

Improved Appearance Rendering for Photogrammetrically Acquired 3D Models

Seth Berrier*, Michael Tetzlaff†, Michael Ludwig† and Gary Meyer†

*University of Wisconsin – Stout, Department of Mathematics, Statistics and Computer Science

†University of Minnesota, Department of Computer Science and Engineering

Abstract—In this paper we present a technique for significantly improved rendering of objects scanned using photogrammetry techniques. We demonstrate the connection between photogrammetry and the unstructured lumigraph, a surface light field representation and rendering algorithm. We use our lumigraph rendering software to demonstrate improved results from photographically acquired cultural heritage artifacts. We discuss our improvements to the rendering of unstructured lumigraphs on modern hardware and offer this rendering tool as free, open source software.

Keywords—Photogrammetry, 3D Scanning, Light Field Rendering, Material Appearance.

I. INTRODUCTION

In recent years there has been increased interest and research in 3D scanning for preservation and curation. This is particularly true of techniques that employ photogrammetry and computer vision. At present 3D scanning serves the academic goals of curation fairly well, including interpretation and evaluation of subjects and original research. It does this primarily by simplifying information exchange and collaboration between museums and other cultural institutions. A curator with very specific expertise can make simple evaluations of an object without physical access to that artifact. Digital representations also afford objective analysis and comparison [1] and hold the potential to expand curatorial research outside of its traditional boxes.

Despite all the success of 3D scanning, one key problem remains: the appearance of the object is not accurately represented. The flat 2D color that is extracted by modern photogrammetry tools only gives a small glimpse of how the material might have behaved in person. Our minds are quite adept at gleaned a wealth of information about an object by simply observing subtle interactions with light and shadow. With current digital artifacts, this play of light and shadow is so simplified as to lend almost no information beyond the shape (which is already well documented by the extracted surface model). This limits their utility for other curatorial tasks that require subtle judgement based on fine details and reflective properties. It also excludes them as a reasonable candidate for preservation and archiving of artifacts when they fall so short of the real thing.

However, we observe that when scanning with photogrammetry tools, there is a wealth of additional information available that is presently being ignored. Information about angular reflectance is well encoded by all the images that are used as input to the photogrammetric process. From various angles,



Fig. 1. An ancient Chinese Yung-cheng bell made of bronze.

Left: results exported from PhotoScan showing only a diffuse texture

Right: the same data viewed in our lumigraph renderer with significantly improved material appearance properties.

both subtle and striking differences in light and shadow are necessarily visible. These differences are deliberately removed by most photographic scanning programs as they would add noise to the resulting surface and this results in discarded information.

In this paper, we describe how to use this additional information by returning to the source images and rendering an object acquired through photogrammetry as an unstructured lumigraph. By sampling the original photographs of the object and treating them as samples of a light field, a dramatic improvement in rendered appearance is possible. We must borrow from another, related computer vision and graphics area known as light field rendering to realize this improvement. Together, tools for photographic scanning and light field rendering (specifically the unstructured lumigraph representation) yield a significantly improved digital artifact representation that conveys a wealth of information beyond a standard 3D model with a color texture (see Figure 1).

In the remainder of this paper, we will examine the state-of-the-art of photogrammetry and light field rendering focusing on cultural heritage applications. We then show how to connect photogrammetry and light field rendering with the use of the

unstructured lumigraph representation. We demonstrate our own unstructured lumigraph rendering software that includes improvements over previous approaches. We discuss several examples and show improvements over the diffuse color of a standard model produced by the same software. Finally, we discuss the implications this may have on wider curation techniques and offer the release of our unstructured lumigraph renderer as free, open source software.

II. BACKGROUND

Photography-based techniques are often utilized in cultural heritage applications to help record artifacts. Sometimes traditional photography can suffice; this is particularly effective when presenting a collection online, such as the *Google Cultural Institute*. This can be extended by taking numerous, structured photographs of the relic, either by varying the orientation of the object, moving the camera, or moving the light. One of the earliest examples of this is the QuickTime VR system [2] where the object is rotated about an axis at fixed intervals. Moving through the images simulates the object rotating, but interaction is restricted to the views that match the original photographs. Approaches like this have been used to create cultural educational tools, such as the Lombards on a tablet app [3].

Polynomial texture mapping [4]–[6], or reflectance transformation imaging (RTI), uses a fixed camera and a moving spot light in each photograph. Instead of allowing an object to be interactively rotated, RTI allows a model to be fit to the lighting data allowing the object to be lit from novel lighting directions. This approach works best for flat objects that can be held in focus easily. It is effective for capturing paintings and bas reliefs at a low cost offering different interactivity than QuickTime VR.

