Composite phase-shifting algorithm for three-dimensional shape compression

Nikolaus Karpinsky
Song Zhang, MEMBER SPIE
Iowa State University
Department of Mechanical Engineering
Ames, Iowa, 50011
E-mail: song@iastate.edu

Abstract. With recent advancements in 3-D imaging and computational technologies, acquiring 3-D data is unprecedentedly simple. However, the use of 3-D data is still limited due to the size of 3-D data, especially 3-D video data. Therefore, the study of how to store and transmit the 3-D data in real time is vital. We address a technique that encodes a 3-D surface shape into a single 24-bit color image. In particular, this image is generated by advanced computer graphics tools with two primary color channels encoded as sine and cosine fringe images, and the third channel encoded as a stair image to unwrap the phase obtained from the two fringe images. An arbitrary 3-D shape can then be recovered from a single image. We test 3-D shapes with differing levels of complexity along with various image formats. Experiments demonstrate that, without significantly losing the shape quality, the compression ratio can go up to 1:36.86, compared with the native smallest possible 3-D data representation method.
[DOI: 10.1117/1.3456632]

Subject terms: phase-shifting algorithm; fringe analysis; compression; three-dimensional.

Paper 090821R received Oct. 22, 2009; revised manuscript received May 6, 2010; accepted for publication May 11, 2010; published online Jun. 30, 2010.

1 Introduction

With recent advancements in 3-D imaging and computational technologies, acquiring 3-D data is unprecedentedly simple. Recent advancements in digital display technology and computers has accelerated research in 3-D imaging techniques. The 3-D imaging technology has been increasingly used in both scientific studies and industrial practices. Real-time 3-D imaging recently emerged, and a number of techniques have been developed.[1–5] For example, we developed[6] a system to measure absolute 3-D shapes at 60 frames/s with an image resolution of 640×480. The 3-D data throughput of this system is approximately 228 Mbytes/s, which is very difficult to store and transmit simultaneously. Therefore, research into how to store and transmit the 3-D data in real time is vital.

Unlike 2-D images, 3-D geometry conveys much more information, albeit at the price of increased data size. In general, for a 2-D color image, 24 bits (or 3 bytes) are enough to represent each color pixel [red (R), green (G), and blue (B)]. However, for 3-D geometry, an (x,y,z) coordinate typically requires at least 12 bytes excluding the connectivity information. Thus, the size of 3-D geometry is at least 4 times larger than that of a 2-D image with the same number of points.

There are numerous ways to represent 3-D data. Wikipedia lists most of the commonly used file formats.[7] Usually, 3-D data are represented in different ways for different purposes. In computer-aided design (CAD), STL is one of the commonly used file formats. It describes a raw unstructured triangulated surface by the unit normal and vertices. This file format does not include texture information. Because STL is a file format native to the stereolithography CAD software created by 3-D systems, it is widely used for rapid prototyping and computer-aided manufacturing.[8] In computer graphics, the OBJ file format is one of the most commonly accepted formats. It is a simple data-format that represents geometry alone: the position of each vertex, the UV coordinate of each texture coordinate vertex, normals, and the faces that make each polygon defined as a list of vertices, and texture vertices.[9] Because these data formats must store connectivity information, 3-D file size is relatively large. Mat5 is a native format that stores the natural data captured by an area 3-D scanner, it stores five matrices:[10] the color, the quality, the x, the y, and the z. This is essentially unstructured data; thus, the connectivity information is naturally stored (captured) by splitting grids into triangles. This file format is thus smaller in comparison with other data formats.

A benefit of the Holoimage format is that it can use existing research of 2-D image processing, which is a well-studied field, and the size of 2-D images is much smaller than that of 3-D geometries. The idea of a reduced data size and existing techniques for 3-D image process is attractive. Since 3-D geometry is usually obtained by 2-D devices (e.g., a digital camera), it is natural to use its originally acquired 2-D format to compress it.

