INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

Performance Evaluation of a Second Generation Metaphase Finder for Chromosome-Based Radiation Dosimetry

By
David Frederic Pollitt

A dissertation submitted to the
School of Graduate Studies and Research,
University of Ottawa,
in partial fulfilment of the requirements
for the degree of
Master of Science (Systems Science)

Systems Science Program
Faculty of Administration
University of Ottawa
December 1997

Abstract

In this thesis an automated chromosome detection system will be described. The biological and clinical background for the need of such a system is presented with emphasis on cell biology. Work was done on three versions of the autofocusing algorithm as well as on two versions of the chromosome detection algorithm. Extensive testing was done to determine the best versions of these algorithms and additional program functionality was used to carry out this testing.

The results of the testing on the autofocusing algorithms are presented in graphs that show a peak when the image is properly focused. The results of the testing and optimisation of the chromosome detection algorithms are displayed in Receiver Operating Characteristic curves. These curves allow the user to pick a point of algorithm functioning which yield an acceptable quantity of well formed chromosome spreads or True Positives along with a given proportion of poorly formed chromosome spreads or False Positives. The accuracy of the final algorithms are well within the desired limits.

Acknowledgements

In the production of this thesis I would like to thank all those who gave me assistance and encouragement, especially advisors Dr. Jean-François Rivest, Dr. David Gibbons, Dr. Jack McLean, and Dr. Frank Johnson. I would also like to thank Dr. Jean-Michel Thizy, the current Director of the Systems Science program at the University of Ottawa, and Dr. Daniel Lane the former Director.

Finally, I would like to thank my parents Edward and Eleanor Pollitt, my brother Chris and his wife Michelle, other family and friends, and the congregation of St. George's Anglican Church, Ottawa for their prayerful support.

This work was funded by Ottawa Instrumentation Ltd.
Work was carried out at Health Canada, Radiation Protection Bureau, and the use of their facilities was appreciated.

Table of Contents

List of Figures

List of Tables

List of Images

Chapter 1

Introduction

1.1 Problem Statement

In the development of new treatments for various diseases there is a need for studies of the effectiveness and risks of these treatments. One such study or test that needs to be carried out is to look for chromosome aberrations which may be caused by certain treatments. Chromosome aberrations in patient's cells are an indicator of cancer risks. In this thesis project we are interested in finding chromosome spreads which contain a collection of 46 individual chromosomes. This, however, requires much tedious effort and time by an analyst looking through a microscope at a slide containing chromosome spreads. Therefore, an automated procedure is required for this process.

Another test that can be performed on a patient is to diagnose the amount of radiation that he or she has been exposed to in an accident. This is represented by an elevated proportion of damaged cells in which there are chromosome aberrations.

There are two problems that need to be solved in the analysis of chromosomes. The first problem it to locate chromosome spreads on a slide under a microscope and the second problem is to analyse these spreads for the presence of aberrations.

To detect chromosome aberrations in a chromosome spread is a very difficult task and requires a human operator to analyse the chromosome spread manually. What we propose to do in

this thesis is to assist the operator in this task by automating the location of chromosome spreads. This location procedure is to take place over night. When the human operator comes in the next day he or she can then scan through the series of chromosomes, that the computer has located during the night, at higher magnification.

If a program can be developed to assist in locating these chromosome spreads this will mean that studies of the effectiveness and risks of new treatment programs and radiation exposure assessment for patients can be done much faster and at a lower cost. This will benefit the patient, reduce treatment development costs, and hopefully mean that the new treatments will become available sooner.

The problem statement for this thesis is to produce a fast and accurate chromosome spread detection system. This system will consist of three components. The first component is the focusing of the microscope on the slide. The second component is the automated movement of the platform containing the slide. Finally the third, and most challenging component, is to automatically locate the chromosomes on that slide.

In this thesis the first and third components will be developed. The second component will be added to the final version of the system and is independent of the other two components.

1.2 Contributions

The contributions that have been made in this thesis are the following:

(i) The development of a fast and accurate autofocusing algorithm.

(ii) The development of a new chromosome spread finder.

Chapter 2

The Images

2.1 Biological Background

Most cells in the human body contain 46 chromosomes. The exceptions to this are the germ cells (or sex cells) that contain only half of the DNA (deoxyribonucleic acid) material, and the red blood cells. A red blood cell (or erythrocyte) does contain chromosomes in its nucleus when it is young, but sheds its nucleus when it matures in order to reduce in size to move throughout the body. Radiation damage, chemical damage, and viral damage to cells can produce aberrations in the chromosomes. These forms of cell damage can rewrite the DNA sequence by introducing errors. If the chromosomes are damaged the cell may malfunction and start reproducing in an uncontrolled manner. This causes cancer, an uncontrollable growth of mutant cells in the body. This uncontrollable growth of cancer cells taps the resources of the body and can eventually cause the death of the patient because there are not enough resources for the normal cells to reproduce. This deterioration can be stopped if the mutant cells are destroyed or removed from the body. Careful examination of a chromosome spread from a cell nucleus will allow for health risks to a patient to be assessed.

2.1.1 Cell Life Cycle

There are many different types of cells in the human body. These different cells all have different life spans. Cells die and need to be replaced. The reproduction of a cell is controlled by the DNA in that cell. The life cycle of a cell has five phases. These phases are as follows: G0, G1, S, G2, and D (or M). These phases are described below. See Figure 2.1 for a graphical description of the Cell Life Cycle. The length of time that the cell is in the G0 phase may be hours, days, or years depending on the cell type. The G1 phase may take from several minutes to several hours to complete. The S phase takes around eight hours; the G2 phase takes around four hours, and finally the D phase takes around an hour to complete.

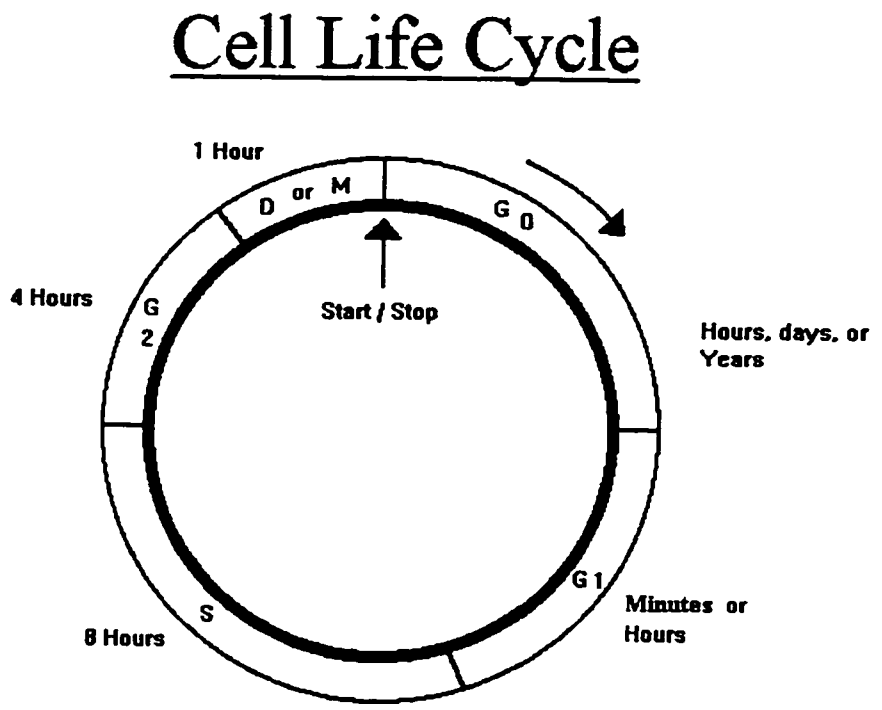


Figure 2.1: The cell life cycle.

The G0 phase is the normal working state for a cell. The cell will stay in this state until it starts the process of dividing. A typical white blood cell (or lymphocyte) may be in this state for many years (approximately three to five years). Other types of cells are in the G0 state for a much

shorter or much longer length of time. Red blood cells live for four to five days. Neurone cells in the brain do not divide at all during the life span of an individual.

Cells form in one of two places. The normal place for a cell to be "born" is in a cell nursery. There are various cell nurseries in the body for the different types of cells. For example skin cells form in a deep layer of the skin. Then they flatten out and move to the outer surface of the skin, die, and eventually flake off. Red blood cells form in the bone marrow, and then enter the blood stream through small fissures in the bones. These cells are formed from special cells that divide regularly in order to replace older cells that die off.

The other place that a cell can form is from a mature cell in it's normal location. For example, a white blood cell can reproduce in the blood stream. If a virus or bacterium attacks the body this will trigger the large production of white blood cells to attack the intruder. Once the intruder has been eliminated then the white blood cell count will return to normal.

The reproduction of cells is heavily regulated by the body system. In a culture outside the body a cell will reproduce by dividing if it has sufficient nourishment. In the body, however, a cell will not reproduce even if it has sufficient nourishment unless it is given the appropriate chemical "go ahead" signal by other cells in the body. If a patient has received damage to the skin then the skin cells around the damaged area reproduce at a high rate to fill in the gap. Once this is done the cells stop reproducing at that accelerated rate.

If a cell's chromosomes are damaged the cell may malfunction and start reproducing even in the absence of chemical signals from other cells. Cancer is caused by an uncontrollable growth of mutant cells in the body. These cancer cells may also lose their effectiveness in terms of what function they used to perform when they were normal.

When a cell starts the process of division it enters the G1 phase. The G1 phase is the first step in the process of cell division. G1 means "Gap #1". Not very much is known about what happens in this stage except material is produced in the cell that will be used at later stages in cell division.

When a cell is going to start the process of dividing it must prepare to enter the S phase. It does this by producing various cell tissues and it enlarges in size. The S phase is when the cell duplicates its 46 chromosomes. The S means "synthesise". There are 23 pairs of chromosomes in a human cell. During the first part of the S phase the chromosomes are very long and stringy. At this stage of reproduction the DNA in the chromosomes splits and unwinds. Then new material attaches to both halves of the DNA strands to duplicate the DNA. During this process of duplication the chromosomes are more susceptible to damage from external radiation.

Once the chromosomes have reproduced the cell enters the G2 phase. G2 means "Gap #2". During this phase the cell is preparing to enter the D phase in which the cell divides. Not much is known about what happens during this phase except that the cell continues to grow in size and produce more cell tissue.

Finally, the cell enters the D phase. The D stands for "Divide". Another term for this phase is the M phase or "mitosis" phase. The two groups of 46 chromosomes separate from each other and form two nuclei for the two daughter cells formed from the parent cell. The cell then divides in two to produce the two new daughter cells. Both of these cells are then in the G0 phase and the cycle continues.

The M phase is subdivided into 5 sub-phases. These are the Prophase, the Metaphase, the Anaphase, the Telaphase and finally Cytokinesis (cell division). In the Metaphase the chromosomes become shorter and more condensed. The appearance of chromosomes at the Metaphase have a recognisable form characterised by two rod-like structures which are attached to each other at a specific point called the "centromere". The two halves are called "chromatids". This is the point at which we are interested in capturing the chromosomes so that any abnormalities may be seen under the microscope at high magnification.

2.1.2 Formation of Chromosome Spreads

Chromosome spreads are formed from a particular type of white blood cell. There are two primary categories of white blood cells, the B cells and the T cells. We are interested in using the T cells to produce the chromosome spreads. These T cells are called "lymphocytes".

Many chromosome aberrations are lost after the first cell division because of mechanical problems in the cell division. Also disconnected chromosome fragments may be lost after the first cell division. As time goes on the number of detectable chromosome aberrations decreases with time (Bender et al., 1988). Therefore, it is important to form the chromosome spreads during the first cell division after exposure to ionising radiation or other mutation causing agents.

To produce chromosome spreads several steps are required. First a blood sample is taken from a patient and put into test tubes. The test tubes are then spun in a centrifuge to allow the heavier red blood cells to settle to the bottom and the lighter white blood cells to float to the top. The white blood cells are skimmed off the top of the test tubes and put into a new test tube. Next an extract from beans called "phytohemagglutinon" or PHA is added to the white blood cells to cause them to reproduce. This causes the cells to enter the G1 phase and then the S phase in which the chromosomes duplicate.

The white blood cell sample is then put into an incubator at 37 degrees Celsius to allow the cells to grow and begin dividing for a period of 48 hours (Bender et al.). At this point most of the cells have entered their first metaphase. A poisonous chemical is then added to the cell sample which prevents the cells from dividing any further. The cell sample is then spun in a centrifuge. This further separates the white blood cells from the remaining blood material. The white blood cells form a white pellet at the bottom of the test tube and this pellet is then extracted and re-suspended in a new test tube. A chemical which causes the cells to swell in size is then added which makes the cell membrane tight. The white blood cells are then dropped onto a microscope slide from approximately one meter and the cells membranes rupture and the chromosomes spill onto the slide. The slide is then heated briefly with a Bunsen burner to spread out the chromosomes to form the chromosome spreads. The slide is given time to dry and then a

chemical die is added to the slide to make the chromosomes and other debris visible. Finally, a cover glass is attached to the top of the slide.

The process of forming chromosome spreads is an art and requires a lot of experience to produce good spreads consistently. Additionally, the health of the blood cells in the original sample will have an effect on how successfully chromosome spreads can be formed.

Once the slide containing the chromosomes has been produced, it is then put under the microscope and the process of finding chromosome spreads begins.

2.1.3 Categorisation of Chromosome Spreads

Chromosome spreads can be put into several categories. A good chromosome spread consists of 46 chromosomes grouped in proximity to each other such that they do not overlap. The first measure of a good chromosome spread is related to its inter-chromosome distance. The inter-chromosome distance is the distance between individual chromosomes. If the distance is too small then some chromosomes may overlap each other. Conversely, if the inter-chromosome distances are too large then the chromosomes will be so far apart that not all of the chromosomes can be seen in one field of view under high magnification. Another problem with the inter-chromosome distance being too large is that it may become unclear if all the chromosomes are accounted for. Also, if the chromosomes from two or more spreads become intermingled this can cause difficulty in sorting out which chromosome belongs to which spread.

The next measure of a good chromosome spread is related to its intra-chromosome distance. The intra-chromosome distance is related to whether the "arms" of the individual chromosomes overlap. If the "arms" of a chromosome overlap then it becomes more difficult to see any possible aberrations in that chromosome. The intra-chromosome distances are normally only seen under high magnification of x1000 or greater. See Figure 2.2 for an illustration of good and poor intra-chromosome distances (or spreads).

Intra-Chromosome Spreads

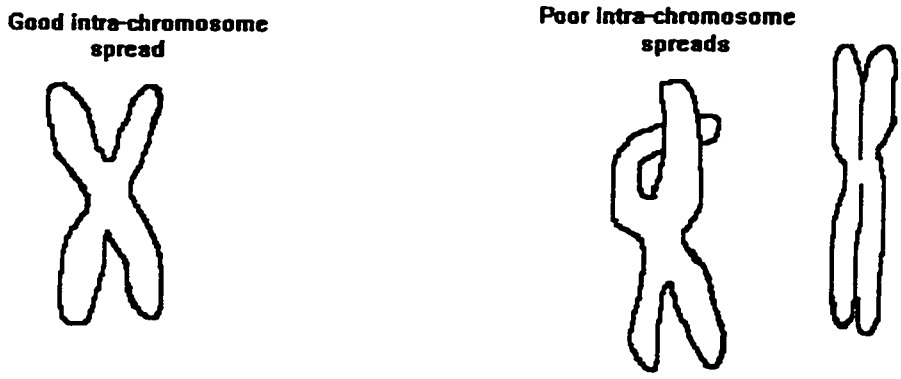


Figure 2.2: Examples of good and poor intra-chromosome spreads.

When the chromosome spread is formed a dye is added to the slide which causes the chromosomes to become dark in colour. Often there is a tendency for the dye to cause streaks and there is also a possibility that not enough dye could have been added to the slide. If this happens then the contrast between the chromosome and the background may be poor. Again this will make the analysis of chromosome aberrations more difficult.

A good chromosome spread will have good inter-chromosome distance, good intra-chromosome distance and the contrast between the chromosomes and their background will be high.

The obstacles to finding good chromosome spreads will now be discussed. Once the slide is produced and put under the microscope the lighting level must be adjusted with care so that the digital camera will be able to detect the chromosome spreads. This is done by locating a good chromosome spread manually under the microscope, putting it into focus, and then looking at the same chromosome spread on the video monitor. The light illumination is adjusted so that the chromosome spread stands out well against its background.

Another obstacle that can appear is that the staining of the slide may not be even from one side of the slide to the other. If the slide becomes darker from left to right for example, then it will be more difficult for the chromosome on the right to be detected. This is not usually a problem on most slides that have been examined thus far.

The final obstacle to locating good chromosome spreads is caused when debris overlaps part or all of the chromosomes in the spread. This debris could be red blood cells or fragments from the membrane of the white blood cells that have been deposited over some of the chromosomes. This prevents the chromosomes from being seen well for aberration analysis.

Once the chromosome spreads have been located by the computer, a categorisation system is used to separate good spreads from poor ones. There are five types of chromosome spreads that can be detected by the computer:

1. False Positive: This is where the computer misclassifies some debris as a chromosome spread. This seems to happen when there are pieces of small debris in close proximity to each other. Pieces of debris are, for the most part, much larger than individual chromosomes.
2. True Positive (Type 1): This is a chromosome spread in which the inter-chromosome and intra-chromosome distances are good.
3. True Positive (Type 2): This is a chromosome spread in which the inter-chromosome distances are good, but the intra-chromosome distances are poor. These images are acceptable, but are much more difficult to analyse for abnormalities.
4. True Positive (Type 3): This is a chromosome spread which has poor inter-chromosome distances in which many of the chromosomes touch each other, and may or may not have good intra-chromosome distances. This type of chromosome spread is unacceptable because it is not possible to analyse it for abnormalities.
5. True Positive (Type 4): This is a chromosome spread which has poor inter-chromosome distances because the individual chromosomes are too far apart from each other and hence it is difficult to determine which chromosomes are from which cell's nucleus. This type of chromosome spread is also unacceptable.

There is a vast range of chromosome spreads that must be analysed and this increases the challenge in categorising them. With the magnification used to locate chromosome spreads in this thesis it is not possible to determine the quality of the intra-chromosome distances, however, the inter-chromosome distances can be taken into account when looking for the chromosome spreads.

2.1.4 Chromosome Aberrations

When an analyst is looking at a chromosome spread he or she will try to determine if there are any chromosome aberrations present. Aberrations are indicators of cell damage that has taken place. When someone is looking for chromosome aberrations a high magnification is used so that the individual chromosome features are easily seen. In the automated process of locating chromosome spreads the magnification used is x200. However, to differentiate between individual chromosomes and to see aberrations in them a x1000 magnification is used.

As stated above one cause of chromosome aberrations is radiation damage. There are two primary types of (ionising) radiation, Low-LET (Linear Energy Transfer) radiation and High-LET radiation. The Low-LET radiation are the x-ray and gamma-ray types and the High-LET radiation are fission neutrons emitted from the decay of Uranium or other highly radioactive elements. (Bender et al.)

The Low-LET type of radiation produces chromosome aberrations according to the following formula:

$Y = c + aD + bD^2$,

where Y is the yield of chromosome aberrations, D is the radiation dose in Rads and c is the baseline chromosome aberrations present in an average person. The coefficients a and b are derived from fitting the curve to the actual data obtained from patients.

The reason that this equation is a quadratic is because there is an effect of multiple hits by radioactive particles on one area of a chromosome. The Low-LET radiation particles penetrate deeply into living matter, but they have a low probability of causing a break in a chromosome.

When the dose is increased the probability that more than one radioactive particle will hit the same place on the same chromosome will increase and this will cause an elevated energy level transfer to the chromosome and this will increase the likelihood of a chromosome break. This produces chromosome fragments. These fragments can recombine with other broken chromosomes to cause aberrations.

In contrast High-LET radiation produces chromosome aberrations according to the following formula:

$Y_2 = c + a_2 D_2$,

where Y_2 is the yield of chromosome aberrations, D_2 is the radiation dose in Rads and c again is the baseline chromosome aberrations present in an average person. The coefficient a_2 is again derived from fitting the curve to the actual data obtained from patients.

In this case the curve is linear. High-LET radiation particles only penetrate shallowly into living matter, but have a much greater probability of hitting a chromosome and causing an aberration. Since there is no additive effect in multiple hits by High-LET radiation, the equation is linear.

If the number of Rads of Low-LET radiation is given over a longer period of time, and hence the magnitude of that dose is reduced, then less aberrations will result because the probability of multiple hits will decrease. On the other hand if the number Rads of High-LET radiation is given over a longer period of time there is not a reduction in the number of aberrations caused by the radiation. This is because there is no additive effect of multiple hits of the radioactive particles. This concludes a brief discussion of radiation damage to chromosomes.

There are many different types of aberrations, see Figure 2.3 for some examples of them.

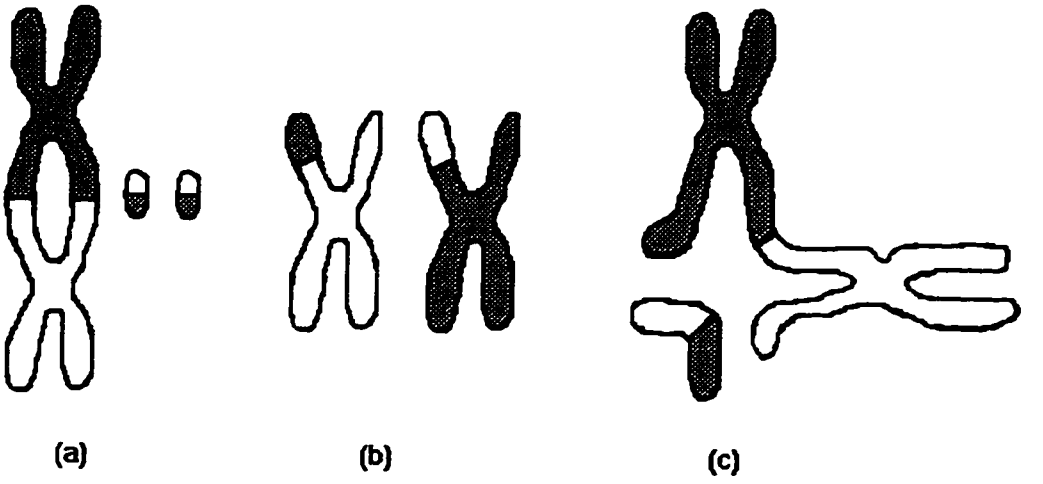


Figure 2.3: Selected chromosome aberrations.

Figure 2.3 (a) is a typical aberration in which the bottom parts of two legs from one chromosome have broken off and the top parts of the two arms from another chromosome have broken off and they have joined to each other. When this happens there will always be the two fragments left over. In Figure 2.3 (b) two arms have broken off two chromosomes and have been reattached to different chromosomes. Unfortunately, this type of aberration can not be detected from a monochrome image. If the time and expense of the procedure is affordable, this type of aberration could be detected with the use of colour marking. In colour marking each different chromosome has a different colour. It is unlikely that this procedure will be used in the near future, because it is much more expensive and difficult to produce colourised slides compared to producing monochrome slides. Another technique is available which produces banding patterns on the chromosomes which are distinct and hence if abnormal banding patterns appear then chromosome aberrations are apparent.

Finally in Figure 2.3 (c) another type of aberration is shown which involves two chromosomes. The legs of the two chromosomes have broken off and they have joined at the leg.

There is also a fragment produced which is the attachment of the two broken off parts from the legs. There are also many other aberrations that are not shown, in particular another common type of aberration is in the form of interconnected rings.

In a potential aberration algorithm the most common aberrations will be sought, and any possible aberrations will be marked on the computer image for closer inspection and confirmation by the user of the software.

The automatic detection of these aberrations is, in itself, a very difficult task. It has not, at the present time, been adequately addressed, even after three decades of research on the topic. The problem has many ramifications, in image processing, image representation, and artificial intelligence. At the present time, the semantics of the problem are too complex. We considered it was more effective to assist the human eye in the simple, repetitive, and tedious aspects of the task than to attempt once more, to replace it.

2.2 Image Characteristics

The images that are captured by the grey scale camera will now be described. The camera is an eight-bit charge coupled device (C.C.D.) monochrome camera. These images are fed into the computer and are then captured, that is to say that they are frozen for the purpose of analysis. The captured images are of size 640 by 480 pixels using one byte for each pixel to produce an image depth of 256 grey levels. The value 0 produces a totally black pixel and the value 255 produces a totally white pixel.

At our magnification power of x200 the chromosomes are small in size and are about 2 to 3 pixels in width. Chromosomes are dark grey or black in colour on a light grey background. Debris, which are not to be detected by the program, are similarly dark grey or black in colour. There are two types of debris. The first are comparable in size to chromosomes, but are isolated in the image and hence will not be treated as part of a chromosome spread. The second are considerably larger in size and so will be eliminated from the image. The background for all image elements is usually a light grey colour.

A valid chromosome spread is circular in shape and has a diameter ranging from 25 to 40 μm. This diameter measurement in pixel values is from approximately 70 to 90 pixels. An individual chromosome has a length of 3 to 7 μm and a width of 1 μm to 3 μm. These measurements in terms of pixel values are a length of 7 to 12 pixels and a width of 2 to 4 pixels. The inter-chromosome distances can be in the range of 1 μm to 10 μm. Again this measurement in pixel values is from approximately 2 to 20 pixels. As stated above the image depth of the field is 256 grey levels. The chromosome grey levels are typically in the 88 to 112 range in intensity, compared to the background grey levels which are in the range of 208 to 224 in intensity. There are about 4000 fields of view on a typical slide (Piper et al.). See Image 2.1 for an example of a typical good quality chromosome spread.

Image 2.1: (TKS1F65.BMP) A typical well formed chromosome spread.

Chapter 3

Literature Survey

In this chapter two other systems that have been developed to find chromosome spreads will be described. The immediate predecessor to the chromosome detection system produced for this thesis was designed and built by Dr. Frank Johnson for and with Dr. Jack McLean. The name of the paper describing Dr. Johnson's and Dr. McLean's system is: "Evaluation of a Metaphase Chromosome Finder - Potential Application to Chromosome-based Radiation Dosimetry" The chromosome detection system of this thesis is called a second generation system because is comes after the above system, and uses new techniques to accomplish the same goals.

This previous chromosome detection system was developed on an Intel 80486-33MHz IBM compatible computer. It has a similar hardware set up as the system of this thesis. That is it has an autofocus component, a platform movement component as well as a chromosome detection component. The mechanism by which the chromosome detection component operates is quite different from the operation of the system of this thesis.

The chromosome detection component of this system will now be described. The field of view has a resolution of 640 by 480 pixels. Each pixel is 0.2 μm on a side and has an area of 0.04 μm^2. This field of view is then divided up into 1200 16 pixel by 16 pixel squares. For any particular square in the field of view there are three types of picture elements that it could contain. The first is that it could contain nothing but the background which is a light grey colour. The second thing that it could contain is debris. Lastly, the third thing that it could contain are some

chromosomes. It is also possible for the 16x16 square to contain a combination of the above three image elements.

The system calculates the sum of the second difference of all adjacent pixels in the square and then compares the result to a predetermined threshold value. If the square's second difference sum is larger than the threshold value then that square is flagged as possibly belonging to a chromosome spread.

If the 16x16 square contains primarily the background then there will be very small variations between adjacent pixels. This will result in the second difference sum being low. If the square contains debris then the second difference sum is likely to be moderate, and if the square contains primarily chromosomes, then the second difference sum should be high.

A typical spread of metaphase chromosomes is circular and has a diameter of 25 to 35 μm. This results in an area of 490 to 960 μm^2. This is equivalent to 48 to 94 contiguous 16x16 squares. Once all of the 1200 squares have been compared to the threshold value they are clumped into contiguous groups. The groups should be fairly circular. If a resulting group is of the desired size and shape then the contiguous group is flagged by the computer as a True Positive (TP). If TPs are not found in the field of view then the field of view is labelled as a True Negative (TN).

As with the chromosome detection system of this thesis, a Receiver Operating Characteristic (R.O.C.) curve was used to evaluate the performance of this system. A human observer must check for False Positives (FP), that is where the computer has incorrectly labelled some debris as a chromosome spread, and for False Negatives (FN), that is where the computer has missed a valid chromosome spread. A FP is classified as a Type I error, and a FN is classified as a Type II error.

The True Positive Rate is the number of detected True Positives divided by the actual number of True positives. Also the False positive Rate is the number of detected True Negatives divided by the actual number of True Negatives.

