Reconstruction of Surfaces of 3-D Objects by M-array Pattern Projection Method

Hiroyoshi Morita, Kazuyasu Yajima, and Shojiro Sakata

School of Production Systems Engineering
Toyohashi University of Technology
Tenpaku, Toyohashi, Aichi 440 JAPAN

ABSTRACT

A common problem of pattern projection methods to measure surfaces of 3-D objects is that an observed pattern possibly includes disorders such as deficiency, displacement, and permutation of subpatterns. These disorders make it difficult to match observed patterns with its position on the projected one and cause wrong measurements as a result. This paper proposes a new technique to correct pattern disorders by using a pattern made from an M-array which is a two-dimensional extension of a well-known M-sequence.

1. Introduction

Pattern projection methods project patterns such as Moire fringe[4], multi slits[5], or structured lights[6] onto the surfaces of objects in a scene and then determine the spatial coordinates of the illuminated pattern on the surfaces by triangulation. Unlike stereo vision methods[1]-[3], pattern projection methods allow a system to obtain the geometry information of a smooth surface so long as the projected pattern on the surfaces can be observed on the image plane. However, pattern disorders such as deficiency, displacement, and permutation of parts of the observed pattern possibly occur because of the difference between the locations of a projector and a receiver. Therefore, an ambiguity can arise in matching the elements (dots or stripes) of the observed pattern with the corresponding elements of the projected pattern. To determine the position corresponding to the elements of the observed pattern on the projection pattern is called indexing according to [6]. Although various methods have been proposed for solving such an index problem[6]-[8], neither of them can completely correct but only detect and eliminate erroneous indices.

Our goal in this study is to design a procedure for detecting and correcting erroneous indices by using a binary M-array pattern as a projection pattern. An M-array has a window property, that is, all the subpatterns except the all-zero subpattern possibly appearing in a prescribed window really exist once at one place within one period. Using this property, the indices of all the elements of the observed pattern can be uniquely determined relative to the projection pattern if no pattern disorder exists. Even if any disorder occurs in the observed pattern, it is shown on certain practical assumption that it is possible to correct erroneous indices by using the window property of an M-array. Furthermore, an experiment to reconstuct surfaces of 3-D objects from measured points data obtained from the proposed method is described.

2. Pattern Projection Method

2.1. System Arrangement

Fig.1 illustrates diagrammatically the measuring system treated in this paper. By projecting light through an M-array mask, the system takes an images of a dot pattern, i.e. a 2-D arrangement of dots on the smooth surfaces of a set of 3-D objects in a scene, and finally determines the 3-D coordinates of the dots on the surfaces of the objects after matching the observed dots in the images with the projected dots.

Both directions of the camera and the projector in Fig. 1 must be parallel to the Z-axis. It is convenient to arrange the projector and the camera so as to set the Y- and Z-distances D_y, D_Z between them equal to zero. Then, the measuring process can be simplified since the positions of dots of the observed pattern in the image plane

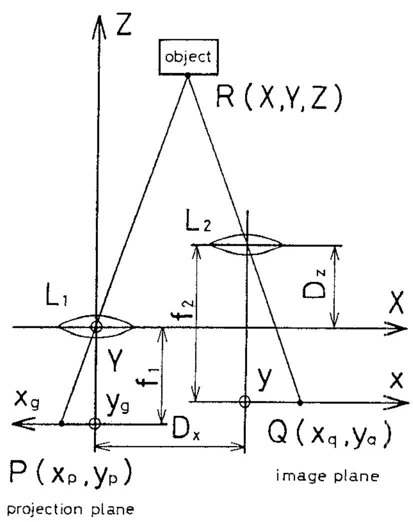

Fig.1 A system arrangment of a project pattern method.

moves only horizontally if an objects moves along the Z-axis direction. Therefore, we assume the following condition holds in the system from now on:

[Condition 1]

$D_y = D_Z = 0.$

2.2 M-array pattern

An M-array is characterized as a solution of a system of linear partial difference equations over the Galois field GF(2)[9][10]. Therefore, we can obtain from the 0-1 values of a window subpattern those of the neighboring elements by using the system of equation. Fig.2 shows an example of M-arrays with 2 x 2 window. Note all the 2 x 2 subpatterns except the all-zero subpattern exist once at a place within one period.

Practically, an M-array pattern consists of circled dark and light dots whose colors are correspondent with elements of M-array. That is, 'dark' and 'light' dots represent 0- and 1-elements on an M-array, respectively.