In this paper, we address a technique that converts 3-D surfaces into a single 2-D color image. The color image is generated using advanced computer graphics tools to synthesize a digital fringe projection and a phase-shifting system for 3-D shape measurement. We propose a new coding method called the "composite phase-shifting algorithm" for 3-D shape recovery. With this method, two color channels

0091-3286/2010/$25.00 © 2010 SPIE

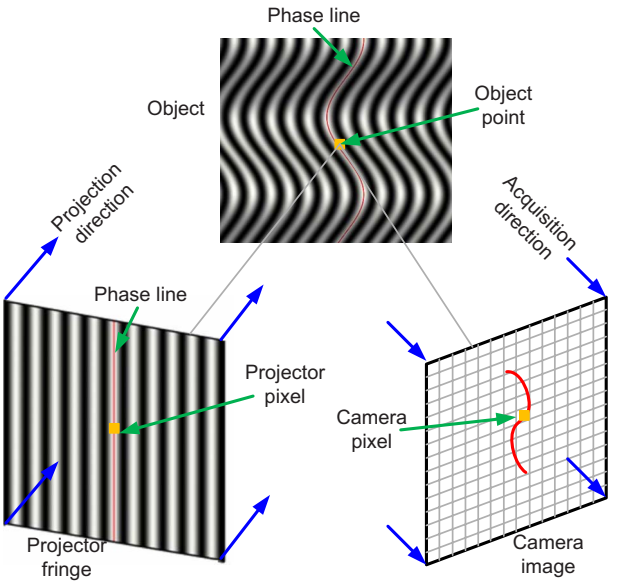


Fig. 1 Virtual digital fringe projection system setup.

(R,G) are encoded as sine and cosine fringe images, and the third color channel (B) is encoded as a stair image; the stair image can be used to unwrap the phase map obtained from two fringe images point by point. By using a 24-bit image and no spatial phase unwrapping, the 3-D shape can be recovered; therefore, the single 2-D image can represent a 3-D surface.

The encoded 24-bit images can be stored in different formats, e.g., bitmap, portable network graphics (PNG), and JPG. If the image is stored in a lossless format, such as bitmap and PNG, the quality of 3-D shape is not affected at all. We found that lossy compression such as JPG compression cannot be directly implemented, as it distorts the blue channel severely affecting the 3-D surface. To circumvent this problem, red and green channels are stored using JPG under different compression levels, while the blue channel remains in a lossless PNG format. Our experiments demonstrated that there is little error for a compression ratio up to 1:36.86, compared with the native smallest possible 3-D data representation method. Experiments are presented to verify the performance of the proposed approach.

Section 2 presents the fundamentals of the virtual fringe projection system and the composite phase-shifting algorithm. Section 3 shows experimental results, and finally Sec. 4 summarizes the paper.

2 Principle

2.1 *Virtual Digital Fringe Projection System Setup*

Figure 1 shows a the virtual fringe projection system setup, which is also known as a Holoimage system.[11] It is very similar to that a real fringe-projection-based 3-D shape measurement system. A projector projects fringe images onto an object and a camera captures the fringe images that the object has distorted. The 3-D information can be retrieved if the geometric relationship between the projector pixels and the camera pixels is known.

The virtual system differs from the real 3-D shape measurement system in that the projector and the camera are orthogonal devices instead of perspective ones, and the relationship between the projector and the camera is precisely defined. Thus, the shape reconstruction becomes significantly simplified and precise. To represent an arbitrary 3-D shape, a multiple-wavelength phase-shifting algorithm[12–15] can be used. However, it requires more than three fringe images to represent one 3-D shape, which is not desirable for data compression.

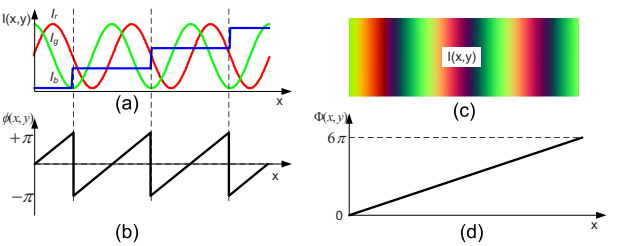


Fig. 2 Schematic diagram of the proposed composite algorithm for single color fringe image generation: (a) cross section of the color fringe images, where red is the sine image (I_r), green is the cosine image (I_g), and blue is the stair image (I_b); (b) the cross section of the phase map $[\phi(x,y)]$ using red and green fringe images from Eq. (3); (c) the real fringe image; and (d) the unwrapped phase after correcting the wrapped phase $[\phi(x,y)]$ by the stair image (I_b). (Color online only.)

