Ray-Tracing Codec for Structured Light 3D Camera

Lam Quang Bui, and Sukhan Lee†, *Fellow, IEEE*

***Abstract*— In this paper, we present a new method for decoding pixel correspondences in structured light based 3D reconstruction, refer to here as Ray-Tracing codec. The key idea of Ray-Tracing codec is to correctly define the region boundaries in real number, for each layer of the Hierarchical Orthogonal Code (HOC) based on an accurate boundary estimator, and to inherit the correct region boundaries between layers sharing common boundaries. Furthermore, each region in lower layer is traced back to the upper layer for the correct correspondence between regions. This is an improvement over existing HOC decoding algorithms as the wrong decoded pixel correspondences can be greatly reduced. The experimental results have shown that the proposed Ray-Tracing codec significantly enhances the robustness and precision in depth imaging, compare with HOC and other well-known conventional approach. The proposed approach opens a greater feasibility of applying structured light based depth imaging to a 3D modeling of cluttered workspace for home service robots.**

I. Introduction

Depth imaging based on structured light has drawn much attention due to its potential for applications in many areas such as robotics, 3D games, precision measurements, etc. Recently, many robots are equipped with a structured light based imaging system instead of traditional passive stereo cameras for object recognition and manipulation in short range, some well-known robots are PR2 [1], TurtleBot [2] and T-Rot [3], etc. This potential is mainly from its capability of measuring depth of textureless objects and also providing higher precision and robustness than passive stereo cameras as well as higher speed and lower cost than laser scanners. The key technology for depth imaging based on structured light lies in the methodology of achieving correct pixel correspondence between DMD (Digital Mirror Device) of a projector and a camera. The accuracy of depth imaging depends directly on the correctness of pixel correspondence.

There are many approaches [4][5] to code light patterns available to date, and they can be classified in to three classes: temporal patterns, spatial patterns and mixed forms. The spatial patterns have a small number of projected patterns that can cope with moving scenarios, their weakness is the low accuracy in 3D data. The temporal patterns have a large number of projected patterns which make them unable to deal with moving scenarios, but they have a very accurate measurement of 3D data. This paper focuses on improving the robustness and precision in depth imaging using temporal patterns based on binary codes.

In the techniques based on binary codes, only two illumination levels are used, which are coded as 0 and 1, and Gray code is widely used to code the pattern. Posdamer et al. [6] were the first to propose a well known effective technique using plain binary coded temporal patterns. Knowing the sensitivity of plain binary code against noise, Inokuchi et al. [7] improved the codification scheme of Posdamer by using Gray code, which is more robust against noise, instead of plain binary. Later, Trobina [8] presented an error model of coded light range sensors based on Gray coded patterns, and demonstrated the importance of accurately locating every stripe in the image. The author proposed two ways of detecting stripe edges with sub-pixel: the first way is to project the reference images (all white and all black) then use the average value as the threshold to determine stripe edge location; an alternative way is to project the additional inverse pattern and find intersection of both profiles, where stripe edge is located.

Some works have been done in order to improve Gray coded patterns. Guhring [9] used Gray code for labeling the code-strings coarsely, and several lines instead of the final few frames to determine the final code-strings. This makes the results more dense and accurate than the Gray code. Kim et al. [10] developed an antipodal Gray code for structured light and showed that the antipodal Gray code provides more robust and accurate results than original Gray code. Sansoni [11] and Wiora [12] combined phase-shift with Gray code patterns to get high-resolution 3D reconstruction.

However, these conventional approaches suffer either from large pixel-wise variations in accuracy or from a number of spurious outliers, especially, near occluding and shading boundaries. This is because the conventional approaches focus on decoding the input signals based only on the intensity values of the pixels at the position to be decoded. It is problematic in decoding the pixel correspondences when the signals are corrupted by system and environmental noise, due to scattering, variation in reflectance and illumination, particularly when the light source is relatively weak.

In this paper, we employ a previously developed structured light pattern, Hierarchical Orthogonal Code [13], and propose a new decoding method to overcome those weaknesses

This research was performed for the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs (F0005000-2010-32), and in part by the KORUS-Tech Program (KT-2010-SW-AP-FSO-0004) funded by the Korea Ministry of Knowledge Economy (MKE). This work was also partially supported by WCU (World Class University) program through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology (R31-10062), by PRCP through NRF of Korea funded by MEST (2011-0018397), by MKE, Korea under ITRC NIPA-2011-(C1090-1121-0008).

