NSG-5002

ELECTRICAL ENGINEERING DEPARTMENT

UNIVERSITY OF TENNESSEE

KNOXVILLE
TN 37916

National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt, Maryland 20771

FINAL REPORT. Contract NSG-5002

Architecture and Data Processing Alternatives for the Tse Computer

VOLUME 4: Image Rotation Using Tse Operations

M. H. Kao

R. E. Bodenheimer

TECHNICAL REPORT TR-EE/CS-76-4

October 1976

ARCHITECTURE AND DATA PROCESSING ALTERNATIVES FOR THE TSE COMPUTER

VOLUME 4: IMAGE ROTATION USING TSE

OPERATIONS

Robert E. Bodenheimer - Principal Investigator
Min-Hwan Kao - Co-Investigator
Department of Electrical Engineering
The University of Tennessee
Knoxville, Tennessee 37916

Final Report. NSG-5002

Period: May 1974 - August 1976

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND 20771

ABSTRACT

This research report contains results of a study on the tse computer's capability of achieving image congruence between temporal and multiple images with misregistration due to rotational differences. The coordinate transformations are obtained and a general algorithm is devised to perform image rotation using tse operations very efficiently. The details of this algorithm as well as its theoretical implications are presented. Step by step procedures of image registration are described in detail. Numerous examples are also employed to demonstrate the correctness and the effectiveness of the algorithm. Conclusions and recommendations are made for futher study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

FIGURE PAGE

FIGURE PAGE

FIGURE PAGE

CHAPTER 1

INTRODUCTION

The explosion of information has already presented a significant challenge to the conventional sequential computers. Especially, for many problems in picture or image processing applications in which very large arrays are required for reasonable resolution, even those "super computers" are becoming too small, too slow, or just simply too expensive to use. For example, in the 1980's, as many as 50,000 images are expected to be generated per day by Earth Observation type spacecraft. Meteorological and planetary spacecraft will increase this number. The need for efficient and simple processors which can handle the huge quantity of image type data sufficiently fast, at a reasonable cost, is becoming more and more urgent. To this end, parallel processing machines [1-4] such as Solomon computer and Illiac IV, etc., have been studied for years. However, most of them have not reached an operational status because of the prohibitive cost involved in their construction. In addition, their speeds have never been satisfactory for handling image type data.

In order to circumvent the challenge of the huge quantity of images generated by NASA's spacecrafts, one research group at Goddard Space Flight Center, (GSFC), generated the concept of a new family of computers, called "tse computers" [5]. These computers, utilize an entire binary image as their basic computational entity, instead of a single bit as in conventional digital computers. These computers are two dimensional expansions of conventional computers. Because of their ability to perform thousands of operations simultaneously, they have the potential of operating orders of magnitudes faster than present computers.

While the processor hardware and its architectural alternatives are still under development, the Computer Engineering Group of the University of Tennessee has been investigating its capability of achieving various image processing problems. An important area of remoting sensing is that of achieving image congruence between temporal and multiple images of the same region of interest [6,7]. Misregistration could result from the inability of the sensing system to produce congruent data due to design characteristics or the fact that the sensors are separated in space and time such that spatial alignment of the sensor is impractical or impossible. Geometric distortion, scale differences, look angle effects, and translational and rotational differences between image pairs can all combine to produce misregistration. The general registration problem is thus one of determining the location of matching context points in multiple images and alteration of the geometric relationships of the images such that the registration of each context point is achieved.

The purpose of this study is to investigate the tse computer's capability of achieving image congruence between temporal and multiple images with misregistration due to rotational differences. The task involves the study of the coordinate transformations, the development of the registration process using tse operations, the derivation of appropriate interpolative techniques, the identification of the required tse operations, and the proposal of possible hardware implementations.

CHAPTER 2

COORDINATE TRANSFORMATIONS

Figure 1 presents an image which is to be rotated clockwise by an angle θ about the tse element (p,q) in the x'y' image plane. In order to facilitate the analysis, an equivalent manipulation is one in which the image is kept fixed while the x'y' image plane is rotated counter-clockwise by the same angle θ to the new position as shown by the dash lines. With such a viewpoint, four coordinate systems are defined for the convenience of the ensuing analysis. From Figure 1, the coordinate systems for an N x N image plane are:

a) x'y' is a coordinate system associated with the orginal image plane in which the tse element at the lower left hand corner is the origin;

b) x"y" is a coordinate system associated with the original image plane, but with the rotation center (p,q) as its origin (note that the x" and y" axes are parallel to the x' and y' axes, respectively);

c) x"'y"' is a coordinate system associated with the rotated image plane and has the rotation center (p,q) as its origin; and

d) xy is a coordinate system associated with the rotated image plane, but with the tse element at the lower left hand corner as its origin (note that the x and y axes are parallel to the x"' and y"' axes, respectively).

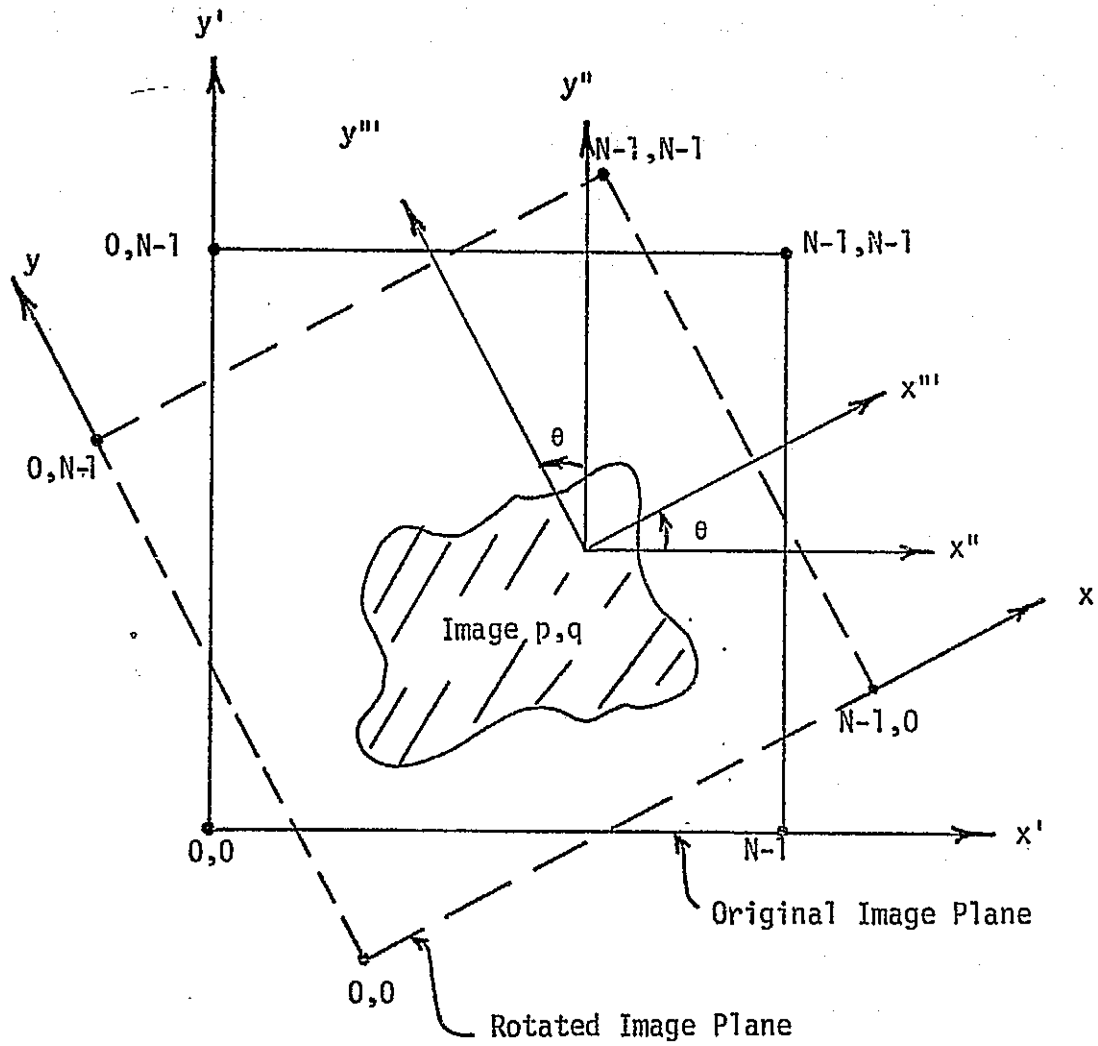


Figure 1. Coordinate transformations necessary for image rotation.

The coordinate representations in these coordinate systems are related by the following set of transformations.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x'' \\ y'' \end{bmatrix} + \begin{bmatrix} p \\ q \end{bmatrix} , \qquad (1)$$

$$\begin{bmatrix} x'' \\ y'' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x''' \\ y''' \end{bmatrix} \qquad (2)$$

and

$$\begin{bmatrix} x''' \\ y''' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} p \\ q \end{bmatrix} . \qquad (3)$$

Combining these three coordinate transformations in succession, the relationship between the original coordinates (x', y') and the final coordinates (x, y) becomes

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \left\{ \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} p \\ q \end{bmatrix} \right\} + \begin{bmatrix} p \\ q \end{bmatrix} ;$$

or,

$$x' = (x - p) \cos\theta - (y - q) \sin\theta + p \qquad (4)$$

and

$$y' = (x - p) \sin\theta + (y - q) \cos\theta + q \quad . \qquad (5)$$

Only the values at the grid points of the new image plane are of interest. For a grid point (j,k), where j and k are integers, the old coordinates (x'_j , y'_k) corresponding to this grid point are from equations (4) and (5)

$$x'_j = (j - p) \cos \theta - (k - p) \sin \theta + p \quad (6)$$

and

$$y'_k = (j - p) \sin \theta + (k - q) \cos \theta + q \quad . \quad (7)$$

Notice that x'_j and y'_k are, in general, not integers. In other words, a grid point in the new image plane is not necessarily at a grid point in the original image plane. However, the new grid point (j,k) is in a square formed by four original grid points, namely (m,n), (m+1,n), (m,n+1), and (m+1,n+1), as shown in Figure 2, where

$$m = [\, x'_j \,] \quad (8)$$

$$n = [\, y'_k \,] \quad (9)$$

and where [] is the notation for the greatest-integer function. The significance of these neighboring grid points is that the values at the new grid point (j,k), or (x'_j ,y'_k), can be obtained by interpolating the known values at the original grid points.

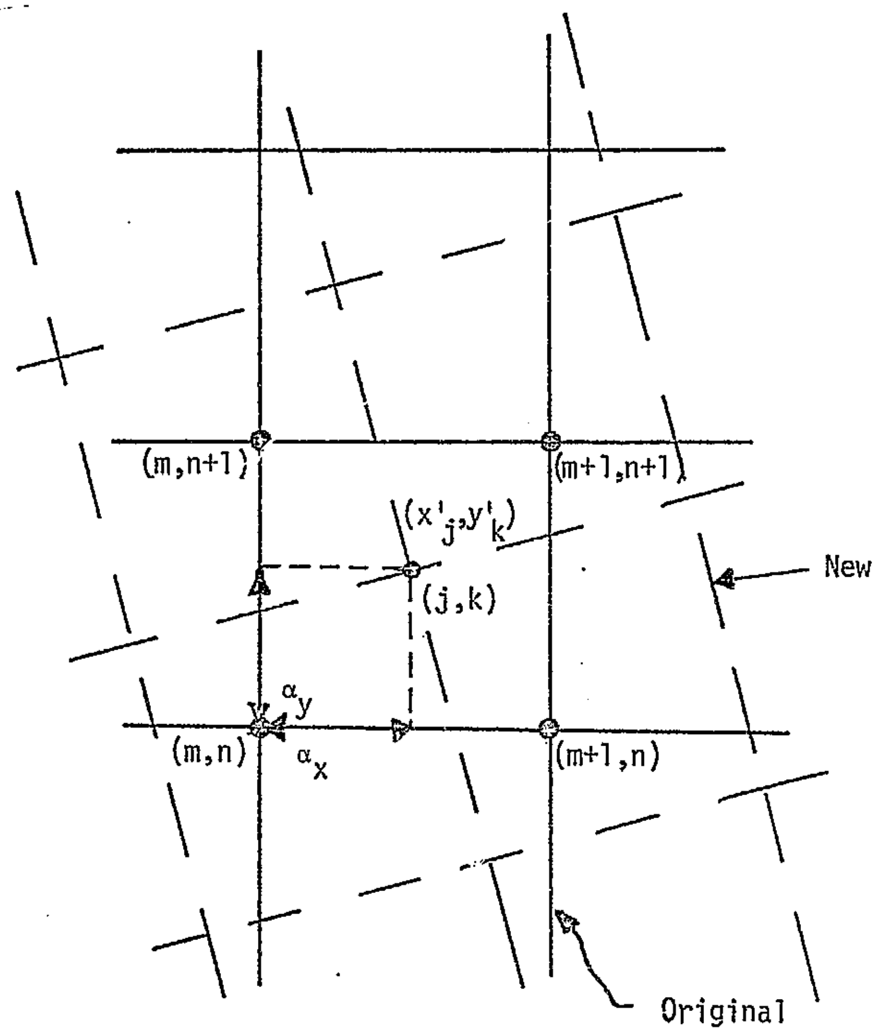


Figure 2. Relationship between original and new grid points.

CHAPTER 3

DATA INTERPOLATIONS

As presented in the previous chapter, the grid points for the new image plane are not necessarily at the grid points of the original image plane; hence, the values at these points must be obtained through some interpolative technique. A large number of interpolative techniques based on different criteria and using different numbers of neighboring grid points are available. The choice of the technique depends upon the resolution and the noise condition of the original image, as well as upon the consideration of computation time and storage. In this chapter two techniques are presented which are simple, easy to implement, and adequate for most cases.

Assigning Technique

The decision as to what value is assigned to a grid point of the new image plane is made dependent on the distance between this point and its neighboring grid points in the original image plane. A grid point is assigned the value corresponding to the value of the nearest neighboring grid point in the original image plane. Mathematically, this may be written as

$$f(j,k) = S_{m,n} \tag{10}$$

where $f(j,k)$ is the value to be assigned to the grid point (j,k) of the new image plane, and $S_{m,n}$ is the known value of the nearest original grid point. Note that m and n are now defined differently from that in equations (8) and (9).

$$m = \langle x'_j \rangle \tag{11}$$

and

$$n = \langle y'_k \rangle \tag{12}$$

where <> is the notation for "round-off function". For example, <3.4> = 3 and <3.6> = 4.

The Assigning Technique is the simplest technique for obtaining values at new grid points. For most of the images which have reasonably good resolutions, this technique is more than adequate. In addition, the technique has the advantage of not blurring the original image.

Linear Least-Square-Error Technique

A second approach is to use a linear least-square-error technique [3]. As presented in Figure 2, the square formed by the four nearest neighbors are used to assign the value of the new grid point. This technique, which has a degree of local averaging and does not use an excessive number of neighboring grid points in the interpolative process, is believed to be a very useful interpolative technique in image processing problems using tse operations. The interpolation formulae of this technique are summarized below, with the details of its derivation presented in Appendix A.

Let the known values at the four neighboring grid points be $S_{m,n}$, $S_{m+1,n}$, $S_{m,n+1}$, and $S_{m+1,n+1}$, respectively. Then the interpolated value at (j,k) is

$$f(j,k) = W_{m,n}S_{m,n} + W_{m+1,n}S_{m+1,n} + W_{m,n+1}S_{m,n+1}$$

$$+ W_{m+1,n+1}S_{m+1,n+1} \qquad (13)$$

where the W's are weights of the form

$$W_{m,n} = -\alpha_x/2 - \alpha_y/2 + 3/4 \qquad (14)$$

$$W_{m+1,n} = \alpha_x/2 - \alpha_y/2 + 1/4 \qquad (15)$$

$$W_{m,n+1} = -\alpha_x/2 + \alpha_y/2 + 1/4$$

$$W_{m+1,n+1} = \alpha_x/2 + \alpha_y/2 - 1/4$$

The distances for α_x and α_y are shown in Figure 2. Note that the linear interpolative technique has the effect of smoothing over the four neighboring grid points.

The two techniques presented above are not the only possible interpolative techniques. A large number of more sophisticated techniques using a large number of neighboring grid points and associated with specific features could be developed. This problem is not explored in this research. However, the point to be stressed is that many important features such as techniques for noise-stripping, edge and curve detection, image restoration, local averaging and image filtering, etc., could be incorporated and embedded in the interpolative procedures using tse operations.

CHAPTER 4

THE INSTRUCTION ROT p, q, θ

A general instruction for image rotation using tse operations is proposed of the form

$$\text{ROT } p, q, \theta$$

where p,q are the coordinates of the tse element about which the image is to be rotated, and θ is the desired angle of rotation. (A positive sign is adopted for clockwise rotation of the image.) With this instruction as well as the hardware 90°, 180°, and 270° image rotator proposed at GSFC/NASA [5], an image can be rotated about any element for any angle. (In fact, the rotation center can be any point, a grid point or non-grid point, within the image plane or a point outside the image plane.)

Since the quantities $\sin\theta$, $\cos\theta$, x'_j, y'_k, W's, etc., are not integers, all these quantities have to be scaled in order to implement the image rotation algorithm with fixed arithmetic. In summary, all the equations discussed above, after being scaled by a factor 2^{ℓ}, become

$$\overline{x'_j} = (j - p)\,\overline{\cos\theta} - (k - q)\,\overline{\sin\theta} + \overline{p} \qquad (18)$$

$$\overline{y'_k} = (j - p)\,\overline{\sin\theta} + (k - q)\,\overline{\cos\theta} + \overline{q} \quad . \qquad (19)$$

For the Assigning Technique,

$$m = \langle \bar{x}'_j / 2^{\ell} \rangle \quad , \tag{20}$$

$$n = \langle \bar{y}'_k / 2^{\ell} \rangle \quad , \tag{21}$$

and

$$f(j,k) = S_{m,n} \quad . \tag{22}$$

For the linear Least-Square-Error Technique,

$$m = [\bar{x}'_j / 2^{\ell}] \quad , \tag{23}$$

$$n = [\bar{y}'_k / 2^{\ell}] \quad , \tag{24}$$

$$\bar{\alpha}_x = \bar{x}'_j - m \cdot 2^{\ell} \ , \tag{25}$$

$$\bar{\alpha}_y = \bar{y}'_k - n \cdot 2^{\ell} \ , \tag{26}$$

$$\overline{W_{m,n}} = -\bar{\alpha}_x/2 - \bar{\alpha}_y/2 + (3/4) \cdot 2^{\ell} \ , \tag{27}$$

$$\overline{W_{m+1,n}} = \bar{\alpha}_x/2 - \bar{\alpha}_y/2 + (1/4) \cdot 2^{\ell} \quad , \tag{28}$$

$$\overline{W_{m,n+1}} = -\bar{\alpha}_x/2 + \bar{\alpha}_y/2 + (1/4) \cdot 2^{\ell} \ , \tag{29}$$

$$\overline{W_{m+1,n+1}} = \bar{\alpha}_x/2 + \bar{\alpha}_y/2 - (1/4) \cdot 2^{\ell} \quad , \tag{30}$$

$$\bar{f}(j,k) = \overline{W}_{m,n} S_{m,n} + \overline{W}_{m+1,n} S_{m+1,n} + \overline{W}_{m,n+1} S_{m,n+1} + \overline{W}_{m+1,n+1} S_{m+1,n+1} \quad , \tag{31}$$

and

$$f(j,k) = \bar{f}(j,k)/2^{\ell} \quad . \tag{32}$$

A super bar is attached to a quantity to denote that the quantity is scaled by the scaled factor 2^{ℓ} (i.e., $\bar{x} = x \cdot 2^{\ell}$).

Basically, the implementation of equations (18) through (32) using tse operations is straightforward. After the constant planes are generated, equations (18) and (19) are computed by at most four multiplications and six addition/subtractions. The divisions and multiplications which appear in equations (20), (21), (23) through (30), and (32) are all powers of 2; hence, expect for the addition/subtraction and shift operations, no division or multiplications are actually involved in the computations of these equations. The generation of the S planes in equation (22) of the Assigning Technique, or in equation (31) of the Linear Least-Square-Error Technique, requires a series of slide operations. This procedure is very involved and is described in the following chapters.

CHAPTER 5

MAGNITUDE OF THE SLIDES

The implementation of the image rotation algorithm requires the generation of the $S_{m,n}$ plane for using the Assigning Technique, or the generation of four planes, namely $S_{m,n}$, $S_{m+1,n}$, $S_{m,n+1}$, and $S_{m+1,n+1}$ for using the Linear Least-Square-Error Technique. Definitions of m,n in both techniques are different in that in the former case, m and n are the "round-off" values of x'_j and y'_k, respectively, as given in equations (11) and (12), while in the latter case, m and n are the "greatest-integer" values of x'_j and y'_k, as given in the equations (8) and (9). In spite of this difference, they resemble each other in appearance and differ in values by at most one in all cases. Therefore, only one technique needs to be discussed in detail. Once the problem is solved for this technique, the other case can be solved with minor modifications.

Note that for the Linear Least-Square-Error Technique, the $S_{m,n}$ plane is the plane in which element (j,k) contains the value originally at the element (m,n), which is the old grid point on the lower left side of the new grid point (j,k) as shown in Figure 2. Similarly, $S_{m+1,n}$, $S_{m,n+1}$, and $S_{m+1,n+1}$ are defined in the same manner. The relationships between (j,k) and (m,n) are given in equations (6) through (9).

Starting with the original image, the $S_{m,n}$ plane is generated by sliding the value at each point (m,n) to its corresponding point (j,k) in a systematic way. With regard to the fact that the image can only be slid either horizontally or vertically, the slides are resolved into "slide-up (down)" and "slide-right (left)" operations.

Notice that the new grid points (j,k) and the squares formed by the four neighboring original grid points are not necessarily a one-to-one correspondence. For instance, as shown in Figure 3, two new grid points, say (j_1,k_1) and (j_2,k_2), may fall into the same square formed by the four original grid points; in other words, the point A is the (m,n) point corresponding to (j_1,k_1) as well as (j_2,k_2). Hence, when generating the $S_{m,n}$ plane, there is a requirement to slide the value at point A to these two neighboring points. On the other hand, there is a possibility that no new grid point falls into a square formed by the four original grid points (the square CDEF as shown in Figure 3). In such a case, when generating the $S_{m,n}$ plane, one does not really need the value originally at point C since the point C is not an (m,n) point corresponding to any new grid point. Because of the fact that (j,k) and (m,n) are not necessarily a one-to-one correspondence and, also, since the slide operation itself cannot slide the data at (m,n) exactly to its destination (j,k), extraction of data through masks is required to complete the generation of the S planes.

Under the rotation of the angle θ, any grid point S(m,n) is rotated to a new position S' through the arc SS' as shown in Figure 4. This displacement can be resolved in to a horizontal slide ℓ_h and a vertical slide ℓ_v. The values of ℓ_h and ℓ_v are found to be

$$\ell_h = \sqrt{(m-p)^2 + (n-q)^2}\ \cos\{\tan^{-1}(\frac{n-q}{m-p}) - \theta\} - (m-p)$$

$$= -(1-\cos\theta)(m-p) + (n-q)\sin\theta \qquad (33)$$

and

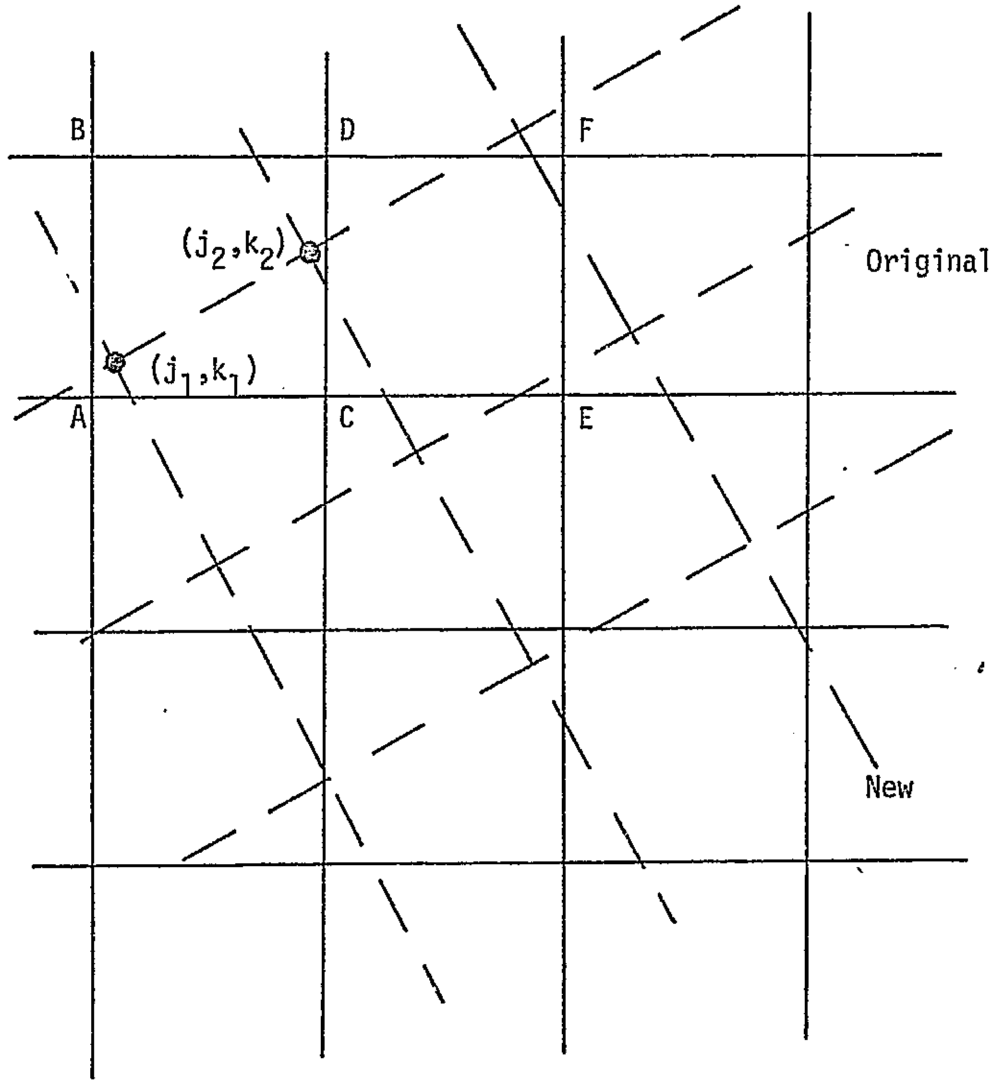


Figure 3. Example that (j,k) and (m,n) are not one-to-one corresponding.

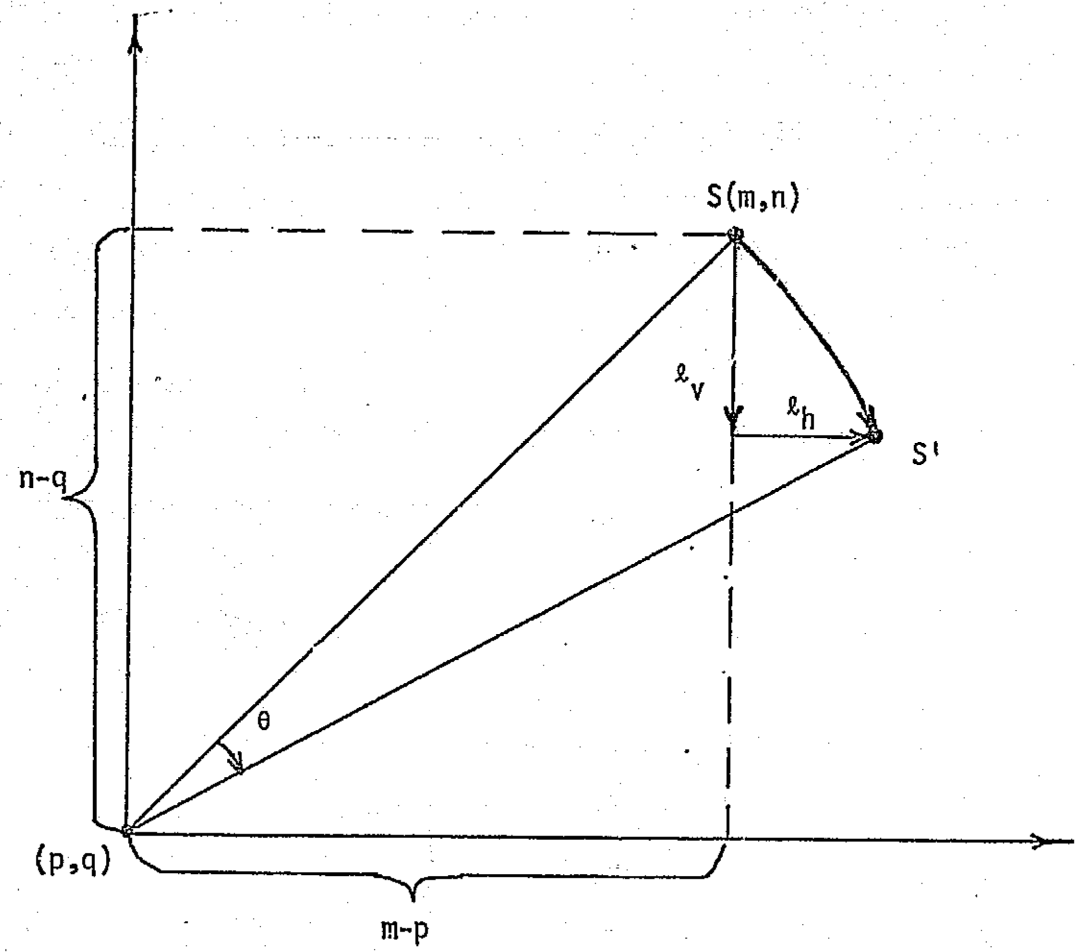


Figure 4. Magnitude of the slide.

$$\ell_v = \sqrt{(m-p)^2 + (n-q)^2} \sin \{ \tan^{-1} (\frac{n-q}{m-p}) - \theta \} - (n-q)$$

$$= - (1 - \cos\theta)(n-q) - (m-p) \sin \theta \qquad (34)$$

Since the magnitude of the slides has to be an integer, and also, because of the relationships between (j,k) and (m,n) as given by equations (11) and (12), the value at (m,n) for the Assigning Technique should be slid to the position (j,k) through a vertical slide,

$$V = \langle \ell_v \rangle \quad , \qquad (35)$$

and a horizontal slide,

$$H = \langle \ell_h \rangle \quad , \qquad (36)$$

in order to generate $S_{m,n}$.

