Pattern Recognition, Vol. 25, No. 6, pp. 609–616, 1992
Printed in Great Britain

0031-3203/92 $5.00 + .00
Pergamon Press Ltd


# GENERATION OF UNIQUELY ENCODED LIGHT PATTERNS FOR RANGE DATA ACQUISITION

PAUL M. GRIFFIN, LAKSHMI S. NARASIMHAN and SOUNG R. YEE
Georgia Institute of Technology, Atlanta, GA 30332, U.S.A.

(*Received* 7 *March* 1991; *in revised form* 12 *September* 1991; *received for publication* 20 *September* 1991)

**Abstract**—A method is presented for generating an encoded pattern which may be used for a special structured lighting system. This pattern consists of a matrix of colored circles which is projected onto a scene by backlighting. A single camera is used to image the projected colored light beams, and range data for the object are obtained. Because the light pattern is encoded, the correspondence between the projected beams from the pattern and imaged beams is easily determined, and hence range data may be solved for quickly. Further, this method may be used in dynamic environments.

Correspondence problem Computer vision Range data Structured light Finite automata

## 1. INTRODUCTION

Computer vision is concerned with the automatic acquisition and analysis of visual information. This visual information may be transformed into many different forms, but one of the most useful for the unambiguous description of three-dimensional (3D) objects is range data. Range data consist of a set of spatial coordinates of the observed object surface, and may be used directly for object matching and recognition.

Structural lighting systems are a common method of acquiring surface range data from a scene. An object is illuminated from a structured light source (which we call the *active* image), and this scene is imaged by a camera (which we call the *passive* image). The structured lighting system considered in this paper consists of projecting a matrix of discrete colored light beams, as shown in Fig. 1. If the coordinate reference frame is object centered, then a transformation from the reference frame to each of the image frames is described by the following:

$$\mathbf{A}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{1}$$

where

$$\mathbf{A} = \begin{bmatrix} T_{11} - T_{13}u & T_{21} - T_{23}u & T_{31} - T_{33}u & T_{41} - T_{43}u \\ T_{12} - T_{13}v & T_{22} - T_{23}v & T_{32} - T_{33}v & T_{42} - T_{43}v \\ L_{11} - L_{13}i & L_{21} - L_{23}i & L_{31} - L_{33}i & L_{41} - L_{43}i \\ L_{12} - L_{13}j & L_{22} - L_{23}j & L_{32} - L_{33}j & L_{41} - L_{43}j \end{bmatrix}$$

**T** is the transformation between the object and passive image, **L** the transformation between the object and active image, $(x, y, z)$ the object range point, $(u, v)$ the corresponding imaged point on the passive image and $(i, j)$ the corresponding point projected from the active image. The values for **T** and **L** may be found using a calibration object.[1]

For an imaged beam point $(u, v)$ in the passive image, if the corresponding beam $(i, j)$ projected from the active image is known, then the range value $(x, y, z)$ on the object surface may be determined by solving equation (1). Since the system of linear equations is overdetermined (four equations and three unknowns), the system may be solved by a best fit method.[1] Determining the corresponding $(i, j)$ for each $(u, v)$ is in general, however, a very difficult task. This is primarily due to the fact that multiple correspondences may occur and so the assignment is ambiguous.

In an effort to simplify this correspondence problem, many researchers have tried to *encode* the projected light pattern. Altschuler *et al.*[1] and Posdamer and Altschuler[2] use a laser shutter/space encoding technique. In this method a grid of 128 × 128 laser beams is projected onto the scene through a programmable spatial light modulator (PSLM)[3] to form a dot pattern on the scene. The PSLM may be programmed to allow certain laser beams through while blocking the passage of others. Correspondence is solved by taking a sequence of images with distinct beam patterns. The sequence is a binary partition, and hence for the 128 × 128 pattern, 15 images are required. An alternative time-sequential encoding scheme using the Gray Code

