

2-28-1989

# Application of Hough Transformation to Detect Ovulatory Patterns in Cervical Mucus Images

Shiliang Wang  
*Portland State University*

Follow this and additional works at: [https://pdxscholar.library.pdx.edu/open\\_access\\_etds](https://pdxscholar.library.pdx.edu/open_access_etds)



Part of the [Electrical and Computer Engineering Commons](#)

Let us know how access to this document benefits you.

---

## Recommended Citation

Wang, Shiliang, "Application of Hough Transformation to Detect Ovulatory Patterns in Cervical Mucus Images" (1989). *Dissertations and Theses*. Paper 3989.

<https://doi.org/10.15760/etd.5873>

This Thesis is brought to you for free and open access. It has been accepted for inclusion in Dissertations and Theses by an authorized administrator of PDXScholar. Please contact us if we can make this document more accessible: [pdxscholar@pdx.edu](mailto:pdxscholar@pdx.edu).

AN ABSTRACT OF THE THESIS OF Shiliang Wang for the Master of Science in Electrical and Computer Engineering presented February 28, 1989.

Title: Application of Hough Transformation to Detect Ovulatory Patterns in Cervical Mucus Images

APPROVED BY MEMBERS OF THE THESIS COMMITTEE:



Marek A. Perkowski, Chairman



Rajinder P. Aggarwal



Bradford R. Crain

A microcomputer system called ovulocomputer is proposed. The system is used to predict and detect ovulation of women. It will use image processing and analysis, pattern recognition theory, spectrophotometry, ionometry and conductometry to measure various physical and chemical properties of cervical mucus that are related to blood hormone levels.

The functions and characteristics of cervical mucus are introduced in this thesis. Among them, the identification of cervical mucus images, named the ferning test, and related features, are well described.

A new idea, called ferning test automation, which has been neither proposed nor developed before, is presented and explained in detail in this thesis.

Some necessary image processing procedures for the cervical mucus images taken by microscope are proposed and the some of the preprocessing results are presented in Appendix C. The results indicate that the proposed preprocessing is feasible.

The application of Hough Transformation are developed and applied to the ferning test automation. The algorithm using modified Hough Transformation to extract ferning features and the results of this algorithm applied in some simulative images are presented. The results have demonstrated the feasibility of this method. The ferning features extracted by this algorithm are introduced and explained in detail. The application program of the Hough Transformation is presented in Appendix A.

The feature spaces are created based on the extracted ferning features. The classical statistical pattern recognition theory (Gaussian Classifier) is applied in these ferning feature spaces. The Gaussian Classifier can make the decisions for the ferning test automation system. Both the training set data processing and classifier procedure are introduced. A program of Gaussian Classifier is also presented in Appendix B. The program contains two routines: training set data processing and classification. The training set data processing is used to "teach" the machine with the "experienced information" and the classifier procedure uses this "experienced information" to perform the classifying functions. The examples of the application of this program demonstrates the results of the ferning pattern recognition.

**APPLICATION OF HOUGH TRANSFORMATION TO DETECT  
OVULATORY PATTERNS IN CERVICAL MUCUS IMAGES**

by

**SHILIANG WANG**

A thesis submitted in partial fulfillment of the  
requirements for the degree of

**MASTER OF SCIENCE  
in  
ELECTRICAL AND COMPUTER ENGINEERING**

Portland State University  
1989

TO THE OFFICE OF GRADUATE STUDIES:

The members of the Committee approve the thesis of Shiliang Wang presented February 28, 1989.



Marek A. Perkowski, Chairman

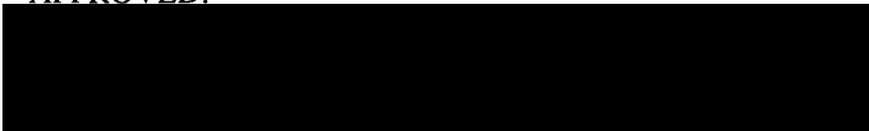


Rajinder P. Aggarwal

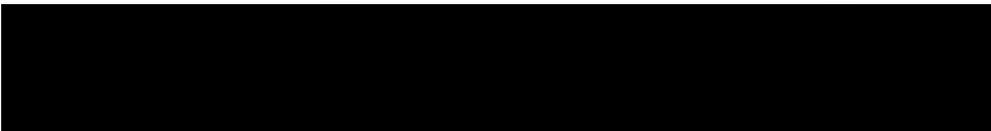


Bradford R. Crain

APPROVED:



Rolf Schaumann, Chairman, Department of Electrical Engineering



Bernard Ross, Vice Provost for Graduate Studies

## ACKNOWLEDGEMENTS

I want to give the best regards to my advisor Dr. Marek A. Perkowski who has been guiding and helping me all the time in the process of this research work. I would like to say thanks for his great confidence and persistence in the tremendous creative work, also for his enthusiastic and enlightening suggestions whenever I needed them. Dr. Perkowski's remarkable intelligence and serious working attitude have made a profound impression on me. It is really very informative and pleasant for me to work with him.

I also want to express my greatest gratitude to Dr. Q. Yu for her professional guidance and generous providing information while she was doing research as a WHO fellow in the Oregon Regional Primate Research Center. I would like to give my thanks and appreciation to Dr. D. P. Wolf for his informative suggestions and to Dr. P. Patton for the providing with the cervical mucus slides.

In addition, a number of people at Electrical and Computer Engineering Department gave instrumental help for this research work. In particular, Ms. Shirley Clark and Mr. Daniel Greenfield are warmly appreciated for their help and convenience in using utility equipment.

Special thanks to my wife Ronglin for her emotional supports.

Also I would like to acknowledge Ms. S. Word and Mr. R. Jory for their help and encouragement. Thanks to Mr. and Mrs. Ashley for their long term residential support.

Portland, Oregon

S. Wang

## TABLE OF CONTENTS

	PAGE
ACKNOWLEDGMENTS .....	iii
LIST OF TABLES .....	vi
LIST OF FIGURES .....	vii
CHAPTER	
I INTRODUCTION .....	1
1.1 General Problems of Ovulometry .....	1
1.2 The Existing and Proposed Ovulation Detecting Devices .....	4
1.3 Our Concept of Ovulometry .....	7
1.4 Ferning Test Automation .....	9
II CERVICAL MUCUS METHOD .....	11
2.1 Functions and Characteristics of Cervical Mucus .....	11
2.2 Mucus Types and Ferning Test .....	14
2.3 Ferning Image Characteristics and Stages .....	17
III FERNING TEST AUTOMATION SYSTEM AND HOUGH TRANSFORMATION .....	21
3.1 Ferning Test Preparation .....	21
3.2 The Ferning Pattern Recognition Method and Algorithm .....	24
IV AN EXPERIMENTAL PROGRAM OF HOUGH	

TRANSFORMATION AND FERNING	
FEATURE EXTRACTION .....	37
4.1 Dissection of the Program .....	37
4.2 The Simulative Results and the Evaluation of the Program .....	47
4.3 Conclusion .....	63
V PATTERN RECOGNITION PROBLEM WITH SPECIAL APPLICATION TO CERVICAL MUCUS PATTERNS .....	65
5.1 Approach to Automated Logic Classification .....	67
5.2 Approach to Statistic Pattern Recognition of Morphological Classification .....	69
5.3 A Program of Gaussian Classifier Application .....	77
VI CONCLUSION AND FUTURE WORKS .....	85
6.1 Conclusion .....	85
6.2 Our Future Works .....	86
REFERENCES .....	89
APPENDIXES	
A: THE PROGRAM OF THE HOUGH TRANSFORMATION APPLICATION .....	92
B: THE PROGRAM OF GAUSSIAN CLASSIFIER .....	102
C: SOME PREPROCESSED MUCUS IMAGES .....	112

## LIST OF TABLES

TABLE	PAGE
I Curves and Parameters .....	34
II Feature Description .....	66
III Description of Criteria for Classification .....	67

## LIST OF FIGURES

FIGURE	PAGE
1. Hormone Changes During a Normal Menstrual Cycle .....	2
2. Cervical Mucus Changes Corresponding to Hormone Changes .....	3
3. An Ovutimer Device .....	5
4. The Spinnbarkeit Characteristic of the Cervical Mucus .....	13
5. Three Types of Cervical Mucus .....	16
6. The Images of (a) Full-ferning, (b) Partly-ferning, (c) Non-ferning .....	18
7. Components of a Digital Image Processing System .....	23
8. A Block Diagram of the Hough Transformation and Detection System .....	25
9. The Hough Transformation with Slope-intercept Parameters .....	28
10. (a) Parameters of the Line $x\cos\theta+y\sin\theta=p$ (b) Curves Corresponding to Three Points on the Line .....	30
11. Flow Diagram of main() Routine .....	40
12. Flow Diagram of hough() and pmtr_spe() Subroutines .....	41
13. Flow Diagram of peak() Subroutine .....	43
14. Flow Diagram of longest() Subroutine .....	44
15. Flow Diagram of parallel() Subroutine .....	45
16. Flow Diagram of remap() Subroutine .....	46
17. The Four-line Binary Image .....	48

18. The Reconstructed Image with Threshold of 15 .....	53
19. The Reconstructed Image with Threshold of 13 .....	54
20. A Full-ferning Simulative Image .....	57
21. A Partial-ferning Simulative Image .....	61
22. A Linear-ferning Simulative Image .....	62
23. Flow Diagram of Logic Classification .....	68
24. A Morphological Pattern Recognition System .....	69
25. The Distribution of Four Classes of Mucus on F2-F4 Two-dimensional Space .....	71
26. The Distribution of Four Classes of Mucus on F1-F2 Two-dimensional Space .....	71
27. The Distribution of Four Classes of Mucus on F1-F6 Two-dimensional Space .....	72
28. The Flow Diagram of the Program of Gaussian Classifier .....	79
29. An Original Mucus Image .....	118
30. The Result of Applying the Median Filter on Figure 29 .....	119
31. The Result of Thresholding of Figure 30 .....	120
32. The Result of Thinning Procedure of Figure 31 .....	121

## CHAPTER I

### INTRODUCTION

So far, thousands of scientists have been making every effort to look for a better and more economical method for prediction of human ovulation. The prediction of ovulation has a significant meaning either in clinic or in medical research, in infertility treatment or in the everyday life natural method of family planning ([1], chapter 1). In recent years one can observe a great progress in the medical developments supporting natural methods of family planning: determining the ovulation moment, research on the properties of the cervical mucus and the conditions of sperm propagation in the female genital tract.

#### 1.1. GENERAL PROBLEMS OF OVULOMETRY

Certain hormonal changes occur during the normal menstrual cycle of a woman [2],[3]. Figure 1 illustrates how four types of hormones, FSH, LH, Progesterone, and Estrogen change during a normal woman's menstrual cycle. These hormonal changes are reflected by changes of many parameters of blood and other body fluids. Recent ovulometry methods is using these symptoms as methods to learn about those hormonal changes. They include:

- Assessment of the sex steroid levels in blood and urine, histological and histochemical examination of endometrial biopsy and laparoscopic examination of the ovaries for follicular growth [4]. For instance, ovulation can be determined by using daily serum LH, FSH, oestradiol ( $E_2$ ) radioimmunoassay. These methods are specific and accurate but require highly specialized techniques which are

costly and not universally available.

- Test the effects of the ovarian sex steroids on target organs and tissues such as vaginal epithelium and uterine cervix. Although they are good indices of ovarian functions, their accuracy and reliability have not been sufficiently established and there are conflicting reports on their validity [5], [6].
- Calendar (rhythm) and measurement of basic body temperature (BBT) has been used to detect normal ovulatory cycles ([1], chapter 7). These are the classic and simplest methods. But they have been found to be unreliable and grossly restrictive. Also, they tend to be disturbed by environmental and emotional factors [7].
- Tests of some other body fluids, like saliva, urine and lymph, reflecting changes of hormones in the blood. These methods can be reliable and accurate but also costly and not available in a clinic.

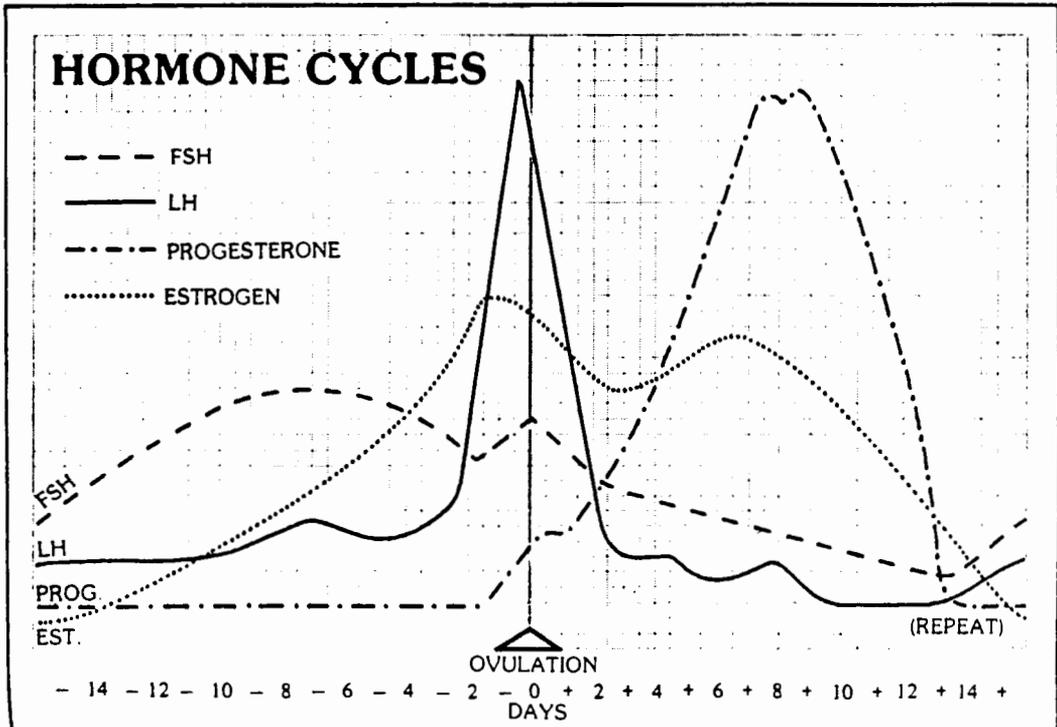
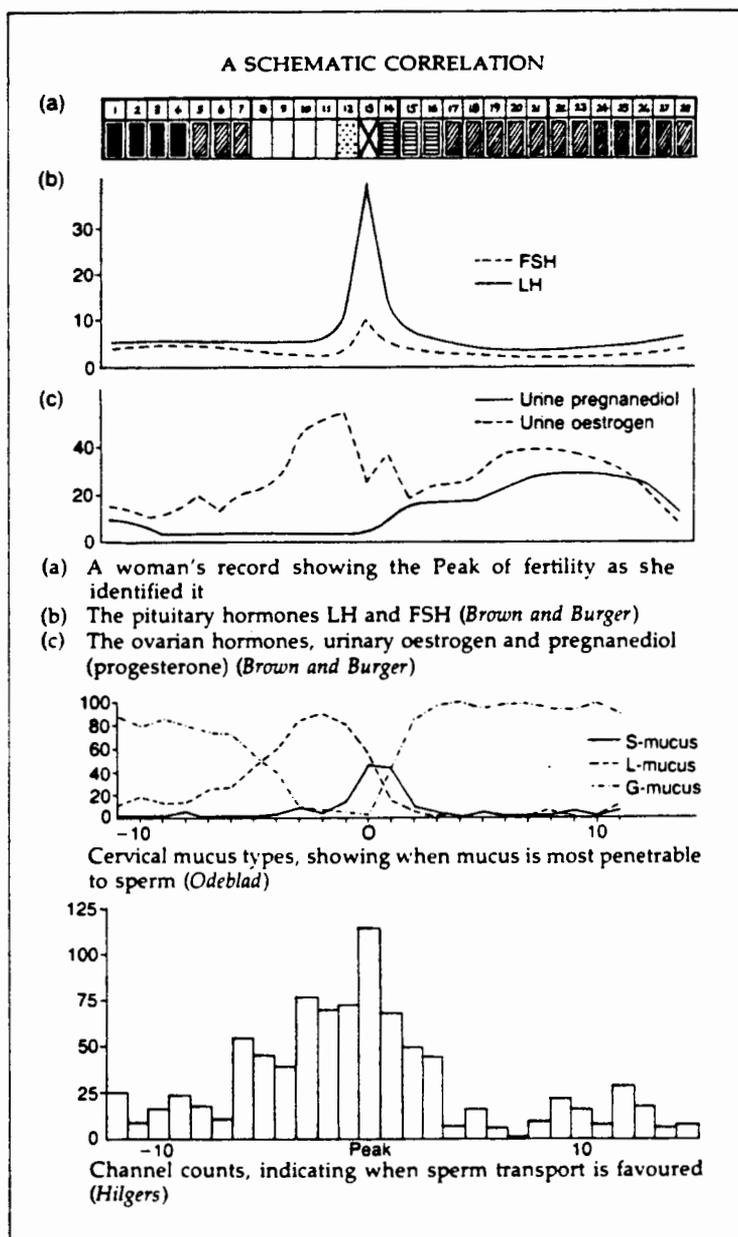


Figure 1. Hormone changes during a normal menstrual cycle.



**Figure 2.** Cervical mucus changes corresponding to hormone changes.

The methods mentioned above have their own advantages and various shortcomings, such as cost, difficulty to make measurements and interpret results, inaccuracy, emotional factors and laboriousness. The recognition of the importance of the cervical mucus as a marker of fertility is a finding of remarkable significance. A great discovery

has been achieved by Dr. John Billings [8]. The basis of the method, so called "Billings Method", is awareness of the mucus. This mucus can indicate whether a woman is fertile or infertile by its sensation and appearance. Figure 2 illustrates an obvious relationship between the secretion of hormones and cervical mucus (some terminologies in Figure 2 will be explained in Chapter II ).

The cervical mucus is produced by the cervix, which is the part of the uterus that joins with the vagina. The mucus is under the control of the reproductive hormones. It is suggested that a series of combinational determinations of cervical mucus characteristics can be used to detect normal ovulatory cycles of patients treated in clinics not having facilities for measurement of hormone secretions. It can be proposed to use a combination of the cervical mucus features with pattern recognition theory and statistic theory to make an accurate final decision of the prediction of the ovulatory moment.

## **1.2 THE EXISTING AND PROPOSED OVULATION DETECTING DEVICES**

In this section, some ovulation detecting devices which have been developed in the past will be introduced. Because of either high cost or inaccuracy, most of these devices have not been successful in the market or in the practical clinical application.

### Blood Test Device

Blood test is in current use. The method is called Radioimmunoassay, which has been described in the above section.

### Urinary Exam Kit

A urinary chemical examination kit has appeared in the market, which contains a sequence of procedures for home use. One kit can be used only for one month. It is quite expensive.

## Saliva Device

In this test, a woman chews a small piece of paraffin in order to stimulate saliva production. She then puts in her month a filter-paper test strip saturated with material which gives a visual color change from white to blue in the presence of alkaline phosphatase. This device has been able to predict ovulation within a period of one to seven days, which still makes it too inaccurate for widespread use.

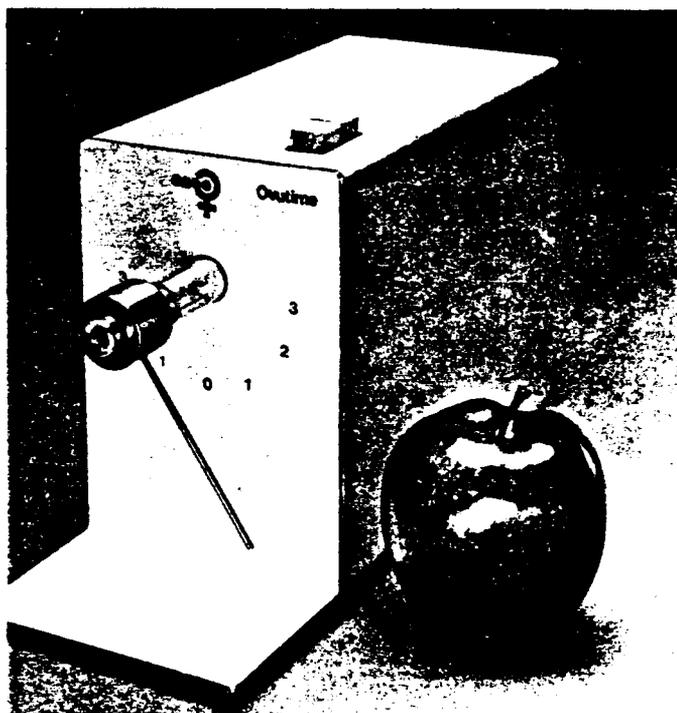


Figure 3. An Ovutimer device.

## Ovutimer device

This instrument called the viscometer, or "Ovutimer", is based on the consistency of the cervical mucus. It was introduced in April, 1976, by three Boston medical researchers and has been used by gynecologists as an office test.

A disposable probe is applied daily against the cervical os and a small sample of mucus is obtained. The probe is then inserted into the ring with the indicator arm

elevated (Figure 3). Before ovulation, when there is little mucus, the indicator arm will not move from the 3 position. However, during the fertile period - when the mucus is thin and not viscous - the arm will fall to the 0 or 1 position. That drop of the indicator arm may take place over four, five, or six days in succession, beginning with the first day of the fertile period. At the end of the fertile period, when the mucus becomes thick, the reading rises to between 2.5 and 4.

### The Ovulometer

The Ovulometer is a simple battery-run device that measures changes in the electrostatic current or voltage naturally present in the body. A woman using the Ovulometer gets an instant reading on a pocket-size meter by touching two electrodes to the index finger of each hand. According to Dr. Howard Lutz, one of its developers, the charge reads positive in all men throughout the month. Women, however, usually give a negative reading, but become positive 3 to 6 days prior to ovulation. Conversion back to negative occurs 24 to 48 hours after ovulation.

### Thermal Device

Another recent development that improved the temperature method is a special heat-sensitive transmitter that has been developed by National Aeronautics and Space Administration. This device can measure the intravaginal temperature prior to and after ovulation.

### Rhythm Based Devices

There is a device called Ritmograph using calendar rhythm only, and a better version of this type was developed by Sherman Brothers of Cambridge Research (Bridgeport, Connecticut) which takes into consideration information from earlier cycles.

### Chemical Kits

Some of the works of WHO lead to tests which can only confirm the ovulation, not predict it. The same type of research is done by Burger, Billings, and Wade. In England and in Ireland they use a test for glucose changes similiar to old Estridex test.

The works of Task Force on Methods for the Determination of the Fertile Period of the WHO Special Programme of Research Development and Training in Human Reproduction include developing assay kits and devices for predicting and detecting the time of ovulation (Speler, 1977). They are, or will be, based on mucus, saliva or urine.

### Other devices

Leonard Ravitz created ovutron which measures female electromagnetic field (or L-Field) which occurs before or after ovulation.

The clinical evaluation of biopotential meter was undertaken in 1976 and showed no significant or consistent pattern of change of polarity or potential which could be correlated with cervical mucus symptoms or the bbt shift associated with ovulation.

## **1.3 OUR CONCEPT OF OVULOMETRY**

As discussed in the previous section, to achieve the prediction of ovulation with the method of combinational cervical mucus features, an ovulometry system is proposed. This ovulometry system measures biochemical and rheologic parameters of cervical mucus. These parameters are analysed by using statistical and pattern recognitional theories. The final decision is based on this analysis. This is our concept of ovulometry.

Our final goal is to develop a microprocessor-based device for the measurement of various parameters related to the prediction and detection of ovulation. In this system the recent developments of endocrinology will be applied. Also, the decision is undertaken not with static rules for a single type of parameters, but with use of advanced

analysis of many parameters. This system will be a useful diagnostic instrument for endocrinologists and gynecologists.

The device would be also useful in sterility treatment. (Some 20% of married women have serious fertilization problems and this index is constantly increasing). It will be helpful in the supervision of the hormonal control of the first phase of pregnancy, and it can serve as an additional diagnostic tool in some gynecological diseases (e.g. cancer). Also, it can be a handy early pregnancy indicator which helps the woman to take special care of her health (e.g. avoiding pharmacotherapy, X-rays, etc).

This is an integrated collection of computer-based, measurement and decision-making systems. The specific measurements it aims are as follows:

- Mucus drying pattern also called crystallography or ferning test, which needs image processing and texture analysis
- pH and ions concentration, usually sodium, potassium, calcium and chloride
- Hormones (FSH, LH, Progesterone, Estrogens) in mucus
- Water comprised in cervical mucus by weight of gel
- Spinnbarkeit or stretchiness test for mucus viscosity (also called stringiness test)
- Quantity of the cervical mucus discharged
- Glucose concentration
- The concentration of nondislyzable solids (NDS) to viscoelasticity
- Other mucus parameters

Also, some very specific and accurate subjective symptoms can be observed by a woman in ovulatory phase, which can be effective input for our system. Those symptoms are subjective, as for instance, somewhere around the time of ovulation a pain in the area of one or both ovaries is felt by many women. The technically called "mittelschmerz" meaning pain in the middle. Another symptom is the inter-menstrual bleeding. Basal

temperature test also can be a basic input for our ovulometry system.

Measurements of all parameters except the ferning test are pretty straight-forward and have been well described in some literatures. Only ferning test automation leaves much to be desired. It is the main topic of this thesis and will be described in the following chapters.

The system uses several different independent rule-based algorithms to make the decision. The most important one is based on classical statistical pattern recognition theory in the feature space. The space is created for all measured parameters. Creation of the discriminant function is adaptive and takes into account changing patterns of the woman's cycle. The final decision is made from partial decisions on the safest basis.

#### 1.4. FERNING TEST AUTOMATION

An idea which has never been proposed before is created and well presented in this thesis. This idea is **ferning test automation** using the techniques of image processing and pattern recognition theories. The automation of ferning test is the main topic of this thesis and will be described in detail in the following chapters.

To my knowledge, the ferning test has never been automatized by other people. In my research, the Hough Transformation [9] was used to detect lines in the images of dry cervical mucus slides under microscope. Before the implementation of Hough Transformation, the routine procedures of video digitization and image processing should be done. These procedures will be discussed in the following chapters.

The pattern features of cervical mucus are extracted from a  $512 \times 512$  pixel digital binary image. These features are the measures of such quantitative concepts as various lengths of lines and the relationship or direction between different lengths of lines. The concept of "length" is expressed here as the number from each array's accumulators in the algorithm of Hough Transformation. The algorithm of Hough Transformation will

also be described in the following chapters.

A feature space, which is a set of measured feature vector values, is created. The extracted feature parameters are listed in the feature space. By examination of the pattern distribution in the feature space, the classifications of the ferning patterns are achieved.

In this automatic ferning test, a quantitative analysis was set up while the qualitative or subjective ones which have usually been used in diagnosis were abandoned.

## **CHAPTER II**

### **CERVICAL MUCUS METHOD**

Studies of the cervical mucus, particularly in relation to infertility research, have confirmed that the mucus must have special characteristics if sperm are to reach and fertilize an ovum [20].

This special characteristics give the fertile mucus its lubricativeness, and its stringy, raw egg-white appearance.

Ovulation occurs only one day in each menstrual cycle [8]. If more than one ovum cell is released, ovulation will still occur within this 24-hour span time. Under the influence of increasing levels of estrogen, certain cells in the cervix secrete the mucus which passes through the vaginal tract and is discharged at the vaginal opening. Through the studies of cervical mucus, some important functions and characteristics have been found [8], [10].

#### **2.1. FUNCTIONS AND CHARACTERISTICS OF CERVICAL MUCUS**

The secretion of the cervical mucus has its important biological purposes:

Sperm need special mucus to survive in a woman's body. Sperm without the special mucus die within a few hours. The acidic environment of the vagina is very hostile to sperm. Special cells in the vagina destroy sperm. With the special mucus of fertility, sperm may live three to five days. Because this mucus is alkaline, it provides a favourable environment in which sperm can live. It also nurtures the sperm [1], [8], [10].

The presence of cervical mucus provides a natural medium in which the sperm

can swim up the genital tract, through the cervical canal and into the uterus on the way to the fallopian tubes.

The cervical mucus has some important biochemical and rheologic characteristics.

The main constituent of cervical mucus is water, which comprises 85% to 98% by weight of the gel and demonstrates, in common with many of the other components of the mucus, a well defined variation in concentration related to the time of the menstrual cycle. The water content (hydration) variation is elicited by the level of the ovarian steroid hormones and it reflects the changes of the hormonal level. When the level of oestrogens is highest in the ovulatory phase, the mucus contains 95% to 98% water, while under progesterone dominance, in the follicular phase and during pregnancy, the level of hydration decreases from 92% to 85% [11].