The performance of the system using an optimum threshold was a TP rate of 0.74 to 0.71 and a FP rate of 0.06 to 0.025 per frame. The FN rate is estimated at 0.25 to 0.29.

The data set used to evaluate this system consisted of 100 fields of view containing 26 metaphase spreads. Therefore the probability of a frame containing a metaphase spread was 0.26.

According to the authors the FNF, which is 1-TPF (the probability of committing a Type II error), was unacceptably high at 0.25 to 0.29. However, the average accuracy of their metaphase finder was described as follows:

Average Accuracy = TPF * P(D+) + TNF * P(D-) = 0.6 * 0.235 + 0.984 * 0.765 = 0.894,

where P(D+) is the probability of a True Positive being detected and P(D-) is the probability of a True Negative being detected. This was an acceptable result according to the authors.

To compare this result to the chromosome detection system described in this thesis the best Average Accuracy result was obtained from the R.O.C. curve in Figure 6.4 at the point (0.0312, 0.6818) which will be more fully described in the Results Section 6.4. This result produced an Average Accuracy of the following:

Average Accuracy = TPF * P(D+) + TNF * P(D-) = 0.6818 * 0.2558 + 0.9688 * 0.7441 = 0.895,

again where P(D+) is the probability of a True Positive being detected and P(D-) is the probability of a True Negative being detected. This result is also acceptable and is very similar to the above result. The advantage of the chromosome detection system described in this thesis is that a much higher True Positive yield can be obtained of 84.6% with only a 13.95% False Positive percentage as can be seen in the R.O.C. curve in Figure 6.7 which again will be discussed later in Section 6.4. This means that fewer slides need to be scanned to obtain a desired amount of chromosome spreads.

One of the strengths of Dr. Johnson's system is that it works very quickly in processing an image. Another strength that it has is that it allows the user to determine what level of Type I error and what level of Type II error are acceptable within its range. A problem with the above system is that is has a higher Type II error rate than was hoped. Additionally, the probability of getting a FP is moderately high on a per frame basis.

In order to compare the metaphase chromosome spread finder of the above system to the system described in this thesis, a normalisation must be performed on the data sets used for both systems.

In the chromosome detection system produced for this thesis, two sets of 100 images were used to produce ROC curves. The number of TP in the first set of 100 images was 44 and the number of TN was 128. There may be more than one TN on a given frame. This would take place if more than one unacceptable metaphase spread was present in the frame. The number of TP in the second set of 100 images was 39 and the number of TN was 86.

We will now look at a commercially available system called the Cytoscan 110 metaphase finder. This system is described by P. Finnon, et al. 1986. In their paper the system is described as being very fast in that it can scan an entire slide in under two minutes. This system scans continuously the whole slide from one end to the other without the need to process each of the 4000 individual frames. For a chromosome detection system such as described in this thesis, even with a very fast autofocusing algorithm that takes just one second to complete, it would take 1.1 hours just to focus on each image. If only one in five images were to be focused it would still take 13.3 minutes to just focus the images. The Cytoscan 110 uses a hierarchical approach on a very expensive computer system. The system uses parallel co-processors and cost more than $100,000.00. No R.O.C. curves were produced, however, the authors state that more chromosome spreads are found by the system than by a human operator. The magnification used was x200.

Four slides of chromosome (metaphase) spreads were used to test the system. Two had a densely distributed metaphase spreads and two had sparsely distributed metaphase spreads. Two different staining methods were used on the slides one type for each type of density. The system categorises the found chromosome spreads into 4 rankings. The highest quality chromosome spreads are categorised into rank 4, the next highest quality chromosome spreads are categorised into rank 3, and so on. The authors state that if the system is commanded to present the user with chromosome spreads first of quality level 4 and then quality level 3, then 75% of the undesirable

objects will be eliminated. The penalty of this is that 30% of good quality chromosome spreads would also be eliminated. This would indicate that the Cytoscan 110 produces a True Positive rate of 70% and a False Positive rate of about 25%.

The speed advantage of the Cytoscan 110 is very impressive compared to the system described in this thesis, but the ability of the system to find good quality chromosome spreads with as few as possible False Positives is a matter of concern. Additionally, the cost of the Cytoscan 110 could be prohibitive to some laboratories.

There have been many attempts to detect chromosome spreads as well as to perform banding pattern feature detection and analysis. (Piper 1980, Piper 1986, Piper 1987, van Vliet 1990, Mayall 1990). The last two of these papers describes the same system for karyotyping. This system has demonstrated some success in analysis, however, it relied on the manual detection of the chromosome spreads. It is with this background that this investigation of a method to automatically detect chromosome spreads was started.

Chapter 4

Morphological Operations

4.1 Dilations and Erosions

In this section three of the basic image processing functions will be discussed. The first is the dilation function, the second is the erosion function, and the third is the inversion function. These operations are the basis of morphological image processing. The essentials of the theory can be found in Serra [Image Processing and Mathematical Morphology I], [Image Processing and Mathematical Morphology II] and Doughecty [An introduction to Mathematical Morphology].

A source image is an array of 640 by 480 pixels. These pixels can have a grey scale of 0 to 255 inclusive. The value 0 represents total black and the value 255 represents total white. In the functioning of the algorithm to detect metaphase spreads described in this thesis, the images used are thresholded images. Hence, these images are binary images which means that they are composed of only black (0) and white (255) pixels. Therefore, in our discussion of dilations and erosions we will only consider the binary versions of these functions.

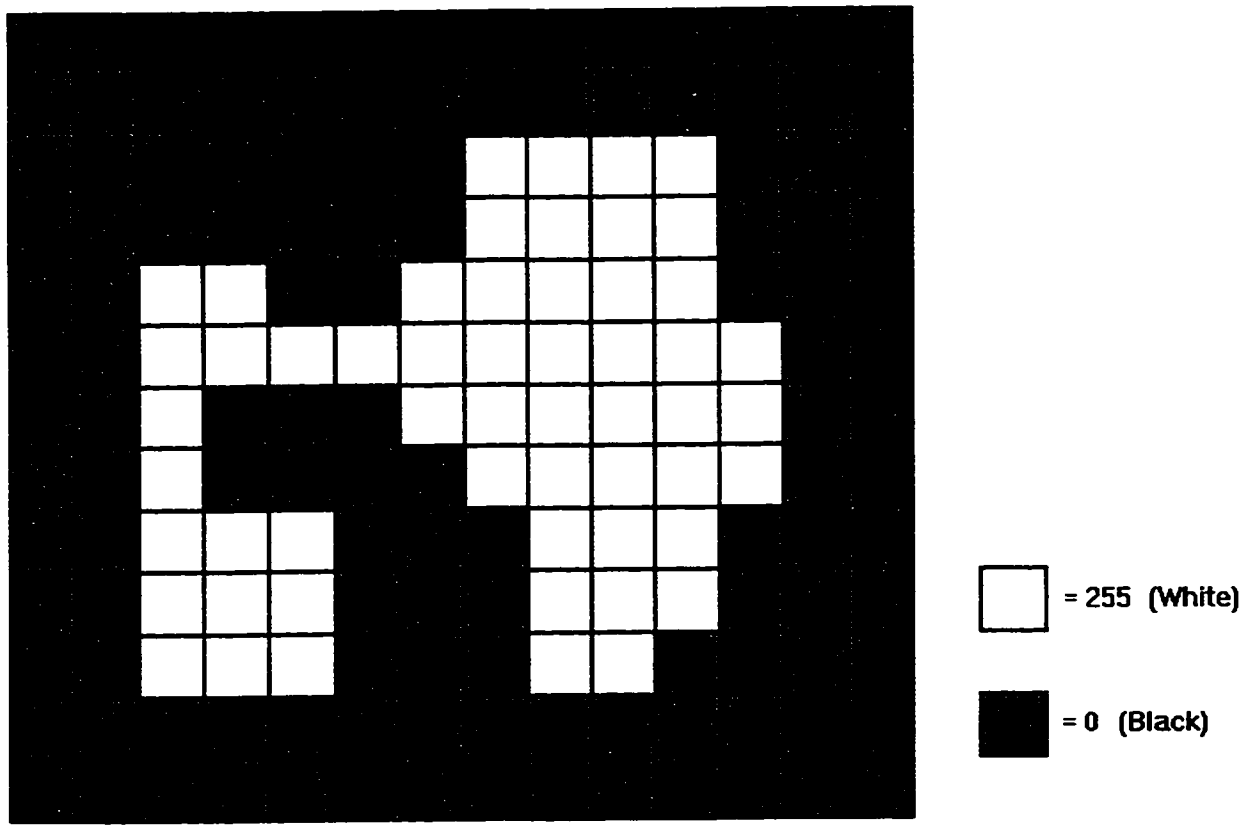
Definitions

Structuring Element

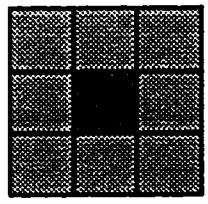
A structuring element is a set of points which generate a shape and an origin which is usually positioned in the geometrical centre of the set. In Figure 4.1 (b) the structuring element is a square. many other shapes can be used such as circles, oriented lines, diamonds, and so on. The

size of the structuring element can also vary. In this example we have a 3x3 square structuring element.

The role of the structuring element is similar to correlation masks in digital filtering, as we will see in the following definitions.



(a) Original Image



(b) Structuring Element

Figure 4.1: Original image (a) and structuring element (b).

Dilation

The dilation is the result of the following operations:

Scan the input image with the structuring element. For each pixel:

- Compute the maximum value over the neighbourhood delineated by the structuring element.
- Output this value to the output image at the location that corresponds to the origin of the structuring element.

Figure 4.2 illustrates how this works.

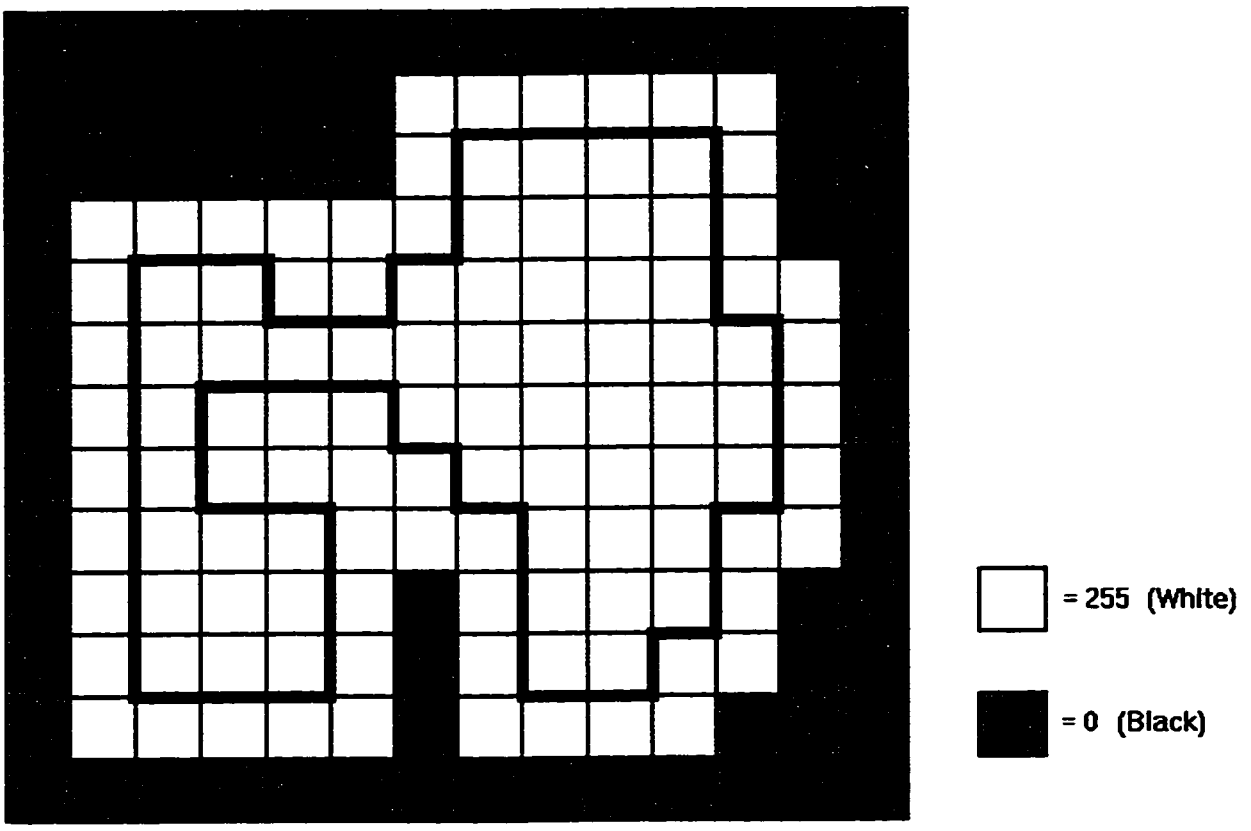


Figure 4.2: Dilated Image.

The dilation function is described by the following algorithm:

```
// B = Dilation(A)
for i = 1 to (rows - 2)
   for j = 1 to (columns - 2)
      B[i, j] = max(
         A[i - 1, j - 1],
         A[i - 1, j],
         A[i - 1, j + 1],
         A[i, j - 1],
         A[i, j],
         A[i, j + 1],
         A[i + 1, j - 1],
         A[i + 1, j],
         A[i + 1, j + 1]
      )
   next j
next i
```

Erosion

The erosion is similar to the dilation with the exception that the minimum value is computed over the neighbourhood in stead of the maximum. Figure 4.3 illustrates an erosion on the input image.

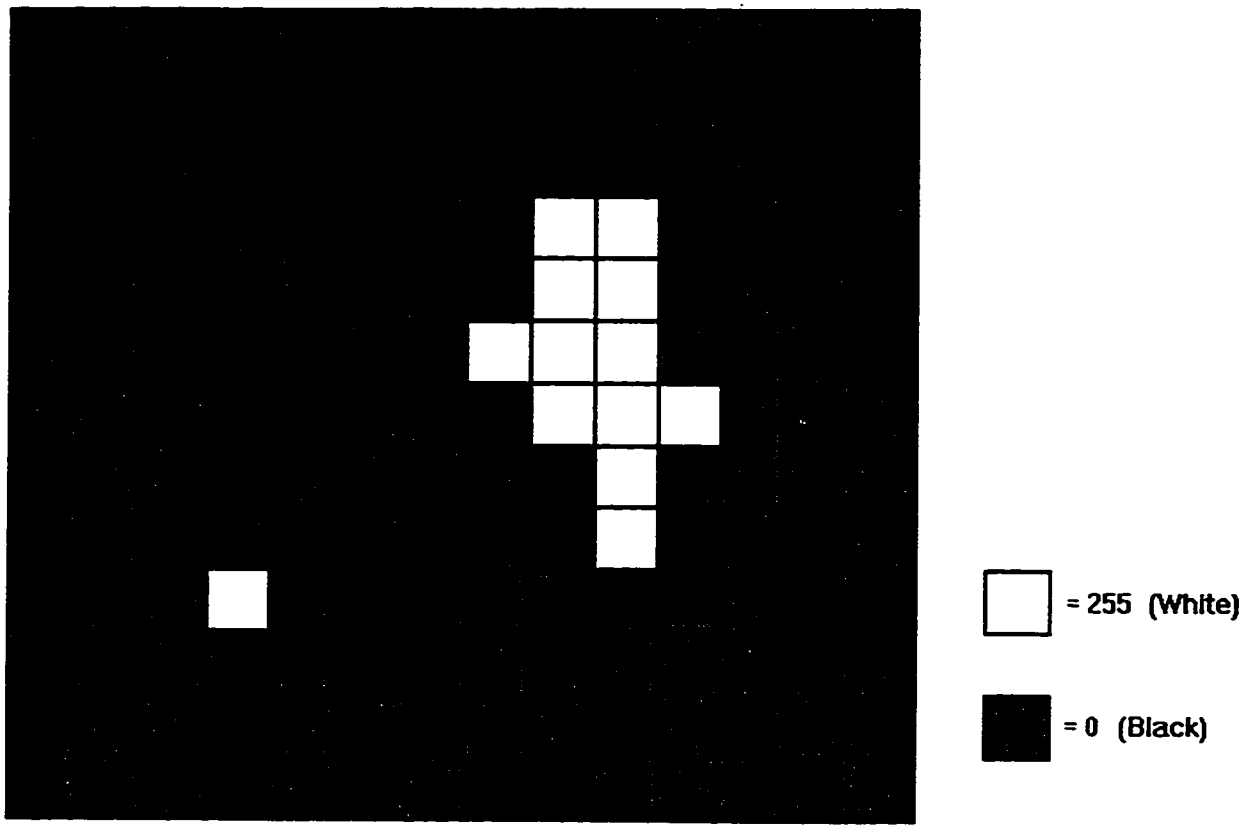


Figure 4.3 Eroded Image.

The erosion function is described by the following algorithm:

```
// B = Erosion(A)
for i = 1 to (rows - 2)
   for j = 1 to (columns - 2)
      B[i, j] = min(
         A[i - 1, j - 1],
         A[i - 1, j],
         A[i - 1, j + 1],
         A[i, j - 1],
         A[i, j],
         A[i, j + 1],
         A[i + 1, j - 1],
         A[i + 1, j],
         A[i + 1, j + 1]
      )
   next j
next i
```

Inversion

An inverted image is produced by scanning an input image and for each pixel in it the output image is set to the opposite value. In a binary image this means that if the input image contains a 0 (Black) value at a pixel then the output image will be set to 255 (White) at that pixel and vice versa. See Figure 4.4 for the inverted form of the input image contained in Figure 4.1.

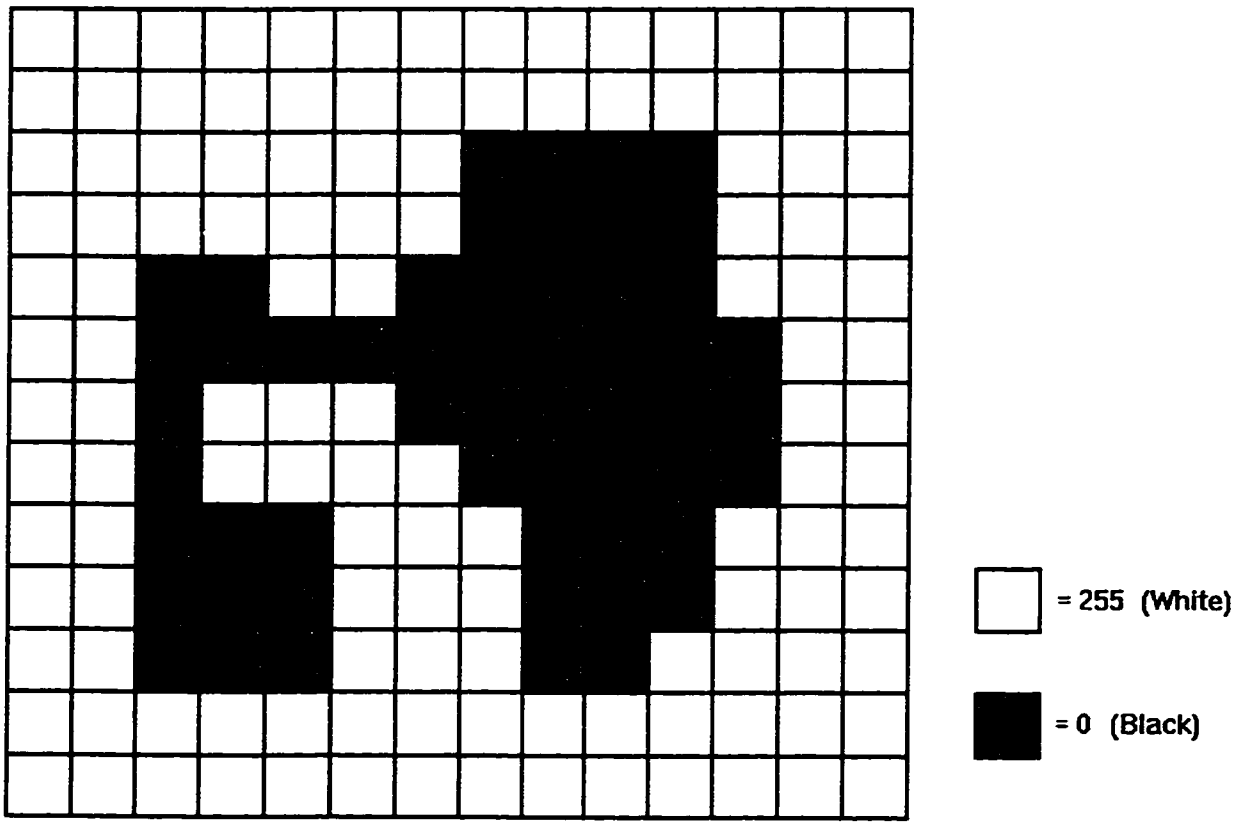


Figure 4.4 Inverted Image.

The inversion function is described by the following algorithm:

```
// B = Inversion(A)
for i = 0 to (rows - 1)
   for j = 0 to (columns - 1)
      if (A[i, j] == 0) then
         B[i, j] = 255
      else
         B[i, j] = 0
      end if
   next j
next i
```

Properties

The operation of the dilation function can be achieved by the use of inversion and erosion functions as follows:

```
B = Dilation(A)
```

Is equivalent to:

```
B = Inversion(Erosion(Inversion(A)))
```

This property is called duality. The erosion function is the dual of the dilation function. Figure 4.5 shows the erosion of the inverted image depicted in Figure 4.4.

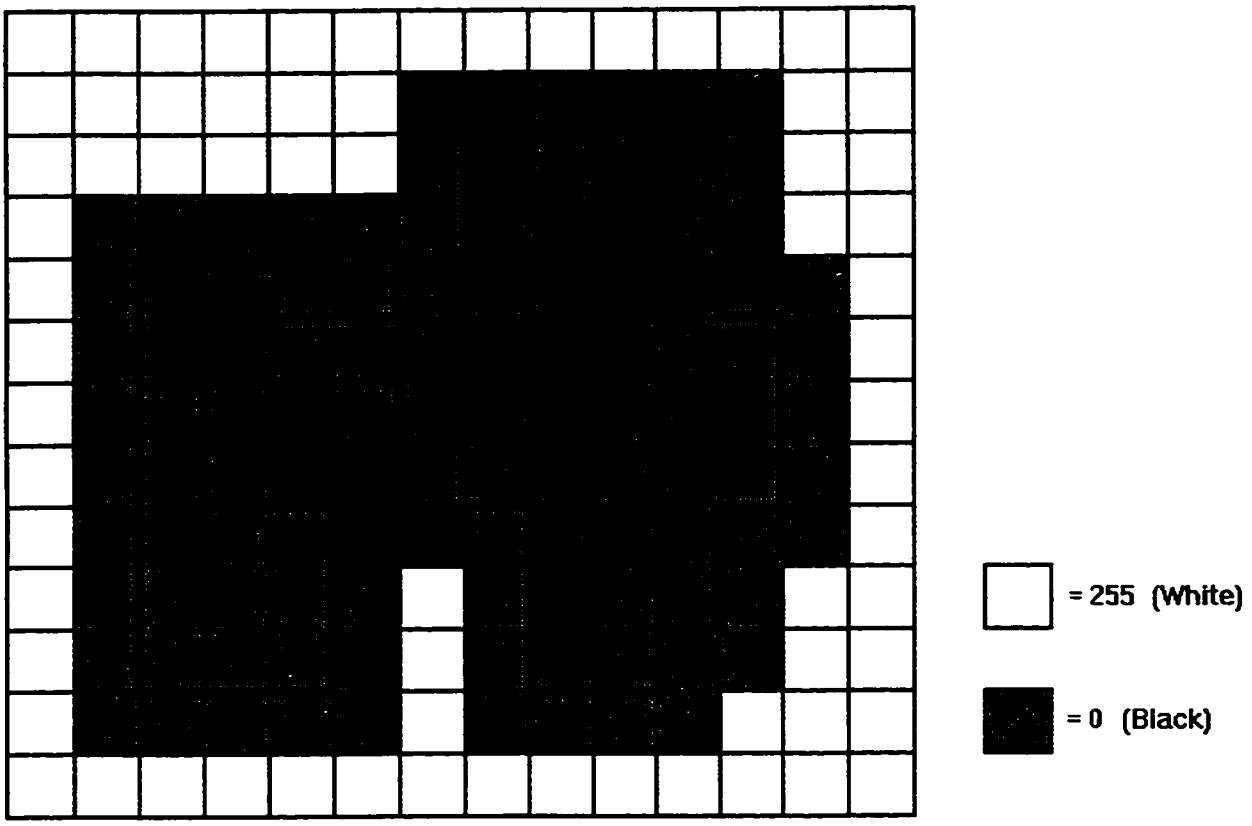


Figure 4.5 Eroded (Inverted) Image.

In a similar manner the operation of the erosion function can be achieved by the use of inversion and dilation functions as follows:

```
B = Erosion(A)
```

Is equivalent to:

```
B = Inversion(Dilation(Inversion(A)))
```

Conversely, the dilation function is the dual of the erosion function. Figure 4.6 shows the dilation of the inverted image depicted in Figure 4.4.

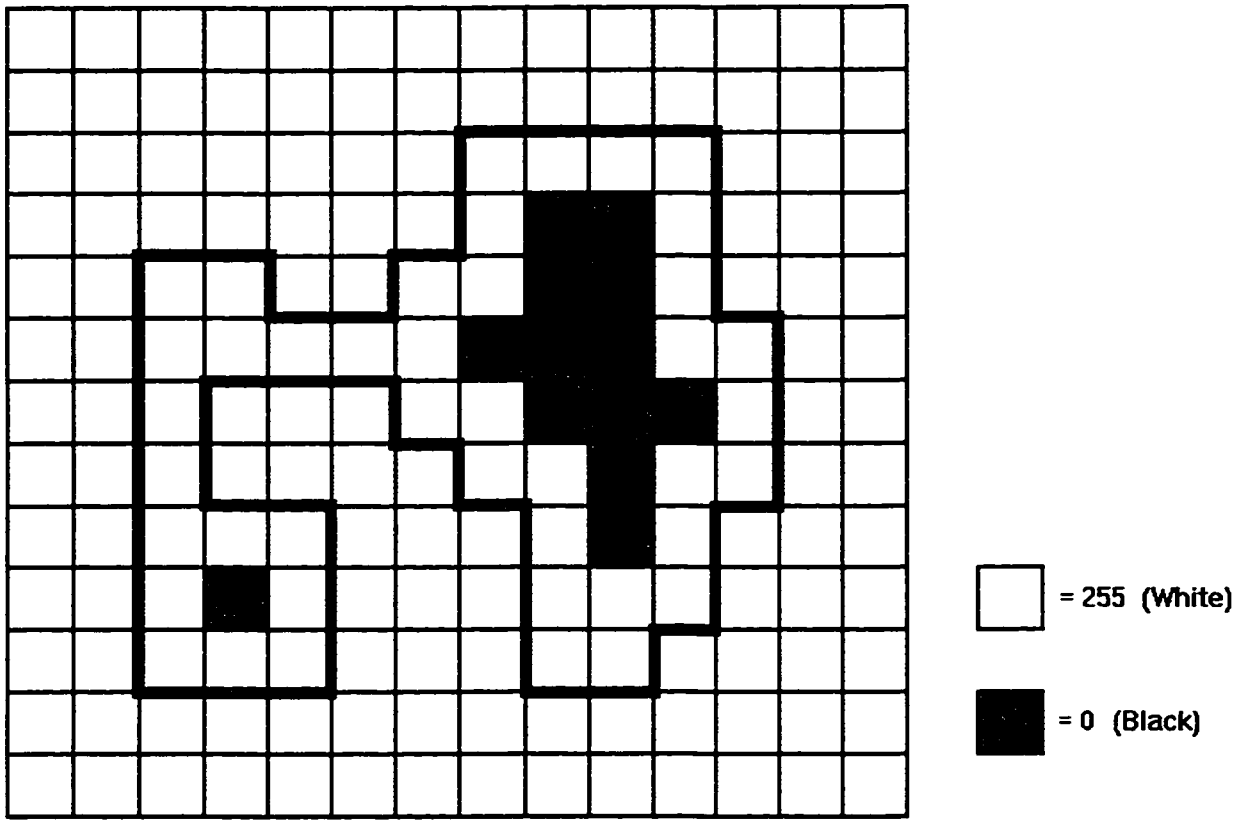


Figure 4.6 Dilated (Inverted) Image.