2.2 *Composite Phase-Shifting Algorithm*

Due to the virtual nature of the system, all environmental variables can be precisely controlled, simplifying the phase-shifting process. To obtain phase, only sine and cosine images are actually required, which can be encoded into two color channels, e.g., the red and green channels. The intensity of these two images can be written as

$$I_r(x,y) = 255/2\{1 + \sin[\phi(x,y)]\}, \tag{1}$$

$$I_g(x,y) = 255/2\{1 + \cos[\phi(x,y)]\}. \tag{2}$$

From Eqs. (1) and (2), we can obtain the phase

$$\phi(x,y) = \tan^{-1}\left(\frac{I_r - 255/2}{I_g - 255/2}\right). \tag{3}$$

The phase obtained in Eq. (3) is in the range $[-\pi,+\pi)$. To obtain a continuous phase map, a conventional spatial phase-unwrapping algorithm can be used. However, it is known that the step height changes between two pixels cannot be larger than π. The phase-unwrapping step is essentially to find the integer number K for 2π jumps for each pixel so that the true phase can be found:[16]

$$\Phi(x,y) = 2\pi K + \phi(x,y). \tag{4}$$

If an additional stair image $I_b(x,y)$ is used whose intensity changes are precisely aligned with the 2π phase jumps (as shown in Fig. 2), the phase-unwrapping step can be performed point by point by using the stair image information. In other words, the unwrapped phase will be

$$\Phi(x,y) = 2\pi I_b(x,y) + \phi(x,y). \tag{5}$$

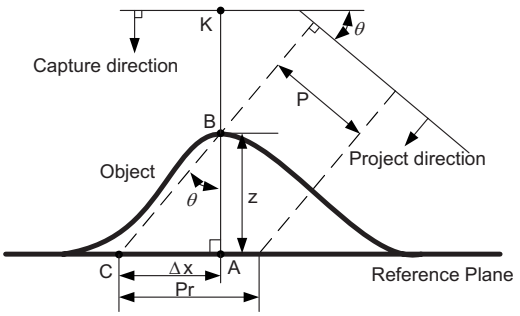


Fig. 3 Schematic diagram for phase-to-coordinate conversion.

In practice, to reduce the problems caused by digital effects, instead of using one gray-scale value for each increment, a larger value is used. In the example shown in Fig. 2, 80 gray-scale values are used to represent one stair.

2.3 *Phase-to-Coordinate Conversion*

Figure 3 illustrates the phase-to-coordinate conversion. To explain the concepts, a reference plane (a flat surface with $z=0$) is used. Assume the fringe pitch generated by the projector is P, and the projection angle is θ; the fringe pitch on the reference plane will be $P_r=P/\cos\theta$. For an arbitrary image point K, if there is no object in place, the phase is Φ_A^r on the reference plane. Once the object is in position, the imaging point on the object is B. From the projector point of view, B on the object and C on the reference plane have the same phase, i.e., $\Phi=\Phi_B=\Phi_C^r$. Then, we have

$$\Delta\Phi = \Phi_C^r - \Phi_A = \Phi_B^r - \Phi_A^r = \Phi - \Phi_A^r. \tag{6}$$

Since the fringe stripes are uniformly distributed on the reference plane, for the pipeline introduced in this paper, the reference plane is well-defined ($z=0$). The phase on the reference plane is defined as a function of the projection angle θ and the fringe pitch P:

$$\Phi^r = 2\pi i/P_r = 2\pi i \cos\theta/P, \tag{7}$$

assuming phase 0 is defined at $i=0$ and the fringe stripes are vertical. Here, i is the image index horizontally. From Eqs. (6) and (7), we have,

$$\Delta\Phi = \Phi - 2\pi i \cos\theta/P. \tag{8}$$

Also we have,

$$\Delta\Phi = \Phi_C^r - \Phi_A^r = 2\pi\Delta i \cos\theta/P. \tag{9}$$

Moreover, the graphics pipeline can be configured to visualize within a unit cube, when pixel size is $1/W$. Here, W is the total number of pixel horizontally, or window width. Then,

$$x = i/W, \tag{10}$$

assuming the origin of the coordinate system is aligned with the origin of the image.