Lam Quang Bui is with the Intelligent Systems Research Institute, School of Information and Communication Engineering, Sungkyunkwan University, South Korea (e-mail: quanglam@ece.skku.ac.kr).

Sukhan Lee†, the Corresponding Author, is with the Intelligent Systems Research Institute, School of Information and Communication Engineering and the Interaction Science Department of Sungkyunkwan University, South Korea (e-mail: lsh@ece.skku.ac.kr).

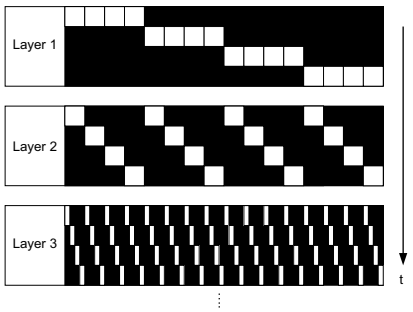

Fig. 1: Hierarchical Layer of Code. Each layer consists of 4 patterns, each code in upper layer is divided in to 4 sub-codes in lower layer.

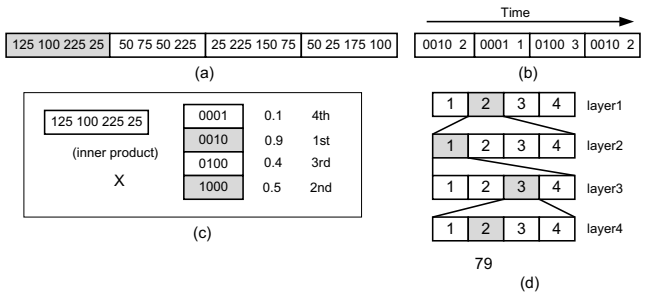

Fig. 2: A conceptual description of an HOC based signal decoding process which is including (a) a signal of pixel values at *i*-th position, (b) signal separating by multiplying an orthogonal matrix, (c) selecting a set of probable codes, and (d) decoding of the correct address.

of the conventional method as well as improve the precision in depth imaging. Unlike the conventional approaches, the proposed approach decodes the pixel correspondences based on the regions and inherits the correspondence information of regions from upper layer to lower layer: First the pattern stripe boundaries are accurately estimated; the boundaries of shadow in camera view as well as in projector view are also estimated. Then in each layer, the regions between boundaries are defined, also their correspondences are determined. Finally, each region in lower layer is traced back to the containing region in upper layer to inherit the correspondence information, thus the correspondence of the current region is correctly updated.

The remainder of the paper is organized as follows: In Section II, we describe our Ray-Tracing codec. The experimental results are provided in Section III. Finally, Section IV concludes the paper.

II. The Proposed Method

A. Structured Light Pattern

In previous work [13], we developed a structured light pattern named as "HOC (Hierarchical Orthogonal Coding)", in which the orthogonal codes are arranged hierarchically in order to reduce the length of codes. The length f of code signals divided into a few layers L and each layer includes H orthogonal codes recursively as shown in Fig. 1. Although the signal codes in the HOC are not orthogonal, each layer has a set of orthogonal codes. For more details, please refer [13].

For example, we assume that a HOC has four layers ($L = 4$) and the number of orthogonal codes in each layer is also four ($H_1 = H_2 = H_3 = H_4 = 4$). In this case, the total number of signal codes is 256 ($H_1 \times H_2 \times H_3 \times H_4 = 256$) and the code length is 16 ($H_1 + H_2 + H_3 + H_4 = 16$), i.e. we need 16 frames of camera image for decoding addresses of the signals.

The decoding process of HOC patterns is as follows (as illustrated in Fig. 2). At the *i*-th pixel position, in each layer we have a set of pixel intensity $y_i = (I1, I2, I3, I4)^T$ from four patterns. The signals are separated by multiplying y_i with an orthogonal matrix: $c_i = Xy_i$, then selecting a probable code for c_i. Repeating for all layers, we have a set of probable codes, and then decode the correct address.

