

# Dual-frequency pattern scheme for high-speed 3-D shape measurement

Kai Liu<sup>1,3</sup>, Yongchang Wang<sup>1,\*</sup>, Daniel L. Lau<sup>1,4</sup>,  
Qi Hao<sup>2,5</sup>, and Laurence G. Hassebrook<sup>1,6</sup>

<sup>1</sup>Department of Electrical Engineering, University of Kentucky, Lexington, KY, 40507, USA

<sup>2</sup>Department of Electrical Engineering, University of Alabama, Tuscaloosa, AL, 35487, USA

<sup>3</sup>kai.liu@uky.edu

<sup>4</sup>dllau@engr.uky.edu

<sup>5</sup>qh@eng.ua.edu

<sup>6</sup>lgh@engr.uky.edu

<http://vis.uky.edu/~realtime3d>

[\\*ywang6@engr.uky.edu](mailto:*ywang6@engr.uky.edu)

**Abstract:** A novel dual-frequency pattern is developed which combines a high-frequency sinusoid component with a unit-frequency sinusoid component, where the high-frequency component is used to generate robust phase information, and the unit-frequency component is used to reduce phase unwrapping ambiguities. With our proposed pattern scheme, phase unwrapping can overcome the major shortcomings of conventional spatial phase unwrapping: phase jumping and discontinuities. Compared with conventional temporal phase unwrapped approaches, the proposed pattern scheme can achieve higher quality phase data using a less number of patterns. To process data in real time, we also propose and develop look-up table based fast and accurate algorithms for phase generation and 3-D reconstruction. Those fast algorithms can be applied to our pattern scheme as well as traditional phase measuring profilometry. For a  $640 \times 480$  video stream, we can generate phase data at 1063.8 frames per second and full 3-D coordinate point clouds at 228.3 frames per second. These achievements are 25 and 10 times faster than previously reported studies.

© 2010 Optical Society of America

**OCIS codes:** (150.0155)Machine vision optics; (150.1135)Algorithms; (150.5670)Range finding; (150.6910)Three-dimensional sensing; (110.6880)Three-dimensional image acquisition.

---

## References and links

1. S. Zhang, D. Royer, and S. T. Yau, "High-resolution, real-time-geometry video acquisition system," ACM SIGGRAPH (2006).
2. C. Guan, L. G. Hassebrook, and D. L. Lau, "Composite structured light pattern for three-dimensional video," *Opt. Express* **11**, 406–417 (2003).
3. Y. Wang, Q. Hao, A. Fatehpuria, D. L. Lau, and L. G. Hassebrook, "Data acquisition and quality analysis of 3-dimensional fingerprints," in IEEE conference on Biometrics, Identity and Security, Tampa, Florida, (2009).
4. J. Salvi, J. Pages, and J. Batlle, "Pattern codification strategies in structured light systems," *Pattern Recognition* **37**, 827–849 (2004).
5. 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).
6. S. Zhang, D. Royer, and S.-T. Yau, "GPU-assisted high-resolution, real-time 3-d shape measurement," *Opt. Express* **14**, 9120–9129 (2006).

7. J. Li, L. G. Hassebrook, and C. Guan, "Optimized two-frequency phase-measuring-profilometry light-sensor temporal-noise sensitivity," *J. Opt. Soc. Am. A* **20**, 106–115 (2003).
8. S. Zhang and P. S. Huang, "High-resolution, real-time three-dimensional shape measurement," *Opt. Eng.* **45**, 123601 (2006).
9. J. M. Huntley and H. O. Saldner, "Error-reduction methods for shape measurement by temporal phase unwrapping," *J. Opt. Soc. Am. A* **14**, 3188–3196 (1997).
10. S. Li, W. Chen, and X. Su, "Reliability-guided phase unwrapping in wavelet-transform profilometry," *Appl. Opt.* **47**, 3369–3377 (2008).
11. Y. Shi, "Robust phase unwrapping by spinning iteration," *Opt. Express* **15**, 8059–8064 (2007).
12. M. Costantini, "A novel phase unwrapping method based on network programming," *IEEE Trans. Geoscience and Remote Sensing* **36**, 813–821 (1998).
13. D. S. Mehta, S. K. Dubey, M. M. Hossain, and C. Shakher, "Simple multifrequency and phase-shifting fringe-projection system based on two-wavelength lateral shearing interferometry for three-dimensional profilometry," *Appl. Opt.* **44**, 7515–7521 (2005).
14. G. Pedrini, I. Alexeenko, W. Osten, and H. J. Tiziani, "Temporal phase unwrapping of digital hologram sequences," *Appl. Opt.* **42**, 5846–5854 (2003).
15. E.-H. Kim, J. Hahn, H. Kim, and B. Lee, "Profilometry without phase unwrapping using multi-frequency and four-step phase-shift sinusoidal fringe projection," *Opt. Express* **17**, 7818–7830 (2009).
16. J.-L. Li, H.-J. Su, and X.-Y. Su, "Two-frequency grating used in phase-measuring profilometry," *Appl. Opt.* **36**, 277–280 (1997).
17. W.-H. Su and H. Liu, "Calibration-based two-frequency projected fringe profilometry: a robust, accurate, and single-shot measurement for objects with large depth discontinuities," *Opt. Express* **14**, 9178–9187 (2006).
18. S. Y. Chen, Y. F. Li, and J. Zhang, "Vision processing for realtime 3-d data acquisition based on coded structured light," *IEEE Trans. Image Processing* **17**, 167–176 (2008).
19. M. Takeda and K. Mutoh, "Fourier transform profilometry for the automatic measurement of 3-d object shapes," *Appl. Opt.* **22**, 3977–3982 (1983).
20. T. P. Koninckx and L. V. Gool, "Real-time range acquisition by adaptive structured light," *IEEE Trans. Pattern Anal. Mach. Intell.* **28**, 432–445 (2006).
21. S. Zhang and S.-T. Yau, "High-speed three-dimensional shape measurement system using a modified two-plus-one phase-shifting algorithm," *Opt. Eng.* **46**, 113603 (2007).
22. P. S. Huang and S. Zhang, "Fast three-step phase-shifting algorithm," *Appl. Opt.* **45**, 5086–5091 (2006).
23. M. Halioua and H. Liu, "Optical three-dimensional sensing by phase measuring profilometry," *Opt. Lasers in Eng.* **11**, 185–215 (1989).
24. S. F. Frisken, R. N. Perry, A. P. Rockwood, and T. R. Jones, "Adaptively sampled distance fields: A general representation of shape for computer graphics," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 249–254 (2000).
25. X. Su, G. von Bally, and D. Vukicevic, "Phase-stepping grating profilometry: utilization of intensity modulation analysis in complex objects evaluation," *Opt. Commun.* **98**, 141–150 (1993).
26. H. Guo and G. Liu, "Approximations for the arctangent function in efficient fringe pattern analysis," *Opt. Express* **15**, 3053–3066 (2007).
27. P. L. Wizinowich, "Phase shifting interferometry in the presence of vibration: A new algorithm and system," *Appl. Opt.* **29**, 3271–3279 (1990).

## 1. Introduction

Structured Light Illumination (SLI) [1, 2, 3] is an active three dimensional (3-D) shape measurement technique widely used for scientific and industrial purposes [4]. Various algorithms have been proposed to achieve high-accuracy and/or high-speed measurement [5, 6].

Higher frequency patterns have been studied and employed to improve measurement accuracy at the extra computational cost of phase unwrapping. Most unwrapping phase strategies can be fallen in two categories [9]: spatial [10, 5, 11, 12] and temporal [7, 13, 14].

Spatial phase unwrapping approaches assume a continuity of scanned surfaces and likely fail in regions with depth discontinuities (step edges) [9]. Temporal methods, alternatively, use the intermediate phase values typically generated by projecting several additional reference patterns and, therefore, are not efficient for real-time operation. Kim *et al.* [15] proposed an algorithm for high-frequency SLI that avoided direct phase unwrapping with  $4N \geq 8$  patterns. Li *et al.* [16] proposed a two-frequency grating that integrated both high-frequency and unit-frequency into each pattern with  $2N \geq 6$  patterns, where the high frequency had to be equal to

N. Su and Liu [17] treated the algorithm of Li *et al.* as one of two bases in their one-shot pattern strategy.

In this paper, we propose a novel dual-frequency pattern scheme that combines, into a single pattern, a high-frequency SLI pattern with a unit-frequency pattern where the high-frequency component generates wrapped phase, resilient to sensor noise, while the unit-frequency component enables phase unwrapping. Differing from Li *et al.*'s two-frequency grating [16], the minimum number of our proposed patterns is 5 for any high frequencies. In Li *et al.*'s method, the minimum number of patterns is 6, and the high frequency term is fixed and related to the number of patterns. Our proposed pattern is defined by a single equation while Li *et al.* employed two equations.

