2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Antipodal Gray Codes for Structured Light

Daesik Kim, Moonwook Ryu, and Sukhan Lee, *Member, IEEE*

***Abstract*—A Gray code and its variants are popular code-patterns for a structured light system. An *n*-bit Gray code is a kind of binary code whose adjacent code-strings differ only in one bit position. We introduce a specified Gray code, called an 'antipodal Gray code.' Since the *n*-bit antipodal Gray code has the additional property that the complement of any code-string appears exactly *n* steps away in the list, the spatial frequency (the width between the white and black stripes) of the antipodal Gray code-pattern is similar along frames. In this paper, we describe the limitations of structured light and the criteria for robust codes. We evaluate the original and antipodal Gray codes, and the experimental results show that the antipodal Gray code provides more robust and accurate results than original Gray codes.**

I. Introduction

AMONG several structured light codes (patterns) [1], a Gray code [2] is one of the popular codes for static environment reconstruction. Gray code characteristics include that they use only black and white colors (i.e. 0 or 1) and differ only in one bit position in adjacent code-strings. These are the main reasons that Gray codes are robust to noise and very suitable to structured light. In order to generate t code-strings, the Gray code requires $\log_2 t$ bits (i.e. frames, pattern images). As a result, the Gray code is less time consuming than the line shifting method, which requires t frames for t code-strings.

The drawback of the conventional temporal codes, including Gray codes, is that the black and white stripes become narrow and dense whenever a frame increases (see Fig. 3). When the camera captures these frames, it is hard to clearly resolve and binarize the patterns, since they look blurry.

Several approaches have so far been attempted for overcoming this problem. Gühring [3] used several lines instead of the final few frames. The Gray code is used for labeling the code-strings coarsely, and the line is used to determine the final code-strings. Though this is more accurate than the Gray code, the decoding result of the Gray code affects the final accuracy since the lines depend on the upper frames of the Gray code. Trobina [4] used the additional inverse (complement) pattern of the Gray code to binarize more robustly. This approach provides a better result but it is still difficult to binarize the images in the lower frames with narrow black and white stripes. Gärtner [5] modified the Gray code and proposed a new code, but it is not easy to re-produce this code when the number of code-strings is changed.

In this paper, we apply a specified Gray code, called an antipodal Gray code to the structured light system. The antipodal Gray code was first generated by Killian and Savage [6]. The n-bit antipodal Gray code has the same characteristics as a Gray code which differs only one bit in successive strings, and the additional property that the complement of any string appears exactly n steps away in the list. Unlike the conventional temporal codes, the spatial frequencies of the pattern images do not vary too much along the frames. This property makes it easier for the camera to decode the patterns, so it is effective in reducing errors due to noise and binarization.

The rest of this paper is organized as follows. In section II, we describe the limitations of structured light; in section III, we propose the criteria for the robust structured light code. Section IV introduces the antipodal Gray code and its characteristics, and section V presents the decoding techniques. Experimental results are shown in section VI, and we discuss the results and conclude in sections VII and VIII, respectively.

II. Problem Description

A. Spreading of Light on the Objects' Surface

In the practical structured light system, even though a single pixel of the DMD or LCD projector (or spotlight of the laser module) is illuminated, it does not correspond to one specific pixel in the camera. This is because of the following reasons. The size of the projected light of the projector (some specific position of the DMD or LCD) and the size of the pixel of the camera can be different; in addition, the focal lengths can be different. Generally, the projected light spreads over onto the surface of the objects depending on the albedo, and this light (precisely speaking, the photons of the imaging device) sometimes overflows and affects its adjacent pixels when the camera captures the scene.

Suppose that the projected light spreads over with some distribution on the surface of the objects. If the light reaches some pixels of the imaging sensor in the camera, the center pixels among these are brighter compared to the neighbor pixels, since the center pixels are more influenced by their

Manuscript received September 14, 2007. This paper was performed for the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Commerce, Industry and Energy of Korea. This work was supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MOST) (No. R01-2006-000-11297-0). This work is supported by the Science and Technology Program of Gyeonggi province.

