OBJECT EXTRACTION AND IDENTIFICATION IN PICTURE PROCESSING

By

Peter Pei-teh Lin

A Dissertation Presented to the Graduate Council of
the University of Florida
in Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy

UNIVERSITY OF FLORIDA

1972

To my parents.

ACKNOWLEDGEMENTS

This work was supported in part by the National Science Foundation Grant GK-2786, the Office of Naval Research Contract N00014-68-A-0173-0001, the Center for Informatics Research, and the Graduate School of the University of Florida. This financial assistance is gratefully acknowledged.

The author would like to thank the members of his supervisory committee and departmental faculty representatives for the fine job they did in reading and therefore improving the presentation made in this dissertation. Dr. Raymond L. Hackett from the Medical School should be acknowledged for the information on medical aspects.

For his time and effort in directing the research, the author would especially like to thank Dr. Julius T. Tou. Without his stimulating ideas, his invaluable suggestions, his honest criticism, and his enlightening guidance, this dissertation would not have been possible. The author's utmost thanks must go to him.

Miss Jean Roman should be acknowledged for proofreading.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

Abstract of Dissertation Presented to the Graduate Council of the University of Florida in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

OBJECT EXTRACTION AND IDENTIFICATION IN PICTURE PROCESSING

By

Peter Pei-teh Lin

March, 1972

Chairman: Dr. Julius T. Tou
Major Department: Electrical Engineering

Any computerized picture processing system can generally be divided into four major units: a picture digitizer, an object extractor, a feature extractor and a classifier. This dissertation is concerned mainly with new approaches to object extraction and feature extraction.

Three information handling methods have been developed which may be used to mechanize the extraction of objects from multi-level pictures. These methods are those of the gradient analysis, the contour analysis and the graph theory approach. In the gradient analysis method, a locally optimal threshold is used to find the boundary points. In this new approach, high efficiency is achieved because the tentative boundary paths are simultaneously found. Then, after filling the gaps along the boundaries or at the intersections and removing the tails, boundary segments are found. Rules are

set to combine the boundary segments in order to decompose the overlapping, self-folding and touching objects in an area picture. (The contour analysis method is developed on the assumption that the threshold used for transfering a multi-level picture to a binary picture is approximately constant in a window.) This method permits very successful object extraction for a multi-level picture with selective deletion of nonboundary contours. In the graph theory approach, a multi-level picture is transfered to a weighted graph. An MSF (Minimal Spanning Forest) of the weighted graph is then found. By finding the principal paths of a tree in the MSF, the skeleton of the object corresponding to this tree can then be found.

The boundary contour of an object has been encoded by a sequence of octal chain codes. A local feature extractor has been designed to find the area, centroid, shape, principal axis direction and the elongation index of an object with the knowledge of the sequence of octal chain codes of the boundary contour. A global feature extractor has been designed to find the inclusion relationship among objects and the distribution of objects in a picture. The inclusion relationship is represented by a Hasse graph. The distribution of objects may be represented by an MST (Minimal Spanning Tree).

The newly designed object extractor and the feature extractor methods have been tested by analysis of the

information in the pictures of chromosomes, skin cells and blood cells. In evaluation of chromosome pictures, the major tasks are to identify and catagorize all chromosomes. In analysis of the histological skin cell photomicrographs, the problem is to find the structure of cells in epidermis in order to detect the degree of the malignancy of possible tumors. In evaluation of the blood cell photomicrographs, the goal is to obtain the histogram of the blood cell photo-intensities in order to reveal critical diagnostic information. In each of these three evaluation tests very promising results were achieved by the use of combinations of the new techniques. A more complete computerized picture processing system is suggested, as an extension of the newly developed techniques.

CHAPTER I

INTRODUCTION

Picture processing is a process which transfers scenes to descriptions. For example, when a picture processor "sees" a writing "日", it should be able to tell that it is the Chinese character for "sun." When a picture processor "sees" the scene as shown in Figure 1.1, it should tell that there are two cubes, A and B, in the scene, where cube A is in front of cube B.

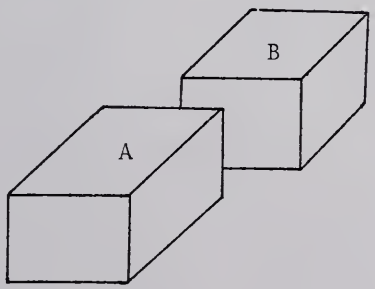


Figure 1.1. A scene of two cubes.

The processor performs two main functions: the first is "to see" and the second is "to give the descriptions"

from what is "seen." "To see" is the process usually called object extraction from the scene. "To give the descriptions" from what has been "seen" includes feature extraction and identification. In general, human beings are the best picture processors up to the present. One drawback of man's ability as a picture processor is that his visual system is easily tired. Mechanization of the picture processing became possible after the invention of the modern computer. This mechanization is very desirable as it frees manpower from routine visual tasks.

There are two principal types of pictures encountered in everyday life. One is the picture of three-dimensional objects. This type of picture is the projection of the three-dimensional object on a picture plane. The projection is supposed to exhibit the depth information. Several researchers[1-3] have conducted research dealing with this type of three-dimensional picture. The other type of picture is two-dimensional. Two-dimensional pictures are either artificial pictures, such as characters[4,5] and maps,[6] or natural images whose depth information is not important and almost cannot be seen in the picture planes, such as pictures of particle tracks in the bubble chamber,[7,8] fingerprints[9] and cell images.[10,11] From here on, "picture processing" means the mechanization of picture processor unless otherwise specified.

A picture processing system can generally be divided into four parts: a picture digitizer, an object extractor, a feature extractor and a classifier. The picture digitizer and the object extractor perform the function of "seeing." The feature extractor and the classifier perform the function of "giving the descriptions." Figure 1.2 is a block diagram of a general picture processing system.

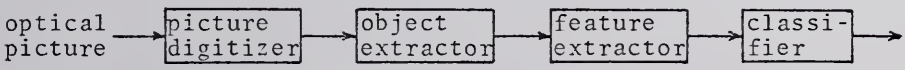


Figure 1.2. A block diagram of a general picture processing system.

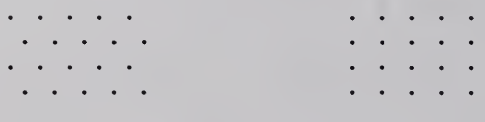
1.1. A Survey of Research in the Area of Picture Processing

A brief survey of the area of picture processing is presented in this section.

1.1.1. Picture Digitizer

A picture digitizer transforms the data of an image to a digitized form which is accessible by a digital computer. An optical picture can be represented mathematically as a real function f on a picture plane D, which is a simply connected subset of the real plane, f: D→R, where R is the set of the intensity values of the picture points. There

are two principal ways to quantize[12] a picture plane: the hexagonal grid and the rectangular grid. Figure 1.3 shows the two types of grids. Hexagonal grids have the advantage of having six neighboring picture points, which are nearest to p, for every picture point p. They have the drawback of being based on an uncommon, non-orthogonal coordinate system. The rectangular grids contain only four neighboring picture points, which are nearest to p, for every picture point p, but it is very easy to access every picture point.



(a) (b)

Figure 1.3. Examples of (a) hexagonal grid and (b) rectangular grid.

The rectangular grid forms an orthogonal coordinate system. Hence the picture plane becomes $I = I_x \times I_y$, where I_x and I_y are subsets of the integer set. From here on, all digitized pictures use rectangular grids. The intensities of a quantized picture are quantized into n levels. Usually n is set equal to 2^k because this maximizes

storage efficiency within the bit-oriented digital computers. A 2^k-level picture is called a k-bit picture. If k = 1, the digitized picture is called a binary picture. An n-level digitized picture is a mapping g: I→N, where N = (0,1,..., n-1) is the set of quantized intensity values. A digitized picture can also be represented as a matrix. The location of a picture point is specified by the location of the element in the matrix. The intensity of a picture point is indicated by the value of the corresponding element in the matrix. Figure 1.4 is a binary picture of a numeral "6" represented as this matrix form.

A picture digitizer performs a transformation from an f mapping to a new mapping g. A complete picture digitizing system, PIDAC (Pictorial Data Acquisition Computer), has been implemented at the CIR (Center for Informatics Research) in the University of Florida. The PIDAC, which is a modification of the FIDAC[13] (Film Input to Digital Automatic Computer) system, is one of the better picture digitizers available today. It consists of a CRT, two lenses, a photomultiplier, an a-d converter and a scan control unit. The digitized pictures are stored on a magnetic tape. The PIDAC can alternately be interfaced with a digital computer to store the digitized form from a picture. The maximum spatial resolution of the PIDAC is 1,240 spots along the long axis and 800 lines per 35 mm film. The maximum digitized level of the PIDAC is 2^4 = 64. Very good 8-level

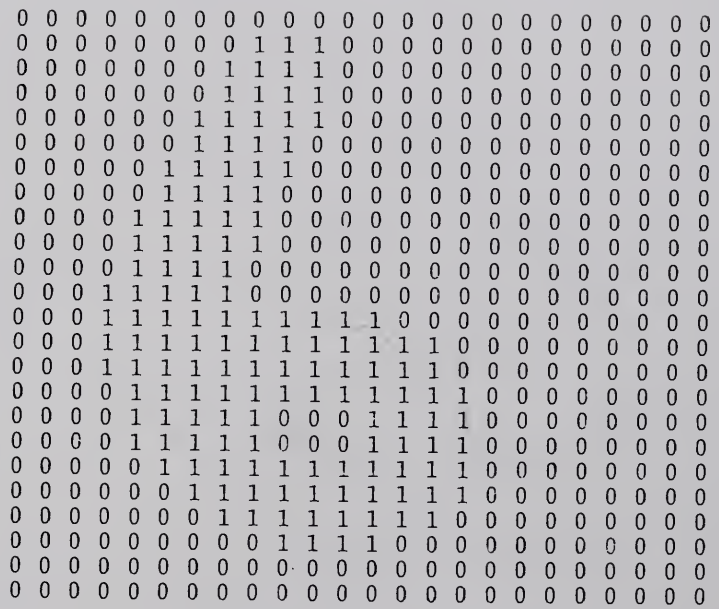


Figure 1.4. A binary picture of numeral "6."

pictures can be achieved by the PIDAC. The scanning speed of the PIDAC is .3 sec/picture. Figure 1.5 is a portion of an 8-level skin cell picture obtained from PIDAC.

Once a digitized picture is obtained, the picture data are then accessible by digital computers. The next process is the extraction of objects from the picture.

1.1.2. Object Extraction

There are mainly three methods used in extracting objects from the scene. The first method finds the boundaries of objects and then decomposes objects from boundaries. The second method finds the thresholds to transfer a multi-level picture to a binary picture and then finds the contours of the binary pictures. The third method finds the clusters in a picture and considers each cluster as an object.

There are two main approaches used to find the boundaries of objects. One approach finds the enhanced picture first and then finds the boundaries. The enhanced pictures can be found either from the spatial domain or from the spatial frequency domain. To find the enhanced pictures directly from the picture plane (spatial domain), the most frequently used methods are the gradient method[1,14] and the Laplacian method.[14] By the gradient method, each picture point in an enhanced picture is set to have a value equal to the gradient of intensity at that picture point.

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

Figure 1.5. A digitized picture of a portion of a skin cell picture.

Since the data are digitized, approximation of gradient is used, such as[1] $h(i,j) = [(g(i,j)-g(i+1,j+1))^2 + (g(i+1,j)- g(i,j+1))^2]^{1/2}$, which is a very good approximation except that square root calculation is involved. Approximation of the Laplacian function is needed for digitized data, such as[14] $L(i,j) = g(i-1,j)+g(i+1,j)+g(i,j-1)+g(i,j+1)-4g(i,j)$. The function L then represents the enhanced picture. The enhanced picture will normally have high values at the boundaries. To find the enhanced picture from the spatial frequency domain,[14] the picture f is transformed to a Fourier spectrum F first. A high-pass filter H is applied to enhance values of F at high frequencies relative to those at low frequencies. The inverse Fourier transformation of FH is the corresponding enhanced picture. After the enhanced picture is found, a threshold is then set to find the boundary points. Boundary points are connected by a multi-step process.[1] The other approach finds the boundaries by use of a matched filter which can extract the boundaries directly from the picture g. The purpose of the edge operator used by Huechel[15] is to fit an ideal edge element to any empirically obtained edge element. In scanning the picture, when an edge is found by the edge operator, scanning is interrupted and the edge is traced until lost.

After the boundaries in a picture are found, objects are to be extracted. Guzman[2] did the work on extracting

three-dimensional objects. The main idea used to extract the three-dimensional objects was based on the a priori knowledge of the possibility of two faces belonging to an object. A vertex is in general a point of intersection of two or more boundaries of regions. A program SEE has been built to examine the configuration of lines meeting at the vertex to obtain evidence relevant to whether the regions involved belong to some object. Two types of links, strong links and weak links, are used. Figure 1.6 shows the linkage of faces at several vertices. A solid line implies a strong link and a dotted line implies a weak link. Figure 1.7 shows the links which are inhibited.

A region is defined as a surface bounded by simply closed curves. A nucleus is a set of regions. Two nuclei, A and B, are linked if the regions a and b are linked where $a \in A$ and $b \in B$. Three rules are set to link the nuclei.

First rule: If two nuclei are linked by two or more strong links, they are merged into a larger nucleus.

Second rule: If nuclei A and B are joined by a strong and a weak link, they are merged into a new nucleus.

Third rule: If nucleus A consists of a single region, has one link with nucleus B and no links with any other nucleus, A and B are merged.

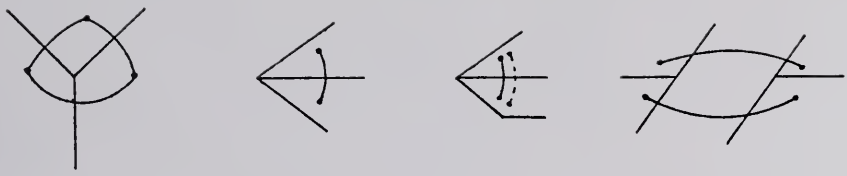


Figure 1.6. Linkage among faces.

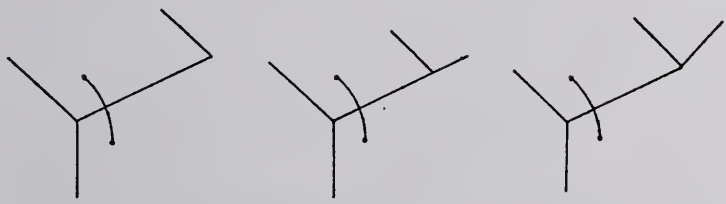


Figure 1.7. Several inhibited links.

The first rule is applied on the picture repeatedly until it is no longer possible to combine nuclei. The second and third rules are then applied successively.

Figure 1.8 is an example of the decomposition of three-dimensional objects. In step 1, every nucleus corresponds to a region; for example, nuclei A, B and C correspond to regions a, b and c, respectively. There are two strong links connecting nuclei A and B. One link comes from the Y intersection of regions a, b and c. The other link comes from the arrow intersection of regions a and b. All other links are derived in the same way. Step 2 is the straightforward application of the three rules to combine nuclei. The regions correspond to nuclei in a group on an object, for example, regions a, b and c form an object. It is obvious that Guzman's method can be applied only to the pictures of three-dimensional objects.

The second method uses thresholds to transfer a multi-level picture and then finds the contours of the binary pictures as object boundaries. Prewitt[11] used the local minima of the optical density frequency distribution of a picture as the thresholds to find the background levels, cytoplasm levels and nucleus levels.

The third method is the clustering method. Zahn[16] proposed a method to group points into objects by the clustering method, which is graph theory oriented. This clustering method is motivated by the perception of two-

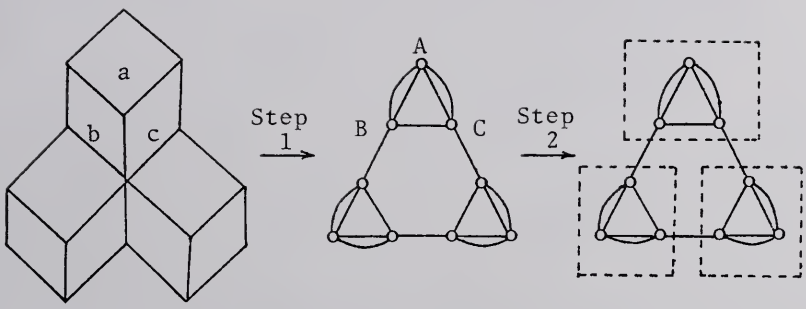


Figure 1.8. An example of the decomposition of three-dimensional objects.

dimensional point sets as separate "gestalts." The principle of grouping used is "proximity" as described by Wertheimer.[17] The proposed method is applicable to binary pictures. For a binary picture, every picture point with grey value 1 is a vertex. Picture points having grey values 1 are called object points. The connection between object points is called an edge. A weight is assigned to every edge. It is equal to the Euclidean distance between the corresponding object points. An MST (Minimal Spanning Tree) T is defined as a spanning tree of G whose weight is minimum among all spanning trees of G. Some edges in the MST can be deleted by using a factor as the measure of the significant edge inconsistency. The MST is then clustered to a forest. Every tree in the forest clusters together all the points in one object.

1.1.3. Feature Extraction

Feature extraction strongly depends on the type of pictures handled. How big the feature set should be depends on the purpose of handling the picture.

There are two main types of features. One is the local feature which depends on individual objects in the picture. Area and centroid were presented by Freeman.[18] Eden[4] has proposed the fundamental strokes as the features of handwritten English characters. Topological features are proposed by Tou and Gonzalez[5] for characterizing handwritten characters. Topological features have been used

for automatic fingerprint interpretation.[9] A skeleton[19] has been proposed to describe indirectly the shape of objects. A skeleton can be thought of as a generalized axis of symmetry of an object. At first the concept was applied to the binary pictures. Rosenfeld,[20] Montanari,[21] Philbrick[22] and others have developed algorithms to find the skeletons in binary pictures. Levi[23] generalized the concept to the multi-level pictures by defining a new distance function which took the grey level intensities into consideration. Ledley[10] used the ratio of the number of concavities to the number of segments of the boundary as the only feature in detecting the mitotic cells.

The other type of feature is the global feature. Global features are the ones which reveal the interrelationship among objects in the picture. Sometimes an object can be described in terms of fundamental components. Global features can also be used to describe the interrelationship among fundamental components of an object. Inclusion relation among regions can be found by MANS.[24] Several linguistic descriptions have been used to describe the global features. Narasimhan[7] used syntax-directed hierarchy labeling to describe the particle tracks in the bubble chamber. Shaw proposed a PDL[25] (Picture Description Language) which may be the most formal and useful linguistic approach in picture processing up to date. Linguistic approach has the advantage of describing the picture formally.

The PDL (Picture Description Language) is a picture or graph algebra over the set of primitive structural descriptions under the operations +, -, x, *, ~ and /. Figure 1.9 shows the local completeness of the operations [+, -, x, *]. Elements in the PDL are considered equal if they are equivalent. The equivalent relation is defined as:

1. S_1 is weakly equivalent to S_2 if there exists an isomorphism between graphs of S_1 and S_2 such that the corresponding edges have identical names.
2. S_1 is equivalent to S_2 if (a) S_1 is weakly equivalent to S_2, and (b) $tail(S_1) = tail(S_2)$ and $head(S_1) = head(S_2)$.

A number of useful algebraic properties are given below:

1. Each of the binary operators is associative.
2. * is the only commutative operator; x and - are "weakly" commutative.
3. The unary operator ~

   (a) ~ acts as complementation in a Boolean algebra.

   $(\sim(S_1+S_2)) = ((\sim S_2)+(\sim S_1))$

   $(\sim(S_1*S_2)) = ((\sim S_2)*(\sim S_1))$

   (b) ~ obeys a "de Morgan's law" with respect to x and -.

   $(\sim(S_1 x S_2)) = ((\sim S_2)-(\sim S_1))$

   $(\sim(S_1-S_2)) = ((\sim S_2)x(\sim S_1))$

   (c) Involution:

   $(\sim(\sim S)) = S.$

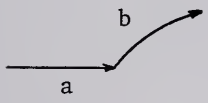
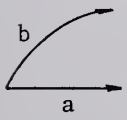
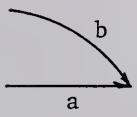
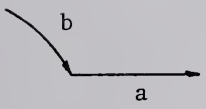
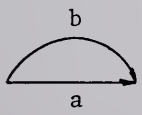
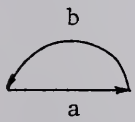
Concatenation	Description
	a+b
	axb
	a-b
	b+a
	a*b
	(a+b)* λ

Figure 1.9. Local completeness of {+, x, -, *}.

4. The / operator.

   (a) $(/(/s)) = S$.

   (b) $(/(S_1 \emptyset_b S_2)) = (/S_1)\emptyset_b(/S_2)$, where $\emptyset_b$ is a binary operator.

5. The null point primitive λ.

   (a) $S + \lambda = \lambda + S$.

   (b) $S + \lambda = S, S-\lambda = S, \lambda x S = S$.

   (c) $\sim\lambda = \lambda$.

   (d) $\lambda \emptyset_b \lambda = \lambda$.

By using some of the algebraic properties of PDL to move unary operators and label designators as far as possible within an expression, a standard form f(S) PDL of an expression S can be obtained. f(S) is defined by:

if $(S = S_1$ V $S = (/S_1)$ V $S = (\sim S_1)$ V S
$= (\sim(/S_1)))$ primitive (S_1), then $f(S) = S$

else

if $S = (S_1 \emptyset_b S_2)$, $\emptyset_b \varepsilon \{+,x,-,*\}$, then $f(S) = (f(S_1)\emptyset_b f(S_2))$

else

if $S = S_1$, then $f(S) = f(g(S))$

else

if $S = (\sim(S_1 \emptyset S_2))$, $\emptyset \varepsilon \{+,*\}$, then $f(S) = (f((\sim S_2))\emptyset f((\sim S_1)))$

else

if $S = (\sim(S_1 x S_2))$, then $f(S) = (f((\sim S_2))\text{-}f((\sim S_1)))$

else

if $S = (\sim(S_1\text{-}S_2))$, then $f(S) = (f((\sim S_2))xf((\sim S_1)))$

else

if S = $(/(S_1 \emptyset S_2))$, $\emptyset \varepsilon \{+,-,x,*\}$, then

$f(S) = (f((/S_1))\emptyset f((/S_2)))$

else

if S = $(\sim(\sim S_1))$, then $f(S) = f(S_1)$

else

if S = $(\sim(/S_1))$ V S = $(/(\sim S_1))$, then $f(S) = f((\sim f((/S_1))))$

else

if S = $(/(/S_1))$, then $f(S) = f((/S_1))$

A valid PDL expression (vPDL) is the one whose standard form is such that if $(/p^{\ell})$ appears in it one or more times for some primitive p and label ℓ, then p^{ℓ} also appears once and only once outside the scope of a /.

The graph described by a vPDL S is defined by the following algorithm:

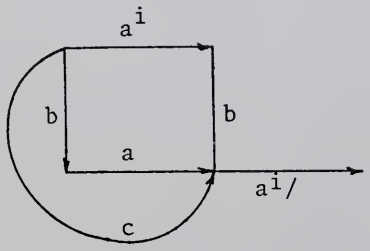
1. Transform S into standard form by applying the function f.
2. Replace each expression of the form $(/p^{\ell})$ by a new primitive $p/^{\ell}$. This removes all / operators.
3. Generate the connectivity graph of the resulting expression.
4. Connect the tail and head nodes of each edge $p/^{\ell}$ to the corresponding nodes of p^{ℓ}.
5. Eliminate all edges of the form $p/^{\ell}$.

The above algorithm formally defines the meaning of labeled expressions and the / operator. Figure 1.10 shows the graph of a vPDL.

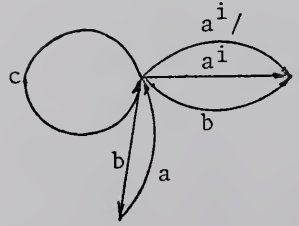
$((((a^i + b) * (b + a))^* c) + (/a^i))$

step 2 $((((a^i + b) * (b + a))^* c) + a^i/)$

step 3



step 4



step 5

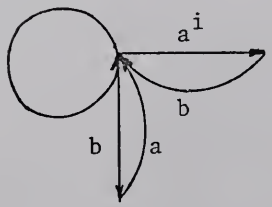


Figure 1.10. The graph of a vPDL.

It has been proved that any vPDL describes a unique primitive connectivity and any connected set of primitives can be effectively described by a vPDL. It has also been shown that the origin (tail) of a picture can be at any convenient place.

The set of rules or grammar G that generates (describes) the class of pictures P_G will be a type 2 (context-free) phrase structure grammar with the following restrictions. Each production is of the form:

$$S \rightarrow pdl_1|pdl_2|pdl_3| \ \ldots \ |pdl_n, \ n \geq 1,$$

where S is a non-terminal symbol and pdl_i is any PDL expression with the addition that non-terminal symbols are allowable replacements for primitive class names. Sentences of L(G) will consist of PDL expressions; thus, the class of terminal symbols of G will be a subset of

{+,x,-,*,~,/,(,)} V{primitive class names} V{label designators}

Each grammar G will have one distinguished non-terminal symbol from which L(G) may be generated; the symbol on the left part of the first production of G will be the distinguished symbol.