Because of the window property, any observed dot can be matched correctly with a projected one if there exist no pattern disorder. Consequently, the spatial coordinates of the dots on the surfaces are determined by triangulation. However, if the

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Fig.2 An example of a binary 3 x 5 M-array. The box in the figure shows a 2 x 2 window of this M-array.

shapes of the objects become more complex, some disorders possibly appear in the observed pattern.

2.3. Pattern Disorders

Pattern disorders can be categorized as the following three types:

1) Deficiency of Dots

In case there are two or more objects in a scene, it may happen that a reflected pattern is obstructed by an object to reach the image plane. An example is shown in Fig. 3. The dots from #2 to #4 do not appear in the image plane since they are behind object #3 in the view from the camera side.

2) Displacement due to Difference in Depth

If the depths of objects are different, the dots projected on a surface of one object may shift horizontally against the others in the image plane. In Fig.3 dots #10 to #12 on object #2 move right by one dot in the image plane.

3) Permutation of Dots

In case a slender object is in front of another, dots may interchange with each other in the image plane. As shown in Fig.3, between object #1 and object #3, dot #5 exchanges place with dots #6 to #8.

3. Correction of Pattern Disorders

3.1 Temporary Array

To match the observed dots to the projected dots, we produce a 2-D integral array which consists of the numbers of the observed dots in the image. On the condition 1, we can acquire an image of observed dots each of which is arranged on the same horizontal lines as the corresponding projected dot. Thus, each row of the array is distinguished clearly from each other, while each

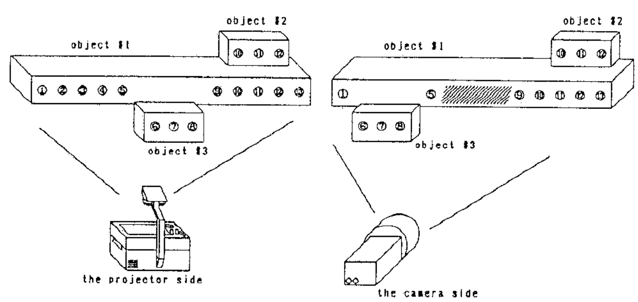

Fig.3 The observed pattern in the camera side contains pattern disorders such as deficiency, desplacement, permutation of subpatterns as described in the text.

column is not always distinguishable. We can assume, however, that every observed dot appears near a lattice point of a 2-D rectangular lattice placed on the image plane adequately. Therefore, the array can be obtained by the following steps:

1) All the elements of the array are initially set equal to zero.
2) As scanning the image from right to left and top to bottom, find dots on it. At each time a dot d_k(k=1,...,N) is found, determine the dot value (0 or 1) and the coordinates of its center (x_k,y_k). Then store them into 1-D arrays VALUE, X, and Y, respectively.
3) Using X and Y, the center coordinates of the found dots are adequately quantized according to the above 2-D lattice so that two adjoining dots on the received image have consecutive quantized coordinates.
4) Substitute the number k of each dot d_k (k=1, ...,N) into an element of the array at the location correspondent with the quantized coordinates of the dot.

The obtained array is called a temporary array or a T-array. Note that a T-array has some vacant elements which do not correspond to any observed dots.

3.2. GROW

Now we try to obtain an 'index' (a column number of an M-array) for each element on the T-array. Initially, set a window of the same form and size as that of a given M-array at a place on the T-array where every element appearing in the window is the number k of a dot d_k found on the image plane. Next, setting a new window on the M-array so as to have the same 0-1 values at the corresponding elements as the T-array's window. Then we can index all the elements in the T-array's window. Now, shift the two windows synchronously by one column or row so long as elements to be entered into each window have the same 0-1 values on both arrays. This process is called a region growing process. And repeat the process until a window can not be set initially in any place on the T-array where each element has not been determined yet.

A region growing process terminates when at least one vacant element exists in the T-array's window or a violation occurs, that is, the value at an element on the T-array is different from that at the corresponding element on the M-array. It is noted that a pattern disorder can be always detected when a violation occurs.

3.3. DETECT

The basic idea of DETECT is to backtrack from the position where a region growing process stopped and to find a place possibly having erroneous indices.

