Three-dimensional data merging using Holoimage

Song Zhang
Shing-Tung Yau
Harvard University
Department of Mathematics
Cambridge, Massachusetts 02138
E-mail: szhang77@gmail.com

Abstract. Three-dimensional data merging is vital for full-field three-dimensional (3D) shape measurement. All 3D range data patches, acquired from either different sensors or the same sensor in different viewing angles, have to be merged into a single piece to facilitate future data analysis. A novel method for 3D data merging using Holoimage is proposed. Similar to the 3D shape measurement system using a phase-shifting method, Holoimage is a phase-shifting–based computer synthesized fringe image. The 3D information is retrieved from Holoimage using a phase-shifting method. If two patches of 3D data with overlapping areas are rendered by OpenGL, the overlapping areas are resolved by the graphics pipeline, that is, only the front geometry can be visualized. Therefore, the merging is performed if the front geometry information can be obtained. Holoimage is to obtain the front geometry by projecting the fringe patterns onto the rendered scene. We also demonstrated that each point of the geometry in the overlapping area can be obtained by averaging the corresponding point of the geometries reconstructed from Holoimage for each patch. Moreover, using Holoimage, the texture can also be obtained. Both simulation and experiments demonstrated the success of the proposed method. [DOI: 10.1117/1.2898902]

Subject terms: holoimage; merging; phase shifting; measurement; 3-D range data.

Paper 070789R received Sep. 20, 2007; revised manuscript received Nov. 19, 2007; accepted for publication Nov. 22, 2007; published online Mar. 21, 2008. This paper is a revision of a paper presented at the SPIE conference on Two- and Three-Dimensional Methods for Inspection and Metrology V, September 2007, Boston, Massachusetts. The paper presented there appears (unrefereed) in SPIE Proceedings Vol. 6762.

1 Introduction

With the development of the range three-dimensional (3D) scanning techniques, full-field panoramic 3D shape measurement is increasingly important, with broad applications including entertainment, manufacturing, and reverse engineering. In general, to obtain full-field panoramic 3D shape measurement, multiview range scanning is unavoidable in situations where merging 3D data patches from different views is critical.

The 3D range data patches, acquired either from different sensors or from the same sensor in different viewing angles, have to be merged into a single piece to facilitate future data analysis. In general, two pieces of 3D data have to be registered before data merging. In this work, we assume that the 3D data are properly registered and only discuss the merging technique. The 3D data merging merges different pieces of 3D data into a single piece by resolving the overlapping areas between pieces. To resolve the overlapping area, one approach is to detect the closest points between difference pieces, such as using the iterative-closest-point (ICP) algorithm,[1,2] and then unify into one point. However, it is very difficult for this approach to guarantee the surface smoothness because one point on one surface might correspond to multiple points on the other. Various methods have been explored by different researchers.[3,4]

0091-3286/2008/$25.00 © 2008 SPIE

However, most of the existing methods have their shortcomings. For example, the volumetric range image processing (VRIP)[3] method loses useful information and generates undesirable holes in the resultant data. We also found that GSI Studio (Geometry System Inc., San Ramon, California) produces a similar result as VRIP does. The method Hu et al.[4] proposed tries to register the geometries from different patches while keeping the raw data point clouds and, therefore, has redundant data after merging.

In this research, we propose a novel method called *Holoimage* for 3D data merging. Holoimage is a novel geometric representation introduced by Gu et al.[5] It encodes both shading and geometry information within the same image. Similar to the fringe images captured in the real world, Holoimage is a phase-shifted fringe image synthesized by the computer. The virtual projector projects the sinusoidal phase-shifted fringe patterns onto the object, the reflected fringe patterns are rendered on the screen to generate the Holoimage. The phase-shifting algorithm is used to reconstruct 3D geometry from the Holoimage once the parameters of the virtual system are known. For the modern graphics pipeline, if two patches of 3D data with overlapping areas are rendered together by OpenGL, the overlapping areas are resolved and only the front geometry is visible. Therefore, the 3D data merging is done if the front geometry is obtained. Holoimage obtains the front 3D geometry by projecting the fringe patterns onto the rendered scene. Unlike the real world, the virtual camera and projector can be used as orthogonal projective devices, and the setup of the system can be controlled accurately and easily. Both simulation and experiments demonstrated the success of the proposed method.

Section 2 addresses the principle of the Holoimage system for 3D data merging. Section 3 shows simulation results. Section 4 shows experimental results from the real 3D data acquired by a range scanner. Section 5addresses the advantages and shortcomings of the Holoimage system, and Section 6 summarizes this work.

2 Principle

2.1 *Three-Step Phase-Shifting Algorithm*

Phase-shifting algorithms are extensively adopted for accurate and rapid 3D shape measurement. Different phase-shifting algorithms including three-, four-, and five-step phase-shifting algorithms have been developed.[6] A three-step phase shifting algorithm with a phase-shift of $2\pi/3$ can be written as

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

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

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

where $I'(x,y)$ is the average intensity, $I''(x,y)$ the intensity modulation, and $\Phi(x,y)$ the phase to be resolved. Phase $\Phi(x,y)$ can be resolved from Eqs. (1)–(3)

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

The 3D information can be retrieved from the phase $\Phi(x,y)$. The value of phase $\Phi(x,y)$ obtained from Eq. (4) ranges from $-\pi$ to π. A phase unwrapping algorithm can be used to generate the continuous phase map.[7] The 3D information can be obtained from a simple phase-to-height conversion algorithm using a reference plane.[8] For a real shape measurement system, more accurate 3D coordinates can be obtained by calibrating the system accurately.[9,10]

2.2 *Holoimage*

Similar to a real 3D shape measurement system using a phase-shifting method to measure the real object in 3D, Holoimage is a technique that is used to virtually measure the 3D object synthesized (or rendered) by the computer. It is a novel geometric representation introduced by Gu et al.,[5] and it encodes both shading and geometry information within the same image. In this research, the Holoimage is a three-step phase-shifted fringe image synthesized by the computer using modern graphics pipeline.