The hierarchic structural description $H_S(C)$ of a picture $C \varepsilon P_G$ having primitive structural description $T_S(C) \varepsilon L(G)$ is defined as the parse of $T_S(C)$ according to G; $H_S(C)$ is conveniently represented as a parenthesis-free tree. A simple example of PDL description of a house is given in

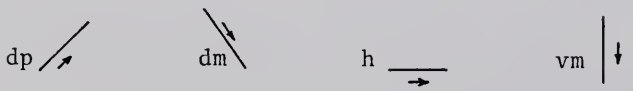
(a)

G:

House → ((vm + (h+(~vm))) * Triangle)

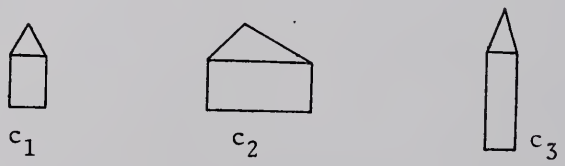
Triangle → ((dp + dm) * h)

L(G) =[((vm + (h+(~vm))) * ((dp + dm)* h))]



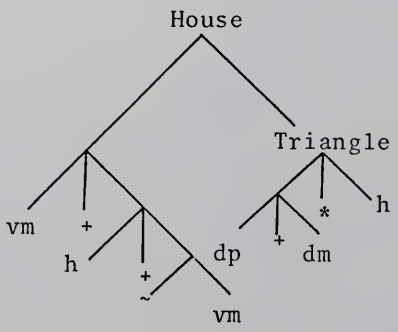
(a) G,L(G), and primitives

(b)



$T_s(c_i)$ = ((vm + (h+(~vm)))* ((dp + dm) * h))

$H_s(c_i)$:



(b) Examples and parse of a "house"

Figure 1.11. An example of the structure descriptions of a picture.

Figure 1.11. Note that all three pictures of houses in Figure 1.11 have the same primitive structural descriptions, which can be accepted by the grammar G.

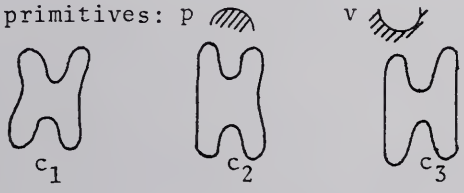
PDL can describe very well the interrelationship between primitives, but it does not have the ability to find the primitives. Figure 1.12 shows three pictures of chromosomes and the accompanying grammar to describe the pictures.

G: Chromosome → K1 * K2

K2 → v+K1+v|v+K1|K1+v|K1

K1 → p+v+p

primitives: p v



$T(C_1)$ = (p+v+p)*(v+p+v+p+v)

$T(C_2)$ = (p+v+p)*(p+v+p+v)

$T(C_3)$ = (p+v+p)*(p+v+p)

Figure 1.12. Grammar for the class of pictures of chromosomes and three examples of chromosomes.

It is seen from Figure 1.12 that the PDL sentences, which describe human chromosomes, are very simple expressions. The main problem in handling the pictures of chromosomes is to find the primitives.

1.1.4. Classification

Once a good set of features has been extracted, many classification techniques[26] are available. If the set of features of different categories are linearly separable, linear classification[27] can be used; otherwise non-linear classification[26] should be used. Multi-level classification[5] is sometimes used. For linguistic description of features, a grammar[25,28] can be designed to accept a sentence S only if S describes a picture of some specific category. A grammar can then serve the purpose of classifying objects. For example, any picture having a PDL expression which is acceptable by the grammar shown in Figure 1.12 is classified as a chromosome. Note that all three pictures shown in Figure 1.12 will be accepted as chromosomes. The main problem of designing a grammar is that it has to be complete in the sense that it should be able to accept all pictures in a category. Here we like to emphasize that in order to have a good result on the classification, a good set of features is required. If the features set is poor, no matter how good the classification technique is the result will be of poor quality.

1.2. Summary of the Remaining Chapters

Chapters II and III present two different methods of object extraction. Chapter II uses the gradient method to find the enhanced picture. Boundary points are found by

adaptively thresholding the gradients. Boundary paths will be found in the process of finding the boundary points. Gaps will then be filled in and the boundary segments are then found. Special laws are used to combine the boundary segments to form the boundaries of individual objects. Overlapping, self-folding and touching objects are decomposed. The gradients used are integers rather than real numbers, such as those used by Roberts.[1] Hence less storage is required. The threshold is adaptive rather than fixed. It is then less sensitive to the noise. Chapter III presents the contour analysis method. This method was motivated in experimenting with area picture data by showing different levels of a picture in a display unit. The main idea is that in a small window section in the picture, the threshold for transferring the picture to a binary picture is approximately constant. The thresholds are adaptive rather than fixed, such as those used by Prewitt. The result of this method is very successful, especially for the area pictures. Chapter IV discusses the graph theory approach to the picture processing. Skeletons of objects can be found by this approach. Further theoretical research should be done in this area.

Chapter V presents the extraction of several important features for area pictures. Area, centroid, shape, principal axis direction and elongation index of an object are the local features discussed in this chapter. Inclusion

relation among objects and the distribution of objects in a picture are the global features presented in this chapter.

Chapter VI discusses the experiments with biomedical images by the above methods and suggests further research. Chromosome, skin cell and blood cell pictures are analyzed. In experimenting with chromosome pictures, the main problem is finding chromosomes in a picture. In experimenting with skin cell pictures, the main problem is to detect the tumors. In experimenting with blood cell pictures, the main problem is to find the histogram of the intensities. All the experiments show very promising results. It is hoped that further research can produce a more sophisticated image analyzing system.

CHAPTER II

OBJECT EXTRACTION BY THE GRADIENT METHOD

When human beings look at any scene, the impact information we get is the shapes of the objects in the scene. The information revealing the shapes of the objects are the boundaries. If the objects are overlapped, it is possible to use the information of boundaries and grey intensities to decompose objects. As we can very easily imagine, one way to find the boundaries is to use the fact that usually the boundaries consist of those points having very high change of intensities from their neighbors. Using this property to find the boundaries is called the gradient method, because the change of intensities is measured by the gradient. The gradient method is very good if there are high contrasts in the boundaries, even if there exist some nonuniform distribution of the intensities in the objects.

Some definitions will be introduced before getting into the problem.

Definition 2.1.--A point p_i in the picture plane I is an 8-neighboring point of the picture point p in I if and only if $0<d(p,p_i)<2$, where d is the Euclidean distance function. In order to make the later discussion easier, the

8-neighboring points of p are labeled as shown in Figure 2.1. The set of all 8-neighboring points of p is denoted as N(p). The octal chain code which encodes the line segment from p to p_i is i. Because the i's values range from 0 to 7, the code is an octal code.

It is obvious that any curve in the digitized picture is approximated by a sequence of line segments which join their points to their 8-neighboring points. Hence any curve in the digitized picture can be represented by a sequence of octal chain codes and the start point of the curve. Let C be a curve represented by the chain codes $c_1 \ldots c_n$ and the start point o. The reverse of the curve C can then be represented by the sequence $C^{-1} = (c_n\text{-}4)_8 \ldots (c_1\text{-}4)_8$ and the start point o* which is the end point of the curve C. Because of the small storage required to store the chain code and because it is easy to manipulate, this technique is used throughout the dissertation to encode the curve in the digitized picture.

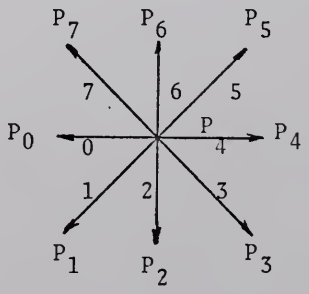


Figure 2.1. The 8-neighboring points of a picture point p and the octal chain codes.

2.1. Boundary Segments Finding

Physically, a boundary point is a picture point which had a high increase of intensity from its neighbors. The rate of change of intensity from p_i to p, where p_i is an 8-neighboring point of p, can be evaluated by a differentiator, which is defined as $h^*(p,p_i) = [g(p)-g(p_i)]/d(p,p_i)$, where g is the picture function and d is the Euclidean distance. It is easily seen from Figure 2.1 that $d(p,p_i) = 1$ if i is even and $d(p,p_i) = \sqrt{2}$ if i is odd. The h^* function can then be redefined as $h^*(p,p_i) = \Psi^*[g(p)-g(p_i),E(i)]$, where E(i) = 0 if i is odd and E(i) = 1 if i is even. The Ψ^* function is defined by the following mapping table.

	g(p)-g(p)								
Ψ^*	-(n-1)	-(n-2)	...	-1	0	1	...	n-2	n-1
E(i) 0	$-(n-1)/\sqrt{2}$	$-(n-2)/\sqrt{2}$	...	$-1/\sqrt{2}$	0	$1/\sqrt{2}$	...	$(n-2)/\sqrt{2}$	$(n-1)/\sqrt{2}$
1	-(n-1)	-(n-2)	...	-1	0	1	...	n-2	n-1

The situation of a drop of intensity from p_i to p is not under consideration, because this situation will be considered as an increase of intensity from p to p_i. Hence we can define an h^{**} function which is equal to the h^* function when the value of h^* is equal to or greater than 0. The value of h^{**} is set equal to 0 when the value of h^* is less than 0. For an n-level picture there are 2n-1 possible values of h^{**} function = {0,1,...,2n-1}. Because the range

of the h** function is discrete, there is a one-to-one mapping h*** from R = range(h**) onto I_b, where I_b is a subset of the integer numbers, and h*** preserves the ordering of the elements. h*** serves the purpose of quantizing the range of h**. A new function h can then be defined as the composition of h*** and h**, that is, h = h***.h**. This function is a measure of the quantized rate of increase of intensities. We can write the h function in terms of a Ψ function, $h(p,p_i) = \Psi[g(p)-g(p_i),E(i)]$. For n = 8, the function can be represented by the following mapping table:

		$g(p)-g(p_i)$														
Ψ		-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
E(i)	0	0	0	0	0	0	0	0	0	1	3	5	6	8	10	11
	1	0	0	0	0	0	0	0	0	2	4	7	9	12	13	14

The magnitude of the gradient at a picture point p is defined as the maximal increase of intensity from the neighboring points to the picture point. The following definition is then yielded.

Definition 2.2.--The gradient b (or sometimes will be called b-value) at a picture point p of the picture g is defined as

$$b(p) = \max_{i\in[0,1,\ldots,7]} \{h(p,p_i)\},$$

where the h function is defined earlier.

The enhanced picture of a digitized picture g is a digitized picture with the intensity at every picture point equal to the gradient at the corresponding picture point in the g picture. The enhanced picture of an n-level picture is of 2n-1 level. It will be seen later that one extra value is required to identify the boundary points. It is then obvious that a k bit picture will yield a k+1 bit enhanced picture. The enhanced picture of Figure 1.4, which is a binary picture of numeral "6", is shown in Figure 2.2.

Because of the unavoidable noise appearing in the picture, there is no way to find the real boundaries in one step. We can break the process of finding boundaries into several steps. The first step is to find all those points which can quite possibly be boundary points. These points are called tentative boundary points. The fact that the boundary points of a boundary path are connected can be used in the process of finding tentative boundaries.

The following section details the scheme of finding the tentative boundary paths. Octal chain codes are used to encode the paths.

2.1.1. Tentative Boundary Path Searching

The method of finding the tentative boundary paths is based on the principle that a tentative boundary point is a point which has a gradient greater than the gradients of the neighboring nonboundary points and is connected to some

other tentative boundary points. Figure 2.3 shows the flow chart used to search the tentative boundary paths.

The input to this tentative boundary path finder is the raw digitized picture. The output is:

1. The tentative boundary picture which is a binary picture having value -1 at the tentative boundary point and 0 elsewhere.

2. A list of tentative boundary paths. For each tentative boundary path, it has

   2a. the start point of the tentative boundary path,

   2b. the length of the tentative boundary path,

   2c. a sequence of octal chain codes which encode the path,

   2d. an indicator which denotes whether the tentative boundary path is closed or open, and

   2e. the end point of the tentative boundary path if the indicator denotes that it is an open path.

The threshold θ in the flow chart, which is used to pick up the first point in a tentative boundary path, is usually decided by the following method.

First, find the histogram of the b values in the picture, which is a plot of number of picture points whose gradients are greater than or equal to a b-value. Figure 2.4 is the histogram of the enhanced picture shown in Figure 2.3. To determine which is θ from this histogram, find

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	2	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	2	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	2	1	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	2	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	2	1	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	2	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	2	1	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	2	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	2	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	2	1	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	2	0	0	0	1	2	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0
0	0	0	2	0	0	0	0	0	0	0	0	1	2	2	0	0	0	0	0	0	0	0	0
0	0	0	2	1	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0
0	0	0	0	2	0	0	0	1	2	2	2	1	0	0	2	0	0	0	0	0	0	0	0
0	0	0	0	2	0	0	0	2	0	0	0	2	0	0	2	0	0	0	0	0	0	0	0
0	0	0	0	2	1	0	0	2	0	0	0	2	0	0	2	0	0	0	0	0	0	0	0
0	0	0	0	0	2	1	0	1	2	2	2	1	0	0	2	0	0	0	0	0	0	0	0
0	0	0	0	0	0	2	1	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	2	2	1	0	0	1	2	2	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
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 2.2. The enhanced picture of Figure 1.4.

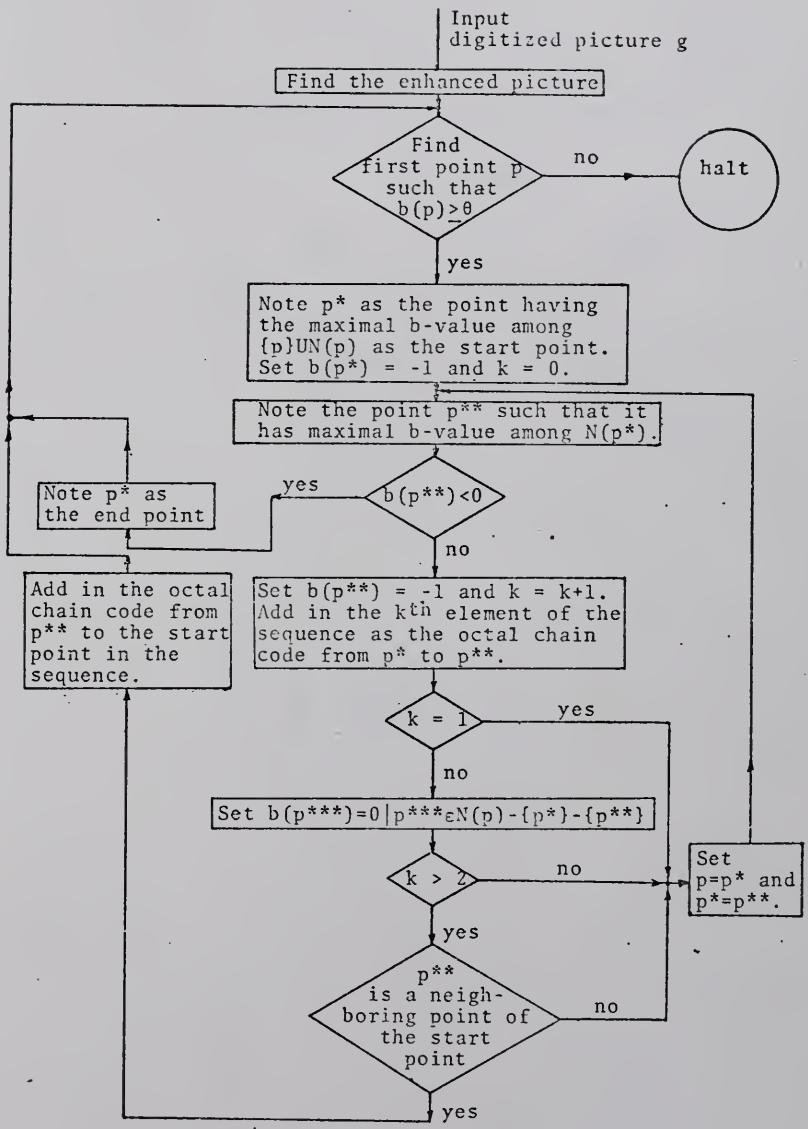


Figure 2.3. Flow diagram for searching for tentative boundary paths.

the greatest drop in the histogram from θ-1 to θ. In this example θ will be set as 1.

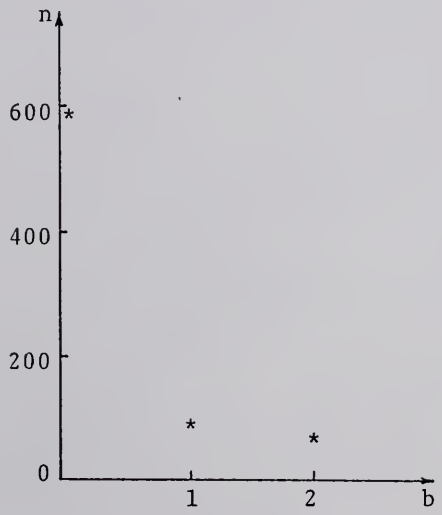


Figure 2.4. Histogram of b-values of Figure 2.3.

Searching the tentative boundary paths of Figure 2.2 yields two tentative boundary paths:

path 1:

start point = (2,9),

length = 52,

octal chain codes = 44222121221234444343222222101000

70777667666566565656 5,

closed contour.

path 2:

start point = (16,10),

length = 10,

octal chain codes = 4432100765,

closed contour.

In this special example, the tentative boundary paths are the final boundary contours. In most of the practical cases, because of the existence of noise, overlapping, self-folding and touching, there will exist gaps between tentative boundary paths and tails of tentative boundary paths. The following sections discuss the strategies of solving these problems.

2.1.2. Procedure of Filling Gaps and Determining Line Segments

To fill the gaps between tentative boundary paths, one has to examine the extreme points of tentative boundary paths. For an extreme point p of a tentative boundary path C, let p' be the nearest tentative boundary point which is not on C or is on C and has more than five points from p along C. If the Euclidean distance between p and p' is less than 3, p is then connected to p' through the shortest path. If p' is an extreme point of a tentative boundary path C', tentative boundaries C and C' will then be combined. If p' is not an extreme point, p' will then be an intersection node. Figure 2.5 shows the flow diagram for filling gaps.

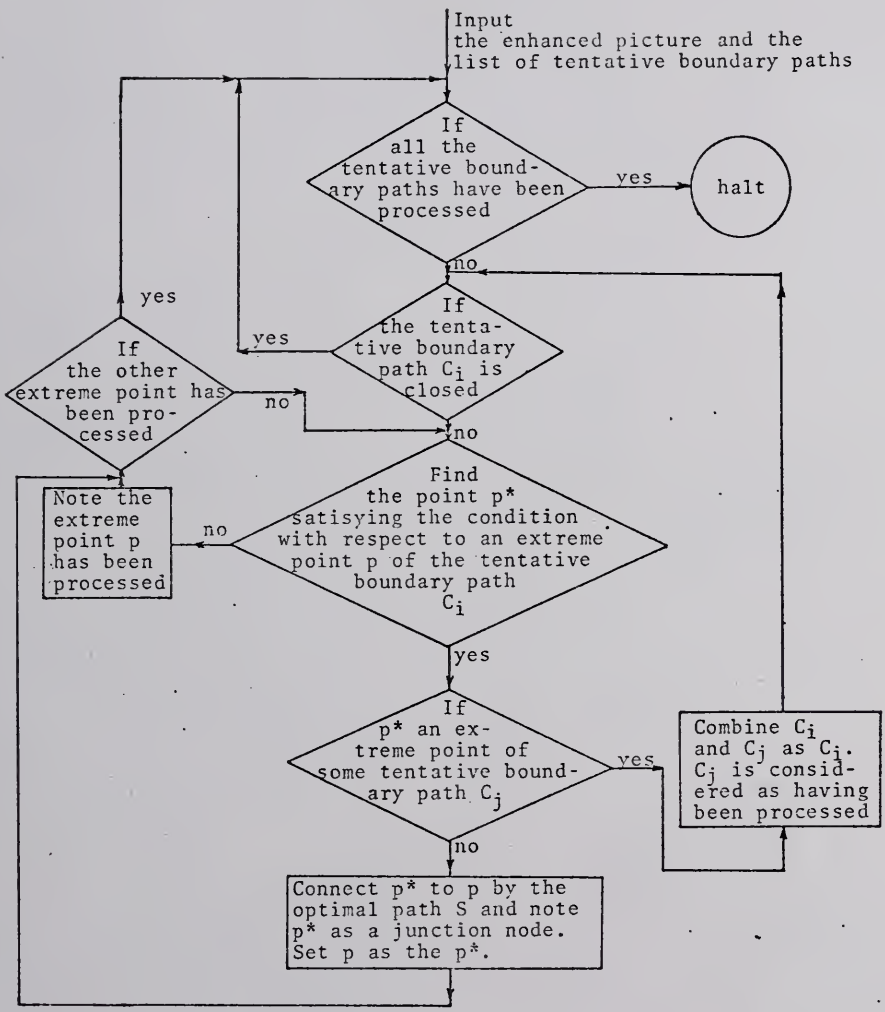


Figure 2.5. Flow diagram for filling gaps between tentative boundary paths.

A boundary segment is defined as a boundary path between successive junction nodes or a boundary path which does not have junction nodes on it. A tail of a boundary is the boundary path between an extreme point, which is not a junction node, and a junction node. Hence the main procedure in determining boundary segments is to order the junction nodes along the boundary paths. Once boundary segments are determined, the boundary segments joined at a junction node can easily be noted. Because only contours of objects are of interest, all tails will be erased.

2.2. Combining Boundary Segments to Form the Boundary Contours of Objects

Sometimes a point p on the boundary will degenerate into several junction nodes after applying the process stated in the previous section to the digitized pictures. It happens most often when p is a real junction point. Let p be a boundary point which degenerates into k junction nodes $n_1,\ldots,n_k$. $N_p = \{n_1,\ldots,n_k\}$ is the complete set of junction nodes degenerated from the point p. The ideal cases (i.e., no degeneration) are:

1. point p is not a junction point and N_p is empty, and
2. point p is a junction point and the cardinal number of N_p is 1.

Figure 2.6 shows some degeneration cases. In the degeneration cases, there must exist singular paths connecting

the degenerated junction nodes. Let N be the set of all junction nodes in the picture. The degenerate relation R is defined on N such that nRn', where n, n'εN, if there exists a sequence of intersection nodes $n_1 = n, n_2, \ldots, n_k = n'$ satisfying the condition that there is a singular segment between junction nodes n_i and n_{i+1}, i=1,...,k-1. The degenerate relation R is obviously an equivalence relation. The equivalence relation R can thus partition N into a collection of equivalence classes. Every equivalence class is then a complete set of junction nodes degenerated from some point. In the processing, we have to find the singular segments first, then we can decide which junction nodes form a complete set. As one would expect, the most reasonable and easiest way to determine if a boundary segment is singular is by the length of the segment.

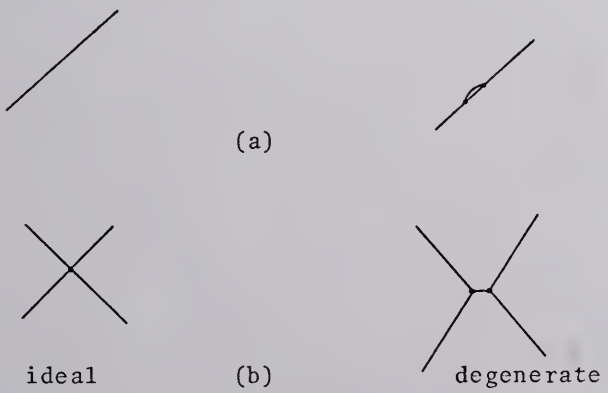


Figure 2.6. Examples of degeneration from (a) a nonjunction boundary point and (b) a junction point on the boundary.

A threshold μ is assigned such that if a boundary segment is of length less than μ, it is classified as a singular segment; otherwise it is nonsingular. Let N be an equivalence class induced by the relation R and let S be the set of all singular segments connecting to the junction nodes in N. (N,S) forms a junction area. Let E be the set of all nonsingular segments connecting to the junction nodes in N. The ordering of the elements in E is very useful in combining boundary segments.

It is important to point out that only the chain codes, which encode the line segments connecting the junction nodes, are used to detect the ordering of the nonsingular boundary segments around the junction area. The ordering of segments around the junction area can be either clockwise or counterclockwise. Because of the line encoding scheme (octal chain code) we used, there are at most eight boundary segments joining at a junction node. Figure 2.7 shows the flow diagram for ordering the nonsingular segments around junction areas.

Every equivalence class of junction nodes can be thought of as a single junction node. Any nonsingular segment which connects to some other nonsingular segment through the equivalence class of junction nodes is considered as through the corresponding single junction node. We can then imagine the boundary segments picture as an ideal one in the sense that no singular path exists.

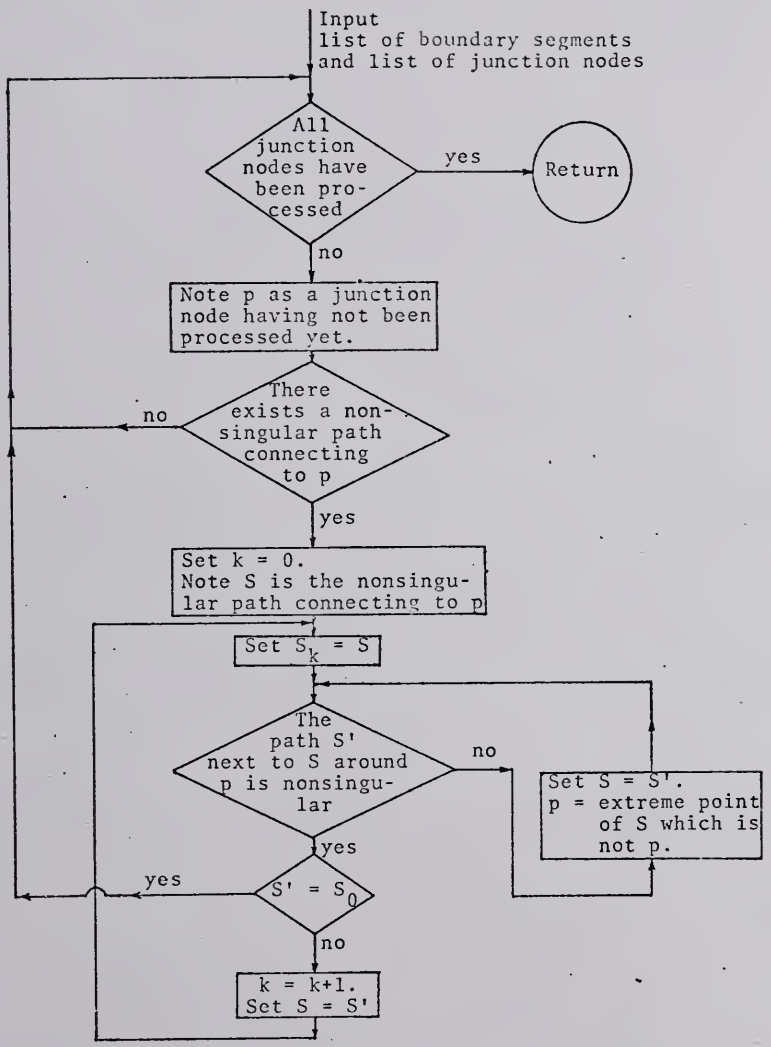


Figure 2.7. Flow diagram for ordering nonsingular paths around junction areas.