The mucus contains a number of proteins, carbohydrates and electrolytes, many of which are common to serum. The levels of electrolytes in mucus have been measured and the most notable variations during the normal menstrual cycle are seen in the sodium and chloride levels. In most determinations the concentration of mucus sodium has been found to be maximal at mid-cycle when it approximates the sodium concentration in serum [12], [20]. In contrast, during the proliferative and follicular phases the level fall to 47% to 63% of that of the serum. If the level of sodium is expressed as a function of the dry weight of mucus, a peak still occurs just before ovulation and is followed by a rapid fall directly after ovulation [12], [13].

The concentration of chloride-ions is also maximized at mid-cycle while the level of potassium showed an inverse relationship, decreasing toward the mid-cycle. As a result, the ratio of chloride concentration to the sum of sodium and potassium concentration showed a smooth increase up to mid-cycle followed by a decline during the luteal phase of the cycle. This forms the basis of the cervical mucus chloride test, which is also

called the "ion test" [12], [13].

It was also found that an increasing concentration of glucose parallels the rising estrogen level in the preovulation phase. The detection of glucose by a glucose oxidase peroxidase test has been used to detect the preovulatory fertile phase [14].

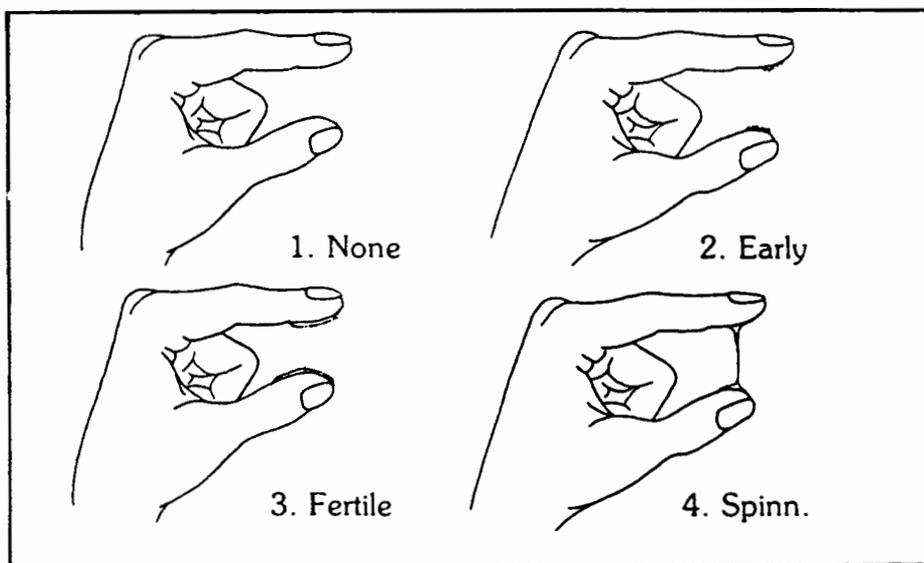


Figure 4. The spinnbarkeit characteristic of the cervical mucus.

For rheologic characteristics of cervical mucus, a great deal of research has been done by Dr. Wolf and his colleagues [13]. Quantitative viscoelasticity measurements were made on individual human cervical mucus samples by microrheometry. Increase in mean value for mucus ferning, spinnbarkeit (also called stretchiness or stringiness, see Figure 4) and daily amount of the mucus discharged were associated with the ovulatory phase of the menstrual cycle. A nadir in mucus nondialyzable solids (NDS) concentration and in viscoelasticity was seen at or near mid-cycle. When the contribution of NDS to viscoelasticity was minimized by data normalization or by sample reconstitution, a significant increase in viscoelasticity was associated with the ovulatory phase of the cycle.

Diagnosis from the cervical mucus can be an accurate method to predict ovulation [2], [15], [16]. Most of the tests are related to the biochemical and physical properties of the cervical mucus as parameters. The following parameters were measured by various authors, from which our system can make advantage:

- the days total amount of mucus
- sperm penetrability
- electrical impedance
- water content
- thixotropy
- moisture
- viscosity
- elasticity
- refractometric factors
- osmotic pressure
- other chemical properties

These parameters change in different phases of the menstrual cycle. The details of how to measure these parameters were introduced in some related literatures. Most of the above measurements can be found in Dr. Wolf's and Moghissi's literatures [13]. However, in this thesis one important and specific parameter, i.e. the features of the mucus image under microscope, which are related to the ferning test will be discussed and introduced in the next two sections.

## **2.2. MUCUS TYPES AND FERNING TEST**

Professor Eric Odeblad and his group [17] have investigated the biological and physical properties of the cervical mucus. They have demonstrated that three different

types of mucus are produced by specialised parts of the cervix during the menstrual cycle. This mucus production is under the control of hormones, in particular, estrogen and progesterone (see Figure 5)

The different types of mucus either impede or encourage the movement of sperm through the reproductive system. The relative amount of each type is crucial in determining a woman's state of fertility.

The mucus during the early infertile days is composed largely of protein fibres which form an impenetrable barrier to sperm cells. This barrier mucus is characteristically opaque and sticky, and it has been termed the G-type mucus.

The next type of mucus to appear is characterized by bead or "loaf" of mucus, giving it a thick, clumpy texture. If stretched between the fingers, this bead-like structure is evident. This type of mucus has been named the L-type. When spread on a glass slide and examined microscopically, this loaf mucus shows a flower-like arrangement of perpendicularly-branched crystals. It is a woman's awareness of this L-type mucus that signals a change from the basic infertile pattern. At first, this L-type mucus mixes with the barrier mucus (G-type), then it eventually replaces it completely. The L-type mucus has a number of very important functions: it neutralises the acidic vagina environment, so that sperm can survive. (Normally the vagina is inhospitable to sperm survival, and it is only in the presence of this protective mucus that sperm retain their ability to fertilize an ovum.) The L-type mucus also plays a part in trapping defective sperm cells. Another important function is to provide a structural support for the third type of mucus, known as S-type fertile mucus.

When the mucus can be stretched, a property of the S-type mucus, the L-type mucus is seen as bead or "loaf" at intervals along it (see Figure 5).

This S-type mucus indicates a high level of fertility and has a lubricative quality, resembling raw egg-white. Due to its lubricative nature, it quickly appears at the entrance

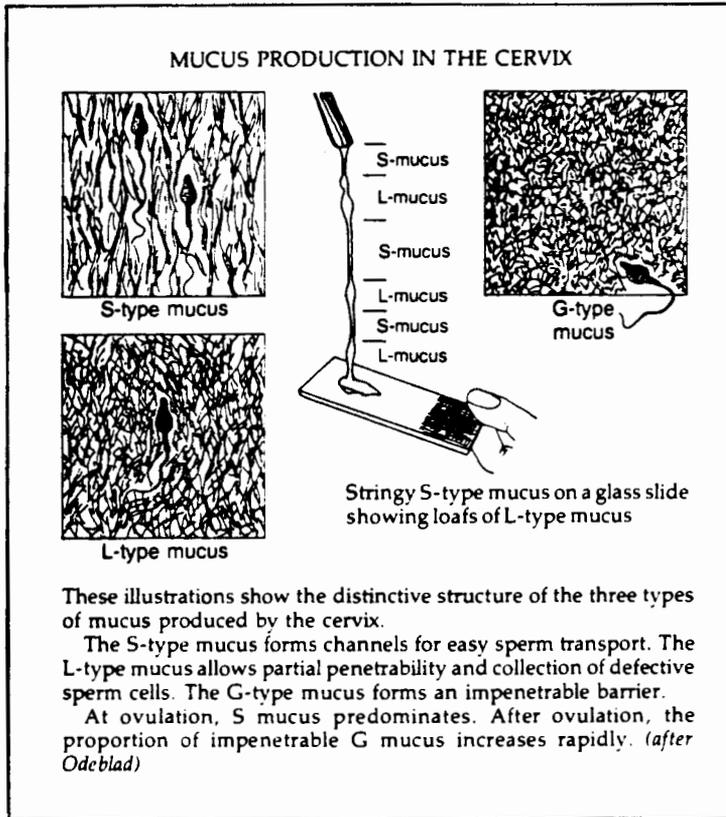


Figure 5. Three types of cervical mucus.

to the vagina, where it produces a sensation of wetness. If stretched it forms strings or loops. Women with normal fertility become aware of the onset of the L-type and S-type mucus on average six days before ovulation, with a range of three to ten days [7], [18], [19]. Some other women may observe or feel the presence of fertile-type mucus for only one half day, and only in some cycles.

The presence of this combination of the L-type and S-type mucus produces a characteristic ferning pattern when smeared on a glass slide. This fern-like visual image conception is the basis of the ferning test. On microscopic examination, the ferning pattern is seen to be due to the presence of channels within the S-type mucus. The nature of these channels and their function are to assist the rapid transport of sperm to the fallopian

tube. It has been found that as fertility increases during the fertile phase of the cycle, so too does the level of the ferning pattern stage seen under microscope. When the S-type mucus reaches its peak, the ferning pattern can be seen fully on the glass slide under microscope.

It has been found that the fertile S-type mucus usually begins to be replaced by the barrier G-type mucus (meanwhile, ferning pattern begins to reduce) just prior to the ovulation. However, some full ferning patterns persist for a day after ovulation, making possible the fertilization of an ovum for its entire lifespan, which is about twelve hours [8].

### **2.3. FERNING IMAGE CHARACTERISTICS AND STAGES**

As discussed above, the cervical mucus is composed of three types of mucus. The ratio among them varies during a woman's cycle. This variation is represented in the microscopic image as variant ferning patterns. When the G-type mucus is dominating, the mucus image is completely lacking in the ferning. When the L-type mucus begins to mix with the G-type mucus, some linear ferning with flower-like side branches attached to those linears can be seen in only a few spots of the microscopical image. Gradually, the L-type mucus replaces the G-type mucus and the S-type mucus begins to increase, good ferning with side branches in part of the image becomes visible. As soon as the s-type mucus reaches its peak, full ferning appears on the whole image.

One of the most important and specific features in cervical mucus image is the line-like feature, which represents the ferning or non-ferning characteristic. Figure 6(a) shows a full-ferning cervical mucus and Figure 6(b) shows a early ferning, while Figure 6(c) shows a non-ferning one. Ferning patterns are composed of rough lines with different length and different direction. The structure is of some main branches which are more straight and longer, and of some side branches which are shorter and not as straight

(a)



(b)



(c)

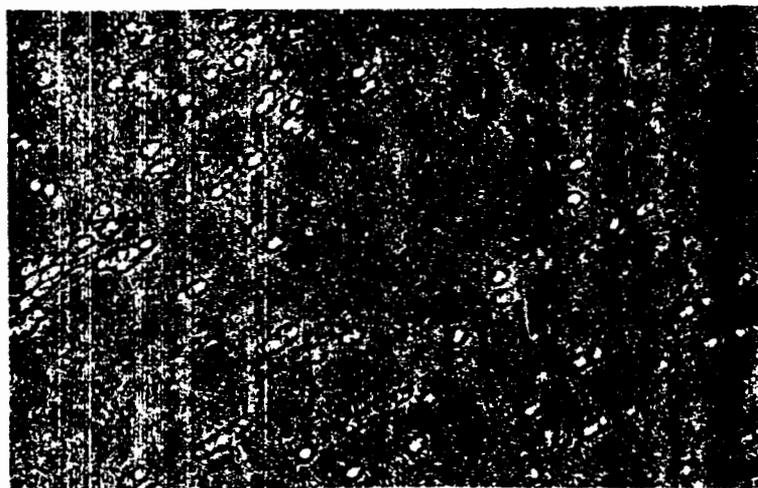


Figure 6. The images of (a) full-ferning, (b) partly-ferning, (c) non-ferning.

as the main branches. It is obvious that the more lines an image has, the higher level (or stage) of the ferning feature this cervical mucus image possesses.

The ferning characteristic stages can be introduced as following:

None ferning: Amorphous mucus.

Linear ferning: Fine linear ferning seen in a few spots. No side branching.

Partial ferning: Good ferning with side branches in part of the slide. Amorphous mucus in other parts.

Complete ferning: Full ferning of the whole preparation.

However, these definitions of mucus stages are quite subjective and qualitative. For a better recognition of these characteristic patterns, a objective and quantitative analysis is needed in our ferning test automation.

A mucus slide under microscope can be divided into several view-fields. If the total number of main branches and the total number of side branches from all view-fields can be counted separately, the ferning stage can be determined approximately. For a digitized image, counting these line-like branches is accomplished by counting those pixels which compose these lines ( so-called colinear pixels ) in a binary picture. Furthermore, the relationship between these lines can be also a significant feature to distinguish these ferning patterns (see Chapter III and Chapter IV). In this way, some numerical criteria for classification can be used to decide which stage a ferning image belongs to. Of course, these criteria of classification must be confirmed by a substantial number of experiments.

The classifications of ferning stages discussed above can only be achieved by analysis of large amount of pre-obtained data collected from well-experienced doctors or technicians.

There are two methods to approach the classification of ferning stages. One is by examination of the feature space and the distribution of ferning features. By recognizing

the region in which the one image features fall, the classification of ferning can be achieved. Another method is using a logic routine to distinguish the input patterns. These two methods will be presented and discussed in Chapter IV and Chapter V. The practical approach can be achieved only by collecting a large amount of data from those experienced doctors and technicians. Only after the necessary data have been obtained and analyzed, would these two methods be practically applied. This procedure will also be discussed in Chapter V.

Obviously, the classification of ferning patterns can never be achieved without close cooperation with some hospitals and clinics. A great deal of work must be done in association with experienced doctors and medical technicians before the final criteria of classification can be set-up.

In the next chapter, the Hough transformation will be introduced. By means of Hough Transformation, those variant lines which compose the fern-like images can be detected. Ferning patterns will be recognized by analysis of the parameters of those lines.

# **CHAPTER III**

## **FERNING TEST AUTOMATION SYSTEM AND THE HOUGH TRANSFORMATION**

### **3.1 FERNING TEST PREPARATION**

#### 3.1.1 Materials and Methods for Obtaining Cervical Mucus

Samples of cervical mucus were obtained from selected patients attending the gynecological clinic at Oregon Health Sciences University. The specimen of mucus was taken with a glass pipette from the lower portion of the endocervical canal, after gently cleaning the portio and external os with a cotton ball. Specimens tinged with blood were discarded. The gathered mucus was smeared on a glass slide for drying. The drying procedure was very simple, just put those slides in the air for about 30 minutes in the usual indoor temperature (sometime it takes less than 30 minutes, depending on the indoor condition). It should be noted that the drying slide must be kept from the dusty and contaminated circumstances. These slides were stored at 4° C for later observation. The observations were made using an Olympus IMT-2 100× – 200× microscope. There are some other methods to obtain cervical mucus [24], such as by direct aspiration of cervical contents into a 1-ml disposable syringe fitted with a disposable polypropylene cone tip. The tip is introduced into the cervical canal while negative pressure is maintained in the syringe. Another method is especially used as evaluation of features of other than the ferning test [25]. This is, by inserting a chemically inert, sterile, soft polyurethane sponge in the lower portion of the endocervical canal and gently pressing against the cervical os. After obtaining the cervical mucus, the sponge can be weighed and washed to gather

other information, such as electrolytes and water.

A mucus slide placed under the microscope is divided into several view-fields. Each view-field is processed by the digitizing system separately.

### 3.1.2 Proposed Equipment for Digital Image Processing

In this section we will describe a digitizer system called PCVISIONplus which will be used in our future work.

The digitizing equipment consists of the television scanning microscope system, a PCVISIONplus Frame Grabber system, an external monitor display and an IBM Personal Computer (PC AT). The PCVISIONplus Frame Grabber is a video digitizer and frame memory capable of digitizing standard RS-170/330 (or CCIR) video input and storing the digitized image signals in a special on-board frame memory. The image can be simultaneously displayed on a video monitor. The PCVISIONplus Frame Grabber is a board placed into one of the IBM PC-AT slots (also IBM PC-XT or 100% hardware compatibles). It is driven by appropriate software, allows to perform complex digital image processing functions, such as: image averaging; image subtraction; convolutions; and edge-enhancing algorithms. But more importantly, it will allow to perform the ferning test automation.

### 3.1.3 How Digital Image Processing Works in PCVISIONplus

The fundamental components of an image processing system are shown in Figure 7. The image originates at a video source, normally a standard RS-170/330 or CCIR camera. The analog signal produced by the camera is then transformed to a digital format through a process called digitization. The digitization process involves taking samples of the analog signal at discrete time intervals and converting each individual sample, or pixel, to a digital value. The digital data is then stored in the frame memory. This memory stores one pixel in each of its memory locations. The frame memory is

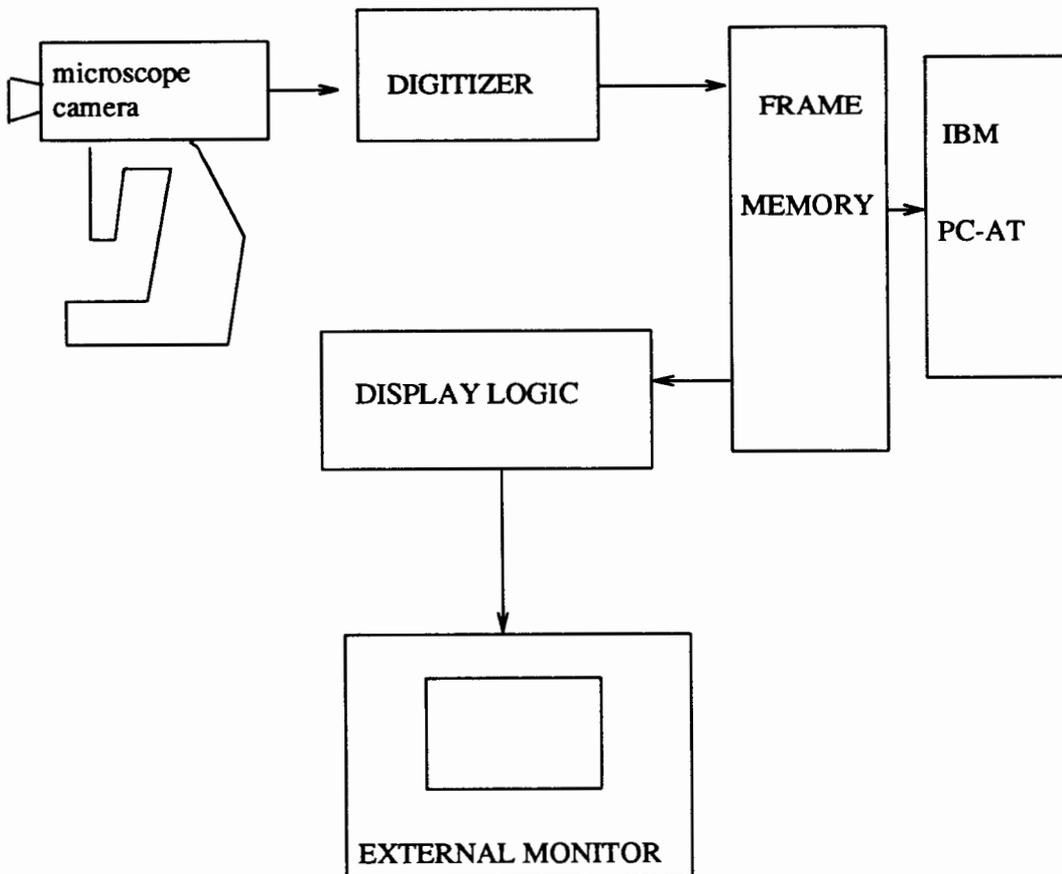


Figure 7. Components of a Digital Image Processing System.

accessible by the CPU in the personal computer.

It is also possible to view the image stored in the frame memory. Display logic transforms the pixels stored in the frame memory back into an analog format so that the image can be viewed on a monitor.

The heart of this PCVISIONplus is the Frame Grabber. It is a single board that plugs directly into an expansion slot in the IBM Personal Computer. It digitizes the incoming video signal to eight bits of accuracy at a rate of 30 frames (60 fields, 30 odd and 30 even) per second, and stores the resulting pixels in frame memory. Each pixel in the frame memory is one of 256 possible intensities or grey levels. Two 512×512 images or one 640×512 image can be stored. Display logic on the Frame Grabber converts the

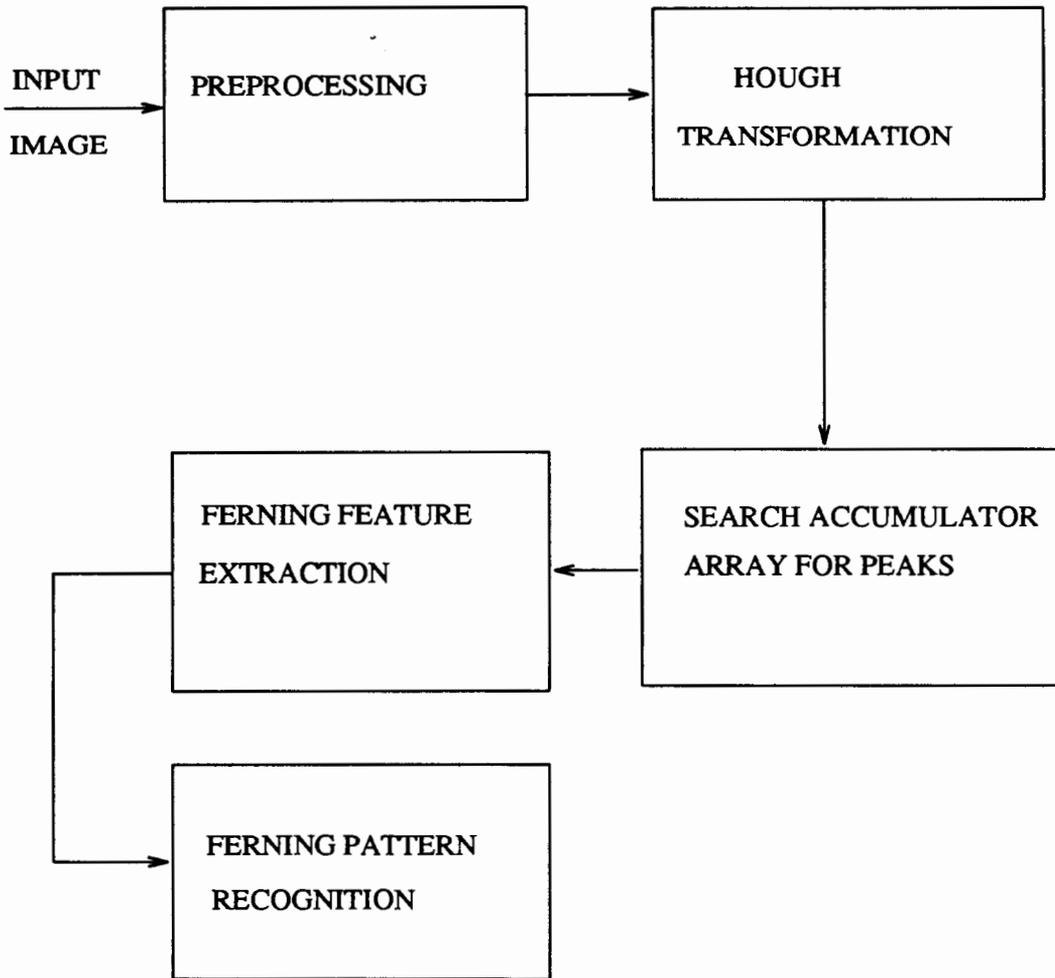
pixels in the frame memory back to an analog RS-170 format for display on a video monitor. Three output channels are provided for pseudocolor display. The input digitization path and each output channel contains eight Look-Up Tables. The Look-Up Tables can be used to change the grey-level of each pixel individually. These 32 tables each contain 256 entries. They are provided for performing transformation of the 256 intensity levels. Simple point transformations can be performed without any calculation or processing delay. For instance, we can program eight different thresholds in the Look-Up Tables of input digitization path to perform eight different thresholding. This is referred to as real-time processing.

### **3.2 THE FERNING PATTERN RECOGNITION METHOD AND ALGORITHM**

In Chapter II, we have introduced the ferning characteristics and stages. As was mentioned, ferning patterns are composed of rough lines with different length and different direction. The method and algorithm we propose to detect these lines is that of Hough Transformation. Our ferning pattern recognition system is proposed in Figure 8. In the following sections this system will be explain in details.

#### **3.2.1 The Preprocessing of Ferning Image**

Images first have to be preprocessed to distinguish between possible line points and a background. Preprocessing techniques which are usually applied in other image processing applications are: edge detection, gradient methods or contour detection. But for the ferning image in our system, the preprocessing is to distinguish those different grey level pixels from the pixels which represent the fern-like lines. This means that before application of Hough Transformation, an image with various grey levels must be transferred into a binary image with only two grey levels: 0 and 1. Therefore, a zero-one threshold operator, and how to decide this threshold automatically, will be introduced in the following sections, which will also be included in the future work.



**Figure 8.** A block diagram of the Hough Transformation and detection system.

After an image picture has been digitized into a set of computer signals and is ready for terminal screen display, assume it is transformed into a  $n \times n$  pixel matrix specified by  $f(x,y)$ ,  $x,y=0,1,2,\dots$ , in which each value of  $f(x,y)$  is integer and nonnegative and has been obtained by quantizing a video signal into one of several grey levels (in PCVISIONplus System there are  $2^8=256$  grey levels). This picture function  $f(x,y)$  may be transformed into another  $n \times n$  function  $t(x,y)$  via the operator  $T[\ ]$ , i.e.  $t(x,y)=T[f(x,y)]$  where the described operator is applied to all pairs  $(x,y)$ .

Zero-one threshold operator:

$$\begin{aligned}
 t(x,y) &= T[f(x,y)] \\
 t(x,y) &= 0, \quad \text{for } f(x,y) < h \\
 t(x,y) &= 1, \quad \text{for } f(x,y) \geq h
 \end{aligned}$$

where  $h$  is a grey level threshold. Among 256 grey levels,  $h_{\max}=255$  represents the grey level of the brightest pixels and  $h_{\min}=0$  that of the darkest ones. Since in a ferning image the fern-like branches are of black pixels, the value of  $h$  used may be chosen as 15 (the grey levels from 0 to 15 represent dark dots which are difficult to be distinguished by human visual capability). In the practical cases, the value of threshold  $h$  can be altered according to the contrast quality of the image. For a high contrast image, the value of  $h$  can be chosen lower than 15, and, vice versa.

For a completely automatic system, the automated decision of a threshold must be taken into account ([21], Chapter 7). Suppose it is known a priori that an image contains two principal brightness ( this is the case of the ferning image ). The histogram of such a picture may be considered as an estimate of the brightnesses probability density function,  $p(x)$ , where  $x$  is the value of grey levels. This overall density function would be the sum or mixture of two unimodal densities, one for the light and one for the dark in the image. If the form of the densities is known or assumed ( in our case it is assumed to be the normal distribution ), then it is possible to determine an optimal threshold for segmenting the image into two brightness regions.

Suppose that an image contains two values combined with additive Gaussian brightness noise. The mixture probability density function is given by

$$p(x) = P_1 p_1(x) + P_2 p_2(x),$$

which, for the Gaussian case, is

$$p(x) = \frac{P_1}{\sqrt{2\pi}\sigma_1} \exp\left[-\frac{(x-\mu_1)^2}{2\sigma_1^2}\right] + \frac{P_2}{\sqrt{2\pi}\sigma_2} \exp\left[-\frac{(x-\mu_2)^2}{2\sigma_2^2}\right]$$

where  $\mu_1$  and  $\mu_2$  are the mean values of the two brightness levels,  $\sigma_1$  and  $\sigma_2$  are the stan-

dard deviations about the mean, and  $P_1$  and  $P_2$  are the a priori probabilities of the two levels. If the variances are equal,  $\sigma^2 = \sigma_1^2 = \sigma_2^2$ , a single threshold  $h$  is sufficient:

$$h = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{\mu_1 - \mu_2} \ln\left(\frac{P_2}{P_1}\right)$$

Above is a brief conclusion of the optimal threshold approach [21]. This approach has been developed and utilized in applications by Chow and Kaneko [22].

In our future work, to obtain accurate results, another preprocessing called **thinning** may be recommended. The thinning processing is used as the preprocessing technique to extract the skeleton of the structure in the character and fingerprint recognition [23]. Thinning algorithms in our system will be used to extract the skeleton of the structure and preserve the connectivity of the skeleton.

