Natural method for three-dimensional range data compression

Pan Ou[1,2] and Song Zhang[1,*]

[1]Department of Mechanical Engineering, Iowa State University, Ames, Iowa 50011, USA

[2]School of Instrumentation Science and Opto-electronics Engineering, Beihang University, Beijing 100191, China

*Corresponding author: song@iastate.edu

Received 3 December 2012; revised 15 February 2013; accepted 17 February 2013;
posted 20 February 2013 (Doc. ID 181031); published 13 March 2013

Prior studies on converting three-dimensional (3D) range data into regular two-dimensional (2D) color images using virtual fringe projection techniques showed great promise for 3D range data compression, yet they require resampling the raw scanned data. Due to this resampling, the natural 3D range data are altered and sampling error may be introduced. This paper presents a method that compresses the raw sampling points without modifications. Instead of directly utilizing the 3D recovered shape, this method compresses the s map, the scale factor of a perspective projection from a 3D space to a 2D space. The s map is then converted to 2D color image for further compression with existing 2D image compression techniques. By this means, 3D data obtained by 3D range scanners can be compressed into 2D images without any resampling, providing a natural and more accurate method of compressing 3D range data. Experimental results verified the success of the proposed method.

OCIS codes: 120.2650, 100.5070, 100.6890.

1. Introduction

Over the past decades, three-dimensional (3D) range geometry/video scanning techniques have advanced rapidly and shown the great potential to penetrate into our daily life [1,2]. Yet, unlike their two-dimensional (2D) counterparts, storing enormously large 3D raw data becomes an issue since 3D range data compression techniques have not been well established. Therefore, it is an important issue to study on how to store and transport 3D range data in an efficient way [3].

At present, 3D range data are predominantly stored as mesh formats (e.g., STL, OBJ, PLY). These formats are designed to be generic and effective such that they could be used to represent arbitrary topologies, and they usually store (x, y, z) coordinates for each vertex of a polygon, the connectivity between vertices, and sometimes surface normal and (u, v) map. However, these methods require a large storage space (for example, to store 640×480 resolution 3D range data, the OBJ file requires approximately 13 MB storage space even without normal information [4]). Over the years, some mesh compression methods [5–7] have been developed, and they achieved reasonable compression ratios with good qualities. However, they often involve a very time-consuming encoding process (ranging from seconds to minutes, even to hours), making them difficult to be employed for real-time applications.

For a digital fringe projection (DFP) system, 3D shapes are recovered from the phase, making it a natural approach to represent 3D range data with phase maps. This approach has been demonstrated successfully for 3D video communication [8]. However, the achievable compression ratio is limited since at least 4 bytes are required to store the floating-point phase map. To further compress the phase data, one could pack the most significant bits of the phase maps into regular 24 bits red, green, and blue (RGB) images. Yet, this method is limited to use lossless 2D image formats since any error on the most significant bit

1559-128X/13/091857-07$15.00/0

will introduce substantially errors on the recovered 3D shape [3].

Virtually creating a 3D range scanner with advanced computer graphics tools can be used to convert 3D shapes to regular 2D images by virtually rescanning the objects with a virtual DFP technique [4,9,10]. Since 3D shapes can be represented as cosine functions, they can be further compressed, and even be extended to range video compression [3,11]. These techniques achieved great compression ratios without substantially losing data quality. However, because one 8 bit channel spatially encodes fringe orders for temporal phase unwrapping, these techniques are limited to use a finite number of fringe stripes, resulting in relatively low-resolution 2D images for representing 3D geometries. To permit higher spatial resolution representation, we proposed a method that directly encodes depth z into regular color fringe images, such that an arbitrary spatial resolution can be used [12]. However, all these methods require resampling the original 3D geometry, resulting in a loss of information during the resampling process. To reduce sampling errors, higher resolution sampling could be utilized, but this approach drastically compromises the compression ratio [13]. Moreover, the resampling angle becomes vital since any deviation from the scanner's camera viewing angle will result in data loss due to occlusions. Furthermore, because of resampling, the original mesh connectivity information may change, which might be problematic if such information is crucial (e.g., texture mapping).