With regard to high-speed or real-time 3-D measurement, researchers have proposed one-shot [2, 18, 19, 20] methods that reconstruct depth from a single, continuously projected pattern, while others have proposed employing a high-speed camera and projector pair. One-shot SLI techniques are attractive for their insensitivity to object motion, where pattern strategies are based on spatial neighborhood coding, but suffering from poor resolution and high computational cost [2]. Therefore, researchers have attempted to achieve the high resolution with the resilience to motion by driving the camera/projector pair at such high frame rates as to minimize impact of object motion. A study of particular importance has been led by Zhang *et al.* [8, 21] who employed a new algorithm for phase generation instead of the time-consuming arctangent function [22] and also took the advantage of the computation capability of GPU to generate the absolute 3-D coordinates of scanned object [6]. For the image in size of  $532 \times 500$ , their new algorithm was 3.4 times faster than using the arctangent function while the frame rate of 3-D reconstruction was sped up from 6 fps to 25.56 fps, by means of the GPU.

To achieve real-time 3-D video acquisition and reconstruction, this paper also introduces a lossless lookup table (LUT) method for phase, intensity/texture, and depth data that works in combination with a high-speed projector/camera pair widely used in phase measurement profilometry (PMP) [23]. LUT-based methods have been known for the low computational cost and employed in many applications [24]. We develop the novel method that extracts phase and intensity/texture information from the incoming PMP video by reducing the computational complexity of the arctangent and square-root functions. Furthermore, since no LUT-based method has been introduced for inverting the camera's calibration matrix to produce 3-D point clouds in real time, we introduce such a method that can also be applied to other triangulation-based 3-D reconstruction techniques.

As will be demonstrated in this paper, our experimental system achieves a processing rate of 714.29 fps to generate phase and texture video and 228.83 fps to produce 3-D point clouds using just one core of an Intel Core 2 Duo Quad Q9650 processor running at 3.0 GHz. No additional cores or GPUs [6] are needed, but using such resources would, of course, increase our frame rate if necessary. For the proposed novel dual-frequency pattern scheme, where the number of patterns is set to 6, as shown in Sec. 4, our fast LUT-based algorithms can be also applied.

## 2. Phase measuring profilometry

PMP is a popular method of SLI due to its high reliability and high accuracy [25]. In practice, either vertical or horizontal sinusoid patterns are projected onto a target object so that the vertical or horizontal correspondence information, between the camera and the projector, can be directly derived from the computed phase data. The PMP patterns are described as:

$$I_n^p(x^p, y^p) = A^p(x^p, y^p) + B^p(x^p, y^p) \cos(2\pi f y^p - \frac{2\pi n}{N}). \quad (1)$$

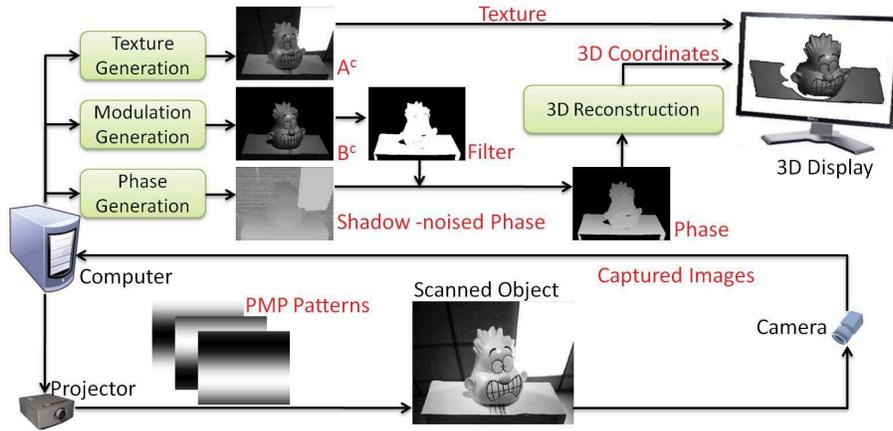


Fig. 1. PMP system diagram.

where  $(x^p, y^p)$  is the coordinates of a pixel in the projector,  $I_n^p$  is the intensity of that pixel,  $A^p$  and  $B^p$  are some constants,  $f$  is the frequency of the sine wave,  $n$  represents the phase-shift index, and  $N$  is the total number of phase shift. Figure (1) shows a group of sine wave patterns with  $N = 3$ ,  $f = 1$ ,  $A^p = 127.5$  and  $B^p = 127.5$  for 8-bits color depth projector are projected.

For reconstruction, a camera captures each image where the sine wave pattern is distorted by the scanned surface topology, resulting in the patterned images expressed as:

$$I_n^c(x^c, y^c) = A^c(x^c, y^c) + B^c(x^c, y^c) \cos\left(\phi(x^c, y^c) - \frac{2\pi n}{N}\right). \quad (2)$$

where  $(x^c, y^c)$  is the coordinates of a pixel in the camera while  $I_n^c(x^c, y^c)$  is the intensity of that pixel. To simplify the notation, the coordinates index both in the camera and projector will be removed from our equation henceforth. The term  $A^c$  is the averaged pixel intensity across the pattern set, which can be derived according to:

$$A^c = \frac{1}{N} \sum_{n=0}^{N-1} I_n^c \quad (3)$$

such that the image  $A^c$  is equal to an intensity or texture photograph of the scene. Correspondingly, the term  $B^c$  is the intensity modulation of a given pixel and is derived from  $I_n^c$  as:

$$B^c = \frac{2}{N} \left\{ \left[ \sum_{n=0}^{N-1} I_n^c \sin\left(\frac{2\pi n}{N}\right) \right]^2 + \left[ \sum_{n=0}^{N-1} I_n^c \cos\left(\frac{2\pi n}{N}\right) \right]^2 \right\}^{0.5}, \quad (4)$$

such that  $B^c$  can be thought of as the amplitude of the sinusoid reflecting off of a point on the target surface. If  $I_n^c$  is constant or less affected by the projected sinusoid patterns,  $B^c$  will be close to zero. Thus  $B^c$  is employed as a shadow noise detector/filter [25] such that the shadow-noised regions, with small  $B^c$  values, are discarded from further processing. Figure 1 shows, an example scene with a background that includes a fluorescent ceiling light, which over saturates the CMOS cameras pixel and, thereby, erases any signal from the SLI projector. In Fig. 1,  $A^c$  looks like a standard video frame absent any indication of the projected pattern sequence  $I_n^p$ . In contrast,  $B^c$ , shown in Fig. 1, looks very similar to  $A^c$  except that it only shows texture in those areas of the scene that significantly reflect the projected sequence  $I_n^p$ . Given the significance of  $B^c$  as an indicator of the projected signal strength, the binarized image in Fig. 1 shows

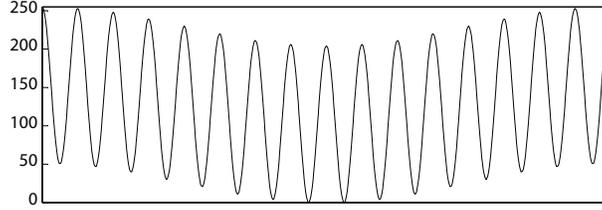


Fig. 2. Cross-section of the proposed pattern strategy for  $N = 5$ ,  $n = 0$ ,  $f_h = 16$ ,  $A^p = 127.5$ ,  $B_1^p = 102$ , and  $B_2^p = 25.5$ .

only those pixels greater in magnitude to a user defined threshold. It is these pixels that will ultimately be used to reconstruct our 3-D surface with ignored pixels being considered too noisy as to relay an reliable depth information.

