LIST OF PUBLICATIONS

Paper Published Author in Related Areas

1. Journal of Solid State Science and Technology Letters, Volume 9, No 1, September 2002, page 316-321.

   Image Processing For Nuclear Track,

   Naharudin Mustafa, Burhanuddin Kamaluddin

2. Proceeding of First Malaysia-France Regional Workshop on Image Processing in Vision Systems and Multimedia Communication, Sarawak, Malaysia, 21st – 22nd April 2003.

   A Mathematical Morphology Approach to the Printed Circuit Board in its Pain Parts Problem,

   Naharudin Mustafa, Burhanuddin Kamaluddin

ABSTRACT

The main aim of this work is to establish a technique to prepare the image of nuclear track detector in the form of thin film, LR 115 Type 2. This research also aims to develop the image processing scripts using SDC Morphology toolbox and Image Processing toolbox in conjunction with MATLAB as a programming platform. The scripts were applied to find the segmentation limit of over-lapping nuclear tracks, nuclear tracks counting and granulometry (area and diameter determination) in the images obtained. We finally compare the toolboxes capability in processing the image. The research found that the SDC Morphology toolbox is more precise than Image Processing toolbox when the exposed time of radiation to track detector is less than 30 hours. Then, the segmented over-lapping objects were obtained if the over-lapping area is less than the segmentation limit of over-lapping area.

Keywords: Morphology Image Processing; MATLAB; SDC Morphology Toolbox

ABSTRAK

Matlamat utama penyelidikan ini adalah untuk menentukan teknik penyediaan imej pengesan jejakan nuklear menggunakan filem nipis LR 115 Type 2. Penyelidikan ini juga bertujuan untuk membangunkan skrip pemprosesan imej dengan menggunakan perisian SDC Morphology dan perisian Image Processing toolbox bersama bersama MATLAB sebagai perisian utama. Skrip-skrip ini telah diaplikasikan untuk menentukan had pembahagian jejakan nuklear yang bertindan, pengiraan bilangan dan penentuan luas jejakan nuklear pada paparan dalam imej. Hasil daripada ini, penyelidikan ini telah membandingkan kebolehan kedua-dua perisian pemprosesan imej. Penyelidikan ini mendapati perisian SDC Morphology lebih tepat berbanding perisian pemprosesan imej semasa dedahan daripada sinaran radioaktif ke atas pengesan jejakan yang kurang daripada 30 jam. Daripada ini, pembahagian objek bertindan berjaya diperolehi jika luas kawasan bertindan adalah kurang daripada had pembahagian luas kawasan bertindan.

ACKNOWLEDGEMENT

Pertama sekali syukur Alhamdulillah ke hadrat Ilahi dengan limpah kurnia dan akhirnya saya diizinkanNya untuk saya menyiapkan tesis MSc. ini.

Saya ingin mengucapkan setinggi-tinggi penghargaan dan terima kasih kepada Prof. Dr. Zainol Abidin Ibrahim selaku penyelia tesis MSc. saya dan juga tidak lupa kepada Allahyarham Dr. Burhanuddin Kamaluddin sebagai penyelia tesis MSc. saya yang asal. Bagi saya mereka berdualah diibaratkan sebagai tulang belakang kepada saya untuk menyiapkan tesis MSc. ini dengan jayanya.

Ribuan terima kasih saya tujukan kepada kedua-dua ibu bapa saya iaitu Allahyarham Mustafa Muhammed dan Puan Siti Minah Abdullah dan juga adik-beradik saya atas galakan, sokongan dan bantuan yang diberikan kepada saya dalam sepanjang pengajian ini. Terima kasih seikhlas hati yang istimewa kepada isteri saya, Dr. Izlina Supaa’t dan anak saya, Athirah Amani Naharudin.

Akhir sekali, saya ingin mengucapkan terima kasih kepada rakan saya iaitu Sidiq Mohamad Khidzir dan rakan-rakan seperjuang yang turut membantu dan berkongsi maklumat sepanjang proses menyiapkan tesis MSc. ini. Tanpa bantuan mereka semua, adalah sukar untuk saya menyiapkan tesis ini.

List of Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

1.1 Chapter Overview

Image analysis is not just important for the applications in science and engineering. It is also very significant for the clarification and understanding of the images taken by satellites, medical imaging tools such as X-ray tomography, magnetic resonance imaging (MRI) and microscopy; which involves image processing, some being much more sophisticated than others (Michielsen & Raedt, 2001). In microscope images of materials such as polymer mixtures and ceramics, the main purpose of image analysis is to provide a quantitative characterisation of the shape, structure and connectivity of the objects image. The purpose of this study is to investigate an easy-to-use and versatile method to compute the morphological properties of such images.

1.2 Research Background

Image processing is a technique to improve and enhance the raw image. There are two methods which are the Digital Image Processing and the Analog Image Processing. Digital Image Processing (DIP) is the method using computer to analyse and enhance the image (Gonzalez & Woods, 2002). It was developed at 1960s at the Massachusetts Institute of Technology, Bell Laboratories, and the University of Maryland. The processing uses image as a two-dimensional signal or input data and apply to it the standard techniques of image processing in order to get the specific output (Rosenfeld, 1969).

After three decades, the techniques of DIP have been developed by researchers in order to get the best output required. These are image representation, image enhancement, image restoration, image analysis, image reconstruction and many more. The developments of these DIP techniques actually are based on using computer algorithms, where the noise and signal distortions can be avoided. The examples of the computer algorithm application techniques in DIP are; Feature Detection, Microscope Image Processing, Remote Sensing and Morphological Image Processing.

Image processing algorithm in Morphological Image Processing was developed based on the Mathematical Morphology (MM). MM is not just a theory but also a technique for the analysis and processing of geometrical structures, derived from the set theory, lattice theory, topology, and random functions. Generally MM is applied to digital images, and can be engaged on graphs, surface meshes, solids, and other spatial structures (Serra, 1982; Serra, 1988; Dougherty, 1992).

Image Processing Toolbox is a tool in MATLAB program. It is a set of reference-standard algorithms including MIP operator and graphical tools for image processing, analysis, visualization, and algorithm development. Application of the MATLAB program and image processing toolbox can be used in detecting an object in an image if the object has sufficient contrast from the background. As an example, the cancer cell can be detected by applying edge detection and basic morphology operations (MathWorks, 2011).

Besides Image Processing Toolbox, there is a toolbox which is known as SDC[1] Morphology Toolbox for MATLAB. It is a collection Morphological Image Processing that provides morphological tools that can be applied to image segmentation, non-linear filtering, and pattern recognition and image analysis (SDC, 2011).

MIP is also used in processing nuclear track detector image. This processing is able to present granulometry and nuclear track counting results (Eghan, Buah-Bassuah, & Oppon, 2007), which is it can be used as reference in this work.

[1] SDC Morphology Toolbox is a product from a company named SDC Information System. There's no further information about the name of 'SDC' abbreviation.

