Short Papers

Structured Light Using Pseudorandom Codes

Raymond A. Morano, Cengizhan Ozturk, *Member, IEEE*, Robert Conn, *Member, IEEE*, Stephen Dubin, Stanley Zietz, and Jonathan Nissanov, *Member, IEEE*

Abstract—We solve the correspondence problem in active stereo vision using a novel pseudorandom coded structured light (SL). This coding scheme performs well in the presence of occlusion. In settings where color coding is feasible, 3D information can be obtained using a single image.

Index Terms—3D imaging, structured light, perfect maps, pseudorandom arrays, active stereo vision, correspondence.

1 Introduction

1.1 Overview

One of the primary aims of machine vision for the past twenty years has been to find a method for rapidly acquiring the three dimensional coordinates of points in a scene. Typically, images of the scene are obtained from two distinct vantage points. Points in the scene are then located in each image and a triangulation is performed to find the 3D (or world) coordinates of these points [3], [4], [14]. The task of identifying the same point in the two images is known as the correspondence problem.

There has been substantial effort put into solving the correspondence problem. Some of the most successful approaches utilize structured light (SL). In this technique, one camera in the stereoscopic pair of cameras is replaced by a projector which illuminates the scene with a pattern of light [9], [10]. This reduces the correspondence problem to matching illuminated scene points on a single image with those on the projection plane. Design of an effective illumination pattern is key to simplifying this indexing task and a number of approaches have been developed. These approaches fall into two general classes–temporal codes, where each point is identified by a unique time-modulation of its intensity [1], [2]; and spatial codes, where the position of a point relative to a set of distinct markings uniquely defines its identity [6], [7], [16]. The coding methods being introduced in this paper can be used in temporal as well as spatial implementations of SL.

2 The PSM Codes

We present here an alternative SL code based on pseudorandom arrays. A pseudorandom array is an A-ary array of size $k \times l$ in which many of the possible A-ary matrices of size of v × w (of which there are a total of $A^{v.w}$) appear exactly once as windows [8]. Clearly, such a structure allows complete global position information of an element based only on the vxw window surrounding it. Note that pseudorandom arrays are related to perfect maps, which are simply arrays in which *every* possible $v \times w$ window appears exactly once [18]. Because of this, we also refer to our arrays as perfect submaps (PSM).

2.1 Properties of the Code

The code we have developed consists of a square pseudorandom matrix, M, of size L ($k = l = L$). Each element m_{ij} within M is assigned one letter from a palette of A possible letters. Since M is a pseudorandom array, the location in M of each element m_{ij} (except those along the border) is completely determined by the letters contained in the $v \times w$ neighborhood of m_{ij}. In this case, the position of each m_{ij} is indexed by a nine-element vector $V_{(ij)}$ where,

$$\mathbf{V}_{(ij)} = \left\{m_{i-1,j-1}; m_{i-1,j}; m_{i-1,j+1}; m_{i,j-1}; m_{i,j}; m_{i,j+1}; m_{i+1,j-1}; m_{i+1,j}; m_{i+1,j+1}\right\}$$

$$\mathbf{V}_{(ij)} = \left\{\mathbf{V}_{(ij)1}; \mathbf{V}_{(ij)2}; \mathbf{V}_{(ij)3}; \mathbf{V}_{(ij)4}; \mathbf{V}_{(ij)5}; \mathbf{V}_{(ij)6}; \mathbf{V}_{(ij)7}; \mathbf{V}_{(ij)8}; \mathbf{V}_{(ij)9}\right\}$$

Assignment of letters to the matrix elements is controlled so that each $V_{(ij)}$ differs from all other $V_{(i'j')}$. The separation between each pair must equal or exceed a specified minimum Hamming distance, h, where the Hamming distance between $V_{(ij)}$ and $V_{(i'j')}$ is given by,

$$H(ij, i'j') = \sum_{r=1}^{w^2} (\delta_r) \quad \text{where} \quad \delta_r = \begin{cases} 0 & \text{if } v_{(ij)r} = v_{(i'j')r} \\ 1 & \text{otherwise} \end{cases}$$

Higher Hamming distance codes can be used to perform error detection and correction via standard techniques of communication theory [15]. The feasibility of constructing a pattern where minimum $H(ij, i'j') \geq h$ (for $1 \leq i, j \leq L$, and $(i, j) \neq (i', j')$) depends on L, A, h, and w. To simplify our discussion, we are limiting our examples to 3×3 window sizes and a Hamming distance of one or three (i.e., $w = 3$, and $h = 1$ or $h = 3$), however our approach applies equally well to other window sizes and Hamming distances.

