applied optics

Multiwavelength depth encoding method for 3D range geometry compression

TYLER BELL AND SONG ZHANG*

School of Mechanical Engineering, Purdue University, West Lafayette, Indiana 47907, USA
*Corresponding author: szhang15@purdue.edu

Received 15 October 2015; revised 18 November 2015; accepted 25 November 2015; posted 25 November 2015 (Doc. ID 252023); published 17 December 2015

This paper presents a novel method for representing three-dimensional (3D) range data within regular two-dimensional (2D) images using multiwavelength encoding. These 2D images can then be further compressed using traditional lossless (e.g., PNG) or lossy (e.g., JPEG) image compression techniques. Current 3D range data compression methods require significant filtering to reduce lossy compression artifacts. The nature of the proposed encoding, however, offers a significant level of robustness to such artifacts brought about by high levels of JPEG compression. This enables extremely high compression ratios while maintaining a very low reconstruction error percentage with little to no filtering required to remove compression artifacts. For example, when encoding 3D geometry with the proposed method and storing the resulting 2D image with Matlab R2014a JPEG80 image compression, compression ratios of approximately 935:1 versus the OBJ format can be achieved at an error rate of approximately 0.027% without any filtering. © 2015 Optical Society of America

OCIS codes: (120.2650) Fringe analysis; (100.5070) Phase retrieval; (100.6890) Three-dimensional image processing.

http://dx.doi.org/10.1364/AO.54.010684

1. INTRODUCTION

Recent advances in 3D scanning technologies have brought about the capabilities to capture high-quality data at very fast speeds [1]. Given such progress, one might consider these technologies to be on the brink of widespread dissemination. One inherent problem that must be further addressed, however, is establishing methods for 3D range data compression that are robust and offer high compression ratios; such methods will ensure efficient storage and fast, high-quality data recovery.

Currently, one conventional storage standard for a single frame of 3D geometry is the mesh format. These formats (e.g., OBJ, PLY, STL) are generic in nature and perform their tasks well, storing the coordinates of each vertex often along with connectivity information. Additional information can also be stored with the mesh, such as a surface normal map and a (u, v) map. Although these formats are able to perform their task of representing a frame of 3D geometry, they also require a large amount of storage to do so. For example, a single 640 × 480 frame of 3D geometry, with only vertex locations and connectivity information, needs about 13 MB of space [2]. For real-time or faster 3D capture systems that want to store or stream each single frame, this large file size becomes an issue.

Given this, other 3D range data compression techniques have been proposed. One such methodology is to encode the raw 3D data in some way such that it can be represented within a 2D image. The information stored within an image's color channels can then be used to recover and reconstruct the compressed geometry. Such approaches are able to take advantage of very well established image formats (e.g., PNG) and the infrastructure built around them.

To compress 3D geometry into a 2D image, one approach is to use the principles of virtual digital fringe projection (DFP) [2–4]. Using the conventions of DFP along with a virtual structured light scanner, this *HoloImage* approach projects fringe images upon the virtual 3D geometry and captures them virtually [3]. The resulting captured fringe images are then packed into the image (e.g., into its color channels) along with any information that may be required to unwrap the phase map between the phase images (e.g., stair image). An additional advantage to using a virtual fringe projection system that converts raw 3D geometry into a 2D image frame is its portability to video storage and streaming [5,6].

Although able to achieve high compression ratios, virtual fringe projection approaches are limited by the spatial resolution of the image. To circumvent this, Zhang proposed a method to directly encode the depth, Z, of 3D geometry directly into the fringe images to be stored within a 2D image's color channels [7]. This method works very well when storing its 2D images in lossless formats yet compression artifacts become difficult to avoid at high levels of lossy storage.

This paper presents a new method of direct depth (DD) encoding, called multiwavelength depth (MWD) encoding,

which uses a different technique to encode within, and recover geometry information from, a 2D image. When stored in a lossless image, the accuracy is identical to the DD method [7] yet boasts a smaller file size. Also, the proposed method improves upon the DD's encoding such that it is not affected as greatly by high levels of lossy compression. Further, DD's encoding technique is limited in its potential depth resolution. MWD's encoding method, however, lifts these restrictions to allow for high-resolution depth encoding. In our experiments, the new MWD method outperformed the current state-of-the-art DD method in terms of file size, recovered data accuracy, amount of postprocessing required, and maximum depth resolution. For example, when the proposed MWD method was used to encode 3D geometry into a 2D image which was then stored using JPEG 80% from Matlab 2014a, the resulting file size for one frame was 30.3 KB. This is approximately a 935:1 compression ratio versus the same data in the OBJ format. In addition, the reconstruction error percent achieved was 0.027% without any postprocessing filtering applied. Further experimental results will be presented to verify the capability of the proposed 3D range compression method.

Section 2 will explain the principles behind the structured light techniques employed, the encoding of the 3D geometry, and the decoding for the recovery of the geometry using the proposed method. Section 3 will present experimental results of the proposed compression method. Section 4 will discuss the results and merits of this paper's method. Lastly, Section 5 will summarize the paper.

2. PRINCIPLE

A. Phase-Shifting Technique for 3D Shape Measurement

The general principles behind the aforementioned 3D range data compression approaches are derived from structured light scanning techniques. Similar to a two-camera stereo-vision method, a structured light method instead replaces one camera with a projector; thus, modern structured light systems are typically comprised of one camera and one projection unit. There are many structured light techniques [8], yet one that has stood out due to its simplicity, speed, and accuracy is DFP. In systems using DFP, patterns are generated which vary in intensity sinusoidally; to generate different patterns, the sinusoid structure can be shifted by some amount. These patterns are then projected sequentially onto some scene to be captured. Figure 1 displays an example of a 3D structured light scanner using a DFP approach. One example merit of phase-shifting techniques is their ability to achieve pixel-by-pixel spatial resolution. Many phase-shifting algorithms have been developed, and a summary of them can be found in this book chapter [9].