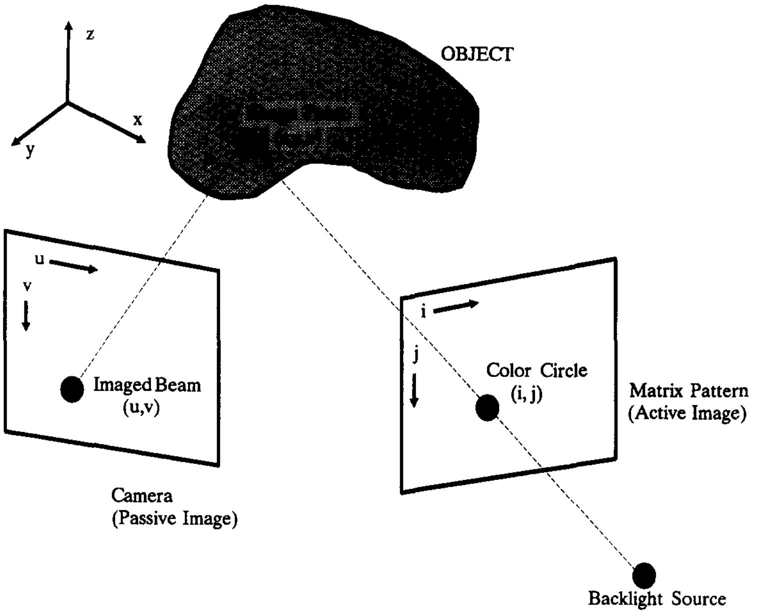


Fig. 1. Structured lighting system where colored circles are projected onto a scene by backlighting and then imaged by a camera.

was developed by Inokuchi *et al.*[4] Both of these methods solve correspondence unambiguously for general 3D static scenes. These methods cannot be used, however, in dynamic environments (i.e. a scene with moving objects).

For dynamic scenes, it is desirable to have an encoding scheme in which correspondence may be solved from a single image. Boyer and Kak[5] use colored light stripes projected onto a scene. Initial codes are established based on adjacent light stripes. The stripes are indexed relative to the initial codes (called seed crystals) by a "crystal growing" process. The process merges stripes as long as they do not violate the overall pattern code. However, since this encoding scheme is based on adjacent subpatterns, small amounts of occlusion can greatly affect the performance. Vuylsteke and Oosterlinck[5] developed a system which projects a square checkerboard grid of black and white points onto the scene which is a binary-encoded light pattern. A binary signature is defined for each point by looking at the values of the neighbors in a window about that point. The encoded light pattern is generated from pseudonoise sequences (which generate the 0 and 1 values for the pattern recursively to define which points are black and which are white). The authors "optimize" the pattern in the sense that the pattern is fault-tolerant to the largest extent. That is, a minimal code distance, $d$, may be specified between nonunique signatures. Since a binary signature is defined in a small window, this method is not as susceptible to occlusion as in the method proposed by Boyer and Kak. However, for a general $n \times m$ pattern it is still possible for multiple correspondences to exist since the pseudonoise sequences do not guarantee uniqueness, but only that nonunique signatures be at least a distance $d$ apart. Further, complete information of the coding primitives is not used. Finally, for a large binary pattern, large windows must be searched for the binary signatures.

In this paper, we present a method which uses an encoded colored light pattern for obtaining range data. This method is not as susceptible to problems with occlusion as in reference (5) since like the method in reference (3), only a small search window is used. Further, this method is more general than that proposed in reference (3) since (i) it is not restricted to binary patterns, (ii) any size pattern may be generated, (iii) it is guaranteed that each position defined in the pattern is unique (i.e. it is completely fault tolerant). In the next section, the notation is defined, the encoded pattern derived is presented, and properties of this pattern are discussed. Section 3 gives encoding examples. Conclusions and future research are discussed in Section 4.

## 2. ENCODING SCHEME

In this section, we show how to assign colors to the light pattern so that the projected pattern is encoded. Let $P$ be a set of color primitives, $P = \{1, 2, \ldots, p\}$ (for instance, 1 = red, 2 = green, 3 = blue, etc.). These color primitives are assigned to an $m \times n$ matrix $M$ to form the encoded pattern which may be projected onto the scene to solve for the range data. We define a *word* from $M$ by the color value at location $(i, j)$ in $M$ and the color values of its 4-adjacent neighbors. If $x_{ij}$ is the assigned color at row $i$ and column $j$, then the word for this location,

| | | | | | |
|---|---|---|---|---|---|
| 1 | 3 | 2 | 1 | 1 | 2 |
| 2 | 2 | 2 | 3 | 1 | 3 |
| 1 | 3 | 3 | 1 | 1 | 2 |
| 1 | 1 | 2 | 3 | 2 | 2 |
| 3 | 2 | 2 | 1 | 3 | 1 |
| 2 | 1 | 1 | 3 | 2 | 3 |

Word = 33212

Fig. 2. A word (33212) defined in matrix $M$ by the color at location $(i, j)$ and the color values of the 4-adjacent neighbors around $(i, j)$.

$w_{ij}$, is the sequence $\{x_{ij}, x_{i,j-1}, x_{i-1,j}, x_{i,j+1}, x_{i+1,j}\}$ where $i \in \{1, 2, \ldots, m\}$ and $j \in \{1, 2, \ldots, n\}$ (see Fig. 2). If a lookup table is maintained for all of the word values in $M$, then each word defines a location in $M$.