However, this decoding method is sensitive to variation of surface reflectance, especially at the boundaries of the patterns; and not accurate enough in depth measurement since the true boundaries of the light stripes cannot be located. In following sections, we describe a new decoding method for HOC patterns which is robust to variation of surface reflectance (by a top-down approach with region based correspondence) and accurate in depth measurement (by accurately estimate the stripe boundaries).

B. Structured Light Pattern Decoding Method

1) Accurate Stripe Boundary Estimation: Accurately estimating the stripe boundary is crucial issue in improving depth measurement of structured light 3D camera systems. Since the projector correspondence is computed regarding the boundaries of light stripes on the camera images.

Here we adopt the result of a previous work on designing an accurate boundary estimator [14]. The captured light stripe signal in camera image, $f_s(x)$, can be modeled as follows:

$$f_s(x) = ((s(x) \otimes g_p(x, \sigma_p))\, R(x) + A(x)) \otimes g_c(x, \sigma_c) + W(x) \quad (1)$$

where $s(x)$ is the pattern for projector illumination, which is a step function:

$$s(x) = \begin{cases} H & x \geq 0 \\ L & x < 0 \end{cases},$$

symbol $\otimes$ represents a convolution operator. In Eq. (1), the two blurring processes associated with the projector and camera lenses are modeled as a convolution with the respective blur kernels, $g_p(x, \sigma_p)$ and $g_c(x, \sigma_c)$. The blur kernels, $g_p(x, \sigma_p)$ and $g_c(x, \sigma_c)$, are chosen to be a normalized Gaussian function with (x, σ_p) and (x, σ_c) representing their respective (center, variance) pairs:

$$g_i(x, \sigma_i) = \frac{1}{\sigma_i \sqrt{2\pi}} e^{\left(-\frac{x^2}{2\sigma_i^2}\right)},$$

where i is p or c.

$R(x)$ is the reflection index of the local surface at x, $R(x) \in [0, 1]$; $A(x)$ is the ambient light, and $W(x)$ is the noise of the imaging sensor.

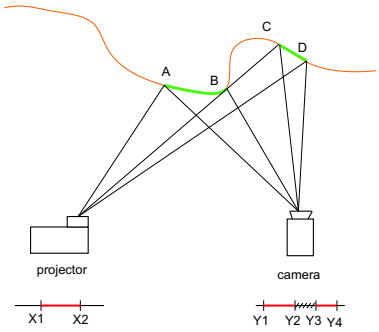

Fig. 3: The shadow in camera view. The region [Y2,Y3] corresponds to the shadow [B,C].

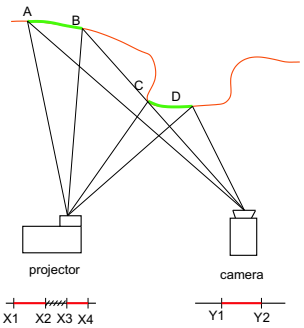

Fig. 4: The shadow in projector view. The region [X2,X3] corresponds to the shadow [B,C].

Also the light patterns captured on the image in correspondence to the respective all-bright and all-dark pattern projections [i.e., $s_1(x) = H$ and $s_0(x) = L$]:

$$f_1(x) = (H * R(x) + A(x)) \otimes g(x, \sigma_c) + W_1(x)$$
$$f_0(x) = (L * R(x) + A(x)) \otimes g(x, \sigma_c) + W_0(x)$$

Then the amount of light from the projector hitting the local surface corresponding to x is estimated as:

$$f_c(x) \approx \frac{deconvlucy\left(\left(f_s(x) - f_0(x)\right), \sigma_c\right)}{deconvlucy\left(\left(f_1(x) - f_0(x)\right), \sigma_c\right)} (H - L) \quad (2)$$

where "$deconvlucy$" represents the Richardson-Lucy deconvolution operator [15][16] that we chose to use for the computation of de-convolution.

The canonical form of light pattern, $f_c(x)$, computed by Eq. (2) is regarded as correcting the edges of $f_s(x)$ corrupted by $R(x)$, as well as by $A(x)$ and $g_c(x, \sigma_c)$, thus providing a mean of recovering the true boundary embedded in $s(x)$. We estimate the boundaries of a light pattern by intersecting the canonical form of a light pattern with that of its inverse pattern. For details, please refer [14].