With phase-shifting algorithms, the more steps used results in a lowered sensitivity to linear phase-shifting errors [10]; however, with more steps brings a longer projection cycle. Given this, real-time 3D structured light scanning applications will traditionally use a three-step phase-shifting algorithm. The three-step algorithm requires a shorter projection cycle and can still render a high-quality phase. The three fringe images with equal phase shifts to be used within a three-step phase-shifting algorithm can be derived via

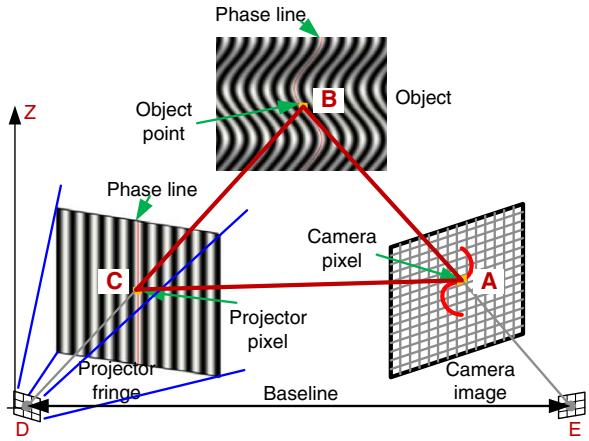

Fig. 1. Example structured light system setup using a digital fringe projection (DFP) technique.

$$I_1(x,y) = I'(x,y) + I''(x,y)\cos(\phi - 2\pi/3), \quad (1)$$

$$I_2(x,y) = I'(x,y) + I''(x,y)\cos(\phi), \quad (2)$$

$$I_3(x,y) = I'(x,y) + I''(x,y)\cos(\phi + 2\pi/3), \quad (3)$$

where I' is the average intensity; I'' is the intensity modulation; and $\phi(x,y)$ is the phase to be solved for. This can be done via

$$\phi(x,y) = -\tan^{-1}\left[\frac{\sqrt{3}(I_1 - I_3)}{2I_2 - I_1 - I_3}\right]. \quad (4)$$

The resulting phase will contain values ranging from $-\pi$ to π with 2π discontinuities. This *wrapped phase*, ϕ, can then be *unwrapped* using a spatial phase-unwrapping algorithm, such as one described in [11], to recover a continuous, unwrapped phase map, Φ. Finally, if the structured light system is calibrated correctly [12], real-world coordinates (X, Y, Z) can be extracted from $\Phi(x,y)$ pixel-by-pixel.

B. HoloImage Encoding

The aforementioned *HoloImage* compression approach uses a *virtual* structured light system (i.e., virtual camera and virtual projection device) to encode the virtual geometry. Being that the resulting system is computer generated, everything can be precisely controlled, including surface reflectivity, ambient light in the scene, and the spatial sampling distribution. As previously mentioned, in real environments, at least three fringe images are required to recover adequate phase; however, given that all of these factors can be controlled in a virtual system, only two fringe images are necessary. Due to the nature of the HoloImage system, the resultant fringe stripes vary horizontally along the x direction and are vertical along the y direction, or vice versa. Since only two images are required to recover phase, a third image can then be used to support the unwrapping of the wrapped phase between the two fringe images. These three images, two fringe images and one stair image, are then packed into each respective channel of a 2D image which can be further compressed with existing 2D image compression techniques (e.g., PNG).

To recover geometry, the fringe images and stair image are extracted from the 2D image's color channels. The wrapped

phase, ϕ, can be computed with the two fringe images similar to Eq. (4). The stair image can then be used to realign the 2π discontinuities such that a continuous phase map, Φ, is recovered. As discussed in Subsection 2.A, (X, Y, Z) can then be recovered. Exact details of the HoloImage approach are covered in [2].

Although able to achieve high-quality compression ratios, HoloImage techniques are not without their drawbacks. One crucial drawback is that HoloImage techniques are limited due to spatial resolution constraints. The fringe images that are virtually projected either vertically or horizontally along the virtual object must have a decent stair height, in practice 10 or more [7], to make the encoded images less sensitive to noise. Using a larger stair height means that there are fewer stairs to use across the image. Given this, dense fringe images are not adequately encoded and therefore high spatial resolutions cannot be reached. To alleviate the constraint of spatial resolution, the depth Z itself can be directly encoded.

C. Direct Depth Z Encoding

To remove the limitations imposed by the spatial resolution of HoloImage techniques, where sampling is done along the x or y direction, the depth Z itself can be encoded. This is achieved with an approach similar to HoloImage except that the sampling takes place along the z direction. Following [7], two fringe images are still used to encode the data with a third image again being used as a stair image. These three images are then stored in the three channels of a 2D image to be further compressed with, traditionally, a lossless format (e.g., PNG). The three encodings to be stored in each channel can be described via

$$I_r(i,j) = 0.5\left[1.0 + \sin\left(\frac{2\pi Z}{P}\right)\right], \tag{5}$$

$$I_g(i,j) = 0.5\left[1.0 + \cos\left(\frac{2\pi Z}{P}\right)\right], \tag{6}$$

$$I_b(i,j) = S \times Fl(Z/P) + 0.5S + 0.5(S-2) \times \cos\left[2\pi \times \frac{\mathrm{Mod}(Z,P)}{P_1}\right], \tag{7}$$

where (i,j) are the image pixel indices; P is the fringe width, or the number of pixels per each fringe stripe; $P_1 = P/(K+0.5)$ is the local fringe pitch where K is an integer; $Fl(x)$ represents the floor function; S represents the gray-scale stair height; and $\mathrm{Mod}(a,b)$ represents the modulus operator.

