MVA2011 IAPR Conference on Machine Vision Applications, June 13-15, 2011, Nara, JAPAN

Slant-invariant stagger grid pattern projection with error-tolerant decoding technique for one-shot shape measurement

Tetsuri Sonoda, Shinichiro Uno, Kenji Saito and Shinji Uchiyama
Canon Inc.
30-2, Shimomaruko 3-chome, Ohta-ku, Tokyo 146-8501, Japan
{sonoda.tetsuri, uno.shinichiro, saito.kenji, uchiyama.shinji}@canon.co.jp

Abstract

We propose a pattern projection method for one-shot active 3D shape measurement to reconstruct movable or deformable scenes. In our method, just one projection pattern which has scene slant-invariant features is used to solve the correspondence between camera and projector pixel coordinates. Additionally, in order to correct correspondence in complex scenes which include discontinuous and overlapping surfaces, we propose an error-tolerant decoding technique with code error detection and correction. Experiment results show the robustness of our method for actual scenes.

1. Introduction

Real−time 3D measurement of various shapes is very useful and promising especially in industries where the shape of various parts needs to be measured.

There are some real-time active shape measurement methods which can measure moving or deforming scenes at high-speed by using just one projection pattern and captured image without time domain scanning. In an active measurement method using a camera and a projector, it is necessary to compute the correspondence between the camera and the projector pixel coordinates. Such one-shot active measurement methods project a pattern which includes many features as cues for correspondence. To use these features for computing correspondence, the features are detected in the captured image and near features on same surfaces in the scene are grouped. To compute correspondence using such features, there are two kinds of approaches.

In one kind of approach which use as many features as possible for computing correspondence [1][2], it is implicitly assumed that scenes consist almost entirely of large and continuous surfaces for grouping many features. When measuring a scene that consists of discontinuous and overlapping surfaces, such approaches may group incorrect features that are not on the same surface, resulting in large measurement errors because of incorrect correspondence. Correcting incorrect correspondence incurs heavy computational load such as belief propagation [3].

By contrast, the other approaches, which use as few features as possible for computing correspondence [4][5], can compute the correspondence for a minimum surface area in the scene for grouping the smallest number of features. Additionally, incorrect correspondences caused by grouping incorrect features affect only minimum surface areas. However, if detected features include errors because of the scene shape, texture or captured image noise, it easily cause incorrect correspondence.

We adopt the latter approach for correspondence with minimum surface areas and we propose a “stagger grid” projection pattern to which scene slant-invariant features are appended in order to reduce feature detection failure. Moreover, for reducing the above incorrect correspondence, we propose a decoding method which can detect and correct code errors caused by feature detection failure and grouping incorrect features.

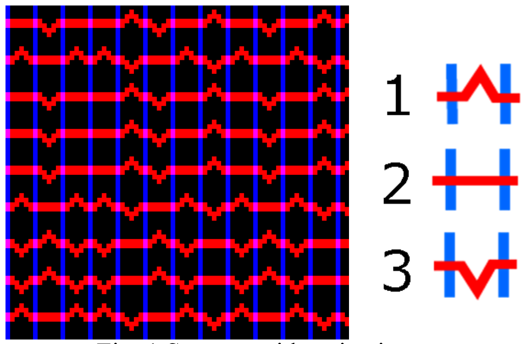


Fig. 1 Stagger grid projection pattern

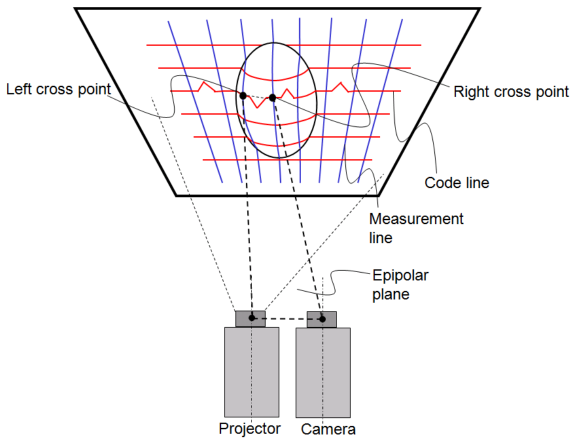


Fig. 2 Camera-Projector configuration

2. Stagger grid projection pattern

First, we describe our stagger grid projection pattern in detail. For one-shot shape measurement, a projection pattern must include enough features for computing correspondence in a robust manner. In order to achieve this, Color, brightness, and geometrical features (combinations thereof) can be used. Zhang et al. proposed a color transition featured pattern [6], and Vuylsteke et al. proposed a black/white binary featured pattern [5]. However these methods have difficulty with measuring scenes that include strongly colored or textured surfaces because of using color or brightness features.

