An Online Model Viewer for Cultural Heritage in Unity 3D

Tyler Garcia, Zhangchi Lyu, Michael Tetzlaff, University of Wisconsin – Stout, Menomonie, WI

Abstract

We present a prototype of a 3D model viewer for cultural heritage built in Unity 3D [1]. The Unity engine allows for the use of custom shaders that can increase realism and provide a uniquely immersive experience with cultural heritage artifacts. Examples of custom shaders that this architecture supports include image-based specular reflections and polynomial texture mapping (PTM) for translucent materials. Furthermore, the software can be built and deployed as a stand-alone WebGL application viewable in the browser without relying on third-party content delivery services. The code for this project is open source and available on GitHub.[1]

Introduction

The demand for cultural heritage in digital spaces is growing rapidly. Knowing that a global pandemic or simply the barrier of physical distance from cultural artifacts may keep much of a potentially interested public away from the physical space of a museum, many cultural heritage organizations are exploring digital options for sharing their artifacts online. Though sharing 3D models of cultural artifacts is nothing new, the methods of deploying these models for public viewing are anything but uniform. No piece of software can boast the claim of serving every need of the cultural heritage community at once, leading institutions to use a range of 3D programs and platforms to share their models. Common to almost all the platforms is a lack of rigor in the visual fidelity of the 3D object's appearance, due in part to the use of a content pipeline designed for entertainment applications like film and games, rather than cultural heritage.

In this work, we develop a new 3D model viewer that advances rendering quality by using the Unity engine's render pipeline and custom shaders designed for specific cultural heritage applications. While dependent on the commercial Unity engine, the source code for this project itself is open source, providing the flexibility and extensibility required by cultural heritage institutions. Through this project, we intend to demonstrate a unique model viewer designed first and foremost for cultural heritage that supports state-of-the-art real-time lighting and shading.

Background

The potential of 3D visualization as a resource for studying cultural artifacts has been widely discussed and explored. Visualization can act as a powerful medium for indexed, archival knowledge [2]. Some researchers have proposed digital tools and machine learning to assist in large-scale preservation and awareness of cultural sites [3]. However, caution is needed when using 3D visualization in the field of cultural heritage. When using 3D models of artifacts for applications like restorations, preserving accuracy should be a priority. Care must also be taken to avoid commodification and devaluing of culture in a digital and reproducible format [4]. Still, 3D visualization for cultural heritage holds great potential for preserving and spreading awareness of cultural artifacts.

***Figure 1:** A 3D model of a bronze statue depicting* Guan Yu *(from the collection of the Minneapolis Institute of Art) hosted on our novel 3D viewer built in the Unity engine.*

Although 3D heritage visualization is not a new concept, there is still no standardized format for sharing and viewing 3D models for cultural heritage. This has led to a lack of any universally recognized solution to the issues present in visualizing and sharing 3D heritage models [5]. Because of this, various solutions have been utilized across the digital heritage community, with many using existing 3D model viewers.

One of the most popular platforms for sharing heritage models is Sketchfab (see **Figure 2**). Though more of a model marketplace than a heritage-oriented online viewer, Sketchfab has been very welcoming of the cultural heritage community on its platform. Attractive features of the platform include ease of upload and the ability to add annotations to a model. Sketchfab has started a heritage program that grants free premium accounts to heritage organizations and museums to facilitate their needs [6].

Researchers in the cultural heritage community have also worked to build their own web-based solutions for model sharing. 3D Heritage Online Presenter (3DHOP, see **Figure 2**) is designed with the intent of visualizing individual high-resolution models. It is not a perfect tool for model viewing, but it is geared towards specific

[1] https://github.com/garciat5597/DH-Model-Viewer

needs of the cultural heritage community [7]. 3DHOP powers the ARIADNE Visual Media Service, a platform designed for sharing high resolution images and 3D models for archaeology. ARIADNE also boasts a very accessible interface for uploading content, making it easy for those inexperienced with technical software to publish [8]. The Smithsonian has their own 3D platform, Voyager, which the museum has embedded into web pages to display 3D models along with annotations and information blocks [9].