Dilation and erosion functions can be iterated on a given image. A second parameter can be supplied to these functions to indicate the number of times to perform the dilation or erosion. For example the following code segments have the same effect:

```
(i)
B = Dilation(Dilation(A))

(ii)
B = Dilation(A, 2)
```

When a function is performed multiple times the time to complete the task is likewise multiplied. When a dilation or erosion is performed multiple times with a small structuring element, this is equivalent to performing a single erosion or dilation with a large structuring element.

For instance, dilating an image twice with a 3x3 structuring element is the same as using a 5x5 structuring element once.

It is in general more efficient to implement a large structuring element by iteration, then by applying the structuring element directly.

The size of a dilation or erosion is the radius of the structuring element expressed in pixels. A 3x3 structuring element has a radius of 1, and 5x5 structuring element has a radius of 2 and so on.

This being said, large dilations or erosions such as those used in this thesis take considerable time. We used the distance function in order to implement a fast dilation function and erosion function.

4.2 Fast Dilations and Erosions using the Distance Function

Distance function

The distance function uses a binary input image to produce a grey scale output image. Each pixel each pixel in the output image represents the distance between that pixel and the boundary of the picture element that it belongs to. This distance can be the city block distance (4-connected) or the chessboard distance (8-connected).

If a pixel in the binary image belongs to the background (set to zero), then the output image will have this pixel also set to zero. The output from the distance function is also the superposition of all the eroded images of the binary input image. Figure 4.7 illustrated this concept.

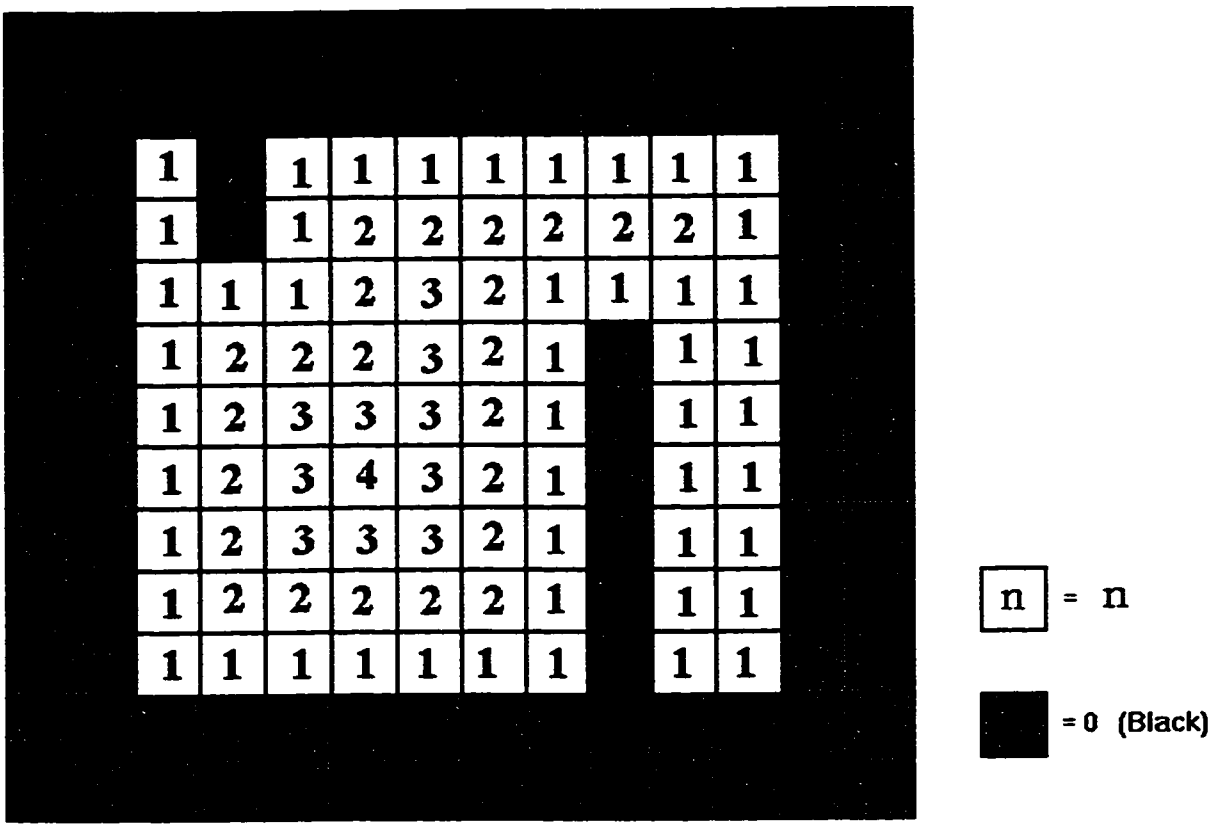


Figure 4.7 Original Image #2.

Once the distance function has produced the output image an erosion of any size can be accomplished by simply performing a threshold of the image with a value set to the size of the erosion wanted plus one. This is because the output image is the superposition of all possible erosions for that image.

The distance function needs to perform two scans of the input image in order to produce the output image. This speed is a great advantage over the iterated erosion function when the structuring element is large.

The implementation of the fast erosion function using the distance function is as follows:

- Binarize the image: 0 stays 0, and 255 becomes 1

- Create the distance function

- Threshold the image produced by the distance function. For instance, an erosion of size 15 needs a threshold equal to 15 + 1 = 16. The number of scans needed is 4. An iterated erosion of size 15 would need 15 scans.

The fast distance function is implemented as follows: The first scan goes from the top left to the bottom right of the image from left to right and in a top to bottom row by row manner. During this scan if the value of the pixel is greater than zero then the new value of the pixel will be one plus the minimum of the following four pixels: the pixel to the top right, the pixel to the top, the pixel to the top left, and the pixel to the left.

The second scan goes from the bottom left to the top right of the image from right to left and in a bottom to top row by row manner. During this scan if the value of the pixel is greater than zero then the new value of the pixel will be the minimum of the value of the pixel and one plus the minimum of the following four pixels: the pixel to the bottom left, the pixel to the bottom, the pixel to the bottom right, and the pixel to the right. The above procedure correctly assigns the distance values to the pixels and only requires two scans of the image.

In the example above from Figure 4.7 an erosion of size 2 can be seen in Figure 4.8.

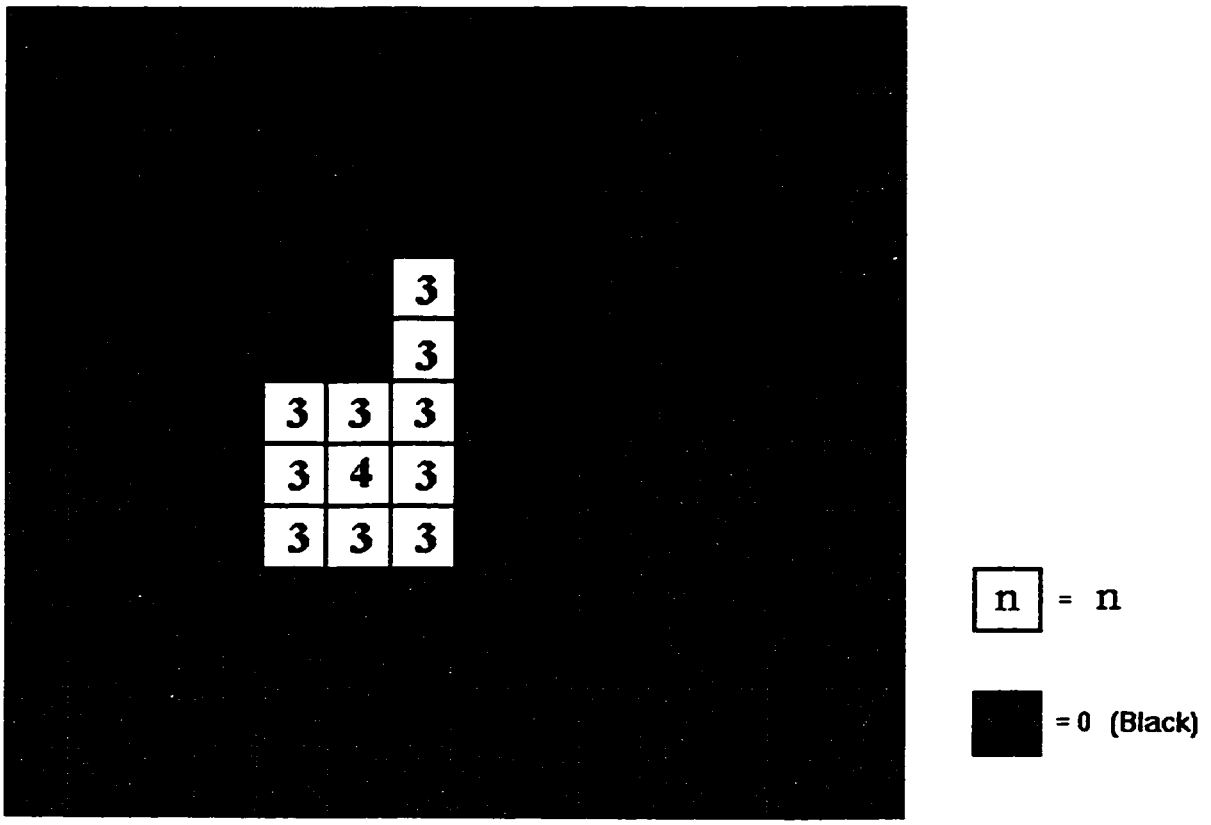


Figure 4.8 Original Image #2 Eroded by size 2
Using the Distance Function.

An erosion of size three is depicted in Figure 4.9.

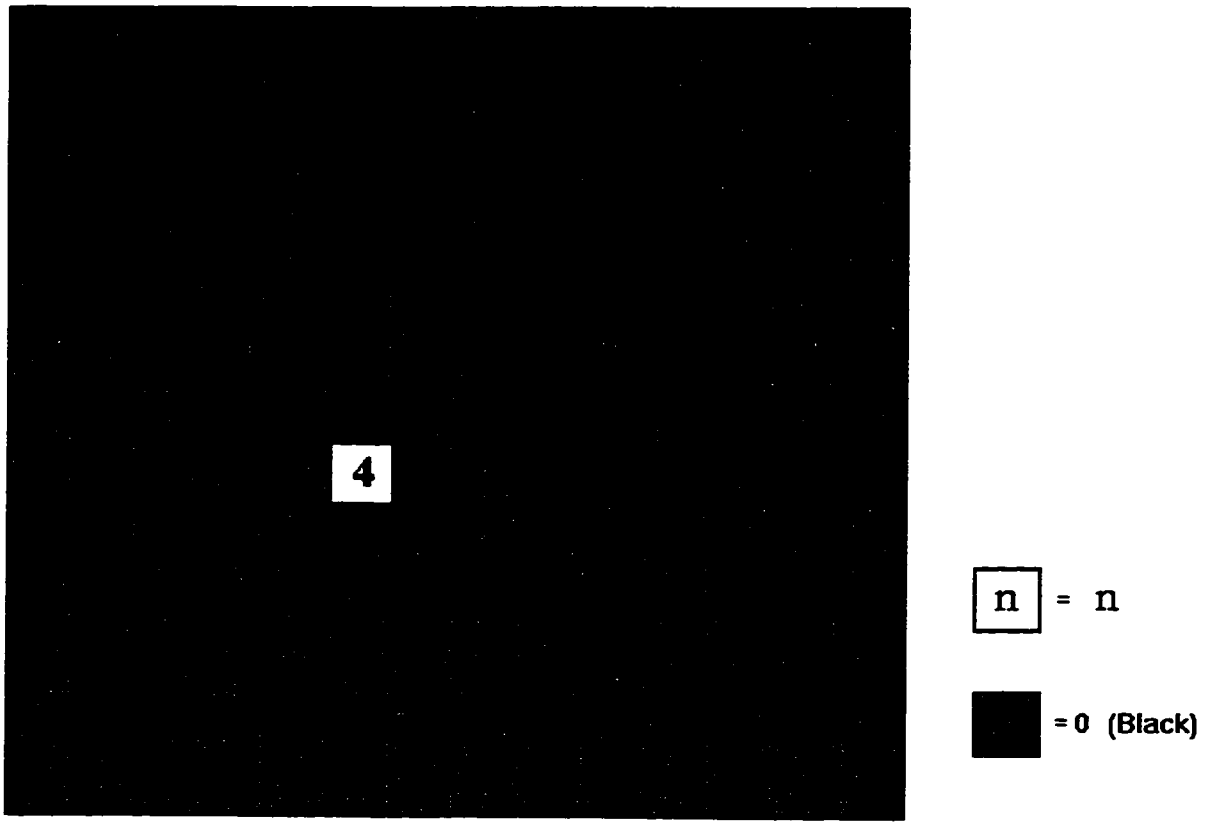


Figure 4.9 Original Image #2 Eroded by size 3
Using the Distance Function.

The erosion by the distance function is described by the following algorithm:

```
// B = ErosionByDistanceFunction(A, givenValue)
//
// B = Threshold(A, "If A(i, j) is greater then 0 then set it to
1.")
for i = 0 to (rows - 1)
   for j = 0 to (columns - 1)
      if (A[i, j] == 0) then
         B[i, j] = 0
      else
         B[i, j] = 1
      end if
   next j
next i
```

```
// Forward scan.
for i = 1 to (rows - 2)
   for j = 1 to (columns - 2)
      B[i, j] = 1 + min(
         B[i - 1, j + 1], // upRight
         B[i - 1, j],     // up
         B[i - 1, j - 1], // upLeft
         B[i, j - 1]      // left
      )
   next j
next i
// Backward scan.
for i = (rows - 2) to 1 step -1
   for j = (columns - 2) to 1 step -1
      B[i, j] = min(B[i, j], 1 + min(
         B[i + 1, j - 1], // downLeft
         B[i + 1, j],     // down
         B[i + 1, j + 1], // downRight
         B[i, j + 1]      // right
      ))
   next j
next i
// B = Threshold(B, "If A(i, j) is greater then givenValue then
set it to 255, otherwise set to 0")
for i = 0 to (rows - 1)
   for j = 0 to (columns - 1)
      if (B[i, j] > givenValue) then
         B[i, j] = 255
      else
         B[i, j] = 0
      end if
   next j
next i
```

As described in section 4.1 the dilation function is simply the dual of the erosion function. Hence to perform a dilation of any size using the distance function first invert the input image, perform the dilation, and then re-invert the output image. This takes 4 scans of the image as well because the thresholding and the inversion can be done in one step simply by changing what values are assigned to the output image during the threshold function.

4.3 Openings and Closings

The simpler image processing functions described in the previous sections can be combined to perform more complex functions. The two image processing functions that will described in this section are openings and closings.

Definitions

An opening is an erosion followed by a dilation of the same size. (Serra) This function has the effect of removing small isolated white picture elements from an image with a background. This operation also has the effect of smoothing the edges of larger white picture elements that have jagged edges. This function is called an "opening" because it opens up gaps between two larger picture elements that are connected by a smaller joining section.

See Figure 4.10 for an example of an opening of size 1 of the original image in Figure 4.1.

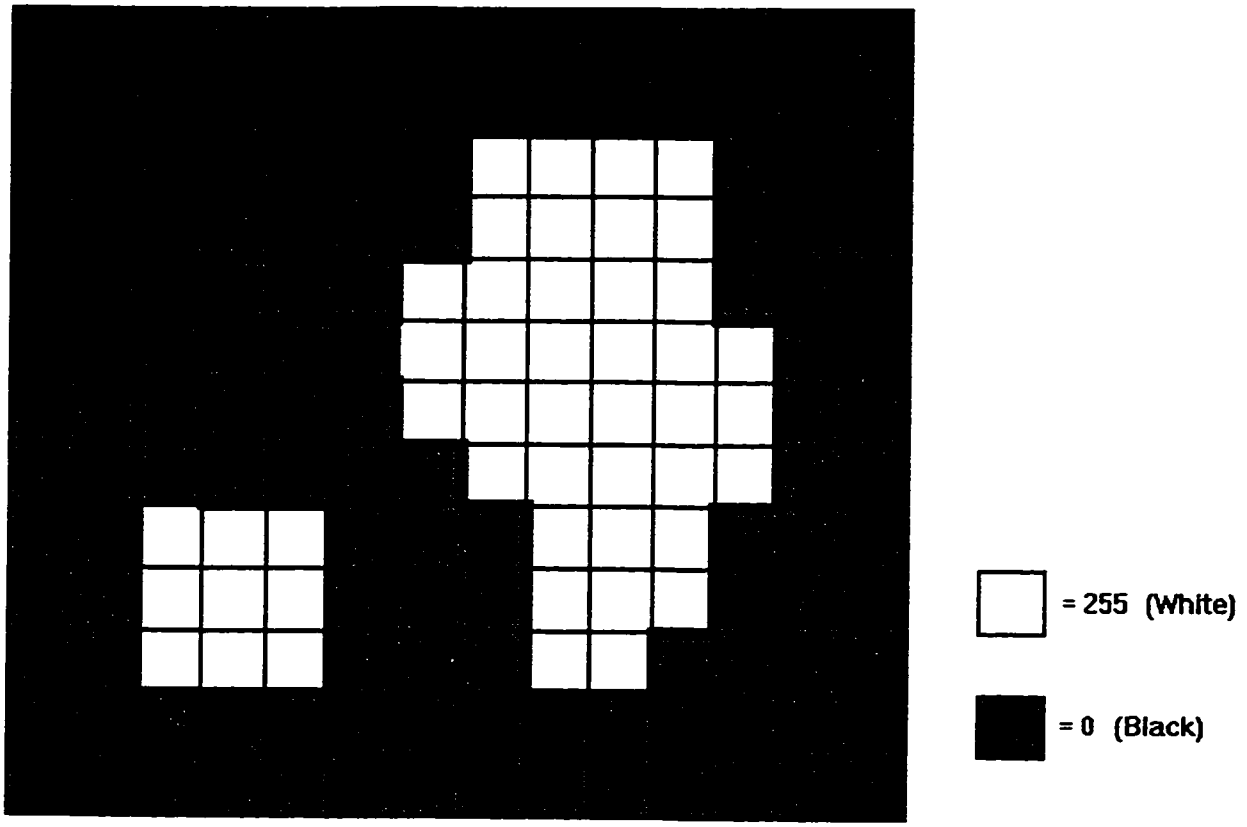


Figure 4.10 Opening of size 1 of the original image in Figure 4.1.

The opening function is described by the following algorithm:

```
// B = Opening(A, givenValue)
B = Erosion(A, givenValue)
B = Dilation(B, givenValue)
```

A closing is a dilation followed by an erosion of the same size. (Serra) This function has the effect of filling in small holes that are smaller than the structuring element. This function also has the effect of smoothing the edges of larger picture elements that have small cracks on their edges. This function is called a "closing" because it closes up small black holes inside larger white picture elements.

See Figure 4.11 for an example of a closing of size 1 of the original image in Figure 4.1.

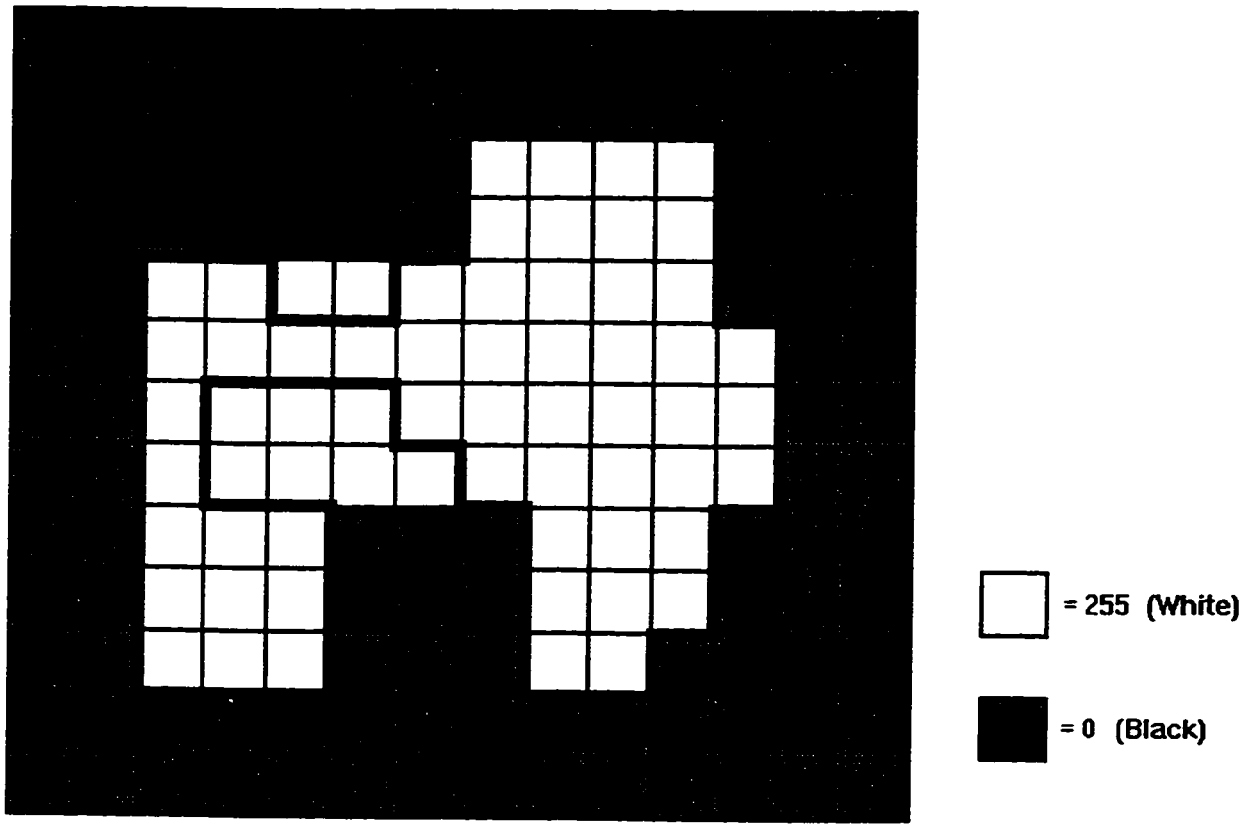


Figure 4.11 Closing of size 1 of the original image in Figure 4.1.

The closing function is described by the following algorithm:

```
// B = Closing(A, givenValue)
B = Dilation(A, givenValue)
B = Erosion(B, givenValue)
```

Duality Properties

As was the case for the duality of dilations and erosions, the dual function of an opening is a closing. Hence the following two code segments accomplish the same effect:

```
B = Opening(A, givenValue)
```

Is equivalent to:

```
B = Inversion(Closing(Inversion(A), givenValue))
```

Also the closing function can be obtained from the opening function and inversion functions by the following code segments:

```
B = Closing(A, givenValue)
```

Is equivalent to:

```
B = Inversion(Opening(Inversion(A), givenValue))
```

This concluded the discussion on Morphological image processing functions.

Chapter 5

System Description

5.1 Experimental Set-up

To implement the autofocus algorithm we used the following experimental set-up. The language of development was ANSI C/C++. The development platform was WIN16. The video card was a Matrox Comet video card. The Libraries used were the standard C Libraries and the Matrox Imaging Library (MIL). The time constraints for obtaining a focused image was one second or less. A motion control card was used to co-ordinate three stepper motors which control the x, y, and z movements of the chromosome slide with respect to the microscope lens.

5.2 Hardware

The computer hardware configuration will now be described for the image autofocusing, acquisition system and chromosome finder program. Refer to Figure 5.1. The chromosome spreads reside on a chromosome slide as described in section 2.1.2 above. The chromosome slide is manually mounted on the platform. Two stepper motors were used to control the x and y movements of the microscope platform. A third stepper motor was used to control the z axis for the focusing of the microscope lens.

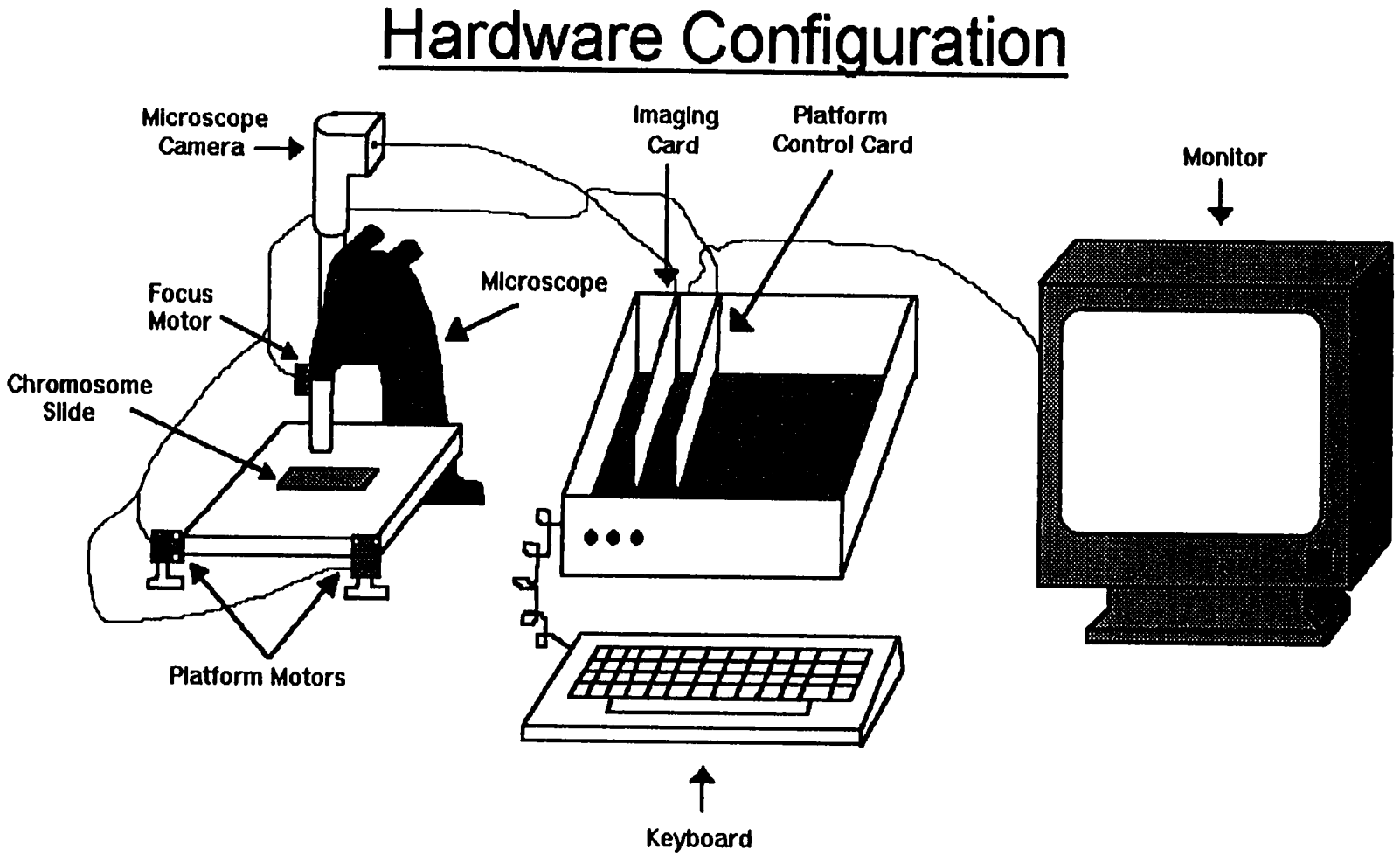


Figure 5.1 Chromosome detection Hardware Set-up.

A 75 MHz Pentium was used as the computer to do the autofocusing and to perform the chromosome detection algorithm. A typical slide contains approximately 4000 frames. A black and white camera is mounted on top of the microscope to capture the images through the lens of the microscope. The image contains 256 shades of grey and has a resolution of 640 by 480 pixels. The images from the camera are sent to the imaging card in the computer. The platform motion and the focusing of the camera are controlled by a motion control card in the computer. The live video image is displayed on the monitor. The computer uses the contrast of the image to determine if it is in focus. When a chromosome spread is detected the computer saves the location of the spread to a file on the hard disk. These images can be recalled at any time by the user for closer inspection. The images can also be viewed through the microscope since the camera does not interfere with the normal viewing of images through the microscope.

5.3 Software

5.3.1 Overview

There are two main parts to the software to control the system. The first part of the system is to control the movement of the platform and the focusing of the microscope. The second part of the program is the chromosome detection system which works on images that it receives from the acquisition system.

After the slides are made they are put under the microscope. The set-up of the microscope involves the adjustment of the light level and the selection of the magnifying power. The light level was set to be just below the saturation threshold of the image. The magnification of the microscope was x200. These factors are well controlled and are easily adjusted.