Of the reliable pixels with sufficiently large  $B^c$ ,  $\phi$  represents the phase value of the captured sinusoid pattern derived as:

$$\phi = \tan^{-1} \left[ \frac{\sum_{n=0}^{N-1} I_n^c \sin(\frac{2\pi n}{N})}{\sum_{n=0}^{N-1} I_n^c \cos(\frac{2\pi n}{N})} \right]. \quad (5)$$

In Fig. 1, we show the resulting phase image corresponding to the scene from sine wave patterns. For reference, Fig. 1 also shows the phase values for all pixels, including those considered unreliable according to  $B^c$ .

Having derived the phase image  $\phi$ , we likewise have derived a unique correspondence value  $y^p$  for every camera pixel  $(x^c, y^c)$  through the linear equation  $\phi(x^c, y^c) = 2\pi y^p$ . The 3-D world coordinates of the scanned object can, therefore, be derived through triangulation with the projector [7] where, under the assumption that the camera and projector are accurately modeled using the pinhole lens model such that their perspective matrices,  $M^c$  and  $M^p$ , are given by:

$$M^c = \begin{pmatrix} m_{11}^c & m_{12}^c & m_{13}^c & m_{14}^c \\ m_{21}^c & m_{22}^c & m_{23}^c & m_{24}^c \\ m_{31}^c & m_{32}^c & m_{33}^c & m_{34}^c \end{pmatrix} \quad \text{and} \quad M^p = \begin{pmatrix} m_{11}^p & m_{12}^p & m_{13}^p & m_{14}^p \\ m_{21}^p & m_{22}^p & m_{23}^p & m_{24}^p \\ m_{31}^p & m_{32}^p & m_{33}^p & m_{34}^p \end{pmatrix}. \quad (6)$$

The 3-D world coordinates  $X^w$ ,  $Y^w$ , and  $Z^w$  are defined according to [7] as

$$\begin{pmatrix} X^w \\ Y^w \\ Z^w \end{pmatrix} = \begin{pmatrix} m_{11}^c - m_{31}^c x^c, & m_{12}^c - m_{32}^c x^c, & m_{13}^c - m_{33}^c x^c \\ m_{21}^c - m_{31}^c y^c, & m_{22}^c - m_{32}^c y^c, & m_{23}^c - m_{33}^c y^c \\ m_{21}^p - m_{31}^p y^p, & m_{22}^p - m_{32}^p y^p, & m_{23}^p - m_{33}^p y^p \end{pmatrix}^{-1} \begin{pmatrix} m_{14}^c - m_{34}^c x^c \\ m_{24}^c - m_{34}^c y^c \\ m_{24}^p - m_{34}^p y^p \end{pmatrix}. \quad (7)$$

It is this matrix inversion as well as the actangent computation in Eq. (5) that will later prove to be the bottleneck preventing real-time surface reconstruction as will be discussed in Sec. 4.3.

### 3. Dual-frequency pattern strategy

For the purpose of minimizing the effects of sensor noise by means of high-frequency PMP while minimizing the computational expense of performing phase unwrapping, we propose a dual-frequency pattern set defined as:

$$I_n^p = A^p + B_1^p \cos(2\pi f_h y^p - \frac{2\pi n}{N}) + B_2^p \cos(2\pi f_u y^p - \frac{4\pi n}{N}), \quad (8)$$

where  $I_n^p$  is the intensity of a pixel in the projector,  $A^p$ ,  $B_1^p$  and  $B_2^p$  are some constants to make value of  $I_n^p$  between 0 and 255 for a 8-bit color depth projector,  $f_h$  is the high frequency of the

sine wave,  $f_u$  is the unit frequency of the sine wave and equals to 1,  $n$  represents the phase-shift index, and  $N$  is the total number of phase shift and is larger or equal to 5. Figure 2 illustrates one proposed pattern along with its profile for  $N = 5$ ,  $n = 0$ ,  $f_h = 16$ ,  $A^p = 127.5$ ,  $B_1^p = 102$ , and  $B_2^p = 25.5$  for 8-bits per pixel grayscale intensity.

Using this novel pattern set, Eq. (2) becomes:

$$I_n^c = A^c + B_1^c \cos\left(\phi_h - \frac{2\pi n}{N}\right) + B_2^c \cos\left(\phi_u - \frac{4\pi n}{N}\right), \quad (9)$$

where  $I_n^c$  is the intensity of a pixel in the camera. The term  $A^c$  is still the averaged pixel intensity across the pattern set, which can be derived by Eq. 3 such that the image  $A^c$  is equal to an intensity or texture photograph of the scene. Correspondingly, the term  $B_1^c$  is the intensity modulation of a given pixel corresponding to  $\phi_h$  and is derived from  $I_n^c$  as:

$$B_m^c = \frac{2}{N} \left\{ \left[ \sum_{n=0}^{N-1} I_n^c \sin\left(m \frac{2\pi n}{N}\right) \right]^2 + \left[ \sum_{n=0}^{N-1} I_n^c \cos\left(m \frac{2\pi n}{N}\right) \right]^2 \right\}^{0.5}, \quad (10)$$

with  $m = 1$  such that  $B_1^c$  can be thought of as the amplitude of the sinusoid reflecting off of a point on the target surface.

Similar to traditional PMP, if  $I_n^c$  is constant or less affected by the projected sinusoid patterns, then  $B_1^c$  will be close to zero, indicating that  $B_1^c$  is very sensitive to noise in either the camera, projector, and/or ambient light. Thus  $B_1^c$  is employed as an indicator of the signal to noise ratio such that excessively noisy regions, with small  $B_1^c$  values, are discarded from further processing [25].  $B_2^c$  is derived from  $I_n^c$  by Eq. 10 with  $m = 2$  and it is the intensity modulation corresponding to  $\phi_u$ . Of the reliable pixels with sufficiently large  $B_1^c$ , the phase-pair,  $(\phi_h, \phi_u)$ , is derived as:

$$(\phi_h, \phi_u) = \left( \tan^{-1} \left[ \frac{\sum_{n=0}^{N-1} I_n^c \sin\left(\frac{2\pi n}{N}\right)}{\sum_{n=0}^{N-1} I_n^c \cos\left(\frac{2\pi n}{N}\right)} \right], \tan^{-1} \left[ \frac{\sum_{n=0}^{N-1} I_n^c \sin\left(\frac{4\pi n}{N}\right)}{\sum_{n=0}^{N-1} I_n^c \cos\left(\frac{4\pi n}{N}\right)} \right], \right) \quad (11)$$

where  $\phi_h$  represents the wrapped phase value of the captured pattern and  $\phi_u$  represents the base phase used to unwrap  $\phi_h$ .

#### 4. LUT-based processing

LUT-based programming is a well known and widely used technique [24] for minimizing processing latency. In this section, we implement LUT-based processing for deriving the modulation,  $B^c$ , and phase,  $\phi$ , terms for traditional PMP that is suitable for all triangulation-based 3-D measurement, including our proposed dual-frequency pattern set. The proposed LUT-based processing takes advantage of the need by real-time systems to use as few patterns as possible by using the 8-bits per pixel, of the captured pattern set, as the indices into the LUT. By having LUTs that account for every possible combination of captured pixel value over the pattern set while storing double-precision results, the proposed scheme is completely lossless compared to traditional processing.

