applied optics

Two-channel depth encoding for 3D range geometry compression

Matthew G. Finley and Tyler Bell*

Department of Electrical and Computer Engineering, The University of Iowa, Iowa City, Iowa 52242, USA
*Corresponding author: tyler-bell@uiowa.edu

Received 21 June 2019; revised 6 August 2019; accepted 9 August 2019; posted 12 August 2019 (Doc. ID 370799); published 29 August 2019

This paper presents a novel method for accurately encoding 3D range geometry within only two channels of a 2D RGB image using a two-frequency phase-shifting approach. Once encoded within a 2D image, 3D geometry can be further compressed with conventional lossless or lossy image compression methods. The nature of the proposed two-channel encoding is relatively smooth; thus, large compression ratios with high reconstruction accuracies can be achieved and are experimentally demonstrated. For example, a compression ratio of 2883:1 was achieved, compared with the STL format, with a reconstruction RMS error of 0.45 mm (99.8% accuracy) when JPEG 85 was used with the proposed method. This paper also demonstrates how a 24-bit color texture map can be encoded alongside 3D geometry within a single 2D image.

https://doi.org/10.1364/AO.58.006882

1. INTRODUCTION

Modern three-dimensional (3D) range scanning devices have the potential to acquire high-resolution, high-accuracy 3D geometry information at real-time speeds [1]. In addition, 3D range scanning devices have recently become more portable and cost effective; further, they are even being included directly on consumer mobile devices [2]. Given the current availability of these devices, they are being increasingly adopted for use in a large variety of applications, including entertainment, security, medicine, manufacturing, and robotics.

Despite the modern breadth of options available for acquiring 3D geometry, many potential applications that utilize 3D scanning devices may be hindered, or are simply not feasible, due to the amount of data being generated. For example, a 3D capture at a modest resolution of only 480×640 points can require anywhere between 12–90 MB to store, if not more, depending on the 3D file format used (e.g., PLY, OBJ, STL). Even if a common 3D file format is not used to represent the data, 4.9 MB of raw data (i.e., floating-point 3D coordinates, color texture, and a mask) are still generated every frame at this resolution. If an application requires real-time (e.g., 30 Hz) data capture, 147 MB of data would be generated per second. To enable applications such as remote 3D geometry video streaming, more than 1.18 Gbps (gigabits per second) of Internet bandwidth would be needed to transmit this 3D video data in real-time. Data at this rate are difficult to store in real-time, let alone stream over a conventional wireless Internet connection.

To enable the efficient handling of 3D geometry, it needs to be compressed. One method actively being explored for compressing 3D range geometry is to encode the floating-point 3D data into the 8-bit color channels of a conventional 2D image. Once in this format, mature 2D image compression techniques can be utilized to reduce the overall size of the data. Lossless image compression techniques (e.g., PNG) can ensure high reconstruction qualities. When some amount of precision loss is tolerable, lossy image compression methods (e.g., JPEG) can be employed to faithfully represent the data with even lower file sizes. Several methods have been proposed that use principles of phase-shifting to precisely encode 3D range geometry into the red, green, and blue (RGB) color channels of a 2D image, which can then be further compressed and stored with PNG or JPEG image compression [3–6]. Although these methods can efficiently achieve high reconstruction accuracies in both lossless and lossy scenarios, they require all three color channels of the output image, leaving no room for auxiliary data, such as a texture map.

In order to further reduce file sizes, methods for encoding 3D data into only two color channels of the output image have also been explored. Hou *et al.* [7] and Wang *et al.* [8] each proposed two-channel encoding methods for 3D range geometry. These methods are able to accurately store 3D geometry information within only two color channels of a 2D image, freeing the third channel to store additional information if necessary. Although these methods were shown to work well with lossless compression (i.e., PNG), lossy compression methods (e.g., JPEG) may be able to achieve smaller file sizes.

To take advantage of the smaller file sizes offered by lossy compression, Bell *et al.* [9] proposed another two-channel encoding method for 3D range geometry. The method's data encodings were relatively smooth, allowing the method to be used in both lossless and lossy scenarios. Although this method was able to achieve high reconstruction accuracies in both storage scenarios, it had constraints on the depth range it could accurately encode, and its decoding procedure was relatively computationally expensive for real-time applications.

This paper presents the two-channel depth (TCD) encoding method for the encoding of 3D range geometry that utilizes two-frequency phase-shifting principles to properly reconstruct the 3D geometry. The proposed method's encoding results in channels that are generally smooth in nature. This allows for small file sizes and low rates of reconstruction error using both lossless and lossy compression. For example, a compression ratio of 2883:1 was achieved, compared with the STL format, with a reconstruction RMS error of 0.45 mm (99.8% accuracy) when JPEG 85 was used to compress 3D geometry encoded with the proposed TCD method. In addition, the encoding and decoding procedures of the method are relatively computationally inexpensive, making the method suitable for deployment within a variety of real-time scenarios, such as 3D video streaming to mobile devices. Last, as only two of the three color channels of an image are used to represent 3D geometry, this paper also discusses a method for incorporating color texture data into the remaining channel.

Section 2 will describe the principle of the proposed method's encoding and decoding procedures, including how the method was used to store 3D geometry and color texture in a single lossy compressed image. Section 3 will present various experimental results of the proposed two-channel depth encoding method. Section 4 will offer a brief discussion of the method's results and considerations. Last, Section 5 will summarize and conclude this paper.

2. PRINCIPLE

A. Phase-Shifting Techniques

When acquiring measurements of the physical world, several widely employed noncontact methods are used to recover 3D geometry, such as stereo vision, time-of-flight, and structured light. In the simplest form, a structured light scanner consists of a single camera and a single projector. The projector is used to project specially designed images onto the world, which spatially encode surfaces within a scene. The camera can then capture the encoded scene within a set of images. If the physical relationship between the camera and projector is known, information within the camera's images can be decoded, and 3D geometry can eventually be recovered.