Daesik Kim and Moonwook Ryu are with the Intelligent Systems Research Center, Sungkyunkwan University, Suwon, Korea, as a Ph. D and a master's student respectively. (e-mail: {daesik80, moonwook}@ece.skku.ac.kr).

Sukhan Lee is with the Intelligent Systems Research Center, Sungkyunkwan University, Suwon, Korea, as a professor and a director. (e-mail: lsh@ece.skku.ac.kr).

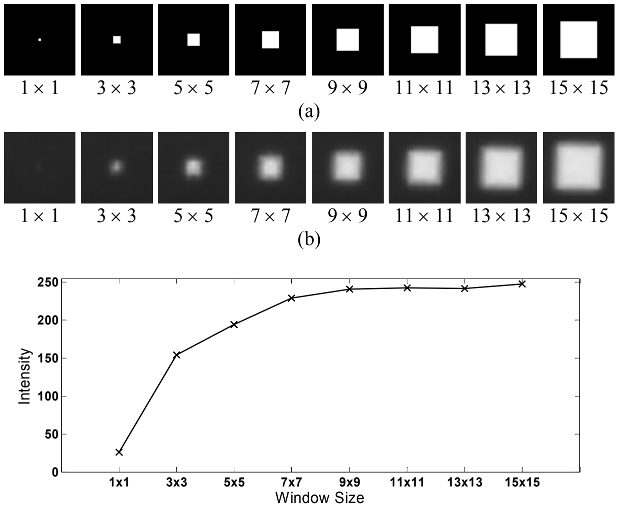


Fig. 1. Example of the Projector-Camera Response. The camera response can vary depending on the size of the projected area even though the projector projects the same intensity light.

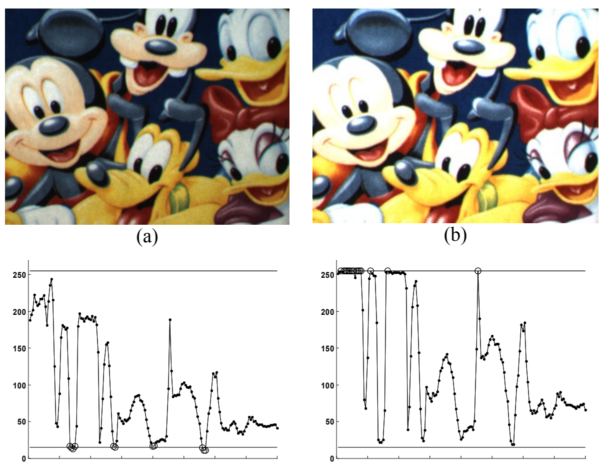


Fig. 2. Limitation of Imaging Device Intensity Range. The imaging device cannot cover the whole intensity range when the objects have different surface reflection properties.

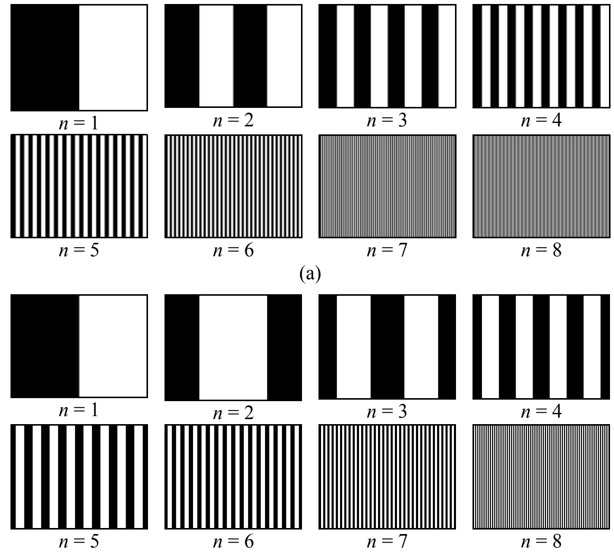


Fig. 3. 8-bit binary and Gray Code. The spatial frequency of the pattern images increases over time.