All these solutions avoid the need to download specialized software onto each machine to view 3D models. These platforms are effective at quickly getting 3D representations of objects out to the public, but none of them fully encompass all the needs of the cultural heritage community.

Sketchfab is a closed source program and cannot be altered to fit the needs of cultural heritage models, such as the use of custom shaders to replicate real world materials. Its lossy compression is not adequate for sharing high resolution objects. Furthermore, the Sketchfab viewer cannot be used for content hosted outside Sketchfab servers, making any integration dependent on Sketchfab as a third party.

On the other hand, while 3DHOP and Voyager are open-source and can technically be modified to suit any user's needs, practically speaking they require active development to keep up with the state-of-the-art in real-time 3D graphics, an effort that may be a challenge for the cultural heritage community to maintain in the long term. These platforms also do not currently support custom shaders as naturally as Unity (although as open-source projects, they could theoretically be modified to add support for such new shaders). 3DHOP is also not suitable for complex scenes with multiple objects (even low-polygon models), making its usage slightly more limiting when it comes to building such scenes [10].

Because of the limitations and barriers we found with existing web viewers, we decided to prototype a 3D viewer using the Unity engine, which has been used in the past for cultural heritage applications [11]–[13], including VR galleries and experiences [14]–[17]. However, unlike prior works using Unity which tend to be stand-alone experiences limited to exploring a single artifact or cultural heritage space, we intend for our platform to be a general-purpose viewer comparable to Sketchfab, Voyager, or 3DHOP.

Method

In consultation with photographers in the Visual Resources department at the Minneapolis Institute of Art (Mia), we developed a prototype of a lightweight, embedded web viewer that supports custom shaders and state-of-the-art real-time lighting. To best address what we perceived as unmet needs of the institutions generating 3D cultural assets, we attempted to strike a balance between Sketchfab and solutions oriented more towards cultural heritage like 3DHOP or Smithsonian Voyager. Our approach leverages the rendering power of Unity (a commercial product like Sketchfab) to build an open-source application that is deployable on institutional servers, like 3DHOP or Voyager. In our embeddable web viewer, we use custom shaders that leverage photographs taken with an on-camera flash, similar to prior work [18], [19] but with substantial performance improvements over the techniques used previously. As in that prior work, we use 3D models built from the flash photographs using off-the-shelf photogrammetry software (Agisoft Metashape).

Figure 2: *Examples of 3D models hosted on existing web platforms. Left: A 5th century BCE bronze* ding *vessel from China (from the collection of Mia), hosted on Sketchfab. Right:* Capsella Samagher *hosted on 3DHOP.*

A good set of guidelines for how to take flash photographs for this workflow can be found in the documentation for IBRelight[2], or in the appendix of Tetzlaff's Ph.D. thesis [20]. One of the most notable differences from standard photography best practices is that photographs typically need to be underexposed by "normal" aesthetic standards, in order to capture the full shape of the specular highlights. Because so many photographs are taken, the graininess that might become apparent when digitally increasing the brightness of an underexposed image is typically not observed after combining all of the images. (In theory, high dynamic range imaging could be used to avoid this issue altogether, but we have found the underexposure method to be sufficient in practice.)

Basis function representation for specular objects

For metallic objects, the source photographs can be fit to specular basis functions using a method similar to Nam et al. [21]. The essence of this technique is that a small number of basis functions (eight to sixteen) are used to represent the specular materials present in the object's surface. To model how the specularity varies on the object's surface, weight textures are estimated, which modulate these basis functions at every pixel. To allow the basis functions to be only one-dimensional (significantly reducing the amount of photographic data required for numerical stability), a normal map is also estimated, which determines where the specular peaks are located. These three components of the solution (the basis functions, weight maps, and normal maps) are alternatingly optimized in a least-squares sense, iterating until some convergence threshold is reached.