For the Linear Least-Square-Error Technique, the situation is more complicated. Four possible cases for clockwise rotation are shown in Figure 5, in which a square formed by four original grid points A,B,C, and D is rotated to the new position A'B'C'D'. Figure 5(a) depicts the case in which point A is the corresponding (m,n) point of the new grid point E; therefore, the relationships between V, H and ℓ_v, ℓ_h are

$$V = [\ell_v] + 1$$

and

$$H = [\ell_h] + 1 \quad .$$

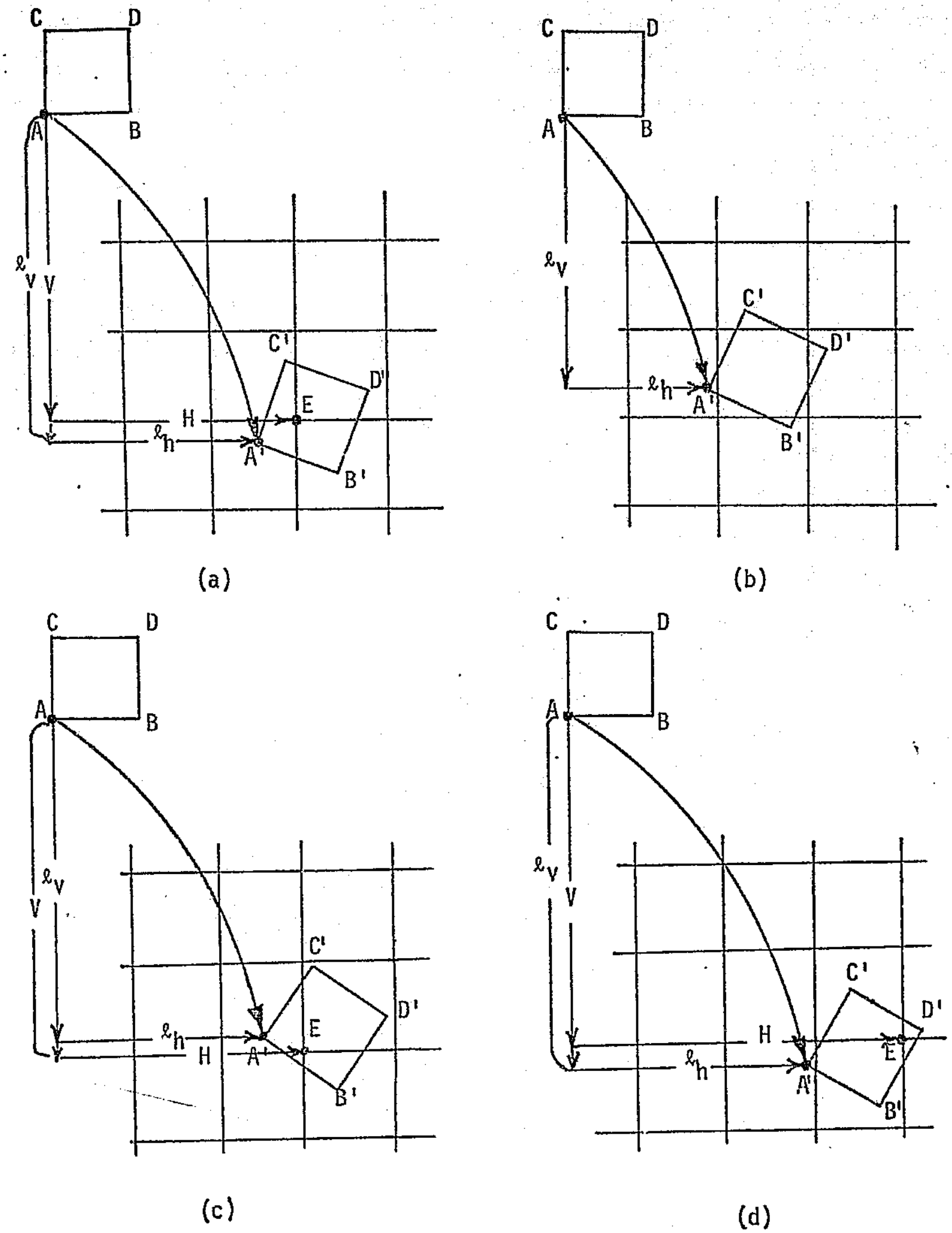


Figure 5. Four possible cases in clockwise rotation.

Figure 5(b) shows the case in which no new grid point falls into the square A'B'C'D'; hence the values of V and H are actually undefined. The point A is not an (m,n) point corresponding to any new grid point. When generating the $S_{m,n}$ plane, one does not require the value originally at point A. Figure 5(c) presents the case in which V and H are given by

$$V = [\ell_v]$$

$$H = [\ell_h] + 1 \quad .$$

Figure 5(d) demonstrates the last possibility, one in which V and H are

$$V = [\ell_v] + 1$$

and

$$H = [\ell_h] + 2 \quad .$$

Since these situations are so involved, and since the V and H values calculated here are only for use in analysis, the above cases are approximated by the following unique expressions, with a possible difference of magnitude 1.

$$V = \langle \ell_v \rangle \tag{37}$$

and

$$H = \langle \ell_h \rangle + 1 \quad ,$$

where <> is the notation for "round-off function".

Similarly, for counter-clockwise rotation, the four possible cases are shown in Figure 6, in which the values of V and H are

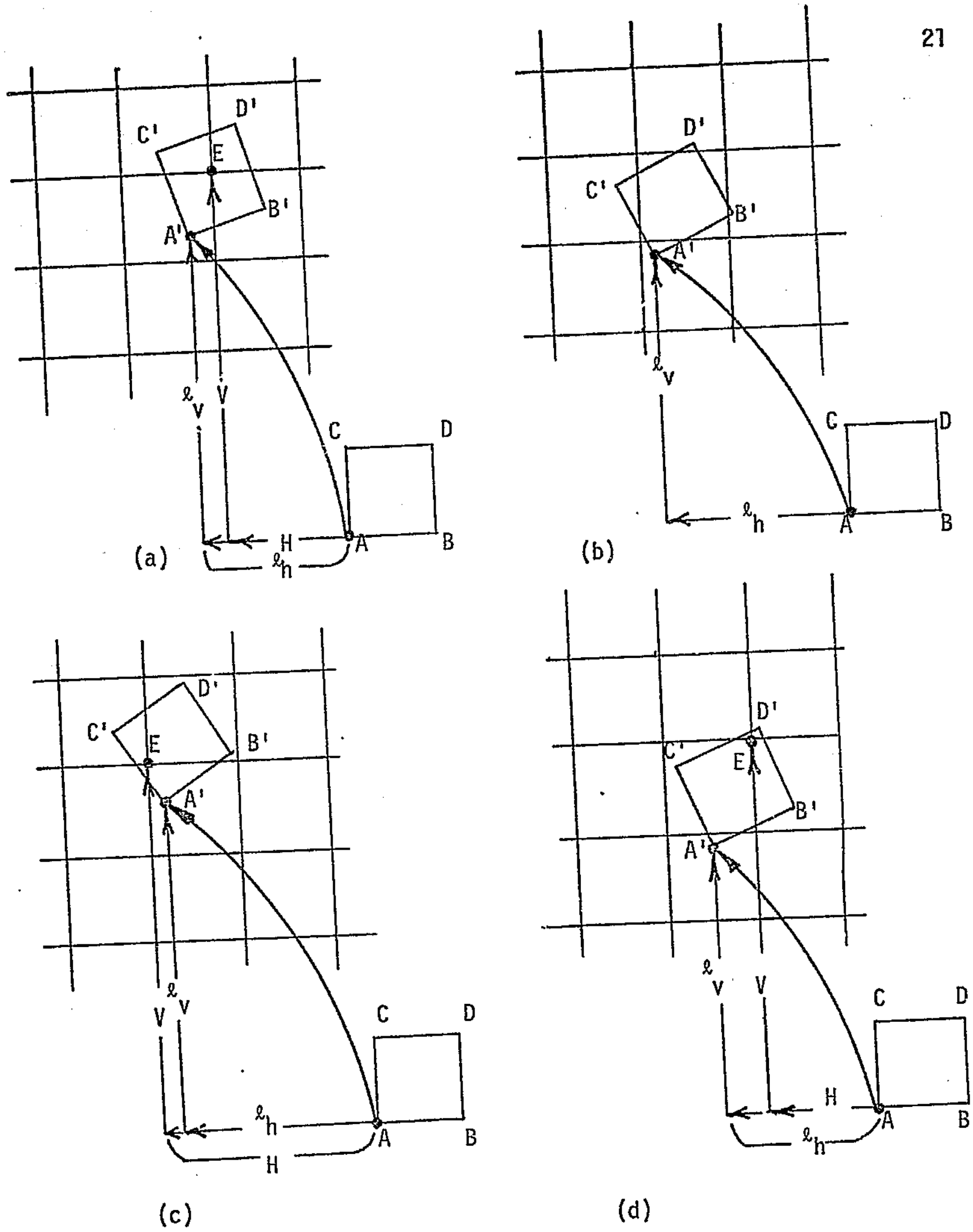


Figure 6. Four possible cases in counter-clockwise rotation.

$$V = [\ell_v] + 1$$

$$H = [\ell_h] + 1 \quad ,$$

$$\left.\begin{matrix} V \\ H \end{matrix}\right\} \text{undefined} \quad ,$$

$$V = [\ell_v] + 1$$

$$H = [\ell_h] \quad ,$$

and

$$V = [\ell_v] + 2$$

$$H = [\ell_h] + 1 \quad .$$

These cases can similarly be approximated by the unique expressions

$$V = \langle \ell_v \rangle + 1 \tag{39}$$

$$H = \langle \ell_h \rangle \quad . \tag{40}$$

Observe that the above three sets of equations, namely equations (35) and (36) for Assigning Technique, equations (37) and (38) for the clockwise rotation using the Linear Least-Square-Error Technique, and equations (39) and (40) for the counter-clockwise rotation using the Linear Least-Square-Error Technique, have exactly the same form. Therefore, only one case needs to be presented and discussed in detail; hereafter, the derivations for other cases are obtained by minor modifications.

As an example, consider the 37° counter-clockwise rotation of a 25 x 25 image about the tse element (9,9). The values of V and H calculated for the whole plane by equations (33) through (36) are shown in Figure 7 and 8, respectively, in which the circled element is the rotation center, (9,9). Notice that a positive value of V means a slide-up operation, while a negative value of V means a slide-down operation. Similarly, a positive value of H means a slide-right operation, while a negative value of H means a slide-left operation. The lines dividing the V-plane and H-plane into zones are employed for the convenience of analysis in the next section. Figures 9 and 10 show the V-plane and H-plane of another example, in which the image is rotated by a smaller angle (17°). Observe that the zones are wider for smaller angles of rotation.

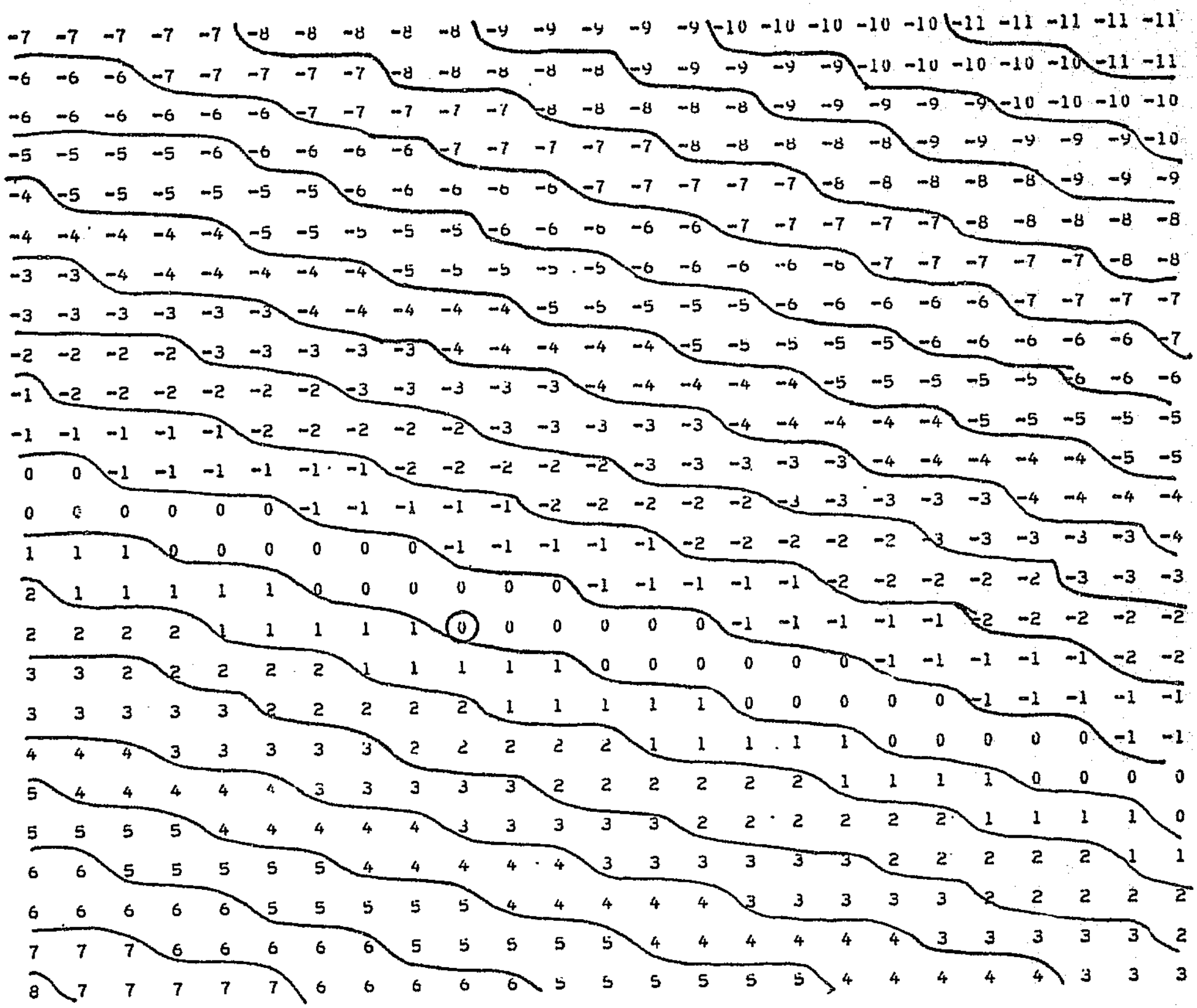


Figure 7. H patterns (rotation angle: -37°). H: number of horizontal slides required.

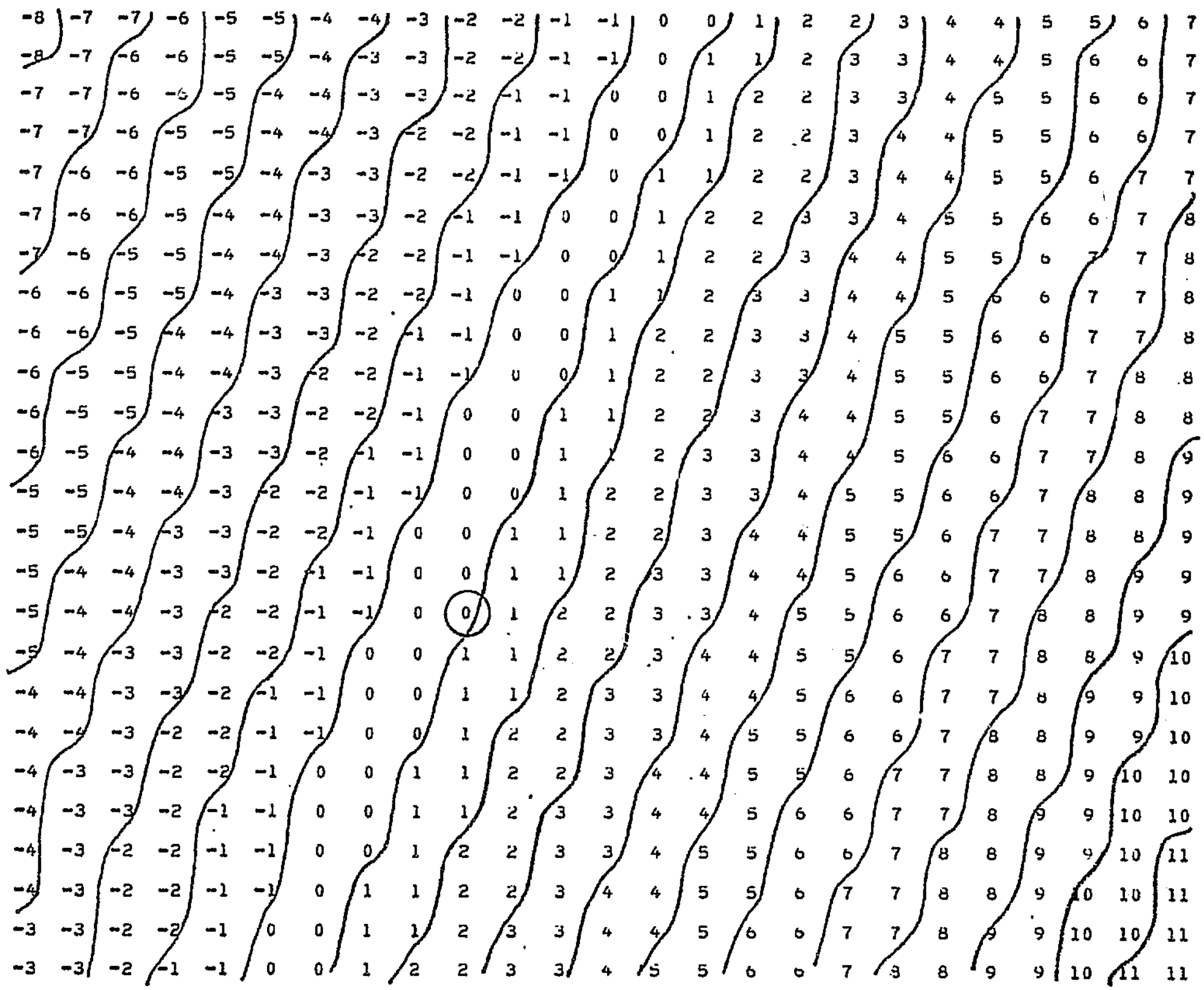


Figure 8. V patterns (rotation angle:-37°). V: number of vertical slides required.

-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5
-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-5	-5	-5	-5	-5	-5	-5
-3	-3	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-5
-3	-3	-3	-3	-3	-3	-3	-3	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-4	-4	-4
-2	-2	-2	-2	-2	-2	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3
-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3
-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-3	-3	-3	-3	-3	-3
-1	-1	-1	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-2	-2	-2	-2	-2	-2	-2	-2
0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-2	-2
0	0	0	0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	-1	-1	-1
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1
3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2	2	2

Figure 9. H patterns (rotation angle: -17°). H:number of horizontal slides required.

-3	-3	-3	-2	-2	-2	-2	-1	-1	-1	0	0	0	1	1	1	1	2	2	2	3	3	3	3	4
-3	-3	-3	-2	-2	-2	-2	-1	-1	-1	0	0	0	1	1	1	1	2	2	2	3	3	3	4	4
-3	-3	-3	-2	-2	-2	-1	-1	-1	-1	0	0	0	1	1	1	1	2	2	2	3	3	3	4	4
-3	-3	-3	-2	-2	-2	-1	-1	-1	-1	0	0	0	1	1	1	2	2	2	2	3	3	3	4	4
-3	-3	-3	-2	-2	-2	-1	-1	-1	0	0	0	0	1	1	1	2	2	2	2	3	3	3	4	4
-3	-3	-3	-2	-2	-2	-1	-1	-1	0	0	0	0	1	1	1	2	2	2	3	3	3	3	4	4
-3	-3	-2	-2	-2	-2	-1	-1	-1	0	0	0	0	1	1	1	2	2	2	3	3	3	3	4	4
-3	-3	-2	-2	-2	-2	-1	-1	-1	0	0	0	1	1	1	1	2	2	2	3	3	3	3	4	4
-3	-3	-2	-2	-2	-1	-1	-1	-1	0	0	0	1	1	1	1	2	2	2	3	3	3	4	4	4
-3	-3	-2	-2	-2	-1	-1	-1	-1	0	0	0	1	1	1	2	2	2	2	3	3	3	4	4	4
-3	-3	-2	-2	-2	-1	-1	-1	-1	0	0	0	1	1	1	2	2	2	2	3	3	3	4	4	4
-3	-3	-2	-2	-2	-1	-1	-1	0	0	0	0	1	1	1	2	2	2	2	3	3	3	4	4	4
-3	-2	-2	-2	-2	-1	-1	-1	0	0	0	0	1	1	1	2	2	2	3	3	3	3	4	4	4
-3	-2	-2	-2	-2	-1	-1	-1	0	0	0	1	1	1	1	2	2	2	3	3	3	3	4	4	4
-3	-2	-2	-2	-2	-1	-1	-1	0	0	0	1	1	1	1	2	2	2	3	3	3	4	4	4	4
-3	-2	-2	-2	-1	-1	-1	-1	0	0	0	1	1	1	1	2	2	2	3	3	3	4	4	4	4
-3	-2	-2	-2	-1	-1	-1	-1	0	0	0	1	1	1	2	2	2	2	3	3	3	4	4	4	4
-3	-2	-2	-2	-1	-1	-1	-1	0	0	0	1	1	1	2	2	2	2	3	3	3	4	4	4	5
-3	-2	-2	-2	-1	-1	-1	0	0	0	0	1	1	1	2	2	2	2	3	3	3	4	4	4	5
-2	-2	-2	-2	-1	-1	-1	0	0	0	0	1	1	1	2	2	2	3	3	3	3	4	4	4	5
-2	-2	-2	-2	-1	-1	-1	0	0	0	1	1	1	1	2	2	2	3	3	3	3	4	4	4	5
-2	-2	-2	-2	-1	-1	-1	0	0	0	1	1	1	1	2	2	2	3	3	3	4	4	4	4	5
-2	-2	-2	-1	-1	-1	-1	0	0	0	1	1	1	1	2	2	2	3	3	3	4	4	4	4	5
-2	-2	-2	-1	-1	-1	-1	0	0	0	1	1	1	2	2	2	2	3	3	3	4	4	4	4	5
-2	-2	-2	-1	-1	-1	0	0	0	0	1	1	1	2	2	2	2	3	3	3	4	4	4	5	5

Figure 10. V patterns (rotation angle: -17°). V: number of vertical slides required.

CHAPTER 6

ALGORITHM FOR THE SLIDE PROCEDURE

The required horizontal and vertical slides vary from point to point in a very irregular manner as presented in Figure 7 and 8. However, by dividing the V-plane and H-plane into zones as shown in the planes, advantage can be taken of the special "sawtooth" patterns to devise the following systematic three-step slide procedure to slide the data at (m,n) to exactly the position (j,k) or to one of its neighboring grid points. Extraction of the data through masks then follows to complete the generation of the S planes, as presented in the next chapter.

Step 1. Alignment. Align the pattern by sliding the image columnwise. As shown in Figure 11, all elements in the same row have the same value of H (that is, require the same amount of horizontal slide) after the columnwise slides except for a truncation difference with a magnitude no more than one. As shown in Figure 12, the V pattern also appears to be more regular after this step.

Step 2. Sliding horizontally. Since the H pattern is aligned, elements in the same row require the same amount of horizontal slide. Rowwise slides are executed in this step. As shown in Figure 13, the resulting H plane (after row slides) contains only 0, +1, or -1 as

***************************							-8	-8	-8	-9	-9	-8	-9	-9	-8	-9	-9	-8	-9	-9	-8	-9	-9	-8
***************				-7	-8	-8	-7	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8
***	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-8	-7	-7	-8	-7	-7	-8	-7	-7	-8	-7	-7	-8	-7
-7	-6	-6	-7	-6	-6	-7	-6	-6	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7
-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6
-6	-5	-5	-5	-5	-5	-5	-5	-5	-5	-6	-6	-5	-6	-6	-5	-6	-6	-5	-6	-6	-5	-6	-6	-5
-5	-5	-5	-5	-4	-5	-5	-4	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5
-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-5	-4	-4	-5	-4	-4	-5	-4	-4	-5	-4	-4	-5	-4
-4	-3	-4	-4	-3	-3	-4	-3	-3	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3
-3	-2	-2	-2	-2	-2	-2	-2	-2	-2	-3	-3	-2	-3	-3	-2	-3	-3	-2	-3	-3	-2	-3	-3	-2
-2	-2	-2	-2	-1	-2	-2	-1	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-2	-1	-1	-2	-1	-1	-2	-1	-1	-2	-1	-1	-2	-1
-1	0	-1	-1	0	0	-1	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	-1	0
0	1	1	0	[illegible]	[illegible]	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	[illegible]	[illegible]	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	[illegible]	2	2	2	2	2	1	1	2	1	1	2	2	1	2	2	1	2	2	1	2
2	3	2	2	3	2	2	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3	2	3	3	2	3	3	3	3	3	2	3	3	3	3
3	4	4	3	4	4	3	4	4	3	3	3	3	3	3	3	3	3	4	3	3	4	3	3****	
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4***************				
5	5	5	5	5	5	5	5	5	5	4	4	5	4	4	5	5	4***************************							
5	6	5	5	6	5	5	6	5	5	5	5	5	5	5**************************************										
6	6	6	6	6	6	6	6	6	6	6	5**													

Figure 11. H patterns after horizontal line-up. Step 1.

***************************							-4	-3	-2	-2	-1	-2	-1	0	0	0	1	1	1	2	1	2	3	3
***************				-4	-4	-3	-3	-3	-2	-2	-1	-1	-1	0	0	0	1	1	1	2	2	2	3	3
***	-5	-5	-4	-4	-4	-3	-3	-3	-2	-1	-1	-1	-1	0	0	0	1	1	2	2	2	3	3	3
-5	-5	-4	-4	-4	-3	-3	-3	-2	-2	-1	-1	-1	0	0	0	1	1	1	2	2	2	3	3	3
-5	-5	-4	-4	-4	-3	-3	-3	-2	-2	-1	-1	-1	0	1	0	1	2	1	2	3	2	3	3	3
-4	-5	-4	-3	-4	-3	-2	-3	-2	-1	-1	0	-1	0	1	1	1	2	2	2	3	2	3	4	3
-4	-4	-4	-3	-3	-3	-2	-2	-2	-1	-1	0	0	0	1	1	1	2	2	2	3	3	3	4	4
-4	-4	-4	-3	-3	-3	-2	-2	-2	-1	0	0	0	1	1	1	1	2	2	2	3	3	3	4	4
-4	-4	-3	-3	-3	-2	-2	-2	-1	-1	0	0	0	1	1	1	2	2	2	3	3	3	4	4	4
-4	-4	-3	-3	-3	-2	-2	-2	-1	-1	0	0	0	1	1	1	2	2	2	3	3	3	4	4	4
-3	-4	-3	-2	-3	-2	-1	-2	-1	0	0	1	0	1	2	1	2	3	2	3	4	3	4	5	4
-3	-3	-3	-2	-2	-2	-1	-1	-1	0	0	1	1	1	2	2	2	3	3	3	4	4	4	5	5
-3	-3	-3	-2	-2	-2	-1	-1	-1	0	0	1	1	1	2	2	2	3	3	3	4	4	4	5	5
-3	-3	-2	-2	-2	-1	-1	-1	0	0	1	1	1	2	2	2	3	3	3	4	4	4	5	5	5
-3	-3	-2	-2	-2	-1	-1	-1	0	0	1	1	1	2	2	2	3	3	3	4	4	4	5	5	5
-2	-3	-2	-1	-2	-1	0	-1	0	0	1	2	1	2	3	2	3	4	3	4	5	4	5	6	5
-2	-2	-2	-1	-1	-1	0	0	0	1	1	2	2	2	3	3	3	4	4	4	5	5	5	6	6
-2	-2	-2	-1	-1	-1	0	0	0	1	1	2	2	2	3	3	3	4	4	4	5	5	5	6	6
-2	-2	-1	-1	-1	0	0	0	0	1	2	2	2	3	3	3	4	4	4	5	5	5	6	6	6
-2	-2	-1	-1	-1	0	0	0	1	1	2	2	2	3	3	3	4	4	4	5	5	5	6	6	6
-1	-2	-1	0	-1	0	1	0	1	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7****	
-1	-1	-1	0	0	0	1	0	1	2	2	3	3	3	4	4	4	5	5	5	6***************				
-1	-1	-1	0	0	0	1	1	1	2	2	3	3	3	4	4	4	5***************************							
-1	-1	0	0	0	0	1	1	1	2	3	3	3	4	4***************************************										
-1	-1	0	0	0	1	1	1	2	2	3	3***													

Figure 12. V patterns after horizontal line-up. Step 1..

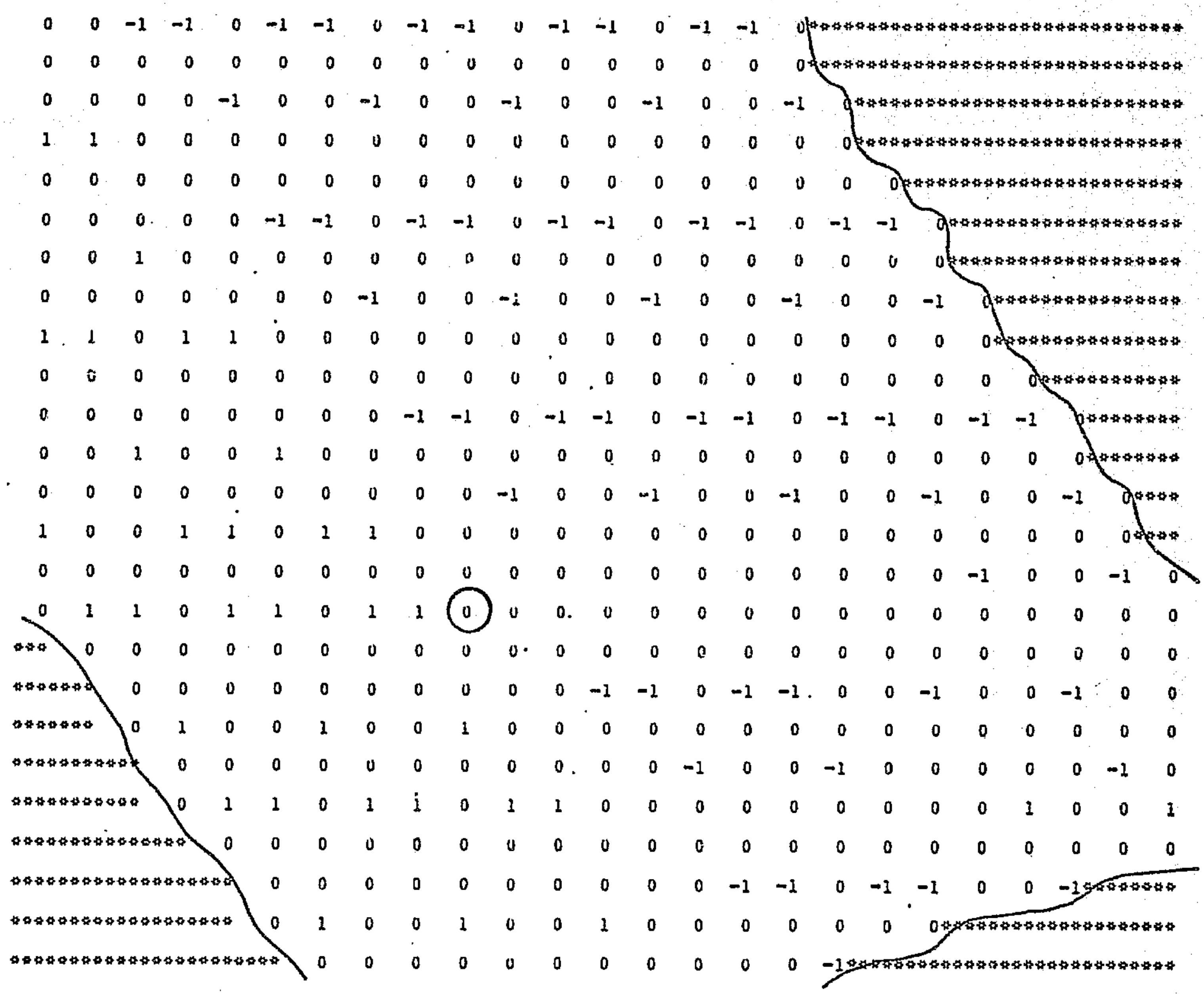


Figure 13. H patterns after horizontal slide. Step 2.

elements, which indicates that all horizontal slides are executed with the exception that some elements need to be slid one more position, left or right.