Although there are many approaches to structured light, one method that has provided simplicity, accuracy, and high speeds is digital fringe projection (DFP). In DFP, the scanner's projection device is used to project a series of *fringe images* onto the scene. The intensity of the fringe images varies sinusoidally, and each successive image has its phase shifted by some amount. Many phase-shifting algorithms are available [10]; however, one of the most simple and efficient is three-step phase-shifting.

In the three-step phase-shifting algorithm, three fringe images with equal phase shifts are generated and projected onto the scene to be captured. The camera then captures each fringe image as it lies upon the scene. Here, each fringe image is defined as

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

$$I_2(x,y) = I'(x,y) + I''(x,y)\cos(\phi), \tag{2}$$

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

where I' is the average intensity, and I'' is the intensity modulation. The phase (ϕ) contains information related to the 3D geometry of the scene, and it can be solved via

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

Due to the limits of the inverse tangent function, the phase value recovered will range from $-\pi$ to π; thus, ϕ is referred to as the *wrapped phase*. To recover true phase values, an *unwrapped phase* (Φ) needs to be determined via a phase unwrapping algorithm. If the relationship between the projector and camera devices is known via system calibration [11], the unwrapped phase can be used pixel-by-pixel to recover 3D geometry of the scene. That is, for each pixel (i,j) in Φ, a unique 3D coordinate (x,y,z) can be determined.

One aspect that is crucial to recovering 3D geometry via DFP is phase unwrapping. Many methods have been proposed for phase unwrapping [12], and they generally fall into the categories of *spatial* and *temporal* phase unwrapping. In spatial phase unwrapping, a *relative* unwrapped phase map, which uses information within the wrapped phase map ϕ, can be estimated. The 3D geometry recovered from a relative phase map is only relative to the surface itself, meaning the exact positions of the coordinates in relation to the 3D scanner are not known. In temporal phase unwrapping, additional images are often projected following the phase-shifted images such that *fringe order* information of ϕ can be determined. If the fringe order information is known, an *absolute* unwrapped phase map can be recovered from ϕ.

One approach to determine fringe order (K) and, thus, absolute phase (Φ) is to use two or more frequencies of phase-shifted fringe images to encode the scene [13]. By appropriately selecting the sinusoidal frequencies for multiple sets of phase-shifted images, lower-frequency phase maps can be used to determine the fringe order of higher-frequency phase maps such that the higher-quality absolute phase can be determined.

B. Two-Channel Depth Encoding

This paper presents a novel method for encoding 3D range geometry within two color channels of a single 2D image that utilize principles of two-frequency phase-shifting. In this method, two different *fringe widths*—depth distances encoded by each period of the encoding function—are used in order to generate two different fringe encodings. The first encoding, I_1, is established to be a high-frequency encoding that encodes depth data Z with a user-defined fringe width of P. The second encoding, I_2, is established to be a low-frequency encoding that encodes Z within a single fringe width that is equivalent to the entire depth range of Z, Range(Z). Mathematically, the proposed method's two-frequency encodings of Z can be described for each pixel (i,j) as

$$I_1(i,j) = \frac{1}{2} + \frac{1}{2}\cos\left(2\pi \times \frac{Z(i,j)}{P}\right), \tag{5}$$

$$I_2(i,j) = \frac{Z(i,j)}{\text{Range}(Z)}. \tag{6}$$

After encoding the depth Z into images I_1 and I_2, they can be placed within two color channels of a traditional, 2D RGB image. The third channel of the output RGB image can be left empty to reduce file size, or it can be used to store 8-bit gray-scale or 24-bit color textures, as will be discussed in Section 2.E. Once the encoded images have been assigned to color channels, the output image is finally stored with 2D lossless (e.g., PNG) or lossy (e.g., JPEG) image compression.

C. Two-Channel Depth Decoding

To recover the encoded depth Z from the compressed image, it is important to note that there is no unique solution to I_1, although it corresponds to some unique value of depth, Z. It can be observed, however, that the direct solution to I_1 results in the absolute value of the wrapped phase, which can be computed for each pixel (i,j) as

$$|\phi(i,j)| = \cos^{-1}(2I_1(i,j) - 1). \tag{7}$$

The $\cos^{-1}$ function in Eq. (7) only has a range from 0 to π; therefore, the sign of ϕ cannot be directly determined. In order to increase the effective range of the $\cos^{-1}$ function from $-\pi$ to π and, thus, determine the sign of ϕ, it can be noted that, where the encoding sinusoid in I_1 is *decreasing*, the sign of ϕ will be positive, and where the encoding sinusoid in I_1 is *increasing*, the sign of ϕ will be negative.

As seen from the definition of a cosine, starting from zero, every other half period in I_1 will have decreasing magnitude. As a result, the regions where the encoding sinusoid in I_1 is increasing or decreasing can be determined through the use of the low-frequency fringe encoding in I_2 via

$$\gamma(i,j) = \text{Floor}\left(\frac{I_2(i,j) \times \text{Range}(Z)}{\beta}\right), \tag{8}$$

where β is defined as one half of the user-defined encoding fringe width, $\beta = P/2$. This gives enough information to determine the sign of the wrapped phase via

$$\phi(i,j) = \begin{cases} +|\phi(i,j)|, & \gamma(i,j) \text{ is even} \\ -|\phi(i,j)|, & \gamma(i,j) \text{ is odd.} \end{cases} \tag{9}$$

Next, a similar method to Eq. (8) can be applied to determine the fringe order of I_1 using the low-frequency fringe encoding in I_2 as

$$K(i,j) = \text{Round}\left(\frac{I_2(i,j) \times \text{Range}(Z)}{P}\right). \tag{10}$$