The result of this is a representation that can be efficiently evaluated in real time at every pixel on the object by performing simple lookups into each basis function and weight texture, multiplying, and summing the results. Unlike prior results like IBRelight [19], which required all the source photographs to be loaded onto the graphics card at runtime, this technique only uses

[2] IBRelight documentation: https://drive.google.com/file/d/17VxAho2rIoRgHWHNE8y6MbnsYE2qc1Hc/view

the source photographs for a one-time “build” step. After that, the basis functions, weight maps, and normal map are all that is theoretically needed for real-time rendering. The time complexity of the shader is also much better; whereas the technique used by IBRelight scaled with the number of photographs loaded at once for real-time rendering (resulting in lower framerates the more images were available), the technique used here scales with the number of photographs when building (more photographs take longer to optimize; similar to other existing tools like Metashape), but requires just a constant amount of time for real-time rendering.

Our implementation differs from Nam et al. in that we use Levenberg-Marquardt non-linear optimization for the normal map, which we found to result in more accurate and robust estimation of surface texture. In addition, to take advantage of Unity’s built-in global illumination, we estimate traditional specular and roughness textures by fitting the basis function representation at each surface position to a GGX microfacet distribution. These textures are a further simplified representation, used only to approximate global illumination (which would otherwise be computationally challenging to evaluate in real time); whereas for direct illumination

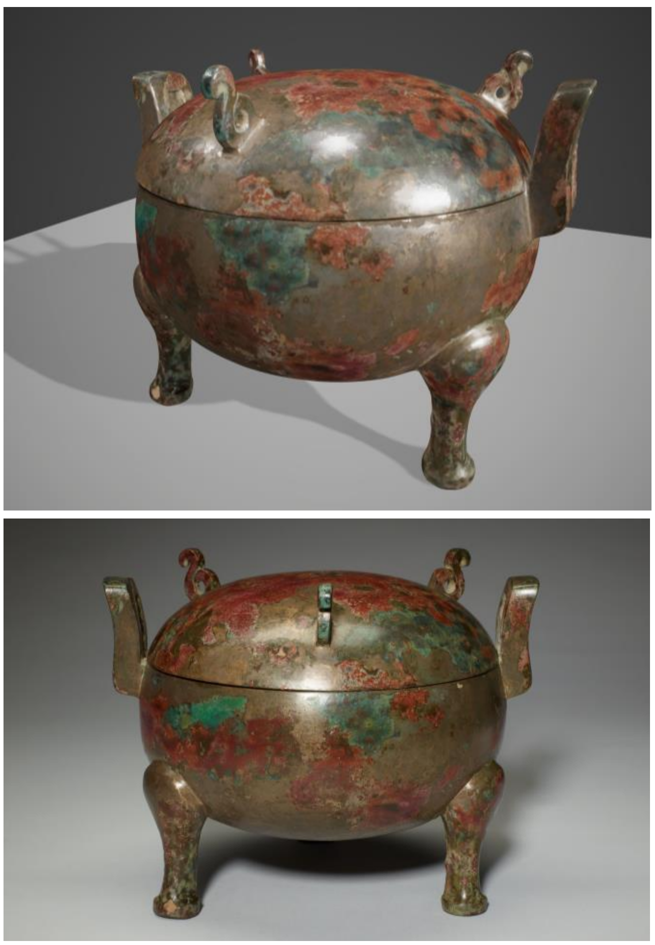

Figure 4: *Top: An depiction of the results of the specular technique for a bronze* ding *vessel from 5th century BCE China (from the collection of Mia). Bottom: catalogue photograph for reference. Although it is not the same* ding *vessel as the one shown in in* ***Figure 2****, one can still appreciate the richness of appearance afforded by our custom shaders, which is not apparent on Sketchfab.*

Figure 3: *Left: An example of the results of the specular technique for a Cloisonné altarpiece from 18th century China (from the collection of Mia). Right: catalogue photograph for reference.*

from discrete light sources, our Unity shader directly evaluates the basis functions and weight maps to maximize highlight quality. This shading technique was used previously for an object in the “Zen Zone” of the game *Vessel*, a capstone project developed by students at UW-Stout [13], but is now presented in a more traditional virtual gallery along with a source code release (see footnote on page 1).