Step 3. Sliding vertically. As is proven in Appendix B, the V pattern is automatically aligned after Steps 1 and 2; in other words, all elements in the same column have the same value of V as can be seen in Figure 14, except for a truncation difference with a magnitude no more than one. This allows the slide procedure to be completed by sliding the image columnwise. As can be seen from the results in Figures 15 and 16, both the H and V planes now contain only 0, +1, or -1, indicating that all the (m,n) points have been slid to exactly the destination (j,k) or to one of its neighboring grid points. With this result, the generation of the S planes can be completed simply by the "data extraction" technique described in the following chapter.

The important point of the three-step slide procedure is the automatic alignment of the V pattern after the second step, which makes the columnwise slide in Step 3 possible.

Observe that the above slide procedure starts with a vertical slide (H pattern alignment), follows with a horizontal slide, and finally concludes with another vertical slide. The roles of horizontal slide and vertical slide can be exchanged. In other words, a slide procedure could be generated which would accomplish the same result with a horizontal slide (V pattern alignment), followed by a vertical slide,

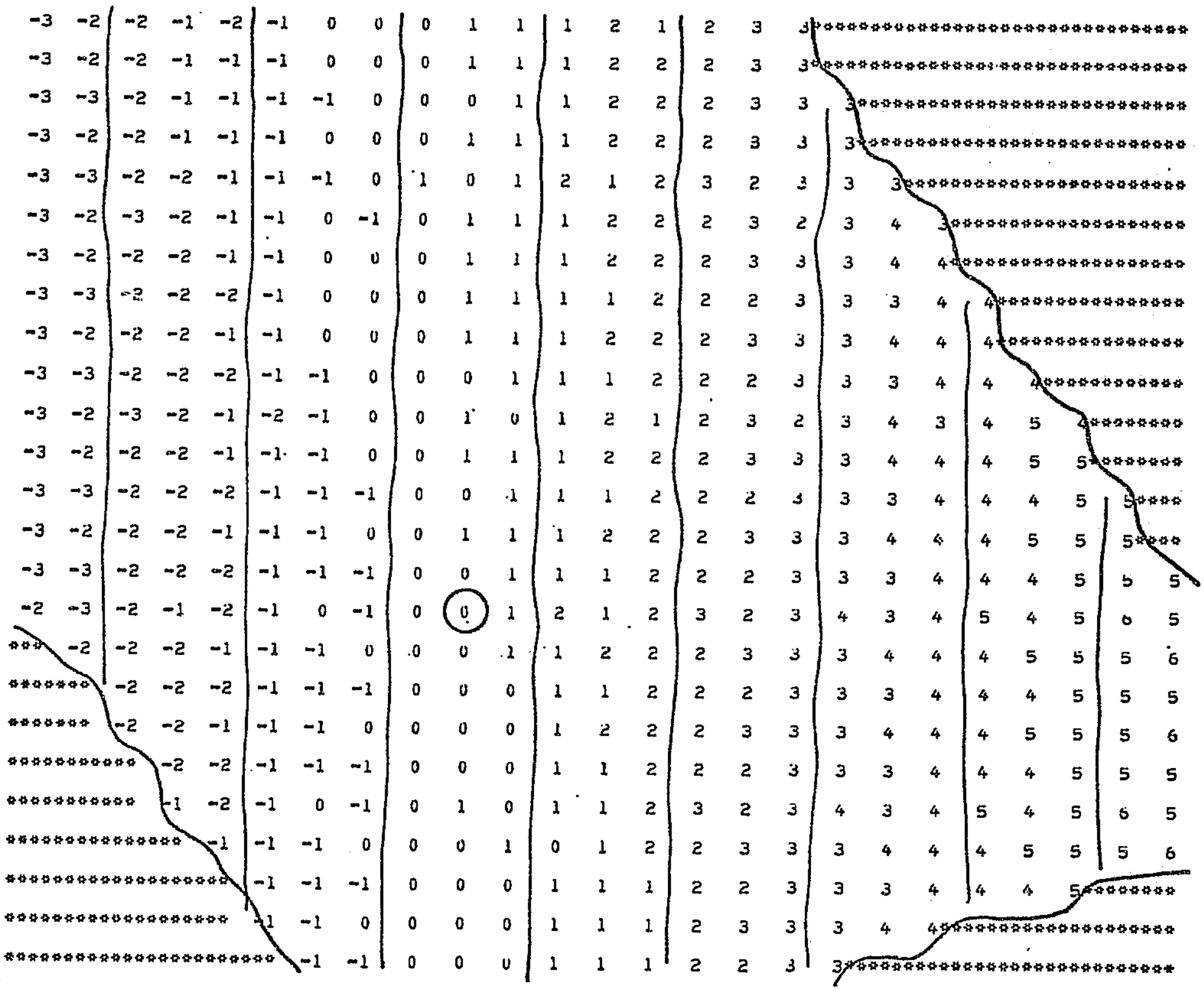


Figure 14. V patterns after horizontal slide. Step 2.

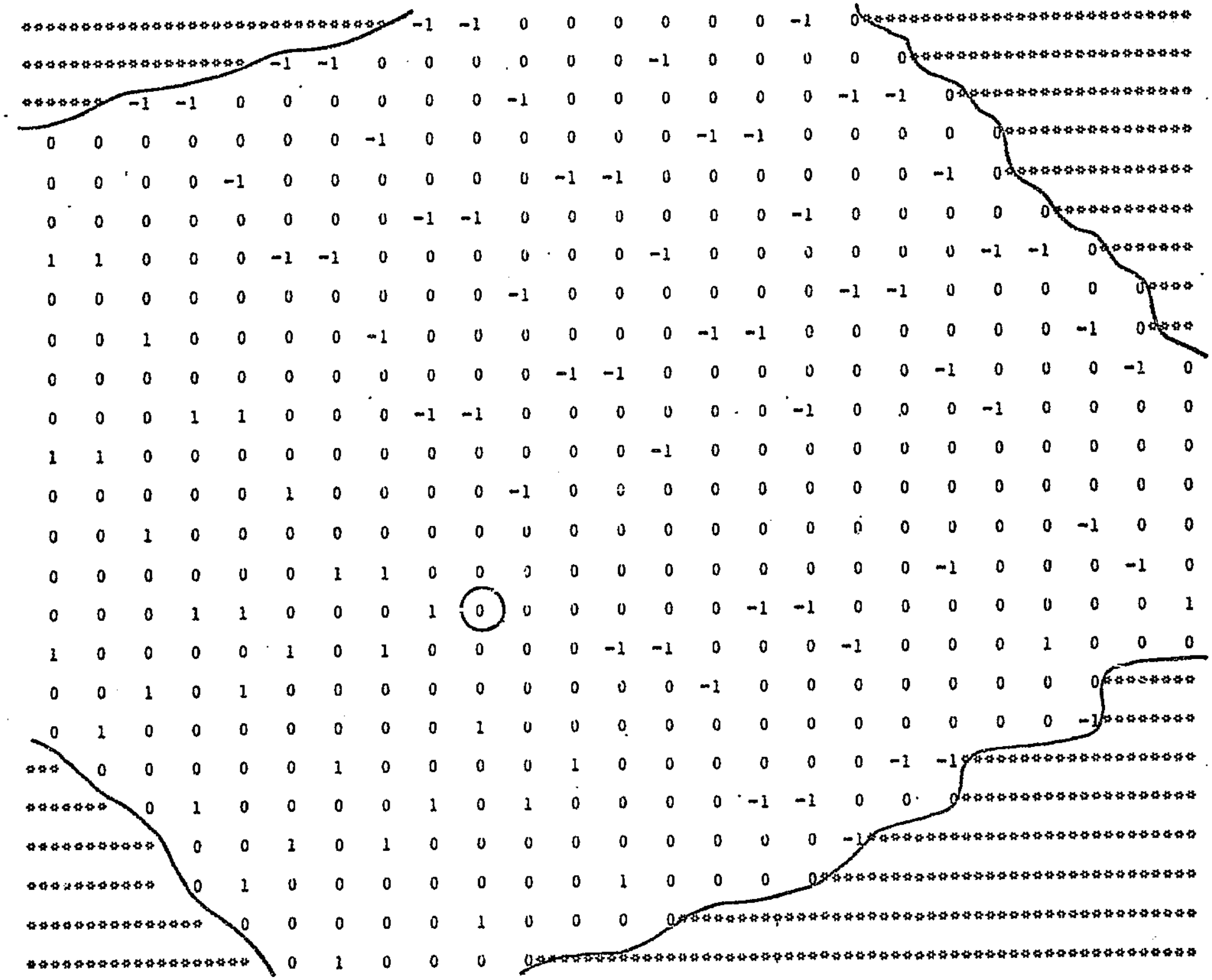


Figure 15. H patterns final. Step 3.

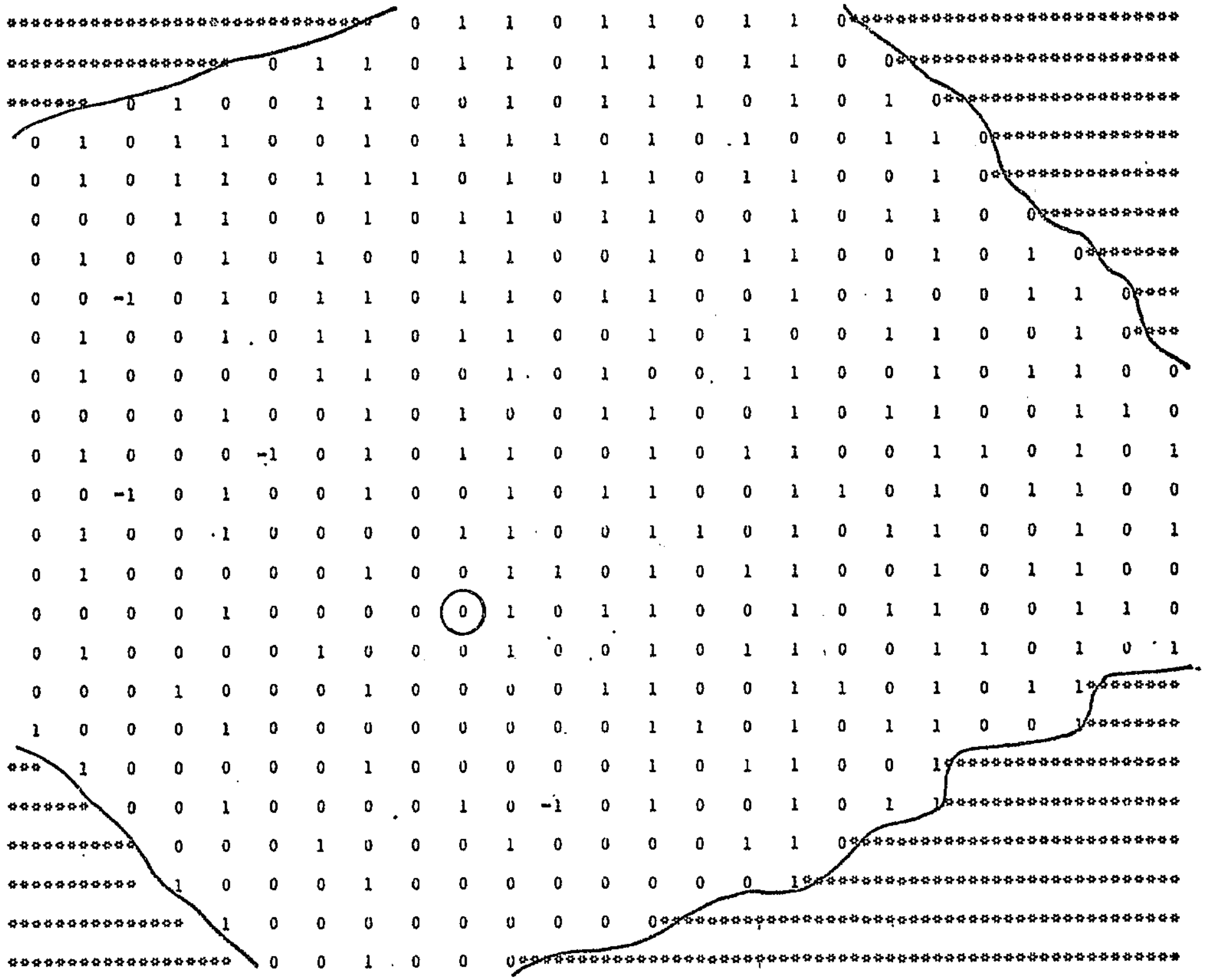


Figure 16. V patterns final. Step 3.

and finally, terminated with another horizontal slide.

CHAPTER 7

ALGORITHM FOR EXTRACTION OF THE DATA

In the last chapter, the use of a V-plane and a H-plane are employed to provide insight into the three-step slide algorithm and to demonstrate the success of this algorithm in sliding the data to the neighboring grid points of their destination. In a practical application, these two planes are not really computed and slid. The only entities which are slid are the elements of the image and the coordinates of the original grid points. On the other hand, the image is slid to generate the desired data plane. The coordinates of the original grid points are slid to keep track of the slides and, finally, are used to generate the masks as required for the process of extracting data.

The details of this data extraction technique are explained through use of an example. Consider a -37° rotation of a 20 x 20 image, with the tse element (0,0) as the rotation center. The desired final destination of (m,n) as computed by equations (8) and (9) are shown in Figure 17; the original coordinates of (m,n) are presented in Figure 18. To generate the S planes, the image is slid by the three-step slide procedure. The coordinate plane is also slid at the same time to keep track of the (m,n) points. Planes in Figures 19, 20, and 21 show the positions of these (m,n) points after Steps 1, 2, and 3, respectively. Carefully checking the actual position after the slid procedure in Figure 21 against the desired destination in Figure 17, one finds that for each desire (m,n) point in Figure 17 its actual position is either

11,15	12,14	12,13	13,13	14,12	15,12	16,11	16,11	17,10	18, 9	19, 9	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**
10,14	11,13	12,13	13,12	13,11	14,11	15,10	16,10	17, 9	17, 8	18, 8	19, 7	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**
10,13	10,12	11,12	12,11	13,11	14,10	14, 9	15, 9	16, 8	17, 8	18, 7	18, 6	19, 6	**,**	**,**	**,**	**,**	**,**	**,**	**,**
9,12	10,12	11,11	11,11	12,10	13, 9	14, 9	15, 8	15, 8	16, 7	17, 6	18, 6	19, 5	19, 5	**,**	**,**	**,**	**,**	**,**	**,**
8,11	9,11	10,10	11,10	12, 9	12, 9	13, 8	14, 7	15, 7	16, 6	16, 6	17, 5	18, 4	19, 4	**,**	**,**	**,**	**,**	**,**	**,**
8,11	9,10	9, 9	10, 9	11, 8	12, 8	13, 7	13, 7	14, 6	15, 5	16, 5	17, 4	17, 4	18, 3	19, 2	**,**	**,**	**,**	**,**	**,**
7,10	8, 9	9, 9	10, 8	10, 7	11, 7	12, 6	13, 6	14, 5	14, 4	15, 4	16, 3	17, 3	18, 2	18, 2	19, 1	**,**	**,**	**,**	**,**
7, 9	7, 8	8, 8	9, 7	10, 7	11, 6	11, 6	12, 5	13, 4	14, 4	15, 3	15, 3	16, 2	17, 1	18, 1	19, 0	19, 0	**,**	**,**	**,**
6, 8	7, 8	8, 7	8, 7	9, 6	10, 5	11, 5	12, 4	12, 4	13, 3	14, 2	15, 2	16, 1	16, 1	17, 0	**,**	**,**	**,**	**,**	**,**
5, 7	6, 7	7, 6	8, 6	9, 5	9, 5	10, 4	11, 3	12, 3	13, 2	13, 2	14, 1	15, 0	16, 0	**,**	**,**	**,**	**,**	**,**	**,**
5, 7	6, 6	6, 5	7, 5	8, 4	9, 4	10, 3	10, 3	11, 2	12, 1	13, 1	14, 0	14, 0	**,**	**,**	**,**	**,**	**,**	**,**	**,**
4, 6	5, 5	6, 5	7, 4	7, 4	8, 3	9, 2	10, 2	11, 1	11, 1	12, 0	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**
4, 5	4, 4	5, 4	6, 3	7, 3	8, 2	8, 2	9, 1	10, 0	11, 0	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**
3, 4	4, 4	5, 3	5, 3	6, 2	7, 1	8, 1	9, 0	9, 0	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**
2, 3	3, 3	4, 2	5, 2	6, 1	6, 1	7, 0	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**
2, 3	3, 2	3, 2	4, 1	5, 0	6, 0	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**
1, 2	2, 1	3, 1	4, 0	4, 0	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**
1, 1	1, 0	2, 0	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**
0, 0	1, 0	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**
0, 0	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**

Figure 17. Final destination (m,n). Desired.

0,19	1,19	2,19	3,19	4,19	5,19	6,19	7,19	8,19	9,19	10,19	11,19	12,19	13,19	14,19	15,19	16,19	17,19	18,19	19,19
0,18	1,18	2,18	3,18	4,18	5,18	6,18	7,18	8,18	9,18	10,18	11,18	12,18	13,18	14,18	15,18	16,18	17,18	18,18	19,18
0,17	1,17	2,17	3,17	4,17	5,17	6,17	7,17	8,17	9,17	10,17	11,17	12,17	13,17	14,17	15,17	16,17	17,17	18,17	19,17
0,16	1,16	2,16	3,16	4,16	5,16	6,16	7,16	8,16	9,16	10,16	11,16	12,16	13,16	14,16	15,16	16,16	17,16	18,16	19,16
0,15	1,15	2,15	3,15	4,15	5,15	6,15	7,15	8,15	9,15	10,15	11,15	12,15	13,15	14,15	15,15	16,15	17,15	18,15	19,15
0,14	1,14	2,14	3,14	4,14	5,14	6,14	7,14	8,14	9,14	10,14	11,14	12,14	13,14	14,14	15,14	16,14	17,14	18,14	19,14
0,13	1,13	2,13	3,13	4,13	5,13	6,13	7,13	8,13	9,13	10,13	11,13	12,13	13,13	14,13	15,13	16,13	17,13	18,13	19,13
0,12	1,12	2,12	3,12	4,12	5,12	6,12	7,12	8,12	9,12	10,12	11,12	12,12	13,12	14,12	15,12	16,12	17,12	18,12	19,12
0,11	1,11	2,11	3,11	4,11	5,11	6,11	7,11	8,11	9,11	10,11	11,11	12,11	13,11	14,11	15,11	16,11	17,11	18,11	19,11
0,10	1,10	2,10	3,10	4,10	5,10	6,10	7,10	8,10	9,10	10,10	11,10	12,10	13,10	14,10	15,10	16,10	17,10	18,10	19,10
0, 9	1, 9	2, 9	3, 9	4, 9	5, 9	6, 9	7, 9	8, 9	9, 9	10, 9	11, 9	12, 9	13, 9	14, 9	15, 9	16, 9	17, 9	18, 9	19, 9
0, 8	1, 8	2, 8	3, 8	4, 8	5, 8	6, 8	7, 8	8, 8	9, 8	10, 8	11, 8	12, 8	13, 8	14, 8	15, 8	16, 8	17, 8	18, 8	19, 8
0, 7	1, 7	2, 7	3, 7	4, 7	5, 7	6, 7	7, 7	8, 7	9, 7	10, 7	11, 7	12, 7	13, 7	14, 7	15, 7	16, 7	17, 7	18, 7	19, 7
0, 6	1, 6	2, 6	3, 6	4, 6	5, 6	6, 6	7, 6	8, 6	9, 6	10, 6	11, 6	12, 6	13, 6	14, 6	15, 6	16, 6	17, 6	18, 6	19, 6
0, 5	1, 5	2, 5	3, 5	4, 5	5, 5	6, 5	7, 5	8, 5	9, 5	10, 5	11, 5	12, 5	13, 5	14, 5	15, 5	16, 5	17, 5	18, 5	19, 5
0, 4	1, 4	2, 4	3, 4	4, 4	5, 4	6, 4	7, 4	8, 4	9, 4	10, 4	11, 4	12, 4	13, 4	14, 4	15, 4	16, 4	17, 4	18, 4	19, 4
0, 3	1, 3	2, 3	3, 3	4, 3	5, 3	6, 3	7, 3	8, 3	9, 3	10, 3	11, 3	12, 3	13, 3	14, 3	15, 3	16, 3	17, 3	18, 3	19, 3
0, 2	1, 2	2, 2	3, 2	4, 2	5, 2	6, 2	7, 2	8, 2	9, 2	10, 2	11, 2	12, 2	13, 2	14, 2	15, 2	16, 2	17, 2	18, 2	19, 2
0, 1	1, 1	2, 1	3, 1	4, 1	5, 1	6, 1	7, 1	8, 1	9, 1	10, 1	11, 1	12, 1	13, 1	14, 1	15, 1	16, 1	17, 1	18, 1	19, 1
0, 0	1, 0	2, 0	3, 0	4, 0	5, 0	6, 0	7, 0	8, 0	9, 0	10, 0	11, 0	12, 0	13, 0	14, 0	15, 0	16, 0	17, 0	18, 0	19, 0

Figure 18. Original position. Coordinates.

0,19	1,19	2,19	3,18	4,18	5,18	6,17	7,17	8,17	9,16	10,16	11,16	12,15	13,15	14,15	15,14	16,14	17,14	18,13	19,13
0,18	1,18	2,18	3,17	4,17	5,17	6,16	7,16	8,16	9,15	10,15	11,15	12,14	13,14	14,14	15,13	16,13	17,13	18,12	19,12
0,17	1,17	2,17	3,16	4,16	5,16	6,15	7,15	8,15	9,14	10,14	11,14	12,13	13,13	14,13	15,12	16,12	17,12	18,11	19,11
0,16	1,16	2,16	3,15	4,15	5,15	6,14	7,14	8,14	9,13	10,13	11,13	12,12	13,12	14,12	15,11	16,11	17,11	18,10	19,10
0,15	1,15	2,15	3,14	4,14	5,14	6,13	7,13	8,13	9,12	10,12	11,12	12,11	13,11	14,11	15,10	16,10	17,10	18, 9	19, 9
0,14	1,14	2,14	3,13	4,13	5,13	6,12	7,12	8,12	9,11	10,11	11,11	12,10	13,10	14,10	15, 9	16, 9	17, 9	18, 8	19, 8
0,13	1,13	2,13	3,12	4,12	5,12	6,11	7,11	8,11	9,10	10,10	11,10	12, 9	13, 9	14, 9	15, 8	16, 8	17, 8	18, 7	19, 7
0,12	1,12	2,12	3,11	4,11	5,11	6,10	7,10	8,10	9, 9	10, 9	11, 9	12, 8	13, 8	14, 8	15, 7	16, 7	17, 7	18, 6	19, 6
0,11	1,11	2,11	3,10	4,10	5,10	6, 9	7, 9	8, 9	9, 8	10, 8	11, 8	12, 7	13, 7	14, 7	15, 6	16, 6	17, 6	18, 5	19, 5
0,10	1,10	2,10	3, 9	4, 9	5, 9	6, 8	7, 8	8, 8	9, 7	10, 7	11, 7	12, 6	13, 6	14, 6	15, 5	16, 5	17, 5	18, 4	19, 4
0, 9	1, 9	2, 9	3, 8	4, 8	5, 8	6, 7	7, 7	8, 7	9, 6	10, 6	11, 6	12, 5	13, 5	14, 5	15, 4	16, 4	17, 4	18, 3	19, 3
0, 8	1, 8	2, 8	3, 7	4, 7	5, 7	6, 6	7, 6	8, 6	9, 5	10, 5	11, 5	12, 4	13, 4	14, 4	15, 3	16, 3	17, 3	18, 2	19, 2
0, 7	1, 7	2, 7	3, 6	4, 6	5, 6	6, 5	7, 5	8, 5	9, 4	10, 4	11, 4	12, 3	13, 3	14, 3	15, 2	16, 2	17, 2	18, 1	19, 1
0, 6	1, 6	2, 6	3, 5	4, 5	5, 5	6, 4	7, 4	8, 4	9, 3	10, 3	11, 3	12, 2	13, 2	14, 2	15, 1	16, 1	17, 1	18, 0	19, 0
0, 5	1, 5	2, 5	3, 4	4, 4	5, 4	6, 3	7, 3	8, 3	9, 2	10, 2	11, 2	12, 1	13, 1	14, 1	15, 0	16, 0	17, 0	**,**	**,**
0, 4	1, 4	2, 4	3, 3	4, 3	5, 3	6, 2	7, 2	8, 2	9, 1	10, 1	11, 1	12, 0	13, 0	14, 0	**,**	**,**	**,**	**,**	**,**
0, 3	1, 3	2, 3	3, 2	4, 2	5, 2	6, 1	7, 1	8, 1	9, 0	10, 0	11, 0	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**
0, 2	1, 2	2, 2	3, 1	4, 1	5, 1	6, 0	7, 0	8, 0	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**
0, 1	1, 1	2, 1	3, 0	4, 0	5, 0	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**
0, 0	1, 0	2, 0	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**

Figure 19. Position after Step 1.

11,16	12,15	13,15	14,15	15,14	16,14	17,14	18,13	19,13	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**
10,15	11,15	12,14	13,14	14,14	15,13	16,13	17,13	18,12	19,12	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**
10,14	11,14	12,13	13,13	14,13	15,12	16,12	17,12	18,11	19,11	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**
9,13	10,13	11,13	12,12	13,12	14,12	15,11	16,11	17,11	18,10	19,10	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**
8,13	9,12	10,12	11,12	12,11	13,11	14,11	15,10	16,10	17,10	18, 9	19, 9	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**
8,12	9,11	10,11	11,11	12,10	13,10	14,10	15, 9	16, 9	17, 9	18, 8	19, 8	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**
7,11	8,11	9,10	10,10	11,10	12, 9	13, 9	14, 9	15, 8	16, 8	17, 8	18, 7	19, 7	**,**	**,**	**,**	**,**	**,**	**,**	**,**
7,10	8,10	9, 9	10, 9	11, 9	12, 8	13, 8	14, 8	15, 7	16, 7	17, 7	18, 6	19, 6	**,**	**,**	**,**	**,**	**,**	**,**	**,**
6, 9	7, 9	8, 9	9, 8	10, 8	11, 8	12, 7	13, 7	14, 7	15, 6	16, 6	17, 6	18, 5	19, 5	**,**	**,**	**,**	**,**	**,**	**,**
5, 9	6, 8	7, 8	8, 8	9, 7	10, 7	11, 7	12, 6	13, 6	14, 6	15, 5	16, 5	17, 5	18, 4	19, 4	**,**	**,**	**,**	**,**	**,**
5, 8	6, 7	7, 7	8, 7	9, 6	10, 6	11, 6	12, 5	13, 5	14, 5	15, 4	16, 4	17, 4	18, 3	19, 3	**,**	**,**	**,**	**,**	**,**
4, 7	5, 7	6, 6	7, 6	8, 6	9, 5	10, 5	11, 5	12, 4	13, 4	14, 4	15, 3	16, 3	17, 3	18, 2	19, 2	**,**	**,**	**,**	**,**
4, 6	5, 6	6, 5	7, 5	8, 5	9, 4	10, 4	11, 4	12, 3	13, 3	14, 3	15, 2	16, 2	17, 2	18, 1	19, 1	**,**	**,**	**,**	**,**
3, 5	4, 5	5, 5	6, 4	7, 4	8, 4	9, 3	10, 3	11, 3	12, 2	13, 2	14, 2	15, 1	16, 1	17, 1	18, 0	19, 0	**,**	**,**	**,**
2, 5	3, 4	4, 4	5, 4	6, 3	7, 3	8, 3	9, 2	10, 2	11, 2	12, 1	13, 1	14, 1	15, 0	16, 0	17, 0	**,**	**,**	**,**	**,**
2, 4	3, 3	4, 3	5, 3	6, 2	7, 2	8, 2	9, 1	10, 1	11, 1	12, 0	13, 0	14, 0	**,**	**,**	**,**	**,**	**,**	**,**	**,**
1, 3	2, 3	3, 2	4, 2	5, 2	6, 1	7, 1	8, 1	9, 0	10, 0	11, 0	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**
1, 2	2, 2	3, 1	4, 1	5, 1	6, 0	7, 0	8, 0	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**
0, 1	1, 1	2, 1	3, 0	4, 0	5, 0	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**
0, 0	1, 0	2, 0	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**

Figure 20. Position after Step 2.

11,16	11,15	12,14	13,14	14,13	15,12	16,12	16,11	17,11	18,10	18, 9	19, 9	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**
10,15	11,14	12,13	13,13	13,12	14,12	15,11	15,10	16,10	17,10	18, 8	19, 8	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**
10,14	10,13	11,13	12,12	12,11	13,11	14,11	15, 9	16, 9	17, 9	17, 8	18, 7	19, 7	**,**	**,**	**,**	**,**	**,**	**,**	**,**
9,13	9,12	10,12	11,12	12,10	13,10	14,10	14, 9	15, 8	16, 8	17, 7	18, 6	19, 6	19, 5	**,**	**,**	**,**	**,**	**,**	**,**
8,13	9,11	10,11	11,11	11,10	12, 9	13, 9	14, 8	15, 7	16, 7	16, 6	17, 6	18, 5	18, 4	19, 4	**,**	**,**	**,**	**,**	**,**
8,12	8,11	9,10	10,10	11, 9	12, 8	13, 8	13, 7	14, 7	15, 6	15, 5	16, 5	17, 5	18, 3	19, 3	**,**	**,**	**,**	**,**	**,**
7,11	8,10	9, 9	10, 9	10, 8	11, 8	12, 7	12, 6	13, 6	14, 6	15, 4	16, 4	17, 4	17, 3	18, 2	19, 2	**,**	**,**	**,**	**,**
7,10	7, 9	8, 9	9, 8	9, 7	10, 7	11, 7	12, 5	13, 5	14, 5	14, 4	15, 3	16, 3	17, 2	18, 1	19, 1	19, 0	**,**	**,**	**,**
6, 9	6, 8	7, 8	8, 8	9, 6	10, 6	11, 6	11, 5	12, 4	13, 4	14, 3	15, 2	16, 2	16, 1	17, 1	18, 0	**,**	**,**	**,**	**,**
5, 9	6, 7	7, 7	8, 7	8, 6	9, 5	10, 5	11, 4	12, 3	13, 3	13, 2	14, 2	15, 1	15, 0	16, 0	17, 0	**,**	**,**	**,**	**,**
5, 8	5, 7	6, 6	7, 6	8, 5	9, 4	10, 4	10, 3	11, 3	12, 2	12, 1	13, 1	14, 1	**,**	**,**	**,**	**,**	**,**	**,**	**,**
4, 7	5, 6	6, 5	7, 5	7, 4	8, 4	9, 3	9, 2	10, 2	11, 2	12, 0	13, 0	14, 0	**,**	**,**	**,**	**,**	**,**	**,**	**,**
4, 6	4, 5	5, 5	6, 4	6, 3	7, 3	8, 3	9, 1	10, 1	11, 1	11, 0	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**
3, 5	3, 4	4, 4	5, 4	6, 2	7, 2	8, 2	8, 1	9, 0	10, 0	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**
2, 5	3, 3	4, 3	5, 3	5, 2	6, 1	7, 1	8, 0	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**
2, 4	2, 3	3, 2	4, 2	5, 1	6, 0	7, 0	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**
1, 3	2, 2	3, 1	4, 1	4, 0	5, 0	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**
1, 2	1, 1	2, 1	3, 0	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**
0, 1	1, 0	2, 0	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**
0, 0	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**	**,**

Figure 21. Final position after Step 3. Actual.

at the exact position A or one of its neighboring grid points, B,C, or D, as shown in Figure 22.