In each layer of HOC patterns, the light stripe in this pattern shares a common boundary with the light stripe in next pattern. Thus two consecutive patterns in the same layer can be treated as a pattern and its inverse pattern, and then the proposed boundary estimation method is applied to find the accurate stripe boundaries. In our experiments, we use 16 HOC patterns and two additional black and white patterns.

2) Boundary of the Shadows in the Camera View: The definition of the shadow in the camera view is the region in captured image that is not reached by the projecting light, as illustrated in Fig. 3.

The region [B,C] is not reached by the projector light, thus in the captured image the corresponding region [Y2,Y3] is the shadow. We need to detect the boundaries of the shadow region which are Y2 and Y3. It is easy to do that after removing out the shadow regions.

3) Boundary of the Shadows in the Projector View: In the structured light imaging system, the projector can be considered as an inverse camera (projecting light instead of capturing light). And similar to the shadow in the camera view, the shadow in the projector view is the region that does not reflect the projecting light to the camera, as described in Fig. 4.

The camera cannot see the region [B,C] which is projected with light pattern, thus the corresponding region in projector image [X2,X3] is the shadow in projector view. Since the region [B,C] does not appear in the captured image, we cannot detect the boundary X2, and X3 directly from the captured image. In order to detect X2, X3, we need to calculate the correspondence values, using all the patterns from layer 1 to layer 4 [13], of pixels from Y1 to Y2 to get the correspondence map X $\in$ [X1,X4], and then the boundaries of shadow in projector view X2, X3 is estimated under the constraint: $|X3 - X2| > 1$.

4) Forming Regions: After detecting all those kinds of boundary, in each layer we define the regions which is formed by two consecutive boundaries, as illustrated in Fig. 5. There are two maps we need to consider: the map of boundaries in camera image and the map of boundaries in projector image.

From captured pattern images, in each layer we can detect all the boundaries including stripe boundaries Y1, Y4, Y5 and shadow boundaries Y2, Y3. All of these boundaries are put in a map. Also from the captured pattern images, in each layer we can compute the correspondence for each pixel [13]. Along each row of the image, we search for positions, named boundaries, where the pixel correspondence value changes; we get the boundary of stripes X1, X2, X5 and the boundaries of shadow X3 and X4 in projector image. Because X1, X2, X5 are the boundaries of black and white fringes in projector image, then corresponding positions in camera image will be Y1, Y4, Y5 respectively.

Combining these two boundary maps we get a region map where two consecutive boundaries enclose a region, as shown in Fig. 4. Where A1 = Y1, A4 = Y4, A6 = Y5; and B1 = X1, B3 = X2, B6 = X5. For the region of shadow [Y2,Y3] in camera image, the corresponding region in the projector image is just only one pixel, so A2 and A3 have the same correspondence value B2. The same way for the region of shadow [X3,X4] in projector image, the corresponding region in camera image is only one pixel, so B4, B5 have the same correspondence value A5.

5) Structured Light Pattern Decoding Algorithm: According to the design of HOC, each layer has regions with local

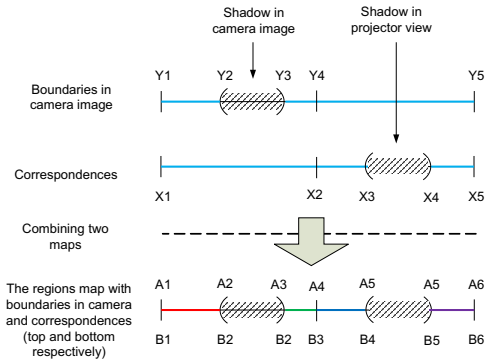

Fig. 5: Region defining procedure in one layer, each color corresponds to one region.

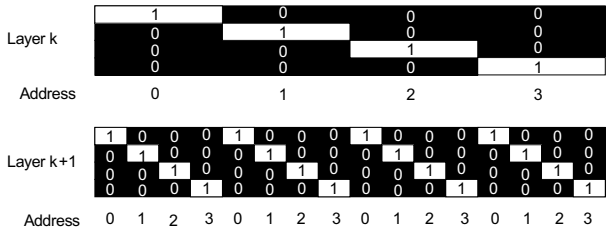

Fig. 6: The HOC and the corresponding addresses.

correspondences. Each of these regions is divided into four sub-regions in next layer, as illustrated in Fig. 6.

