

*Conjecture 1:* Let  $\ell(m)$  denote the linear span of  $S_m$ , where  $S_m$  is the 0,1 sequence from the Segre hyperoval defined in the last paragraph. Then

$$\ell(m)/m = 1 + \ell(m-2)/(m-2) + \ell(m-4)/(m-4)$$

for all  $m \geq 9$ .

#### ACKNOWLEDGMENT

The author wishes to thank H. Dobbertin and R. M. Wilson for their help and encouragement.

#### REFERENCES

- [1] W. E. Cherowitzo and L. Storme, " $\alpha$ -flocks with oval herds and monomial hyperovals," preprint.
- [2] T. Cusick and H. Dobbertin, "Some new 3-valued crosscorrelation functions of binary  $m$ -sequences," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1238–1240, 1996.
- [3] D. Glynn, "Two new sequences of ovals in finite Desarguesian planes of even order," *Lecture Notes in Mathematics*, vol. 1036. Berlin, Germany: Springer, 1983, pp. 217–229.
- [4] R. Gold, "Maximal recursive sequences with 3-valued recursive cross-correlation functions," *IEEE Trans. Inform. Theory*, vol. IT-14, pp. 154–156, Jan. 1968.
- [5] S. W. Golomb, "The use of combinatorial structures in communication signal design," in *Application of Combinatorial Mathematics*, C. Mitchell, Ed. Oxford, U.K.: Clarendon, 1997, pp. 59–78.
- [6] A. Maschietti, "Difference sets and hyperovals," *Des., Codes Cryptogr.*, vol. 14, pp. 89–98, 1998.
- [7] M. K. Simon, J. K. Omura, R. A. Scholtz, and B. K. Levitt, *Spread Spectrum Communications*, vol. 1. Reading, MA: Computer Science, 1985.

## Design of Shapes for Precise Image Registration

Alfred M. Bruckstein, Larry O’Gorman, *Fellow, IEEE*,  
and Alon Orlitsky

**Abstract**—This correspondence deals with the problem of designing planar shapes for subpixel image registration. Basic theoretical considerations are shown to lead to a lower bound on location accuracy. Optimal registration marks achieving this bound are discussed. These optimal designs, however, require very high printing or etching resolution and are inherently very sensitive to variations in the image sampling model (like scaling of grid size and rotation). More robust, optimal and suboptimal “topology-preserving” registration marks are then introduced and analyzed.

**Index Terms**—Grid geometry, image registration, information-theoretic precision bounds.

#### I. INTRODUCTION

Suppose that a planar shape is digitized by point sampling at lattice points defined by a square grid. The result is a binary two-dimensional “digital image” of the shape: a pattern of zeros and ones indicating whether the corresponding grid point belongs to the shape or its background (see Fig. 1). In case the planar shape is known up to an arbitrary translation in the plane, the two-dimensional pattern of zeros and ones that form its digital image provides information about its location in the plane. Formally, the planar shape  $S$  can be described by an indicator function over  $R^2$

$$\chi_S(x, y) = \begin{cases} 1, & \text{if } (x, y) \in S \\ 0, & \text{if } (x, y) \notin S. \end{cases} \quad (1.1)$$

If the shape is translated by a vector  $(\mathbf{X}, \mathbf{Y})$ , the translated shape  $S(\mathbf{X}, \mathbf{Y})$  has an indicator function given by

$$\chi_{S(\mathbf{X}, \mathbf{Y})}(x, y) = \chi_S(x - \mathbf{X}, y - \mathbf{Y}). \quad (1.2)$$

We digitize translated versions of  $S$  on the unit grid  $\{(i, j) \in Z^2\}$ . The result of digitizing  $S(\mathbf{X}, \mathbf{Y})$ , on the lattice is

$$B(i, j) = \chi_{S(\mathbf{X}, \mathbf{Y})}(i, j) = \chi_S(i - \mathbf{X}, j - \mathbf{Y}). \quad (1.3)$$

We address in this correspondence the following questions.

1) Given a planar shape of finite support, say  $S \subset [0, A) \times [0, A) \subset R^2$  (i.e.,  $\chi(x, y) = 0$  for  $(x, y) \in R^2 \setminus [0, A) \times [0, A)$ ), how should we estimate the translation vector  $(\mathbf{X}, \mathbf{Y})$  from  $B(i, j)$ ?

2) What is the best that we can do in estimating the location of  $S$  over all the possible planar shapes of finite extent and what shapes achieve minimum error in location?

3) How to design “good” shapes for location estimation when the shapes are constrained to obey certain further restrictions on size, topology, etc.

Manuscript received November 11, 1995; revised December 20, 1997.

A. M. Bruckstein was with Bell Laboratories, Murray Hill, NJ 07974 USA. He is now with Computer Science Department, Technion–Israel Institute of Technology, 32000 Haifa, Israel.

L. O’Gorman was with Bell Laboratories, Murray Hill, NJ 07974 USA. He is now with VERIDICOM Inc., Chatham, NJ 07928 USA.

A. Orlitsky was with Bell Laboratories, Murray Hill, NJ 07974 USA. He is now with the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093 USA.

Publisher Item Identifier S 0018-9448(98)06738-8.

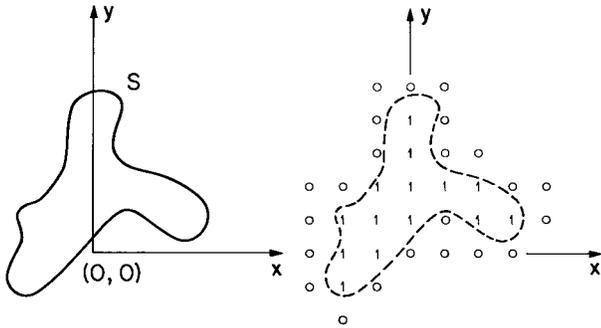


Fig. 1. Image digitization by point sampling.

The motivation for the above problems arises from the necessity to design fiducials—standard marks or shapes—for locating objects in images in various machine-vision applications. For example, such standard marks are etched into printed circuit boards and on VLSI wafers, with the aim of enabling subpixel (i.e., less than grid size) registration accuracies when placing overlay masks, mounting integrated circuit packages on PC boards, etc. Since the problem arose in industrial applications, it has received some attention both by theoreticians, e.g., [1], and by application-oriented engineers, e.g., [2]–[8]. However, the problem was approached mostly via a correlation or template-matching type of analysis [5], [8], [9]. When dealing with binary digitizations of two-level images the approach to subpixel accuracy registration was based upon either centroid computations for the digitized images [2], [4], [3] or the theoretically amenable problem of digitized straight edge location [1].

