Hindawi
Journal of Control Science and Engineering
Volume 2021, Article ID 6676704, 16 pages
https://doi.org/10.1155/2021/6676704

Research Article

One-Shot M-Array Pattern Based on Coded Structured Light for Three-Dimensional Object Reconstruction

Xiaojun Jia and Zihao Liu

College of Information Science and Engineering, Jiaxing University, Jiaxing, Zhejiang 314001, China

Correspondence should be addressed to Xiaojun Jia; xjjiad@sina.com

Received 14 December 2020; Revised 4 May 2021; Accepted 20 May 2021; Published 3 June 2021

Academic Editor: Radek Matušů

Pattern encoding and decoding are two challenging problems in a three-dimensional (3D) reconstruction system using coded structured light (CSL). In this paper, a one-shot pattern is designed as an M-array with eight embedded geometric shapes, in which each 2×2 subwindow appears only once. A robust pattern decoding method for reconstructing objects from a one-shot pattern is then proposed. The decoding approach relies on the robust pattern element tracking algorithm (PETA) and generic features of pattern elements to segment and cluster the projected structured light pattern from a single captured image. A deep convolution neural network (DCNN) and chain sequence features are used to accurately classify pattern elements and key points (KPs), respectively. Meanwhile, a training dataset is established, which contains many pattern elements with various blur levels and distortions. Experimental results show that the proposed approach can be used to reconstruct 3D objects.

1. Introduction

Regarding three-dimensional (3D) object reconstruction techniques, structured light is considered one of the most reliable techniques in stereovision with two or more cameras. When using this technique, one of the stereovision cameras is replaced by a light source that is used to project one or more patterns comprised of points, lines, or complex structure pattern elements into the field of view [1, 2]. The position of the pattern can be retrieved from the captured image with a camera by fathering local information around this encoded point. Since the light pattern is designed with a set of encoded points, this technique is also known as coded structured light (CSL) [2]. CSL has been widely used in many fields, such as 3D reconstruction, industrial inspection, object recognition, reverse engineering, biometrics, and others [1–5].

A proper encoding pattern and decoding method for a CSL system play decisive roles in detection accuracy for complex objects. Salvi et al. [2] and Jeught and Dirckx [6] presented extensive explanations regarding previous approaches of CSL. Among them, one-shot techniques [2, 6, 7], where a unique pattern is projected, are considered suitable for dynamic environments. One group of one-shot techniques relies on encoding strategies using De Bruijn or pseudorandom sequences with color multi-slit or stripe patterns [7, 8]. The technique of accurately locating color multi-slit or stripe patterns could provide better results because the image segmentation step is easier. However, because several colors were used, these color stripes were typically sensitive to object albedo or texture. The other group of one-shot techniques using M-arrays (perfect maps) or pseudorandom array patterns was found to be robust against occlusions (up to a certain limit), had unique subwindow characteristics (or window property, which denotes that the subwindow appears only once in the array or pattern) of the array, and was suited to dynamic scenes with a monochrome encoded pattern. In this paper, we focus on one-shot techniques based on array patterns. M-arrays, first presented by Etzion [9], are $n_1 \times n_2$ pseudorandom arrays in which a $k_1 \times k_2$ submatrix ($k_1 \leq n_1$ and $k_2 \leq n_2$) appears only once in the pattern. M-arrays were constructed theoretically with dimensions of $n_1 \times n_2 \leq q^{k_1 k_2}$. In practice, zero submatrices are not considered, and the maximum number of

the matrices is $q^{k_1 k_2} - 1$ given by MacWilliams and Sloane [10].

Choosing an appropriate window property will determine the robustness of the pattern against pattern occlusions and object shadows for a given application. Lu et al. [11] presented a large M-array using pseudorandom numbers to generate the pattern. Color points were replaced with geometric symbols. Because generating adequate code words with binary modulation is difficult, Morano et al. [12] proposed a color pattern based on pseudorandom codes. The use of colors reduced the size of the windows. Vandenhouten et al. [13] focused on the design and evaluation of a subset of symmetric isolated binary toroidal perfect submaps for structured light patterns, and several valuable images related to the practical application of perfect submaps in a 3D sensor were defined. A 20×20 M-array and window property of 3×3 was designed by Pagès et al. [14], based on an alphabet of three symbols. A window property of 3×3 with three different symbols (black circle, circumference, and stripe) was used to represent the codeword. Jia et al. [15, 16] presented an M-array pattern with ten special symbol elements and a 2×2 window property, which had many turning points and intersections that were used for detection. Fang et al. [17] proposed using a symbol density spectrum to choose ten pattern elements for improving resolution and decreasing decoding error. The pattern elements were classified by recognizing the feature of the connected components with eight connected region. Li et al. [18] presented a high-accuracy and high-speed structured light 3D imaging method developed for optical applications. They introduced the digital fringe projection (DFP) method to the intelligent robotics community. Huang et al. [19] proposed a CSL method using a spatially distributed polarization state of the illuminating patterns with the advantage of enhancing target in 3D reconstruction. To improve measurement accuracy, some researchers have proposed using deep learning in 3D reconstruction methods. Tang et al. [20] designed a grid pattern with embedded geometric shapes and proposed a pattern decoding method. Pattern elements were accurately classified using a deep neural network. Eigen and Fergus [21] proposed a classical deep learning method for acquiring depth data using VGG-16. This method requires a massive dataset for training, and the size of the dataset limits its application range. Garg et al. [22] proposed an unsupervised convolutional neural network (CNN) for estimating depth data according to a single image. This method was convenient for training and satisfactory performance of reconstruction was obtained on less than half of the KITTI dataset. Li et al. [23] proposed a method for combining structured light and unsupervised CNN networks in stereo matching to calculate depth.

Based on the aforementioned studies, many researchers have sought ideal methods for 3D reconstruction with CSL. However, direct use of these methods to reconstruct 3D objects in a CSL system, as proposed herein, faces three issues:

(1) Color stripes or grids with location information are preferred encoding patterns, which fail in bright environments.

(2) Geometric shapes or adjacent images with obvious features are taken as encoding patterns. When these patterns are projected onto scenes with rich colors or complicated textures, decoding is performed using traditional image processing, such as image segmentation, feature extraction, and simple pattern matching, which will reduce decoding accuracy and feature positioning accuracy.

(3) At present, most CSL decoding methods use simple image segmentation and template matching algorithms. However, due to the complexity and uncertainty of the target surface, including color, texture, deformation, reflection, and discontinuities, the pattern elements in the image are unclear, and the elements cannot be accurately determined when they change dramatically, which makes decoding difficult.

In this study, we designed a structured light pattern using an M-array with a 2×2 window property and eight geometric elements. A decoding method is proposed to process the distorted pattern image obtained by the camera. A deep convolutional neural network (DCNN) is used to accurately classify the pattern elements. A training dataset containing various fuzzy and distorted pattern elements is also compiled. The chain angle method was used to determine the detection point information. Finally, 3D reconstruction can be achieved using an established detection system. The remainder of this article is arranged as follows. The framework for this approach, including encoding, capturing image, decoding, system calibration, and 3D reconstruction, is described in Section 2. Experimental results are presented in Section 3, and conclusions are presented in Section 4.

2. The Proposed Method

The proposed method of one-shot 3D reconstruction involves encoding, image capture, decoding, system calibration, and 3D reconstruction, shown in the flowchart in Figure 1. First, a one-shot pattern based on an M-array is designed. Second, before obtaining the image, the structured light detection system must be established in advance. Third, decoding is implemented using the proposed pattern element tracking algorithm (PETA), and the independent elements in the pattern can be separated. The training dataset with various fuzzy and distorted pattern elements was compiled and a DCNN was applied to classify the pattern elements. Fourth, 3D calibration data is used to estimate system parameters. Finally, the point cloud is transformed into a 3D shape using bilinear interpolation.

2.1. One-Shot CSL Pattern. An improved encoding scheme is proposed in this paper. This scheme follows the pattern generation method described in an earlier study by our

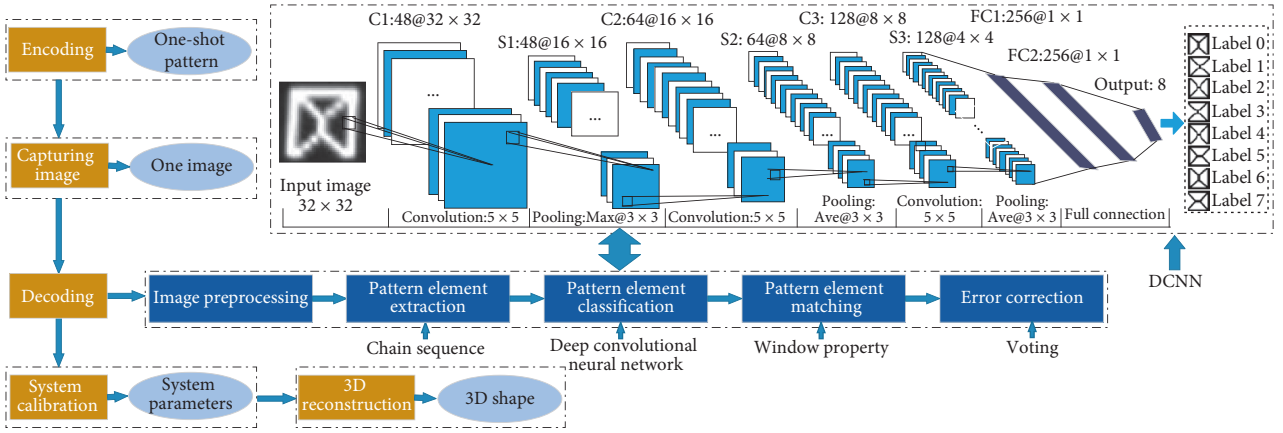


Figure 1: Flowchart showing one-shot 3D reconstruction with CSL.

research group [15, 16] and can be used to obtain a M-array with a 2×2 window size. As shown in Figure 2, eight special geometrical shapes were used as array elements. A pattern constructed from eight elements is shown in Figure 3, where black is selected as the background. The four corners of the shape and the internal intersection points are used as key detection points. Compared with Jia et al. [15, 16], the current encoding pattern has been improved in two aspects. First, the number of elements is reduced to eight, and each element has its own detection point, which reduces the recognition error rate and increases the detection accuracy. Second, the size of the encoding pattern is reduced from 79×59 to 39×29, which improves the decoding speed of the whole pattern.