Sagawa et. al [1] and Ulusoy et. al [2][3] proposed methods in which a grid pattern consisting of vertical and horizontal lines is projected, the trajectories of these cross-points on each epipolar line are used as geometric features for correspondence. These methods are robust against strong colors and textures in the scene, but the ability to detect features depends on the calibration quality of camera-projector and aberration of optical systems.

To solve these difficulties, we propose the "stagger grid" projection pattern, as shown in Fig. 1 with the configuration shown in Fig. 2. The stagger grid projection pattern consists of blue colored measurement lines which are almost orthogonal to the camera-projector epipolar plane and red colored code lines which are almost parallel to the same epipolar plane. Each measurement line is used for computing depth by the light-section method. The code lines are used as cues for computing the correspondence of each measurement line in the captured image. Each code line can be divided into one feature by the cross bar of two neighboring measurement lines. Each feature varies 1) upper, 2) none, or 3) lower "staggered", relative to the cross bars on both sides, and can thus render three code values. This feature utilizes the property that orthogonal direction displacement of the epipolar plane is conserved in the captured image, and is independent of surface slants in the scene. Hence this feature is robust and easy detectable almost regardless of the various shapes, colors and textures in the scene. In addition, with the stagger grid pattern high density measurement lines can easily be made since the only constant thickness narrow lines are used for measurement and feature.

3. Error-tolerant coding and decoding

Next, we describe the coding and decoding method for correspondence. We define each intersection of code lines with measurement lines as one code which is rendered by the code line on the right of the cross bar. These cross bars are the minimum units for correspondence. By tracing the four adjacent (up, down, left and right) cross bars and gathering their code values, the data code-array is acquired.

Here, the 2D code-array which is applied across the whole projection pattern is called a projection code-array, so the above data code-array is a sub array of the projection code-array. Correspondence can be done if data code-array's 2D position is fixed in projection code-array by matching between them.

When the method can acquire a data code-array with correct adjacencies of cross bars, a longer length of data code-array makes it more robust against incorrect code by failure of feature detection. Therefore, some methods were developed, which use a maximum number of features on the same surface in the scene for matching [1][2][3]. However, a number of features are required for correspondence, so the method can only be used for scenes which have large area surfaces. Furthermore, if the projection pattern has spurious connections at the border between discontinuous and overlapping surfaces in the captured image, the data code-array is acquired with incorrect adjacencies of cross bars, causing seriously incorrect correspondence and a very high computing cost for correction [3].

Griffin's coding method

Therefore, we need a coding method which can compute correspondence with the smallest data code-array. As an optimal code, we apply Griffin's coding method [5] for the projection code-array, as described below.

Letting the number of values that can be rendered by a feature be denoted by k, and the horizontal × vertical size of a data code-array be denoted by $m \times n$, then the first row of the projection code-array f_{i1} is given by

$$f_{i1} = V_{HMi} \ldots (i = 1 \ldots k^m) \qquad (1)$$

V_{HM} denotes a de Bruijn sequence which has k values and the unique minimum subsequence length is m. After the first column, the projection code-array f_{ij} is described by:

$$f_{ij} = 1 + \left(\left(f_{ij-1} + V_{VNj}\right) \bmod k\right) \qquad (i = 1 \ldots k^m,\ j = 2 \ldots k^{n-1}) \quad (2)$$

V_{VN} denotes a de Bruijn sequence which has k values and the unique minimum subsequence length is $n\text{-}1$. The $k^m \times k^{n-1}$ projection code-array which is generated in this manner has the following properties. For example, in the case of $m \times n = 4 \times 3$, at an arbitrary w_{ij} as the origin position, the data code-array (w_{ij}, w_{ij-1}, w_{ij+1}, w_{i-1j}, w_{i+1j}, w_{i+2j}) (shown at Fig. 3) is always unique in the whole $k^m \times k^{n-1} = 81 \times 9$ projection code-array. Hence, in theory, when six length cross-figured data code-array is acquired on a surface in the scene, the data code-arrays position can be fixed in the projection code-array by matching.