Thus we want to update the region-wise correspondences, not pixel-wise correspondences. By doing this way we can eliminate the outliers caused by uncertainties of the pixel-wise correspondence at the boundaries of the regions. The decoding process is as follows. In each layer, we detect all the possible boundaries: stripe boundaries, camera shadow boundaries and projector shadow boundaries. Then the regions are formed from pairs of the consecutive boundaries. The global correspondences of sub-regions in lower layer are updated by inheriting the global correspondence of the containing region in the upper layer plus their own local correspondence. The inheritance is described in Fig. 7.

For example, in the layer k, the region [Y4,Y5] has the correspondence [X4,X5]. In lower layer $k+1$, there are sub-regions [a6,a7] and [a7,a8] having correspondences [b6,b7] and [b7,b8] respectively. Thus the inherited correspondences of the sub-regions are [X4+b6, X4+b7] and [X4+b7, X5+b8].

This procedure is repeated from the first to the last layer, the global correspondence of the regions in the last layer will be the output of the structured light pattern decoding process. The whole decoding algorithm is presented in Fig. 8.

III. Experimental Results

A. Experimental Setup

The proposed Ray-Tracing codec has been implemented in a light weight structured light system, namely Pico system, which can be used for service robot applications. The system consists of an Optoma PK301 Pico projector, a PGR Flea2 1394 digital camera mounted a TV LENS 8*mm* 1:1.3 and

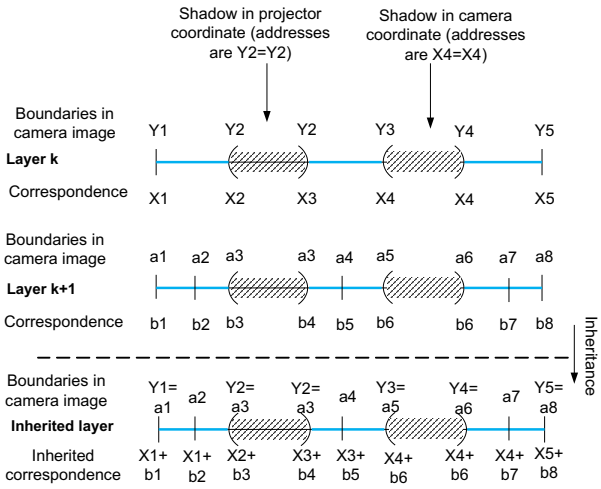

Fig. 7: The inheritance principle. Each sub-region in the lower layer inherits the global correspondence in upper layer to update its correspondence.

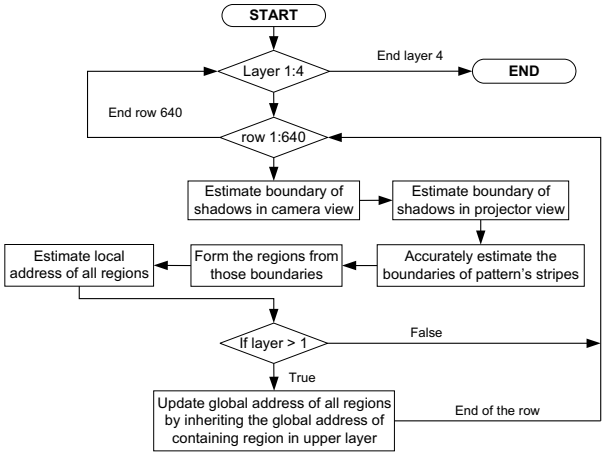

Fig. 8: Region based correspondence decoding algorithm flow chart.

a computer, as illustrated in Fig. 9. The resolution of the projector was 800x600 and that of the camera 640x480. The position of the camera was about 13*cm* on the right of the projector. The computer generates signal patterns, acquires images, and computes depth images. Typical distance of the structured light system from the scene is 1*m*, and the system was calibrated.