##### 4.1. Modulation

As a first step in reducing the computational complexity associated with Eqs. (4), (5), and (7), we simplify the determination of  $B^c$  by rewriting Eq. (4) as:

$$B^c = \frac{1}{3} \left[ 3(I_1^c - I_2^c)^2 + (2I_0^c - I_1^c - I_2^c)^2 \right]^{0.5} \quad (12)$$

for  $N = 3$ ,

$$B^c = \frac{1}{2} [(I_1^c - I_3^c)^2 + (I_0^c - I_2^c)^2]^{0.5} \quad (13)$$

for  $N = 4$ , or

$$B^c = \frac{1}{6} [3(I_1^c + I_2^c - I_4^c - I_5^c)^2 + (2I_0^c - 2I_3^c + I_1^c - I_2^c - I_4^c + I_5^c)^2]^{0.5} \quad (14)$$

for  $N = 6$ . Noting that we need only solve these equations for 8-bits per pixel  $I_n^c$  images, we can implement a modulation look-up table, MLUT, that, for  $N = 3$ , is defined according to

$$MLUT[U, V] = \frac{1}{3} [3V^2 + U^2]^{0.5}, \quad (15)$$

where integer indices  $V$  and  $U$  are derived from

$$V = I_1^c - I_2^c \quad \text{and} \quad U = 2I_0^c - I_1^c - I_2^c; \quad (16)$$

for  $N = 4$ , is defined according to

$$MLUT[U, V] = \frac{1}{2} [V^2 + U^2]^{0.5}, \quad (17)$$

where integer indices  $V$  and  $U$  are derived from

$$V = I_1^c - I_3^c \quad \text{and} \quad U = I_0^c - I_2^c; \quad (18)$$

or for  $N = 6$ , is defined according to

$$MLUT[U, V] = \frac{1}{6} [3V^2 + U^2]^{0.5}, \quad (19)$$

where integer indices  $V$  and  $U$  are derived from

$$V = I_1^c + I_2^c - I_4^c - I_5^c \quad \text{and} \quad U = 2I_0^c - 2I_3^c + I_1^c - I_2^c - I_4^c + I_5^c. \quad (20)$$

The double-precision results of Eq. (15), (17) or (19) are stored in the MLUT. In contrast to Eq. (4), MLUT reduces the run-time computation cost of modulation without losing accuracy, where the size of the LUT is determined by the number of bits per pixel of the camera sensor and the number of patterns being projected.

For our proposed dual-frequency pattern scheme, the minimum number of patterns is 5, but to take advantage of our LUT-based processing, we set the number of patterns to 6 where the modulation terms  $B_1^c$  and  $B_2^c$ , in Eq. (9), are described as:

$$B_1^c = \frac{1}{6} [3(I_1^c + I_2^c - I_4^c - I_5^c)^2 + (2I_0^c - 2I_3^c + I_1^c - I_2^c - I_4^c + I_5^c)^2]^{0.5} \quad (21)$$

and

$$B_2^c = \frac{1}{6} [3(I_1^c - I_2^c + I_4^c - I_5^c)^2 + (2I_0^c + 2I_3^c - I_1^c - I_2^c - I_4^c - I_5^c)^2]^{0.5}. \quad (22)$$

In practice, we need only calculate  $B_1^c$  as a shadow filter and for representing object texture. The MLUT for  $B_1^c$  is the same as Eq. (19).

#### 4.2. Phase

As our second step to producing real-time 3-D video with texture, we intend to minimize the computational complexity associated with generating phase video where the arctangent function has long been considered a significant obstacle for fringe pattern analysis [26]. Previous approaches to this problem include Huang *et al.*'s cross ratio algorithm [22] and Guo *et al.*'s approximation algorithm [26]. However, these methods reduce computational cost at the expense of accuracy and are not faster than our LUT-based algorithms.

As we did with Eq. (4) for  $B^c$ , we simplify Eq. (5) according to the number of patterns  $N$  such that

$$\phi = \tan^{-1} \left[ \frac{3^{0.5}(I_1^c - I_2^c)}{2I_0^c - I_1^c - I_2^c} \right] \quad (23)$$

for  $N = 3$ ,

$$\phi = \tan^{-1} \left[ \frac{I_1^c - I_3^c}{I_0^c - I_2^c} \right] \quad (24)$$

for  $N = 4$ , or

$$\phi = \tan^{-1} \left[ \frac{3^{0.5}(I_1^c + I_2^c - I_4^c - I_5^c)}{2I_0^c - 2I_3^c + I_1^c - I_2^c - I_4^c + I_5^c} \right] \quad (25)$$

for  $N = 6$ . Again based on the fact that the intensity values of grabbed images are range-limited integers, we can implement these calculations through a phase LUT (PLUT), for  $N = 3$ , defined according to

$$PLUT[U, V] = \tan^{-1} \left[ \frac{3^{0.5}V}{U} \right], \quad (26)$$

where  $U$  and  $V$  are defined in Eq. (16); for  $N = 4$ , defined according to

$$PLUT[U, V] = \tan^{-1} \left[ \frac{V}{U} \right], \quad (27)$$

where  $U$  and  $V$  are defined in Eq. (18); or for  $N = 6$ , defined according to

$$PLUT[U, V] = \tan^{-1} \left[ \frac{3^{0.5}V}{U} \right], \quad (28)$$

where  $U$  and  $V$  are defined in Eq. (20). The double-precision results are stored in the PLUT. Thus, the time-consuming arctangent operation is pre-performed, and phase values are obtained by accessing the pre-computed PLUT whose size is, again, determined by the number of bits per pixel of the sensor as well as the number of patterns projected with no loss in accuracy. Compared to Eqs. (5), PLUT avoids computing arctangent function at run-time such that the computational cost of phase is greatly reduced without introducing distortion.

For our proposed dual-frequency pattern scheme, the more accurate wrapped phase  $\phi_h$  and the coarse unit frequency phase  $\phi_u$  pair in the Eq. (8) is rewritten as

$$(\phi_h, \phi_u) = \left( \tan^{-1} \left[ \frac{3^{0.5}(I_1^c + I_2^c - I_4^c - I_5^c)}{2I_0^c - 2I_3^c + I_1^c - I_2^c - I_4^c + I_5^c} \right], \tan^{-1} \left[ \frac{3^{0.5}(I_1^c - I_2^c + I_4^c - I_5^c)}{2I_0^c + 2I_3^c - I_1^c - I_2^c - I_4^c - I_5^c} \right] \right) \quad (29)$$

The PLUT for  $\phi_h$  is the same as the Eq. (28), and the PLUT for  $\phi_u$  is defined as

$$PLUT[U, V] = \tan^{-1} \left[ \frac{3^{0.5}V}{U} \right], \quad (30)$$

where  $V$  and  $U$  are derived from

$$V = I_1^c - I_2^c + I_4^c - I_5^c \quad \text{and} \quad U = 2I_0^c + 2I_3^c - I_1^c - I_2^c - I_4^c - I_5^c. \quad (31)$$

Once  $\phi_h$  and  $\phi_u$  are obtained, phase unwrapping can be also achieved in real-time.

#### 4.3. 3-D point cloud

Having obtained both the phase and modulation images by means of LUTs, we begin our derivation of a LUT for the purpose of implementing Eq. (7). After matrix inverse and matrix multiplication operations, Eq (7) can be expanded into direct algebraic forms as

$$\begin{aligned} X^w &= M_x(x^c, y^c) + N_x(x^c, y^c)T \\ Y^w &= M_y(x^c, y^c) + N_y(x^c, y^c)T \\ Z^w &= M_z(x^c, y^c) + N_z(x^c, y^c)T \end{aligned}$$

where  $T = (C(x^c, y^c)y^p + 1)^{-1}$  and  $M_x, M_y, M_z, N_x, N_y$  and  $N_z$  are defined in the Appendix.

We then note that, based on the mapping of world coordinates to the camera plane [7] by  $M^c$ , if  $Z^w$  is calculated according to

$$Z^w = M_z(x^c, y^c) + N_z(x^c, y^c)T, \quad (32)$$

then  $X^w$  and  $Y^w$  can be computed as

$$X^w = E_x(x^c, y^c)Z^w + F_x(x^c, y^c) \quad \text{and} \quad Y^w = E_y(x^c, y^c)Z^w + F_y(x^c, y^c), \quad (33)$$