Different elements of A in M are identified at the camera side by giving them different attributes on the projection side. These can be color, shape, temporal encoding, etc. It is therefore desirable to use the smallest possible value of A. For example, in a color encoded perfect submap using fewer colors will simplify the determination of a given point's color in a scene with complex background and surface reflectivity; in a temporal encoded implementation it will decrease the acquisition time. We know that there are only A^9 distinct $V_{(ij)}$ (code words), and that $(L-2)^2$ words are necessary to construct M. Therefore, a lower bound on A can be defined as $A \geq (L-2)^{(2/9)}$. Unfortunately, a direct solution of M for an arbitrary set of A, h, w is not possible. We employed a pseudorandom method of generating M that uses a fill-and-test-as-you-go-along construction scheme, which is explained next. We note here that for certain values of L, A, and w with $h = 1$, there exist construction methods; the constructed arrays are usually nonsquare [8], [12], [13], [15].

2.2 Pseudorandom Code Generation

We begin the pattern generation process by seeding the top left 3×3 window of the matrix with random letter assignments (Fig. 1). The

- *R.A. Morano, C. Ozturk, S. Dubin, S. Zietz, and J. Nissanov are with the School of Biomedical Engineering, Science, and Health Systems, Drexel University, Philadelphia, PA 19104. E-mail: nissanoj@dunx1.ocs.drexel.edu.*
- *R.A. Morano, C. Ozturk, and J. Nissanov are with the Imaging and Computer Vision Center, Drexel University, Philadelphia, PA 19104.*
- *R. Conn is with the Department of Electrical and Computer Engineering, Drexel University, Philadelphia, PA 19104.*

Manuscript received 8 June 1995; revised 18 Dec. 1997. Recommended for acceptance by K. Boyer.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 106194.

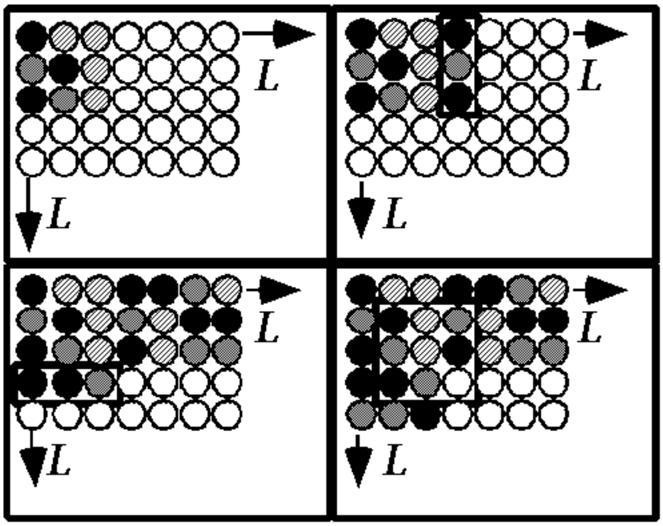

Fig. 1. Code generation. The matrix elements are represented here by gray values (black dot, letter 1; gray dot, letter 1; hatched dot, letter 3; and white dot, unassigned).

codes are generated by iterating between adding one new code word and checking it against all preceding ones. Pattern growth at the top and left margins of the matrix consists of a random assignment of letters from the available palette for each of the three positions adjoining the established pattern (Fig. 1 top right and bottom left). This action generates one new $V_{(ij)}$. For the rest of the matrix, a new $V_{(ij)}$ is generated by the addition of a single point to the pattern (Fig. 1 bottom right). In all cases, the Hamming distance between the new code word and all of those previously embedded in the matrix is computed. If all of these distances are equal to or greater than the preset minimum Hamming distance h, the new code is accepted and the algorithm proceeds with the next iteration. If not, a new letter assignment is generated. If, in a particular iteration, all possible letter assignments are exhausted (i.e., none were found to have a sufficient Hamming distance), the matrix is cleared and the process started over from the beginning.