2.2. The Imaging System. The mathematical model of a CSL measurement system is derived from the pin-hole model. We used the CSL imaging system shown in Figure 4, which was proposed by our research group [15, 16]. Spotlight S_0 at the projection side and camera D_0 at the receiving side are both in a flat plane, and the baseline distance between them is S. Note that $GXYZ$ is a right-handed coordinate system where the Z axis is perpendicular to the GXY plane and points inwards, and G is the center of the line S_0D_0. S_0 is located at $(S/2, 0, 0)$ and D_0 is located at $(-S/2, 0, 0)$. On the spotlight side, P is the geometric center of the pattern plane (projecting plane) and f_p is the length of line PS_0. The angle between PS_0 and the X axis is θ_S, and PS_0 is perpendicular to the pattern plane. On the camera side, D is the geometric center of the image plane, and f_c is the length of the line DD_0. The angle between DD_0 and the X axis is θ_D, and DD_0 is perpendicular to the image plane. $T(x_w, y_w, z_w)$ is an arbitrary 3D point on the object in the scene. Figure 4(a) is projected onto the GXZ plane; the camera side and projection side are shown in Figures 4(b) and 4(c), respectively. On the camera side, as shown in Figure 4(b), $DH_D$is perpendicular to the X axis, and H_D is the endpoint. The X' axis is parallel to the X axis and passes through D. RH_R is vertical to the X' axis, and H_R is the endpoint. The angle between the line RD_0 and the X axis is α_D. On the projection side, as shown in Figure 4(c), PH_S is perpendicular to the X axis, and H_S is the endpoint. The X' axis is parallel to the X axis and passes through P. EH_E is perpendicular to the X' axis, and H_E is the endpoint. The angle between ES_0 and the X axis is α_S.

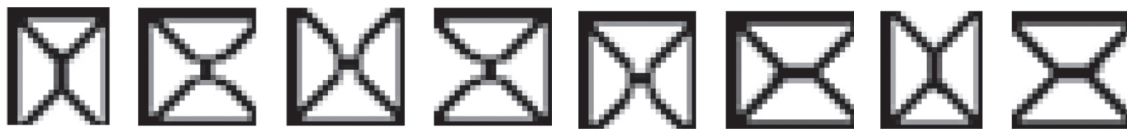
In Figure 4(a), it is assumed that the line TS_0 and the pattern plane intersect at point E, and its coordinates are (x_{pu}, y_{pu}) (mm). The line TD_0 and the plane D intersect at point R, and its coordinates are (x_{cu}, y_{cu}) (mm), where $x_{cu} = (u^c - u_0^c) \times dx$, $y_{cu} = (v^c - v_0^c) \times dy$. (u^c, v^c) (mm) denotes the coordinate in the image coordinate system, and (u_0^c, v_0^c) (pixels) denotes the center of the image. dx (mm) and dy (mm) are the physical pixel sizes in the x and y directions, respectively. Based on the triangle principle, 3D information of T can be calculated as follows:

$$\begin{cases} x_w = S\dfrac{\sin(\alpha_S - \alpha_D)}{2\sin(\alpha_S + \alpha_D)}, \\ y_w = S\dfrac{y_{pu}\sin\alpha_S \sin\alpha_D}{\sin(\alpha_D + \alpha_S)\left(f_p \sin\theta_S + x_{pu}\cos\theta_S\right)}, \\ z_w = S\dfrac{\sin\alpha_S \sin\alpha_D}{\sin(\alpha_S + \alpha_D)}, \end{cases} \tag{1}$$

where α_S and α_D are

$$\begin{cases} \alpha_S = \theta_S + \arctan\left(\dfrac{x_{pu}}{f_p}\right) \\ \alpha_D = \theta_D - \arctan\left(\dfrac{x_{cu}}{f_c}\right). \end{cases} \tag{2}$$

2.3. Decoding. Decoding in a CSL system is a challenging and complex problem. The goal of decoding is to establish a one-to-one correspondence between the pattern elements in the projection pattern and the acquired distortion pattern. In this section, the process of decoding includes image preprocessing, pattern element extraction, pattern element



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 2: Eight pattern elements. (a–h) Elements with labels 0–7.

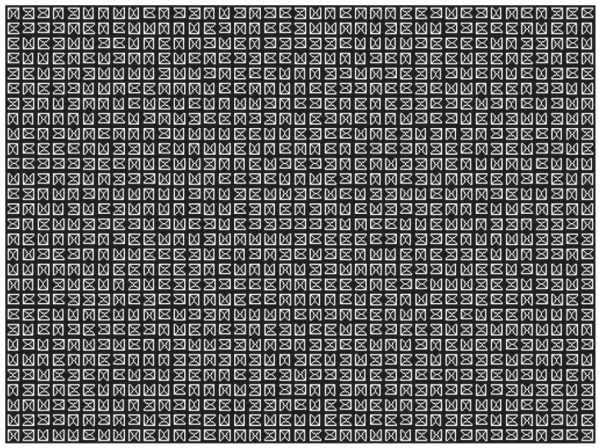


Figure 3: 39×29 M-array with 2×2 window property.

classification, pattern element matching, and error correction.

2.3.1. Image Preprocessing. Due to modulation of the object surface, the shape and intensity of the projected pattern may not be uniform. The traditional image segmentation detection method is unsuitable for pattern element detection. Image preprocessing should be performed in order to accurately detect the pattern elements. The initial captured image is first transformed into a greyscale image; then the greyscale image is binarized by applying a segmentation threshold [24]. A classical thinning algorithm [25] is finally applied to the binary image until a skeleton of each pattern element is obtained.

2.3.2. Pattern Element Extraction. To extract pattern elements, an effective pattern element tracking algorithm (PETA) for the binary image is proposed in this paper. The principle of PETA is as follows. We first scan the thinned image from left to right, top to bottom, to locate the starting point of each pattern element (the point with a pixel value of 1). The skeleton is then tracked to form an ordered chain sequence. The skeleton of each element is tracked twice, and an intersection can be visited several times, so that the complete skeleton of an element can be obtained and a chain list of ordered pixel coordinates can be formed. When tracking a pattern element, a certain tracking order should be enforced. The first time is tracked in the clockwise direction, and the second time is tracked by returning to the starting point and following the counter clockwise direction. PETA is shown in Algorithm 1.

As shown in Algorithm 1, the untracked target pixels in a pattern element are in layer 0, and the tracked pixels are in layer 1. When tracking in layer 0, the target pixel in layer 0 in the pixel's eight connected region is compared with the Euclidean distance between the pixel and the other pixels in an eight connected region. A shorter distance of the pixel is preferred. If there are no pixels in layer 0, tracking proceeds in layer 1 while attempting to locate the next pixel in layer 0. This continues until the starting point is reached.

After tracking the thinned image, the pattern elements are composed of a few pixels, which are sorted automatically along a certain direction, such as counterclockwise. Suppose a pattern image has t $(t \geq 1)$ elements, and each element belongs to a chain sequence. Thus, the entire thinned image will have t chains. The chains of all elements in the pattern image can be represented as follows:

$$\begin{cases} F_c(x, y) = \{C_1, C_2, \cdots, C_t\}, \\ C_i = \left\{p_i^1, p_i^2, \cdots, p_i^k\right\}, \quad i = 1, 2, \cdots, t, \end{cases} \tag{3}$$

where $F_c(x, y)$ is the image generated by the chain extraction algorithm after thinning, and (x, y) are the coordinates of the pixel. C_i is the i-th chain sequence, p_i^k is the k-th pixel in the i-th chain sequence, and each chain is composed of k $(k \geq 1)$ ordered pixels. There will be some defects or artefacts in the obtained image. These defects usually exist in the form of outliers, which represent a few pixels in the chain, and they need to be deleted. In the process of generating a chain sequence, a threshold value T_{num} can be used to remove the chain if the chain has less than T_{num} pixels, which can be calculated as follows:

$$\begin{cases} F_{Nc}(x, y) = \left\{C_{N1}, C_{N2}, \cdots, C_{Nq}\right\}, \\ \text{Num}\left(C_{Nq}\right) \geq T_{\text{num}}, \quad q \leq t, \end{cases} \tag{4}$$

where $F_{Nc}(x, y)$ is the final image generated using PETA, Num is a function used to calculate the number of pixels in the contour, and C_{Nq} is the newly generated q-th chain.