where

$$\begin{aligned} E_x(x^c, y^c) &= \frac{(m_{22}^c m_{33}^c - m_{23}^c m_{32}^c)x^c + (m_{13}^c m_{32}^c - m_{12}^c m_{33}^c)y^c + (m_{12}^c m_{23}^c - m_{13}^c m_{22}^c)}{(m_{21}^c m_{32}^c - m_{22}^c m_{31}^c)x^c + (m_{12}^c m_{31}^c - m_{11}^c m_{32}^c)y^c + (m_{11}^c m_{22}^c - m_{12}^c m_{21}^c)}, \\ F_x(x^c, y^c) &= \frac{(m_{22}^c m_{34}^c - m_{24}^c m_{32}^c)x^c + (m_{14}^c m_{32}^c - m_{12}^c m_{34}^c)y^c + (m_{12}^c m_{24}^c - m_{14}^c m_{22}^c)}{(m_{21}^c m_{32}^c - m_{22}^c m_{31}^c)x^c + (m_{12}^c m_{31}^c - m_{11}^c m_{32}^c)y^c + (m_{11}^c m_{22}^c - m_{12}^c m_{21}^c)}, \\ E_y(x^c, y^c) &= \frac{(m_{23}^c m_{31}^c - m_{21}^c m_{33}^c)x^c + (m_{11}^c m_{33}^c - m_{13}^c m_{31}^c)y^c + (m_{13}^c m_{21}^c - m_{11}^c m_{23}^c)}{(m_{21}^c m_{32}^c - m_{22}^c m_{31}^c)x^c + (m_{12}^c m_{31}^c - m_{11}^c m_{32}^c)y^c + (m_{11}^c m_{22}^c - m_{12}^c m_{21}^c)}, \quad \text{and} \\ F_y(x^c, y^c) &= \frac{(m_{21}^c m_{32}^c - m_{22}^c m_{31}^c)x^c + (m_{12}^c m_{31}^c - m_{11}^c m_{32}^c)y^c + (m_{11}^c m_{22}^c - m_{12}^c m_{21}^c)}{(m_{21}^c m_{32}^c - m_{22}^c m_{31}^c)x^c + (m_{12}^c m_{31}^c - m_{11}^c m_{32}^c)y^c + (m_{11}^c m_{22}^c - m_{12}^c m_{21}^c)}. \end{aligned}$$

Implementing the 7 parameters  $M_z, N_z, C, E_x, E_y, F_x$ , and  $F_y$  by means of table look-up for indices  $(x^c, y^c)$  (camera column and row indices), reduces the total computational complexity associated with deriving the 3-D point cloud from the phase term to 7 look-ups, 4 additions, 3 multiplications, and 1 division, which is significantly less than what is involved in performing matrix inversion and matrix multiplication, as required by Eq. (7). It should be noted that the method presented in Eqs. (32) and (33) can be applied to all triangulation-based, 3-D coordinate, reconstruction techniques including stereo vision, 3-D from video, time-of-flight, and so on.

## 5. Experiments

In order to test the performance of the proposed pattern scheme and LUT-based processing, we programmed the experimental system of Fig. 3 using Microsoft Visual Studio 2005 with managed C++. The imaging sensor is an 8-bits per pixel, monochrome, Prosilica GC640M, gigabit ethernet camera with a frame rate of 120 fps and  $640 \times 480$  pixel resolution. The projector is

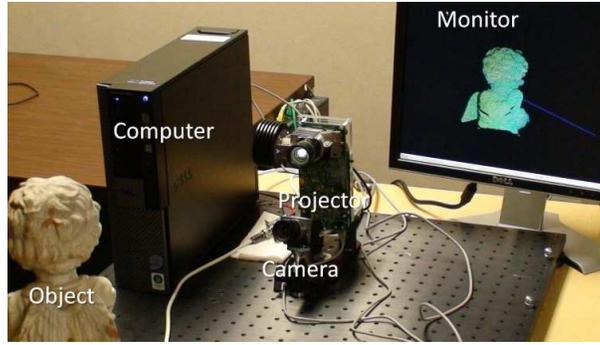


Fig. 3. Experimental setup.

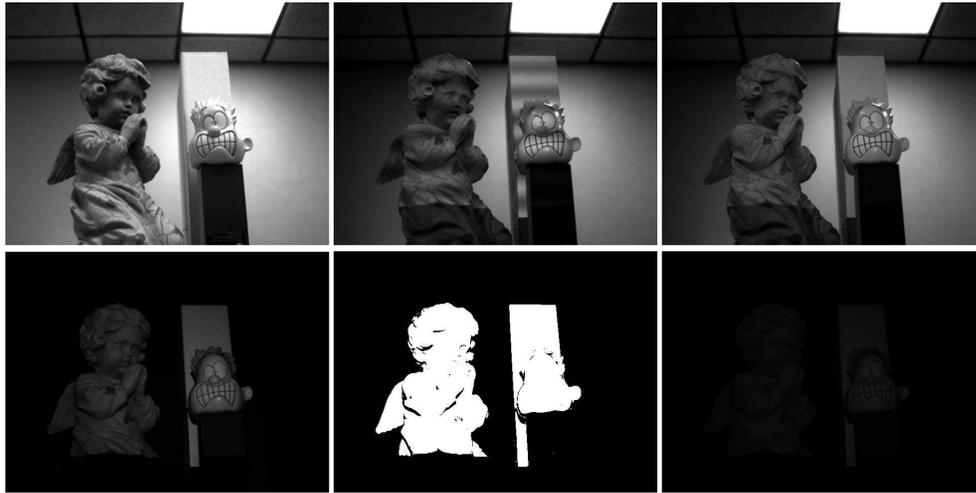


Fig. 4. Visualization of the scanned object (top left), one of captured images with projected pattern,  $I_0^c$  (top center),  $A^c$  (top right),  $B_1^c$  (bottom left), binarized  $B_1^c$  with threshold of 10 (bottom center), and  $B_2^c$  (bottom right).

composed of a Texas Instrument's Discovery 1100 board with ALP-1 controller and LED-OM with 225 ANSI lumens. The projector has a maximum frame rate of 150 fps at a resolution of  $1024 \times 768$  and 8-bits per pixel grayscale. The camera and projector are synchronized by an external triggering circuit. As our processing unit, we used a Dell Optiplex 960 with an Intel Core 2 Duo Quad Q9650 processor running at 3.0 GHz.

Our experiments include: (1) scanning a static scene with the proposed dual-frequency pattern scheme employing 5 patterns without using LUT-based processing; (2) scanning static scene with the traditional PMP with unit frequency for  $N = 3, 4$  and 6 involving LUT-based processing; (3) scanning a moving hand with dual-frequency patterns for  $N = 6$  by means of LUT-based processing; and (4) scanning a moving white table tennis ball and reconstructing it with a known sphere model.

In the first experiment, the image of Fig. 4 (top-left) shows the scanned static objects. The projected patterns' parameters, in the Eq. (8), are  $N = 5$ ,  $f_h = 6$ ,  $f_u = 1$ ,  $A^p = 155$ ,  $B_1^p = 80$  and  $B_2^p = 20$ . The minimum value in each pattern is 55 in order to prevent underflow of captured images using our camera. The visualizations of the extracted parameters  $I_0^c$ ,  $A^c$ ,  $B_1^c$  and  $B_2^c$

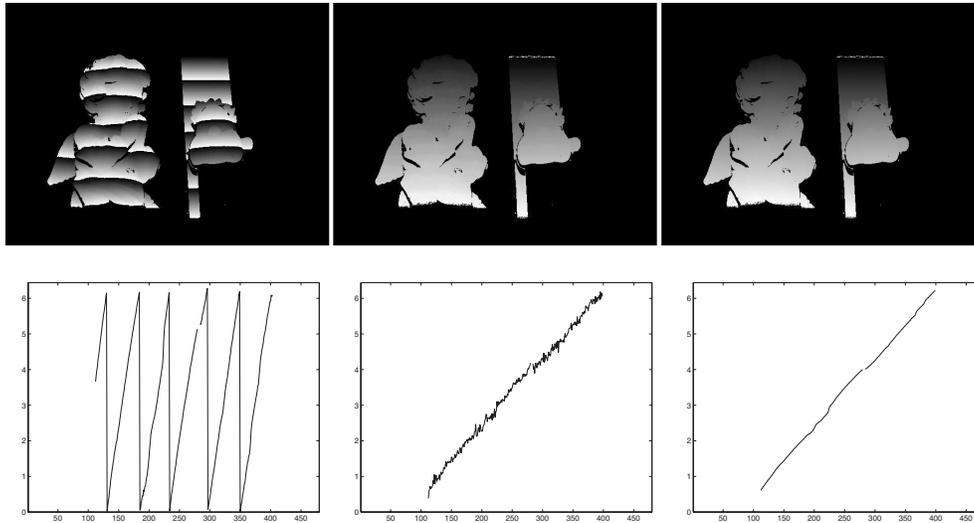


Fig. 5. The (top) visualization and (bottom) cross-section at the 250th column of the (left) wrapped phase  $\phi_h$  filtered by binarized  $B_1^c$ , the (center) base phase  $\phi_u$  filtered by  $B_1^c$ , and the (right) final unwrapped phase filtered by  $B_1^c$  (right).

are shown in Fig. 4 (top center), (top right), (bottom left), and (bottom right). The image in Fig. 4 (bottom center) shows the binarized  $B_1^c$ , to be used as a shadow noise filter for phase generation, using a threshold of 10.