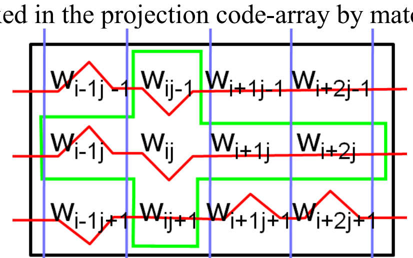


Fig. 3 Data code-array and redundant code-array

Code error detection and correction

The above coding method enables us to compute correspondence with the shortest length of code-array. However, failure to detect just a single feature in the data code-array easily causes incorrect correspondence unless any code error is detected or corrected. To reduce such incorrect correspondence, we developed an error-tolerant decoding technique with code error detection and correction which uses additional codes around the data code-array as described below.

First, we assume that there is a surface in the scene with an area larger than the minimum size of data code-array. On that surface, a "redundant" code-array in addition to a data code-array can be acquired in an $m \times n$ square matrix. For example, in the case of $m \times n = 4 \times 3$, at arbitrary w_{ij} as the origin position, a redundant code-array (w_{i-1j-1}, w_{i+1j-1}, w_{i+2j-1}, w_{i-1j+1}, w_{i+1j+1}, w_{i+2j+1}) is acquired with the data code-array (Fig. 3). By definition of eq. (2), the vertical differences of the code-array in the $m \times n$ square matrix are all the same for each column. We use this constraint for code error detection and correction. Computing mod

k differences of each $m\times n$ matrix row, hash value H_{ij} is computed as:

$$H_{ij} = (w_{ij} - w_{ij+1} + k) \bmod k \quad ....(i = 1....m, j = 1...n-1) \qquad (3)$$

Verifying H_{ij} between each column i=1…m, error code column position $i_{incorrect}$ can be detected if H_{ij}=H_{ij+1} or not in the case of $n\geqq 2$. Additionally, in the case of $m\geqq 3$ and $n\geqq 3$, if uneven H_{ij} exists only in one column $i_{incorrect}$, the correct hash array $H_{correctj}$ and incorrect hash array $H_{incorrectj}$ can be determined by voting. And error code row position $j_{incorrect}$ is fixed by using $H_{correctj}$ and $H_{incorrectj}$ as follows:

$$\begin{cases} j_{incorrect} = (j : H_{correctj} \neq H_{inccorectj}) & (j=1) \\ j_{incorrect} = (j : H_{correctj-1} \neq H_{incorrectj-1} \cap H_{correctj} \neq H_{incorrectj}) & (j = 2...n-2) \\ j_{incorrect} = (j : H_{correctj-1} \neq H_{incorrectj-1}) & (j = n-1) \end{cases} \qquad (4)$$

Then, using $H_{correctj}$ and code w_j around $j_{incorrect}$ on $i_{incorrect}$, error the code is corrected to $w_{correctj}$:

$$\begin{cases} w_{correctj} = (H_{correctj} + w_{j+1} + k) \bmod k ....(j=1) \\ w_{correctj} = (w_{j-1} - H_{correctj-1} + k) \bmod k \\ \qquad = (H_{correctj} + w_{j+1} + k) \bmod k ....(j = 2...n-2) \\ w_{correctj} = (w_{j-1} - H_{correctj-1} + k) \bmod k ...(j = n-1) \end{cases} \qquad (5)$$

As described above, code error in the data code-array can be detected and corrected with a very simple computation. After error correction, better matching can be done using six length cross-figured data code-array, the same as in the Griffin's coding method. In the case for an insufficient redundant code-array for error detection and correction being acquired, only Griffin's coding method is applied.

4. Evaluation

Prototype system

This section describes a prototype system to prove the effectiveness of our method. In software implementation, for separation of measurement and code lines, we simply use a color channel. And for extraction of each line, we use a Sobel filter for finding zero-cross points as center of line width, and assign the same label for all points on the same line. Whole software for the prototype is implemented in C++. The projection pattern has 80 code lines at 8 pixel intervals and 79 measurement lines at 6 pixel intervals, suitable for the projector's resolution (640×480). We set a data code-array horizontal size of $m=4$ corresponding to 79 measurement lines and a vertical size $n=3$ for error detection and correction. The projection code-array horizontal size is set to 78 out of V_{HM} ($m=4$, sequence length: $3^4=81$) and the vertical size is set to 80 repeating V_{VN} ($n=3$, sequence length: $3^{3-1}=9$) in eq. (6). In our method, horizontal code line repeating correspondences are also determined with vertical measurement line correspondences, but they are not used in this evaluation.