To recover 3D geometry from the 2D image, the image's different color channels are used. In the direct encoding method X and Y coordinates are uniformly sampled and are therefore proportional to their image indices (i,j). Given this, X and Y can be recovered by some user-specified scaling constant, described as

$$X = j \times c, \tag{8}$$

$$Y = i \times c. \tag{9}$$

The depth map, Z, can be recovered from the stored 2D image via

$$Z = P\left[Fl\left(\frac{I_b - 0.5S}{S}\right) + \frac{1}{2\pi} \times \tan^{-1}\left(\frac{I_r - 0.5}{I_g - 0.5}\right)\right]. \tag{10}$$

The DD encoding technique works quite well [7] and eliminates the spatial resolution constraints imposed by the HoloImage technique; however, the third channel, I_b as given in Eq. (7), limits this approach when high levels of JPEG compression are used. When reconstruction takes place from low-quality JPEG images using this approach, the stair image may degrade, which causes artifacts during phase unwrapping; thus, large filters must be used to remove such artifacts. Also, some ringing artifacts may be imposed due to the cosine-wrapping of the stair image. Further, in both lossy and lossless domains, this technique has a limited depth resolution. The gray-scale stair height, S, constrains I_b as the range of the depth image and the number of stairs used to encode the geometry becomes larger. To alleviate these drawbacks, the third channel needs to be replaced with an alternative that is less sensitive to lossy, JPEG compression and can encode large depth resolutions.

D. Multiwavelength Depth Encoding

This paper presents a novel method for 3D range data compression based on DD encoding and utilizes principles of multifrequency phase-shifting techniques [13]. In the proposed approach, two different fringe widths are used to generate four fringe encodings: one of a shorter, user-defined width, P, and one longer with a width equivalent to the range of Z, $\mathrm{Rng}(Z)$. Mathematically, the encoding of each fringe image can be described via

$$I_r(i,j) = 0.5\left[1.0 + \sin\left(\frac{2\pi Z}{P}\right)\right], \tag{11}$$

$$I_g(i,j) = 0.5\left[1.0 + \cos\left(\frac{2\pi Z}{P}\right)\right], \tag{12}$$

$$I_b^{\sin}(i,j) = 0.5\left[1.0 + \sin\left(2\pi \times \frac{\mathrm{Mod}(Z,\mathrm{Rng}(Z))}{\mathrm{Rng}(Z)}\right)\right], \tag{13}$$

$$I_b^{\cos}(i,j) = 0.5\left[1.0 + \cos\left(2\pi \times \frac{\mathrm{Mod}(Z,\mathrm{Rng}(Z))}{\mathrm{Rng}(Z)}\right)\right]. \tag{14}$$

The first two fringe encodings stored in the red and green color channels are identical to the red and green channels of the DD method. Instead of storing a cosine-wrapped stair image in the third channel, however, the wrapped phase of the two fringes encoded with the longer wavelength is stored and can be computed via

$$\phi_b(i,j) = \tan^{-1}\left(\frac{I_b^{\sin} - 0.5}{I_b^{\cos} - 0.5}\right). \tag{15}$$

It should be noted that the resulting phase map encoded within ϕ_b has a range between $(-\pi, \pi]$. To represent this range instead within a range $(0, 1]$ such that it can be stored in the color channel properly, the following is done:

$$I_b(i,j) = \frac{\phi_b + \pi}{2\pi}. \tag{16}$$

The resulting 2D image can then be stored within a lossless format, such as PNG, or within a lossy format, such as JPEG, at varying levels of compression.

E. Multiwavelength Depth Decoding

To recover geometry from a compressed 2D image, the wrapped phase map between the two fringe images of the shorter wavelength is first computed via

$$\phi_{rg}(i,j) = \tan^{-1}\left(\frac{I_r - 0.5}{I_g - 0.5}\right). \quad (17)$$

Now, as the Rng(Z) was used as the fringe width for the longer wavelength, the encoding of each fringe image generated with this longer width spans the entire depth of the scene. Given this, in the wrapped phase map ϕ_b stored within I_b, there will be no 2π jumps; it will be continuous. This continuous wrapped phase map, ϕ_b, can be used to unwrap the more dense phase map, ϕ_{rg}, which does contain 2π discontinuities.

The first step is to convert the wrapped phase information in I_b back into its original range of $(-\pi, \pi)$ from its current range of $(0, 1)$ via

$$\phi_b(i,j) = I_b \times 2\pi - \pi. \quad (18)$$

Next, a stair image is computed from the continuous ϕ_b, which in turn can be used to reversely unwrap the phase of the shorter wrapped phase, ϕ_{rg}. The stair image is described via

$$K(i,j) = \frac{\phi_b \times \text{Rng}(Z)/P - \phi_{rg}}{2\pi}. \quad (19)$$

The stair image K's values determine the scale of how many 2π must be added to ϕ_{rg} to remove its 2π discontinuities. Removing the phase jumps results in a continuous, unwrapped phase, Φ, of the shorter wavelength. This process can mathematically be described as

$$\Phi(i,j) = \phi_{rg} + 2\pi \times \text{Rnd}(K), \quad (20)$$

which is finally used to recover depth Z via

$$Z(i,j) = \frac{\Phi \times P}{2\pi}. \quad (21)$$

X and Y can be recovered as they were in Eqs. (8) and (9).

