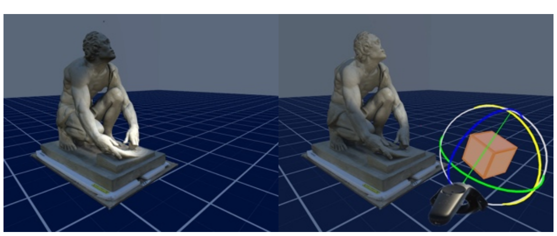

Fig. 3. User modification of a spot light.

As with lighting, shading, and shadows, in an interactive visualization of 3D digital models for analytical purposes, the user should be given the option of enabling or disabling materiality and texture. In Figure 4, the specular highlights on the bronze sculpture's back help reveal the "fine details of surface structure" [18], that is invisible in the simple textured model. However, those highlights also obscure the surface texture with their brightness, which the simple textured model does not suffer from. In fact, for an artist or sculptor, who might be more interested in the pure form of a digital model, any sort of texturing or materiality might prove distracting, and such a user might wish to completely change the material of the model to a flat color, so as to better study the play of light on the surface.

Fig. 4. A model with specular highlights and reflectivity (left), rendered unlit flat-color (middle), and rendered in untextured MatCap (right).

In the present application, users have the ability to change a model's material: they can toggle the jpeg texture on and off, switch the shader to a flat color without reflections, or a simple "MatCap" shader, for inspection of form and surface. Users can also create or import their own shaders to add to the possible options.

C. *Potential for Reuse*

Our application was specifically designed and maintained with reuse in mind. Both the Unity project source code and a pre-built application are available via GitHub at https://github.com/leifchri92/VR-Model-Inspector. Using GitHub, one may download the entire unity project and all associated code. Detailed instructions contained in the project walk users through how to add their own models and build and run the application. To add models, users may simply drag and drop their models, in a Unity supported file format [19], into the appropriate project folder. Users may then go on to modify or add model and light modification behaviors by editing the provided scripts. If users would rather use the application as is, they may simply run the executable, which provides runtime support for the loading of 3D model files in the .obj format.

IV. Future Work

Future work involves implementation of collaborative analysis tools, so that multiple users, in different locations, can interact with each other, and 3D models, in the same environment. Additionally, the authors hope to solicit feedback on the usefulness and usability of the virtual reality framework. This may be done informally, through the sharing and distribution of the application via GitHub, and formally, through the use of user tests focused on the usability and intelligibility of the implemented tools.

V. Conclusion

The functionality outlined here is but a first attempt at bringing to bear upon 3D cultural heritage models the great potential of analysis facilitated by virtual reality environments. By combining accurate and realistic 3D models of cultural heritage artifacts with the immersive qualities of VR, it is possible to offer a very near to real experience of interaction with the actual artifact. The VR environment and 3D model may in fact afford opportunities and avenues of inquiry that would be altogether impossible in reality. In short, the authors hope that this publicly available framework and its tools will serve as a foundation for further iteration and collaboration on a useful and extensible VR environment tailored to the analysis of cultural heritage 3D models by scholars, students, and enthusiasts alike.

References

[1] B. Frischer, G. Zotti, Z, Mari, and G. C. Vittozzi, "Archaeoastronimical experiments supported by virtual simulation environments: Celestial alignments in the Antinoeion at Hadrian's Villa (Tivoli, Italy)," Digital Applications in Archaeology and Cultural Heritage, vol. 3, pp. 55-79, 2016.

[2] C. Costa, A. Arêde, M. Morais, and A. Aníbal, "Detailed FE and DE Modelling of Stone Masonry Arch Bridges for the Assessment of Load-carrying Capacity," Procedia Engineering, vol. 114, pp 854-861, January 2015.

[3] K. Sengoku, S. Buseki, S. Ono, T. Oishi, T. Masuda, and K. Ikeuchi, "Polykleitos and His Followers at Work: How the Doryphoros Was Used," in New Approaches to the Temple of Zeus at Olympia, A. Patay-Horváth Ed. Cambridge Scholars Publishing, 2015, pp. 201-222.

[4] "The Uffizi Digitization Project," The Uffizi Digitization Project, https://www.digitalsculpture-uffizi.org/, accessed September 3, 2018.

[5] "3D-ICONS," Europeana Pro, https://pro.europeana.eu/project/3d-icons, accessed September 3, 2018.

[6] B. Witmer and M. J. Singer, "Measuring presence in virtual environment: A presence questionnaire," Presence, vol. 7, pp. 225-240, 1998.

[7] C. Kiourt, A. Koutsoudis, S. Markantonatou, and G. Pavladis, "THE 'SYNTHESIS' VIRTUAL [7] MUSEUM," Mediterranean Archaeology & Archaeometry, vol. 16, pp. 1-9, 2016.

[8] B. Jiménez Fernández-Palacios, D. Morabito, and F. Remondino, "Access to complex reality-based 3D models using virtual reality solutions," Journal of Cultural Heritage, vol. 23, pp. 40-48, January 2017.

[9] T. P. Kersten, G. Büyüksalih, F. Tschirschwitz, T. Kan, S. Deggim, Y. Kaya, and A. P. Baskaraca, "THE SELIMIYE MOSQUE OF EDIRNE, TURKEY-AN IMMERSIVE AND INTERACTIVE VIRTUAL REALITY EXPERIENCE USING HTC VIVE," International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences, vol. 42, pp. 403-409, 2017.

[10] "Vive Virtual Reality Systmem," Vive, https://www.vive.com/us/product/vive-virtual-reality-system/, accessed September 3, 2018.

[11] "Oculus Rift," Oculus, https://www.oculus.com/rift/, accessed September 3, 2018.

[12] Samsung Gear VR with Controller," Samsung, https://www.samsung.com/global/galaxy/gear-vr/, accessed September 3, 2018.

[13] "Oculus Go," Oculus, https://www.oculus.com/go/, accessed September 3, 2018.

[14] "Google Cardboard," Google, https://vr.google.com/cardboard/, accessed September 3, 2018.

[15] "GitHub," GitHub, https://github.com/, accessed September 3, 2018.

[16] A. J. Hanson, "The rolling ball," in Graphics Gems III, pp. 51-60, 1992.

[17] K. Henriksen, J. Sporring, and K. Hornbæk, "Virtual trackballs revisited," IEEE Transactions on Visualization and Computer Graphics, vol. 10, pp. 206-216, 2004.

[18] C. Ware. *Information Visualization : Perception for Design*. Third edition. Waltham, MA: Morgan Kaufmann, 2013.

[19] "Model File Formats," Unity, https://docs.unity3d.com/Manual/3D-formats.html, accessed September 5, 2018.