Take the lower left corner portion of the 20 x 20 image problem as an example. From Figure 17, the desired destination is

	⋮			
(2,3)	(3,2)	(3,2)		
(1,2)	(2,1)	(3,1)		
(1,1)	(1,0)	(2,0)	...	(41)
(0,0)	(1,0)	*		
(0,0)	*	*		,

and the actual final position after the slide procedure is, from Figure 21,

		⋮			
(2,4)	(2,3)	(3,2)	(4,2)		
(1,3)	(2,2)	(3,1)	(4,1)		
(1,2)	(1,1)	(2,1)	(3,0)	...	(42)
(0,1)	(1,0)	(2,0)	*		
(0,0)	*	*	*		.

Associated with this plane, there is a slid image,

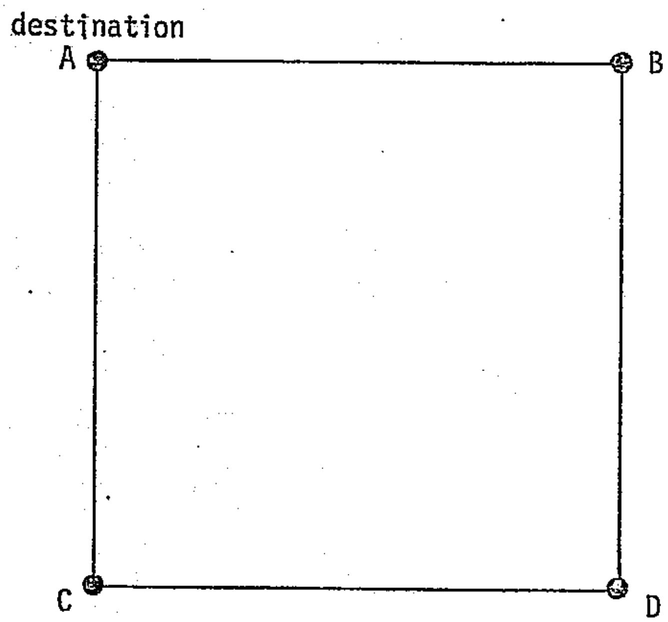


Figure 22. Four positions to search for data.

$$\left|\begin{array}{cccc} & & \vdots & \\ S_{2,4} & S_{2,3} & S_{3,2} & S_{4,2} \\ S_{1,3} & S_{2,2} & S_{3,1} & S_{4,1} \\ S_{1,2} & S_{1,1} & S_{2,1} & S_{3,0} \cdots \\ S_{0,1} & S_{1,0} & S_{2,0} & * \\ S_{0,0} & * & * & * \\ \hline \end{array}\right. , \qquad (43)$$

where $S_{i,j}$ denotes the data which is originally at position (i,j). Now,

$$(41) - (42) = \left|\begin{array}{ccc} & \vdots & \\ (0,-1) & (1,-1) & (0,0) \\ (0,-1) & (0,-1) & (0,0) \\ (0,-1) & (0,-1) & (0,-1) \cdots \\ (0,-1) & (0,0) & * \\ (0,0) & * & * \\ \hline \end{array}\right. .$$

The (0,0) elements in the resulting plane indicates that these points are at the exact destination. Hence, a mask M_1 is generated for these points.

$$M_1 = \left|\begin{array}{ccc} & \vdots & \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \cdots \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ \hline \end{array}\right. .$$

$$(43) \cdot \text{AND} \cdot M_1 = \left| \begin{array}{cccc} & \vdots & & \\ 0 & 0 & S_{3,2} & \\ 0 & 0 & S_{3,1} & \\ 0 & 0 & 0 & \cdots \\ 0 & S_{1,0} & 0 & \\ S_{0,0} & 0 & 0 & \end{array} \right. \qquad (44)$$

Next,

$$(41) - \{\text{Slide left } (42)\} = \left| \begin{array}{cccc} & \vdots & & \\ (0,0) & (0,0) & (-1,0) & \\ (-1,0) & (-1,0) & (-1,0) & \\ (0,0) & (-1,-1) & (-1,0) & \cdots \\ (-1,0) & (-1,0) & * & \\ * & * & * & \end{array} \right. ,$$

generating a mask

$$M_2 = \left| \begin{array}{cccc} & \vdots & & \\ 1 & 1 & 0 & \\ 0 & 0 & 0 & \\ 1 & 0 & 0 & \cdots \\ 0 & 0 & 0 & \\ 0 & 0 & 0 & \end{array} \right. .$$

$$\{\text{Slide left (43)}\} \cdot \text{AND} \cdot M_2 = \left| \begin{array}{ccc} & \vdots & \\ S_{2,3} & S_{3,2} & 0 \\ 0 & 0 & 0 \\ S_{1,1} & 0 & 0 \;\ldots \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{array} \right. . \qquad (45)$$

Next

$$(41) - \{\text{Slide up (42)}\} = \left| \begin{array}{ccc} & \vdots & \\ (1,0) & (1,0) & (0,1) \\ (0,0) & (1,0) & (1,0) \\ (1,0) & (0,0) & (0,0) \;\ldots \\ (0,0) & * & * \\ * & * & * \end{array} \right. ,$$

generating a mask

$$M_3 = \left| \begin{array}{ccc} & \vdots & \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \;\ldots \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{array} \right. .$$

$$\{\text{Slide up (43)}\} \cdot \text{AND} \cdot M_3 = \left|\begin{array}{llll} & \vdots & & \\ 0 & 0 & 0 & \\ S_{1,2} & 0 & 0 & \\ 0 & S_{1,0} & S_{3,0} & \cdots \\ S_{0,0} & 0 & 0 & \\ 0 & 0 & 0 & \end{array}\right. \qquad (46)$$

Finally,

$$(41) - \{\text{Slide up and left (42)}\} = \left|\begin{array}{cccc} & \vdots & & \\ (0,1) & (1,0) & (-1,1) & \\ (0,1) & (0,0) & (0,1) & \cdots \\ (0,1) & (-1,0) & * & \\ * & * & * & \\ * & * & * & \end{array}\right.$$

$$M_{\text{[illegible]}} = \left|\begin{array}{cccc} & \vdots & & \\ 0 & 0 & 0 & \\ 0 & 1 & 0 & \\ 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & \\ 0 & 0 & 0 & \end{array}\right.$$

and

$$\{\text{Slide up and left (43)}\} \cdot \text{AND} \cdot M_4 = \left| \begin{array}{cccc} & \vdots & & \\ 0 & 0 & 0 & \\ 0 & S_{2,1} & 0 & \\ 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & \\ 0 & 0 & 0 & \end{array} \right. \qquad (47)$$

$$(44) \cdot \text{OR} \cdot (45) \cdot \text{OR} \cdot (46) \cdot \text{OR} \cdot (47) = \left| \begin{array}{cccc} & \vdots & & \\ S_{2,3} & S_{3,2} & S_{3,2} & \\ S_{1,2} & S_{2,1} & S_{3,1} & \\ S_{1,1} & S_{1,0} & S_{3,0} & \cdots \\ S_{0,0} & S_{1,0} & * & \\ S_{0,0} & * & * & \end{array} \right.$$

This is the desired $S_{m,n}$ plane. Similarly, $S_{m+1,n}$, $S_{m,n+1}$, and $S_{m+1,n+1}$ can be generated by extracting the data through masks in the same way.

Observe that the above example is a special case in which the data can always be found from one of its four neighboring grid points, after the slide procedure, as shown in Figure 22. In general, however, as proven in Appendix B and presented in the examples of the next chapter, data can be at any one of its eight neighboring grid points as depicted in Figure 23. In other words, nine masks, instead of four, are required to extract the data and generate the desired S planes,

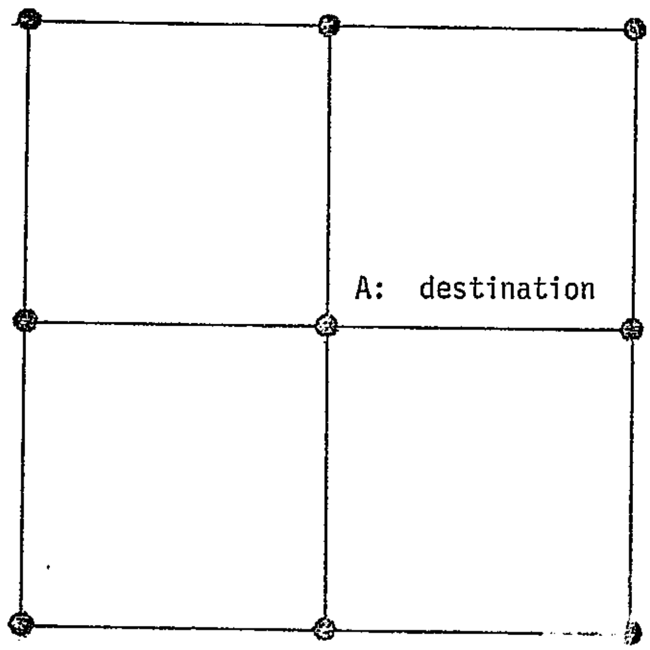


Figure 23. Nine postions to search for data.

CHAPTER 8

AUTOMATIC CONTROL FOR THE SLIDE PROCEDURE

A control unit is required for the tse computer to undertake the task of sequencing the stored instructions in the proper order, selecting the correct information source and destination, and providing the appropriate processing path through the tse processor. One possible control organization is shown in Figure 24 [8], in which processor control is achieved by the selection of a "control word" which is output from the control unit and interfaced with the tse processor. Each bit of the control word is used to activate or deactivate one or more of the elements in the tse processor. Proper data paths are thus provided by activating the elements which lie in the specific processing paths chosen by the instruction.

Control implementation may be achieved by utilizing small computers. Basically, any of the microprocessors available could be used. However, the control unit must observe the timing constraints dictated by the tse logic device propagation delay. A control unit organized around a microprocessor must be sufficiently fast for the tse processor.

Most of the operations involved in the image rotation algorithm are simple slides, shifts, additions, subtractions, and comparisons, etc.. The function of the tse processor control unit for these basic operations has been studied [2,4] and is not duplicated in this research. However, since a large number of slide operations is required in the three-step slide procedure, an efficient control algorithm for the slide process is

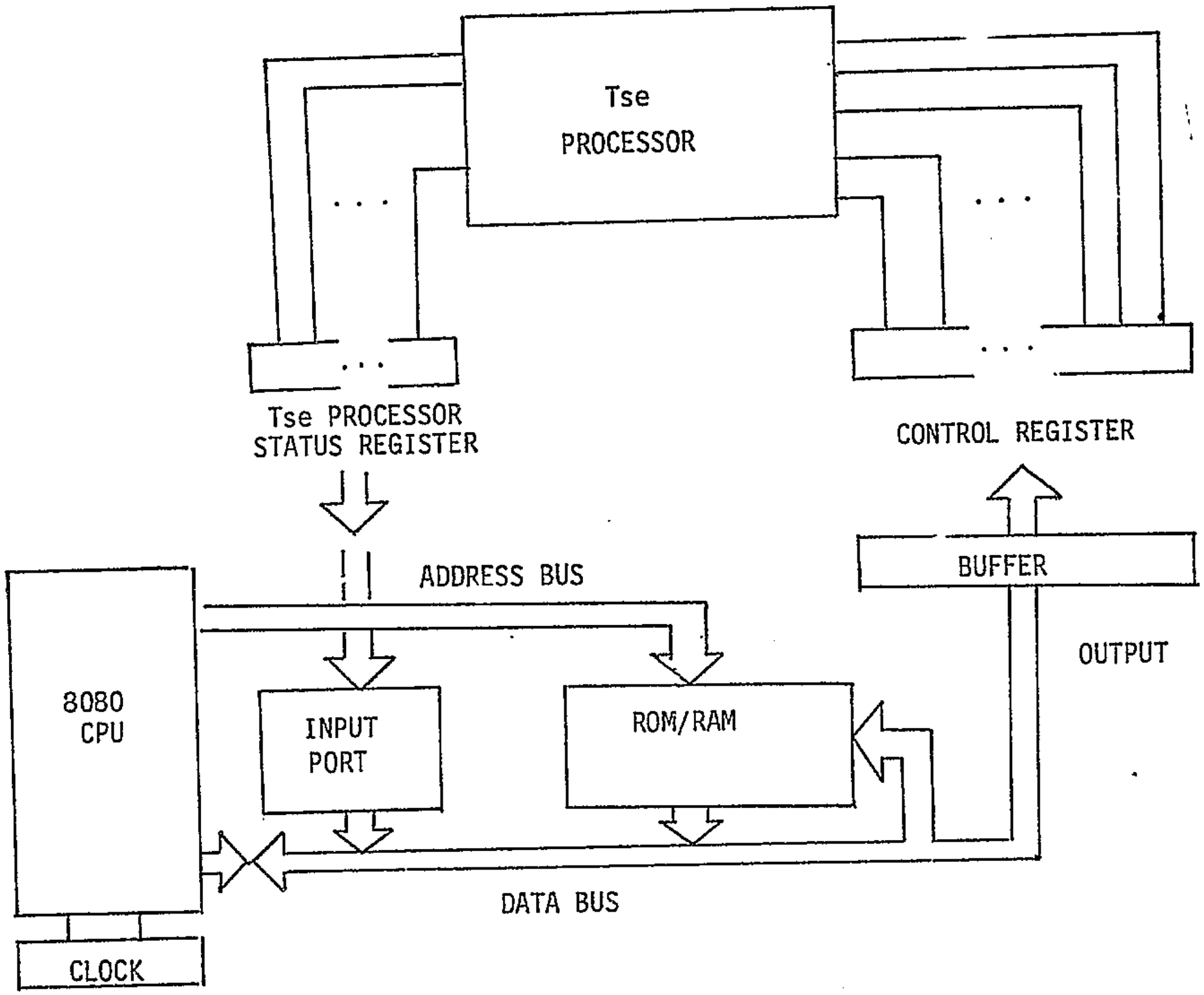


Figure 24. Tse processor control.

highly necessary.

The task of the control unit in the three-step slide procedure is to accurately determine the desired number of slides for each column, or each row, at each step. Notice that, in the actual implementation of the algorithm, V and H planes are not generated and slid for the following two reasons:

(1) As presented in Chapter 5, the exact relationships between V, H and ℓ_v, ℓ_h are involved.

(2) Even if the V and H planes were generated and slid along with the image for the purpose of providing useful guides, the task is very time-consuming to determine the proper amount of slides for each column (row) at each step by the man-machine mode, as was done in Chapter 6. The development of a hardware sensor or a software algorithm to check the V and H planes and to determine the required number of slides would be very impractical as far as cost and operation time are concerned.

Therefore, there is no obvious guide from which one can decide upon the required number of slides during the slide process. For this reason, an algorithm which can provide the control unit with the necessary information on the required slides for each row, or each column, at each step of the sliding process, is desirable. The efficiency of such a control is crucial to the success of the image rotation.

The following text is devoted to the development and the explanation of a method serving for the automatic control of the slide procedure. This method is simple, fast, and easy to implement. The number of required slides for each column, or each row, can be precalculated and coded as control words before the execution of the sliding procedure. An example is employed to illustrate the details of the proposed method.

Step 1

As presented in Appendix B, the amount of vertical slides generated in this step, for column m, is

$$V' = \left\langle \frac{\cos\theta - 1}{\sin\theta} (m-p) \right\rangle \tag{48}$$

where θ is the angle of rotation and p is the x-coordinate of the rotation center. As an example, Figure 25 shows a 32 x 32 image plane which contains a binary image "T", whose edges are outlined as shown in the figure. The image is to be rotated 37.24° about the tse element (15,15), as shown by the circled element. The Assigning Technique is used in this example.

Equation (48) becomes

$$V' = \left\langle \frac{\cos 37.24° - 1}{\sin 37.24°} (m-15) \right\rangle = \left\langle -0.337(m-15) \right\rangle$$

The amount of vertical slides for each column can be calculated by the control unit, by substituting m = 0 to 31. The results are shown in Table 1.

Figure 26 showns the image after performing the vertical slides. As expected, Step 1 has the effect of twisting the image vertically.

Step 2

The horizontal slides in Step 2 are given by Equations (33) and (34), or any set of equations (35) and (36), equations (37) and (38), or

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 25. Original image.

TABLE 1

Amount of Vertical Slides in Step 1

m	v'
0	5
1	5
2	4
3	4
4	4
5	3
6	3
7	3
8	2
9	2
10	2
11	1
12	1
13	1
14	0
15	0

m	v'
16	0
17	-1
18	-1
19	-1
20	-2
21	-2
22	-2
23	-3
24	-3
25	-3
26	-4
27	-4
28	-4
29	-5
30	-5
31	-5

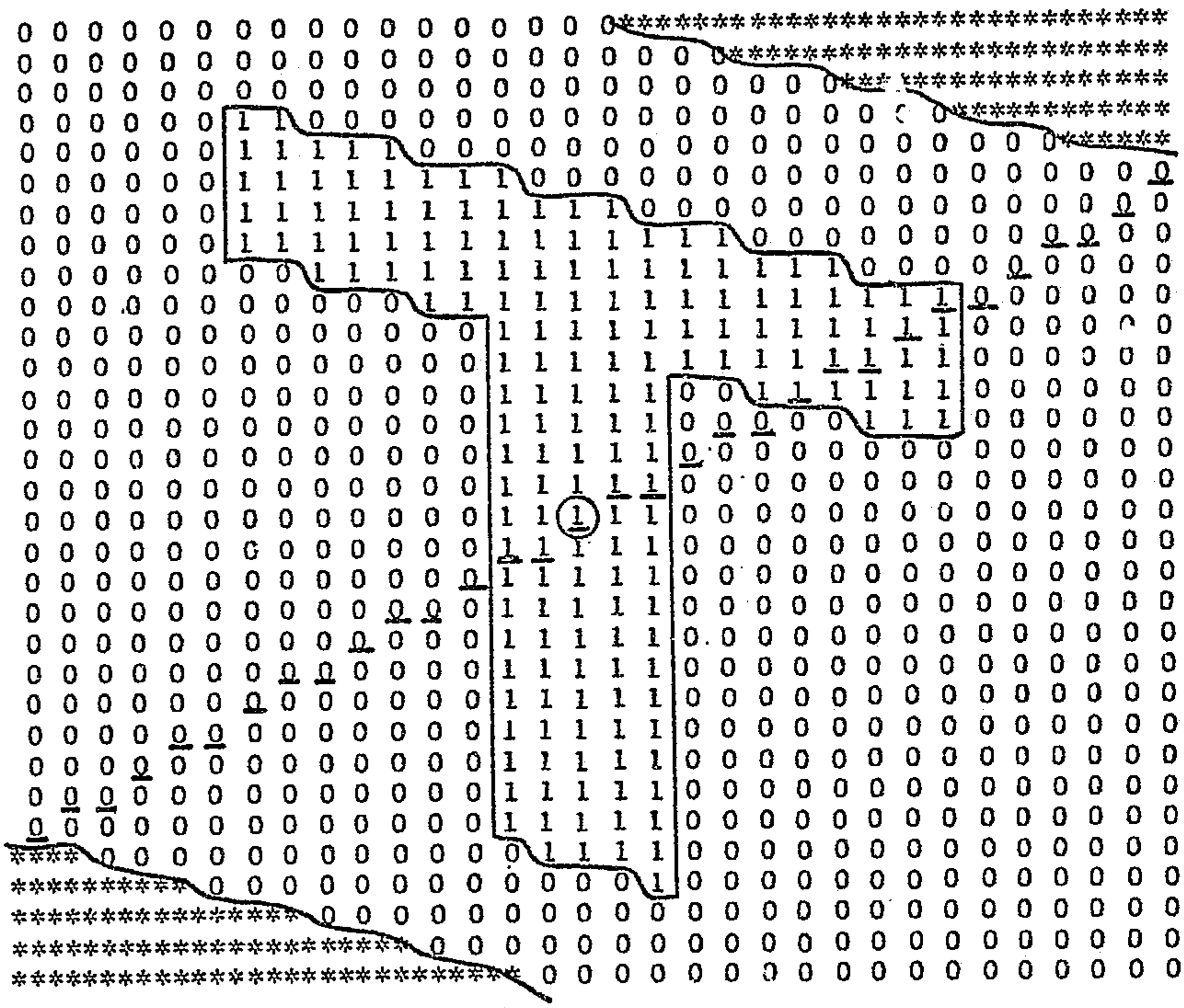


Figure 26. Slid image After Step 1.

equations (39) and (40), depending upon which technique is used and in which direction the image is rotated. In Step 1 , the H-pattern is aligned with respect to the elements at the same column as the rotation center. Since this column is the only column which is not slid during Step 1, the simplest way to determine the required number of horizontal slides for each row is to use these elements as references. Substituting the coordinates of these elements (p,n) into equation (33), and then into equations (35), (37), and (39), the following results are obtained:

(1) For the Assigning Technique,

$$H = \langle (n-q)\ \sin\theta \rangle \quad . \tag{49}$$

(2) For the Linear Least-Square-Error Technique (clockwise rotation,

$$H = \langle (n-q)\ \sin\theta \rangle + 1 \tag{50}$$

(3) For the Linear Least-Square-Error Technique (counter-clockwise rotation),

$$H = \langle (n-q)\ \sin\theta \rangle \tag{51}$$

For the example, equation (49) becomes

$$H = \langle (n-15)\cdot \sin 37.24 \rangle$$

$$= \langle 0.605 \cdot (n-15) \rangle \quad .$$

The amount of required horizontal slides for each row calculated by this equation are summarized in Table 2.

TABLE 2

Amount of Horizontal Slides in Step 2

n	H
0	-9
1	-8
2	-8
3	-7
4	-7
5	-6
6	-5
7	-5
8	-4
9	-4
10	-3
11	-2
12	-2
13	-1
14	-1
15	0

n	H
16	1
17	1
18	2
19	2
20	3
21	4
22	4
23	5
24	5
25	6
26	7
27	7
28	8
29	8
30	9
31	10

The result of the image slid by the algorithm is shown in Figure 27. As can be seen, Step 2 has the effect of twisting the image horizontally.

Step 3

Since almost all image elements are displaced from their original positions after Steps 1 and 2, there is no obvious row or column similar to the p-column used in Step 2 which can be employed as the reference for the decision of the required vertical slides for each column in Step 3. The intended reference elements, after being slid in Steps 1 and 2, must span the whole range from the leftmost column to the rightmost column such that the required number of vertical slides for each column can be determined from the reference elements at that column. After thorough study, the elements on the diagonal of the image plane are found to be the only set of elements which has the above property in all cases. These points are indicated as underlined elements in Figure 25. Their new positions after Steps 1 and 2 are shown in Figures 26 and 27, respectively. Notice that, although some of these elements are slid out of the image plane during Steps 1 and 2, the remaining elements cover the whole plane horizontally. Also, observe the fact that reference elements are absent in some columns, which can be taken care of by filling appropriate data into these positions. Since the number of absent elements is small, no difficulty is imposed by this condition.

The task of determining the required number of slides in Step 3 includes:

(a) Determining the new column positions of the reference elements after

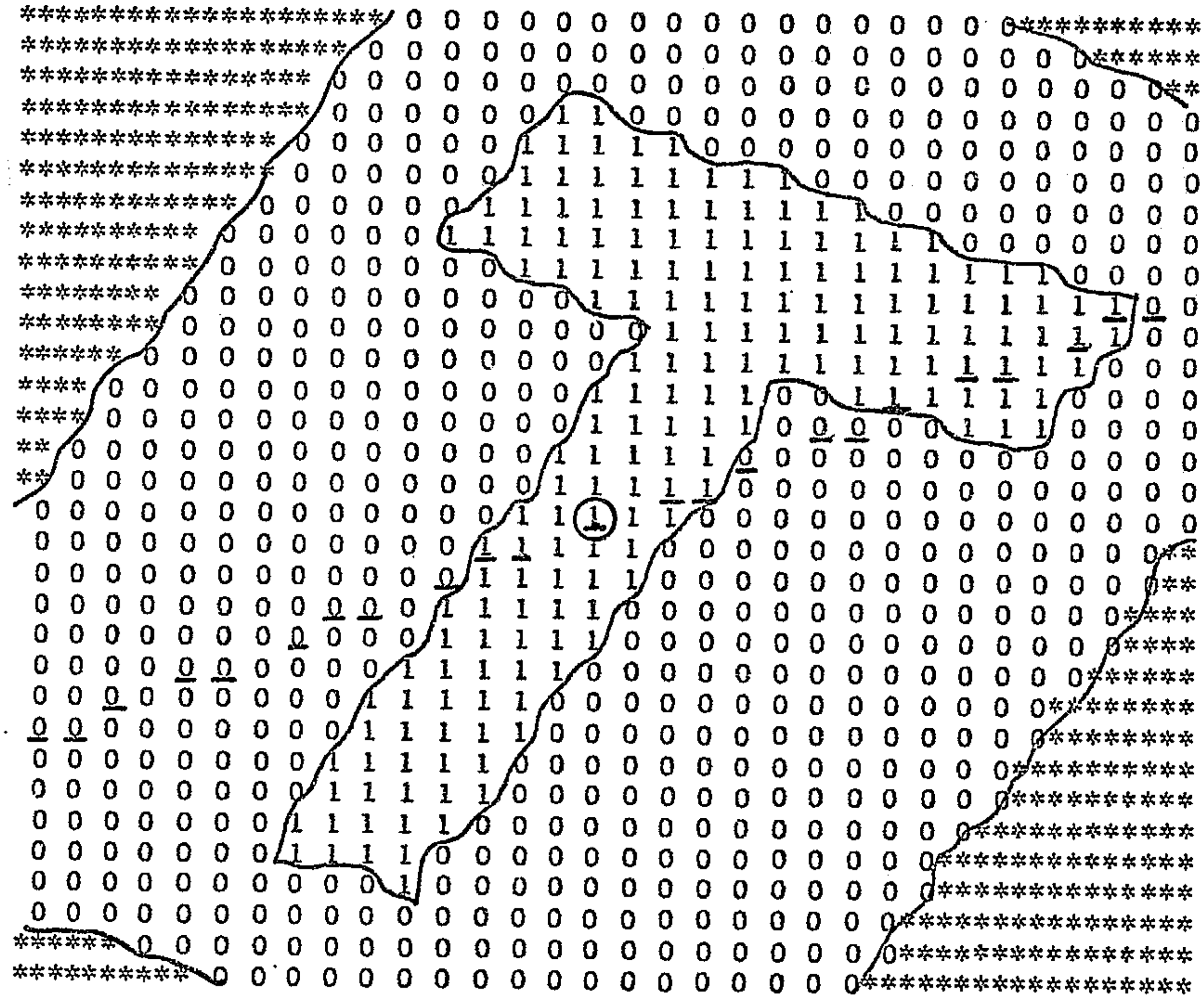


Figure 27. Slide image after Step 2.

Step 2.

The column position of the diagonal element (n,n), after a horizontal slide H in Step 2, is

$$i = n + H \tag{52}$$

where H is given by equations (36), (38), or (40), depending upon which interpolative technique is used and in which direction the image is rotated.

For the example, equation (52) becomes

$$i = n + \langle -(1-\cos 37.24°)(n-15) + (n-15)\sin 37.24° \rangle = n + \langle 0.4(n-15) \rangle$$

The calculated values of new column positions i are shown in Table 3(a). Those elements which are slid out of the image plane ($i < 0$ or $i > 31$) are no longer considered and are deleted in the table. Observe that, for n = 14 and 16, the calculated values of i are 14 and 16, respectively, which are different from the actual column positions of these two elements, 13 and 17, as can be seen from Figure 27. However, these differences are always no more than 1, and are within the tolerance of the algorithm for data extraction.

(b) Determining the required number of vertical slides in Step 3 for reference elements.

The total number of required vertical slides, V, for the diagonal element (n,n) is given by equations (35), (37), or (39). Since the element has already been slid by an amount V' in Step 1, the required vertical slides in Step 3 is

$$V'' = V - V' \quad , \tag{53}$$

where V' is given by equation (48).

For the example, equation (53) becomes

$$V'' = \langle -(1 - \cos 37.24°)(n-15) - (n-15)\sin 37.24° \rangle - \langle \frac{\cos 37.24° - 1}{\sin 37.24°}(n-15) \rangle$$

$$= \langle -0.809\ (n-15) \rangle - \langle -0.337\ (n-15) \rangle \quad , \quad n = 4 \sim 26$$

The values of V" calculated by this equation are shown in Table 3(b). Notice that for some i, the values of V" are not given, since reference elements are absent at these columns.

(c) Smoothing V"

For clockwise rotation, V" should be a decreasing function of column i. However, as indicated in Table 3(b), contradiction occurs for i = 5 as a result of rounding-off. Data smoothing is thus required to make the corrections. One of the simplest ways to restore V" to a decreasing function is the operation

$$V''(i+1) = V''(i) \quad , \quad \text{if } V''(i+1) > V''(i) \quad . \tag{54}$$

The results of the smoothing operation are shown in Table 3(c).

(d) Filling the voids in V"

Since the voids in V" are few and are distributed evenly among the specified ones, each void can be filled by simply assigning the void the same value as the column to its left; that is,

TABLE 3

n	i
0	-6
1	-5
2	-3
3	-2
4	0
5	1
6	2
7	4
8	5
9	7
10	8
11	9
12	11
13	12
14	14(13)
15	15
16	16(17)
17	18
18	19
19	21
20	22
21	23
22	25
23	26
24	28
25	29
26	30
27	32
28	33
29	35
30	36
31	37

(a)

i	V"	
0	5	
1	5	
2	4	
3		
4	3	
5	4	←
6		
7	3	
8	2	
9	2	
10		
11	1	
12	1	
13		
14	1	
15	0	
16	-1	
17		
18	-1	
19	-1	
20		
21	-2	
22	-2	
23	-3	
24		
25	-4	
26	-4	
27		
28	-4	
29	-5	
30	-5	
31		

(b)

i	V"
0	5
1	5
2	4
3	
4	3
5	3
6	
7	3
8	2
9	2
10	
11	1
12	1
13	
14	1
15	0
16	-1
17	
18	-1
19	-1
20	
21	-2
22	-2
23	-3
24	
25	-4
26	-4
27	
28	-4
29	-5
30	-5
31	

(c)

i	V"
0	5
1	5
2	4
3	4
4	3
5	3
6	3
7	3
8	2
9	2
10	2
11	1
12	1
13	1
14	1
15	0
16	-1
17	-1
18	-1
19	-1
20	-1
21	-2
22	-2
23	-3
24	-3
25	-4
26	-4
27	-4
28	-4
29	-5
30	-5
31	-5

(d)

(a) The New Column Positions of The Reference Elements After Step 2

(b) Required Number of Vertical Slides For Reference Elements

(c) Results of the Smoothing of V"

(d) Results of the Filling of V"

$$V''(i+1) = V''(i) \quad , \quad \text{if } V''(i+1) \text{ is not given} \quad .$$

The final results of V" after the filling process are shown in Table 3(d). After sliding the image by the amounts determined in Step 3, the results are shown in Figure 28. Step 3 has the effect of twisting the image further vertically.