It is easy to synthesize a Holoimage using a modern graphics pipeline. Three sinusoidal fringe patterns can be precomputed and stored as a 3-channel 24-bit color texture image. To simplify the analysis of the Holoimage system, a canonical configuration is preferred, where both the projective texture and the camera use orthogonal projection, and the geometric object is normalized to be inside a unit cube. The vertical-stripe color Holoimage image is encoded as

$$I_r(i,j) = 255/2[1 + \cos(2\pi j/P - 2\pi/3)], \tag{5}$$

$$I_g(i,j) = 255/2[1 + \cos(2\pi j/P)], \tag{6}$$

$$I_b(i,j) = 255/2[1 + \cos(2\pi j/P + 2\pi/3)], \tag{7}$$

where P is the fringe pitch, or number of pixels per fringe period. The projected fringe images are virtually distorted by the 3D object. The Holoimage is generated by recording the rendered scene. The following three equations represent the three channels of the color Holoimage:

$$I_r(x,y) = I'(x,y) + I''(x,y)\cos[\Phi(x,y) - 2\pi/3], \tag{8}$$

$$I_g(x,y) = I'(x,y) + I''(x,y)\cos[\Phi(x,y)], \tag{9}$$

$$I_b(x,y) = I'(x,y) + I''(x,y)\cos(\Phi(x,y) + 2\pi/3), \tag{10}$$

From Eq. (4), we have

$$\Phi(x,y) = \tan^{-1}\left[\frac{\sqrt{3}(I_r - I_b)}{2I_g - I_r - I_b}\right]. \tag{11}$$

Similarly, the phase unwrapping step is needed to generate the continuous phase map, and a phase-to-coordinate conversion algorithm is required to reconstruct the 3D information. However, unlike the real measurement system, each device (camera or projector) of the Holoimage system can be controlled easily. Therefore, the phase-to-coordinate conversion algorithm is very simple and accurate. Section 2.3 will introduce the phase-to-coordinate conversion methods.

In practice, if only geometric information is required, the OpenGL texture environment can be set to *replace* using glTexEnvf(GL_TEXTURE, GL_TEXTURE_ENV_MODE, GL_REPLACE). If both geometry and shading information are required, the texture environment should be set as *modulate* using glTexEnvf(GL_TEXTURE, GL_TEXTURE_ENV_MODE, GL_MODULATE). If a texture is also to be rendered on the surface, we need to use a multitexturing technique to generate the Holoimage. Of course, if color texture is desirable, the color Holoimage may have errors where the monochromatic fringe image may have to be adopted.

Figure 1 shows a typical setup of a Holoimage system that uses orthogonal projection for both the camera and the projector. The projector projects fringe patterns orthogonally onto the object, and the camera viewing from another angle captures the deformed images orthogonally.

2.3 *Phase-to-Height Conversion*

In our Holoimage system, both the camera and the projector are treated as orthogonal projection devices, that is, the focal length of the projector f^p and that of the camera f^c are ∞.

2.3.1 *Reference-plane–based method*

To convert the phase to depth, and thereby generate coordinates, we first use a reference-plane–based method. Figure 2 shows the diagram of the Holoimage system. Before

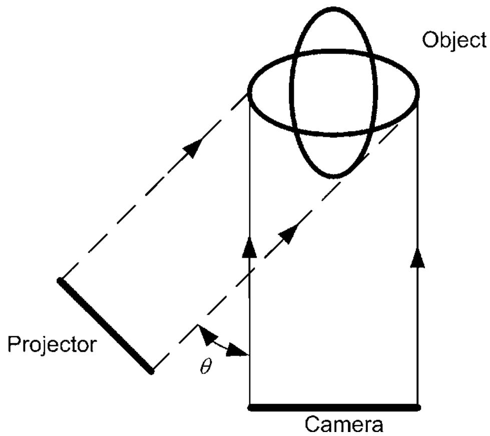


Fig. 1 Setup of Holoimage system.

any measurement, a reference plane is measured. The reference plane is a plane with height 0 in the depth, or z, direction. The subsequent measurement is relative to the reference plane. An arbitrary point K in the captured image corresponds to point A on the reference plane and B on the object surface. The distance between A and B is the depth z of the measured object, $z=\overline{AB}$. The corresponding phase for the reference plane is Φ_A if no object is placed, and it is Φ_B when the object is there. From the projector's point of view, phase Φ_B on the object is the same as Φ_C on the reference plane, that is, $\Phi_C=\Phi_B$. The phase difference between point A and C is $\Delta\Phi=\Phi_A-\Phi_C$. Because the fringe image is uniformly distributed on the reference plane, the actual distance is proportional to the phase difference of the fringe images. In other words, $\overline{AC}=k\Delta\Phi$. Because triangle ΔABC is a right triangle, we have

$$z=\frac{\overline{AC}}{\tan\theta}=k\frac{\Delta\Phi}{\tan\theta}=k\frac{\Phi_A-\Phi_C}{\tan\theta}=k\frac{\Phi_A-\Phi_B}{\tan\theta}.$$