V_{HM} = {111121113112211231132113312121312221 223123212331313221323133213332222322332323333}
V_{VN} = {312113223} (6)

We used an xw8400 HP workstation (Windows XP, Intel Xeon X5355 2.66GHz CPU, 3GB memory) for the computation, and constructed measurement hardware consisting of a camera (PGR GRAS-50S5C, resolution: 1024×1024 cropping) with a Fujinon 9mm lens (exposure: F8) and a projector (3M MPro120, resolution: 640×480) (Fig. 5). The camera-projector baseline is set to about 50 mm.

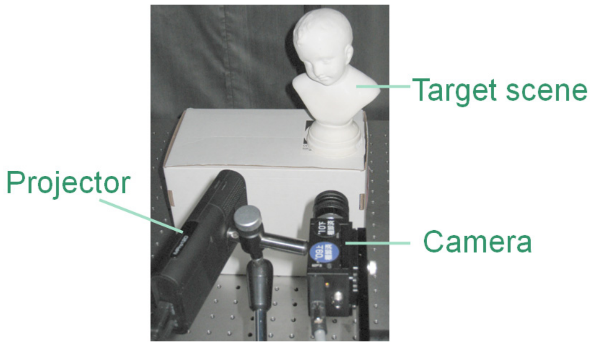


Fig. 5 Measurement system and target scene

Performance

We describe experimental details to compare the correspondence abilities between Griffin's original coding method and our error detection and correction method with our stagger grid projection pattern. We chose two target scenes for the experiment: one is a plaster figure of a small boy consisting of large continuous surfaces, and the other is a pile of plastic parts consisting of discontinuous and overlapping surfaces. We set the experiment conditions to a measurement range of around 200 mm, measurement area of 50×50 mm and shutter speed of 83.4 ms.

For comparison, we obtained reference correspondence results by projecting each measurement line and capturing them one by one. The correspondence results are obtained as a group of point arrays for each corresponded lines. Using these reference correspondence results, we define performance index values **A**, **B** and **C** as below.

A: The ratio of correct corresponding number of point array to reference result
B: The ratio of failure corresponding number of point array to reference result
C: The ratio of incorrect corresponding number of point array to reference result

A+**B**+**C**=100%. Higher **A** indicates a higher performance of correspondence. Higher **B** shows a lower performance of correspondence because it means that larger no corresponded area exists in the scene. And higher **C** shows the most adverse result for shape measurement because it causes serious measurement errors.

The measurement was made 20 times. Measurement results are shown as the ratio of average and distribution in Fig. 6 and Fig. 7 (Index **B** and **C** are shown as negative values), and the captured image and reconstruction result are shown in Fig. 8(a)(b) and Fig. 9(a)(b). In this experiment, the shape measurement error is about 50 µm RMS in the plane of the plastic parts and average computing time is around 200 ms.

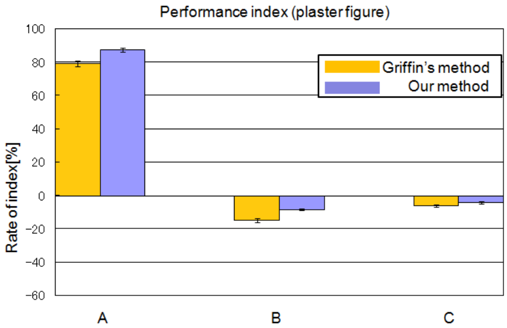


Fig. 6 Performance index of plaster figure

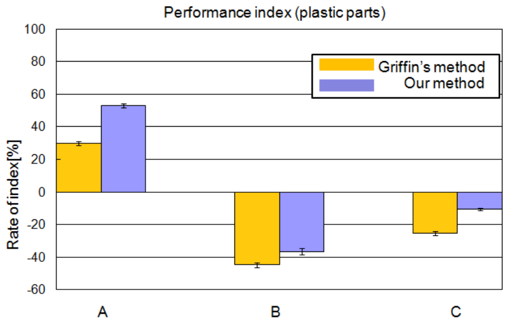
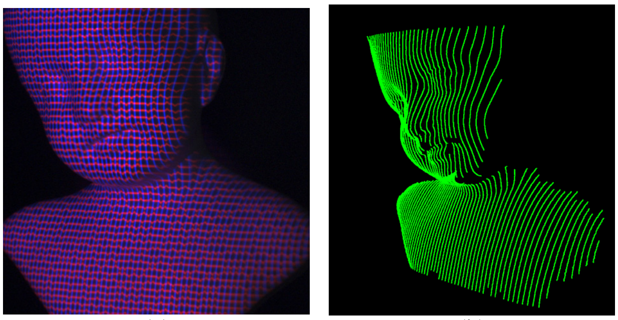
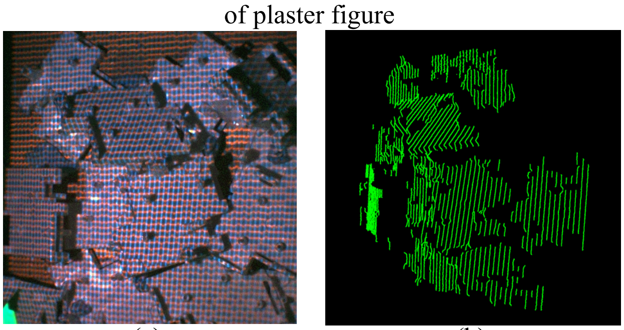