Notice that the above discussion for Step 3 is only for a clockwise rotation. For counter-clockwise rotation, the following two operations are changed:

(1) Diagonal elements (0,N-1), (1,N-2), ..., (N-1,0), instead of (0,0), (1,1), ..., (N-1,N-1), are used as reference elements;

(2) The V" should be an increasing function, rather than a decreasing function of i.

Finally, the detailed algorithm for making the decision as to the required number of slides in each step are summarized in Table 4. Different formulae are included for both interpolative techniques, and both directions of rotation. The step by step procedures are presented in such a way that direct implementation of this algorithm is not difficult.

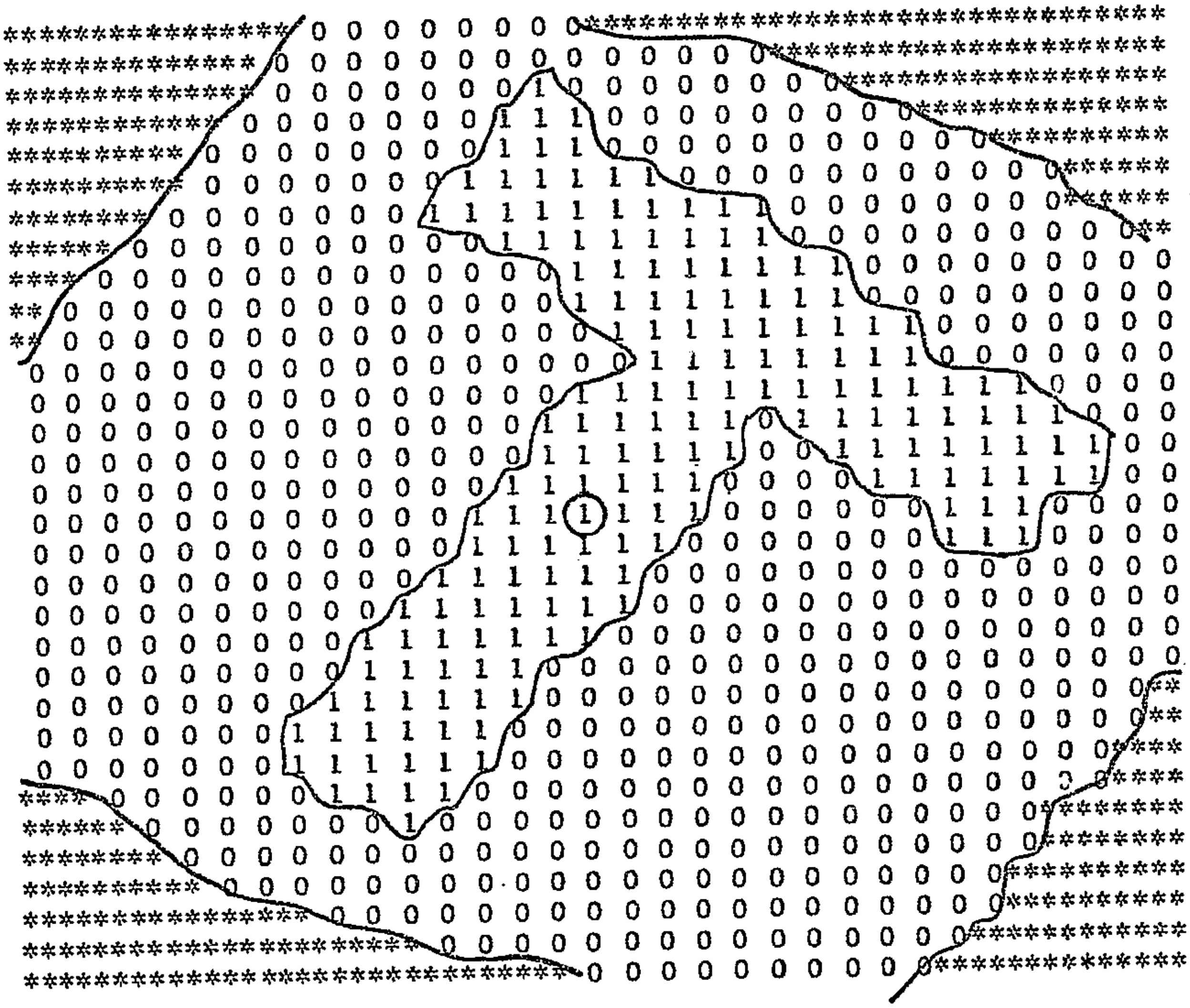


Figure 28. Slid image after Step 3.

TABLE 4

Steps to Perform the Three-Step Slide Algorithm

	Assigning Technique		LLSE Technique	
	clockwise rotation	counter-clockwise rotation	clockwise rotation	counter-clockwise rotation
Step 1 $V'(m) =$	$\left\langle \frac{\cos\theta - 1}{\sin\theta}(m-p) \right\rangle$			
Step 2 $H(n) =$	$\langle (n-q)\sin\theta \rangle$		$\langle (n-q)\sin\theta \rangle + 1$	$\langle (n-q)\sin\theta \rangle$
Step 3 (a) $i(n) =$	$n + \langle -(1-\cos\theta)(n-p) + (n-q)\sin\theta \rangle$	$n + \langle -(1-\cos\theta)(n-p) - (N-1-n-q)\sin\theta \rangle$	$n + \langle -(1-\cos\theta)(n-p) - (n-q)\sin\theta \rangle$	$n + \langle -(1-\cos\theta)(n-p) - (N-1-n-q)\sin\theta \rangle$
(b) $V''(i) =$	$\langle -(1-\cos\theta)(n-q) - (n-p)\sin\theta \rangle$	$\langle -(1-\cos\theta)(n-q) - (N-1-n-p)\sin\theta \rangle$	$\langle -(1-\cos\theta)(n-q) - (n-p)\sin\theta \rangle$	$\langle -(1-\cos\theta)(n-q) - (N-1-n-p)\sin\theta \rangle + 1$
(c) Smoothing	$V''(i+1) = V''(i)$ if $V''(i+1) > V''(i)$	$V''(i+1) = V''(i)$ if $V''(i+1) < V''(i)$	$V''(i+1) = V''(i)$ if $V''(i+1) > V''(i)$	$V''(i+1) = V''(i)$ if $V''(i+1) < V''(i)$
(d) Filling	$V''(i+1) = V''(i)$ if $V''(i+1)$ is not given			

CHAPTER 9

EXAMPLES

In this chapter, many examples are given to demonstrate the correctness and the effectiveness of the algorithm derived in the previous chapters. Various cases, including clockwise rotation and counter-clockwise rotation, small angle rotation and large angle rotation, rotation about a point in the middle of the image plane, rotation about a point close to the boundary of the image plane, using the Assigning Technique and using the Linear Least-Square-Error Technique, are presented in order to show that the algorithms are applicable to all cases of rotation. The 32 x 32 image plane shown in Figure 25 (page 55)is used for the examples. All examples are simulated using FORTRAN language on an IBM/360 System. The simulation program, which is developed for the general case, is included in Appendix C. One should make the point that because of the printing facility used, the images shown in the following examples do not appear to be exact squares as they should be. Except for this print-out distortion, the results of all examples appear to be what one would expect.

<u>Example 1</u>

rotation center $(p,q) = (8,4)$
rotation angle $\theta = 28.6°$
The Assigning Technique

As the first step, the desired destinations, m and n planes, are calculated by equations (6), (7), (11), and (12), and shown in Figures 29 and 30, respectively. The three-step slide algorithm is then used to slide the image as well as the original m and n planes, which are the coordinates of the tse elements as shown in Figures 31 and 32. The image after each slide step is shown in Figures 33, 34, and 35. As expected, Step 1 has the effect of twisting the image vertically and Step 2 has the effect of twisting the image horizontally. Step 3 completes the remaining vertical slides. The result in Figure 35 shows that, after the slide procedure, the image has been rotated to the desired orientation, but its edges are rather coarse.

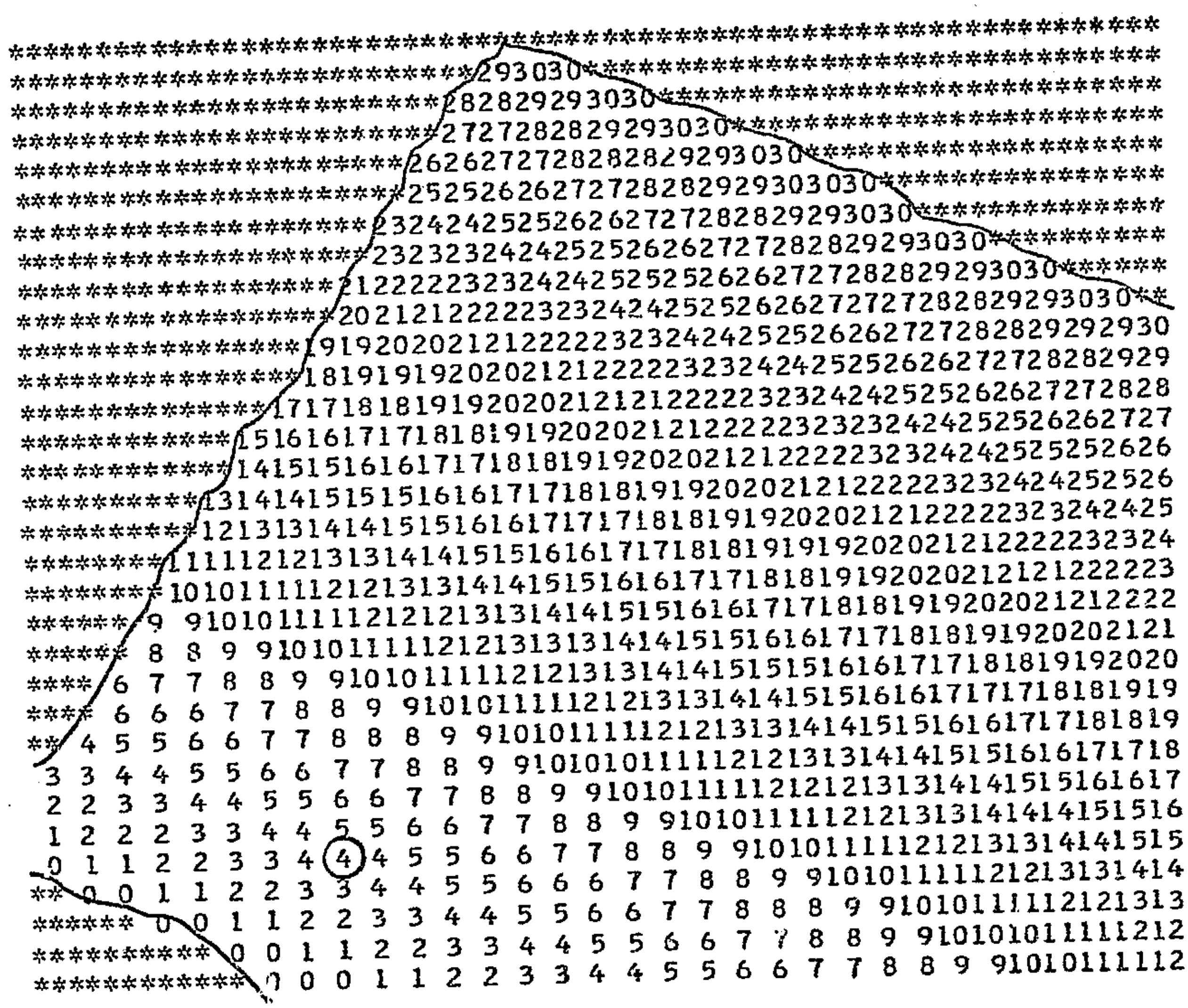
Notice that the original positions, m and n planes, which are shown in Figures 31 and 32, are slid along with the image plane. Figures 36 and 37 present the m and n planes after the slide procedure. Comparing the calculated m and n planes (Figures 29 and 30), which show the desired destinations, with the slid m and n planes (Figures 36 and 37), which show the actual positions after the slide procedure, data can be extracted by the algorithm described in Chapter 7 to form the desired rotated image f(i,j), as shown in Figure 38. Observe that this image has smoother edges than the one in Figure 35. The data extraction process has rearranged the displaced elements and positioned them where they should be.

Careful comparison of Figures 29, 30 and Figures 36, 37 reveals that the boundaries of the calculated m, n planes and those of the slid m,n planes are not exactly the same. The differences are due to the round-off approximations. Even though this "boundary effect" is small, one should be aware of its presence.

In the actual implementation, elements slid from outside the image plane are presented by O's rather than stars as shown in Figures 33

```
******************************************************************
****************************  0 1 2**********************************
*************************  0 0 1 2 3 4******************************
*************************  0 1 2 3 4 4 5 6*************************
***********************  0 0 1 2 3 4 5 6 7 8 8*********************
***********************  0 1 2 3 4 4 5 6 7 8 91011*****************
*********************  0 1 1 2 3 4 5 6 7 8 8 9101112***************
*********************  0 1 2 3 4 5 5 6 7 8 91011121213 14*********
*******************  0 1 2 2 3 4 5 6 7 8 9 9101112131415 16 16******
******************  0 1 2 3 4 5 6 6 7 8 910111213131415161718 19**
*****************  0 1 2 2 3 4 5 6 7 8 9101011121314151617171819 20
****************  0 1 2 3 4 5 6 6 7 8 910111213131415161718192021
***************  0 1 2 3 3 4 5 6 7 8 910101112131415161717 18192021
*************  0 0 1 2 3 4 5 6 7 7 8 91011121314141516171819202121
*************  0 1 2 3 4 4 5 6 7 8 91011111213141516171818192021 22
**********  0 0 1 2 3 4 5 6 7 8 8 91011121314151516171819202122 22
**********  0 1 2 3 4 4 5 6 7 8 9101112121314151617181919202122 23
********  0 1 1 2 3 4 5 6 7 8 8 910111213141515161718192021222323
********  0 1 2 3 4 5 5 6 7 8 910111212131415161718191920212223 24
******  0 1 2 2 3 4 5 6 7 8 9 91011121314151616171819202122232324
******  0 1 2 3 4 5 6 6 7 8 9101112131314151617181920202122232425
****  0 1 2 2 3 4 5 6 7 8 910101112131415161717181920212223242425
****  0 1 2 3 4 5 6 6 7 8 91011121314141516171819202121222324 2526
**  0 1 2 3 3 4 5 6 7 8 9101011121314151617171819202122232425 2526
  0 0 1 2 3 4 5 6 7 7 8 910111213141415161718192021212223242526 27
  0 1 2 3 4 4 5 6 7 8 9101111121314151617181819202122232425252627
  0 1 2 3 4 5 6 7 8 8 9101112131415151617181920212222232425262728
  1 2 3 4 4 5 6 7 8 9101112121314151617181919202122232425262627 28
** 2 3 4 5 6 7 8 8 910111213141516161718192021222323242526272829
****** 5 5 6 7 8 91011121213141516171819192021222324252627272829
********** 7 8 9 91011121314151616171819202122232324252627282930
************ 8 9101112131314151617181920202122232425262727282930
```

Figure 29. Final m (calculated).



Final 30, Final n (calculated),

```
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031
```

Figure 31. Original m.

```
3131313131313131313131313131313131313131313131313131313131313131
3030303030303030303030303030303030303030303030303030303030303030
2929292929292929292929292929292929292929292929292929292929292929
2828282828282828282828282828282828282828282828282828282828282828
2727272727272727272727272727272727272727272727272727272727272727
2626262626262626262626262626262626262626262626262626262626262626
2525252525252525252525252525252525252525252525252525252525252525
2424242424242424242424242424242424242424242424242424242424242424
2323232323232323232323232323232323232323232323232323232323232323
2222222222222222222222222222222222222222222222222222222222222222
2121212121212121212121212121212121212121212121212121212121212121
2020202020202020202020202020202020202020202020202020202020202020
1919191919191919191919191919191919191919191919191919191919191919
1818181818181818181818181818181818181818181818181818181818181818
1717171717171717171717171717171717171717171717171717171717171717
1616161616161616161616161616161616161616161616161616161616161616
1515151515151515151515151515151515151515151515151515151515151515
1414141414141414141414141414141414141414141414141414141414141414
1313131313131313131313131313131313131313131313131313131313131313
1212121212121212121212121212121212121212121212121212121212121212
1111111111111111111111111111111111111111111111111111111111111111
1010101010101010101010101010101010101010101010101010101010101010
 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
 4 4 4 4 4 4 4(4)4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Figure 32. Original n.

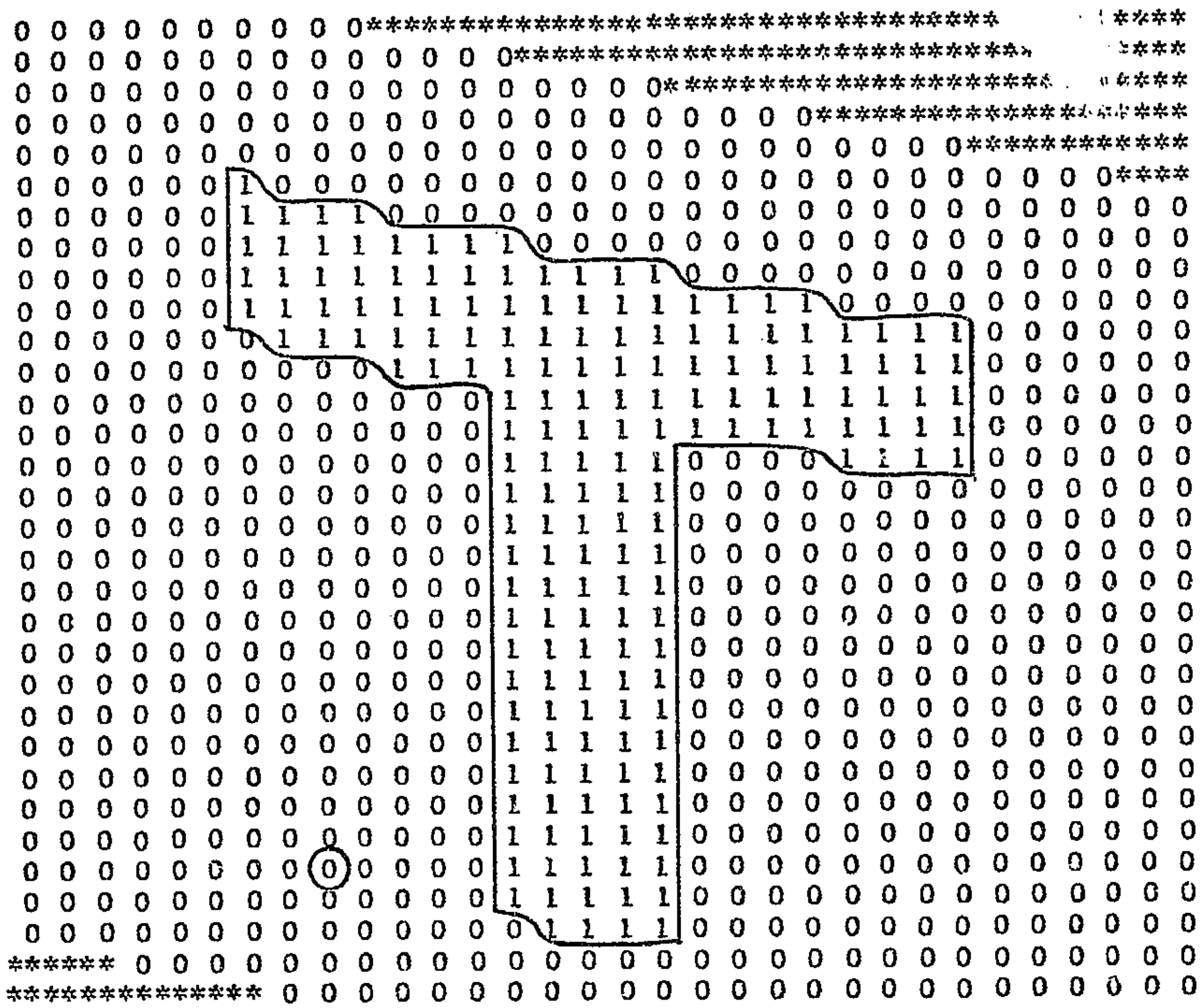


Figure 33. Slid image after Step 1.

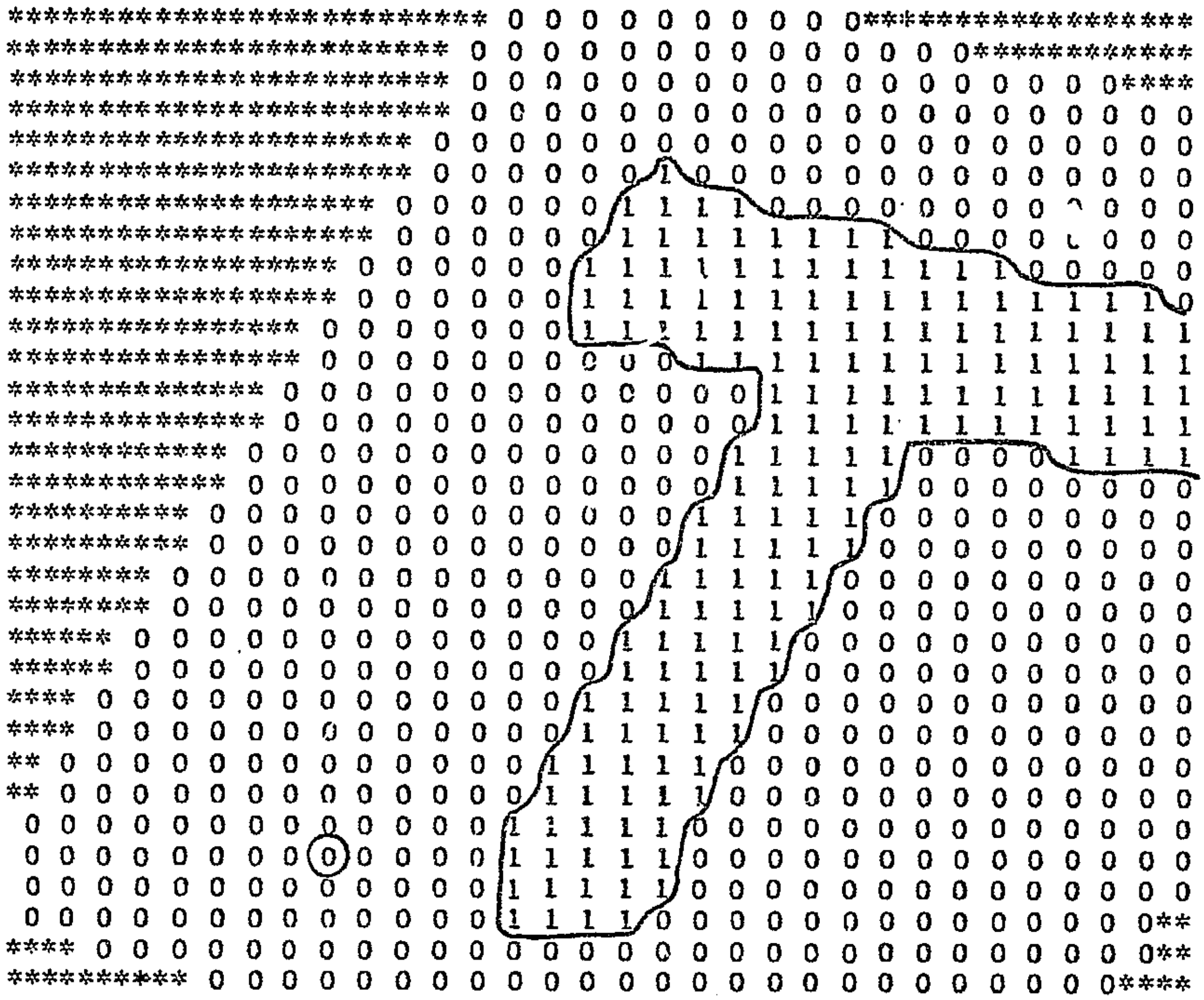


Figure 34. Slid image after Step 2.

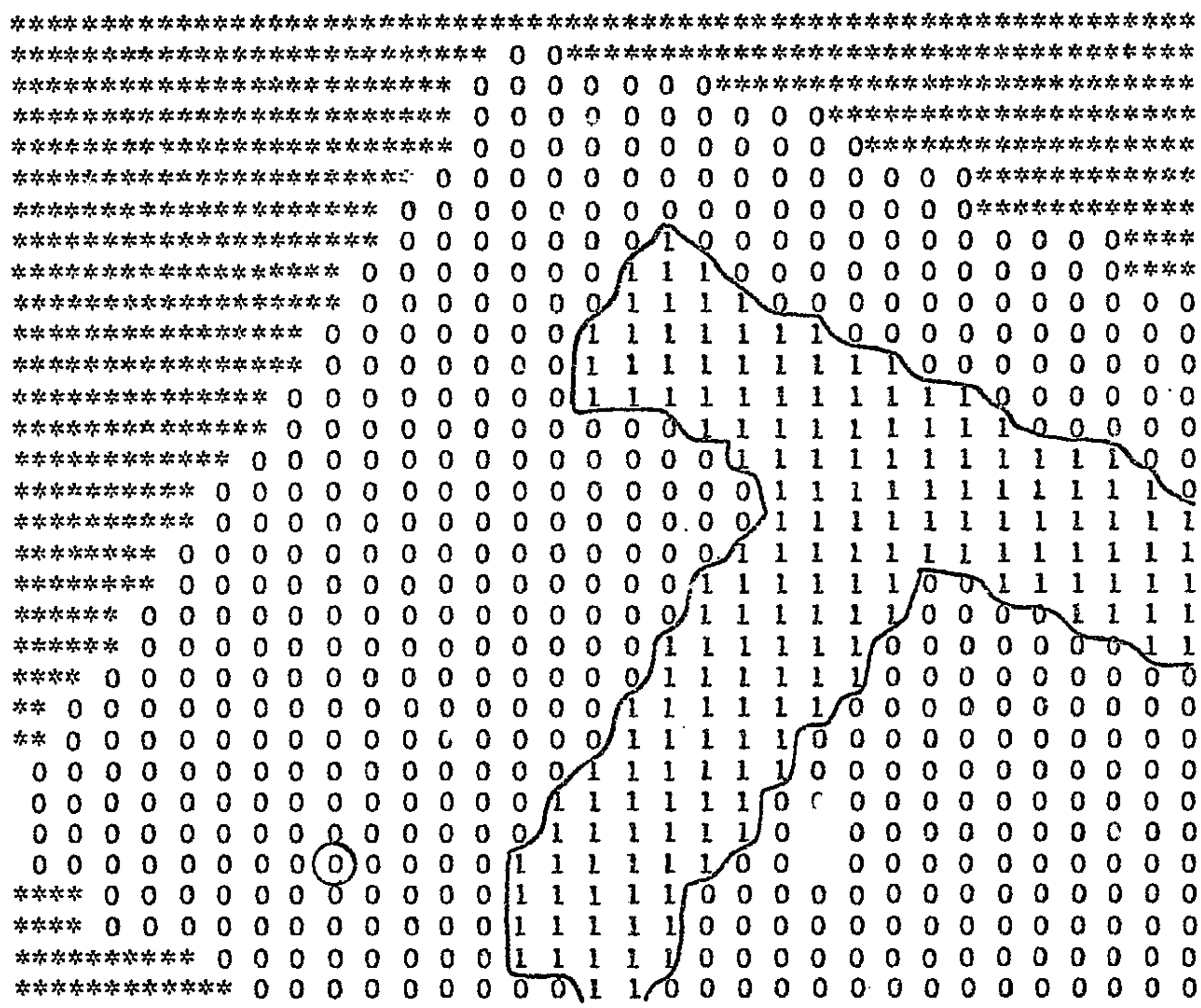


Figure 35. Slid image after Step 3.

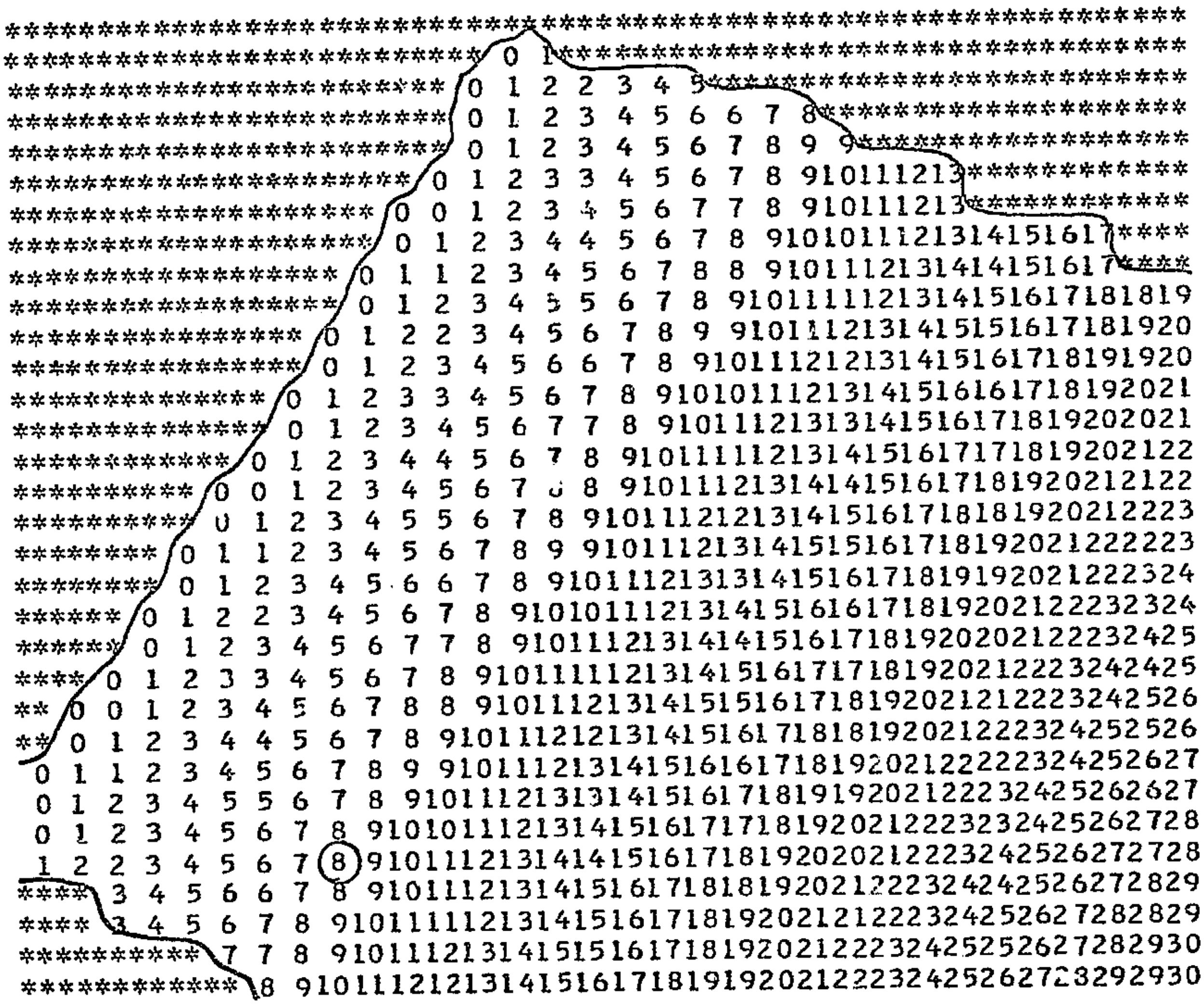


Figure 36. Final m (slid).