The science of photogrammetry (the measurement of objects from photographs) has a long history that predates modern digital computers. Growing out of its original use in surveying and reconnaissance, it now represents an alternate approach for digitizing cultural objects. The lighting is held relative constant with respect to the object while many views are captured. By relying on computer vision techniques the full 3D geometry can be reconstructed. Modern photogrammetry tools leverage a variety of classic vision and graphics algorithms [7]–[11]. Many tools provide a high-resolution surface texture that captures the spatially-varying color of the item. Photogrammetry works particularly well in general lighting with diffuse materials that are not shiny or cause reflections of the environment. There is significant ongoing interest in the use of these techniques and tools for cultural heritage interests with support from organizations like the *Smithsonian Institution*, *CyArk*, and *Cultural Heritage Imaging*.

Bidirectional texture functions (or BTFs) [12] are an approach that can be considered as the hybrid of photogrammetry and RTI. An artifact is photographed from many points of view *and* many different lighting directions simultaneously. BTFs can model complex 3D geometry as well as photogrammetry, and preserve the full reflectance properties of the materials as well or better than RTIs. However, they require structured sampling of both viewing and lighting directions necessitating complex gantry systems and increased

time and storage. Schwartz et al. [13] have demonstrated the effectiveness of bidirectional texture functions when applied to cultural heritage. Linear light source reflectometry [14] and specialized continuous spherical harmonic lighting [15] also rely on specialized gantry systems to make the required capture process reasonable. All three of these methods require substantial monetary and time investment that may make them difficult for institutions to adopt.

Light fields or lumigraphs are another option for digitizing cultural heritage artifacts. They are a compromise over the BTF approach where the lighting remains fixed but significantly less data has to be captured. Light fields model the 5D plenoptic function that describes how light moves through an environment. The initial formulation for a light field relied on a rectangular grid of lenses to approximate the plenoptic function [16], [17]. This approach allowed for small translations in the view point and refocusing of images. Capturing light fields of this form is now easily done with a single shot by using a light field camera such as the *Lytro*. Surface light fields in various forms [18]–[21] reparameterize the light field directly on the surface of the objects within the scene. Unstructured lumigraph rendering, the approach we build upon in this paper, only needs an approximate description of the surface and operates on the uncompressed photographs directly.

Buehler et al’s unstructured lumigraphs can be considered an advanced form of view-dependent texturing [22]. An unstructured lumigraph consists of an approximate shape of the object and a set of photographs of the object. In addition the photographs are annotated with information on the camera’s position and orientation relative to the object. This allows each photograph to be projected onto the approximate shape and blended together appropriately based on the interactive viewing position. Because the blended photographs are view-dependent, the resulting rendering accurately conveys glossy materials, although it is limited to how that glossy material looked while the photographs were taken. As we demonstrate in later sections of this work, when combined with photogrammetry, unstructured lumigraph rendering drastically improves the appearance of the digitized artifact.

Palma et al. [23] demonstrate the approximation of material properties of artifacts from frames of a video sequence which has similar applications as our work. They later extend this to approximate a surface light field using a spherical harmonic fitting [24]. Both of these works offer a straightforward capture process but end up fitting the material appearance properties to a model. Our own approach avoids this fitting stage and maintains a closer connection to the input image data by using it directly. Also, by coupling with the photogrammetry process, our approach uses the same images for geometry approximation and light field rendering.

III. BRIDGING PHOTOGRAMMETRY & LUMIGRAPHS

To render a lumigraph, more than just photographs are required (see Figure 2). It is necessary to know exactly where the camera was located and oriented for each view (referred to as the camera’s *extrinsic* properties). It is also necessary to account for how the camera projects and distorts the image (the camera’s *intrinsic* properties). Finally, an approximation of the object’s surface is needed as a proxy for rendering.

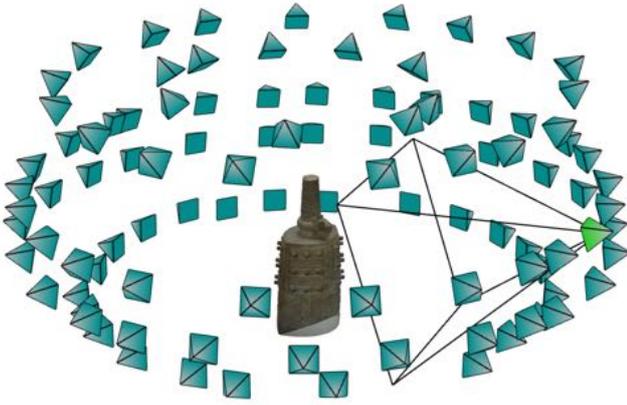


Fig. 2. All the extra information needed to render a lumigraph. In the center is a triangle mesh computed by PhotoScan entirely from uncalibrated photographs. The extrinsic properties of the photographs (position and orientation) are represented by the pyramids surrounding the object.