5.3.2 Autofocus Algorithm

The principles in creating an autofocus algorithm are to make a measure of the sharpness of an image and then to compare that measure to the results of other sharpness measures of the same image at different focal points. This is a contrast based measure since the more contrast in the image, the sharper the image. When the maximum sharpness calculation is obtained then that image is in focus. Since we were only dealing with flat images on the microscope slide the depth of the image is very small.

Groen et al. 1985 analysed various autofocus algorithms and the best results were obtained using a gradient function. We chose to implement this algorithm because is was fast and accurate. This algorithm used only integer arithmetic and was thus promising in its potential to be performed at high speed.

However, we experienced problems with this algorithm because the initial results were disappointing. These results were due to noise in the captured images. After modifying the algorithm we observed significant improvement, which we will detail in the next sections.

Morphological Algorithms

There are several criteria required for a good autofocus algorithm:

(i) The image that is to be put into focus must have a single focal point. There must not be more than one maximum in the sharpness function. This can been seen in Figure 5.2.

Autofocus Sample Plot

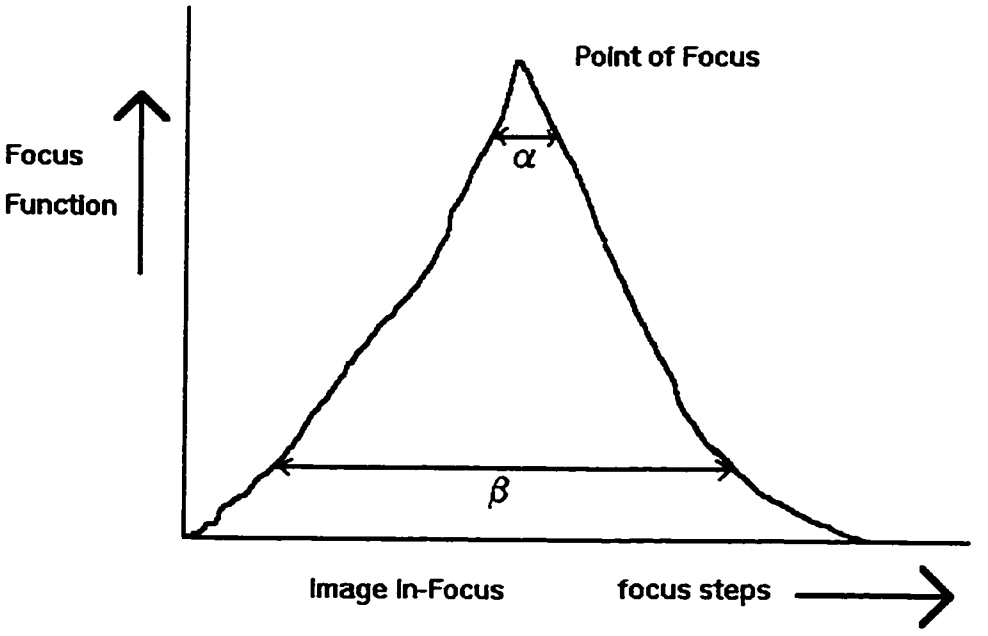


Figure 5.2 Autofocus Sample Plot.

(ii) The point at which the image is deemed to be in focus must be fairly narrow. This is depicted by the length alpha in Figure 5.2.

(iii) The sweeping distance in which the focusing algorithm operates must be large. The sweeping distance represents the range over which the autofocus algorithm can successfully operate. This is indicated by the length beta in Figure 5.2.

(iv) The algorithm must be repeatable. This means that the same location in the alpha range must be found for a given image by the same focusing algorithm when the algorithm is repeated.

(v) The lighting level of the image must not affect the focusing algorithm.

Groen, et al. mentioned there were three classes of focusing algorithms. The first class of algorithms looked at the distances in the height of adjacent pixels. The equation for the first class of algorithms is as follows:

$$F^1_{n,m,\vartheta} = \iint_{image} E\left\{\left|\frac{\partial^n g(x,y)}{\partial x^n}\right| - \theta\right\}^m dxdy$$

Integrated over the image.

Where: $\frac{\partial g(x,y)}{\partial x} = \frac{\Delta g}{\Delta x} = g(i,j) - g(i,j-1)$ and *g(x, y)* is a pixel in the image, *n* is the degree of the partial derivative, *m* is the scaling factor of the differences, and θ is an arbitrary threshold.

The second class of algorithms looked at the heights and depths of the peaks and valleys of the image. The equation for the second class of algorithms is as follows:

$$F^2_{f,\vartheta} = \iint_{image} f[g(x,y) - \theta]dxdy$$

Integrated over the image.

Where g(x, y) is a pixel in the image, *f* is a function chosen to measure the size of peaks and valleys in the image, and θ is an arbitrary threshold.

The third class of algorithms looked at the variability of each pixel from the mean of all the pixels. The equation for the third class of algorithms is as follows:

$$F^3_{m,c} = \frac{1}{c}\iint_{image} |g(x,y) - \bar{g}|^m dxdy$$

Integrated over the image.

Where *g(x, y)* is a pixel in the image, *m* is the scaling factor of the differences, and *c* is a normalising constant.

All three for these algorithms were compared in the above paper on three types of images. The images were (a) a grid, (b) a metaphase (chromosome) spread (this was our interest) and, (c) a portrait.

We chose the first method. Groen et. al. provided experimental evidence in favour of this method over the other classes of algorithms. It was also computationally efficient.

We initially used the following parameters for equation (1):

$\theta = 0$

$m = 2$

$n = 1$

To acquire the sample data, first the image of the chromosome spread was manually focused and then the z co-ordinate of the microscope was noted. Then the z co-ordinate was decreased to a realistic out of focus setting so as to insure that the blurred image we obtained was realistic and representative of what we would get in the application. Then the z co-ordinate was decreased in small even steps and snapshots of the image were taken. These snapshots were then put into files and were later used by the focusing algorithm.

Essentially the squared gradient algorithm did the following calculation:

$$F = \sum_{image} [g(x, y) - g(x - 1, y)]^2,$$

Where g(x, y) is a pixel at coordinates (x, y), and F is the contrast figure of merit.

This process was repeated for all the images in the sequence of images. The maximum calculated result of the algorithm should be the correctly focused image.

5.3.2.1 Preliminary Results

We tested the focusing algorithm on four image sequences, which were AUTO00xx, AUTO10xx, AUTO80xx, and AUTO90xx. The four in-focus images of these series can been seen on the following pages in Image 5.1 to Image 5.4. All chromosome spreads were taken from white blood cells.

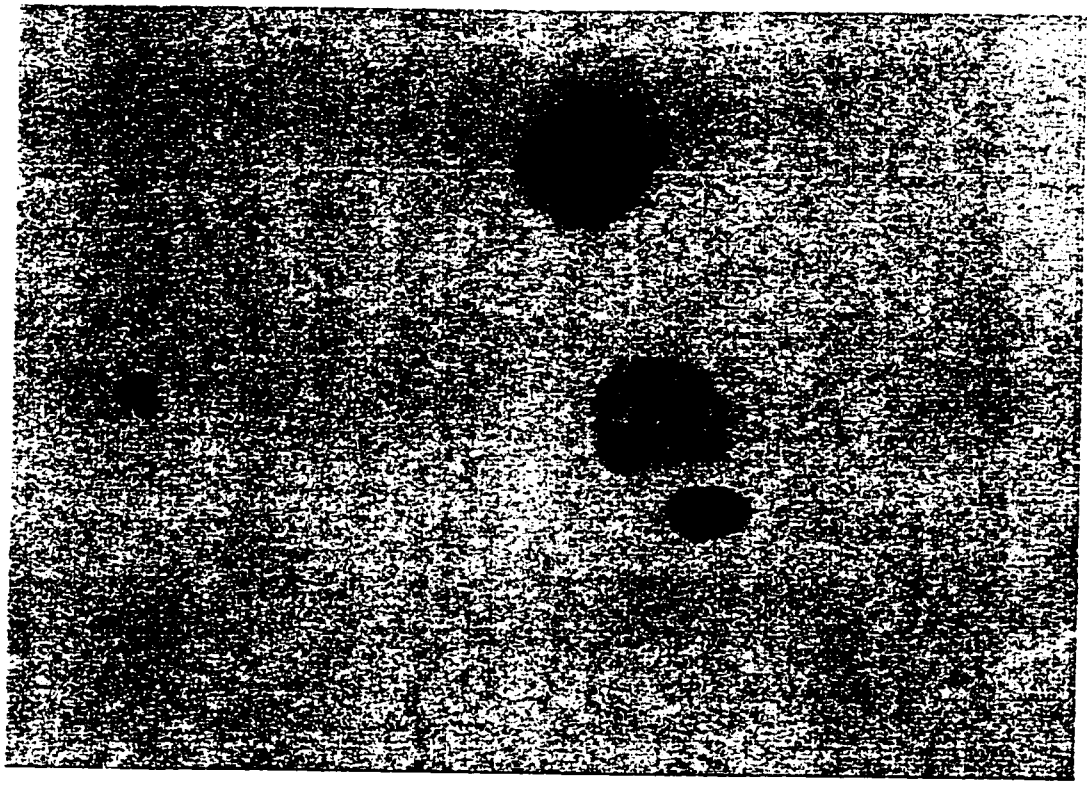


Image 5.1: (AUTO0033.BMP) A poorly formed chromosome spread.

Image 5.2: (AUTO1025.BMP) A well formed chromosome spread.

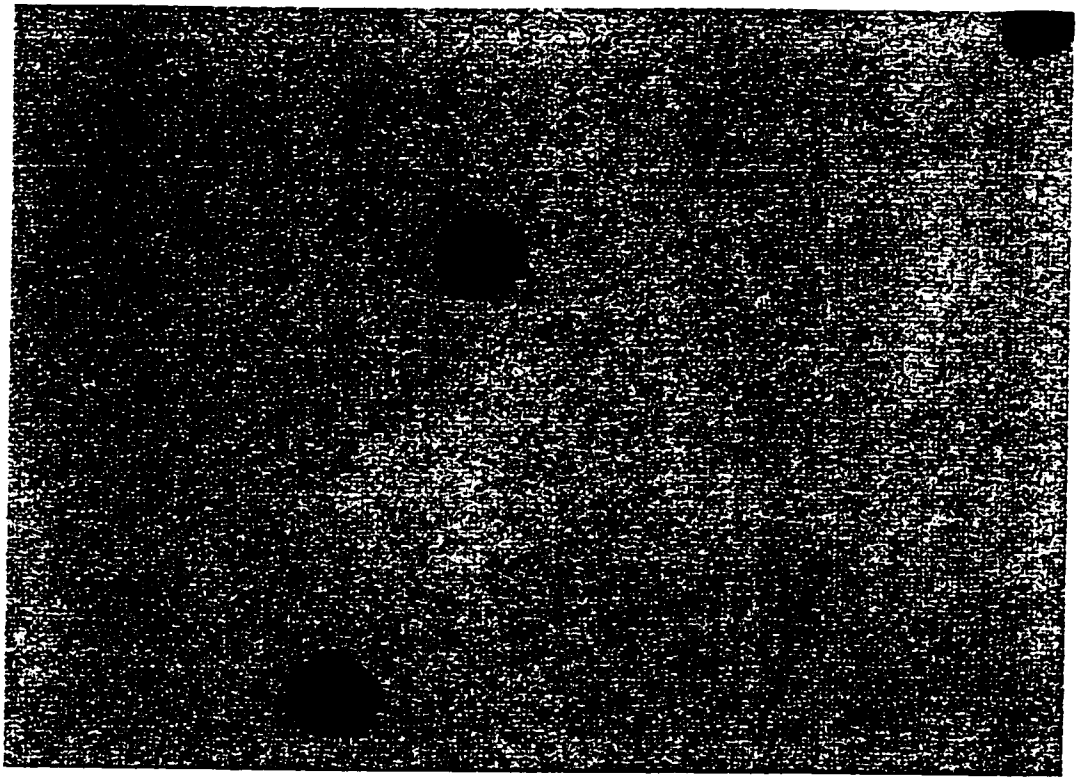


Image 5.3: (AUTO8052.BMP) Three large debris.

Image 5.4: (AUTO9054.BMP) Image filled with debris.

The AUTO00xx series of images consists of 61 images of a chromosome spread. This particular chromosome spread is of poor quality because the inter-chromosome distances are too small. The image that is in focus is image number 33.

The AUTO10xx series of images consists of 54 images of a chromosome spread that could be used for further analysis because it is a good chromosome spread. The image that is in focus is image number 25.

The AUTO80xx series of images consists of 72 images of three debris items. These debris are probably red blood cells. There are very few features seen in these debris and hence it is difficult to put this image into focus. There is no chromosome spread in this image. The image that is in focus is image number 52.

The AUTO90xx series of images consists of 82 images of a large number of medium and small debris. There was no chromosome spread in this image. Because of the large amount of debris in this image there are a lot of image elements that can be used to put the image in focus. Therefore, of the four series of images this produces the best initial result. The image that is in focus is image number 54.

The plots of the series of these images in varying degrees of focus can be seen on the following pages. The initial plots were generally poor. See Figure 5.3 for the OUT0.TXT plot of the AUTO00xx series of images. OUT0 means the output of the algorithm for the AUTO00xx series images. The AUTO00xx series of images did correctly produce the highest peak as the correct in-focus point, however there are two smaller peaks on the left and the right of this central peak. If the focusing algorithm looked only in the vicinity of one these smaller peaks then an incorrect focus point would be obtained.

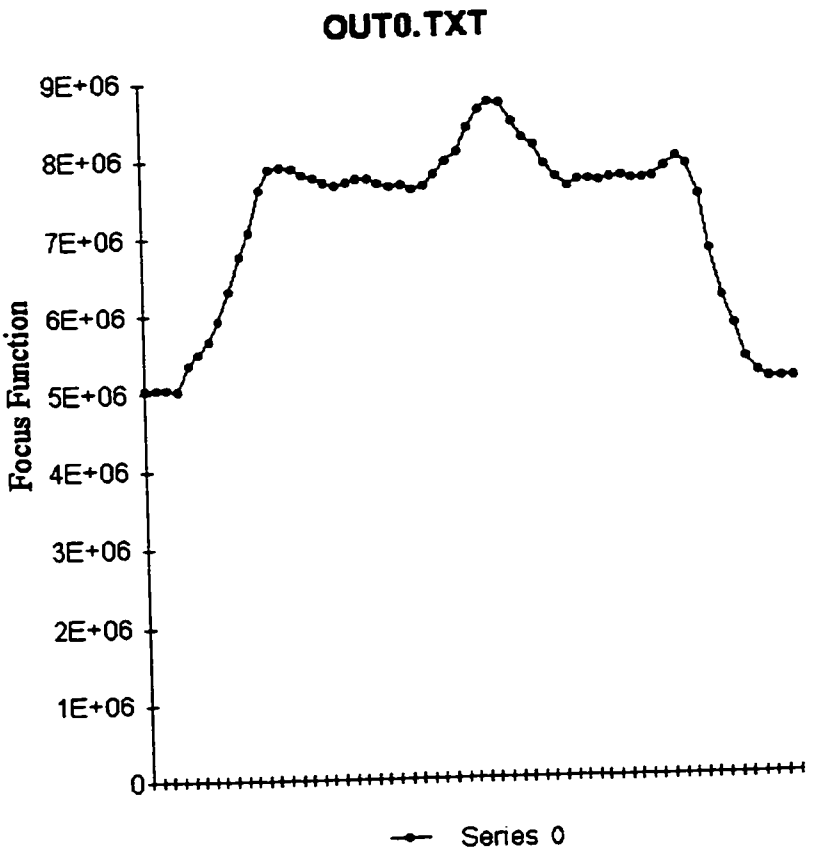


Figure 5.3: Output of the squared gradient function on the AUTO00xx series of images.

The AUTO10xx series of images did not produce an adequate peak. See Figure 5.4 for the OUT1.TXT plot of the AUTO10xx series of images. In the plot the highest point is still the correct focus point, however there is sufficient variability in the graph to easily confuse the auto focus algorithm.

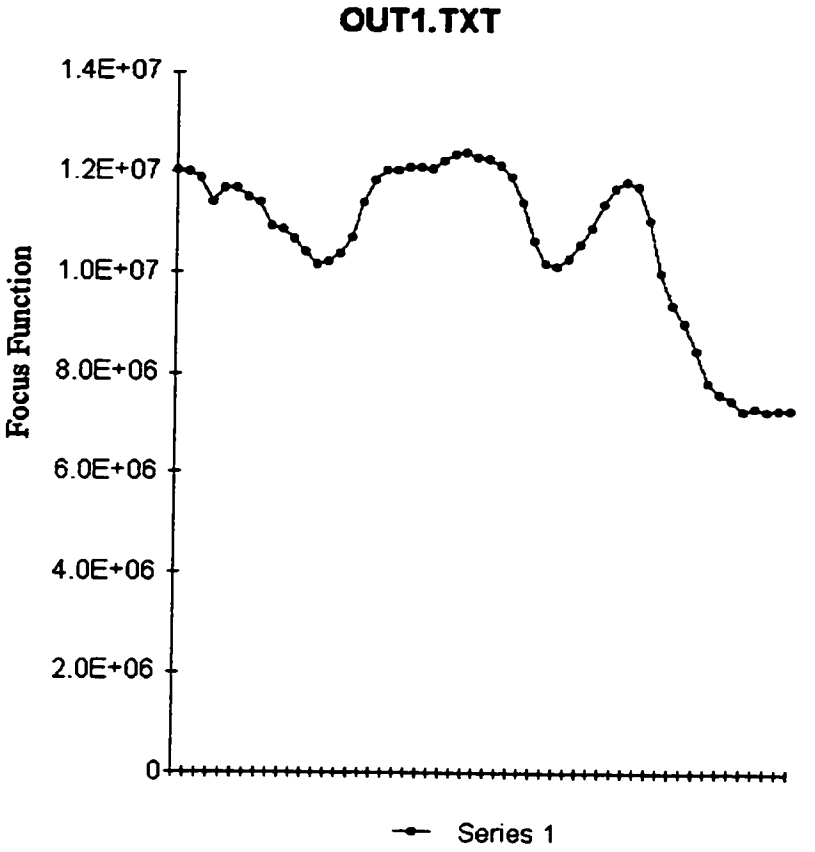


Figure 5.4: Output of the squared gradient function on the AUTO10xx series of images.

The AUTO80xx series of images also did not produce an adequate peak. See Figure 5.5 for the OUT8.TXT plot of the AUTO80xx series of images. In the plot, the highest point is still the correct focus point, however there is a flat plateau with a jagged top which would confuse the autofocus algorithm.

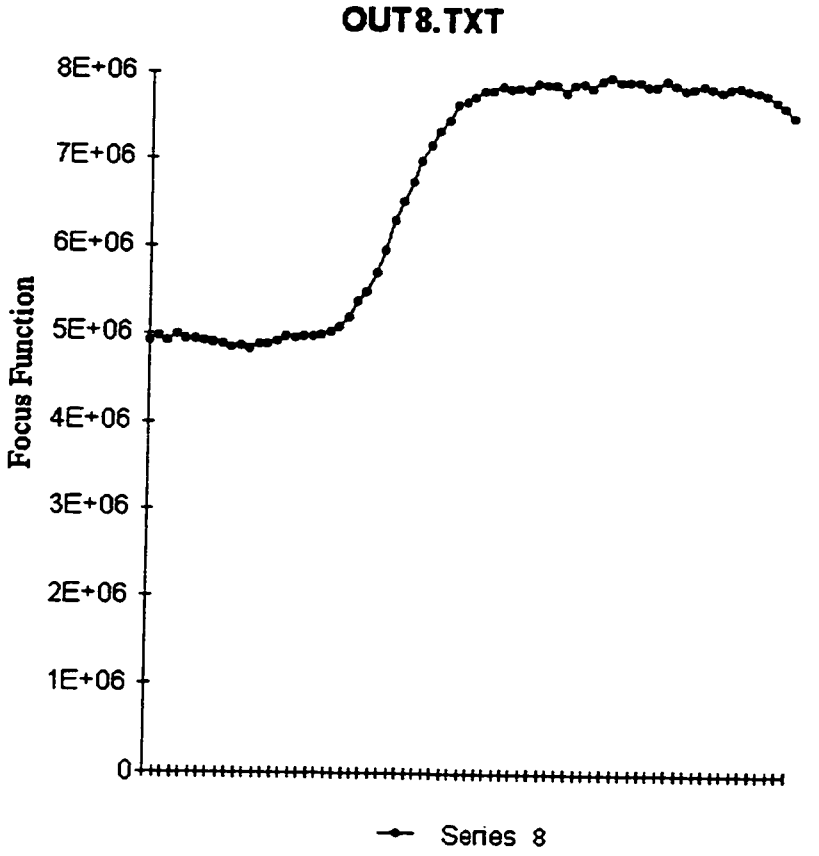


Figure 5.5: Output of the squared gradient function on the AUTO80xx series of images.

Finally, the AUTO90xx series of images did produce a very good peak. See Figure 5.6 for the OUT9.TXT plot of the AUTO90xx series of images. There is only one peak in this plot and the sides of the peak are fairly smooth. This is the type of result that we had hoped to obtain in all of the graphs. Unfortunately we can not guarantee that such a large number of objects will be available in an image to produce such a good result. This image for example does not contain a chromosome spread and has only debris. A typical good spread contains a large light grey background and a small dark grey foreground, so the above plot result would be difficult to obtain.

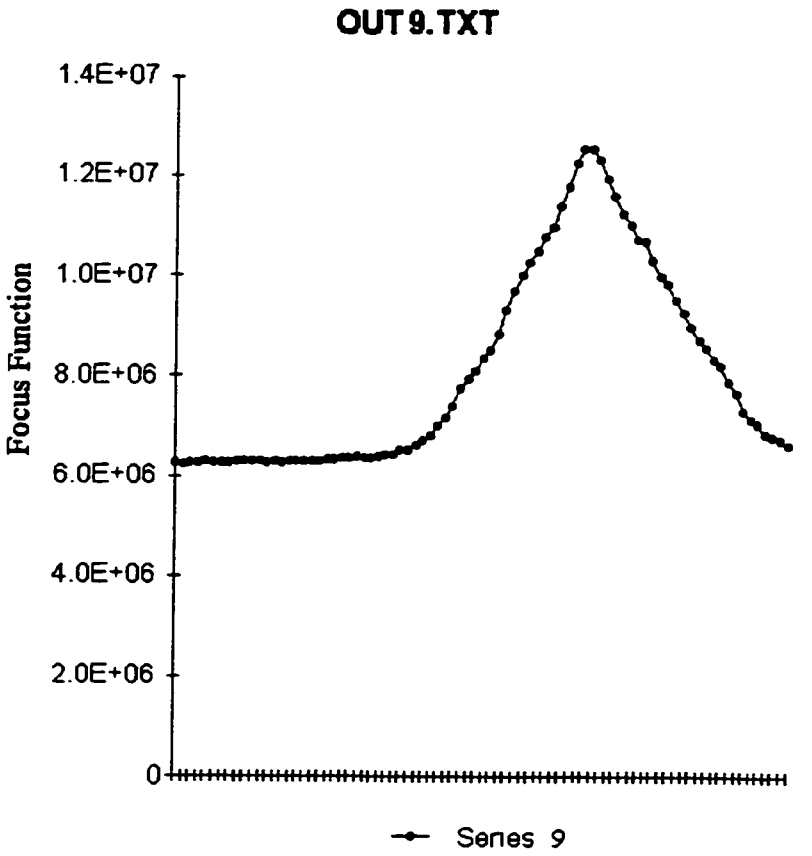


Figure 5.6: Output of the squared gradient function on the AUTO90xx series of images.

Based on the above results it was hypothesised that they were due to the following:

(i) The lighting may have been a problem. This could cause inconsistent results if the light level changed over the time needed to aquire all the images in the sequence.

(ii) If there was sizeable noise in the video camera, this would cause errors in the calculation of the gradients.

(iii) Any fingerprints or dust on the chromosome slides may have caused a false peak in the focusing plots. Possible fingerprints or dust that are deposited on the cover glass or under the glass slide would be in separate focal planes, and this could cause the focusing algorithm to pick one of these focal planes as a false peak. This is especially

possible in the AUTO00xx series of images, that contain two false peaks on either side of the main (correct) peak.

(iv) Finally any problems in the camera gain may have caused poor reception of image details. This could be the case if some of the light sensitive receptors in the camera misfired or did not fire at times.

5.3.2.2 Algorithm Development and Selection

In order to offset the effects of the noise in the images, we decided to filter the gradients of the images. By using this method we hoped to produce better results. See Figure 5.7 for a block diagram of the first filtering technique we used.

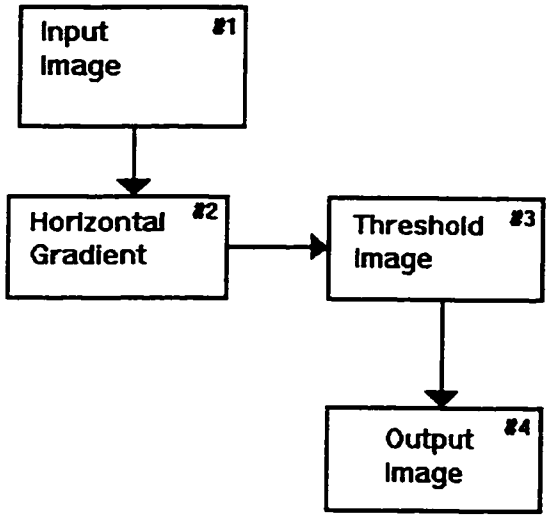


Figure 5.7: First filtering technique.

To start the program is given the input image. The gradient of this image is computed in the x direction and is put into image number 2. The gradient is thresholded and the output is put into image number 3. This threshold was set to 32. This means that all of the differences less than or equal to 32 are ignored in making the calculation. This helped compensate for the noise effects which in the original image seemed to have

overwhelmed the signal. The results obtained from this procedure produced the well-behaved plots seen in Figures 5.8 to Figure 5.11 on following pages. The one exception was the series AUTO80xx in which the result was poor. This result, however, is not a major concern because the image was almost featureless and had no chromosome spread in it.

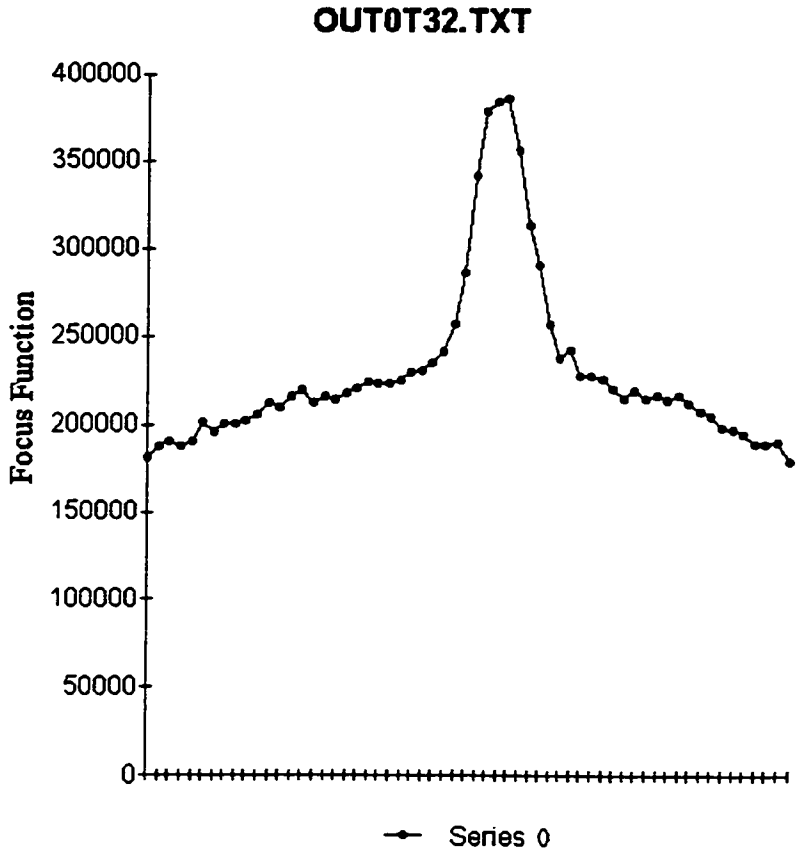


Figure 5.8: First filtering technique on the AUTO00xx series of images.

This plot does produce a good peak for the first set of images.