1.3 Mathematical Morphology (MM)

Mathematical Morphology is a nonlinear image processing technique that is engaged in the properties analysis of materials and material textures. It is a set of mathematical derivations that can be used as texture analysis. It has been developed almost 4 decades ago and produces many of the image processing applications. MM signifies a special method that is supported by classical linear processing. Furthermore, it improves the application of various mathematical developments of the processing and analysis of images. These areas include medical image segmentation, non-linear statistics, logic, geometry, geometrical probability, topology and various algebraic systems such as lattice and group theories. The disciplines that have been used in the classical image processing techniques normally come into view within the context of mathematical morphology, based on its development and application. There have been vast growths of interest in mathematical morphology over the past few years, which were developed by Georges Matheron (Matheron, 1975). As a result, many research approaches ranging from noise analysis to image analysis are currently being explored. Even though it was developed based on binary images, morphological applications are expanding into different fields beyond the image analysis field. For instance, lattice image processing (Maragos, 2005) and symmetry groups (Roerdink, 1993).

The application of Mathematical Morphology in image processing is a concept for operating shapes of objects in images quantitatively. All morphological methods are reduced to only two basic operations called *erosion* and *dilation*. Logical Operations also is one of the basic morphological methods (Refer to Chapter 2 for detailed information on the subject of Mathematical Morphology).

1.4 Introduction to Nuclear Track

The basics of the nuclear track operations are based on the fact that a heavy charge particle will cause extensive ionisation of the material if it passes through a medium. Some nuclear particles will be ionised along the path it takes. The path that has been made by a nuclear particle is a zone enriched area with free chemical radicals and other chemical elements. This damaged zone is the track of a nuclear, which is known as **latent track** (Nikezic & Yu, 2004).

If a piece of material or thin film that is sensitive to radioactivity is exposed to radioactive source for a certain period of time, the nuclear particles that passes through it will ionise the film and create the latent track. Theoretically, the number of this nuclear latent track produced from a radioactive source is proportional to time. However, the thin film containing latent tracks needs to be exposed to some chemically aggressive solution, where chemical reaction will occur extensively along the latent tracks. This chemical solution is required to form the 'track' of the particle, which may be seen under an optical microscope. This procedure is known as detector etching or track visualization.

The thin film that is used in the nuclear track processing is called track detector. They are three common types of track detectors that are used in nuclear track. They are CR-39 detector which is based on polyallyldiglycol carbonate, cellulose nitrate, and the well-known LR 115, which will be used in these studies.

In order to continue with the track analysis and image processing procedure, the nuclear track detector requires conversion to an image. As an image, granulometry analysis and nuclear tracks counting can be succeeded (Eghan, Buah-Bassuah, & Oppon, 2007).

1.5 Motivation

We know there are many image processing problems that can be solved using certain techniques, which is suitable with their problems characteristic. We also know that image processing can be run or processed using many types of software. For that reason, the main motivation for conducting this research is to find out the capability of the combination between MATLAB program and also a MATLAB toolbox developed by third party (SDC Morphology Toolbox) in processing the images.

There are various problems in the field of image processing, which can be solved through certain techniques and algorithms based on the types of problems. Based on this known problems, we would like to apply an image processing method known as the Morphological Image Processing in order to find more efficient approaches or techniques to deal with the problems. In addition, the SDC Morphology Toolbox was built based on Morphological Image Processing principle.

1.6 The Purpose and Objective of the Research

The main goal of this research is to study Morphology Image Processing (MIP) and its application to nuclear track images by comparing the results obtained from using Image Processing toolbox and SDC Morphology toolbox application. This thesis has the following objectives;

i. To study the capability of SDC Morphology Toolbox as a morphology image processing tool in conjunction with the MATLAB platform, regardless whether the image is real or artificial (Pritchard, 2002; Marshall, 1992; Quadrades & Sacristán, 2001).

ii. To use these toolboxes to perform dot counting (nuclear track counting) profile at arbitrarily set areas and compare with result from simple particle (nuclear track) counting simulation. We also use these toolboxes to perform granulometry (area measurement) to obtain the average of the mean area profile of exposed nuclear track over various exposure times and thus obtain the diameter of the mean area.

iii. To study how the toolboxes perform nuclear track counting on images of nuclear tracks in which some of the tracks are over-lapping with each other. We then separate the over-lapping tracks based on how much the objects overlap and with different object shapes to obtain the segmentation limit value.

iv. To compare the SDC Morphology Toolbox and the Image Processing Toolbox and determine which toolbox gives the more precise and accurate nuclear track counting and granulometry (area measurement).

1.7 Chapters Overview

This dissertation contains five main chapters. The overview of each chapter is given below.

- ☑ **First Chapter**

  This is the introduction chapter with discussion on research background, mathematical morphology, nuclear tracks, motivations of the research and purpose and objective research.

- ☑ **Second Chapter**

  In this chapter theoretical background of Image Processing and Mathematical Morphology are discussed and explained including the algorithms, which are relevant to the objectives of the research. A section related to Mathematical Morphology will give some ideas on what MIP is all about. Then the chapter included a brief of theory discussion of object extraction, granulometry and nuclear track formation.

- ☑ **Third Chapter**

  The descriptions, that focuses on MATLAB as a programming platform for Image Processing including Image Processing Toolbox and SDC Morphology Toolbox.

- ☑ **Fourth Chapter**

  The chapter is all about research methodology.

- ☑ **Fifth Chapter**

  All of the research findings and the discussions relating to the result obtained are found in this chapter.

- ☑ **Sixth Chapter**

  This section provides the final conclusions and further deductions, specifically from the research findings.

CHAPTER 2

THEORY AND BACKGROUND

2.1 Chapter Overview

In this chapter, theoretical background of image processing and mathematical morphology, including the relevant algorithms are discussed. The basic concepts used in the SDC Morphology Toolbox for MATLAB are also covered.

2.2 Theoretical Background of Image Processing

There are many possible definitions for the term of Image Processing and Computer Vision. The most common one that will be stated in this thesis is that image processing aims to process an image, usually by a computer to produce informative image output. On the other hand, computer vision processes the image and produces it into some form of generalised information about the image, such as labelled regions. Ballard and Brown (1982) came out with the statement that "*Computer Vision is the enterprise of automating and integrating a wide range of processes and representations for visual perception*". They also include the term of image processing within this definition by implying that image processing is one of the many steps in computer vision. Niblack (1986) describes image processing as "*the computer processing of pictures*", whereas Computer Vision includes many techniques from image processing, but is broader in the sense that it is concerned with a complete system.

In many situations, image processing is operated with taking an array of pixels (image) as input and constructing a new array of pixels as output which somehow represents an improvement and enhancement to the original array. Image processing methods have been divided into Real Space methods and Fourier Space methods. The

Real Space methods are engaged in processing the input pixel array such as Grey Level Thresholding, Image Smoothing, and Detection of Points, Lines and Edges. The Fourier Space methods work by deriving a new representation of the input data by performing a *Fourier Transform*, which is then processed. Finally, an *Inverse Fourier Transform* is performed on the resulting data to give the final output image.