B. Results

1) Qualitative Evaluation: To evaluate the qualitative performance of the proposed Ray-Tracing codec, we compare the depth image captured by Ray-Tracing codec with other methods: HOC and Gray code inverse (Trobina's method). We acquired depth images from a real scene (Fig. 11) by Ray-Tracing codec, Gray code inverse and HOC, as shown in Fig. 12, 13 and 14 respectively. As can be seen, outliers in the 3D point cloud are significantly reduced in the result of Ray-Tracing codec compared with HOC and Gray code inverse. Since the brightness of the Optoma PK301 Pico projector is 50 ANSI Lumens, the signal to noise ratio is lower compared

Fig. 9: The setup of the structured light system (Pico system).

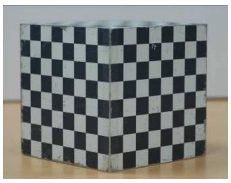

Fig. 10: The calibration block to be measured and compared.

with using other normal presentation projectors. When the signal of pixels around the boundaries of light stripes or of shadows is corrupted by noise/variation in reflectance, the decoded pixel correspondences of HOC or Gray code inverse are wrong, thus outliers in 3D data are created. Whereas, the Ray-Tracing codec uses region based decoding algorithm, the wrong decoding correspondences of single pixels is eliminated, thus the 3D data of Ray-Tracing codec contains much less outliers. We can see that the Ray-Tracing codec contains much less outliers than others methods.

To statistically evaluate the number of outliers, we manually and carefully delete the outliers from the 3D point cloud shown in Fig. 12-14 using Rapidform software [17], also count the number of outliers have been deleted. Table I shows the total number of 3D points in the result of each method and corresponding number of outliers. We can see that the Ray-Tracing codec contains much less outliers than others methods.

TABLE I: Number of 3D points and outliers in the results of Ray-Tracing codec, Gray code inverse (Trobina's method), and HOC

	Total number of 3D points	Number of outliers	Percentage of outliers
Ray-Tracing codec	238397	387	0.16%
Gray code inverse (Trobina's method)	290938	8779	3.02%
HOC	181305	5307	2.93%

2) Quantitative Evaluation: To evaluate the accuracy as well as the effect of variation in surface reflection on the proposed Ray-Tracing codec, Gray code inverse (Trobina's method), and HOC; we captured the 3D data of a face of calibration block made from aluminum (Fig. 10), which has the flatness-values as: standard deviation is 0.260365 μm and max error is 1.149865 μm, by these methods and we measured the errors in the captured data. The error was defined as the distance from the reconstructed points to the fitted plane. As can be seen in Table II, with approximately the same number of points, the accuracy of Ray-Tracing codec is significantly improved compared with other methods.

TABLE II: Errors of Reconstructed 3D Data Using Ray-Tracing codec, Gray code inverse and HOC

	Standard deviation of error (mm)	Error max (mm)	Number of points
Ray-Tracing codec	0.332	3.529	27887
Gray code inverse	0.788	20.997	27888
HOC	6.870	95.434	27077

Fig. 11: The scene to be captured.

C. Compare Pico System with Kinect Sensor

Kinect sensor is the commercial version of Primesense device [18], which is widely used recently in robotics and computer vision communities. In this Section, we show a comparison between Pico system endowed with Ray-Tracing codec and Kinect sensor.

In first experiment, a Venus statue is placed about $1m$ in front of the devices, as shown in Fig. 15. Pico system and Kinect sensor captured 3D point cloud of the scene. Fig. 16 and 17 show the 3D points in different views captured by Pico system and Kinect sensor respectively. As can been seen, the Pico system with Ray-Tracing codec gives a clean, dense and detailed 3D point cloud. Meanwhile the 3D point cloud resulted by Kinect sensor is sparse, loses the details of the object.

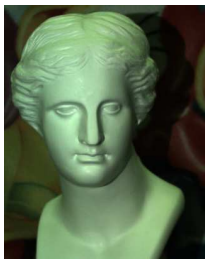

Fig. 15: The object to be captured.

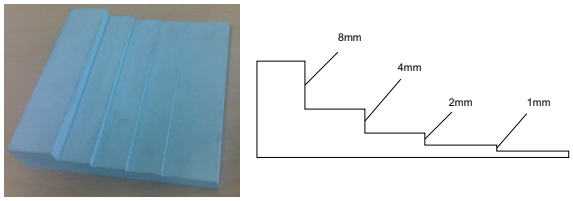

Fig. 18: The exponential staircase. The steps are approximately $1mm$, $2mm$, $4mm$ and $8mm$.