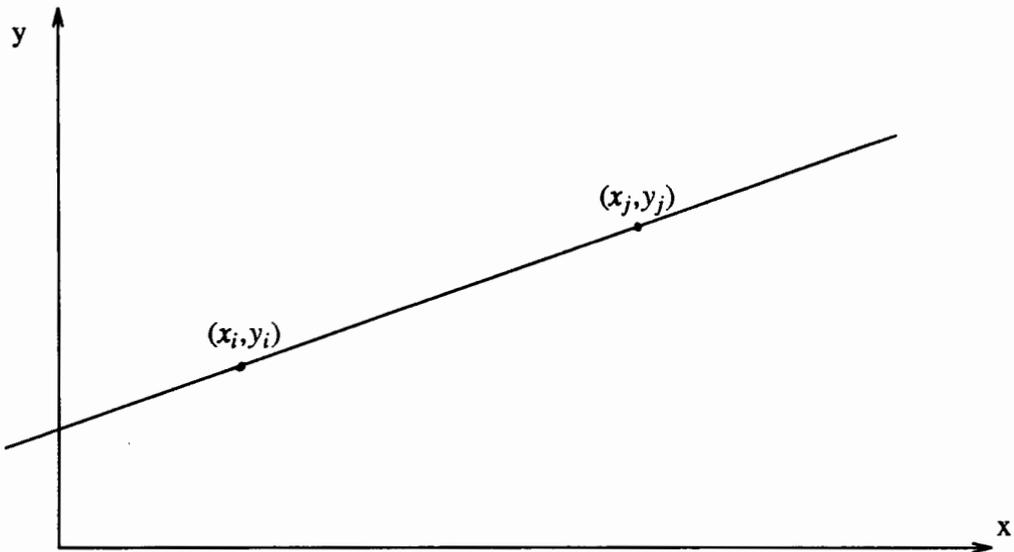
Two thinning algorithms, fast parallel thinning algorithm [24] and 2D thinning algorithm preserving 8-point and 4-point neighbor connectivities [25], can be used. The reason for which the thinning preprocessing is recommended in this thesis is that the line segments in a ferning image always have a certain width and also the ferning branches are of zigzag edges. After the thinning preprocessing, an image with neat line segments will be obtained. For more details about thinning processing, the reader can look to references [21], [24], [26], [27].

### 3.2.2 A Review of Hough Transformation

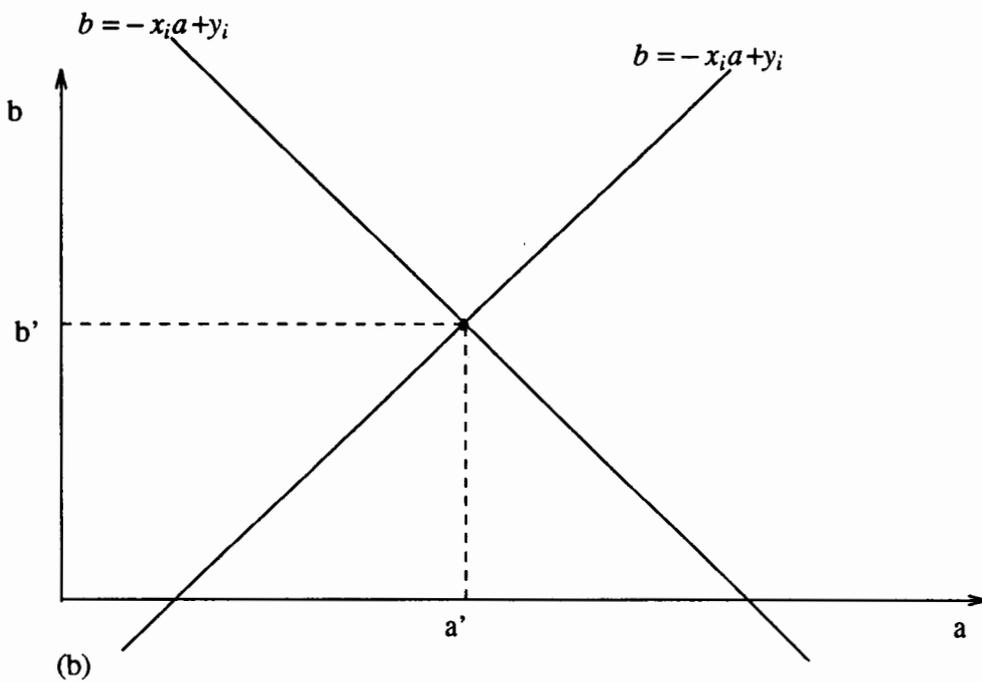
The detection of straight line-segments in images is a problem that often occurs in image analysis. This detection of collinear points is possible, among others, with the Hough Transformation [9], described by Rosenfeld [28]. Consider a point  $(x_i, y_i)$  and the general equation of a straight line in slope-intercept form:

$$y_i = ax_i + b \tag{3.1}$$

There is an infinite number of lines that pass through  $(x_i, y_i)$ , but they all satisfy the equation (3.1) for varying values of  $a$  and  $b$ . However, if we write this equation as



(a)



(b)

Figure 9. The Hough Transformation with slope-intercept parameters.

$$b = -x_i a + y_i$$

and consider the a-b parameter plane called slope-intercept space, then we have the equation of a single line for a fixed pair  $(x_i, y_i)$ . Furthermore, a second point  $(x_j, y_j)$  will also have a line in parameter space associated with it, and this line will intersect the line associated with  $(x_i, y_i)$  at  $(a', b')$ , where  $a'$  is the slope and  $b'$  the intercept of the line containing both  $(x_i, y_i)$  and  $(x_j, y_j)$  in the x-y plane. In fact, all points contained on this line will have lines in parameter space that intersect at  $(a', b')$ . These concepts are illustrated in Figure 9. Points in the image are transformed in this method into lines in a slope-intercept space. Lines in the slope-intercept space corresponding to collinear points will cross each other in one point. This point defines the slope and intercept of the line through the collinear points. Quantizing the slope-intercept space into cells and counting the number of lines crossing each cell, reduces searching for collinear points in the image to looking for the cells in the slope-intercept space, which are locally maximum.

However, a problem with using the equation  $y = ax + b$  to represent a line is that both the slope and the intercept approach infinity as the line approaches the vertical position.

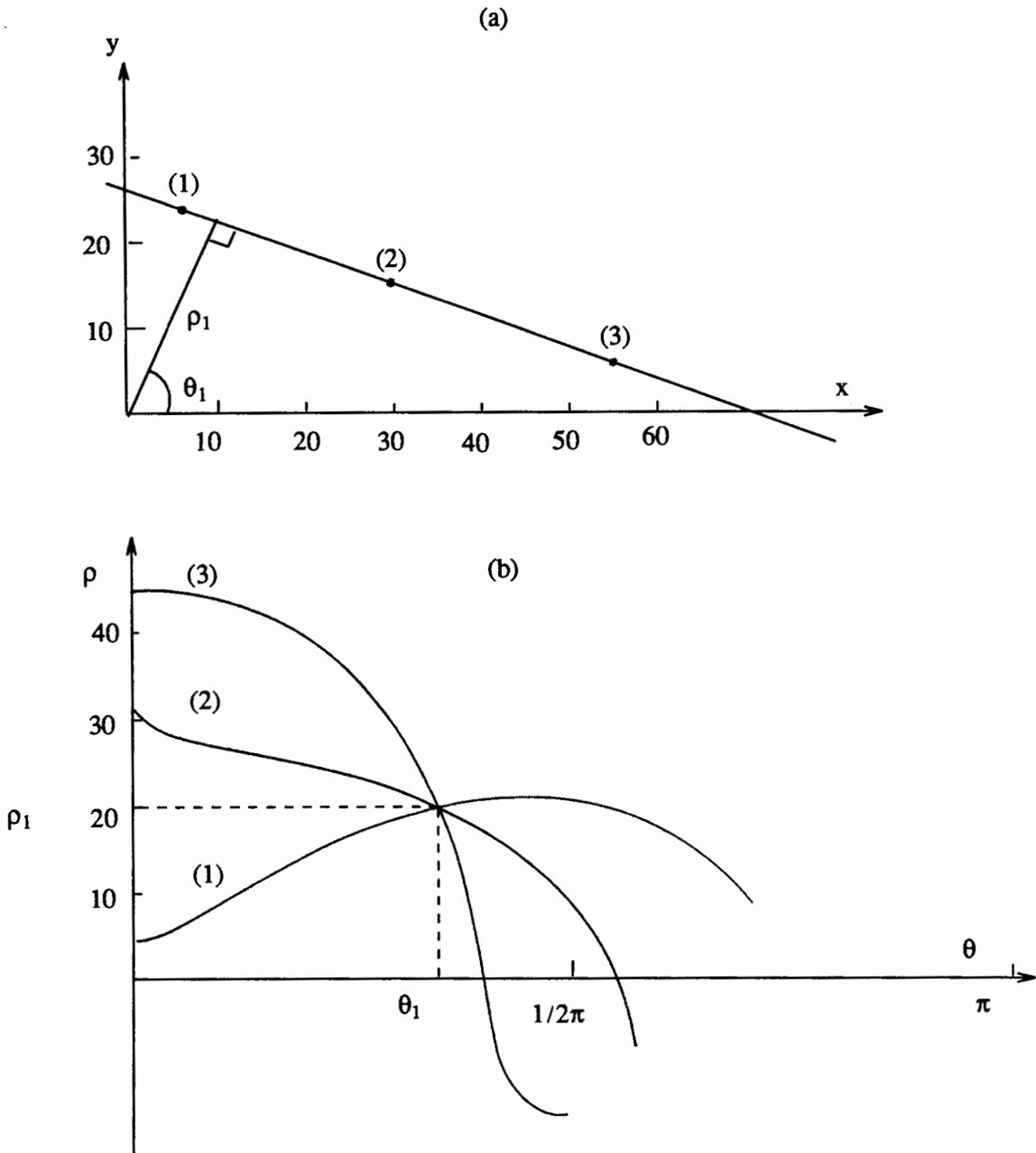
The method was improved by Duda and Hart [9] by using an **angle-radius** parametrization (also called **normal parametrization**), avoiding therefore the problem of the unboundedness of slope and intercept. With this improvement, collinear points in the image show up now as peaks in the angle-radius space.

An experimental program accomplishing this transform using angle-radius parametrization as well as implementation of T[] operator preprocessing, peak search in parameter space and parallel-line feature extraction will be described in Chapter 4.

Duda and Hart used the normal parametrization of a line in an image given by

$$\rho_1 = x_i \cos \theta_1 + y_i \sin \theta_1 \quad (3.2)$$

where  $\rho_1$  is the distance of the line to the origin,  $\theta_1$  is the angle between the normal and



**Figure 10.** (a) Parameters of the line  $x\cos\theta + y\sin\theta = \rho$ .  
 (b) curves corresponding to three points on the line.

the  $x$ -axis and  $(x_i, y_i)$  are points on the line (see Figure 10(a)).

For a given point  $(x_i, y_i)$ , this point can be mapped onto  $\rho$ - $\theta$  plane as a sinusoidal curve. Also, if we restrict  $\theta$  to interval  $(0, \pi)$ , then the normal parameters for a line are

unique. With this restriction, every line in x-y plane corresponds to a unique point in the  $\theta$ - $\rho$  plane. Furthermore, the parameters of all lines going through a point  $(x_i, y_i)$  in the image constitute a sinusoidal curve in the  $(\rho, \theta)$  space, given by

$$\rho = x_i \cos \theta + y_i \sin \theta \quad \pi > \theta \geq 0 \quad (3.3)$$

(see Figure 10(b))

The sinusoidal curves corresponding to collinear points of a line with parameters  $(\rho_1, \theta_1)$  will cross each other in the point  $(\rho_1, \theta_1)$  in the  $\rho$ - $\theta$  space, as for these points  $(x_i, y_i)$  equation (3.3) holds. So line parameters can be obtained from the crossing-point of sinusoidal curves in the parameter space. An example is illustrated in Figure 10 showing the three collinear points in x-y plane mapped into  $\rho$ - $\theta$  plane as three curves with a common intersection  $(\theta_1, \rho_1)$ .

Thus the problem of detecting collinear points can be converted to the problem of finding concurrent curves.

We can summarize some interesting properties of the point-to-curve transform as follows:

**Property 1.** A point in the picture plane corresponds to a sinusoidal curve in the parameter plane.

**Property 2.** A point in the parameter plane corresponds to straight line in the picture plane.

**Property 3.** Points lying on the same straight line in the picture plane correspond to curves through a common point in the parameter plane.

**Property 4.** Points lying on the same curve in the parameter plane correspond to lines through the same point in the picture plane.

On an image, the points which compose the ferning lines are called **feature points**. As the images are processed by computer both the image and the  $(\rho, \theta)$  space

are quantized. On the image, feature points will be undertaken in the transform, which are checked to be collinear. The feature points (e.g. the points composing the ferning branch) are found in the preprocessing stage.

The Hough Transformation of feature point  $(x_i, y_i)$  is performed by computing  $\rho$  from equation (3.3) for all  $n$  values of  $\theta_k$  into which  $\theta$  is quantized ( $0 \leq \theta < \pi$ ). The values of  $\rho$  are then quantized in  $m$  intervals of width  $\Delta\rho$ . In this way a quantized sinusoidal curve is obtained (in which, strictly speaking,  $\theta$  is sampled and  $\rho$  is quantized) and along the quantized curve each cell is incremented with an equal amount, for instance one. This procedure is repeated for all feature points. Collinear feature points in the image show up as peaks in the  $\rho$ - $\theta$  space.

Exactly colinear subset of feature points can be found, at least in principle, by finding coincident points of intersection in the parameter plane. Unfortunately, this approach is essentially exhaustive (we will explain it later), and the required computation grows linearly with the number of feature points. When it is, however, not necessary to determine the line exactly, i.e. when a certain quantizational errors are allowed, the computational burden can be reduced considerably.

Following Hough's basic proposal, we specify the acceptable error in  $\theta$  and  $\rho$  and quantize the  $\theta$ - $\rho$  plane into a grid. This quantization can be confined to the region  $\pi > \theta \geq 0, R > \rho \geq -R$ , where  $R$  is the size of the quantized  $\theta$ - $\rho$  plane, since the points outside this rectangle correspond to lines in the picture plane that do not exist in the view field. The quantized region is treated as a two-dimensional array of accumulators. For each point  $(x_i, y_i)$  in the picture plane, the corresponding curve given by (3.3) is entered to the array by incrementing the count in each cell along the curve. Thus, a given cell in the two-dimensional accumulator eventually records the total number of curve passing through it. After all feature points have been treated, the array is inspected to find cells having high counts (they are called **peaks**). If the count in a given cell  $(\theta_i, \rho_i)$  is  $k$ , then

precisely  $k$  feature points lie ( within quantization error ) along the line whose normal parameter are  $(\theta_i, \rho_i)$  .

In the application of Hough Transformation, one problem must be handled with care: the quantization of the parameter space, which may either influence the discretization errors ( **scatter of the peaks** ) or the burden of computation ( **oversampling** ). T.M.van Veen and F.C.A.Groen have analyzed the discretization errors of the Hough Transformation [30]. According to their theory, neither oversampling nor undersampling occurs when

$$\Delta\rho \approx \tau \sin\left(\frac{1}{2}\Delta\theta\right)$$

where  $\tau$  is the length of a line segment in picture space,  $\Delta\rho$  and  $\Delta\theta$  are the quantized intervals of  $\rho$  and  $\theta$  separately. In the experimental program in the next chapter this equation will be used to decide the quantization of  $\rho$ .

Let us investigate how the computation required by the accumulator method varies with the number of the feature points. To be more specific about the quantization, suppose that we restrict our attention to  $n_\theta$  values of  $\theta$  uniformly spaced in the interval  $[0, \pi)$ . Suppose further that the  $\rho$  axis in the interval  $[-R, R]$  is quantized into  $n_\rho$  cells. For each feature point  $(x_i, y_i)$ , we use (3.3) to compute the  $n_\theta$  different values of  $\rho$  corresponding to the  $n_\theta$  possible values of the independent variable  $\theta$ . Since there are  $N$  feature points, we need to carry out this computation  $N \times n_\theta$  times. When these computations are complete, the  $n_\theta \times n_\rho$  cells of the two-dimensional accumulator are inspected to find high counts. The total number of computations  $N_c$  is then:

$$N_c = N \times n_\theta + n_\theta \times n_\rho$$

Clearly, the computation required grows linearly with the number of feature points.

Since in some cases the ferning lines can be some kinds of curves, they can also be detected by using Hough's basic theory with different parametrization. In principle, the transform method extends to arbitrary curves but the curve type which should be

expressed by certain equation could be difficult to determine, especially when limited computation is desired. It is very important to have a small number of parameters since the accumulator implementation requires quantization of the entire parameter space and the computation grows exponentially with the number of parameters.

Table I describes four kinds of curves together with their parameters.

TABLE I  
CURVES AND PARAMETERS

analytic form	parameters	equations
line	$\rho, \theta$	$x \cos \theta + y \sin \theta = \rho$
circle	$a, b, r$	$(x - a)^2 + (y - b)^2 = r^2$
parabola	$a, b, c$	$(y - b)^2 = 4c(x - a)$
ellipse	$a, b, m, n$	$\frac{(y - b)^2}{m^2} + \frac{(x - a)^2}{n^2} = 1$

In dealing with those pictures with random noise and lines composed of non-adjacent black points, two noise removal-techniques were recommended by M. Cohen and G.T. Toussaint [31]. The first method is applicable when the distribution of the noise is known and the other one can be used when it is not. Also, O'Gorman and Clowes [32] have proposed a gradient weighted Hough Transformation which reduces the number of feature points in a window not belonging to the line of interest. The noise-removal techniques mentioned above may be applied in our practical ferning pattern recognition system in the future.

In the last few years, Hough Transformation has been applied in many fields. D.

H. Ballard has generalized the Hough Transformation to detect arbitrary shapes [33]. Ianino and Shapiro have presented a review of detection of circular and parabolic segments [34]. G. C. Stockman et al have presented how to build hardware implementation inferred from software techniques [35]. Teresa M. Silberberg has described the Hough Transformation on the Geometric Arithmetic Parallel Processor [36]. F. Matthew et al have designed a monolithic Hough Transformation processor based on restructurable VLSI [37]. Hungwen Li et al have developed a fast algorithm for Hough Transformation with k-tree data structure which can be applied to design respective hardware and VLSI chip [38].

All the achievements mentioned above will be applied in our ovulocomputer system to build up necessary hardware and chips with the most appropriate algorithm and design techniques, if the practical results of the Hough Transformation itself proves sufficient comparison to experienced doctors when recognizing ferning patterns.

Now let us see how the different ferning patterns can be recognized after the Hough Transformation has been applied and those lines composing ferning branches have been detected. Two methods are used to extract ferning features. They are described as following:

Examining Figure 6(a),(b) and (c) in chapter 2, it is easy to see that images with ferning patterns have different numbers and different lengths of lines. By using Hough Transformation with certain count array, those lines can be detected separately. For instance, we may set three criteria  $n_1, n_2$ , and  $n_3$ , Here  $n_1, n_2$  and  $n_3$  represent the criteria of the length of the ferning main branches, the length of the full ferning side branches and the length of the partial ferning side branches, respectively. If we obtain in a picture N accumulator peaks which are bigger than  $n_1$ , M accumulator peaks which are bigger than  $n_2$ , and P accumulator peaks which are bigger than  $n_3$ , we can determine that the picture with bigger M+N has a more ferning stage than that with a smaller M+N.

If a picture has a small  $M$  and a rather big  $P$  but is almost lacking in  $N$ , we may determine that it is in a partial ferning stage. A picture whose  $M, N, P$  are almost zero indicates that it is a non-ferning picture. The details of how to use these ferning features to classify ferning categories will be discussed in Chapter VI.

Also, a comparison of Figure 6(a), (b) and (c) in Chapter II indicates that an image with a full ferning pattern has many more side branches which are parallel to each other than those with a partial or linear ferning patterns. Perhaps, parallel lines are one of the most significant characteristics of the full ferning image. In a picture space these parallel lines can be expressed as in the equation (3.3), with the same  $\theta$  but different  $\rho$ 's. After all feature points are transferred into the parameter space and the accumulator arrays are inspected to find out the cells with peaks. The cells found with peaks are located, that represent these detected lines. Among these cells, if the coordinates of any of them in the  $\theta$  axis are of the same quantized value of  $\theta$ , then the corresponding lines are parallel to each other. In another words, the peaks with the same  $\theta$  in the parameter space define the parallel lines (or approximately) in the picture space. The experimental program from the next chapter will demonstrate how these two features are extracted by using the methods described above.

## CHAPTER IV

### AN EXPERIMENTAL PROGRAM OF THE HOUGH TRANSFORMATION AND FERNING FEATURE EXTRACTION

An experimental program in C language is presented in this chapter. The program will be explained by using one test image of four-line and three artificial mucus images. Among them, the four-line image is just a testing image to examine the routine and the subroutines of this program and the other three images, which are similar to real ferning images under microscope, are for the demonstration of the ferning feature extraction.

#### 4.1 DISSECTION OF THE PROGRAM

The program has one **main()** routine and several subroutines. They are explained as follows:

In the **main()** routine there are some definitions of variables and some fundamental procedures.

- A monochrome image is a two-dimensional light-intensity (grey level) function which is defined as a two-dimensional array  $\text{pix}[\text{IMY}][\text{IMX}]$ . Since the size of the artificial image is  $80 \times 80$ , so the size of the array is defined as  $\text{IMY}=80$ ,  $\text{IMX}=80$ .
- Define by  $\Delta\theta$  the quantizing interval of dimension  $\theta$ :  $\Delta\theta=3^\circ$ , so the total number of samples within  $\pi > \theta \geq 0$  is  $180/3 = 60$ . Since the estimated minimum length of a line in those picture is of 10 pixels, so the maximum length of this 10-pixel line is  $\sqrt{10^2+10^2} = 10\sqrt{2}$  (actually even with the same pixels, the length of a line can be varying). Using the discretization error analysis theory [30] discussed in

Chapter 3, according to the equation:

$$\Delta\rho = \rho \sin\left(\frac{1}{2}\Delta\theta\right)$$

we have

$$\Delta\rho = 10\sqrt{2}\sin\left(\frac{1}{2}\times 3^\circ\right) \approx 0.4$$

For the convenience of computation, we define  $\Delta\rho = 0.5$ .

- The  $\rho$ - $\theta$  space is defined as 2-dimensional accumulator array  $ac[PMTR\_Q][PMTR\_P]$ , here  $PMTR\_P$  is the size of the array in  $\rho$  axis (or the number of elements in  $\rho$  axis) of  $\rho$ - $\theta$  space and  $PMTR\_Q$  is the size of the array in  $\theta$  axis. Therefore,

$$PMTR\_P = \frac{\sqrt{IMX^2 + IMY^2}}{\Delta\rho} \times 2 + 1 \approx 453$$

$$PMTR\_Q = 180/3 = 60$$

- Since the accumulator  $ac[][]$  is a matrix, it needs positive addresses (coordinates). Therefore, each  $\rho_i$  value has to be recalculated to a new coordinate,  $\rho_i'$ .  $\rho_i$  and  $\rho_i'$  are given by the relation,

$$\rho_i' = \rho_i + zero$$

where

$$zero = \frac{(PMTR\_P - 1)}{2} + 1 = 227$$

- Each feature point in the picture space has to be computed  $PMTR\_Q$  times and each time the  $\cos()$  and  $\sin()$  mathematical functions have to be called once from the  $\langle\text{math.h}\rangle$  library of the C compiler. Therefore, instead of calculating this tedious procedure, a table containing all necessary values of  $\sin()$  and  $\cos()$  was put in the **main()** routine. Each time the calculation is executed, the transform routine needs only to find the  $\sin()$  and  $\cos()$  values from the table. This method tremendously reduces the time for calculation.

- Prompting a question to require for the input the name of the image file to be processed.
- Initiating all the elements in the accumulator array `ac[][]` to zero value.
- Call `hough()` subroutine.
- Call the `preprocess()` subroutine to perform the image preprocessing (refer to Section 3.2.1).

The `hough()` routine which is called by `main()` routine performs most of the interactive tasks, such as, input the threshold for search of peaks, input the file name to store the data which are the results of transformed and the results of processed features, etc. It implements the following procedures:

- Call `pmtr_spe()` subroutine to perform the computation of the Hough Transformation.
- Call `peak()` subroutine function to scan the accumulators in parameter space, looking for those peaks which satisfy the input threshold.
- Call `remap()` subroutine. Since this is an experimental simulative program, it is necessary to examine how the detected lines diverse from the actual ones in the original image. The `remap()` subroutine reconstructs the detected lines into an image again so that this image can be compared with the original one. The `main()` routines described above are illustrated in Figure 11.

The `hough()` and `pmtr_spe()` subroutines are presented in Figure 12.

A complete Hough algorithm is implemented in `pmtr_spe()`, `peak()`, and `remap()` routines.

In `hough()` routine, during the scanning of the picture area, each time when a feature point is found, the `pmtr_spe()` is called once.

After all the feature points in the picture area are transformed into the parameter

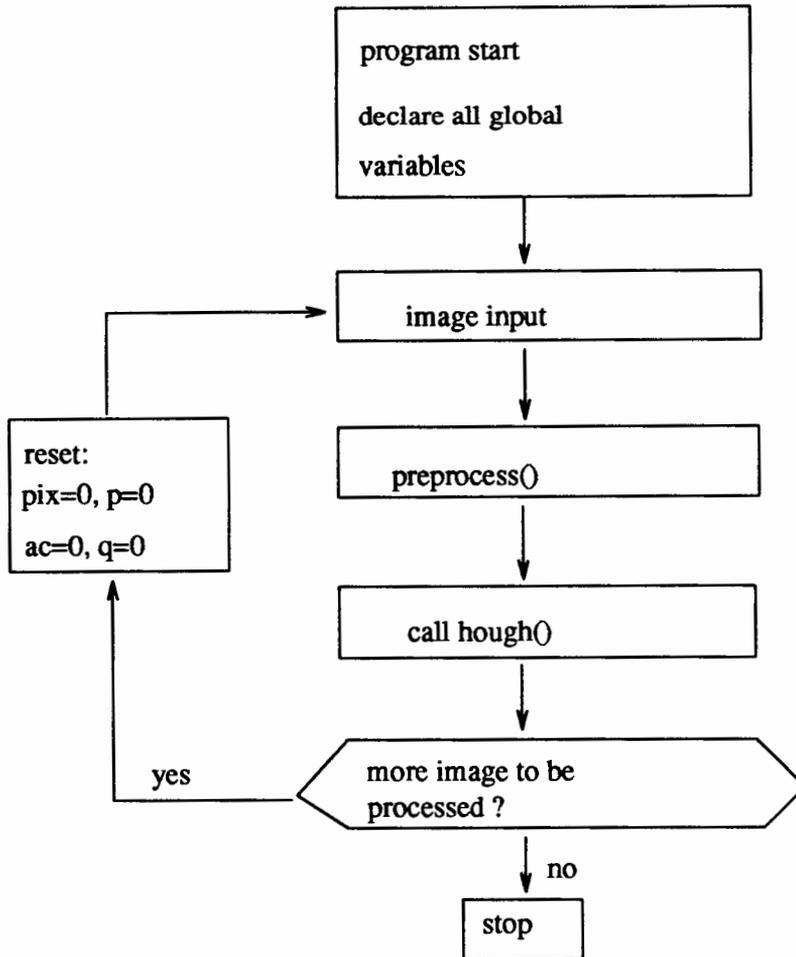


Figure 11. Flow diagram of main() routine.

space, the accumulator array  $ac[][]$  records all the intersections of the curves. The analysis of the accumulator array is performed by **peak()** subroutine using the algorithm presented in Figure 13.