Using the above method, 1,000 trials were executed in an attempt to generate a 45×45 size array for various values of h, A, and w. The values tested were all combinations of A, w, and h where $h = 1$ to 4; $A = 3$ to 9; $w = 3$ to 5. The search results for all of these cases are given in Table 1 in the Appendix. With window size restricted to $w = 3$, we required $A \geq 8$ to achieve $h \geq 3$, while to achieve $h \geq 1$, we required $A \geq 3$. Although not shown in the table, it was possible to obtain a minimum Hamming distance of five, but only with $w = 5$ and $A = 6$.

2.3 Correspondence Assignment in PSM

Our SL system consists of a CCD camera and an LCD or a slide projector. The projector is used to illuminate the object with a two dimensional, PSM encoded dot array. The camera located at a second vantage point captures the scene and digitized images are fed to a computer. Fig. 2a shows an experiment where a step is put in the scene which hides part of the projected pattern from the camera. SL using a projected grid with a fixed sampling frequency may fail in this setting (Fig. 2b). Since perfect submap based SL uses only local information this hidden step does not substantially effect performance. This capability comes from the coding properties of the projected pattern as discussed next.

Analysis starts with extraction of the coordinates of the dots' centroids from the captured images. For each dot, the eight closest neighbors are found, forming a 3×3 window. Each dot also carries a property, like a color or a temporal on-off pattern which

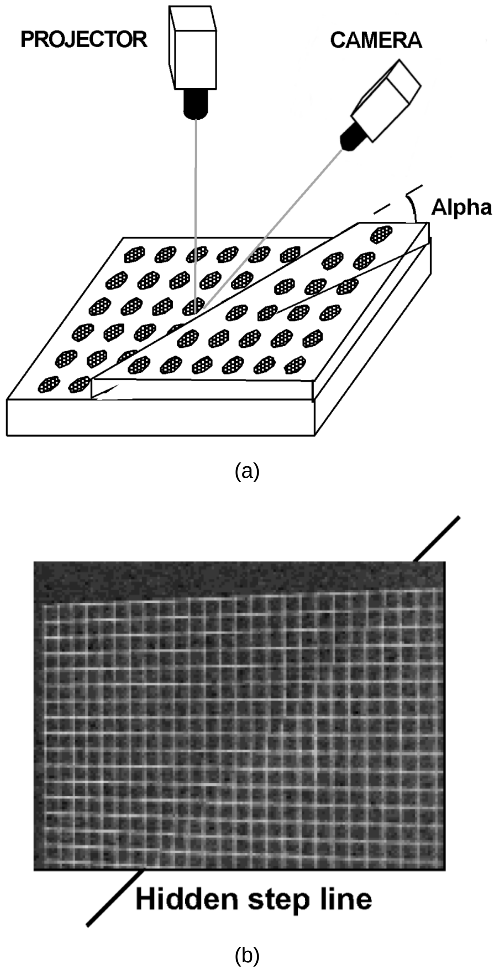

Fig. 2. (a) Structured light setup and an example of occlusion. Part of the scene illuminated with the SL pattern is not visible to the camera. In this image, a diagonal step (alpha = 45°) is present in the scene. (b) Structured light using grid illumination. Using a grid on the setup shown in (a) may not detect the occlusion step.

gives it a letter assignment. As a result, each window generates a nine-letter code word which can be used to index its projection coordinates. This correspondence process varies slightly when different Hamming distance codes are used.

In a Hamming distance one map, once a 3×3 window is established and its location in the projected code is found, all of its nine elements can be tentatively assigned a correspondence. Each dot will be assigned nine times since it is a member of nine different but overlapping 3×3 windows. These nine assignments can be corroborative or contradictory. The extent of agreement for a particular match defines the confidence level of that correspondence. Typically, for a planar object at the middle of the field of view, all points are matched with a confidence of nine, while at the sides and corners, the confidence levels will be lower (Fig. 3a).