This correspondence is based on a technical memorandum issued in 1989 at Bell Laboratories, [10], which had a limited circulation. At the time this memo was published, a paper [11] dealing with related topics appeared. A subsequent paper by Havelock [12] deals with some extensions of ideas from [11] and [10]; however, the results to be presented here have never been published before.

The important topic of circularly symmetric and bull's-eye fiducials discussed in [10] and here, in Sections IV and V, has been the subject of several papers since 1989. In [13], bull's-eye patterns were studied by simulation while [14] analyzed rotation-invariant fiducials in the spirit of [10], using results from lattice geometry and number theory. The paper [15] deals with some practical topics arising from the use of circularly symmetric fiducials, while [16] advocates the idea of exploiting gray-scale information to improve the location accuracy for circular fiducials. This idea, also discussed in [11] and [17] is, in our opinion, an important direction of research in the context of fiducial design and should be the subject of further theoretical investigations.

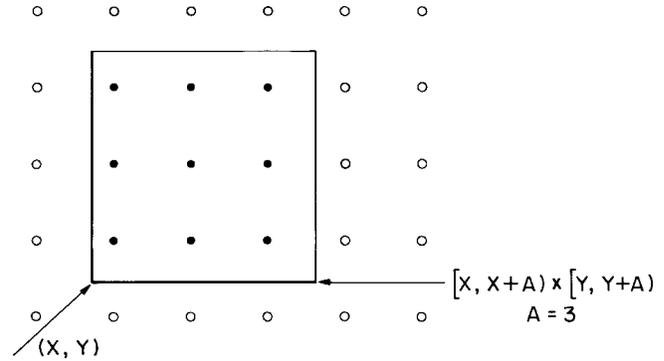
## II. THEORETICAL CONSIDERATIONS

Suppose that a shape  $S$  translated by the unknown  $(\mathbf{X}, \mathbf{Y})$  yields a digitized image  $B(i, j)$ . Returning to the problems defined in Section I, we shall first address the issue of estimating the vector  $(\mathbf{X}, \mathbf{Y})$  from the data  $B(i, j)$ . Clearly, the digitized image can, and usually will be the same for a range of different translation vectors (the mapping of  $(\mathbf{X}, \mathbf{Y})$  via  $\chi_S(x - \mathbf{X}, y - \mathbf{Y})$  into  $B(i, j)$  is many to one). The best we can hope for is to determine the region  $\mathcal{R}$

$$\mathcal{R} = \{(\mathbf{X}, \mathbf{Y}); \chi_{S(x, Y)}(i, j) = B(i, j), \forall i, j\} \subset \mathbb{R}^2. \quad (2.1)$$

This, however, is easy to do since from the value  $B(i, j)$  at each  $(i, j)$  and the assumed complete knowledge of  $\chi(x, y)$  we get constraints on  $(\mathbf{X}, \mathbf{Y})$ . Indeed, if  $B(i_0, j_0)$  is known, we have from

$$\chi_S(i_0 - \mathbf{X}, j_0 - \mathbf{Y}) = B(i_0, j_0)$$


 Fig. 2. The meaningful samples for  $S \subset [0, A) \times [0, A)$ .

that  $(\mathbf{X}, \mathbf{Y})$  is constrained to be in the region

$$R_{(i_0, j_0)} \triangleq \{(\mathbf{X}, \mathbf{Y}); \chi_{S(x, Y)}(i_0, j_0) = B(i_0, j_0)\} \subset \mathbb{R}^2 \quad (2.2)$$

providing the observed sample value  $B(i_0, j_0)$  via (2.2). Then we clearly have

$$\mathcal{R} = \bigcap_{(i_0, j_0) \in \mathbb{Z}^2} R_{(i_0, j_0)} \quad (2.3)$$

and all points in  $\mathcal{R}$  are equally eligible as estimates of  $(\mathbf{X}, \mathbf{Y})$ .

### A Lower Bound on Location Accuracy

Recall that we have assumed the registration/location shape to be of bounded support, i.e.,  $S \subset [0, A) \times [0, A)$  for some integer  $A$ . Therefore, we know that the digitization of  $S$  can have at most  $A^2$  “meaningful bits.” Of course, all “pixel digits”  $B(i, j)$  are meaningful, however, the prior assumption of finite support implies that, if we know the pixel  $(i_0, j_0)$  (i.e., the region  $(i_0 - 1, i_0] \times (j_0 - 1, j_0] \subseteq \mathbb{R}^2$ ) to which  $(\mathbf{X}, \mathbf{Y})$  belongs, we know *in advance* that all  $B(i, j)$  outside the range  $i, j \in [i_0, i_0 + A) \times [j_0, j_0 + A) \subset \mathbb{Z}^2$  will be zero, see Fig. 2.

Clearly, the output pattern for  $(\mathbf{X}, \mathbf{Y}) = (m + \epsilon_x, n + \epsilon_y)$  is identical to that of  $(\mathbf{X}, \mathbf{Y}) = (\epsilon_x, \epsilon_y)$  shifted by  $(m)$  to the right and by  $(n)$  upwards. (Here  $\epsilon_x, \epsilon_y \in (0, 1]$ .) In other words, up to translation by a pair of integers, we can have at most  $2^{A^2}$  different digital patterns.