Similarly, for the y coordinate, if we assume the y direction has the same scaling factor, we have

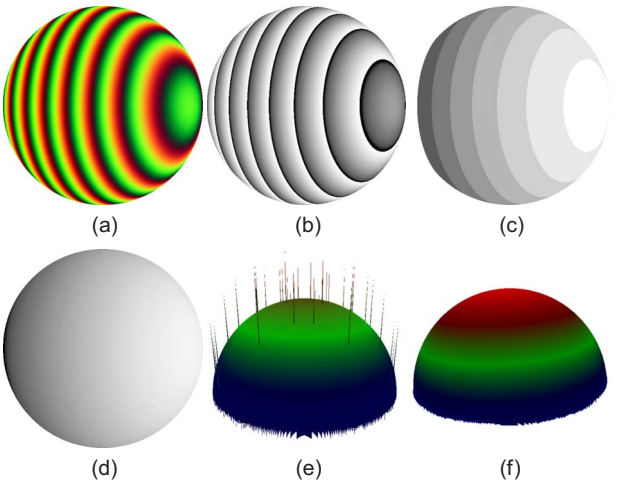


Fig. 4 Three-dimensional recovery using the single color fringe image: (a) fringe image; (b) phase map using red and green channels of the color fringe image; (c) stair images (blue channel); (d) unwrapped absolute phase map; (e) 3-D shape before applying median filtering; and (f) 3-D shape after applying median filtering. (Color online only.)

$$y = j/W, \tag{11}$$

where j is the image index vertically.

From the geometric relation of the diagram in Fig. 3, it is obvious that

$$z = \Delta x/\tan\theta. \tag{12}$$

Combining this equation with Eqs. (9) and (10), we have

$$z = \frac{P\Delta\Phi}{2\pi W \sin\theta}. \tag{13}$$

Finally, the equation governing the z coordinate calculation is

$$z = \frac{P(\Phi - 2\pi i \cos\theta/P)}{2\pi W \sin\theta}, \tag{14}$$

which is a function of the projection angle θ, the fringe pitch P, and the phase Φ obtained from the fringe images.

3 Experiment

To verify the performance of the proposed approach, we first tested a sphere with a diameter of 1 mm (we can use any units since it is normalized into a unit cube), as shown shortly in Fig. 5, whose color fringe image is shown as Fig. 4(a). In this example, we used a stair step height of 5, projection angle of $\theta=30$ deg, and a fringe pitch of $P=16$ pixels. All of the fringe images used in the rest of the paper have exactly the same setup. From the red and green channels, the phase map can be calculated by Eq. (3), which is shown in Fig. 4(b). The blue channel [shown in Fig. 4(c)] is then applied to unwrap the phase map point by point using Eq. (5), an the result is shown in Fig. 4(d). On this unwrapped phase map, there are some artifacts (white dots) that are not clearly shown in this figure [these are seen more clearly in Fig. 4(e)]. Using the phase-to-coordinate conversion algorithm introduced in Sec. 2.3, the

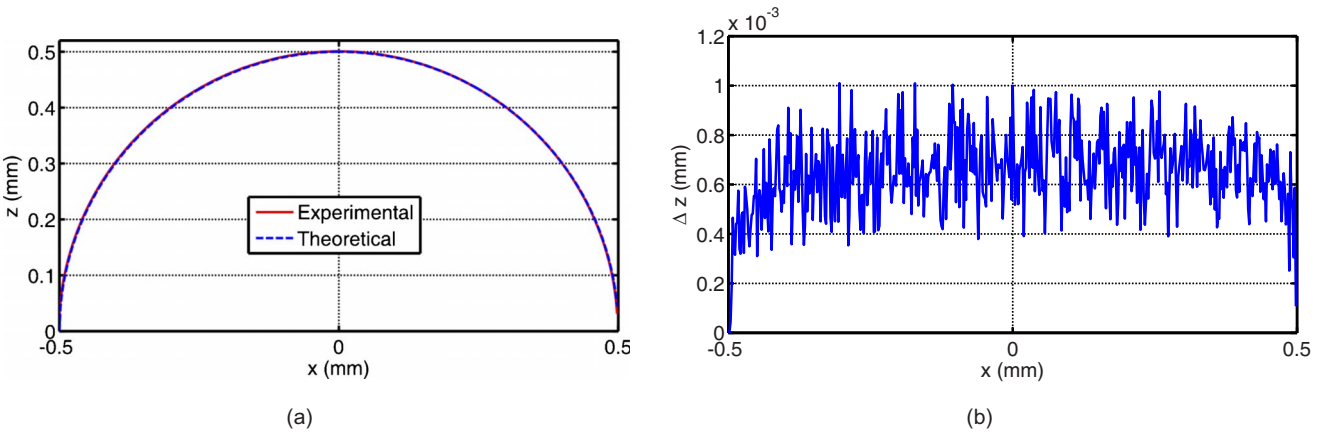


Fig. 5 Comparison between the reconstructed 3-D shape and the theoretical one: (a) cross section of the 256th row and (b) difference [root mean square (rms): 1.68×10^{-4} mm or 0.03%].