The visualizations of Eq. (11) are illustrated in Fig. 5 (top-left) and (top-center) where shadow noise was filtered by the binarized  $B_1^c$  while Fig. 5 (bottom-left) and (bottom-center) show the 250th column curve crossing the angel. In comparison, the unit phase term is noisy while the wrapped phase has high quality but needs to be unwrapped. Because we obtain the wrapped phase and the base phase simultaneously, the unwrapping algorithm for multi-frequency PMP can be performed to generate the final, high-quality and non-ambiguous phase. No neighborhood technique need be employed. The final unwrapped phase image without shadow noise is shown in Fig. 5 (top-right) and the 250th column curve crossing the angel is shown in Fig. 5 (bottom-right). Now once the unwrapped phase is obtained, 3-D world coordinates of the scanned object can be computed by triangulation between the camera and projector. Figure 6 shows the reconstructed 3-D point clouds with texture (top) and depth rendering (bottom) in front (left), side (center), and top view (right).  $A^c$  acts as the texture for the 3-D point clouds in this experiment.

For our second experiment, a static object was scanned using traditional PMP with unit frequency with  $N = 3, 4,$  and  $6$  for the purpose of measuring the speed at which our LUT-based algorithms perform without phase unwrapping. Although the processor has four cores, our reported processing rates for each LUT algorithm are based on using just a one core. And although  $B^c$  could be used as a shadow noise filter, thus reducing the number of pixels that would need to be processed while deriving  $\phi$ , we computed a phase value for the full set of  $640 \times 480$  pixels such that our observed processing times represent an upper limit. Under these conditions, the average computation time over 1,000 runs for modulation  $B^c$  and phase  $\phi$  was then reported in Table 1.  $B^c$  acts as texture in this experiment.

In an overview of Table 1 for the total processing time including texture generation, phase generation, and 3-D reconstruction, our LUT-based computations are 173.31 fps for  $N = 3,$



Fig. 6. the reconstructed 3-D point clouds with texture (top) and depth rendering (bottom) in front view (left), side view (center) and top view (right)

Table 1. Processing time and rate, in milliseconds and frames per second (in parentheses), respectively, for various stages of PMP processing by means of the equations and LUT described in this paper.

Process	$N = 3$	$N = 4$	$N = 6$
Eq. (4)	58.88 (17.58)	67.34 (14.85)	127.72 (8.22)
Eq. (5)	71.40 (14.01)	79.53 (12.57)	135.31 (7.39)
*1 Eq. (7)	45.00 (22.22)	44.85 (22.30)	45.16 (22.14)
*2 Eq. (12), (13), and then (14)	6.72 (148.81)	2.34 (427.35)	10.00 (100.00)
*3 Eq. (23), (24), and then (25)	17.97 (55.65)	18.90 (52.91)	18.75 (53.33)
MLUT: Eq. (15), (17) and then (19)	0.94 (1063.83)	1.25 (800.00)	2.18 (458.72)
PLUT: Eq. (26), (27) and then (28)	0.94 (1063.83)	1.25 (800.00)	2.19 (456.62)
MLUT and PLUT, combined	1.40 (714.29)	1.72 (581.40)	2.81 (355.87)
MLUT + PLUT for 2 plus 1 [21]	1.39 (719.81)	-	-
3-D LUT (Eq. (32) and (33))	4.37 (228.83)	4.37 (228.83)	4.38 (228.31)
Traditional PMP: *2, *3, and then *1	65.97 (15.16)	65.0 (15.38)	68.75 (14.55)
MLUT and PLUT, and then 3-D LUT	5.77 (173.31)	6.09 (164.20)	7.18 (139.28)

164.20 fps for  $N = 4$ , and 139.28 for  $N = 6$  versus, using traditional PMP, 15.16 fps for  $N = 3$ , 15.38 fps for  $N = 4$ , and 14.55 fps for  $N = 6$ , which means our LUT-based algorithms is  $10\times$  faster than traditional PMP algorithms. Using the general modulation equation, Eqs. (4), and the general phase equation, (5), the processing time is increased with larger  $N$ , because larger  $N$  means more  $\sin(\cdot)$ ,  $\cos(\cdot)$  and summation computations. The processing time and rates are listed in Table 1.

With the simplified modulation equations of Eq. (12), (13), and (14), the processing time decreases significantly for  $N = 4$  (2.34 ms) compared with  $N = 3$  (6.72 ms) because, before performing square-root computation, there are only 2 subtractions, 1 addition, and 2 square computations for  $N = 4$ . There are 3 subtractions, 1 addition, 2 multiplications, and 2 square computations for  $N = 3$ . Using the simplified phase equation, Eq. (23), (24), and (25), the processing time is almost the same for different  $N$  because, although the processing time for the basic computations, such as addition and subtraction, is still varied for different  $N$ . Such basic processing time is negligible compared to the processing time for the processing hungry arctangent function.

Furthermore, it is found from Table 1 that the performance of MLUT and PLUT is the same for different  $N$  because the computations for the integer indices,  $U$  and  $V$ , are the same for both



Fig. 7. Sample video clips. (Media 1), (Media 2), (Media 3)

MLUT and PLUT with the same  $N$  as is the time for accessing MLUT and PLUT. However, the processing time for MLUT/PLUT is increased with increasing  $N$  because the time for accessing the image buffer,  $I_n^c$ , increases; therefore, accessing this buffer is the predominant factor in this case. In practice when we want to access MLUT and PLUT, the computations of  $U$  and  $V$  need only be performed once. So Table 1 also shows the processing time of the row marked as “MLUT and PLUT, combined” is less than the summation time of the row marked as “MLUT: Eq. (15), (17) and then (19)” and the row marked as “PLUT: Eq. (26), (27) and then (28)” in which their  $U$  and  $V$  were computed respectively.

For testing 3-D reconstruction using a single processing core, we implemented reconstruction by means of matrix-inversion (Eq. (7)) versus our LUT-based implementation of Sec. 4.3. With our 3-D LUT algorithm, the frame rate of 3-D reconstruction is 228.83 fps for  $640 \times 480$  image resolution, which is 10.3 times faster than matrix inversion by means of Eq (7). For added comparison, Zhang *et al.* reported a reconstruction frame rate of 25.56 fps with a resolution of  $532 \times 500$  when performing matrix inversion by means of GPU processing on an nVidia Quadro FX 3450 [6]. For Zhang *et al.*'s  $2 + 1$  pattern strategy [21], our LUT analysis can be also applied, and the performance is shown in Table 1.

In our third experiment, we scanned a human subject's hand gestures with our dual-frequency pattern set ( $N = 6$ ) using LUT-based processing. Different from our second experiment, the most recent  $N$  video frames are captured live and stored in a shared image buffer by a camera thread. At the same time, a second thread performs modulation and phase generation from the shared buffer and stores the computed data into a modulation and phase buffer. A third thread, simultaneously, performs 3-D reconstruction using data from this modulation buffer, storing the 3-D point cloud results into a 3-D data buffer. Finally, a fourth thread displays the data from the 3-D data buffer using OpenGL. Because the speed of the camera/projector pair is 120 fps, the speed of final 3-D reconstruction is also 120 fps while the display speed is limited by the LCD display at 40 – 50 fps.

Figure 7 shows a human subject's hand gestures. In the case of Fig. 7 (left), we show the entire system along with the subject's hand as a way of demonstrating that video is being acquired and displayed on screen and in real-time where there is no noticeable delay between the gesture's performed by the subject and displayed on screen. On the LCD panel, you can see the XYZ axes labeled as blue, green, and red lines, respectively, while the 3-D point cloud is also being reflected off the XZ plane as a way of better illustrating the hand coming forward and away from the camera. In the video clips of Fig. 7 (center) and (right), we show the subject's hand held in front of a white cardboard box. The purpose of these clips is to show that we can track multiple, disjoint objects, since there are no phase ambiguities when using dual-frequency pattern scheme. For added inspection, the images in Fig. 8 show the live frames of hand poses with front view and top view where, from visual inspection, one can see the texture/intensity encoding of the points derived from  $B^c$ . Points with low modulation strength are not displayed.