Assume the projection fringe has a fringe pitch of P. Once it is projected onto the reference plane from an angle of θ, it becomes $P_r=P/\cos(\theta)$. Assume the size of the pixel in actual dimension is c mm, which is determined by the projection window size and the setup of the OpenGL environment. The constant k, therefore, becomes

$$k=c\frac{P_r}{2\pi}=c\frac{P}{2\pi\cos(\theta)}.$$

Hence,

$$z=c\frac{P(\Phi_A-\Phi_B)}{2\pi\sin(\theta)}. \tag{12}$$

The x and y values are proportional to their indexes in the x and y directions with a constant of pixel size c. If the data is always normalized into a unit cube ($0\leqslant x\leqslant 1$, $0\leqslant y\leqslant 1$, and $0\leqslant z\leqslant 1$), the fringe stripes are vertical, and the resolution of the Holoimage is $W\times H$

$$c=\frac{1}{W}. \tag{13}$$

Because the data is all normalized into a unit cube, the x and y coordinates for each pixel (i,j) become

$$x=\frac{j}{W}, \tag{14}$$

$$y=\frac{i}{H}. \tag{15}$$

It should be noted that there is no approximation involved. Therefore, the phase-to-height conversion method is accurate. In contrast, a similar phase-to-height conversion method used in a real 3D measurement system[8] is only an approximation, which produces error.[9]

2.3.2 *Absolute-phase–based method*

For the reference-plane–based method, the phase map of the reference plane is precomputed from the captured fringe images and stored for future measurement. It is good for a real measurement system because it can reduce some systematic error of the system, such as the error caused by the nonlinearity of the projector lens and the camera lens. However, it requires loading the reference plane phase map for any measurement. For this ideal Holoimage system setup, it is not necessary to use the reference plane to convert the phase to depth. In this section, we will introduce a new method that converts the absolute phase to depth without the use of the reference plane.

If the fringe image has vertical stripes, then the phase map of the reference plan can be written as

$$\Phi_r(i,j)=2k\pi+\frac{2\pi}{P_r}j, \tag{16}$$

where $2k\pi$ describes the 2π differences of the reference plane, which is related to the starting point of the phase-unwrapping step. If at least one point (i_0,j_0) on the reference plane has a known phase value Φ_{r0}, then

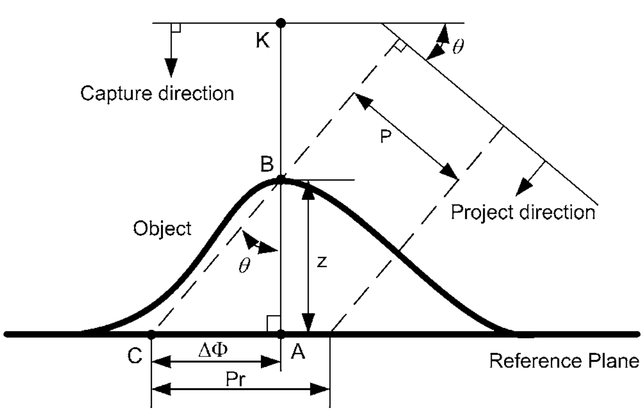


Fig. 2 Schematic diagram of a Holoimage image using orthogonal projection model.

$$\Phi_r(i_0,j_0)=\Phi_{r0}=\frac{2\pi j_0}{P_r}, \tag{17}$$

and the phase map of the reference plane is uniquely determined. That is,

$$k=\frac{\Phi_{r0}-\dfrac{2\pi}{P_r}j}{2\pi}. \tag{18}$$

Then, the absolute phase map of the reference plane becomes

$$\Phi_{ar}(i,j)=\frac{2\pi}{Pr}(j-j_0). \tag{19}$$

Similarly, we can obtain the absolute phase map for any measured object

$$\Phi_a(i,j)=\Phi(i,j)-\Phi_0. \tag{20}$$

Here Φ_0 is determined by projecting one or more marker points with known phase values, and detect them from the camera-captured Holoimages. In this research, a simple "cross" marker is encoded into the projected fringe images. By detecting the cross center of the Holoimage, the absolute phase value on the measured object can be computed. If the center of the cross marker is the center of the projected fringe image, then j_0 in Eq. (19) is half of the image width.

Because the same marker point on the projected fringe image is used for the reference plane and any measured object, from the projected fringe image point of view, the phase value for the marker point is always the same, that is, $\Phi_{r0}=\Phi_0$. Therefore, the phase difference between the measured object and the reference plan becomes

$$\Delta\Phi(i,j)=\Phi(i,j)-\Phi_r(i,j)=\Phi_a(i,j)-\Phi_{ar}(i,j). \tag{21}$$

Once the phase difference is known, depth z can be computed using Eq. (12),

$$z=cP\frac{\Delta\Phi}{2\pi\sin(\theta)}. \tag{22}$$

From Eqs. (19) and (22), we have

$$z=c\frac{P}{2\pi\sin(\theta)}\left[\phi_a-\frac{2\pi(i-i_0)\cos(\theta)}{P}\right]. \tag{23}$$

Similarly, if the data is normalized into a unit cube, the x and y coordinates are

$$x=\frac{j}{W}, \tag{24}$$

$$y=\frac{i}{H}. \tag{25}$$

In Eq. (23), c is $1/W$ if the fringes are vertical stripes.

The advantage of using the absolute phase-to-depth conversion method is that it only uses one single-color image to represent the whole geometry data. It is good for geometric data storage as well as geometric data communication.