Figure 1, **Figure 3**, and **Figure 4** illustrate the results of this method. In all three cases, the specularity would not be visible in the color texture produced by Metashape. (An example of this can be seen by comparing the *ding* vessel in **Figure 4** with the similar vessel in **Figure 2**, which uses a simple Metashape texture with no specularity.) The Cloisonné altarpiece in **Figure 3** is an example of an object for which it would be very challenging to manually specify the specularity parameters, as the specular properties are highly spatially varying. Even the more homogeneous statue of *Guan Yu* shown in **Figure 1** has a certain richness to its metallicity that might not be easily captured by a traditional specular model.

PTM / RTI representation for translucent objects

For translucent objects, polynomial texture mapping (PTM; sometimes referred to as reflectance transformation imaging or RTI) may be applied. PTM is a powerful way to represent and render the non-Lambertian reflectance of important objects in cultural heritage and other fields which demand high fidelity. As with the method for specular objects, the primary inputs to the PTM technique are photographs of the object taken from different angles using a single camera-mounted flash light source. In contrast with the classical PTM rig which requires a dome filled with many discrete light sources, our solution is much simpler, requiring just a dark room and a camera with a single flash. Moreover, whereas PTMs are typically captured from only a single viewpoint, we extend the PTM technique to full 3D by moving the camera along with the light source.

Figure 5: *An illustration of the results of the 3D PTM technique for* Traveling in Autumn Mountains*, a jade sculpture in Mia's collection. The images on the left use a camera-mounted flash light source, while the images on the right are a more natural studio lighting configuration.*
Top row: Renderings using the color texture produced by Agisoft Metashape (comparable to the standard shaders used by most existing 3D viewers).
Middle row: Renderings using the custom PTM shader introduced in this work.
Bottom row: Photographs: one of the source flash images on the left, and a catalogue image of the object on the right for reference. While a standard shader may be effective for enhancing geometric details in the sculpture, the PTM shader better matches the translucent appearance of real jade.

Our implementation of PTM is a 3D variant of the technique described by Zhang and Drew [22]. They used the following formulation for the PTM basis functions:

$$P(u,v) = (1, u, v, w, u^2, uv)$$

In this formula, the world-space light direction is parameterized using the coordinates (u, v), from which the third coordinate can be derived: $w = \sqrt{1 - u^2 - v^2}$. We make two small changes to this set of basis functions. First, we perform the fitting in tangent space (the same coordinate space typically used for "normal mapping" in traditional computer graphics), letting u and v correspond to the components of the light direction along the tangent and bitangent axes. This ensures that these basis functions are equally valid for all sides of the 3D object. Second, we replace the u^2 term with $u^2 + v^2$ to maintain symmetry, since the orientation of the tangents and bitangents within the tangent plane is arbitrary:

$$P(u,v) = (1, u, v, w, u^2 + v^2, uv)$$

With these small changes to the basis function, we perform the PTM fitting as in prior work, using a linear least squares regression to produce six weight maps, each corresponding to one of the basis functions. In practice, luminance data is represented as an RGB color, so we calculate a separate set of weight maps for each RGB channel. The results are encoded as six RGB texture images, which are ideal for use in a shader within a game engine like Unity.

In order to render a novel image of the model using these weight maps, we evaluate the basis functions with the virtual light direction, then multiply by the corresponding weights from the weight maps, and sum the results. The polynomial approximation is only valid when the object is front-lit, so when the light source is coming from behind the object, we clamp w (corresponding to the angle between the light direction and the surface normal) to 0 and gradually transition the reflectance itself to 0 as the light gets further behind the object. As with the specular technique (or traditional PTM), the original photographs are not required for real-time rendering, so both the amount of graphics memory used and the time to render each frame are constant with respect to the number of images originally available for optimization.