The subroutine **peak()** inspects all the cells in the accumulator array using the threshold to obtain the peaks. The location of each peak (coordinates of  $\rho_i, \theta_i$ ) defines the lines in the picture area. The **peak()** also calls two subroutines **longest()** and **parallel()**. The **longest()** subroutine is used simply to find the maximum peak, which is the longest line in the picture area. The **parallel()** is a very important subroutine for the

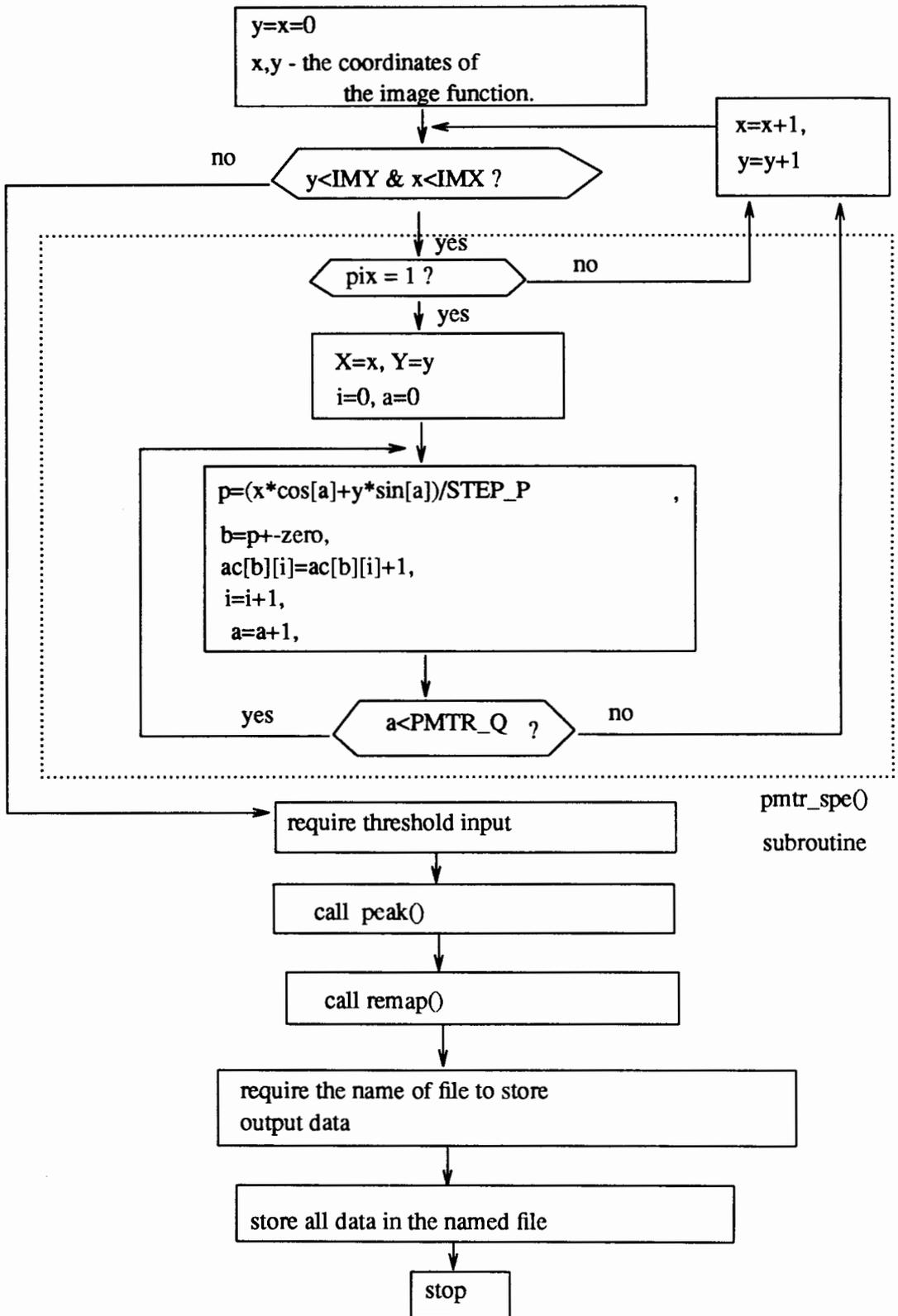


Figure 12. Flow diagram of `hough()` and `Pmtr_spec()` subroutines.

extraction of the ferning features. It has been discussed in Section 3.2.2 in Chapter 3 that all the peaks in the parameter space with the same  $\theta$  coordinate define a set of parallel lines in picture space. The purpose of subroutine **parallel()** is to search for those peaks that have the same  $\theta$  coordinate. Its procedure is as follow:

- Subsequent cells in the accumulator array were analyzed in a loop.
- Two one-dimensional arrays  $p[]$  and  $q[]$  were used as pointers to store the  $\rho_i$  and  $\theta_i$  of each peak found. Here, the values stored in  $p[]$  and  $q[]$  were transferred parameters referred to the picture space, i.e. they have been transferred from coordinate of  $\rho$ - $\theta$  space into the actual values of line parameters in picture space.
- The total number of the peaks was assigned to the variable  $k$ .
- Print out every peak value (here "peak value" is used for the value of the accumulator found to be a peak) and its converted  $\rho$  and  $\theta$  values according to picture space.
- Call **longest()** subroutine.
- Call **parallel()** subroutine.

The **longest()** and **parallel()** subroutines are dissected in the following: (see Figure 14 and Figure 15)

After all peaks which exceed the THRESHOLD have been obtained in the first loop of **peak()** routine, the coordinates of every peak are assigned to the two one-dimensional arrays,  $p[]$  and  $q[]$ . These values are passed to the subroutine **longest()**. The value of the first peak is assigned to a variable  $max$  while the coordinates of this peak are assigned to the variables  $mp$  and  $mq$ . Each time through the loop the value of each peak is compared with the prior ones and the bigger one is assigned to  $max$  while its coordinates are assigned to  $mp$  and  $mq$ , i.e.  $max$ ,  $mp$  and  $mq$  is updated in each time of the loop. At the end of the loop, the variable  $max$  eventually obtains the largest value among

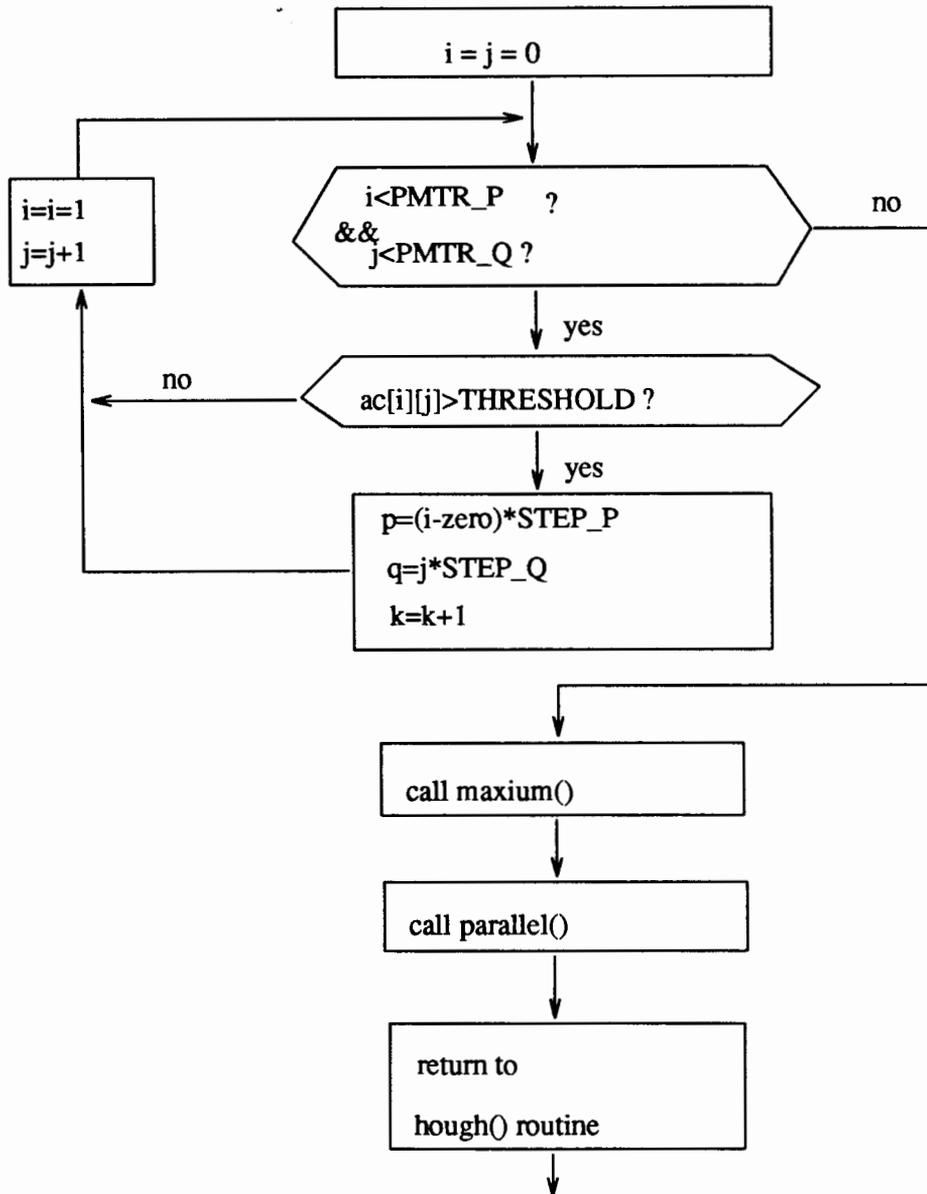


Figure 13. Flow diagram of peak() subroutine.

the peaks, which indicates the longest line in the picture space (in our simulated images, it refers to the main branch of the ferning)

The **parallel()** subroutine detects the parallel lines in an image. It searches for those peaks with the same  $\theta$  coordinate in the accumulator array:

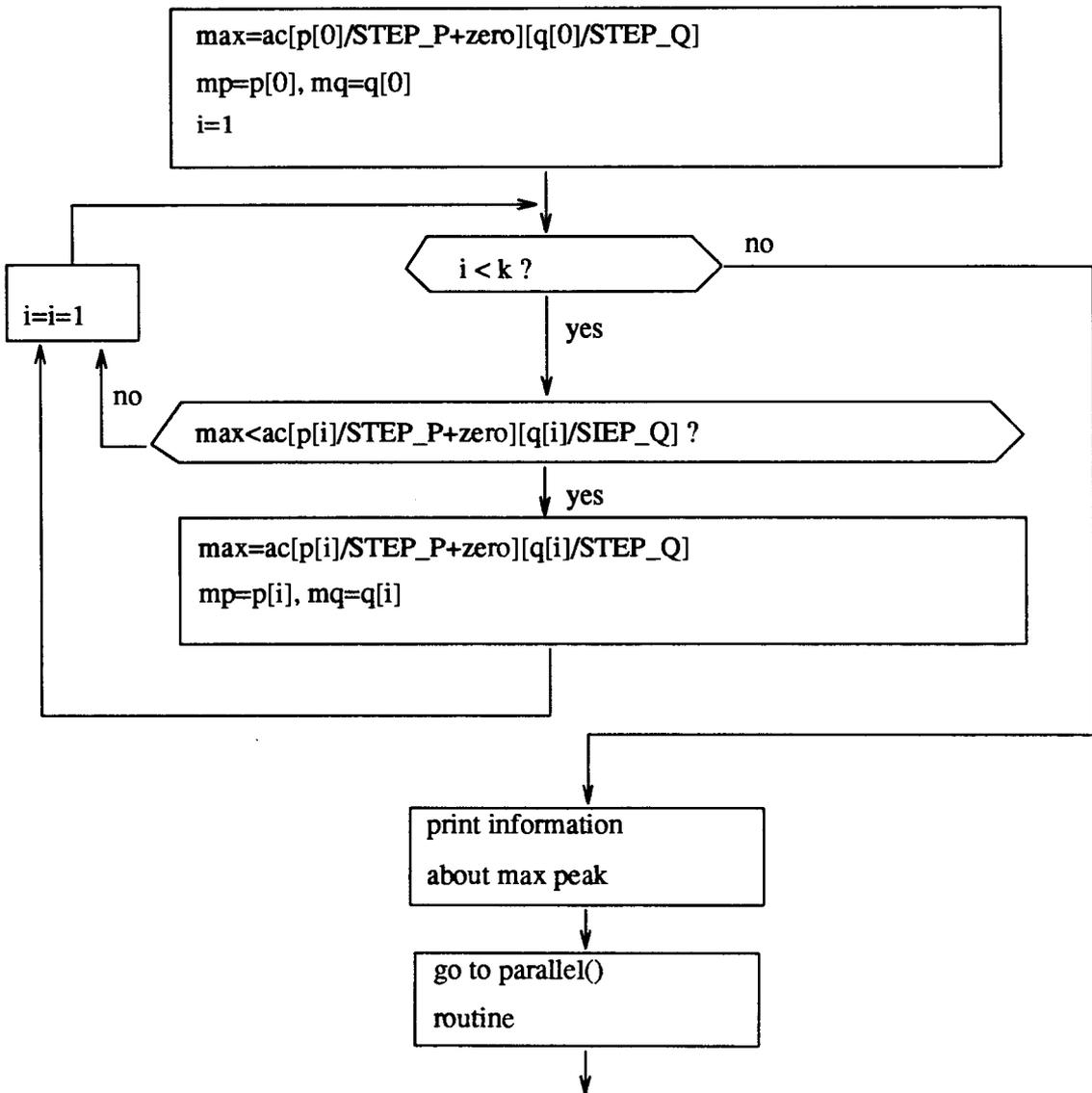


Figure 14. Flow diagram of longest() subroutine.

- The variable K is assigned the number of sets of parallel lines.
- One-dimensional array Q[] is used to store the  $\theta$  value of each set of parallel lines. The size of Q[] equals to the number of the sets of parallel lines, K.
- One dimensional array pl[] is assigned the number of parallel lines in each set, corresponding to each  $\theta$  in which each set of parallel lines exists. Obviously, the size of pl[] equals to the size of Q[] array.

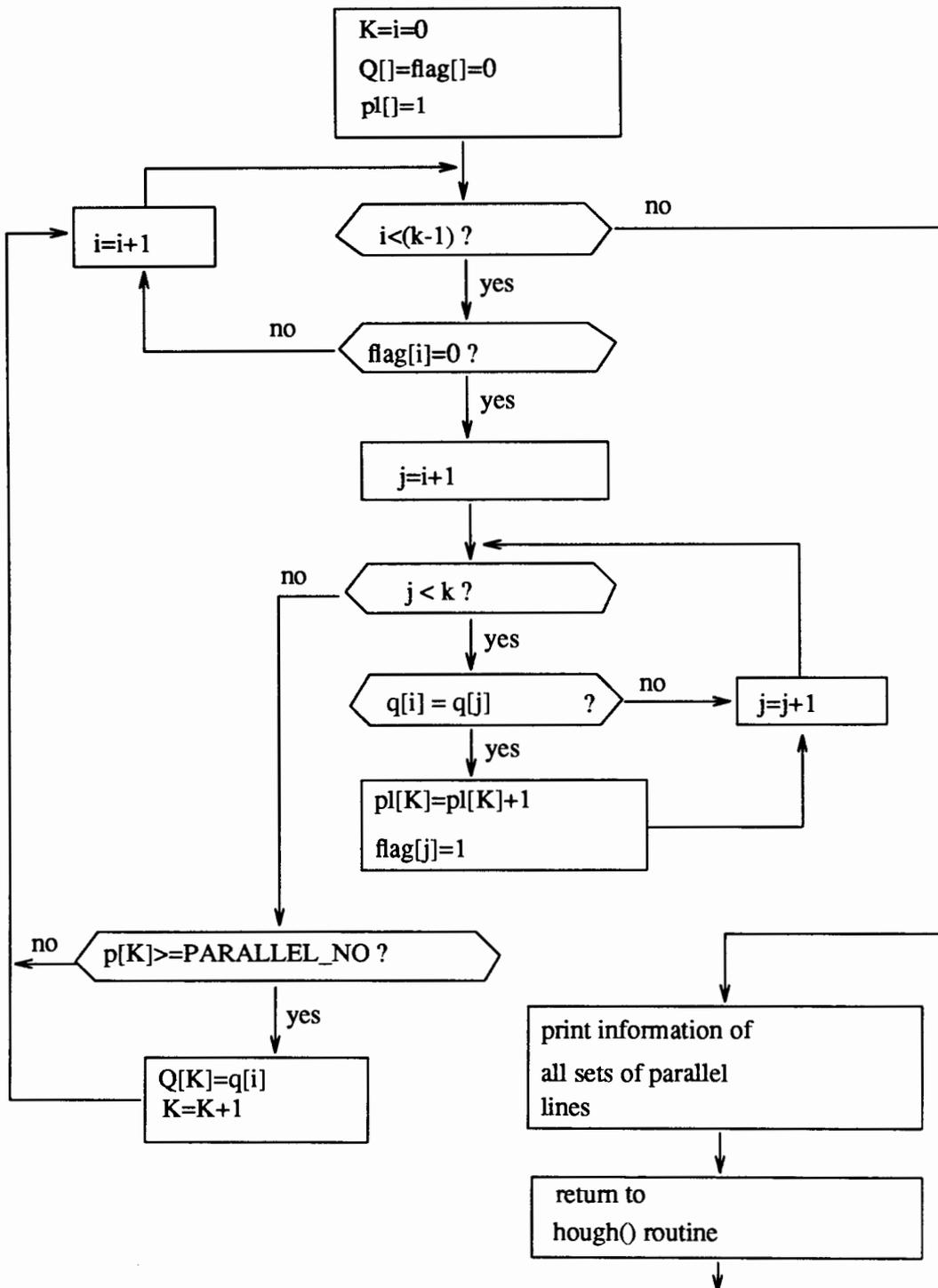


Figure 15. Flow diagram of parallel() subroutine.

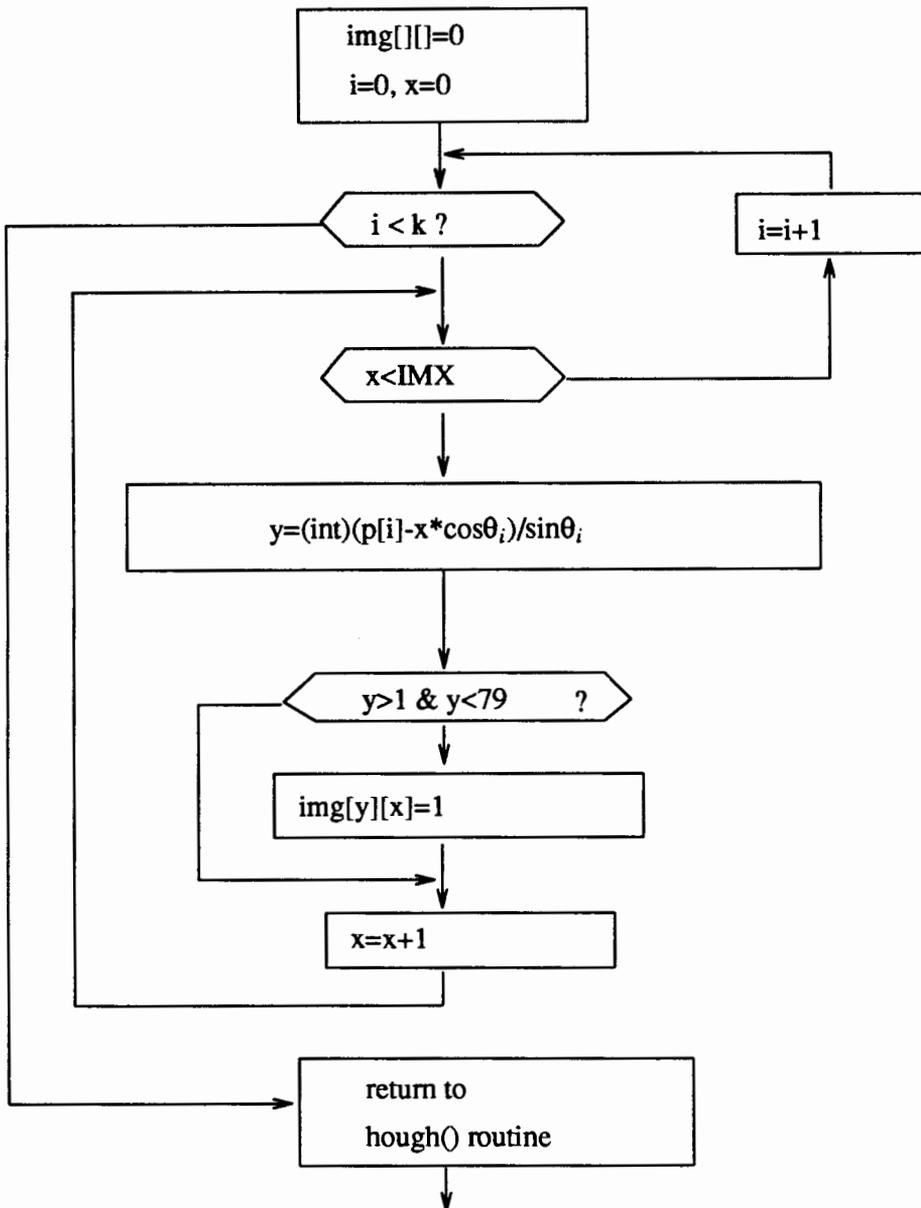


Figure 16. Flow diagram of remap() subroutine.

- flag[] array is used as a label for those peaks of which the  $\theta$  values are found to be the same as the previous ones.
- PARALLEL\_NO variable is another threshold. Since in an image there are always certain number of parallel lines detected, sometimes we are only interested in those sets of which the number of parallel lines are larger than this

threshold.

Figure 14 and Figure 15 illustrate how the subroutines **longest()** and **parallel()** proceed.

The last subroutine that **hough()** calls is called **remap()**. The **remap()** simply uses the values of  $\theta$  and  $\rho$  of detected peaks as parameters to reconstruct an image. It performs the reverse transformation using the equation

$$y_i = \frac{\rho_1 - x_i \cos \theta_1}{\sin \theta_1}$$

In our  $80 \times 80$  -pixel binary image,  $x_i = 1, 2, \dots, 80$ , for each pair of  $\theta$  and  $\rho$  values the computation has to be carried out 80 times to obtain  $y_i$  values. ( during the computation, the  $y_i$  values are usually the double type variables, they have to be rounded into the type of integer). After each time of calculation, a pair of  $x_i$  and  $y_i$  is obtained and the corresponding pixel is set to one. The pixels which are not corresponding to any coordinates of the calculated  $x_i$  and  $y_i$  are set to zero. These zero-pixels compose the background of the reconstructed binary image.

The array `img[][]` is assigned as reconstructed binary image function, just like the original image function `pix[][]`. Figure 16 presents the algorithm of the **remap()** subroutine.

## 4.2 THE SIMULATIVE RESULTS AND THE EVALUATION OF THE PROGRAM

### 4.2.1. An Example of a Four-line Image Processing

A four-line image is shown in Figure 17. By visual inspection, we can realize that these lines are composed of 64, 62, 62 and 18 feature points with  $\rho$  values of -2.0, 0.5, 28.5 and 57.5 and  $\theta$  values of  $135^\circ, 135^\circ, 135^\circ$  and  $45^\circ$ , separately. Since the total number of pixels in the image is  $80 \times 80 = 6400$  and only  $64 + 62 + 62 + 18 = 206$  of them are

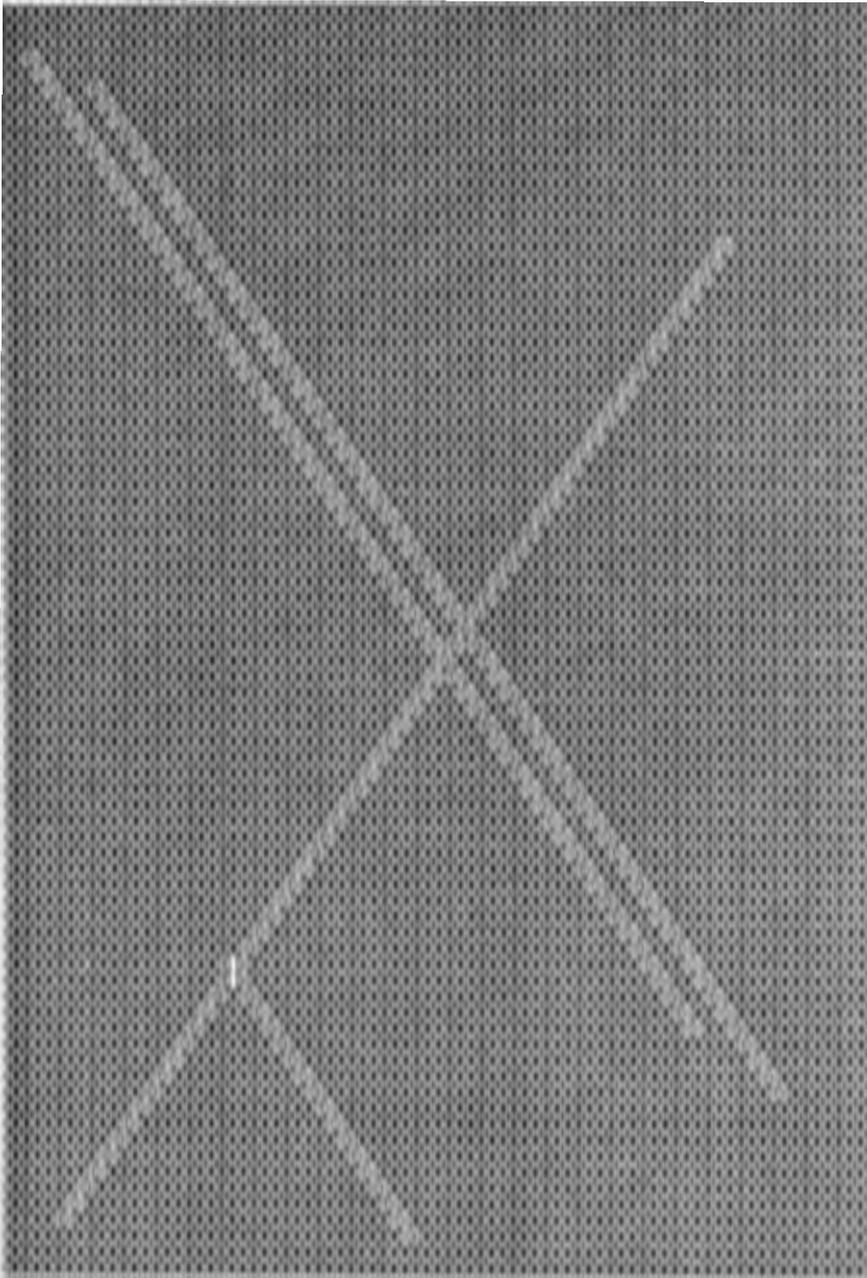


Figure 17. The four-line binary image.

feature points, it means that 3.2% pixels in this image are feature points. When this program is run on SUN machine in EE Department, the time taken for the Hough Transformation and accumulator adding was about 3.45 seconds, the search for the peaks and



011030200111122221101012020101011202212230001114010100000000  
4222010210100000000101102021223222322223411100031000100000000  
011121101111100011111020101010111113222300000140010010000000  
43222011010111222110101001112332121322232011104010100000000  
442120201110110011110102021012322202322230000011000000000000  
001202020111122110010202011210222322222311111000101100000000  
441120002101000000111201020112311121223310000040101000000000  
0022212001111121211011002121022213112200000171010100000000  
4311130210101110100101102002123112211333111110700000100000000  
0121202011011000001111020211210221233223000000111010000000000  
422213020111122222101010201202311311221300000110010100000000  
022121001010000000101100210310220212233111110001000100000000  
4222112011111000011111020012113012320222000000010010000000000  
022222020101122221101020210212032201222200001110010100000000  
42111130211100000001102011121312133323311110000101000000000  
023123010110100001210110202212021321222200001801000100000000  
41122011101112222100111002212131200133311111110010000000000  
0322023011010000001111020012120213220220000000001010000000000  
412232120111100011110102112213121212120000000110001000000000  
04321020101011222101011101312302111333311111100010000000000  
4012131111010000001011012001103123212220000010001010000000000  
04322111011111012111020032113022112133000007011010000000000  
40123131201011121001010200222032211221211118100100000000000  
042312130211100000111102212130111212220000030010010000000000  
402021211001111222101111022110321213333100000110010000000000  
041313111110110000111012102220122212200111110001010000000000  
403120130110100001110020023111321133110000020001000000000000  
04123302011122221101102111122011101231111030110010000000000  
4033120301010000000111102212213211212200001510001000000000000  
031120312111100001211010112112013233331000030001000000000000  
413334112010122221010102312221321201010111411100100000000000  
021001131111000000101110131112011233200000300001000000000000  
423333312101110001111112111211311121231100300001010000000000  
021221122110112221011101223221012212210011311110000000000000  
3221223211110000001100211101123211223200002000001000000000000  
02332310301121001110110222221011221001110200011000000000000  
3201202312000122110111112120122122133000014111001000000000000  
014324212211100000111012010221012121211002200001000000000000  
24021122211111101210010233321332222200112000100000000000000  
0032313313010112100111211011200211121100033111001000000000000  
341113112211100000111123221212212321201102000001000000000000  
00323124221012221101112221200211232000120001100000000000000  
221214201211001000001030011213312211210012110000000000000000  
003220214121100001211113322110021133101122000001000000000000  
3322223412201222210111211111223121011000221011100000000000000  
002213112221000000110123122200022331011012010000000000000000  
321230133231110001101311211122311123000132000001000000000000  
001124321210112221011221112211022201110112111100000000000000  
333311212321000000111123331112311233001110000000000000000000  
0001323432322100011003111121110221211003300000100000000000000  
3211121012130122210121311122223212110101111110000000000000000  
012221134231100000113212220021012122002220000000000000000000  
3212243212231100121012211121123112301111210000100000000000000  
01222021243301221001222332122102320200211011110000000000000000  
321113343123100000113211111101311120101220000000000000000000  
022322011231221012122220212330222300111110000000000000000000  
31112122232320121002112231120031120101220011110000000000000000  
022102333234300001233211102123021130111210000000000000000000  
301231201213222102223222121032221102210110010000000000000000  
0321222442333000001332111120230111001112000110000000000000000  
3012210113243000013213210212203222201211100000000000000000000  
0312222122224322212221222122130122011121011100000000000000000  
302113233233230000333211121120322120200200000000000000000000  
0311302112235500014212213112130113010210110000000000000000000  
3021141233333422223211201312032201121200011100000000000000000  
03121121112334000033232101302120113021211000000000000000000000  
3022222112325500064321233122213220011120110000000000000000000  
0321123433232552232212111120120211202111001100000000000000000  
301221211123338003322223212113112020210100000000000000000000  
0221140212224570763211120121220220212020010000000000000000000



important feature only in the case of the extraction of ferning images but it has no meaning in this example. We will explain this feature in the ferning-like image simulation.