```
****************************************************************
**************************2929**********************************
************************28282829303030**************************
************************27272729292929303131********************
************************2626262828282830303031******************
**********************252525262727272729292931313131************
********************23242424252626262728282830303030************
********************2223232424252525262727282929292931313131****
******************212122222323242425252626272828282930303030****
******************2020212222222323242425262627272728292929303131
****************191919202121212223232324252526262727282828293030
****************181818202020202122222224242425252626272728282929
**************17171718191919192121212223232324252525262627272828
**************16161617181818192020202122222223242424252626262727
************1515151616171717181919202021212123232323242525252627
**********131414141515161617171818191920202122222222222424242526
**********121313141414151516161718181819192021212122232323232525
********11111212131313141515151617171718191920202021222222232424
********10101112121212131414141616161617181819192020212121222323
****** 9 9 91011111111131313141515151517171718181919202021212222
****** 8 8 81010101011121212131414141516161617181818191920202121
**** 7 7 7 8 9 9 9 910111112121313131415151516171717181919192020
** 5 6 6 6 7 8 8 8 9 9101011111212131314141416161616171818181920
** 4 5 5 6 6 7 7 7 8 8 91010101111111212131314151515151717171819
 3 3 4 4 5 5 6 6 7 7 7 8 9 9 91011111112121314141415161616161818
 2 2 3 4 4 4 5 5 6 6 6 8 8 8 8 910101011121213131314151515161717
 1 1 2 3 3 3 4 5 5 5 6 7 7 7 7 9 9 9 910111112121313141414151616
 0 0 1 2 2 2 3 4 4 4 5 6 6 6 7 8 8 8 810101011111212131314141515
**** 1 1 1 1 2 3 3 3 4 5 5 5 6 7 7 7 8 9 9 910111111121213131414
**** 0 0 0 0 2 2 2 3 3 4 4 4 5 6 6 6 7 8 8 8 9101010111212121313
********** 0 1 1 1 2 2 3 3 4 4 5 5 5 6 7 7 7 9 9 910111111111213
************ 0 0 1 1 1 2 2 3 3 4 4 5 5 6 6 6 8 8 8 8101010101112
```

Figure 37. Final n (slid).

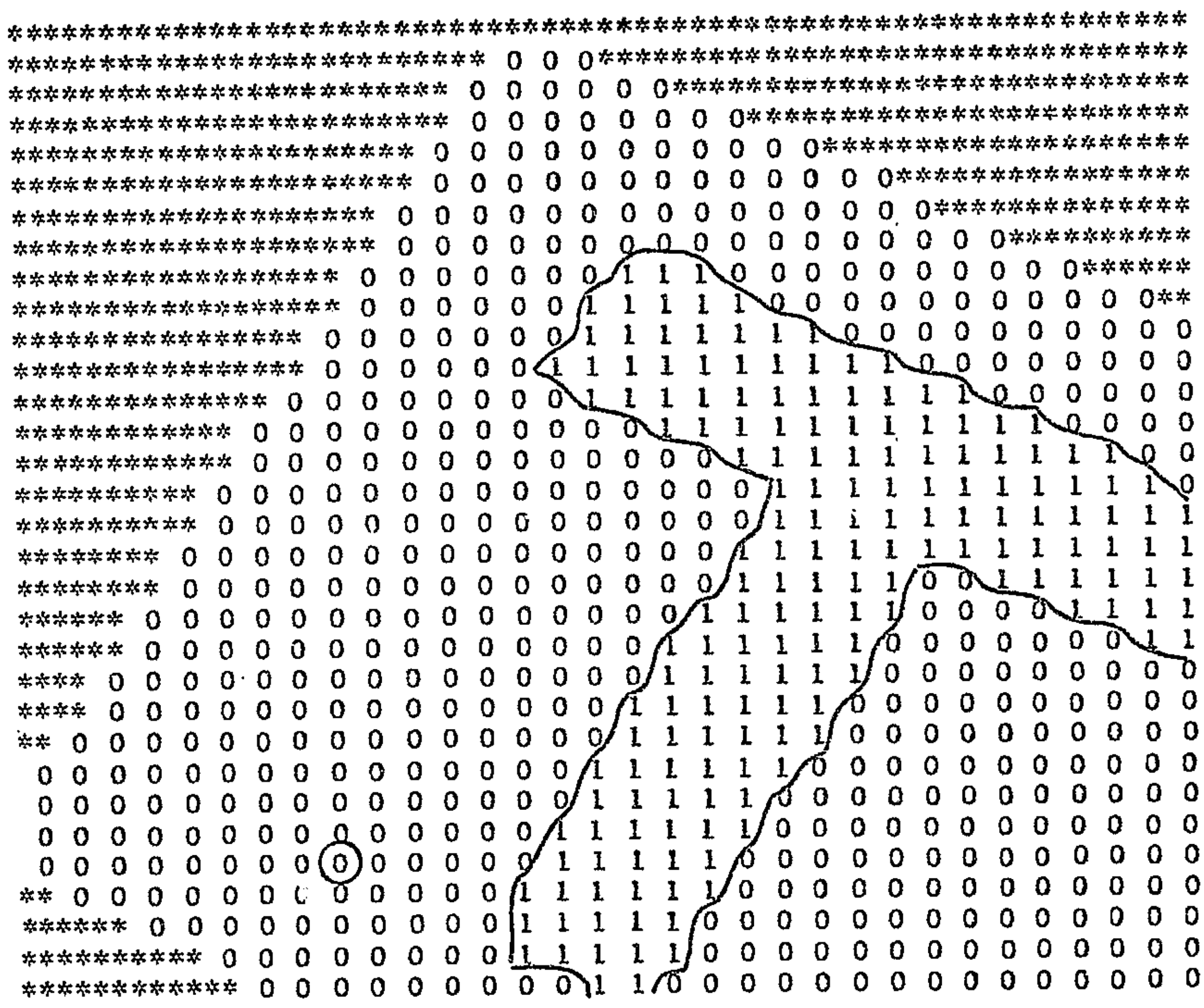


Figure 38. Rotated image f(i,j).

through 38. The actual boundary of the rotated image can be obtained by generating a mask from the calculated final m and n planes, as shown in Figures 29 and 30, where elements with coordinates less than 0 or greater than 31 are the area originally undefined.

Example 2

rotation center (p,q) = (8,4)
rotation angle $\theta = -28.6°$
The Assigning Technique

This example is the same as Example 1, except that the image is rotated in a counter-clockwise direction. The rotated image is presented in Figure 39. Observe that some elements on the boundary of the rotated image plane (for example, the squared element on the left boundary in Figure 39) do not obtain the data from its neighboring elements (Obviously, the value at the squared element should be 1 rather than 0). This is because of the fact that, for boundary elements, some of their eight neighboring elements have already been slid out of the image plane; therefore, the task becomes impossible for these boundary elements to extract the required data from lost elements. As a result, one should realize that the boundary of the rotated image has already deteriorated.

Example 3

rotation center (p,q) = (8,4)
rotation angle $\theta = 28.6°$
The Linear Least-Square-Error Technique

This example is the same as Example 1, except that the Linear Least-Square-Error Interpolative Technique is used to extract data,

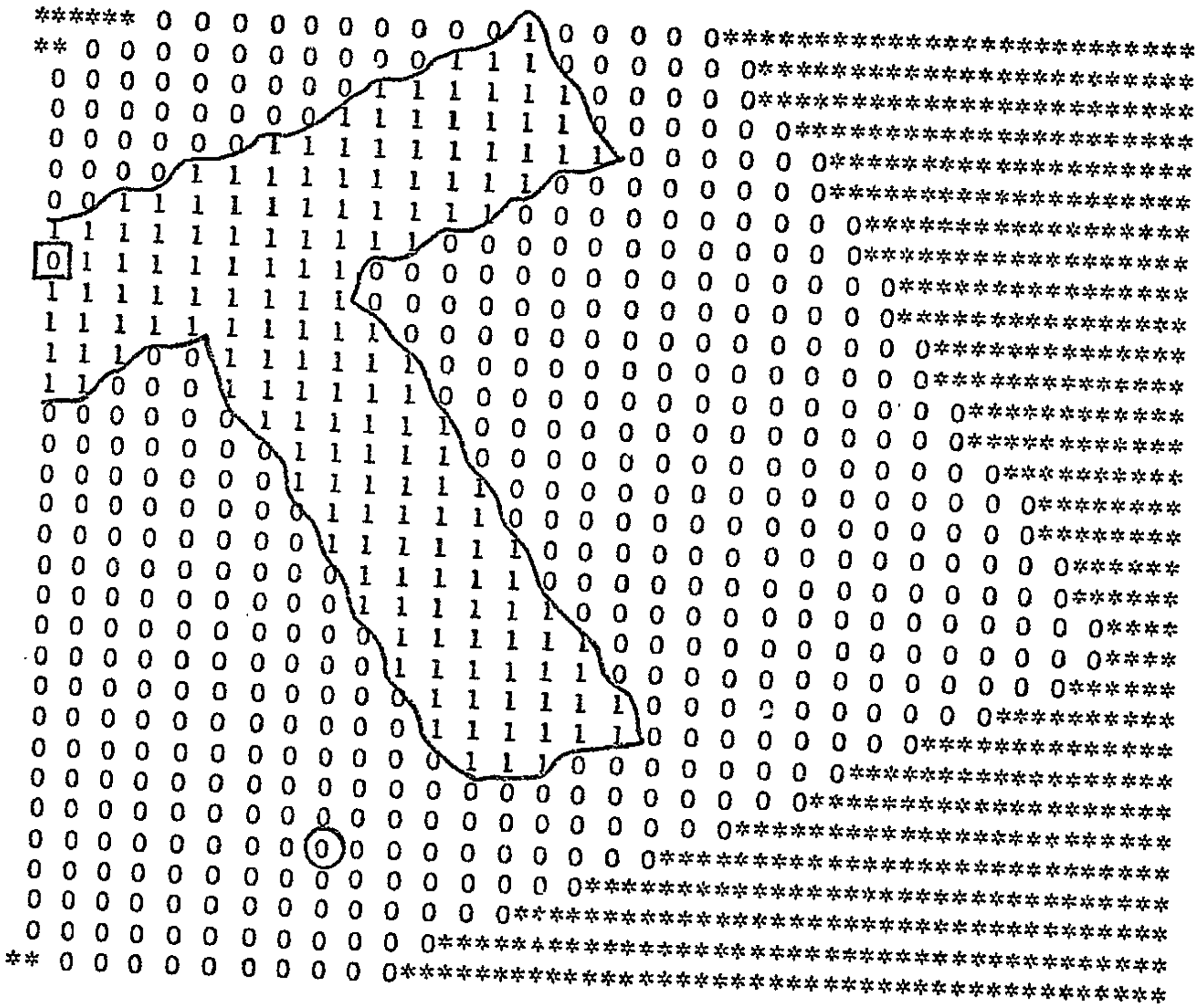


Figure 39. Example 2: (p,q) = (8,4), θ = -28.7° (Using Assigning Technique).

Figure 40 presents the $S_{m,n}$ plane. Image planes $S_{m+1,n}$, $S_{m,n+1}$, and $S_{m+1,n+1}$ can similarly be generated and the final rotated image can be obtained by interpolating these four planes. However, since the image in this example is only a binary image, the effort is not taken.

Example 4

rotation center (p,q) = (8,4)
rotation angle θ = -28.6°
The Linear Least-Square-Error Technique

Again the Least-Square-Error Interpolative Technique is used to replace the Assigning Technique used in Example 2. The resulting $S_{m,n}$ plane is presented in Figure 41.

Example 5

rotation center (p,q) = (15,15)
rotation angle θ = 45°
The Assigning Technique

The rotated image f(i,j) is presented in Figure 42.

Example 6

rotation center (p,q) = (15,15)
rotation angle θ = 10°
The Assigning Technique

Figure 43 presents the rotated image. Observe that since the rotation angle is small and the image is a binary image, the rotated image looks rather crude.

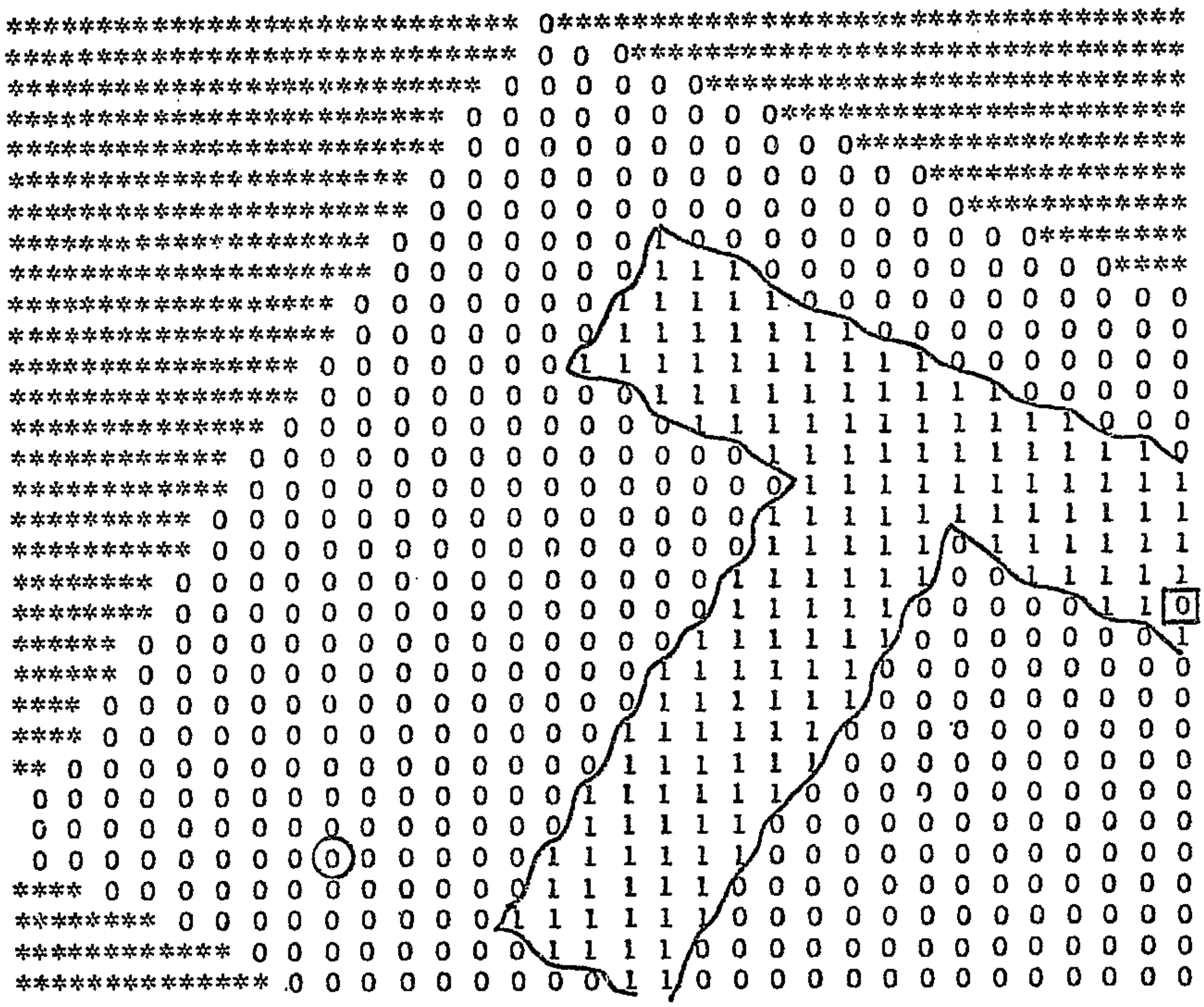


Figure 40. Example 3: (p,q) = (8,4), θ = 28.7° (Using Linear Least-Square-Error Interpolative Technique).

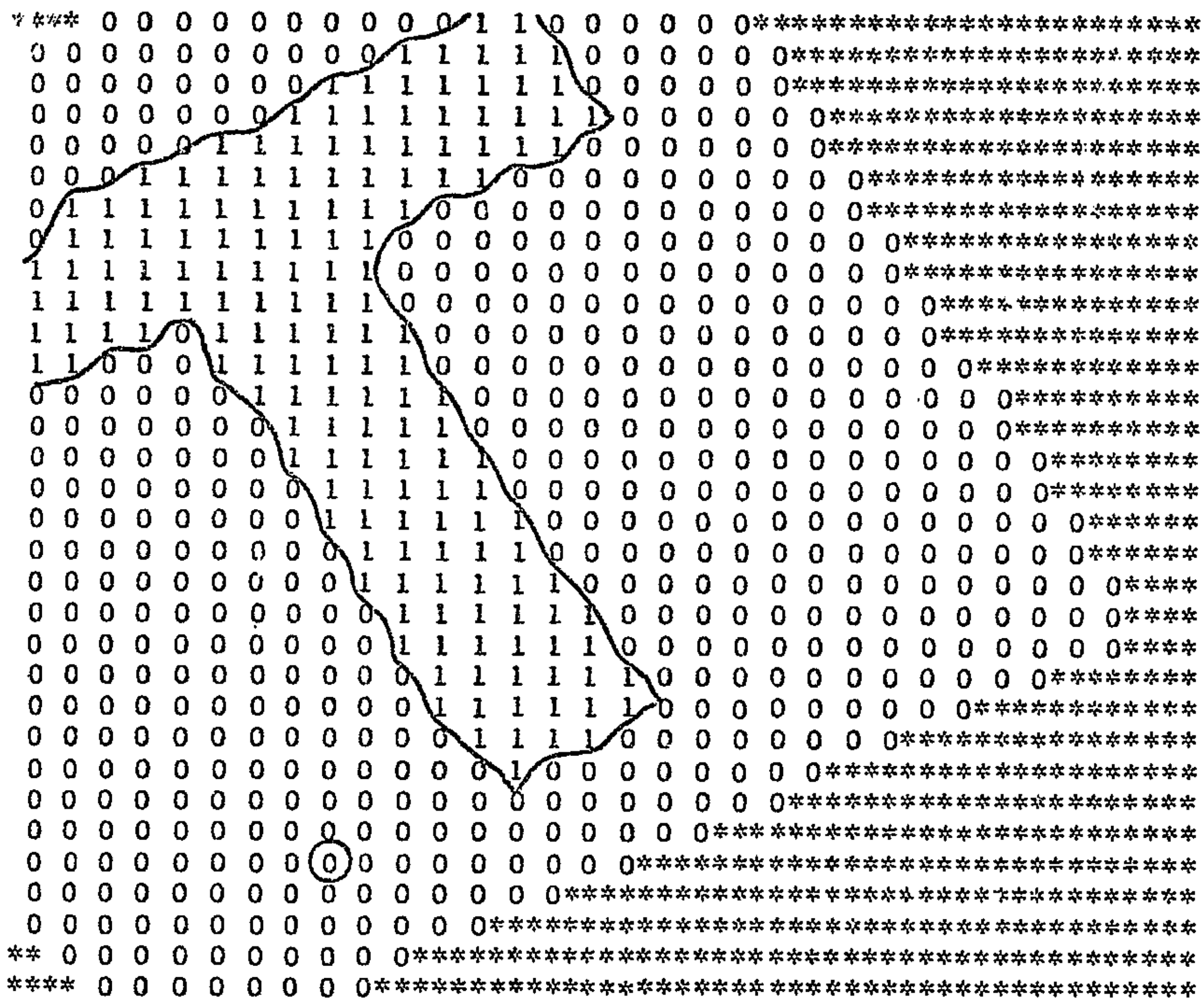


Figure 41. Example 4: (p,q) = (8,4), θ = -28.7° (Using Linear Least-Square-Error Interpolative Technique).

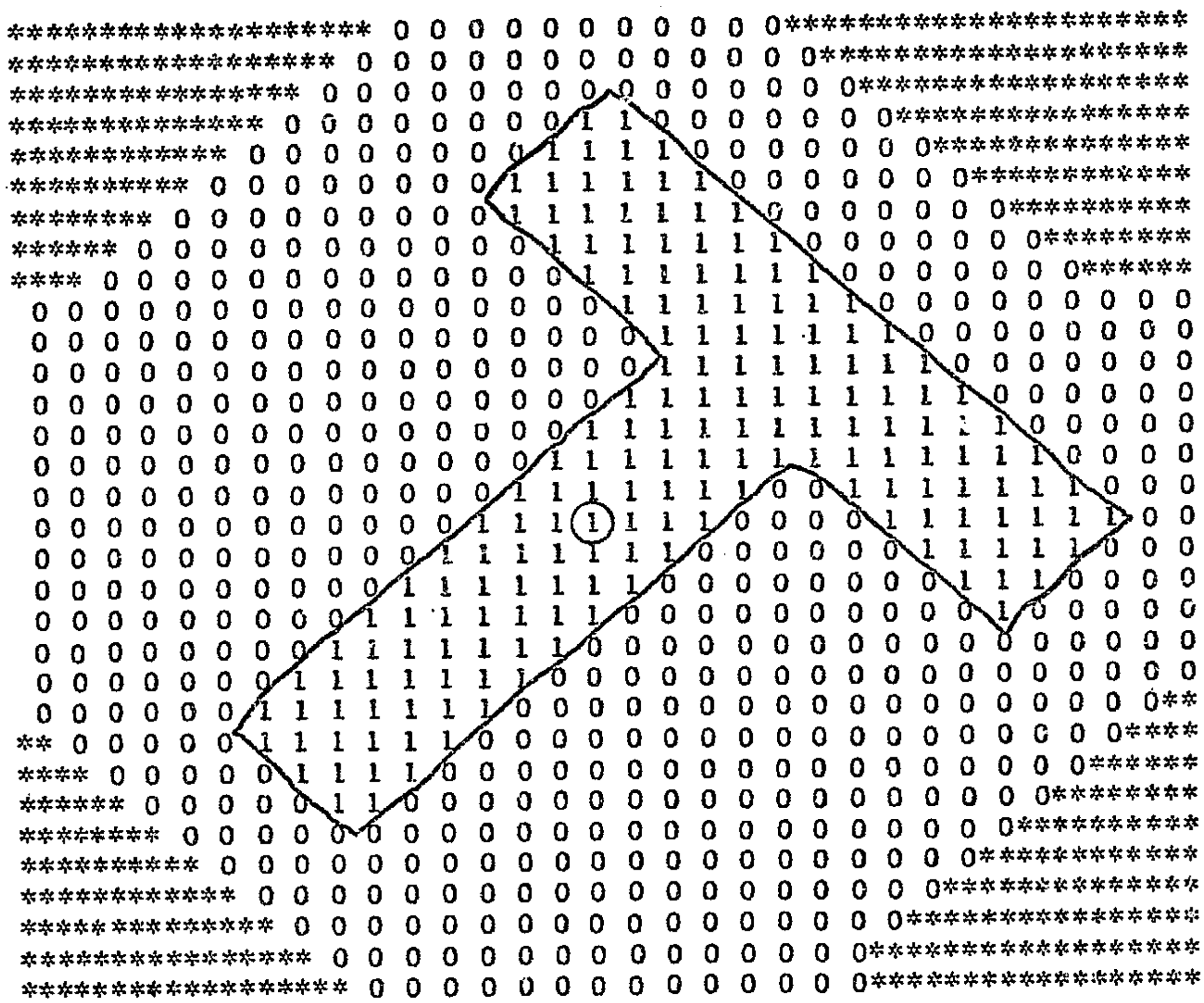


Figure 42. Example 5: $(p,q) = (15,15)$, $\theta = 45°$ (Using Assigning Technique).

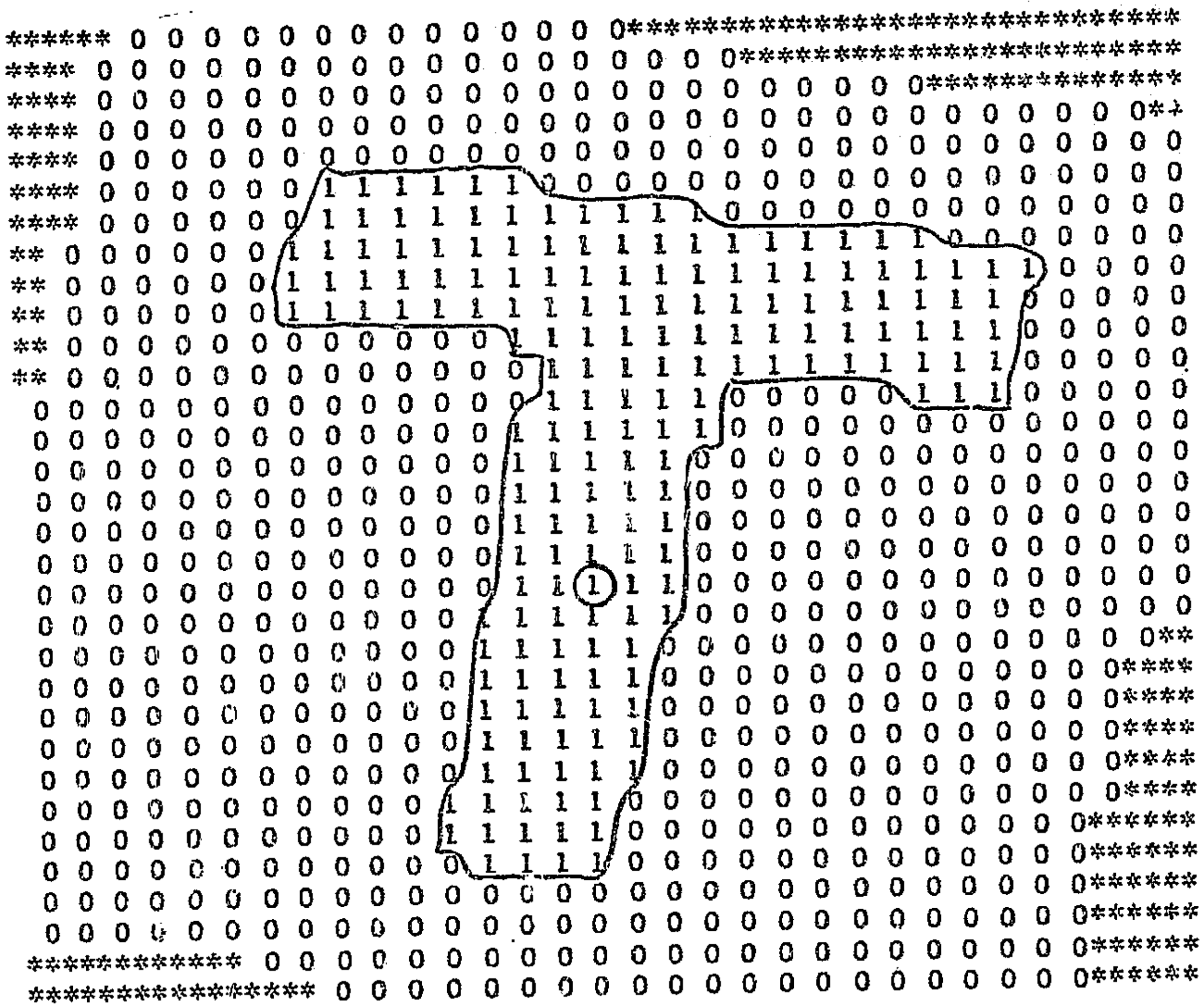


Figure 43, Example 6: (p,q) = (15,15), θ = 10° (Using Assigning Technique).

Example 7

rotation center (p,q) = (15,15)
rotation angle θ = 5°
The Assigning Technique

The rotated image is presented in Figure 44.

Example 8

rotation center (p,q) = (15,15)
rotation angle θ = -15°
The Assigning Technique

Figure 45 shows the rotated image.

Example 9

rotation center (p,q) = (31,0)
rotation angle θ = 10°
The Assigning Technique

In this example, the image is rotated about the lower right corner of the image plane. The final image is presented in Figure 46.

Example 10

rotation center (p,q) = (0,31)
rotation angle θ = -15°
The Assigning Technique

The image is rotated about the upper left corner of the image plane. Figure 47 presents the rotated image.

Example 11

rotation center (p,q) = (31,31)
rotation angle θ = 10°
The Assigning Technique

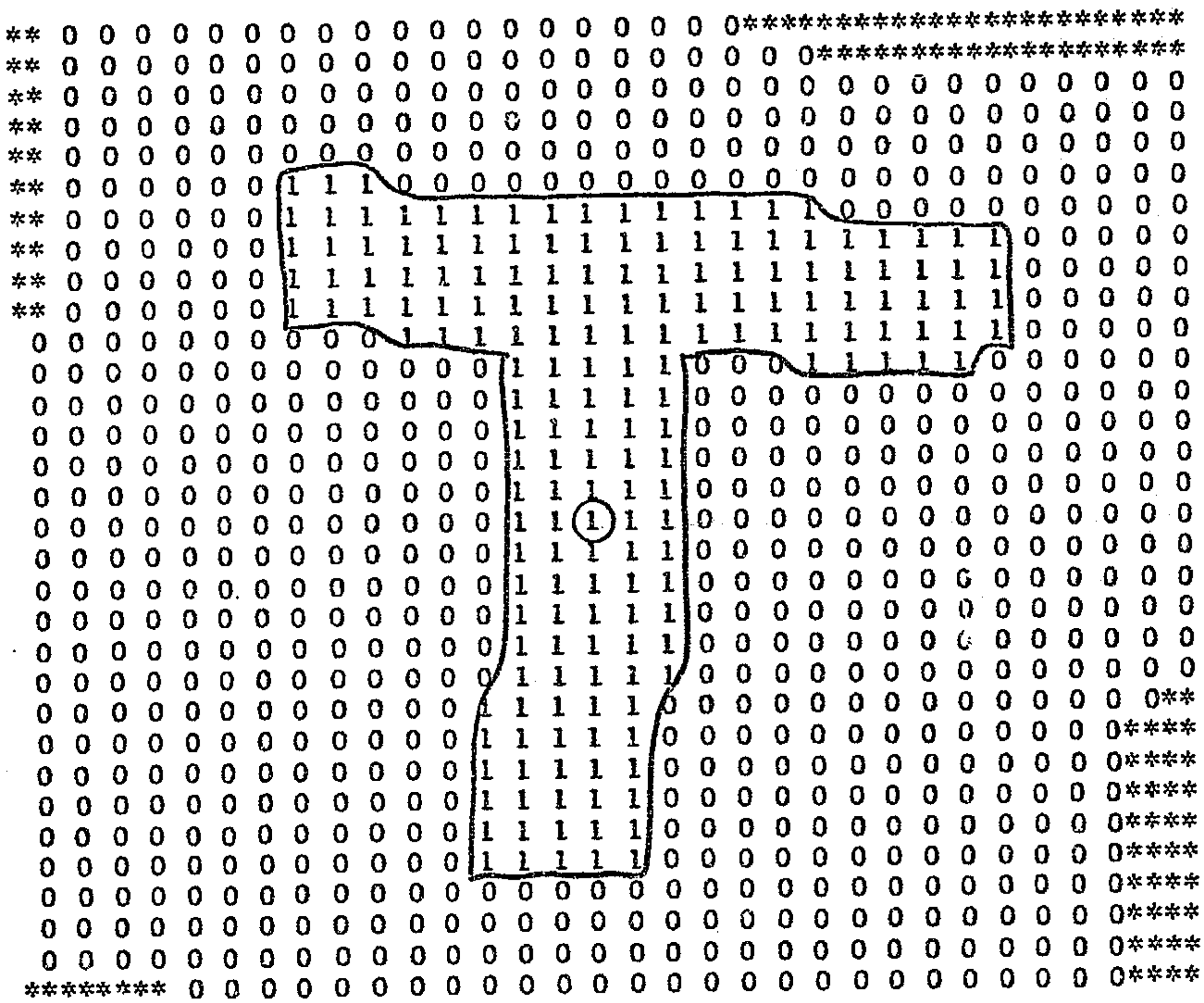


Figure 44. Example 7: $(p,q) = (15,15)$, $\theta = 5°$ (Using Assigning Technique).

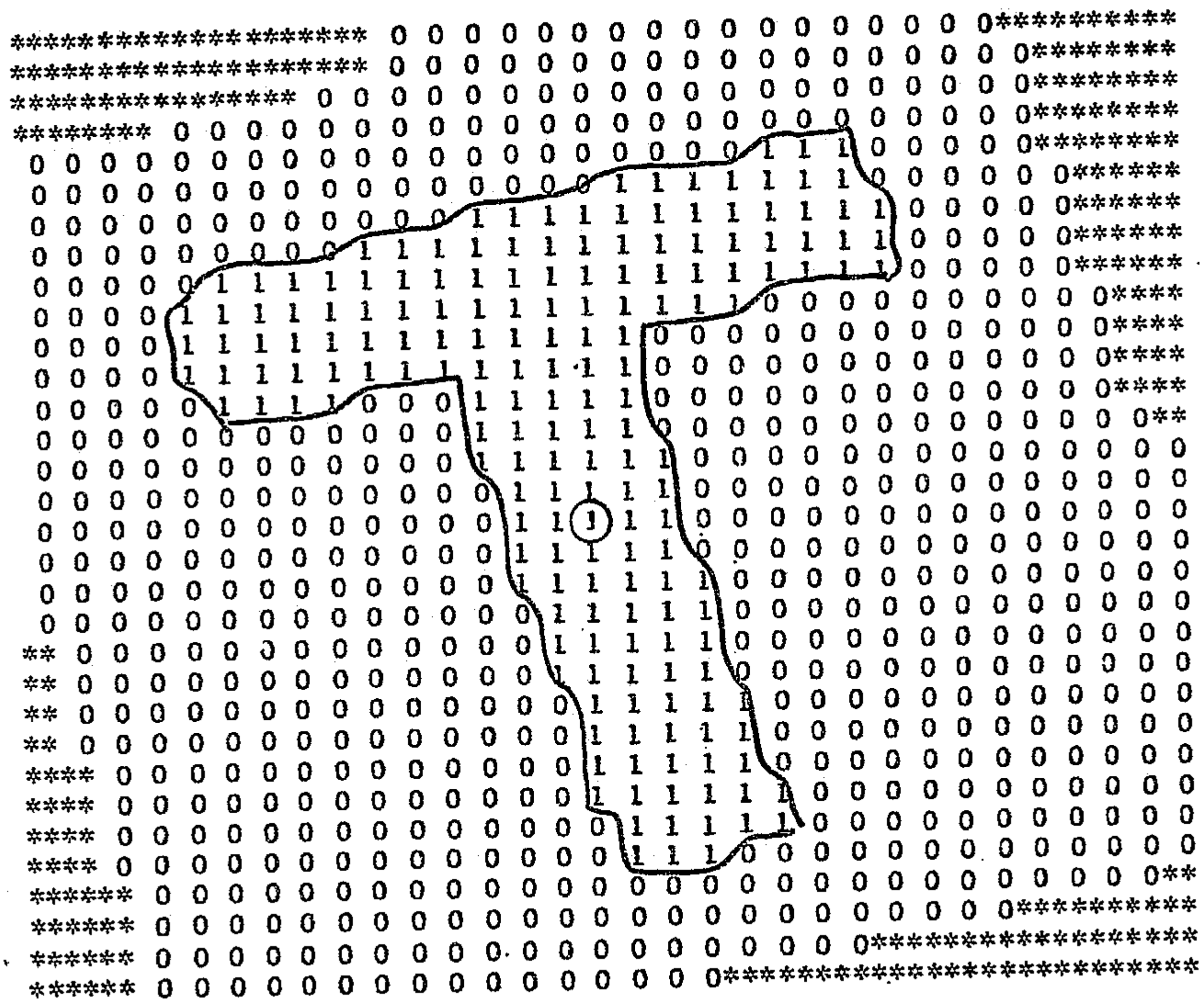


Figure 45. Example 8:(p,q) = (15,15), θ = -15° (Using Assigning Technique).

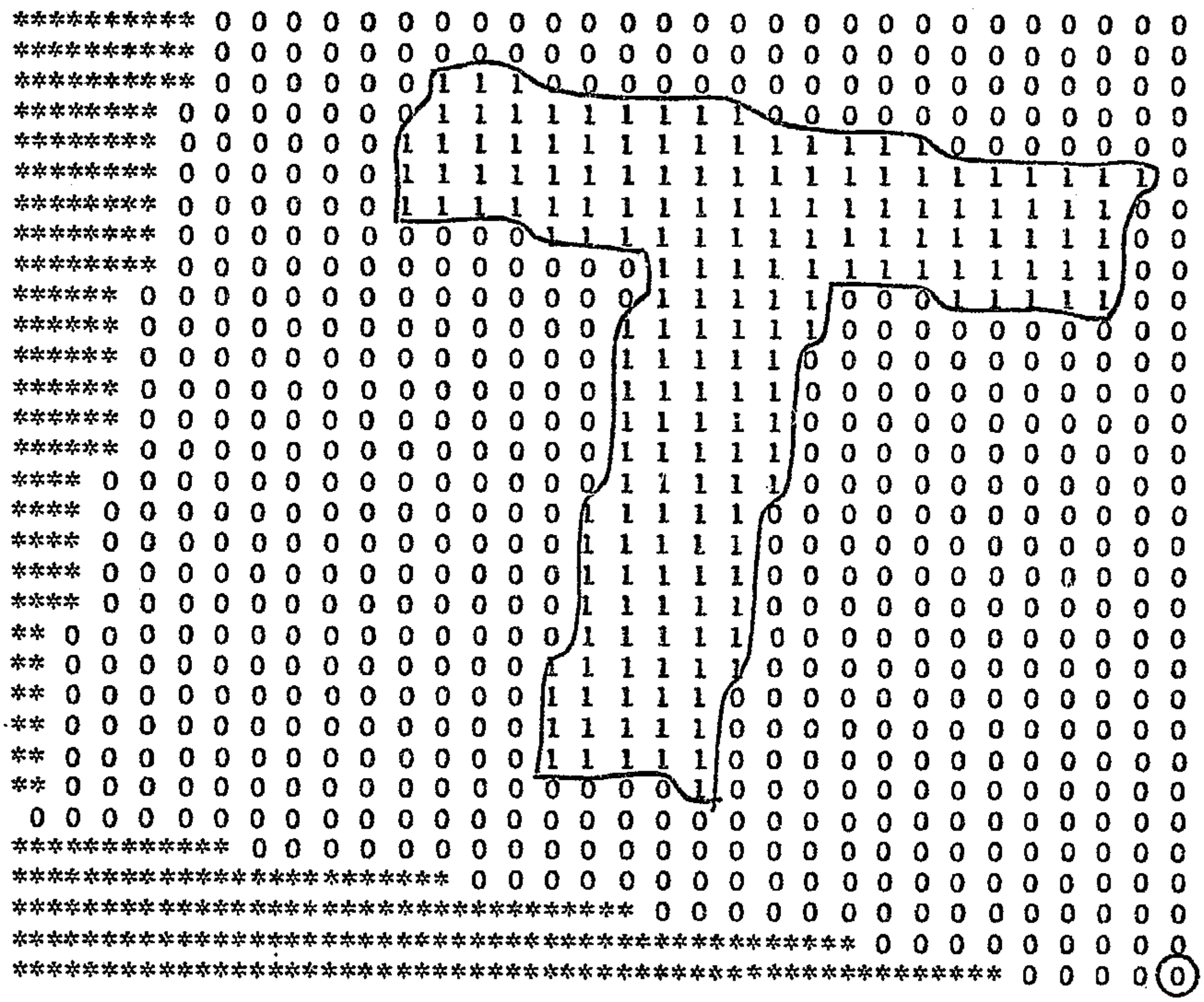


Figure 46. Example 9: (p,q) = (31,0), θ = 10° (Using Assigning Technique).