A nonsingular segment S_i is connected to a nonsingular segment S_j if and only if there exists an equivalence class N of junction nodes such that both S_i and S_j contain some junction nodes in N. This relation is denoted by E. The relation E' of connectivity of two nonsingular paths is a transitive closure of the relation E. The picture consisting only of the nonsingular segments (and the singular paths which connect them) of the equivalence class induced by the relation E' is called an isolated picture.

A contour is defined as a simply closed curve. We can partition a contour into a sequence of successive adjacent boundary segments. Hence the concatenation of successive adjacent boundary segments can form a contour, if the condition of being a contour is satisfied. A picture point p is included by a contour if p is a point on the contour or if every ray initiated from p will meet an odd number of times with the contour. If the above condition is not satisfied, the picture point is said to be excluded by the contour. The set of all picture points which are included by a contour is called the region enclosed by the contour.

Definition 2.3.--In an isolated picture, an elementary region is defined as a region enclosed by a minimum contour in the sense that the region does not include any region which is enclosed by a contour in the picture. A region in an isolated picture, which is the union of all elementary regions, is called the whole region of the picture. The

contour in an isolated picture which encloses the whole region is called the exterior boundary of the isolated picture. Any nonsingular segment in the exterior boundary is called an exterior segment.

Let m be the number of junction areas and k be the number of nonsingular segments in an isolated picture; there are k-m+1 elementary regions[29] and one exterior boundary. The above facts are useful in terminating the searching process. Figure 2.8 shows the flow diagram for finding all the minimum contours and the exterior boundary in an isolated picture.

From here on, the terms "path" and "node" are used to imply "nonsingular segment" and "junction area," respectively, unless otherwise specified.

Let us look at the different examples shown in figure 2.9 to get a feeling of "how our visual systems combine paths into object boundary contours."

It is amazing that we don't have to know the intensity in each elementary region to find out that in (a) there are two objects: one is enclosed by the boundary contour consisting of paths S_0 and S_2, the other is enclosed by the boundary contour consisting of paths S_1 and S_3; in (b) there is a self-folding object enclosed by the contour consisting of paths S_0 and S_1 with the region enclosed by the contour consisting of paths S_1 and S_2 as the folded part; and in (c) there are two touching objects: one is enclosed by the

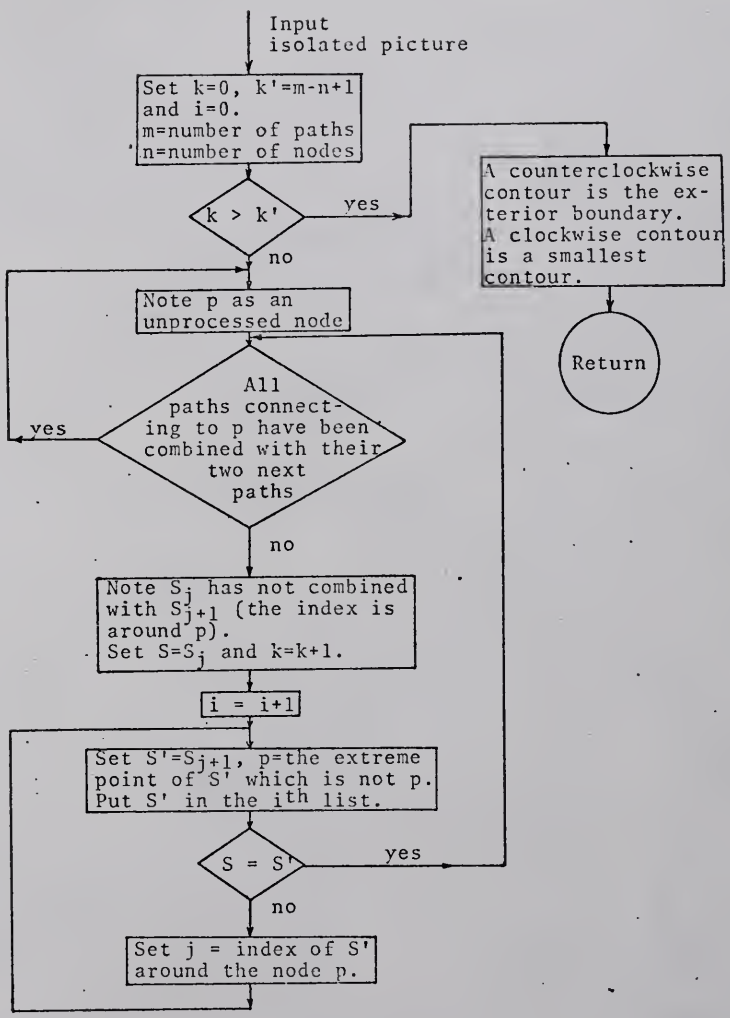


Figure 2.8. Flow diagram for finding all the smallest contours and the exterior boundary in an isolated picture.

boundary contour consisting of paths S_0 and S_1, the other is enclosed by the boundary contour consisting of paths S_1 and S_2.

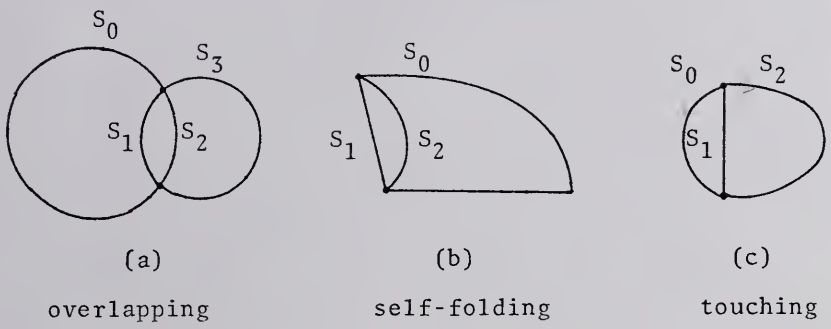


Figure 2.9. Examples of (a) overlapping, (b) self-folding and (c) touching.

Sometimes objects touching at a point may occur such as the examples shown in Figure 2.10. We call this kind of node the looping node.

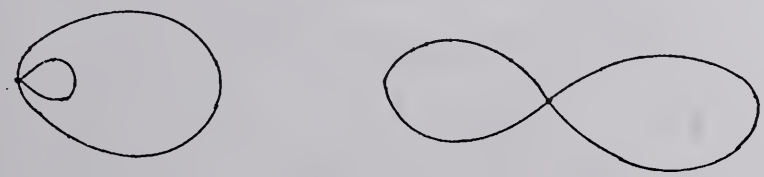


Figure 2.10. Examples of looping nodes.

If there is an odd number of paths connecting to a junction node, self-folding or touching may occur. Usually the path which belongs to two objects (here we consider self-folding as a special case of touching) is quite "straight." The straightness of a path can be determined by the filtered sequence of differences of successive chain codes, which can be obtained by the digital filtering method. If each filtered difference of successive chain codes of a path has absolute value less than 1, the path is then considered as "straight." Figure 2.11 is a flow diagram for combining the paths in forming the boundary contours of objects.

Now we have to find out which objects are overlapped, which objects are touched and which objects are self-folded. Let C_1 and C_2 be the boundaries of two objects O_1 and O_2, respectively. If there does not exist a common node between C_1 and C_2, O_1 and O_2 are separate objects. If there exists a common node between C_1 and C_2 and if there does not exist a common path between C_1 and C_2, O_1 and O_2 are overlapped objects, such as Figure 2.9(a). If there exists a common path between C_1 and C_2, O_1 and O_2 are either touching or self-folding. Assume that both C_1 and C_2 have a common path S. Trace C_1 and C_2 so that S is traced in the same direction. If both the contours C_1 and C_2 are traced in the same direction (that is, either both are clockwise or both are counterclockwise), O_1 and O_2 form a self-folding object.

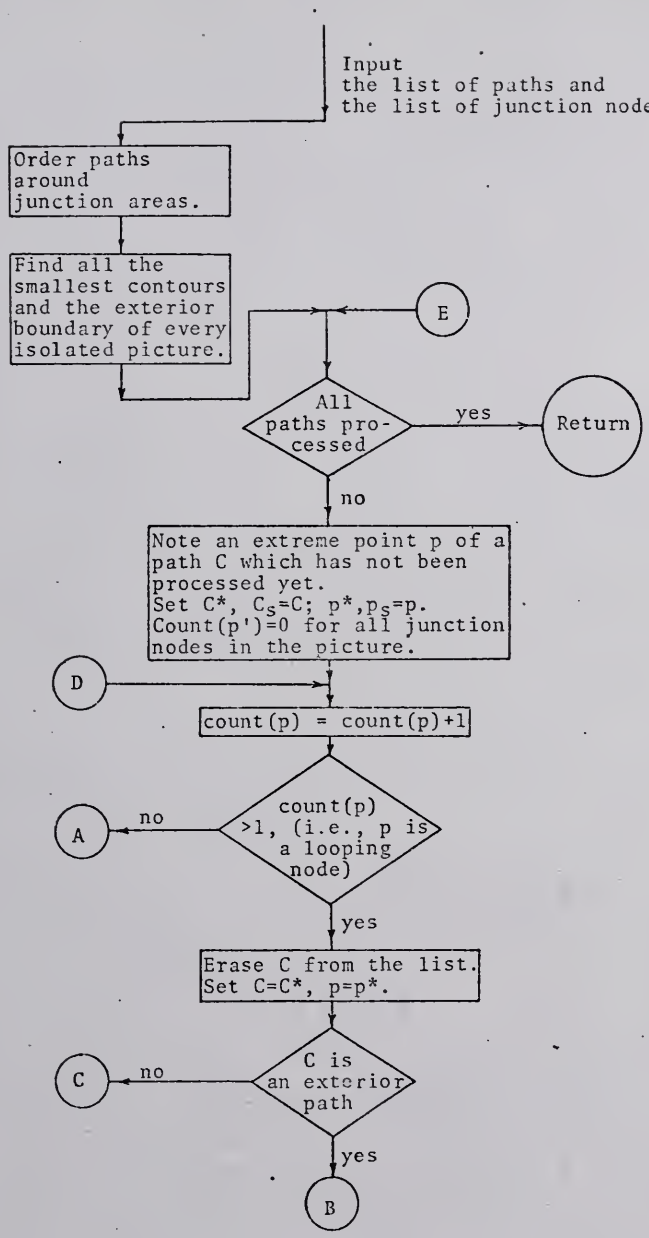


Figure 2.11. Flow diagram for combining paths in forming the boundary contours of objects.

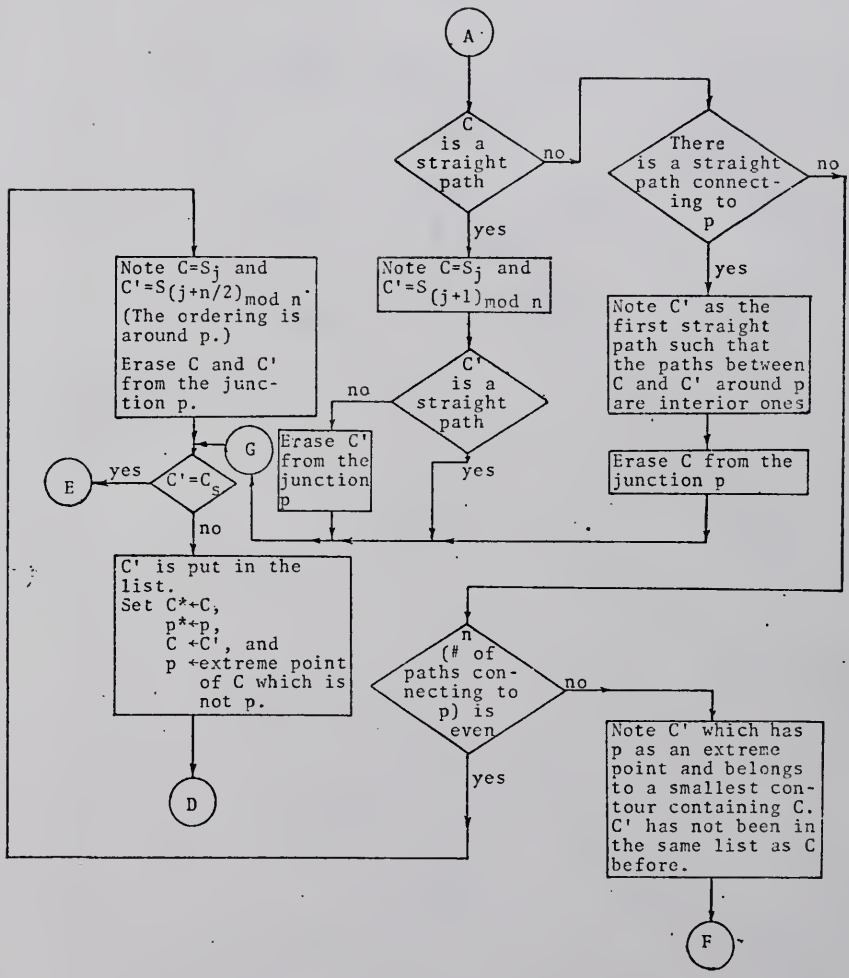


Figure 2.11. Continued

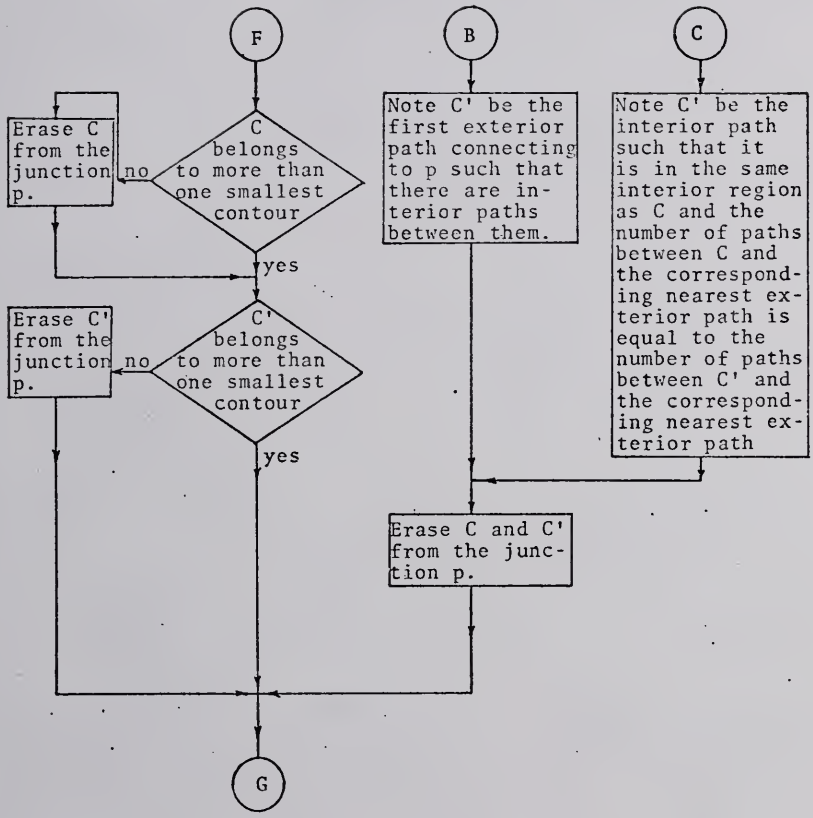


Figure 2.11. Continued

If C_1 and C_2 are traced in different directions, O_1 and O_2 are touching each other. For example, in Figure 2.9(b), the common path is S_1. Trace $C_1 = S_1S_2$ and $C_2 = S_1S_2$ so that S_1 is traced upward in both cases. Both C_1 and C_2 are traced in the clockwise direction. Hence O_1 and O_2 are self-folding. In Figure 2.9(c), trace $C_1 = S_0S_1$ and $C_2 = S_2S_1$ so that the common path S_1 is traced upward in both contours. C_1 is then traced in counterclockwise direction, while C_2 is traced in clockwise direction. Hence C_1 and C_2 are touching each other.

CHAPTER III

OBJECT EXTRACTION BY THE CONTOUR ANALYSIS

It is known[6] that boundary paths are very easily extracted from a binary picture. An n-level picture g can be transformed into n-1 binary pictures $\beta_j = \phi_j(g)$, j=1, ...,n-1 such that

$$\beta_j(p_i) = \begin{cases} 1 & \text{if } g(p_i) \geq j \\ 0 & \text{otherwise} \end{cases}$$

$\forall p_i \in I$, where I is the picture plane and j is the threshold to transform a grey picture into a binary picture. The transformation from an n-level picture into a collection of (n-1) binary pictures $\{\beta_1,\ldots,\beta_{n-1}\}$ is denoted by $\phi = \{\phi_1,\ldots,\phi_{n-1}\}$. The reason for excluding β_0 pictures from consideration is that it is a trivial picture with 1's everywhere in the picture plane. A binary picture β_j is said to be a subpicture of a binary picture β_i, denoted by $\beta_j \subset \beta_i$, if β_j and β_i have the same picture plane I and $\forall p \in I$, $\beta_i(p) = 0$ implies $\beta_j(p) = 0$. It is easily seen that $\beta_1 \subset \ldots \subset \beta_{n-1}$. Obviously ϕ is a one-to-one transformation from an n-level picture into a collection of (n-1) binary pictures $\{\beta_1,\ldots,\beta_{n-1} | \beta_1 \subset \ldots \subset \beta_{n-1}\}$. Figure 3.1 shows the binary pictures transformed from the 8-level picture shown in

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0
```

(a) β_1

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0
1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0
1 1 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0
1 1 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0
1 1 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0
1 1 1 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0
1 1 1 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0
1 1 1 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0
1 1 1 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0
1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0
1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0
1 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0
1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0
1 1 1 1 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0
1 1 1 1 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0
```

(b) β_2

Figure 3.1. The binary pictures transformed from Figure 1.5.

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

(c) β_3

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

(d) β_4

Figure 3.1. Continued

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	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	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(e) β_5

Figure 3.1. Continued

Figure 1.5. Since the highest level in Figure 1.5 is 5, both β_6 and β_7 are trivial pictures, having 0 in every picture point, and are not shown in Figure 3.1.

Boundary contours can be extracted from β_j, $j=1,\ldots,n-1$. Inclusion relations can be set up among boundary contours. A boundary contour C_1 in β_j will be included in a boundary contour C_2 in β_i, where $i<j$. If the shapes of C_1 and C_2 are similar, C_2 is more likely to include the whole object. This fact is quite obvious from the experiments. Using this fact to extract objects from the scene is very effective, especially for area pictures, those having only objects consisting of areas.

3.1. Some Fundamental Concepts in Binary Pictures

A picture point in the picture plane of binary picture β is said to be an object point if $\beta(p) = 1$, otherwise it is a background point. A picture point $p \varepsilon I$ is said to be directly connected to $p' \varepsilon I$ if $d(p,p')<2$, where d is the Euclidean distance function. An object point p is said to be connected to an object point p' if there exists a sequence of object points $(p_0,\ldots,p_m)$ such that $p = p_0$ and $p' = p_m$ and p_i is directly connected to p_{i-1}, $i=1,\ldots,m$. A maximal set of connected object points in a binary picture β is an element in that picture. An object point p in a binary picture β is a boundary point if there exists a background point

p_i such that $d(p,p_i)<2$. A contour is a sequence of picture points $(p_0,\ldots,p_{m-1})$ such that $0<d(p_i,p_{i+1})<2$, where $i = 0, \ldots,m-1,p_m=p_0$ and m is the length of the contour. As stated in the last chapter, it can also be represented by a sequence of octal chain codes. A picture point p is included by a contour if p is a point in the sequence, or every ray initiated from p will meet an odd number of times with the contour. The set of all picture points which are included by a contour C is called the region enclosed by C. An exterior contour of an element B in a picture is a contour such that when the contour is traced in a clockwise direction, then all points in the element will be on the right-hand side. An interior contour of an element in a picture is a contour such that when the contour is traced in a clockwise direction, then all picture points in B are in the left-hand side. It is easily seen that for any element B, there is only one exterior contour and there is a finite number of interior contours. There is one exterior contour and one interior contour in the binary picture shown in Figure 1.4.

Let β_j be a subpicture of the binary picture β_i. For any exterior contour $C^{(j)}$ in β_j there exists an exterior contour $C^{(i)}$ in β_i such that the region enclosed by $C^{(j)}$ is a subset of the region enclosed by $C^{(i)}$. For any interior contour $C^{(j)'}$ in β_j there may exist an interior contour $C^{(i)'}$ in β_i such that the region enclosed by $C^{(i)'}$ is a subset of the region enclosed by $C^{(j)'}$.

3.2. Contours Finding in Multi-level Picture

In a multi-level picture g, a picture point p is a boundary point if there exists a point p_i such that $d(p,p_i)<2$ and $g(p)>g(p_i)$. p_i is called an adjacent background point, with respect to the contours passing through the boundary point p in binary pictures β_j, $g(p_i)<j<g(p)$. In order to find the contours, a labeling scheme is used. Let g be a n-level picture. Assume that n is even, which is generally the case, as n=8 for the picture shown in Figure 1.5. The rules of the labeling scheme are:

1. all boundary points in the i^{th} contour are labeled as n-1+2i,
2. all adjacent background points with respect to the i^{th} contour are labeled as n-2+2i.
3. the labeling of the odd numbers has priority over that of the even numbers, and
4. for odd numbers, the labeling of large numbers has priority over the small numbers. For even numbers, the labeling of small numbers has priority over the large numbers.

It is easily seen that a contour C in a multi-level picture is a contour in the binary pictures β_j, $b(C)<j\leq g(C)$, where $g(C) = \min_{p\in C}\{g(p)\}$ and $b(C) = \max\{g(q) \mid q$ is an adjacent background point with respect to C$\}$. g(C) and b(C) are called the intensity and the background intensity of the contour C, respectively.

A rotator is a vector initiating from an object point to its eight-neighboring points, and hence can be represented by an octal chain code. A rotator rotating in counterclockwise direction is needed in the process of finding contours. If a rotator pointed to the picture point p_i, it will next point to $p_{(i+1)_{mod8}}$. The rotator is the index of the neighboring point.

In finding the contours in a picture g, first scan the picture g in forward direction. Let p be the first picture point satisfying the conditions that $g(p)>g(p_o)$ and either the picture point p has not been labeled before, or the picture point p is on some contour C', which has been found, and $g(C')<g(p)$. p is then the start point of a contour to be found. Set the initial position of the rotator in the direction indicated by octal chain code 1. Find the first object point pointed by the rotator which rotates in a counterclockwise direction. After the second contour point has been found, the initial position is set at the direction indicated by the octal chain code $(i+4)_{mod8}$, where i is the octal chain code indicating the position of last rotator. By this method of finding the contours, an interior contour will be encoded in a clockwise direction, while an exterior contour will be encoded in a counterclockwise direction.

Let $c_1 \ldots c_m$ be the sequence of the octal chain codes of a contour C. $c_i c_{i+1}$ is deleted if $|c_i - c_{i+1}| = 4$. By this procedure, the sequence of the chain code will be

reduced to $c_1,\ldots c_m,$. If the reduced sequence is empty, then the contour is one which can be broken into two paths $S_1 = c_1\ldots c_{m_1}$ and $S_2 = c_{m_1+1}\ldots c_m$ such that $S_2 = S_1^{-1}$. A contour having an empty reduced sequence is considered to be an exterior contour. If the reduced sequence is not empty, then the sum of the differences of adjacent octal chain code is the parameter to indicate whether the contour is encoded clockwise or counterclockwise. If the sum is -8, the contour is traced in a clockwise direction. Hence the contour is an interior one. If the sum is +8, the contour is traced in a counterclockwise direction. Hence the contour is an exterior one. Figure 3.2 shows the flow diagram used to find contours in a multi-level picture.

Figure 3.3 is the labeled picture of the 8-level picture shown in Figure 1.5.

There are 14 contours in Figure 1.5.

Contour 1: start point = (1,1),

octal chain codes = 22222222222222224444444444
44565656665666566600000000
000000000,

length = 61,

intensity = 1,

background intensity = 0,

exterior contour.

Contour 2: start point = (1,1),

octal chain codes = 22222222222222224444534454
55656656766765500000000000
000

length = 55,
intensity = 2,
background intensity = 1,
exterior contour.

Contour 3: start point = (1,1),
octal chain codes = 22222222666656660
length = 17,
intensity = 3,
background intensity = 2,
exterior contour.

Contour 4: start point = (1,6),
octal chain codes = 44344540000000
length = 14,
intensity = 3,
background intensity = 2,
exterior contour.

Contour 5: start point = (1,10),
length = 0,
intensity = 4,
background intensity = 3,
exterior contour.

Contour 6: start point = (2,1),
octal chain codes = 22650,
length = 5,
intensity = 4,
background intensity = 3,
exterior contour.

Contour 7: start point = (2,6),
octal chain codes = 444432210211007666656653,
length = 25,
intensity = 2,
background intensity = 1,
interior contour.