We wish to assign the color primitives of $P$ to the matrix $M$ so that no two words in the matrix are the same. In this way, each defined location is unique and hence correspondence will be unique. That is, if the pattern is projected onto a scene, and the word value for an imaged point $(u, v)$ is determined (by determining the color of that imaged point and the colors of its imaged 4-adjacent neighbors), then the corresponding position $(i, j)$ in $M$ of this imaged point is uniquely defined.

In addition to having each word of $M$ be unique, we also want to make the color assignments so that matrix $M$ is as large as possible. Since each word $w_{ij}$ is made up of five color primitives, then the largest number of words that can be present in $M$ is $|P|^5$. We provide a matrix $M$ with $|P|^3 + 2$ columns and $|P|^2 + 2$ rows that forms each of these $|P|^5$ words exactly once, and thus is maximal. Note that since a word is defined by the color at a position $(i, j)$ in $M$ and the colors of the four adjacent positions to $(i, j)$, then there is the additional two columns and two rows which are needed.

We begin by examining a one-dimensional problem. Specifically, given a set of color primitives, $P = \{1, 2, \ldots, p\}$, we wish to construct a string of $|P|^3 + 2$ colors such that each triplet of adjacent colors in the string is distinct from every other triplet of adjacent colors. We call this string the maximal

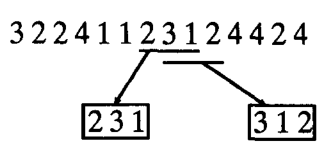


Fig. 3. Two adjacent triplets defined on a horizontal string. Since the triplets overlap, the two color primitives on the first triplet (31) are the first two color primitives of the adjacent triplet.

horizontal string. Since there is overlap of the triplets in the string, independence of the triplets does not hold. For instance, suppose that one triplet has value 231, then the next triplet to the right of it must have value $31x$ (see Fig. 3).

Define $\{f_h(p)\}$ to be the following sequence:

$$\begin{aligned}
\{f_h(p)\} = \{&pp1, p(p-1)1, \ldots, p11,\\
&(p-1)p1, \ldots, (p-1)11,\\
&, \ldots,\\
&2p1, \ldots, 211, 1\},\\
\{&pp2, p(p-1)2, \ldots, p22,\\
&(p-1)p2, \ldots, (p-1)22,\\
&, \ldots,\\
&3p2, \ldots, 322, 2\},\\
&.\\
&.\\
&.,\\
\{&pp(p-1), p(p-1)(p-1), p-1\},\\
\{&ppp\}.
\end{aligned}$$

For instance, if $P = \{1, 2, 3, 4\}$ then $\{f_h(4)\} = 441$, 431, 421, 411, 341, 331, 321, 311, 241, 231, 221, 211, 1, 442, 432, 422, 342, 332, 322, 2, 443, 433, 3, 444. Removing the commas from the sequence forms a horizontal string. We can define each row $i$ specified between the brackets { } in $\{f_h(p)\}$ as substring $s_h^i(p)$. For the case of $\{f_h(4)\}$ there are four substrings: $s_h^1(4) = 4\,4\,1\,4\,3\,1\,4\,2\,1\,4\,1\,1\,3\,4\,1\,3\,3\,1\,3\,2\,1\,3\,1\,1\,2\,4\,1\,2\,3\,1\,2\,2\,1\,2\,2\,1\,1\,1$, $s_h^2(4) = 4\,4\,2\,4\,3\,2\,4\,2\,2\,3\,4\,2\,3\,3\,2\,3\,2\,2\,2$, $s_h^3(4) = 4\,4\,3\,4\,3\,3\,3$, and $s_h^4(4) = 4\,4\,4$.

*Theorem* 1. The sequence $\{f_h(p)\}$ defines a maximal horizontal string.

*Proof.* To show that the sequence $\{f_h(p)\}$ defines a maximal horizontal string is equivalent to showing that all triplets are found in the string with no redundancies.