In the Hamming distance one case, when a point in the center of the field of view is not detected, the confidence level of its neighbors decreases, but the effect remains localized (Fig. 3b). If that point is detected but its property (e.g., color) is misidentified, every window containing that false letter will be mislabeled. Mislabeling of a window results in a false assignment of all of its elements. At the end, a given element may receive its assignments from both correctly labeled and mislabeled windows. While the former are corroborative, the latter are not. By tabulating the assignments for each dot, the correspondence with the highest confidence value is determined. If that value

1	2	3	3	3	3	3	3	3	2	1
2	4	6	6	6	6	6	6	6	4	2
3	6	9	9	9	9	9	9	9	6	3
3	6	9	9	9	9	9	9	9	6	3
3	6	9	9	9	9	9	9	9	6	3
3	6	9	9	9	9	9	9	9	6	3
3	6	9	9	9	9	9	9	9	6	3
3	6	9	9	9	9	9	9	9	6	3
3	6	9	9	9	9	9	9	9	6	3
2	4	6	6	6	6	6	6	6	4	2
1	2	3	3	3	3	3	3	3	2	1

(a)

.	.	.	.	.	.	.	.	.	.	.
.	9	9	9	9	9	9	9	9	9	.
.	9	9	9	9	9	9	9	9	9	.
.	9	9	8	7	6	7	8	9	9	.
.	9	9	7	5	3	5	7	9	9	.
.	9	9	6	3	0	3	6	9	9	.
.	9	9	7	5	3	5	7	9	9	.
.	9	9	8	7	6	7	8	9	9	.
.	9	9	9	9	9	9	9	9	9	.
.	9	9	9	9	9	9	9	9	9	.
.	.	.	.	.	.	.	.	.	.	.

(b)

.	.	.	.	.	.	.	.	.	.	.
.	81	81	81	81	81	81	81	81	81	.
.	81	81	81	81	81	81	81	81	81	.
.	81	81	73	65	57	65	73	81	81	.
.	81	81	65	49	33	49	65	81	81	.
.	81	81	57	33	9	33	57	81	81	.
.	81	81	65	49	33	49	65	81	81	.
.	81	81	73	65	57	65	73	81	81	.
.	81	81	81	81	81	81	81	81	81	.
.	81	81	81	81	81	81	81	81	81	.
.	.	.	.	.	.	.	.	.	.	.

(c)

.	.	.	.	.	.	.	.	.	.	.
.	9	9	9	6	3	3	6	9	9	.
.	9	9	9	6	3	3	6	9	9	.
.	9	9	9	6	3	3	6	9	9	.
.	9	9	9	6	3	3	6	9	9	.
.	9	9	9	6	3	3	6	9	9	.
.	9	9	9	6	3	3	6	9	9	.
.	9	9	9	6	3	3	6	9	9	.
.	9	9	9	6	3	3	6	9	9	.
.	9	9	9	6	3	3	6	9	9	.
.	.	.	.	.	.	.	.	.	.	.

(d)

.	.	.	.	.	.	.	.	.	.	.
.	9	9	9	9	9	6	3	1	1	.
.	9	9	9	9	6	3	1	1	3	.
.	9	9	9	6	3	1	1	3	6	.
.	9	9	6	3	1	1	3	6	9	.
.	9	6	3	1	1	3	6	9	9	.
.	6	3	1	1	3	6	9	9	9	.
.	3	1	1	3	6	9	9	9	9	.
.	1	1	3	6	9	9	9	9	9	.
.	1	3	6	9	9	9	9	9	9	.
.	.	.	.	.	.	.	.	.	.	.

(e)

.	.	.	.	.	.	.	.	.	.	.
.	9	9	9	9	9	8	7	6	4	.
.	9	9	9	8	7	6	4	2	1	.
.	9	8	7	6	4	2	1	1	2	.
.	7	6	4	2	1	1	2	4	6	.
.	4	2	1	1	2	4	6	7	8	.
.	1	1	2	4	6	7	8	9	9	.
.	2	4	6	7	8	9	9	9	9	.
.	6	7	8	9	9	9	9	9	9	.
.	8	9	9	9	9	9	9	9	9	.
.	.	.	.	.	.	.	.	.	.	.