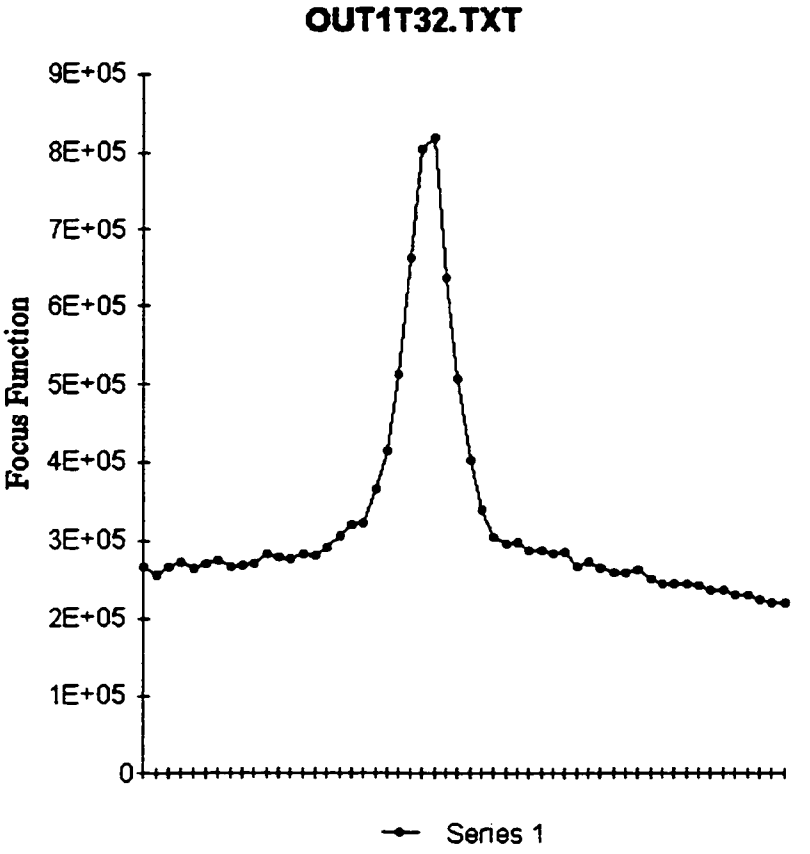


Figure 5.9: First filtering technique on the AUTO10xx series of images.

This plot does produce a good peak for the second set of images.

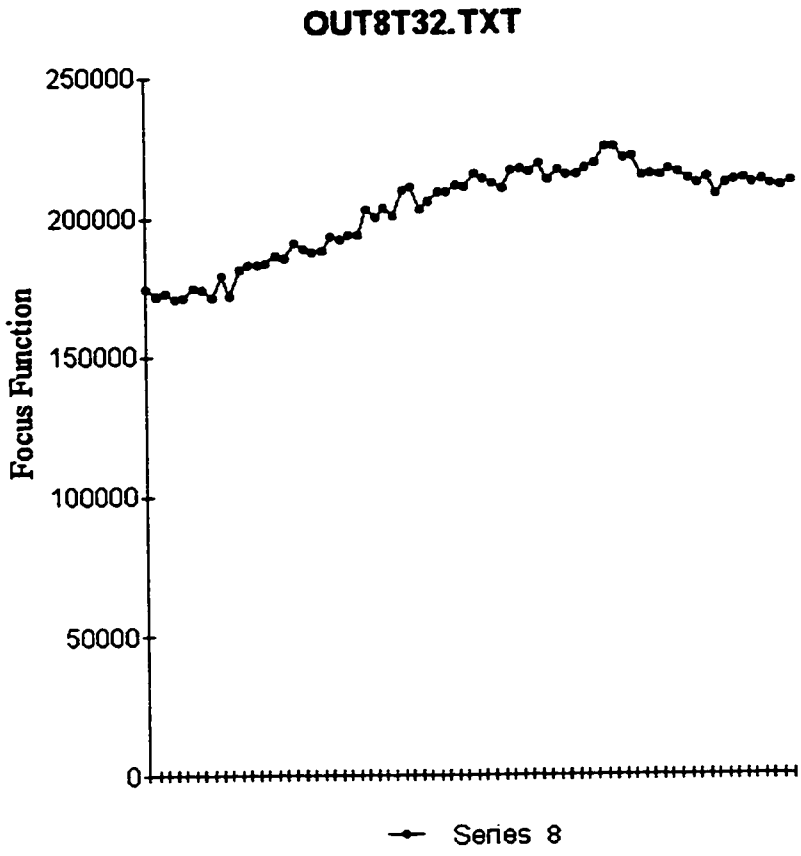


Figure 5.10: First filtering technique on the AUTO80xx series of images.

This plot does not produce a good peak for the third set of images. Hence the in focus point could not be determined from this input image.

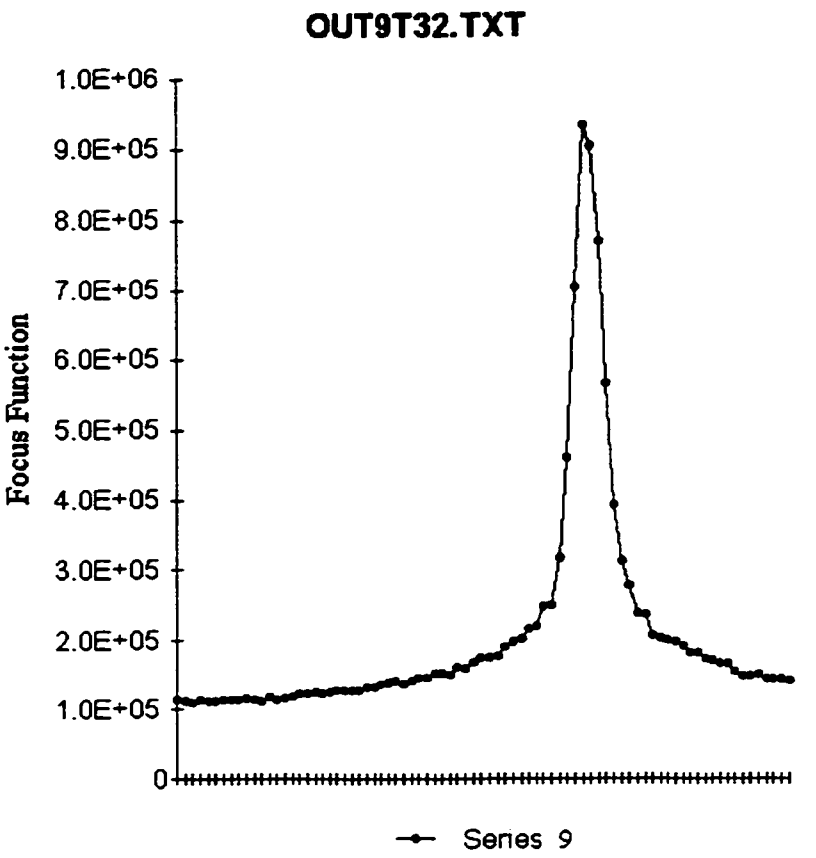


Figure 5.11: First filtering technique on the AUTO90xx series of images.

This plot does produce a very good peak for the fourth set of images.

In chromosome spread images such as series AUTO00xx and AUTO10xx there were enough features to produce a good result. In later analysis of focused images the image containing just three objects would be rejected anyway, so it is not as important to get it in perfect focus.

The initial filtering results were not perfect though. They did produce very sharp peaks, but around the peaks there tended to be some jagged regions. This could produce a false peak and was of concern.

Another filtering technique was tested in which only the regions of interest in the image would be used to calculate the gradient. The block diagram number 2 for this technique is depicted in Figure 5.12.

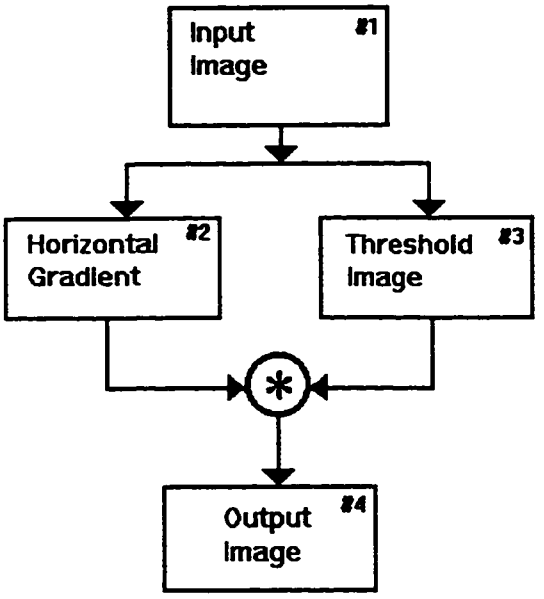


Figure 5.12: Second filtering technique.

Here the gradient of the x co-ordinate was performed on the input image and the result was stored in image number 2. Also a threshold on image number 1 was performed and the result was stored in image number 3. The threshold sets all pixels equal to or above intensity level 100 to zero and the rest of the pixels were set to one. The larger pixel values represent lighter levels of grey in the image. A value of 0 represents black and a value of 255 represents white. The gradient image number 2 was then multiplied with image number 3 to produce the final result in image number 4. The squared gradient calculation is then made on this final image. This filtering technique had the effect of ignoring unimportant parts of the image that tended to produce noise effects.

The results of this filtering effort produced the well-behaved plots seen in Figures 5.13 to Figure 5.16 on following pages:

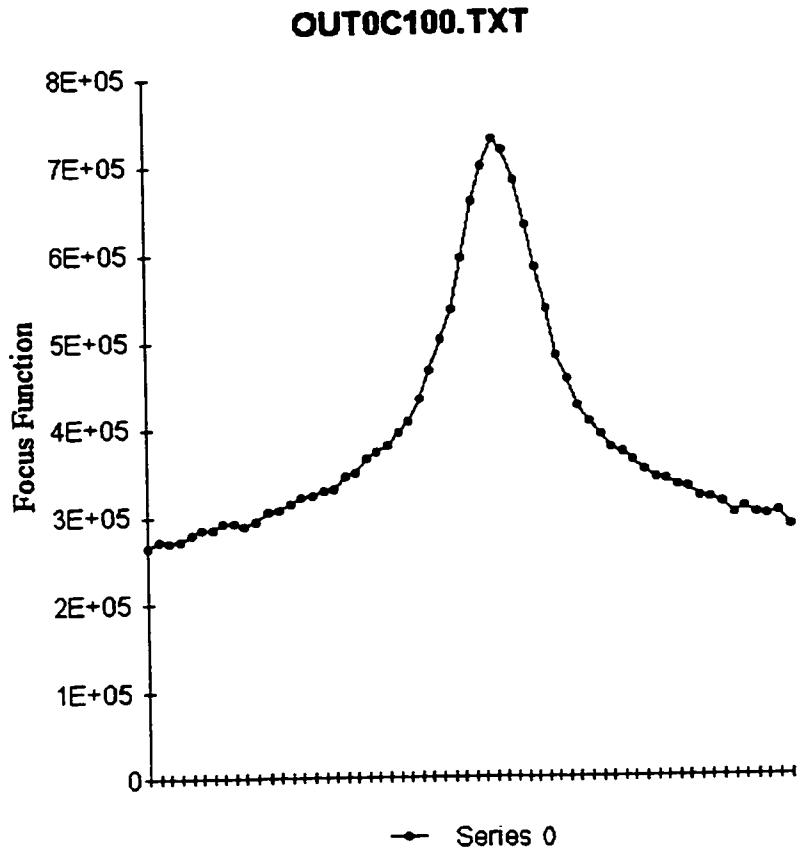


Figure 5.13: Second filtering technique on the AUTO00xx series of images. This plot does produce a good peak for the first set of images.

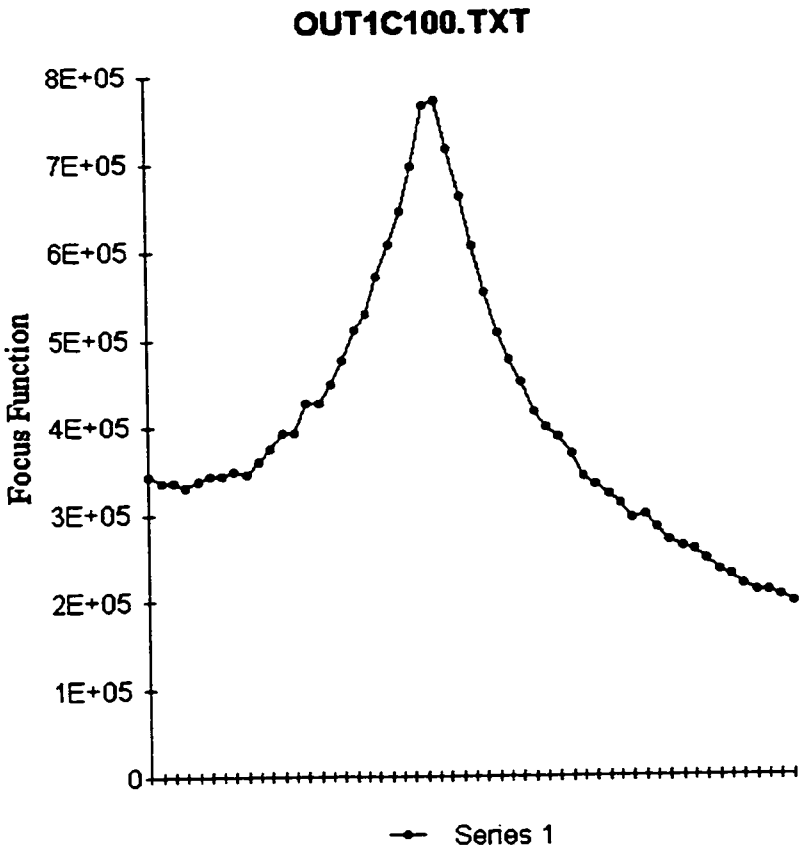


Figure 5.14: Second filtering technique on the AUTO10xx series of images. This plot does produce a good peak for the second set of images.

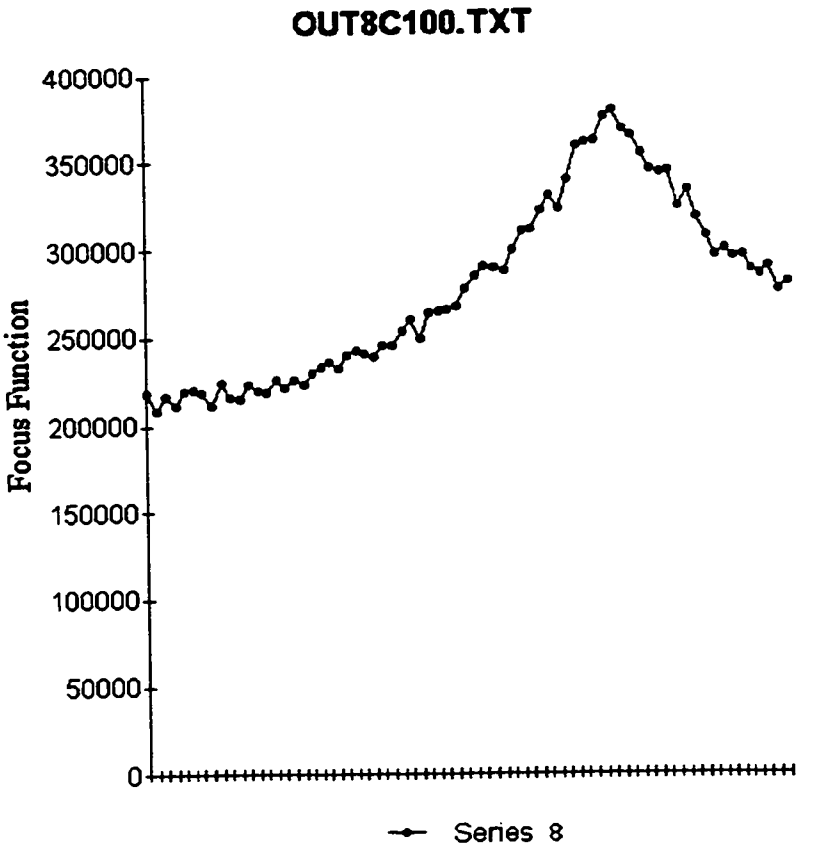


Figure 5.15: Second filtering technique on the AUTO80xx series of images. This plot does produce a reasonably good peak for the third set of images.

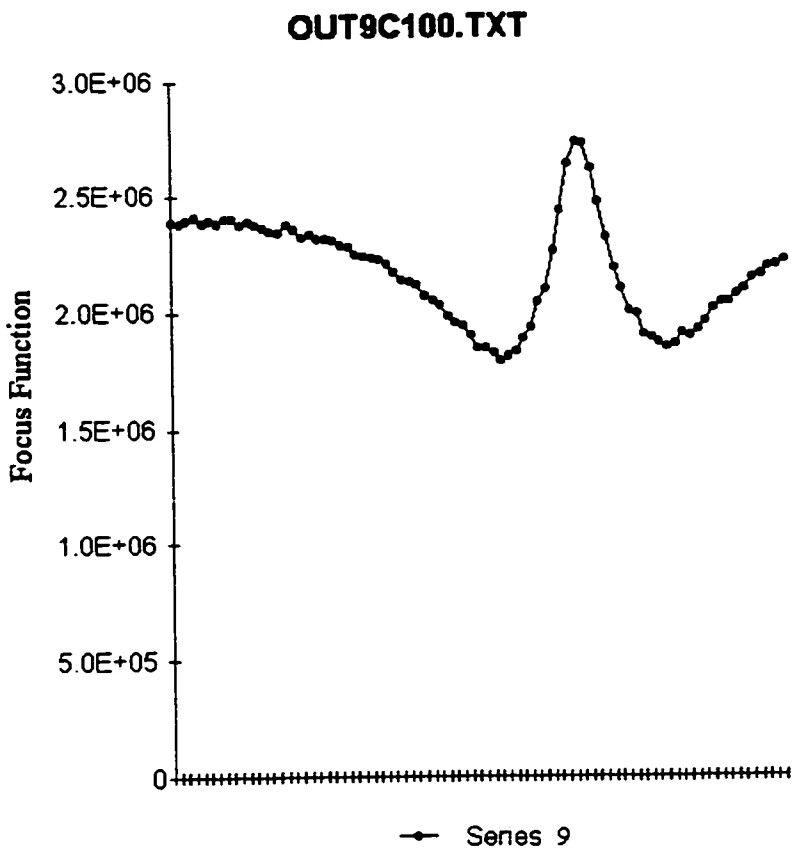


Figure 5.16: Second filtering technique on the AUTO90xx series of images.

This plot does produce a good peak for the fourth set of images.

This filtering technique produced the best results. Even series AUTO80xx produced a reasonable though not acceptable peak. An interesting result of this filtering approach was that image series AUTO90xx produced a plot that starts high goes down then produces a good peak, goes down again and then goes up. This algorithm could find one of the end points as a false peak, but this would be relatively unlikely since the previous image would also be in focus and we would only be looking in the area immediately around the peak. In general these heavily cluttered images are fairly rare. Since this later algorithm does produce a correct in focus point for all of the images look at so far, we close this algorithm for the autofocusing algorithm. Additionally, this later algorithm had smother slopes leading to the peak and this was an additional benefit.

5.3.3 Spread Detector

There are two main goals in the chromosome spread detection part of the system. The first is to increase the number of True Positives of types #1 and #2 and reduce the number of False Positives and True Positives of types #3 and #4. The second goal is to speed up the system as much as possible. The methodology has been tried under x200 an x400 magnification powers with similar results.

Figure 5.17 illustrates the block diagram of the methodology. The procedure that is used to locate chromosome spreads is a two phase band pass filter. The filter removes large picture elements on the high side of the particle size spectrum as well as small isolated particles on the low side of the spectrum. The algorithm works according to the following steps:

Chromosome Detection Algorithm

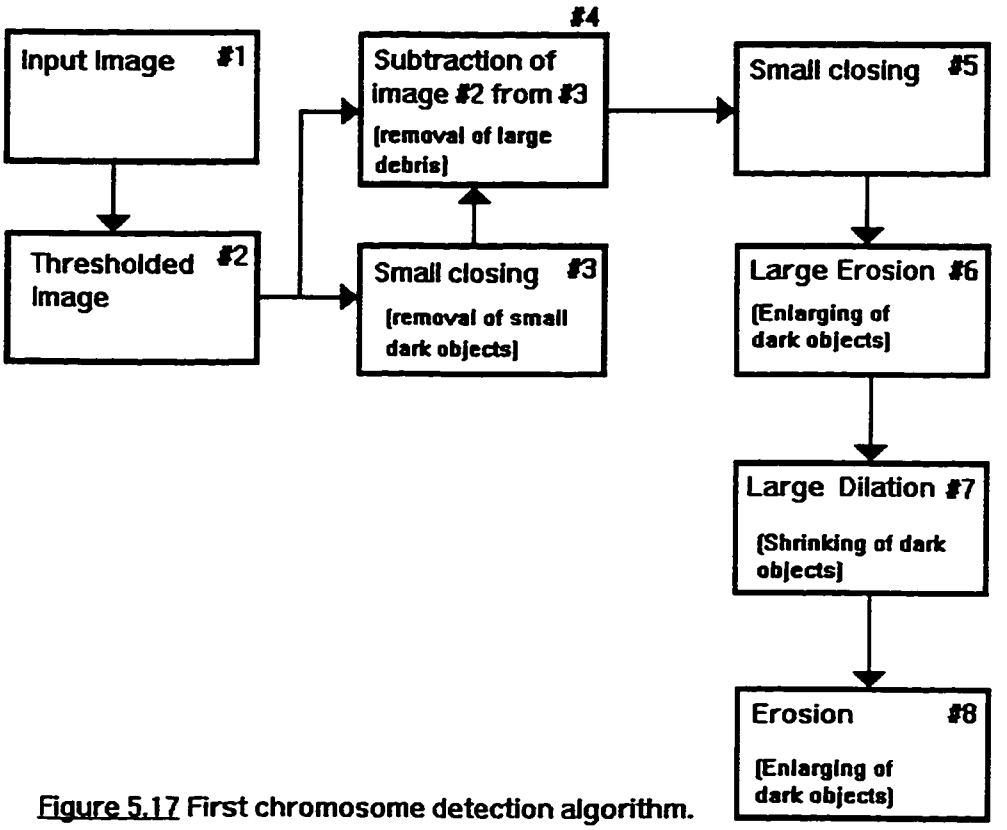


Figure 5.17 First chromosome detection algorithm.

The first step in the image processing is to do a threshold of the input image (#1) to make a binary image. This initial threshold will make the foreground dark grey picture elements all black and the light grey background all white. This threshold will also greatly increase the speed of subsequent image processing operations. No grey levels exist after this point since the image contains just total black (0) and total white (255) values. This produces image (#2).

The closing function is performed on image (#2) to produce image (#3). The closing function consists of a dilation followed by an erosion. When a dilation is performed the white areas of the picture are expanded. This causes the dark areas to be made smaller. Then an erosion is performed. During an erosion the white areas of the picture shrink and the dark picture elements are enlarged. The result of this closing is

that the small dark picture elements are removed from the image (#3). Image (#4) is made by subtracting image (#2) from image (#3). For the actual algorithm the image is now in inverse video, but for the purposes of this discussion we shall presume that the image still has a white background and black picture elements.

The subtraction used to produce image (#4) has the effect of removing the large dark picture elements. These large dark picture elements consist of debris that are not wanted in the image.

After this has taken place there are some small fragments left over from the edges of the larger picture elements that are caused by the smoothing effect of the above closing operation. These small edge fragments are then removed by an additional small closing operation. This produces image (#5).

Next an erosion is performed with a sufficient magnitude to cause the chromosome picture elements to overlap and become one large picture element (#6). The isolated small picture elements will then be eliminated in image (#7) by a large dilation, which is larger than the erosion used to produce image (#6). Then an erosion is performed to make the remaining picture element the approximate size of the chromosome spread (#8).

A series of images showing the operation of the above algorithm can be seen on the following pages in Image 5.5 to Image 5.12:

Image 5.5: (CHRO1.BMP) Original image.

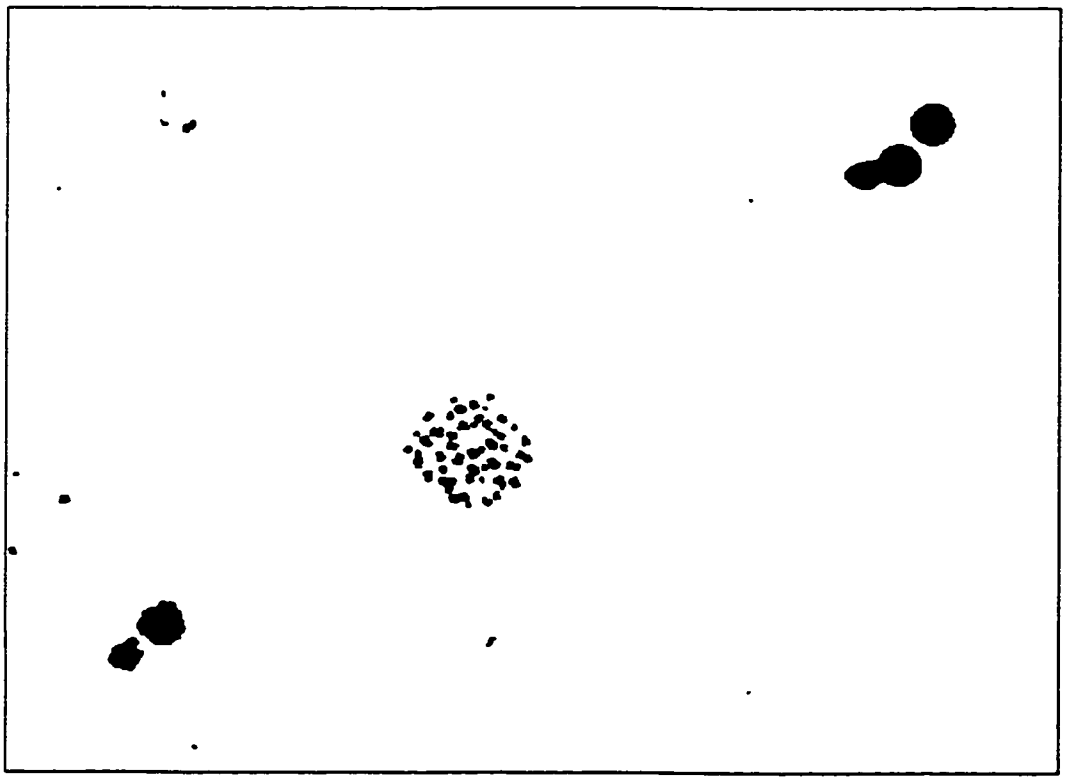


Image 5.6: (CHRO2.BMP) Thresholded image.

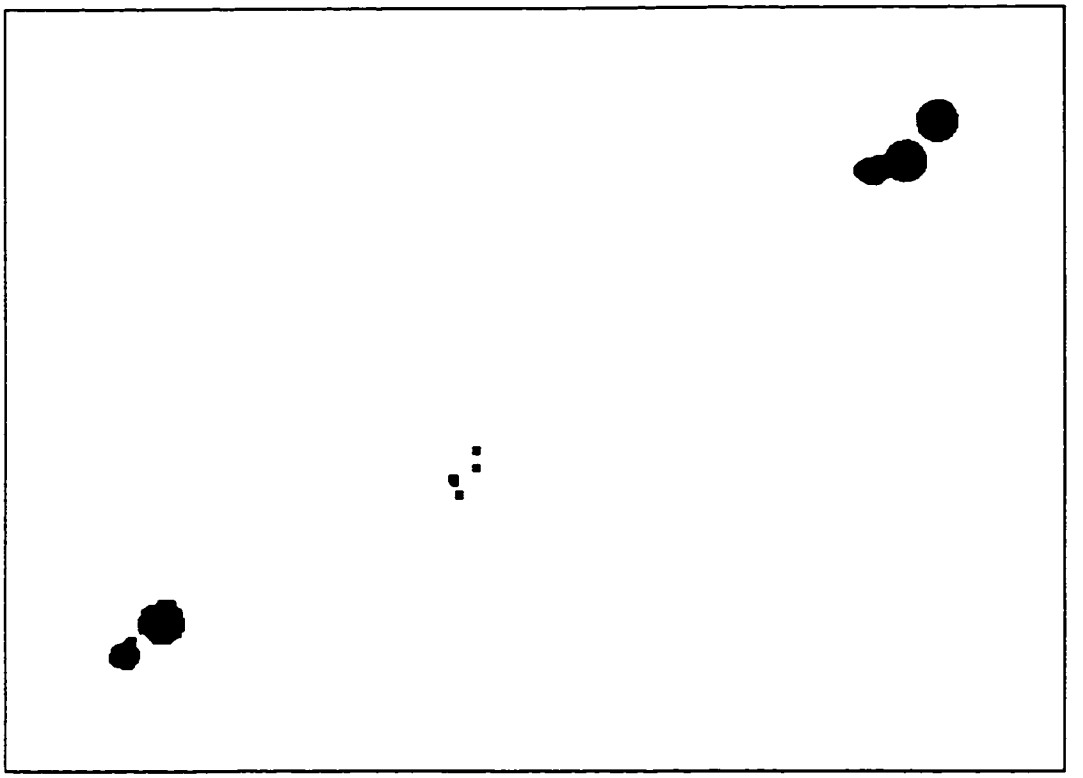


Image 5.7: (CHRO3.BMP) Image produced by the closing function.