This paper presents a novel technique to overcome the aforementioned limitations by allowing compressing the raw data points without any modifications. Instead of utilizing 3D recovered geometry, the proposed approach compresses the s map, the scale factor of a perspective projection from a 3D space to a 2D plane. For a given 3D range system, the s map is unique to every measurement, and it can be determined during the 3D reconstruction. Similar to the depth map encoding method proposed in [12], an s map can also be converted to a 2D color image that can be further compressed using existing 2D image compression techniques. Since the s map comes along with the camera image pixel by pixel, the proposed technique permits compressing 3D range data into regular 2D images without any resampling. Therefore, this is a natural and more accurate method for compressing 3D range data. We will present experimental results to verify the performance of this proposed technique.

Section 2 explains the principle of encoding and decoding. Section 3 shows experimental results, and Section 4 summarizes this paper.

2. Principle

A. Principle of Structured-Light Technique

The structured-light method has been extensively adopted in both scientific study and industry due to its flexibility [2]. The structured-light method is similar to a stereo-based method, except that it uses a projector to replace one of the cameras [14]. Over the years, there have been numerous techniques developed to recover 3D shape, among which the DFP technique stands out due to its speed and accuracy. The DFP technique is a special kind of structured-light method where the structured pattern intensity varies sinusoidally.

Figure 1 shows the schematic diagram of a 3D shape measurement system using a DFP technique. The system includes the image acquisition unit (A), the projection unit (C), and the object to be measured (B). These three basic units form a triangulation base. The projector shines vertically straight structured stripes that vary horizontally on the object surface. Each vertical stripe is also called a phase line that is used to establish one of the correspondence constraints between the camera and the projector. The object surface distorts straight phase lines to curved ones if the surface is not flat and if they are perceived from another viewing angle. A camera captures the distorted structured images from an angle differs from the projection angle for 3D reconstruction. For a DFP system, the correspondence is established through the phase carried by the sinusoidal structured patterns. Once the correspondence is established, (x,y,z) coordinates of a point on the object surface can be recovered if both the camera and the projector are calibrated [15].

To calibrate a camera, a pinhole model is usually used. The pinhole camera model describes a camera with intrinsic parameters (e.g., focal length, principal point) and extrinsic parameters describing the rotation and translation between the world coordinate system $(0;x,y,z)$ and the camera coordinate system $(0^c;x^c,y^c,z^c)$ [16]. The imaging process is essentially a projection from a 3D space to a 2D plane, which can be mathematically described as

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A[R,T]\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \tag{1}$$

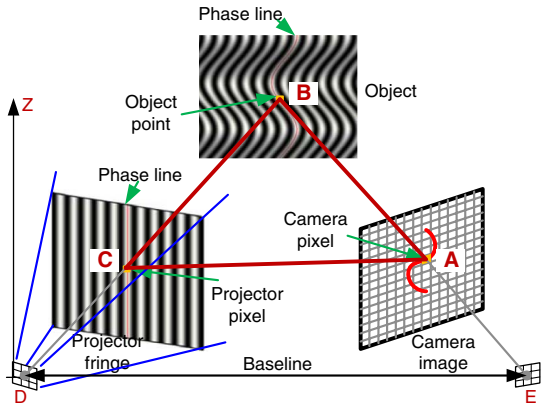


Fig. 1. (Color online) Schematic diagram of a 3D shape measurement using a structured-light-based technique.

where s is a scale factor, $[R,T]$ is the extrinsic matrix, R is the 3×3 rotation matrix, and T is the 3×1 translation vector. A is the camera intrinsic matrix that can be described as

$$A=\begin{bmatrix}\alpha & \gamma & u_0\\ 0 & \beta & v_0\\ 0 & 0 & 1\end{bmatrix}. \tag{2}$$

Here, α and β are, respectively, the focal lengths along x and y axis, γ is the camera skew factor, and (u_0,v_0) is the principal point where the optical axis intersects with the camera sensor. Once the extrinsic and intrinsic matrices are determined through calibration, the projection from the world coordinates (x,y,z) to the camera image coordinates (u,v) is unique.