The substring made from the first row in the sequence $\{f_h(p)\}$, $s_h^1(p)$, contains the triplets $xy1$ for all $x, y \in P$, the triplets $x1y$ for all $x, y \in P$ with the exception of $11p$, and the triplets $1xy$ for all $x, y \in P$ with the exception of $1pp$. However, $11p$ is found from the last two elements of $s_h^1(p)$ and the first element of $s_h^2(p)$, and $1pp$ is found from the last element of $s_h^1(p)$ and the first two elements of $s_h^2(p)$. Further, no other triplets containing a 1 are found. The substring made from the $n$th row in the sequence $\{f_h(p)\}$, $s_h^n(p)$, contains the triplets $xyn$ for all $x \in P\backslash\{1, 2, \ldots, n-1\}$ and $y \in P\backslash\{1, 2, \ldots, n-1\}$, the triplets $xny$ for all $x \in P\backslash\{1, 2, \ldots, n-1\}$ and $y \in P\backslash\{1, 2, \ldots, n-1\}$ with the exception of $nnp$, and the triplets $nxy$ for all $x \in P\backslash\{1, 2, \ldots,$

$n-1\}$ and $y \in P \backslash \{1, 2, \ldots, n-1\}$ with the exception of $npp$. However, $nnp$ is found from the last two elements of $s_h^n(p)$ and the first element of $s_h^{n+1}(p)$, and $npp$ is found from the last element of $s_h^n(p)$ and the first two elements of $s_h^{n+1}(p)$. All of the other triplets in $s_h^n(p)$ have been previously defined in substrings $s_h^1(p)$ to $s_h^{n-1}(p)$. Further, no other triplets containing $n$ are found in any subsequent rows. The substring made from the $p$th row in the sequence $\{f_h(p)\}$, $s_h^p(p)$, contains the triplet $ppp$. All other triplets $pxy$, $xpy$ and $xyp$ have been defined in the previous rows. Therefore, all triplets exist in the string with no redundancies. □

The maximal horizontal string may be used as an initial row for the generation of the matrix $M$ in the original problem. Theorem 1 shows that each triplet in this string is unique. Adding a constant to each element in this string creates a new string in which each triplet is also unique. We develop a string of color primitives (called the maximal vertical string) which is used to modify the maximal horizontal string to construct $M$. Before this construction is discussed, the generation of the maximal vertical string is presented.

Given the same set of color primitives used for the generation of the maximal horizontal string, $P = \{1, 2, \ldots, p\}$, we wish to assign them to a string of $|P|^2 + 1$ numbers such that each adjacent pair of colors in the string is distinct from every other pair of colors. Since there is overlap of the pairs, no adjacent pairs are independent. For instance pair 32 must be followed by pair $2x$ in the string.

Define $\{f_v(p)\}$ to be the following sequence:

$$\begin{aligned}\{f_v(p)\} = \{&p1, (p-1)1, \ldots, 21, 1\},\\ \{&p2, (p-1)2, \ldots, 32, 2\},\\ &\ldots,\\ \{&pp\}.\end{aligned}$$

For instance, if $P = \{1, 2, 3, 4\}$ then $\{f_v(4)\} = 41, 31, 21, 1, 42, 32, 2, 43, 3, 44$. Removing the commas from the sequence forms a vertical string (4 1 3 1 2 1 1 4 2 3 2 2 4 3 3 4 4). As in the case for the horizontal string, we can define each row $i$ contained in the brackets { } in $\{f_v(p)\}$ as substring $s_v^i(p)$. For the case of $\{f_v(4)\}$ there are four substrings: $s_v^1(4) = 4\,1\,3\,1\,2\,1\,1$, $s_v^2(4) = 4\,2\,3\,2\,2$, $s_v^3(4) = 4\,3\,3$, and $s_v^4(4) = 4\,4$.

*Theorem* 2. The sequence $\{f_v(p)\}$ defines a maximal vertical string.

*Proof.* To show that the sequence $\{f_v(p)\}$ defines a maximal vertical string is equivalent to showing that all pairs are found in the string with no redundancies.

The substring made from the first row in the sequence $\{f_v(p)\}$, $s_v^1(p)$, contains the pairs $x1$ for all $x \in P$. Further no other row contains a 1, and so there are no redundant $x1$ pairs. The substring made from the second row in the sequence, $s_v^2(p)$, contains the pairs $x2$ for all $x \in P \backslash 1$. However, 12 is contained in the first row. Further, no other $b2$ values are contained in the sequence. Continuing in this manner, pick the substring made from the $n$th row in the sequence, $s_v^n(p)$. This row contains all the pairs $xn$ for all $x \in P$ with the exceptions of $(n-1)n$, $(n-2)n, \ldots, 2n, 1n$. However, $1n$ is contained in the first row and is the only pair in this row ending in $n$, $2n$ is found in the second row and is the only pair in this row ending in $n$, and so forth up to $(n-1)n$ in the $(n-1)$ row. Since no $n$ values are found in rows below $n$, then there are no redundant pairs of the form $cn$. Finally, the substring made from the last row, $s_v^p(p)$, contains $pp$. All other $ip$ values for $i \in \{1, 2, \ldots, (p-1)\}$ are found by taking the last element of row $i$ and the first element of row $i+1$. Since $p$ is the first element of every row, and no other $p$ values exist, then there are no redundant $ip$ values. Therefore, all pairs exist in the string with no redundancies. □