The test subject used for this study was a jade object from Mia's collection: *Traveling in Autumn Mountains* from the 17th-18th century CE. We used an off-the-shelf 3D photogrammetry tool (Agisoft Metashape), which successfully estimated the camera locations and built a usable 3D model, despite the non-Lambertian reflectance of the object's surface. At the same time, when standard rendering techniques were applied to the color texture produced by Metashape, important appearance details related to how light passes through the translucent object were lost due to the translucency of the jade material.

The results of our PTM technique are shown in **Figure 5**, alongside the results using the Metashape texture. The PTM rendering has softer shading that better matches the translucency seen in the source photograph than the color texture map produced by Metashape. A screenshot of the model in the 3D viewer is also shown along with a catalogue image of the object for reference.

Use of shaders and lighting in Unity

Both the specular representation and the PTM / RTI format used for translucent materials can be efficiently rendered in real-time using custom shaders. The flexibility of being able to use custom shaders stands in contrast to most existing viewers, which only offer the ability to specify a color texture map along with possibly normal and specular maps. This is a format that makes sense in the entertainment industry where many assets are hand-drawn by artists and fidelity is less important than flashiness. For cultural heritage, however, formats like PTM or specular basis functions make more sense as a direct compression of real-world information from photographs. Use of these formats improves the resemblance of the virtual 3D rendering to the original physical object by mathematically optimizing similarity to the original photographs within the constraints of the chosen representation. This eliminates the need to resort to a "guess-and-check" approach with specularity / shininess or translucency sliders, or "magic" tools for "generating" specular textures from just one color texture.

Since there will probably never be an exhaustive list of all possible formats useful to cultural heritage, allowing for custom shaders makes sense as a means of support for heritage-oriented representations of material appearance. These shaders could be archived with the data itself and can, in a sense, be thought of as a form of metadata: any software application can use a shader to interpret the contents of the appearance data package produced by an optimization or compression technique like the ones discussed and used in this work (i.e. specular basis or PTM).

For lighting, we created a studio or gallery-like environment in Unity with multiple dynamic light sources. Rather than generating pre-baked light maps for each model, we opted to apply Unity's dynamic lighting in real time. This would theoretically make it possible to update which models are available after build time. In future work, we plan to create a secondary program to help import 3D models into the viewer's collection of objects, which would allow cultural heritage professionals to add new models at any time without even opening the Unity editor.

The use of Unity as an engine also enables advanced lighting features such as soft shadows, global illumination, and the like. These capabilities lead to more realistic images without any additional effort on the designers of the virtual exhibit or experience.

Results and Discussion

Our goal for this project was to create a 3D model viewer that supports custom shaders and state-of-the-art lighting. Unity was an ideal choice for achieving these goals. We also wanted the usability of the viewer to match other programs like Sketchfab and we wanted the program to be embeddable in a web browser and deployable without third-party services. We successfully created a deployable WebGL build of the model viewer using artifacts from Mia. We have published a build of the viewer on itch.io with some demo objects from Mia[3]. The source code is on GitHub (see footnote on page 1) so anyone may explore it and create their own extensions of the project. Similarly, the implementations of the specular basis and PTM optimization techniques that were used to build the appearance data packages for each object are housed within the IBRelight code base[4], also a publicly available open-source repository (although it should be emphasized that this functionality has not been thoroughly documented and is not quite ready for public use).