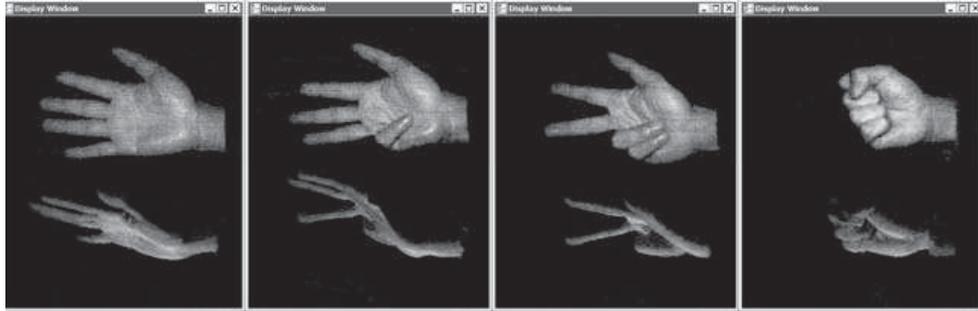


Fig. 8. Sample point clouds, using dual-frequency pattern scheme for  $N = 6$ , live show of various hand poses. (top) Front view and (middle) Top view.

As a final and quantitative evaluation of our proposed methodology, we scanned a texture-less, white table tennis ball, with a radius of 19 mm, swinging like a pendulum on a 20 cm long string. We used our proposed dual-frequency pattern scheme with  $N = 6$ ,  $f_h = 4$ ,  $f_u = 1$ ,  $A^p = 155$  and  $B_1^p = B_2^p = 50$ . Our PLUT and 3-D LUT were employed for phase generation and 3-D reconstruction. From the reconstructed point clouds of the moving ball, we found the best-fit sphere with variable position and radius. From this, we compared the estimated sphere radius with the true radius. Plotted in Fig. 9 (top), is this estimated sphere radius. The averaged distance of the table tennis ball point cloud, in Fig. 9 (bottom), was estimated as the centroid of all points in the cloud.

The exact position of the point cloud centroid is of no quantitative significance since we do not have a true measure to compare, but it does indicate the points in time when the ball was at the extreme points of the pendulum swing. What Fig. 9 shows is that we get an accurate and stable measurement of the ball radius at the extreme points of the swing, where the ball comes to a complete stop before changing direction. The mid-way point has the least stable radius estimate due to this being the point in the swing where the ball is moving at its highest velocity. At its farthest point from the sensor, noise in the reconstruction results in a sphere radius that is its smallest because the points of the cloud are not, in any way, constrained to be side by side in the reconstruction. So two neighboring pixels may end up being on the front and back of the best-fit sphere. This behavior is evident in the distance estimate, which has a high degree of variability. At its closest point to the sensor, the point cloud is sufficiently flat as to have the entire point cloud residing on the front-side of the best-fit sphere, resulting in a smooth and consistent distance estimate.

## 6. Conclusion and future works

In this paper, we presented a novel dual-frequency pattern strategy along with three LUT-based algorithms for calculating the modulation and phase terms along with reconstructing 3-D point clouds based upon a popular method of SLI. The proposed pattern coding technique embeds low and high frequency components into a single pattern where high-quality, high-frequency phase is unwrapped according to low-frequency phase. No any intermediate patterns are needed nor are surface discontinuities an issue. With regards to reconstruction quality, we note that scan accuracy is strongly influenced by features of pattern design, which were also considered in this paper. Being that our proposed method requires at least five patterns, future work will look at reducing this pattern number further.

With regards to LUT-based processing, especially, phase generation and 3-D reconstruction contributions of this paper are of particular importance to real-time 3-D reconstructing. Com-

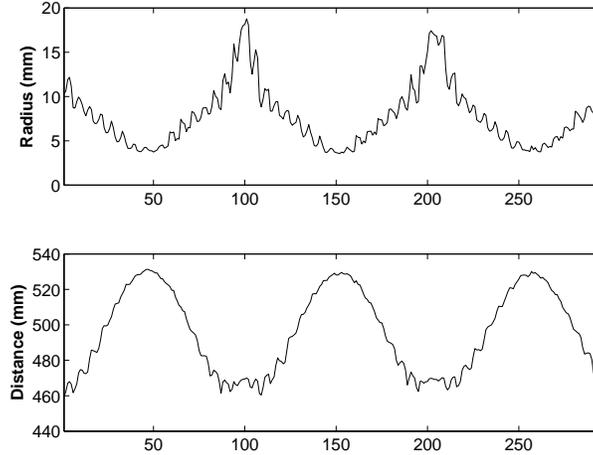


Fig. 9. Radius of best-fit sphere (top) and distance between the camera and the center of the ball (bottom) for 295 measurements.

pared with both traditional PMP algorithm (Table 1) and previous published literatures [6,8,22], the improvement in frame rate performance for phase generation and 3-D reconstruction is significant. Furthermore, the methods presented here are not strictly tied to traditional PMP and the proposed dual-frequency approach where the 2+1 method, introduced by Wizinowich [27] and modified by Zhang in his SLI system [21], can be performed at the equivalent of 719 fps (Table 1) when implemented by means of MLUT and PLUT.

As presented using an 8 bits per pixel sensor, the LUT-based method is lossless but requires significant memory, depending on the number of patterns used. This memory consumption can, of course, be manipulated to use only the most significant bits of the incoming video stream to reduce the overall memory requirements for MLUT and PLUT. The LUTs used for 3-D reconstruction are limited in size by the number of pixels forming the sensor, so reducing the memory requirements here corresponds to reducing the overall field of view of the camera. While losing accuracy in MLUT may be insignificant for most applications, quantizing the incoming video to reduce the memory consumed by PLUT incurs a penalty in phase accuracy. For our experimental system, the variance in the phase error (in radians) ranged from 0.031 for 3-pattern, 4-bits per pixel LUTs to 0.001811 for 3-pattern, 10-bits per pixel LUTs.

#### Appendix: Definitions of $M_x$ , $M_y$ , $M_z$ , $N_x$ , $N_y$ and $N_z$

$$M_x(x^c, y^c) = \frac{\left(\frac{a_{x1}x^c + a_{x2}y^c + a_{x3}}{d_1x^c + d_2y^c + d_3}\right)}{\left(\frac{c_1x^c + c_2y^c + c_3}{d_1x^c + d_2y^c + d_3}\right)}, \quad N_x(x^c, y^c) = \frac{b_{x1}x^c + b_{x2}y^c + b_{x3}}{d_1x^c + d_2y^c + d_3} - M_x(x^c, y^c), \quad (34)$$

$$M_y(x^c, y^c) = \frac{\left(\frac{a_{y1}x^c + a_{y2}y^c + a_{y3}}{d_1x^c + d_2y^c + d_3}\right)}{\left(\frac{c_1x^c + c_2y^c + c_3}{d_1x^c + d_2y^c + d_3}\right)}, \quad N_y(x^c, y^c) = \frac{b_{y1}x^c + b_{y2}y^c + b_{y3}}{d_1x^c + d_2y^c + d_3} - M_y(x^c, y^c), \quad (35)$$

$$M_z(x^c, y^c) = \frac{\left(\frac{a_{z1}x^c + a_{z2}y^c + a_{z3}}{d_1x^c + d_2y^c + d_3}\right)}{\left(\frac{c_1x^c + c_2y^c + c_3}{d_1x^c + d_2y^c + d_3}\right)}, \quad N_z(x^c, y^c) = \frac{b_{z1}x^c + b_{z2}y^c + b_{z3}}{d_1x^c + d_2y^c + d_3} - M_z(x^c, y^c), \quad (36)$$