Assume first that  $(\mathbf{X}, \mathbf{Y}) \in (0, 1] \times (0, 1]$  and consider all the patterns that  $S$  generates when the origin is placed within this region. Each pattern corresponds to a region  $\mathcal{R}$  of possible translations within  $(0, 1] \times (0, 1]$ , and clearly, different patterns must induce disjoint  $(\epsilon_x, \epsilon_y)$ -translation subsets of the unit square. Therefore, the output patterns induce a decomposition of the unit square into some number, say  $P$ , of disjoint regions  $\mathcal{R}_1, \dots, \mathcal{R}_P$ . If  $P$  patterns are generated, the area of the largest region—inducing the worst uncertainty in estimating  $(\mathbf{X}, \mathbf{Y})$ —must be greater than  $1/P$ . Since we can have at most  $2^{A^2}$  different patterns we obtain that the worst case uncertainty region for any shape  $S$ ,  $\mathcal{R}_{\text{worst}}$ , must obey

$$\text{Area} \{\mathcal{R}_{\text{worst}}\} \geq \frac{1}{2^{A^2}}. \quad (2.4)$$

The above argument assumed that we know the pixel to which  $(\mathbf{X}, \mathbf{Y})$  belongs. Relaxing this assumption, requires  $S$  to be designed so that the “rough translation”  $(i_0, j_0)$  can always be determined. We can easily ensure this by assigning part of the support of  $S$ ,  $[0, A) \times [0, A)$ , to rough location, for example, by requiring that

$$\chi_S(x, y) = 1, \quad \text{for all } (x, y) \in [0, 1) \times [0, 1) \quad (2.5)$$

(see Fig. 3). This implies that the lower left corner of the digitized pattern is always an ON (= 1) bit and we can immediately locate

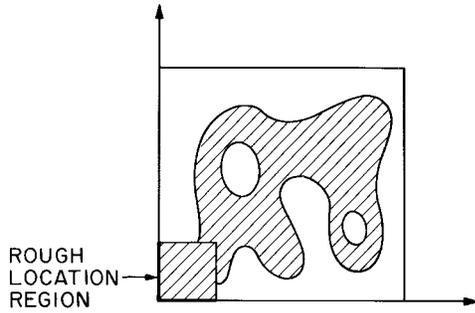


Fig. 3. Shape providing immediate rough translation information.

the pattern up to a pixel size region. Indeed, if  $(i_0, j_0)$  is the lower leftmost grid point that has value 1, i.e.,

$$B(i_0, j_0) = \chi_S(i_0 - \mathbf{X}, j_0 - \mathbf{Y}) = 1$$

we know from (2.5) that  $\mathbf{X} \in (i_0 - 1, i_0]$  and  $\mathbf{Y} \in (j_0 - 1, j_0]$ .

From the  $A^2$  meaningful bits, one bit has thus been allocated to “rough location” and the remaining  $A^2 - 1$  may yield further location information. This could also be explained as follows: the lower leftmost nonzero bit assumed here to be  $(i_0, j_0)$  provides that

$$R(i_0, j_0) = (i_0 - 1, i_0] \times (j_0 - 1, j_0]$$

and the rest of the information bits in the pattern can serve to further refine  $R(i_0, j_0)$  down to  $\mathcal{R}$ —the final uncertainty region. In this case the bound for  $\mathcal{R}_{\text{worst}}$  becomes

$$\text{Area } \{\mathcal{R}_{\text{worst}}\} \geq \frac{1}{2^{(A^2-1)}}. \tag{2.6}$$

If, say,  $A = 3$  as in the example of Fig. 2, the worst  $\mathcal{R}$  area is larger than  $1/2^8$  implying that the “best” balanced shape  $S$  should yield for  $\mathbf{X}$  and  $\mathbf{Y}$  an estimate within  $\Delta\mathbf{X} = \Delta\mathbf{Y} = 1/2^4 = 1/16$ .

For any given shape, the procedure of finding  $\mathcal{R}$  can readily be carried out computationally and the side of  $\mathcal{R}_{\text{worst}}$ , compared to the lower bound, will yield the location properties of  $S$ .

Havelock [11] also considers equivalence classes of locations that yield identical sampled and quantized images (and calls them “locales”) and stresses their importance in evaluating the precision of locating a translated bivariate function. His paper, however, does not ask the question of shape design and is aimed at analyzing tradeoffs between grey levels and sampling density, in various mensuration tasks.

### III. DESIGN OF OPTIMAL LOCATION MARKS

The lower bound presented in the previous section enables us to evaluate various designs for  $S$ . The question of optimal shapes naturally arises in this context: can we design a shape  $S$  that actually achieves the lower bound? To answer this question let us first consider the one-dimensional counterpart of our problem as follows.

Suppose we have to design a binary function  $\beta(x)$  with finite support  $[0, A)$  with  $A \in \mathbb{Z}$ , and taking values in  $\{0, 1\}$ , so that when  $\beta(x - \mathbf{X})$  is sampled at the integer coordinates the resulting 0/1 pattern locates  $\mathbf{X}$  with the highest precision. From the developments of the previous section we know that in the sampled (digitized) image we get  $A$  meaningful bits.

Therefore, if we design a function  $\beta(x)$  that uses one output bit for “rough location” and enables us to determine  $\mathbf{X}$  within an interval of length  $1/2^{A-1}$  in the worst case, we have an optimal design. The

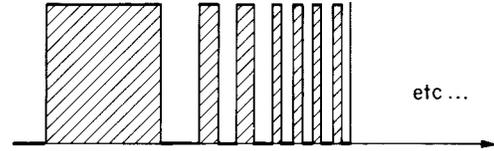


Fig. 4. Optimal one-dimensional design.

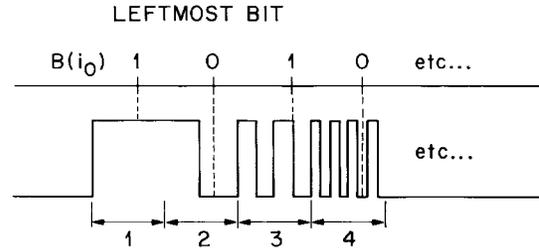


Fig. 5. Each bit doubles the location precision in the optimal design.

following function is an optimal  $\beta(x)$ :

$$\beta^0(x) = \begin{cases} 0, & \text{for } x \in (-\infty, 0) \\ 1, & \text{for } x \in [0, 1) \\ 1, & \text{for } x \in [1, 3/2); & 0, & \text{for } x \in [3/2, 2) \\ 1, & \text{for } x \in [2, 9/4); & 0, & \text{for } x \in [9/4, 10/4) \\ 1, & \text{for } [10/4, 11/4); & 0, & \text{for } [11/4, 3); \text{ etc...} \\ \vdots & \\ 0, & \text{for } x \in [A, \infty) \end{cases}$$

a concatenation of Haar-type basis functions [18] (see Fig. 4).