phase map can be converted to a 3-D image, which is shown in Fig. 4(e). The artifacts (spikes), which are now more obvious, are caused by the sampling of the projector and the camera. Because the projector and the camera are digital devices, the discrete signal of the fringe images and the stair image introduce a subpixel shift between the jumps. Fortunately, because this shift is limited to 1 pixel either left or right, this problem can be fixed using a conventional image-processing technique, e.g., median filtering in phase domain. Figure 4(f) shows the corrected result.

The cross section of the reconstructed 3-D shape and the theoretical sphere is shown in Fig. 5(a), and the difference is shown in Fig. 5(b). It is very obvious that the difference is negligible.

Because this algorithm enables point-by-point phase unwrapping, it can be used to reconstruct arbitrary shapes of an object with an arbitrary number of steps. To verify this, we tested a step-height surface: a flat object with a deep squared hole. The color image is shown in Fig. 6(a). Even though the object has height variations greater than the step height, the fringe image does not appear to have discontinuities; this is because the virtual system is different from a real 3-D shape measurement system in that the light can pass through objects. The phase map obtained from the fringe images is shown in Fig. 6(b), the phase jumps are very obvious. Because this technique uses the third channel to unwrap the phase, the 3-D shape can be correctly reconstructed, which is shown in Fig. 6(c). This 3-D shape has large height variations, greater than one period of phase range, yet it is correctly reconstructed. Figure 7 shows a cross section of the 3-D shape.

An actual scanned 3-D object was then tested for the proposed algorithm. Figure 8 shows the experimental result. The original shape is shown in Fig. 8(a), the color fringe image is shown in Fig. 8(b), and the unwrapped phase map and the recovered 3-D shape are shown in Figs. 8(c) and 8(d), respectively. If the original shape and the recovered shape are rendered in the same window, the results are shown in Fig. 8(e) in shaded mode and Fig. 8(f) in wireframe mode. This clearly demonstrates that the recovered 3-D shape and the original shape are almost perfectly aligned, that is, the recovered 3-D shape and the original 3-D shape do not have significant difference.

All these experiments demonstrate that the proposed single image technique can be used to represent an arbitrary 3-D surface shape, and thus, can be used for shape compression. We performed further experiments using different image formats and compared the 3-D reconstruction quality. In this research, we tested bitmap, PNG, and differing compression levels of JPG. The typical 3-D surface shown in Fig. 8(a) was used to verify the performance. In a natively binary format (xyzm), a 512×512 3-D surface together, storing x, y, and z coordinates and the mask infor-

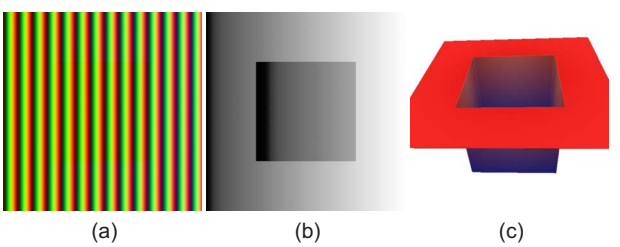


Fig. 6 Our 3-D recovery for step height object: (a) color fringe image, (b) unwrapped phase map, and (c) 3-D shape. (Color online only.)

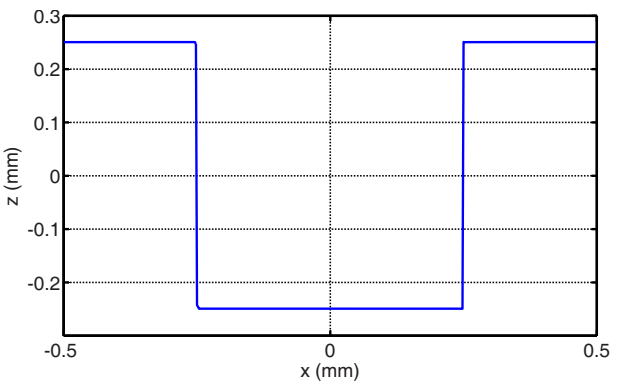


Fig. 7 Cross section of the 256th row of the step height object.