Let us explain how DETECT works by using an example of Fig.4. The size of window is now assumed as 3 x 4. Both column patterns #4 and #5 on the M-array enclosed by dashed lines do not appear on the T-array. We assume that a region growing process has started at (0',0') and (0,0) on the T- and M-arrays, respectively and proceeded to the right, where the location of a window is defined as that of the upper leftmost element in it. When the time the window has reached (0',3') on the temporary array, a violation occurs between column #6' and #6. Then shift the window to the right so that it contains column #6' as the leftmost one. At the same time, move the M-array's window to such a place where every element has the same 0 or 1 value as the corresponding element in the T-array's window. In the example, the window has been moved to (0,8). Backtracking now begins toward the left. And reset the indices of all the dots on the T-array appearing in the window until a violation occurs again. In the above example, a violation occurs at (0',3') and (0,5) on the T- and M-arrays, respectively. Consequently, the indices of dots in

the temporary array

	0'	1'	2'	3'	4'	5'	6'	7'	8'	9'
0'	1	0	0	1	1	1	1	1	1	1
1'	0	0	1	0	1	0	0	1	0	0
2'	0	0	0	0	1	1	1	0	0	1

the M-array

	0	1	2	3	4	5	6	7	8	9	10	11
0	1	0	0	1	1	1	1	1	1	1	1	1
1	0	0	1	0	1	0	1	0	0	1	0	0
2	0	0	0	0	1	1	1	1	1	0	0	1

Fig.4 Column patterns #4 and #5 on the M-array do not appear in the temporary array since the camer angle is defferent from the projector angle. Such a deficiency of subpatterns causes the wrong matching between the temporary array and the M-array.

column #4' and #5' have been reset.

Backtracking causes the region on the T-array where indices of elements have been obtained to shrink. This means that the opposite region on the T-array expands. Then, a new window can be placed on the expanded region, and a region growing process may start again to another direction which has been never tried. After all, most element on the T-array can be determined their own indices by GROW and DETECT. However, some elements corresponding to the dots which have been projected near the boundary between objects can not be determined yet. Such elements are called undecoded elements. Undecoded elements tend to appear alone or to form a chain on the T-array.

3.4. CORRECT

Based on the fact described at the end of the previous section, we adopt a set of several heuristic rules as follows:

1) If both of its upper and lower neighboring elements have the same index, the current element has also the same one.
2) If both of its right and left neighboring elements have particular indices such as k-1 and k+1, then the current element has just the middle index.
3) The index of the current element is bigger (or smaller) by one than that of its left (or right) neighboring element whose position on the image plane is closer to that of the current element.
4) If some of undecoded elements form a consecutive vertical segment, their indices are equal to that of a decoded element just below or above the segment.
5) If some of undecoded elements form a consecutive horizontal segment, the index of each element of the segment is consecutive to one another relative to that of the left or right side element of the segment.

The above rules work well for many cases except pathological cases described below. For examples, suppose that some undecoded elements remain after GROW and DETECT as shown in Fig.5(a). Although these elements might correspond to the dots on object #2, their indices are obtained as 5 by using the above rules. Furthermore, in a part where a window can not be placed completely as shown in Fig.5(b), a displacement might occur. In spite of such a displacement, the wrong indices are obtained as 3, 4, and 5 from left to right. In case of Fig.5(c), the indices of the dots on object #2 are wrongly decoded as 3 to 5 from left to right. However, it is noted that the above examples scarcely occur in practice.

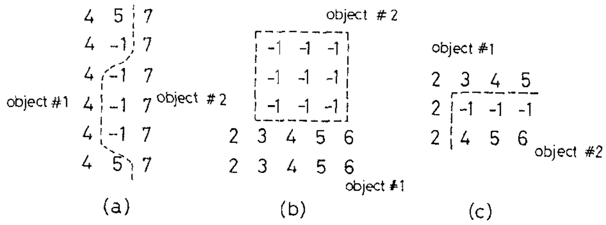

Fig.5 Pathological examples that CORRECT misinterprets the column numbers of dots on the M-array.