We can clip each pattern element from the original deformed pattern image by using the obtained chain sequences. Each pattern element can be clipped from the original image in order of its index number in chain sequences to generate an independent subimage of pattern elements, which is then saved. Elemental extraction involves four sequential processes. First, four margin coordinates (top, bottom, left-most, and right-most) are computed as follows. The top and bottom margin coordinates are the smallest and largest y-coordinates of pixels in the chain, respectively. Similarly, the left-most and right-most margin coordinates are the smallest and largest x-coordinates in the

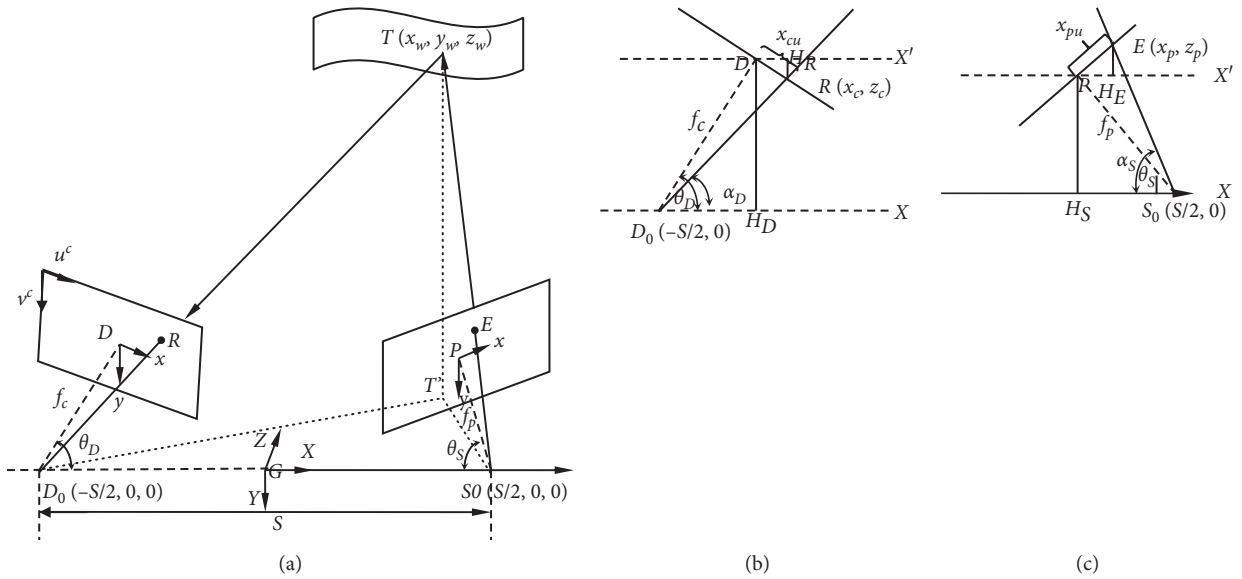


Figure 4: The CSL imaging system. (a) The triangular calculation model of CSL. (b) Camera side. (c) Projection side.

chain sequence, respectively. We can obtain a blue rectangular box, labelled as skeleton border shown in Figure 5. Second, with the help of four margin coordinates, we can locate the pattern element in the binary image. The yellow rectangular box in Figure 5 represents the binary subimage of the pattern element. Some pixels are retained from each margin, to avoid shape-descriptive overflow errors when using the edge extraction algorithm, which is five in our study. The binary pattern subimage is obtained by using segmentation border, shown in right-top subimage in Figure 5. Third, once we have obtained the binary subimage, we can easily locate the pattern element in original captured image. It has the same coordinates and size as the binary subimage, shown in the right-bottom subimage in Figure 5. Finally, the classical bilinear interpolation method (fast speed and less computation) was used to resize the resulting images, and the final image has 32×32 pixels, which we call image size normalization. Figure 5 shows that the mathematical topological structure of the original image's internal structure is retained, although the shape of the pattern element image is changed. The above four steps are used to extract the original pattern element image from the original captured image. This is then converted to a 32×32 image to form the dataset for training the DCNN.

2.3.3. Pattern Element Classification. Results from the pattern element classification are presented in this subsection. A DCNN based on LeNet [26] is used for classification. The overall structure of the DCNN is illustrated in Figure 1. The DCNN learns the relevant local features from low to high-level layers. There are three convolutional layers and three pooling layers. When interleaved with Max and Ave pooling strategies, this method can capture deformable, invariant features by using an affine transformation. The latter fully-connected layers can capture complex co-occurrence statistics, which improves the learning ability. The final layer outputs a decision that is produced using the end-to-end network feature map. This architecture is appropriate for learning local features from the element image dataset:

(1) *Input Image Layer*. This layer accepts pattern elements with 32×32 pixels.

(2) *Convolution Layer*. The convolution layer is used to extract image features. C1, C2, and C3 in Figure 1 are convolution layers. The size of the convolution core is 5×5 with a step size of 1, and the width and height are padded with 2 pixels. The three convolution layers C1, C2, and C3 produce 48 feature maps of 32×32 pixels, 64 feature maps with 16×16 pixels, and 128 feature maps with 8×8 pixels. The ReLU activation function is used to extract the image features.

(3) *Pooling Layer*. S1, S2, and S3 in Figure 3 are pooling layers, each of which is connected to a 3×3 neighborhood of the previous convolution feature map with a step size of two. Among them, a Max pooling strategy is used in S1, S2, and S3 to adopt the Ave pooling strategy. The three pooling layers, S1, S2, and S3, produce feature maps with 16×16 pixels, 8×8 pixels, and 4×4 pixels, respectively.

(4) *Fully Connected Layer*. FC1 and FC2 are the fully connected layers, in which 256 features are output by FC1 and FC2. The classifier is used to calculate the probability of belonging to each output category. At the output end of FC1 and FC2, there is a dropout layer to reduce the risk of overfitting. Finally, the classification output is generated as one of eight possibilities using the SoftMax classifier.

To accurately identify the pattern element labels with DCNN, sufficient labelled data is required to train the network. Therefore, the encoding pattern was projected onto various scenario targets, such as a plane, curve surfaces,

```
Img←Thinned image;
H←Image height;
W←Image width;
DO WHILE (j < H)
  DO WHILE (i < W) {
    pixelA←Information in the first object pixel;
    //∗pixelA is a structured variable, including pixel value, pixel coordinates, number of layers, type of pattern element, order of
pattern elements, and direction angle∗//
    IF (pixelA is the layer 0 pixel) {
      Track the pixels using layer 0 rule;
    IF (The Euclidean distance of pixelA is the shortest)
      pixleA is pushed into the chain sequence;
      Look for the next pixel;
    ELSE
      Look for the next pixel;
    }
    ELSE {
      Track the pixels using layer 1 rule;
      pixleA is pushed into the chain sequence;
      Look for the next pixel;
  }
  //∗ Complete first element tracking, continue looking for the second element∗//
}
```

Algorithm 1: PETA.

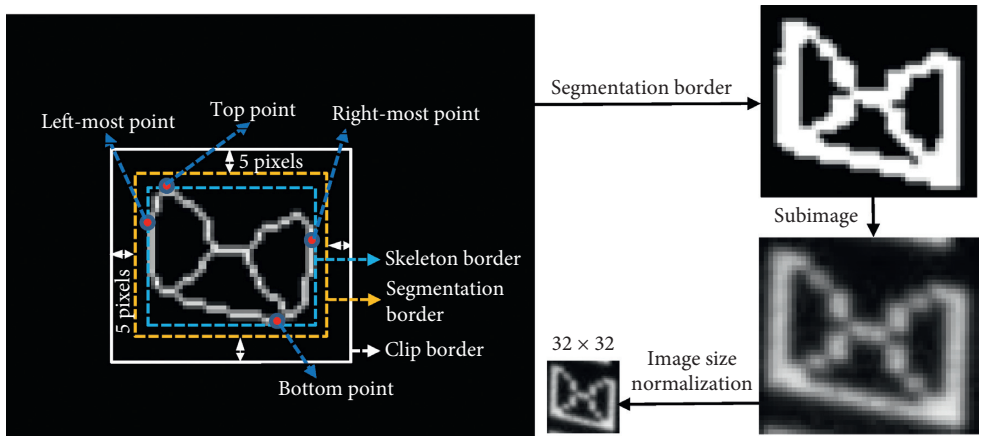


Figure 5: Element extraction and image normalization.

ladder, ball, and statues. We extracted the elements from the original captured image, segmented images, and thinned images. The training dataset contained 20346 images. However, this set is not large enough to train a DCNN to high accuracy. Moreover, a large-scale dataset should be constructed to prevent overfitting and increase the training accuracy. Therefore, the following data augmentation technique was used to increase the size of the training dataset. For each original image, an additional seven images were expanded. These newly formed images were created by rotating the images clockwise by two different angles (5° and 15°), translating the images in the lower left direction by 50 and 100 pixels, enlarging the images by factors of 2.0 and 0.5 using bilinear interpolation, and downsampling the images in two intervals. After several image transformations, the size of the dataset increased to 142422 images. Sample images of the pattern elements are shown in Figure 6.

2.3.4. Pattern Element Matching. The generated M-array pattern is an array with 29 rows and 39 columns composed of eight elements, which are labelled from 0 to 7 with 2×2 window property. That is, any subwindow with size of 2×2 only appears once in the whole pattern. This feature is used to locate a subwindow. We have constructed a lookup table of project pattern. If the value of the subwindow in the projection pattern is equal to the value of the subwindow in the captured distortion pattern, the two subwindows will match. The following function is used to calculate the value of this subwindow, which is globally unique throughout the pattern:

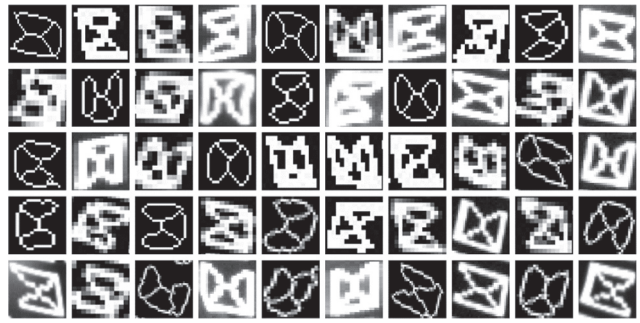


Figure 6: Sample images showing pattern elements with blur and distortions.

$$K(r,c) = 1000 \times f(r,c) + 100 \times f(r,c+1) + 10 \times f(r+1,c) + f(r+1,c+1), \quad (5)$$

where r and c represent the row and column numbers of the element in the standard pattern array, respectively, $0 \le r \le 29$, and $0 \le c \le 39$. The value of f ranges from 0 to 7, and the value of $K(r, c)$ ranges from 0 to 7777. The entire pattern array can be used to build a lookup table using the calculated subwindow value in equation (5). Each value in the array is unique, which corresponds to a unique subwindow.