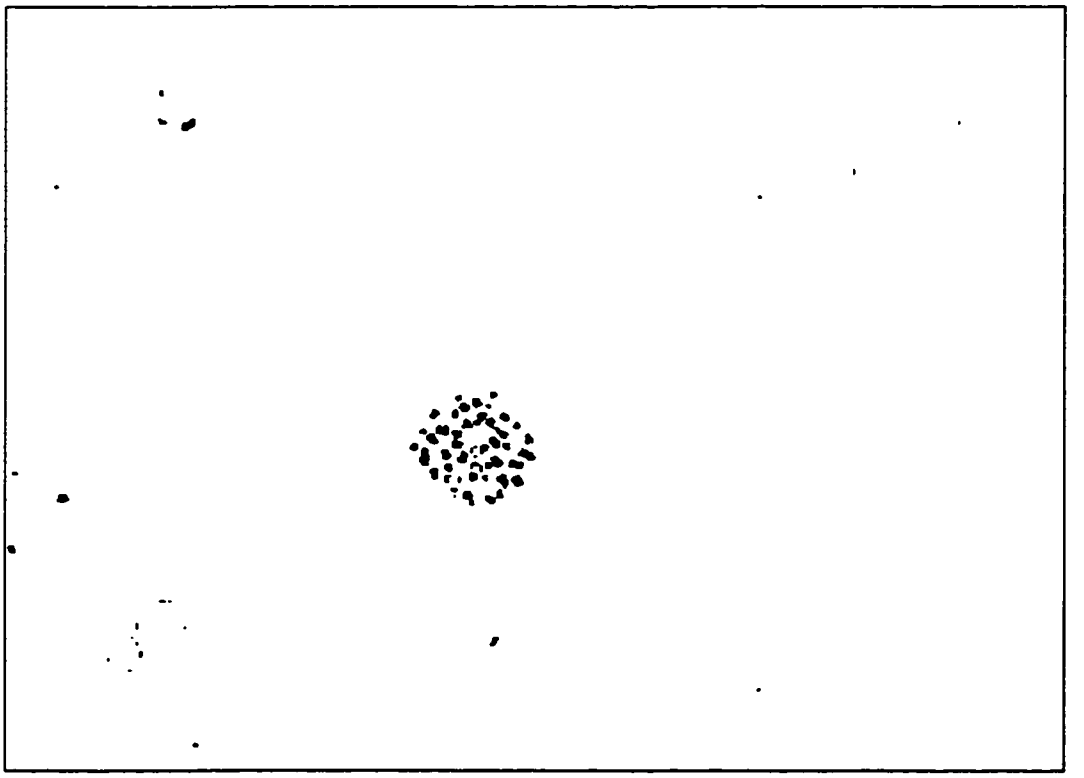


Image 5.8: (CHRO4.BMP) Image produced by subtraction of image 5.7 from image 5.6.

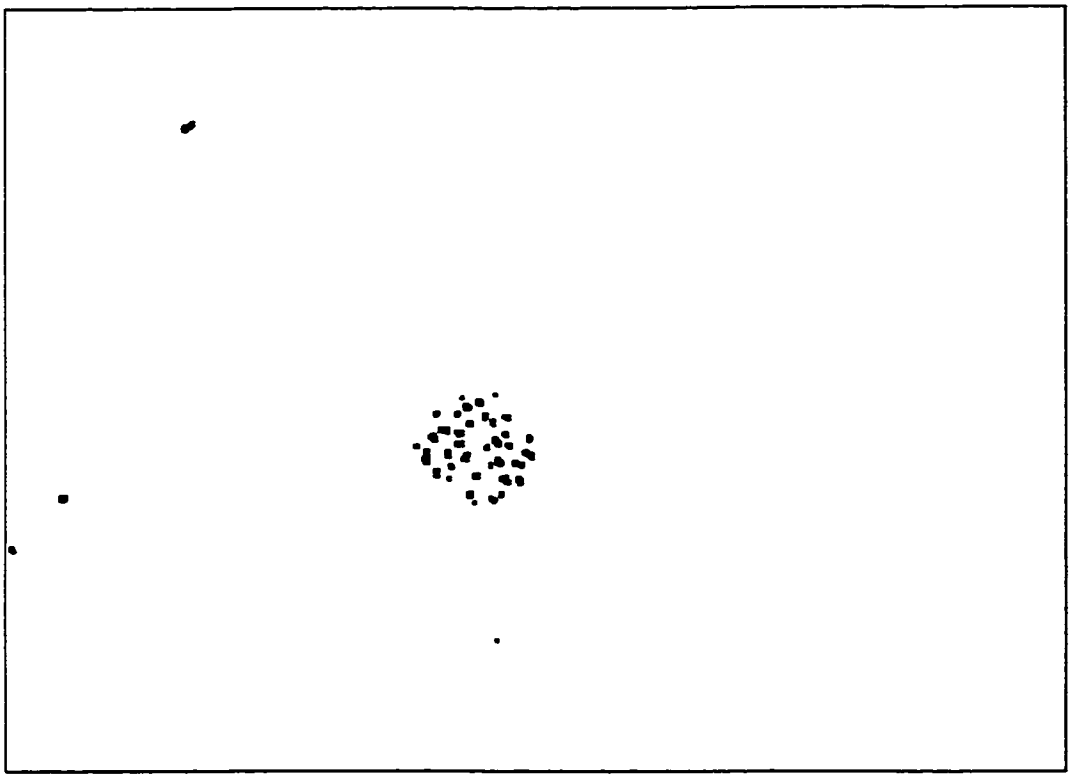


Image 5.9: (CHRO5.BMP) Image produced by the small closing function.

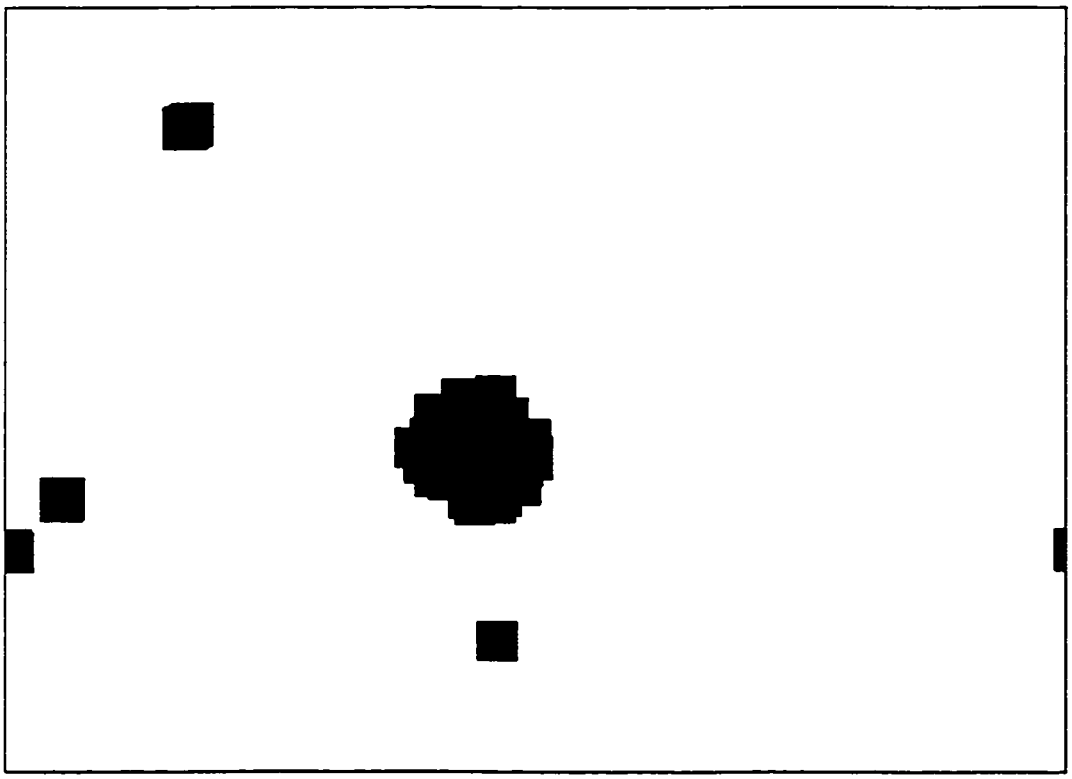


Image 5.10: (CHRO6.BMP) Image produced by the large erosion function.

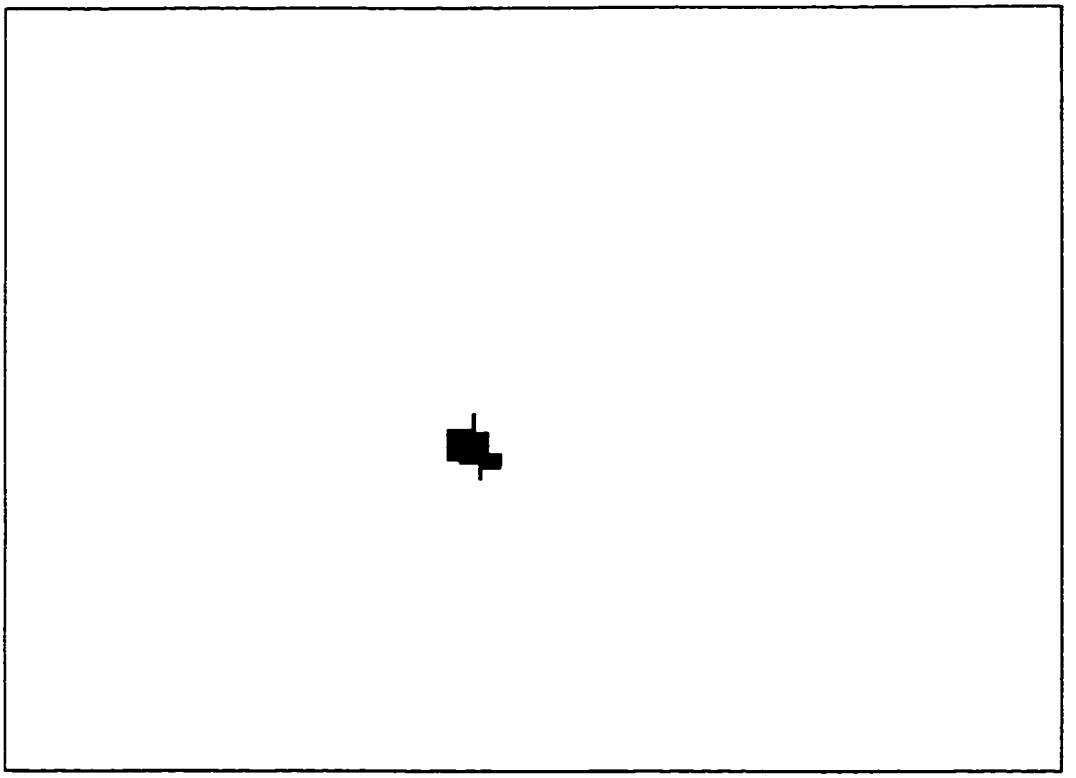


Image 5.11: (CHRO7.BMP) Image produced by the large dilation function.

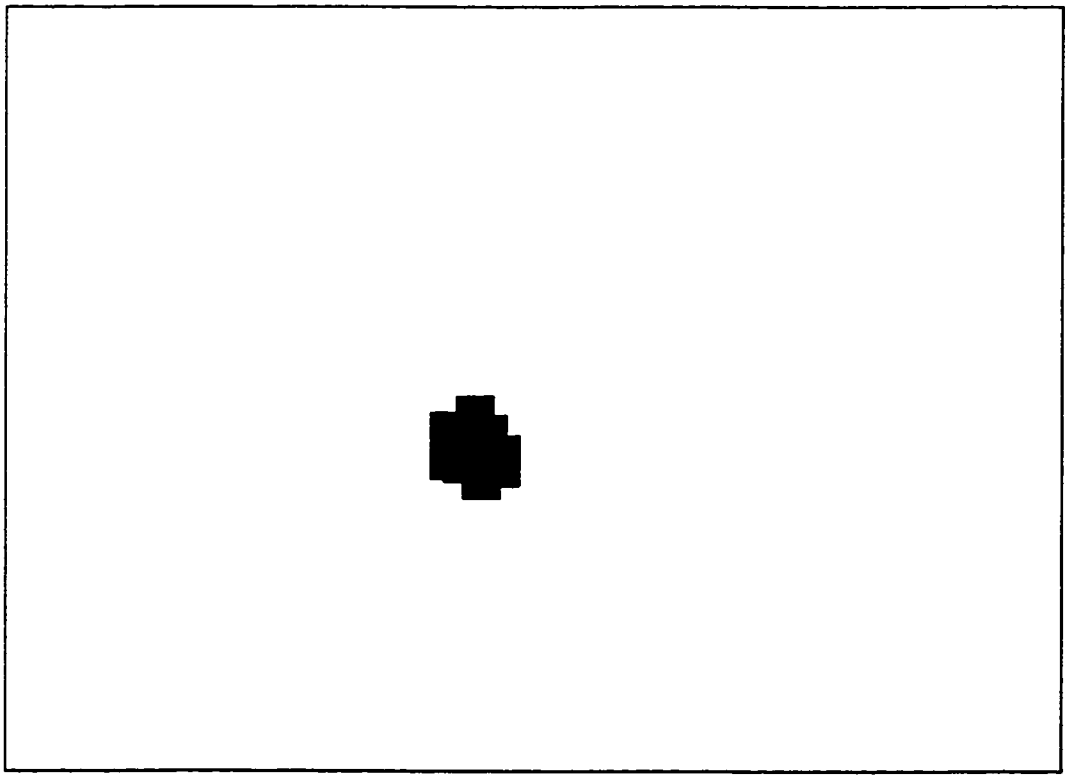


Image 5.12: (CHRO8.BMP) Final Image produced by the large erosion function.

The parameters used for the above algorithm are stored in an initialisation file. Two different sets of initial values for the parameters are hard coded into the program. This first set is for the x200 magnification and the second set is for the x400 magnification. All the image processing used in this thesis was done under the x200 magnification. The lower the magnification the more quickly the program can scan the slide because fewer images need to be processed.

The parameters depicted in Figure 5.17 will now be listed:

- The input image: (Image #1): This grey scale image may contain one or more chromosome spreads.

- The initial threshold (Image #2, Parameter #0): Depends on the light intensity of the foreground and background objects.

- Size of the small closing (Image #3, Parameter #1): This is the maximum size of the individual chromosomes that we are looking for.

- Subtracted image (Image #4): This is the result of subtracting image #2 from image #3.

- Size of the small closing (Image #5, Parameter #2): This is the maximum size of the small debris and edge fragments left over from the subtraction.

- Size of the large erosion (Image #6, Parameter #3): This parameter represents the maximum inter-chromosome distance that will be accepted.

- Size of the large dilation (Image #7, Parameter #4): This is the minimum spread size that will be accepted.

- Size of the large erosion (Image #8): This parameter produce a final image that is pleasing to the eye in terms of it matching the original chromosome spread size. It is not necessary to use this parameter simply to get the x-y co-ordinates of the chromosome spread.

The speed of the basic algorithm described above was increased by the use of the following methods: First the speed of the throughput has been greatly increased by the

use of an early stop mechanism. This was accomplished by doing a scan of the image at three different stages in the chromosome detection process and checking to see if there was anything still remaining in the image to be processed. If there was nothing to be processed the program reports that nothing has been found and a negative is returned. The program then goes on to the next image to be processed. Another time saving feature that has been added to the program is that a separate text file is now used to report the results of the image processing. This text file contains the file names of the processed images and the letter "V" is appended to the front of the given image file name if one or more chromosome spreads are found and the image is also written to the disk. Saving the image file to disk can be time consuming. If no chromosome spreads are found then the letter "X" is appended to the front of the given image file name to be saved to the text file and no image is actually saved to the disk. This results in a substantial time savings in the performance of the program. On the 75 MHz computer used to test the algorithm this results in a time savings of 21 seconds for images that contain one or more chromosome detections. This time will vary according to the computer and hard disk used. This time saving feature will not be necessary in the production version of the program since only the co-ordinates of the chromosome spreads will be saved to disk, however, for the purposes of this thesis the actual images are saved to disk. The reason for this is the need for data to produce R.O.C. curves which will now be discussed.

As mentioned above the distance function also greatly improves the speed of the algorithm in a transparent way. The dilations and erosions that utilise the distance function have the same effect as the normal dilation and erosions except they complete much more quickly.

After all the time saving features had been added to the program the average time to process an image that contains one or more chromosome spreads was 7.83 seconds on a 75 MHz computer compared to 15.89 seconds for the basic algorithm. The last step in

the image processing algorithm is only for cosmetic purposes and can be skipped which will then leave a smaller marker on any chromosome spreads that are detected. If this happens then the final time to detect one or more chromosome spread is 5.55 seconds. This time of the algorithm is also equivalent to a rejection (early stop) taking place after image (#7) has been scanned in Figure 5.17. For a rejection which takes place after image (#5) has been scanned a negative will be returned after only 2.26 seconds. Finally, if a rejection that takes place just after the initial threshold in image (#2) this will take just 0.5 seconds.

The time needed to locate 1000 chromosome spreads has yet to be determined by the functioning of the final version of system. Normally, 1000 chromosome spreads need to be analysed under high magnification (usually x1000) to give a reliable determination of cell damage for low radiation doses (< 1.5 Grays). For higher doses of radiation about 500 chromosome spreads need to be detected. The time needed to obtain these 1000 chromosome spreads will depend to a large extent to the individual slide used in the analysis. Some slides contain very few chromosome spreads, while others contain many more chromosome spreads. At the time of the writing this thesis, the chromosome detection, platform movement and autofocusing components of the system had not been totally integrated.

It is estimated that a typical chromosome slide will contain approximately 100 to 150 good quality chromosome spreads. Since the algorithm only finds 84.6% of these spreads we would only find 85 to 127 good quality chromosome spreads per slide. Therefore, to collect 1000 good quality chromosome spreads 8 to 12 slide would need to be scanned. The following calculation is used to calculate the time to collect a single slide:
Low density: 85 chromosome spreads per slide

{stage .1} {stage .4} {stage .6} {stage .7}

4000 * (0.05 * 0.5s + 0.05 * 2.26s + 0.87875 * 5.55s + 0.02125 * 7.83s) = 20725.8s

= 5.757 hours for 1 slide

If we add 1.5 hours for autofocusing and platform movement we get:

7.257 hours for one slide

= 87.084 hours for 12 slides = 3.63 days for 12 slides

High density: 127 chromosome spreads per slide

{stage .1} {stage .4} {stage .6} {stage .7}

4000 * (0.05 * 0.5s + 0.05 * 2.26s + 0.86825 * 5.55s + 0.03175 * 7.83s) = 20821.56s

= 5.784 hours for 1 slide

If we add 1.5 hours for autofocusing and platform movement we get:

7.284 hours for 1 slide

= 58.27 hours for 8 slides = 2.43 days for 8 slides

As stated previously multitasking could be used to perform the chromosome detection, autofocusing, and platform movement in parallel. It would probably be true that only one in five images need to be focused since the image does not go out of focus quickly form frame to frame. Also, with a faster computer this time could be greatly reduced and also if two microscope systems were working in parallel then the time required to locate 1000 spreads would be cut in half. For future work, more testing of the final production version of the system will be needed to give a reliable estimate of total time to locate 1000 good quality chromosome spreads.

Compared to the Cytoscan 110 system discussed in chapter 3 the system described by this thesis has greater accuracy. The Cytoscan 110 system takes 2 minutes to scan a single slide. With the poorer accuracy more slides would be needed to get 1000 good quality chromosome spreads. It is estimated that 15 slides would be requited and hence 1000 good quality chromosome spreads could be found in one half hour. This is significantly faster than the system described by this thesis, however, it takes considerable time to manually check the found chromosome spreads for aberrations. Taking this into account, it is reasonable to suggest that the system described by this thesis could keep up

with the human analyst and at the same time display better quality chromosome spreads to be analysed.

Another approach was tried after these initial results were obtained. This new approach attempted to deal with the problem caused by the light intensity of the images. The above algorithm is sensitive to the general light intensity of the chromosomes compared to their background. If the light level is set too high then the lighter chromosomes will not show up in image (#2) from Figure 5.17 above because they will be over the initial threshold level. This can cause valid chromosome spreads to go undetected. Alternatively, if the light level is set to low then a chromosome spread that is sitting on a darker than normal background will tend to be thresholded as one or more large picture elements. They will then be eliminated during the image processing steps outlined above.

To compensate for these problems a technique was used that delayed the threshold to a later point in the chromosome detection algorithm. This has the effect that the grey levels are still present during the first part of the image processing and thereby taking advantage of certain properties of these grey level images. As a result of delaying the threshold the grey level versions of the image processing functions are used up to the point that the threshold is performed instead of the black and white versions of these functions. Since the initial sizes of the image processing function calls are small and the image is converted to black and white before the larger dilations and erosions take place, the distance function versions of the dilations and the erosions are still used. As stated in a previous section, for small dilations and erosions there is no time savings achieved by using the distance function versions of the dilation and erosion functions.

The new version of the chromosome detection algorithm is outlined by the following algorithm which is depicted in Figure 5.18:

Chromosome Detection Algorithm

(Delayed Threshold Version)

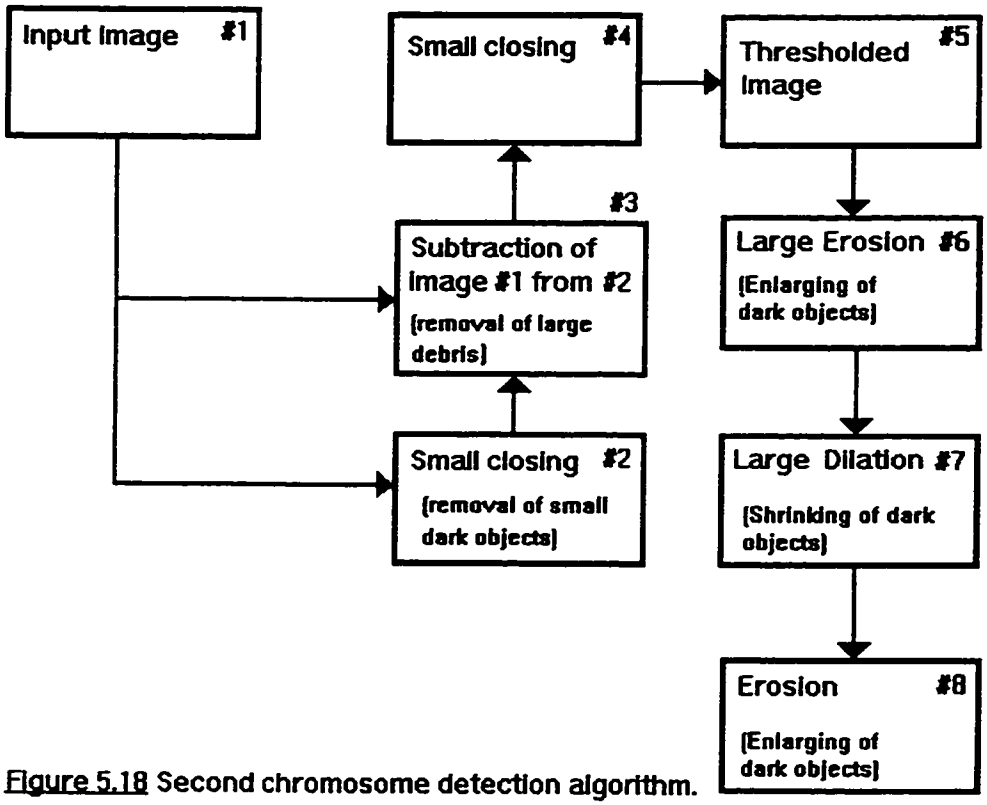


Figure 5.18 Second chromosome detection algorithm.

First the input image (#1) has a small grey scale closing performed on it to form image (#2). Next image (#1) is subtracted from image (#2) to produce image (#3). Since this is a grey scale subtraction the relative brightness of the background light level is not as critical to the functioning of the algorithm. This is because the closing that takes place will remove the small dark elements from the image while leaving the background grey level untouched. When the subtraction takes place only small light elements will survive. As was the case in the previous version of the algorithm, the picture is reversed so that what will be produced for image (#3) will be some small lighter picture elements on a very dark background. Again, for the purpose of this explanation, the image will be inverted.

Next a small opening is produced to remove the edge fragments that are a side effect of the above subtraction. This produces image (#4). Now the threshold takes place which produces image (#5). This threshold is less sensitive to the initial brightness of the original image since it is only calibrated to distinguish the difference between the grey level of the background and the foreground. This makes the algorithm less sensitive to the actual grey levels involved.

Now the algorithm is completed as in the original version by doing a large erosion to produce image (#6). This again clumps the small dark objects into a large picture element or blob. This image is then dilated to produce image (#7) which removes the small isolated picture elements. These large dilations and erosions again use the distance function as in the first version of the algorithm, since we are now dealing with a black and white image, which contains no grey levels.

Finally, a large erosion is performed to produced image (#8), which makes the final blob representing the chromosome spread similar to the actual size of the spread.

This concludes this discussion of the chromosome detection part of the program. Additional minor enhancements were made to the program, which relate mostly to the ease of user interaction with the program.

Chapter 6

Validation

6.1 Methodology

The above chromosome detection algorithm was validated by the use of several Receiver Operating Characteristic (R.O.C.) curves. In this chapter the R.O.C. curve results used to optimise the algorithm will be presented as well as the method used in the automatic acquisition of R.O.C. curve data.

R.O.C. curves are graphs contained in the unit square. The x-axis represents the probability that a False Positive will be detected, and the y-axis represents the probability that a True Positive will be detected. A R.O.C. curve is produces by varying one parameter and recording the results this has the detection rates. An ideal R.O.C. curve goes from the point (0, 0) to the point (0, 1) and then goes from the point (0, 1) to the point (1, 1). See the example R.O.C. curve in Figure 6.1. In this graph the x-axis represents the probability that a given spread in the sample will be a False Positive and the y-axis represents the probability that a given spread in the sample will be a True Positive.

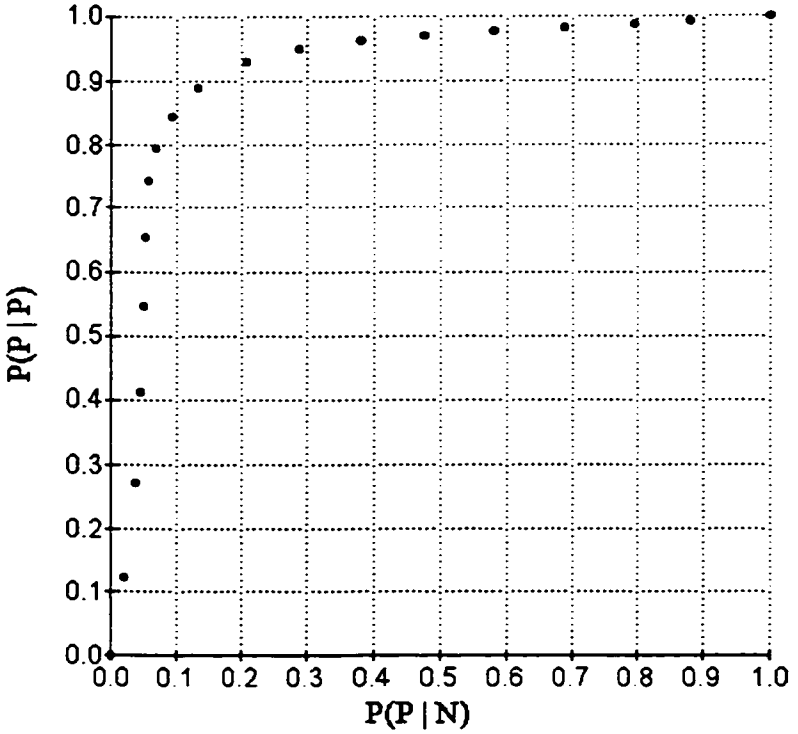


Figure 6.1: An example R.O.C. curve.