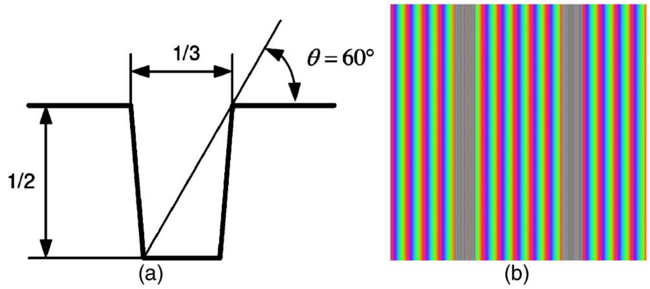


Fig. 3 A 3D surface with a deep hole. (a) Cross section of 3D surface. (b) Holoimage.

2.4 *Holoimage System Versus Real System*

There are fundamental differences between the synthesized Holoimage and the captured fringe images in the real world. The major advantages of a Holoimage system over a real 3D shape measurement system are:

- No shadow or self-occlusion: The projective texture mapping (similar to the projector in the real world) of a synthetic Holoimage does not include shadows or self-occlusion. Figure 3 shows an example. The left image shows the cross section of the 3D object, the right image shows the Holoimage with the projection angle of 60 deg. The bottom of the notch cannot be illuminated by a real projector; however, it can be illuminated by the virtual projector.
- No color coupling: Three monochromatic fringe projective textures can be combined into one color projective texture to generate one color 24-bit Holoimage. Because each color channel of the Holoimage system is separate, using color to represent three phase-shifted fringe images will not cause any problems. However, using color fringe images in a real measurement system is undesirable because the measurement accuracy is affected by the color of the object due to the color coupling problem of the projector and the camera. Therefore, three monochromatic fringe images are usually needed to reconstruct the geometry.
- No complex system calibration: The parameters of the projector and the camera, as well as their relationships, are accurately controlled and known; therefore, no calibration is required for 3D reconstruction for a Holoimage system. However, for a real measurement system, it usually involves time-consuming complex system calibration procedures, and the measurement accuracy highly depends on the calibration accuracy.
- Simple phase-to-height conversion: The phase-to-height conversion is simple and accurate without any approximation. However, if the same phase-to-height conversion algorithm is used for a real 3D shape measurement system, it will introduce significant error,[9] especially when the measured object has a relatively large depth range.

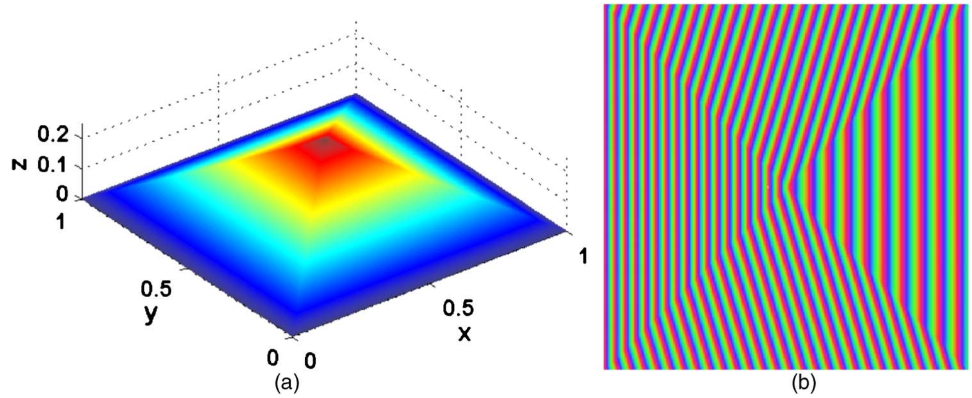


Fig. 4 Pyramid and the corresponding Holoimage ($P=30$; $\theta=30$ deg; image size: 512×512). (a) A 3D plot of the pyramid. (b) Holoimage.

- Linearity: Both the projector and the camera of the Holoimage system can be treated as exact linear devices; therefore, the error caused by sensor nonlinearity does not exist for such a system. In contrast, a real system using a projector always has a nonlinear gamma of the projector.

In the meantime, the Holoimage system has some shortcomings when compared with the real measurement system. For a real 3D shape measurement system based on digital fringe projection and phase-shifting method, the projector can be defocused so that the projected fringe images can be regarded as ideal analog sinusoidal fringe images. Therefore, the measurement resolution is only dependent on the camera resolution, that is, the digitation error is only introduced by the camera. However, for the Holoimage system, both the projector and the camera are always digital, and the error is caused by the digitation introduced by both the projector and the camera. Therefore, the digitalization error is larger than with the real measurement system, albeit the digitation error is very small.

2.5 *Merging Procedure*

It is easy to merge the 3D data patches using Holoimage. The procedures are

1. Normalize 3D data: Normalize 3D data patches S^k by translating and scaling of the original data into a unit cube: $s^k(x,y,z)=S^k[c(X-x_0),c(Y-y_0),c(Z-z_0)]$, so that $0\leqslant x\leqslant 1$, $0\leqslant y\leqslant 1$. Here c is a scaling constant for all geometries and (x_0,y_0,z_0) is the center of all geometries.
2. Generate Holoimage: By using the graphics pipeline, rendering all geometries into the same scene, and projecting fringe images onto the rendered scene, the Holimage of different patches is generated.
3. Obtain absolute phase map: Compute the absolute phase by using phase wrapping, phase unwrapping, and relative-phase-to-absolute-phase conversion algorithms.
4. Obtain normalized merged 3D coordinates: Compute the normalized absolute coordinates using the absolute phase-to-coordinates conversion algorithm addressed in Sec. 2.3.2.
5. Transform normalized coordinates to original coordinates: Retrieve the original coordinates by translating and scaling the normalized coordinates $S^k(X,Y,Z)=s^k(x/c+x_0,y/c+y_0,z/c+z_0)$.