Since the projector and the camera are optically the same, the projector can also use the same pinhole camera model [15]. The calibration of a structured-light system essentially estimates the intrinsic and extrinsic parameters of the projector and the camera. If the world coordinate system for the camera is defined to be the same as that of the projector, absolute (x,y,z) coordinates can be reconstructed from absolute phase [17].

B. s Map Encoding

Equation (1) indicates that, for a calibrated structured-light system, there is a unique relationship between (x,y,z) coordinates and the scale factor s given (u,v) map, which is the camera image. That is, instead of storing recovered (x,y,z) coordinates, one can store the camera calibration parameters [i.e., A,R,T in Eq. (1)] and s map. Since only s rather than three x, y, and z information are stored, this method reduces the file size to be 1/3 without scarifying any quality. Furthermore, because s map is a natural camera grid, it can be converted to 2D color images for further compression.

To convert s map to a 2D color image, a similar method to that introduced in [12] can be adopted. Instead of converting the resampled and normalized depth z map into regular RGB image, we encode normalized s map, s^n, into three color channels of the regular 2D image as

$$I_r(i,j)=127.5+127.5\sin(2\pi s^n/P), \tag{3}$$

$$I_g(i,j)=127.5+127.5\cos(2\pi s^n/P), \tag{4}$$

$$I_b(i,j)=c\cdot Fl(s^n/P+0.5)+0.5c+0.5(c-2)\cdot\cos\left[2\pi\cdot\frac{\mathrm{Mod}(s^n,P)}{P_1}\right]. \tag{5}$$

Here, P is the fringe period, $P_1=P/(N+0.5)$ is the local fringe period, N is an integer number, $c=255P$ is the stair height in grayscale value, $\mathrm{Mod}(a,b)$ is the modulus operator to get a over b, and $Fl(x)$ is to get the integer number of x by removing the decimals. Since all these equations vary sinusoidally, this compression technique permits high-quality lossy compression [3]. In these equations, the normalized s map, s^n, is determined by

$$s^n=(s-s_{\min})/(s_{\max}-s_{\min}), \tag{6}$$

where $s_{\max}$ and $s_{\min}$ are, respectively, the maximum and the minimum s values.

Solving Eqs. (3) and (4) leads to

$$\phi(i,j)=\tan^{-1}\left(\frac{I_r-127.5}{I_g-127.5}\right). \tag{7}$$

This equation provides the wrapped phase $\phi(i,j)$ ranging from 0 to 2π with 2π discontinuities. The wrapped phase can be unwrapped by combing with the fringe order, $k(i,j)$, information carried by the blue channel,

$$k(i,j)=Fl\left(\frac{I_b}{c}\right). \tag{8}$$

It should be noted that the blue channel encodes a modified stair image to ensure that the stair changes perfectly align with the 2π discontinuities for temporal phase unwrapping, and the normalized s map, s^n, can be uniquely determined for each point by