3. EXPERIMENTAL RESULTS

In our primary experiments, an ideal, computer-generated semi-sphere was used to verify the performance of the MWD encoding method. After encoding the 3D geometry into a 2D image of 512 × 512 as described in Subsection 2.D, the image was either saved out in a lossless format (PNG) or a lossy format (JPEG) at different levels of compression using Matlab 2014b.

Figure 2 shows each channel and step along the encoding and decoding process of the MWD technique. Figure 2(a) shows the compressed, encoded 2D color image with Figs. 2(b)–2(d) showing the image's individual red, green, and blue color channels, respectively. To recover geometry from the saved out image, the wrapped phase is needed between the fringe images of the shorter wavelength; the resulting wrapped phase is shown in Fig. 2(e). As described previously, the continuous, wrapped phase of the longer wavelength, shown in Fig. 2(d), can be used to derive a stair image, shown in Fig. 2(f). The stair image is then used to unwrap the phase of the shorter wavelength to get the unwrapped phase map, Fig. 2(g), which can then be used to recover 3D geometry, as displayed in Fig. 2(h).

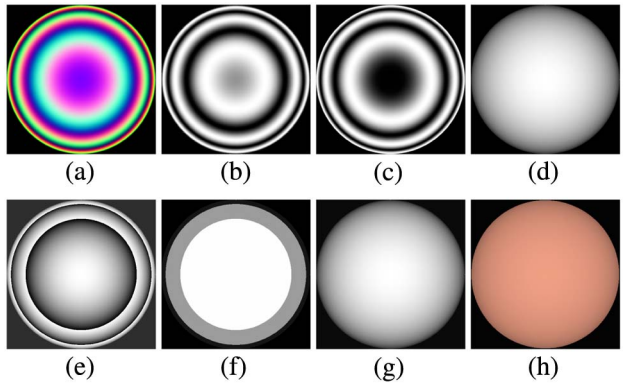

Fig. 2. Experimental results of encoding and decoding an ideal sphere using the multiwavelength depth encoding method (a) encoded 2D color image; (b)–(d) the three encoded color channels, red, green, blue, respectively; (e) wrapped phase between the two shorter wavelength fringe images, (b) and (c); (f) stair image derived from the wrapped phase map of the longer, continuous wavelength, (d); (g) the unwrapped phase map; (h) recovered 3D result.

To help demonstrate the results of this paper's proposed MWD encoding method, comparisons were made with a similar and current state-of-the-art 3D range geometry compression method, the DD encoding [7] technique. It is important to note that all comparisons between MWD and DD were done so on the same geometry and with an identical experimental setup; the only difference being the principle behind the encoding methods themselves. In both cases, a semi-sphere was encoded into a 512 × 512 image using four stairs.

Figure 3 shows comparisons of the visual quality of the recovered data using different JPEG compression levels with each column representing JPEG 100%, JPEG 80%, JPEG 60%, JPEG 40%, and JPEG 20%, respectively, using Matlab 2014a JPEG compression. The top row displays the recovered sphere as encoded via the DD method without any filter applied to recover from JPEG compression artifacts. The second row shows the same DD encoding method with a 25 × 25 filter applied to address compression artifacts. The third row demonstrates the quality of the proposed MWD encoding as no filter was applied to achieve the results. It can be seen that the quality of the recovered geometry using direct depth quickly degrades under JPEG compression and requires a large amount of filtering to recover geometry. MWD on the other hand remains spherical and uniform at high levels of compression without the need for filtering. Even using JPEG 20%, which stores the compressed sphere using only 14 KB for a 512 × 512 image, the recovered data using the MWD approach maintains its original geometric shape and does not require any filtering to do so.

It can be seen visually that the recovered data remains spherical, even at low JPEG qualities. To empirically verify the accuracy of the recovered 3D sphere, a cross section of the ideal data was compared with the same cross section of the decoded data. Figure 4 shows the results for both the DD and MWD approaches where 3D data was recovered from a PNG image as well as JPEG images stored at various levels of compression: 100%, 80%, and 60%, respectively.