Fig. 12: The 3D point cloud of the scene captured by Ray-Tracing codec (left: front view; center: top view; right: side view).

Fig. 13: The 3D point cloud of the scene captured by Gray code inverse (Trobina's method) (left: front view; center: top view; right: side view).

Fig. 14: The 3D point cloud of the scene captured by HOC (left: front view; center: top view; right: side view).

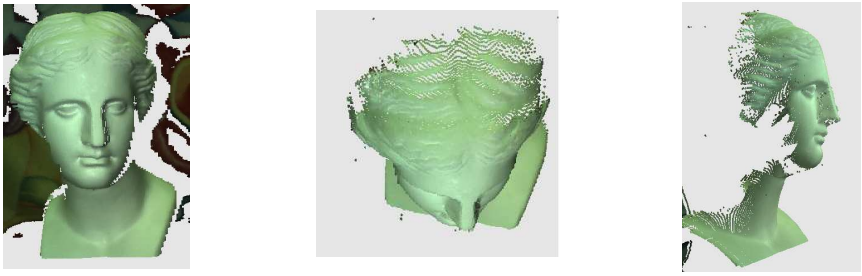

Fig. 16: The 3D point cloud of the Venus statue captured by Pico system (left: front view; center: top view; right: side view).

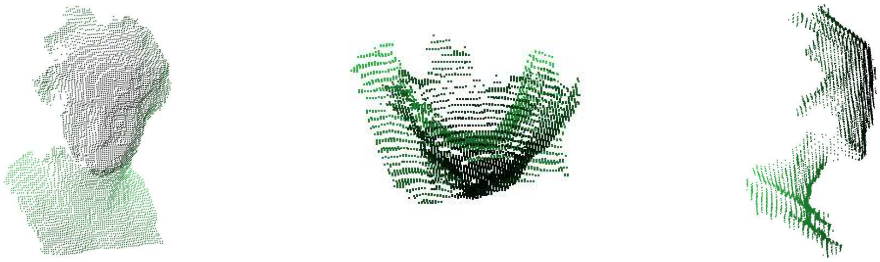

Fig. 17: The 3D point cloud of the Venus statue captured by Kinect sensor (Trobina's method) (left: front view; center: top view; right: side view).

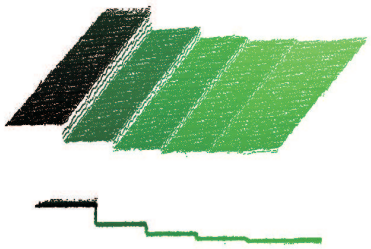

Fig. 19: The 3D point cloud of the staircase captured by Pico system, with 45^0 view angle (top) and side view (bottom).

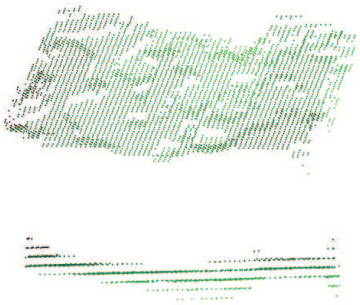

Fig. 20: The 3D point cloud of the staircase captured by Kinect sensor, with 45^0 view angle (top) and side view (bottom).

In second experiment, a staircase, in which the steps are: $1mm$, $2mm$, $4mm$ and $8mm$ (Fig. 18), is positioned at a distance of about $90cm$ from the imaging devices. The acquired 3D point clouds of the staircase by Pico system and Kinect sensor are shown in Fig. 19 and 20 respectively. As can be seen in those figures, we can distinguish the bottom stair with $1mm$ height in 3D point cloud of Pico system, but we cannot even locate the position of the top stair with $8mm$ height in the 3D point cloud of Kinect sensor. Table III describes the measurement accuracy of the staircase.