After image processing and element classification, we can recognize elements in the acquired image according to the pattern element matching. Equation (5) shows that we can calculate the value of every subwindow and search it in the lookup table constructed from the standard pattern array. If the same value can be found, the window will establish a corresponding relationship with the subwindow in the pattern. If all subwindows are matched in this way, the entire image can be matched.

2.3.5. Error Correction. An element recognition error may cause mismatching (wrong matching element) of a subwindow because the same element may belong to more than one subwindow. Therefore, an error correction algorithm based on a voting mechanism is proposed for error correction. As shown in Figure 7, W_1, W_2, W_3, and W_4 are 2×2 subwindows, each of which contains four pattern elements (circles). When the window slides, each element may be contained in at most four subwindows, as shown by the red circle in the middle of Figure 7. Of course, an element may be contained in one, two, or three subwindows. That is, the position of each element can be determined in at most four subwindows in the pattern image.

In the generated projection pattern, the number of votes of each element is an ideal value, called a theoretical number of votes V_{TNV}. The number of votes for each element in the acquired distortion pattern is V_{ANV}. To uniquely determine the matching position of a distorted element in the standard pattern, the following condition should be satisfied:

$$V_{ANV} \ge V_{TNV} - 1. \quad (6)$$

The following approach is used to calculate V_{TNV} for an element. Each element in the element's eight connected region is searched, forming a 2×2 subwindow, and V_{TNV} will increase by 1. V_{TNV} cannot exceed four votes. The following method is used to calculate V_{ANV} for an element. Using the M-array window property, for each element in the acquired distortion pattern, once we have matched the position of the element in the standard pattern, we will add 1 to V_{ANV} for the element and finally the position and V_{ANV} for the element with the most votes is recorded. In practice, the voting mechanism follows the rule of "minority is subordinate to the majority." For example, three of the four subwindows of an element determine that the element's position in the standard pattern is (r, c), but the fourth subwindow determines that the element's position in the pattern is (r', c'), where $r' \ne r$ and $c' \ne c$. According to the voting mechanism, the credibility of the position determined from three windows is higher than that determined from one subwindow, so the position (r, c) is the matching element.

According to the voting mechanism, there are 0, 1, 2, 3, and 4 cases of vote in the matching process. For example, the elements on the top, bottom, left, and right edges will have two votes, as these elements are located in two subwindows at most. The number of votes for the element in the four corners is only 1, as these elements are located in only one subwindow. An element with 0 vote corresponds to an element that is isolated or unrecognized.

2.4. System Calibration. Calibration for CSL is the first step towards 3D reconstruction of the measured object. We refer to the comparative review presented by Zhang [27] for mathematical details of camera calibration and Chen et al. [28] for projector calibration. The calibration technique of Zhang's method only requires the camera be used to observe a planar pattern from a few (at least two) different orientations. The projector is claimed to be conceptually reciprocal to camera, yet it always adopts a reduced projection model. The calibration procedure for the projector can refer to the calibration procedure [28].

2.5. 3D Reconstruction. Once we have finished the decoding and calibration processes mentioned above, the 3D coordinates of a feature point can be computed as follows:

$$\begin{bmatrix} x_t & y_t & z_t & 1 \end{bmatrix}^T = T_m \begin{bmatrix} x_w & y_w & z_w & 1 \end{bmatrix}^T, \quad (7)$$

where $\begin{pmatrix} x_t & y_t & z_t \end{pmatrix}$ is the final 3D information. T_m is the homogeneous transformation matrix from the camera,

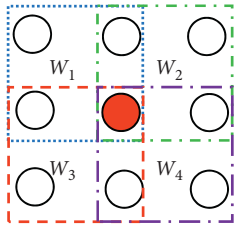


Figure 7: The relationship between pattern elements and subwindows.

which can be computed using the aforementioned calibration method. $(x_w \ y_w \ z_w)$ comes from equation (1).

To confirm the feature points that are required to reconstruct 3D information, we present an angle variation method to detect the key points (KPs) in element. If we define the angle variation at one point, e.g., point p in Figure 8, we can describe the method as follows.

Giving an integer r that denotes the number of pixels, and assuming I_p is the index of r in the contour sequence obtained in the above section, the coordinates of two points a and b in the chain sequence can be calculated as follows:

$$\begin{cases} x_a = \dfrac{1}{r}\displaystyle\sum_{i=Ip-r}^{Ip-1} x_i, \quad y_a = \dfrac{1}{r}\sum_{i=Ip-r}^{Ip-1} y_i, \\ \\ x_b = \dfrac{1}{r}\displaystyle\sum_{i=Ip+1}^{Ip+r} x_i, \quad y_b = \dfrac{1}{r}\sum_{i=Ip+1}^{Ip+r} y_i, \end{cases} \tag{8}$$

where x_i and y_i are the coordinates of point i in the contour sequence. We can define vectors $\overrightarrow{ap}$ and $\overrightarrow{pb}$ as

$$\begin{cases} \overrightarrow{ap} = \text{complex}(x_p, y_p) - \text{complex}(x_a, y_a), \\ \overrightarrow{pb} = \text{complex}(x_b, y_b) - \text{complex}(x_p, y_p), \end{cases} \tag{9}$$

where the function complex denotes application of equation (8). The angle variation at point p is

$$\theta_p = \text{angle}(\overrightarrow{pb}) - \text{angle}(\overrightarrow{ap}), \tag{10}$$

where the function angle calculates the angle between the x axis and the vector $\overrightarrow{ap}$ or $\overrightarrow{pb}$. θ_p ranges from -180° to 180°.

Horns and intersections between elements are settled in a pattern acting as KPs, and these points have large angle variations. The KPs of a pattern element numbered 0 are shown in Figure 9(a). The angle variation graph based on the points of the pattern element is shown in Figures 9(b) and 9(c), where $r = 3$. Figure 9(b) shows the external angle variation of Figure 8(a), the x axis is the point index of the chain sequence, and the y axis is the angle variation of the point index. Figure 9(c) shows the internal angle variation graph.

As shown in Figure 9(a), the waveform showing the angle variation has a positive peak at KPs 1, 2, 3, 5, 6, 8, and 10; at intersections 4 and 7, the angle waveform has a negative trough. The angle variation is closely related to a given integer r. A larger value of r aids observation of the overall shape of the chain, and a smaller value is used to obtain the details of the angle variation. At these KPs, the chain variation will be large. Thus, these points have high positioning accuracy and can be used as detection points. Waveforms showing variations in the angles of the other seven elements can be obtained using the angle variation algorithm, as shown in Figure 10. These waveforms will show that the positions of the KPs and intersection points of different elements are not the same, which can be used to locate the KPs.

Different elements have different angle variations, and two elements exhibit remarkable differences between the positive and negative peaks. Meanwhile, the number of positive and negative peaks for these eight elements is different. Thus, these features can help us identify the eight elements.

An unknown region in a pattern element can be greatly shrunk by identifying KPs, such as horns and intersections. Subsequently, the unidentified elements will be mapped in succession to a position in the standard pattern. The globally unique characteristic of a subwindow in M-arrays is used to identify four elements, and the KPs act as reference points during the mapping process. The two nearest reference points are chosen to implement mapping to minimize identification error. After all elements are identified, the elements in the subwindow can be determined. 3D information at KPs in the elements will be obtained.

3. Experiments and Results

Experimental results are presented in this section to demonstrate the feasibility of our proposed method. The experiments were conducted with a structured light system consisting of commercial projectors (Epson EMP-821 Series LCD Projectors, 1024×768 resolution, 20 Hz frame rate) and a CCD camera (CoolSNAP cf CCD, 1040×1392 resolution, $4.65\mu\text{m} \times 4.65\mu\text{m}$ pixel size, Kowa Lens LM16HC or LM25HC with 35.0 mm focal length). A server running Windows 7 with an Intel®CoreTM i7-7700K CPU@ 4.20 GHz x8 processor and 8 GB RAM (DDR4 2400 MHz × 2) was used for data training and image processing. The captured images were decoded using C++ and Python version 2.7. Subsequent DCNN construction and training algorithms were implemented using the Caffe framework. In this study, Matlab 2017a was used for postprocessing and 3D data visualization. The measurement distance and baseline distance of the system are approximately 1.35 m and 0.218 m, respectively. This section is organized as follows. First, the classification accuracy and measurement precision were presented, which were obtained with the proposed decoding method. Then, six objects with reflectivity, surface discontinuities, and color were chosen for use in the experiments.

3.1. Evaluation of Classification Accuracy. We constructed a dataset of 142422 element images extracted from the structured light image using Algorithm 1. These element images can be divided into eight categories. To evaluate the classification accuracy, we divided the sample data into a

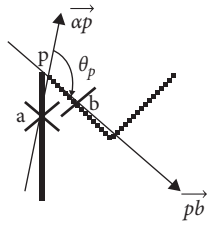


Figure 8: Angle variation definition.

training set, a verification set, and a test set including 12726 training samples, 4242 validation samples, and 4242 test samples, respectively (approximately 6 : 2:2 ratio). A detailed sample distribution for each class is listed in Table 1.

Stochastic gradient descent (SGD) and the sigmoid actuation function were used for training. The DCNN was trained using batches of 512 images. Weight decaying and dropout probability of 0.4 in the last two fully connected layers were used during recognition. A learning rate of 5×10^{-4} was chosen. The experimental classification results are presented as an average of ten repeated experiments.