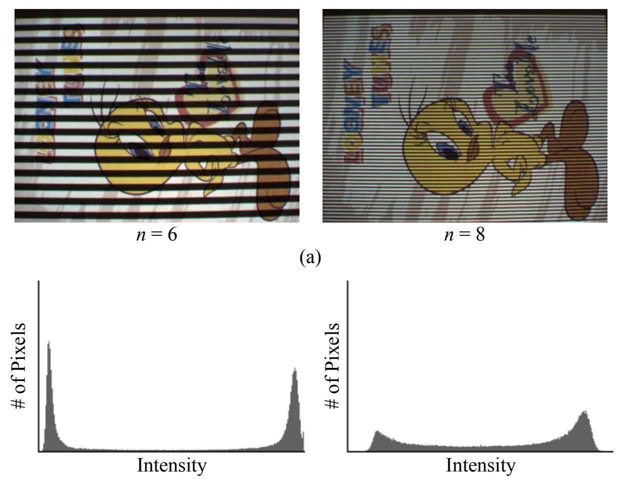


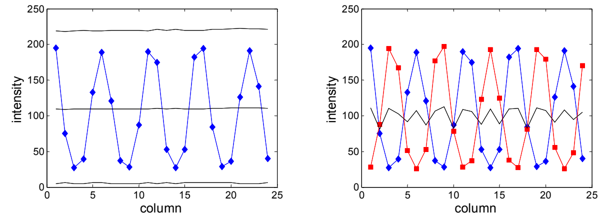
Fig. 4. Histogram of the Captured Image with Gray code. The two peaks of the histogram are lower when the frame increases.

neighbors' light distribution. Therefore, even though the projector projects the same intensity light (e.g. white, 255 intensity), the camera can respond differently depending on the size of the projected area though the albedo of the objects' surface is the same.

Fig. 1 (a) shows examples of when the projector projects from a 1×1 to a 15×15 white square, and Fig. 1 (b) is the corresponding image in the camera. Fig. 1 (c) shows the gray-level intensity of the center pixels of the square in the imaging sensor. As shown in Fig. 1 (c), the intensity value increases up to a limited amount. In practice, a 1×1 single-pixel-size light affects 3~5 neighbor pixels in the camera, but this can vary depending on the properties of the object's surface, exposure time, the sensitivity of the imaging device, etc.

B. Limitation of Intensity Range of an Imaging Device

The intensity range of an imaging device, such as a standard CCD or CMOS, is 2^8 (0~255). Fig. 2 gives the example of when a white image is projected to the wall which is covered by textile. As shown in Fig. 2 (a), if we set the camera exposure time to short, or the aperture size is small, the camera may not detect the light at some pixels. These pixels are represented as circles in Fig. 2 (c). Similarly, as shown in Fig. 2 (b), if we set the camera parameters and lens aperture to obtain a brighter image, some pixels may be saturated and the light overflows to its neighbor pixels. These pixels are represented as circles in Fig. 2 (d). Consequently, the imaging device cannot cover the whole intensity range when the objects have different surface reflection properties



(a) (b)

Fig. 5. Two Binarization Methods. (a) Project the reference images and use the average value as the threshold, (b) project the additional inverse pattern images, and use the zero-crossing value as the threshold.

0	0	0	0	1	1	1	1
0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**

(a)

0	0	0	0	1	1	1	1
0	0	1	1	1	1	0	0
0	1	1	0	0	1	1	0
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**

(b)

Fig. 6. Appearance of Incorrect Code-string. If the code does not have the hamming distance one, a wrong code-string may appear due to noise.

such as color, albedo, etc. This is the fundamental limitation of structured light.