The values of the fringe order K determine the integer multiples of 2π that must be added to the wrapped phase ϕ in order to remove the 2π discontinuities and recover the continuous, absolute unwrapped phase, Φ, of the high-frequency fringe encoding. This can be described mathematically as

$$\Phi(i,j) = \phi(i,j) + 2\pi \times K(i,j). \tag{11}$$

Last, this absolute, unwrapped phase can be scaled to compute the recovered depth map Z' via

$$Z'(i,j) = \Phi(i,j) \times \frac{P}{2\pi}. \tag{12}$$

If the calibration parameters of the capture system are known, Z' can then be used to directly recover X' and Y' coordinates via

$$X'(i,j) = \frac{Z'(i,j) \times (i - u_0)}{f_u},$$
$$Y'(i,j) = \frac{Z'(i,j) \times (j - v_0)}{f_v}, \tag{13}$$

where f_u and f_v are the focal lengths along the u and v directions, and u_0 and v_0 are the offset of the principle point. If the calibration parameters of the capture system are not known, the X and Y coordinates can simply be uniformly sampled, in proportion to image indices (i,j), to later be recovered via some scaling constants, c_i and c_j. For example, $X'(i,j) = i \times c_i$ and $Y'(i,j) = j \times c_j$.

Figure 1 illustrates the proposed two-channel encoding and decoding process on an ideal hemisphere. Figure 1(a) is the 3D geometry of the sphere to be encoded. Figures 1(b) and 1(c) are the encodings I_1 and I_2, given by Eqs. (5) and (6), respectively. Figure 1(d) is the 512 × 512 output image when I_1 is stored in the green channel and I_2 is stored in the red channel of a compressed PNG image. Figure 1(e) is the wrapped phase ϕ derived from Fig. 1(d) via Eqs. (7)–(9). Figure 1(f) is the fringe order information K derived from I_2 in Eq. (10). Last, Fig. 1(g) is the absolute unwrapped phase Φ solved in Eq. (11) that is used by Eq. (12) to determine the recovered depth map Z' rendered in Fig. 1(h).

D. Decoding Error Correction

The previously defined method of encoding and decoding 3D range geometry using two color channels of an RGB image results in visual surface artifacts at periodic regions within the depth range of Z'. Error arises at these locations due to the inability to recover the correct values of γ in Eq. (8), which allow for the correct sign of ϕ to be determined by Eq. (9). In practice, as compression quality decreases, the ability to correctly determine γ around integer multiples of β also decreases due to the reliance on the low precision depth encoding stored in I_2.

One approach in detecting the exact locations of these errors would be to search for statistical outliers in the difference between the low-quality depth map stored in I_2 and the recovered depth map Z'. Although this would work well when using lossless compression methods, lossy compression can result in a significant degradation of I_2. Because the magnitude of the periodic errors discussed here is small, they cannot be detected in the presence of this compression noise with any confidence.

Instead, a simpler approach is implemented in order to correct the periodic errors that occur as a result of this two-channel compression method. In this approach, error regions within the recovered depth map, Z', are defined around each integer multiple of β with some additional user-defined depth distance, α. Once the error regions have been identified, they can be corrected. A simple method of correcting these errors involves replacing the defined error regions with points from the corresponding regions of a heavily filtered version of Z'. After this

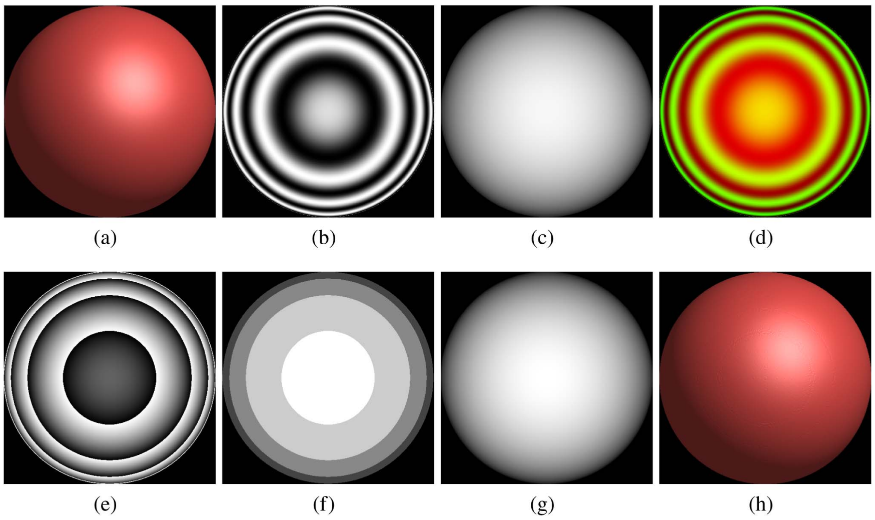

Fig. 1. Proposed two-channel encoding and decoding process. (a) Original 3D geometry to be encoded; in this case, an ideal hemisphere. (b) I_1 encoding of (a) given by Eq. (5). (c) I_2 Encoding of (a) given by Eq. (6). (d) Encoded image E, which is the output of the proposed method. (e) Wrapped phase ϕ derived from (b). (f) Fringe order information K derived from (c). (g) Unwrapped phase map Φ. (h) 3D rendering of the recovered Z'.

replacement, the corrected depth map can then be lightly filtered to further reduce compression artifacts.

Figure 2 demonstrates the proposed method's error correction process on an ideal hemisphere. Figure 2(a) shows the uncorrected Z', given by Eq. (12), with periodic errors around every β within the encoding depth range. Figure 2(b) shows the error points segmented around each β with some user-defined distance α. Figure 2(c) shows Z' with the identified error locations removed. Last, Fig. 2(d) shows the corrected Z' after error locations have been filled from a filtered version of Z'.