(f)

Fig. 3. Correspondence assignments in PSM based SL. (a) Decreased confidence numbers at the edges and corners of a projected 9 × 9, Hamming 1 array. (b) Decreased confidence numbers around a missing point in an h = 1 array. The 9 × 9 arrays in (b) through (f) are taken from a center area of a larger projected array, so edge and corner effects are not present. (c) Confidence numbers around a misidentified central point in an h = 3 array. (d) Confidence numbers around a perpendicular occlusion, Hamming distance of the array is 1. (e) Confidence numbers around the occlusion placed at 45° (h = 1). (f) Confidene numbers around the occlusion placed at 60° (or 30°), h = 1.

exceeds the next highest confidence value by a fixed certainty factor then the correspondence with the highest value is assigned to the dot. If it does not, that dot is left unlabeled.

In Hamming distance three case, each code word is separated by at least three letters from all others. In other words, each code in a Hamming distance three map has a unique Hamming distance one basin (attractor). This basin allows correct window identification even if one of the letters is incorrect or missing. In our SL implementations with Hamming distance three PSM codes, we label each 3 × 3 window nine times, each time by disregarding one of its nine elements. If there are no errors in any of the elements, all of them will be assigned a confidence of nine from this window's label. If there is a one element error, a window containing the erroneous element can be correctly labeled only once out of the nine assignments. As before, each point is also a member of nine different windows, so in the case of Hamming distance three, each point's confidence can be as high as 81 (nine different windows, each window assigned nine times). In the case of a misidentified center dot, we get a confidence of nine for the correct correspondence candidate for that dot, since all the nine windows which it is a member of can be assigned once correctly (Fig. 3c). The remaining 72 assignments will yield low confidence candidates.

The effects of occlusion on confidence levels can be seen in Figs. 3d, 3e, and 3f for the case of Hamming distance one. The arrangement of erroneous elements in windows straddling the occlusion depends on the angle of the discontinuity. In Figs. 3d, 3e, and 3f, the number of correct windows is shown for every element. This also equals the confidence values of correct correspondences. For points near the occlusion, there will also be incorrect correspondence assignments propagated from false window labels.

Depending on the orientation of the occlusion in relation to the projected code, the Hamming properties of the code, and the certainty factor, the percentage of successfully matched points will differ. These dependencies will be examined in an experimental setting next. In the real world, missing or misidentified points, corner and edge effects and occlusion problems may overlap and may result in dots left unlabeled. One has to choose the code properties and the certainty level to optimize sensitivity and specificity of correspondence, tailoring the SL system to the individual needs.

3 Experimental Results

To evaluate performance of the PSM-based SL, we imaged the scene illustrated in Fig. 2a. It consists of an occlusion step positioned at three different angles to the horizontal (alpha = 90°, 60°, and 45°). For each angle, part of the projected array on the lower plane was occluded from the camera. Three different types of binary temporal coding sequences were used in the experiment:

1) A classical 2D binary sequence (CBS) where no neighborhood relations are employed and each point is uniquely identified by its own temporal on-off pattern of illumination, as in Altshuler et al. [1],
2) Hamming distance one PSM, and
3) Hamming distance three PSM.

The images were processed and a correspondence was established for each type of coding sequence at all three angles of the step. The correspondence efficiency ratio is defined as, the number of sample points correctly identified relative to the number of visible projected points. The ratios for 60, 45, and 90 degrees of occlusions for Hamming 1 and certainty factor two were 366/431 (85 percent), 373/430 (87 percent), and 387/428 (90 percent), respectively. The ratios increased when a hamming distance three code has been used: 402/431 (93 percent), 417/430 (97 percent), 421/28 (98 percent). All of the assigned correspondences were correct. The unassigned points includes points lost due the edge and corner effects.