Commercial photogrammetry tools (like *Agisoft PhotoScan*, *Eos Systems PhotoModeler* and *Autodesk Recap 360*) do not require this additional information, instead, they produce it. These tools start only with photographs of the object taken from many angles, ‘uncalibrated’ and taken directly from the camera into the program. These tools use a bootstrapping process where the same images that serve to reconstruct the object shape can be used to calibrate the camera model. Consequently, photogrammetry tools produce more than just a surface model with color; they also produce intrinsic and extrinsic camera properties. While these photogrammetry tools are robust, the camera parameter estimation process can fail to converge. For best results, care must be taken to control the environment around the object as well as all the settings of the camera, especially when the viewing angles used are unstructured. Regardless, we have been successful using this unstructured approach with unskilled photographers.

When photogrammetry and lumigraph rendering are brought together, it is easy to see how one compliments the other (see Figure 3). The output of photogrammetry serves to fill in the missing data for an unstructured lumigraph which provides superior rendering results from the same information. It is important that the photogrammetry software offers the camera calibration information for export (minimally the extrinsic properties of the camera for each input view of the object). Most will offer this information in one form or another but some make it much easier to retrieve by offering explicit export options and a variety of formats. We found that Agisoft PhotoScan is superior in this regard among the commercial options. PhotoScan offers many options for exporting both the extrinsic and intrinsic properties of the camera in each view. PhotoScan will also ‘undistort’ the photos automatically eliminating the need to invert and apply distortion parameters to the images. The free, open source tools *Bundler* and *CMVS* (and the growing body of front-end applications that simplify their use) are also a viable option.

The Full Scanning Process

One key advantage of photogrammetry techniques is that they do not require expensive equipment. One person with a

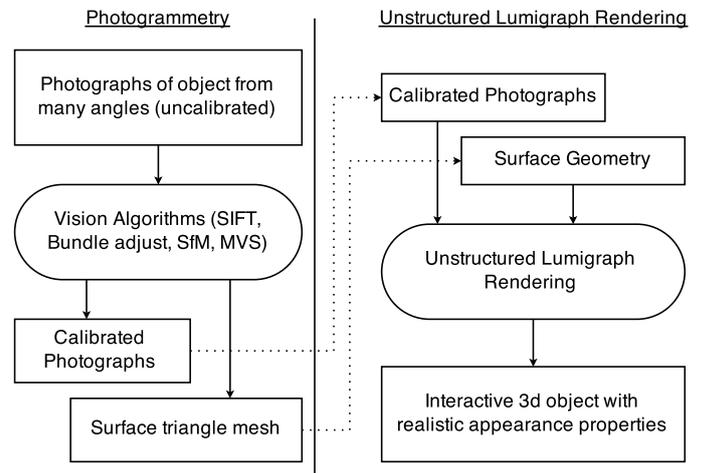


Fig. 3. Left: the flow of data in a photogrammetry program (simplified). Right: data flow for rendering unstructured lumigraphs. Photogrammetry provides precisely the missing information needed for a lumigraph.

decent camera and tripod (and a few hours time) can capture all the input data needed. While unstructured viewing angles and a noisy background environment can present challenges to convergence of the photogrammetry steps, some relatively simple best practices help minimize these problems. The same best practices will maximize the effectiveness of rendering the results as a lumigraph. In this section we describe the end-to-end process from photographs to lumigraph emphasizing any practices that can help with convergence and a superior end result.

Preparing the environment: You should begin by preparing the object and environment as you would for product photography providing a high-contrast background wherever possible (black or white). Light needs to fill in shadows from all sides as any area of the object that is in shadow will be indiscernible by the photogrammetry process and cause noise. Unlike photogrammetry lighting, where saturating the room in diffuse light is often recommended, having some structure to the light is preferred for a lumigraph result. With photogrammetry the governing principle is that all shape must be visible to be extracted from the photos. However, taken too far this would obscure material properties. With lumigraphs there is the additional principle that all *material appearances* should be visible to be rendered in the final result. In particular, if the object has a shiny finish, or a metallic or satin sheen, a few stronger lights should be placed to catch that sheen and produce a specular highlight from several angles. Flooding the object with diffused light can often have the effect of washing out these highlights causing the final result to appear dull and unrealistic. Expensive studio lighting is not necessary for good results (see Figure 4). A collection of inexpensive consumer lamps combined with consistent light bulbs work well both for photogrammetry and lumigraph rendering.

Capturing photos: As with the lighting, an expensive, high-end camera is not necessary. There are a few features that are very useful, mostly for the photogrammetry process:

- Manual controls to hold exposure constant.
- An APS-C or larger sensor.



Fig. 4. The basic components of a capture environment. Ideally there should be several lights filling in all the shadowed areas on the object. The camera should be moved as opposed to the object.

- The ability to export losslessly compress images.