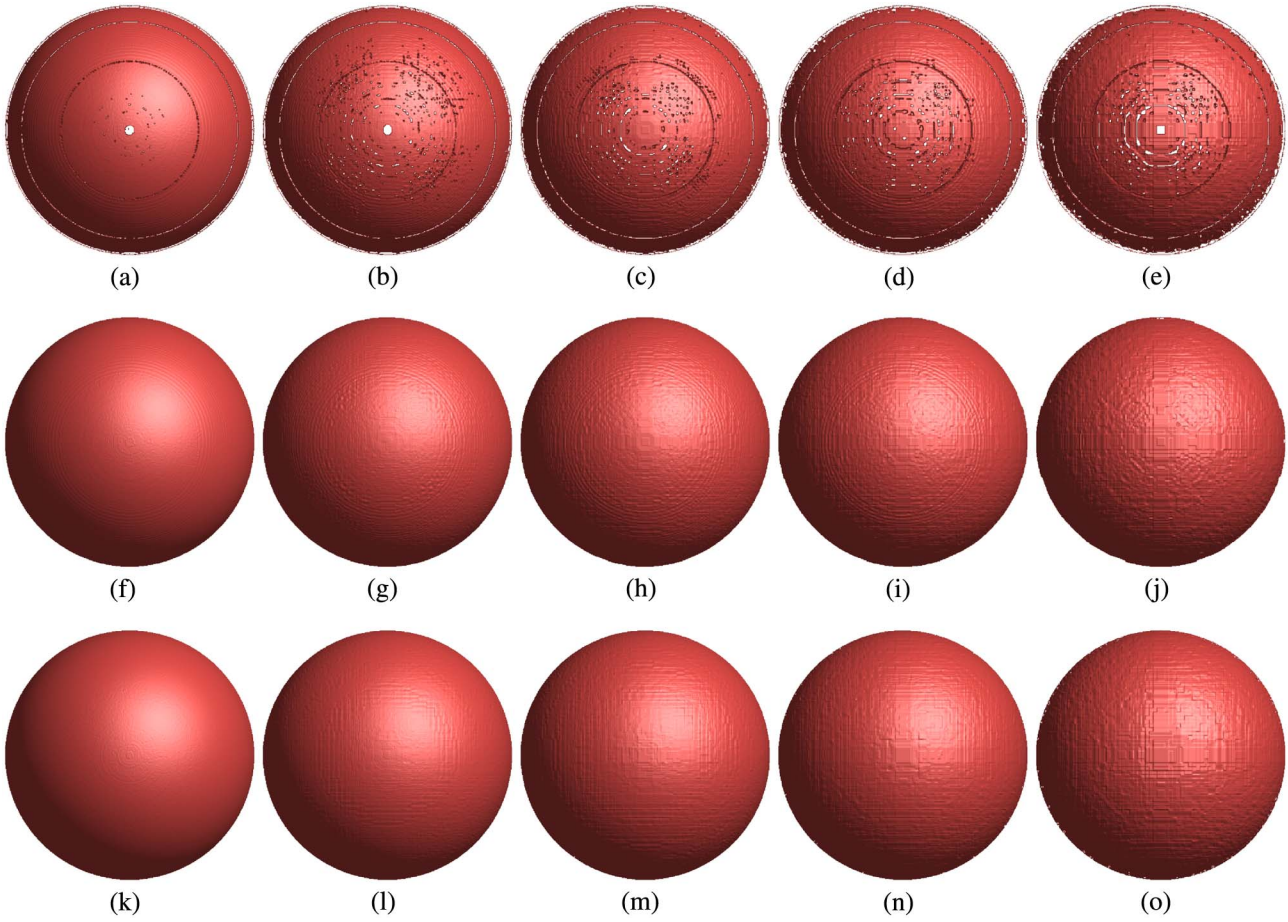

Fig. 3. Visual comparison of recovered sphere geometry using different encoding methods, JPEG compression levels, and filtering amounts. Each column represents JPEG storage levels 100%, 80%, 60%, 40%, and 20%, respectively, using Matlab 2014a JPEG compression. (Row 1) direct depth method without a filter applied to the recovered geometry; (Row 2) direct depth method with a 25 × 25 filter applied to the recovered geometry; and (Row 3) multiwavelength depth method without any filter applied.

The normalized RMS error percentages of the visual results seen in Fig. 3 are given in Table 1; it should be noted that when calculating the error in both approaches the five boundary pixels were ignored to eliminate outliers. From this table, it can be seen that the DD and the proposed MWD approaches are identical when using lossless image compression; however, the accuracies differ when lossy compression is used. This is due to direct depth's third channel, which does not compress well and is the cause of lossy compression artifacts. The continuous nature of the third channel of the MWD approach, however, allows for less JPEG artifacts and thus higher geometry reconstruction accuracies.

To get extremely small sizes (7–10 KB), JPEG qualities can be reduced to 5%–10%. At this very low quality, both approaches begin to exhibit compression artifacts in the recovered data. To remove these spikes from either approach, a median filter is traditionally used. Throughout our experiments with compressing the sphere and other geometries, MWD always required a smaller filter size than DD when recovering data from JPEG encoded images; this results in a faster decode speed for MWD which would have great implications if in use within a real-time streaming environment where compressed 3D frames must be decoded and displayed as they are arriving. For example, as shown in Fig. 5, MWD performs better at the very low JPEG quality of 10% than DD when using the same filter size of 5 × 5. Even the very large filter size of 25 × 25 was unable to rid the DD method completely of artifacts due to such high levels of JPEG compression whereas MWD's 5 × 5 filter proved enough.

Not only does the MWD approach require less filtering, the subject of the filter is different from other state-of-the-art depth compression approaches. Traditionally, what is filtered to remove spikes is the final recovered phase thus affecting the accuracy of the reconstruction. With the proposed MWD approach, however, only the third channel must be filtered in most cases, leaving the red and green channels, the ones containing the encoded geometry data, untouched. This contributes to a higher accuracy of data recovery.

As demonstrated above, the quality, both visually and statistically, of recovered data even at high levels of lossy compression is quite good for the proposed method. This robustness to such low JPEG qualities provides this technique the ability to represent 3D geometry using very little information, thus achieving high compression ratios. Table 2 below shows the

