METHODS OF EDGE DETECTION IN VISUAL SCENES *

by

S. A. Underwood and J. K. Aggarwal
Department of Electrical Engineering

Technical Report No. 144
February 1973

INFORMATION SYSTEMS RESEARCH LABORATORY

ELECTRONICS RESEARCH CENTER
THE UNIVERSITY OF TEXAS AT AUSTIN
Austin, Texas 78712

* The research reported herein was supported by NSF Grant GK 32595X and JSEP Contract F44620-71-C-0091.

ABSTRACT

Several methods of detecting edges in visual scenes by a digital computer are presented. There are two broad procedures of edge detection, called one- and two-pass methods. Examples illustrating some of the methods in each catagory are given.

TABLE OF CONTENTS

LIST OF FIGURES

1. Introduction

Visual pattern recognition is a process whereby a digital computer uses a visual scene as an input and deduces certain properties about the content of the input data. The usefulness of visual pattern recognition is obvious when one considers how much the human decision process is dependent on vision as the major input media. Several applications of visual pattern recognition are briefly reviewed in the following.

The earlier goals of visual pattern recognition centered around the computer recognition of machine imprinted alphanumeric characters [1]. This application has been widely accepted for the automatic reading of commercial documents. Present research in this area is directed toward the automatic recognition of hand printed alphanumeric characters [2]. Research into the use of vision as a controlling input to a computer operated mechanical manipulator is being performed at MIT [3], Stanford [4], and Hitachi Ltd. [5]. In these three projects the vision input to the computer consists of the output of a television camera which observes the environment in which the manipulator is to operate. Use of a television camera input for the control of mobil automatons is being implemented at SRI [6], MIT Lincoln Labs [7], and JPL [8]. Here the vision information is used to control the movement of the automaton in an unknown environment and enable it to perform given tasks.

One area of considerable current interest is the automatic extraction of information from biomedical images. A major effort has been devoted to chromosome karyotyping [9] and X-ray analysis [10]. Eventually the automation and standardization of laboratory processes will be realized by the results of biomedical image processing. Another major research area of interest is the automatic analysis of multi-channel spectral data from airborn scanners [11]. The earth is scanned with twelve spectral scanners and one digitizes the intensities in twelve bands from 0.40

to 1.00 microns. From the digitized data, the types and conditions of agriculture are automatically determined by using pattern recognition techniques. Other applications of this research effort include the discovery of minerals. The twelve-channel scanners and the use of 3-color visual scanners [12] add another dimension to the pattern recognition process .

In the above mentioned applications, the visual information consists of binary black or white or multi-gray-level digitization of the spectral intensities in a visual scene. The goal of the visual pattern recognition is to extract descriptions of objects present in the visual scene. An object is a body which contrasts light intensity with its surrounding area. Therefore, edge detection is a necessary first step in order to determine the presence and location of objects in the visual scene.

An "edge point" is defined as a point in a two-dimensional visual scene which contrasts with its adjacent points. An "edge" is a string of adjacent edge points of length greater than one. The process of detecting the edges in a visual scene can be broken into two steps:

(a) Edge point detection

(b) Edge fitting

In the present paper we shall discuss both edge point detection and edge fitting techniques. Algorithms and procedures will be presented for edge detection together with examples which use these techniques. The visual data for the examples was produced by an Image Dissector television camera, connected to a SDS-930 computer in the College of Engineering, The University of Texas at Austin.

It is not imperative but generally desirable to filter the data before the process of edge detection is implemented on the data. This filtering may be performed in the time domain via a digital filter or in the frequency domain via fast fourier transform. The process of prefiltering may be used to emphasize certain frequencies, de-emphasize certain other frequencies, remove noise, and is called image enhancement. Image enhancement will not be discussed in present paper and it will be the subject of a companion paper. However it may be observed that the results of the edge detection algorithm will, in general, depend upon the type of image enhancement used on the data.

2. Edge Detection

The edges which appear in visual scenes are formed by the meeting of two areas of somewhat uniform intensity. An edge is characterized as a distinct change in light intensity between neighboring points in the scene. The detection of an edge can, therefore, be accomplished by looking for these intensity changes in the digitized scene. The method of examining the scene can be global or local. The two methods are characterized by the amount of input data which is considered by the algorithm at one time.

Global methods consider all, or a large subset, of the input information which is available at one time. The algorithm can be characterized as a parallel processor which takes all of the available data and reduces it to a single output describing the input data. The implementation of perceptrons [13] is an example of a global method.

Local methods are algorithms which consider only a small part of the input data base at one time. Small subsets of the available input data are analyzed and local decisions about the information contained in each subset are made. Information which has been derived from these local subsets is then connected to form decisions about the complete input data base. Most of the algorithms used for edge detection in visual scenes are of the local type. The reason for this fact may lie in the sequential way in which computers (and of course the computer programmers) operate. The present paper is directed at the discussion of local methods.

Local methods of edge detection in visual scenes are analogous to looking at a scene through a small window. The portion of the scene which is seen through the window is analyzed and information about the edge(s) which are present in the view are stored. The window is then moved to another area of the scene and the process is repeated until all edges in the scene are detected.

Two types of local methods are presented. The two-pass method consists of operating on the digitized picture array with a local operator which accentuates or detects the presence of edges in the window. A second picture array is formed by this process which contains only the edges which have been detected by the local operator. The new picture array is then used to detect the edges in the scene. The one-pass method locates an edge in the window and moves the window along the detected edge. This is an edge follower technique, and does not require the intermediate picture array storage of the two-pass method. The following sections present two pass and one pass techniques, together with examples illustrating these methods.

3. Two-Pass Methods

The popular local operators consist of an n by n window on a digitized scene, where n is an integer ≥ 2. Given a two dimensional matrix of intensities, S, the n by n window is shifted along S and a differential matrix, D, is computed. This is a transformation of the intensity domain into the differential domain. Some two by two operators are shown below:

$$(1) \quad d_{i,j} = \sqrt{(s_{i,j} - s_{i+1,j+1})^2 + (s_{i+1,j} - s_{i,j+1})^2}$$

in Roberts [14]

(2) $d_{i,j} = |s_{i,j} - s_{i+1,j+1}| + |s_{i+1,j} - s_{i,j+1}|$

in Forsen [15]

(3) $d_{i,j} = \frac{1}{2} \{ |s_{i,j} - s_{i+1,j} + s_{i,j+1} - s_{i+1,j+1}|$

$+ |s_{i,j} + s_{i+1,j} - s_{i,j+1} - s_{i+1,j+1}| \}$

in Ejiri [5]

(4) $d_{i,j} = |s_{i,j} - s_{i+1,j}| + |s_{i,j} - s_{i,j+1}|$

in Brice [16].