```
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0
** 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0
** 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0
** 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
** 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
**** 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
**** 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
**** 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
**** 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
****** 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
****** 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
****** 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
****** 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
******** 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
******** 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
******** 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0
********** 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0
********** 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0
********** 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0
********** 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0
************ 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0
************ 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0
************ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
************ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
************** 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0******
************** 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**************
************** 0 0 0 0 0 0 0 0 0 0 0 0 0 0**********************
************** 0 0 0 0 0 0 0 0 0 0 0****************************
**************** 0 0 0 0 0 0************************************
**************** 0 0********************************************
****************************************************************
```

Figure 47. Example 10, (p,q) = (0,31), $\theta = -15°$ (Using Assigning Technique).

The image is rotated about the upper right corner of the image plane. Figure 48 presents the rotated image.

Example 12

rotation center $(p,q) = (0,0)$
rotation angle $= 5°$
The Assigning Technique

The image is rotated about the lower left corner of the image plane. Figure 49 presents the rotated image.

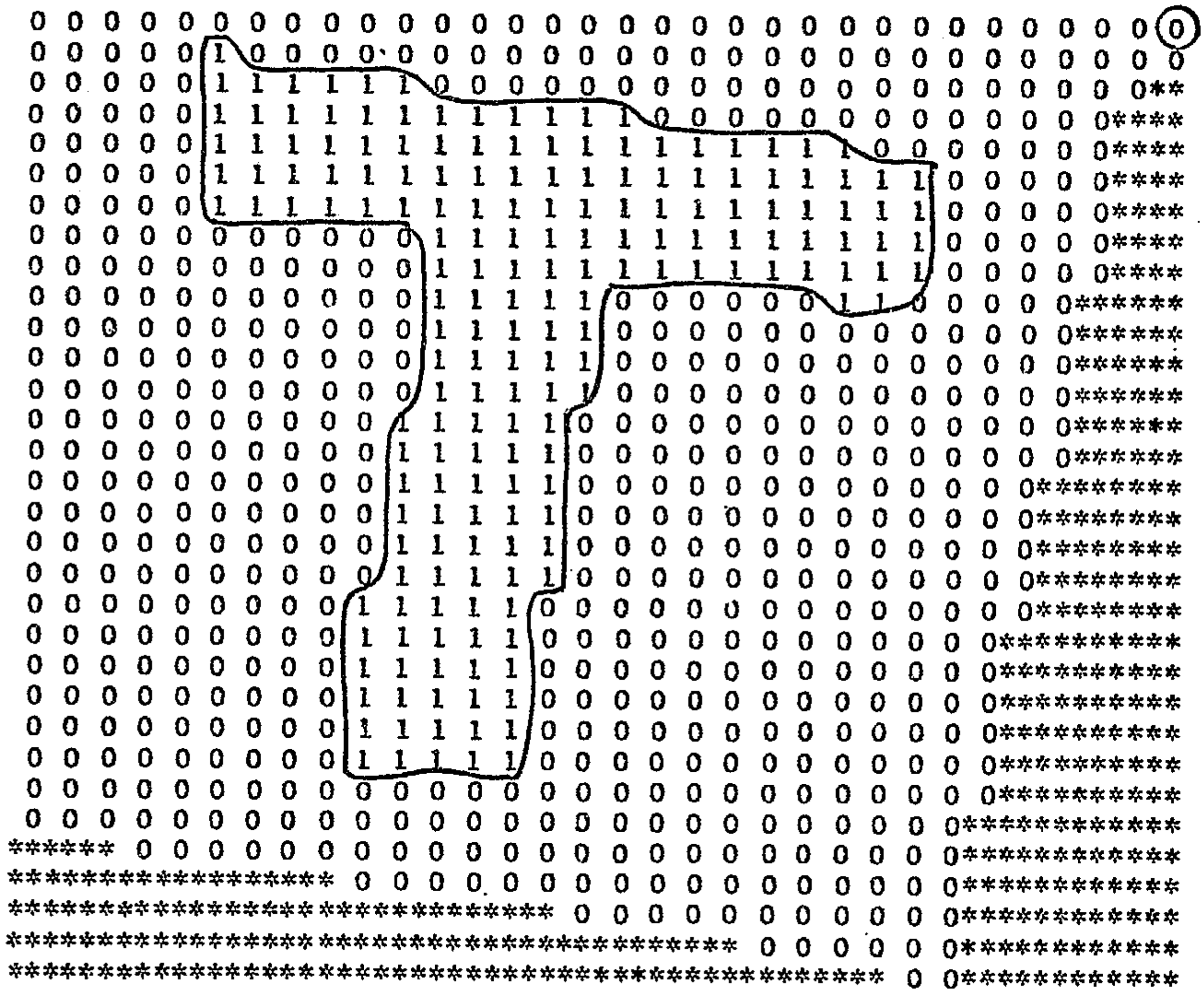


Figure 48. Example 11: $(p,q) = (31,31)$, $\theta = 10°$ (Using Assigning Technique)

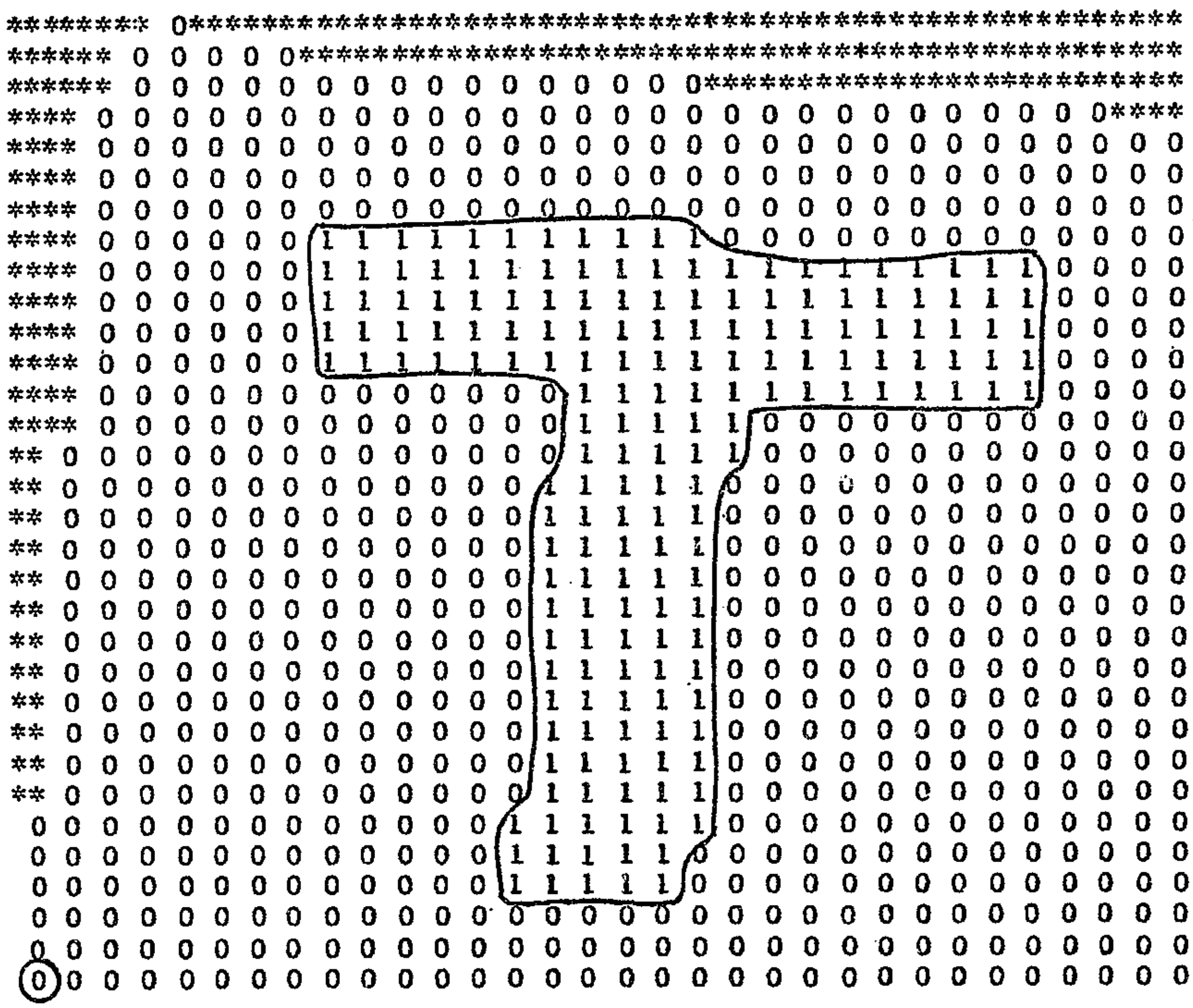


Figure 49. Example 12. (p,q) = (0,0), θ = 5° (Using Assigning Technique).

CHAPTER 10

IMPLEMENTATION

The slide procedure is the crucial step in the image rotation problem. Special hardware is required to facilitate the fast execution of the slide process. A simple and efficient hardware structure is proposed in this chapter. Such an organization can be easily incorporated with the tse computer architectures previously proposed [8,9,10].

As an example, Figure 50 presents a 5 x 5 image plane, in which columns 2 and 3 need to be slid upward one position, columns 4 and 5 need to be slid upward 2 and 3 positions, respectively. Obviously, sliding the image plane column by column would be very inefficient as far as operation time is concerned. For this example, 1 + 1 + 2 + 3 = 7 slide operations would be required. In addition, a mask for each column is also required. The slide procedure for the image plane can be resolved into three steps as shown in Figure 50. Since each column, with the exception of column 1, requires at least one vertical slide, a mask is used to slide the part of the image to the right of column 1 one position. Similarly, since the image to the right of column 3 needs at least another vertical slide, another mask, which can be obtained by sliding the mask in step 1 right two positions, can be used to slide columns 4 and 5. This same procedure holds true for Step 3. Observe that, in such a procedure, only 3 vertical slides are required. In general, the total number of slides is equal to the maximum number of slides required for any column, as compared to the sum of the slides required for each column

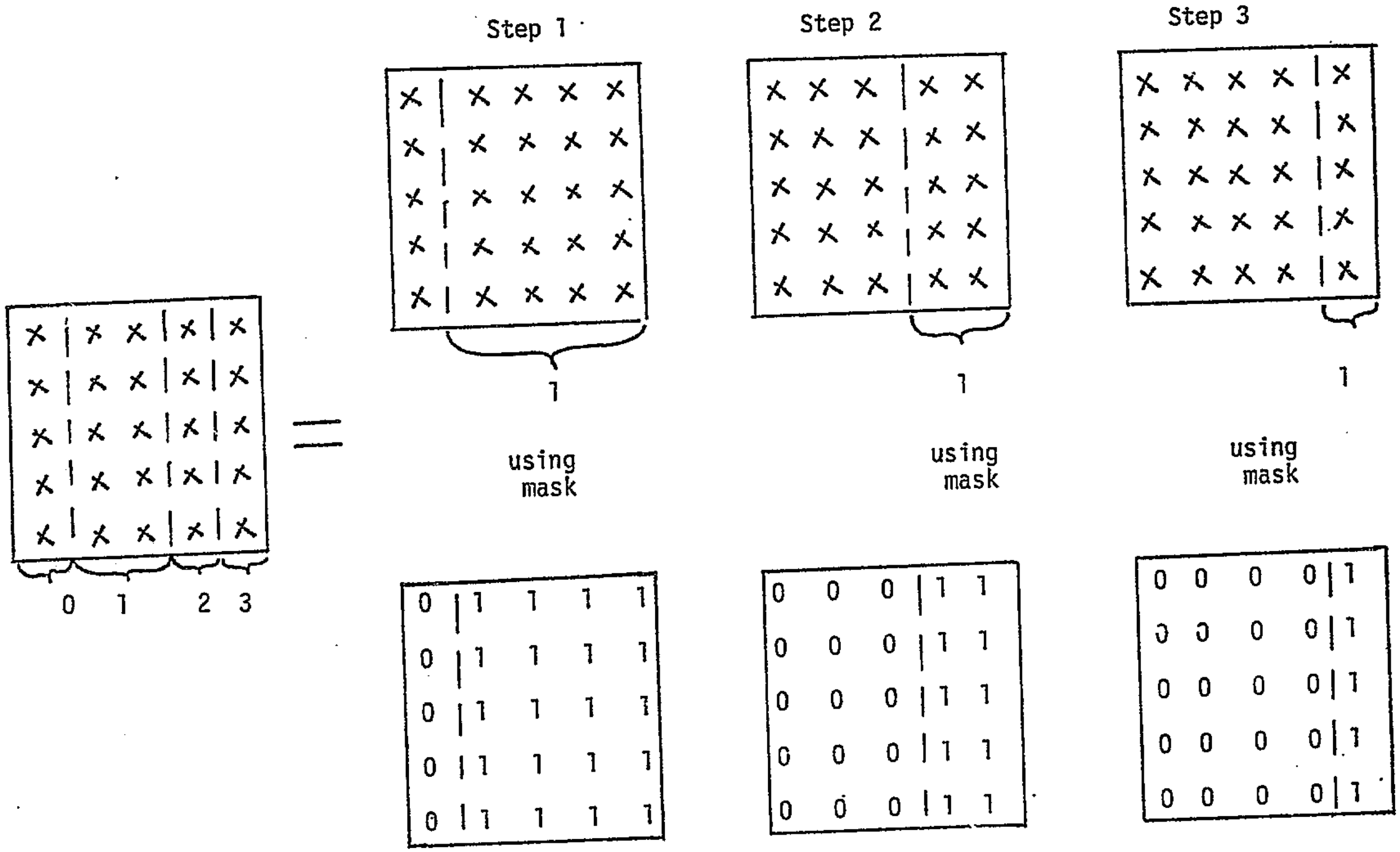


Figure 50. Image sliding using masks.

by a columnwise slide. The horizontal sliding of the image can be accomplished in a similar way.

A simple hardware organization which can efficiently execute the slide procedure for the above algorithm is proposed in Figure 51. The functional capabilities of this implementation are described below.

The Mask Generator is used to generate successively the necessary mask for the slide operations. Initially, an all 1's tse is loaded, which is then slid vertically (for horizontal sliding of the image) or horizontally (for vertical sliding of the image) to generate a series of masks. The mask and its complement are output to the Image Slider, where the image and its coordinates are slid tse by tse through the action of these masks.

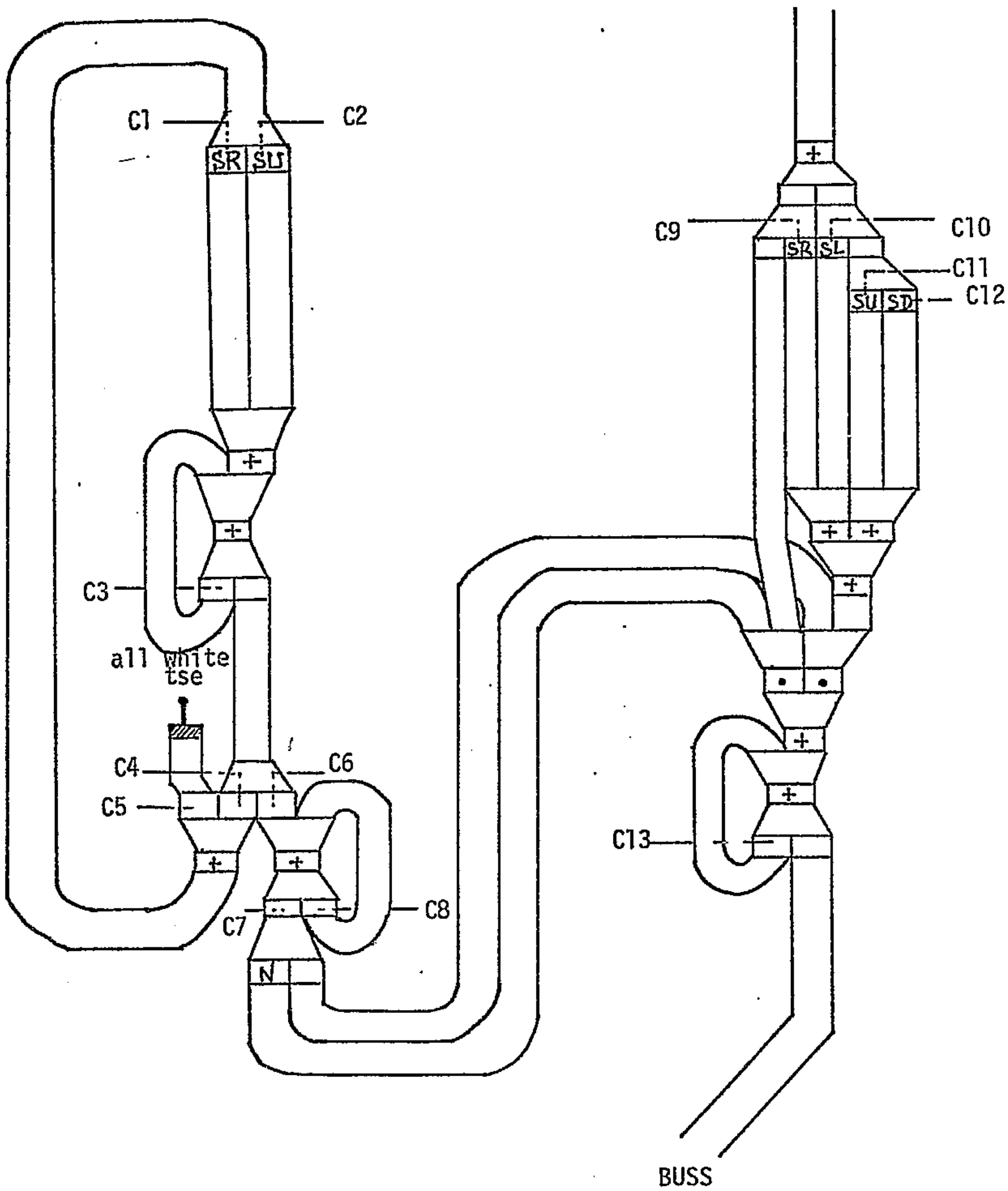


Figure 51. Implementation for slide operations.

CHAPTER 11

CONCLUSION

The coordinate transformations involved in the image rotation problem have been generated. General algorithms are proposed to perform image rotation using tse operations. Two simple and useful interpolative techniques have been developed. Various examples have been employed to demonstrate the correctness and the effectiveness of the proposed algorithms. By utilizing the hardware implementation of Figure 51, the lengthy slide procedure can be accomplished speedily and efficiently.

The algorithms of the three-step slide procedure and data extraction are essential to the problem. Their derivations are intended to be based upon as rigorous a mathematical treatment as possible. These developments have provided a successful solution to the image rotation problem. However, the method for the automatic control of the slide procedure has not been fully explored. The following subjects are recommended for further study.

Additional Simulations and Possible Refinements of the Proposed Control Method

Because of the experimental nature of the image rotation problem due to many round-off approximations in the derivations, a large number of simulation results should be generated to confirm the correctness of the proposed control method. Although many examples are employed in Chapter 9, most of these are simulated for the Assigning Technique. At most, one concludes that the proposed control method is successful for the cases using the Assigning Technique. However, the proposed control method has not been tested extensively for the cases using the Linear Least-Square-Error Interpolative Technique. The proposed control method may not be accurate enough to be suitable for all cases. For instance, the smoothing and filling of V" from the left may underestimated the complexity of the problem and thus may result in the failure of sliding some elements to one of their neighboring elements in some cases. These operations need to be tested extensively and refined, if necessary, to provide an accurate procedure. The procedure developed in this research has been successful in all cases tested.

Alternatives to the Proposed Control Method

Since the instruction cycle time of a conventional computer is significantly less than that of the tse computer, a control unit organized around a microprocessor is sufficiently fast for the computations of the required number of slides in the three-step slid procedure. Sooner or later, the instruction cycle time of the tse computer might become comparative with that of a conventional computer. In this situation, the microprocessor control of sliding may be too slow. A task should be initiated to determine if the control of the sliding can be accomplished within the tse processor itself.

REFERENCES

1. Unger, S.H., "A Computer Oriented Toward Spatial Problems," Proceedings of IRE, Vol. 46, October 1958, pp. 1744-1750.

2. Kruse, B., "A Parallel Picture Processing Machine," IEEE Transactions on Computers, Vol. C-22, No. 12, December 1973, pp. 1075-1086.

3. Lewin, D., Theory and Design of Digital Computers. London: Thomas Nelson and Sons LTD, 1972, pp. 307-311,

4. Edelstein, L.A., "'Picture Logic' For 'Bacchus' a Fourth-Generation Computer," The Computer Journal, Vol. 6, July 1963, pp. 144-153.

5. Schaefer, D.H., and Strong, J.P., Tse Computers. X-943-75-14, Goddard Space Flight Center, 1975.

6. Levy, H.H., "Earth-Resources Technology Satellite: NASA Data-Processing Facility," Geoscience Electronics, Vol. 8, No. 4, October 1970, pp. 348-352.

7. Nagy, G., "Digital Image Processing Activities in Remote Sensing for Earth Resources," Proceedings of the IEEE, Vol. 60, No. 10, October 1972, pp. 1177-1200.

8. Jones, J.R., "Extraction of Topological Information From An Image By the Tse Computer," M.S. Thesis, The University of Tennessee, June, 1975.

9. Metcalfe, A.G. and R.E. Bodenheimer, "Organizations of Array Logic (tse) Devices For Extracting A Parallel Counting Algorithm," IEEE Southeastern Symposium on System Theory, March 1975.

10. Rickard, D.A. and R.E. Bodenheimer, "Tse Logic Design Doncepts and the Development of Image Processing Machine Architectures," Technical Report TR-EE/CS-76-1, Department of Electrical Engineering, The University of Tennessee, Knoxville, Tennessee, September 1976.

APPENDIX A

DERIVATION OF THE LINEAR LEAST-SQUARE-ERROR INTERPOLATION TECHNIQUE

Figure A shows the grid point (j,k), or (x'_j, y'_k), of the new image plane and its four neighboring grid points. The value at the grid point is unknown and needs to be determined through some interpolation technique. A linear interpolation technique using four neighboring points is derived here, based on the "least squared error" criterion.

In this technique, the value at (x'_j, y'_k) will be determined through the linear interpolation over its four neighbors to satisfy the least-square-error condition. In other words, the plane which fits the four points with the least squared error will be found. Then, the interpolation value at (x'_j, y'_k) can be determined.

Let the known values at the four grid points (m,n), (m+1,n), (m,n+1), and (m+1,n+1) be $S_{m,n}$, $S_{m+1,n}$, $S_{m,n+1}$, and $S_{m+1,n+1}$, respectively, and let the equation of the linear interpolation plane be

$$f(x',y') = a(x' - m) + b(y' - n) + c \; . \qquad \text{(A1)}$$

Note that the summed squared error is

$$\begin{aligned} e^2 &= [f(m,n) - S_{m,n}]^2 + [f(m+1,n) - S_{m+1,n}]^2 + [f(m,n+1) - S_{m,n+1}]^2 \\ &\quad + [f(m+1,n+1) - S_{m+1,n+1}]^2 \\ &= (c - S_{m,n})^2 + (a + c - S_{m+1,n})^2 + (b + c - S_{m,n+1})^2 \\ &\quad + (a + b + c - S_{m+1,n+1})^2 \; . \qquad \text{(A2)} \end{aligned}$$

The necessary conditions for the least square error are

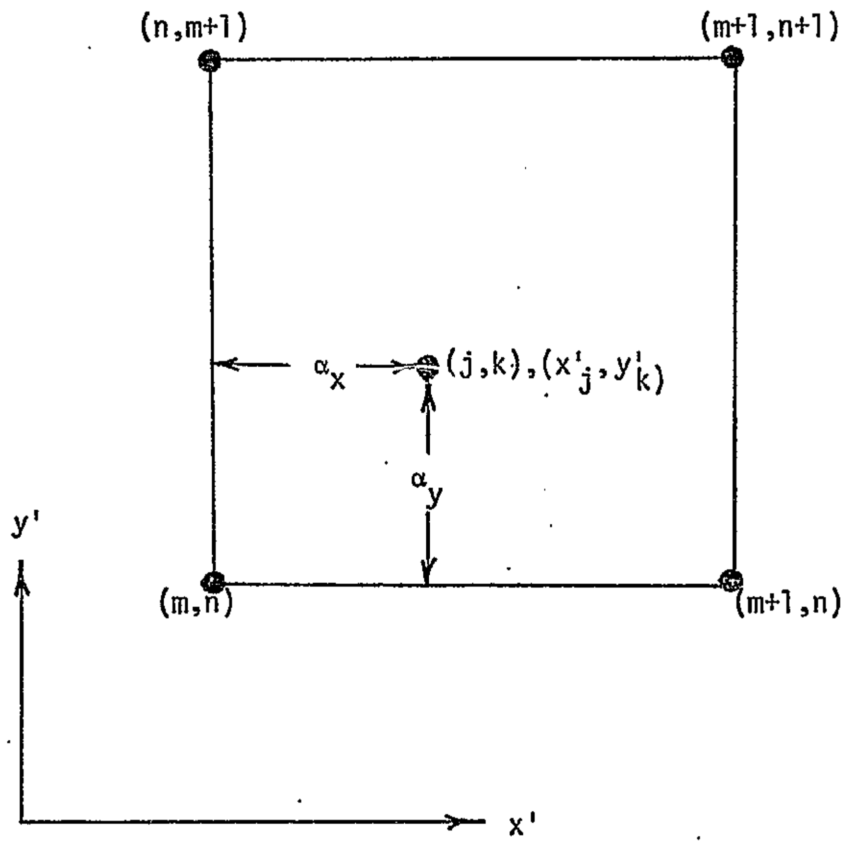


Figure A. Relationship between new grid point (j,k) and its four neighboring original grid points.

$$\frac{\partial e^2}{\partial a} = 0 \quad , \qquad \frac{\partial e^2}{\partial b} = 0 \quad \text{and} \quad \frac{\partial e^2}{\partial c} = 0 \; ; \quad \text{that is,}$$

$$2a + b + 2c = S_{m+1,n} + S_{m+1,n+1}, \tag{A3}$$

$$a + 2b + c = S_{m,n+1} + S_{m+1,n+1} , \tag{A4}$$

and

$$2a + 2b + 4c = S_{m,n} + S_{m,n+1} + S_{m+1,n} + s_{m+1,n+1} . \tag{A5}$$

Solving equations (A3), (A4), and (A5), gives

$$a = \frac{1}{2}\,(S_{m+1,n+1} + S_{m+1,n} - S_{m,n+1} - S_{m,n}), \tag{A6}$$

$$b = \frac{1}{2}\,(S_{m+1,n+1} + S_{m,n+1} - S_{m+1,n} - S_{m,n}), \tag{A7}$$

and

$$c = \frac{1}{4}(-S_{m+1,n+1} + S_{m,n+1} + S_{m+1,n} + 3\,S_{m,n}) . \tag{A8}$$

Substituting equations (A6), (A7), and (A8) into equation (A1), the equation of the least-square-error interpolation plane is obtained as

$$\begin{aligned} f(x',y') = {} & (\tfrac{1}{2})(S_{m+1,n+1} + S_{m+1,n} - S_{m,n+1} - S_{m,n})(x' - m) \\ & + (\tfrac{1}{2})(S_{m+1,n+1}) + S_{m,n+1} - S_{m+1,n} - S_{m,n})(y' - n) \\ & + (\tfrac{1}{4})(-S_{m+1,n+1} + S_{m,n+1} + S_{m+1,n} + 3\,S_{m,n}) . \end{aligned}$$

Hence the desired value at (x_j,y_k), or (j,k) is

$$f(j,k) = f(x_j,y_k) = \frac{1}{2}(S_{m+1,n+1} + S_{m+1,n} - S_{m,n+1} - S_{m,n})\alpha x$$
$$+ \frac{1}{2}(S_{m+1,n+1} + S_{m,n+1} - S_{m+1,n} - S_{m,n})\alpha y$$
$$+ \frac{1}{4}(-S_{m+1,n+1} + S_{m,n+1} + S_{m+1,n} + 3S_{m,n}) . \tag{A10}$$

Equation (A10) can be written as

$$f(j,k) = f(x_j,y_k) = W_{m,n}S_{m,n} + W_{m+1,n}S_{m+1,n} + W_{m,n+1}S_{m,n+1}$$
$$+ W_{m+1,n+1}S_{m+1,n+1} , \tag{A11}$$

where

$$W_{m,n} = -\frac{1}{2}\alpha x - \frac{1}{2}\alpha y + \frac{3}{4} , \tag{A12}$$

$$W_{m,n+1} = -\frac{1}{2}\alpha x + \frac{1}{2}\alpha y + \frac{1}{4} , \tag{A13}$$

$$W_{m+1,n} = \frac{1}{2}\alpha x - \frac{1}{2}\alpha y + \frac{1}{4} , \tag{A14}$$

and

$$W_{m+1,n+1} = \frac{1}{2}\alpha x + \frac{1}{2}\alpha y - \frac{1}{4} . \tag{A15}$$

The W's are the weights of the values at the four grid points with

$$W_{m,n} + w_{m,n+1} + W_{m,n+1} + W_{m+1,n+1} = 1 \tag{A16}$$

APPENDIX B

DERIVATION OF THE THREE-STEP SLIDE ALGORITHM

The success of the slide algorithm devised in Chapter 6 is due to the fact that the V-pattern is automatically aligned after Steps 1 and 2. This allows for the completion of the slide procedure by simply sliding the image columnwise as described in Step 3. The following is devoted to explain and to prove the essence of this algorithm. Equations (35) and (36) for the Assigning Technique are used for the derivation.

Step 1

Assume that element (m,n) is within the same H-zone as (p,n_0), which is an element at the same column as the rotation center (p,q), as shown in Figure B; in other words, their H values are equal.

$$H_0 = H \quad ,$$

where H_0 and H are given by equations (34) and (36),

$$H_0 = \langle (n_0 - q) \sin\theta \rangle$$

and

$$H = \langle -(1-\cos\theta)(m-p) + (n-q) \sin\theta \rangle \ .$$

Therefore,

$$\langle (n_0 - q) \sin\theta \rangle = \langle -(1-\cos\theta)(m-p) + (n-q) \sin\theta \rangle \ . \qquad \text{(B1)}$$

As shown in Figure B, more than one element at the p-column may lie within the same H-zone as (m,n). The ℓ_h values corresponding to these points are spaced by a value of $\sin\theta$. Therefore, at least one of these points can always be chosen as the (p,n_0) point corresponding to this (m,n) point, such that, dropping the "round-off function", equation (B1)

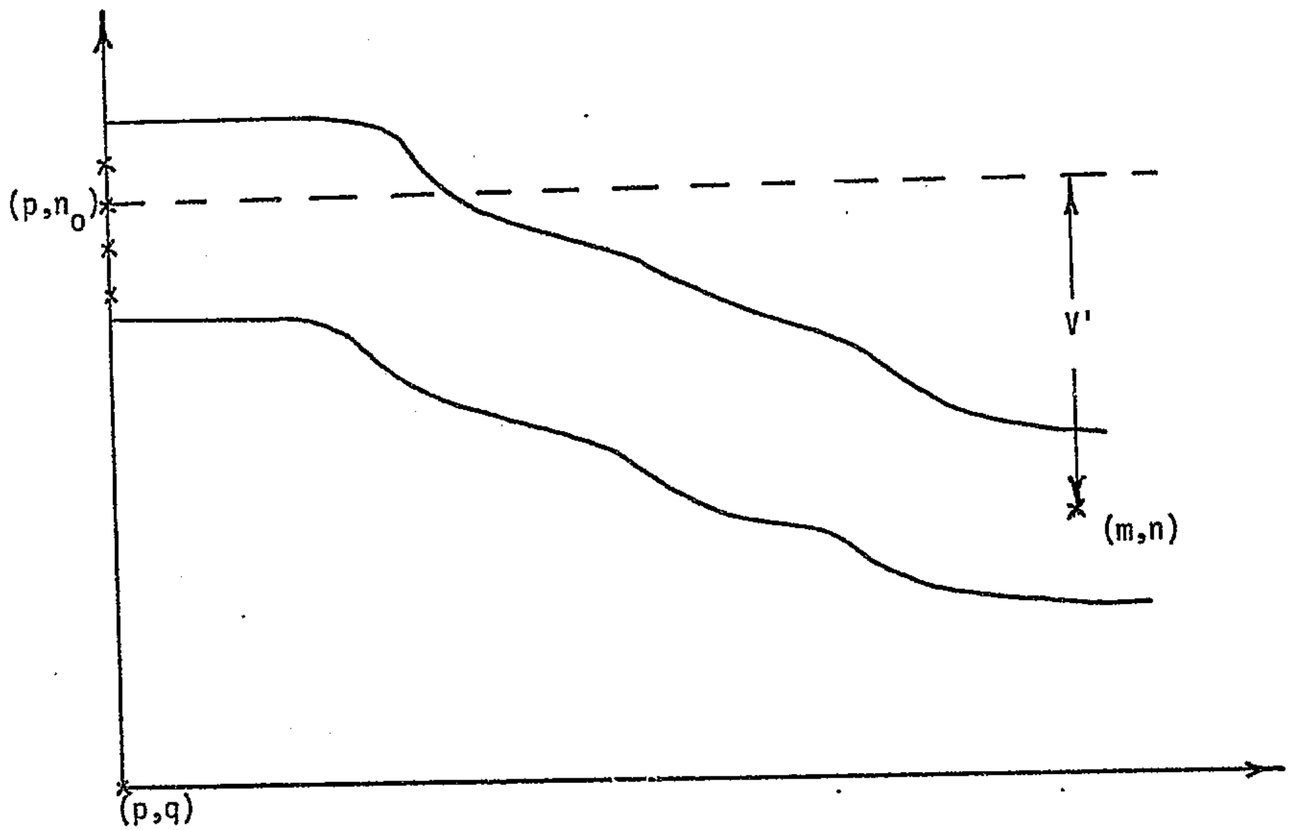


Figure B. Amount of H-pattern lining-up slides.

becomes

$$(n_0-q)\sin\theta = -(1-\cos\theta)(m-p) + (n-q)\sin\theta + E\{\sin\theta\};$$

or

$$(n_0-n)\sin\theta = (\cos\theta-1)(m-p) + E\{\sin\theta\} \quad , \tag{B2}$$

where $E\{\sin\theta\}$ is an error term of magnitude less than $\sin\theta$. Dividing both sides of equation (B2) by $\sin\theta$ yields

$$(n_0-n) = \frac{(\cos\theta-1)}{\sin\theta}(m-p) + \frac{E\{\sin\theta\}}{\sin\theta} \quad ;$$

or,

$$(n_0-n) = \frac{(\cos\theta-1)}{\sin\theta}(m-p) + E\{1\} \quad , \tag{B3}$$

where $E\{1\}$ is an error term of magnitude less than 1.

Observe that in order to align the H-pattern, the (m,n) point needs to be slid up by an amount $V' = n_0 - n$ such that (p,n_0) and (m,n) will be at the same horizontal position. Since the amount of the slides has to be an integer, the amount of vertical slide in Step 1 can be chosen as

$$V' = <\frac{\cos\theta-1}{\sin\theta}(m-p)> \tag{B4}$$

With this amount of H-pattern slides, any two points, say (m_1,n_1) and (m_2,n_2) will be slid vertically in Step 1 by

$$V_1' = <\frac{\cos\theta-1}{\sin\theta}(m_1-p)> \tag{B5}$$

and

$$V_2' = \left\langle \frac{\cos\theta - 1}{\sin\theta} (m_2 - p) \right\rangle \quad , \tag{B6}$$

respectively,

Step 2

Once the H-pattern is aligned, these two points are to be slid horizontally by the amount given by equations (34) and (36).

$$H_1 = \langle -(1-\cos\theta)(m_1-p) + (n_1-q)\sin\theta \rangle \tag{B7}$$

and

$$H_2 = \langle -(1-\cos\theta)(m_2-p) + (n_2-q)\sin\theta \rangle \quad , \tag{B8}$$

respectively.

Originally element (m_1,n_1) is at column m_1. After being slid horizontally by the amount H_1 in this step, this element is now at the column

$$m_1 + H_1 = m_1 + \langle -(1-\cos\theta)(m_1-p) + (n_1-q)\sin\theta \rangle \ . \tag{B9}$$

Similarly, element (m_2,n_2) is now at the column

$$m_2 + H_2 = m_2 + \langle -(1-\cos\theta)(m_2-p) + (n_2-q)\sin\theta \rangle \ . \tag{B10}$$

Step 3

What remains to be proven is that the V-pattern is aligned after Steps 1 and 2; in other words, any two elements at the same column will now have the same value of V.

Let (m_1,n_1) and (m_2,n_2) be at the same column after Steps 1 and 2,

$$m_1 + H_1 = m_2 + H_2 \quad ;$$

or, from equations (B9) and (B10),

$$m_1 + \langle -(1-\cos\theta)(m_1-p) + (n_1-q)\sin\theta \rangle = m_2 + \langle -(1-\cos\theta)(m_2-p) + (n_2-q)\sin\theta \rangle .$$

Dropping the round-off functions subject to a round-off error of the fractional part, the above equation becomes

$$m_1\cos\theta + n_1\sin\theta = m_2\cos\theta + n_2\sin\theta \quad . \qquad \text{(B11)}$$

The total amount of the vertical slide to be made for elements (m_1,n_1) and (m_2,n_2) is given by equations (33) and (35),

$$V_1 = \langle -(1-\cos\theta)(n_1-q) - (m_1-p)\sin\theta \rangle$$

and

$$V_2 = \langle -(1-\cos\theta)(n_2-q) - (m_2-p)\sin\theta \rangle ,$$

respectively.

Since these two elements have already been slid vertically in Step 1 by the amounts V_1' and V_2', respectively, the remaining amount of the required vertical slide for (m_1,n_1) is

$$V_1'' = V_1 - V_1' = \langle -(1-\cos\theta)(n_1-q) - (m_1-p)\sin\theta \rangle - \langle \frac{\cos\theta - 1}{\sin\theta}(m_1-p) \rangle$$

Again, dropping the round-off functions subject to a round-off error of the fractional part gives

$$V_1' = \frac{\cos\theta-1}{\sin\theta}\{(m_1-p)\cos\theta + (n_1-q)\sin\theta\} \quad . \tag{B12}$$

Similarly, for point (m_2,n_2),

$$V_2'' = \frac{\cos\theta-1}{\sin\theta}\{(m_2-p)\cos\theta + (n_2-q)\sin\theta\} \quad . \tag{B13}$$

From equations (B12) and (B13)

$$V_1'' - V_2'' = \frac{\cos\theta-1}{\sin\theta}\{(m_1-m_2)\cos\theta + (n_1-n_2)\sin\theta\}. \tag{B14}$$

Substituting equation (B11) in equation (B14) gives

$$V_1'' - V_2'' = 0 \quad , \tag{B15}$$

which means that (m_1,n_1) and (m_2,n_2) at the same column after Steps 1 and 2 require the same amount of vertical slide.

Notice that the above derivationsare subject to the round-off errors, which result in the differences of magnitude 1 as shown in the example in Chapter 6. Extraction of the data is presented in Chapter 7 to take care of these differences.

APPENDIX C

SIMULATION PROGRAM