C. High Spatial Frequency of the Pattern Image

Fig. 3 (a) and (b) show the 8-bit binary code and Gray code images respectively. When the frame increases, the frequency of the pattern images increases. Even though the spatial frequency of the Gray code pattern image is half that of the binary code at the same frame (i.e. the width of the white and black stripes in the Gray code is twice as broad than in the binary code at the same frame), the camera still may not resolve the pattern.

Fig. 4 (a) is the captured images when the frame of the Gray code is 6 and 8, and Fig. 4 (b) is the corresponding histogram. When the frame is 8, the number of pixels where it is unclear whether they are black or white increases (the pixels which are located near the middle of the intensity). In this example, the camera resolution was 640 × 480, the number of code-strings was 256, and the width of the stripes was about 3.75 pixels in the image. If the camera resolution is lower, the two peaks of the histogram will also be lower.

When the high spatial frequency pattern images are illuminated, the stripes captured are normally dark and blurred. This is for a similar reason to what we described in section II.A. Thus, it is hard to determine the correct white and black stripe position with this pattern. Consequently, if the spatial frequencies of the pattern images are too high, the camera has difficulty resolving the images, so the probability of the decoding error increases.

D. Binarization Error

When we use conventional temporal codes such as a Gray code (Fig. 3), we should determine whether the pixels from the obtained images are white or black. In order to binarize these, local thresholds need to be defined (which can be different for each pixel). The following methods are

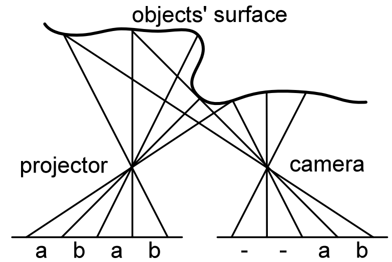


Fig. 7. Pattern Ambiguity. When the projector projects the patterns labeled as 'a, b, a, b', the camera captures 'a, b'. In this case, we cannot define where these 'a' and 'b' come from in the projector.

commonly used:

1. project the reference images (fully-projected and non-projected, i.e. white and black) and use the average value as the threshold, or
2. project the additional inverse (complement) pattern images, and use the zero-crossing value as the threshold.

These methods are illustrated in Fig. 5. In many cases, we cannot trust the first method since the intensity of the white image obtained by the camera cannot represent the white stripe in each frame. As mentioned above, this is because the gray-level intensity of the incident light varies over time (frames) according to the influence from neighbor pixels. For example, the white stripes of the first frame (n = 1) appear brighter than those of the last frame (n = 8) in practice, similar to Fig. 4 (a). The second method is more reliable. However, as shown in Fig. 4 (b), if the response of the camera is too weak, we may not detect the projected light in the image. This normally occurs when the stripes are narrow (i.e. high spatial frequency image) or the albedo is low.

E. Appearance of Incorrect Code-string

When the hamming distance between the code-strings is not equal to one, decoding error can be increased due to noise. For example, as shown in Fig. 6 (a), in a 3-bit binary code pattern, if the first bit (i.e. frame) of the code-string '0 1 1' (labeled as 3) is affected by that of the code-string '1 0 0' (labeled as 4), it may be mis-decoded as code-string '1 1 1' (labeled as 7). In contrast to this issue in binary code, decoding error is small in a 3-bit Gray code pattern, as shown as shown in Fig. 6 (b).

At the occluding boundary, where the depth changes dramatically in the depth image, the Gray code may be mis-decoded. This is because code-strings which are apart from each other can be located together in the image captured by the camera, and these strings cannot have a hamming distance of one.

III. Criteria of Robust Codes

In this section, we present the criteria for robust codes in structured light. These criteria are similar to the one proposed by Gartner et al [5].