These operators require that a distinct change in intensity must occur between two adjacent points in the intensity matrix of the visual scene. This is not a practical restraint for most image input devices or objects in visual scenes which are to be analyzed. When a visual image is digitized by an input device, random noise from the input device along with bluring and pincushion effects of the lens system tend to form gradual changes in intensity across boundary points of the edges in the visual scene. Therefore, if the maximum resolution and sensitivity of the image input device is to be obtained, the two by two window will not provide sufficient information to detect edges optimally.

More important as a limiting factor of the two by two window are the types of edges which must be detected. Only very sharp edges with high intensity contrast between the surfaces which form the edges will produce a single step response of intensity. The detection of ill defined edges (edges which are formed by a gradual change in intensity across the edge) requires that the area adjacent to both sides of an edge be examined to determine the contrast which defines the edge. Therefore, larger windows will be required and the local operator must consider more than four points in the visual scene to detect the presence of edges. Several three by three operators are given below:

(5) The Laplacian

$$d_{i,j} = [(s_{i,j} - s_{i-1,j}) - (s_{i+1,j} - s_{i,j})]$$

$$+ [(s_{i,j} - s_{i,j-1}) - (s_{i,j+1} - s_{i,j})]$$

in Rosenfeld [17]

(6) $$d_{i,j} = |(s_{i-1,j+1} + 2s_{i,j+1} + s_{i+1,j+1})$$

$$- (s_{i-1,j-1} + 2s_{i,j-1} + s_{i+1,j-1})|$$

$$+ |(s_{i+1,j-1} + 2s_{i+1,j} + s_{i+1,j+1})$$

$$- (s_{i-1,j-1} + 2s_{i-1,j} + s_{i-1,j+1})|$$

in Duda [18]

(7) $$d_{i,j} = |(s_{i-1,j+1} + s_{i,j+1} + s_{i+1,j+1})$$

$$- (s_{i-1,j-1} + s_{i,j-1} + s_{i+1,j-1})|$$

$$+ |(s_{i-1,j-1} + s_{i-1,j} + s_{i-1,j+1})$$

$$- (s_{i+1,j-1} + s_{i+1,j} + s_{i+1,j+1})|$$

in Holmes [19]

(8)

$$d_{i,j} = \begin{bmatrix} s_{i-1,j-1} & s_{i-1,j} & s_{i-1,j+1} \\ s_{i,j-1} & s_{i,j} & s_{i,j+1} \\ s_{i+1,j-1} & s_{i+1,j} & s_{i+1,j+1} \end{bmatrix} * \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$\text{where } A * B = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{i,j} \times b_{j,i}$$

in Sutro [7]

The operator given by (8) is called convolving the digital image with a filter of mask. Prewitt [20] uses this type of operation with three by three and four by four element masks to extract the gradients in blood-cells. Each of the operators given by (1) - (7) may be expressed in terms of the convolution operation.

Otto [21] uses a five by five operator and looks for the maximum change in intensity in eight directions from the point (i, j), as shown in equation (9).

(9) $$d_{i,j} = \max_{m,n} |(s_{i+m,j+n} - s_{i,j})|$$

where $(m,n) = \{(0,2),(-2,2),(-2,0),(-2,-2),(0,-2),(2,-2),(2,0),(2,2)\}$

Hueckel [22] operates on the elements within a circular window. The window will contain 32, 52, 69, 88, or 137 elements of the picture array, depending on the diameter of the window. A low pass filter is applied to these elements which produces the 8 lower order Fourier components of wavelengths in terms of the window diameter. These eight components are then used to determine the best fit of an idealized edge (unit step intensity) to the input intensity information within the window. The operator selects the best values of (c, s, ρ, d, b) which describe a straight edge that passes through the circular window. (c, s, ρ) are the parameters of the normal equation $cx + sy = \rho$. d is the starting value and $d+b$ is the terminating value of the idealized edge intensity.

An adaptive local operator is presented by Rosenfeld [23, 24]. As a test for the presence of a vertical edge through the point (i, j), the difference of average intensities of two 2^h by 2^h windows adjacent to the edge i~ computed for various values of h.

$$(10) \quad d_{i,j}^{h} = 1/2^{2h} \left\{ \sum_{u=0}^{2^h-1} \sum_{v=0}^{2^h-1} s_{(i-2^{h-1}+u),(j-v)} - \sum_{u=0}^{2^h-1} \sum_{v=0}^{2^h-1} s_{(i-2^{h-1}+u),(j+v)} \right\}$$

The smallest window such that

$$3/4 \; |d_{i,j}^{h}| > |d_{i,j}^{h+1}|$$

is chosen as the optimal window size. This procedure has the advantage of being able to adapt to a large variety of edges which appear in a visual scene. It also detects surfaces of different textures which meet to form an edge. The operators described above determine the differential intensity matrix. In general, different operators will yield different differential intensity matrices. The use of these operators on visual scenes is illustrated in the next section. It may be emphasized that the solutions are not always correct or unique. In general the solutions will depend upon the order in which the algorithm is applied or the various parameters which are used. The type of algorithm which is used is generally determined by the type of visual scene to be processed.

The differential intensity matrix D, obtained by the use of local operators, consists of a matrix of numbers. The second pass of the two-pass method consists of selecting a threshold λ, and fitting straight or curved line segments to all of the points in D such that $|d_{i,j}| > \lambda$. A histogram of the values of D will give some initial indication of what value of λ can be used. Most methods of edge fitting require finding the location of connected (adjacent) points in D which are greater than λ and fitting the best line or curve connecting these points by any of the following techniques:

(a) Forming short straight line segments from connected points and regrouping the connected segments of approximately the same slope into longer straight line segments [14].

(b) Fitting a least squares straight line or a polynomial on a subset of points of D [18].

(c) Finding the maximum value of the convolution of the points in D with a template [18].

(d) Thining of local areas of connected points in D to form a simple line [25].

(e) Performing a Hough transformation [26].

Once the edges in D have been located, they can be represented in several ways. Straight edges can be characterized by their end points. Curved edges can be represented by a sequence of straight edges, or the coefficients of a polynomial, or chain encoding (a polynomial representation of the changes in slopes along the curve), or a set of templates. Certain descriptions may be more desirable in a given application. For example, the description by the coefficients of the Fourier transformation of the angle along the perimeter of a simple closed curve will produce a position and rotation invariant description [27].

4. Edge Detection Examples

A black and white pencil drawing (Figure 1) was digitized into a 256 by 256 matrix of intensities S. Figure 2 shows an eight gray-level computer printout of the S matrix. Curve A of Figure 3 shows the histogram of S, $h_s(x)$, with 0 being the darkest and 63 the brightest intensities in S. The S matrix was operated upon by the eight differential operators that were given in Equations (1) through (8) in the previous section to form the differential matrix D. Six of the operators produced similar results and are shown in Figures 4 through 9. The operators given in Equations (5) and (8) did not produce usable differential matrices. These two operators produced excessive random noise of unconnected points in D and did not produce well defined edges. These results do not produce any criticism or evaluation of the specific operators. Only the results from the specific visual image and intensity digitization can be observed. Another type of visual scene or digitizing method may have produced different results.