4. Experimental Results

4.1. Apparatus

The setup of equipments consists of an over head projector with a 340 mm lens and a CCD monochrome video camera with a 25 mm lens. The x, y, and z directional distances D_X, D_Y, D_Z between them are set up as 185mm, 0mm, and 140mm, respectively. An thin aluminum plate with 32 x 27 holes 1mm in diameter and pitched five holes to the centimeter is placed on the over head projector. This plate generates an all-dots-pattern, while any M-array pattern can be generated by masking this plate. Such a mask sheet consists of a transparent sheet with 1.2mm diameter black dots on the appropriate places. Each black dot is correspondent with an element having the value 1 in a 91 x 45 M-array with a 3 x 4 window size. By projecting first the all-dots-pattern and secondly the M-array pattern to objects, two digital image data of size 512 x 768 with 256 gray-scales are obtained. Then, the white(bright) dots in the image can be separated from the background by using a thresholding technique on a 16-bit personal computer.

Contrary to Condition 1, D_Z has a positive value as described above. Therefore, the observed pattern can move not only horizontally but also vertically in proportion to its distance from the projector. However, when objects in a scene are

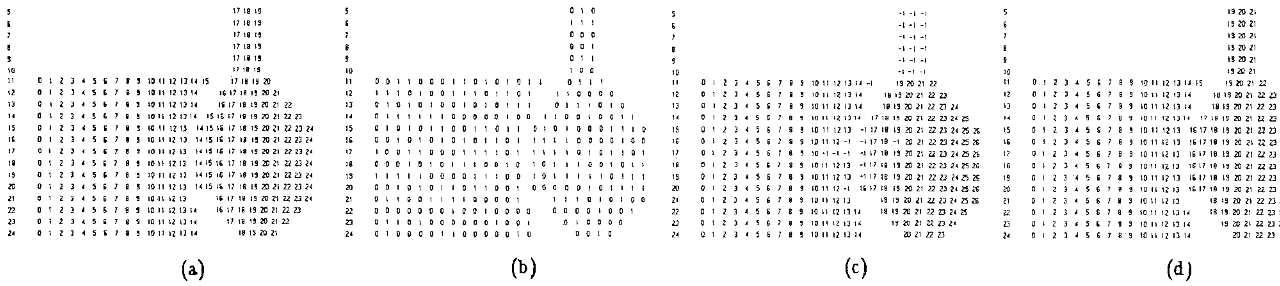

Fig.6 (a) This shows column numbers on the temporary array obtained from the observed dots. The location of each dot corresponds to its position of the image plane. (b) The values of the M-array corresponding to the observed dots. (c) The results of the procedure GROW and DETECT. There are undecoded dots. (d) The outputs of CORRECT. The position of the M-array of every dot has been obtained correctly.

moved about 400mm, each dot on the image plane may move at most 5 pixels. This movement value is less than a half of the distance (= 14 pixels, at least) between neighboring dots on the image plane, which means it is still possible to determine the row number on the M-array of each dot uniquely. As a result, the proposed decoding algorithm is still suitable for the system without any restriction.

4.2. Results

The following four scenes were considered in the experiments:

a) a bottle lying on a table,

b) three boxes,

c) a cone and a box,

d) a bottle with dimples and a box.

The distance of these objects from the focal point of the projector were about 1500mm to 1850mm. Fig. 6(a) presents the observed pattern in case d). All the number except n the leftmost column represent the column numbers of dots on the observed array while those in the leftmost column represent their row numbers. The corresponded M-array values are shown in Fig. 6(b). Fig. 6(c) show the result of GROW and DETECT. Since several dots have remained undecoded, CORRECT has been applied. Finally, correct matching has been obtained as shown in Fig. 6(d). For the rest of scenes, the proposed algorithm has also worked correctly for the rest of scenes.

4.3 Reconstruction of Surface Shapes of 3-D Objects

By using the proposed M-array pattern projection system, we have done an experiment to reconstruct the surface shapes of objects. The surface reconstruction is to represent the surface of an object as a mathematical function such as Spline from a set of spatial coordinates of measured points by means of some surface interpolating or smoothing techniques. Once such a surface description is obtained, it is easy to determine various geometric quantities such as sur-