Since we have  $\beta^0(x) = 1$  for  $x \in [0, 1)$ , the leftmost 1 of the output pattern generated by  $B(i) = \beta^0(i - \mathbf{X})$  will provide the information that  $\mathbf{X} \in (i_0 - 1, i_0]$ . Then (see Fig. 5) the output pattern will look like

$$000 \cdots 0 \underbrace{1b_1b_2b_3 \cdots b_{A-1}}_{\text{meaningful bits}} 0000 \cdots$$

It is now easy to realize that

$$\begin{aligned} b_1 = 1, & \quad \text{iff } \epsilon_x \in (1/2, 1] \\ b_2 = 1, & \quad \text{iff } \epsilon_x \in (1/4, 2/4] \text{ or } (3/4, 1] \\ b_3 = 1, & \quad \text{iff } \epsilon_x \in (1/8, 2/8] \text{ or } (3/8, 4/8] \\ & \quad \text{or } (5/8, 6/8] \text{ or } (7/8, 1], \text{ etc., } \cdots \end{aligned}$$

and that the  $A - 1$  bits beyond  $b_0$  determine a straightforward binary coding for an uncertainty interval of size  $1/2^{A-1}$ . For example, if we have  $A = 4$  and see the pattern

$$00 \cdots 0 \underbrace{1010}_{\text{meaningful bits}} 000$$

we know that  $b_1 = 0$  means  $\epsilon_x \leq 1/2$  and given this,  $b_2 = 1$  means that  $\epsilon_x > 1/4$ . Now  $b_3 = 0$  indicates that  $\epsilon_x \in (2/8, 3/8]$  and this is the last significant piece of information we get, as we know that  $A = 4$ . The  $\epsilon_x$  has been located within an interval of length  $1/8$ , i.e.,  $1/2^{A-1} = 1/2^3$ , the best theoretically achievable precision.

Thus we have produced an example of an optimal one-dimensional design. It is clear from the above discussion that we can readily produce other designs of identical performance, for example, by reversing the levels 0 and 1 over the regions  $(1, 2]$ ,  $(2, 2\frac{1}{2}]$ ,  $(2\frac{1}{2}, 3]$ ,  $(3, 3\frac{1}{4}]$ ,  $(3\frac{1}{4}, 3\frac{1}{2}]$ ,  $(3\frac{1}{2}, 3\frac{3}{4}]$ ,  $(3\frac{3}{4}, 4]$ , etc, in arbitrary

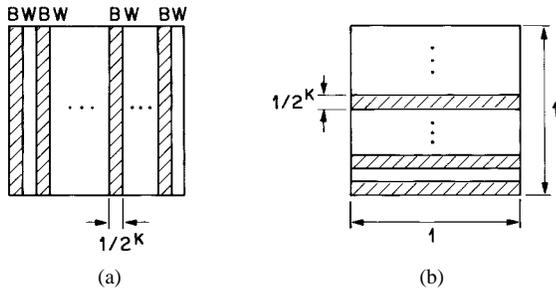
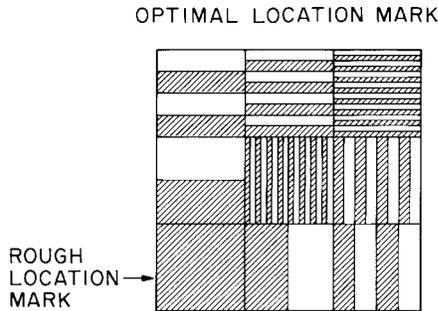


Fig. 6. The basic cells for 2-D optimal fiducial designs.


 Fig. 7. An optimal 2-D fiducial of area  $3 \times 3$ .

ways. Thus we can have  $2^P$  equivalent “optimal” designs where  $P = 0+1+2+4+\dots+2^{(A-2)}$ , for a  $\beta(x)$  with support  $[0, A)$ . Some of these designs will be better than others in terms of the number of 0/1 transitions and the distances between them. The original design  $\beta^0(x)$  has  $1+2+4+\dots+2^{A-2} = (2^{A-1} - 1)$  1-valued connected components ( $1+2+4=7$  if  $A=4$ , as in Fig. 5) but reversing 0/1 over the regions  $(2, 2\frac{1}{2}]$ ,  $(3, 3\frac{1}{4}]$ ,  $(3\frac{1}{2}, 3\frac{3}{4}]$ , etc., we obtain an optimal design with  $1+1+2+\dots+2^{A-3} = 2^{A-2}$  (4 if  $A=4$ ) connected 1-valued runs. These latter designs achieve a location precision of  $1/2^{A-1}$  with a shape that has  $2^{A-1}$  level transitions. The connection between level transitions and location accuracy will be discussed in detail in Section IV.

Two-dimensional (2-D) optimal shapes  $\chi^0(x, y)$  can now be found as immediate generalizations of the one-dimensional (1-D) designs  $\beta^0(x)$ . Given a region  $[0, A) \times [0, A)$  we partition it into unit squares, use the lower leftmost pattern as a rough location sign, and assign the remaining  $A^2 - 1$  squares, half to  $\epsilon_x$  location patterns and half to  $\epsilon_y$  location patterns. The location patterns will implement the 1-D coding idea as follows.

A horizontal  $1/2^k$  unit cell is defined as in Fig. 6(a) and a vertical  $1/2^k$  unit cell as its  $90^\circ$  rotation, Fig. 6(b). Then we compound the horizontal and vertical unit cells for  $k = 1, 2, \dots, (A^2 - 1)/2$  to a square pattern over  $[0, A) \times [0, A)$  in any predetermined way. The result will be a registration shape with the “rough location” property and yielding a precision of  $\Delta \mathbf{X} = \Delta \mathbf{Y} = 1/2^{\lfloor (A^2-1)/2 \rfloor}$ .

Suppose, for example, that  $A = 3$ . Then an optimal  $3 \times 3$  location mark is the pattern depicted in Fig. 7. Of course, except for the rough location mark, we could arbitrarily permute the other regions with no effect on the precision achieved by this shape. Also we could flip the values of  $\chi^0(x, y)$  as described following the discussion of 1-D designs. Note that it is advisable to have  $A$  an odd number, since then we can have identical  $\Delta \mathbf{X}$  and  $\Delta \mathbf{Y}$  precision, otherwise, we must do some more thinking about the assignment of one, pairless, unit square. Therefore, 2-D optimal location patterns do exist, however, the examples we have seen so far are based on high-precision etching, are very sensitive directionally, and assume “perfect” knowledge of

the pixel size. As we move the shape  $\chi(x, y)$  within a given pixel-sized area (located via the rough location bit) all  $2^{A^2-1}$  possible 2-D 0/1 patterns arise as digitized images equally often, this in fact being the secret of achieving optimality.