The harder question to answer is whether the improved shading and lighting offered by Unity provide a higher degree of engagement to users of the application when compared to other heritage-focused 3D model viewers. To get a preliminary user experience assessment of the program, we had several meetings with the photographers from Mia to get direct feedback, which was generally focused on feature requests. Through these engagements, we were able to get feedback on existing features as well as requests for what would be useful for them as end users. As an example, one of the features that was near the top of the list was the ability to include annotations associated with 3D points on the object; a basic prototype of this feature is included in the current build of the 3D viewer. In future work, we plan to perform a more rigorous study of the impact of higher quality lighting and shading for end users in the general public.

Another direction for future work is to release the tools used to optimize the specular and PTM surface appearance models, for use by museum professionals. We also hope to support an authoring platform (possibly Smithonian's Voyager Story); currently, the models are hardcoded into the viewer app, but we hope to eliminate this limitation in the near future. Scenes from the authoring platform would ideally be loaded in dynamically so that museum professionals could add exhibits to the viewer without needing to install or run the Unity editor.

It is worth noting that we have intentionally chosen not to provide a platform for file hosting. This allows the museum to use their own internal solution for file hosting, reducing dependency on third party servers such as Sketchfab for their model hosting. This infrastructure also provides security in how the museum's resources are deployed by keeping control of the content in the hands of the museum. (As with all content delivered over the web, it is possible for users to take the raw data and use it for their own purposes – but institutions would at least have control over the source of the data.)

Conclusion

The use of Unity's rendering pipeline in this project puts our viewer in a unique position for future development. The implementation of custom shaders on top of the underlying power of the Unity engine allows for the possibility of larger virtual gallery projects with a higher degree of realism. Although not yet demonstrated by our prototype, this format also allows for more variety in environments than the typical gray void of 3D model viewers, which is where the real time lighting and shaders would really shine. The ability to view artifacts under a variety of lighting scenarios would offer users a deeper level of control and interaction, bringing them closer to the artifact with the ability to manipulate its environment. Alternatively, a professional 3D artist could construct an environment for the object in the Unity engine that places it in its original historical context, enhancing the educational experience much like a well-designed gallery, but without all the limitations of a physical space. We hope that our prototype is but the beginning of an exploration of the many ways that the power of real-time 3D engines can enhance the future of cultural heritage storytelling.

Acknowledgements

This work was produced in part with funding provided by the McNair Scholars Program at the University of Wisconsin – Stout.

Special thanks to Dan Dennehy, Charles Walbridge, and the Visual Resources team from the Minneapolis Institute of Art, who photographed all of the objects shown in the app and produced the 3D models from photogrammetry.

The 3D viewer app also benefited from the contributions of software developer Anton Nikitin.

References

[1] Unity Technologies, "Unity Real-Time Development Platform | 3D, 2D VR & AR Engine." https://unity.com/ (accessed Mar. 17, 2022).

[3] https://tetzlaff.itch.io/ch-3d-model-viewer

[4] https://github.com/michaelt919/IBRelight

[2] R. Scopigno *et al.*, “3D models for cultural heritage: beyond plain visualization,” *Computer*, vol. 44, no. 7, pp. 48–55, 2011.

[3] A. Yasser, K. Clawson, C. Bowerman, M. Lévêque, and others, “Saving cultural heritage with digital make-believe: machine learning and digital techniques to the rescue,” in *HCI’17: Proceedings of the 31st British Computer Society Human Computer Interaction Conference*, 2017, no. 97, pp. 1–5.

[4] N. Statham, “Scientific rigour of online platforms for 3D visualization of heritage,” *Virtual Archaeology Review*, vol. 10, no. 20, pp. 1–16, 2019.

[5] A.-M. Boutsi, C. Ioannidis, and S. Soile, “An integrated approach to 3D web visualization of cultural heritage heterogeneous datasets,” *Remote Sensing*, vol. 11, no. 21, p. 2508, 2019.

[6] T. Flynn, “What happens when you share 3D models online (In 3D)?,” *J. Grayburn, Z. Lischer-Katz, K. Golubiewski-Davis, & V. Ikeshoji-Orlati (Eds.) D*, vol. 3, pp. 73–86, 2019.