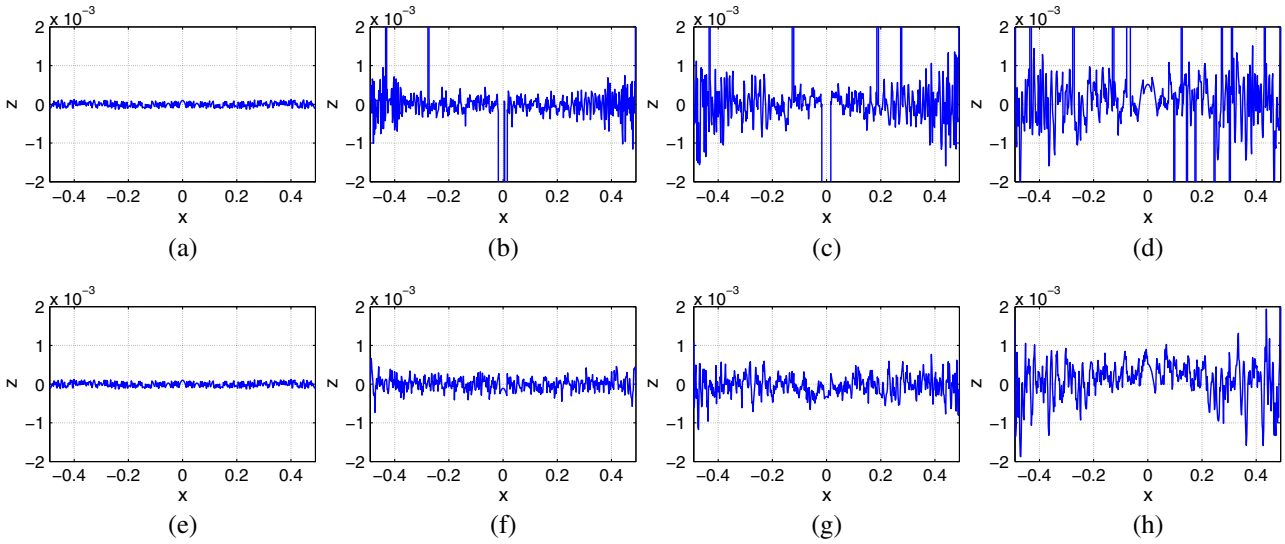

Fig. 4. Experimental results comparing the ideal sphere and the recovered 3D geometry, using DD or MWD, from different qualities of 2D images. Each column represents storage qualities PNG, JPEG 100%, JPEG 80%, and JPEG 60%, respectively; for each plot, the x-axis represents the normalized pixel coordinates along the cross section and the y-axis represents the normalized difference between the ideal and recovered geometries for each coordinate. (Row 1) direct depth cross-section error difference versus ideal geometry. RMS error percentages for (a)–(d) are 0.0061%, 2.9557%, 3.4834%, and 3.397%, respectively. (Row 2) multiwavelength depth cross-section error difference versus ideal geometry. RMS error percentages for (e)–(h) are 0.0061%, 0.0167%, 0.0271%, and 0.0508%, respectively.

Table 1. Comparison of RMS Error between a Normalized Sphere Compressed into a 512 × 512 Image Using the Proposed MWD Encoding Versus the Direct Depth (DD) Encoding[a]

	PNG	JPG100	JPG80	JPG60	JPG40	JPG20
DD No Filter	0.0061%	2.9557%	3.4834%	3.3970%	3.8545%	4.5275%
DD 25 × 25 Filter	0.0061%	0.0263%	0.0438%	0.0578%	0.0725%	0.0924%
MWD No Filter	0.0061%	0.0167%	0.0271%	0.0508%	0.0651%	0.0928%

[a]Associated JPEG levels represent the compression quality at a percentage out of 100 using Matlab 2014a.

ratios of MWD in comparison to popular 3D storage types. In each case, the geometry stored was the ideal sphere. As comparison, Table 3 shows the ratios of using DD. It has now been shown that in general, the proposed method has both larger compression ratios, or smaller data sizes, and lower error percentages.

To ensure that the proposed method worked on complex geometries, a statue was tested. Figure 6 compares the DD and MWD approaches to encoding the same complex geometry, each using four stairs to encode 512 × 512 resolution data into a 512 × 512 compressed image. The results shown in the top row compare DD with a 15 × 15 filter applied and MWD with a 5 × 5 filter applied both using JPEG 95%. The bottom row compares DD and MWD using the same filter sizes except this time using JPEG 40%. It can be seen that the much larger filter does not rid DD's recovered geometry of its compression artifacts whereas MWD's complex geometry is preserved and recovered from this low JPEG quality using only the small 5 × 5 filter.

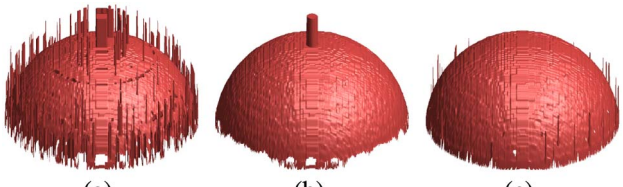

Fig. 5. Comparison of filtered recovered data from JPEG 10% encodings; (a) 3D recovery from DD at JPEG 10% using a 5 × 5 filter; (b) 3D recovery from DD at JPEG 10% using a 25 × 25 filter; (c) 3D recovery from MWD at JPEG 10% using a 5 × 5 filter. The RMS errors for (a)–(c) are 6.13%, 3.93%, and 0.15%, respectively.