The PSM based structured light has high efficiency (85 percent or higher), but not as high as CBS (100 percent). In CBS, $\log_2(N) + 1$ images have to be captured in order to uniquely label N beamlets. In our noncoplanar setting, CBS acquisition needed a minimum of 10 slides (nine-bit code for each point and, as will be explained below, one sampling frame), while imaging with a Hamming 1 PSM can be achieved with three slides and a Hamming 3 PSM employs only four temporal slides. In each of these cases, one frame, the sampling frame, is used in the extraction of centroids. For this frame, all dots are illuminated. The use of a sampling frame provides some tolerance for subject movement between the different temporal slides. The remaining slides are indexing frames with each letter defined by the temporal sequence of illumination. Thus, a four-frame temporal code consists of three letter indexing frames and one sampling frame, and these three indexing frames can accommodate up to $8(= 2^3)$ letters.

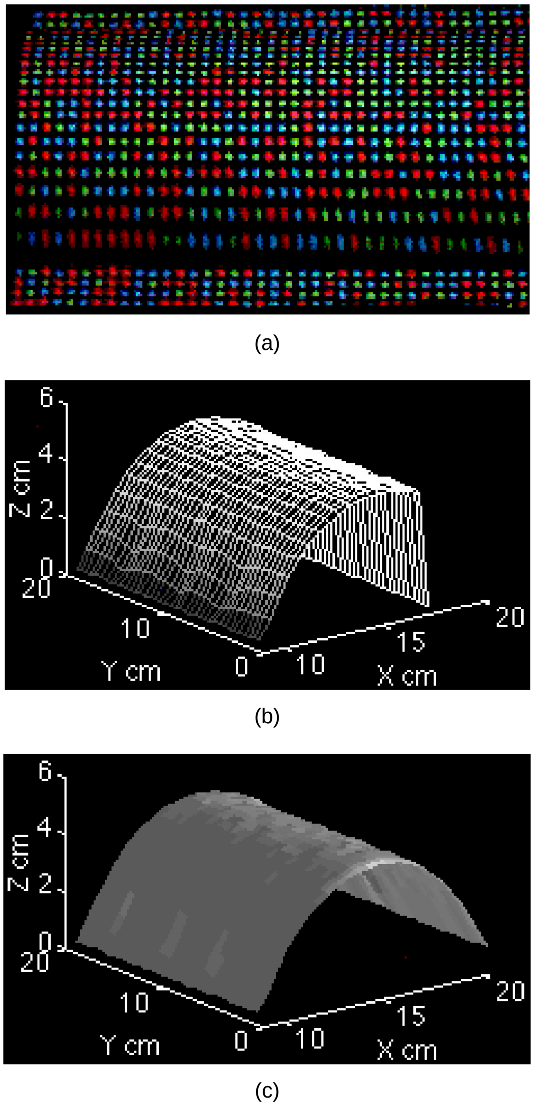

Fig. 4. Color encoded PSM. A hemicylinder was illuminated by slide projector from above containing a 35-mm slide with a 45 × 45, Hamming 1, three-color PSM dot array. The view from a color camera mounted lateral to the projector is shown in (a) The computed surface (b) is incomplete due to difficulty in color dot segmentation in receding surface. A second camera provided a better vantage view of that section of the hemicylinder and combination of the points extracted from both views and surface fitting yielded amore complete rendition of the object (c).

3.1 Color Code Implementation

An attractive implementation of PSM employs color encoding. The number of image frames required can then be reduced further. If a three-chip camera is employed, a single frame may suffice. We implemented a code generated using the search algorithm described above (L = 45, w = 3, A = 3, and h = 1) as a red, blue, and green dot array on a 35-mm slide. In Fig. 4a, projection of this code on a simple geometric object, a hemicylinder, is illustrated. From such a single frame both indexing information as well as centroid locations can be obtained. At the receding top edge of the cylinder, labeling efficiency decreases (Fig. 4b). Nonetheless, due to the local nature of PSM coding, this has no global consequences. If mapping a more complete surface is desired, views from multiple cameras may be readily combined (Fig. 4c).