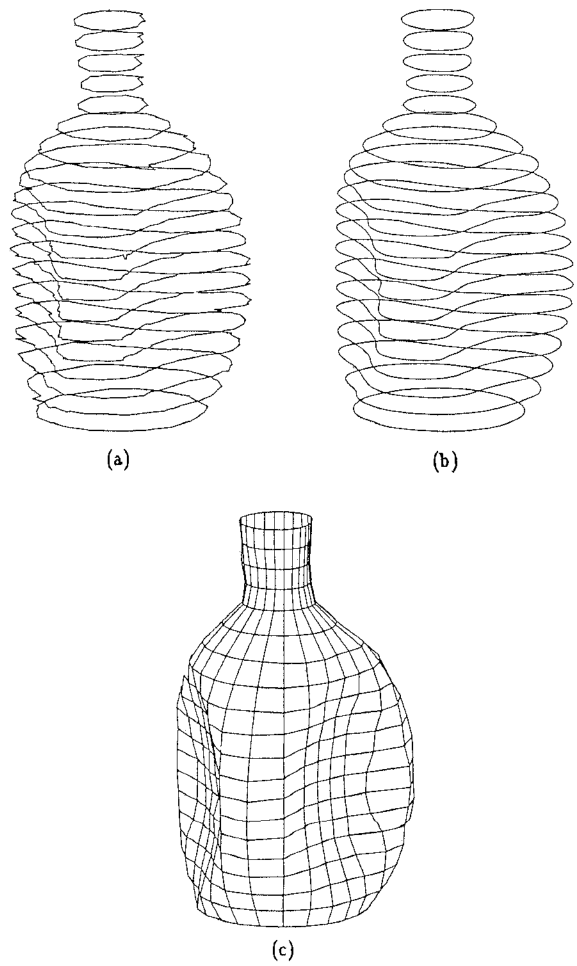

Fig.7 (a) A wire-framed model of a bottle with dimples. (b) From the wire-framed model, B-spline curves have been calculated by means of a smoothing technique. (c) A smooth solid model consisting of bicubic Bézier patches.

face area, volume, and Gauss curvatures. These quantities are quite useful in recognition of curved objects. Our approach to realize surface shape reconstruction is as follows; first, we unify the spatial coordinates of dot points on an object obtained from multi views by the proposed method. Then bicubic Bézier patches are calculated by means of a surface smoothing technique. Fig.7(a) displays a wire-framed model whose vertices are corresponding to the observed dot points. The intermediate result of our smoothing process is shown in Fig.7(b); for each set of the dot points with the same height, a B-spline curve has been obtained. Fig.7(c) shows a smooth solid model finally obtained. In our experiments, by rotating an object placed on a turn table every 90±1.8 degrees, the measurements have been done 8 times. The total number of obtained coordinates data was 1187 and 380 patches have been generated from these data.

5. Conclusion

We have proposed an M-array projection pattern method which enables us not only to obtain the surface shape of objects in a scene but also to detect and correct pattern disorders. Furthermore, we applied the proposed method for surface econstruction problem. Then, we have obtained a smooth solid model to represent an observed object. This solid model enables us to calculate various geometric quantities of the object. These quantities are useful for object recognition.

REFERENCE

[1] Yakimovsky, Y., and Cunningham, R. "A system for extracting three-dimensional measurements from a stereo pair of TV cameras," *Comput. Graphics Image Processing* Vol. 7, pp.195–210, 1978.

[2] Barnard, S. T., and Fischler, M. A. "Computational stereo," *ACM Comput.Surv.*, Vol. 4, no. 4, pp.553–572, 1982.

[3] Yachida, M., Kitamura, Y., and Kimachi, M. "Trinocular vision: new approach for correspondence problem," *Proc. of ICPR*, 1986.

[4] Takasaki, H. "Moire topography from its birth to practical application," *Optics and Letters in Eng.*, Vol. 3, pp.3–14, 1982.

[5] Will, P. M. "Grid coding: a novel technique for image processing," *Proc. IEEE*, Vol. 60, no. 6, pp.669–680, 1972.

[6] Boyer,K. L. and Kak,A. C. "Color-encoded structured light for rapid active ranging," *IEEE Trans.on PAMI*, Vol. PAMI-9, pp.14–28, 1987.

[7] Inokuchi, S., Sato, K. and Matsuda, F. "Range imaging system for 3-D object recognition," *Proc. of ICPR*, pp.806–808, 1984.

[8] Posdamer, J. L and Altschuler, M. D. "Surface Measurement by Space-encoded projected beam systems," *Comput. Graphics and Image Processing* Vol. 18, pp.1–17, 1982.

[9] MacWilliams, F. J. and Sloane, N. J. A. "Pseudo Rando Sequences and Arrays," *Proc. IEEE*, Vol. 64 no. 12, pp.1715–1729, 1976.

[10] Sakata, S. "Finding a minimal set of linear recurring relations capable of generating a given finite 2-D array," *J. of Symbolic Computation* Vol. 4, 1987.