$$s^n(i,j)=P\left[k(i,j)+\frac{\phi(i,j)}{2\pi}\right]. \tag{9}$$

Once s^n map is obtained, s map can be obtained by a linear transformation,

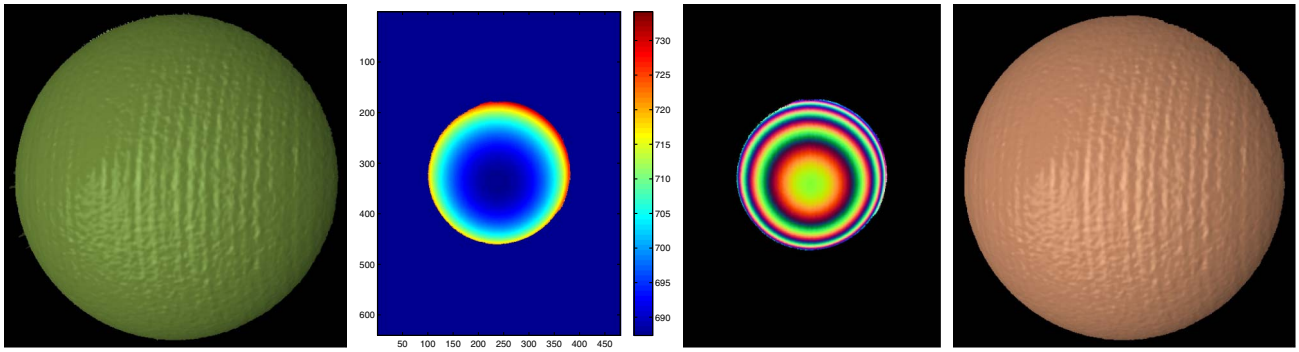
$$s=s^n\cdot(s_{\max}-s_{\min})+s_{\min}. \tag{10}$$

Recovering (x,y,z) coordinates from s map is straightforward. From Eq. (1), we have

$$\begin{bmatrix}x\\ y\\ z\end{bmatrix}=(A\cdot R)^{-1}\cdot s\cdot\begin{bmatrix}u\\ v\\ 1\end{bmatrix}-R^{-1}\cdot T. \tag{11}$$

3. Experiments

In this research, we used a previously developed DFP system to scan the testing objects, and compressed the raw scanned data with the proposed method. The hardware system includes a digital-light-processing projector (Model: Samsung SP-P310MEMX), and a CCD camera (Model: the Imaging Source Digital USB CCD camera DMK 21BU04). The camera uses a 12 mm focal length megapixel lens (Model: Computar M1214-MP) at $F/1.4$ to 16C. The camera resolution is 640×480. The projector has a resolution of 800×600 with a projection distance of 0.49–2.80 m. We employed



(a) (b) (c) (d)

Fig. 2. (Color online) Experimental results of an ideal sphere. (a) Original 3D shape, (b) s map, (c) encoded 2D color image, and (d) recovered 3D shape.

the multiple wavelength phase-shifting algorithm described in [18] for absolute phase capture, and the calibration method discussed in [15] for 3D coordinates recovery.

We first experimented with a spherical object. Figure 2 shows the measurement results. The raw data are shown in Fig. 2(a), whose s map is show in Fig. 2(b). The s map was encoded into a 24 bit color image using Eqs. (3)–(5), and the result is shown in Fig. 2(c). From this encoded color image, the original s map can be recovered, and the original (x, y, z) coordinates can also be reconstructed by combining the camera calibration parameters. Figure 2(d) shows the recovered 3D result if 2D color image is stored as the lossless PNG format. There is no visible difference between the original 3D data and the recovered one.

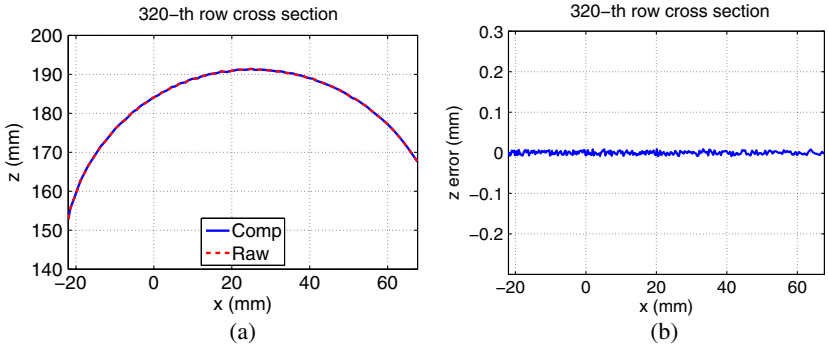
Taking the differences between the recovered 3D shape and the original 3D shape point by point, the error can be quantified. In this research, the error is quantified by the mean distance error that is defined as

$$\Delta_d = \frac{1}{M}\sum_{i=1}^{M}\sqrt{(x_i - x_i^r)^2 + (y_i - y_i^r)^2 + (z_i - z_i^r)^2}. \quad (12)$$

Here, (x, y, z) are the original coordinates, (x^r, y^r, z^r) are the recovered coordinates, and M is the total number of valid measurement points (i.e., the background points are ignored).