TABLE III: The average and standard deviation of the depth measurement in each stair of the staircase with respect to the bottom stair

		Pico system	Kinect sensor
Top stair	Average depth $[mm]$	14.600	Cannot locate
	Standard deviation $[mm]$	0.2640	the stair
2nd stair	Average depth $[mm]$	6.8000	Cannot locate
	Standard deviation $[mm]$	0.2179	the stair
3rd stair	Average depth $[mm]$	2.9000	Cannot locate
	Standard deviation $[mm]$	0.1891	the stair
4th stair	Average depth $[mm]$	0.9989	Cannot locate
	Standard deviation $[mm]$	0.1776	the stair
Bottom stair	Average depth $[mm]$	0.1337	Cannot locate
	Standard deviation $[mm]$	0.1074	the stair

IV. Conclusions

We have presented a new method to decode the pixel correspondences for HOC structured light pattern in order to improve the robustness and accuracy of structured light 3D imaging systems. The key idea of the proposed method is to define regions in each layer of the code with the support of an accurate boundary estimator, and then each region in lower layer is traced back to the containing region in upper layer to inherit the correct correspondence information. With better quality projectors, we can achieve more accurate measurements, yet in a same projector-camera system, we have shown that the Ray-Tracing codec is more robust and accurate than other conventional methods. The results show the improvements of the proposed Ray-Tracing codec compared with other methods. However, the processing time is still high, the Ray-Tracing codec is implemented on a NVIDIA Tesla C2050 GPU and running at 3Hz, whereas Kinect sensor outputs video at a frame rate of 30Hz. Our future work is to improve the algorithm in order to speedup the system.

References

[1] PR2, Willow Garage, *www.willowgarage.com/pages/pr2/overview*

[2] TurtleBot, Willow Garage, *www.willowgarage.com/turtlebot*

[3] M. Kim, S. Kim, S. Park, et al., "Service robot for the elderly," *Robotics & Automation Magazine, IEEE*, vol. 16, no. 1, 2009, pp. 34-45.

[4] F. Blais, "Review of 20 Years of Range Sensor Development," *Journal of Electronic Imaging*, vol. 13, no. 1, 2004, pp. 231-240.

[5] J. Salvi, S. Fernandez, T. Pribanic et al., "A state of the art in structured light patterns for surface profilometry," *Pattern Recognition*, vol. 43, no. 8, 2010, pp. 2666-2680.

[6] J.L. Posdamer, M.D. Altschuler, "Surface measurement by space-encoded projected beam systems," *Comput. Graph. Image Process*, vol. 18, no. 1, 1982, pp. 1-17.

[7] S. Inokuchi, K. Sato, F. Matsuda, "Range imaging system for 3-D object recognition," *Proceedings of the International Conference on Pattern Recognition*, 1984, pp. 806-808.

[8] M. Trobina. "Error model of a coded-light range sensor," *Technical Report BIWI-TR-164*, Communication Technology Laboratory, ETH-Zentrum, 1995.

[9] J. Guhring, "Dense 3-D surface acquisition by structured light using off-the-shelf components," *Proc. of the SPIE Photonics West, Electronic Imaging 2001, Videometrics and Optical Methods for 3D Shape Measurement VII*, vol. 4309, pp. 220-231, 2001.

[10] D. Kim; M. Ryu; S. Lee; , "Antipodal gray codes for structured light," *IEEE International Conference on Robotics and Automation*, 2008, pp.3016-3021.

[11] G. Sansoni, A. Patrioli, F. Docchio, "OPL-3D: a novel, portable optical digitizer for fast acquisition of free-form surfaces," *Review of Scientific Instruments*, vol. 74, no. 4, 2003, pp. 2593-2603.

[12] G. Wiora, "High resolution measurement of phase-shift amplitude and numeric object phase calculation," *Proceedings of SPIE Vision Geometry IX*, 2000, pp. 289-299.

[13] S. Lee, J. Choi, D. Kim et al., "Signal Separation Coding for Robust Depth Imaging Based on Structured Light.", *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005, pp. 4430-4436.

[14] S. Lee, L.Q. Bui, "Accurate estimation of the boundaries of a structured light pattern," *J. Opt. Soc. Am. A*, vol. 28, no. 6, 2011, pp. 954-961.

[15] W. H. Richardson, "Bayesian-Based Iterative Method of Image Restoration," *J. Opt. Soc. Am. A*, vol. 62, 1972, pp. 55-59.

[16] L. B. Lucy, "An iterative technique for the rectification of observed distributions," *Astronomical Journal*, vol. 79, 1974, pp. 745-754.

[17] Rapidform, *www.rapidform.com*

[18] Primesense, *www.primesense.com*