A. Unique Code-string

The code-strings need to be uniquely identified. This

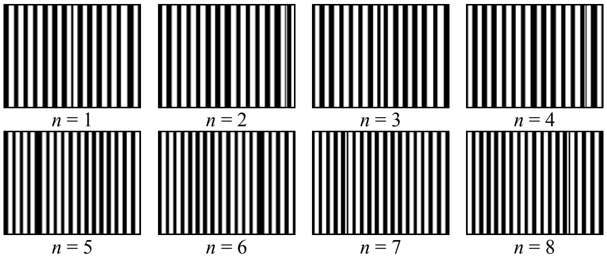


Fig. 8. 8-bit Antipodal Gray Code. The successive strings differ only in one bit position, but this code has the additional property that the binary complement of any code-string appears exactly n steps away in the list.

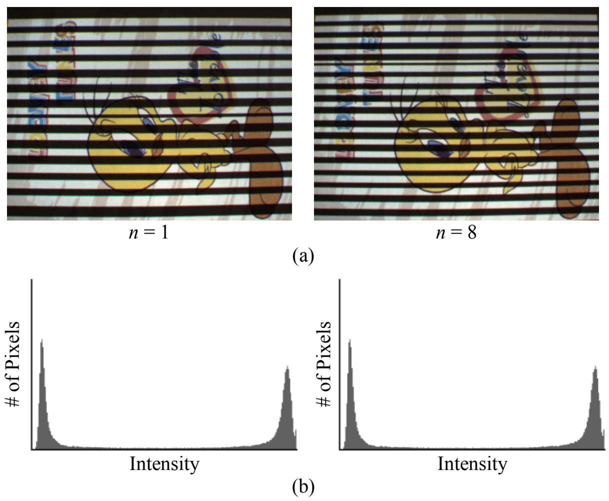


Fig. 9. Histogram of the Captured Image with antipodal Gray code. The two peaks of the histogram are similar even though the frames are different.

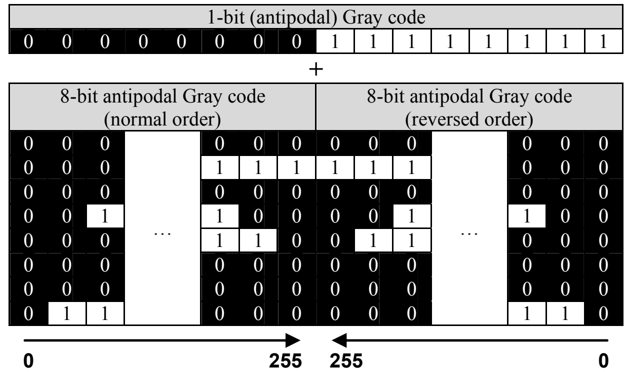


Fig. 10. Expansion of antipodal Gray code (9-bit code). Expand t times (t is a multiple of 2) the 8-bit antipodal Gray code with its reverse order code recursively, then add the additional $\log_2 t$ frame with original Gray code or antipodal Gray code

means that the each code-string has to be appear only once among the strings. This criterion is required when we measure the several objects located near and far from the structured light system, since some code-strings may not be captured by the camera. For example, as shown in Fig. 7, when the projector projects the code-strings ‘a, b, a, b’, the camera can only capture ‘a, b’. In this case, we cannot define where ‘a’ and ‘b’ are from in the projector.

B. Hamming Distance of One

As described in section II.E, the hamming distance is one of the important factors in the structured light pattern. In the case of Gray code, as shown in Fig. 3, even though the error occurs due to the influence of the neighboring pixels on a continuous surface, the error can be small. So, this is very well adopted criterion for a robust temporal code.

C. Low Spatial Frequency Pattern Image

As described earlier, even though the projector projects the same intensity light, the corresponding gray-level intensity of the camera varies depending on the size of the projected areas. If the spatial frequencies of the pattern images are low and similar along time, the camera would resolve the pattern well; consequently, binarization error would be reduced.