The above information is coincident with the visual inspection in the accumulators. Also, the parameters and feature points obtained by the program were coincident with our visual inspection in the picture space.

The `remap()` subroutine is just a procedure for reverse transformation and reconstruction of the lines back into picture space. This `remap()` uses only the values of  $\rho$  and  $\theta$  of each detected peak, it does not consider the value of the peaks itself. So the reconstructed lines are obtained from the reverse transformation using only parameters  $\theta_s$  and  $\rho_s$ . The computation uses every  $x_i$  to obtain the corresponding  $y_i$ . The reconstructed lines are, therefore, across the whole picture region, rather than, as in the original picture, just the line segments inside one picture's region. Figure 18 illustrates the reverse transformation image from the parameter space with the threshold of 15.

Let us investigate how the threshold effects the detection of lines. In our program, when the threshold of 13 was applied, the following was output Dat file:

Total number of peaks bigger than 13 is 7,  
so 7 line(s) have been detected:

Q= 138, p= -3.5, peak= 15

Q= 135, p= -2.0, peak= 64

Q= 135, p= 0.5, peak= 62

Q= 132, p= 2.0, peak= 14

Q= 132, p= 2.5, peak= 15

Q= 135, p= 28.5, peak= 18

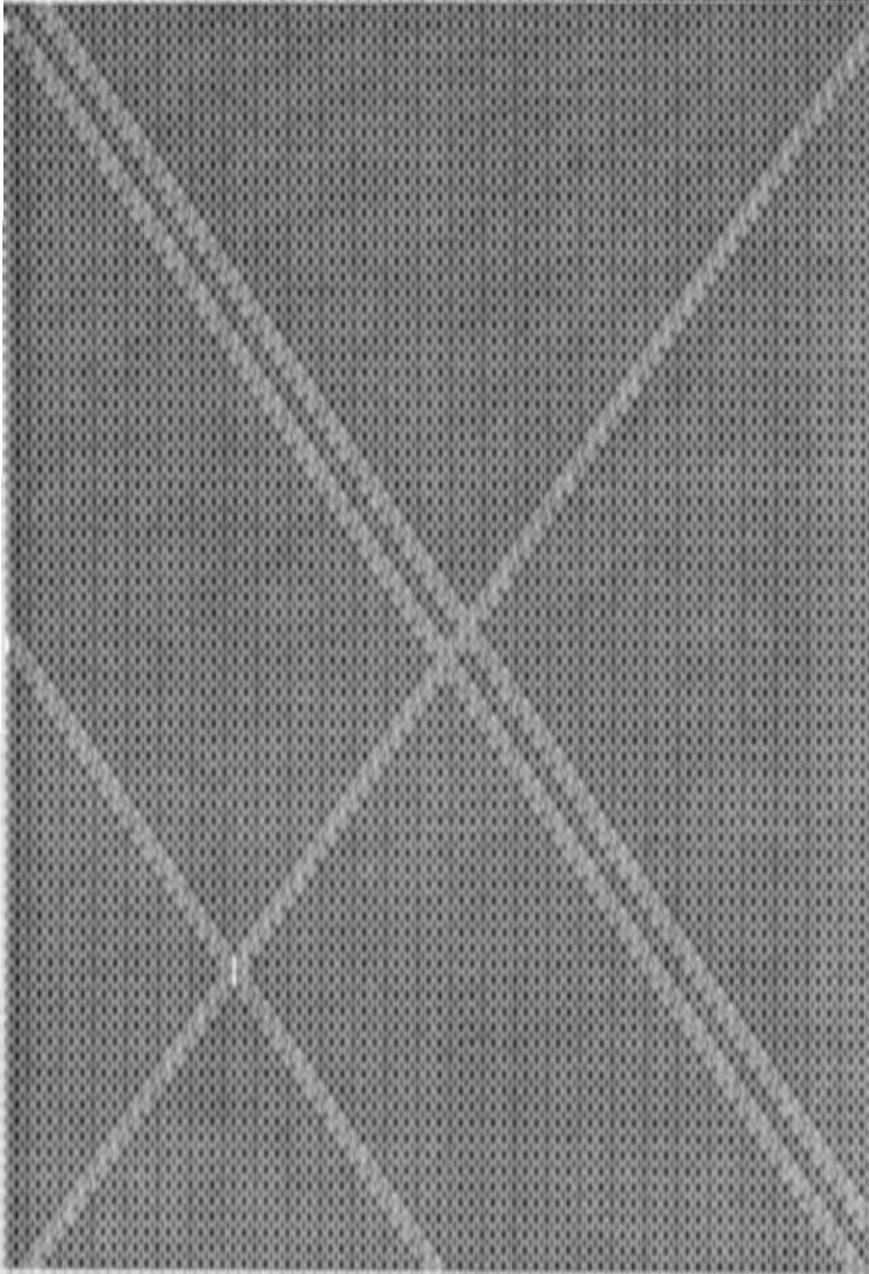


Figure 18. The reconstructed image with threshold of 15.

Q= 45, p= 57.5, peak= 62

Maximum peak: Q = 135 , p = -2.0 peak= 64.

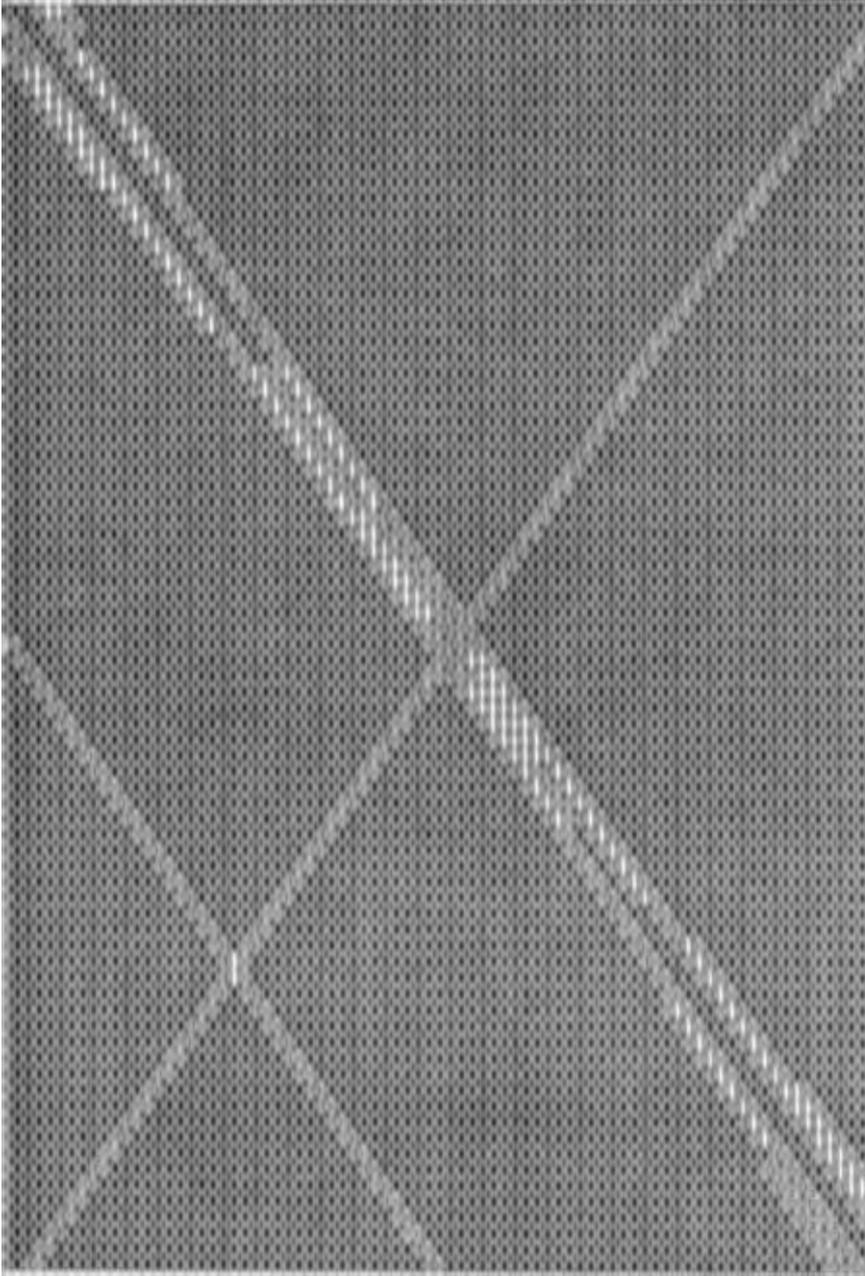


Figure 19. The reconstructed image with threshold of 13.

This is the longest line in the picture space.

When  $Q = 135$ , 3 lines are in parallel.

When  $Q = 132$ , 2 lines are in parallel.

The percentage of the parallel lines is: 71.4%.

This time 7 peaks were obtained. To reconstruct an image by using these  $\rho$  and  $\theta$  parameters, the `remap()` routine was called. The respective results are shown in Figure 19. The difference between Figure 19 and Figure 18 is obvious, but not tremendous. The difference occurs near the two lines ( $\theta=135^\circ$ ,  $\rho=-2.0$ ;  $\theta=135^\circ$ ,  $\rho=0.5$ ) which were long and close. This is because a quantization error usually happens when two long lines are close enough. This problem has been discussed in the literature [30]. If we decrease the value of the threshold to be much smaller, the difference between the reconstructed image and the original one will increase significantly. In this example, the points in the picture space were mapped to the parameter space and then they were mapped back to the picture space again. This two-way calculation obviously increased the errors. In the practical case of the mucus picture with  $256 \times 256$  or  $512 \times 512$  pixels, the ferning lines usually are not so close in respect to their length, there would be no such problem.

The selection of threshold should be considered carefully. In our future processing and detection of real ferning images, the threshold will be decided on the base of a lot of practical tests.

#### 4.2.2 The Simulative Application of Ferning-like Images

The Hough Transformation and ferning feature extraction were applied to three artificial ferning-like images (see Figure 20 - Figure 22). They are full-ferning, partial-ferning and linear-ferning, respectively (refer to the introduction of ferning characteristics in Chapter II).

##### Full-ferning image simulation

By visual inspection, there are 18 lines in the full-ferning image shown in Figure

20 and the longest line is the main ferning branch, with 9 side-branches in one side of the main branch and 8 side-branches in the other side. All these side-branches, in either side, are parallel with each other. As it was pointed out in Chapter III, the two most distinguished characteristics of a full-ferning image are:

- (1) full-image has much more lines,
- (2) most of the lines (side-branches) are parallel with each other.

In the image from Figure 20, the total number of feature points is 583, so  $583/6400=9.1\%$  pixels are feature points. The time taken in a SUN machine to carry out the Hough Transformation and the accumulator addition was about 9.2 seconds, for peak search and printing out, the time was about 0.7 seconds, and for storage of the data was about 3.5 seconds. The same program was run in PC-AT using Optimizing C5.1 compiler and the time taken was 47.15 seconds, 1 second and 14 seconds, respectively. Look at the time taken in Section 4.2.1, one can see that the time taken to run this program on PC-AT is 5 times of that of the SUN machine.

The result of this processing and feature extraction can be found in the Dat file:

Total number of peaks bigger than 15 is 19,  
so 19 line(s) have been detected:

Q= 135, p= -25.5, peak= 22

Q= 135, p= -11.5, peak= 32

Q= 135, p= -8.5, peak= 33

Q= 135, p= -3.0, peak= 38

Q= 135, p= 5.5, peak= 38

Q= 138, p= 12.0, peak= 18

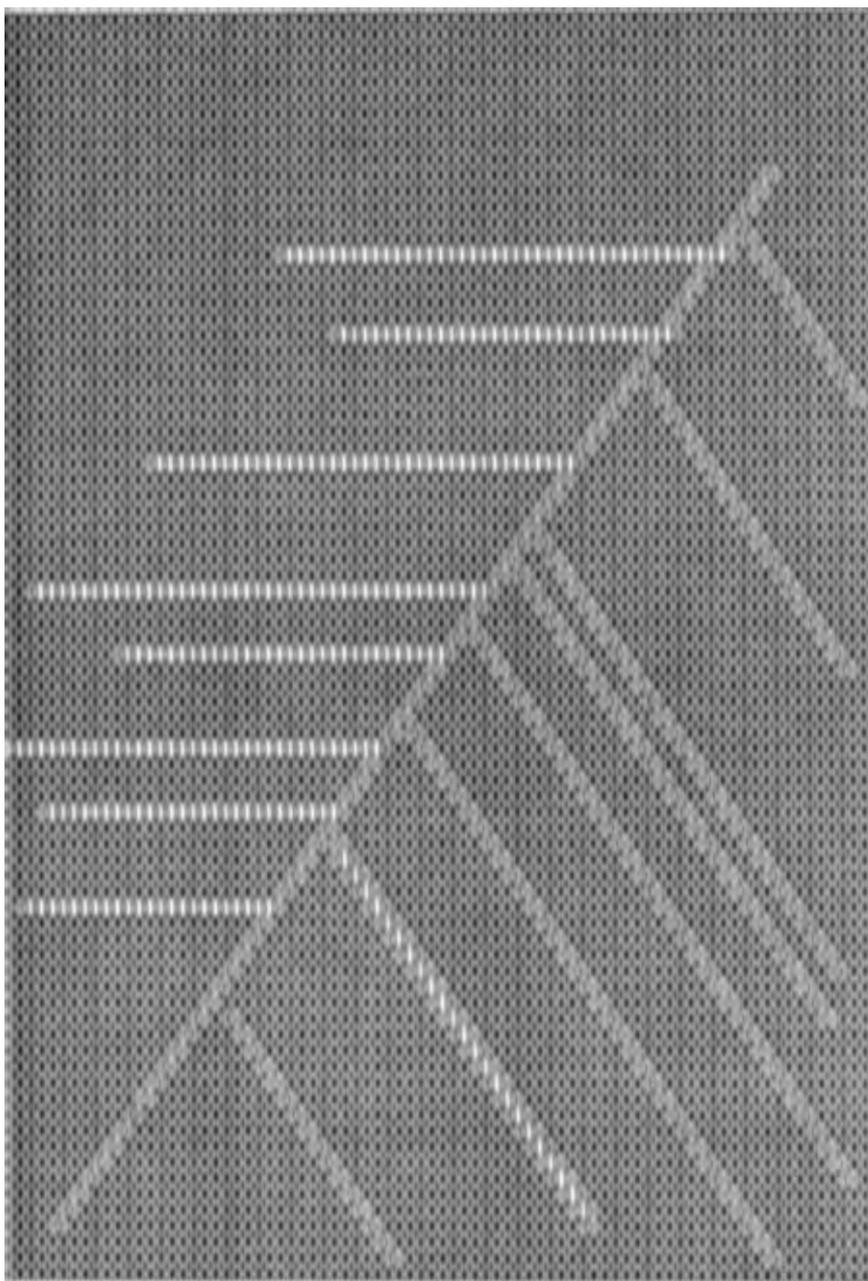


Figure 20. A full-ferning simulative image.

Q= 90, p= 15.0, peak= 43

Q= 135, p= 15.5, peak= 29

Q= 135, p= 16.5, peak= 28

Q= 90, p= 20.0, peak= 33

Q= 132, p= 20.0, peak= 18

Q= 90, p= 28.0, peak= 41

Q= 135, p= 30.5, peak= 19

Q= 90, p= 36.0, peak= 46

Q= 90, p= 40.0, peak= 35

Q= 90, p= 46.0, peak= 39

Q= 90, p= 50.0, peak= 32

Q= 90, p= 56.0, peak= 30

Q= 45, p= 56.5, peak= 67

Maximum peak: Q = 45 , p = 56.5, peak= 67,

This is the longest line in the picture space.

When Q = 135 , 8 lines are in parallel.

When Q = 90 , 8 lines are in parallel.

The percentage of the parallel lines is: 84.2%.

The program detected nearly all the lines just as a visual inspection would, and the **parallel()** subroutine gave out the important information about the parallel-line characteristics. Usually, when the percentage of the number of lines which are parallel with each other becomes close to a certain prespecified value, the full-ferring stage can be confirmed. This criteria will be finally set up after sufficient number of practical applications have been tested and necessary experiences have been obtained.

There are some inevitable errors in the above:

(1) one line was missing, which was one of the side-branches with 11 feature points. It was the shortest line in this image and it was below the threshold.

(2) Two false lines were detected, which were ( $\theta=138^\circ, \rho=12.0$ ) and ( $\theta=132^\circ, \rho=20.0$ ). Since the quantization interval  $\Delta\theta=3^\circ$ , when some line just happened with  $\theta=135^\circ$ , those false lines had the parameter of  $135^\circ \pm 3^\circ$ . So obviously, this was a quantization error. In the practical cases, this kind of errors may happen. But compared with the total number of correctly detected lines, it should not effect the final classification decision.

### Partial-ferning image simulation

The definition of partial-ferning characteristic is: good ferning with side branches in part of the image, amorphous mucus in other parts. It is obvious that the most distinguished difference between the full-ferning characteristic and partial-ferning characteristic is that the total number of lines in the full-ferning image is much larger than that of the partial-ferning image (a slide can be divided into several view-fields, i.e. several images, the total number of lines is the sum of the numbers of lines from all images from a slide).

A partial-ferning image is similar to that shown in Figure 21.

The program was applied to a similar artificial partial-ferning image. Since the number of feature points in this image was less than that of the full-ferning (Figure 20), the time taken for the processing of the partial-ferning image was smaller than that for the full-ferning. The result of this processing was in the Dat file as follows:

Total number of peaks bigger than 15 is 6,

so 6 line(s) have been detected:

Q= 135, p= -1.5, peak= 42

Q= 90, p= 12.0, peak= 24

Q= 90, p= 21.0, peak= 19

Q= 45, p= 31.0, peak= 18

Q= 90, p= 33.0, peak= 29

Q= 45, p= 48.0, peak= 17

Maximum peak: Q = 135 , p = -1.5, peak= 42.

This is the longest line in the picture space.

When Q = 90 , 3 lines are in parallel

When Q = 45 , 2 lines are in parallel

The percentage of the parallel lines is: 83.3%.

One line was missing because the number of feature points of this line was smaller than the threshold. From the data shown above, three features of this image can be confirmed:

- (1) The percentage of the parallel lines is like in the full-ferning case.
- (2) The number of the detected lines is smaller than that of the full-ferning case.
- (3) The average lengths of all these lines are smaller than those of the full-ferning case.

These three features will be used in the next chapter to recognize the ferning patterns.

### Linear-ferning image simulation

The linear-ferning image is similar to that shown in Figure 22.

The most significant characteristic for an linear-ferning image is that there are fine linear fernings seen in few spots, but there is almost no side branching. From this

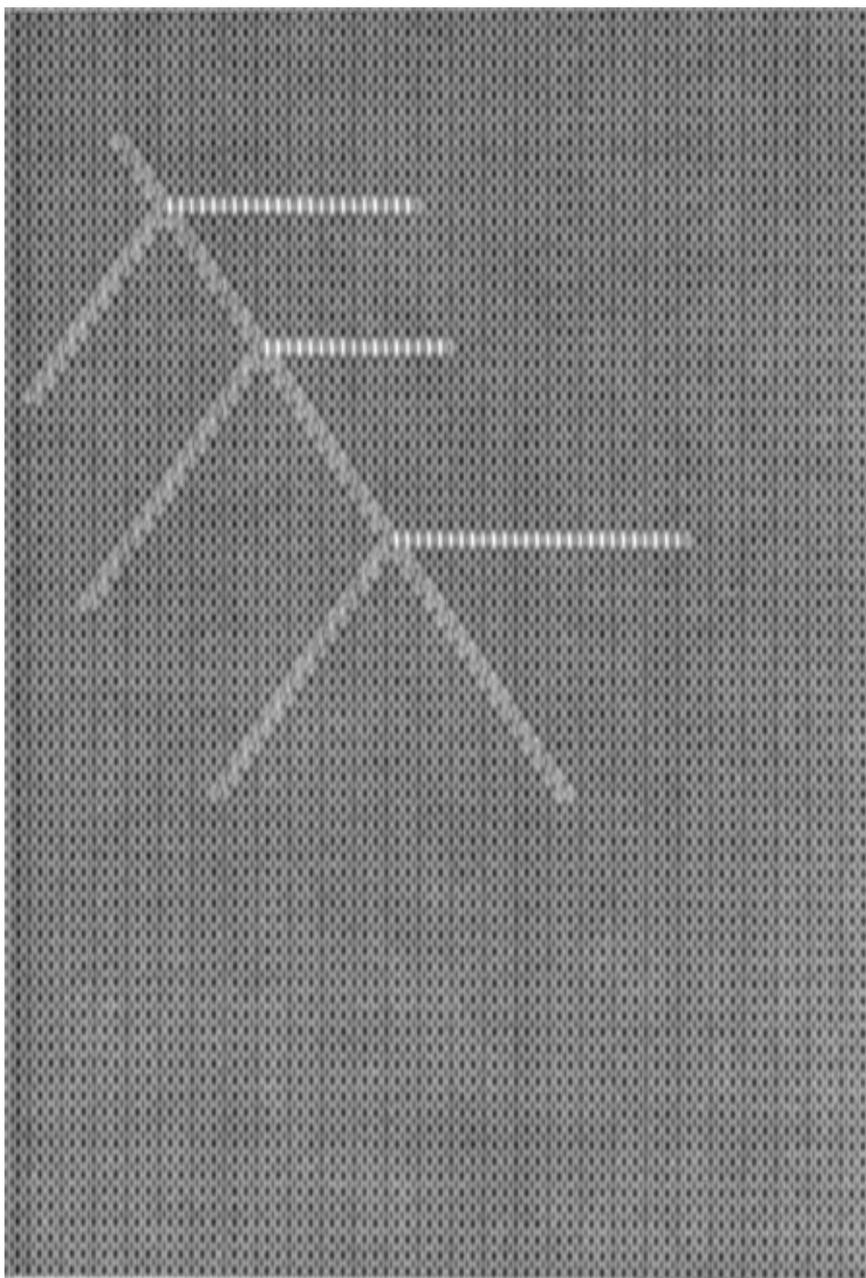


Figure 21. A partial-ferning simulative image.

point of view, one can easily understand that in a linear-ferning image there are some lines being detected, but among these detected lines, few are parallel with each other. When the image from Figure 22 was applied to this program, the processing procedure

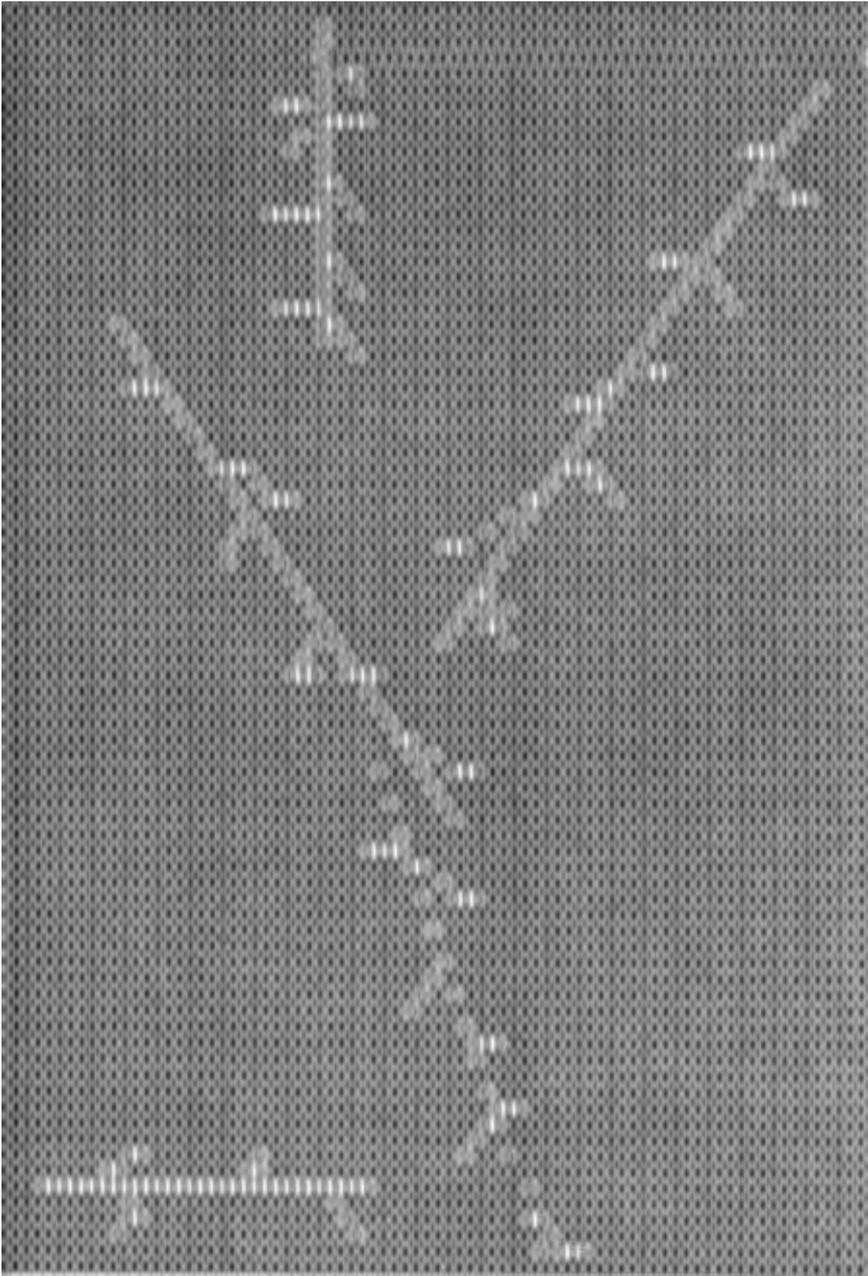


Figure 22. A simulative linear-ferning image.

was the same as for the last two images. The results of the parallel characteristic from the **parallel()** subroutine are, however, very different. The processing result in the Dat file is as follows:

Total number of peaks bigger than 15 is 5,  
so 5 line(s) have been detected.

Q= 153, p= -7.5, peak= 18

Q= 135, p= 8.0, peak= 32

Q= 0, p= 28.0, peak= 21

Q= 45, p= 56.0, peak= 38

Q= 90, p= 74.0, peak= 32

Maximum peak: Q = 45 , p = 56.0, peak = 38

This is the longest line in the picture space.

No lines bigger than the threshold are in parallel!

Comparing the detected lines with the image from Figure 22, it is obvious that all straight lines have been detected and none of them are parallel. It can come to the conclusions that:

- (1) the total number of all detected lines from all images in a slide is less than that of a full-ferning slide but approximately at the same level of that of a partial-ferning image.
- (2) almost no lines are parallel (the percentage of the number of parallel lines is very small).