The optimal shapes described above achieve their goal by being fine tuned to the sampling grid size; they are complicated topologically and are quite difficult to produce, if high location precisions are to be achieved. In fact, any shape that achieves optimal-location performance will necessarily require high-precision etching, tuned to the pixel size. This is obvious from the observation that translating an “optimal” shape by more than  $1/2^{\lfloor (A^2-1)/2 \rfloor}$  in either the vertical or the horizontal direction will have to induce some changes in the digitized patterns seen. We, therefore, are led to further consider shapes that are simpler, more robust to errors in our knowledge of grid size. And, although they use area less efficiently, they still yield good location performance.

#### IV. DESIGN OF ROBUST LOCATION MARKS

Let us again consider a 1-D example first. Suppose we have a function  $\beta(x)$  of the form

$$\beta(x) = \begin{cases} 0, & \text{for } x \in [-\infty, 0) \\ 1, & \text{for } x \in [0, l_1) \\ 0, & \text{for } x \in [l_1, l_2) \\ 1, & \text{for } x \in [l_2, l_3) \\ \vdots & \vdots \\ 1, & \text{for } x \in [l_{K-1}, l_K) \\ 0, & \text{for } x \in [l_K, +\infty) \end{cases} \quad (4.1)$$

parametrized by the sequence of increasing positive numbers  $\{l_1, l_2, \dots, l_K\}$ . As before, assume that we digitize  $\beta(x - \mathbf{X})$  by sampling it at the integers. If  $l_1 \geq 1$ ,  $\beta(x)$  has the rough location property and we can assume that  $\mathbf{X} = \epsilon_x \in (0, 1]$ . The digitization of  $\beta(x - \epsilon_x)$  will result in a pattern of zeros and ones of the form

$$B(i) = 00 \dots 0 \underbrace{1^{\rho_1} 0^{\rho_2} 1^{\rho_3} 0^{\rho_4} 1^{\rho_5} \dots 0^{\rho_{K-1}} 1^{\rho_K}}_{\text{pattern}} 00$$

where  $b^{\rho_s}$  means  $(b b \dots b)$ ,  $\rho_s$  times (and if  $\rho_s = 0$  no  $b$  at all). Denote by  $\{n_r\}$  the sequence

$$n_r = \sum_{s=1}^r \rho_s, \quad \text{for } r = 1, 2, \dots, K. \quad (4.2)$$

From the digitization process we have that the sequence  $\{n_r\}$  is a function of  $\epsilon_x$  and

$$n_r(\epsilon_x) = \lfloor l_r + \epsilon_x \rfloor \quad (4.3)$$

for  $\epsilon_x \in (0, 1]$ , and  $n_r(\epsilon_x)$  jumps from  $\lfloor l_r \rfloor$  to  $\lfloor l_r \rfloor + 1$  as  $\epsilon_x$  increases from 0 to 1. The jump point will be at

$$\epsilon_{xr} = 1 - (l_r - \lfloor l_r \rfloor) = 1 + \lfloor l_r \rfloor - l_r. \quad (4.4)$$

Suppose that we have, from the sequence  $B(i) = \beta(i - \epsilon_x)$ , complete information on the  $\{\rho_s\}$  and, consequently, on the  $\{n_r(\epsilon_x)\}$  sequence for each  $\epsilon_x$ . Then we can look at the vector function  $\mathbf{N}(\epsilon_x) = [n_1(\epsilon_x), n_2(\epsilon_x), \dots, n_K(\epsilon_x)]$  and realize that the vector  $\mathbf{N}(\epsilon_x)$  changes at each  $\epsilon_{xr}$ . Therefore, given  $\mathbf{N}(\epsilon_x)$  we can determine uniquely to what interval  $(\epsilon_{xi}, \epsilon_{xj}]$ ,  $\epsilon_x$  belongs. The worst case precision in locating  $\epsilon_x$  will be determined by the length of the longest interval induced on  $(0, 1]$  by the collection of breakpoints

$$\{\epsilon_{xr} = 1 + \lfloor l_r \rfloor - l_r \mid r = 1, 2, \dots, K\} \quad (4.5)$$

induced by the design  $\{l_1 l_2 \dots l_K\}$  of  $\beta(x)$ . In the previous section we have seen one basic example of a function  $\beta(x)$ , composed of concatenating Haar-type 0/1 functions of increasing spatial resolution,

that had the design

$$l_1 = 1\frac{1}{2}, \quad l_2 = 2, \quad l_3 = 2\frac{1}{4}, \quad l_4 = 2\frac{1}{2}, \quad l_5 = 2\frac{3}{4},$$

$$l_6 = 3, \quad \text{etc.,} \dots$$

For this design it was shown that the breakpoints  $\{\epsilon_{xr}\}$  divided the  $(0, 1]$  span of  $\epsilon_x$  into  $2^{A-1}$  equal intervals of length  $1/2^{A-1}$  thereby achieving the optimal precision possible for the given support of  $\beta(x)$ . However, for  $A = 5$ , for example, we had there a total of  $2[0 + 1 + 2 + 4 + 8] = 2[15] = 30$  level transitions (15 regions) achieving a  $1/16$  precision. Therefore, two breakpoints always coincided in the optimal design. We could, however, find an improved optimal design having distinct breakpoints by using the level flipping trick discussed in the previous section. With  $A = 5$  we could find an optimal design with  $8 (= 2^{A-2})$  connected components with value 1. This design has  $K = 15$  and the  $\{\epsilon_{xr}\}$ 's are distinct and divide  $[0, 1)$  into 16 equal intervals. We note that this design works because we do have in this case *complete information* on the  $\rho_s$  sequence from the data  $B(i)$ .

The point illustrated by the above argument is the following. To obtain good designs one should try to place (by appropriately choosing the  $\{l_r\}$  sequence) the breakpoints  $\{\epsilon_{kr}\}$  as uniformly as possible over  $(0, 1]$ —however, one must also ensure the unambiguous recovery of all  $n_r(\epsilon_x)$  of  $\mathbf{N}(\epsilon_x)$  from the digitized data. One particularly straightforward way to ensure that  $\mathbf{N}(\epsilon_x)$  is completely determined by the  $B(i) = \beta(i - \epsilon_x)$  sequence is to use 1-D designs that “preserve topology” in the following sense. If the  $\beta(x)$  function has  $C$  “connected components” (connected regions where  $\beta(x) = 1$ ), i.e.,  $K = 2C - 1$ , the image  $B(i)$  should always have  $C$  connected components, or runs of 1 as well.