3 Simulation

To show the capability of the 3D reconstruction using Holoimage, we use the computer to generate a simple geometric shape object, a pyramid; use Holoimage to reconstruct it; and compare the reconstructed 3D result with the ideal one.

Figure 4 shows the 3D shape and the corresponding Holoimage of the pyramid. The pyramid has the height of 0.25 and bottom size of 1×1. Figure 5 shows the cross section of the pyramid. The measurement error of depth z is approximately 2.7×10^{-5} rms, which is very small in comparison with digitization error.

We then use Holoimage to merge two surfaces. We simulate two sinusoidal profile surfaces, $S1: z=0.25\sin(2\pi x/P)$ and $S2: z=-0.25\sin(2\pi x/P)$, and merge them using the Holoimage, as shown in Fig. 6. In this example, we used a sinusoidal period of $P=256$ and $0\leq x\leq 1$. The 3D data merging error is approximately 2.9×10^{-5} rms, which is slightly larger than in the previous example. This example and the previous example both show overall shift approximately to the order of 10^{-4}. This is caused by the error of cross marker detection, because the cross detection can only be accurate to the pixel level, which caused error. Figure 6(e) shows the zoom-in view of the region $0.744\leq x\leq 0.755$.

4 Experiments

In this research, we measured different patches of the object using our previously developed 3D shape measurement system.[11] This system is based on a digital fringe projection and phase-shifting method, it can measure absolute coordinates of the object at 30 frames per second.

Figure 7 shows the result of merging two pieces of 3D data from different viewing angles. Figure 7(a) shows the first geometry, and Figure 7(b) shows the second geometry. Two sets of the 3D data are rendered together in the same scene using OpenGL as shown in Figure 7(c), where the

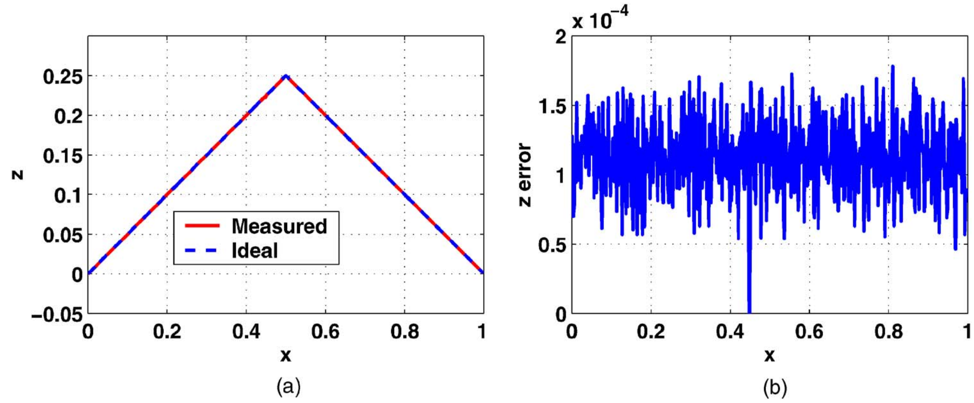


Fig. 5 (a) Cross section of the pyramidic shape object. The solid line represents the measured data and the dashed line represents the ideal line. (b) Difference between the measured data and the ideal data (RMS error: 2.7×10^{-5}).

brighter color represents the first geometry and the darker color represents the second geometry. It can been seen here that only the front geometry is visible. The color fringe images are projected onto the object virtually using the orthogonal projection, the Holoimage is shown in Fig. 7(d). Figure 7(e) shows the phase map of the Holoimage. Figures 7(f)–7(h) show the merged results. It can been seen here that the 3D geometries are merged well.

To verify the quality of the merged data, we plot one horizontal cross section of the 3D data before and after merging (280th row for the merged 3D data and $93.93 \leq y \leq 95.27$ for the original 3D data). Figure 8 shows the result. It can be seen clearly that the 3D surface after merging represents the overlapped 3D geometry very well.

All previous examples show the merging result of the top geometry (i.e., the geometry closer to the camera); however, obtaining averaged geometry in the overlapping area is also useful, which is also very easy to realize using Holoimage. Because two geometries are rendered in the same scene setup, Holoimage can be generated for each