Not all R.O.C. curves are this well behaved, sometimes the curve will dip later on, but usually the curve goes up fairly steeply from the starting point at (0, 0) to a point close to the point (0, 1) and then it goes horizontally to the end point (1, 1). Real world R.O.C. curves never reach the "perfect" point (0, 1), but they can get close to that point.

All R.O.C. curves represent the variation of a parameter as it is adjusted. There is a trade-off between an increased yield of True Positives which also tends to produce an increased rate of False Positives and a decreased yield of these two items.

The idea is that we wish to maximise the area under the curve. The user of the R.O.C. curve may pick the point which he or she is comfortable with form the various points available in the graph. In our case if the user wishes a high yield of chromosome spreads and is willing to have a higher level of False Positives then a point high and more to the right on the graph will be chosen. However, if a very low rate of False Positives

with an associated lower rate of True Positives is desired then a point to the left of the graph and farther from the top of the graph will be chosen. If True Positives are found at a reduced rate then the overall time to locate a desirable number of True Positives will be extended. These trade-offs are up to the user of the program to decide.

One problem with the use of R.O.C. curves in our application is that they only work with one parameter at a time. We have five parameters that need to be adjusted and measured by R.O.C. curves. For two of these parameters R.O.C. curves do not need to be produced. The first parameter is parameter #1 (used to produce image #3 in Figure 5.17) which is the small closing used to remove the individual chromosome. This parameter does not need to be tested because the width of chromosomes is very consistent for all the images used in the image sets. The second parameter is parameter #2 (used to produce image #5 in Figure 5.17) which is the small closing used to remove the edge fragments and other very small debris after the subtraction. This parameter does not need to be tested because theses are again very consistent for all the images.

The parameters that will be tested are parameters #0, #3, and #4. Of these three parameters parameter#3 and parameter#4 are closely related because parameter #3 is the maximum inter-chromosome distance that is permitted and this will have an effect on parameter #4 which is the maximum size of a chromosome spread. Parameter #0 which is the initial threshold is independent of these other two parameters.

There are five primary parameters that are used in the functioning of the algorithm. These parameters are stored in a control array which allows them to be adjusted as needed by the user of the program. For the initial set-up of these parameters a good quality chromosome spread was found and used. The program provides a step by step process of viewing the images as they flow through the algorithm. Using this process some initial guesses for the parameters were set by the programmer.

As stated above there are five parameters used in the chromosome detection algorithm. The parameters that were not tested with R.O.C. curves were parameter #1

(small closing to remove small dark objects) and parameter #2 (small closing). Parameter #5 was not used for the first version of the chromosome detection algorithm. Parameter #0 is the width of a chromosome that is used to produce image (#3) which removes the chromosomes and other small picture elements. There is very little variability in this parameter and hence it did not need to be adjusted. Parameter #2 is used to produce image (#5) in which the small edge fragments are removed after the subtraction used to produce image (#4). Again this parameter does not change with different images since the edge fragments are very consistent in width.

6.2 Image Data Set

The image data was chosen from a typical slide produced by the method described in section 2.1.2. The slide was manually moved around and various images of chromosome spreads, as well as images of debris were captured for later processing by the computer. A total of 200 images were captured from two different slides.

The lighting level was manually set by the user of the program to be just below the saturation level of the video display. This is to say that the brightness was made as bright as possible without causing the details of the image to be lost.

Two sets of 100 images were used to test and evaluate the functioning of the algorithm. The first set of 100 images was called "TKS1Fxx" and was used to optimise the parameters of the algorithm and the second set was used as a check of the optimised algorithm to verify that the program works well with independent data. The second set of 100 test images was called "TKS2Fxx". Using this approach the accuracy of the algorithm can be verified and shown to be stable with independent data sets.

After all the images were captured each image was scored by the analyst. A "Gold Standard" was produced in which the location and number of each well formed chromosome spread was logged and each poorly formed chromosome spread was also

logged. This represents what the user of the program expects the computer to find when it process the images.

6.3 Automatic Acquisition of R.O.C. Curve Data for Plot Generation

The algorithm used to generate the data for the R.O.C. curves will now be described. The first step is to collect the images used for the tests. This usually consists of 100 images as described in the previous section. All the file names of the images are stored in a text file. The analyst then manually scores each image in this set and determines how many True Positives and True Negatives each image contains. If for example an image consists of only debris and no chromosome spreads then it is categorised as one True Negative. If an image has 3 poorly formed chromosome spreads (that cannot be used subsequently for aberration analysis) and additional debris then this image will be scored as having three True Negatives. Lastly, if an image contains two well formed chromosome spreads and one poorly formed chromosome spread then the image will have two True Positives and one True Negative.

This set of 100 images is then analysed by the computer to produce a set of analysed images and a text file containing the image file names and the number of chromosomes spreads in each image.

After this has happened the user of the program then goes back and reviews the results that the computer has made. Any incorrectly analysed images are recorded by the user. Next the user produces correctly analysed images, usually by manually adjusting the program parameters and then saving these new images over the incorrectly analysed images on the disk.

Next the user uses the program to construct a text file which is the "Gold Standard" that has the file names of all of perfectly analysed images. Adjustments need to be made to the number of True Negatives that a particular image contains.

Now the user uses the program to analyse the original set of images and stores them on the hard disk. To produce the R.O.C. data the computer compares the stored images for a particular test with the "Gold Standard". The algorithm to do this is as follows:

A correct image is loaded into buffer A. The number of blobs, indicating the correct number of chromosome spreads is then stored in the variable lCB (for Correct Blobs). The image in buffer A is then transferred to buffer B. The test image is then loaded into buffer A. The number of blobs that it contains is then stored in the variable lTB (for Total Blobs).

The two buffers are then ANDed together and the result is stored in buffer A. This buffer is then eroded a small amount to remove the possibility that two blobs that are in close proximity to each other in the two images and overlap somewhat do not produce a phantom chromosome spread in the ANDed image.

The number of blobs in this image is stored in the variable lFB (for found blobs). The number of stored True Negatives is stored in the variable lSN (for stored negatives). To get the final results for the number of True Positives (lTP), False Positives (lFP), False Negatives (lFN), and True Negatives (lTN) the following calculations are made:

```
lTP = min(lCB, lFB)
lFP = max(0, (lTB - lTP))
lFN = lCB - lTP
lTN = max(0, (lSN - lFP))
```

These results are then stored in a final output text file which also contains the names of the images. This file can be used to show the final results of the analysis of the chromosome spreads so that the user can then go back and check to see that the computer actually produced the correct results. A plot data file is also produced for each set of test

images, and as a particular parameter is changed the points for each plot are produced. This concluded this section on the automatic production of R.O.C. data. The formal procedure used by the user is listed in Appendix A.

6.4 Results

The results of the R.O.C. curve testing of the algorithm will now be described. There are three parameters that were optimised using R.O.C. curves. This first parameter to be optimised was parameter #0 (the threshold). This threshold is the value that is used to initially separate the foreground objects from the background objects. The next parameter that was optimise was the parameter #3 (large erosion used to produce image #6) which represents the inter-chromosome distances. A large value of this parameter indicates that chromosomes that are more distant from each other will be considered to be part of the same chromosome spread. A small value for this parameter indicated that individual chromosomes must be very close together in order to be considered part of the same chromosome spread. The third parameter to be optimised was parameter #4 (large dilation used to produce image #7). This parameter represents the overall size of a chromosome spread. If this parameter is large then only chromosome spreads that have a large area will be found by the algorithm. If this parameter is small then chromosome spreads that have a smaller area will be found by the algorithm.

All of the parameters used by the chromosome detection algorithm are stored in a control array.

In the process of optimising the three parameters the first parameter was optimised initially. This threshold parameter is largely independent of the other two parameters to be optimised. Next, parameter #3 was optimised. Then parameter #4 was optimised. After parameter #4 was optimised parameter #3 was again optimised in the light of the change made to parameter #4. The reason for this is the fact that parameters #3 and #4 are related

to each other. It was the case that the parameters did converge to stable values which indicates the stability of the overall algorithm.

As stated above several R.O.C. curves were used to optimise the algorithm. Each one of the R.O.C. curves will now be described in detail. As can be seen in Figure 6.2 the R.O.C. curve of the threshold, the graph goes up to a peak and then comes back down the same way to a point close to the starting point again. In all of the R.O.C. curves presented in this section the points (0, 0) and (1, 1) have been added to the graph to make the graph square when plotted by the spreadsheet software.

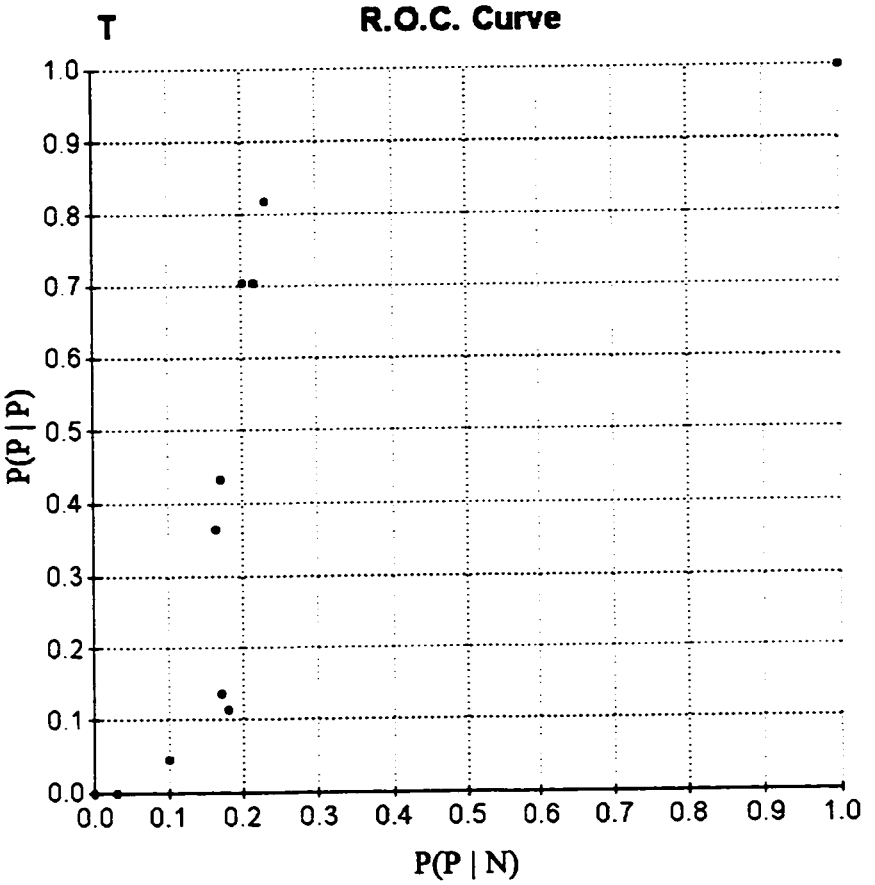


Figure 6.2: R.O.C. curve produced by varying the threshold.

T	R.O.C.	DATA
	x	y
110	0.0312	0
120	0.1015	0.0454
130	0.1796	0.1136
140	0.164	0.3636
145	0.2187	0.7045
150	0.2187	0.7045
153	0.2343	0.8181
155	0.2343	0.8181
157	0.2343	0.8181
160	0.2031	0.7045
170	0.1718	0.4318
180	0.1718	0.1363
	0	0
	1	1

Table 6.1: Data for R.O.C. curve in Figure 6.2.

The reason the curve goes up and then back down again is because when the threshold parameter is set to a low value (110) many large dark picture elements are present which are eliminated form the image when processed. Distinct chromosomes tend to lump together and form large objects that are rejected by the chromosome detection algorithm. As the threshold is increased smaller blobs appear in the image and more chromosome spreads start to appear. The peak value for the yields in chromosome detections was found when the threshold was set to the value 155. When the threshold is still further increased fewer and fewer picture elements are present in the image and hence fewer chromosome are detected. This explains the behaviour of this R.O.C. curve. The value 155 was chosen as the value to set the control array.

Next parameter 3 was tested with varying values to produce the R.O.C. curve in Figure 6.3.

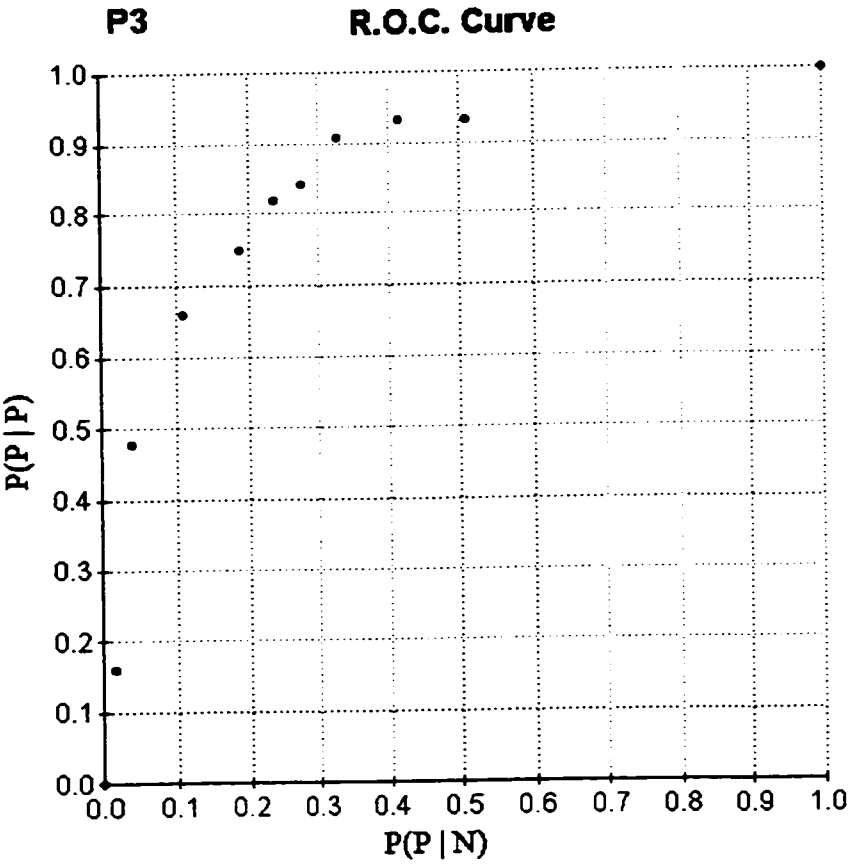


Figure 6.3: R.O.C. curve produced by varying parameter #3.

P3	R.O.C.	DATA
	x	y
7	0.0156	0.159
8	0.039	0.4772
9	0.1093	0.659
10	0.1875	0.75
11	0.2343	0.8181
12	0.2734	0.8409
13	0.3281	0.909
14	0.414	0.9318
15	0.5078	0.9318
	0	0
	1	1

Table 6.2: Data for R.O.C. curve in Figure 6.3.

This graph was produced with the threshold set to 155 and parameter #4 set to 20. This R.O.C. curve behaves normally compared to other R.O.C. curves in that it starts close to the point (0, 0), rises and then levels off heading toward the point (1, 1). The optimal point that was chosen was the point closest to the intersection of the line:

$y = -x + 1$

and the R.O.C. curve. The point that was chosen was the point (0.2343, 0.8181) at which parameter #3 had the value 11.

The next R.O.C. curve tested parameter #4. For this graph parameter #3 was set to 11 and the threshold was set to 155.

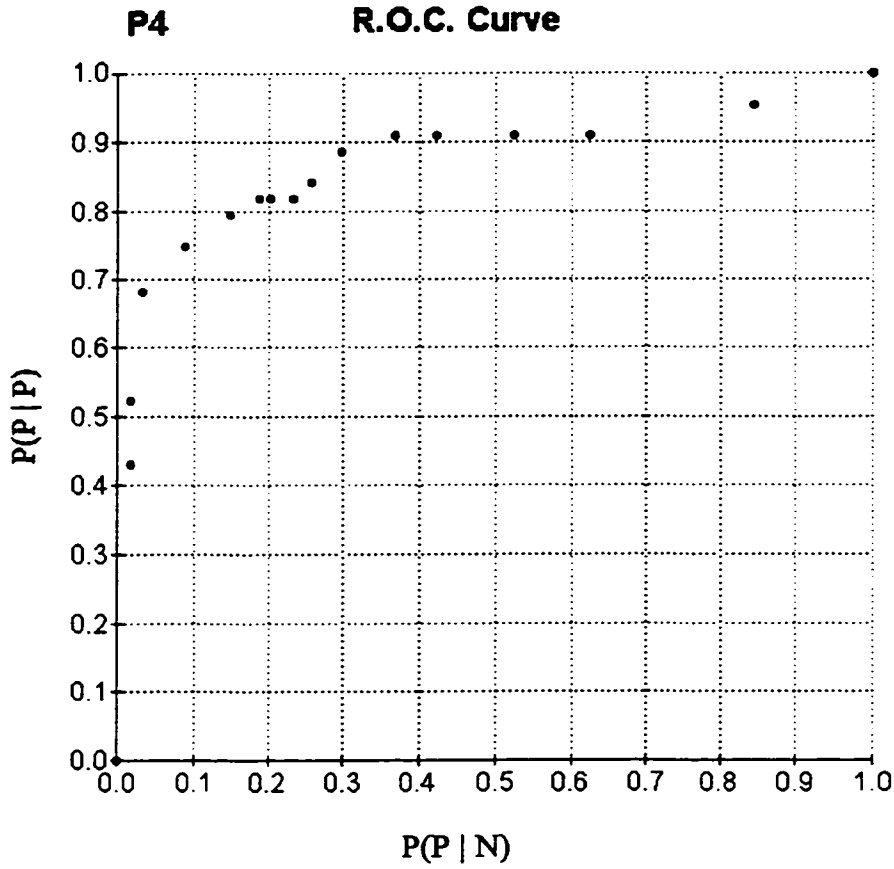


Figure 6.4: R.O.C. curve produced by varying parameter #4.

P4	R.O.C.	DATA
	x	y
13	0.8437	0.9545
14	0.625	0.909
15	0.5234	0.909
16	0.4218	0.909
17	0.3671	0.909
18	0.2968	0.8863
19	0.2578	0.8409
20	0.2343	0.8181
21	0.2031	0.8181
22	0.1875	0.8181
23	0.1484	0.7954
24	0.0859	0.75
25	0.0312	0.6818
26	0.0156	0.5227
27	0.0156	0.4318
	0	0
	1	1

Table 6.3: Data for R.O.C. curve in Figure 6.4.

Again this graph produced a good result. There are three points in a row which produces a slight shelf in the graph. The optimal point that was chosen was (0.1875, 0.8181) at which parameter #4 had the value 22.

We then went back and produced another R.O.C. curve depicted in Figure 6.5 which re-tested parameter #3.

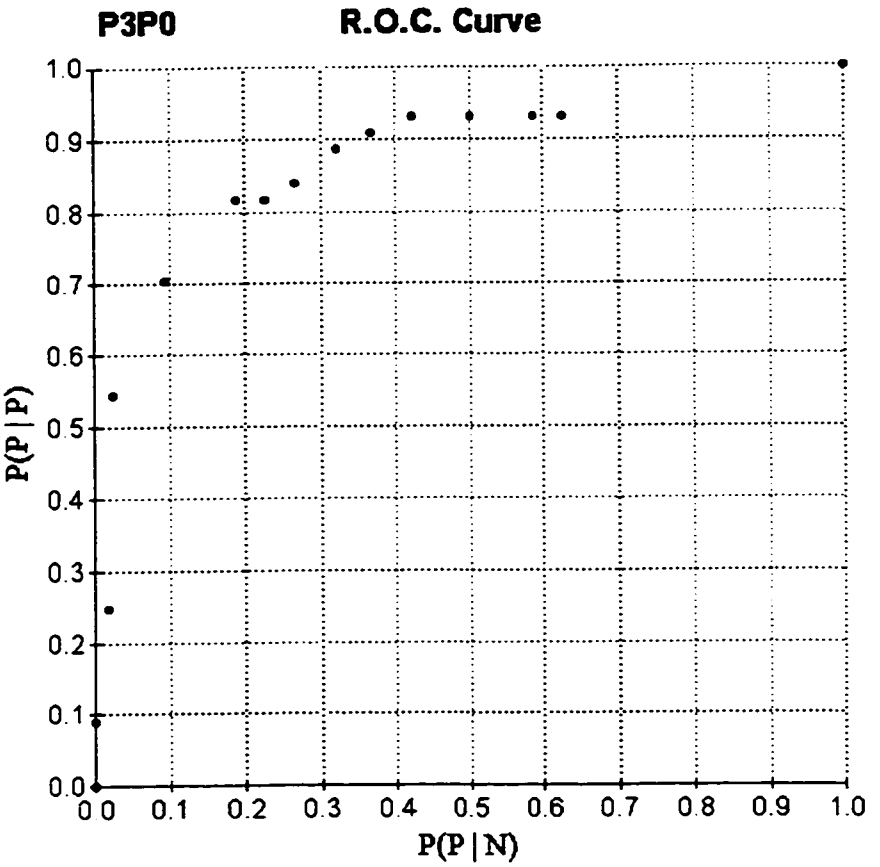


Figure 6.5: R.O.C. curve produced by varying parameter #3 again.

P3P0	R.O.C.	DATA
	x	y
7	0	0.0909
8	0.0156	0.25
9	0.0234	0.5454
10	0.0937	0.7045
11	0.1875	0.8181
12	0.2265	0.8181
13	0.2656	0.8409
14	0.3203	0.8863
15	0.3671	0.909
16	0.4218	0.9318
17	0.5	0.9318
18	0.5859	0.9318
19	0.625	0.9318
	0	0
	1	1

Table 6.4: Data for R.O.C. curve in Figure 6.5.

For this graph parameter #4 was set to 22 and the threshold was set to 155. After viewing the results of this graph the value of parameter #3 being set at 11 was still the optimal value, but the graph as a whole was superior to that in Figure 6.3. This was because the point (0.1875, 0.8181) was closer to the point (0, 1) than the optimal point in Figure 6.3 which was (0.2343, 8181).

After all of the parameters were optimised using the first set of 100 images, the second set of images was used to verify the high yields of good chromosome spreads. This led to the formation of the R.O.C. curve in Figure 6.6.

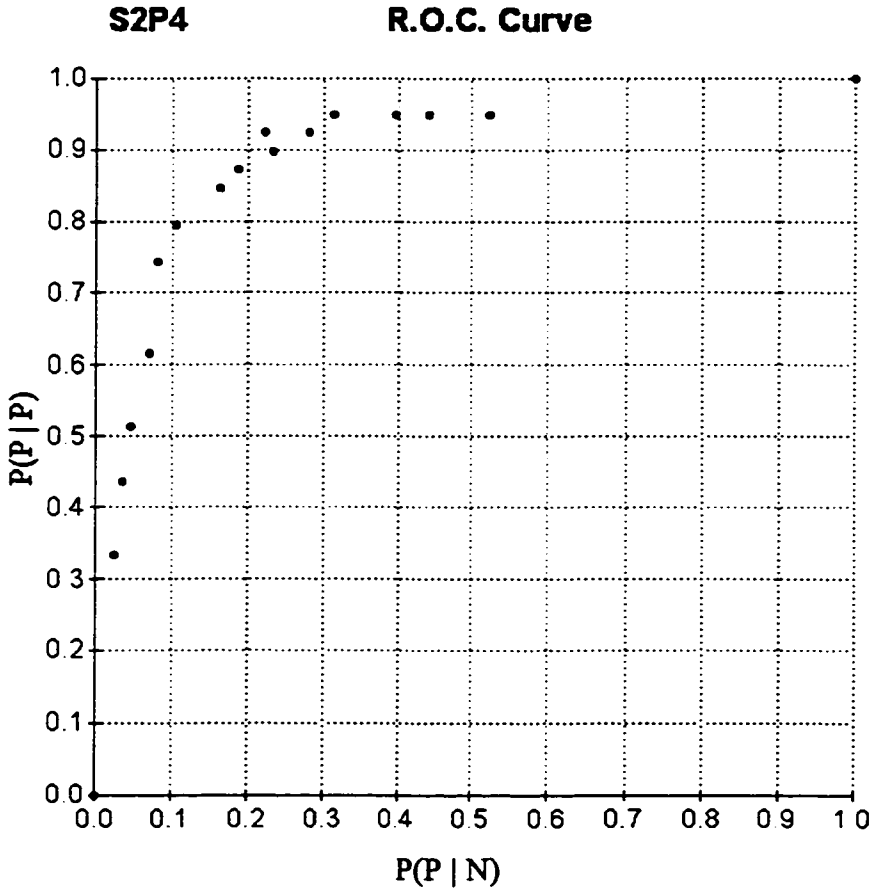


Figure 6.6: R.O.C. curve produced by varying parameter #4 using series #2.

S2P4	R.O.C.	DATA
	x	y
14	0.5232	0.9487
15	0.4418	0.9487
16	0.3953	0.9487
17	0.3139	0.9487
18	0.279	0.923
19	0.2209	0.923
20	0.2325	0.8974
21	0.186	0.8717
22	0.1627	0.8461
23	0.1046	0.7948
24	0.0813	0.7435
25	0.0697	0.6153
26	0.0465	0.5128
27	0.0348	0.4358
28	0.0232	0.3333
	0	0
	1	1

Table 6.5: Data for R.O.C. curve in Figure 6.6.

For this graph the threshold was set to 155, parameter #3 was set to 11, and parameter #4 was set to 22. The graph appears to quite good with the optimal point being (0.1627, 0.8461). There is a small hiccup at the point (0.2325, 0.8974), but otherwise the graph appears well formed. This graph confirms that the algorithm is effective in detecting chromosome spreads with varying data sets.

Now we will look at the R.O.C. curve produced by using the delayed threshold. It appears in Figure 6.7.

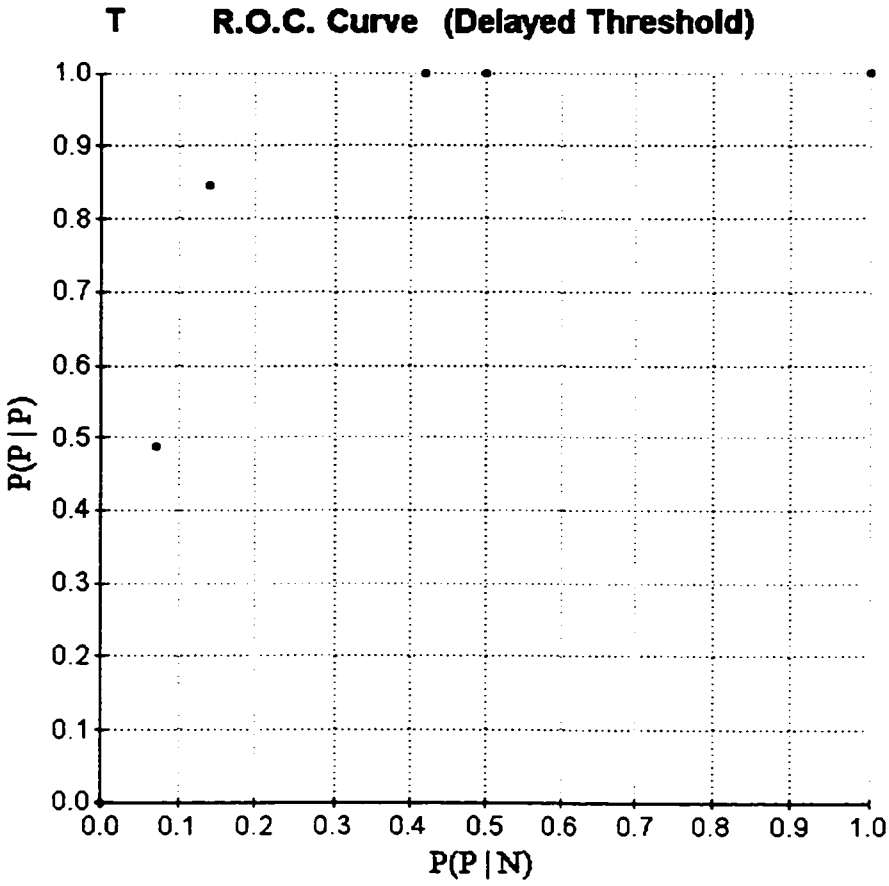


Figure 6.7: R.O.C. curve produced by varying the (delayed) threshold.

T	R.O.C.	DATA
	x	y
10	0.5	1
15	0.4186	1
20	0.1395	0.8461
25	0.0697	0.4871
30	0.0697	0.4871
35	0	0
40	0	0
45	0	0
	0	0
	1	1

Table 6.6: Data for R.O.C. curve in Figure 6.7.