A digital image can be written down as $a[m, n]$, which '*a*' is matrix with m rows and n columns, and also represented in a 2-D discrete space. It is derived from an analogue image $a(x, y)$ in a 2-D continuous space through a *sampling* process, also known as digitisation. The process will digitise the image function, $f(x, y)$ spatially and in amplitude (Russ, 2008).

The digitisation process of an image is shown in Figure 2:1, which the analogue image or 2-D continuous image can be presented as $a(x, y)$ and divided into N rows and M columns. The intersections point of a row and a column is known as pixel. The value assigned to the integer coordinates of a pixel $[m, n]$ with $\{m = 0, 1, 2, ..., M\text{-}1\}$ and $\{n = 0, 1, 2, ..., N\text{-}1\}$ is $a[m, n]$. In many cases, the 2-D continuous image $a(x, y)$ which is considered to be the physical signal that gives limit effect on the face of a 2-D sensor, where the signal is an actual function of variables together with depth (z), colour (λ), and time (t). In this study, the images are 2-D, monochromatic and static images.

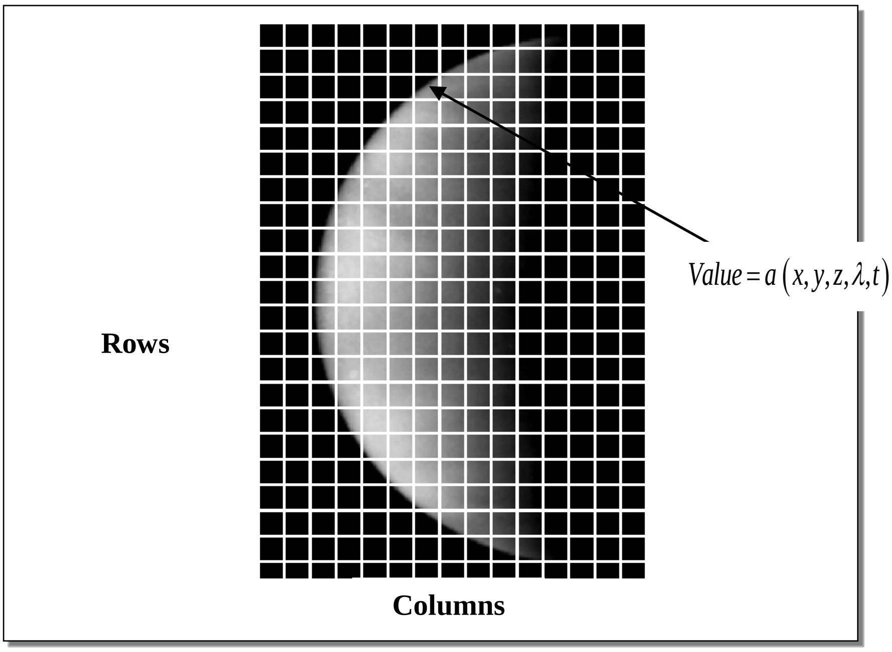


Figure 2:1 Digitisation of a continuous image. The pixel at coordinates [m = 7, n = 3] has the integer brightness value 212.

The image shown in Figure 2:1 has been divided into pixels of N = 22 rows and M = 15 columns. A value assigned to each pixel is the average brightness of the pixel rounded to the nearest integer value. The process of representing the amplitude of the 2 dimensional signals, at a given coordinate as an integer value with different grey levels is usually referred to as amplitude quantization or simply *quantization*.

2.3 Mathematical Morphology Theoretical Background (The *Concepts of Mathematical Morphology*)

One of the abilities of image processing and analysis refers to geometrical concepts such as size, shape, and orientation. Mathematical morphology uses concepts from set theory, geometry and topology to analyse geometrical structures in an image (Heijmans, 1994). The name Mathematical Morphology (MM) was introduced in 1966 (Serra & Soille, 1994) and the theoretical treatment presented by Matheron (1975) and Serra (1982; 1988). MM provides an approach to image processing, which is based on the shape of an object. In the context of image processing, it is the name of a specific methodology, designed for the analysis of the geometrical structure in an image. The method has also found applications in several other fields, such as medical diagnostics, histology, industrial inspection, computer vision, and character recognition (Kukielka & Woznicki, 2001).

MM examines the shapes or geometrical structure of an image by probing it with small patterned images which is called ***structuring elements***. These structuring elements can take in various shape and size. A non-linear image will be produced from this procedure, which is suitable for studying geometrical and topological structures. These MM operators are then applied to an image in order to make certain features perceptible and apparent. The process of reducing the object shape in the image to a sort of caricature (skeletonization) will distinguish the meaningful information from inappropriate distortions. For example, the shape of an object can be transformed to the digital image of a symbol by reducing each connected component to a one pixel thick skeleton while retaining the symbol's object shape. The effectiveness of this skeletonisation process is sufficient for recognition and it can be handled much more economically than the full symbol.

Theoretically, mathematical morphology studies operators between complete lattices, specifically nonempty sets equipped with a partial order relationship for which every subset has an *infimum* and a *supremum*. Appendix 2A gives further details about infimum and supremum. In fact, MM was developed from translation invariant operator between complete lattices and it can be represented by means of elementary morphological operators. An image operator can be created by composing elementary morphological operators. However, this approach is not practical when a large number of elementary operators are used. Fortunately, most applications can be successfully operated within below the limited reasonable number of morphological operators.

Therefore, an image analysts need to identify the category of morphological operators for a particular problem, whether it is for shape detection, extraction, or filtering.

Morphological operations tend to simplify, enhance, extract or describe image data. Figure 2:2 shows the relationships among the basic elements of MM.