Contour 8: start point = (5,12),
octal chain codes = 12102110221343454566556760660,
length = 28,
intensity = 3,
background intensity = 2,
exterior contour.

Contour 9: start point = (9,11),
octal chain codes = 2171222245556660,
length = 16,
intensity = 4,
background intensity = 3,
exterior contour.

Contour 10: start point = (11,4),
octal chain codes = 2221776553,
length = 10,
intensity = 2,
background intensity = 1,
interior contour.

Contour 11: start point = (12,6),
length = 0,

intensity = 4,
background intensity = 3,
exterior contour.

Contour 12: start point = (12,10),
length = 0,
intensity = 5,
background intensity = 4,
exterior contour.

Contour 13: start point = (14,1),
octal chain codes = 222451766,
length = 9,
intensity = 3,
background intensity = 2,
exterior contour.

Contour 14: start point = (16,6),
octal chain codes = 1753,
length = 4,
intensity = 2,
background intensity = 1,
interior contour.

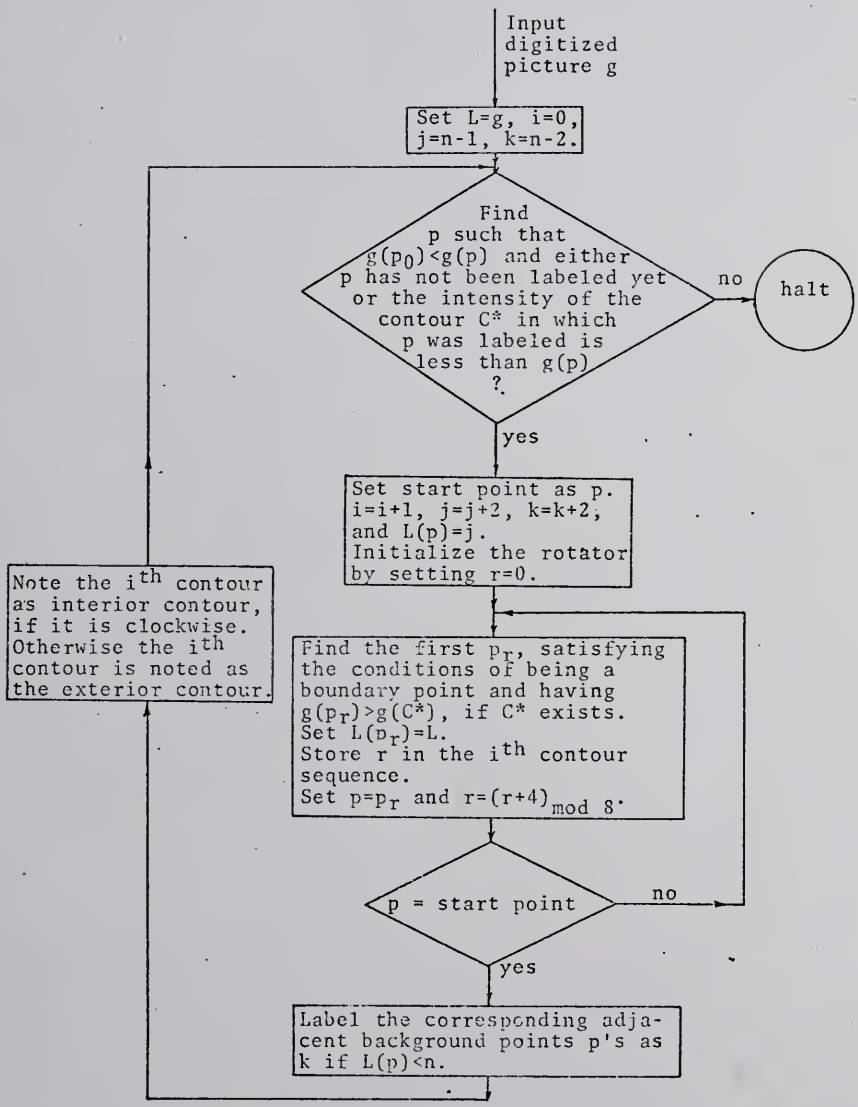


Figure 3.2. Flow diagram for finding contours in a multi-level picture.

```
13131111211515151517151515111 9 9 9 8
19191221202121212121151414111010 1 9 8
19131221202020202020211411010 1 1 9 8
1913122120 1 1 1 12021221110 1 1 9 8
1312212020 1 120202021232311 0 1 9 8 8
13122120 1 1 120212123 3231110 1 9 8 0
13122120 1 12020212223 3231110 10 9 8 0
13122120 120202123232424232211 0 9 8 0
131221202020212222324252524231 1 9 8 8 0
111227212121222325242525242310 9 8 0 0
112726272223232530253025231110 9 8 0 0
272626272229242530313025221110 9 8 0 0
27262627222324253030252311 0 9 8 8 0 0
3327262723 32425 42524231110 9 8 0 0 0
33322722352323252523231110 9 8 8 0 0 0
333233353435222323111110 10 9 8 0 0 0 0
3333111135 9111111 9 9 9 9 9 8 8 0 0 0 0
```

Figure 3.3. The labeled picture obtained from Figure 1.5.

3.3. Inclusion Relation Among Contours

As stated in the beginning of the chapter, inclusion relation must be set among contours to extract objects. Inclusion relation can be easily found from the labeled picture L and the original n-level picture g. The labels and intensities of two successive picture points p_o and p (in the forward raster direction, i.e., p is at the right-hand side of the picture point p_o) are required. The information from the label of the picture point p_o is stored as a state. Three kinds of labels exist. State 1 is that $L(p_o)<n$, that is, p_o is not labeled in the labeling process. State 2 is that $L(p_o)\geq n$ and is even, that is, p_o is an adjacent background point. State 3 is that $L(p_o)\geq n$ and is odd, that is, p_o is a boundary point. Figure 3.4 is the state diagram for finding the inclusion relation among contours. The actions in Figure 3.4 should be explained. An array is initiated every time a line in the picture is scanned. State 1 is the initial state. If the action consists of entering the region, the region name is put in the array. If the action consists of leaving the region, the region name is taken out of the array. The inclusion relation is a partially ordered relation, which can be represented by a Hasse graph.[30] Every time a contour C is put in the array, the contour C is included in the contour which is next to C in the array. At a picture point p, there may pass several

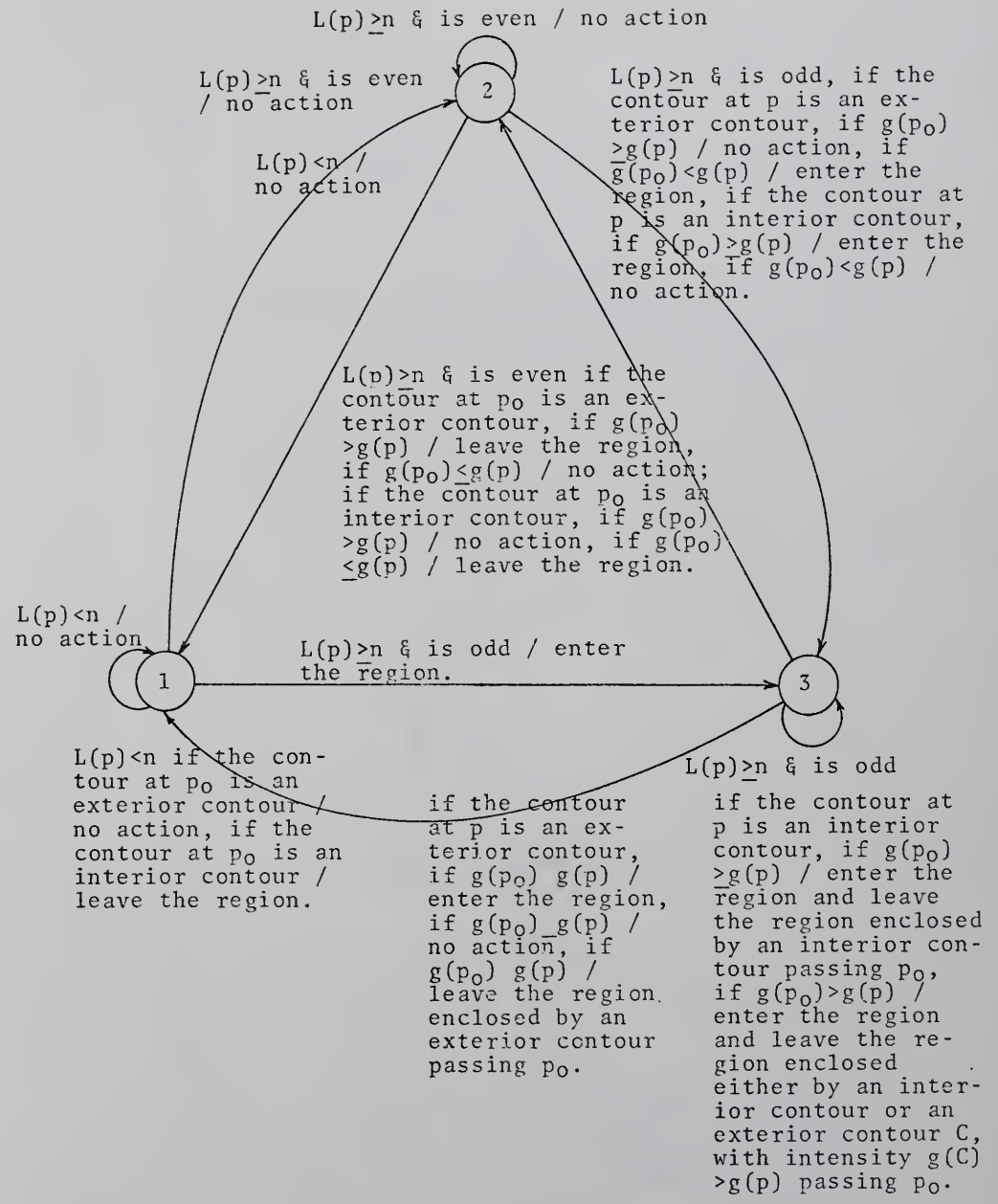


Figure 3.4. State diagram for finding the inclusion relation among contours.

contours $C_1,\ldots,C_m$. Let $C_{1'},\ldots,C_{m'}$, where the indices are ordered according to the ascending order of the corresponding label values, be the contours satisfying the condition of entering the region at the picture point p. $C_{1'},\ldots,C_{m'}$ will be put in the array in the order.

Applying this inclusion relation finding process to the contours in Figure 1.5, the Hasse graph will turn out to be the one shown in Figure 3.5.

3.4. Object Extraction by Comparison

Now we are in the final stage of extracting objects. Let H be the Hasse graph representing the inclusion relation among contours in a picture. Let the area of the region enclosed by a contour C be denoted as A (which can be found by the method presented in Section 5.1.1.). A threshold α_1 is set such that if $A < \alpha_1$, the node corresponding to C is deleted from H. It is obvious that if a contour C satisfies the above condition, all its descendants will satisfy the condition and will thus be deleted. A reasonable value for α_1 would be 9, because it is usually impossible to filter out the noise disturbing the shape of a contour if the area enclosed by the contour is less than 9. Let H' be the subgraph of H obtained by this deletion process. The H' obtained from the Hasse graph H shown in Figure 3.5 is shown in Figure 3.6.

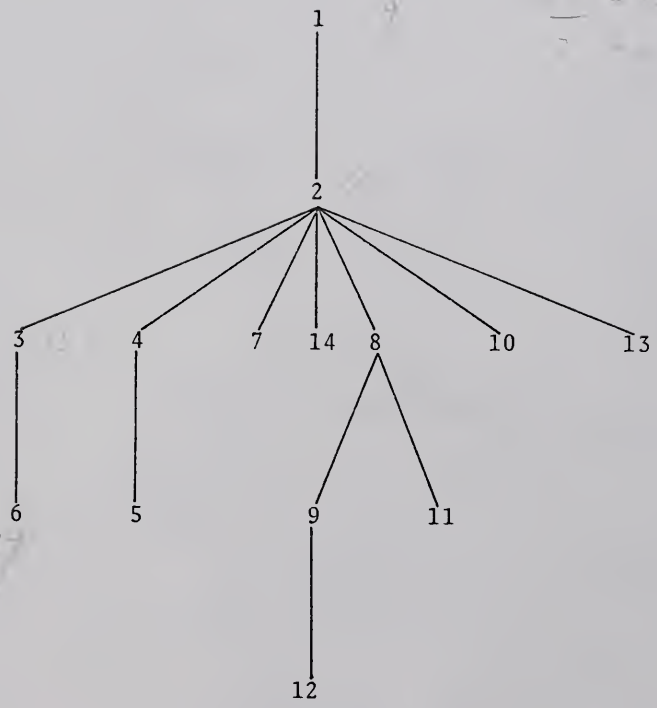


Figure 3.5. The Hasse graph representing the inclusion relation among contours in Figure 1.5.

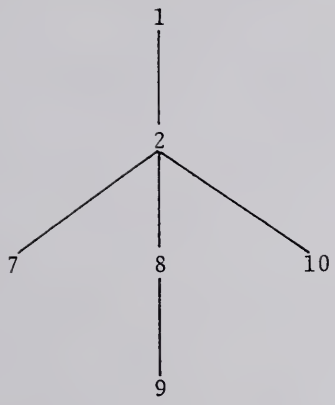


Figure 3.6. The Hasse graph obtained through the deletion of small contours.

A dissimilarity measurement between two contours C_1 and C_2 is defined as

$$D(C_1,C_2) = |M_c^{(1)}-M_c^{(2)}|+|M_p^{(1)}-M_p^{(2)}|+|M_v^{(1)}-M_v^{(2)}| ,$$

where $M_c^{(i)}$, $M_p^{(i)}$ and $M_v^{(i)}$ are the number of critical points, the number of peak points and the number of valley points on the contour C_i, i=1, 2.

A threshold α_2 is set to extract objects from H'. If a node corresponding to a contour C is the only son of a node corresponding to a contour C*, and if $D(C,C^*) < \alpha_2$, the node corresponding to the contour C is deleted from H'.

First the levels of the nodes in H' are assigned. The level of the root in H' is assigned as 1 and the levels of

all sons of nodes of level k are assigned as k+1. The deletion procedure is then applied to H' from the nodes with largest level assignment to the nodes with level 1. Let the resulting graph be denoted as H". Every node in H" can possibly correspond to the contour of an object. Figure 3.7 shows H" obtained from Figure 3.6.

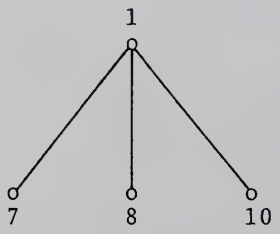


Figure 3.7. The Hasse graph obtained through the deletion of similar contours.

Contours 7 and 10 are not under consideration because they are interior contours. Contour 1, which touches the picture frame, is also not under consideration. Hence only one object, which is enclosed by contour 8, is extracted.

CHAPTER IV

GRAPH THEORY APPROACH TO PICTURE PROCESSING

In this chapter we are proposing a method to extract objects in a multi-level picture by the clustering method. This approach can detect the gestalt clusters, which are objects, in the picture and can presumably give the "skeletons" of objects. We first transfer the n-level picture into a weighted graph G and then find an MST (Minimal Spanning Tree) of every isolated weighted graph G_i of G. Based on the statistics of an MST. we can cluster an MST. Every cluster is an object in the picture. Some major paths of an MST restricted to a cluster form a "skeleton" of an object. Some graph theory backgrounds and properties of the MST will be discussed in the following section.

4.1. Some Graph Theory[29] Backgrounds and the Properties of an MST

An undirected finite graph G = {V,E,F} consists of a set V of m vertices, where V = $\{v_1,\ldots,v_m\}$, a set E of k edges, where E = $\{e_1,\ldots,e_k\}$, and a function F, a mapping from E into V and V, the set of all unordered pairs of members of V. Figure 4.1 shows an example of an undirected

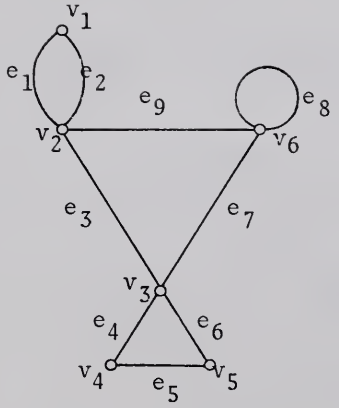


Figure 4.1. An example of an undirected finite graph.

graph. In Figure 4.1, $V = \{v_1,\ldots,v_6\}$ and $E = \{e_1,\ldots,e_9\}$. If e_j is in E, then $F(e_j) = (v_{i_1} \& v_{i_2})$, for some vertices v_{i_1} and v_{i_2} in V, such as $F(e_3) = (v_2 \& v_3)$ in Figure 4.1. An edge e_j is incident with vertices v_{i_1} and v_{i_2}, if $F(e_j) = (v_{i_1} \& v_{i_2})$. For example, in Figure 4.1, e_3 is incident with v_2 and v_3. If $F(e_j) = (v_{i_1} \& v_{i_2})$, then e_j is a loop, such as e_8 in Figure 4.1. The number $\eta(v_i)$ of edges, which are incident with a vertex v_i, is called the degree of the vertex v_i. For example, $\eta(v_2) = 4$ in Figure 4.1. v_{i_1} and v_{i_2} are adjacent vertices, if there exists an edge e_j such that $F(e_j) = (v_{i_1} \& v_{i_2})$. For example, v_1 and v_2 are adjacent vertices in Figure 4.1. Let e_{j_1} and e_{j_2} be two distinct edges. If $F(e_{j_1}) = (v_{i_1} \& v_{i_2})$, and if $F(e_{j_2}) = (v_{i_2} \& v_{i_3})$, then e_{j_1} and e_{j_2} are adjacent edges. Furthermore, if $v_{i_3} = v_{i_1}$, then e_{j_1} and e_{j_2} are parallel edges. For example,

in Figure 4.1, e_1 and e_3 are adjacent edges, and e_1 and e_2 are parallel edges. A simple graph is a graph having no loop and no pair of parallel edges. The graph, shown in Figure 4.1, is not a simple graph, because e_1 and e_2 are parallel edges, and e_8 is a loop. Figure 4.2 shows an example of a simple graph.

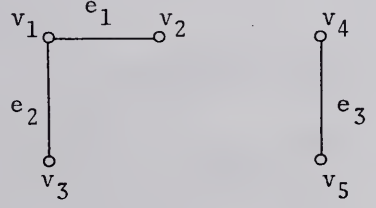


Figure 4.2. An example of a simple graph.

A graph G' = {V',E',F'} is a subgraph of a graph G = {V,E,F}

1. if V' is a subset of V and E' is a subset of E,
2. if for every e_j in E', F'(e_j) = F(e_j), and
3. if for every e_j in E', F(e_j) = (v_{i_1}&v_{i_2}), v_{i_1} and v_{i_2} are in V'.

A finite sequence of edges, $e_{j_1},\ldots,e_{j_t}$, is an edge progression (or edge sequence) of length t if there is a sequence of vertices, $v_{i_0},v_{i_1},\ldots,v_{i_t}$, such that for each $\mu = 1,\ldots,t$, $F(e_{j_\mu}) = (v_{i_{\mu-1}}\&v_{i_\mu})$. If $v_{i_0} \neq v_{i_t}$, the edge progression is open (or non-cyclic), such as e_3, e_4, e_5, e_6 in Figure 4.1. If $v_{i_0} = v_{i_t}$, the edge progression is

closed (or cyclic), such as e_3, e_1, e_2, e_3 in Figure 4.1. An edge progression is said to be from v_{i_0} to v_{i_t}; v_{i_0} is the initial vertex and v_{i_t} is the terminal vertex of the progression. For $\mu = 1,\ldots,t-1$, v_{i_μ} is an intermediate vertex of the progression. A chain progression (or non-cyclic path) is an open edge progression in which no edge is repeated in the sequence, such as e_3, e_2, e_1 in Figure 4.1. A circuit progression (or cyclic path) is a closed edge progression in which no edge is repeated in the sequence, such as e_3, e_4, e_5, e_6, e_7, e_9 in Figure 4.1. A simple chain progression (or simple path or arc) is a chain progression in which no vertex is repeated in the vertex sequence, such as e_3, e_2 in Figure 4.1. A simple circuit progression (or circuit) is a circuit progression in which $v_{i_0} = v_{i_t}$ but there is no other duplication of any vertex in the vertex sequence, such as e_3, e_7, e_9 in Figure 4.1.

Let v_{i_0} and v_{i_t} be two vertices of a graph G, v_{i_0} and v_{i_t} are connected vertices if $v_{i_0} = v_{i_t}$ or if there exists an edge progression, $e_{j_1},\ldots,e_{j_t}$ with vertex sequence v_{i_0}, $v_{i_1},\ldots,v_{i_t}$. The existence of an edge progression from v_{i_0} to v_{i_t} implies the existence of an arc from v_{i_0} to v_{i_t}, so a pair of distinct vertices is connected if and only if there is an arc joining them. G is a connected graph if for any vertices v_i and v_j in V, v_i and v_j are connected.

A mxm matrix $A = (a_{i_j})$ can be defined such that

$$a_{i_j} = \begin{cases} 1 \text{ if } i=j \text{ or if } v_i \text{ and } v_j \text{ are adjacent vertices} \\ 0 \quad \text{otherwise.} \end{cases}$$

For example, the A matrix for the graph shown in Figure 4.2 is

$$A = \begin{pmatrix} 11100 \\ 11000 \\ 10100 \\ 00011 \\ 00011 \end{pmatrix}$$

For some integer S such that $A^S = A^{S+1}$, the matrix $A^S = (a_{i_j}^{(s)})$ is called the connection matrix in that $a_{i_j}^{(s)} = 1$ if and only if the vertices v_i and v_j are connected. For example, the connection matrix of the graph shown in Figure 4.2 is

$$A^2 = \begin{pmatrix} 11100 \\ 11100 \\ 11100 \\ 00011 \\ 00011 \end{pmatrix}$$

The connectivity relation C on the vertices of a graph is an equivalence relation. Let the partition of V, by the connectivity relation, be q equivalence classes, $\{V_1,\ldots, V_q\}$. For example, $V_1 = \{v_1,v_2,v_4\}$ and $V_2 = \{v_3,v_5\}$ in Figure 4.1. Let E_i be the subset of E each of which is incident with vertices in V_i. For any $i \neq j$, there does not exist an edge e_t in E such that it joins a vertex in V_i and a vertex in V_j. Thus, if e_t is in E_i and $F(e_t) = (v_{i_1} \& v_{i_2})$, then both v_{i_1} and v_{i_2} are in V_i. Hence $\{E_1,\ldots,E_q\}$ is a partition of E. For example, $E_1 = \{e_1,e_2\}$ and $E_2 = \{e_3\}$ in Figure 4.2. Therefore, $G_i = (V_i,E_i,F_i)$, where F_i is the restriction of F to E_i, defines a subgraph of G. Each such G_i is obviously connected. If G_i is a subgraph

of a connected subgraph G' of G, then $G_i = G'$; that is, G_i is a maximal connected subgraph of G and is called an isolated component of G.

Definition 4.1.--A tree is a connected graph having no circuit. A circuit-free graph having q connected components is a forest of q trees.

If T = {V,E,F} is a tree and e is an edge of T, then the subgraph G = $\{V,E-(e), F_{E-(e)}\}$ of T is disconnected, where $F_{E-(e)}$ implies the function F restricted on the domain E-(e). Hence no subgraph derived from a tree, which has all the vertices and lesser number of edges, is connected. Thus a tree is a minimal connected graph.

Definition 4.2.--Let G = {V,E,F} be a connected graph, and let v_i and v_j be two distinct vertices in V. The distance $d(v_i,v_j)$ between v_i and v_j is defined as the minimum length of the arcs from v_i to v_j. If $v_i = v_j$, $d(v_i,v_j)$ is defined equal to 0.

The distance function defined above satisfies the metric axioms:

1. $d(v_i,v_i) = 0$,
2. $d(v_i,v_j) = d(v_j,v_i)$, and
3. $d(v_i,v_t) \leq d(v_i,v_j) + d(v_j,v_t)$ $\forall$ $v_i,v_j,v_t \in V$.

Definition 4.3.--Let T = {V,E,F} be a tree and v_i is a vertex in T. If $\eta(v_i) = 1$, vertex v_i is termed as a leaf of the tree T. An arc $a(v_i,v_j)$ from v_i to v_j is called a diametral path when its length ℓ is maximal among the

distances between any two vertices; ℓ is the diameter of the tree T. A vertex c in V is a center of T if

$$r(c) = \min_{v_i \in V} \{r(v_i)\} = r_0$$

where $r(v_i)$ is defined as $\max_{v_j \in V}\{d(v_i,v_j)\}$. r_0 is called the radius of T. Let v_i be a leaf of T. The longest arc from v_i is called a major path from v_i.

The following theorem reveals the properties of the centers of a tree.

Theorem 4.1.--Let T be a tree of diameter ℓ and $a(v_{i_0}, v_{i_\ell})$ be a diametral path, having the corresponding sequence of vertices $v_{i_0}, v_{i_1}, \ldots, v_{i_\ell}$. When ℓ is even T has a single center $c = v_{i_{(\ell/2)}}$ and has a radius $r_0 = \ell/2$. All major paths go through c. When ℓ is odd T has two centers, $c_1 = v_{i_{(\ell-1)/2}}$ and $c_2 = v_{i_{(\ell+1)/2}}$, and has a radius $r_0 = (\ell+1)/2$. All major paths pass through both centers.

Definition 4.4.--Let G = {V,E,F} be a connected graph, and T = $\{V_T,E_T,F_T\}$ be a tree and a subgraph of G. If $V_T = V$, then T spans G. T is termed a spanning tree of G.