In this case, the association of a digitized image  $B(i)$  to the corresponding  $\rho_r$ 's and  $n_r(\epsilon_x)$  clearly becomes a trivial task. The “topology-preservation” property will be ensured for all  $\epsilon_x \in (0, 1]$  if and only if we have all intervals  $[l_i, l_{i+1})$  at least of length 1, i.e., we need

$$l_1 \geq 1, \quad l_2 - l_1 \geq 1, \quad l_3 - l_2 \geq 1, \dots, \quad l_K - l_{K-1} \geq 1. \quad (4.6)$$

Now we may ask the following question. What are the best achievable location accuracies within the class of “topology-preserving” designs? The answer to this question lies in the distribution of the jump points  $\epsilon_{xr}$ , induced by the designs. Given  $K = 2C - 1$  for some  $C$ , we shall have a topology-preserving design provided (4.6) is satisfied and the worst case precision will be lower-bounded by

$$\text{length [worst interval]} \geq \frac{1}{K+1} = \frac{1}{2C} \quad (4.7)$$

since the  $K$  jump points determine at most  $K + 1$  intervals. We can achieve this lower bound with any design that would place the jump points at the locations

$$\left\{ \frac{1}{K+1}, \frac{2}{K+1}, \dots, \frac{K}{K+1} \right\}$$

$$\equiv \{\epsilon_{xr} = 1 + [l_r] - l_r \mid r = 1, 2, \dots, K\}. \quad (4.8)$$

Indeed, setting

$$\begin{cases} l_1 = 1 + 1/(K+1) \triangleq D \\ l_2 = 2 + 2/(K+1) = 2D \\ l_3 = 3 + 3/(K+1) = 3D \\ \vdots \\ l_K = K + K/(K+1) = KD \end{cases}$$

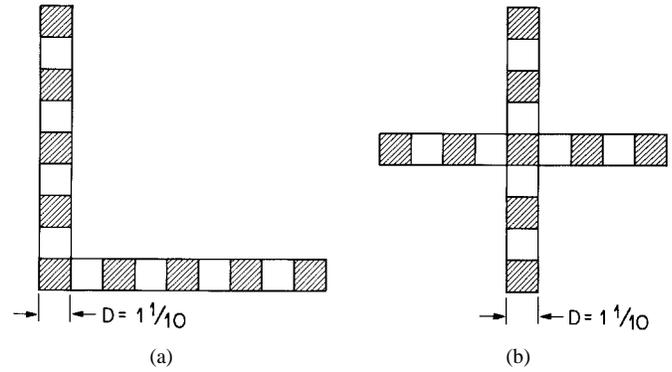


Fig. 8. Two topology-preserving location marks.

we obtain the desired jump point distribution, since

$$\begin{cases} \epsilon_{x1} = 2 - (1 + 1/(K+1)) = K/(K+1) \\ \epsilon_{x2} = 3 - (2 + 2/(K+1)) = (K-1)/(K+1) \\ \vdots \\ \epsilon_{xK} = K + 1 - (K + K/(K+1)) = 1/(K+1). \end{cases}$$

Therefore, with  $C$  connected components of length  $D = 1 + 1/(K+1)$ , spaced at a distance  $D$  from one another, and for a total support of  $(K + K/(K+1)) \cong K + 1 = 2C$  for  $\beta(x)$ , we can get a design that achieves both “topology preservation” and subpixel registration to within  $1/2C$ . This is seen to be an *optimal* “topology-preserving” design. To see what happens, consider the case of  $C = 3$ , i.e.,  $K + 1 = 6$ . The predicted accuracy is  $\Delta X = 1/6$  and we readily realize that six equal intervals for  $\epsilon_x$  are encoded as follows:

$$\begin{aligned} \epsilon_x \in (0, 1/6] &\rightarrow 01010100\dots \\ \epsilon_x \in (1/6, 2/6] &\rightarrow 01010110\dots \\ \epsilon_x \in (2/6, 3/6] &\rightarrow 01010010\dots \\ \epsilon_x \in (3/6, 4/6] &\rightarrow 01011010\dots \\ \epsilon_x \in (4/6, 5/6] &\rightarrow 01001010\dots \\ \epsilon_x \in (5/6, 6/6] &\rightarrow 01101010\dots \end{aligned}$$

Note that this design too implies complete knowledge of the sampling, knowledge of grid size, however, it does not require the etching of very thin lines.

From the above 1-D analysis it follows that location marks as shown in Fig. 8 achieve a precision of  $\Delta X = \Delta Y = 1/2C$  and clearly have the topology-preservation property. Note that it is superfluous to fill the  $2C \times 2C$  square with a regular checkerboard pattern since no further increase in location accuracy will result. However, we should be able to use area more efficiently in two dimensions. We could strive to obtain better designs with a “topology preservation” like property and the 1-D analysis serves as a good guidance in this endeavor. For a general design we shall first require the rough location property, i.e.,  $\chi(x, y) = 1$  for  $(x, y) \in [0, 1) \times [0, 1)$ . The registration shape properties then clearly depend on the set of lengths  $\{l_{Hk}^{(j)}\}$  and  $\{l_{Vk}^{(i)}\}$  that are induced by the intercepts of  $S(\epsilon_x, \epsilon_y)$  with the horizontal and vertical grid lines having equations  $y = j$  and  $x = i$  (see Fig. 9).

The location resolution for  $\epsilon_x$  and  $\epsilon_y$  can be theoretically predicted provided we can determine the length  $\{l_{Hk}^{(j)}\}$  and  $\{l_{Vk}^{(i)}\}$ , without having to know the translation precisely. This might seem to be a rather stringent requirement; however, it is met in an obvious way by a rather large class of shapes, and two immediate examples are

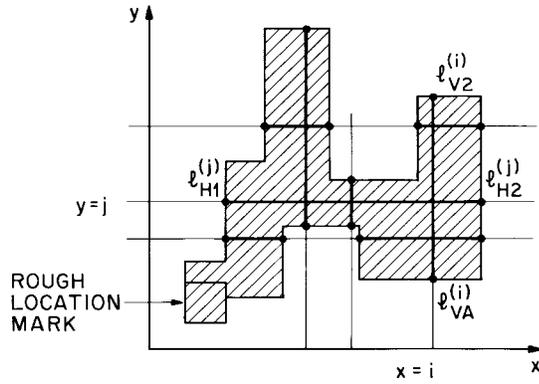


Fig. 9. Illustration of the general conditions for topology preservation.