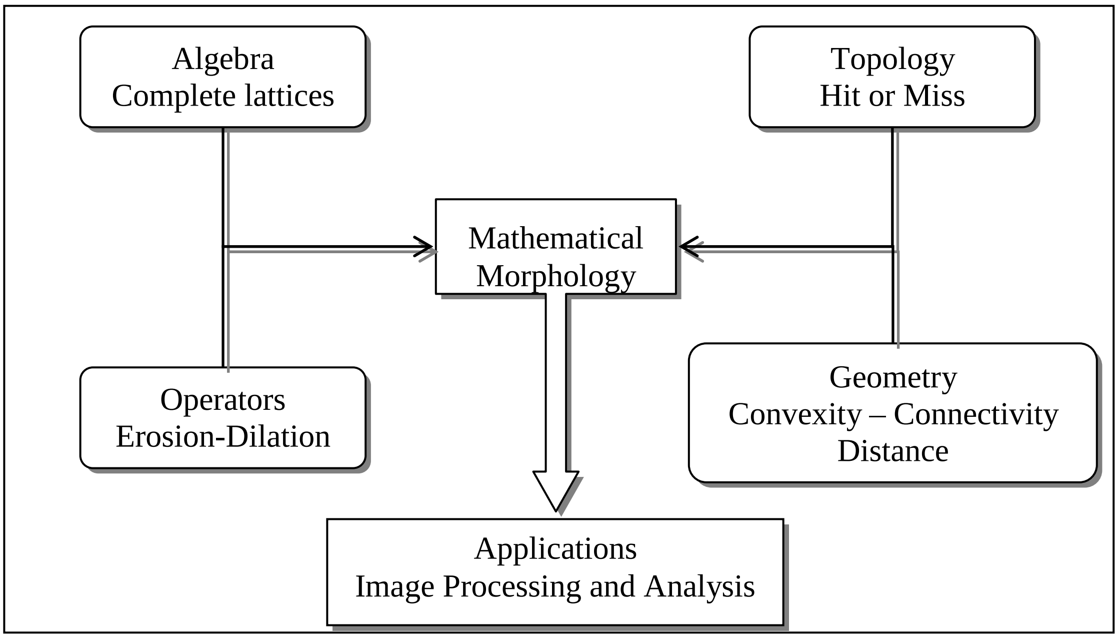


Figure 2:2 A mathematical tools diagram that studies operators on complete lattices.

Morphology originally comes from a branch of biology that refers to the form and structure of animals and plants. Also, it is used for the study of geometry and topology of patterns. In addition, integral-geometry Morphological Image Analysis (MIA for short) employs additive image functionals to assign numbers to the shape and connectivity of patterns formed by the pixels in the image. A part from that, integral geometry provides the precise mathematical framework to define these image functionals (Santaló, 2004; Stoyan, Kendall, & Mecke, 1996; Michielsen & Raedt, 2001). The fundamental theorem of integral geometry states that under certain conditions, the number of different additive image functionals is equal to the dimension of the pattern plus one (SDC, 2011). Thus, in the case of a 2-D image, there are exactly three of these functionals, called quermassintegrals or Minkowski functionals. For a given image, the first step in MIA is to compute these functionals themselves. The second step is to study the behaviour of the three or four numbers as a function of some control parameters, such as time, density, and so on (Michielsen & Raedt, 2001).

A significant feature of MIA is the massive contrast between the simplicity of implementation or use and the level of sophistication of the mathematical theory. Without a doubt, as follows, the calculation of the image functionals merely amounts to the proper counting of, for example faces, edges and vertices of pixels. The application of MIA requires little computational effort. Another appealing feature of MIA is that the image functionals have a geometrically and topologically intuitive and hence, also have perceptually clear interpretation for 2-D images as they correspond to the area, boundary length, and connectivity number. The four functionals for 3D images are the volume, surface area, integral mean curvature and connectivity number (Michielsen & Raedt, 2001).

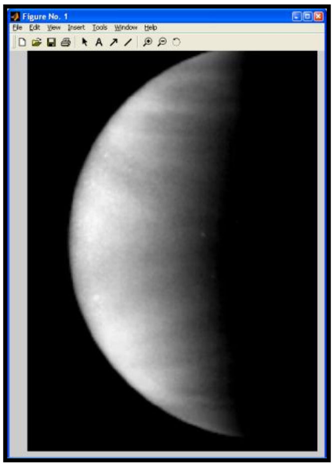
2.3.1 Morphological Image Processing

Basically, the digital form image can be produced as an output from any image processing program (software), which the MATLAB itself could be used as a digitised image generator software. The digitised images are free of noise and other artifacts that may affect the geometry and topology of the image structures or area of interest. Such perfect images are easily generated by a computer and are very useful for the development of theoretical concepts and models. Unfortunately, genuine pictures or patterns obtained from computer simulations are usually imperfect. Therefore, some special form of image processing may be necessary before attempting to make measurements of the features in the image (Michielsen & Raedt, 2001).

In Morphological Image Analysis, it is very important to acquire the geometric and topological content of the image from the operations that are used to enhance the image quality. The morphological image processing (MIP) technique will be reviewed in the following section and is well adapted for this purpose. This is because MIP and MIA are based on the same mathematical concepts. Most importantly, it is very flexible, fast and easy to use. Pioneering work in this field was carried out by Matheron (1975) and Serra (1982).

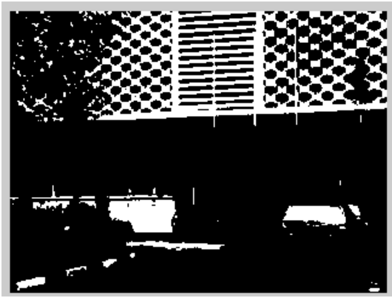
The most important reason why we use mathematical morphology as the main principle in solving various Image Processing problems in this research is because it is a special mathematical tool or method for investigating geometric structure in binary and also greyscale images. In other words, mathematical morphology is not just a theory of geometric, but it also known as a geometric approach to Image Processing and Analysis. By using this approach, it makes the images become easier (***becoming noiseless***) to analyse. This is what we need in this research where the visual perception requires transformation of images so as to make particular shape information observable. The goal of this approach is to distinguish meaningful shape or structure information from irrelevant one (*images*). The vast majority of shape processing and analysis techniques are based on designing a shape operator which satisfies desirable properties. Figure 2:3 and Figure 2:4 are shown below to display the difference between greyscale images and binary images. The binary images only show white and black colour, which the pixel values only give '0' and '1' as discussed earlier.

(a)

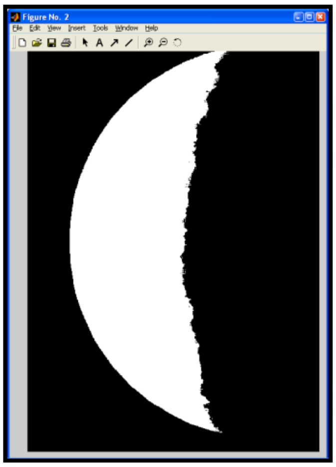


(b)

Figure 2:3 Greyscale images



(a)



(b)

Figure 2:4 Binary Images

Image analysis consists of obtaining measurement characteristics to images under consideration. For example, geometric measurements determine object location, orientation, area, and length of perimeter.

The basic operations associated with an object are the standard set operations *union* $\{\cup\}$, *intersection* $\{\cap\}$, and *complement* $\{C\}$ and also translation, which is given by a vector x and a set A. The translation, $A + x$, is defined as equation 2:1 below;

$$A + x = \{\alpha + x | \alpha \in A\} \qquad 2:1$$

Since we are dealing with a digital image composed of pixels at integer coordinate positions (Z^2), thus implies restrictions on the allowable translation vectors x.