The operator (1) produced a differential matrix $D_{(1)}$, for which the histogram, $h_d(x)$, is shown by curve B of Figure 3. The values of $d_{i,j}$ ranged from 0 to 78. Figure 4 shows an eight gray-level printout of $D_{(1)}$. The seven printout thresholds for the D matrices were choosen by the use of the relationship

$$g(x) = \log_{10} \sum_{i=x}^{n} h_d(i)$$

where $h_d(x)$ is the histogram of D. The thresholds were (0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2) of the maximum value of g(x). Curve C shows g(x) for $D_{(1)}$. The eight operators are summarized below.

Equation No.	max $(d_{i,j})$	Figure No.	Minimum g(x) Threshold
1	78	4	7
2	111	5	9
3	58	6	6
4	102	7	7
5	151	-	8
6	312	8	49
7	209	9	29
8	297	-	20

5. One-Pass Methods

The two pass methods discussed in Section 3 required storage of a large amount of intensity data. The edges to be detected in a visual scene are characterized by the largest intensity change occuring along lines which are perpendicular to the edge. Therefore, the largest amount of information about the presence of an edge from the least amount of data can possibly be obtained from such single line scans. In the following discussions, $d(x)$ is defined as the differential along the vector $s(x)$ of intensities. The one pass approach lends itself best to systems which require the least amount of computer data storage. It is used mainly in systems which use a sequential scanning device, such as a vidicon television camera, or a random scanning device, such as an image dissector television camera or a random flying spot scanner. Only the points in the visual scene where the edges are detected are stored in the computer. This produces a large reduction of storage requirement. Picture reduction techniques discussed in [28] use similar techniques for visual storage and image transmission.

Sutro [21] used a vidicon television camera which first sweeps the raster with horizontal scans and then repeats with vertical scans. The video signal is digitized and processed by a digital filter, which detects the edge gradients. The horizontal and vertical gradients are then merged to form the visual image. The random position image devices produce the capability of scanning along a line (not necessarily straight) anywhere in the visual raster. Only those intensities along the scan line must be digitized and a minimum amount of computer intensity data storage is required.

One pass methods differ from two pass methods in that (i) the one pass method does not store the differential intensity matrix, (ii) the edge fitting is performed as one determines the edge. There is no separate edge fitting operation. Thus the one pass methods are also called edge-follower algorithms.

Greenblatt [29] used perpendicular scans along the proposed straight edge. The coordinates, where the maximum $[s(x) - s(x-1)]$ occured along each perpendicular scan are stored and a narrow rectangular box is proposed to enclose these coordinate points. The rectangular box is lengthened and more perpendicular scans are produced until the box cannot enclose the coordinates. A straight edge in the visual scene is thereby defined by the rectangular box.

Herskovits [30] discusses methods of detecting the edges of objects which are viewed with an image dissector television camera. Δ is a range of intensity points along a straight line scan which are examined before and after the location x in the intensity vector for the presence of an edge. If first difference is

$$(11) \quad d_1(x) = s(x+\Delta) - s(x-\Delta) ,$$

then the average of $d_1(x)$ over an interval of 2Δ is

$$(12)\quad \bar{d}_1(x) = 1/2\Delta \sum_{i=-\Delta}^{+\Delta} d_1(x+i) ,$$

and $\bar{d}_1(x)$ may be used to detect edges. Further, the image dissector produces random noise which is assumed to be Gaussian, with a standard deviation of σ. Assuming that an ideal step function of intensity is being detected, $\bar{d}_1(x)$ will detect the step function if it has an amplitude of greater than $\sigma/\sqrt{\Delta}$. A constant slow change of intensity, such as a blurred edge, with a slope of α can be detected by $\bar{d}_1(x)$ if the intensity changes by more than $2\Delta\alpha$. As a compromise between these two values of detectable intensity change, the optimal value of Δ was found to be $\Delta^{3/2} \approx \sigma/2\alpha$.

Another type of operator was found to be more satisfactory with sloping types of edges, the most common type, based on the second difference

$$(13)\quad d_2(x) = s(x+\Delta) - 2s(x) + s(x-\Delta).$$

The difference between the mean of $d_2(x)$ over Δ points after and before x is given by

$$(14)\quad \bar{d}_2^*(x) = 1/\Delta \left\{ \sum_{i=1}^{\Delta} d_2(x+i) - \sum_{i=1}^{\Delta} d_2(x-i) \right\} .$$

Here $\bar{d}_2^*$ will detect intensity changes of greater than $\sqrt{3/\Delta}\,\sigma$ and sloping changes caused by defocusing have no appreciable effect on this value.

Underwood [31] used the first difference average $\bar{d}_1(x)$ of equation (12) as an edge operator with $\Delta = 2$. The edges in the visual scene were traced by performing perpendicular scans along a proposed edge and storing the coordinates of the closest peak in $|\bar{d}_i(x)|$ to the midpoint of the scan for each perpendicular. An additional operation of performing circular scans about an edge point was performed. The purpose of the circular scan was to determine the direction of an edge and if another edge is present in the neighborhood which interesects with the present edge which is being followed. The close proximity of edges near an object corner causes the value of $\bar{d}_1(x)$ to not approach zero when two edges are near each other. Instead, local peaks and valleys in $|\bar{d}_1(x)|$ occur. Therefore, a peak at location x was determined to indicate an edge iff:

$$(15) \quad |\bar{d}_1(x)| - |\bar{d}_1(x^{-a}_{+b})| > \frac{1}{n} \sum_{i=1}^{n} |\bar{d}_1(i)|$$

where a is distance to the nearest valley clockwise of x, b is distance to the nearest valley counter-clockwise of x, and n is the number of sample points. The use of the average value of $|\bar{d}_1(x)|$ as a threshold value automatically compensates for different contrast situations which occur in the visual scene.

6. Example of Edge-following

The edge follower reported in [31] was designed to form line drawing descriptions of three-dimensional planar objects which were viewed by a random access Image Dissector television camera. The routine determines all of the nodes of the object under view and returns a line drawing description in terms of the node (end point) coordinates of each visible edge of the object. An initial edge of the object is found by scanning across the scene and locating the maximum peak in $\bar{d}_1(x)$. A circular scan finds the direction of this edge. Scans along perpendiculars to the edge determine the coordinates of points along the object edge. Nodes of the object are found by periodically performing circular scans to determine the presence of other edges which intersect to form nodes. A followed edge can only be terminated by such an intersection. Once a node has been determined, a circular scan about the node will locate the other edges of the object. Figure 10 shows the computer plotter output of an icosahedron which was viewed by the system. The line drawing description output is used to perform automatic learning and recognition of planar three-dimensional objects.