IV. Antipodal Gray Codes

A. Definition

An antipodal Gray code was invented by Killian and Savage [6]. An n-bit antipodal Gray code has the same property as an ordinary n-bit Gray code, which is that the successive strings differ only in one bit position, but it has the additional property that the binary complement of any code-string appears exactly n steps away in the list. Fig. 8 shows the example of the 8-bit antipodal Gray code. According to Killian and Savage, the antipodal gray code does not exist when odd $n > 3$ or $n = 6$, and they showed the existence of the codes for $n=2^k$. However, whether an antipodal Gray code whose n=10 exists or not is not proven.

B. Characteristics

The width of the white or black stripes of the antipodal Gray code is mostly uniform, so the camera response to the projected white stripe is similar over time; consequently, this code suffers less from the intensity variation which occurs when there is variation of the projected white-black stripes’ widths.

Fig. 9 (a) shows the images captured when the frame of the antipodal Gray code is 1 and 8, and Fig. 9 (b) is the corresponding histogram. The histograms look similar between the two frames. In this example, the camera resolution was 640×480, and the number of code-strings was 256.

In the original 8-bit Gray code, the number of black-white stripe boundaries in the 8^{th} frame is 128 and the other 128 boundaries are located from the 1^{st} to the 7^{th} frame. On the other hand, in the 8-bit antipodal Gray code, the number of stripe boundaries is about 32 from the first to the last frame, but the total number of stripe boundaries is the same, 256.

C. Expansion

Since only 2^k-bit antipodal Gray code was proved by Killian and Savage, it is required to modify the code to generate n-bit codes other than 2^k-bit. In a structured light system, more than 8-bit code is commonly used, especially when a high resolution camera (e.g. more than 1024×768 pixels) is utilized. Here, we propose a method to expand the Antipodal Gray code. First, expand t times (t is a multiple of 2) the 8-bit antipodal Gray code with its reverse order code recursively, then add the additional $\log_2 t$ frame with original Gray code or antipodal Gray code. Fig. 10 gives the example of the expansion of the antipodal Gray code into 9-bit code. In the case of 10-bit, we can expand the code with 2-bit (antipodal) Gray code and four 8-bit Antipodal Gray code. This is the simplest way to expand the antipodal Gray code. Though the property whereby the binary complement of any code-string appears exactly n steps away in the list is broken, the Gray code property whereby the successive strings differ only in one bit position is preserved.

D. Drawbacks

When the objects are located outside of the depth-of-field of the projector, the pattern becomes blurry. In this case, the code-strings are affected not only by the next strings but also by the nearby strings. When the code-strings are mixed with the strings located some pixels away in antipodal Gray code, the decoding error tends to increase to a greater extent than in the original Gray code. In the case of Gray code, though the code-strings are mixed with the neighbor strings, the decoding error is relatively small, while in the case of antipodal Gray code, the decoding error is a kind of random; it can be big or small. That is, antipodal Gray codes work well under well-focused environments; otherwise, they can provide poor results.

V. Sub-pixel Decoding

The following sub-pixel decoding method can be applied for more precise decoding. Here, we describe the method when the pattern stripe is vertical. Even in case the pattern stripe is horizontal, we can apply the following method in a similar way.

A. Initial Labeling

Sub-pixel location technology is widely used in structured light. In this paper, we used the original and its inverse patterns for binarization. We first computed the difference $f_i(c,r)$ between the original $I_i^{original}(c,r)$ and its inverse $I_i^{inverse}(c,r)$ patterns. That is,

$$f_i(c,r) = I_i^{original}(c,r) - I_i^{inverse}(c,r) \qquad (1)$$

where the subscript i denotes the frame. Then, the convoluted value $g_i(c,r)$ is computed with three different masks, 1×1, 3×1, or 5×1 (first derivative filters):

$$g_i(c,r) = f_i(c,r) \qquad (2)$$

or

$$g_i(c,r) = f_i(c\text{-}1,r) + f_i(c+1,r) \qquad (3)$$

or

$$g_i(c,r) = f_i(c\text{-}2,r) + f_i(c\text{-}1,r) + f_i(c+1,r) + f_i(c+2,r). \qquad (4)$$