The basic Minkowski set operations addition and subtraction can now be defined. The individual elements that comprise of B are not only pixels but also vectors, as they have a clear coordinate position with respect to [0, 0]. Given two sets equation A and B, as stated below;

Minkowski addition: $A \oplus B = \bigcup_{\beta \in B} (A + \beta)$

Minkowski subtraction: $A \ominus B = \bigcap_{\beta \in B} (A + \beta)$

2.3.2 Hit-or-Miss Transformation (HMT)

The language of MM is that of a set theory, where the set represents binary and grey level images. For instance, the set of all white pixels in a black and white image form a complete description of the image and can also be regarded as an *image object*. MM extracts information about the geometrical structure of an object by transforming it through its interaction with another object, called the structuring element, which is of simpler shape than the original object. The information about size, spatial distribution, shape, connectivity, convexity, smoothness, and orientation can be obtained by transforming the image object using different structuring elements. An example of a structuring element, B_x centred at a point *x* is given in Figure 2:5. The point *x* can be seen as the origin of a coordinate system, which permits to address any of the positions of B_x by a vector *b*.

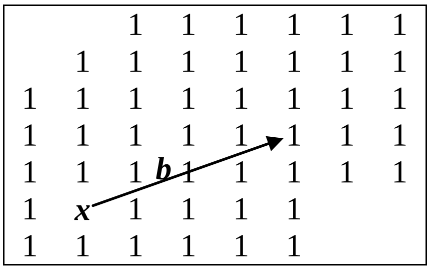


Figure 2:5 Example of a (non-symmetric) structuring element, centered at *x*. The vector *b* permits to address different positions.

The most fundamental operator for shape detection is known as *Hit or Miss Transformation (HMT)* (Serra, 1982). It is a point by point transformation of a set *X*. Structuring element, B_x is composed from two sub-sets, B_x^1 and B_x^2 which is centred at the same point *x*, as follows;

$$B_x = B_x^1 \cup B_x^2 \qquad 2:2$$

The set B_x^1 only contains ones and the set B_x^0 only contains zeros. Both sets may contain points of indifferent values, which means one or zero. Apart from that, point *x* belongs to the HMT, $X \otimes B$ of X, if and only if B_x^1 is included in X and B_x^0 is included in the complement of X, as follows;

$$X \otimes B = \{x : B_x^1 \subset X ; B_x^0 \subset X^c\} \qquad 2:3$$

An illustrative example for an image on a discrete grid is given in Figure 2:6. The structuring element is 1 at its centre and 0 on the right side, and thus B_x^1 is one at the centre and of indifferent value on the right (represented by a point in Figure 2:6). Whereas B_x^0 is of indifferent value at the centre and zero on the right side. It is easy to observe that only the right border points result from the HMT transformation. The sample structuring element shown below (Figure 2:6) is taken from the online documentation of SDC Morphology Toolbox, which re-groups a number of structuring elements B and is also interesting for various image analysing tasks.

$$\begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \xrightarrow{\text{HMT}} \begin{array}{|c|c|c|}\hline 1 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \longrightarrow \begin{array}{|c|c|c|}\hline 0 & 1 & . \\ \hline 1 & 1 & . \\ \hline . & . & 1 \\ \hline \end{array}$$

Figure 2:6 Example of a HMT transformation

2.3.3 Erosion

Erosion reduces the size of an object and eliminates features that are less than the scale of the neighbourhood as defined by the structuring element. It can be obtained by applying the HMT where B_x^0 is the empty set. Hence, the eroded set Y of X are the locus of the centres x of B_x^1 included in the set X. It can also be obtained by the classical **Minkowski subtraction** (Haralick & Shapiro, 1991), $X \ominus B = \bigcap_{b \in B} X_b$, where X_b is the translated version of X by the vector b. Erosion is thus defined as (Serra, 1982);

$$Y = \{x : B \subset X\} = \bigcap X_{-b} = X \ominus \breve{B} = \epsilon_{1,b}\ (x) \qquad 2:4$$

From the definition $\breve{B} = \bigcup_{b \in B} \{-b\}$ is the transposed set of *B*, where the reflected set of *B* is with respect to the origin, and $\in_{1,B}$ is the erosion of *X* by *B* of size 1. This process is viewed as a removing process of a certain number layers of the object. An example of erosion is given in Figure 2:7 and Figure 2:9.

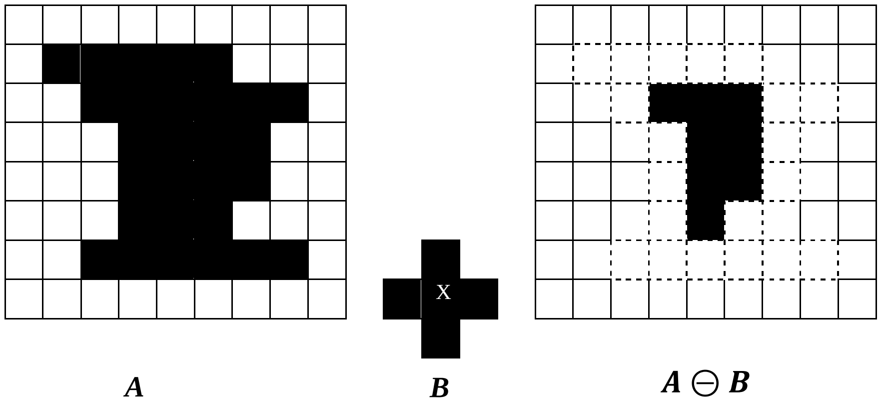


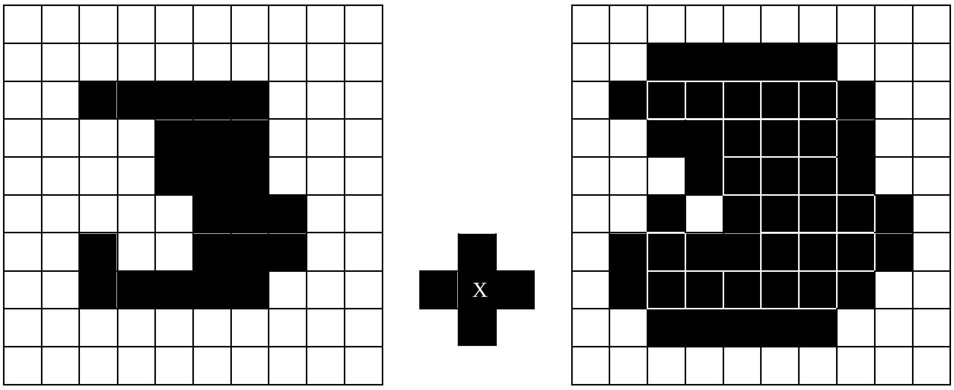
Figure 2:7 Minkowski subtraction is $A \ominus \breve{B}$

2.3.4 Dilation

Dilation expands the size of objects on the scale of structuring element neighbourhood. It is the dual operation of the erosion with respect to complementation, meaning a dilation of X^C is equal to an erosion of *X*. It is the locus of the centres of the B_x^1 which hit the set *X*. Dilation can also be expressed in terms of Minkowski addition and is defined as follows (Serra, 1982):