A modification of the routine has added the ability to follow the curved edges of non-planar objects. Instead of describing an object edge as a straight line between two node coordinates, an edge becomes a string of short straight edge segments between two nodes. The presence of intersecting edges to form an edge end must now take into account not only the direction of the last edge segment, but the curvature which is formed by the preceeding edge segments. Figure 11 shows the plotter output of a cup which was viewed.

7. Bibliography

[1] G. Nagy, "Current Problems in Character Recognition," Proceedings of the Purdue Symposium on Information Processing, April 1969.

[2] John H. Munson, " Experiments in the Recognition of Hand-Printed Text: Part I - Character Recognition," Proceedings of the Fall Joint Computer Conference, 1968.

[3] Patrick H. Winston, "Heterarchy in the M.I.T. Robot," Flash #8, Vision Group, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

[4] J. A. Feldman, G. M. Feldman, G. Falk, G. Grape, J. Pearlman, I. Sobel, J. M. Tenebaum, "The Stanford Hand-Eye Project," Proceedings of the International Joint Conference on Artificial Intelligence, Washington, D. C., May 1969.

[5] Masakazu Ejiri, Takeshi Uno, Haruo Yoda, Tatsuo Goto, Kiyoo Takeyasu, "A Protype Intelligent Robot that Assembles Objects from Plan Drawings," IEEE Trans. on Computers, Vol. C-21, No. 2, February, 1972.

[6] Nils J. Nilsson, "A MobilAutomaton: An Application of Artificial Intelligence Techniques," Proceedings of the International Joint Conference on Artificial Intelligence, Washington, D. C., May 1969.

[7] Louis L. Sutro, William L. Kilmer, "Assembly of Computers to Command and Control a Robot," Proceedings of the Spring Joint Computer Conference, 1969.

[8] Douglas A. O'Handley, "Recent Developments in Digital Image Processing at the Image Processing Laboratory of JPL," paper presented at the First National Conference on Remotely Manned Systems (RMS), California Institute of Technology, Pasadena, California, 13-15, September 1972.

[9] T. Caspersson, K. R. Castleman, G. Lomakka, E. J. Modest, Aa. Moller, R. Nathan, R. J. Wall, L. Zech, "Automatic Karyotyping of Quinacrine Mustard Stained Human Chromosomes," Exptl Cell Res 67.

[10] S. J. Dwyer, III, C. A. Harlow, D. A. Ausherman, G. S. Lodwick, "Computer Diagnosis of Radiographic Images," Proceedings of the Spring Joint Computer Conference, 1972.

[11] "Remote Multispectral Sensing in Agriculture," Laboratory for Agricultural Remote Sensing, Annual Report, Vol. 4, Research Bulletin 873, Purdue University, Lafayette, Indiana, December, 1970.

[12] Masahiko Yachida, "Application of Color Information to Visual Perception," Pattern Recognition, Pergamon Press, Vol. 3, 1971.

[13] M. L. Minsky, S. Papert, "Perceptrons, an Introduction to Computational Geometry," M.I.T. Press, Cambridge, Mass., 1969.

[14] L. G. Roberts, "Machine Perception of Three-Dimensional Solids," in Optical and Electro-Optical Processing of Information, M.I.T. Press, Cambridge, Massachusetts, 1965.

[15] G. E. Forsen, "Processing Visual Data with an Automaton Eye," in Pictorial Pattern Recognition, Thompson, Washington, D. C., 1968.

[16] Claude R. Brice, Claude L. Fennema, "Scene Analysis Using Regions," Artificial Intelligence 1, American Elsevier Publishing Co., Inc., 1970.

[17] Azriel Rosenfeld, "Picture Processing by Computer," Academic Press, New York, N.Y., 1969.

[18] Richard O. Duda, Peter E. Hart, "Pattern Classification and Scene Analysis," in press, 1971, Wiley.

[19] W. S. Holmes, H. R. Leland, G. E. Richmond, "Design of a Photointerpretation Automaton," Proceedings of the Fall Joint Computer Conference, 1962.

[20] Judith M. S. Prewitt, "Object Enhancement and Extraction," in Picture Processing and Psychopictorics, Academic Press, New York, N. Y., 1970.

[21] Jacqueline L. Otto, Charles A. Harlow, "EDGE5--Line Identification Algorithm," Technical Report, Image Analysis Laboratory, Dept. of Electrical Engineering, University of Missouri - Columbia, Columbia, Missouri, March 1971.

[22] Manfred H. Hueckel, "An Operator Which Locates Edges in Digitized Pictures," Jour. ACM, Vol. 18, No. 1, January 1971.

[23] A. Rosenfeld, M. Thurston, "Edge and Curve Detection for Visual Scene Analysis," IEEE Trans. on Computers, Vol. C-20, May 1971.

[24] A. Rosenfeld, M. Thurston, Y. H. Lee, "Edge and Curve Detection: Further Experiments," IEEE Trans. on Computers, Vol. C-21, July 1972.

[25] R. Stefanelli, A. Rosenfeld, "Some Parallel Thinning Algorithms for Digital Pictures," Jour. ACM, Vol. 18, No. 2, April 1971.

[26] Richard O. Duda, Peter E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," Communications ACM, Vol. 15, No. 1, January 1972.

[27] C. T. Zahn, R. Z. Roskeid, "Fourier Descriptors for Plane Closed Curves," IEEE Trans. on Computers, Vol. C-21, March 1972.

[28] L. C. Wilkins, P. A. Wentz, "Bibliography on Data Compression, Picture Properties and Picture Coding," IEEE Trans. on Information Theory, Vol. IT-17, March 1971.

[29] Richard Greenblatt, Jack Holloway, "SIDES21," Massachusetts Institute of Technology, Project MAC, Artificial Intelligence Project, Vision Memo No. 101, Memorandum MAC-M-320, August 1, 1966.

[30] Annette Herskovits, Thomas O. Binford, "On Boundary Detection," Massachusetts Institute of Technology, Project MAC, Artificial Intelligence Project, Memo No. 183, July 1970.

[31] Stephen A. Underwood, "Visual Learning and Recognition by Computer," Ph.D. Dissertation, The University of Texas at Austin, May 1972, also as S. A. Underwood, C. L. Coates, Technical Report No. 123, Electronics Research Center, The University of Texas at Austin, Austin, Texas, April 6, 1972.