TABLE 1
THE RESULTS OF THE PSEUDORANDOM SEARCH ALGORITHM FOR A MATRIX WITH L = 45

H	W		a=3	a=4	a=5	a=6	a=7	a=8	a=9
1	3	Average	45.3	100.0	100.0	100.0	100.0	100.0	100.0
		Number Completed	1	1000	1000	1000	1000	1000	1000
		Maximum Completed	100.0	100.0	100.0	100.0	100.0	100.0	100.0
1	4	Average	98.6	100.0	100.0	100.0	100.0	100.0	100.0
		Number Completed	936	1000	1000	1000	1000	1000	1000
		Maximum Completed	100.0	100.0	100.0	100.0	100.0	100.0	100.0
1	5	Average	99.8	100.0	100.0	100.0	100.0	100.0	100.0
		Number Completed	997	1000	1000	1000	1000	1000	1000
		Maximum Completed	100.0	100.0	100.0	100.0	100.0	100.0	100.0
2	3	Average	2.9	16.5	44.5	75.4	91.3	96.8	98.7
		Number Completed	0	0	28	371	747	902	964
		Maximum Completed	14.0	58.8	100.0	100.0	100.0	100.0	100.0
2	4	Average	22.0	77.9	96.5	99.5	99.7	99.9	100.0
		Number Completed	0	399	879	981	993	997	1000
		Maximum Completed	57.1	100.0	100.0	100.0	100.0	100.0	100.0
2	5	Average	73.1	98.7	99.8	100.0	100.0	100.0	100.0
		Number Completed	216	955	996	1000	1000	1000	1000
		Maximum Completed	100.0	100.0	100.0	100.0	100.0	100.0	100.0
3	3	Average	0.2	1.8	5.5	12.4	22.7	37.8	55.0
		Number Completed	0	0	0	0	0	6	97
		Maximum Completed	1.6	9.1	22.6	50.6	75.9	100.0	100.0
3	4	Average	4.6	18.8	44.8	62.7	68.1	71.7	57.6
		Number Completed	0	0	24	429	764	922	981
		Maximum Completed	14.8	63.5	100.0	100.0	100.0	100.0	100.0
3	5	Average	20.0	68.6	96.3	99.5	99.7	100.0	100.0
		Number Completed	0	179	865	981	994	1000	1000
		Maximum Completed	52.6	100.0	100.0	100.0	100.0	100.0	100.0
4	3	Average	0.0	0.0	0.6	1.7	3.3	5.8	8.8
		Number Completed	0	0	0	0	0	0	0
		Maximum Completed	0.0	1.0	4.0	8.7	17.0	23.4	34.6
4	4	Average	0.8	4.9	12.1	23.6	41.3	63.0	80.9
		Number Completed	0	0	0	0	5	145	448
		Maximum Completed	3.9	16.1	35.5	64.3	100.0	100.0	100.0
4	5	Average	8.5	23.0	53.3	86.5	96.0	98.7	99.5
		Number Completed	0	0	31	538	857	960	981
		Maximum Completed	19.6	62.8	100.0	100.0	100.0	100.0	100.0

A difficulty with color encoding of PSM arises if the imaged subject has complex reflectivity properties. Color identification may then be compromised. This is particularly severe if more than three colors are employed as would be required for codes with Hamming distance greater than one.

4 SUMMARY AND CONCLUSION

The structured light coding scheme discussed here simplifies the identification of points in the illumination pattern (the correspondence problem). Because it employs only local information, the labeling of the projection pattern is less sensitive to missing points than other previously described spatial codes such as the ones which have used grids [7]. Labeling of the grid intersection points can be time consuming and complex, especially if parts of lines are occluded. Although they only require one image to uniquely label many points, it requires a depth first search in which each new label is dependent on previously labeled points.

Our method does not depend on the acquisition of as many images as do classical binary temporal codes [1], so it can be better suited for applications where movement is a problem and a small reduction in the sampled points can be tolerated. Furthermore, the technique offers the possibility of employing neighborhood-level error detection and correction based only on the local characteristics of the code, similar to the approach taken by Boyer and Kak [5]. Their approach relies on local information and attempts to index individual stripes by a "crystal growing" process where subpatterns of stripes act as seeds to the stripe indexing decision process. Although each subpattern is unique and the total code is formed by putting them side by side, the uniqueness of the overlapping codes is not assured. Pseudorandom arrays eliminate this problem since all code words must be unique. In addition, our two dimensional codes allow the flexibility of working in noncoplanar settings and require less number of colors (or other attributes) for a given horizontal resolution. Unlike the crystal growing method, correspondence assignments with PSM are accomplished without iterating. In the scenes where color imaging is feasible, our technique has the potential of capturing the 3D information using only a single color frame.