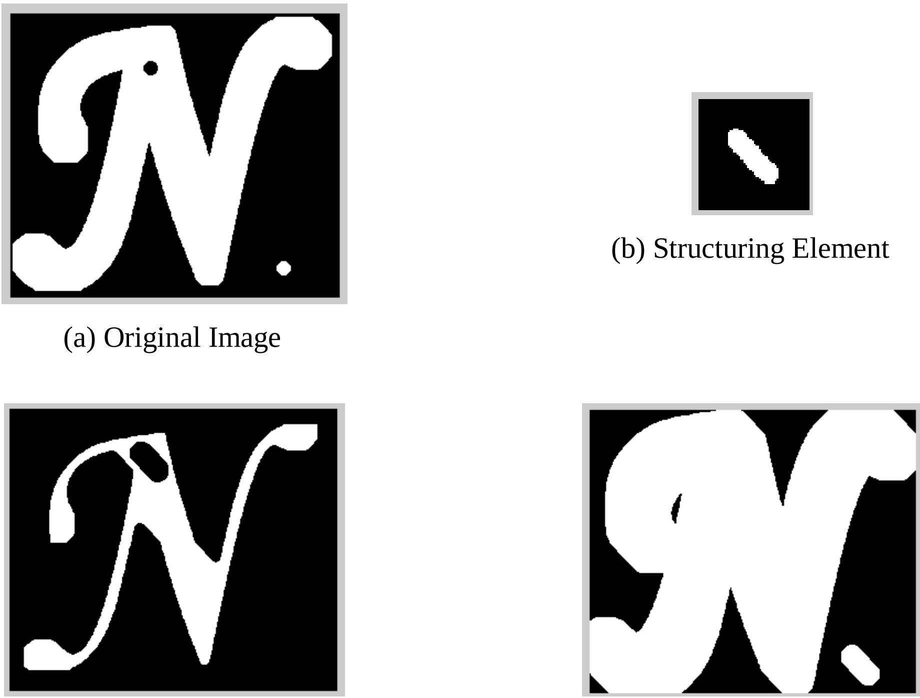
$$\emptyset Y = \left\{x: B_x \bigcap X \neq \emptyset\right\} = \bigcup_{b \in \breve{B}} X_b = X \oplus \breve{B} = \delta_{1,B}(X) \qquad 2:5$$

Consequently, the transposed set $\breve{B}$ is occupied to be able to make the analogy with the HMT. An example of dilation is given in Figure 2:8, where the process can be seen as the addition of layers to the object. See Figure 2:9 for more examples of dilation processes using MATLAB.



X *Y* $X \oplus Y$

Figure 2:8 Binary Morphological Dilation process



(a) Original Image

(b) Structuring Element

(c) Erosion image, $\in_B (X)$

(d) Dilation Image, $\delta_B(X)$

Figure 2:9 Example of erosion and dilation of a binary set *X*

2.4 SDC Morphology Toolbox for MATLAB

The SDC Morphology Toolbox for MATLAB is an accumulation of greyscale morphological tools that can be applied to image segmentation, non-linear filtering, pattern recognition, and image analysis (SDC, 2011). It includes effective segmentation functions such as watershed and connected filters based on reconstruction. The SDC Morphology Toolbox comes with several scripts showing the morphological solution to many real life image processing problems. For instance, several images in the field of machine vision, medical imaging, desktop publishing, document processing, food industry, agriculture, and so on, are included in the toolbox.

The SDC Morphology Toolbox is a set of functions or programs for Image and Signal Processing. The functions in this toolbox are based on the theory of Mathematical Morphology, which have been discussed in section 2.3. It is a non-linear approach for the representation of image and signal transformation, which makes it practical for analysing some Image and Signal Processing problems generally, such as segmentation, and extraction of quantitative information, noise reduction and compression. Mathematical Morphology has a special characteristic which allows us to learn generic transformations of discrete signals directly, without appealing to continuous approximations. This characteristic allows for the development of efficient algorithms of compact integer data structures.

Nowadays, researchers in image processing and computer vision field are increasingly recognising that simple algorithms derived from mathematical morphology can be extremely useful in all kinds of applications. Hence, MATLAB is becoming increasingly important as the programming environment for image processing. Furthermore, the addition of the MORPHOLOGY TOOLBOX for MATLAB enhances MATLAB image processing capabilities.

2.4.1 Availability of the Toolbox

The SDC Morphology Toolbox is generated mostly using automatic methods. This ensures that the syntax and conventions used are very consistent with the toolbox. The images loaded in SDC Morphology Toolbox are represented as 1-D, 2-D, or 3D arrays, with pixels of types *uint8*, *uint16*, and *logical uint8* (binary). The image data type in *uint8* represents positive integers from 0 to 255, the *uint16* image data type represents integers from 0 to 65535, and the logical *uint8* represents just the numbers 0 and 1. Structuring elements and a sub-image are needed to process the image.

The toolbox has Interval functions which are also known as *hit-or-miss templates*. They are useful to process the morphological images. Generally, these functions are the functions that create interval to detect end-points of curves, and interval for homotopic thickening and thinning of binary image. The toolbox also has a few functions that can be applied to manipulate the interval, whether to rotate by an angle or to visualize it. This Interval function also has unique functions that are capable to create hit-or-miss template or an interval itself, based on a pair of structuring elements. For more details about Interval functions, please see its example in Appendix 2B.

2.4.2 General rules

The SDC Morphology toolbox has several general rules which we need to comprehend before applying it. These general rules make it easier for us to understand and handle the toolbox.

The first rule stated that each function of the toolbox can return only one variable, for example; the application of *mmreadgray* and *mmbinary* functions in the box below will return each, a variable of '*a*' and '*abin*', which are the images. Figure 2:10 shows the MATLAB script, which the functions return one variable each.