The performance of validation accuracy, training loss, and validation loss during training is shown in Figure 11. The classification accuracy of the validation set kept improving during training (Figure 11(a)), while the training loss and test loss kept decreasing (Figure 11(b)). After 10000 iterations, the accuracy tended to stabilize. During the first 10000 training iterations, the validation accuracy increased rapidly, and the accuracy eventually basically stabilized at more than 99%. The training loss and test loss value decreased rapidly during the first 10000 iterations and then remained below 0.03. In particular, the training loss tended to be stable and close to 0.0001. The training accuracy of the DCNN was approximately 99.5%. When the trained DCNN was used for testing, the network could produce classification accuracy of approximately 98.9% for the pattern elements.

3.2. Evaluation of Measurement Precision. To evaluate the measurement precision, a flat white board was chosen as the target object and was placed 1.35 m from the baseline. Since the Kinect v2 (Kinect for windows v2 sensor) and time of flight (ToF) sensor (SR4000) have reconstruction object ability (by calculating the total flight time of light from the light source to the surface of the object and then back to the sensor, the distance between the object and the sensor can be obtained), they were used to implement the other two sophisticated methods of Li et al. [18] and Jia et al. [15] for comparison with the proposed method. The correspondence of KPs could be obtained with the proposed decoding method. We calculated the distance of the flat white board ten times without moving or vibrating the devices. Real distance was calculated using the average value. We used the root mean square error (RMSE) of the reconstructed object as an indicator of accuracy. The performance of the five different methods is shown in Figure 12. As the figure shows, the measurement precision with our proposed method is higher than that provided by the other four methods.

3.3. 3D Reconstruction of Complex Surfaces. To further evaluate the performance of the proposed method, more complex objects were selected for experiments, as shown in Figure 13. The first object was four white stairs with high reflectivity, as shown in Figure 13(a). The second object in Figure 13(b) is a polygon. The third object in Figure 13(c) is a yellow bottle. The fourth object in Figure 13(d) has many subshapes. The fifth and sixth objects in Figures 13(e)–13(f) are a head model and a mouth model with deep slopes and surface discontinuities, respectively. By applying the established experimental platform, the designed pattern was projected onto the objects. The camera captures the distorted images, which are then converted to greyscale images, as shown in Figure 14. Meanwhile, Figure 14 shows the pattern projected on the objects that different objects have different distortions due to their different depths.

Figure 15 shows 3D point clouds for all objects determined with the proposed decoding method. The point clouds of the second, third, and fifth objects are incomplete. This is expected because some patterns in the areas with edges or deep slopes were not extracted during pattern element detection, and it was difficult to correctly classify some pattern elements with abnormal blurring or drastic distortions. Once the pattern elements were not extracted or the pattern elements were falsely classified, it would be difficult to obtain matched points from every element because the subwindow size was 2×2. We adopted two parameters presented by Jia et al. [15] to quantify decoding performance: recognition rate for pattern elements, and the erroneous judgment rate for subwindows. Some valuable conclusions could be obtained. The recognition rate for pattern elements was greater than 97%, and the error judgment rate for the subwindow was less than 6%. In general, these experimental results show that the proposed decoding method can be used to reconstruct objects with surface color and complex textures.

Figure 16 shows the depth information reconstruction results, with bilinear interpolation used for all objects. Figure 17 shows the 3D reconstruction results using point cloud and mesh processing software (VRMesh 11.0) for all objects. Although some areas without 3D points could be completely resolved, the reconstructed 3D shapes for all objects were nearly acceptable. These experimental results show that the proposed encoding and decoding method can be used with objects that have surface colors and texture.

The experimental results provide the following contributions:

(1) A one-shot 3D imaging approach is proposed to reconstruct an object's shape from one single image. This projected pattern is constructed by a 39×29 M-array with a 2×2 window property, with only eight geometrical shapes.

(2) The pattern elements dataset is established after extracting the elements using PETA. Then, an improved DCNN based on LetNet is proposed.

(3) An interesting decoding method that combines deep learning based on DCNN and chain angle is proposed to complete decode.

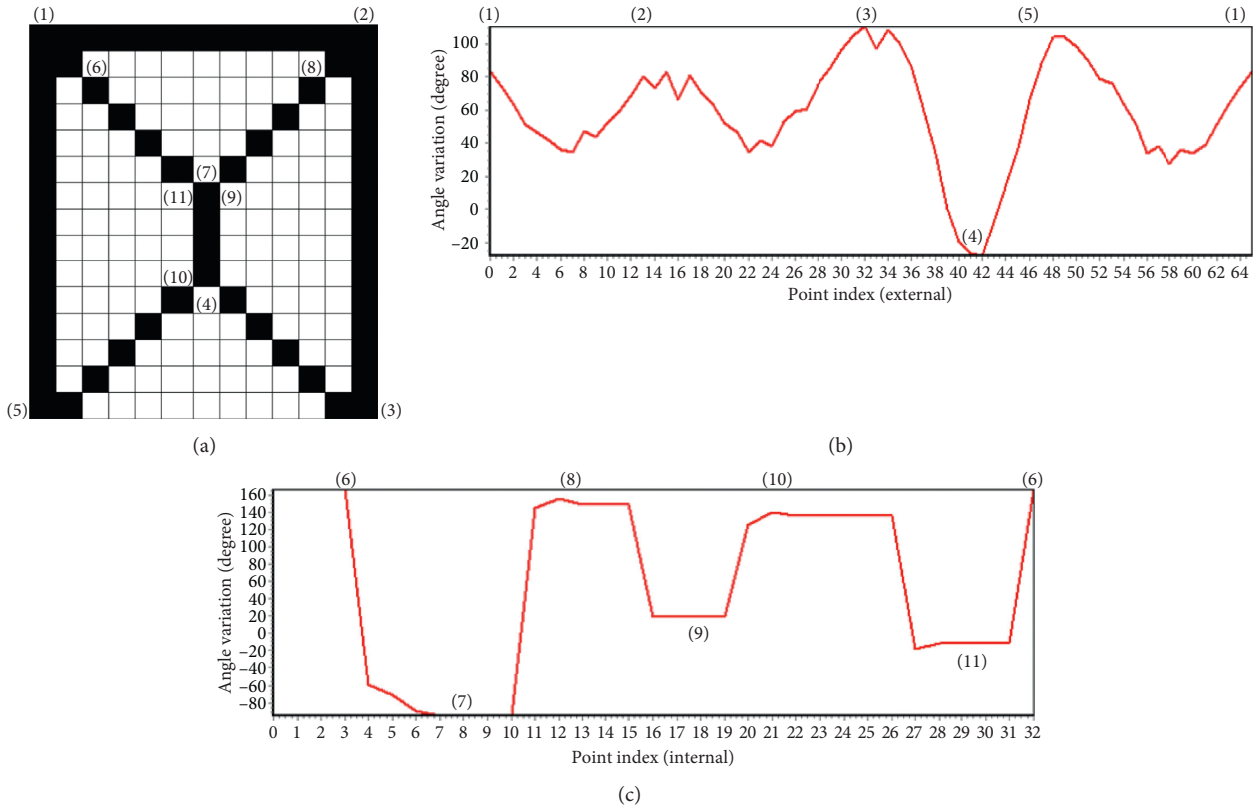


Figure 9: KPs definition and angle variation of pattern element 0 when $r = 3$. (a) KPs definition of the element with label 0. (b) Angle variation of external point of chain sequence for the pattern element. (c) Angle variation of internal point of chain sequence for the pattern element.

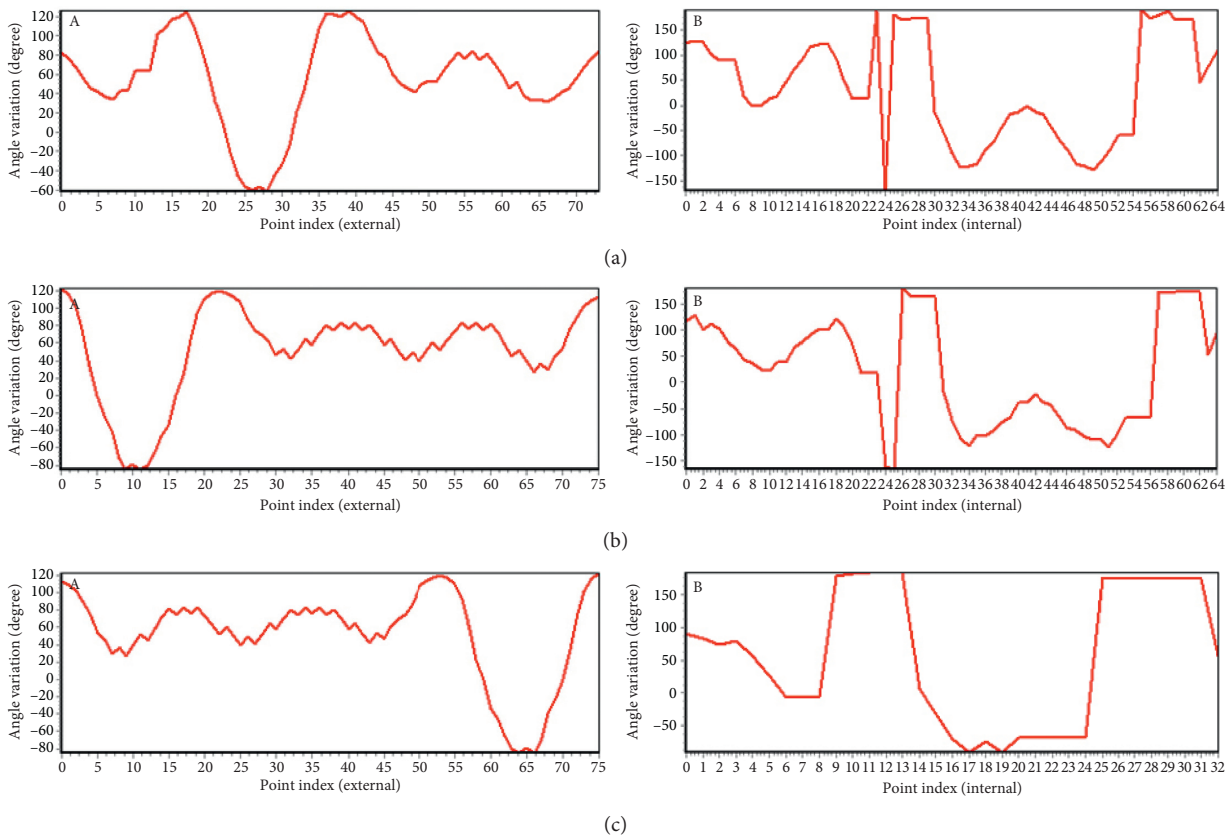


Figure 10: Continued.