This graph was produced by varying the threshold. Parameter #3 was set to 11 and parameter 4 was set to #22. The optimal point was (0.1395, 0.8461). This is the best point so far of all the R.O.C. curves and shows the potential of the alternative algorithm. Also it is interesting to observe that the point (0.4186, 1) has been achieved by the algorithm. This is the first time that algorithm has located 100% of the True Positives, however, this point is not acceptable due to the high False Positive rate. More work needs to be done to show that this new algorithm behaves well with varying data sets.

Of the three parameters that were tested the threshold parameter was the most sensitive to variation. If the magnifying power is changed then the parameters that need to be adjusted are parameter #1 (small closing), parameter #3 (large erosion), and parameter #4 (large dilation). The parameters tat do not need to be changed are parameter #0 (threshold) and parameter #2 (small closing).

This concludes the R.O.C. curve analysis for the two versions of the algorithm.

Chapter 7

Conclusion

In this conclusion we will now discuss what has been done. First, an automated system to detect good quality chromosome spreads was needed to reduce the amount of manual labour involved in determining radiation induced cell damage in patients.

This system consists of two subsystems. This first subsystems is the image acquisition subsystem and the second subsystem is the chromosome detection subsystem. This first subsystem consists of two components. The first component is the autofocusing algorithm to accurately bring the chromosome slide into focus. The second component is the platform movement algorithm to move the chromosome slide in the x-y plane. The second subsystem uses the chromosome detection algorithm which is used to locate the good quality chromosome spreads for later viewing by the analyst for manual aberration analysis. This thesis dealt with the first component of the image acquisition subsystem and the chromosome detection subsystem.

7.1 Autofocusing

The autofocusing component will now be summarised. After initially poor results, filtering techniques proved effective in producing accurate and repeatable in-focus images.

Three different autofocusing algorithms were tested to determine the one best suited for our purposes. To test the autofocus algorithms four different images were used ranging from images that had very few picture elements to images that contained a great

many picture elements. For each of the images a set images was produced ranging from very out of focus to in focus images. These sets of images were used to produce the graphs of the focus points for all three images. The third algorithm produced the best results. These results are characterised by a sharp distinct peak with broad smooth slopes on either side of that peak. From this sort of graph there is little opportunity for the algorithm to find a local peak that is not the correct global peak.

7.2 Chromosome Detection

The chromosome detection subsystem will now be summarised. Image processing techniques consisting primarily of dilations and erosions were combined to produce an algorithm that put markers on the location of valid chromosome spreads. The initial speed of the algorithm was enhanced by two different techniques. The first technique consisted of an early stop mechanism which returned a True Negative result when nothing remained in the image after the completion of certain image processing steps. The second technique was the use of a distance function which greatly improved the speed of large dilations and erosions.

After producing a faster chromosome detection algorithm, attention turned to the accuracy of that algorithm. In that regard several R.O.C. curves were produced which showed the sensitivity of the algorithm to various parameters. For this testing of the algorithm an initial set of test images was manually collected. Three of these parameters were then optimised. To produce data points for an R.O.C. curve a significant amount of manual work is normally required. To reduce the amount of time consuming and tedious work a mechanism was developed to automate the acquisition of R.O.C. curve data. The accuracy of the algorithm was demonstrated by the good quality of these R.O.C. curves. For verification purposes a second set of test images was used to produce an R.O.C. curve which replicated the good results obtained from the initial set of test images.

The sensitivity of the algorithm to the light levels then became an issue of concern. To address this issue a modified version of the algorithm was produced in which the threshold was delayed to a later stage in the algorithm. A preliminary R.O.C. curve was produced which showed even better results than the initial version of the algorithm.

The above algorithms used image processing functions arranged in new ways. Such an algorithm would have been impractical in the past due to the highly C.P.U. intensive nature of these image processing functions. The algorithms were designed to run on inexpensive commercially available systems that are reliable, simple to use and highly available.

More testing of these algorithms need to be carried out especially in regard to the delayed threshold version of the algorithm. Also trials of the system need to be performed by users of the system to get feed back on the user friendliness and ease of use of the system.

The goals of the system were to produce a fast and accurate autofocusing algorithm and chromosome detection algorithm. These goals were reached with good results. In terms of the accuracy of the algorithm, very good results were obtained which met or exceeded the initial requirements of the system. Speed enhancements of the system will naturally take place as hardware improves and has done so in the past two years.

The overall goal of the system was to prove that the techniques used in the system were sound and produced the desired results.

7.3 Future Work

The future work that needs to be done on the system will now be discussed. The speed of the chromosome detection algorithm needs to be improved. This can be done in several ways. The most obvious method is to buy another computer with a faster C.P.U. and memory bus. When this project began two years ago the state of the art Pentium computer was 100 MHz, today 300 MHz Pentium II computers are available. With the 75

MHz Pentium used in this thesis, it was estimated that the time to detect 1000 good quality chromosomes spreads would take from approximately 58 hours to 87 hours to complete. Since a Pentium II computer of the same clock speed is approximately twice as fast as a regular Pentium computer it is estimated that on a 300 MHz Pentium II, the computer would only take 7.25 to 10.9 hours to locate 1000 good quality chromosomes spreads. This would require an automated system to change the chromosome slides to work unattended over night. In this way the 1000 good quality chromosome spreads could be collected over night which would meet the initial operating requirements of the system.

Additionally, this thesis used Windows 3.1 which is a 16 bit operating system. Windows 95 and Windows NT are 32 bit operating systems and hence the data transfer rate is twice as fast again. The MIL libraries used in this Thesis are written in 16 bit code. This library now comes in a 32 bit version for Windows 95 and Windows NT. Hence these new versions of the image processing functions now operate faster on the same speed of computer due to the fact that the new version of imaging library uses the full available bus width. This would further reduce the time needed to locate 1000 good chromosome spreads needed for aberration analysis.

It is proposed that a preliminary delineation of the input image into regions of interest (R.O.I.) might be used to reduce the amount of the image that the image processing functions will operate on. This idea is that a grid be placed on the image of a certain granularity and that at the intersection of the vertices the value of the image be determined. If, for example, the top of the image only produces background values, while the bottom part of the image has a few foreground points then the top of the image could be ignored during the image processing steps. This idea has to be considered to see if the added overhead of doing the R.O.I. is worth the additional time in comparison to simply performing the image processing steps on the entire image. There may also be a reduction

in the ability of the algorithm to detect all of the available chromosome spreads if the R.O.I. misses all of the chromosomes in a given spread.

Finally, the image acquisition subsystem and the chromosome detection subsystem need to be integrated into one system. The autofocusing, platform movement, and the chromosome detection algorithms all need to be put together to form one system for the user. Additionally a good G.U.I. Windows interface should be added to the system to enhance the ease of use of the system. On line help and a user manual will also need to be produced. The system as it is has the autofocusing chromosome detection algorithms in one program. Additional code for the platform movement could easily be added along with a user interface to complete the system. Much hard work went into the design and testing of the system which produced good results. These good results obtained by the system indicate that a viable system is not too far off from now.

Bibliography

1. B. Batchelor and F. Waltz, "Interactive Image Processing for Machine Vision", London, England, Springer-Verlag, 1993.

2. R. Bayley, A. Carothers, X. Chen, S. Farrow, J. Gordon, L. Ji, J. Piper, D. Rutovitz, M. Stark, and N. Wald, Radiation dosimetery by automatic image analysis of dicentric chromosomes, Mutation Research, 253, pp. 223-235., 1991.

3. M. A. Bender, A. A. Awa, A. L. Brooks, H. J. Evans, P. G. Groer, L. G. Littlefield, C. Pereira, R. J. Preston, and B. W. Wachholz, Current status of cytogenetic procedures to detect and quantify previous exposures to radiation, Mutation Research, pp. 103-159, 1988.

4. P. Finnon, D. C. Lloyd, and A. A. Edwards, An assessment of the metaphase finding capability of the Cytoscan 110, Mutation Research, 164, pp. 101-108, 1986.

5. J. Graham, Automation of Routine Clinical Chromosome Analysis, Analytical and Quantitative Cytology and Histology, Volume 9 Number 5, pp. 383-390, October 1987.

6. F. C. A. Groen, I. T. Young, and G. Lighthart, A Comparison of Different Focus Functions for Use in Autofocus Algorithms, Cytometry, 6, pp. 81-91, 1985.

7. R. Jain, R. Kasturi, and B. G. Schunck, "Machine Vision", New York, St. Lewis, Toronto, McGraw-Hill Inc., 1995, Chapter 2, pp. 63-69.

8. L. Ji, Intelligent Splitting in the Chromosome Domain, Pattern Recognition, Volume 22 No. 5, pp. 519-532, 1989.

9. B. John and K. R. Lewis, Somatic Cell Division, Oxford Biology Readers, London England, pp. 1-14, 1978.

10. F. Johnson and J. McLean, Evaluation of a Metaphase Chromosome Finder - Potential Application to Chromosome-based Radiation Dosimetry, Micron Volume 26, No. 5, pp. 489-492, 1995.

11. B. H. Mayall, J. D. Tucker, M. L. Christensen, L. J. van Vliet, and I. T. Young, Experience With the Athena Semi-Automated Karyotyping System, Cytometry, Volume 11, pp. 59-72, 1990.

12. Microsoft (R) Encarta, "Blood," Copyright (c) 1994 Microsoft Corporation, Copyright (c) 1994 Funk & Wagnall's Corporation.

13. R. W. Parry, H. Bassow, P. Merrill, and R. Tellefsen, "Chemistry Experimental Foundations", Englewoods Cliffs, New Jersey, Prentice-Hall Inc., Copyright 1982, Chapter 20, pp. 558-583.

14. J. Piper, E. Granum, D. Rutovitz, and H. Ruttledge, Automation of Chromosome Analysis, Signal Processing, Volume 2, No. 3, pp. 203-221, July, 1980.

15. J. Piper and C. Lundsteen, Human chromosome analyses by machine, TIG, Volume 3 no. 11, pp. 309-313, November 1987.

16. J. Piper and D. Rutovitz, A parallel processor implementation of a chromosome analysis system, Pattern Recognition Letters, Volume 4, pp. 397-404, October 1986.

17. D. M. Prescott, The Reproduction of Eukaryotic Cells, Carolina Biology Readers, Edited by J. J. Head, pp. 1-16.

18. S. K., Reed, Pattern recognition and categorisation. Cognitive Psychology, 3, pp. 382-407, 1972.

19. J. Serra, Image Analysis and Mathematical Morphology, Academic Press, London, 1982.

20. L. J. van Vliet, I. T. Young, and B. H. Mayall, The Athena Semi-Automated Karyotyping System, Cytometry, Volume 11, pp. 51-58, 1990.

Appendix A

Procedure to produce automatic data for R.O.C. curves

STEP A

1. Collect 100 images and store them into numbered files starting from "image00.TIF" to "image99.TIF", where image can be any file name (as long as the total file name is at most 8 characters plus a 3 character extension).

2. Create the "`AAAA.TXT`" file containing the names of all the image files.

This is done as follows:

(i) Go to the DOS prompt in the directory that contains the 100 image files.

(ii) type: `DIR *.TIF /B > AAAA.TXT`

(iii) And press "Enter".

(iv) type: `EDIT AAAA.TXT`

(v) And press "Enter".

(vi) type: `100`

(vii) And press "Enter". This will add "`100`" as the first line of the text file.

(viii) Save the file and return to the DOS prompt.

(ix) type: `EXIT`

3. Now run the "GETSPRED" program in the same directory as the files.

4. Type `10` and press "Enter".

5. After the image files have been processed enter option 12 to view the results. Press "Enter" again and then use the "+" key to move forward and the "−" key to move backward when viewing the images. Then go to STEP B.

STEP B

If you find an image that has the correct number of chromosome spreads then you do the following:

{Remember what the image number is that you are looking at. For example TKS2F37.TIF in image number 37.}

1. Type the letter 'Q' or 'q'.
2. Press "Enter".
3. Type 5 and press "Enter".
4. Press "Enter".
5. Press "Enter".
6. Type 3 and press "Enter".
7. Type the letter 'C' or 'c' and press "Enter".
8. Press "Enter".

{Wait for file to save.}

9. Press "Enter".
10. Type 12 and press "Enter".
11. Press "Enter".
12. Type the letter 'R' or 'r'.
13. Type the number of the image that you were looking at in step 1.
14. Continue on in the process until you reach the end of the set of images.
15. If an image does not come out as expected then do the following:

    (i) Save the image as "A".

(ii) Try changing a parameter so that the desired result appears.

(iii) If so save the image with the appropriate name.

(iv) If not it may be necessary to manually colour the desired area in as white on an all black image to get the desired result.

(v) Once successful save the image with the appropriate name.

(vi) Go to step 10 until all of the images have been processed.

STEP C

The next thing that you need to do is produce the "CCCC.TXT" text file. This file will contain the "Gold Standard for the set of 100 images that you just complete in STEP B. To create this file do the following steps:

1. From the main menu type 14 and press "Enter".
2. Type the letter 'Y' or 'y' and press "Enter".
3. Press "Enter".
4. Press "Enter".

{Wait for the computer to finished processing the files.}

5. Press "Enter".
6. Now open a DOS box and edit the CCCC.TXT file with EDIT.
7. Change the number of TN to equal the correct number of True Negatives that the user has observed.
8. Save CCCC.TXT and return to the program.
9. Verify the results of the CCCC.TXT file be going back and reviewing the results.

STEP D

Once a set of test image results has been produced for a given parameter, you do the following. From the main menu you type:

1. Type 14 and press "Enter".

2. Press "Enter".

3. Type the letter 'Y' or 'y' and press "Enter".

4. Press "Enter".

5. Press "Enter".

6. Press "Enter".

7. Press "Enter".

{Wait for the computer to beep to signify that the process has finished.}

8. Press "Enter".

9. Change a parameter of interest and then produce a new set of test images using that new parameter or if enough runs have been done go to step 11.

10. Go back to step 1.

11. All of the data results are now in the file called "PLOTDATA.TXT". Use Microsoft Excel and that data file to produce the ROC curve plots.

Appendix B

Program Menus

In this appendix the menus in the program will be described. The program contains a main menu of the most frequently used functions and a sub menu containing the less frequently used functions. Some of these functions ask the user for input and some do not. When the program is first started the parameters that are stored in the GETSPRED.INI file are displayed if the GETSPRED.INI file exists. The user then presses the "Enter" key and then the main menu is displayed as follows:

```
#####################################
# Please Enter your choice: (1-17) #
#####################################
# 1. Display Buffer A. [M == x200] #
# 2. Load an image into Buffer A.  #
# 3. Save Buffer A into a file.    #
# 4. Locate chromosome spread(s)   #
#    in Buffer A.   [VTKS1F01.TIF] #
# 5. Dilate Buffer A.(Dist. Funct.)#
# 6. Erode Buffer A. (Dist. Funct.)#
# 7. TEMP: Fix edges of Buffer A.  #
# 8. Go to the Sub Menu. [T == 155]#
# 9. Quit Program.                 #
#10. Automatic Mode.               #
#11. Display part of Buffer A.     #
#12. Automatically view images.    #
#13. Change a parameter.           #
#14. Calculate TP, FP, TN, and FN  #
#    for a set of verified images. #
#15. Set Buffer A to a grey level. #
#16. Swap Buffer A and Buffer B.   #
#17. Invert Buffer A.              #
#####################################

Choice>
```

Each option of the main menu will now be described. The first menu option is as follows:

```
# 1. Display Buffer A. [M == x200] #
```

This menu item causes the Buffer A image to be displayed on the screen. By doing so the user can see all of the effects of the image processing functions on Buffer A. The [M == x200] display show what magnification power the parameters are set to. This can be changed in the sum menu to x400.

The next menu item is:

```
# 2. Load an image into Buffer A.  #
```

This menu item allows the user to load an image stored in a disk file for display in Buffer A.

The next menu item is:

```
# 3. Save Buffer A into a file.    #
```

This menu item allows the user to save the image displayed in Buffer A to a disk file. If the file already exists then the user will be asked if they wish to overwrite the old file.
The next menu item is:

```
# 4. Locate chromosome spread(s)   #
#    in Buffer A.   [VTKS1F01.TIF] #
```

This menu item will perform the algorithm to locate the chromosome spreads. If the user wishes the algorithm can be stepped through operation by operation so that the effects of each operation can be seen. This will allow the user to determine if any parameters need to be adjusted when calibrating the algorithm.

The area between the square brackets contains the file name of the last image loaded.

The next menu item is:

```
# 5. Dilate Buffer A.(Dist. Funct.)#
```

This menu item allows the user to perform a black and white (no grey levels) dilation on the image displayed in Buffer A. The previous default value of this dilation is suggested, but a new value can be specified. The speed difference of this dilation can be compared to the normal dilation performed sub menu item 5. The advantage shows up for large dilations.

If the image has a white foreground and a black background then the function will produce exactly the same output as the normal dilation function. However, if the image has a black foreground and a white background then the function will have a side effect of producing a thin black line on the top and the bottom of the image. This is a minor problem which does not affect the operation of the chromosome detection algorithm since the algorithm normally uses a white foreground and a black background.

The next menu item is:

```
# 6. Erode Buffer A. (Dist. Funct.)#
```

This menu item allows the user to perform a black and white (no grey levels) erosion on the image displayed in Buffer A. The previous default value of this erosion is suggested, but a new value can be specified. Again, the speed difference of this erosion can be compared to the normal erosion performed sub menu item 6. The advantage shows up for large erosions.

If the image has a white foreground and a black background then the function will produce exactly the same output as the normal erosion function. However, if the image has a black foreground and a white background then the function will have a side effect of producing a black line on all of the edges of the image which is the same thickness as the erosion. This is a larger problem than that produced in the dilation, but again it does not affect the operation of the chromosome detection algorithm since the algorithm normally uses a white foreground and a black background.

The next menu item is:

```
# 7. TEMP: Fix edges of Buffer A.  #
```

This menu item was added to the program to fix a minor problem in the images captured by the C.C.D. camera. The captured images had a black line along the top and right part of the image. When the image was processed the early stop mechanism did not work because of these black lines. To fix this problem the above function was written to put white lines over the black lines and thereby remedy the problem. In automatic mode in menu item 10 this is done each time an image is loaded from the disk.

The next menu item is:

```
# 8. Go to the Sub Menu. [T == 155]#
```

This causes the sub menu to be displayed. The menu items in the sub menu will be described after the main menu items have been described. The '[T == 155]' in the display indicates the current threshold value.

The next menu item is:

```
# 9. Quit Program.                    #
```

This menu item allows the user to quit the program. A conformation is request by the user. The Buffer A image is closed and all the image buffers are cleaned up.

The next menu item is:

```
#10. Automatic Mode.                  #
```

This menu item asks the user for the text file name of the input list of image files names. This file contains the number of input images and the file names of each of those images. This file is typically called "AAAA.TXT". The user is then asked for the output file name to store the number and the names of the processed images. This file name is typically called "VBBB.TXT".

The next menu item is:

```
#11. Display part of Buffer A.      #
```

This menu item allows the user to zoom in on a particular part of Buffer A to see the actual values of each pixel. A small region is displayed and the user can see the effects of the previous image processing function in great detail.

The next menu item is:

```
#12. Automatically view images.     #
```

This menu item allows the user to view all of the images in a list of image file names stored in a text file. This allows the user to scan through every image and see the effects of the chromosome detection algorithm on those images. The user can view the next, previous, and a user specified image. Additionally, dilations and erosions can be done on the current image and the image can be saved to a new name while viewing the images without the need to return to the main menu.

The next menu item is:

```
#13. Change a parameter.            #
```

This menu item allows the user to change any of the parameters that affect the operation of the chromosome detection algorithm. These changes will only be in effect for the running of the program unless they are saved to the GETSPRED.INI file from sub menu item 10.

The next menu item is:

```
#14. Calculate TP, FP, TN, and FN  #
#    for a set of verified images. #
```

This menu item calculates the True Positives, False Positives, True Negatives, and False Negatives for a set of processed images.

The next menu item is:

```
#15. Set Buffer A to a grey level. #
```

This menu item sets the grey level of Buffer A to a user specified value. This value can range from 0 (Black) to 255 (White) and any grey level in between.

The next menu item is:

```
#16. Swap Buffer A and Buffer B.   #
```

This menu item allows the user to swap Buffer A and Buffer B so that the user can compare the two images. Buffer C is used as an intermediate image buffer during the swap, so its contents are over written.

Finally, the last main menu item is:

```
#17. Invert Buffer A.              #
```

This menu item performs a binary inversion of Buffer A. It does not work for grey levels.

As stated above if main menu option 8 is selected then the sub menu is displayed as follows:

```
************************************
* Please Enter your choice: (1-12) *
************************************
* 1. Perform gradient on part of   *
*    Buffer A.          [M == x200] *
* 2. Change Stop At location.      *
* 3. Automatically calculate the   *
*    gradient of a set of images in*
*    a file and store the results. *
* 4. Get and/or enter the value of *
*    a pixel.                      *
* 5. Dilate Buffer A.              *
* 6. Erode Buffer A.               *
* 7. Reset parameters for magnifi- *
*    cation of x200.               *
* 8. Reset parameters for magnifi- *
*    cation of x400.               *
* 9. Go to the Main Menu.          *
*10. Save options to GETSPRED.INI. *
*11. Add (or subtract) a value to  *
*    every pixel of buffer A.      *
*12. Draw a rectangle on buffer A. *
************************************

uiSubChoice>
```

Each option of the sub main menu will now be described. The first sub menu option is as follows:

```
* 1. Perform gradient on part of   *
*    Buffer A.          [M == x200] *
```

This menu item performs a gradient on part of Buffer A. This gradient function is also used for the autofocusing, but here it is just being used for testing purposes.

The next sub menu item is:

```
* 2. Change Stop At location.      *
```

This option changes the stop point in the autofocus algorithm. This allows the user to see the image at the end of any particular step in the algorithm.

The next sub menu item is:

```
* 3. Automatically calculate the   *
*    gradient of a set of images in*
*    a file and store the results. *
```

This menu item is used for the autofocusing of the microscope lens. A set of images which range from very out of focus to in focus are scanned by the program and the most in focus image is determined. In the final version of the program the images will be obtained directly from the camera rather from stored files.

The next sub menu item is:

```
* 4. Get and/or enter the value of *
*    a pixel.                      *
```

This menu item allows the user to display and optionally set the value of any pixel in the image. This is used for diagnostic purposes.

The next sub menu item is:

```
* 5. Dilate Buffer A.              *
```

This menu item implements the regular dilation function that is built into the MIL (Matrox Imaging library). It is fast for small dilations, but very slow for large ones. It does work correctly for any image, regardless of the foreground and background being any combination of black or white.

The next sub menu item is:

```
* 6. Erode Buffer A.                *
```

This menu item implements the regular erosion function that is built into the MIL library. It is fast for small erosion, but very slow for large ones. It does work correctly for any image, regardless of the foreground and background being any combination of black or white.

The next sub menu item is:

```
* 7. Reset parameters for magnifi- *
*    cation of x200.                *
```

This menu item resets the chromosome detection algorithm's parameters to the default values for a magnification of x200.

The next sub menu item is:

```
* 8. Reset parameters for magnifi- *
*    cation of x400.                *
```

This menu item resets the chromosome detection algorithm's parameters to the default values for a magnification of x400.

The next sub menu item is:

```
* 9. Go to the Main Menu.           *
```

This causes the main menu to be displayed again.

The next sub menu item is:

```
*10. Save options to GETSPRED.INI. *
```

This menu item saves the parameters of the chromosome detection algorithm to the GETSPRED.INI file. These parameters will be read in again the next time the program is run.

The next sub menu item is:

```
*11. Add (or subtract) a value to  *
*    every pixel of buffer A.      *
```

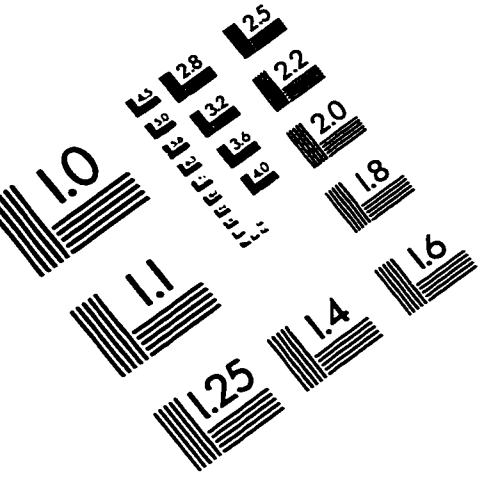
This menu item allows the user to adjust the brightness of the image in Buffer A. If a small value is added to the value of every pixel in the image then the image will appear brighter. If a small value is subtracted from the value of every pixel in the image then the image will appear darker.

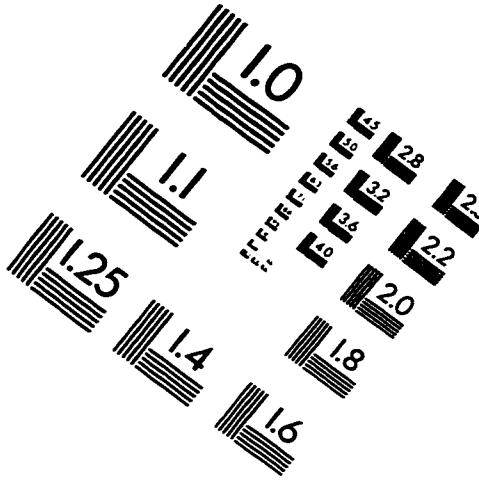
Finally, the last sub menu item is:

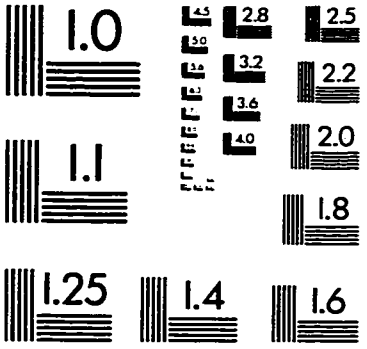
```
*12. Draw a rectangle on buffer A. *
```

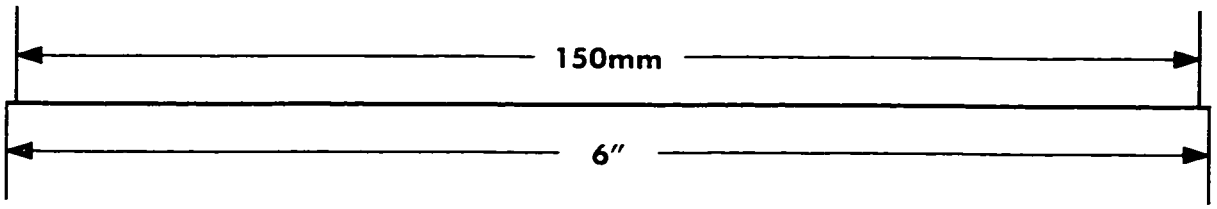
This menu item draws a rectangle of any grey level and of any size and location on the image in Buffer A. Again this function is used for diagnostic purposes and to manually mark the location of a chromosome spread.

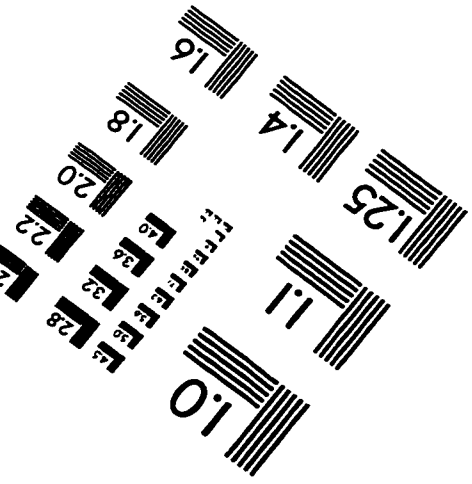
IMAGE EVALUATION TEST TARGET (QA-3)

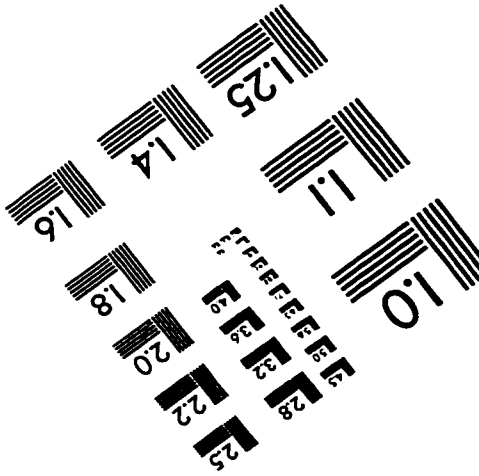












APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989