The actual capture process is essentially identical to that used for photogrammetry:

- Set the aperture stopped down to increase depth of field¹ then find a shutter speed that works for the brightest angle.
- Use a slightly increased ISO (like 400) to avoid excessively long shutter speeds.
- Photos are taken in the following pattern:
 - In a circle around the object at 10 degree intervals.
 - A full circle is made at eye level as well as two levels above and two below.
 - The highest and lowest levels can take bigger steps around the circle (30 degree increments).

This process should yield between 90 and 120 photos depending on how precise you are with your steps. While decent photogrammetry results can be achieved with fewer views than this, we have found that this higher number of views gives better results when rendering as a lumigraph (especially if the object is somewhat shiny). When an object has high gloss or a shiny finish, the specular highlight will not travel properly across the surface if the viewing angle is under sampled. Instead, it will seem to jump from view to view with no smooth transition. The need for higher sampling of viewing angles for accurate lumigraph rendering results is a well studied limitation of all light field rendering techniques and an active area of research.

When capturing for photogrammetry, it is important to have views that are well above and below eye level. Without these views, the top and bottom of the object will only be seen at grazing angles making it difficult for the software to discern their distance from the camera. These views are even more important when rendering as a lumigraph. Without high and low angle views the object will look very blurry or even disappear when rotated to these angles in the lumigraph renderer.

¹We find in practice, the images are processed well below the native resolution of the camera sensor (to minimize computation time and memory constraints). At these lower resolutions the system is not diffraction limited even at a high aperture setting (such as f22).

Rotating the object: While the above process is recommended for best results as a lumigraph, it is possible to simplify everything significantly by rotating the object instead of the camera. It is worth noting that what is rendered in the end is not a proper lumigraph/light field as the lighting will be changing from view to view (since the lights do not rotate with the object). Despite this, the end results are quite pleasing and informative and the authors were surprised by how effectively the material properties are communicated with this approach. Most of the cultural heritage examples shown in Section V were captured in this manner.

IV. RENDERING UNSTRUCTURED LUMIGRAPHS

Basic Process: We have improved upon the unstructured lumigraph approach of Buehler et al. [20] for our rendering algorithm. Their approach was inspired by Debevec et al.’s “view-dependent texture mapping” [22]. In view-dependent texture mapping, a geometric mesh is rendered with “projective texture mapping,” where rather than having a 2D texture image wrapped over the object, the texture is projected much like a real projector onto the object’s geometry. Both Debevec et al. and Buehler et al. repeatedly apply projective texture mapping with multiple views of the object in order to achieve full coverage of the 3D geometry. Each view is assigned a weight for every pixel in the final rendering, which is used to blend with the other views in the dataset. Buehler et al. note that as the number of views grows, the technique becomes a legitimate light field rendering approach as each view can be thought of as a sample of the light field, with an appropriate interpolation scheme in between.

The original unstructured lumigraph renderer computed the top k views for each vertex of the geometry on the CPU and then transferred that data to the GPU every time the user adjusted their viewing position. This implementation was in part determined by the computational limits at the time, and was also designed to minimize ghosting with imprecise geometry. For cultural heritage applications, however, modern photogrammetry techniques can compute very accurate geometric models with many vertices, which would be quite inefficient to render with the original implementation. Modern hardware is also able to store much more information on the GPU, so it is preferable to minimize the amount of per-frame CPU computation. Our software addresses these issues through several optimizations outlined below.

Optimization for Modern Hardware: With modern graphics cards, it is no longer necessary to render one texture at a time, as was the case in the original approach. Modern hardware is capable of storing every view simultaneously in graphics memory at a very acceptable resolution. A typical lumigraph with 2048x2048 images requires less than 3 GB of memory, which is comfortably within the limits of a high-end consumer laptop card such as the NVidia GTX 780M. For lower end graphics cards, the images can be downscaled to 1024x1024 which further reduces the memory requirements to roughly half a gigabyte.

With all imagery available on the GPU, programmatic shaders can be used to compute the blending weights on the fly without needing to perform costly transfers of data between the CPU and GPU. However, some newer GPUs are streamlined to

perform texture lookups with negligible overhead while general computations are more costly. Consequently, the task of finding the “top k views” required for unstructured lumigraph rendering becomes a bottleneck. Instead we develop a new blending function described in the next section.