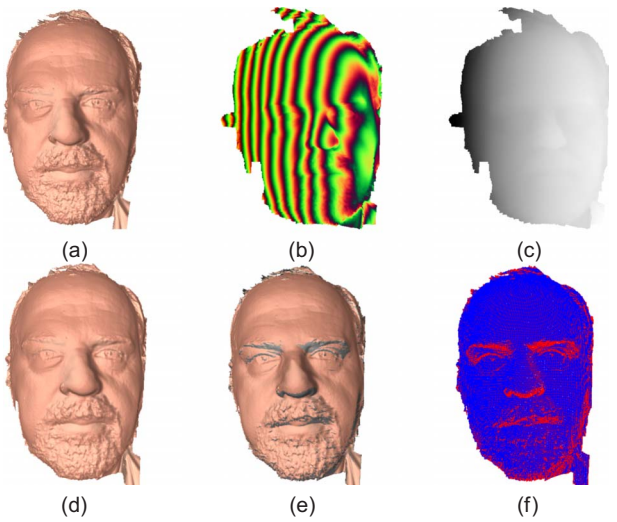


Fig. 8 Results of 3-D recovery using the color fringe image for scanned data: (a) original 3-D scanned data, (b) color fringe image, (c) unwrapped phase map, (d) 3-D reconstructed shape, (e) overlap original 3-D shape (yellow) and the recovered 3-D shape (gray) in shaded mode, and (f) overlap original 3-D shape (blue) and the recovered 3-D shape (red) in shaded mode. (Color online only.)

mation requires at least 3,407,872 bytes (4 bytes floating point for each coordinate, and 1 byte for the mask). Most popular 3-D formats, such as OBJ and STL, use much more space. The bitmap color image has a size of 786,486 bytes, which is approximately 4.33 times smaller. In this experiment, we used the bitmap color image as it is uncompressed lossless.

Figure 9 shows the results. The PNG image format was first used to compress the image data. Since the PNG format is lossless, the original 3-D data can be recovered without any loss, while the file size is reduced to 171,257 bytes (compression ratio of 1:19.90). Figure 9(a) shows the reconstructed 3-D shape and Fig. 9(e) shows the difference between the reconstructed 3-D shape and the original 3-D shape. We can see that there is no difference at all. We found that the color image cannot be directly compressed into JPG format because the third channel (blue) is intolerant of noise. To circumvent this problem, we compressed the red and green channels using a JPG format, while retaining the blue channel in PNG format. In this manner, the file size was reduced to 92,446 bytes, while retaining the 3-D shape quality with a compression ratio 1:36.86. Figures 9(b) and 9(f) show the reconstructed 3-D shape and the difference map, respectively. When we further compressed the red and green channels to a size of 92,192 bytes, the image quality slightly drops, as shown in Figs. 9(c) and 9(g). Note that the boundary dropped more than the inside