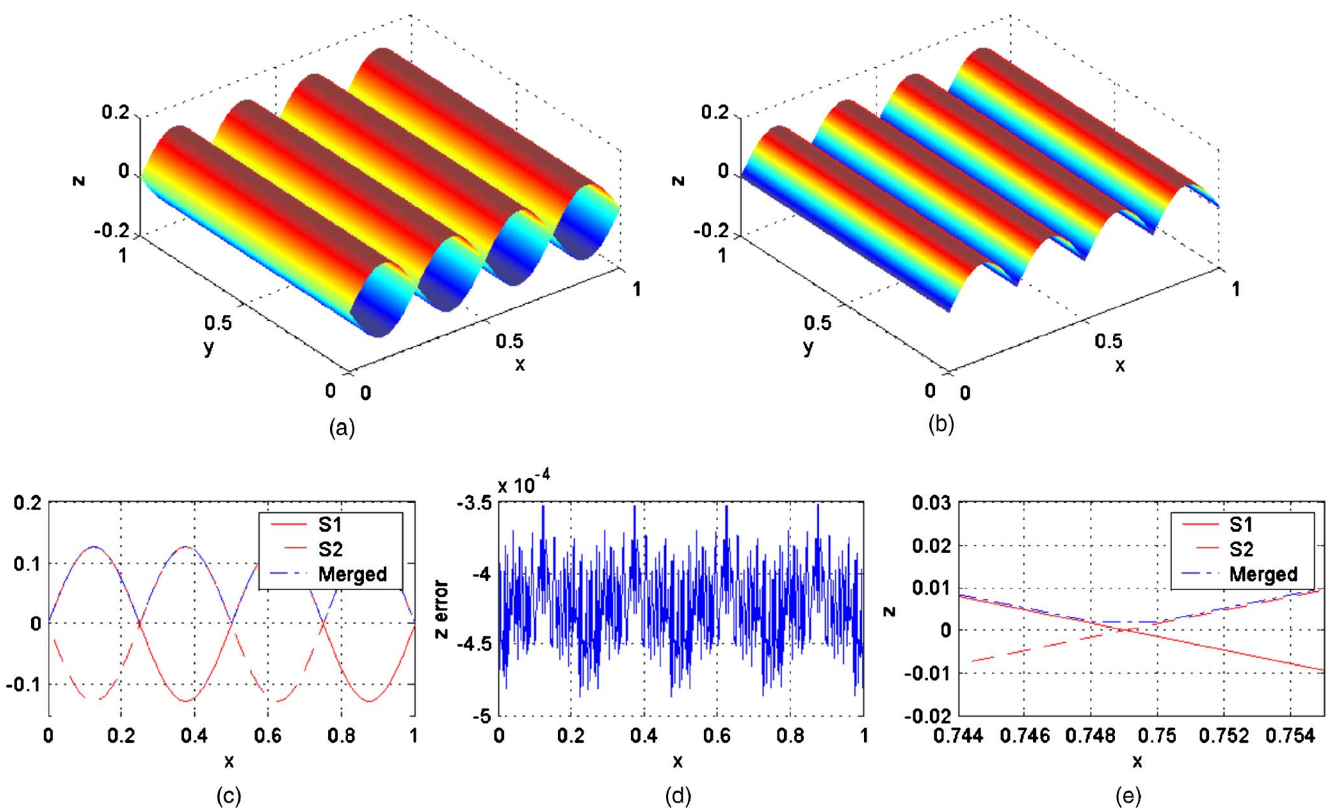


Fig. 6 Overlapping of two sinusoidal surfaces ($P=30$; $\theta=30$ deg; image size: 512×512). (a) Cross section of two planar surfaces before merging and after merging. (b) Error before merging and after merging (RMS error: 2.9×10^{-5}).

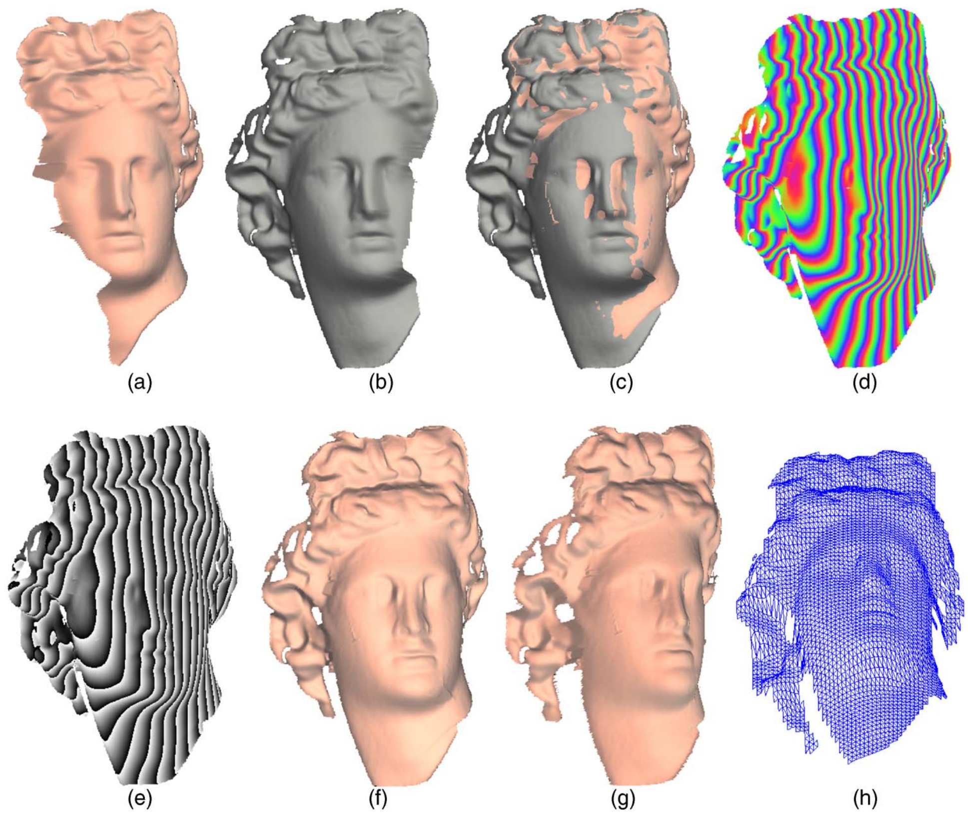


Fig. 7 The 3D data merging using Holoimage ($P=30$; $\theta=30$ deg; image size: 512×512). (a) The first geometry. (b) The second geometry. (c) Two geometries are rendered together in the same OpenGL scene. (d) Holoimage. (e) Phase computed from the Holoimage. (f) to (h) The resultant 3D geometry after merging in different viewing angles and modes.

individual geometry by removing the other. Once two geometries are reconstructed using Holoimage, the resultant merged geometry can be obtained from the two geometries: for the overlapping area, averaging the geometry pixel-by-pixel, and for the remaining region, taking the one that has geometry information. Figure 9 shows an example.

Moreover, the texture can also be obtained using Holoimage. Figure 10 shows an example. It can be seen that the texture is preserved well. Similarly, for the overlapping area, the texture information can be averaged from the texture of both geometries.