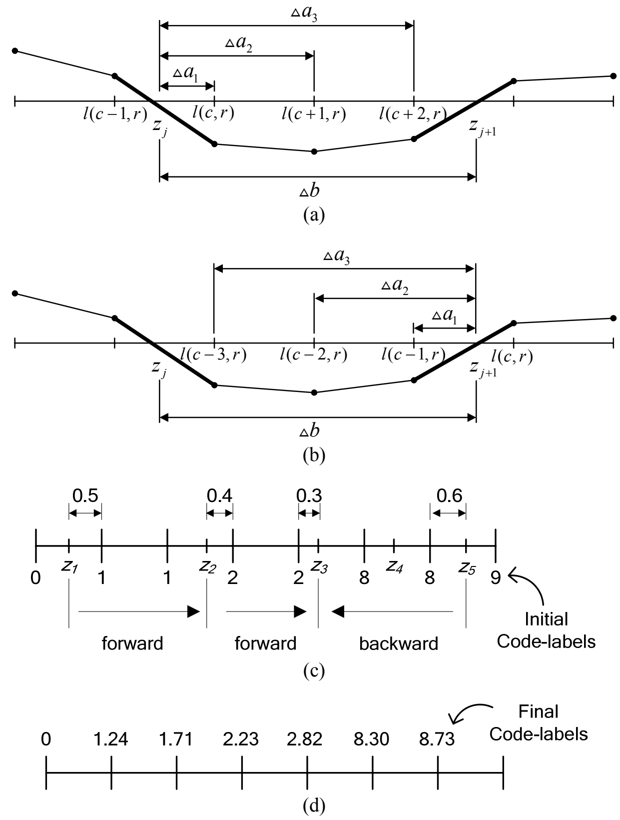


Fig. 11. Forward and Backward Code Labeling. The pattern is labeled according to the zero-crossing points' position.

If the sign of $g_i(c,r)$ is negative, binarize $g_i(c,r)$ as 0, otherwise as 1; then decode with binary numbers and label it as $l(c,r)$. In addition, if the sign of $g_i(c-1,r)$ and $g_i(c-1,r)$ change, compute the zero-crossing point z_j.

B. Forward Code Labeling

We only trust the initially decoded code-label $l(c,r)$, whose label is the same or one bigger than $l(c-1,r)$. That is,

$$0 \le l(c,r) - l(c-1,r) \le 1 \qquad (5)$$

In addition, we trust the code-label $l(c,r)$ next to a "shadow" which does not have any code-label due to light limitation, occlusion, etc. Since we project a coded pattern which has the hamming distance 1 between the code-strings, we can say that when the difference between $l(c,r)$ and $l(c-1,r)$ is 0 or 1, these are correctly decoded, and we can assume that the surface is continuous from $l(c-1,r)$ to $l(c,r)$.

If the zero-crossing point z_j is positioned between the trusted code-labels $l(c-1,r)$ and $l(c,r)$, we sub-decode the pixels which are located between z_j and z_{j+1} as shown in Fig. 11 (a). In order to avoid confusion, let $l_{initial}(c,r)$ be the initially decoded labels and let $l_{sub}(c,r)$ be the sub-decoded value. Let the zero-crossing point z_j be the label $l_{initial}(c,r)$, and then sub-decode and update with the following equation:

$$l_{sub}(c-1+n,r) = l_{initial}(c,r) + \frac{\Delta a_n}{\Delta b}, \qquad (6)$$

where Δb is the distance $|z_{j+1} - z_j|$ in pixels, and Δa_n is the distance between the pixel positions and z_j as shown in Fig. 11 (a).

C. Backward Codex Labeling

When $l(c, r) - l(c-1, r)$ is bigger than 1 or less than 0, we can assume that the surface's depth changes dramatically, and we do not trust the labels $l(c, r)$ and $l(c-1, r)$ since the error occurs frequently in this area.