Definition 4.5.--A weighted graph G = {V,E,F,W} is a graph {V,E,F} with the assignment of a weight to each edge in E. W is the weight function which maps E into real numbers, that is, the weight of an edge e_j is $W(e_j)$. The weight of an edge progression $e_{j_1}, \ldots, e_{j_t}$ is defined equal to $W(e_{j_1})+\ldots+W(e_{j_t})$. The weight of G is defined equal to

the sum of the weights of all edges in G. Let $T = \{V_T, E_T, F_T, W_T\}$, where W_T is the restriction of W on E_T, be a spanning tree of G, which is also connected. T is said to be a Minimal Spanning Tree (MST) of G if the weight of T is minimal among all spanning trees of G. Figure 4.3 shows an example of a weighted graph G. Figure 4.4 shows the corresponding MST of G.

Definition 4.6.--Let $\{V_1, V_2\}$ be a partition of the vertex set V of a weighted graph $G = \{V,E,F,W\}$. The weight $W(V_1,V_2)$ across the partition is defined as the smallest weight among all edges which join one vertex in V_1 and the other in V_2. The set of edges $E(V_1,V_2)$ which span a partition will be referred to as the cut set of $\{V_1,V_2\}$ and a link is any edge in $E(V_1,V_2)$ whose weight is equal to the weight $W(V_1,V_2)$. The set of all links in $E(V_1,V_2)$ is called link set $L(V_1,V_2)$ of $\{V_1,V_2\}$.

The following theorem allows us to find an MST of a weighted graph from the link sets.

Theorem 4.2.--An MST contains at least one edge from the link set $L(V_1,V_2)$ of every partition $\{V_1,V_2\}$. Every edge of an MST is a link of some partition of V.

Theorem 4.3 reveals that the appropriate clusters can be found as subtrees of any MST.

Theorem 4.3.--If V_i is a non-empty subset of V with the property $W(V_{i_1},V_{i_2}) < W(V_i,V-V_i)$ for all partitions

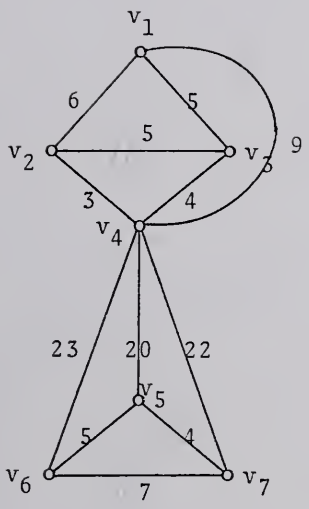


Figure 4.3. An example of a weighted graph.

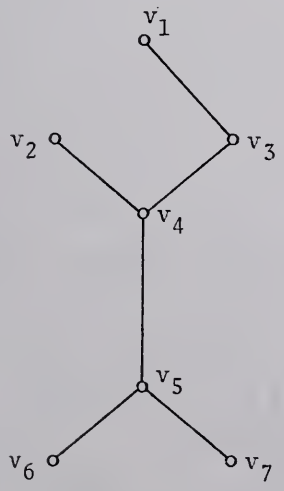


Figure 4.4. An MST of the weighted graph shown in Figure 4.3.

$\{V_{i_1}, V_{i_2}\}$ of V_i, then the restriction of any MST to V_i forms a subtree of the MST.

4.1.1. Finding an MST of a Weighted Graph

From Theorem 4.2, it is clear that an MST can be found from a connected graph by building up a subtree T', to which a link of $\{V_{T'}, V_G - V_{T'}\}$ is added. Let m be the total number of vertices of G. We can set $V_G = \{v_1, \ldots, v_m\}$. Three arrays are required to achieve the purpose of finding an MST from a weighted graph G.[31]

1. Vertex array X: It indicates which vertices are in $V_{T'}$, that is, if X(i) = 1, then $v_i \in V_{T'}$, while if X(i) = 0, then $v_i \notin V_{T'}$.
2. Reference array R: If X(i) = 1, R(i) specifies the index of the vertex v_j in $V_{T'}$, if v_i is adjacent to v_j in T'. If X(i) = 0, R(i) specifies the index of vertex v_j in $V_{T'}$, such that $W(v_i, v_j) = \min_{v_q \in V_{T'}} \{W(v_i, v_q)\}$, where $W(v_i, v_q)$ is the weight of the edge joining vertices v_i and v_q.
3. Weight array Z: Z(i) is equal to the weight of the edge incident with v_i and $v_{R(i)}$, that is, $Z(i) = W(v_i, v_{R(i)})$.

The link of $\{V_{T'}, V_G - V_{T'}\}$ can be found from the vertex array X and the weight array Z by noting that the edge connecting a vertex v_i and $v_{R(i)}$, where v_i is not in the subtree T' of MST (i.e., X(i) = 0), has a weight Z(i) equal to $\min_{j=\{1,\ldots,m\}} \{W(j) \mid X(j) = 0\}$. The edge is then added to T' by setting X(R(i)) = 1. Figure 4.5 shows the flow diagram for finding an MST of a connected weighted graph.

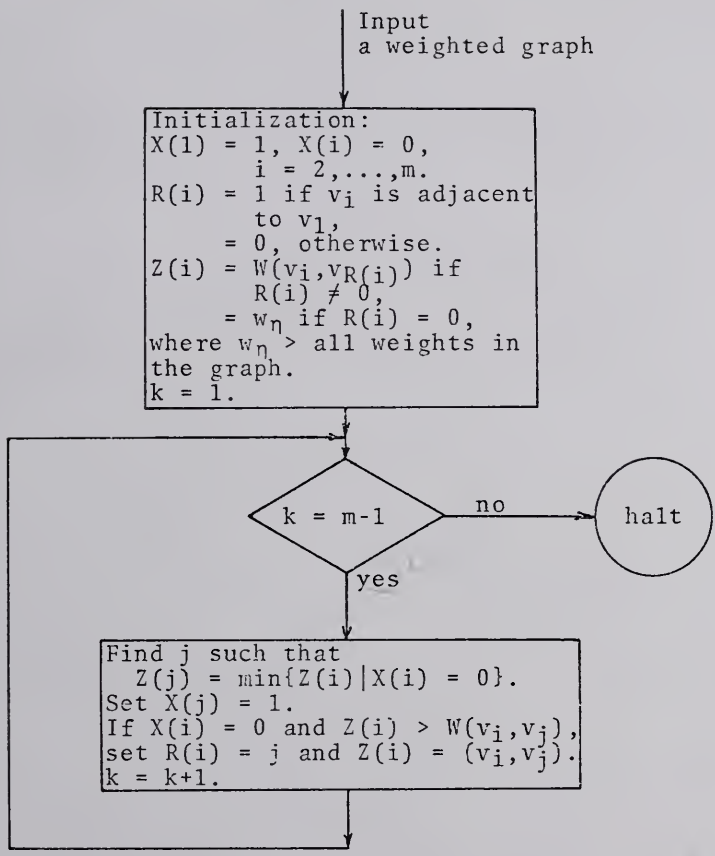


Figure 4.5. Flow diagram for finding an MST of a weighted graph.

4.1.2. Finding Major Paths

As is easily seen, the most straightforward clustering can be done by setting a threshold w, such that if an edge in an MST is of weight greater than w, the edge is deleted from the MST. The deletion of heavily weighted edges from an MST will yield a forest of subtrees. Every subtree corresponds to a cluster of the connected weighted graph G. It should be pointed out that an MST of a connected weighted graph is not unique. From Theorem 4.3, it is known that this non-uniqueness of the MSTs does not restrict the formation of the same clusters from different MSTs of a graph. In many cases, more sophisticated consideration should be taken to cluster the graph. Statistics of the weights of edges on major paths should be taken into consideration. For a tree, usually there exist many major paths. A systematic method should be set to find all major paths. Well-distinguished major paths are of interest. Two major paths can be considered as well distinguished if they have only a small portion of paths in common. Also, the branches from a diametral path are of interest. Since a tree is a simple graph, any arc in a tree can be represented by a sequence of vertices. The following are useful definitions.

Definition 4.7.--Let T_1 be a tree of diameter ℓ, which is even. There is only one center c in T_1. The arc a(v,c) from a leaf v to the center c is called a radial path. If $d(v,c) = \ell/2$, a(v,c) is called a maximal radial path. Let

T_2 be a tree of diameter ℓ, which is odd. There are two centers, c_1 and c_2, in T_2. For any leaf v, if $d(v,c_i) > d(v,c_j)$, where $i \neq j$, the arc from v to c_i, $a(v,c_i)$, is called a radial path. If $d(v,c_i) = (\ell+1)/2$, $a(v,c_i)$ is called a maximal radial path. Let $a(v_i,v_j)$ and $a(v_t,v_j)$ be two arcs, where both v_i and v_t are leaves, and v_j be the only common vertex of the two arcs. If $d(v_i,v_j) \leq d(v_t,v_j)$, then $a(v_i,v_j)$ is called a branch and any arc containing $a(v_t,v_j)$ is called a stem. v_j is called a branching vertex.

A relation R is defined on all radial paths in a tree T. Let s_1 and s_2 be two radial paths in T. If s_1 and s_2 contain a common subsequence of more than one vertex, then $s_1 R s_2$. It is obvious that R is an equivalence relation. Let T be a tree having only one center c and $v_{i_1},\ldots,v_{i_t}$ be all vertices adjacent to c. Every radial path has a subsequence of vertices (v_{i_j},c) for some vertex v_{i_j}, $j=1,\ldots,t$. Hence there are t equivalence classes of radial paths induced by the equivalent relation R. Let S_j denote all radial paths having a subsequence of vertices (v_{i_j},c). Any radial path of S_j can be combined with any maximal-radial path of $S_{j'}$ to form a major path, where $j \neq j' \in \{1,\ldots,t\}$. Let $s_u^{(j)} = (v_{u_1}^{(j')},v_{u_2}^{(j')},\ldots,v_{i_{j'}},c)$ be a maximal-radial path in $S_{j'}$. $(v_{u_1}^{(j)},v_{u_2}^{(j)},\ldots,v_{i_j},c,v_{i_{j'}},\ldots,v_{u'_2}^{(j')},v_{u'_1}^{(j')})$ then forms a major path. If T is a tree having two centers, c_1 and c_2, every radial path S in T either contains the subsequence of vertices (c_1,c_2), or contains the subsequence

of vertices (c_2,c_1). Hence there are two equivalence classes induced by the relation R. Let $S_u = (v_{u_1},\ldots,c_i,c_j)$ be a radial path and $S_{u'} = (v_{u'_1},\ldots,v_{u'_{(\ell-1)/2}},c_j,c_i)$ be a maximal radial path. S_u and $S_{u'}$ can be combined into a major path $(v_{u_1},\ldots,c_i,c_j,v_{u'_{(\ell-1)/2}},\ldots,v_{u'_1})$.

Because of less storage required and easy combination into major paths, the storage structure of radial paths would be that only one radial path, which has maximum length among all radial paths in the same equivalence class, of every equivalence class is stored in the full sequence. Any other radial paths are stored as branches.

Every leaf in a tree initializes a sequence. Trim all the leaves from the tree. If the adjacent vertex v_i of a leaf v_j does not turn out to be a leaf after v_j is trimmed, the corresponding sequence will represent a branch. The sequence having v_i as a leaf will be the corresponding stem. The procedure is iterated until either there are only two vertices left or there is only one vertex left. The vertices finally left are the centers of the tree. Figure 4.6 shows the flow diagram for finding radial paths and centers in a tree.

The tree shown in Figure 4.4 has only one center v_4. The diameter of the tree is 4. The set of radial paths, having full sequences, is $\{(v_1,v_3,v_4),(v_7,v_5,v_4)\}$. (v_6,v_5) is a branch depending on the maximal radial path (v_7,v_5,v_4). (v_2,v_4) is a branch having (v_1,v_3,v_4) and (v_7,v_5,v_4) as

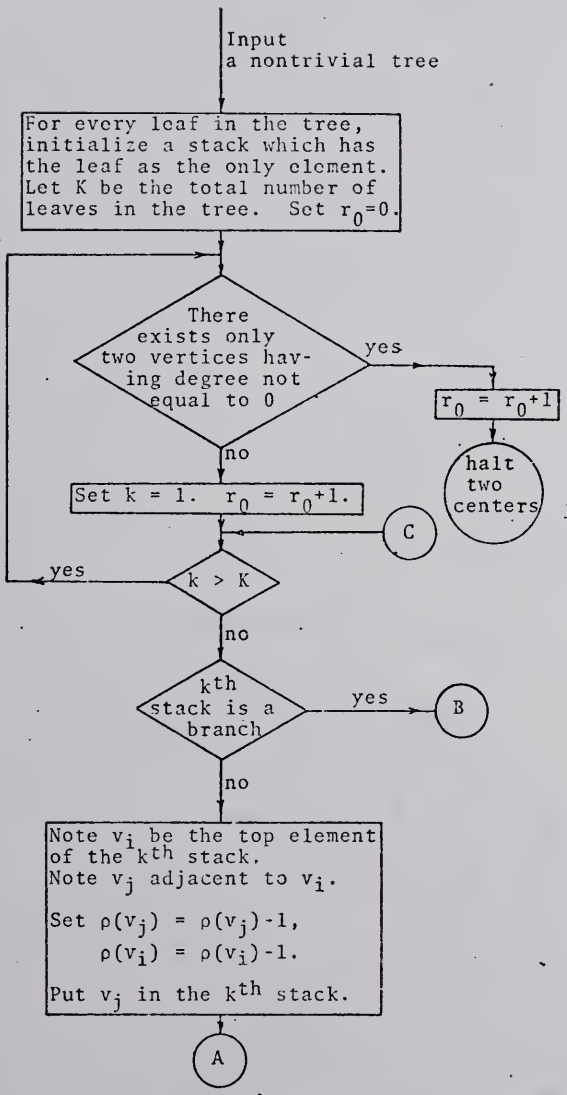


Figure 4.6. Flow diagram for finding radial paths and centers in a tree.

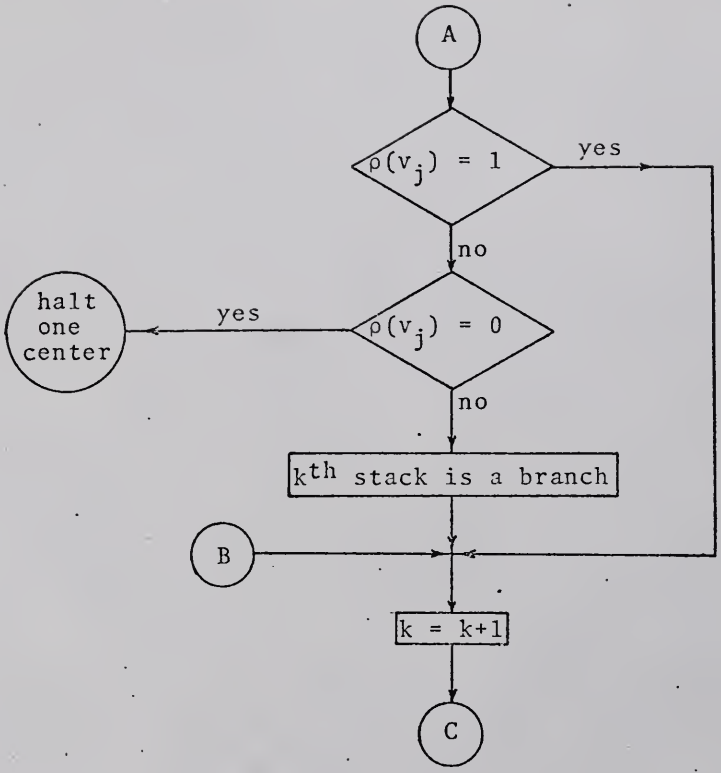


Figure 4.6. Continued

stems. (v_1,v_3,v_4) and (v_7,v_5,v_4) can be combined into a diametral path (v_1,v_3,v_4,v_5,v_7). From the branch (v_6,v_5) we can find a maximal radial path (v_6,v_5,v_4) which is dependent on the maximal radial path (v_7,v_5,v_4). (v_1,v_3,v_4) and (v_6,v_5,v_4) can be combined into a diametral path (v_1,v_3,v_4,v_5,v_6). (v_2,v_4) is independent of any other radial paths in the tree, hence we can combine

1. (v_2,v_4) with (v_1,v_3,v_4) to a major path (v_2,v_4,v_3,v_1),
2. (v_2,v_4) with (v_7,v_5,v_4) to a major path (v_2,v_4,v_5,v_7), and
3. (v_2,v_4) with (v_6,v_5,v_4) to a major path (v_2,v_4,v_5,v_6).

4.2. The Representation of a Digitized Picture by a Weighted Graph

A digitized picture g can be represented by a weighted graph G in the following manner. Every picture point is considered as a vertex. Several possible methods are used to define the connection of vertices and the weight of the corresponding edges.

1. Method 1 is that every picture point p is connected to any of the four-neighboring picture points p' with a weight $1/(g(p)+g(p'))$ except that $g(p)+g(p') = 0$.
2. Method 2 is that every picture point p is connected to any of the eight-neighboring picture points p' with a weight of $d(p,p')/(g(p)+g(p'))$ except that $g(p)+g(p') = 0$.
3. Method 3 is that every picture point p is connected to the 16-neighboring picture points p' with a weight of

d(p,p')/(g(p)+g(p')) except that g(p)+g(p') = 0. Figure 4.7 shows the three methods for connecting nodes.

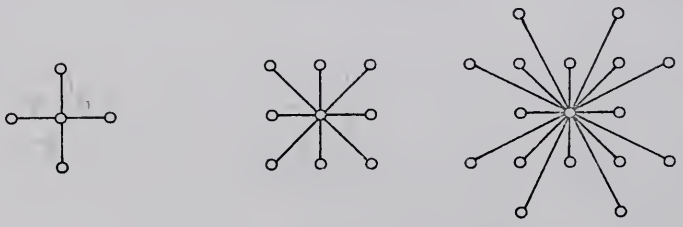


Figure 4.7. Three methods for connecting picture points.

Because the MST will be invariant under the monotone transformation of the weights, a quantizer can be used to transfer monotonically the weights assigned by the previous paragraph to a subset W of the natural numbers. Let g be a n-level picture. It is easily seen that for method 1, the cardinal number of W is 2n-3; for method 2, the cardinal number of W is 4n-5; and for method 3, the cardinal number of W is 6n-7.

The weighted graph G, representing a multi-level picture, is not necessarily a connected graph. Once the weighted graph G of a digitized picture is found, the flow diagram shown in Figure 4.1 can be applied to G to find an MST. The edges in the MST found, having weights equal to

the initial weight w_η, are deleted. Hence the result would be a minimal spanning forest.

For the same reason of encoding a curve by the chain code (discussed in detail in Chapter II), chain codes are used to represent edges, instead of using a pair of vertices or assigning a name to an edge. In method 1, from a vertex to another vertex, there are only four possible edges, hence a 2-bit chain code is used to encode edges. In method 2, from a vertex to another vertex, there are only eight possible edges, hence an octal chain code is used to encode the edges. In method 3, there are only 16 possible edges from a vertex to another vertex, hence a hexadecimal chain code is used to encode the edges.

CHAPTER V

FEATURE EXTRACTION

As stated in Chapter I, feature extraction strongly depends on the type of pictures handled. Here in this chapter, we are discussing only area pictures. There are mainly two types of features: local features and global features. Local features are the ones which are dependent on the individual objects and are independent of any other objects, for example, area, centroid, shape of object, principal axis direction, elongation index, etc. Global features are the ones which describe the interrelationship of objects in the picture, for example, the inclusion relation of objects in the picture, overlapping, touching of objects in the picture, the distribution of objects in the picture, etc. Assume that the objects are described by the boundary contours. The boundary contours are described by the octal chain codes. An object which has one exterior contour C_0 and q interior contours $C_1,\ldots,C_q$ can be described by $R_{C_0} - \bigcup_{j=1}^{q} R_{C_j}$, where R_{C_j} is the region enclosed by the contour C_j, $j = 0,\ldots,q$, where - is the substraction in the set theory and U is the union.

5.1. Some Fundamental Local Features

The most important and useful local features are areas, centroids, shapes, major principal axis directions and the corresponding elongation index of objects. We are attacking the problems one by one. Remember that the boundary contours are described by the start points and the associated sequences of octal chain codes.

5.1.1. Area

Since the contours are encoded into a sequence of straight lines, the area inside the contour is the algebraic sum of areas of the strips between the line segments and the x axis. Figure 5.1 shows the x-y coordinate system and the eight possible line segments with the corresponding octal chain codes.

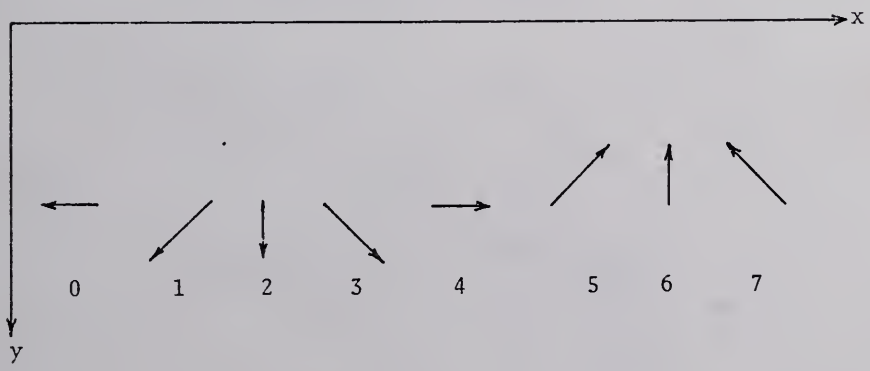


Figure 5.1. X-y coordinate system and the eight possible line segments with the corresponding octal chain codes.

The following is a table of the area a_i of the strip r_i between a segment, corresponding to the octal chain code s_i which starts at (x_i,y_i), and the x axis, and the increment Δx_i and Δy_i in the x and y coordinates, respectively. From Table 5.1 we can get $a_i = \Delta x_i(y_i + \Delta y_i/2)$. The sign of the area enclosed by a contour is positive if the contour is encoded in the clockwise direction and is negative otherwise. The area of an object which is represented by $R_{C_0} - \bigcup_{j=1}^{q} R_{C_j}$ is then equal to $|\text{area}(R_{C_0})| - \sum_{j=1}^{q} |\text{area}(R_{C_j})|$. The ratio $|\text{area}(R_{C_j})|/|\text{area}(R_{C_0})|$ is a normalized measurement of the size of the hole enclosed by the interior contour C_j, $j=1,\ldots,q$.

Table 5.1

Table of area, x-increment and y-increment

Octal chain code s_i	Area a_i	x-increment Δx_i	y-increment Δy_i
0	$-y_i$	-1	0
1	$-(y_i+1/2)$	-1	1
2	0	0	1
3	$y_i+1/2$	1	1
4	$1+y_i$	1	0
5	$y_i-1/2$	1	-1
6	0	0	-1
7	$-(y_i-1/2)$	-1	-1

5.1.2. Centroid

The centroid of a region enclosed by a contour C can be found by the mean value technique. The y coordinate of the centroid is $\bar{y} = (\sum_{i=1}^{m} a_i \bar{y}_i)/A$, where a_i is the area of the strip between the corresponding segment and the x axis, $\bar{y}_i$ is the mean y coordinate of the strip, m is the length of the sequence representing C and $A = \sum_{i=1}^{m} a_i$ is the area of the region. Table 5.2 gives the $a_i \bar{y}_i$ values for the eight possible segments. A formula can be derived, $a_i \bar{y}_i = \Delta x_i(y_i^2 + \Delta y_i(y_i + \Delta y_i/3))/2$.

Table 5.2

Table of moment

Octal chain code s_i	Moment of a_i about the x-axis $a_i \bar{y}_i$	Moment of a_i' about the y-axis $a_i' \bar{x}_i'$
0	$-y_i^2/2$	0
1	$-(y_i^2 + y_i + 1/3)/2$	$(x_i^2 - x_i + 1/3)/2$
2	0	$x_i^2/2$
3	$(y_i^2 + y_i + 1/3)/2$	$(x_i^2 + x_i + 1/3)/2$
4	$y_i^2/2$	0
5	$(y_i^2 - y_i + 1/3)/2$	$-(x_i^2 + x_i + 1/3)/2$
6	0	$-x_i^2/2$
7	$-(y_i^2 - y_i + 1/3)/2$	$-(x_i^2 - x_i + 1/3)/2$

The x coordinate of the centroid is $\bar{x} = \sum_{i=1}^{m} a_i' \bar{x}_i'/A'$, where a_i' is the area of the strip between the corresponding segment and the y axis, $\bar{x}_i'$ is the mean x coordinate of the strip and $A' = \sum_{i=1}^{m} a_i'$ is the area of the region enclosed by the contour C which has opposite sign from that of A. By symmetry a formula can be derived,

$$a_i' \bar{x}_i' = \Delta y_i(x_i^2 + \Delta x_i(x_i + \Delta x_i/3))/2.$$

For an object which has holes, the net centroid is

$$\bar{x} = (|A_{C_0}|\bar{x}_{C_0} - \sum_{j=1}^{q} |A_{C_j}|\bar{x}_{C_j})/A$$

and

$$\bar{y} = (|A_{C_0}|\bar{y}_{C_0} - \sum_{j=1}^{q} |A_{C_j}|\bar{y}_{C_j})/A,$$

where A_{C_j} is the area, $(\bar{x}_{C_j}, \bar{y}_{C_j})$ is the centroid of the region enclosed by the contour C_j, $j=0,1,\ldots,q$ and A is the net area of the objects.

5.1.3. Shape

The shape of an object can be described by the shape of the exterior boundary contour of the object and the shapes of interior boundary contours of the object if they exist. A method to describe the shape of a contour is then required. Let us consider a contour in the real plane first. Second order differentiation can very well describe the shape of the contour because it can partition the contour into concave and convex portions. In the digitized picture, the

octal chain code describes the first order differentiation of the curve. The difference $c \div c'$ of two successive octal chain codes c and c', which is defined in Section 3.2, i.e., $c \div c' = c - c' + 8p$, where p is an integer such that the absolute value of the difference is less than or equal to 4, can describe the second order differentiation.