E. Encoding Depth with Color Texture

As mentioned above, the proposed method only utilizes two channels of the 2D output image, leaving one channel free to store additional information such as a texture image. If the texture image to be stored is an 8-bit gray-scale image, it can simply be placed within the free channel before the output image is saved with lossless or lossy compression. Many modern 3D scanning devices, however, produce a 24-bit color texture image, which simply does not fit within the remaining 8-bit color channel. To avoid this issue, 3D-to-2D compression methods typically place the color texture *next* to the encoded output image in a mosaic format. Although this well preserves the color texture, it doubles the resolution of the output image, thus reducing the amount of available data savings.

One option to fit the color image within the remaining encoding channel is to quantize the red, green, and blue color channels of the texture image such that, when combined, they will all fit within the remaining channel. Although viable, this method may not work well when the output image is stored using various qualities of lossy compression. To store color texture within the remaining channel of this paper's method, a two-step method is proposed that consists of (1) Bayer color filter [14] simulation and (2) color clustering. In Step (1), a Bayer color filter (e.g., RGGB) is applied to the color image T_c such that the resulting image T_b is a gray-scale Bayer encoded image. At this point, T_b is an 8-bit image and can be stored within the remaining channel of the output image; however, there are high levels of local intensity changes due to the Bayer filter; thus, the image may not compress well with lossy methods. To make the resulting Bayer encoded image smoother, Step (2) separates and clusters the colors encoded within T_b such that all of the encoded red pixels are adjacent to one another, all of the encoded green pixels are adjacent to one

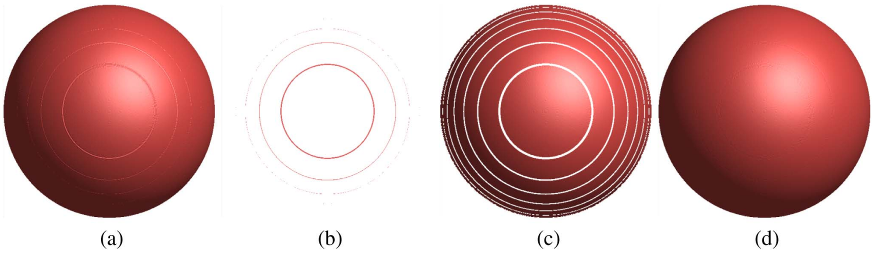

Fig. 2. Illustration of the proposed method's error correction process. (a) Uncorrected Z' with periodic errors around every β within the encoding's depth range. (b) Segmented error in Z' around each β. (c) Z' with the segmented errors removed. (d) Corrected Z' with error locations replaced with data from a filtered Z'.

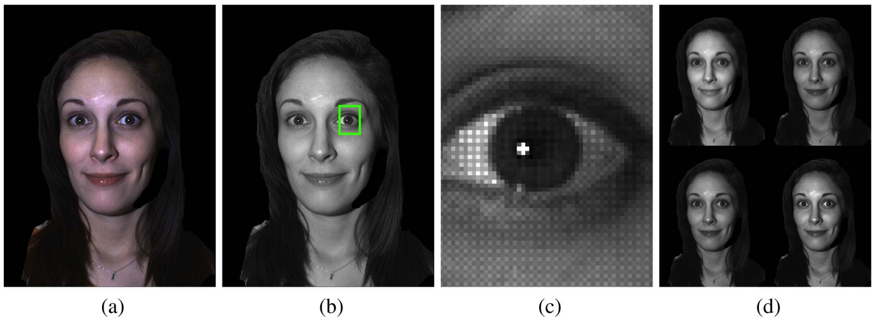

Fig. 3. Storing 24-bit color texture within the remaining 8-bit channel of the proposed method's output image. (a) 24-bit color texture image, T_c, to be encoded. (b) 8-bit color encoded image, T_b, generated from (a) after simulating an RGGB Bayer color filter. (c) Magnified portion of (b) that highlights the RGGB Bayer color filter. (d) Result T_B after separating the colors encoded within (c). This is the final 8-bit result that can be stored within the free channel of the proposed encoding method for both lossless and lossy compression.

another, and all of the encoded blue pixels are adjacent to one another. After performing this operation, T_B is an 8-bit grayscale representation of a color texture image that can more efficiently be stored within the remaining channel of the proposed method's output image, even if lossy compression is utilized. To recover the color texture, the colors are first sorted back into the pattern that corresponds to the original Bayer filter that was applied. Last, a demosaicing operation can be used to reconstruct a color image from the Bayer encoded image.

Figure 3 illustrates the process for encoding color texture alongside the proposed method's encoded 3D geometry. Figure 3(a) is the original 24-bit color texture image (T_c) that corresponds to some captured 3D geometry. Figure 3(b) is the color encoded image (T_b) generated via Bayer filter simulation with a RGGB Bayer grid, and Fig. 3(c) is a magnified view that better shows the resulting Bayer pattern. Figure 3(d) is the final encoded image (T_B) that shows the result after separating and clustering the colors encoded within T_b. The top left quadrant contains all of the red pixels in T_b, the top right and bottom left quadrant contain all of the green pixels in T_b, and the bottom right quadrant contains all of the blue pixels in T_b; this matches the structure of the RGGB Bayer grid used to generate T_b. The image T_B in Fig. 3(d) is the encoded color image that can be stored in the proposed method's remaining 8-bit channel, allowing the proposed method to represent both encoded 3D geometry and color texture within a single 2D image that can be compressed with lossless and lossy image compression technologies.