Computation of Blending Weights: Buehler et al. originally proposed a weighting function based on the angle between the viewing direction of the source camera C_i and the viewing direction of the virtual camera D . That is, they initially assign a weight of 1 for each view and then subtract a penalty proportional to the angle between the source viewing direction and the target viewing direction. The weights are then normalized by dividing each weight through by the sum of the weights. More precisely, given a source camera position C_i , a virtual camera position D , and a surface position p , w_{Buehler} (pre-normalized) is defined as:

$$w_{\text{Buehler}}(i) = 1 - \frac{\text{penalty}(i)}{\text{thresh}} = 1 - \frac{\text{acos}\left(\frac{(C_i-p) \cdot (D-p)}{\|C_i-p\| \|D-p\|}\right)}{\text{thresh}}$$

“thresh” is a threshold for which every view with a higher penalty should have a weight of zero. This allows views which are clearly inferior to be excluded completely. Buehler et al. then comment that to ensure epipole consistency (that is, if the target viewing direction is equivalent to one of the source viewing directions, the source view should be used exclusively), each weight should be divided through by $\text{penalty}(i)$. The resulting weight w'_{Buehler} is then:

$$w'_{\text{Buehler}}(i) = \frac{1}{\text{penalty}(i)} - \frac{1}{\text{thresh}} = \frac{1}{\text{acos}\left(\frac{(C_i-p) \cdot (D-p)}{\|C_i-p\| \|D-p\|}\right)} - \frac{1}{\text{thresh}}$$

The difficulty with this approach is determining how to compute the threshold. Setting the threshold too low may cause some pixels to have a shortage of views which in the extreme case could cause them to not be rendered at all. On the other hand, setting it too high results in inferior views blending with the desired views, making the image blurrier, particularly for lumigraphs with strong specular highlights. Buehler et al.’s solution was to compute a different threshold at every point where the weighting function is evaluated. They find the k nearest source cameras to the desired virtual camera (i.e. the k cameras with the lowest penalties) and set $\text{thresh} = \text{penalty}(k)$,

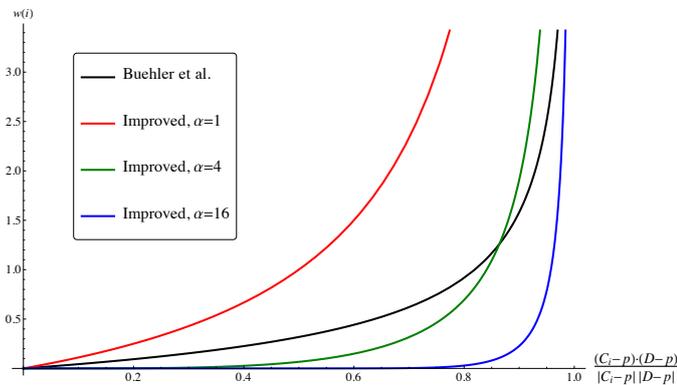


Fig. 5. A plot of Buehler et al.’s blending weight function (with maximum threshold) compared with the improved weight function with various values of α .

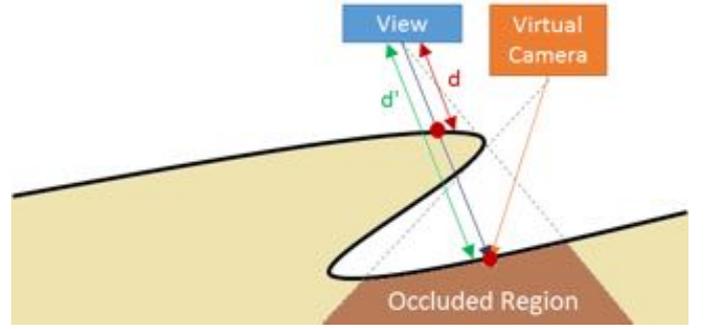


Fig. 6. A diagram indicating how occlusion artifacts can occur and how they can be corrected. The precomputed depth d stored into the depth buffer is compared with the actual depth d' and found to be closer. Thus the view is ignored, eliminating the occlusion artifact.

using the remaining $k-1$ views for rendering. Our solution is different; we instead use a slightly modified weighting function which is not dependent on a per-vertex threshold to reduce ghosting. For simplicity, we use the full set of input views for each pixel, since on modern hardware as many as 100 texture lookups per pixel can be performed fast enough that it is no longer prohibitive.

The first optimization we make is to eliminate computing the inverse cosine, as this is an unnecessary computation, and instead scale the weights using the normalized dot product term directly. Second, we eliminate the hard threshold, essentially allowing the penalty to be unbounded above. We only exclude views where the normalized dot product is less than or equal to zero, since this would correspond to back projection, which we do not want. Finally, to prevent inferior views from contaminating pixels where there are an abundance of good views, we add an exponent which forces the weighting function to asymptotically approach zero faster as the normalized dot product decreases. The effect of this is that in cases where there are plenty of good views, these views dominate, but for cases with a shortage of views, any views available can still be used with valid non-zero weights, within the typical limitations of floating point representations.