### 4.3 CONCLUSION

In this chapter we have provided the experimental simulation results obtained using the routines to several images. It paves the way to approach the automation of the ferning test. It seems that some feature extractions worked successfully on the artificial binary images. But the actual application is left to be worked out. In this experimental

program, some ferning features can be obtained by analyzing the results of the detection (since the program detects only some of the features, it would be worthwhile to propose a few others). We can conclude in the following:

- The number of the straight lines. As shown above, different length of lines can be detected by using different thresholds. Those different number of different length of lines can be an important feature for different ferning stages.
- The percentage of the parallel lines among all detected lines. This is a significant feature for full-ferning stage.
- The average length of all maximum lines from all images (view-fields) in a slide. This feature represents the average length of main branches in a ferning slide.
- The rate of total number of maximum lines to total number of side-branch lines. This is only a proposed feature that will be tested in our future application.

In the next chapter, the feature space, as well as some important concepts of pattern recognition will be introduced. The main ideas in our ferning pattern recognition are the partition of the feature space and the distribution of the ferning feature in the feature space. The classification will be achieved by using the pattern recognition theory in the n-dimensional feature space.

## CHAPTER V

### PATTERN RECOGNITION PROBLEM WITH SPECIAL APPLICATION TO CERVICAL MUCUS PATTERNS

In this Chapter, two methods of classification will be proposed:

(G) An automated **decision logic** will be presented in Section 5.1, which can be used to classify the extracted ferning features of the cervical mucus into relevant categories.

(2) A brief description of approaches to morphological pattern recognition and classification will be presented, and with the extracted ferning features obtained in the previous chapter, a method for cervical mucus classification using **statistical distribution on n-dimensional space** will be described in Section 6.2.

It must be emphasized that the second method is much more important and universal than the first one. The reason that the logic method is proposed here is that the number of the ferning features is small, so they may be easily classified by this simple method. Since our final goal is to construct an ovulometry system for prediction of the ovulation moment, a large amount of biochemical and rheologic parameters of cervical mucus, together with the ferning features, will be analyzed and classified. The situation can be, therefore, very sophisticated and complex. It is impossible to solve this problem using the logic method, so the classification using the statistical pattern recognition theories will become the most accurate method. This method, described in the Section 5.2, will be used not only in the ferning pattern classification but in the decision-making of the whole ovulometry system.

In Chapter II, we have introduced four kinds of ferning stages which represent the features of the cervical mucus appearing in four periods of time. These four kinds of ferning stages will be referred as four "classes" or "categories". The terms "class" and "category" are terminologies used in the pattern recognition classifications, as in this chapter.

TABLE II  
FEATURE DESCRIPTION

Feature		Description
F1	Total lines	Number of all lines above threshold T1
F2	Long lines	Number of all lines above threshold T2
F3	Short lines	Number of all short lines = $F1 - F2$
F4	Parallel lines	Number of all parallel lines
F5	Percentage of long lines	Percentage of long lines = $F2 / F1$
F6	Percentage of parallel lines	Percentage of parallel lines = $F4 / F1$

In the Chapter IV, The algorithm for the extraction of ferning features has been presented. Table II provides a brief description of each of these features used in this chapter. T1 and T2 in the Table II are two thresholds which were used in searching of the accumulators of the Hough Transformation to detect the collinear pixels (refer to Chapter IV).

TABLE III  
DESCRIPTION OF CRITERIA FOR CLASSIFICATION

Criteria	Description
t1	Criterion for number of total lines, used with F1
t2	Criterion for number of total lines (t2>t1), used with F1
t3	Criterion for number of long lines, used with F2
t4	Criterion for number of long lines (t3>t4), used with F2
t5	Criterion for percentage of long lines, used with F5
t6	Criterion for percentage of parallel lines, used with F6

### 5.1 APPROACH TO AUTOMATED LOGIC CLASSIFICATION

Figure 23 indicates the automated classification logic used. Six criteria, t1 - t6, are used in the decision flow-logic. These criteria function as six decision points to classify which class one ferning image belongs to. Table III describes the six criteria. Here it is necessary to point out that two pairs, t1 and t2, t3 and t4 serve as the criteria of the number of total lines and the number of long lines, respectively, but the values in each pair are not the same. As pointed out in Chapter IV, the total number of lines in a full-ferning image was larger than that of either partial-ferning or linear-ferning, and so was in the case with the number of long lines. This pair of criteria is used to distinguish the full-ferning category from the others. Here, the conditions:

$$t1 < t2 \quad \text{and} \quad t4 < t3$$

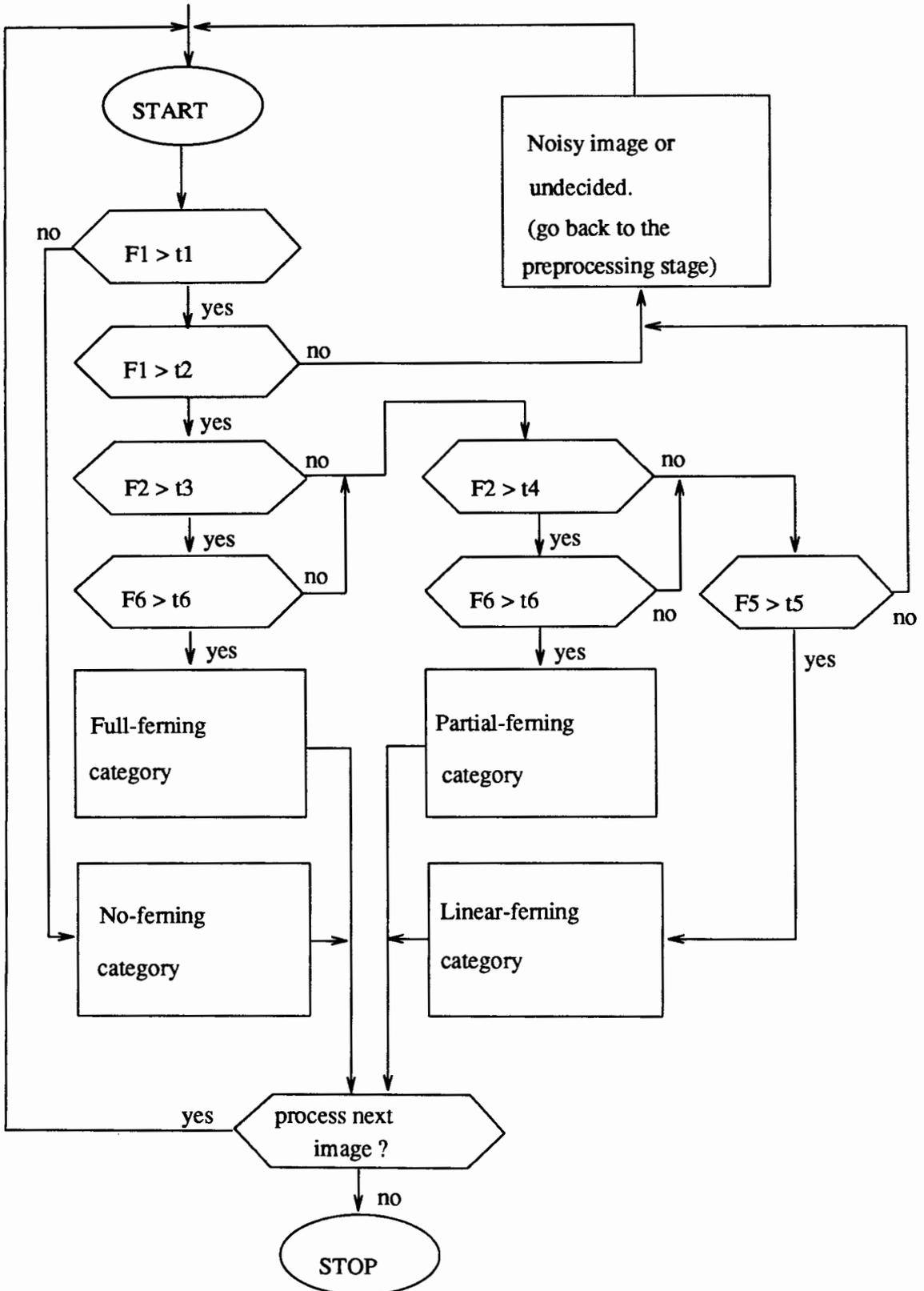


Figure 23. Flow diagram of logic classification.

must be satisfied.

Notice in the flow diagram shown in Figure 23 that if a pattern is an undecided or noisy image, the preprocessing has to go back to the preprocessing procedure to have more averaging procedures to remove the noise.

It must be pointed out that the logic classification is a simple and easy-implementable, but inaccurate method, because it lacks of a theoretical basis. The statistical pattern recognition classification is more preferable, which will be proposed in the next section.

## 5.2. APPROACH TO STATISTIC PATTERN RECOGNITION OF MORPHOLOGICAL CLASSIFICATION

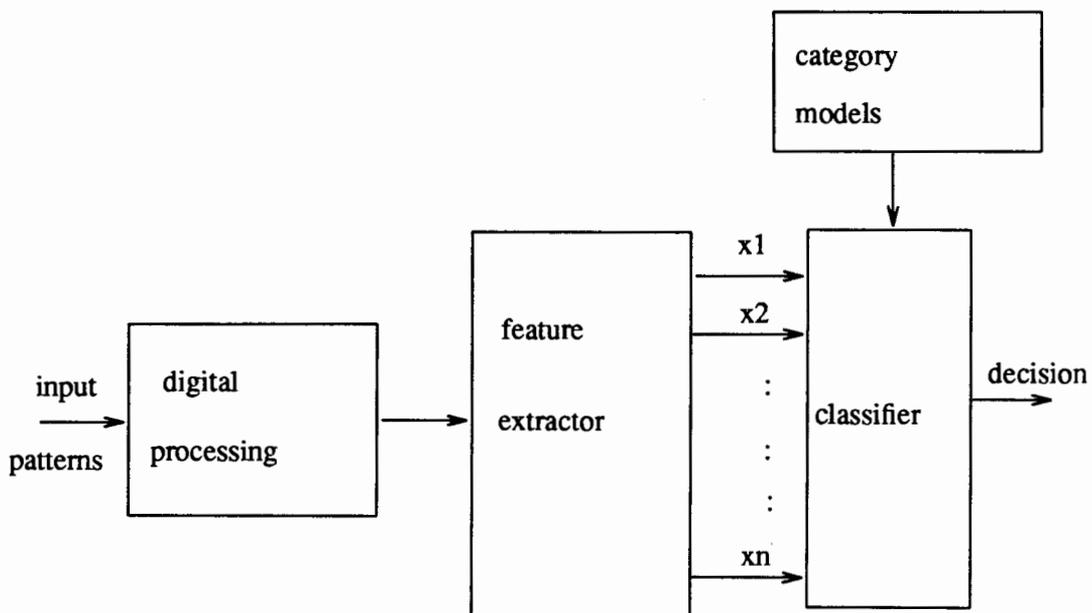


Figure 24. A morphological pattern recognition system.

The many different mathematical techniques used to solve pattern recognition problems may be grouped into two general approaches, namely, the **statistical approach**

and the **syntactic approach** [39]. In the statistical approach, a set of characteristic measurements, called **features** , are extracted from the patterns; the recognition of each pattern assignment to a pattern class is usually made by partitioning of the **feature space** [40]. More than half of the developments in the pattern recognition research during the past decade deal with the **statistical approach. Morphological analysis and classification** is one of the researches in **statistical approach** [39].

Morphological approach for cervical mucus pattern recognition can be considered as consisting of two subproblems. The first subproblem is what measurements should be taken from the input patterns, we refer to it as **feature extraction**. This subproblem has been presented in Chapter IV, the measured features of a mucus image are of the line-like features, such as the length of the lines, the number of the lines and the percentage of the parallel lines etc., which are intuitively related to the visual image and can be considered as some parameters of the input patterns.

The second subproblem in morphological pattern recognition is the problem of classification ( or making a decision on the class assignment for the input patterns ) based on the measurements taken from the selected features. Here, the machine or routine which performs the function of classification is called a **classifier**. A block diagram of this pattern recognition approach is shown in Figure 24.

### 5.2.1 The Classification in Feature Space with Distribution of Ferning Features

As it was mentioned above, statistical pattern recognition methods will be mainly applied in our approach. Therefore, they are discussed in this section. More details can be found in the referenced literature [39],[41].

Given is a set  $\underline{X}$  of **pattern vectors**  $X$  ( they will be also called **patterns** ) of real coordinates.

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad X \in \underline{X}, \quad x_i \in \Phi \quad i=1,2,\dots,n$$

Each pattern represents a point in space  $\Phi$ . The space will be called **feature space** or **pattern space**. Additionally, certain regions  $\omega_j$ , called classes will be distinguished in the feature space. Recognition ( classification ) of the given pattern  $X$  consists in determination to which class it belongs ( in other words, to which class belongs the point described by the coordinates of vector  $X$  ). This is achieved by computing and estimating the conditional probability of  $p(j|X)$ . Here  $p(j|X)$  is defined as: the probability of the pattern class  $j$  under the condition of vector  $X$  has occurred. This conditional probability  $p(j|X)$  can be computed using **Bayes' Theorem** [42]. This theorem has been developed into **Gaussian classifier** with multivariate and normal distribution classifier which will be discussed in detail in the next section. By comparison of  $p(j|X)$ 's value in each class, one can confirm with no difficulty to which space class belongs the given pattern vector.

Consider that  $x_1, x_2, \dots, x_n$  are random variables where  $x_i$  is the measurement of  $i$ -th feature. Let us assume that for each pattern class  $\omega_j, j=1,2,\dots,m$ , the multivariate (n-dimensional) probability density (or distribution) function of the vector  $X, p(X|\omega_j)$  and the probability of occurrence of  $\omega_j$  are known. Here,  $p(X|\omega_j)$  is the probability that, given the set of measurement  $X$ , the pattern belongs to class  $\omega_j$ . One can think of it as simply the probability of class membership. On the basis of a priori information  $p(X|\omega_j)$  and  $p(\omega_j)$ , the function of a classifier is to perform the classification task for minimizing the probability of misrecognition.

Consider that  $F1 - F6$  are six-dimensional space vectors and any subset of them also can be an  $n$ -dimensional space vector, where  $n$  is the number of features taken from any of  $F1 - F6$ . As may be expected from Chapter IV, those ferring features are relevant

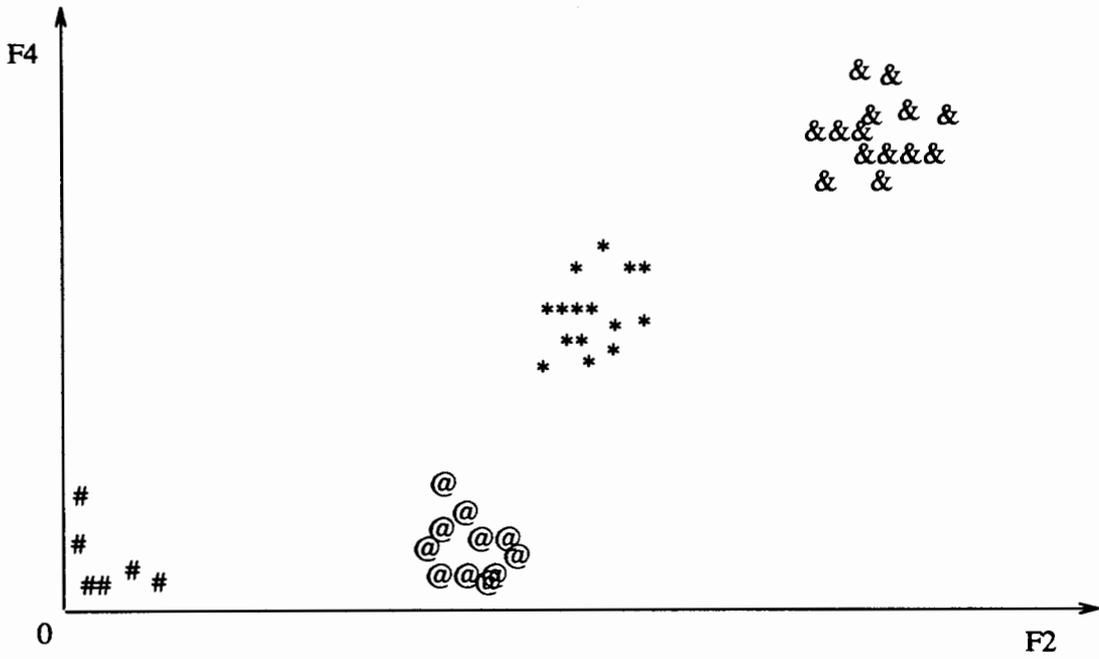


Figure 25. The distribution of four classes of mucus on F2-F4 two-dimensional space.

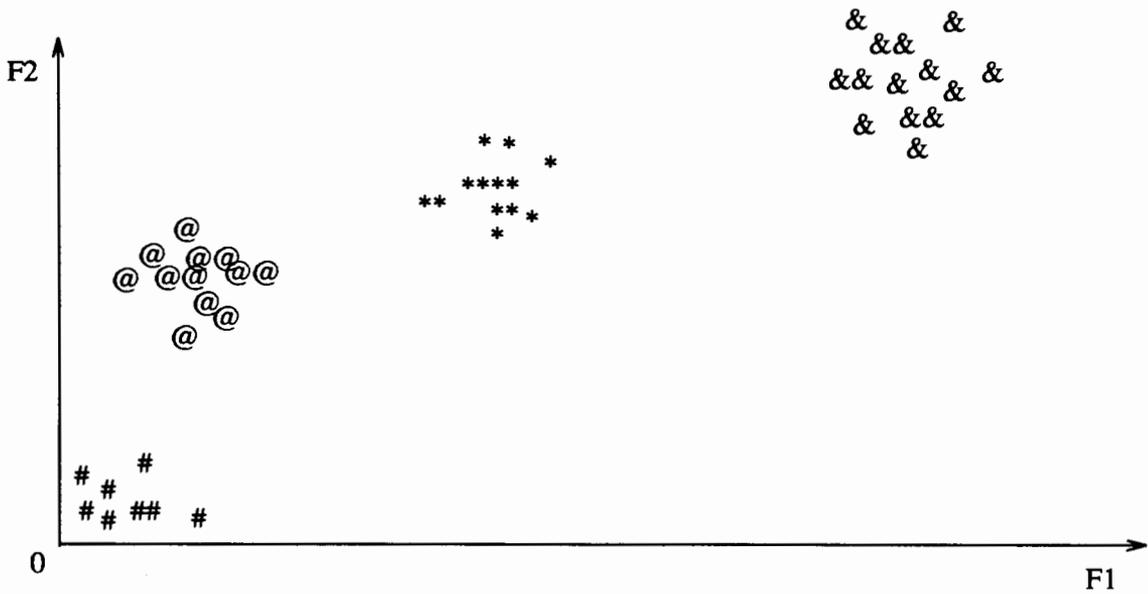


Figure 26. The distribution of four classes of mucus on F1-F2 two-dimensional space.

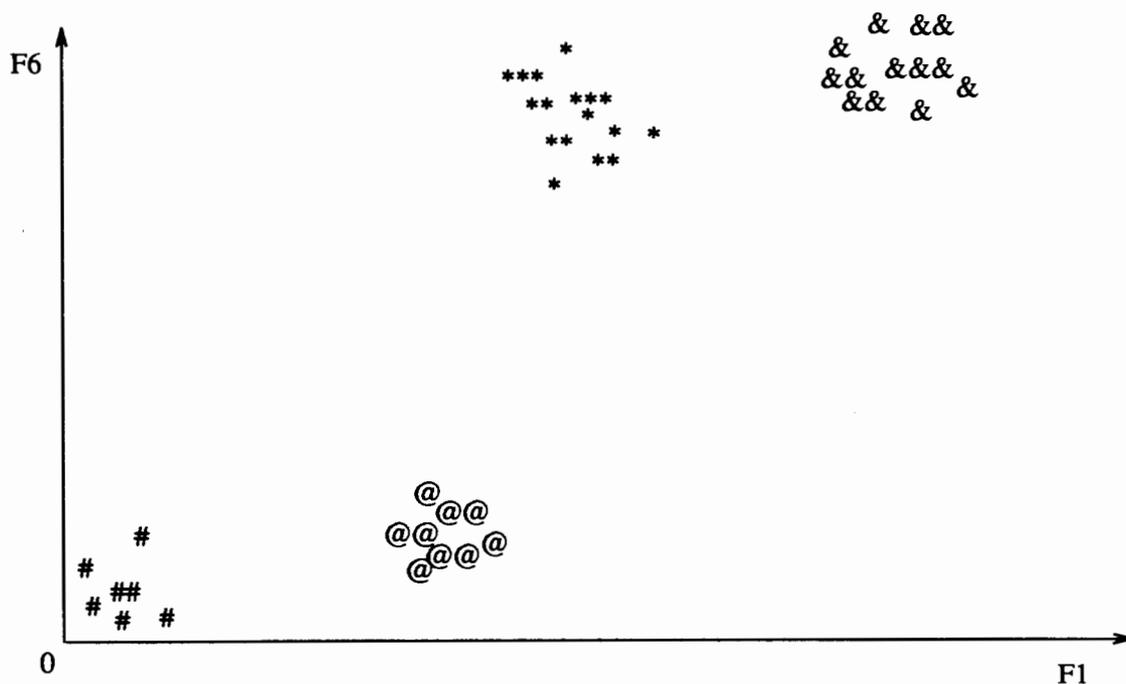


Figure 27. The distribution of four classes of mucus on F1-F6 two-dimensional space.

features for the classification of the cervical mucus categories. Their combined effect (or the distribution) can be, therefore, also an excellent classifier. Let us assume that one has a training set of samples from which a set of features has been extracted ( it is also called the **training feature vector** ). Hypothetically, the training set of feature vectors distributes in the feature spaces as illustrated in Figure 25 - Figure 27. In these figures, the training sets distribution of four mucus classes are non-ferning, linear-ferning, partial-ferning and full-ferning denoted by the signs #, @, \*, &, respectively. The distributions of these ferning classes shown in Figure 25 - Figure 27 are just assumptions illustrating how the features can partition the feature space. For the convenience of illustration, only a two-dimensional space is created and only three independent pairs of features are selected. In the practical and sophisticated situation, more dimensional space will be used. This will depend on how accurate the classification results are expected. The practical distribution will be obtained after analyzing a large amount of data in the future.

### 5.2.2 Computer Classification with Statistical Pattern Recognition Theory

Before the application of computer classification, a large amount of data must be obtained from all necessary experiments. These data are what we refer to as **training sets**. The probability distribution of all vectors in the feature spaces must be calculated from the training sets. Only by using this data base, can the automatic computer classification be accomplished.

A multivariate statistic approach to the classification of patterns is proposed in this thesis. The pattern vectors are assumed to be normally distributed, and the probability of class membership for an unknown pattern vector  $X$  is estimated by a multivariate **Gaussian classifier** of the form:

$$p(j|X) = \frac{p(j) |Q_j|^{-1/2} \exp\{-\frac{1}{2}[(X-M_j)'Q_j^{-1}(X-M_j)]\}}{\sum_{i=1}^n p(i) |Q_i|^{-1/2} \exp\{-\frac{1}{2}[(X-M_i)'Q_i^{-1}(X-M_i)]\}} \quad (5.1)$$

for each class  $j = 1, \dots, n$ . Where  $X$ ,  $M$ ,  $Q$  are matrices and the sizes of them are dependent on the dimension of the feature space. In the real life, neither the number of dimensions of the feature space nor the number of pattern classes are too large. For each class the equation (5.1) needs to be computed only once. The time taken for calculation of equation (5.1) will not be a real problem. This is one of the advantages of the **Gaussian classifier**. The meaning of  $p(j)$ ,  $Q_j$ ,  $M_j$  and  $X$  will be explained later in this section.

The details of theoretical basis of the above formula can be found in some statistics books related to the Gaussian distribution and classification [42], [43]. This classification model is considered to be a reasonable approximation because the distributions of this kind of variables appear to be approximately Gaussian [43]. The computation implies knowledge of the **mean vector**  $M_j$ , the **covariance matrix**  $Q_j$ , and the **a priori probability**  $p(j)$  for each class.  $M_j$  and  $Q_j$  will be computed from the training set for the class analysis [42], [39]. The a priori probabilities will be taken as the proportion

of each class in the training set. Of course, the probability  $p(j)$ ,  $M_j$  and  $Q_j$  can only be obtained after thousands of patients are surveyed and necessary amount of training sets are analyzed. In a practical situation these could be chosen according to the experience of a particular laboratory, and could be further adjusted if the costs of the various types of errors are unequal [43].

Now let us investigate how the **covariance matrix**  $Q_j$  and the **mean vector** of  $M_j$  in equation (5.1) can be computed from the training sets. For the convenience of the explanation, the subscript  $j$  of both  $M_j$  and  $Q_j$  are omitted in the following. In general cases, the **mean vector**  $M$  can be expressed as:

$$M = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix} = \begin{bmatrix} E(x_1) \\ E(x_2) \\ \vdots \\ E(x_n) \end{bmatrix}$$

where each  $\mu_i$  value (or  $E(x_i)$ ) approximately equals the average value of each feature type from the training set. For instance, if there are  $r$  patterns (cervical mucus images) in the training set, one can obtain  $r$  parameter values in each type of feature. The  $E(x_i)$  can be approximately estimated as:

$$E(x_i) \approx \hat{E}(x_i) = \frac{1}{r} \sum_{j=1}^r x_{ij} = \bar{x}_i$$

where  $\hat{E}(x_i)$  is the estimated value of  $E(x_i)$ . The **covariance matrix** is

$$Q = \begin{bmatrix} q_{11} & q_{12} & \dots & q_{1n} \\ q_{21} & q_{22} & \dots & q_{2n} \\ \dots & \dots & \dots & \dots \\ q_{n1} & q_{n2} & \dots & q_{nn} \end{bmatrix}$$

where each  $q_{ij}$  in the matrix can be obtained from the equation:

$$q_{ij} = Cov(x_i, x_j) = E\{[x_i - E(x_i)][x_j - E(x_j)]\} \quad i, j = 1, 2, \dots, n \quad (5.2)$$

It is supposed here that all these  $q_{ij}$  exist.

Since each training set is independent from each other, and if each extracted feature is also independent (it will be in our case), the covariance between any other

different features is zero, which means that when  $i \neq j$ , equation (5.2) can be

$$q_{ij} = Cov(x_i, x_j) = E\{[x_i - E(x_i)][x_j - E(x_j)]\} = 0 \quad \text{when } i \neq j$$

In this case all off diagonal elements in the covariance matrix will be zero and each diagonal element will be the corresponding **variance**. So the **covariance matrix** can be rewritten as

$$Q = \begin{bmatrix} q_{11} & 0 & \dots & 0 \\ 0 & q_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & q_{nn} \end{bmatrix}$$

where each  $q_{ii}$  of the diagonal elements can be computed by

$$q_{ii} = Var(x_i) = E\{[x_i - E(x_i)]^2\} = \sigma_i^2 \quad i = 1, 2, \dots, n$$

In the practical cases,  $\hat{\sigma}_i$  is used as an estimated value of  $\sigma_i$  which comes from the equation

$$\sigma_i^2 \approx \hat{\sigma}_i^2 = \frac{1}{(r-1)} \sum_{j=1}^r [x_{ij} - \hat{E}(x_i)]^2$$

As having been mentioned above, the pattern vectors are assumed to be normally distributed. An example of computation of equation (5.1) in a two-dimensional normal distribution is the following.