3. EXPERIMENTS

To evaluate the performance of the proposed two-channel 3D range geometry encoding method, several experiments were conducted. The first experiment encoded an ideal hemisphere, with a radius of 256 mm, into a 512 × 512 image via the proposed method, as illustrated in Fig. 1. For all encodings of the hemisphere, the user-defined fringe width P was selected such that Z was encoded within $n = 4$ periods. During the encoding process, I_1 was stored into the output image's green channel, and I_2 was stored in the output image's red channel; the blue channel was left empty for this experiment. After the hemisphere was encoded into a 2D image, the image was then compressed with lossless compression (e.g., PNG) and varying levels of lossy compression (e.g., JPEG).

Figure 4 shows the reconstruction results of the encoded hemisphere for each compression quality. Each column represents reconstructions when the output image in Fig. 1(d) was stored with PNG, JPEG 100, JPEG 95, JPEG 90, and JPEG 85, respectively. The first row of Fig. 4 shows the uncorrected hemisphere recovered for each quality; the second row shows the corrected hemispheres following the error correction procedure described in Section 2.D; and the third row shows the corrected hemispheres with a small filter applied to further reduce compression artifacts. As can be seen, the method is able to accurately reconstruct the sphere in both lossless and lossy scenarios. The main form of error on the uncorrected hemispheres in the first row appear as *rings* within the sphere's depth range. As visualized in the second and third rows, however, the proposed error correction method can adequately detect and address these errors.

In addition to the visual results of the proposed method shown in Fig. 4, cross sections of the reconstruction errors were also evaluated. Figure 5 shows error plots for the 256th row between the hemisphere's depth map Z and the recovered depth map Z'. The first column shows errors when PNG was used to store the encoded output image; the second, third, and fourth columns show errors for JPEG 100, JPEG 95, and JPEG 85, respectively. Similar to Fig. 4, the first row of Fig. 5 shows cross sections of error for uncorrected depth maps Z', the second row shows error for corrected depth maps following the procedure in Section 2.D, and the third row shows error for corrected depth maps after applying light filtering to further reduce compression artifacts.

Table 1 provides RMS errors when the proposed method was used to compress the ideal hemisphere with a radius of 256 mm into a 512 × 512 image. It should be noted that, when calculating these errors, the outside perimeters of the reconstructed hemispheres were eroded with a width of 5 pixels to remove the influence of outliers. As shown in this table, the proposed method results in a small amount of error due to compression relative to the radius of the sphere. The table also highlights the effectiveness of the error correction procedure described in Section 2.D. For example, even at JPEG

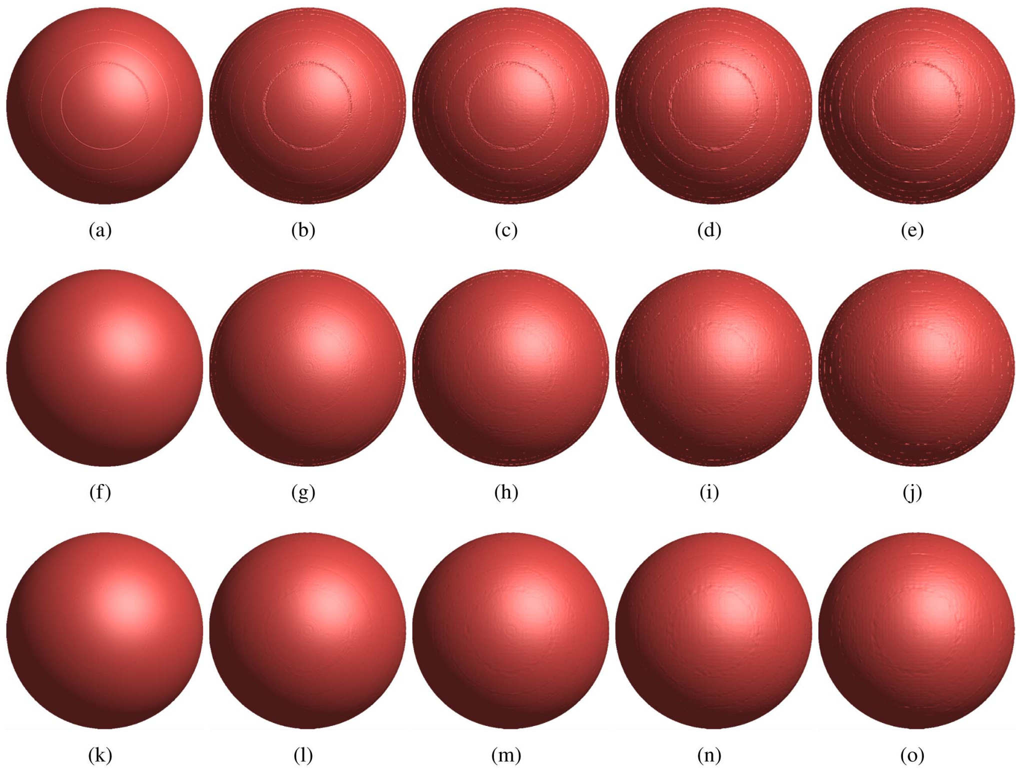

Fig. 4. Visual results of recovered hemisphere using the proposed method with various levels of image compression. Each column represents reconstruction results after encoding into a PNG, JPEG 100, JPEG 95, JPEG 90, and JPEG 85 quality image, respectively. (a)–(e) Row 1: Reconstructions using the proposed method. (f)–(j) Row 2: Results after the proposed error correction has been applied. (k)–(o) Row 3: Results after applying a filter to the error corrected data.

85, the proposed method only incurs an RMS error of 0.450 mm (99.8% accuracy) after correction.