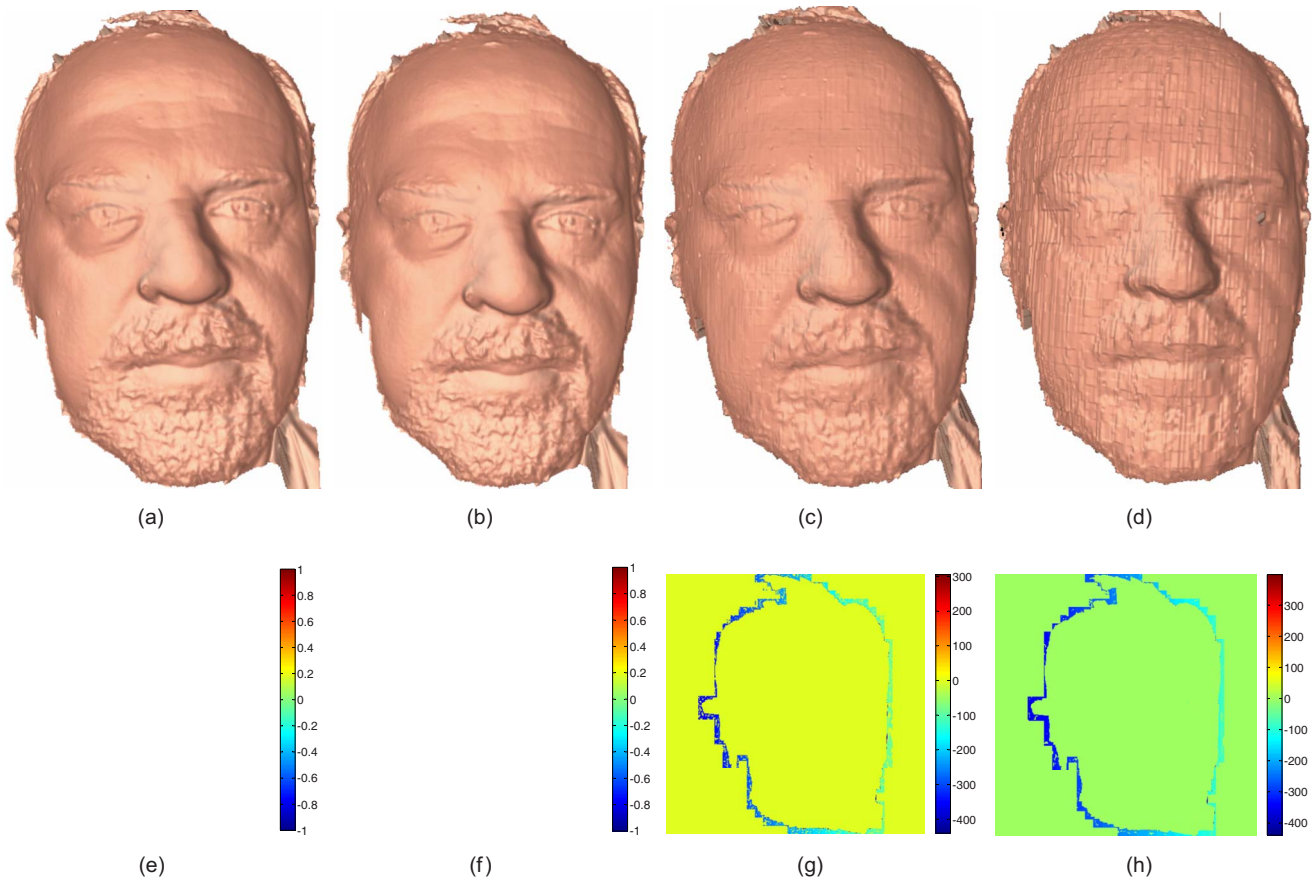


Fig. 9 Results of 3-D reconstruction under different compression ratios: (a) PNG format (1:19.90), (b) JPG+PNG format (1:36.86), (c) JPG+PNG format (1:36.96), and (d) JPG+PNG format (1:41.71). (e)-(h) Error maps of the above corresponding compressed 3–D shapes.

Table 1 Compression comparison of various 3-D formats compared to the Holoimage format.

	Compressed	PNG	xyzm	MAT5	PLY	DAE	OBJ	STL
File size	92 KB	210 KB	3.4 MB	5.5 MB	6.5 MB	10.6 MB	12.8 MB	17.0 MB
Ratio	1:1	1:2.28	1:36.86	1:59.78	1:70.65	1:115.22	1:139.13	1:184.78

Formats contain only vertices and connectivity if required, and are in binary format if applicable to the format; no point normals or texture coordinates are stored.

of the shape, because the boundary has sharp edges. We also demonstrated that when the file size is further reduced to 81,713 bytes, the 3-D shape quality is reduced substantially. The results are show in Figs. 9(d) and 9(h). This experiment showed that a color image can be substantially compressed without losing the data quality.

In addition, we compared the file size with some other commonly used 3-D data formats. Table 1 gives a comparison of various 3-D shape formats. In general, the 3-D data format requires connectivity information (e.g., OBJ, STL), the compression ratio is over 139. Comparing the formats, the native binary format (xyzm) gives the best compression as it was designed specifically to store point cloud data from 3-D scanners disregarding polygon links; even this format is over 36 times larger than a compressed Holoimage.