Let $c_0, c_1, \ldots, c_{m-1}$ be the sequence of octal chain codes representing the contour C and $X = x_0, \ldots, x_{m-1}$, where $x_i = c_{i+1} - c_i$, $i = 0, 1, \ldots, m-1$ and $c_m = c_0$ be the corresponding difference sequence. Because of the quantization error and noise in the picture, high frequency noise appears in the X sequence. A digital filter[32] is then required to filter out this high frequency noise. A Hanning window function W, which is defined as

$$w_k = 1 - \cos(2\pi k/K), \quad 0 \leq k \leq K,$$

is used here to convolve the input sequence X to filter out the high frequency noise. The output Y of the digital filter W is then

$$y_{(i+K/2)_{\text{mod } m}} = \sum_{j=0}^{K} w_j \, w_{(i=j)_{\text{mod } m}},$$

$i = 0, \ldots, m-1$. The output sequence Y is called the smoothed difference sequence.

The change of signs between successive y_i's in the Y sequence is important to note. If the value of y_i is positive, the i^{th} point in the contour is in the concave portion

of the contour. If the value of y_i is negative, the i^{th} point in the contour is in the convex portion of the contour. The Y sequence is periodic because the last element in Y is one ahead of the first element in Y. We can thus group the Y sequence into a sequence of t subsequences $Y_0,\ldots,Y_{t-1}$ such that $Y = Y_0 \ldots Y_{t-1}$, The signs of all y_i values in a subsequence are the same and the signs of y_i values in a subsequence are different from those of y_i values in an adjacent subsequence.

Let $N = (n_0,\ldots,n_{t-1})$ be the sequence such that the n_i^{th} point in the contour is the start point of the Y_i portion of the contour, i = 0,...,t-1. We call these points the critical points. It is noted that

$$Y_i = (y_{n_i}, y_{(n_i+1)_{\text{mod } m}}, \ldots, y_{(n_{(i+1)_{\text{mod } e}}-1)_{\text{mod } m}}),$$

i = 0,...,t-1.

A sign specification σ should be accompanied with the N sequence such that $\sigma = \text{sgn}(y_{n_0})$. From the value of σ we can tell that the portion of the contour from the n_i^{th} point to the $n_{(i+1)_{\text{mod } t}}^{th}$ point, i = 0,...,t-1, is a concave portion or a convex portion. If σ = +1 and if i is even (odd), it implies that $\text{sgn}(y_{n_i}) = +1$ (-1). Hence the portion of the contour from the n_i^{th} point to the $n_{(i+1)_{\text{mod } t}}^{th}$ point is a concave (convex) one. If σ = -1, the situation is just the opposite to that of σ = +1.

The window size K of the Hanning filter is chosen such that if $K' \geq K$ is used as the window size, then for every contour C in the picture, any two successive critical points are within δ length along the contour δ, where $4 \leq \delta \leq 9$ is a threshold, and if $K' < K$ is used as the window size, then there exists a contour C having the N sequence (if $t \neq 1$) such that $(n_i - n_{i-1})_{\text{mod } m} \leq \delta$ for some $1 \leq i \leq t$, where m is the length of the contour C.

Let the i^{th} contour point have a local maximum $|y_i|$ value and $|y_i| > \alpha$. The i^{th} contour point is said to be an extreme point. If $y_i > 0$, then the i^{th} contour point is on a concave portion. Hence the i^{th} contour point is also called a valley point. If $y_i < 0$, the i^{th} contour point is also called a peak point.

If there is one or less extreme point between two successive critical points, the $n_{i-1}{}^{th}$ and the $n_i{}^{th}$ points, the curvature of the portion of the contour between the successive critical points can well be defined as $\text{sgn}(y_{n_{i-1}})((n_i - n_{i-1})_{\text{mod } m}/d_i(n_i, n_{i-1})-1)$, where $d(n_i - n_{i-1})$ is the Euclidean distance between $n_{i-1}{}^{th}$ and the $n_i{}^{th}$ points.

It should be pointed out that the Euclidean distance between the starting point and the end point of a curve can be found without the specification of the locations of the starting point and the end point, if the octal chain code sequence of the curve is known. Let $c_1 \ldots c_m$, be the

octal chain code sequence of a curve C. We can calculate the x(y)-projection of the vector from the start point of C to the end point of C by summing up the x(y) increments contributed by the octal chain codes as defined in Table 5.1, i.e., $\Delta X = \sum_{i=1}^{m'} \Delta x_i$, $\Delta Y = \sum_{i=1}^{m'} \Delta y_i$. The Euclidean distance between the start point and the end point of the curve C is then $\Delta X^2 + \Delta Y^2 = (\sum_{i=1}^{m'} \Delta x_i)^2 + (\sum_{i=1}^{m'} \Delta y_i)^2$.

5.1.4. Principal Axis Direction and Elongation Index

If a contour C has no very deep concavity, we can assume that the region enclosed is an elliptical one. Generally there are two principal axes of the region through the centroid. In order to find the principal axes, we have to find I_x, I_y, the moments of inertia of the region about the x and y axes, respectively, and P_{xy}, the product of inertia of the region. Here the x-y coordinate system is the translation of the old x-y coordinate system obtained by setting the centroid of the region as the new origin. Note that both I_x and I_y are positive and are given by $I_x = \sum_{i=1}^{n} a_i \overline{y_i^2}$ and $I_y = \sum_{i=1}^{n} a_i' \overline{x_i'^2}$, where $a_i \overline{y_i^2}$ is the moment of inertia of a_i about the y-axis and $a_i' \overline{x_i'^2}$ is the moment of inertia of a_i about the x-axis. $P_{xy} = \text{sgn}(A) \sum_{i=1}^{n} a_i \overline{x_i y_i}$, where $a_i \overline{x_i y_i}$ is the product of inertia of a_i.

Table 5.3 shows the $a_i \overline{y_i^2}$, $a_i' \overline{x_i'^2}$ and $a_i \overline{x_i y_i}$ values. Formulas for those calculations can be derived:

Table 5.3

Table of moment of inertia and product of inertia

Octal chain code s_i	Moment of inertia of a_i about the x-axis $a_i\overline{y_i}^2$	Moment of inertia of a_i about the y-axis $a_i\overline{x_i}^2$	Product of inertia of a_i $a_i\overline{x_i y_i}$
0	$-y_i^3/3$	0	$-y_i^2(2x_i-1)/4$
1	$-(4y_i^3+6y_i^2+4y_i+1)/12$	$(4x_i^3-6x_i^2+4x_i-1)/12$	$-y_i^2(2x_i-1)/4-x_iy_i/2+y_i/3-x_i/6+1/8$
2	0	$x_i^3/3$	0
3	$(4y_i^3+6y_i^3+4y_i+1)/12$	$(4x_i^3+6x_i^2+4x_i+1)/12$	$y_i^2(2x_i+1)/4+x_iy_i/2+y_i/3+x_i/6+1/8$
4	$y_i^3/3$	0	$y_i^2(2x_i+1)/4$
5	$(4y_i^3-6y_i^2+4y_i-1)/12$	$-(4x_i^3+6x_i^2+4x_i+1)/12$	$y_i^2(2x_i+1)/4-x_iy_i/2-y_i/3+x_i/6+1/8$
6	0	$-x_i^3/3$	0
7	$-(4y_i^3-6y_i^2+4y_i-1)/12$	$-(4x_i^3-6x_i^2+4x_i-1)/12$	$-y_i^2(2x_i-1)/4+x_iy_i/2-y_i/3-x_i/6+1/8$

$$a_i\overline{y_i^2} = \Delta x_i(4y_i^3 + \Delta y_i(6y_i^2 + 4(\Delta y_i)y_i + 1))/12,$$

$$a_i\overline{x_i^2} = \Delta y_i(4x_i^3 + \Delta x_i(6x_i^2 + 4(\Delta x_i)x_i + 1))/12, \text{ and}$$

$$a_i\overline{x_i y_i} = \Delta x_i(y_i^2(2x + \Delta x_i))/4 + (\Delta x_i \Delta y_i)x_i y_i/2$$
$$+(\Delta x_i)(\Delta y_i)y_i/3 + (\Delta y_i)(\Delta x_i)x_i/6 + 1/8.$$

Let θ be the angle of rotation from the x,y axes to the two principal axes x' and y', respectively, as shown in Figure 5.2.

A formula[33] has been derived such that $\tan 2\theta = -2P_{xy}/(I_y - I_x)$. Hence we can find this angle of rotation very easily. A major principal axis is defined as the axis about which the moment of inertia is minimum. Hence, in order to find the major principal axis, we have to calculate $I_{x'}$ and $I_{y'}$, and to find out which has less value. The formula for $I_{x'}$ and $I_{y'}$ are:

$$I_{x'} = (I_x + I_y)/2 + ((I_x - I_y)/2) * \cos 2\theta - P_{xy} \sin 2\theta$$

and $$I_{y'} = (I_x + I_y)/2 - ((I_x - I_y)/2) * \cos 2\theta + P_{xy} \sin 2\theta.$$

The major principal direction can be represented by θ or $\theta + \pi/2$ depending on x' or y' being the major principal axis, respectively. The ratio of the moment of inertia about the major principal axis to the moment of inertia about the minor principal axis is called the elongation index. The elongation index ranges from 0 to 1. It indicates how

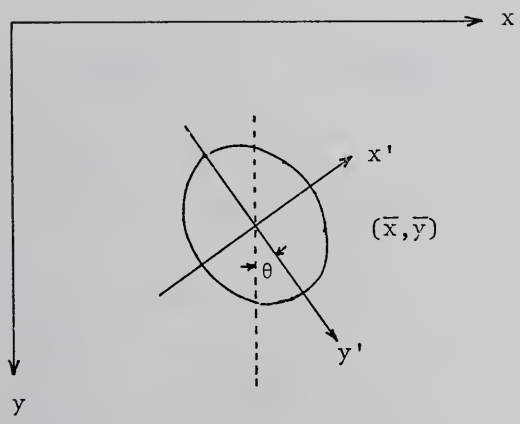


Figure 5.2. Principal axis direction.

sharp the region is. The less the elongation index, the sharper the region is. For example, if the elongation index is 1, the region is a circle, while if the elongation index is almost 0, the region is almost approaching a line.

5.2. Global Features

The global features are the ones which describe the interrelationship of objects in the picture. Since sometimes an object may contain several holes, we can consider the regions enclosed by the boundary contours as individual objects and find the global features among the regions to describe the object itself. To find the global features among objects, only the regions enclosed by the exterior boundary contours need to be considered. The following are several nontrivial global features.

5.2.1. Inclusion Relation Among Objects

Inclusion relations among objects can be derived directly from the inclusion relation among contours. Let H be the Hasse graph representing the inclusion relation among contours and N' be the set of all nodes in H which correspond to exterior contours of objects in the picture. Let H' be the graph which represents the transitive reduction of the reachability among nodes in N' revealed by the graph H. H' is then a Hasse graph which represents the inclusion relation among objects. Since there is a one-to-one

correspondence between exterior contours of objects and objects, the nodes in N' are in one-to-one correspondence with the objects.

5.2.2. Distribution of Objects in the Picture (With Consideration to the Distances Between Objects)

The distance between two objects O and O' is defined as the Euclidean distance between the corresponding centroids o and o'. One method to describe the distribution of objects in the picture is using the Minimal Spanning Tree (MST) technique, which is discussed in detail in Chapter IV. A K-nearest-neighbor graph G = (N, E, W) can be set by the following rules:

1. For any object O in the picture, there exists one and only one node n in N.
2. There exists an edge connecting nodes n and n' if either O' is a K-nearest-neighbor of object O or O is a K-nearest-neighbor of object O', where objects O and O' are corresponding to nodes n and n', respectively.
3. Let e be an edge connecting nodes n and n', the weight W(e) assigned to edge e is defined as the distance between objects O and O', where O and O' are the objects corresponding to nodes n and n', respectively.

Let T be an MST of the weighted graph G. T can be found by the method presented by the flow diagram 4.1. T can reveal the information describing gestalt clusters of objects in the picture. The philosophy is the same as

stated in Chapter IV, except we consider every object in the picture as an individual vertex.

The definition of the distance between objects stated in the last paragraph is a good one if the object size and shape are quite uniform through the picture. If the objects' sizes and shapes are not uniform through the picture, the distance between two objects O and O' may better be defined as $\min\{d(p,p') | p \varepsilon O \text{ and } p' \varepsilon O'\}$.

5.2.3. Distribution of Objects in the Picture (With Consideration to Both the Distances Between Objects and Relative Principal Axes' Directions Among Objects)

If all objects in the picture are quite elliptical in shape, to our visual systems the reasonable gestalt clusters of objects should take not only the distance between objects into consideration, but also the angle between the principal axes of objects into consideration. One practical example is the cluster of cells in the epidermis, which will be discussed in detail in Chapter VI.

The method used to describe the distribution of objects by considering not only the distance between objects, but also the relative principal axes' directions among objects is the same as the one presented in the last section, except in setting the K-nearest-neighbor weighted graph, the weights assignment should be modified (i.e., rule 3 should be changed). Let e be an edge connecting nodes n and n', the weight W(e) assigned to edge e is defined as d(O,O') +

$\gamma\theta(O,O')$, where $d(O,O')$ is the distance between objects O and O', $\theta(O,O')$ is the angle between the major principal axes of objects O and O' and O and O' are the objects corresponding to nodes n and n', respectively.

The weighted graph in the last section is a special case of the weighted graph in this section under the assumption that $\gamma = 0$.

CHAPTER VI

EXPERIMENTS AND CONCLUSIONS

All techniques stated in the previous chapters have been implemented by FORTRAN programs. Biomedical images are chosen as the experiment data. It should be pointed out that the techniques are not restricted to biomedical images. The reason that the author chose this special type of pictures is due to the great need for the handling of biomedical images. Chromosome pictures, skin cell pictures and blood cell pictures are the main pictures we work with in the experiments.

6.1. Experiments With Chromosome Pictures

Figure 6.1 is an 8-level digitized chromosome picture.[23] Figure 6.2 is the enhanced picture and Figure 6.3 is the boundary picture obtained by the programs based on the gradient technique. There is only one boundary in the picture with a length equal to 207. The corresponding sequence of octal chain codes is

44433120110101010100011010121210121345454555554554545
44545445654433211101000001011100010121101122233234 33
43343332100070777077767070013233323333433332100707 77

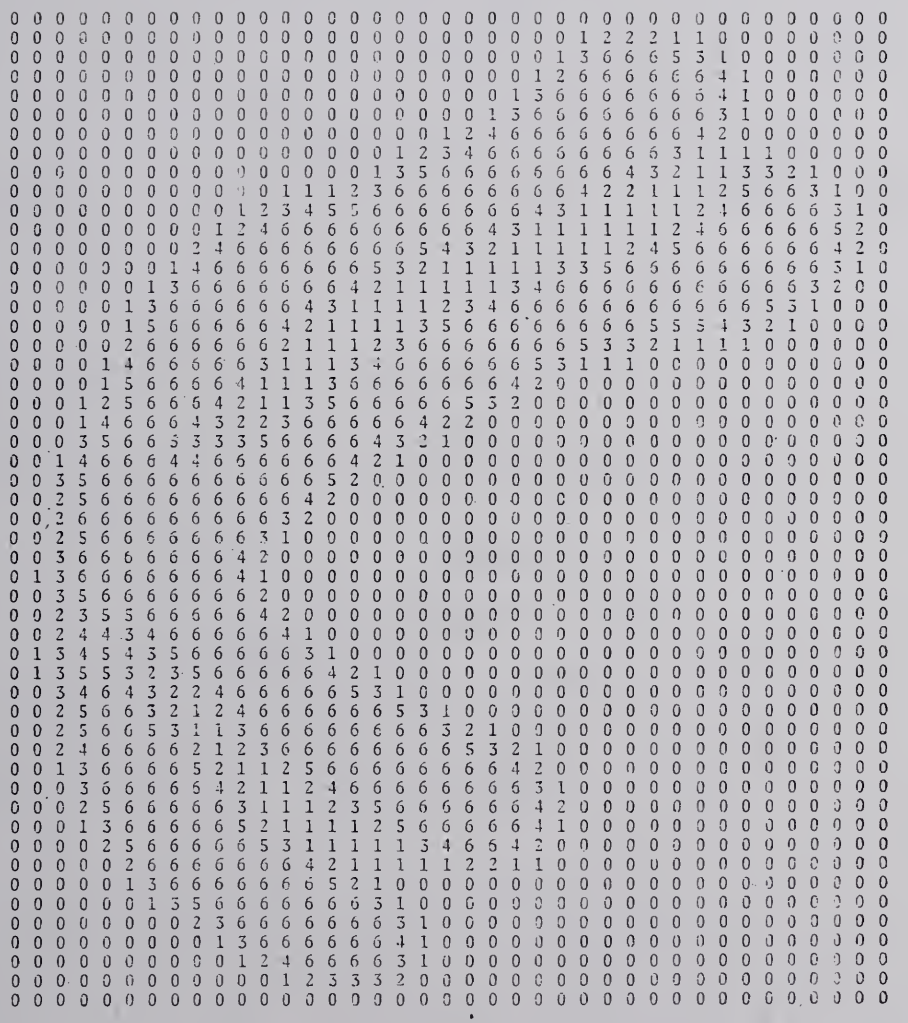


Figure 6.1. An 8-level picture of a human chromosome.

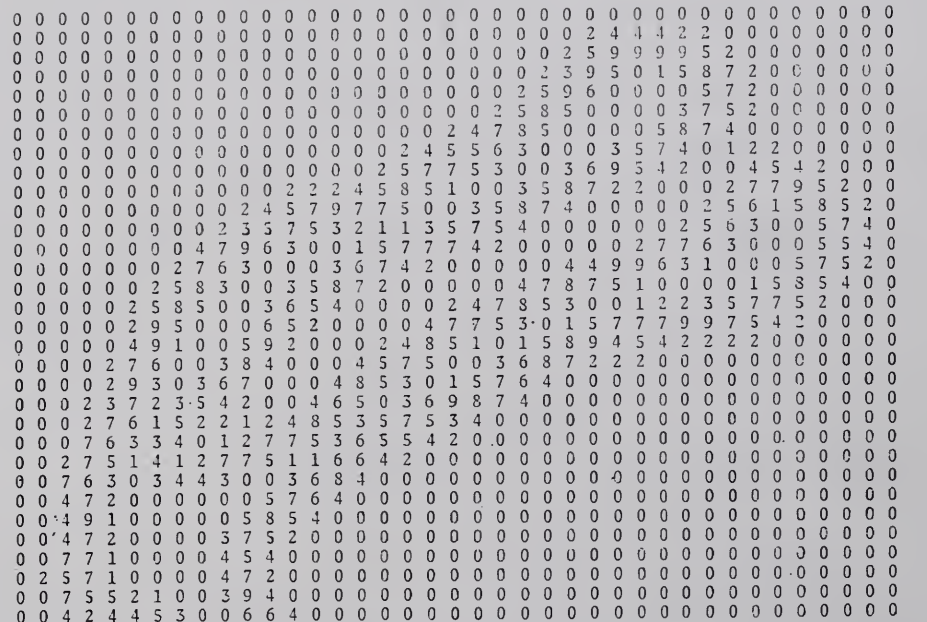


Figure 6.2. The enhanced picture of Figure 6.1.

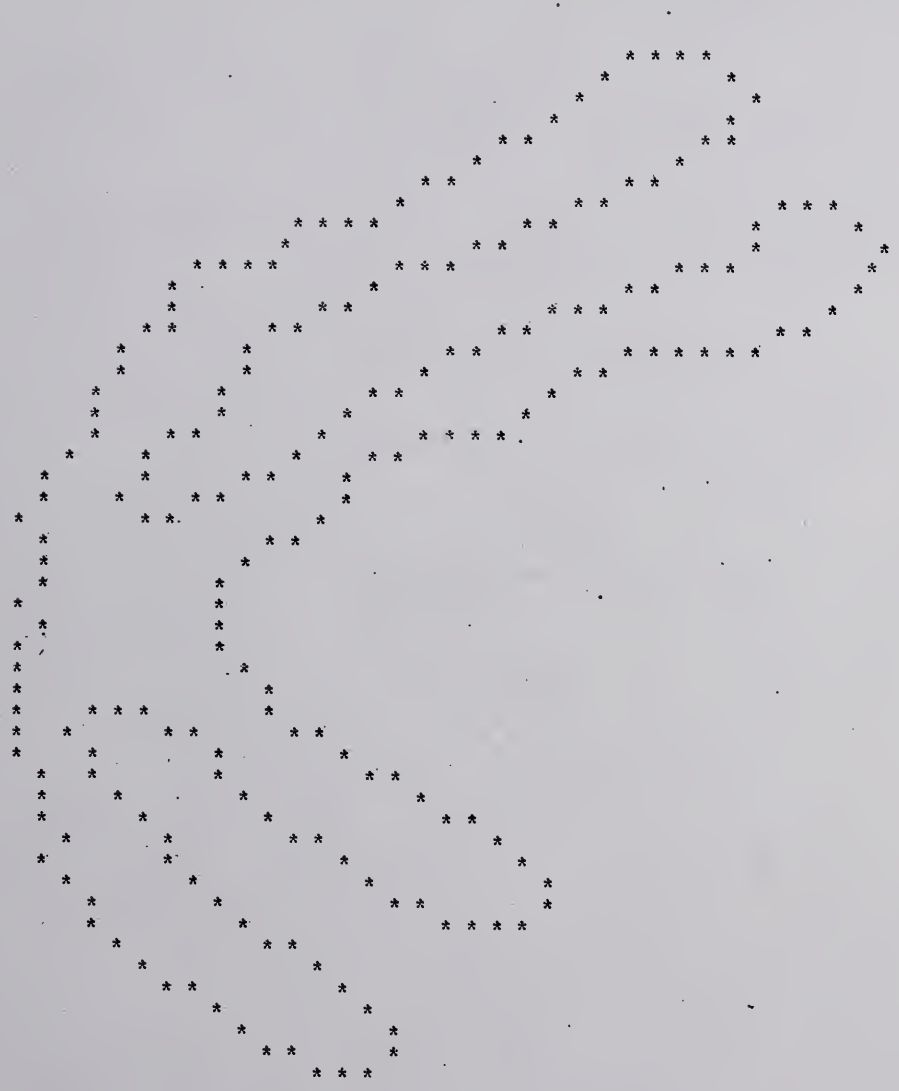


Figure 6.3. The boundary picture obtained from Figure 6.2 by the gradient method.

07776775766766666575667565566565565445544455455555555.

The optimal Hanning window filter, used to filter out the noise in the sequence of differences of successive octal chain codes, has a window size equal to 26. The filtered sequence of differences of successive octal chain codes in shown in Figure 6.4. There are six critical points, the 17^{th}, 45^{th}, 77^{th}, 104^{th}, 122^{nd} and 140^{th} points on the contour. The 5^{th}, 65^{th}, 113^{th} and 148^{th} points on the contour are the peak points. The 33^{rd}, 97^{th} and 131^{st} points in the contour are the valley points. The parameter is +1, which implies that the portions (17, 44), (77, 103) and (122,139) are the concave portions and the portions (45, 76), (104, 121) and (140, 16) are the convex portions. The portion (i, j) denotes the portion of the contour from the i^{th} point to the j^{th} point in the contour.

Figure 6.5 is the labeled output picture of Figure 6.1 using the contour analysis. Figure 6.6 is the contour picture obtained by the method stated in Chapter III. There is only one object contour in the final picture with a length equal to 201. The corresponding sequence of octal chain codes is

44444322111001101010010110112112345555455454545454445
55564343222110100001001011001101111112233333434334332
22100007707770767701232333234333432210070707707677677
66667665676656675655665665554554455454555456.

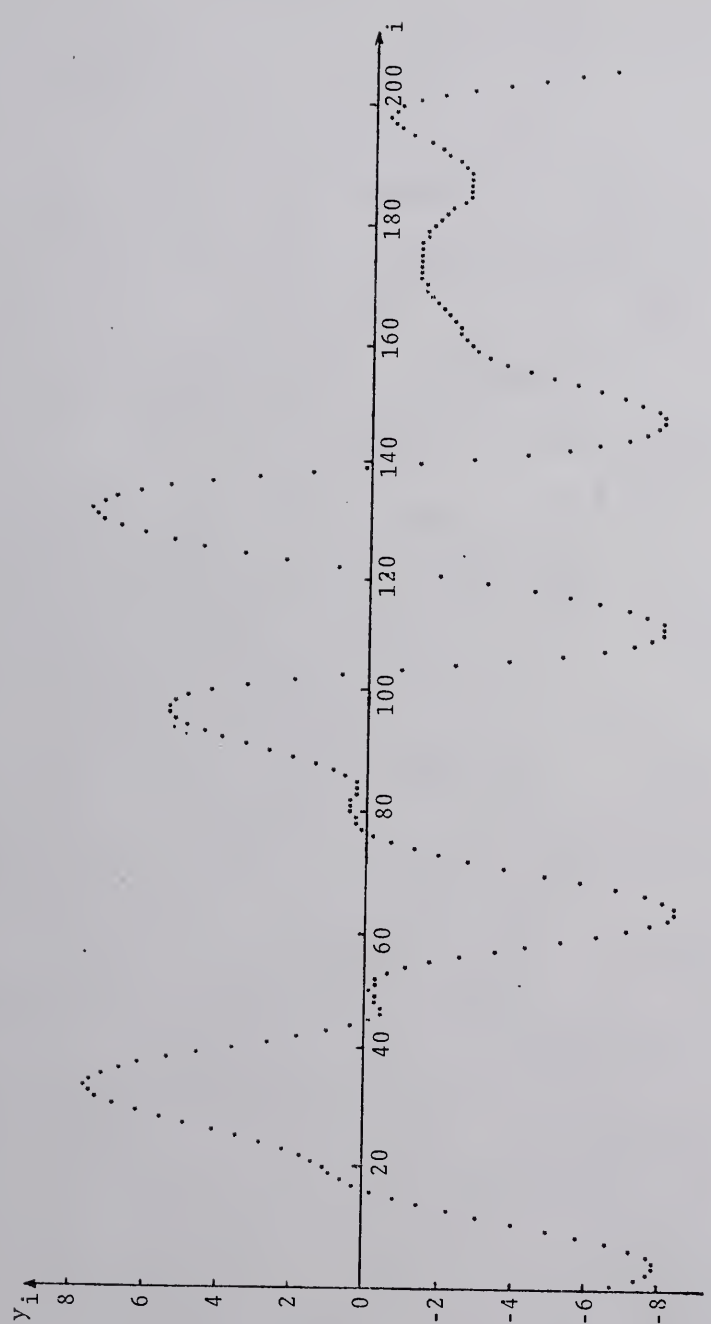


Figure 6.4. The smoothed difference function of the boundary shown in Figure 6.4.