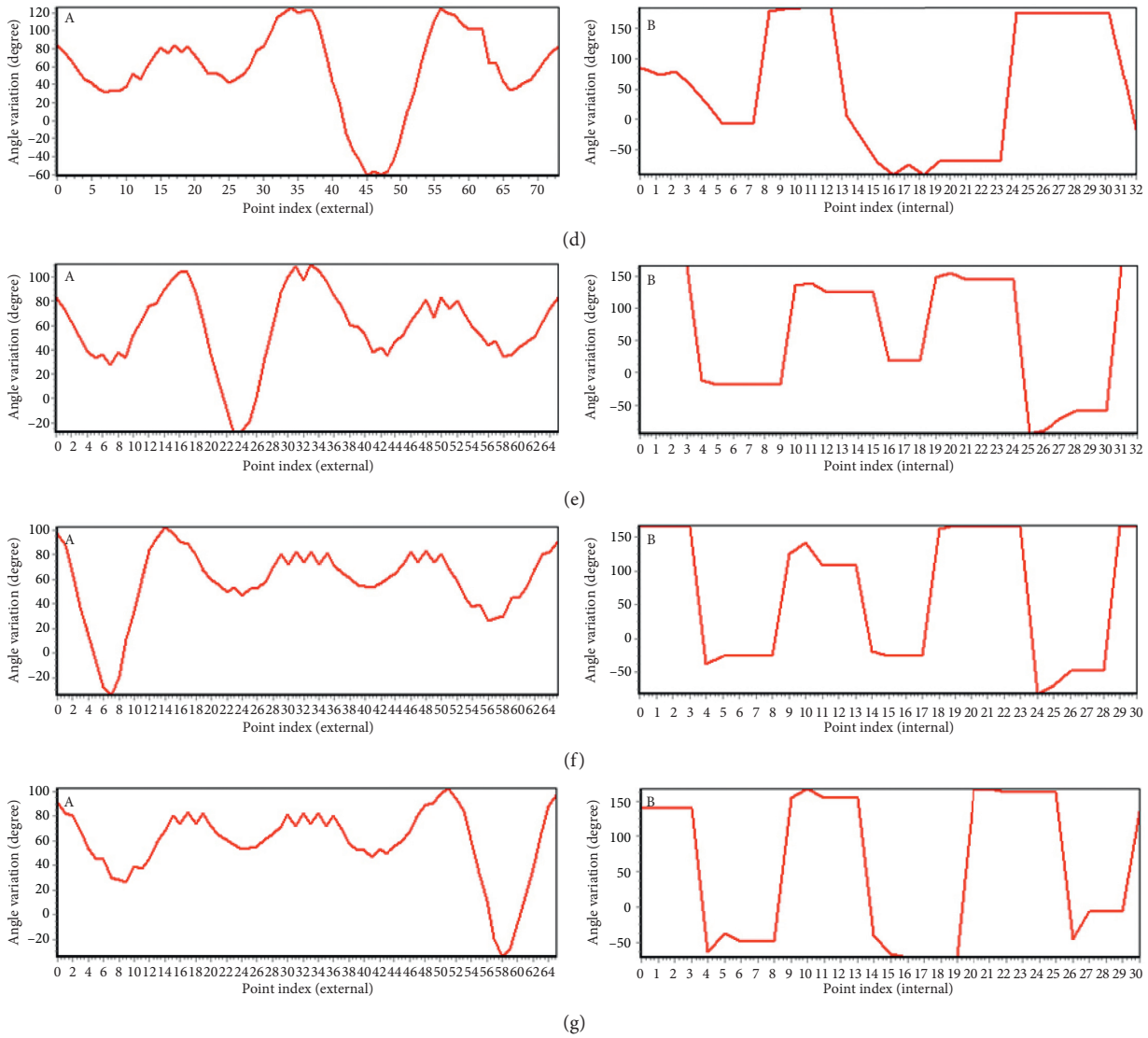


Figure 10: Angle variation of seven other elements. (a–g) Elements with labels 1–7.

Table 1: Distribution of samples in dataset.

Data style	Eight different pattern elements classes							
Label	0	1	2	3	4	5	6	7
Training set	10986	10412	10386	10680	10762	10550	10823	10857
Validation set	3662	3470	3462	3560	3587	3516	3607	3619
Test set	3662	3470	3462	3560	3587	3516	3607	3619

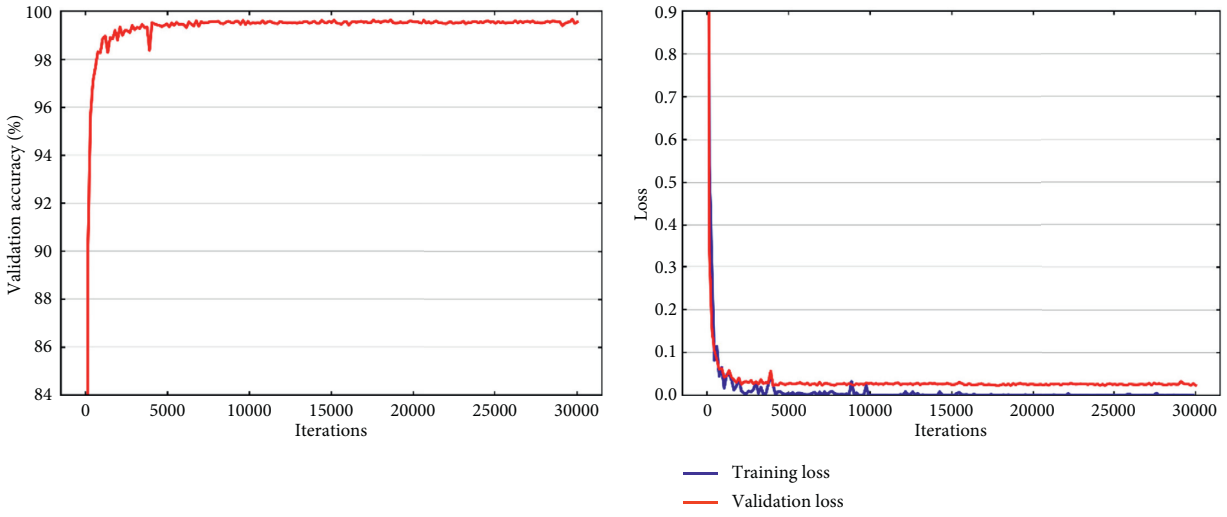


Figure 11: Validation accuracy, training loss, and validation loss results. (a) Validation accuracy and (b) training loss and validation loss.

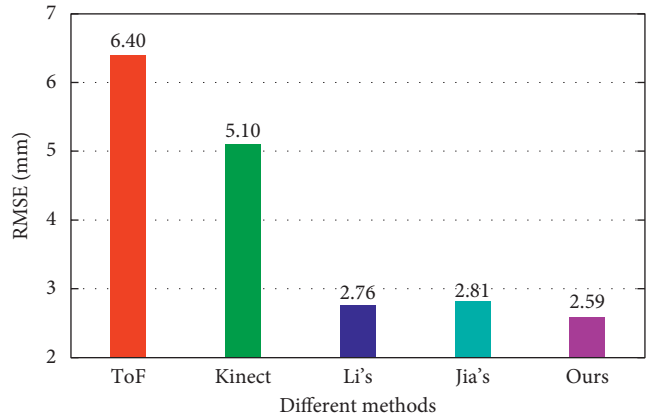


Figure 12: RMSE of ToF, Kinect, Li's method [18], Jia's method [15], and the proposed method.