Figure 3 shows the comparison between the original 3D shape and the recovered shape. Figure 3(a) plots one cross section of the original geometry and the recovered geometry, which clearly shows that they are almost perfectly aligned. The differences between these two curves were plotted in Fig. 3(b). The difference is very small with a mean distance error (Δ_d) of 0.003 mm for the whole measurement points. It is negligible small for the object with a range of $x \in [-22.98, 70.45]$ mm, $y \in [48.35, 144.07]$ mm, and $z \in [144.79, 191.44]$ mm.

Even without further 2D image compression, the file size is already much smaller comparing with storing the data using the standard 3D mesh formats. If the encoded image is stored as a lossless PNG file format (file size is approximately 114 kbytes), the compression ratio is approximately 52:1 comparing with the STL format. The 2D image can be further stored as lossy image formats, such as the JPG formats with different levels of quality. Figure 4 shows the comparing results when the 2D images were stored as JPG files with quality levels 12, 10, 8, and 6. The JPG quality levels are defined by Adobe Photoshop CS3 with 12 being the best quality. This experiment shows that the encoded color image can be stored as lossy image formats. Of course, higher-quality 2D images will result in less compression ratio but better quality 3D shape recovery. One may also notice that the recovered data have some artifacts if a low-quality JPG is used: spikes on



(a) (b)

Fig. 3. (Color online) Comparison between the recovered sphere and the original sphere. (a) z cross sections and (b) z difference ($\Delta_d = 0.003$ mm).