Fig. 7 Performance index of plastic parts



(a) (b)

Fig. 8 (a) Captured image and (b) reconstruction result of plaster figure



(a) (b)

Fig. 9 (a) Captured image and (b) reconstruction result of plastic parts

5. Discussion and Conclusions

The captured images in Fig. 8 (a) and Fig. 9 (a) show that the stagger features indicated by red code lines are almost invariant, independent of surface slants and small asperities in the scene. Thus, the stagger grid projection pattern is feasible for scene areas which have sufficient smoothness for rendering code as detectable lines.

For the plaster figure, even Griffin's coding method shows good correspondence (Fig. 6 and Fig. 8(a)(b)). Although our method shows less failure/incorrect correspondence **B** and **C**, the differences are not notable (Fig. 6). Because this scene has relatively large and smooth surfaces that are not overlapping, the projection pattern can be captured almost without spurious connections. In such a simple scene, good measurement results can be obtained merely by using our stagger grid projection pattern without code error handling.

On the other hand, the results for the plastic parts show a significant difference between each decoding method (Fig. 7 and Fig. 9(a)(b)). This scene has local sharp bumps on the surface and each surface is discontinuous and overlapping. With such overlapping discontinuous surfaces, the projection pattern tends to have a number of spurious connections in the captured image [3]. In such scenes, methods that use a maximum number of features for correspondence [1][2] cannot work correctly without detecting spurious connections [3]. Different from those methods, Griffin's coding method is not broken by using the smallest number of features. However, failure correspondence **B** and incorrect correspondence **C** increased significantly because of code errors caused by sharp bumps and overlapping surfaces. Our code error detection and correction technique is very effective for such complex scenes.

As described above, we show that our one-shot robust projection pattern and error-tolerant decoding technique can be used to measure complex scenes. Furthermore, our error detection and correction technique is very simple and has low computational costs.

The limitation of our method is that a minimum surface area for computing correspondence is required in the scene, the same as for Griffin's coding method. To overcome this limitation, we plan to append time domain varying features in our stagger projection pattern and track them. For real-time measurement, we also plan to apply parallel image processing by using GPGPU.

Another limitation is that projection pattern uses two colors for separation between code and measurement lines. To measure strong color or textured object, we will improve this pattern to monochrome.

References

[1] R. Sagawa, Y. Ota, Y. Yagi, R. Furukawa, N. Asada, and H. Kawasaki, "Dense 3D reconstruction method using a single pattern for fast moving object," in Proc. of 2009 IEEE 12th ICCV, pp.1779-1786, Kyoto, Sep. 2009

[2] A. O. Ulusoy, F. Calakli, and G. Taubin. "One-shot Scanning using De Bruijn Spaced Grids," in Proc. of 3DIM'09, 2009.

[3] A. O. Ulusoy, F. Calakli, and G. Taubin. "Robust One-Shot 3D Scanning Using Loopy Belief Propagation", in Proc. of ACVA 2010, 2010

[4] P. Vuylsteke and A. Oosterlinck, "Range Image Acquisition with a Single Binary-Encoded Light Pattern," IEEE PAMI, Vol.12, No.2, pp.148-164, 1990

[5] P. M. Griffin, L.S. Narashimhan and S.R. Yee, "Generation of uniquely encoded light patterns for range data acquisition," Pattern Recognition, Vol.25, Issue 6, pp.609-616, 1992

[6] L. Zhang, B. Curless and S.M. Seitz, "Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming," in Proc. of IEEE 3DPVT, pp. 24-36, 200