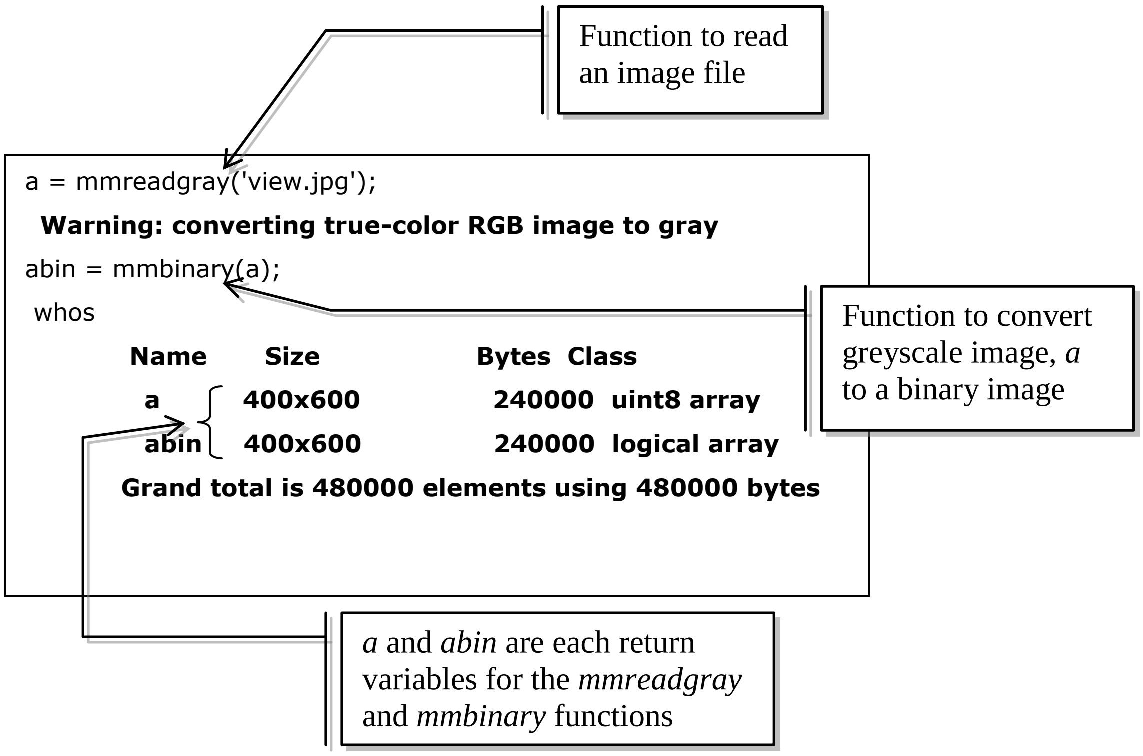


Figure 2:10 The MATLAB script of the functions that returns one variable each

In addition, the command line of MATLAB can be engaged from left to right, and the SDC Morphology toolbox's operands can also be optional from left to right only. *Mmconv* is one of the functions in SDC Morphology Toolbox. Please refer to Appendix 2C for more detailed information regarding conventions that are used in the SDC Morphology Toolbox.

The toolbox default parameter of the structuring element is *mmsecross*, which is the elementary 3×3 cross. The *mmsecross* function generates the structuring element *B*, formed by *r* successive Minkowski additions of the elementary cross. In other words, the 3×3 cross centred is at the origin with itself. If $r = 0$, *B* is the unitary set that contains the origin. If *r* = 1, *B* is the elementary that cross itself. Equation 2:6 below which proves how the *mmsecross* functions;

$$B(r) = \{(x_1, x_2) \in Z^2 : |x_1| + |x_2| \leq r\} \qquad 2:6$$

And then, Figure 2:11 demonstrates the MATLAB script to demonstrate the function of *mmsecross*.

```
se1 = mmsecross;
mmseshow(se1)
 ans =
     0   1   0
     1   1   1
     0   1   0
se2 = mmsecross(2);
mmseshow(se2)
 ans =
     0   0   1   0   0
     0   1   1   1   0
     1   1   1   1   1
     0   1   1   1   0
     0   0   1   0   0
```

Function to display structuring as an image

The default structuring element created by *mmsecross* (r = 1)

Structuring element by r = 2

Figure 2:11 The MATLAB Script of *mmsecross* function

In other words, the radius, r needs to be determined to *mmsecross* in order to create the structuring elements with the radius more than 1.

All the images in most functions must have the same data type and size. This rule states that the data type and size of images should be in the same value in order to make the functions of SDC Morphology Toolbox run smoothly. This can be seen below in the example of MATLAB script with regard to this matter.

```
im1 = mmreadgray('cat_fr_gray.jpg');
im2 = mmreadgray('pcb2.tif');
whos
  Name      Size              Bytes  Class
  im1     400x600            240000  uint8 array
  im2     400x600            240000  uint8 array
uniim = mmunion(im1, im2);
mmshow(im1);
mmshow(im2);
mmshow(uniim);
```

whos function shows all the variables in the workspace, and displays the size of *im1* and *im2* are same.

mmunion creates the image by taking the pixel-wise maximum between the images and represents the union of them.

Figure 2:12 The MATLAB Script that shows data type and size are in the same value

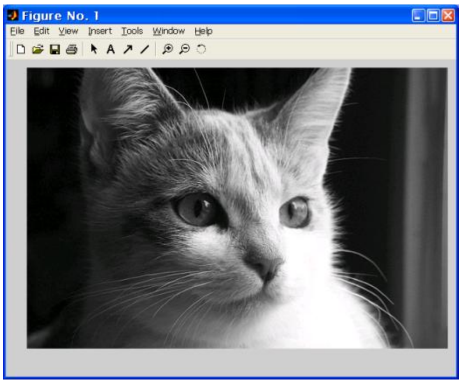


Figure 2:13 Image, ***im1***

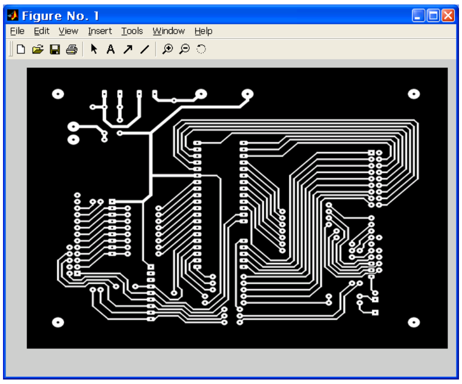


Figure 2:14 Image, ***im2***

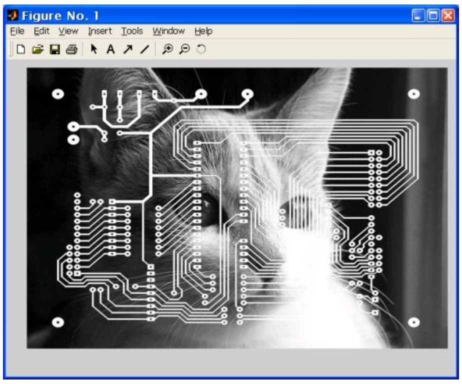


Figure 2:15 Final image, ***uniim***

As the result of the script which finally produces three images. This can be seen in Figure 2:13, which shows the first image from the script, and then Figure 2:14, which shows the second image, and the final image in Figure 2:15 represents their union image.

The toolbox has the capability to operate an image with a constant value. This constant is regarded as a constant image and is the same size as the other images. This can be applied to *mmunion* function where the image besides the first images could be a constant (SDC, 2011), Refer to Appendix 2D for more detail about *mmunion* function.

Furthermore, inside the SDC Morphology Toolbox, there is a function called *mmfreedom*, which controls the automatic data type conversions. There are 3 possible levels, called FREEDOM levels (SDC, 2011). In this research, we use type conversion ‘1’. The FREEDOM levels are set or inquired by *mmfreedom* function.