[7] M. Potenziani, M. Callieri, M. Dellepiane, M. Corsini, F. Ponchio, and R. Scopigno, “3DHOP: 3D heritage online presenter,” *Computers & Graphics*, vol. 52, pp. 129–141, 2015.

[8] F. Ponchio, M. Potenziani, and M. Callieri, “ARIADNE Visual Media Service: easy web publishing of advanced visual media,” *CAA2015*, vol. 1, pp. 433–442, 2016.

[9] Smithsonian Digitization, “Smithsonian Voyager: Open source 3D explorer and authoring tool suite.” https://3d.si.edu (accessed Mar. 17, 2022).

[10] R. Scopigno, M. Callieri, M. Dellepiane, F. Ponchio, and M. Potenziani, “Delivering and using 3D models on the web: are we ready?,” *Virtual Archaeology Review*, vol. 8, no. 17, pp. 1–9, 2017.

[11] K. Giles, A. Masinton, and G. Arnott, “Visualising the Guild Chapel, Stratford-upon-Avon: digital models as research tools in buildings archaeology,” *Internet Archaeology*, vol. 32, 2012.

[12] H. Richards-Rissetto and J. von Schwerin, “A catch 22 of 3D data sustainability: Lessons in 3D archaeological data management & accessibility,” *Digital Applications in Archaeology and Cultural Heritage*, vol. 6, pp. 38–48, 2017.

[13] B. Bissell, M. Morris, E. Shaffer, M. Tetzlaff, and S. Berrier, “Vessel: a cultural heritage game for entertainment,” in *IS&T Archiving*, 2021, pp. 1–6.

[14] 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, 2017.

[15] P. Clini, L. Ruggeri, R. Angeloni, and M. Sasso, “Interactive immersive virtual museum: digital documentation for virtual interaction,” *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. XLII–2, pp. 251–257, 2018.

[16] H. Rahaman, E. Champion, and M. Bekele, “From photo to 3D to mixed reality: A complete workflow for cultural heritage visualisation and experience,” *Digital Applications in Archaeology and Cultural Heritage*, vol. 13, 2019.

[17] A. Jääskeläinen, “The dynamic virtual museum and promoting wellbeing,” in *Xamk Beyond 2021. Sustainable Development and Social Responsibility*, 2021.

[18] M. Tetzlaff and G. Meyer, “Image-based relighting using environment maps,” in *IS&T Archiving*, 2017, pp. 23–27.

[19] M. Tetzlaff and G. Meyer, “IBRelight: an image-based 3D renderer for cultural heritage,” in *IS&T Archiving*, 2018, pp. 93–98.

[20] M. Tetzlaff, “Image-Based Relighting of 3D Objects from Flash Photographs,” University of Minnesota, 2019. Accessed: May 24, 2021. [Online]. Available: http://conservancy.umn.edu/handle/11299/206362

[21] G. Nam, J. H. Lee, D. Gutierrez, and M. H. Kim, “Practical SVBRDF acquisition of 3D objects with unstructured flash photography,” *ACM Transactions on Graphics*, vol. 37, no. 6, p. 267:1-267:12, 2018.

[22] M. Zhang and M. Drew, “Efficient robust image interpolation and surface properties using polynomial texture mapping,” *Journal on Image and Video Processing*, vol. 2014, no. 1, p. 25, 2014.

Author Biographies

Tyler Garcia is a recent graduate of the University of Wisconsin – Stout with a B.S. in Computer Science.

Zhangchi Lyu is a recent graduate of the University of Wisconsin – Stout with a B.S. in Computer Science. His research interests include artificial intelligence and computer graphics.

Michael Tetzlaff is an Assistant Professor of Computer Science at the University of Wisconsin – Stout. He received his B.S. from Bethel University in St Paul, and his M.S. and Ph.D. from the University of Minnesota. His research interests include image-based computer graphics techniques and especially their applications in cultural heritage.