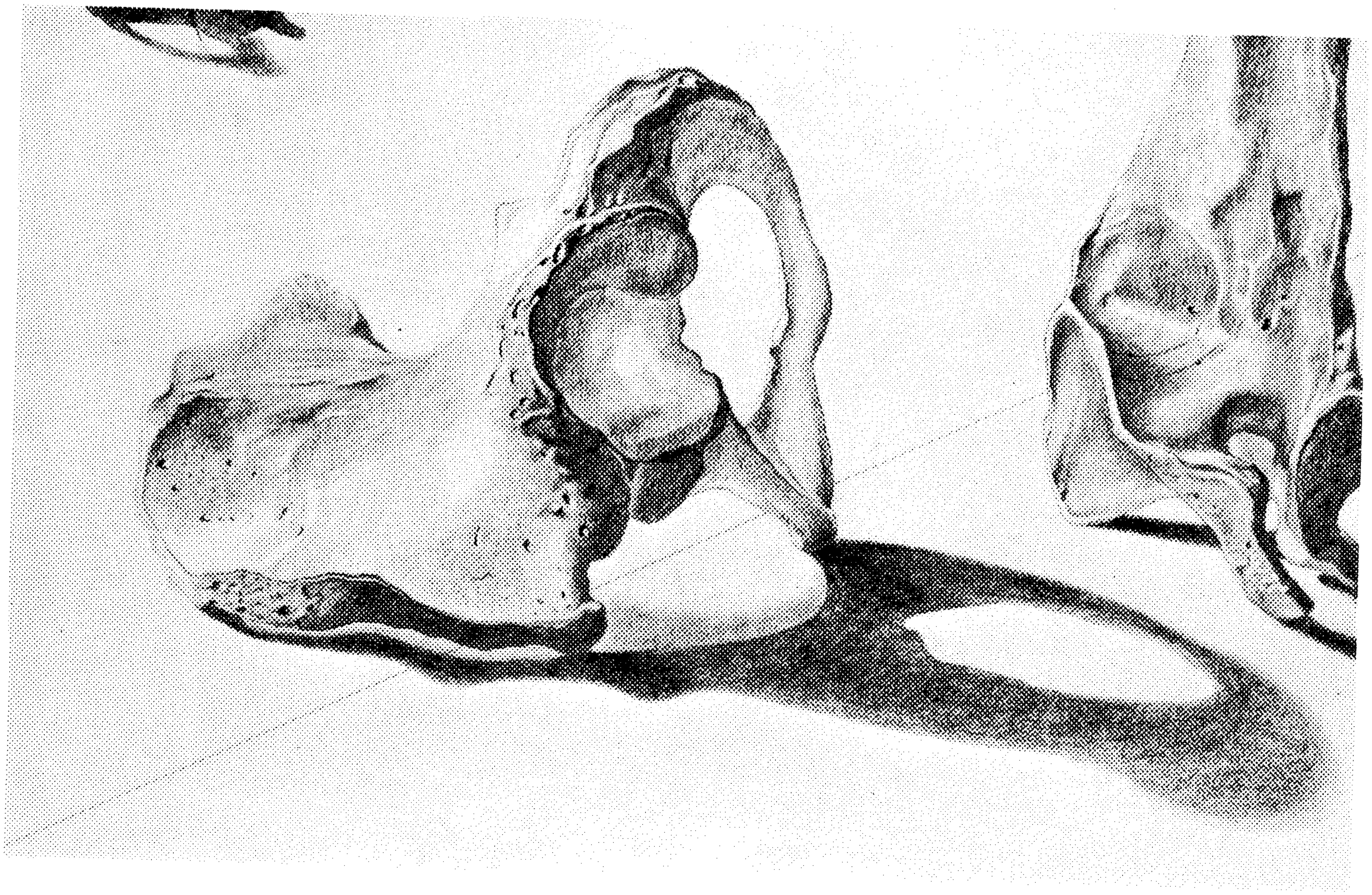

Figure 1. Original Scene

Figure 2. Intensity Matrix S

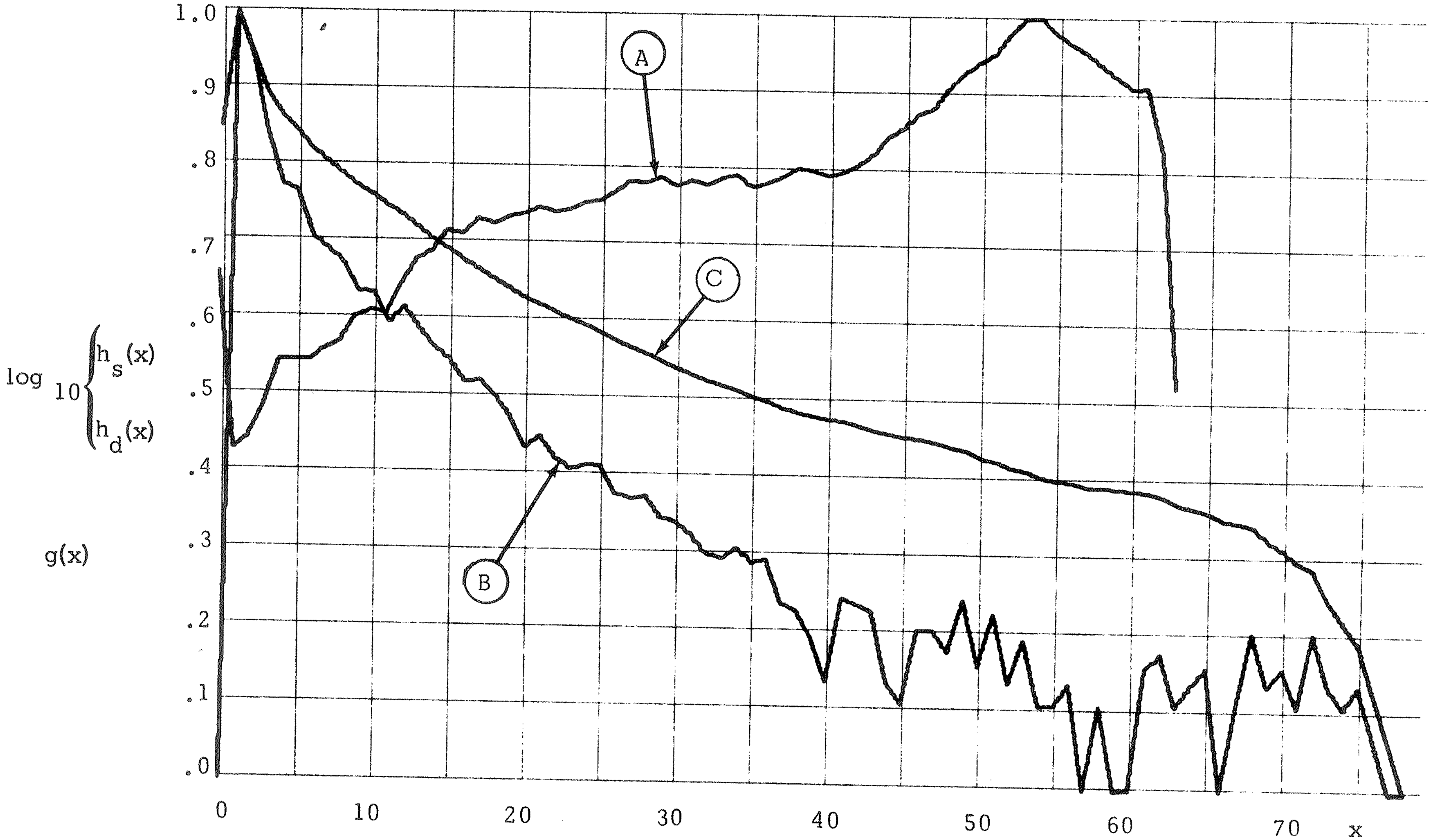

Figure 3. Distribution of S and $D_{(1)}$