```
      DIMENSION XJ(32,32),YK(32,32),NVP(32),NH(32),NVPP(32)
      INTEGER*2 OUTBND,M(32,32),N(32,32),MF(32,32),NF(32,32),
     $          IM(32,32),SMN(32,32)
      INTEGER P,Q
      DATA OUTBND/' *'/,SMN/1024*0/
C
C******** ORIGINAL POSITION (M,N) ********
C
      DO 1 I=1,32
      DO 1 J=1,32
      M(I,J)=I-1
    1 N(I,J)=J-1
      DO 11 K=1,32
      J=33-K
C        WRITE OUT THE ORIGINAL COORDINATES
   11 WRITE(6,100) (M(I,J),I=1,32)
      WRITE(6,200)
  200 FORMAT('1')
      DO 12 K=1,32
      J=33-K
   12 WRITE(6,100) (N(I,J),I=1,32)
  100 FORMAT(1X,32(I2))
      WRITE(6,..0)
C
C******** READ IN AND WRITE OUT IMAGE ********
C
C        P,Q: COORDINATES OF ROTATION CENTER
C        TET: ROTATION ANGLE
C        LLL=1: USING ASSIGNING TECH(CLOCKWISE ROTATION)
C        LLL=2: USING ASSIGNING TECH(COUNTER-CLOCKWISE ROTATION)
C        LLL=3: USING LLSE TECH(CLOCKWISE ROTATION)
C        LLL=4: USING LLSE TECH(COUNTER-CLOCKWISE ROTATION)
      DO 6 K=1,32
      J=33-K
    6 READ(5,300) (IM(I,J),I=1,32)
  300 FORMAT(32I1)
      READ(5,310) TET,P,Q,LLL
  310 FORMAT(F10.6,I2,I2,I1)
      DO 7 K=1,32
      J=33-K
    7 WRITE(6,100) (IM(I,J),I=1,32)
      WRITE(6,200)
C
C******** FINAL POSITION (MF,NF) ********
C
      COSTET=COS(TET)
      SINTET=SIN(TET)
      DO 2 J=1,32
      DO 2 K=1,32
      XJ(J,K)=FLOAT(M(J,K)-P)*COSTET-FLOAT(N(J,K)-Q)*SINTET
     $        +FLOAT(P)
```

```
      YK(J,K)=FLOAT(M(J,K)-P)*SINTET+FLOAT(N(J,K)-Q)*COSTET
     $        +FLOAT(Q)
      GO TO(24,24,25,25),LLL
C          FOR ASSIGNING TECHNIQUE
   24 IF(XJ(J,K).GE.-0.5) GO TO 21
      MF(J,K)=INT(XJ(J,K)+0.5)-1
      GO TO 22
   21 MF(J,K)=INT(XJ(J,K)+0.5)
   22 IF(YK(J,K).GE.-0.5) GO TO 23
      NF(J,K)=INT(YK(J,K)+0.5)-1
      GO TO 2
   23 NF(J,K)=INT(YK(J,K)+0.5)
      GO TO 2
C          FOR LINEAR LEAST-SQUARE-ERROR TECHNIQUE
   25 IF(XJ(J,K).GE.0.0) GO TO 26
      MF(J,K)=INT(XJ(J,K))-1
      GO TO 27
   26 MF(J,K)=INT(XJ(J,K))
   27 IF(YK(J,K).GE.0.0) GO TO 28
      NF(J,K)=INT(YK(J,K))-1
      GO TO 2
   28 NF(J,K)=INT(YK(J,K))
    2 CONTINUE
C          DEFINE THE BOUNDARY OF THE ROTATED IMAGE
      DO 3 J=1,32
      DO 3 K=1,32
      IF(MF(J,K).LT.0.OR.MF(J,K).GE.31.OR.NF(J,K).LT.0.OR.
     $   NF(J,K).GE.31) GO TO 31
      GO TO 3
   31 MF(J,K)=OUTBND
      NF(J,K)=OUTBND
    3 CONTINUE
      DO 32 K=1,32
      J=33-K
   32 WRITE(6,100) (MF(I,J),I=1,32)
      WRITE(6,200)
      DO 33 K=1,32
      J=33-K
   33 WRITE(6,100) (NF(I,J),I=1,32)
      WRITE(6,200)
C
C******** STEP 1 ********
C
      CALL CONTR1(SINTET,COSTET,P,Q,NVP)
C  NVP(I): NUMBER OF VERTICAL SLIDES FOR COLUMN (I) IN STEP 1
      DO 4 I=1,32
      K=NVP(I)
      IF(K.GT.0) GO TO 46
      IF(K.LT.0) GO TO 47
      GO TO 4
```

```
C         SLIDE UP
   46 KK=32-K
      DO 41 J=1,KK
      JJ=33-J
      M(I,JJ)=M(I,JJ-K)
      N(I,JJ)=N(I,JJ-K)
   41 IM(I,JJ)=IM(I,JJ-K)
      DO 42 J=1,K
      M(I,J)=OUTBND
      N(I,J)=OUTBND
   42 IM(I,J)=OUTBND
      GO TO 4
C         SLIDE DOWN
   47 KK=32+K
      DO 43 J=1,KK
      M(I,J)=M(I,J-K)
      N(I,J)=N(I,J-K)
   43 IM(I,J)=IM(I,J-K)
      KK1=KK+1
      DO 45 J=KK1,32
      M(I,J)=OUTBND
      N(I,J)=OUTBND
   45 IM(I,J)=OUTBND
    4 CONTINUE
      DO 44 K=1,32
      J=33-K
   44 WRITE(6,100) (IM(I,J),I=1,32)
      WRITE(6,200)
C
C******** STEP 2 ********
C
      CALL CONTR2(SINTET,COSTET,P,Q,NH,LLL)
C  NH(I): NUMBER OF HORIZONTAL SLIDES FOR ROW(I) IN STED 2
      DO 5 J=1,32
      K=NH(J)
      IF(K.GT.0) GO TO 53
      IF(K.LT.0) GO TO 54
      GO TO 5
C         SLIDE LEFT
   54 KK=32+K
      DO 51 I=1,KK
      M(I,J)=M(I-K,J)
      N(I,J)=N(I-K,J)
   51 IM(I,J)=IM(I-K,J)
      KK1=KK+1
      DO 52 I=KK1,32
      M(I,J)=OUTBND
      N(I,J)=OUTBND
   52 IM(I,J)=OUTBND
      GO TO 5
```

```
C         SLIDE RIGHT
   53 KK=32-K
      DO 82 I=1,KK
      II=33-I
      M(II,J)=M(II-K,J)
      N(II,J)=N(II-K,J)
   82 IM(II,J)=IM(II-K,J)
      DO 83 I=1,K
      M(I,J)=OUTBND
      N(I,J)=OUTBND
   83 IM(I,J)=OUTBND
    5 CONTINUE
      DO 81 K=1,32
      J=33-K
   81 WRITE(6,100) (IM(I,J),I=1,32)
      WRITE(6,200)
C
C******** STEP 3 ********
C
      CALL CONTR3(SINTET,COSTET,P,Q,NVP,NVPP,LLL)
C  NVPP(I): NUMBER OF VERTICAL SLIDES FOR COLUMN(I) IN STEP 3
      DO 10 I=1,32
      K=NVPP(I)
      IF(K.GT.0) GO TO 110
      IF(K.LT.0) GO TO 111
      GO TO 10
C         SLIDE UP
  110 KK=32-K
      DO 101 J=1,KK
      JJ=33-J
      M(I,JJ)=M(I,JJ-K)
      N(I,JJ)=N(I,JJ-K)
  101 IM(I,JJ)=IM(I,JJ-K)
      DO 102 J=1,K
      M(I,J)=OUTBND
      N(I,J)=OUTBND
  102 IM(I,J)=OUTBND
      GO TO 10
C         SLIDE DOWN
  111 KK=32+K
      DO 201 J=1,KK
      M(I,J)=M(I,J-K)
      N(I,J)=N(I,J-K)
  201 IM(I,J)=IM(I,J-K)
      KK1=KK+1
      DO 202 J=KK1,32
      M(I,J)=OUTBND
      N(I,J)=OUTBND
  202 IM(I,J)=OUTBND
   10 CONTINUE
```

```
      DO 203 K=1,32
      J=33-K
  203 WRITE(6,100) (IM(I,J),I=1,32)
      WRITE(6,200)
      DO 204 K=1,32
      J=33-K
  204 WRITE(6,100) (M(I,J),I=1,32)
      WRITE(6,200)
      DO 205 K=1,32
      J=33-K
  205 WRITE(6,100) (N(I,J),I=1,32)
      WRITE(6,200)
C
C******** EXTRACT DATA AND GENERATE SMN ********
C
      DO 30 I=1,32
      DO 30 J=1,32
      IF(MF(I,J).EQ.M(I,J).AND.NF(I,J).EQ.N(I,J)) GO TO 301
      GO TO 30
  301 SMN(I,J)=IM(I,J)
   30 CONTINUE
      DO 40 I=1,31
      DO 40 J=1,32
      I1=I+1
      IF(MF(I,J).EQ.M(I1,J).AND.NF(I,J).EQ.N(I1,J)) GO TO 401
      GO TO 40
  401 SMN(I,J)=IM(I1,J)
   40 CONTINUE
      DO 50 I=2,32
      DO 50 J=1,32
      I1=I-1
      IF(MF(I,J).EQ.M(I1,J).AND.NF(I,J).EQ.N(I1,J)) GO TO 501
      GO TO 50
  501 SMN(I,J)=IM(I1,J)
   50 CONTINUE
      DO 60 I=1,32
      DO 60 J=1,31
      J1=J+1
      IF(MF(I,J).EQ.M(I,J1).AND.NF(I,J).EQ.N(I,J1)) GO TO 601
      GO TO 60
  601 SMN(I,J)=IM(I,J1)
   60 CONTINUE
      DO 70 I=1,32
      DO 70 J=2,32
      J1=J-1
      IF(MF(I,J).EQ.M(I,J1).AND.NF(I,J).EQ.N(I,J1)) GO TO 701
      GO TO 70
  701 SMN(I,J)=IM(I,J1)
   70 CONTINUE
```

```
      DO 80 I=1,31
      DO 80 J=1,31
      I1=I+1
      J1=J+1
      IF(MF(I,J).EQ.M(I1,J1).AND.NF(I,J).EQ.N(I1,J1)) GO TO 801
      GO TO 80
  801 SMN(I,J)=IM(I1,J1)
   80 CONTINUE
        DO 90 I=2,32
      DO 90 J=2,32
      I1=I-1
      J1=J-1
      IF(MF(I,J).EQ.M(I1,J1).AND.NF(I,J).EQ.N(I1,J1)) GO TO 901
      GO TO 90
  901 SMN(I,J)=IM(I1,J1)
   90 CONTINUE
      DO 1000 I=1,31
      DO 1000 J=2,32
      I1=I+1
      J1=J-1
      IF(MF(I,J).EQ.M(I1,J1).AND.NF(I,J).EQ.N(I1,J1)) GO TO 1001
      GO TO 1000
 1001 SMN(I,J)=IM(I1,J1)
 1000 CONTINUE
      DO 2000 I=2,32
      DO 2000 J=1,31
      I1=I-1
      J1=J+1
      IF(MF(I,J).EQ.M(I1,J1).AND.NF(I,J).EQ.N(I1,J1)) GO TO 2001
      GO TO 2000
 2001 SMN(I,J)=IM(I1,J1)
 2000 CONTINUE
      DO 2002 K=1,32
      J=33-K
 2002 WRITE(6,100)  (SMN(I,J),I=1,32)
      WRITE(6,200)
      STOP
      END
```

```
      SUBROUTINE CONTR1(SINTET,COSTET,P,Q,NVP)
C
C******** CONTROL OF STEP 1 *********
C
      DIMENSION VP(32),NVP(32)
      INTEGER P,Q
      DO 10 I=1,32
      VP(I)=-(1.0-COSTET)*FLOAT(I-1-P)/SINTET
      IF(VP(I).GE.-0.5) GO TO 11
      NVP(I)=INT(VP(I)+0.5)-1
      GO TO 10
   11 NVP(I)=INT(VP(I)+0.5)
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CONTR2(SINTET,COSTET,P,Q,NH,LLL)
C
C******** CONTROL OF STEP 2 ********
C
      DIMENSION H(32),NH(32)
      INTEGER P,Q
      GO TO(22,22,23,22),LLL
   22 DO 20 I=1,32
      H(I)=FLOAT(I-1-Q)*SINTET+0.5
      IF(H(I).GE.0.0) GO TO 21
      NH(I)=INT(H(I))-1
      GO TO 20
   21 NH(I)=INT(H(I))
   20 CONTINUE
      GO TO 24
   23 DO 30 I=1,32
      H(I)=FLOAT(I-1-Q)*SINTET+0.5
      IF(H(I).GE.0.0) GO TO 31
      NH(I)=INT(H(I))
      GO TO 30
   31 NH(I)=INT(H(I))+1
   30 CONTINUE
   24 RETURN
      END
```

```
      SUBROUTINE CONTR3(SINTET,COSTET,P,Q,NVP,NVPP,LLL)
C
C******** CONTROL OF STEP 3 **********
C
      DIMENSION VD(32),HD(32),NVD(32),NHD(32),NVPP(32),NVP(32)
      INTEGER P,Q
      DO 30 I=1,32
      GO TO(33,34,33,34),LLL
   33 J=I
      GO TO 35
   34 J=33-I
   35 VD(I)=-(1.0-COSTET)*FLOAT(J-1-Q)-FLOAT(I-1-P)*SINTET+0.5
      HD(I)=-(1.0-COSTET)*FLOAT(I-1-P)+FLOAT(J-1-Q)*SINTET+0.5
      IF(VD(I).GE.0.0) GO TO 31
      NVD(I)=INT(VD(I))-1
      GO TO 32
   31 NVD(I)=INT(VD(I))
   32 IF(HD(I).GE.0.0) GO TO 23
      NHD(I)=INT(HD(I))-1
      GO TO 30
   23 NHD(I)=INT(HD(I))
   30 CONTINUE
      GO TO(51,51,52,53),LLL
   52 DO 10 I=1,32
      NHD(I)=NHD(I)+1
   10 CONTINUE
      GO TO 51
   53 DO 20 I=1,32
      NVD(I)=NVD(I)+1
   20 CONTINUE
   51 IIL=0
C
      DO 40 I=1,32
      II=I+NHD(I)
      IF(II.LT.1) GO TO 40
      IF(II.GT.32) GO TO 61
C        FILLING NVPP
   43 IF(II-IIL-1) 41,41,42
   42 NVPP(IIL+1)=NVPP(IIL)
      IIL=IIL+1
      GO TO 43
C
   41 NVPP(II)=NVD(I)-NVP(I)
      IF(II.EQ.1) GO TO 46
C        SMOOTHING NVPP
      GO TO(44,45,44,45),LLL
   44 IF(NVPP(II).GT.NVPP(II-1)) NVPP(II)=NVPP(II-1)
      GO TO 46
```

```
45 IF(NVPP(II).LT.NVPP(II-1)) NVPP(II)=NVPP(II-1)
   GO TO 46
48 IF(I.EQ.32.AND.II.NE.32) GO TO 47
   GO TO 40
47 NVPP(II+1)=NVPP(II)
   II=II+1
   GO TO 48
46 IIL=II
40 CONTINUE
61 IF(IIL.EQ.32) GO TO 49
   NVPP(IIL+1)=NVPP(IIL)
   IIL=IIL+1
   GO TO 61
49 RETURN
   END
```