Our modified weighting function, w_{Improved} , is then:

$$w_{\text{Improved}}(i) = \frac{1}{1 - \left(\frac{(C_i-p) \cdot (D-p)}{\|C_i-p\| \|D-p\|}\right)^\alpha} - 1$$

Some care must be taken to choose α appropriately. Setting it too low causes inferior views to contaminate the image more substantially, much like setting a high threshold with Buehler’s approach. Setting too high, on the other hand, causes harsher transitions between virtual camera positions. Additionally, in some cases, if α is too high, all the weights may fall below the limits of floating point precision, particularly with sparse datasets. In practice, α should depend on the sparsity of the dataset - for a dense dataset, α should be set higher to maximize image quality, but for a sparse dataset, it should be set lower to smooth out the view interpolation and avoid precision issues. We have found that for all of our datasets, α can be set once for a single lumigraph and provide satisfactory results for all camera positions.

Occlusion Detection: We also address the problem that occurs when the fundamental assumption of 4D light fields is violated. 4D light fields assume that directed lines are free of occluders and have constant luminance. In an implementation based on projective texturing, a feature in the foreground of an input image may be projected onto a surface in the background. If the virtual camera is in a position to view the background, a ghosted image of the occluder will be present on the background. This is because the line from the point on the surface where the feature is projected passes unoccluded through the virtual camera, but the line from the surface point to the source camera intersects the foreground object (see Figure 6). Consequently, the color data in the source view corresponds to the foreground object rather than the background surface.

Buehler et al. briefly consider the issue of visibility, but indicate that they did not find it necessary to address it in their implementation because of the nature of their datasets. However, within the domain of cultural heritage, it is very possible for an artifact to have a protrusion such as a handle on a vase, or a decorative ornament, which causes occlusion issues. There are a number of ways to address this issue. Debevec et al. [22] took an approach where a list of visible views is precomputed for each polygon in the geometry. We chose to use an approach proposed by Wood et al. [18], which is conceptually similar to “shadow mapping.”

The shadow mapping process is to render the scene from the perspective of each virtual light source and for each pixel rendered, store the distance from the light source to the nearest point of intersection along the corresponding ray in a “depth buffer.” Depth buffers are already supported by graphics hardware for the purpose of performing hidden surface removal. When the scene is rendered from the perspective of the virtual camera and shaded based on the light sources, a lookup is performed for each light into the corresponding depth buffer.

We implement the same process as shadow mapping, but we render the scene from the perspective of every source camera and maintain a depth buffer for each view. Then when applying projective texturing we check if the distance stored in the depth buffer is closer than the actual distance from the surface position to the source camera (see Figure 6). If it is, then there must be some other surface in between and we ignore the contribution from that view (setting its weight to zero). If the depth is essentially the same (within floating-point precision considerations), then there must be no closer surface, and the source view is valid. In practice this technique works well for eliminating occlusion related artifacts.

We find in practice that some GPUs with ample video memory and a modern architecture perform better with the approach described above. Using the approach of Buehler et al. was in fact slower on these cards despite the reduction in texture lookups. We achieved interactive rates with all the data sets presented in this paper ranging from a full 60+ FPS down to 15 for the majority of data sets. The frame rate can vary with the resolution of the input data, the number of images and the size of the object in the viewport. The frame rate remains consistent with the majority of objects, the one exception being the fish hat which slowed to 10 FPS when viewed close up. The efficiency of our renderer is an area of active development



Fig. 7. An unstructured lumigraph of the Chinese bell rendered without occlusion detection (left) and with depth-based occlusion detection (right). The objectionable artifacts caused by occlusion are largely eliminated in the image on the right.

and something that will be improved as we optimize and test prior to release.

V. RESULTS

We have worked with several museum and photography partners to capture several objects from their collections. Some of these objects were originally photographed only with Quicktime VR or photogrammetry intentions while others were photographed with lumigraph rendering in mind from the start.

In Figure 9 and the left of Figure 8 two ancient Chinese bronzes (circa 13th and 5th c. BCE) are shown as lumigraphs. These images are generated by our lumigraph rendering software and are the result of photographing the objects on a turntable at several intervals. As described in Section III the objects were first processed with Agisoft PhotoScan to generate the necessary intrinsic and extrinsic camera properties as well as a detailed geometry mesh. Rendering as a lumigraph affords a perception of the shiny metallic reflection still present in the bronze artifacts despite centuries of patina buildup. Note that only the bell was photographed with the intention of rendering it as a lumigraph, the wine vessel was captured with the intention of making a Quicktime VR. A direct comparison between the static texture of the bell and its lumigraph counterpart is presented in Figure 1.

The right of Figure 8 shows a design artifact from a museum collection rendered as a lumigraph. This is a high fashion hat modeling a fish made of satin and metallic ribbons and cord with translucent buttons for eyes. The subtle reflections of the satin ribbon are much more apparent in the lumigraph rendering. This particular object was under highly diffused lighting which washes out the stronger reflective properties but despite this, the object under motion as a lumigraph does convey the subtle reflection.

For all of these examples, even despite being captured without lumigraph rendering in mind, the final results are compelling and convey the materials far better than the typical static color texture produced by photogrammetry tools.