The maximal vertical and horizontal strings generated by $\{f_h(p)\}$ and $\{f_v(p)\}$ can be used to form the $|P|^3 + 2$ by $|P|^2 + 2$ matrix $M$. Define $\{f_h(p)\} \otimes \{f_v(p)\}$ as the application of $\{f_v(p)\}$ to $\{f_h(p)\}$ in the following way. A first row is defined by $\{f_h(p)\}$. Create a second row by adding the first element of $\{f_v(p)\}$ to each element of $\{f_h(p)\}$ modulo $p$. Create a third row by adding the second element of $\{f_v(p)\}$ to each element of $\{f_h(p)\}$ modulo $p$. This process is repeated for each element of $\{f_v(p)\}$. The construction is a $|P|^3 + 2$ by $|P|^2 + 2$ matrix. For instance, $\{f_h(4)\} \otimes \{f_v(4)\}$ is shown in Fig. 4. Note that the modulo operation used in the construction is only on the set $\{1, 2, \ldots, p\}$ and does not include the 0 element as does the traditional modulo operation.

Since $p$ is finite, the matrix generation scheme explained above is in fact a finite automata. The color primitive values in the string $\{f_h(p)\}$ define initial states, and the values of $\{f_v(p)\}$ define the number of transitions (or "jumps") from a given state to the state to be determined.

From the first row $\{f_h(p)\}$, the second row is created by applying the number of transitions determined by the first element of $\{f_v(p)\}$. In other words, the second row consists of the values of the states, which are arrived at by applying a certain number of jumps to the states of the first row. The next number of transitions to be applied to the second row is the value of the second element of $\{f_v(p)\}$. This process is repeated until the entire sequence of $\{f_v(p)\}$ is applied.

The finite automata for $p = 3$ is illustrated in Fig. 5. If the color primitive value of a horizontal row is 2, and the value of 2 from $\{f_v(p)\}$ is applied, then the corresponding element of the next row is 1. We can now present the main result of the paper.

```
441431421411341331321311241231221211144243242234233232224434333444
441431421411341331321311241231221211144243242234233232224434333444
112142132122412442432422312342332322211314313341344343331141444111
441431421411341331321311241231221211144243242234233232224434333444
112142132122412442432422312342332322211314313341344343331141444111
334324314344234224214244134124114144433132131123122121113323222333
441431421411341331321311241231221211144243242234233232224434333444
112142132122412442432422312342332322211314313341344343331141444111
112142132122412442432422312342332322211314313341344343331141444111
334324314344234224214244134124114144433132131123122121113323222333
223213243233123113143133423413443433322421424412411414442212111222
441431421411341331321311241231221211144243242234233232224434333444
223213243233123113143133423413443433322421424412411414442212111222
223213243233123113143133423413443433322421424412411414442212111222
112142132122412442432422312342332322211314313341344343331141444111
441431421411341331321311241231221211144243242234233232224434333444
441431421411341331321311241231221211144243242234233232224434333444
441431421411341331321311241231221211144243242234233232224434333444
```

Fig. 4. The color primitive assignments for $\{f_h(4)\} \otimes \{f_v(4)\}$.

*Theorem* 3. The matrix formed by $\{f_h(p)\} \otimes \{f_v(p)\}$ contains the largest number of elements possible with the requirement that each word $w_{ij}$ be distinct.

*Proof.* The maximum number of distinct words possible for $P = \{1, 2, \ldots, p\}$ is $p^5$. These words could be contained in a $p^3 + 2$ by $p^2 + 2$ matrix. This is exactly the size of the matrix produced by $\{f_h(p)\} \otimes \{f_v(p)\}$. We have left to show that no two words in the matrix $M$ formed by $\{f_h(p)\} \otimes \{f_v(p)\}$ are the same.

By theorem 1, the first row of $M$ (which is $\{f_h(p)\}$) contains all distinct triplets. Since we apply the same number of jumps to every element of the row to obtain the next row, each of the triplets of the second row remain distinct. For this reason, if we pick any row in $M$, the row will have distinct triplets, and hence unique words on it. Suppose that there are two words, $W_1$ and $W_2$, in $M$ that are the same. Since each row has unique triplets, then these words must be on different rows.