Figure 4. $D_{(1)}$ Matrix

2

Figure 5. $D_{(2)}$ Matrix

3

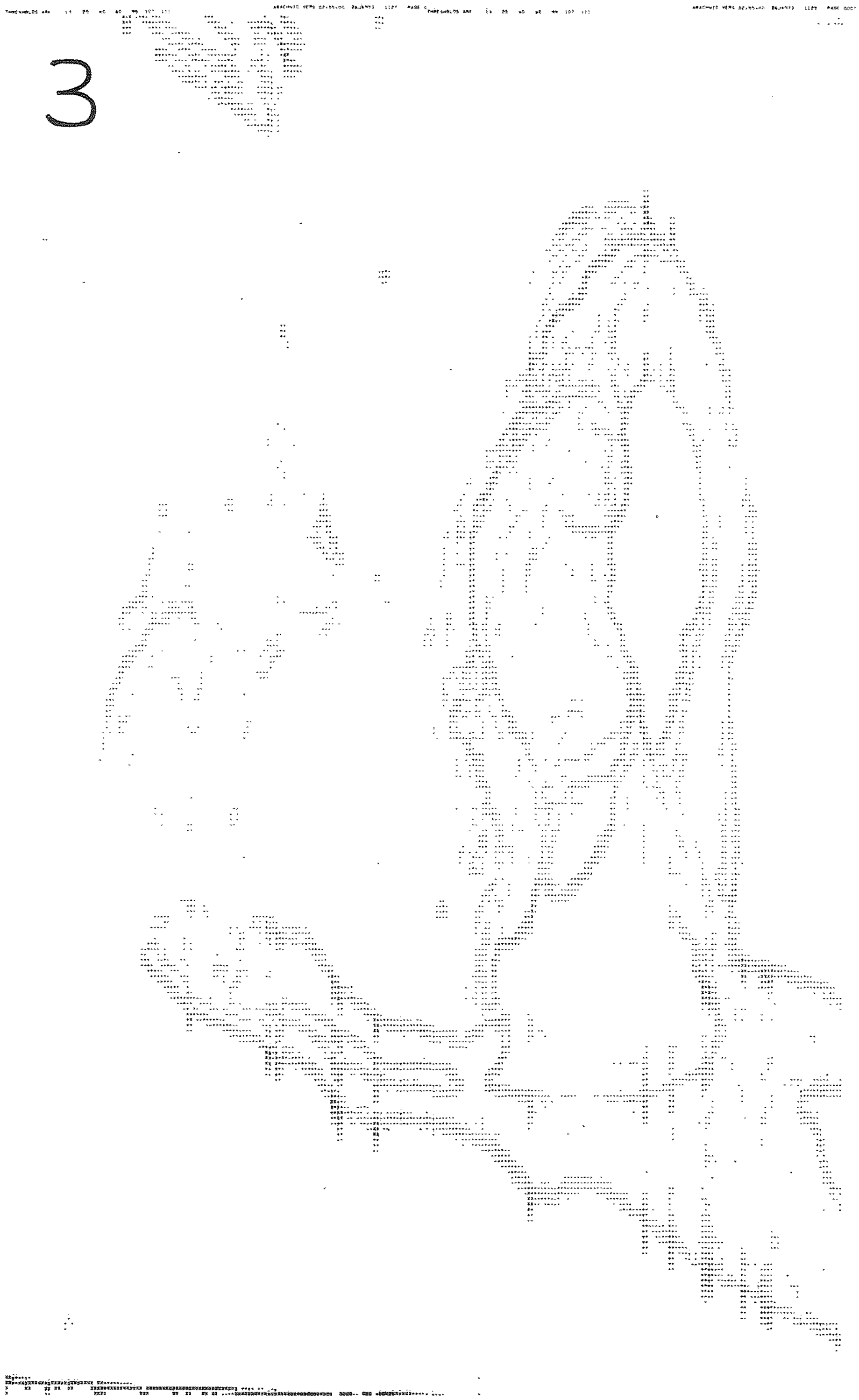

Figure 6. $D_{(3)}$ Matrix