The probability function of a two-dimensional normal distribution can be written as follows:

$$p(x_1, x_2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left\{ \frac{-1}{2(1-\rho^2)} \left[ \frac{(x_1-\mu_1)^2}{\sigma_1^2} - 2\rho \frac{(x_1-\mu_1)(x_2-\mu_2)}{\sigma_1\sigma_2} + \frac{(x_2-\mu_2)^2}{\sigma_2^2} \right] \right\}$$

where  $\sigma_1, \sigma_2; \mu_1, \mu_2$  and  $\rho$  are **variance** (standard deviation), **mean value** and **correlation coefficient**, of the variables  $x_1$  and  $x_2$ , respectively. It should be noted that  $\rho$  is a conventional notation for **correlation coefficient** and it has not thing to do with the same notation used in the Hough Transformation (where  $\rho$  is sued as the normal distance from the origin to a line). In a normal distribution, the value of variance can be also estimated from the training set:

$$\sigma_i^2 \approx \frac{1}{(r-1)} \sum_{j=1}^r (x_{ij} - \bar{x}_i)^2$$

And the matrix forms of  $X$  and  $M$  are

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad M = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$$

According to equation (5.2), the **covariance matrix**  $Q$  can be computed as:

$$Q = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}$$

Therefore, its determinant is

$$|Q| = \sigma_1^2 \sigma_2^2 (1 - \rho^2)$$

and

$$Q^{-1} = \frac{1}{|Q|} \begin{bmatrix} \sigma_2^2 & -\rho\sigma_1\sigma_2 \\ -\rho\sigma_1\sigma_2 & \sigma_1^2 \end{bmatrix}$$

After computation, the product of the matrices  $(X-M)'Q^{-1}(X-M)$  is

$$(X-M)'Q^{-1}(X-M) = \frac{1}{2(1-\rho^2)} \left[ \frac{(x_1-\mu_1)^2}{\sigma_1^2} - 2\rho \frac{(x_1-\mu_1)(x_2-\mu_2)}{\sigma_1\sigma_2} + \frac{(x_2-\mu_2)^2}{\sigma_2^2} \right] \quad (5.3)$$

So for a two-dimensional classification, the equation (5.1) will be easy to compute by using the formula from (5.3).

As having been pointed out that in most of the cases the variables are independent each other. For the practical cases, just substitute the  $\rho$  in the above equations with zero.

The feature space shown in Figure 25 - Figure 27 and the example discussed above are only two-dimensional. A three-dimensional program of Gaussian Classifier will be presented in the next section which performs both the training set data processing and classification decision-making. For a more precise and sophisticated classification, more dimensional space may be required in our future work.

### **5.3 A PROGRAM OF GAUSSIAN CLASSIFIER APPLICATION**

A program of Gaussian Classifier in F1, F2, F4 three-dimensional space is presented in the section. The program is written in C language. The program will be explained by using a training set of patterns with three parameters for each ferning class as the basic input data to calculate the M and Q matrices. Then any features from a mucus image are used as the pattern of an unknown class to compute the probability of each class to which this pattern belongs. The program will print out the probability values of the class to which this pattern belongs.

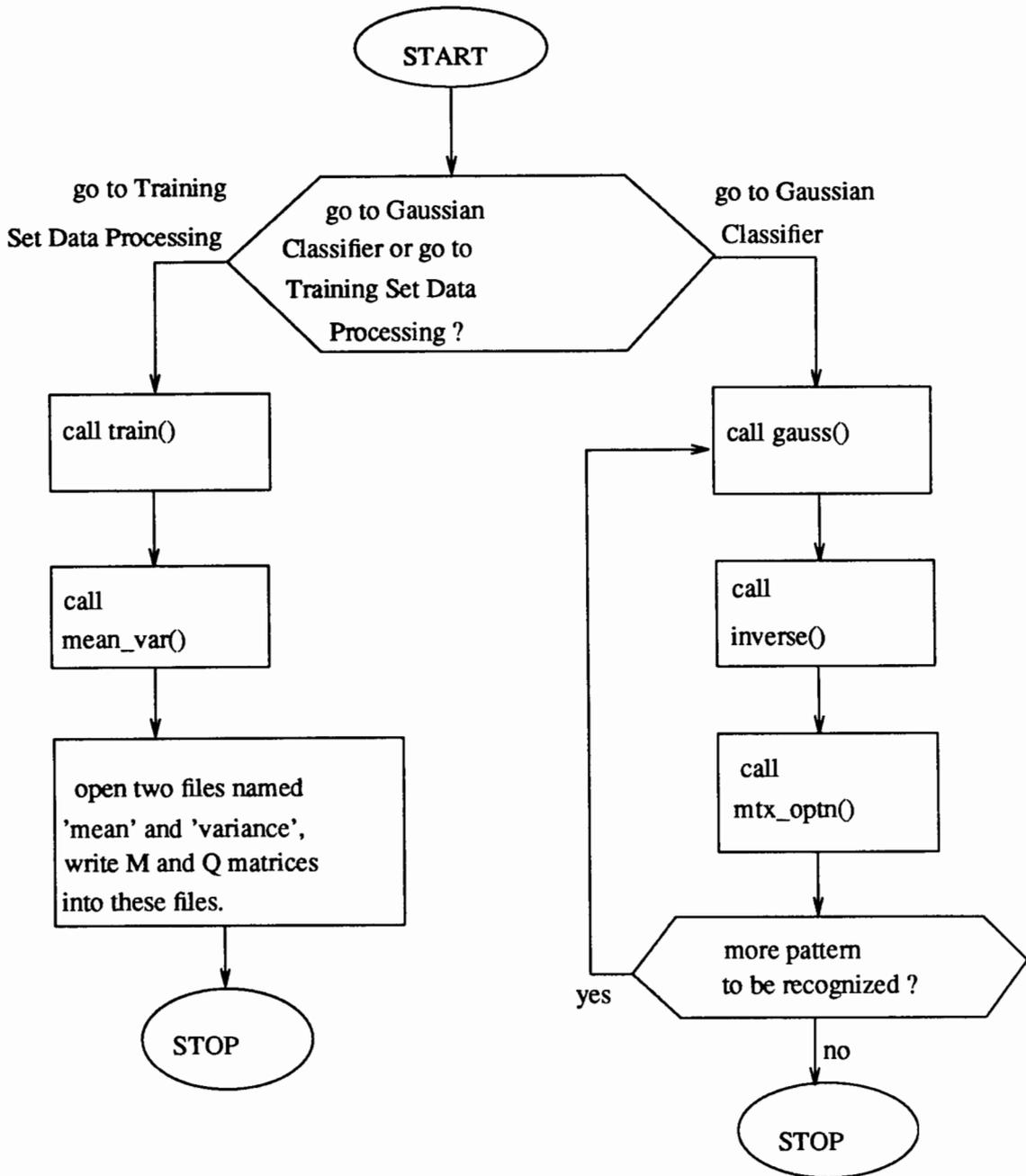
The reason that three features, F1, F2 and F3, are chosen is that they are mutually independent. Actually, one can choose any kind of features from F1 - F6. By choosing, however, the independent features, the calculation can be simplified. Because only the diagonal elements in the Q matrix need to be computed and the off-diagonal elements can be set to zero when the chosen features are independent of each other. This has been explained in the last section.

#### **5.3.1 The Main Routines of the Program**

The program contains two main parts: the training set data processing and the Gaussian Classifier procedure. They are explained in the following:

##### **Training set data processing**

As mentioned in the last two sections, the principle of the classifier is based on the large amount of training data which teaches the classifier to classify each input pattern to the correct class. In the training set data processing, the program requires the user to input the total number of full-ferning training sets, the total number of partial-ferning training sets, the total number of linear-ferning training sets and the total number of non-ferning sets, respectively. Then the program requires the user to input three kinds of



**Figure 28.** The flow diagram of the program of Gaussian Classifier.

parameters for each class: the total number of detected lines, the total number of detected long lines and the total number of detected parallel lines. These information now becomes the "experience information" stored in the matrices M and Q calculated by the

program. Both M and Q are 3×4 matrices. After the subroutine **mean\_var()** calculates the M and Q matrices, it creates two files under the same directory of the program named **mean** and **variance** into which the M and Q matrices are stored. The advantage of this procedure is to simplify the processing. Once the entire training set data have been input to this program, the results of the calculation are written into these two files and they remain unchanged, unless the user obtains more training sets and wants to recalculate the M and Q matrices again. The M and Q matrices are the only important "experience information" which then used by the Gaussian Classifier to perform the recognition.

### Gaussian Classifier procedure

The **gauss()** subroutine performs the main task of pattern recognition. It prompts the user to input the parameters of a given pattern. These parameters are the total number of detected lines, the total number of long lines and the total number of parallel lines, respectively. These three parameters are assigned to the matrix X[]. Then the routine opens the two files **mean** and **variance** which contain the "experience information", M and Q matrices, to read the data from them. The computation of the equation (5.1) is done by the subroutines **inverse()** and **mtx\_optn()**. The **inverse()** subroutine computes the determinant of the matrix Q and the **mtx\_optn()** subroutine computes the formula

$$p(i) |Q_i|^{-1/2} \exp\{-\frac{1}{2}[(X-M_i)^t Q_i^{-1} (X-M_i)]\}$$

and calculates the sum of them for  $i = 1, 2, \dots, n$ . After this sum is computed, the routine can easily calculate the results of equation (5.1) by just substituting each element with different j's in the equation (5.1).

After each  $p(j|X)$  is computed, the **gauss()** subroutine prints out the value of each  $p(j|X)$ ,  $j = 1, \dots, 4$  which indicates the probability of this input pattern belonging to the class of full-ferning, partial-ferning, linear-ferning and non-ferning, respectively. By inspection of the values of each probability, one can easily tell which class this pattern

belongs to.

### 5.3.2 An Example of the Application in the Program of Gaussian Classifier

In this section, an example of the application of the program will be presented. An experimental training set was applied to the routine of **Training Set Data Processing** to obtain the "experience information", in the form of matrices M and Q. Two patterns of unknown class are then input to the **Gaussian Classifier** routine to test the program.

After the program execution, the **Training Set Data processing** routine was chosen and the following data were input:

From full-ferning training set:

The total number of full-ferning training sets = 3.

The total number of lines (F1) = 138, 103, 82.

The total number of long lines (F2) = 10, 7, 6.

The total number of parallel lines (F4) = 121, 74, 69.

From partial-ferning training set:

The total number of partial-ferning training sets = 3.

The total number of lines (F1) = 34, 56, 71.

The total number of long lines (F2) = 5, 7, 4.

The total number of parallel lines (F4) = 28, 41, 49.

From linear-ferning training set:

The total number of linear-ferning training sets = 3.

The total number of lines (F1) = 15, 11, 19.

The total number of long lines (F2) = 3, 1, 4.

The total number of parallel lines = 6, 2, 7.

From the non-ferning training set:

The total number of non-ferning training sets = 3.

The total number of lines (F1) = 4, 6, 0.

The total number of long lines (F2) = 0, 1, 0.

The total number parallel lines (F4) = 0, 2, 0.

After the training data were input into the **Training Set Data Processing** routine, the routine calculated the **M** and **Q** matrices and wrote **M** matrix to the **mean** file and wrote **Q** matrix to the **variance** file. These files are in the same directory as the program. One can open both files to view the content. In the **mean** file, the values of  $E(x_i)$  are

107.666664	7.666667	88.000000
53.666668	5.333333	39.333332
15.000000	2.666667	5.000000
3.333333	0.333333	0.666667

Each row of this matrix is the **mean vector** of each class (refer to Section 5.2.2) and each column in a row indicates the mean value of each feature, i.e. the total number of lines, the total number of long lines and the total number of parallel lines, respectively. Here the name "mean value" is actually the estimated mean value.

In the **variance** file, the values of  $\sigma_i$  are:

800.333313	4.333333	823.000000
346.333344	2.333333	112.333336
16.000000	2.333333	7.000000
9.333333	0.333333	1.333333

Each row of this matrix is the **variance vector** of each class and the elements in each row are also the diagonal elements the Q matrix. Each column indicates the variance value of each kind of feature, i.e. the variances of features F1, F2 and F4, respectively. Also these variances are actually the estimated values of the variances.

The above data stored in the **mean** and **variance** files are "experience information" which was then used in the later Gaussian Classification.

Let us suppose now that there is a ferning pattern the class of which is unknown. The detected features, however, were: F1 = 92, F2 = 8, F4 = 71, and these data were input to the **Gaussian Classifier** routine. The results of the probability were quickly printed out on the screen:

The probability of the full-ferning pattern:  $p(0|x) = 0.997631$

The probability of the partial-ferning pattern:  $p(1|x) = 0.002369$

The probability of the linear-ferning pattern:  $p(2|x) = 0.000000$

The probability of the non-ferning pattern:  $p(3|x) = 0.000000$

It is obvious that the pattern should belong to the full-ferning class.

When the second pattern with F1 = 17, F2 = 2, F4 = 6 were input to the routine, the results were:

The probability of the full-ferning pattern:  $p(0|x) = 0.000000$

The probability of the partial-ferning pattern:  $p(1|x) = 0.000007$

The probability of the linear-ferning pattern:  $p(2|x) = 0.999993$

The probability of the non-ferning pattern:  $p(3|x) = 0.000000$

Since the probability of the linear-ferning pattern is 0.999993, the conclusion is obvious.

The above are just two simple examples of ferning patterns. The program will be tested under many more practical cases in order to be verified and to obtain further decision-making improvement.

Our future work will be dealing with large amounts of feature parameters. Dr. Perkowski has adopted a more mature and accurate theory called parallel pattern recognition method to the ferning pattern recognition. In his method, the classification is achieved by separation of discriminant boundaries in parameter space. The most important is the case of separation with use of the hyperplanes and discriminant equations. Disposing the discriminant boundaries and equations, as well as the classified pattern's coordinates, one can confirm with better accuracy to which space class belongs the given pattern vector. The parallel pattern recognition method requires, however, huge computational resources and, at this time, it is not certain if it can be applied on the IBM PC-AT class computers without additional DSP co-processors and memory. The method will be, however, pursued in the future, after all kinds of parameters are collected from the cervical mucus images.

## CHAPTER VI

### CONCLUSION AND FUTURE WORKS

#### 6.1 CONCLUSION

(1) In this thesis a microcomputer system called ovulocomputer has been proposed and partially developed. The reasons for development of this ovulocomputer have been stated. The medical theories on which our device is based also have been briefly presented. And the design concepts of this micro-processor based ovulation predictor/detector have been described. The most important parameters used by this device are extracted from the cervical mucus which are definitely related to blood hormone levels.

(2) Almost all of the physical and chemical properties of cervical mucus have been introduced, among which, the ferning test is emphasized and described in details. The ferning test performed by this ovulocomputer is based on the image processing and pattern recognition techniques, instead of just by human visual judgement.

(3) The studies of this thesis demonstrate the feasibility of recognition of ferning patterns by Hough transformation with adaptive improvements suitable to our purpose, together with the classical statistic pattern recognition theories. The ferning features can be extracted using the presented algorithm, which mainly is a method to obtain the line-properties in cervical mucus images. This algorithm has been applied to several simulative 80×80-pixel images and the results have demonstrated the feasibility of this method (obviously, the further verification must be obtained from practical applications).

(4) This thesis has presented how to classify the ferning patterns. The feature

spaces are created. The Gaussian classifier has been introduced and will be applied in our ferning pattern recognition system. And the two-dimensional classifier of Gaussian theory has been presented in detail.

## 6.2 OUR FUTURE WORKS

Since the main topic in this thesis is the application of adapted Hough algorithm which has only been proven only to be a feasible method in the simulative images to obtain ferning features. A lot of work needs to be done in our future research.

(1) Many routines will be needed for our image preprocessing, either in spatial-domain or in frequency-domain. Considering the calculation speed and the methods available, the spatial-domain method is preferable. We will focus on two areas of image processing: image enhancement and image description.

### Image Enhancement

The cervical mucus images have their specific characteristics. The pictures shown in Figure 6 can be considered as an image corrupted by impulse "noise", which makes it difficult to apply Hough Transformation (here the word "noise" does not mean the real noise, it is just what the ferning images look like). To solve this problem, the processing procedure of image smoothing is recommended. Two routines are proposed: one is **neighborhood averaging** [21], Chapter IV), the other is **median filtering** ([21], Chapter IV). These two procedures can smooth the mucus images to make them suitable for the image description.

### Image Description

As we have mentioned in Chapter III, ferning branches are not single-pixel lines. To obtain the image of single-pixel line to apply Hough algorithm, the **thinning preprocessing** is needed (it is also called the **skeleton of the region** ). The approach which has

been developed by Zhang and Suen [24] and has been successfully applied in binary images will be used in our image preprocessing.

(2) We have discussed the optimal thresholding in Chapter III, which can automatically decide the threshold operation to convert a multiple grey-level image to a binary image. We need to obtain and develop this software to perform the automated threshold operation.

The image preprocessing mentioned above has been basically applied to some mucus images and the results are illustrated in Appendix C.

(3) The most serious problem facing us is the speed of calculations for the Hough Transformation. It is obvious that much of the computation can be done in parallel. Recently several efforts have been made to speed up the computation of the Hough transformation by utilizing parallelism [44],[45], one of which uses the systolic array processor for straight line detection. The systolic array processor maximizes the degree of parallel processing by using systolic arrays to implement the Hough Transformation for digital line detection. This method was developed and improved by H. Chuang and C. Li [46]. The basic theory of this method will be used with necessary modification in our system so that the speed and the required memory can meet the practical case.

(4) Our ferning pattern recognition system will never be properly and practically working until thousands of training sets have been obtained. It means that we have to cooperate with hospitals and experienced gynecologists and obstetricians to obtain the necessary help and information. Only after thousands of cervical mucus samples with relevant information have been obtained, can the practical ferning feature spaces be created and our ferning pattern recognition theories be applied. This is an urgent step that we will undertake in the future.

(5) There is much more work to be done in our whole ovulocomputer system than just in the ferning test automation. Although some of the parameter measurements of

physical and chemical properties are quite straightforward, not all of them are completely available and automatized. Not only should we develop the necessary device of those physical and chemical parameter extractions, computer I/O interface also must be built up. Dr. Perkowski has adopted the more sophisticated pattern recognition algorithm to perform the classification in feature spaces for the decision-making. But it may need to be developed to be more mature in application and develop into software suitable for our system.

(6) Right now we are not knowledgeable about how a ferning slide looks like under a high zooming microscope ( $1000\times$  - ). It is possible that the shape parameters can be used to recognize the ferning patterns when the cervical mucus slide is observed under a high zooming microscope. The method will be studied in the future.

(7) The integration of the whole ovulocomputer system is an absolutely necessary and final goal for us. It means that we must build up and design various kinds of chips for the system, including the Hough algorithm chip. Once the specific chips have been designed, the analysis and processing speed will meet the requirements and become competitive in the market. This is one of the inevitable steps in the future.

## REFERENCES

- [1] Guay, T., "The Personal Fertility Guide", San Francisco, CA, Pub. 1980.
- [2] Flynn, A. M., "The Value of a Cervical Score in the Assessment of Ovarian Function", *Journal of Obs. Gyn.*, Feb. 1973. Vol.80, pp. 152-159.
- [3] Nofziger, M., "A Cooperative Method of Natural Birth Control", The Book Publishing Co., Summertown, Tennessee, 1979.
- [4] World Health Organization Colloquim, "The Cervical Mucus in Human Reproduction", Geneva, 1972.
- [5] Luisi, M., "Radio-immunoassay of Salivary Progesterone for Monitoring Function in Female Infertility", *Ann. Biol. Clin.* 1978, 45, pp. 449-452.
- [6] Soudheimer, J., "Cervical Mucus and Contraception.", Thieme Medical Publisher, Seminar in Reproductive Endocrinology, 4:4, November 1986.
- [7] Kippley, J., Sheila, "The Art of Natural Family Planning", The Couple to Couple League International Inc., 1984.
- [8] Billings and Westmore, "The Billings Method", Anne O'Donovan Pty. Ltd., 1980.
- [9] Hough, P.V.C., "method and Means for Recognizing Complex Patterns", U.S. Patent 3069654, Dec. 18, 1962.
- [10] Billings, E., Billings, J., Catarinich, M., "Atlas of the Ovulation Method", Advocate Press Pty. Ltd. Melbourne, Australia, May 1977.
- [11] Kopito, LE; Kosasky HJ; Stargis SH; et al, "Water and Electrolyte Levels in Human Cervical Mucus", *Fertil. Steril.* 24:499-506, 1973.
- [12] Chantler, E. and Elstein, M., "Structure and Function of Cervical Mucus", Thieme Medical Publishers, Seminars in Reproductive Endocrinology, 4:4, Nov. 1986.
- [13] Wolf, D. and Blasco, L., "Human Cervical Mucus: I. Rheologic Characteristics II. Changes in Viscosity During the Ovulatory Menstrual Cycle", *Fertility and Sterility*, Vol.28, No.1, January 1977.
- [14] Birnberg CH; Wexler DJ; Gross M., "Estimation of Ovulation Phase by Serial Test of Cervical and Vaginal Glucose", *Obstet Gynecol* 21:194-200, 1963.
- [15] Dubelew, W. R. and Erwin, J., "Comparative Primate Biology." Vol. 3, Alan R. Liss, Inc., New York.
- [16] McSweeney, I. D. J and Sbarra, HA., "A New Cervical Mucus Test for Hormone Appraisal", *Am. J. Obstet. Gynecol.*, 88:705-709, 1964.

- [17] Odeblad, E. and Hoglund, A.; et al, "The Dynamic Model of the Human Cervical Mucus", Proc. Nord. Fert. Sco. Meeting, Umea, Jan, 1978.
- [18] Billings, E.L.; Billings, J.J.; Brown, J.B.; Burger, H.G., OP. cit., 282-4.
- [19] Billings, E.; Westmore, A., "The Billings method" Anne O'Donovan Pty. Ltd., Approved by WOOMB., 1980.
- [20] Moghissi, K.; Neuhaus, O.; Stevenson, S., "Composition and Properties of Human Cervical Mucus", May, 1960.
- [21] Gonzalez, R. and Wintz, P., "Digital Image Processing.", Addison-Wesley Publishing Company, May 1987.
- [22] Chow, C. K. and Kaneko, T., "Automatic Boundary Detection of the Left Ventricle from Cineangiograms." *Compt. and Biomed. Res.*, Vol. 5, pp. 388-410, 1972.
- [23] Fu, K. S., "Syntactic Pattern Recognition and Applications.", Prentice-Hall, Englewood Cliffs, NJ, 1982.
- [24] Zhang, T. Y. and Suen, C. Y., "A Fast Parallel Algorithm for Thinning Digital Patterns." *Commun. Ass. Compt. Mach.* 27, pp. 236-239, 1984.
- [25] Tsao, Y. F., "Skeleton Processing for Shape Analysis and Image Generation." Ph.D. Thesis, Purdue University, 1982.
- [26] Dyer, C. R. and Rosenfeld, A., "Thinning Algorithm for Grayscale Pictures.", *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. PAMI-1, no. 1, pp. 88-89, 1979.
- [27] Salari, E. and Siy, P., "The Ridge-Seeking Method for Obtaining the Skeleton of Digital Images." *IEEE Trans. Syst. Man. Cyb.*, Vol. SMC-14, no. 3, pp. 524-528, 1984.
- [28] Rosenfeld, A., "Picture Processing by Computer.", Academic Press, New York, 1969.
- [29] Duda, R. O. and Hart, P. E., "Use of the Hough Transformation to Detect Lines and Curves in Pictures.", *Commun. ACM* 15, pp. 11-15, 1972.
- [30] Veen, T. M. V. and Groen, F. C. A., "Discretization Errors in the Hough Transformation." *Pattern Recognition*, Vol. 14, no. 1-6, pp. 137-145, 1981.
- [31] Cohen, M. and Toussaint, G. T., "On the Detection of Structures in Noisy Pictures.", *Pattern Recognition Pergamon Press*, 1977, Vol. 9, pp. 95-98.
- [32] O'Gorman, F. and Clowes, M. B., "Finding Picture Edege Through Collinearity of Feature Points." *IEEE Trans. Comput.*, Vol. C-25, no. 4, pp. 449-454, 1976.
- [33] Ballard, D. H., "Generalizing the Hough Transformation to Detect Arbitrary Shapes.", *Pattern Recognition*, Vol. 13, no. 2, pp. 111-122, 1981.
- [34] Iannino, A. and Shapiro, S. D., "A Survey of the Hough Transformation and its Extension for Curve Detection.", in *Proc. 1978 Int. Conf. Pattern Recognition and Image Processing*, IEEE, pp. 32-38, 1978.
- [35] Stockman, G. C. and Agrawala, A. K., "Equivalence of Hough Curve Detection of Template Matching.", *Commun. ACM*, Vol. 20, no. 11, November, 1977.

- [36] Silberberg, T. M., "The Hough Transformation on the Geometric Arithmetic Parallel Processor.", 1985, Center for Automation Research, Univ. of Maryland.
- [37] Rhodes, M. F., et al, "A Monolithic Hough Transform Processor Based on Restructurable VLSI.", IEEE Trans. Pattern Anal. Mach. Intell. Vol. 10, no. 1, January 1988.
- [38] Li, H. and Lavin, M. A., "Fast Hough Transformation Based on Bintree Data Structure.", IEEE, Trans. Pattern Anal. Mach. Intell. Vol. 10, Feb. 1986.
- [39] Fu, K. S., "Application of Pattern Recognition.", CRC Press, new York, 1982.
- [40] Fu, K. S., "Sequential Method in Pattern Recognition and Machine Learning.", Academic Press, New York, 1968.
- [41] Bacus, J. and Gose, E., "Leakocyte Pattern Recognition.", IEEE Trans. on System Man. and Cybernetics SMC-2(4), pp. 513-526, 1972.
- [42] Devre, J., "Probability & Statistics for Engineering and the Sciences.", Brooke/Cole Publishing Co., 1982.
- [43] Bacus, J., and Weens, J., "An Automated Method of Differential Red Cell Classification with Application to the Diagnosis of Anemia", Histochem. Soc., Inc., Vol. 25, No. 7 pp. 614-632, 1977.
- [44] Rosenfeld, A. and Pfaltz, J., "Sequential Operation in Digital Picture Processing.", Journal of Ass. Comput. Mach., Vol. 13, no. 4, pp. 471-494, October 1966.
- [45] Hanahara, K. and Maruyama, T., "A Real-time Processor for the Hough Transformation.", IEEE Trans. Pattern Anal. Mach. Intell. Vol. 10, no. 1, January 1988.
- [46] Chuang, H. Y. and Li, C. C., "A Systolic Array Processor for Straight Line Detection by Modified Hough Transformation.", Dept. of Computer and Electrical Engineering, Univ. of Pittsburgh, PA, 1985.