Additional experiments were conducted to test MWD's ability to encode high-resolution depth images. Instead of a sphere, geometry of a flat plane with a constantly increasing depth was encoded. It was found that the MWD approach was able to encode large depth ranges, with the largest tested being a 4096 × 4096 depth image having a depth resolution of 4096 (over 16 million points at a 4 K range). Similarly, values of 1024 and 2048 were tested, as well. The MWD method was able to retain high qualities of information after decoding being that up to 256 stair values could be used to encode the depth. The same tests, however, did not bode well for the direct depth approach; as the number of stairs increase, the smaller the step height becomes. The result of this is phase unwrapping issues, which induce 2π-tall spike artifacts in the recovered phase;

Table 2. Compression Ratios of MWD Using PNG and Different JPEG Storage Levels Versus 3D Mesh File Formats

	PNG	JPG100	JPG80	JPG60	JPG40	JPG20	JPG10
OBJ	146.8:1	244.3:1	935.3:1	1294.1:1	1583.3:1	2147.0:1	2834.1:1
PLY	52.0:1	86.5:1	331.0:1	458.0:1	560.3:1	759.8:1	1003.0:1
STL	103.7:1	172.6:1	660.7:1	914.1:1	1118.4:1	1516.6:1	2001.9:1

Table 3. Compression Ratios of DD Using PNG and Different JPEG Storage Levels Versus 3D Mesh File Formats

	PNG	JPG100	JPG80	JPG60	JPG40	JPG20	JPG10
OBJ	91.7:1	187.7:1	708.5:1	1069.5:1	1375.8:1	1968.1:1	2725.1:1
PLY	32.5:1	66.4:1	250.8:1	378.5:1	486.9:1	696.5:1	964.4:1
STL	64.8:1	132.6:1	500.5:1	755.4:1	971.8:1	1390.2:1	1924.9:1

these are unable to be removed with a simple filter. Even with lossless encoding, the direct depth method was unable to encode the plane at a depth resolution of 1024 using 128 stairs, for example. Figure 7's first row shows a normalized small window of the unwrapped phase and the phase's cross section with its continuous slope removed when using the direct depth method to encode a scene with a depth range of 1024 using 128 stairs into a PNG image. The second row shows the same small window and cross section with its slope removed, this time using MWD to encode the scene with the same depth range of 1024 using 128 stairs. As it can be seen, the MWD approach correctly unwraps the phase and can therefore recover proper geometry.

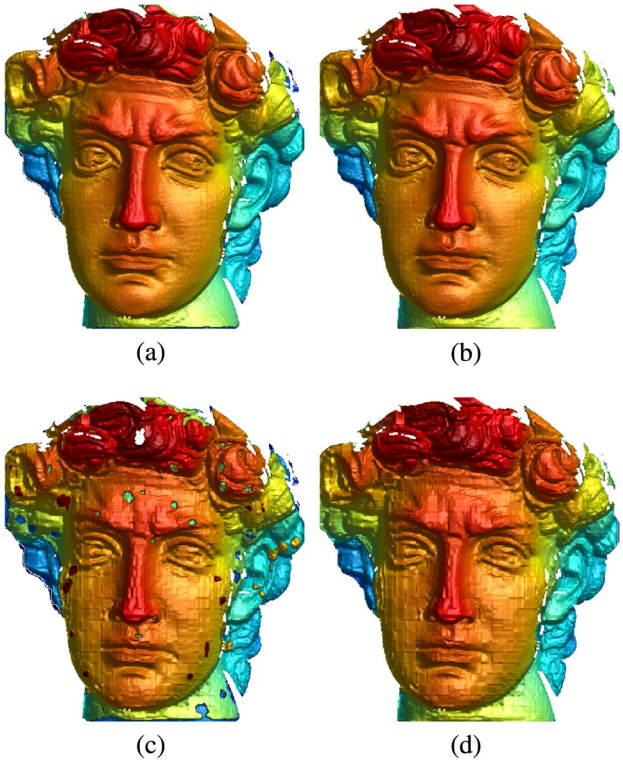

Fig. 6. Encoding complex geometries. (a) DD recovered geometry using a 15 × 15 filter and JPEG 95%; (b) MWD recovered geometry using a 5 × 5 filter and JPEG 95%; (c) DD recovered geometry using a 15 × 15 filter and JPEG 40%; (d) MWD recovered geometry using a 5 × 5 filter and JPEG 40%.

For the direct depth approach to feasibly encode large depth ranges, the number of stairs must be lowered to increase the gray-scale stair height, which results in poor quality reconstruction at high-resolution depths. The paper's MWD approach, however, was able to encode the plane at experimental resolutions of 1024, 2048, and 4096 with upto 256 stairs. Using a 4096 × 4096 depth image of the experimental plane with a depth resolution of 4096, the MWD method was able to encode the plane with 256 stairs at a normalized RMS error rate of 0.00015%. These high-resolution depth images could be encoded using lossy storage with the MWD approach, as well, yet a fewer number of stairs had to be used to avoid compression artifacts. For example, using the MWD with 34 stairs along with JPEG 100%, the same 4096 × 4096 depth image

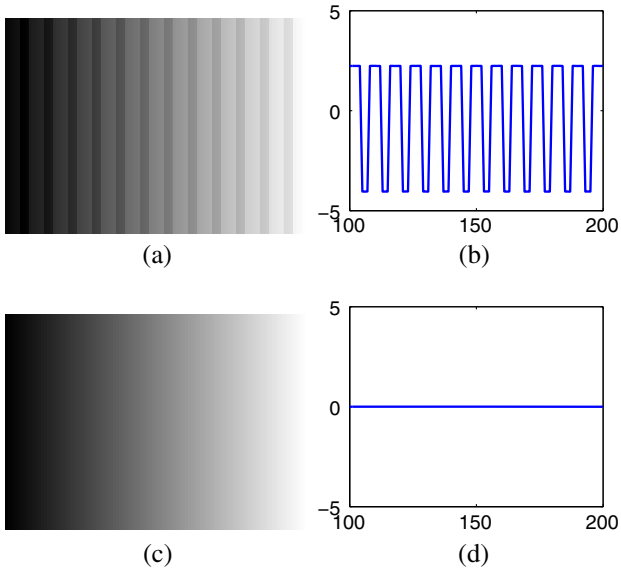

Fig. 7. Visual demonstration showing the DD's inability and MWD's ability to encode high-resolution depths. (a) A normalized subwindow of the recovered phase when using the direct depth technique to encode geometry with a depth resolution of 1024 and 128 stairs; (b) a cross section of the recovered phase (a) with its slope removed, the resulting 2π-tall spikes leave artifacts within the data; (c) the normalized subwindow of recovered phase when using the multiwavelength technique to encode the same geometry with the same number of stairs; (d) cross section of the recovered phase (c) with its slope removed.

with a depth resolution of 4096 was able to be encoded and recovered with a RMS error of 0.0024%.