5 Discussion

As addressed above, the proposed Holimage system is an efficient and accurate method for 3D data merging. The major advantages of the proposed method are

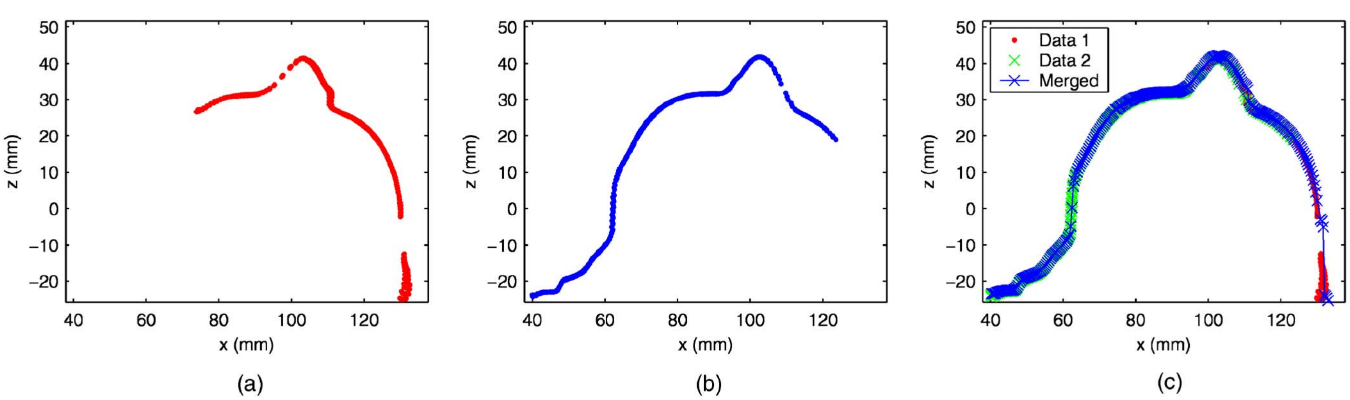
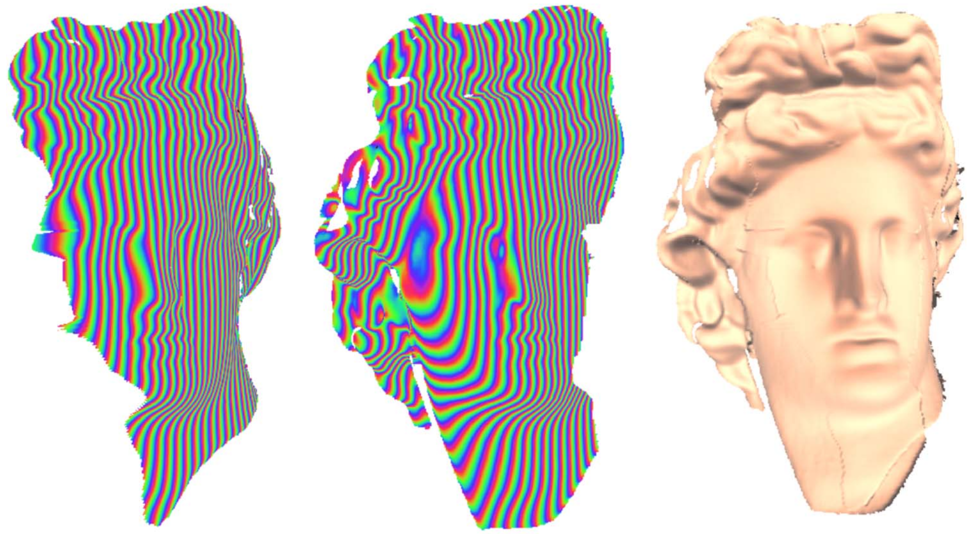


Fig. 8 280th ($93.93\leq y\leq 95.27$) row cross-section view before and after merging. (a) Cross section of the first geometry. (b) Cross section of the second geometry. (c) Cross section of two geometries and the merged result.



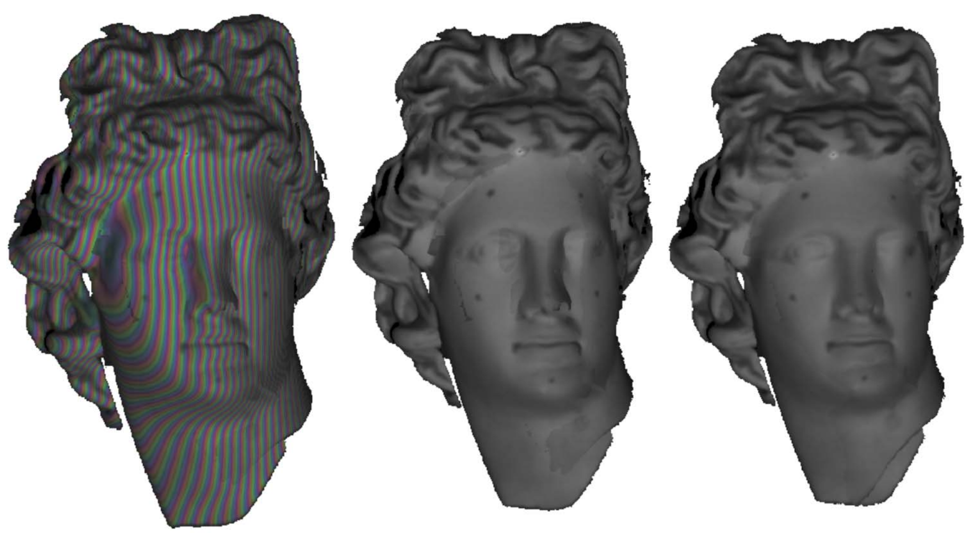
(a) (b) (c)

Fig. 9 Merging geometry by averaging overlapping area point by point. (a) Holoimage of the first 3D geometry. (b) Holoimage of the second 3D geometry. (c) Merged 3D geometry.