where

$$\begin{aligned}
a_{x_1} &= m_{22}^c m_{34}^c m_{33}^p + m_{23}^c m_{32}^c m_{34}^p + m_{24}^c m_{33}^c m_{32}^p - m_{22}^c m_{33}^c m_{34}^p - m_{23}^c m_{34}^c m_{32}^p - m_{24}^c m_{32}^c m_{33}^p, \\
a_{x_2} &= m_{12}^c m_{33}^c m_{34}^p + m_{13}^c m_{34}^c m_{32}^p + m_{14}^c m_{32}^c m_{33}^p - m_{12}^c m_{34}^c m_{33}^p - m_{13}^c m_{32}^c m_{34}^p - m_{14}^c m_{33}^c m_{32}^p, \\
a_{x_3} &= m_{12}^c m_{24}^c m_{33}^p + m_{13}^c m_{22}^c m_{34}^p + m_{14}^c m_{23}^c m_{32}^p - m_{12}^c m_{23}^c m_{34}^p - m_{13}^c m_{24}^c m_{32}^p - m_{14}^c m_{22}^c m_{33}^p, \\
a_{y_1} &= m_{21}^c m_{33}^c m_{34}^p + m_{23}^c m_{34}^c m_{31}^p + m_{24}^c m_{31}^c m_{33}^p - m_{21}^c m_{34}^c m_{33}^p - m_{23}^c m_{31}^c m_{34}^p - m_{24}^c m_{33}^c m_{31}^p, \\
a_{y_2} &= m_{11}^c m_{34}^c m_{33}^p + m_{13}^c m_{31}^c m_{34}^p + m_{14}^c m_{33}^c m_{31}^p - m_{11}^c m_{33}^c m_{34}^p - m_{13}^c m_{34}^c m_{31}^p - m_{14}^c m_{31}^c m_{33}^p, \\
a_{y_3} &= m_{11}^c m_{23}^c m_{34}^p + m_{13}^c m_{24}^c m_{31}^p + m_{14}^c m_{21}^c m_{33}^p - m_{11}^c m_{24}^c m_{33}^p - m_{13}^c m_{21}^c m_{34}^p - m_{14}^c m_{23}^c m_{31}^p, \\
a_{z_1} &= m_{21}^c m_{34}^c m_{32}^p + m_{22}^c m_{31}^c m_{34}^p + m_{24}^c m_{32}^c m_{31}^p - m_{21}^c m_{32}^c m_{34}^p - m_{22}^c m_{34}^c m_{31}^p - m_{24}^c m_{31}^c m_{32}^p, \\
a_{z_2} &= m_{11}^c m_{32}^c m_{34}^p + m_{12}^c m_{34}^c m_{31}^p + m_{14}^c m_{31}^c m_{32}^p - m_{11}^c m_{34}^c m_{32}^p - m_{12}^c m_{31}^c m_{34}^p - m_{14}^c m_{32}^c m_{31}^p, \\
a_{z_3} &= m_{11}^c m_{24}^c m_{32}^p + m_{12}^c m_{21}^c m_{34}^p + m_{14}^c m_{22}^c m_{31}^p - m_{11}^c m_{22}^c m_{34}^p - m_{12}^c m_{24}^c m_{31}^p - m_{14}^c m_{21}^c m_{32}^p, \\
b_{x_1} &= m_{22}^c m_{33}^c m_{24}^p + m_{23}^c m_{34}^c m_{22}^p + m_{24}^c m_{32}^c m_{23}^p - m_{22}^c m_{34}^c m_{23}^p - m_{23}^c m_{32}^c m_{24}^p - m_{24}^c m_{33}^c m_{22}^p, \\
b_{x_2} &= m_{12}^c m_{34}^c m_{23}^p + m_{13}^c m_{32}^c m_{24}^p + m_{14}^c m_{33}^c m_{22}^p - m_{12}^c m_{33}^c m_{24}^p - m_{13}^c m_{34}^c m_{22}^p - m_{14}^c m_{32}^c m_{23}^p, \\
b_{x_3} &= m_{12}^c m_{23}^c m_{24}^p + m_{13}^c m_{24}^c m_{22}^p + m_{14}^c m_{22}^c m_{23}^p - m_{12}^c m_{24}^c m_{23}^p - m_{13}^c m_{22}^c m_{24}^p - m_{14}^c m_{23}^c m_{22}^p, \\
b_{y_1} &= m_{21}^c m_{34}^c m_{23}^p + m_{23}^c m_{31}^c m_{24}^p + m_{24}^c m_{33}^c m_{21}^p - m_{21}^c m_{33}^c m_{24}^p - m_{23}^c m_{34}^c m_{21}^p - m_{24}^c m_{31}^c m_{23}^p, \\
b_{y_2} &= m_{11}^c m_{33}^c m_{24}^p + m_{13}^c m_{34}^c m_{21}^p + m_{14}^c m_{31}^c m_{23}^p - m_{11}^c m_{34}^c m_{23}^p - m_{13}^c m_{31}^c m_{24}^p - m_{14}^c m_{33}^c m_{21}^p, \\
b_{y_3} &= m_{11}^c m_{24}^c m_{23}^p + m_{13}^c m_{21}^c m_{24}^p + m_{14}^c m_{23}^c m_{21}^p - m_{11}^c m_{23}^c m_{24}^p - m_{13}^c m_{24}^c m_{21}^p - m_{14}^c m_{21}^c m_{23}^p, \\
b_{z_1} &= m_{21}^c m_{32}^c m_{24}^p + m_{22}^c m_{34}^c m_{21}^p + m_{24}^c m_{31}^c m_{22}^p - m_{21}^c m_{34}^c m_{22}^p - m_{22}^c m_{31}^c m_{24}^p - m_{24}^c m_{32}^c m_{21}^p, \\
b_{z_2} &= m_{11}^c m_{34}^c m_{22}^p + m_{12}^c m_{31}^c m_{24}^p + m_{14}^c m_{32}^c m_{21}^p - m_{11}^c m_{32}^c m_{24}^p - m_{12}^c m_{34}^c m_{21}^p - m_{14}^c m_{31}^c m_{22}^p, \\
b_{z_3} &= m_{11}^c m_{22}^c m_{24}^p + m_{12}^c m_{24}^c m_{21}^p + m_{14}^c m_{21}^c m_{22}^p - m_{11}^c m_{24}^c m_{22}^p - m_{12}^c m_{21}^c m_{24}^p - m_{14}^c m_{22}^c m_{21}^p, \\
c_1 &= m_{21}^c m_{32}^c m_{33}^p + m_{22}^c m_{33}^c m_{31}^p + m_{23}^c m_{31}^c m_{32}^p - m_{21}^c m_{33}^c m_{32}^p - m_{22}^c m_{31}^c m_{33}^p - m_{23}^c m_{32}^c m_{31}^p, \\
c_2 &= m_{11}^c m_{33}^c m_{32}^p + m_{12}^c m_{31}^c m_{33}^p + m_{13}^c m_{32}^c m_{31}^p - m_{11}^c m_{32}^c m_{33}^p - m_{12}^c m_{33}^c m_{31}^p - m_{13}^c m_{31}^c m_{32}^p, \\
c_3 &= m_{11}^c m_{22}^c m_{33}^p + m_{12}^c m_{23}^c m_{31}^p + m_{13}^c m_{21}^c m_{32}^p - m_{11}^c m_{23}^c m_{32}^p - m_{12}^c m_{21}^c m_{33}^p - m_{13}^c m_{22}^c m_{31}^p, \\
d_1 &= m_{21}^c m_{33}^c m_{22}^p + m_{22}^c m_{31}^c m_{23}^p + m_{23}^c m_{32}^c m_{21}^p - m_{21}^c m_{32}^c m_{23}^p - m_{22}^c m_{33}^c m_{21}^p - m_{23}^c m_{31}^c m_{22}^p, \\
d_2 &= m_{11}^c m_{32}^c m_{23}^p + m_{12}^c m_{33}^c m_{21}^p + m_{13}^c m_{31}^c m_{22}^p - m_{11}^c m_{33}^c m_{22}^p - m_{12}^c m_{31}^c m_{23}^p - m_{13}^c m_{32}^c m_{21}^p, \\
d_3 &= m_{11}^c m_{23}^c m_{22}^p + m_{12}^c m_{21}^c m_{23}^p + m_{13}^c m_{22}^c m_{21}^p - m_{11}^c m_{22}^c m_{23}^p - m_{12}^c m_{23}^c m_{21}^p - m_{13}^c m_{21}^c m_{22}^p.
\end{aligned}$$

## Acknowledgements

This work was sponsored, in part, by National Institute of Hometown Security, Somerset, KY, contract 17-07-UK, and by the Kentucky Science and Technology Company Inc., under contract KSEF-148-502-08-226.