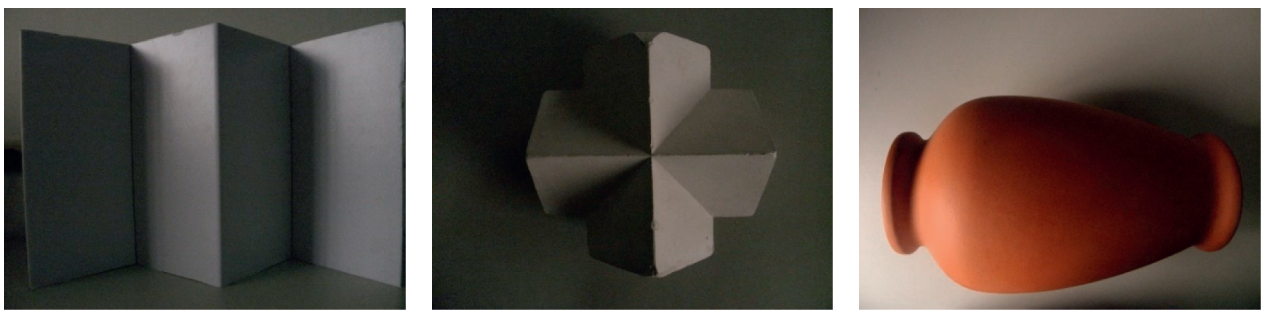
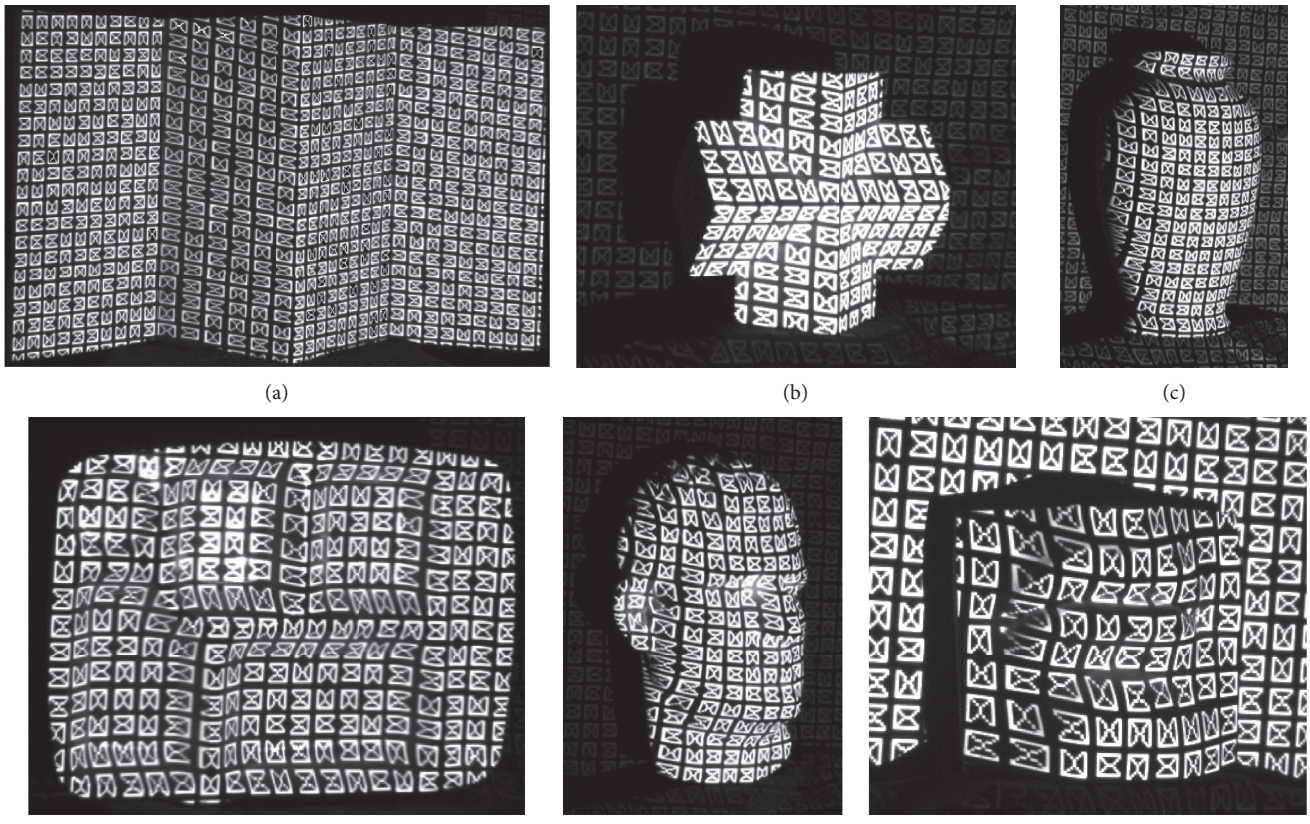


Figure 13: Continued.

(d) (e) (f)

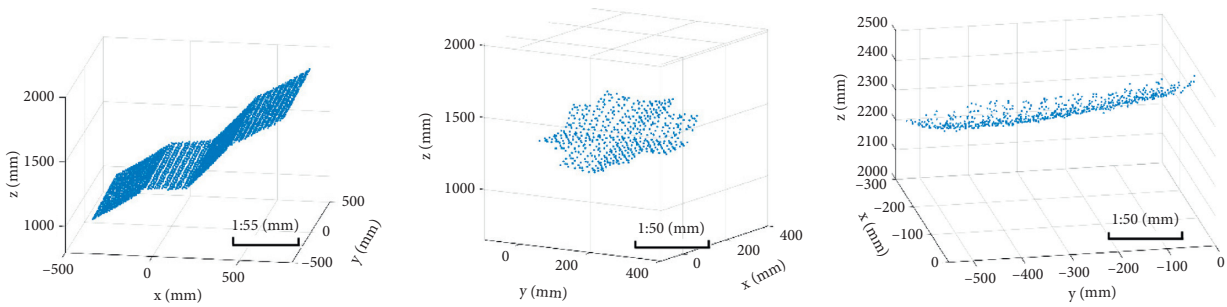
Figure 13: Different objects used in object reconstruction experiments. (a) Four stairs. (b) Polygon. (c) Yellow bottle. (d) Tableware. (e) Head model. (f) Mouth model.



(a) (b) (c)

(d) (e) (f)

Figure 14: The deformed patterns of projecting the pattern on the objects captured by the camera for (a) four-stairs, (b) polygon, (c) yellow bottle, (d) tableware, (e) head model, and (f) mouth model.



(a) (b) (c)

Figure 15: Continued.

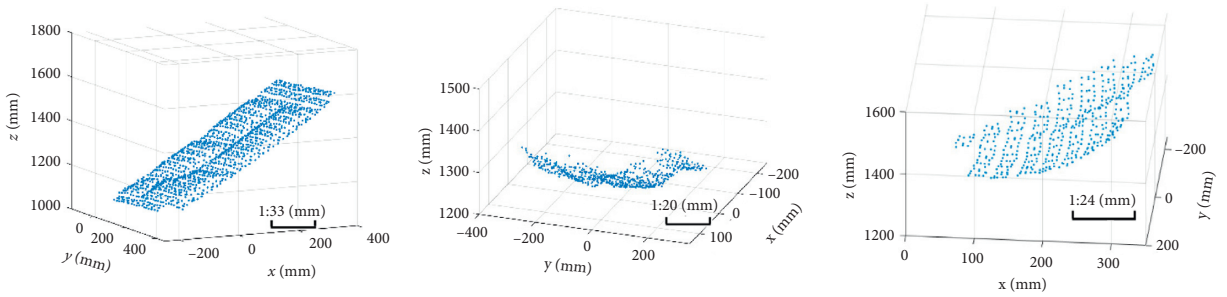


Figure 15: 3D point clouds for (a) four-stairs, (b) polygon, (c) yellow bottle, (d) tableware, (e) head model, and (f) mouth model.

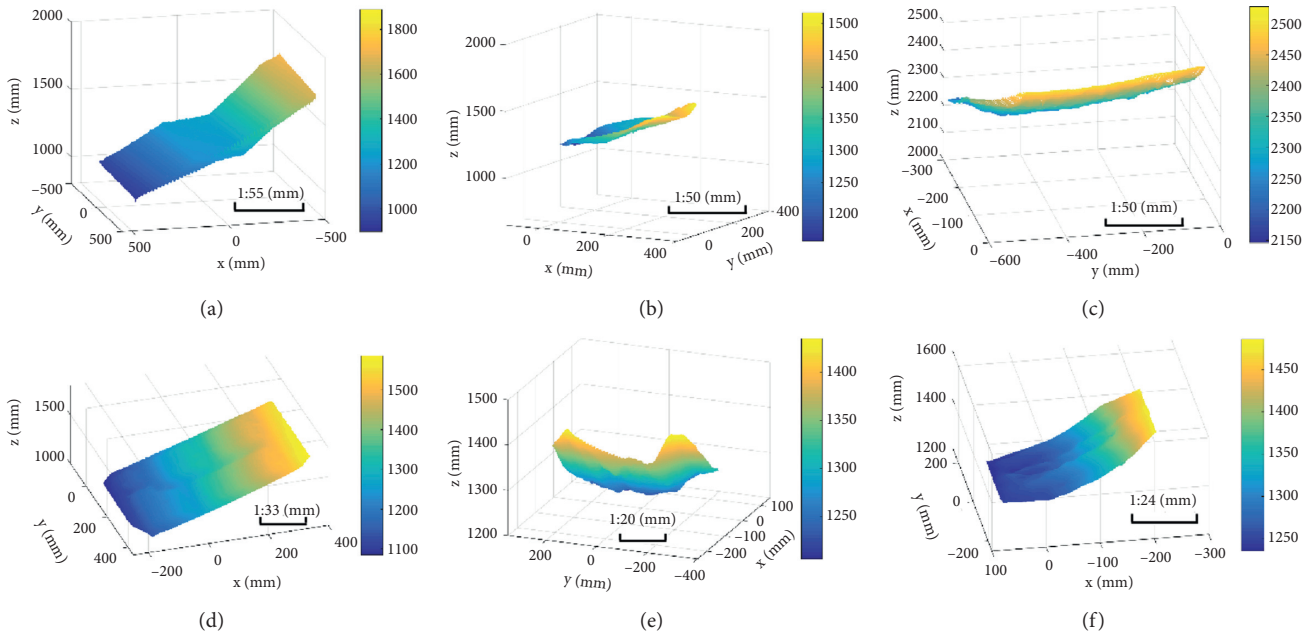


Figure 16: 3D depth maps for (a) four-stairs, (b) polygon, (c) yellow bottle, (d) tableware, (e) head model, and (f) mouth model.