- Simple system setup: Both the projector and the camera are treated as an orthogonal projective device and all parameters of the system can be controlled accurately and easily. Therefore, no system calibration is needed, the phase-to-coordinate conversion algorithm is simple and efficient.
- High accuracy: Both simulation and experiments prove that this method can preserve the original geometry with very high accuracy.
- Fast processing speed: Because this Holoimage is an ideal sinusoidal fringe image with three-step phase shifting, the 3D data reconstruction can be realized in real-time once the real-time 3D shape measurement algorithms (phase wrapping,[12] phase unwrapping,[13] and absolute coordinate computation[11]) are employed.
- Both geometry and texture: Holoimage can obtain both geometry and texture information for the merged data easily.
- Small data size: Using a single 24-bit color fringe image to represent a geometry data, the file size is very small. Table 1 shows the comparison between the data size of the Holoimage and other popular data formats (OBJ and PLY). The data used is the merging result as shown in Fig. 7(f). This can be a good method for 3D data compression, 3D data transmission, and such.

6 Conclusions

We have presented a 3D data merging technique using Holoimage. For the modern graphics pipeline, if two different



(a) (b) (c)

Fig. 10 Preserving textures after merging. (a) Holoimage with texture. (b) Merged result with texture. (c) Merged result with texture by averaging the overlapping area.

Table 1 Data size comparison between different file format.

Total Vertices	Holoimage (bytes)	OBJ (bytes)	PLY (bytes)
113,898	786,486	22,210,732	9,882,359

Note: Holoimage used here is 24-bit 512×512. OBJ and PLY file formats are widely used in graphics to represent 3D geometry.

geometries are rendered, only the front geometry is visible and the overlapping areas are resolved automatically. We projected fringe patterns onto the 3D scene and generated the Holoimage, from which the merged 3D data is retrieved using a phase-shifting algorithm and the phase-to-height conversion method. Our simulation demonstrated that the merging can successfully merge two pieces of geometries. Due to the digitization error of the projector and the camera, we found that for a complex surface with an image resolution of 512×512, the depth merging accuracy is as small as 1.4×10^{-3}. We also verified that the merging technique could successfully merge real 3D data (both geometry and texture) captured by the real 3D scanner. Because a single 24-bit color image can represent the whole geometry, the data size is drastically smaller than other 3D geometric data representation methods. Moreover, it is very fast to reconstruct the 3D geometry from Holoimage images; we have demonstrated that real-time reconstruction is feasible for a real 3D shape measurement system.[14] Therefore, Holoimage can be potentially used for applications such as 3D data compression, 3D data communication, and so on.

Acknowledgments

The author would like to thank the team at Geometric Informatics Inc. (GI) for their assistance and discussions. This work was done at GI.

References

1. P. Besl and N. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(2), 239–256 (1992).
2. Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image Vis. Comput.* **10**, 145–155 (1992).
3. B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *SIGGRAPH*, ACM, pp. 303–312 (1996).
4. Q. Hu, K. Harding, D. Hamilton, and J. Flint, "Multiple views merging from different cameras in fringe-projection based phase-shifting method," *Proc. SPIE* **6762**, 676207 (2007).
5. 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).
6. D. Malacara, Ed., *Optical Shop Testing*, John Wiley and Sons, New York (1992).
7. D. C. Ghiglia and M. D. Pritt, *Two-Dimensional Phase Unwrapping: Theory, Algorithms, and Software*, John Wiley and Sons, Inc., New York (1998).
8. C. Zhang, P. S. Huang, and F.-P. Chiang, "Microscopic phase-shifting profilometry based on digital micromirror device technology," *Appl. Opt.* **48**, 5896–5904 (2002).
9. S. Zhang and P. S. Huang, "Novel method for structured light system calibration," *Opt. Eng.* **45**, 083601 (2006).
10. R. Legarda-Sáenz, T. Bothe, and W. P. Jüptner, "Accurate procedure for the calibration of a structured light system," *Opt. Eng.* **43**, 464–471 (2004).
11. S. Zhang and S.-T. Yau, "High-resolution, real-time 3d absolute coordinate measurement based on a phase-shifting method," *Opt. Express* **45**, 2644–2649 (2006).
12. P. S. Huang and S. Zhang, "Fast three-step phase shifting algorithm," *Appl. Opt.* **45**, 5086–5091 (2006).
13. S. Zhang, X. Li, and S.-T. Yau, "Multilevel quality-guided phase unwrapping algorithm for real-time three-dimensional shape reconstruction," *Appl. Opt.* **46**, 50–57 (2007).
14. S. Zhang and P. S. Huang, "High-resolution, real-time three-dimensional shape measurement," *Opt. Eng.* **45**, 123601 (2006).

Song Zhang is a research fellow at Harvard University. He received his doctoral degree in mechanical engineering from Stony Brook University in 2005. His major research interests include real-time 3D optical metrology, 3D machine and computer vision, and geometry processing.

Shing-Tung Yau is a professor of mathematics at Harvard University. He received his doctoral degree in mathematics from University of California—Berkeley in 1971. He received a number of awards including Fields Medal in 1982, a MacArthur Fellowship in 1984, the Crafoord Prize in 1994, and the (U.S.) National Medal of Science in 1997.