While there are a myriad of options for the photogrammetry part of this workflow there are no software tools for rendering unstructured lumigraphs available for public use. Our lumigraph renderer represents a modernized implementation of the original approach described by Buehler et al. [20]. At present, our tool can be pointed to a PhotoScan PSZ file and will extract and convert the necessary information about camera properties into our own format. Combined with undistorted photos from PhotoScan, the renderings shown in Section V were generated.

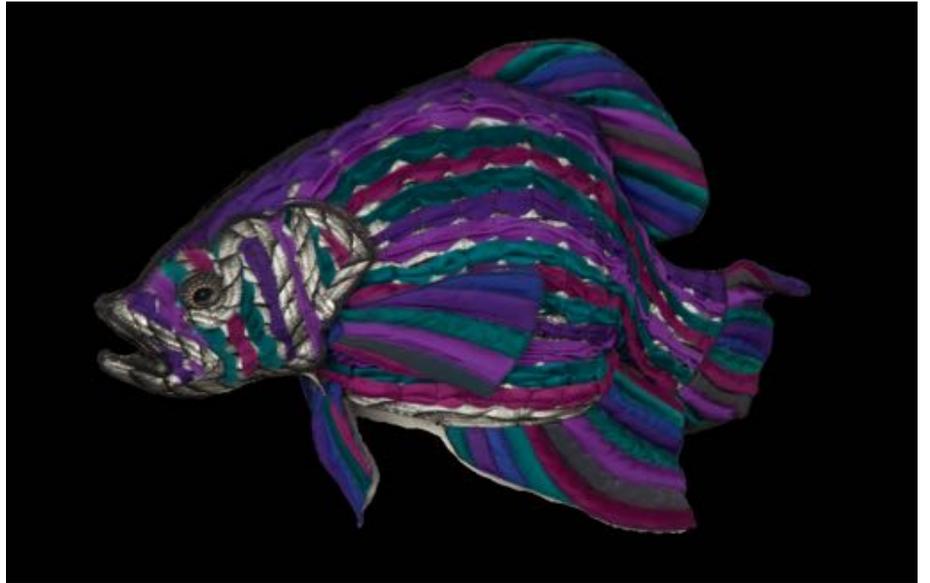


Fig. 8. Two artifacts rendered as lumigraphs. **Left:** A bronze *Chinese Léi ritual wine vessel* from 13th century BCE. **Right:** A modern hat with complex ribbon and cord materials.



Fig. 9. An ancient *Chinese Yung-cheng bell* made of bronze from the 6th – 5th century BCE. Rendered with our software as a lumigraph.

To help support the advancement of this area and empower other researchers and curators we are offering our renderer as free, open source software. A link to more information will be available at Dr. Berrier's web site: <http://www.uwstout.edu/faculty/berriers/>.

VI. CONCLUSIONS

In this paper we have shown that there is a significant connection between photogrammetry and lumigraph rendering. We demonstrate how the original views and calibration data from photogrammetry tools constitute a complete unstructured lumigraph. Our software is able to take this data and present a significantly improved rendering of the object where material properties are communicated much more effectively. We show several examples of the improved results using cultural heritage objects from museum collections that were captured using photogrammetry techniques. Finally, we intend to make our lumigraph rendering software available for public use and continued development.

As 3D scanning and photogrammetry have opened new doors to curation and research in the cultural heritage domain, we believe the improved results afforded by lumigraph rendering have the potential to propel this into a viable digital heritage technique. It is a path to the future of accurate preservation and broad digital access to knowledge and heritage and a large step beyond the current state-of-the-art. As more heritage institutions realize the power of modern photogrammetry and its synergy with light field rendering we hope to empower them to bring this to their communities and the world at large through our efforts described here and our continued work in light field rendering.

ACKNOWLEDGEMENTS

The authors would like to thank the Minneapolis Institute of Arts and staff photographer Charles Walbridge for providing the photographs of the bronze artifacts used as examples in this paper. Additionally we are grateful to the the Goldstein Museum of Design and staff member Ellen Skoro for photographing the hat shown in this work. Finally, we would like to acknowledge and express our gratitude to one of the reviewers who clearly spent a considerable amount of time and thoughtful consideration in their assessment of our work. Their feedback was welcome and exceptionally valuable.