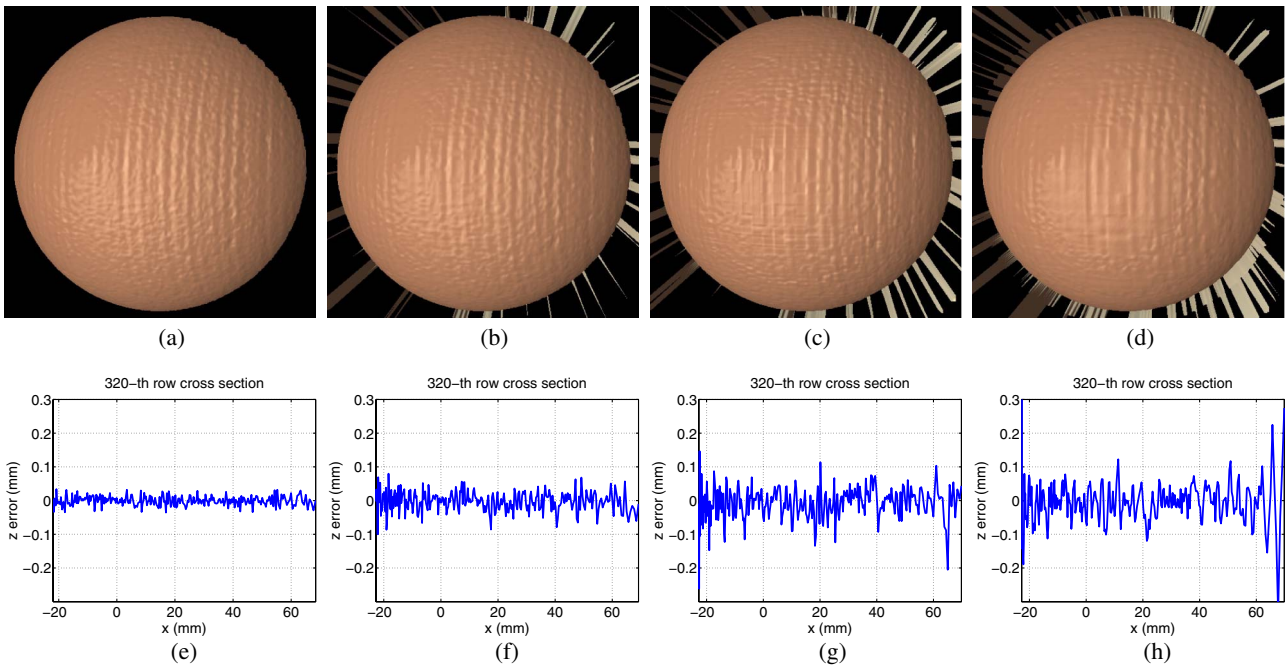


Fig. 4. (Color online) Results of using different quality JPG file formats. (a)–(d) 3D results from JPG file with quality levels 12, 10, 8, and 6, respectively; (e)–(h) z differences for 320-th row of above results comparing with the original 3D data. $\Delta_d = 0.008$, 0.035, 0.063, and 0.192 mm for (e)–(h), respectively.

the boundary. Even with low-quality JPG formats, the overall 3D geometry was still properly recovered. It should be noted that for these 3D results recovered from lossy formats, median filters (9×9 and 5×5) were used to reduce the incorrectly unwrapped phase points.

A more complex 3D object (David head) was also scanned and tested. Figure 5 shows the results. Figures 5(a)–5(c) show the raw scanned 3D shape, the s map, and the encoded 2D color image. Figure 5(d) shows the image that overlays the original 3D shape with the 3D shape recovered from a lossless PNG image. This figure shows that they are almost perfectly aligned. Quantitatively, Δ_d is approximately 0.022 mm, which is, again, negligibly small comparing with the size of 3D statue: $x \in [-30.58, 166.46]$ mm, $y \in [-13.07, 214.93]$ mm, and $z \in [-47.58, 125.38]$ mm. This once again confirmed that the proposed encoding technique can represent the original complex 3D shape with a high quality.

To further compress the data, the encoded 2D color images were compressed with lossy JPG file formats. Again, we used Adobe Photoshop CS3 to save the lossless image to JPG formats with quality levels of 12, 10, 8, and 6. Figure 6 shows the comparing results. Figures 6(a)–6(d) show the recovered 3D results if the 2D image is stored with these quality levels of JPG files. Again, there are artifacts on

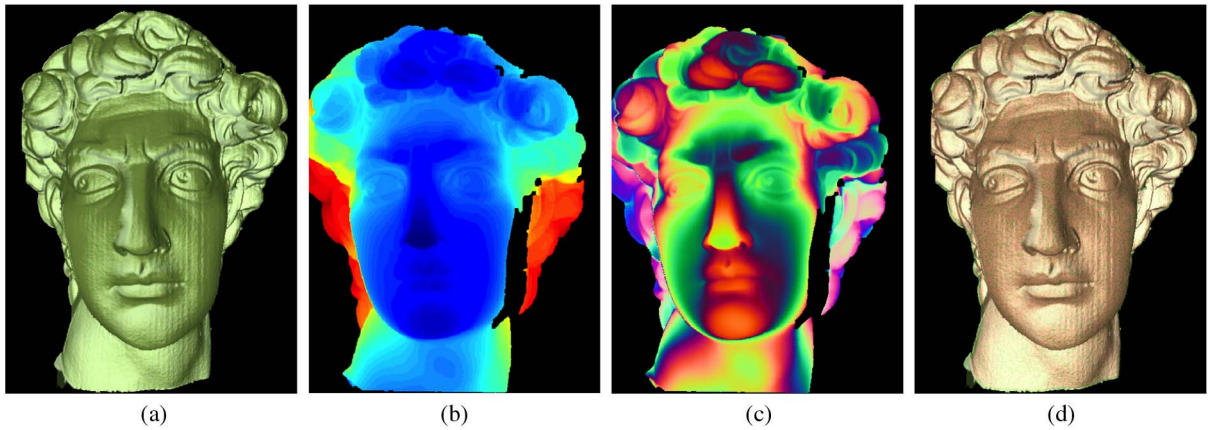


Fig. 5. (Color online) Experimental results of on a complex statue, David head. (a) Original 3D shape, (b) s map, (c) encoded 2D color image, and (d) overlapping original and recovered 3D results. Green color represents the original 3D shape and golden color represents the recovered 3D shape ($\Delta_d = 0.022$ mm).