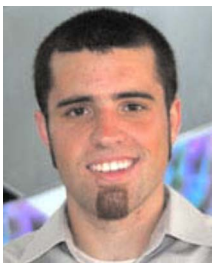
4 Conclusion

We successfully demonstrated that an arbitrary 3-D shape can be represented as a single color image, with the red and green channel being represented as sine and cosine fringe images, and the blue channel encoded as a phase-unwrapping stair function. Storing 3-D geometry in a 2-D color image format enables conventional image compression methods to be employed to compress the 3-D geometry. However, we found that lossy compression algorithms cannot be incorporated because of the third channel containing sharp edges. Lossless image formats, such as PNG or bitmap, must be used to store the blue channel because it contains sharp edges, while the red and green channels can be stored in any image format. Compared with the native smallest possible 3-D data representation method, we demonstrated that with a compression ratio of 1:36.86, the shape quality was not reduced at all. The compression ratio is much larger if other 3-D formats are used.

By compressing 3-D geometry into 24-bit color images, the compression ratio is very high. However, after conversion, the original 3-D data connectivity information is lost and the data are resampled. Note that because the shape reconstruction can be conducted pixel by pixel, our method is suitable for parallel processing, thus enabling real-time shape transmission and visualization. In the future, we will explore higher image compression methods to store the red and green channels, and investigate higher compressed lossless strategies to store blue channel.

References

1. S. Rusinkiewicz, O. Hall-Holt, and M. Levoy, "Real-time 3D model acquisition," *ACM Trans. Graphics* **21**(3), 438–446 (2002).
2. C. Guan, L. G. Hassebrook, and D. L. Lau, "Composite structured light pattern for three-dimensional video," *Opt. Express* **11**(5), 406–417 (2003).
3. L. Zhang, B. Curless, and S. Seitz, "Spacetime stereo: shape recovery for dynamic scenes," in *Proc. Computer Vision and Pattern Recognition*, pp. 367–374, IEEE, Piscataway, NJ (2003).
4. J. Davis, R. Ramamoorthi, and S. Rusinkiewicz, "Spacetime stereo: a unifying framework for depth from triangulation," *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(2), 296–302 (2005).
5. S. Zhang and P. S. Huang, "High-resolution, real-time three-dimensional shape measurement," *Opt. Eng.* **45**, 123601 (2006).
6. S. Zhang and S.-T. Yau, "High-speed three-dimensional shape measurement using a modified two-plus-one phase-shifting algorithm," *Opt. Eng.* **46**(11), 113603 (2007).
7. http://en.wikipedia.org/wiki/List_of_file_formats.
8. http://en.wikipedia.org/wiki/STL_(file_format).
9. http://en.wikipedia.org/wiki/obj.
10. http://www.engr.uky.edu/~lgh/soft/softmat5format.htm.
11. X. Gu, S. Zhang, P. Huang, L. Zhang, S.-T. Yau, and R. Martin, "Holoimages," in *Proc. ACM Solid and Physical Modeling*, pp. 129–138 (2006).
12. D. P. Towers, J. D. C. Jones, and C. E. Towers, "Optimum frequency selection in multi-frequency interferometry," *Opt. Lett.* **28**, 887–889 (2003).
13. C. E. Towers, D. P. Towers, and J. D. C. Jones, "Absolute fringe order calculation using optimised multi-frequency selection in full-field profilometry," *Opt. Lasers Eng.* **43**, 788–800 (2005).
14. Y.-Y. Cheng and J. C. Wyant, "Multiple-wavelength phase shifting interferometry," *Appl. Opt.* **24**, 804–807 (1985).
15. P. K. Upputuri, N. K. Mohan, and M. P. Kothiyal, "Measurement of discontinuous surfaces using multiple-wavelength interferometry," *Opt. Eng.* **48**, 073603 (2009).
16. D. C. Ghiglia and M. D. Pritt, *Two-Dimensional Phase Unwrapping: Theory, Algorithms, and Software*, John Wiley & Sons, New York (1998).



Nikolaus Karpinsky came to Iowa State University in the fall of 2009 seeking his MS degree in human computer interaction. He received his BS degree in software engineering from Milwaukee School of Engineering in the spring of 2009. He is currently working with Dr. Zhang on enhancing the performance of 3-D imaging devices developed by the 3-D Machine Vision Laboratory. His research interests include augmented reality, computer vision, data representation, and human computer interaction.

Song Zhang is an assistant professor of mechanical engineering at Iowa State University. He received his PhD degree in mechanical engineering from Stony Brook University in 2005, and was a postdoctorral fellow with Harvard University from 2005 to 2008. His major research interests include superfast 3-D optical metrology, biophotonic imaging, 3-D machine and computer vision, human computer interaction, and virtual reality. He serves as a reviewer for over a dozen journals.