REFERENCES

- [1] A. Bevan, X. Li, M. Martín-Torres, S. Green, Y. Xia, K. Zhao, Z. Zhao, S. Ma, W. Cao, and T. Rehren, "Computer vision, archaeological classification and china's terracotta warriors," *Journal of Archaeological Science*, vol. 49, pp. 249 – 254, September 2014.
- [2] S. E. Chen, "Quicktime vr: an image-based approach to virtual environment navigation," in *Proceedings of SIGGRAPH '95*. New York, NY, USA: ACM, 1995, pp. 29 – 38.
- [3] M. D. D. Padova and M. Pontisso, "The lombards on a tablet, an experiment of 3d visualization with multipart belt-sets," Abstracts of CAA2011, 39th Annual Conference of Computer Applications and Quantitative Methods in Archaeology, 2011.
- [4] T. Malzbender, D. Gelb, and H. Wolters, "Polynomial texture maps," in *Proceedings of SIGGRAPH '01*. New York, NY, USA: ACM, 2001, pp. 519 – 528.
- [5] P. Gautron, J. Krivanek, S. Pattanaik, and K. Bouatouch, "A novel hemispherical basis for accurate and efficient rendering," in *Proceedings of the Fifteenth Eurographics Conference on Rendering Techniques*, ser. EGSR'04. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2004, pp. 321–330.
- [6] M. Zhang and M. S. Drew, "Efficient robust image interpolation and surface properties using polynomial texture mapping," *EURASIP Journal on Image and Video Processing*, vol. 2014, no. 1, 2014.
- [7] J. Wu, Z. Cui, V. S. Sheng, P. Zhao, D. Su, and S. Gong, "A comparative study of sift and its variants," *Measurement Science Review*, vol. 13, no. 3, pp. 122–131, June 2013.
- [8] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment — a modern synthesis," in *Vision Algorithms: Theory and Practice*, ser. Lecture Notes in Computer Science, B. Triggs, A. Zisserman, and R. Szeliski, Eds. Springer Berlin Heidelberg, 2000, vol. 1883, pp. 298–372.
- [9] Y.-m. Wei, L. Kang, B. Yang, and L.-d. Wu, "Applications of structure from motion: a survey," *Journal of Zhejiang University SCIENCE C*, vol. 14, no. 7, pp. 486–494, 2013.
- [10] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1, June 2006, pp. 519–528.
- [11] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," *ACM Trans. Graph.*, vol. 32, no. 3, pp. 29:1–29:13, Jul. 2013.
- [12] K. J. Dana, B. van Ginneken, S. K. Nayar, and J. J. Koenderink, "Reflectance and texture of real-world surfaces," *ACM Trans. Graph.*, vol. 18, no. 1, pp. 1 – 34, Jan 1999.
- [13] C. Schwartz, M. Weinmann, R. Ruiters, and R. Klein, "Integrated high-quality acquisition of geometry and appearance for cultural heritage," in *Proceedings of the 12th International Conference on Virtual Reality, Archaeology and Cultural Heritage*, ser. VAST'11. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2011, pp. 25–32.
- [14] A. Gardner, C. Tchou, T. Hawkins, and P. Debevec, "Linear light source reflectometry," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 749 – 758, July 2003.
- [15] B. Tunwattanapong, G. Fyffe, P. Graham, J. Busch, X. Yu, A. Ghosh, and P. Debevec, "Acquiring reflectance and shape from continuous spherical harmonic illumination," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 109:1–109:12, Jul. 2013.
- [16] M. Levoy and P. Hanrahan, "Light field rendering," in *Proceedings of SIGGRAPH '96*. New York, NY, USA: ACM, 1996, pp. 31 – 42.
- [17] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Proceedings of SIGGRAPH '96*. New York, NY, USA: ACM, 1996, pp. 43 – 54.
- [18] D. N. Wood, D. I. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. H. Salesin, and W. Stuetzle, "Surface light fields for 3d photography," in *Proceedings of SIGGRAPH '00*. New York, NY, USA: ACM, 2000, pp. 287 – 296.
- [19] W.-C. Chen, J.-Y. Bouguet, M. H. Chu, and R. Grzeszczuk, "Light field mapping: Efficient representation and hardware rendering of surface light fields," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 447 – 456, July 2002.
- [20] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen, "Unstructured lumigraph rendering," in *Proceedings of SIGGRAPH '01*. New York, NY, USA: ACM, 2001, pp. 425 – 432.
- [21] M. Eisemann, B. D. Decker, M. Magnor, P. Bekaert, E. D. Aguiar, N. Ahmed, C. Theobalt, and A. Sellent, "Floating textures," *EuroGraphics 2008*, vol. 27, no. 2, 2008.
- [22] P. Debevec, Y. Yu, and G. Borshukov, "Efficient view-dependent image-based rendering with projective texture-mapping," in *9th Eurographics Rendering Workshop*, Vienna, Austria, June 1998, pp. 105 – 116.
- [23] G. Palma, M. Callieri, M. Dellepiane, and R. Scopigno, "A statistical method for svbrdf approximation from video sequences in general lighting conditions," *Comp. Graph. Forum*, vol. 31, no. 4, pp. 1491–1500, Jun. 2012.
- [24] G. Palma, N. Desogus, P. Cignoni, and R. Scopigno, "Surface light field from video acquired in uncontrolled settings," in *Digital Heritage International Congress (DigitalHeritage), 2013*, vol. 1, Oct 2013, pp. 31–38.