APPENDIX

TABLE 1

The results of the pseudorandom search algorithm for a matrix with $L = 45$. Three parameters summarize the 1,000 searches for a given h and A: Number completed, Maximum completed and Average. Number completed gives the number out of 1,000 trials which were successful. Maximum completed will be 100 if a full code is generated. Otherwise, it gives the maximum percentage of the matrix that was filled by the algorithm. The average of the filling percentage of the matrix in all of the 1,000 attempts is given in Average. Other tables for $L = 23$ and $L = 35$ can be found in [11].

ACKNOWLEDGMENTS

We wish to thank the anonymous reviewers for their helpful comments. This work was supported by a grant from Ross Laboratories, a division of Abbott Laboratories, Columbus, Ohio, and NIH award P41RR01638, a Biomedical Technology Resource grant to the Computer Vision Center for Vertebrate Brain Mapping.

REFERENCES

[1] M.D. Altschuler, K. Bae, B.R. Altschuler, J.T. Dijak, L.A. Tamburino, and B. Woolford, “Robot Vision by Encoded Light Beams,” *Three Dimensional Machine Vision*, T. Kanade, ed. Norwell, Mass.: Kluwer Academic Publishers, 1987.

[2] I. Amir and F.P. Higgins, “3D Line-Scan Intensity Ratio Sensing,” *Proc. Optics, Illumination, and Image Sensing for Machine Vision IV*, SPIE-1614, Nov. 1991.

[3] D.H. Ballard and C.M. Brown, *Computer Vision*. Englewood Cliffs, N.J.: Prentice Hall, 1982.

[4] S.T. Barnard and M.A. Fischler, “Computational Stereo,” *ACM Computing Surveys*, vol. 14, no. 4, Dec. 1982.

[5] K.L. Boyer and A.C. Kak, “Color-Encoded Structured Light for Rapid Range Sensing,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 9, no. 1, pp. 14-28, Jan. 1987.

[6] B. Carrihill and R. Hummel, “Experiments With the Intensity Ratio Depth Sensor,” *Computer Vision Graphics Image Processing*, vol. 32, pp. 337-358, 1985.

[7] S.M. Dunn, R.L. Keizer, and J. Yu, “Measuring the Area and Volume of the Human Body With Structured Light,” *IEEE Trans. Systems, Man, and Cybernetics.*, vol. 19, no. 6, pp. 1,350-1,364, 1989.

[8] T. Etzion, “Constructions for Perfect Maps and Pseudorandom Arrays,” *IEEE Trans. Information Theory*, vol. 34, pp. 1,308-1,316, 1988.

[9] R.A. Jarvis, “A Perspective on Range Finding Techniques for Computer Vision,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 5, no. 2, pp. 122-139, Mar. 1983.

[10] V. Llario and B. Martinez, “Active Methods for Obtaining Depth Maps,” *Computer Vision Theory and Industrial Applications*. Berlin: Springer-Verlag, 1992.

[11] R.A. Morano, “Noninvasive Measurement of Body Composition,” M.S. thesis, Drexel Univ., Philadelphia, Pa., 1994.

[12] C. Ozturk, E. Schmutz, and J. Nissanov, “Decoding of Multidimensional Perfect Maps,” *Proc. Am. Math. Soc. Conf.*, Md., in press.

[13] K.G. Paterson, “Perfect Maps,” *IEEE Trans. Information Theory*, vol. 40, no. 3, pp. 743-753, 1994.

[14] M.A. Penna and R.R. Patterson, *Projective Geometry and Its Application to Computer Graphics*. Englewood Cliffs, N.J.: Prentice Hall, 1986.

[15] W.W. Peterson and E.J. Weldon, Jr., *Error-Correcting Codes*, 2nd ed. Cambridge, Mass.: MIT Press, 1972.

[16] Y.F. Wang, A. Mitiche, and J.K. Aggarwal, “Computation of Surface Orientation and Structure of Objects Using Grid Coding,” *IEEE Trans. Pattern Analysis and Machine Intelligence.*, vol. 9, no. 1, pp. 129-137, 1987.