The proposed method also achieves small file sizes due to (1) its use of only two channels of an RGB image and (2) the relatively smooth nature of the encodings within the two channels. When storing the hemisphere into a 512 × 512 output image, the file sizes were 129.0 KB for PNG, 123.4 KB for JPEG 100, 61.5 KB for JPEG 95, 45.1 KB for JPEG 90, and 37.4 KB for JPEG 85. To put these numbers into context, Table 2 provides compression ratios for the proposed method when compared with popular 3D file formats OBJ, PLY, and STL.

In addition to validating the proposed two-channel encoding method with ideal 3D geometry, a more complex 3D model was also tested. Figure 6 displays the method's reconstruction performance for various levels of image compression. Figures 6(a) and 6(b) show 3D geometry and color texture, respectively, of a 480 × 640 3D model. The proposed two-frequency encoding method was applied to generate I_1 and I_2, as shown in Figs. 6(c) and 6(d). As with the sphere, the user-defined fringe width P for I_1 was defined such that Z was encoded within $n = 4$ periods. The 24-bit color texture in Fig. 6(b) was then processed as described in Section 2.E to generate T_B, the color-separated, Bayer-encoded 8-bit image shown in Fig. 6(e). Images I_1, I_2, and T_B were then, respectively, placed into the green, red, and blue channels of a single 2D image, as shown in Fig. 6(f). This image was then stored with PNG and various levels of JPEG compression.

Figure 6(g) shows depth and color reconstructions of the proposed method. The first row shows 3D geometries rendered in shaded mode; the second row shows 3D geometries rendered with the recovered color texture mapped onto the geometry. The first column of Fig. 6(g) shows the original 3D geometry that was encoded. The remaining columns show reconstructions from the image in Fig. 6(f) when it was compressed with PNG, JPEG 100, JPEG 95, JPEG 90, and JPEG 85, respectively. It is worth mentioning that all of the depth and color texture reconstructions in Fig. 6(g) come from the single 480 × 640 image [Fig. 6(f)], which only required 280.4 KB for PNG, 166.9 KB for JPEG 100, 77.2 KB for JPEG 95, 54.3 KB for JPEG 90, and 43.9 KB for JPEG 85.

4. DISCUSSION

The proposed two-channel depth (TCD) encoding method has several important aspects that make it suitable for adoption within practical applications where 3D geometry may need to be stored or transmitted in real-time, such as live 3D video communications. The following summarizes the main contributions of the proposed TCD method.

1. **Small file sizes.** The proposed method only requires two color channels to store encoded geometry within an image that can be further compressed with image compression techniques. With the combination of depth encoding and image compression, small file sizes (i.e., large compression ratios)