**APPENDIX A**

**THE PROGRAM OF THE HOUGH  
TRANSFORMATION APPLICATION**

```

/*****

This is a program of application of Hough Transform
to detect lines and other features of these
lines in the cervical mucus images. It extracts the
features of the lines such as the number of the
lines and the length and parallel situation of them

*****/

#include <stdio.h>
#include <math.h>

#define PI 3.14159
#define IMX 80    /* The size of the image */
#define IMY 80    /* The size of the image */

#define STEP_Q 3
/* The quantizing interval in parameter space */

#define STEP_P 0.5
/* The quantizing interval in parameter space */

#define PARALLEL_NO 2
/* The threshold for the number of parallel lines*/

unsigned char ac[454][180/STEP_Q];
int zero, PMTR_Q, PMTR_P, THRESHOLD;
unsigned char pix[IMY][IMX], img[IMY][IMX];
/* pix[][] is the image to be performed the Hough Transform
img[][] is the reconstructed image */

unsigned char q[3000], pl[100], Q[100];
int k, K, max, mq;
double pet, sqrt(), p[3000], mp, COS[60], SIN[60];
/* COS[] and SIN[] are the array to be loaded with the values
of mathematical functions sin() and cos() */

main()
{
    char file_name[10];
    int i, c, x, y;
    double pmtr_p;
    FILE *fp, *fopen();

    pmtr_p = sqrt((double)(IMX * IMX + IMY * IMY));
/* The longest line(diagonal) in 80 by 80 image */

    PMTR_P = ((int)(pmtr_p / STEP_P + 0.5)) * 2 + 1;
/* the size of P coordinate */

```

```

    PMTR_Q = (int)(180 / (double)STEP_Q + 0.5);
/* the size of Q coordinate */

    for(y=0; y<PMTR_P; y++)
        for(x=0; x<PMTR_Q; x++)
            ac[y][x] = 0;
    zero = (PMTR_P - 1)/2 + 1;
/* In the parameter space when p=0 */

    printf("\n Input the name of the image file to be processed: ");
    scanf("%s", file_name);
    fp = fopen(file_name, "r");
/* Open the input file to read the image */

    x=0;
    y=0;

/* following input the character 0 and 1 from the image file
and assigns them to the matrix pix[][] */

    while((c=getc(fp)) != EOF)
    {
        if ( c=='0' || c == '1')
        {
            if (x < IMX)
                pix[y][x] = c;

            else
            {
                x=0;
                y++;
                pix[y][x]= c;
            }
            x++;
        }
        else
            ;
    }

/* following loads the values of cos() and sin() function
so that it speeds the calculation. */

    for (i=0; i<60; ++i)
    {
        COS[i] = cos(PI/60*i);
        SIN[i] = sin(PI/60*i);
    }

    hough();

}

```

```

/*****

This subroutine calls some subroutines to perform
the Hough Transformation and feature extraction.
It also outputs the results to a file of which users
are prompted to input the name.

*****/

hough()
{
    FILE *fp, *fopen();
    char file_name[10];
    int i, a, b, x, y;

    for (y=0; y<IMY; y++)
        for (x=0; x<IMX; x++)
        {
            if(pix[y][x] == '1')
                pmtr_spe(x,y);
        }
    /* When the point is a feature point, call pmtr_spe() to
    perform the parameter computation */

    printf("\nInput the threshold for parameter space: ");
    scanf("%d",&THRESHOLD);

    peak();
    /* peak() function searches the peaks which are bigger the threshold */

    remap_image();
    /* remap_image() function reconstructs the image from parameter space */

    printf("\n nIn what file would you like to store the data ? ");
    scanf("%s", file_name);

    if (strcmp(file_name, "stdout") == 0)
        fp = stdout;
    else
        fp = fopen(file_name, "w");
    /* Open the named file to write the results to this file */

    fprintf(fp, "\nThis is the mapped image in parameter space:\n\n");

    for(b=0; b<PMTR_P; ++b)
        for(a=0; a<PMTR_Q; ++a)
        {
            fprintf(fp, "%2d", ac[b][a]);
            if(a == 59)
                fprintf(fp, "\n");
        }
}

```

```

for(i=0; i<k; ++i)
{
    if (p[i] >= 0)
        fprintf(fp, "\n Q= %d p= %.1f %d ",
            q[i],p[i],ac[(int)(p[i]/STEP_P+0.5)+zero][q[i]/STEP_Q]);
    else
        fprintf(fp, "\n Q= %d p= %.1f %d ",
            q[i],p[i],ac[(int)(p[i]/STEP_P-0.5)+zero][q[i]/STEP_Q]);
}

fprintf(fp, "\n\n Total number of peaks bigger than %d is %d ",
        THRESHOLD, k);
fprintf(fp, "\n");
fprintf(fp, "\n Maximum peak = %d when Q = % d , p = %.1f \n",
        max, mq, mp);
fprintf(fp, "\n\n\n");

if (K == 0)
    fprintf( fp, "\n No lines bigger than thredhold are in parallel !\n ");
else
{
    for(i=0; i<K; ++i)
        fprintf(fp, "\n When Q = % d, % d lines are in parallel", Q[i], pl[i] );
}

fprintf(fp, "\n\n %.2f percent of all lines are in parallel.", pet);
fprintf(fp, "\n\n\n");
fprintf(fp, "Following is the image remapped from parameter space:\n\n\n");

fprintf(fp, "\n");

for(i=0; i<IMY; ++i)
    for(x=0; x<IMX; ++x)
    {
        fprintf( fp, "%d", img[i][x]);
        if(x == 79)
            fprintf(fp, "\n");
    }

fprintf(fp, "\n\n\n");
}

```

```

/*****

```

This subroutine does the main task of Hough Trans-formation. It calculates the transform from picture space to parameter space.

```

*****/

```

```

pmtr_spe(X, Y)
int X, Y;
{
    int a, b, i, j;
    double p;

    i = 0;

    for(a=0; a<PMTR_Q; a++)
    {
        p = (X * COS[a] + Y * SIN[a]) / STEP_P;
        /* p is the normal parameter in the parameter space */

        if (p < 0)
            b = (int)(p - 0.5) + zero;
        /* shift to accumulator coordinate */

        else
            b = (int)(p + 0.5) + zero;
            ac[b][i]++;
            i++;
        /* Each time the accumulator ac[][] increases by 1 */

    }
}

/*****

This subroutine searches for the peaks which are
bigger than the desired threshold in the parameter
space.

*****/

peak()
{
    int i, j;

    k = 0;

    for(i=0; i<PMTR_P; ++i)
        for(j=0; j<PMTR_Q; ++j)
        {
            if ( ac[i][j] > THRESHOLD )
            {
                p[k] = (i - zero) * STEP_P;
                q[k] = j * STEP_Q;
                k++;
            }
        }
    /* Load the coordinates of each peak to p[] and q[] array */
}

```

```

        }
        else
        ;
    }

for(i=0; i<k; ++i)
{
    if(p[i] >=0 )
        printf("\n Q=%d , p=%.1f, peak=%d  ", q[i], p[i],
            ac[(int)(p[i]/STEP_P+0.5)+zero][q[i]/STEP_Q]);

    else
        printf("\n Q=%d , p=%.1f, peak=%d  ", q[i], p[i],
            ac[(int)(p[i]/STEP_P-0.5)+zero][q[i]/STEP_Q]);
}

printf("\n\n Total number of peaks bigger than  %d  is %d \n\n",
        THRESHOLD,k);

longest();

parallel();

}

```

```

/*****

```

This subroutine looks for those peaks with the same Q values, which refer to the parallel lines in the picture space.

```

*****/

```

```

parallel()
{
    int i, j, flag[100], pl_sum;
/* flag[100] supposes there are no more than 100 parallel lines
for each Q (angle), is may need to be changed if the parallel
lines for each Q are more than 100 in the future processing */

    K = 0;

    for(i=0; i<100; ++i)
    {
        Q[i] = flag[i] = 0;
        pl[i] = 1;
/* pl[] stores the number of parallel lines for each Q found */
/* Q[] stores the corresponding line angles */
    }

    for(i=0; i<(k-1); ++i)
    {

```

```

    if(flag[i] == 0)
    {
        for(j=i+1; j<k; ++j)
        {
            if(q[i] == q[j])
            {
                pl[K]++;
                flag[j] = 1;
            }
            else
            ;
        }
        if(pl[K] >= PARALLEL_NO)
        {
            Q[K] = q[i];
            K++;
        }
    }
    else
    ;
}

pl_sum = 0;

for (i=0; i<K; ++i)
    pl_sum = pl_sum + pl[i];
/* pl_sum is the total number of parallel lines in a image */

pet = (double) pl_sum / k * 100;

if(K == 0)
    printf("\n No lines bigger than threshold are in parallel !\n ");
else
    {
        for(i=0; i<K; ++i)
            printf("\n When Q = %d , %d lines are in parallel", Q[i], pl[i] );
    }

    printf("\n\n %.2f percent of all lines are in parallel.", pet);
}

```

```

/*****

```

This subroutine looks for the maximum peak in the parameter space, which refers to the longest line in the picture space.

```

*****/

```

```

longest()
{
    int i, m, n;

```

```

if(p[0] >=0)
    max = ac[(int)(p[0]/STEP_P+0.5)+zero][q[0]/STEP_Q];
else
    max = ac[(int)(p[0]/STEP_P-0.5)+zero][q[0]/STEP_Q];

    mp = p[0];
    mq = q[0];

for(i=1; i<k; ++i)
{
    if(p[i] >= 0)
        n = (int)(p[i]/STEP_P+0.5)+zero;
    else
        n = (int)(p[i]/STEP_P-0.5)+zero;

    m = q[i]/STEP_Q;

    if( max < ac[n][m])
    {
        max = ac[n][m];
        mp = p[i];
        mq = q[i];
    }
    else
        ;
}

printf("\n Maximum peak = %d when Q = %d , p = %.1f\n",
        max, mq, mp );
}

```

```

/*****

```

This subroutine reconstructs the image from the information obtained in parameter space. It verifies the accuracy of the information obtained from the above routines of Hough Transformation.

```

*****/

```

```

remap_image()
{
    int i, x, y;

    for(i=0; i<IMY; ++i)
        for(x=0; x<IMX; ++x)
            img[i][x] = 0;
}

```

```
for(i=0; i<k; ++i)
  for(x=0; x<IMX; ++x)
  {
    y=(int)((p[i]-x*COS[q[i]/STEP_Q])/SIN[q[i]/STEP_Q]+0.5);
    if(y < 0 || y > 79 )
      ;
    else
      img[y][x] = 1;
  }
}
```

**APPENDIX B**

**A PROGRAM OF**

**GAUSSIAN CLASSIFIER**

```

/*****

```

This is a Gaussian Classifier program which contains two main procedures. One is the training set data processing and the other is classifier procedure.

```

*****/

```

```

#include <stdio.h>
#include <math.h>

```

```

#define DIM 3
/* 3 dimensional space */

```

```

#define CLASS 4
/* Four class ferning patterns */

```

```

#define P 0.25
/* assume the probability of each pattern is the same */

```

```

float Q_inv[DIM][DIM], product;
float M1[DIM], M2[DIM], M3[DIM], M4[DIM];
/* M1, M2, M3, M4 are mean vectors of full-ferning, partial-ferning,
linear-ferning and non-ferning patterns, which are obtained from
the training sets. M[0], M[1], M[2] elements are the total numbers
of lines, long lines and parallel lines, respectively. */

```

```

float Q1[DIM][DIM], Q2[DIM][DIM], Q3[DIM][DIM], Q4[DIM][DIM];
/* The covariance matrix Q1, Q2, Q3, which are obtained from
the training sets */

```

```

main()

```

```

{
    float mm[CLASS*3], vv[CLASS*3];
/* mm[] and vv[] array are used to load the mean and variance values
form the files named 'mean' and 'variance', respectively. */

```

```

    int i, j;
    char d;
    FILE *fp, *fopen();

```

```

    printf ("\n*****\n\n");
    printf ("\nThere are two procedures in this routine:\n");
    printf ("(1)Training set data processing, and (2)Gaussian Classifier.\n\n");
    printf ("Type 't' to go to the Training set data processing.\n");
    printf ("Type 'g' to go to the Gaussian Classifier procedure\n\n");
    printf ("*****\n\n");
    scanf ("%c", &d);

```

```

    if (d == 't')
        train();

```

```

else if (d == 'g')
{
    fp = fopen ("mean", "r");
    for(i=0; i<(3*CLASS); ++i)
    {
        fscanf(fp, "%f", &mm[i]);
    }
/* Input the data from the file named 'mean' */

    for(i=0; i<(3*CLASS); i++)
    {
        if(i<3)
            M1[i] = mm[i];
        else if(i>=3 && i<6)
            M2[i-3] = mm[i];
        else if(i >= 6 && i<9)
            M3[i-6] = mm[i];
        else
            M4[i-9] = mm[i];
    }
/* Assign the data to the mean vectors to M1, M2, M3 and M4 matrices */

    for(i=0; i<DIM; ++i)
    for(j=0; j<DIM; ++j)
        Q1[i][j] = Q2[i][j] = Q3[i][j] = Q4[i][j] = 0;

    fp = fopen("variance", "r");

    for(i=0; i<(3*CLASS); ++i)
        fscanf(fp, "%f", &vv[i]);
/* Input the data from file named 'variance' */

    for(i=0; i<(3*CLASS); ++i)
    {
        if(i<3)
            Q1[i][i] = vv[i];
        else if(i>=3 && i<6)
            Q2[i-3][i-3] = vv[i];
        else if(i>=6 && i<9)
            Q3[i-6][i-6] = vv[i];
        else
            Q4[i-9][i-9] = vv[i];
    }
/* Assign the data to Q1, Q2, Q3 and Q4 matrices */

    gauss();

}
else
;
}

```

```

question()
{
    char answer;

    printf("\nMore pattern to be recognized ? (y/n)\n");
    scanf("%s", answer);
    if (!strcmp(answer, "y"))
        gauss();
    else
        ;
}

/*****

This subroutine computes the  $p(j|X)$  which is the probability
of the class to which the input pattern belongs.

*****/

gauss()
{
    float *Qp[CLASS], *Mp[CLASS];
    int X[DIM];
    int i,j,k;
    float qq, s[3][3], p_j[CLASS], p_jx[CLASS], p_ix;
/* p_j[] is the numerator of probability p(j|X),
   p_ix is the denominator of probability p(j|X),
   p_jx is the probability p(j|X) */

    FILE *fp, *fopen();

    printf ("\nInput the number of total lines (feature F1): ");
    scanf ("%d",&X[0]);
    printf ("\nInput the number of long lines (feature F2): ");
    scanf ("%d", &X[1]);
    printf ("\nInput the number of parallel lines (feature F4): ");
    scanf ("%d", &X[2]);

/* Following passing the basic addresses of the matrix to
   the pointers. */

    Qp[0] = &Q1[0][0];
    Qp[1] = &Q2[0][0];
    Qp[2] = &Q3[0][0];
    Qp[3] = &Q4[0][0];
    Mp[0] = &M1[0];
    Mp[1] = &M2[0];
    Mp[2] = &M3[0];
    Mp[3] = &M4[0];

    p_ix = 0.0;

    for ( i=0; i<CLASS; ++i)

```

```

{
    p_j[i] = 0;

    inverse( Qp[i], &qq );

    mtx_optn( X, Mp[i] );

    p_ix = P * sqrt( 1/qq ) * exp(-0.5 * product) + p_ix;
/* Compute the denominator of p(j|X) */

    p_j[i] = P * sqrt(1/qq) * exp(-0.5 * product);
/* Compute the numerator of p(j|X) */

}

for (i=0; i<CLASS; ++i)
    p_jx[i] = p_j[i] / p_ix;
printf("\n*****\n\n");

printf("The probability of full-ferning pattern:
    p(0|x) = %f\n\n", p_jx[0]);
printf("The probability of partial-ferning pattern:
    p(1|x) = %f\n\n", p_jx[1]);
printf("The probability of linear-ferning pattern:
    p(2|x) = %f\n\n", p_jx[2]);
printf("The probability of non-ferning pattern:
    p(3|x) = %f\n\n", p_jx[3]);

fp = fopen("prob", "w");

fprintf(fp, "The probability of full-ferning pattern:
    p(0|x) = %f\n\n", p_jx[0]);
fprintf(fp, "The probability of partial-ferning pattern:
    p(1|x) = %f\n\n", p_jx[1]);
fprintf(fp, "The probability of linear-ferning pattern:
    p(2|x) = %f\n\n", p_jx[2]);
fprintf(fp, "The probability of non-ferning pattern:
    p(3|x) = %f\n\n", p_jx[3]);

question();

}

```

/\*\*\*\*\*\*

This subroutine performs the matrix's multiplication operation.

\*\*\*\*\*/

```

mtx_optn( x, m )
float m[];
int x[];

```

```

{
  int i, j, k;
  float sum, T[DIM], x1[DIM];

  for (i=0; i<DIM; ++i)
    x1[i] = x[i] - m[i];
  for(i=0; i<3; ++i)
  {
    T[i] = 0.0;

    for(j=0; j<3; ++j)
      T[i] = x1[j] * Q_inv[j][i] + T[i];

  }

  product = 0.0;
  for (i=0; i<DIM; ++i)
    product = T[i] * x1[i] + product;
}

/*****

```

This subroutine performs the inverse operation of the matrix and it also calculates the algebra remainder of the matrix.

\*\*\*\*\*/

```

inverse( cov, pp )
float cov[DIM][DIM], *pp;
{
  int i, j, a, b, m, n;
  float q, p[2][2], A[3][3];

  /* q is the determinant of the matrix Q */

  q = cov[0][0] * cov[1][1] * cov[2][2];

  /* Following calculates the algebra remainder of the matrix cov[][] */

  for (a=0; a<3; ++a)
    for (b=0; b<3; ++b)
    {
      m = n = 0;
      for (i=0; i<3; ++i)
        for (j=0; j<3; ++j)
        {
          if (i == a || j == b)
            ;
          else
            {
              p[m][n] = cov[i][j];
            }
        }
    }
}

```

```

        ++n;
        if ( n>1 )
        {
            n = 0;
            ++m;
        }
        else
        ;
    }
}

if ( (a+b) % 2 == 0)
    A[b][a] = p[0][0]*p[1][1] - p[0][1]*p[1][0];
else
    A[b][a] = - p[0][0]*p[1][1] + p[0][1]*p[1][0];
}
for (i=0; i<3; ++i)
    for (j=0; j<3; ++j)
        Q_inv[i][j] = A[i][j] / q;
*pp = q;
}

```

\*\*\*\*\*

This subroutine performs the main task of the training set data processing. It requires the user to input the features of a pattern to be recognized and it calculates the mean vector matrix and the variance vector matrix. Then it stores them in files named 'mean' and 'variance', respectively.

\*\*\*\*\*/

```

train()
{
    int n, i;
    float m1[DIM], m2[DIM], m3[DIM], m4[DIM];
    /* m1[], m2[] and m3[] are used to load the mean vectors of
       the number of total lines, long lines and parallel lines,
       respectively. */

    float v1[DIM], v2[DIM], v3[DIM], v4[DIM];
    /* v1[], v2[] and v3[] are used to load the variance vectors
       of the number of total lines, long lines and parallel lines,
       respectively. */

    float mw[3*CLASS], vw[3*CLASS];
    FILE *fp, *fopen();

    printf ("\n***** Full-ferning Training Set Processing *****\n");
    printf ("\nInput the total number of full-ferning training sets: ");

    mean_var(&m1[0], &m1[1], &m1[2], &v1[0], &v1[1], &v1[2]);
}

```

```

printf ("\n***** Partial-ferning Training Set Processing *****\n");
printf ("\nInput the total number of partial-ferning training sets: ");

mean_var(&m2[0], &m2[1], &m2[2], &v2[0], &v2[1], &v2[2]);

printf ("\n***** Linear-ferning Training Set Processing *****\n");
printf ("\nInput the total number of linear-ferning training sets: ");

mean_var(&m3[0], &m3[1], &m3[2], &v3[0], &v3[1], &v3[2]);

printf ("\n***** Non-ferning Training Set Processing *****\n");
printf ("\nInput the total number of non-ferning training sets: ");

mean_var(&m4[0], &m4[1], &m4[2], &v4[0], &v4[1], &v4[2]);

for(i=0; i<DIM; ++i)
{
    mw[i] = m1[i];
    mw[i+3] = m2[i];
    mw[i+6] = m3[i];
    mw[i+9] = m4[i];
    vw[i] = v1[i];
    vw[i+3] = v2[i];
    vw[i+6] = v3[i];
    vw[i+9] = v4[i];
}

fp = fopen("mean", "w");
for(i=0; i<12; ++i)
{
    fprintf(fp, "%20f", mw[i]);
    if(i==2 || i==5 || i== 8 || i==11)
        fprintf(fp, "\n");
}

fp = fopen("variance", "w");
for(i=0; i<12; ++i)
{
    fprintf(fp, "%20f", vw[i]);
    if(i==2 || i==5 || i==8 || i==11)
        fprintf(fp, "\n");
}
}

/*****

```

This subroutine calculates the mean values and the variance of every kind of class from the input training set data and returns these values to the train() routine.

```

*****/

```

```

mean_var(totl, lng, parall, v_totl, v_lng, v_parall)
float *totl, *lng, *parall, *v_totl, *v_lng, *v_parall;
{
    int n, i;
    int X1[100], X2[100], X3[100];
    float sum1, sum2, sum3;
    float v_sum1, v_sum2, v_sum3;

    scanf ("%d", &n);
    printf ("The total number of training sets is %d !\n", n);

    for (i=0; i<n; ++i)
    {
        printf ("\nFromn the NO.%d training image,
                input the following:", i+1);
        printf ("\nThe total number of lines detected: ");
        scanf ("%d", &X1[i]);
        printf ("\nThe total number of long lines detected: ");
        scanf ("%d", &X2[i]);
        printf ("\nThe total number of parallel lines detected: ");
        scanf ("%d", &X3[i]);
    }

    for (i=0; i<n; ++i)
        printf ("X1=%d, X2=%d, X3=%d\n", X1[i], X2[i], X3[i]);

    sum1 = sum2 = sum3 =0;
    v_sum1 = v_sum2 = v_sum3 = 0;

/* The following calculates the mean value of each kind of parameter */

    for (i=0; i<n; ++i)
    {
        sum1 = X1[i] + sum1;
        sum2 = X2[i] + sum2;
        sum3 = X3[i] + sum3;
    }
    printf ("sum1 = %f, sum2 = %f, sum3 = %f\n", sum1, sum2, sum3);
    *totl = sum1 / n;
    *lng = sum2 / n;
    *parall = sum3 / n;

/* The following calculates the variance of each kind of parameter */
    for(i=0; i<n; ++i)
    {
        v_sum1 = (X1[i]-*totl)*(X1[i]-*totl) + v_sum1;
        v_sum2 = (X2[i]-*lng)*(X2[i]-*lng) + v_sum2;
        v_sum3 = (X3[i]-*parall)*(X3[i]-*parall) + v_sum3;
    }

/* The following returns the results to the train() routine. */
    *v_totl = v_sum1 / (n-1);

```

```
*v_lng = v_sum2 / (n-1);  
*v_parall = v_sum3 / (n-1);  
}
```

## **APPENDIX C**

### **SOME PREPROCESSED MUCUS IMAGES**

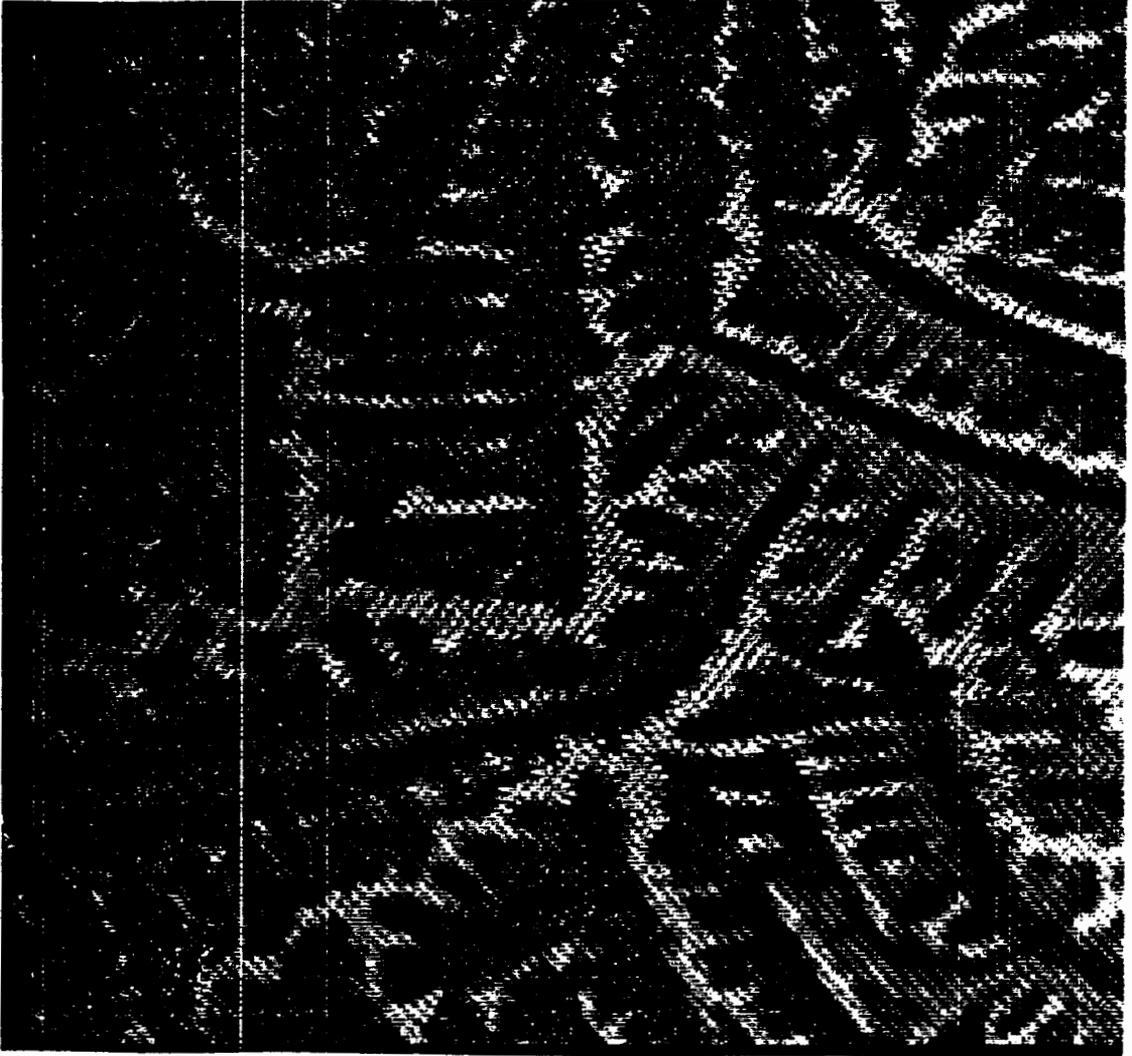


Figure 29. An original full-ferming mucus image.

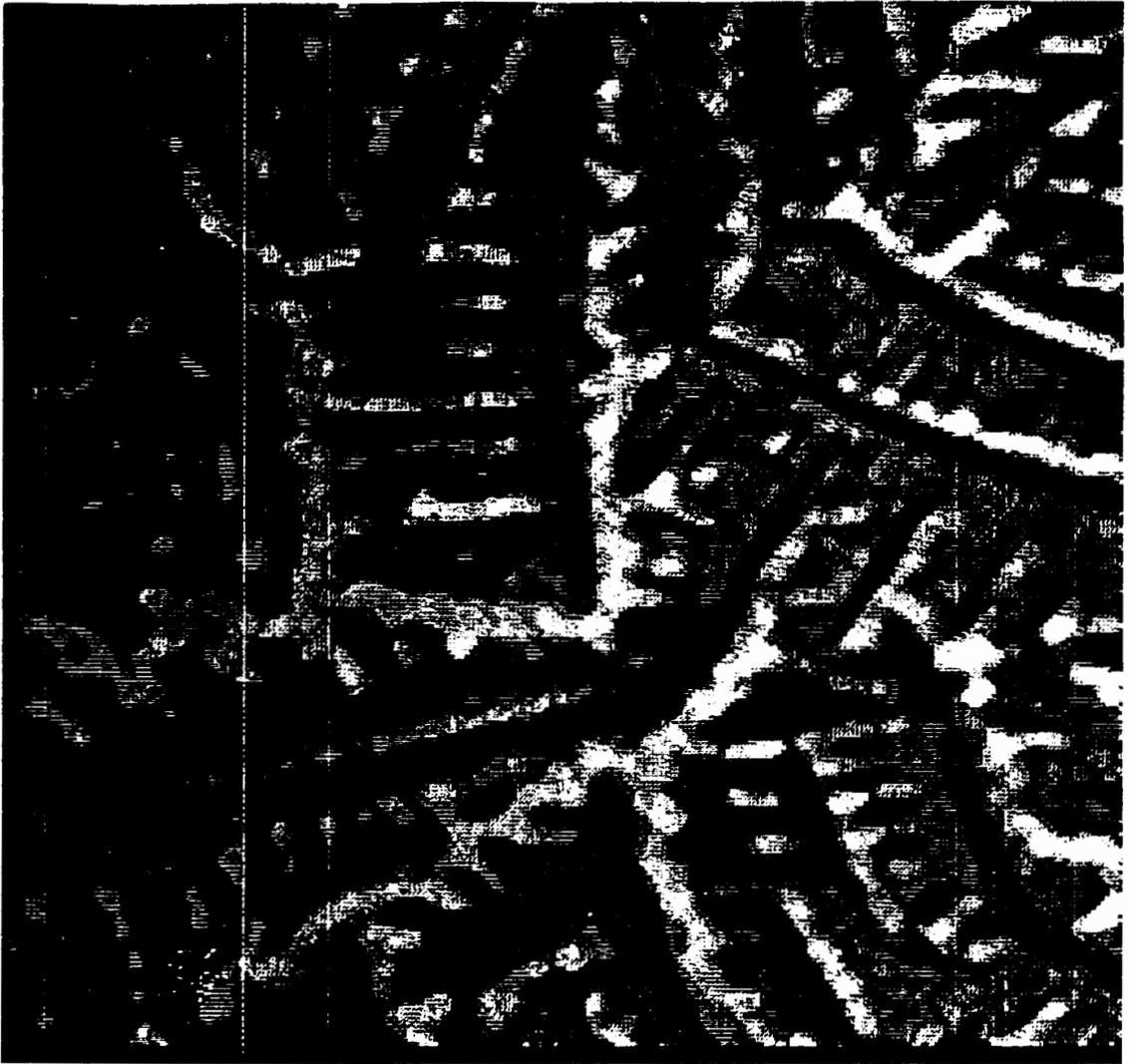


Figure 30. The result of applying the median filter on Figure 29.



Figure 31. The result of thresholding of Figure 30, the image became a binary image.



Figure 32. The result of thinning procedure of Figure 31.