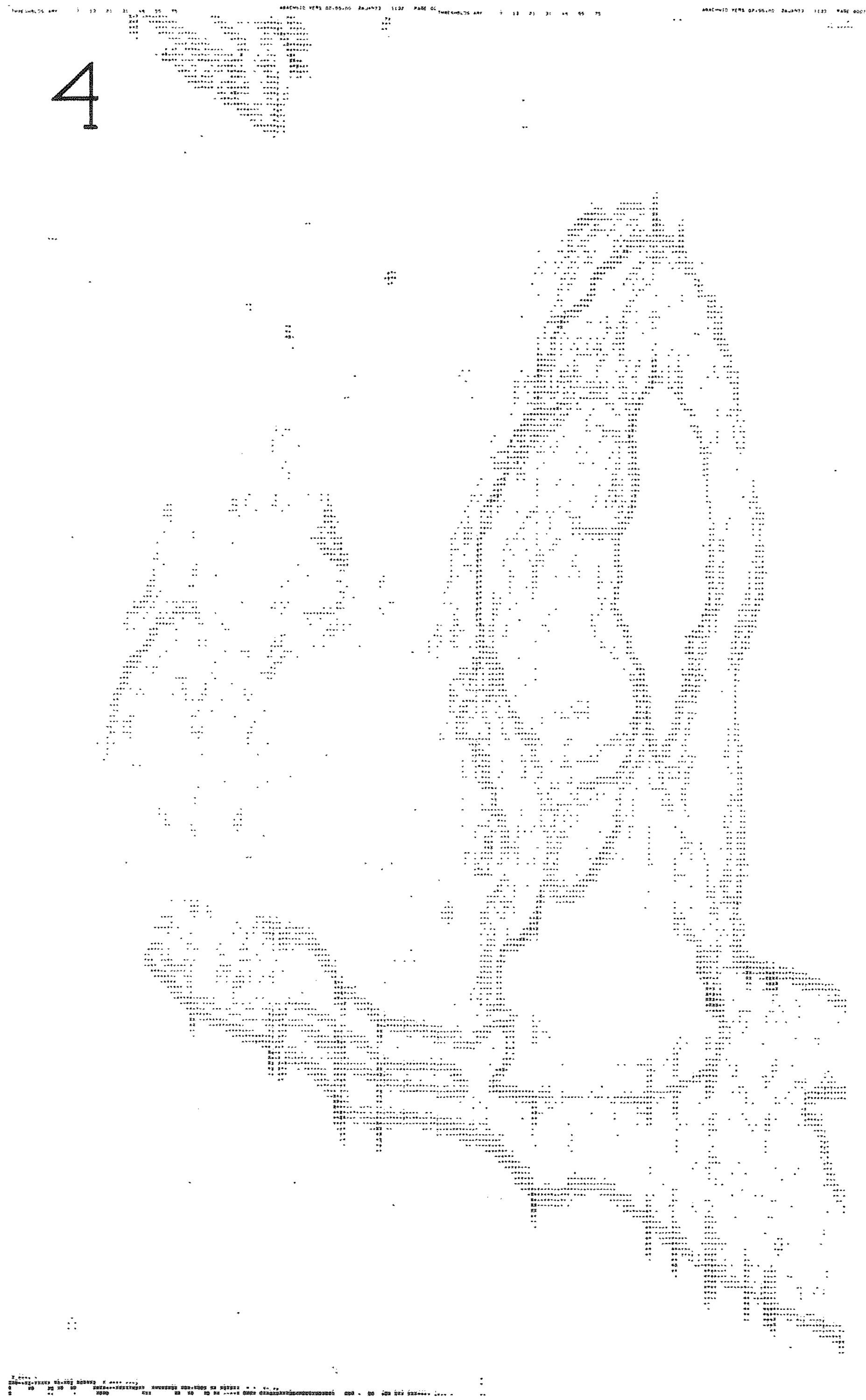

Figure 7. $D_{(4)}$ Matrix

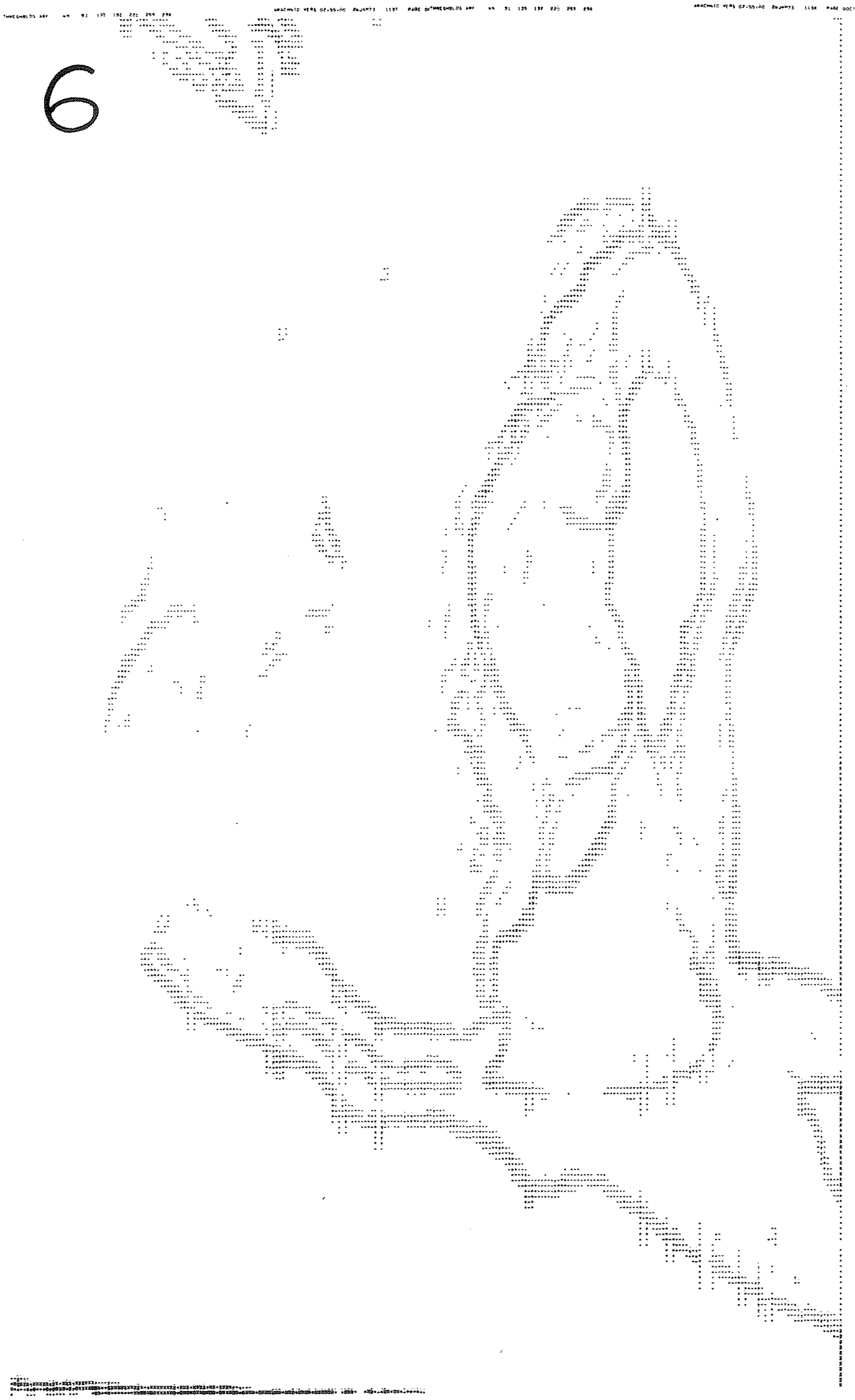

Figure 8. $D_{(6)}$ Matrix

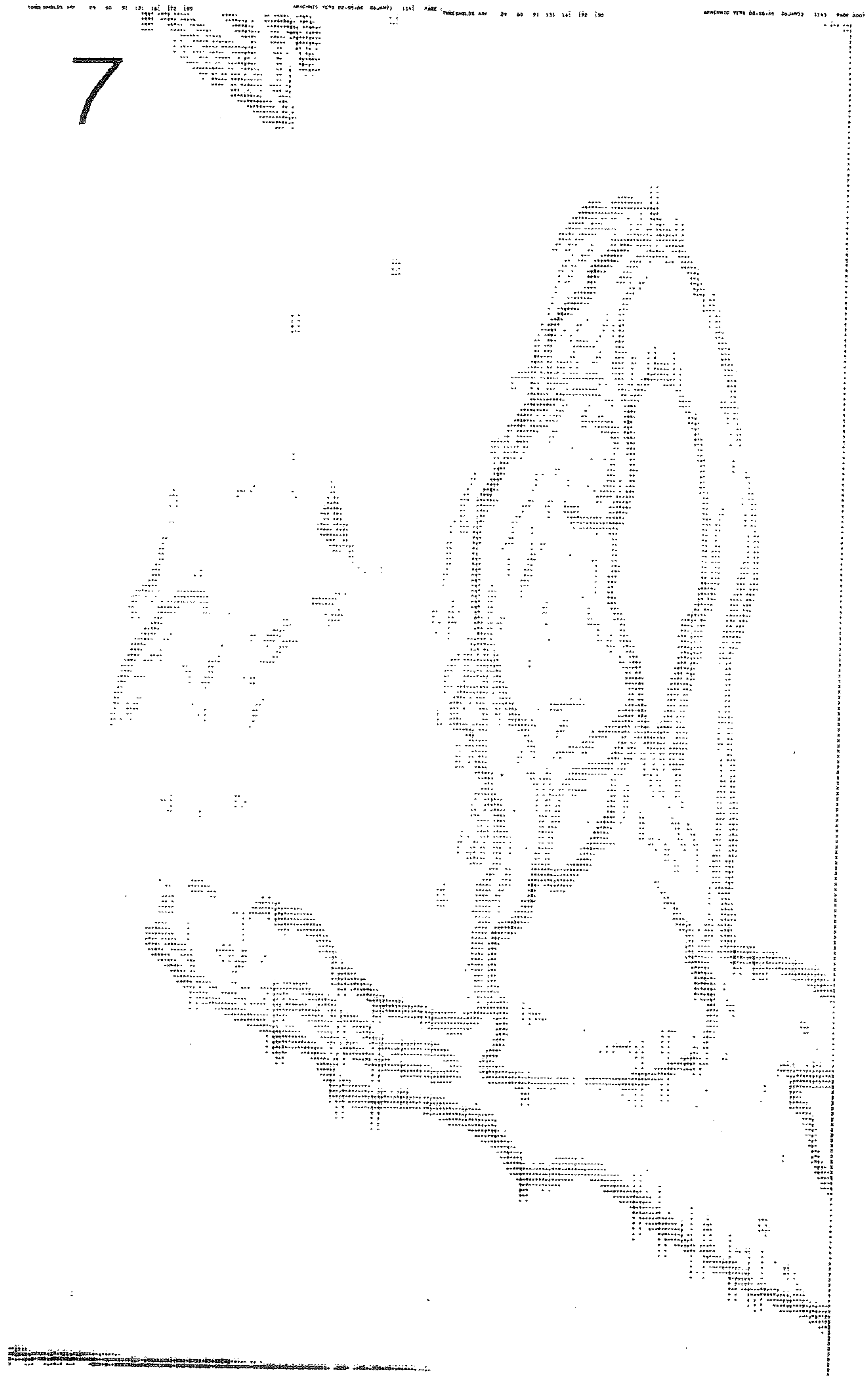