Let $k_1 l_1 m_1$ be the vertical element of $W_1$, and $k_2 l_2 m_2$ be the vertical elements of $W_2$. Since we claim that $W_1$ and $W_2$ are the same, then $k_1 = k_2$, $l_1 = l_2$ and $m_1 = m_2$. Define $x^{(z)}$ to be the state after $z$ jumps from state $x$, and let $k_1 = k_2 \equiv k \in P$. If $ab$ and $cd$ are different substring pairs from $\{f_v(p)\}$, and applied to the components of $W_1$ and $W_2$, respectively. The following relationship can then be set up:

$$l_1 = k^{(a)} = l_2 = k^{(c)}$$

$$m_1 = [k^{(a)}]^{(b)} = k^{(a+b)} = m_2$$

where $a, b, c, d \in P$. By recognizing $k^{(np)} = k$, where $n \in I^+$, we obtain

$$|a - c| = n_1 p \tag{2}$$

$$|(a + b) - (c + d)| = n_2 p \tag{3}$$

where $n_1, n_2 \in I^+$.

Since $a, c \leq p$, then from equation (2), we obtain

$$0 \leq n_1 = |a/p - c/p| < 1.$$

However, the only possible value for $n_1$ which satisfies this relation is 0, which means that $a = c$. Similarly from equation (3) we get that it must be true that $b = d$. Therefore, $ab$ is the same as $bd$. This contradicts theorem 2. □

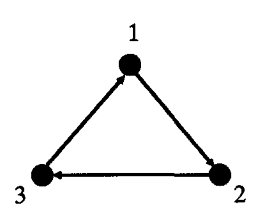


Fig. 5. Finite automata for $p = 3$.

### 3. COLORED LIGHT SYSTEM

In this section we discuss the method used for the colored light projection system. For the colored light pattern, we use three colors: red, green and blue. The size of matrix $M$, therefore, will be 11 rows by 29 columns. Applying the sequences presented in Section 2, we get the maximal horizontal string (3 3 1 3 2 1 3 1 1 2 3 1 2 2 1 2 1 1 1 3 3 2 3 2 2 2 3 3 3) and the maximal vertical string (3 1 2 1 1 3 2 2 3 3). Assigning red to 1, green to 2 and blue to 3, and applying the vertical string to the horizontal, we generate the light pattern shown in Fig. 6.

The availability of several vision systems which support color image processing, reduces the recognition of the color primitives to a relatively simple task. However, it is possible to make false color readings based on the color content in the scene. This approach works best for environments that are color neutral.

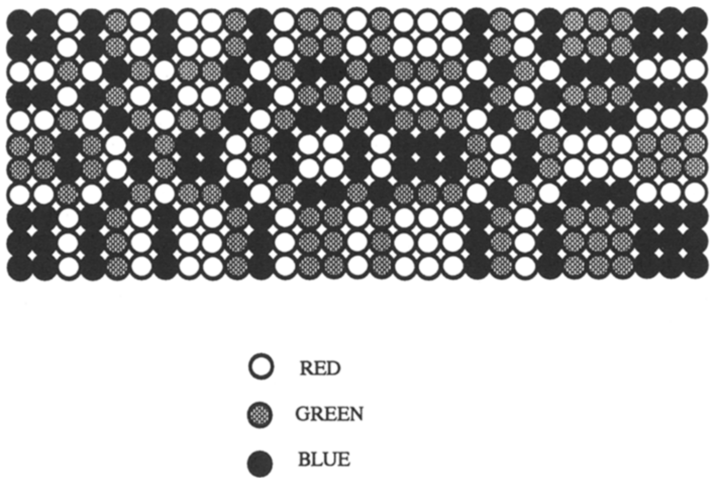


Fig. 6. Pattern $M$ used for the colored lighting system with the colors red, green and blue.

The method used for generating the color pattern is relatively simple. The pattern shown in Fig. 6 is generated on a computer screen using colored circles. A color photographic slide is then made of this pattern. The slide may be projected onto a scene by backlighting (such as a slide projector). Using a single camera (as long as the transformation between camera and projector is known) range data from the scene may be determined using equation (1). Figure 7(a) shows an object with the light pattern projected onto it and the beam locations found (note that in this figure, the colored points all appear black since it is a binary image of it). For each imaged point $(u, v)$, the color of its 4-adjacent neighbors is determined, and hence the word value of each point is found. From the word values, the location $(i, j)$ in $M$ of $(u, v)$ is determined. These values are plugged into equation (1) to solve for the range data values. In Fig. 7(b), an edge is drawn between two imaged points if their corresponding locations in $M$ are adjacent.