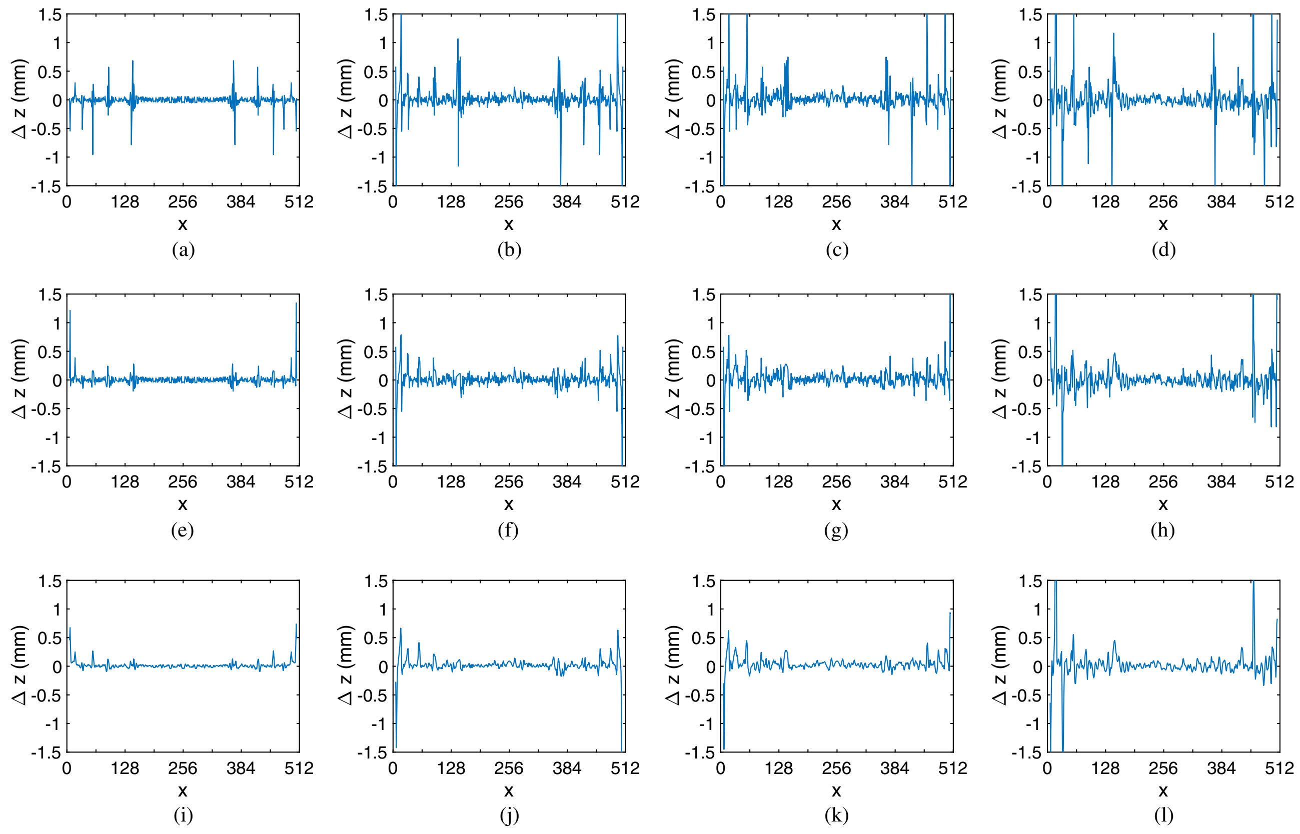

Fig. 5. Graphical results of error between the 256th row of the original depth map, Z, and the reconstructed depth map, Z', when various levels of image compression were used to encode an ideal hemisphere with a radius of 256 mm. Each column represents reconstruction results after encoding into a PNG, JPEG 100, JPEG 95, and JPEG 85 quality image, respectively. Row 1: Reconstruction error plots using the proposed method; RMS errors for (a)–(d) are 0.141, 0.733, 0.748, and 0.843 mm, respectively. Row 2: Error plots after the proposed error correction has been applied. RMS errors for (e)–(h) are 0.115, 0.688, 0.693, and 0.787, respectively. Row 3: Error plots after applying a filter to the corrected data. RMS errors for (i)–(l) are 0.090, 0.408, 0.406, and 0.450, respectively.

Table 1. RMS Error of a Reconstructed Ideal Hemisphere (with a Radius of 256 mm) That Was Compressed into a 512 × 512 Image Using the Proposed Two-Channel Encoding

	PNG	JPEG 100	JPEG 95	JPEG 90	JPEG 85
Uncorrected	0.141 mm	0.733 mm	0.748 mm	0.766 mm	0.843 mm
Corrected	0.115 mm	0.688 mm	0.693 mm	0.709 mm	0.787 mm
Corrected (Filtered)	0.090 mm	0.408 mm	0.406 mm	0.413 mm	0.450 mm

can be achieved. It is worth noting that, while only PNG and JPEG file sizes and ratios were reported in this paper, other options can be used to compress the generic 2D RGB images constructed by the TCD method. For example, image formats such as the free lossless image format may also be employed.

2. **Low rates of reconstruction error.** The proposed method encodes 3D range geometry in a manner that results in images that are relatively smooth in nature. This is especially important when lossy image compression is used, as it (1) helps to achieve smaller file sizes (i.e., tens of kilobytes) and (2) helps

Table 2. Compression Ratios of the Proposed Method Using PNG and JPEG to Encode an Ideal Hemisphere into a 512 × 512 Image Compared With 3D Mesh Formats

	PNG	JPEG 100	JPEG 95	JPEG 90	JPEG 85
OBJ	123.5: 1	129.0: 1	258.9: 1	353.6: 1	425.9: 1
PLY (Binary)	60.5: 1	63.2: 1	126.8: 1	173.1: 1	208.6: 1
PLY (ASCII)	98.0: 1	102.4: 1	205.4: 1	280.6: 1	338.0: 1
STL (Binary)	158.9: 1	166.1: 1	333.1: 1	455.0: 1	548.1: 1
STL (ASCII)	836.0: 1	873.5: 1	1752.2: 1	2393.3: 1	2883.0: 1

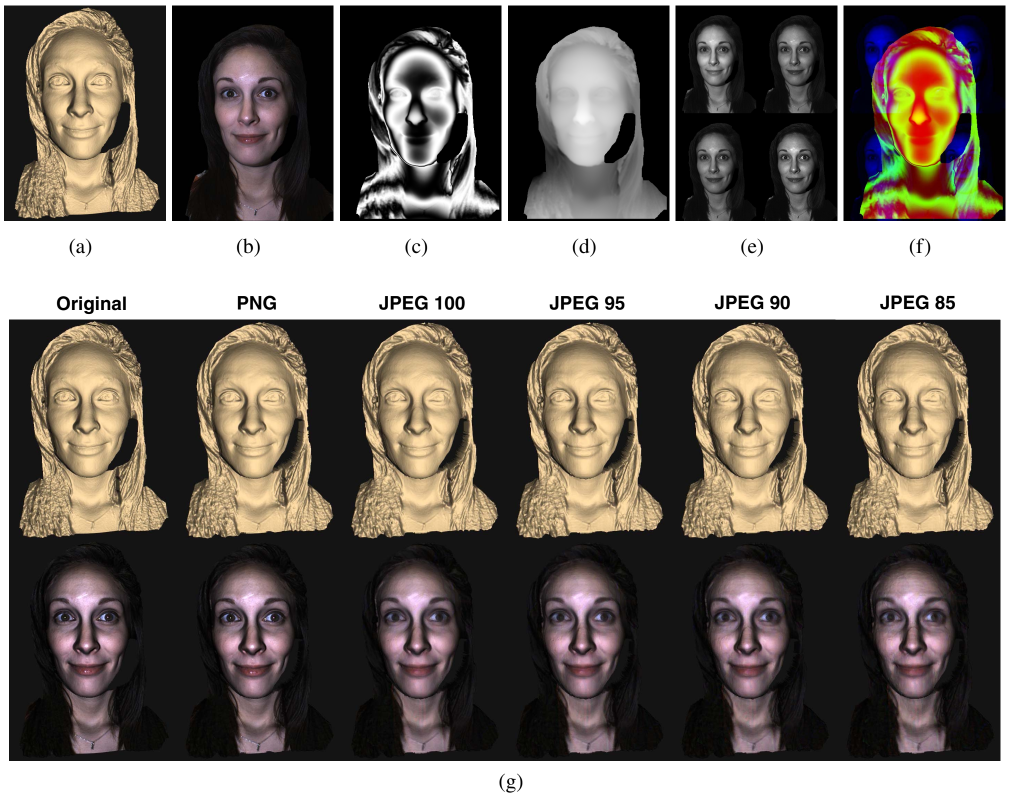

Fig. 6. Results of the proposed method encoding complex geometry. (a) Original 3D geometry. (b) Original color texture. (c) I_1 encoding of (a) given by Eq. (5). (d) I_2 encoding of (a) given by Eq. (6). (e) Encoded color texture T_B following the process in Section 2.E. (f) Final encoded output image that contains encoded 3D geometry and encoded color texture. (g) Visual results of reconstructions when (f) was stored with various levels of 2D compression. The first column shows the original geometry and color texture that was encoded. Each successive column of (g) shows reconstructions when (f) was stored with PNG, JPEG 100, JPEG 95, JPEG 90, and JPEG 85, respectively. The first row of (g) shows reconstructed 3D geometry; the second row shows reconstructed 3D geometry mapped with recovered color texture.

to reduce compression artifacts within the recovered geometry. Between the method's general robustness to lossy compression artifacts and the decoding correction described in Section 2.D, TCD is able to recover depth maps with high degrees of accuracy.