Fig. 6. (Color online) Experimental results of on a complex statue, David head. (a)–(d) Raw 3D results from JPG images at levels 12, 10, 8, 6, respectively, and $\Delta_d = 0.046$, 0.128, 0.215, and 0.371 mm, respectively; (e)–(h) cleaned 3D results from JPG images at levels 12, 10, 8, 6, respectively, and $\Delta_d = 0.046$, 0.079, 0.114, and 0.153 mm, respectively.

the boundary and detail losses for lower-quality JPG formats, but the overall 3D geometry is all maintained quite well. To better illustrate the recovered geometry, all those boundary spikes were removed by a simple thresholding method. Figures 6(e)–6(h) illustrate the cleaned 3D results with different quality levels of JPG files.

The ultimate goal of this research is to compress complex 3D range data effectively. Table 1 summarizes the achieved compression ratios and the associated errors if the David head data were stored in different mesh formats (i.e., OBJ, PLY, and STL), and in different 2D image formats. These data indicate that (1) even converting 3D range data to a lossless BMP file will substantially save the storage (21.3:1 comparing with the STL format), (2) PNG seems to be a good option if lossless compression is needed, and (3) overwhelming compression ratios can be achieved if the encoded 2D color images are stored in lossy JPG formats. It is important to note that the OBJ and PLY files we used only store the connectivity and vertices information. In other words, texture, *uv* map, and normal map were not preserved. If the *uv* maps were stored, the file size would be drastically larger, and even greater compression ratios could be achieved.

It is important to note that comparing with previously presented methods [4,12], this compression method requires to store camera calibration information [i.e., $(AR)^{-1}$ and $R^{-1}T$] and the range of data ($s_{\max}$ and $s_{\min}$). The background pixels are indicated as black pixels ($R = G = B = 0$), and thus it is not necessary to store such information. Coincidentally, encoding *s*-map directly could more effectively achieve higher accuracy than the method of spatially resampling the data with higher densities. This holds true for structured-light range scanning systems, where the ratio between the depth recovery

Table 1. Compression Ratios for Different Encoded Image Formats Versus 3D Mesh File Formats[a]

	BMP	PNG	JPG12	JPG10	JPG8	JPEG6
OBJ	15.7:1	51.0:1	62.7:1	110:1	159.1:1	213.5:1
PLY	16.1:1	52.4:1	64.4:1	112.9:1	163.4:1	219.3:1
STL	21.3:1	69.3:1	85.2:1	149.5:1	216.2:1	290.3:1
Δ_d (mm)	0.022	0.022	0.046	0.079	0.114	0.153

[a]Mean distance errors Δ_d refer to cleaned results.

error and the lateral extension is a constant. Under an optimal setup, the ratio is approximately 10^{-4} [19,20].

4. Summary

We have presented the idea of naturally encoding 3D range data into regular 2D images without resampling the 3D geometry. Experimental data showed that: (1) high accuracy reconstruction can be achieved if the 2D images are stored as lossless image formats, (2) larger compression ratios can be realized if the image is stored as lossy image formats (e.g., JPG), and (3) even with very low quality lossy JPG compression, the overall 3D shape can still be well preserved. Since this novel compression method takes the s map naturally without modifications, it is a natural way of storing 3D range data without those major limitations suffered by most of the state-of-the-art 3D data compression techniques. Moreover, because this technique is straightforward without intensive computation requirements, this proposed technique has the potential to instantaneously compress and deliver 3D live video contents for a 3D range video acquisition system. Though the linear camera model is used, this proposed method can be extended to the nonlinear camera model by converting the nonlinear to linear model through prior rectification of the camera images before compression. By this means, the same compression techniques can be used. For the 3D shape measurement system we have, because the camera lens is quite good, the nonlinear effect is not obvious, and thus a linear calibration model is sufficient.