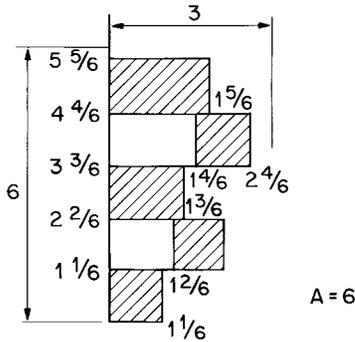


Fig. 10. An efficient topology-preserving design.

the ones depicted in Fig. 8. We must also extend the “topology-preservation” property to two dimensions, in order to ensure a simple and robust way to solve the correspondence problem, i.e., to be able to determine for each intersection patterns defined by  $\{I_{Hk}^{(j)}\}$  or  $\{I_{Vk}^{(i)}\}$  the corresponding  $\{\rho_i\}$  sequences by looking at runs of zeros and ones. A straightforward way to ensure the “topology-preservation” property is to require that for all  $\epsilon_x, \epsilon_y$  the lines  $y = j$  and  $x = i$  intersect  $S(\epsilon_x, \epsilon_y)$  in “topology-preserving” 1-D patterns. If both these requirements are met, the set of horizontal and vertical intersection patterns can readily be exploited to yield estimates of  $\epsilon_x, \epsilon_y$  with theoretically predictable precision.

The examples of Fig. 8 show that within an area of  $A \times A$  ( $A = 2C$ ) we can always have a simple registration shape design that achieves a prevision of  $\Delta X = \Delta Y = 1/A$ , however, the area is not used efficiently at all. Recall that in the previous section we have seen an optimal design that does not obey the “topology-preservation” property and achieves a precision of  $\Delta X = \Delta Y = \sqrt{1/2^{(A^2-1)}}$  in area  $A^2$ . Fig. 10 shows how we could more densely pack a topologically preserving design. The area used would be less than  $3A$  for the same location resolution of  $\Delta X = \Delta Y = 1/A$ . It is clear that the design exhibited in Fig. 10 has all the required properties. From the 1-D analysis we realize that the first nonzero column will yield the vertical location to a precision of  $\Delta Y = 1/2C = 1/A$  and the horizontal location will be provided by the shape intercepts with the grid lines  $y = j$ . Since the horizontal intercepts are topology-preserving and boundaries occur at  $(integer + k/2C)$  locations for all  $k = 1, 2, \dots, 2C - 1$  we also have a horizontal positioning precision of  $\Delta X = 1/2C = 1/A$ .

Thus we readily get topologically preserving designs that achieve  $\Delta X = \Delta Y = 1/A$  with area  $\cong 3A$ , as opposed to optimal designs with  $\Delta X = \Delta Y = \sqrt{1/2^{A^2-1}}$  with area  $A^2$ .

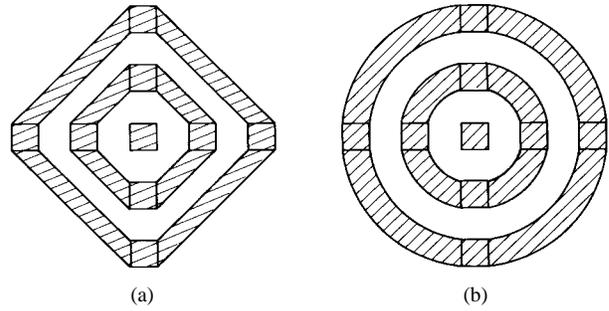


Fig. 11. A bad (a) and a good (b) topology-preserving design.

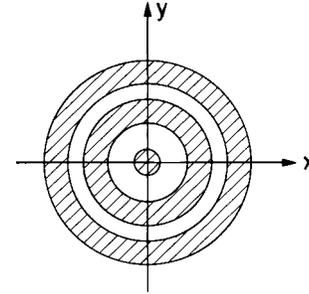


Fig. 12. The classical “bull’s eye.”

The above discussion shows that we should attempt to provide designs that are topologically preserving in the sense of being able to associate Object/Background transitions to 1/0 transitions in the digitized patterns in a unique way and that also ensure as many object background transitions as possible at the intersections of the translated shape  $S(\mathbf{X}, \mathbf{Y})$  with the grid. The fractional parts of the ordinates and abscissas of these transitions should be spread as evenly as possible on the  $(0, 1]$  interval.

With these considerations in mind we should clearly discard the design shown in Fig. 11(a) since the fractional parts of slope 1 boundary intersections with the unit grid are identical, but should consider as potentially good a design as the one shown in Fig. 11(b). Of course, an exact analysis of these designs would require the general methods described in Section II, since these designs, although topologically preserving, do not obey the condition that their grid intersections occur at *a priori* predictable locations in the shape’s master coordinate system. We refer the reader to [13] and [14] for further discussions and theoretical results concerning circularly symmetric fiducials suggested by these designs.

### V. CONCLUDING REMARKS

We have seen that the precision in locating the shift parameters of a given shape from its digitized image is determined by the way the shape interacts with the digitization grid as the shape is translated in the plane. The design of the location mark should attempt to maximize the number of digital 2-D patterns generated at all translations. If the grid size is perfectly known and the shape is known to undergo translations only, optimal registration marks can be designed. If, however, for robustness considerations we impose further conditions on the fiducial, like for example 1-D “topology” preservation at all translations, we considerably reduce the precision that can even theoretically be obtained. Practically, however, a fiducial should still induce as many digitized shapes as possible, while satisfying additional constraints. A particularly attractive fiducial comes to mind when considering the analysis done in Section IV is the classical “rotationally invariant” shape of Fig. 12.

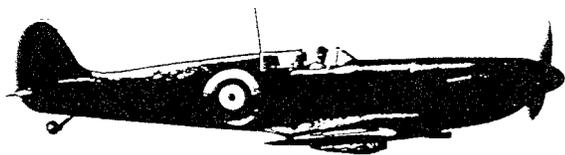


Fig. 13. A British WWII Supermarine Spitfire fighter plane featuring a dangerous fiducial.

A set of  $C$  concentric circles with  $r_1 = \frac{1+1/2C}{2}$  and  $\Delta r = 1+1/2C$  is expected to be a very good (although suboptimal) location shape. Experimental tests in [13] and further theoretical developments recently published in [14], led to the conclusion that the “bull’s-eye” fiducial is indeed a very good, robust and practical location mark.