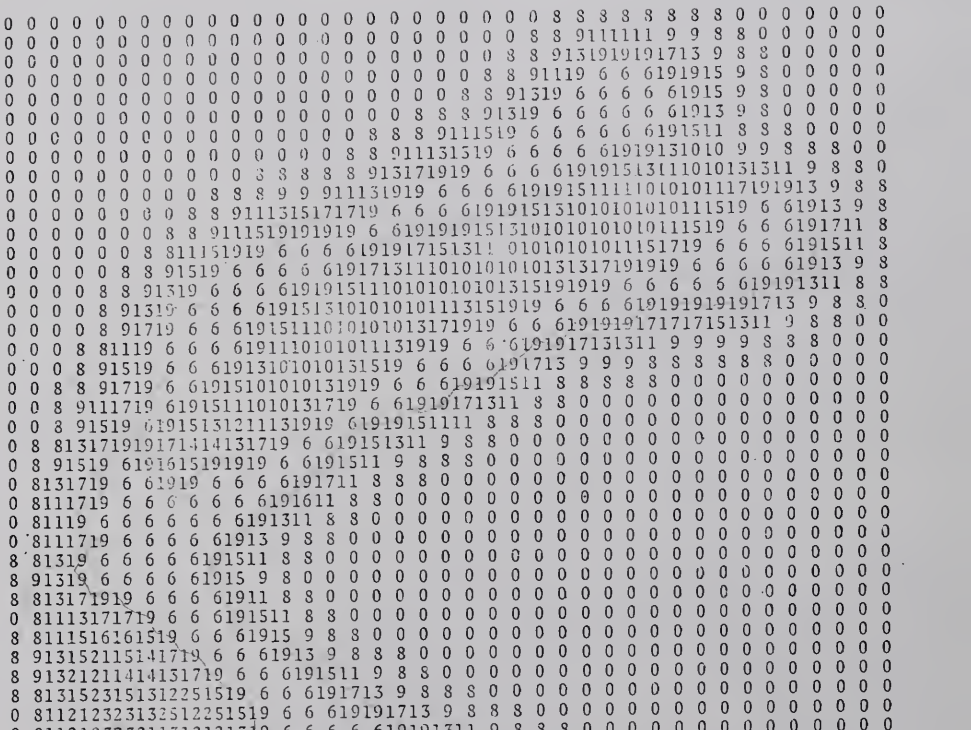


Figure 6.5. The labeled picture obtained from Figure 6.1.

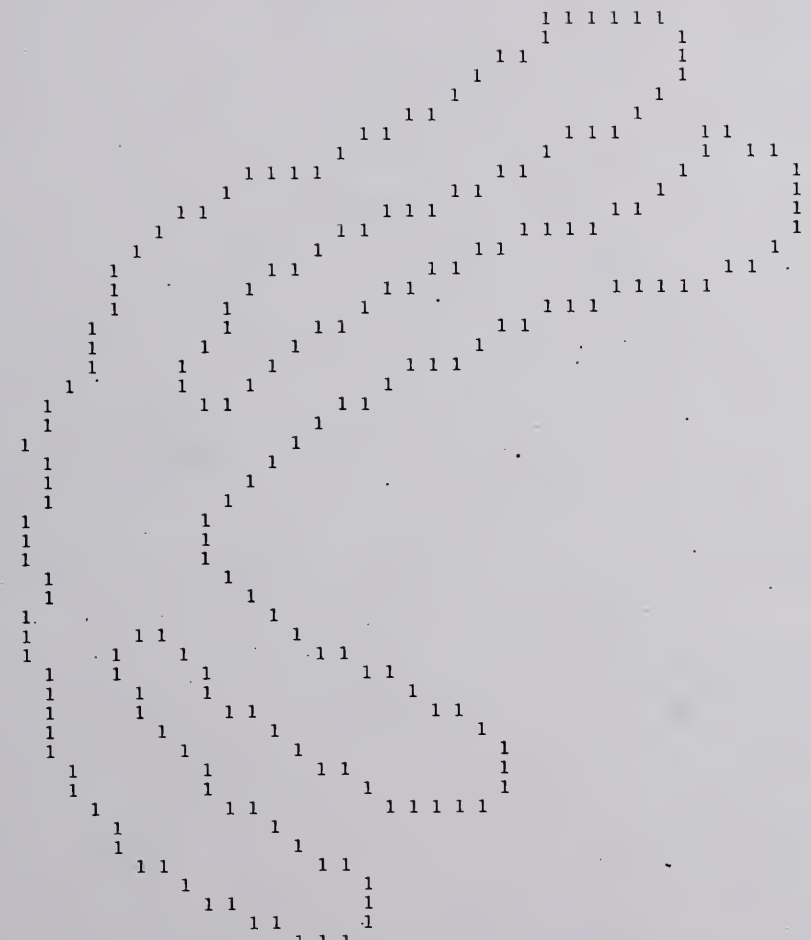


Figure 6.6. The boundary picture obtained from Figure 6.1 by the contour analysis.

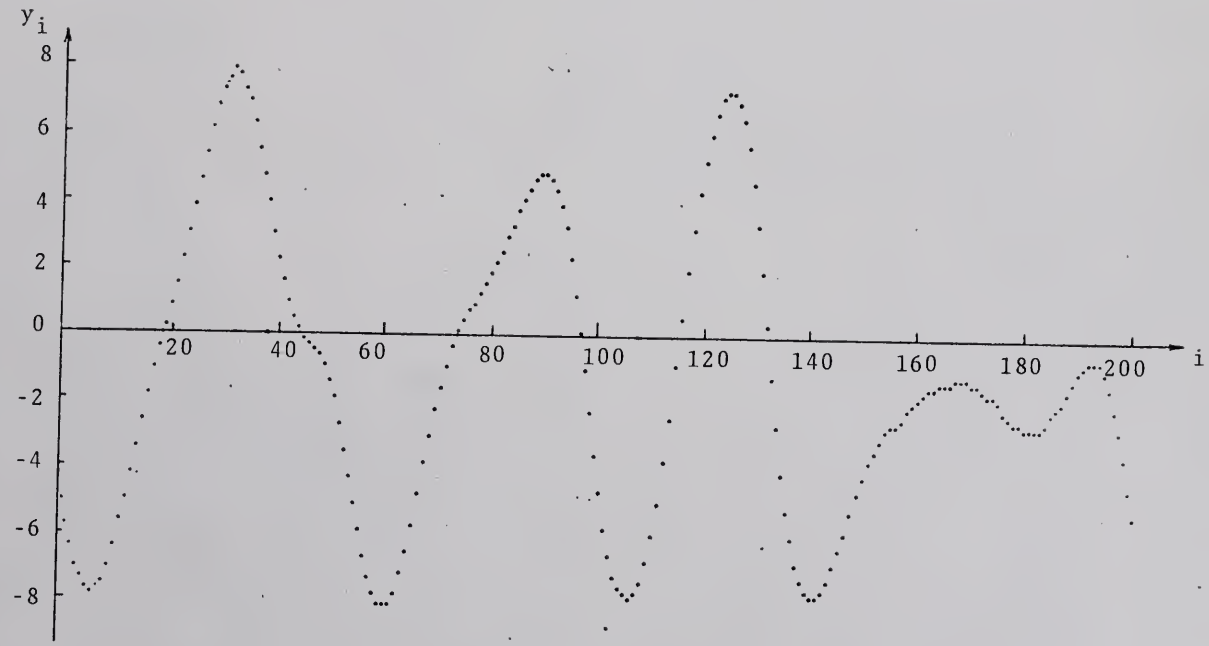


Figure 6.7. The smoothed difference function of the contour shown in Figure 6.6.

The optimal Hanning window filter, used to filter out the noise in the sequence of differences of successive octal chain codes, has a window size equal to 28. The filtered sequence of differences of the successive octal chain codes is shown in Figure 6.7. There are six critical points, the 19^{th}, 45^{th}, 74^{th}, 98^{th}, 116^{th} and 133^{rd} points in the contour. The 6^{th}, 62^{nd}, 107^{th} and 141^{st} points in the contour are the peak points. The 31^{st}, 90^{th} and 125^{th} points in the contour are the valley points. The σ parameter is +1, which implies that the portions (19, 44), (74, 97) and (116, 132) are the concave portions and the portions (45, 73), (98, 115) and (133, 18) are the convex portions. It should be pointed out that the dissimilarity of the contours obtained by the gradient method and the contour analysis is equal to 0 as is expected.

Method 2 as stated in Section 4.2 is used to find the weighted graph of Figure 6.1. An MST has been found as shown in Figure 6.8. The radius of the MST is $r_0 = 51$. There are two centers (23, 7) and (24, 7) in the MST. α is set as 1/4, hence any branch of length less than $[\alpha r_0] = 12$ is removed. Only four radial paths (including two maximal radial paths) are remaining, which are shown in Figure 6.9.

Radial path 1:

start point = (6, 34),

chain code sequence = 000660060020202020020020200
0020020202022022222,

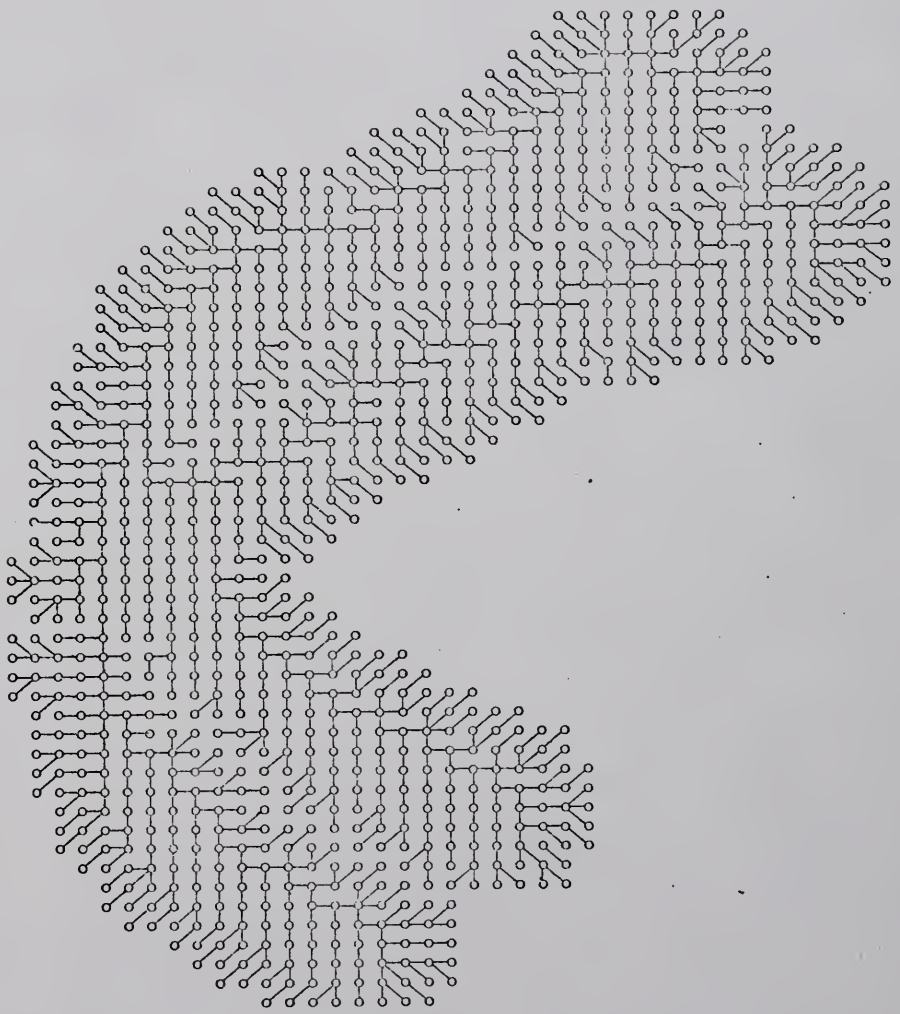


Figure 6.8. An MST obtained from Figure 6.1.

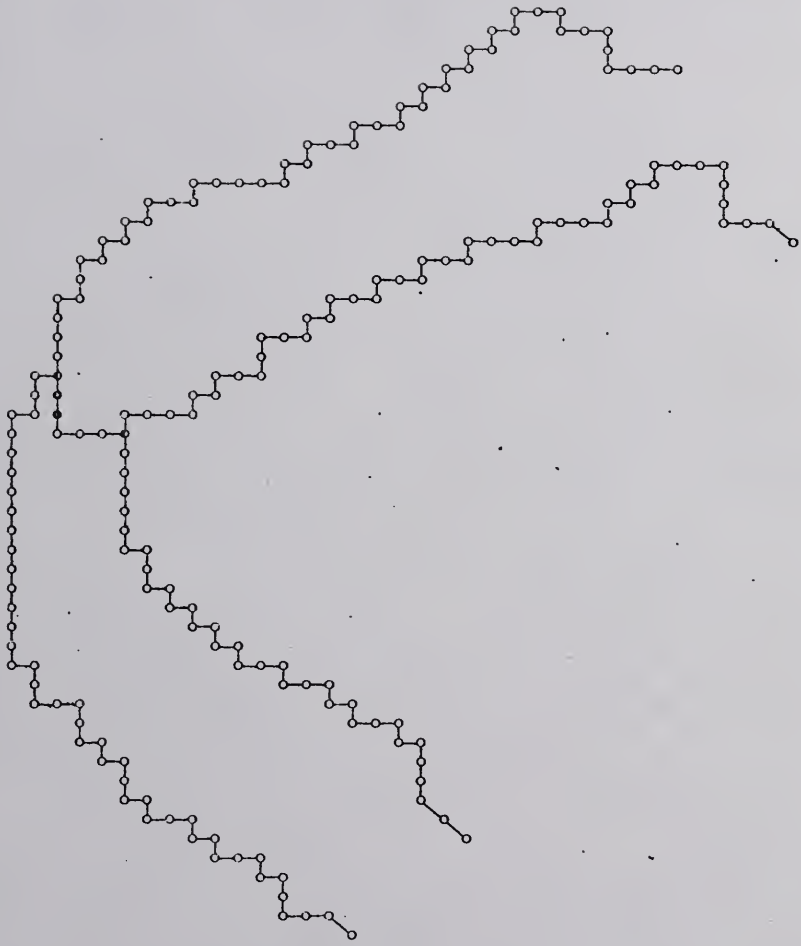


Figure 6.9. The four radial paths in Figure 6.8 obtained by deleting small branches.

length = 51 (i.e., it is a maximal radial path).

Radial path 2:

start point = (15,39),

chain code sequence = 70066600020202000200020020020020020022002000200066,

length = 51 (i.e., it is a maximal radial path).

Radial path 3:

start point = (46,25),

chain code sequence = 776660600606006006060606066066666600066,

length = 39.

Radial path 4:

start point = (50, 20),

chain code sequence = 70066060060600606606066006606666666666666066022,

length = 47.

Radial path 3 is dependent on maximal radial path 2, while radial path 4 is dependent on maximal radial path 1. (22, 7) and (25, 10) are the only two branching vertices in the MST with the removal of small branches. The octal chain code sequence of the path from (22, 7) to (25, 10) is 222444 which is of length 6. Since this length is much smaller than the length of any radial path, the skeleton of the picture can be considered as consisting of four arms. Each arm is a subpath of a relative diametral path from a leaf to the nearest branching vertex. The branching vertices are shown by ◐, and the centers are shown by • in Figure 6.9.

6.2. Experiments on Skin Cell Pictures

It is known[34] that there are two types of cells in the epidermis, basal cells and squamous cells. If the epidermis is normal, the basal cells' nuclei are generally perpendicular to the dermis-epidermal junction, and the squamous cells' nuclei are parallel to the epidermal-dermal junction. The basal cell carcinoma will appear in an island. The main features of the tumor cells in the island are:

1. the nuclei are crowded, and
2. there is no uniform pattern of the alteration of the major principal axis direction of the nuclei.

The hair follicle shaft can be distinguished from the basal cell carcinoma island because the hair follicle has an acellular protein part which makes the hair follicle look like a doughnut instead of a solid island. The sebaceous glands can be easily distinguished from the basal cells carcinoma in that the sabaceous cells' nuclei are quite widely separated. All other structures in the skin are characterized by having scattered nuclei or being in the form of a small island.

In our experiment we first extract nuclei from the picture. Figure 6.10 is a portion of the skin cell picture obtained through the quantization of the PIDAC. Contour analysis is used to extract objects from the picture. Figure 6.11 shows the intermediate result, which is the

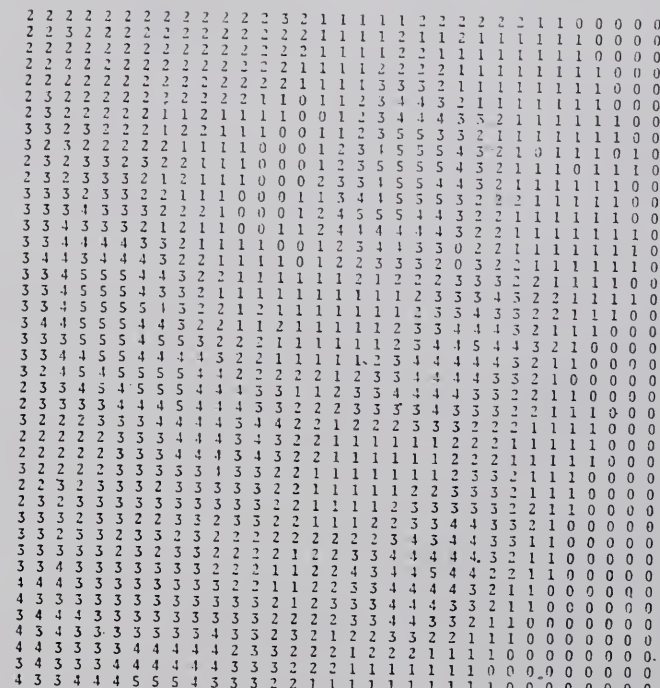


Figure 6.10. An 8-level picture of a portion of a skin cell picture.

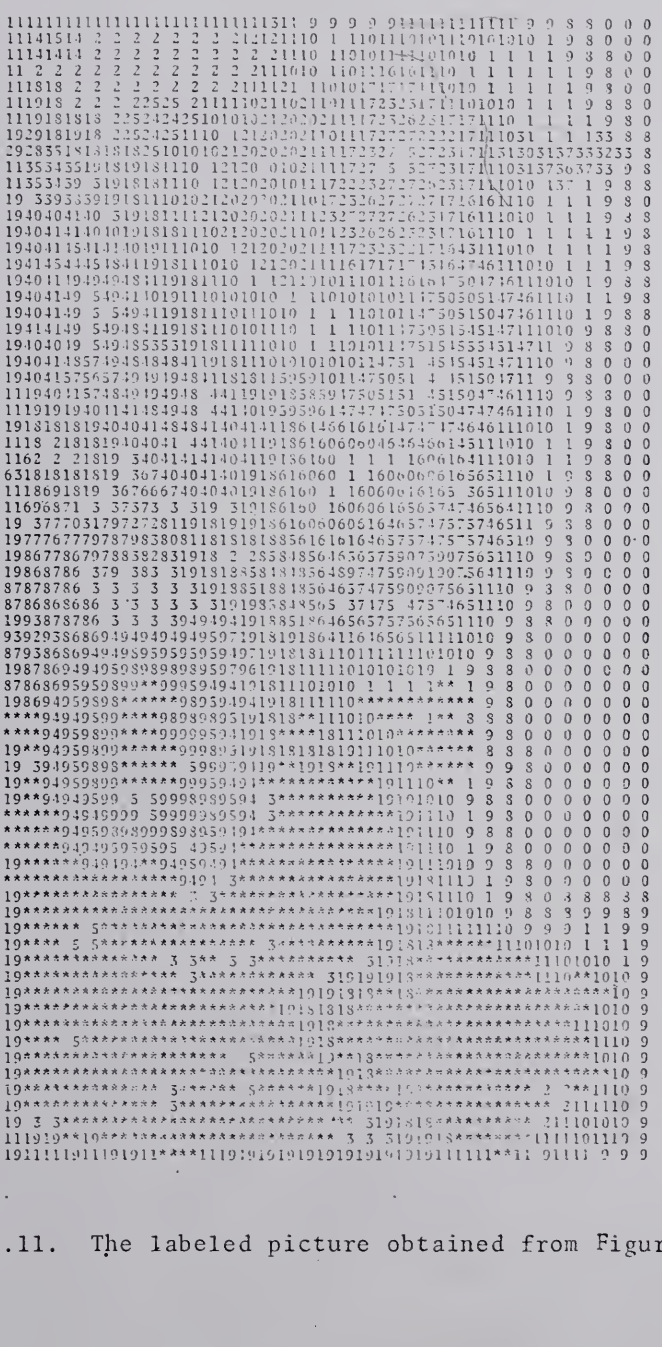


Figure 6.11. The labeled picture obtained from Figure 6.10.

labeled picture of Figure 6.10. Since this is only a portion of the epidermis, the cytoplasma part touches the picture frame, which is then not under consideration. It should also be pointed out that the cytoplasma parts of the cells are merged together, hence it is very hard to tell the cytoplasma boundary of each individual cell. The objects extracted are then only the nuclei as is expected. Figure 6.12 shows (nuclei) object boundaries. Picture points which have the value i = 1,...,9 are the boundary points of object i. There are in total 9 nuclei in the picture.

Local features of individual nuclei are then found by the methods stated in Chapter V. Table 6.1 shows the results of the local features of the 9 nuclei. XBAR and YBAR specify the location of the centroid of an object. AREA is the area of an object. THETA is the angle (in radians) of the major principal axis of an object with respect to the positive y axis. The elongation index of an object is what is defined in Section 5.1.4.

The distribution of nuclei in the picture is described by the MST as stated in Section 5.2.3. The weight assigned to the edge connecting two nodes n and n' is $D(O,O') + \gamma\theta(O,O')$, where O and O' are the nuclei corresponding to the nodes n and n', respectively. The appropriate MST (i.e., the appropriate γ value) is found through the flow diagram, shown in Figure 6.13, by changing γ value until proper

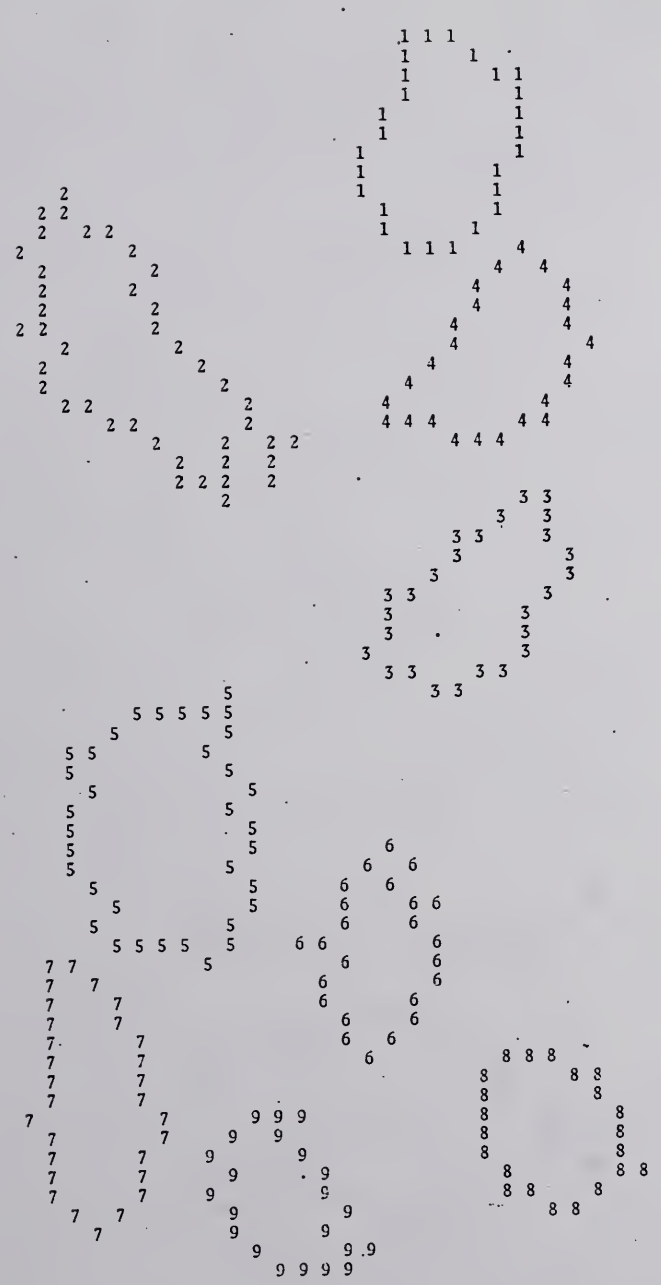


Figure 6.12. The boundary picture obtained from Figure 6.10 by the contour analysis.