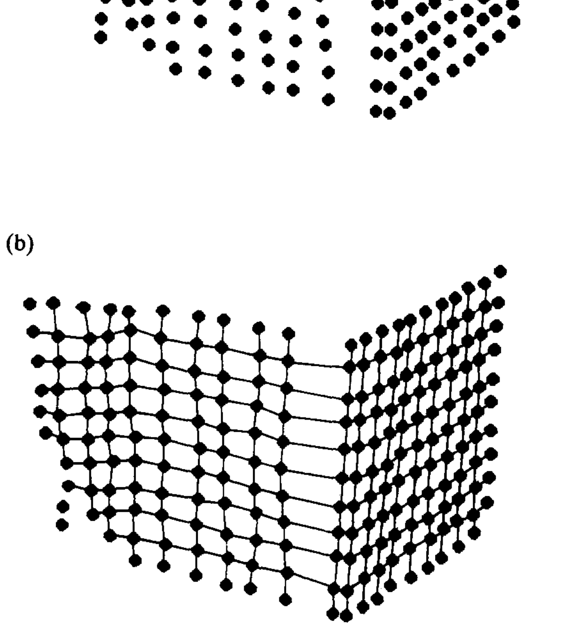


Fig. 7. (a) Binary image of the pattern from Fig. 6 projected onto an object. (b) Edges drawn between imaged points if these points are adjacent in $M$.

The resolution of the pattern may be increased by simply using more color primitives. Depending on the color discriminating capability of the system employed, almost any degree of resolution can be obtained.

In many applications, the scene may not be color neutral, or the vision system used may only have grey-scale capability. In this case, the structured light must be monochromatic. A possible encoding scheme in this case is to use a modified grid pattern where a primitive is assigned to each grid intersection point. For example, five grid primitives which may be used are shown in Fig. 8.

Since there are five grid primitives, the size of matrix $M$ is 27 rows by 127 columns. The horizontal and vertical strings are easily computed from the sequences given in Section 2. The horizontal string is (5 5 1 5 4 1 5 3 1 5 2 1 5 1 1 4 5 1 4 4 1 4 3 1 4 2 1 4 1 1 3 5 1 3 4 1 3 3 1 3 2 1 3 1 1 2 5 1 2 4 1 2 3 1 2 2 1 2 1 1 1 5 5 2 5 4 2 5 3 2 5 2 2 4 5 2 4 4 2 4 3 2 4 2 2 3 5 2 3 4 2 3 3 2 3 2 2 2 5 5 3 5 4 3 5 3 3 4 5 3 4 4 3 4 3 3 3 5 5 4 5 4 4 4 5 5 5) and the vertical string is (5 1 4 1 3 1 2 1 1 5 2 4 2 3 2 2 5 3 4 3 3 5 4 4 5 5).

Fig. 8. The five modified grid primitives.

Each time a grid primitive is assigned to $M$, its fault tolerant grid primitive is assigned to the same position. In this way, each set of five grid primitives defines two words. For verification, each word may be checked with its word pair. This greatly reduces the possibility of word identifications. This also makes this method much less susceptible to problems of occlusion than previously published methods. A portion of the light pattern generated in this way is shown in Fig. 9.

The method used for generating the light pattern for this case again involves producing the pattern on a computer screen and taking a photographic slide of this pattern. The slide is then projected onto the scene of interest and range data are obtained from a single camera.

## 4. CONCLUSIONS

In this paper, we have presented a method for generating an encoded colored light pattern for use in obtaining range data from a scene. For a given set of color primitives, the patterns generated are guaranteed to be the largest size possible for the restriction that each word in the pattern be unique. We have also shown how an alternative encoding scheme can be used to produce a modified grid pattern. We feel that in addition to the two coding schemes presented here, many other types of coded patterns can be developed depending upon the application. By using a light pattern generated from the method presented here, correspondence may be solved from a single image. Therefore, this scheme is applicable in dynamic environments such as in medical imaging and manufacturing. The authors have implemented both coding systems by using photographic slides of computer generated patterns projected onto the scene of interest.

As large a matrix as desired can be generated simply by using more primitive elements in the code. However, there is some price to be paid for increasing the size of $M$ since in general the problem of primitive recognition in the acquired images becomes more difficult as the number of primitive types increases. The issue of range data resolution versus ease of processing is very much application specific, and must be considered.

## 5. SUMMARY

This paper represents a method for generating an encoded light pattern which may be used for a special structured lighting system consisting of a matrix of colored light beams. This matrix of colored light may be projected onto a scene, and from a single camera, range data from the scene may be obtained. The pattern is encoded to allow the correspondence between projected beams and imaged beams to be determined unambiguously. By solving a system of linear equations with this correspondence, range data are determined.