4. DISCUSSION

The proposed multiwavelength depth encoding method has the following advantages and merits over the previously proposed direct depth encoding technique:

- *Higher compression ratios*. The nature of the encoding within the proposed technique, being continuous in the third channel, can be represented with less information. The result of this is smaller file sizes both using PNG and JPEG to compress and store the color-encoded 2D image.
- *Higher accuracy 3D data recovery*. Although identical under most circumstances when using PNG, the proposed method is much more robust, due to its more continuous and uniform nature, to high levels of JPEG compression. This results in less compression artifacts due to high levels of lossy compression.
- *Little to no filtering required*. Depending on the data to be encoded, the level of JPEG compression, and the number of stairs used to encode the data, the proposed method requires little to no postprocessing to fix artifacts; thus, decode computational complexity and time are lessened. Also, if postprocessing must occur, only filtering the third channel was found to be necessary in most circumstances. Whereas in the direct depth approach the recovered data itself was filtered, only having to filter the third channel in the proposed technique keeps the integrity of the two data channels in tact. The result is an improvement of the overall quality of recovered data.
- *High-resolution depth encoding*. In the proposed technique, up to 256 stairs can be used to encode the data. The direct depth technique must use a lower number of stairs to account for the requirement of a decently large gray-scale stair height. As more stairs can be used, high-resolution depth images can be encoded using the proposed technique. Experiments were able to encode geometries with depth resolutions of up to 4096, for example.

It should be noted that the technique proposed by this paper suffers from the same drawback as the aforementioned HoloImage and direct depth techniques in that only one side of a 3D surface is able to be encoded into a 2D image. This drawback is alleviated, however, if the proposed method is properly paired with a 3D scanning device (in which only one side of an object is captured). If the goal is to encode an object in 360°, many encodings could be taken from differing positions *around* the object. These scans need to be decoded, registered, and fused together to regenerate the object's original geometry.

5. SUMMARY

We have presented the MWD encoding method for 3D range compression, which represents 3D geometries in 2D images. These 2D images can be stored within both lossless and lossy formats to achieve additional data compression. In the past, traditional approaches may suffer compression artifacts due to high levels of lossy compression, yet it has been demonstrated that the proposed method works well under various, and even extreme, levels of lossy compression. The result of this is high compression ratios, small file sizes, and high 3D data reconstruction accuracies with little to no filtering required.

Funding. Directorate for Engineering (ENG) (CMMI-1300376); National Science Foundation (NSF) (CMMI-1523048)

Acknowledgment. The views expressed in this paper are those of the authors and not necessarily those of the NSF.

REFERENCES

1. S. Zhang, "Recent progresses on real-time 3D shape measurement using digital fringe projection techniques," Opt. Lasers Eng. **48**, 149–158 (2010).
2. N. Karpinsky and S. Zhang, "Composite phase-shifting algorithm for three-dimensional shape compression," Opt. Eng. **49**, 063604 (2010).
3. X. Gu, S. Zhang, P. Huang, L. Zhang, S.-T. Yau, and R. Martin, "Holoimages," in *Proceedings of the ACM Symposium on Solid and Physical Modeling* (ACM, 2006), pp. 129–138.
4. Z. Hou, X. Su, and Q. Zhang, "Virtual structured-light coding for three-dimensional shape data compression," Opt. Lasers Eng. **50**, 844–849 (2012).
5. N. Karpinsky and S. Zhang, "Holovideo: Real-time 3D range video encoding and decoding on GPU," Opt. Lasers Eng. **50**, 280–286 (2012).
6. N. Karpinsky and S. Zhang, "3D range geometry video compression with the H.264 codec," Opt. Lasers Eng. **51**, 620–625 (2013).
7. S. Zhang, "Three-dimensional range data compression using computer graphics rendering pipeline," Appl. Opt. **51**, 4058–4064 (2012).
8. T. Bell, N. Karpinsky, and S. Zhang, "Real-time 3D sensing with structured light techniques," in *Interactive Displays: Natural Human-Interface Technologies*, A. K. Bhowmik, ed. (Wiley, 2014), pp. 181–213.
9. H. Schreiber and J. H. Bruning, "Phase shifting interferometry," in *Optical Shop Testing* (Wiley, 2007), pp. 547–666.
10. J. Novák, P. Novák, and A. Mikš, "Multi-step phase-shifting algorithms insensitive to linear phase shift errors," Opt. Commun. **281**, 5302–5309 (2008).
11. D. C. Ghiglia and M. D. Pritt, *Two-Dimensional Phase Unwrapping: Theory, Algorithms, and Software* (Wiley, 1998).
12. S. Zhang and P. S. Huang, "Novel method for structured light system calibration," Opt. Eng. **45**, 083601 (2006).
13. Y. Wang and S. Zhang, "Superfast multifrequency phase-shifting technique with optimal pulse width modulation," Opt. Express **19**, 5149–5155 (2011).