Table 6.1

Local features of nuclei in Figure 6.12

Nucleus	XBar	YBar	Area	Theta	Eccentricity
1	11.6728	21.5994	54.5	0.18481	0.455348
2	22.1863	7.7279	68.0	1.03434	0.407566
3	35.4360	22.8333	47.0	-0.91540	0.325312
4	22.7193	23.9228	47.5	-1.02386	0.553201
5	47.1780	9.0922	79.5	-0.06773	0.377898
6	53.7212	18.4975	33.5	0.06664	0.212766
7	61.1953	5.9293	49.5	-0.07382	0.145627
8	62.8950	25.9561	38.0	1.15331	0.921100
9	66.3389	14.0645	31.0	1.15437	0.869634

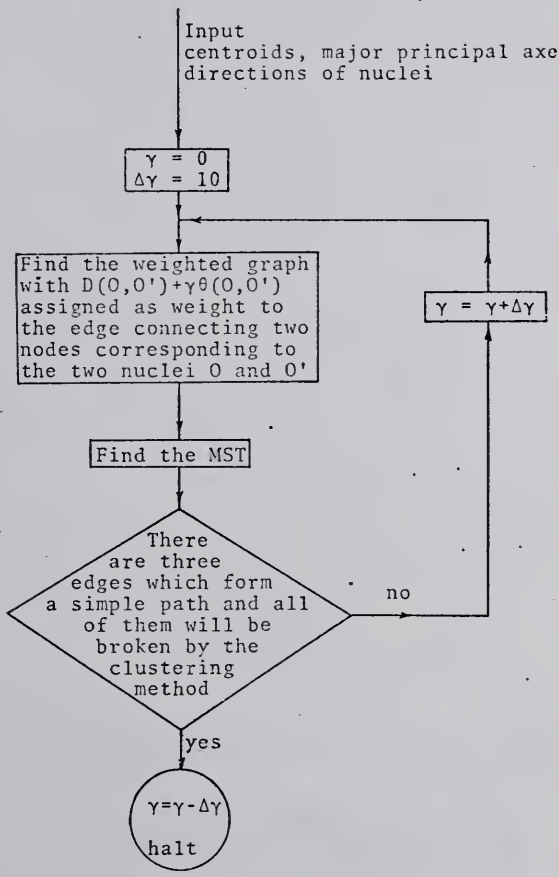


Figure 6.13. Flow diagram of finding the distribution of objects in Figure 6.10.

clusters are obtained. The clustering method used in Figure 6.13 is that if $W(e) > \overline{w}+\alpha\sigma_w$, then the edge e is erased, where $\overline{w}$ is the mean weight in the tree and σ_w is the standard deviation of the weights in the tree and α is an adjustable parameter and is set equal to 3 in the experiment. Figure 6.13 is then a flow diagram which not only gives the result of the appropriate MST but also gives the result of the clusters of the MST. Figure 6.14 is the MST obtained by this method and the corresponding γ value is 20. Figure 6.15 shows the clusters of the MST.

From Figure 6.15 we can tell that the nuclei in the same cluster are of the same type. Nucleus 1 is a basal nucleus. Nuclei 3 and 4 are basal nuclei. Nuclei 9 and 8 are basal nuclei. Nucleus 2 is a squamous nucleus. Nuclei 5, 6 and 7 are squamous nuclei.

6.3. Experiments on Blood Cell Pictures

The histogram of blood cell density via the cell intensity reveals information for some disease diagnosis. Blood cell pictures are different from skin cell pictures in that there are some blood cells whose intensities are lower than their corresponding background intensities, while every skin cell intensity is higher than its corresponding background intensity. Hence a hole can be a cell in the blood cell pictures. Figure 6.16 is a portion of the blood cell pictures in our experiment.

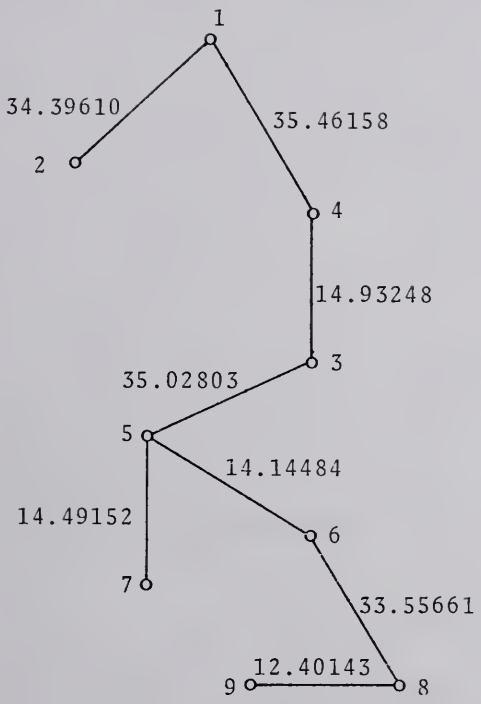


Figure 6.14. An MST used to describe the distribution of objects in Figure 6.10.

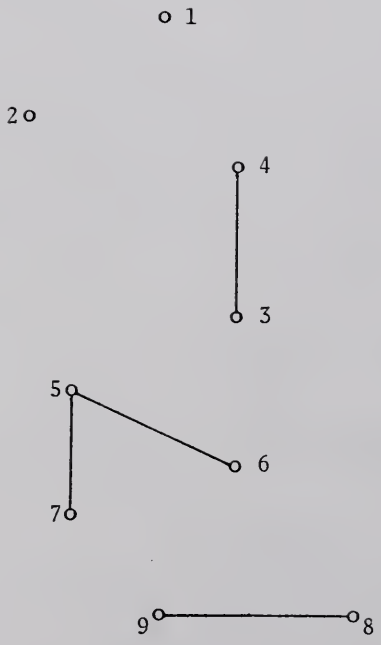


Figure 6.15. The clusters of objects in Figure 6.10.

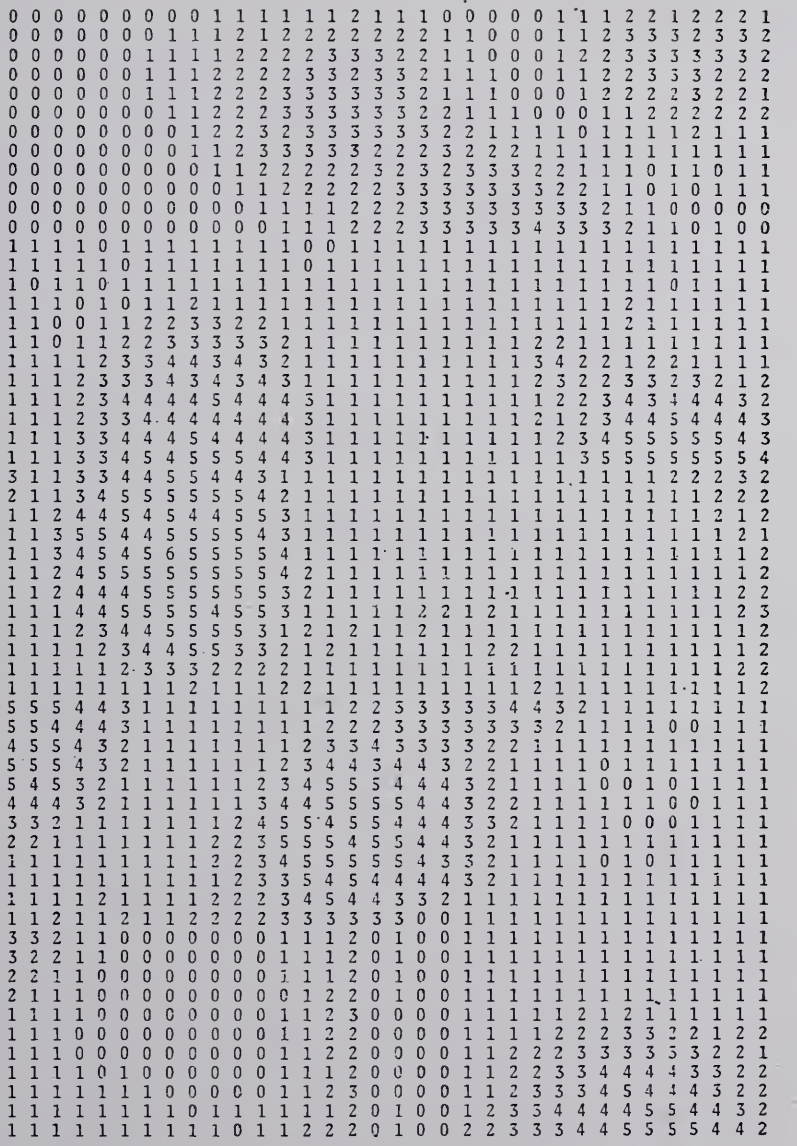


Figure 6.16. An 8-level picture of a portion of a blood cell picture.

The intensity of a cell is defined as the average of the intensities of picture points inside the cell. Let H be the Hasse graph of all the contours in the picture which represent the inclusion relation. Let C be the object contour of a cell O and H_C be the subtree of H with C as the root. A level assignment to the subtree H_C is very easily done from the level assignment of H. Assume that the level of C in H is L(C), the level of any contour C' in H_C is then L(C') - L(C) + 1, where L(C') is the level of C' in H. The intensity of the cell O is then

$$g(O) = \frac{\sum_{i=1}^{n_O} \sum_{j=1}^{m_i} \left(A(C_j^i) - \sum_{k=1}^{m_j^i} A(C_k^{(i,j)})\right) g(C_j^i)}{A(C)}$$

where n_O is the highest level of the subtree H_C, m_i is the number of level i nodes in H_C, A(C') is the area enclosed by the contour C', C_j^i is the contour corresponding to a level i node in H_C, m_j^i is the number of sons of the node corresponding to the contour C_j^i and $C_k^{(i,j)}$ is a contour corresponding to a son of the node corresponding to the contour C_j^i. In this special experiment, the intensity of the cell is assigned as the integer which is closest to the value g(O) calculated from the above equation. The counting of overlapping cells is done by the flow diagram shown in Figure 6.17. K in the flow diagram is the number of cells which jointly form a single object boundary found by the contour analysis.

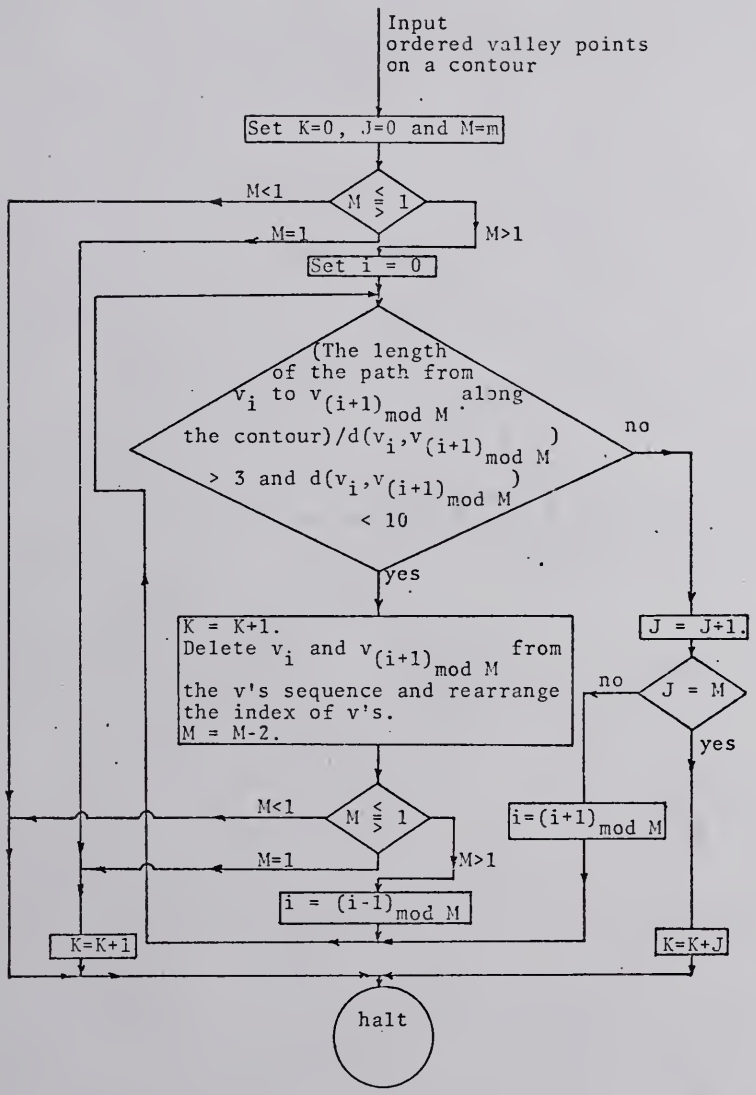


Figure 6.17. Flow diagram for counting overlapping blood cells.

Figure 6.18 shows the histogram of intensities of a blood cell picture in our experiment. In Figure 6.18, n represents the total number of cells.

6.4. Conclusions and Further Research

From the experiments, it is found that the two methods of object extraction are good in different situations:

1. Object extraction by the gradient method can work very well if the boundary points have very high gradients. That is, it works well if the boundaries are sharp.
2. Object extraction by contour analysis works well under the condition that the objects in a picture occupy quite uniform intensity regions. This method is especially good for area pictures.

By using the graph-theory approach to picture processing, a clustering method can then be applied. A multi-level picture g can be transferred into a weighted graph G. From the weighted graph G, a minimal spanning forest of G can then be found. The clustering method is then applied to every MST. Let r_o be the radius of an MST T. Small branches in T are removed if their lengths are less than αr_o, where $o<\alpha<1$ is a threshold. The remaining radial paths are combined to form major paths by the method stated in Section 4.1.2. Let $s_i = (v_{i_o},\ldots,v_{i_t})$ be a major path of

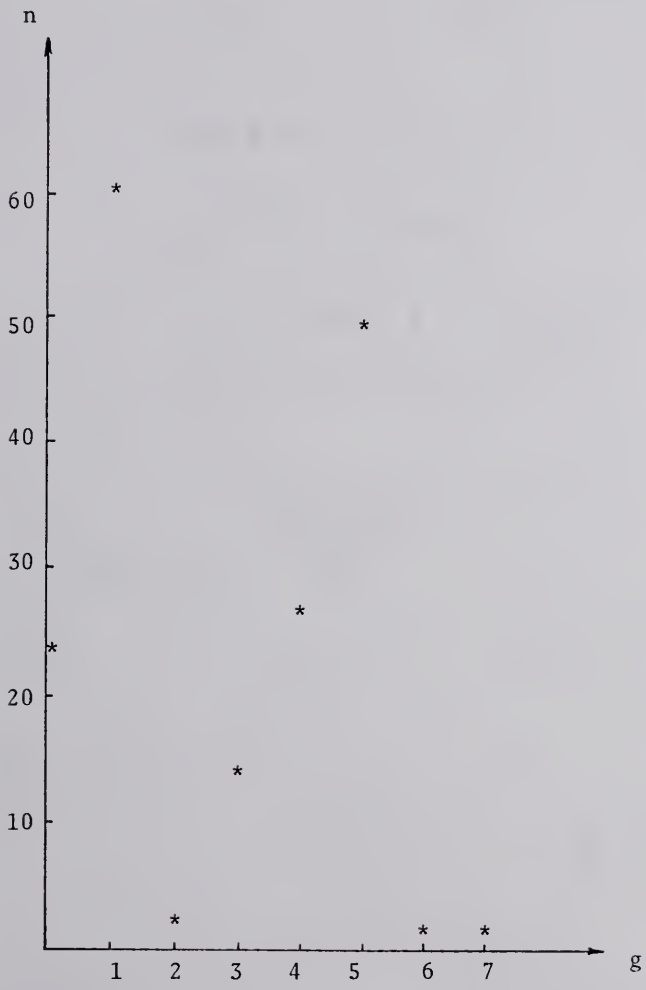


Figure 6.18. The histogram of the intensities of a blood cell picture.

length t and $(w_{i_1},\ldots,w_{i_t})$ be the corresponding weight sequence, that is, w_{i_j} is the weight of the edge connecting $v_{i_{(j-1)}}$ and v_{i_j}. Significant local minima weights are noted from the histograms of the weights along the major paths. Several cluster ranges of weights can be found between significant local minima. If w_{i_j} and $w_{i_{(j-1)}}$ are in different cluster ranges and if $w_{i_j} < w_{i_{(j-1)}}$, then the edge connecting vertices $v_{i_{(j-1)}}$ and v_{i_j} are broken. After this clustering method is applied to all major paths in T, T will be clustered into several subtrees. Hence the minimal spanning forest will be clustered into several trees. Let two clusters be represented by two trees T_1 and T_2. T_1 and T_2 will be linked together if (1) the average weights w_1 and w_2 of T_1 and T_2 are very close, that is, $|w_1-w_2| < w_\alpha$, and (2) if a leaf v_1 in T_1 and a leaf v_2 in T_2 are adjacent in G_T, and if the weight w' of the edge incident with v_1 and v_2 are very close to $\frac{w_1+w_2}{2}$, that is $|\frac{w_1+w_2}{2} - w'| < w_\alpha$.

After this linking process, every tree will represent an object in the picture. The skeleton of an object can be represented by the significant major paths of the tree representing the object.

From the experiments on the skin cell picture, it is found that object extraction by contour analysis is a very promising method to handle cell pictures. The global feature used in the experiment also shows that the description of the distribution of nuclei in the picture is a very reasonable global feature.

It is suggested that a further development of the system is needed in which a slide is taken as a roll of films and the PIDAC can handle a roll of films. It is possible to work on the individual film and to store the local features of objects in the individual films. In describing the global features, it is possible to consider all films at one time by knowing the location of each film in the slide. Once such a system is set, the distribution of the nuclei in the picture can be found more completely.

The completion of the clustering stated in Section 6.2 can be done by linking together the nuclei which definitely belong to the same type. When one scans through the picture, if the object contour point of nucleus O is always just ahead of a contour point of a high level contour C and if another nucleus O' satisfies the same condition for the same contour C, nuclei O and O' are said to be the same type. Let each cluster obtained by the method stated in Section 6.2 be considered as a vertex. Clusters S_i and S_j are connected if there exists the definite same type nuclei O_k, and O_ℓ^j, in S_i and S_j, respectively. A weight $W(S_i,S_j)$ is then assigned to the edge connecting S_i and S_j by

$$W(S_i,S_j) = D(O_k^i,O_\ell^j) = \min\{D(O_{k'}^i,O_{\ell'}^j) \mid O_{k'}^k \in S_i, O_{\ell'}^j \in S_j \text{ and } O_{k'}^i \text{ and } O_{\ell'}^j \text{ are definitely the same type}\}.$$

A weighted graph is then set. Find the corresponding minimal spanning forest. If the clusters S_i and S_j are connected in the minimal spanning forest, connect O_k^i and O_ℓ^j in the graph obtained through the cluster shown in Section 6.2.

After this process, it is easily seen that for a normal epidermis, there should be two large clusters. An island of nuclei can be found to be a cluster through the distribution of the objects in a picture by considering only the distance between objects (i.e., $\gamma = 0$). A tumor detection system can then be set by observing the resulting clusters of nuclei in an island.

The blood cell extraction is more complicated than the skin cell extraction because some blood cells appear as holes in the picture. While the result we get in the blood cell picture analysis is quite reasonable, a more detailed research should be conducted to achieve a more complete object extraction system in dealing with the pictures which have holes as objects. Again the PIDAC should be developed to be able to handle a rool of films which can then find more reasonable data about the density of the cells, because the result from one film is too restricted. The average of the density of a roll of films can give a very satisfactory result.

BIBLIOGRAPHY

1. L. G. Roberts, "Machine Perception of Three-dimensional Solids," in Optical and Electro-optical Information Processing (edited by J. T. Tippett et al.), M.I.T. Press, Cambridge, Mass., 1965.

2. A. Guzman,"Computer Recognition of Three-dimensional Objects in a Visual Scene," Ph.D. Dissertation, M.I.T., 1968.

3. B. Raphael, "Programming a Robot," Proc. IFIPS, Edinburgh, 1968.

4. M. Eden and M. Halle, "Characterization of Cursive Handwriting," Proceedings of the Fourth Symposium on Information Theory (edited by C. Cherry), Butterworth Scientific Publications, London, 1961.

5. J. T. Tou and R. C. Gonzalez, "A New Approach to Automatic Recognition of Handwritten Characters," Technical Report No. 70-101, CIR, University of Florida, 1970.

6. R. H. Cofer,"The Automation of Map Reading," Ph.D. Dissertation, University of Florida, 1971.

7. R. Narasimhan, "Syntax-directed Interpretation of Classes of Pictures," Comm. ACM, Vol. 9, No. 3, March 1966.

8. H. S. White, "DAPR - Digital Automatic Pattern Recognition for Bubble Chambers," in Pictorial Pattern Recognition (edited by G. C. Cheng et al.), Thompson Book Co., Washington, D.C., 1968.

9. W. J. Hankley and J. T. Tou, "Automatic Fingerprint Interpretation and Classification via Contextual Analysis and Topological Coding," in Pictorial Pattern Recognition (edited by G. C. Cheng et al.), Thompson Book Co., Washington, D.C., 1968.

10. R. S. Ledley, "Automatic Pattern Recognition for Clinical Medicine," Proc. of the IEEE, Vol. 57, No. 11, November 1969.

11. J. M. S. Prewitt and W. S. Mendelsohn, "The Analysis of Cell Images," Annals of the New York Academy of Sciences, Vol. 128, 1966.

12. H. Freeman, "On the Encoding of Arbitrary Geometric Configurations," IRE Trans. on Electronic Computers, Vol. EC-10, No. 2, June 1961.

13. R. S. Ledley et al., "FIDAC: Film Input to Digital Automatic Computer and Associated Syntax-directed Pattern Recognition Programming System," in Optical and Electro-optical Information Processing, M.I.T. Press, Cambridge, Mass., 1965.

14. J. K. Hawkins, "Image Processing Principles and Techniques," in Advances in Information Systems Science, Vol. 3 (edited by J. T. Tou), Plenum Publishing Co., N. Y., 1970.

15. M. H. Huechel, "An Operator Which Locates Edges in Digitized Pictures," J.ACM, Vol. 18, No. 1, January 1971.

16. C. T. Zahn, "Graph-theoretical Methods for Detecting and Describing Gestalt Clusters," IEEE Trans. on Computers, Vol. C-20, No. 1, January 1971.

17. M. Wertheimer, "Principles of Perceptual Organization," in Readings in Perception (edited by D. Beardsley and M. Wertheimer), Van Nostrand, Princeton, N. J., 1958.

18. H. Freeman, "Techniques for the Digital Computer Analysis of Chain-coded Arbitrary Plane Curves," Proc. National Electron Conference, Chicago, Ill., Vol. 17, 1961.

19. H. Blum, "A Transformation for Extracting New Descriptors of Shape," in Models for the Perception of Speech and Visual Form (edited by W. Wathen et al.), M.I.T. Press, Cambridge, Mass., 1967.

20. A. Rosenfeld and J. L. Pfaltz, "Sequential Operations in Digital Picture Processing," J.ACM, Vol. 13, October 1966.

21. U. Montanari, "A Method for Obtaining Skeletons Using a Quasi-Euclidean Distance," J.ACM, Vol. 15, No. 4, 1968.

22. O. Philbrick, "Shape Description with the Medial Axis Transformation," in Pictorial Pattern Recognition (edited by G. C. Cheng et al.), Thompson Book Co., Washington, D. C., 1968.

23. G. Levi and U. Montanari, "A Grey-weighted Skeleton," Information and Control, Vol. 17, No. 1, August 1970.

24. J. L. Pfaltz, J. W. Snively, Jr., and A. Rosenfeld, "Local and Global Picture Processing by Computer," in Pictorial Pattern Recognition (edited by G. C. Cheng et al.), Thompson Book Co., Washington, D. C., 1968.

25. A. C. Shaw, "A Formal Picture Description Scheme as a Basis for Picture Processing Systems," Information and Control, Vol. 14, No. 1, January 1969.

26. J. T. Tou, "Engineering Principles of Pattern Recognition," in Advances in Information Systems Science, Vol. 1 (edited by J. T. Tou), Plenum Publishing Co., N. Y., 1968.

27. N. J. Nilsson, Learning Machines, McGraw-Hill Book Co., N.Y., 1965.

28. S. K. Chang, "The Analysis and Synthesis of Two-dimensional Patterns Using Picture-processing Grammars, Ph.D. Dissertation, University of California, Berkeley, 1969.

29. O. Ore, Theory of Graphs, American Mathematical Society, Providence, 1962.

30. F. Hohn,"Algebraic Foundation for Automata Theory," in Applied Automata Theory (edited by J. T. Tou), Academic Press, N. Y., 1968.

31. J. C. Gower and G. J. S. Ross, "Minimum Spanning Trees and Single Linkage Cluster Analysis," Applied Statistics, Vol. 18, No. 1, 1969.

32. B. Gold and C. M. Rader, Digital Processing of Signals, McGraw-Hill Book Co., N. Y., 1969.

33. F. P. Beer and E. R. Johnston, Mechanics for Engineers, McGraw-Hill Book Co., N. Y., 1956.

34. R. L. Hackett, Private communication, Medical School, University of Florida, 1971.

BIOGRAPHICAL SKETCH

Peter Pei-teh Lin was born December 8, 1945, in Chekiang, China. In July, 1962, he graduated from Chien-Kuo High School. In July, 1966, he received the degree of Bachelor of Science in Electrical Engineering from National Taiwan University. In 1967 he enrolled in the Graduate School of the University of Florida. He worked as a graduate assistant in the Center for Informatics Research until August, 1969, when he received the degree of Master of Science in Engineering. From September, 1969, until the present time he has pursued his work toward the degree of Doctor of Philosophy.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Julius T. Tou, Chairman
Graduate Research Professor
of Electrical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Arnold H. Nevis
Professor of Medicine, Electrical Engineering and Biophysics

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Zoran R. Pop-Stojanovic
Associate Professor of
Mathematics

This dissertation was submitted to the Dean of the College of Engineering and to the Graduate Council, and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

March, 1972

Dean, College of Engineering

Dean, Graduate School