3. **Efficient encoding and decoding.** The encoding and decoding process of the proposed TCD method takes place mostly pixel-by-pixel. That is, each pixel (i, j) within Z gets encoded into a pixel of the 2D output image. Each pixel within the output image then results in some pixel (i, j) within the reconstructed depth map Z'. Given this, the method is well suited to be parallelized for implementation on a device's graphics processing unit (GPU). The result of this is potentially faster encoding and decoding, on a broader range of devices, for applications that may have real-time data storage or transmission requirements.

4. **Ability to include color texture.** In similar 3D-to-2D encoding methods, the resolution of the 2D output image may often be doubled in order to store both geometry and color texture in a mosaic format. The proposed method, however, only requires two channels to encode geometry, leaving one channel free to store additional information. This paper demonstrated how the remaining 8-bit channel could be used to store a 24-bit color texture map. The result is a single method that can be used to compress both 3D range geometry and color texture within a single 2D image.

Last, it is worth noting that, in practice, there is a trade-off between the number of periods used to encode the geometry and the resulting reconstruction accuracy. In theory, as the number of periods increase, encoding precisions increase as well. Due to the periodic errors around each β within the recovered depth map, however, as the number of encoding periods increases, the frequency of periodic errors within the recovered depth map also increases. In addition, as the level of lossy compression increases, the periodic errors become more difficult to correct. One future work may be to automatically determine the number of periods to use that will provide the optimal trade-off between encoding precision and reconstruction accuracy for some 3D geometry.

5. CONCLUSION

This paper presented the two-channel depth (TCD) encoding method that utilized two-frequency phase-shifting to properly reconstruct 3D geometry from only two channels of a 2D image. The proposed method's encoding resulted in small file sizes and low rates of reconstruction error with both lossless and lossy compression. For example, a compression ratio of 836:1 was achieved versus the STL format with a reconstruction RMS error of 0.09 mm (99.9% accuracy) when PNG was used, and a compression ratio of 2883:1 was achieved versus STL with a reconstruction RMS error of 0.45 mm (99.8% accuracy) when JPEG 85 was used. As only two of the three color channels of the output 2D image are needed by TCD to represent 3D geometry, this paper also showed how color texture could be stored in the remaining channel. Using this approach, both 3D range geometry and a color texture image can be compressed within a single 2D image. Finally, given that the proposed encoding and decoding are mostly performed pixel-by-pixel, it can be parallelized, thus making the method suitable for deployment within real-time scenarios, such as 3D video streaming to and from mobile devices.

Funding. University of Iowa (ECE Department Faculty Startup Funds).

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. T. Bell and S. Zhang, "Holo reality: real-time, low-bandwidth 3d range video communications on consumer mobile devices with application to augmented reality," in *Electronic Imaging 2019: 3D Measurement and Data Processing* (Society for Imaging Science and Technology, 2019), pp. 3DMP-007-1–3DMP-007-5.
3. N. Karpinsky and S. Zhang, "Composite phase-shifting algorithm for three-dimensional shape compression," Opt. Eng. **49**, 063604 (2010).
4. S. Zhang, "Three-dimensional range data compression using computer graphics rendering pipeline," Appl. Opt. **51**, 4058–4064 (2012).
5. P. Ou and S. Zhang, "Natural method for three-dimensional range data compression," Appl. Opt. **52**, 1857–1863 (2013).
6. T. Bell and S. Zhang, "Multiwavelength depth encoding method for 3d range geometry compression," Appl. Opt. **54**, 10684–10691 (2015).
7. Z. Hou, X. Su, and Q. Zhang, "Virtual structured-light coding for three-dimensional shape data compression," Opt. Lasers Eng. **50**, 844–849 (2012).
8. Y. Wang, L. Zhang, S. Yang, and F. Ji, "Two-channel high-accuracy holoimage technique for three-dimensional data compression," Opt. Lasers Eng. **85**, 48–52 (2016).
9. T. Bell, B. Vlahov, J. P. Allebach, and S. Zhang, "Three-dimensional range geometry compression via phase encoding," Appl. Opt. **56**, 9285–9292 (2017).
10. H. Schreiber and J. H. Bruning, "Phase shifting interferometry," in *Optical Shop Testing* **3**rd ed. (Wiley, 2007), Chap. 14, pp. 547–666.
11. B. Li, N. Karpinsky, and S. Zhang, "Novel calibration method for structured-light system with an out-of-focus projector," Appl. Opt. **53**, 3415–3426 (2014).
12. S. Zhang, "Absolute phase retrieval methods for digital fringe projection profilometry: a review," Opt. Laser. Eng. **107**, 28–37 (2018).
13. Y. Wang and S. Zhang, "Superfast multifrequency phase-shifting technique with optimal pulse width modulation," Opt. Express **19**, 5149–5155 (2011).
14. B. E. Bayer, "Color imaging array," U.S. patent 3,971,065 (July 20, 1976).