Figure 9. $D_{(7)}$ Matrix

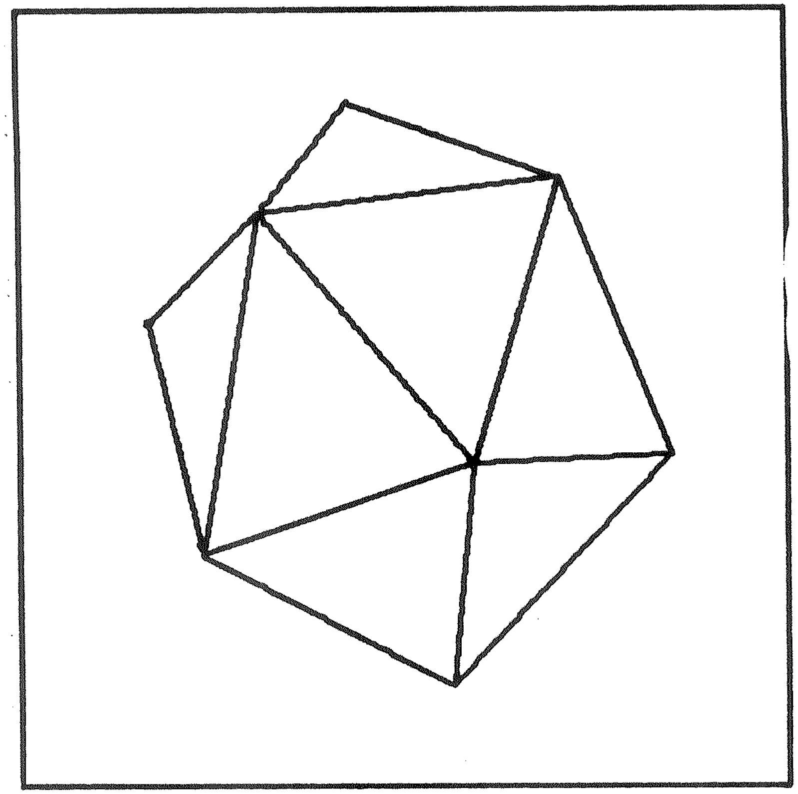

Figure 10. Isosahedron

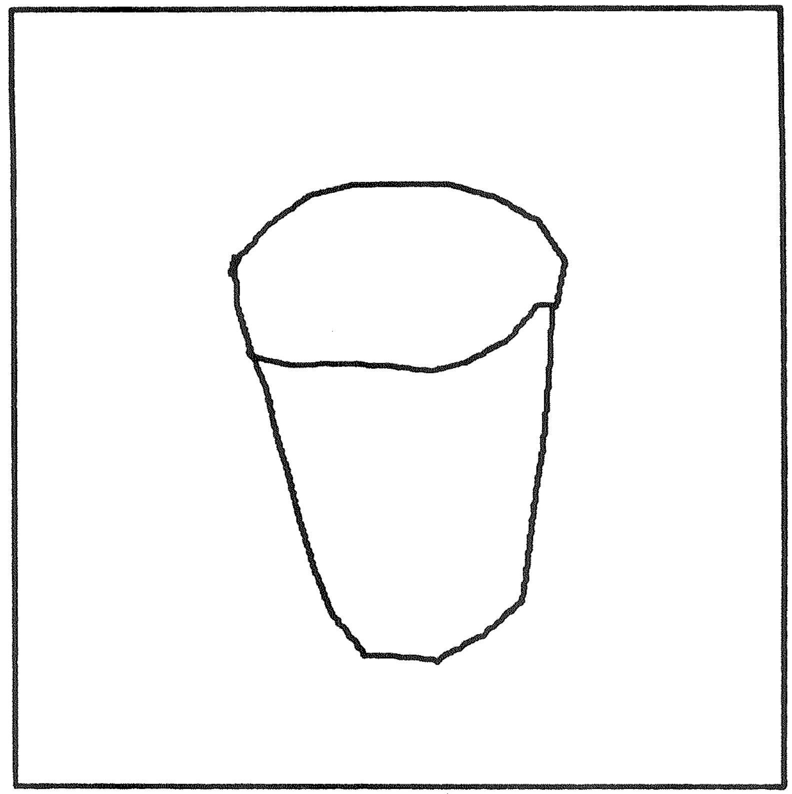

Figure 11. Cup