If the zero-crossing point z_j is positioned between the non-trusted indices, we do not carry out forward decoding. However, if the zero-crossing point z_{j+1} is positioned between the trusted labels, we carry out backward decoding from z_{j+1} to z_j as shown in Fig. 11 (b). Let the zero-crossing point z_{j+1} have the label $l_{initial}(c, r)$, and then sub-decode and update with following equation:

$$l_{sub}(c-n, r) = l_{initial}(c, r) - \frac{\Delta a_n}{\Delta b}, \quad (7)$$

D. Example

Fig. 11 (c), (d) show examples of forward and backward code labeling. Fig. 11 (c) is the result of the initial decoding and Fig. 11 (d) is the final decoding which is carried out under the proposed decoding methods.

VI. Experimental Results

A. Experimental Setting

For our experiments, a Canon projector and a PGR flea2 IEEE 1394 digital camera were used. The resolution of the projector was 1024×768 and the camera was 640×480. The position of the camera was about 20cm above the projector. 512 code-strings (in a horizontal direction) and 384 code-strings (in a vertical direction) were used for projection. The distance between the system and the objects was about 1m. The original Gray code (GC) and antipodal Gray code (APGC) were implemented and evaluated.

B. Results

We projected the coded-patterns 15 times to the arbitrary located calibration plane (checkerboard plane) with 9-bit codes (512 code-strings) and their inverse codes, and decoded with the proposed methods in section V. Then, we calibrated the structured light system using OpenCV, and reconstructed again. The error was defined as the average distance between the original checkerboard corner position and the reconstructed position. In table 1-3, we summarize the results carried out with three different masks.

TABLE 1
ERROR IN X COORDINATE

	1×1	3×1	5×1
APGC	4.88	2.69	1.96
GC	5.35	2.94	6.38

(unit: mm)

TABLE 2
ERROR IN Y COORDINATE

	1×1	3×1	5×1
APGC	2.71	1.53	1.09
GC	2.96	1.65	3.45

(unit: mm)

TABLE 3
ERROR IN Z COORDINATE

	1×1	3×1	5×1
APGC	13.85	7.64	5.53
GC	15.17	8.34	17.87

(unit: mm)

VII. Discussion

As shown in table 1-3, in the case of antipodal Gray code, the accuracy increased when the mask size was increased from 1×1 to 5×1; however, in case of Gray code, the accuracy was best when the mask was 3×1. This is because the stripe width can often be smaller than the filter size in Gray code, so the first derivative may not be computed correctly. On the other hand, since most stripe widths in antipodal Gray code were broader than the filter size, the convolution results was better. In practice, the relationship between the accuracy and the mask size depends on the camera resolution and the number of code-strings, etc. From the results, we can infer that the antipodal Gray code is more robust and accurate than the original Gray code.

VIII. Conclusion and Future Works

In this paper, we applied the antipodal Gray code to the structured light system, and evaluated the codes. We proposed the criteria for the robust codes and showed that the antipodal Gray code can be one of the proper codes suited for structured light.

After submitting this paper, we found another paper related to the antipodal Gray code by Chang et al. [7], who established other various antipodal Gray codes. In our future work, we will compare and evaluate these antipodal Gray code variations in the structured light system.

References

[1] J. Salvi, J. Pagès, and J. Batlle, "Pattern codification strategies in structured light systems," *Pattern Recognition*, vol. 37, no. 4, pp. 827-849, 2004.

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

[3] J. Gühring, "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.

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

[5] H. Gärtner, P. Lehle, and H. J. Tiziani, "New, highly efficient, binary codes for structured light methods," *Proc. of SPIE*, vol. 2599, pp. 4-13, 1996.

[6] C. E. Killian and C. D. Savage, "Antipodal Gray codes," *Discrete Mathematics*, vol. 281, no. 1-3, pp. 221-236, 2002.

[7] G. J. Chang, S.-P. Eu, and C.-H. Yeh, "On the (n,t)-antipodal Gray codes," *Theoretical Computer Science*, vol. 374, no. 1-3, pp. 82-90, 2007.