Hence the Royal Air Force chose, rather poorly, a deadly targeting shape to be painted on its planes during World War II (see Fig. 13).

#### REFERENCES

- [1] C. A. Berenstein, L. N. Kanal, D. Lavine, and E. C. Olson, “A geometric approach to subpixel accuracy,” *CVGIP*, vol. 40, no. 3, pp. 334–360, 1987.
- [2] J. Hill, “Dimensional measurements from quantized images,” in D. Nitzan, S. Bavnard, R. Bolles, R. Cain, J. Hill, W. Park, and R. Smith, Eds., *Machine Intelligence Research Applied to Industrial Automation*, SRI Report prepared for the NSF, Nov. 1980.
- [3] I. Amir, “An algorithm for finding the center of circular fiducials,” *CVGIP*, vol. 49, pp. 398–406, 1990.
- [4] C. Bose and I. Amir, “Design of fiducials for accurate registration using machine vision,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 1196–1200, Dec. 1990.
- [5] H. K. Nishihara and P. A. Crossley, “Measuring photolithographic overlay accuracy and critical dimensions by correlating binarized Laplacian of Gaussian convolutions,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, pp. 17–30, Jan. 1988.
- [6] J. Sanz, Special Issue of PAMI/III on Machine Vision. vol. 10, 1988.
- [7] R. T. Chin, “Automated visual inspection 1981–1987,” *CVGIP*, vol. 41, pp. 346–381, 1988.
- [8] Gi Tian and M. N. Huhns, “Algorithms for subpixel registration,” *CVGIP*, vol. 35, no. 2, pp. 220–233, 1986.
- [9] V. N. Dvornichenko, “Bounds on the deterministic correlation functions with application to registration,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-5, pp. 206–213, Mar. 1983.
- [10] A. M. Bruckstein, L. O’Gorman, and A. Orlicsky, “Design of shapes for precise image registration,” AT&T Bell Lab. Tech. Memo, Murray Hill, NJ, Oct. 1989.
- [11] D. I. Havelock, “Geometric precision in noise-free digital images,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 1065–1075, Oct. 1989.
- [12] ———, “The topology of locales and its effects on position uncertainty,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 380–386, Apr. 1991.
- [13] L. O’Gorman, A. M. Bruckstein, C. B. Bose, and I. Amir, “Subpixel registration using a concentric ring fiducial,” in *Proc. 10th ICPR Conf.* (Atlantic City, NJ, June 1990).
- [14] A. Efrat and C. Gotsman, “Subpixel image registration using circular fiducials,” *Int. J. Comp. Geom. and Appl.*, vol. 4, no. 4, pp. 403–422, 1994.
- [15] M. Tichem and M. S. Cohen, “Sub- $\mu$ m registration of fiducial marks using machine vision,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 791–794, Aug. 1994.
- [16] G. Chiorboli and G. P. Vecchi, “Comments on ‘Design of fiducials for accurate registration using machine vision’,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 1330–1332, Dec. 1993.
- [17] N. Kiryati and A. M. Bruckstein, “Gray levels can improve the performance of binary image digitizers,” *CVGIP*, vol. 53, no. 1, pp. 31–39, 1991.
- [18] S. Karlin and H. Taylor, *A First Course in Stochastic Processes*. New York: Academic, 1975.

## An Analysis of the Timed $Z$ -Channel

Ira S. Moskowitz, *Member, IEEE*, Steven J. Greenwald,  
and Myong H. Kang

**Abstract**—Golomb analyzed the  $Z$ -channel, a memoryless channel with two input symbols and two output symbols, where one of the input symbols is transmitted with noise while the other is transmitted without noise, and the output symbol transmission times are equal. We generalize to the *timed  $Z$ -channel*, where the output symbol transmission times are not equal. The timed  $Z$ -channel appears as the basis for a large class of covert (communication) channels appearing in multilevel secure computer systems. We give a detailed mathematical analysis of the timed  $Z$ -channel and report a result expressing the capacity of the timed  $Z$ -channel as the log of the root of a characteristic equation. This generalizes Shannon’s work on noiseless channels for this special case. We also report a new result bounding the timed  $Z$ -channel’s capacity from below. We show how an interesting observation that Golomb reported for the  $Z$ -channel also holds for the timed  $Z$ -channel.

**Index Terms**—Channel capacity, computer network, computer security, covert channel, CPU scheduling, timing channel,  $Z$ -channel.

#### I. INTRODUCTION

In Shannon’s seminal paper, he expressed the capacity of a discrete noiseless channel with variable symbol time durations as the logarithm of the zero of an associated “polynomial,” see [26], [6] (in this correspondence we abuse the term polynomial since technically they are polynomials in the inverse of the variable). The nontrivial exponents of this polynomial are the negatives of the symbol time durations. Our main theoretical result is to extend this algebraic solution to a particular type of noisy binary input channel which has two output times.

Discrete noiseless channel with variable symbol time durations and their “noisy” generalizations are of great interest to designers of multilevel secure (MLS) computer systems [22]. We discuss an application of our analysis to such a system. In this correspondence we will restrict ourselves to a two-level system, consisting of the (security/sensitivity) levels Low and High. The Bell–LaPadula (BLP) [2] requirements are that a lower level user/process (Low) may not read from a higher user/process (High) and that High may not write to Low. However, due to the exigencies of realistic system design, it may be possible for High to *covertly* pass information to Low in violation of the BLP requirements. Such a violation is called a *covert channel* [14]. Mathematically, a covert channel is simply a communication channel with High acting as the transmitter and Low being the receiver. In this correspondence, as in [22], we are interested in covert channels where time values comprise the output alphabet. Such a covert channel is called a covert timing channel, or more simply, a *timing channel*. In the literature, timing channels often include the class of channels that use the notion of a “clock” to count the number of symbols sent in a certain time period [29].

Manuscript received November 10, 1996; revised February 27, 1998. The work of I. S. Moskowitz and S. J. Greenwald was supported by the Office of Naval Research. The material in this correspondence appeared in part, under the same title, in *Proc. IEEE Symp. Security and Privacy* (Oakland, CA, May 1996), pp. 2–11.

The authors are with the Center for High Assurance Computer Systems, Information Technology Division, Mail Code 5540, Naval Research Laboratory, Washington, DC 20375 USA.

Publisher Item Identifier S 0018-9448(98)07364-7.