References

1. S. Zhang, "Recent progresses on real-time 3-D shape measurement using digital fringe projection techniques," Opt. Lasers Eng. **48**, 149–158 (2010).
2. G. Geng, "Structured-light 3D surface imaging: a tutorial," Adv. Opt. Photon. **3**, 128–160 (2011).
3. N. Karpinsky and S. Zhang, "Holovideo: real-time 3D video encoding and decoding on GPU," Opt. Lasers Eng. **50**, 280–286 (2012).
4. N. Karpinsky and S. Zhang, "Composite phase-shifting algorithm for three-dimensional shape compression," Opt. Eng. **49**, 063604 (2010).
5. S. Gumhold, Z. Kami, M. Isenburg, and H.-P. Seidel, "Predictive point-cloud compression," in *Proceedings of ACM SIGGRAPH 2005 Sketches* (ACM, 2005) p. 137.
6. B. Merry, P. Marais, and J. Gain, "Compression of dense and regular point clouds," Comput. Graph. Forum **25**, 709–716 (2006).
7. R. Schnabel and R. Klein, "Octree-based point-cloud compression," in *Proceedings of the Eurographics Symposium on Point-Based Graphics* (Eurographics, 2006), pp. 111–120.
8. A. Jones, M. Lang, G. Fyffe, X. Yu, J. Busch, I. McDowall, M. Bolas, and P. Debevec, "Achieving eye contact in a one-to-many 3D video teleconferencing system," in SIGGRAPH '09 (ACM, 2009).
9. X. Gu, S. Zhang, L. Zhang, P. Huang, R. Martin, and S.-T. Yau, "Holoimages," in *Proceedings of the ACM Symposium on Solid and Physical Modeling* (ACM, 2006), pp. 129–138.
10. Z. Hou, X. Su, and Q. Zhang, "Virtual structured-light coding for three-dimensional shape data compression," Opt. Lasers Eng. **50**, 844–849 (2012).
11. N. Karpinsky and S. Zhang, "Generalizing holovideo to H.264," Proc. SPIE **8290**, 829012 (2012).
12. S. Zhang, "Three-dimensional range data compression using computer graphics rendering pipeline," Appl. Opt. **51**, 4058–4064 (2012).
13. S. Zhang, L. Ekstrand, T. Grieve, L. S. Chumbley, and M. Morris, "Three-dimensional data processing with advanced computer graphics tools," Proc. SPIE **8493**, 84931 (2012).
14. J. Salvi, S. Fernandez, T. Pribanic, and X. Llado, "State of the art in structured light patterns for surface profilometry," Pattern Recogn. **43**, 2666–2680 (2010).
15. S. Zhang and P. S. Huang, "Novel method for structured light system calibration," Opt. Eng. **45**, 083601 (2006).
16. Z. Zhang, "A flexible new technique for camera calibration," IEEE Trans. Pattern Anal. Mach. Intell. **22**, 1330–1334 (2000).
17. S. Zhang and S.-T. Yau, "High-resolution, real-time 3-D absolute coordinate measurement based on a phase-shifting method," Opt. Express **14**, 2644–2649 (2006).
18. S. Zhang, "Digital multiple-wavelength phase-shifting algorithm," Proc. SPIE **7432**, 74320N (2009).
19. J. Vargas and J. A. Quiroga, "Novel multiresolution approach for an adaptive structured light system," Opt. Eng. **47**, 023601 (2008).
20. J. Vargas and J. A. Quiroga, "Multiresolution approach based on projection matrices," Appl. Opt. **48**, 1295–1302 (2009).