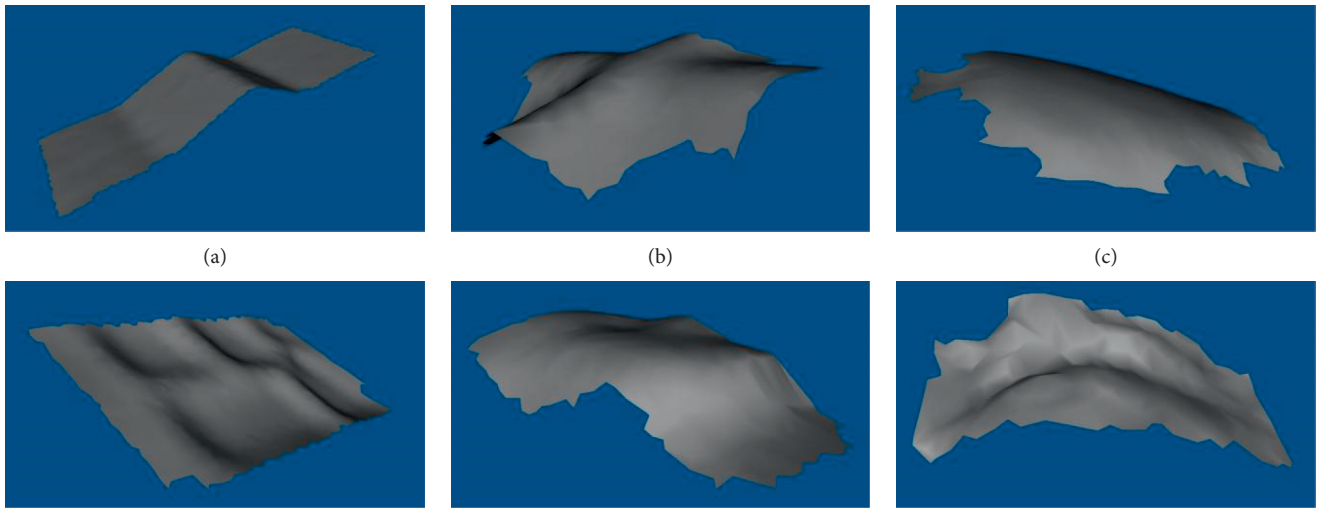


Figure 17: 3D reconstruction results for (a) four-stairs, (b) polygon, (c) yellow bottle, (d) tableware, (e) head model, and (f) mouth model.

4. Conclusions

3D reconstruction using structured light has seen tremendous growth over the past decades. In a one-shot CSL system, encoding and decoding methods are the two major concerns. A solution where an encoded pattern is constructed with eight elements and some KPs and then used for detection is presented in this paper. Pattern element tracking and a DCNN were used for decoding. The proposed pattern was designed as a geometric shape with a 2×2 window property, and some KPs were used for detection. The decoding procedure involved image preprocessing, pattern element extraction, pattern element classification, pattern element matching, and error correction. Because the feature points are defined as the intersection points between elements in the projected pattern, a chain angle method was used to precisely detect the KPs. And, pattern elements can be extracted from the structured light image using PETA. A training dataset with over 1×10^5 samples was compiled, and a DCNN based on LetNet was used to identify pattern elements. Finally, window matching was implemented to determine the correspondence of pattern elements between the projecting pattern and the distorted pattern, and to reduce the number of false matches. The experimental results provide sufficient evidence to show that the method can be used for 3D reconstruction of objects with a variety of surface colors and complex textures. Future research will focus on integration of broken patterns caused by discontinuous surfaces, different colored surfaces, and application for dynamic scenes. To increase the measurement precision and resolution, each pattern element should include more measurement points. But, as the number of measurement points increases, identifying the pattern elements becomes more complex due to noise resistance. Thus, it would be interesting to perform a quantitative evaluation of the effect of reducing the window width while increasing the number of pattern elements, analyzing noise resistance, and finding a compromise between these advantages and disadvantages.

Data Availability

Data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported in part by the Zhejiang Public Welfare Technology Research Project Fund of China under Grant nos. LGG20F010010 and LGG21F030013 and the City Public Welfare Technology Application Research Project of Jiaxing Science and Technology Bureau of China under Grant nos. 2018AY11008 and 2020AY10009. The authors thank LetPub for its linguistic assistance during the preparation of this manuscript.

References

[1] S. Zhang, "High-speed 3d shape measurement with structured light methods: a review," *Optics and Lasers in Engineering*, vol. 106, pp. 119–131, 2018.

[2] J. Salvi, S. Fernandez, T. Pribanic, and X. Llado, "A state of the art in structured light patterns for surface profilometry," *Pattern Recognition*, vol. 43, no. 8, pp. 2666–2680, 2010.

[3] B. X. Gong and G. Y. Wang, "Underwater image recovery using structured light," *IEEE Access*, vol. 7, pp. 77183–77189, 2019.

[4] H. J. Chen, Y. Cheng, D. D. Liu et al., "Color structured light system of chest wall motion measurement for respiratory volume evaluation," *Journal of Biomedical Optics*, vol. 15, no. 2, Article ID 026013, 2010.

[5] Z. Wang, "Robust three-dimensional face reconstruction by one-shot structured light line pattern," *Optics and Lasers in Engineering*, vol. 124, Article ID 105798, 2020.

[6] S. V. Jeught and J. J. Dirckx, "Real-time structured light profilometry: a review," *Optics and Lasers in Engineering*, vol. 87, pp. 18–31, 2005.

[7] J. Pagès, J. Salvi, C. Collewet, and J. Forest, "Optimised De Bruijn patterns for one-shot shape acquisition," *Image and Vision Computing*, vol. 23, no. 8, pp. 707–720, 2005.

[8] C. Je, S. W. Lee, and R.-H. Park, "Colour-stripe permutation pattern for rapid structured-light range imaging," *Optics Communications*, vol. 285, no. 9, pp. 2320–2331, 2012.

[9] T. Etzion, "Constructions for perfect maps and pseudorandom arrays," *IEEE Transactions on Information Theory*, vol. 34, no. 5, pp. 1308–1316, 1988.

[10] F. J. MacWilliams and N. J. A. Sloane, "Pseudo-random sequences and arrays," *Proceedings of the IEEE*, vol. 64, no. 12, pp. 1715–1729, 1976.

[11] J. Lu, J. R. Han, E. Ahsan, G. H. Xia, and Q. Xu, "A structured light vision measurement with large size M-array for dynamic scenes," in *Proceedings of the 35th Chinese Control Conference*, pp. 3834–3839, Chengdu, China, July 2016.

[12] R. A. Morano, C. Ozturk, R. Conn, S. Dubin, S. Zietz, and J. Nissano, "Structured light using pseudorandom codes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 322–327, 1998.

[13] R. Vandenhouten, A. Hermerschmidt, and R. Fiebelkorn, "Design and quality metrics of point patterns for coded structured light illumination with diffractive optical elements in optical 3D sensors," *SPIE Digital Optical Technologies*, vol. 10335, Article ID 1033518, 2017.

[14] J. Pagès, C. Collewet, F. Chaumette, J. Salvi, S. Girona, and F. Rennes, "An approach to visual serving based on coded light," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4118–4123, Orlando, FL, USA, May 2006.

[15] X. J. Jia, Z. J. Zhang, F. Cao, and D. Zeng, "Model and error analysis for coded structured light measurement system," *Optical Engineering*, vol. 49, no. 12, Article ID 123603, 2010.

[16] X. J. Jia, G. X. Yue, and M. Fang, "The mathematical model and applications of coded structured light system for object detecting," *Journal of Computers*, vol. 4, no. 1, pp. 53–60, 2009.

[17] M. Fang, W. Shen, D. Zeng, X. Jia, and Z. Zhang, "One-shot monochromatic symbol pattern for 3D reconstruction using perfect submap coding," *Optik*, vol. 126, no. 23, pp. 3771–3780, 2015.

[18] B. Li, Y. An, D. Cappelleri, J. Xu, and S. Zhang, "High-accuracy, high-speed 3d structured light imaging techniques and potential applications to intelligent robotics,"

International Journal of Intelligent Robotics and Applications, vol. 1, no. 1, pp. 86–103, 2017.

[19] X. Huang, J. Bai, K. Wang et al., "Target enhanced 3D reconstruction based on polarization-coded structured light," *Optics Express*, vol. 25, no. 2, pp. 1173–1184, 2017.

[20] S. Tang, X. Zhang, Z. Song, L. Song, and H. Zeng, "Robust pattern decoding in shape-coded structured light," *Optics and Lasers in Engineering*, vol. 96, pp. 50–62, 2017.

[21] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proceedings of the International Conference on Computer Vision*, pp. 2650–2658, Santiago, Chile, December 2015.

[22] R. Garg, B. G. V. Kumar, G. Carneiro, and I. Reid, "Unsupervised CNN for single view depth estimation: geometry to the rescue," in *Proceedings of the European Conference on Computer Vision*, pp. 740–756, Amsterdam, Netherlands, October 2016.

[23] F. Li, Q. Li, T. Zhang, Y. Niu, and G. Shi, "Depth acquisition with the combination of structured light and deep learning stereo matching," *Signal Processing: Image Communication*, vol. 75, pp. 111–117, 2019.

[24] Z. Liu, F. Cheng, and W. Zhang, "A novel segmentation algorithm for clustered flexional agricultural products based on image analysis," *Computers and Electronics in Agriculture*, vol. 126, pp. 44–54, 2016.

[25] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Communications of the ACM*, vol. 27, no. 3, pp. 236–239, 1984.

[26] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[27] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[28] X. Chen, J. Xi, Y. Jin, and J. Sun, "Accurate calibration for a camera-projector measurement system based on structured light projection," *Optics and Lasers in Engineering*, vol. 47, no. 3-4, pp. 310–319, 2009.