Patterns are generated by assigning color primitives to elements of a matrix $M$. A word of $M$ is defined by the color at location $x_{ij}$ and its 4-adjacent neighbors. The color primitives are assigned in such a way that no two words in $M$ are the same. The

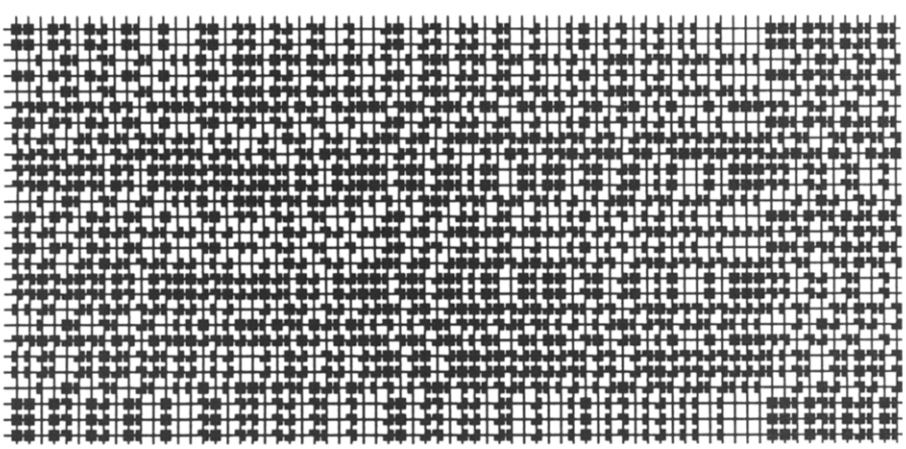

Fig. 9. Pattern $M$ used for the modified grid lighting system using the primitives shown in Fig. 8.

assignment of the color primitives to $M$ is based on two sequences presented in the paper. We prove that the matrix so defined contains the largest number of elements possible with the requirement that each word be distinct. Finally, an alternative structured lighting scheme is discussed using modified grid primitives as opposed to color primitives.

*Acknowledgements*—We gratefully acknowledge John Taboada for introducing us to this problem, and the anonymous reviewers for their helpful comments and suggestions.

## REFERENCES

1. M. D. Altschuler, B. R. Altschuler and J. Taboada, Laser electro-optic system for rapid three-dimensional (3-D) topographic mapping of surfaces, *Opt. Engng* **20**(6), 953–961 (1981).
2. J. Posdamer and M. D. Altschuler, Surface measurement by space-encoded projected beam systems, *Comput. Graphics Image Process.* **18**, 1–17 (1982).
3. J. Taboada, Laser-generated beam array for commutation of spatially modulated optical signals, *SPIE* **698**, 198–206 (1986).
4. S. Inokuchi, K. Sato and F. Matsuda, Range-imaging system for 3-D object recognition, *Proc. 7th Int. Conf. on Pattern Recognition*, pp. 806–808 (1984).
5. K. L. Boyer and A. C. Kak, Color-encoded structured light for rapid active ranging, *IEEE Trans. Pattern Analysis Mach. Intell.* **9**(1), 14–28 (1987).
6. P. Vuylsteke and A. Oosterlinck, Range image acquisition with a single binary-encoded light pattern, *IEEE Trans. Pattern Analysis Mach. Intell.* **12**(2), 148–164 (1990).

**About the Author**—PAUL M. GRIFFIN is Assistant Professor at the School of Industrial and Systems Engineering at the Georgia Institute of Technology. He received a B.S. degree in chemical engineering from the University of Texas at Austin, an M.S. degree in industrial engineering from the University of Texas at El Paso, and the Ph.D degree in industrial engineering from Texas A & M University. His research interests include computer vision, pattern recognition and tolerance verification. He is a member of IEEE, IIE, ORSA and SME.

**About the Author**—LAKSHMI NARASIMHAN was born in India in 1962. He received his B.Tech. degree in mechanical engineering from the Institute of Technology, Banaras Hindu University, India, in 1984, and his Ph.D. in management science from the University of Texas at Dallas in 1988. He is currently Assistant Professor at the School of Management, Georgia Tech. His research interests include computational geometry, computer vision, spatial theory of voting, and systems analysis and design. He is a member of the Operations Research Society of America, the Institute of Management Science, the IEEE Computer Society, and the Association for Computing Machinery.

**About the Author**—SOUNG R. YEE received the MSIE from the Korea Advanced Institute of Science and Technology in 1984. From 1984 to 1987 he was with Samsung Electronics as a research engineer. He is currently a doctoral candidate in industrial and systems engineering at the Georgia Institute of Technology working towards the development of a 3D object classification system.