Level of FREEDOM = 2

Level of FREEDOM = 1

Level of FREEDOM = 0

```
f = uint8([2 3 4]);
mmfreedom(2);
mmaddm(f, 5)
 ans =
   7    8    9
mmfreedom(1);
mmaddm(f, 5)
 Warning: Converting image from scalar to uint8.
 ans =
   7    8    9
mmfreedom(0);
mmaddm(f, 5)
 ??? Cannot convert arguments. Freedom level is 0.
 Error in ==>
 C:\MATLAB6p5\morph1.2\morph\mmaddm.dll
```

An error output if level of FREEDOM is equal to 0

Figure 2:16 MATLAB FREEDOM levels script

Figure 2:16 shows the result of the MATLAB script, the level of FREEDOM that is equal to ‘1’ is more appropriate in applying the toolbox because the user will realize to what is actually happening to the programming script and the variables, especially when the warning sign appear after the image type conversion is completed.

2.5 Objects Extraction Review

This sub-topic is about reviewing theoretically the extraction of the objects from the image. Firstly, the operator that is always used as a first step to start the script in this research is called Opening operator. Then, the derivation from the erosion operation occurs followed by a dilation operation (Pratt, 2001). A part from that, there are a few functions of this operator that will always make the objects in the image smooth.

Concurrently, the small images or islands will be eliminated depending on the structuring elements condition and they will also divide the narrow connection (Gatos & Perantonis, 2004). Object extraction can also be improved as fast opening functions to granulometries applications (Vincent, 1993; Vincent, 1994a). Figure 2:17, Figure 2:18 and Figure 2:19 illustrate the example of the Opening operator application using disk shape structuring element to the artificial (synthetic greyscale) image.

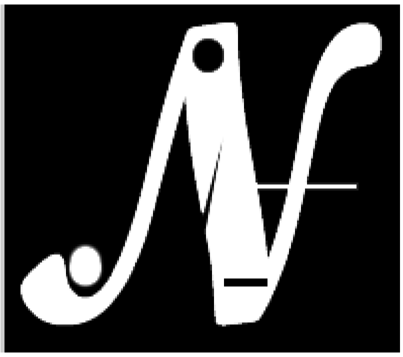


Figure 2:17 Original image

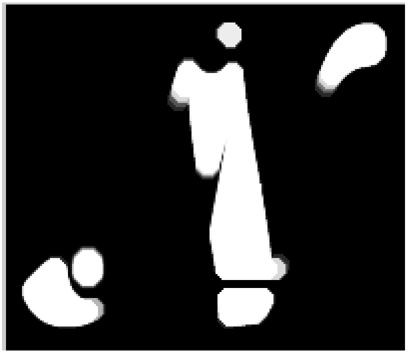


Figure 2:18 The image after Opening operation

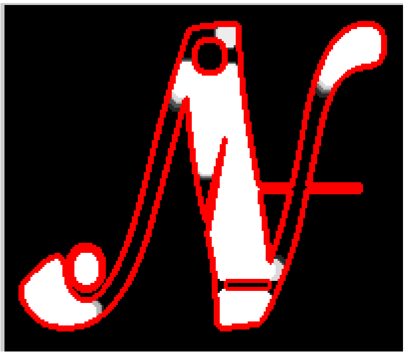


Figure 2:19 The overlay of original and open image

Hence, after the Opening operation has been done to the research image, the next process that follows is image reconstruction, which is also known as Infinite Reconstruction (Serra, 1983). Infinite Reconstruction (inf-reconstruction) constructs or rebuilds the portion of interest in the image by means of image filtering, 'top-hat', segmentation, etc. (Vincent, 1993; Heijmans, 1994). In other words, the purpose of the process is to give back the required objects into the image. The following figures present the example of the inf-reconstruction processing to the synthetic greyscale image, where Figure 2:20 shows the original image, Figure 2:21 shows the image after erosion processing and Figure 2:22 shows a reconstructed image.

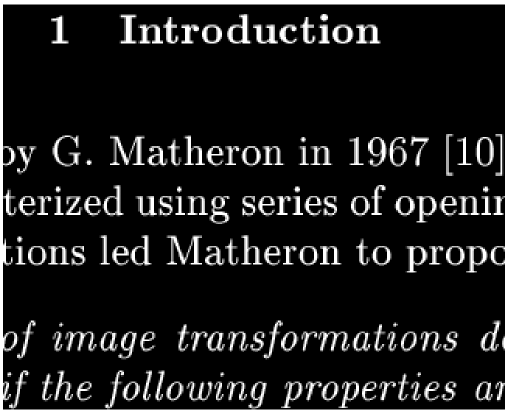


Figure 2:20 Original image

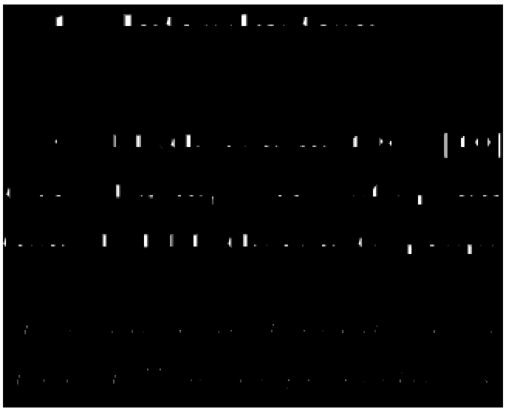


Figure 2:21 Eroded Image

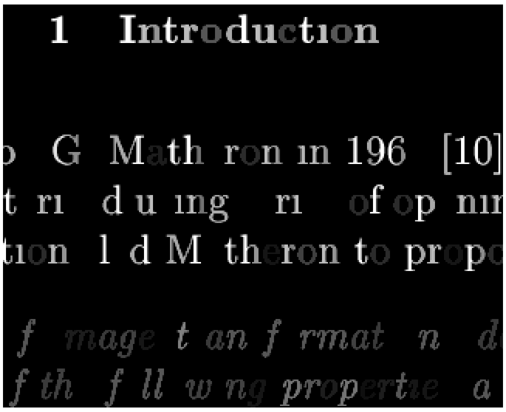


Figure 2:22 Reconstructed image

The following step is to subtract the research image in order to obtain a subtracted image based on the reconstructed image, where the portion of interest is more significant with the background pixel value uniformly (Serra, 1983; Gonzalez & Woods, 2002). See Figure 2:23, Figure 2:24 and Figure 2:25 that illustrate the example of subtraction operation for the sample image taken from SDC Information System's site. Besides that, the operation of image comparison would be useful in order to obtain the image with uniformly pixel value background (SDC, 2011).

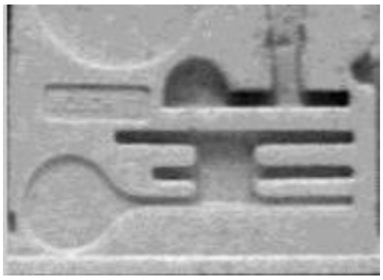


Figure 2:23 First image

Figure 2:24 Second image

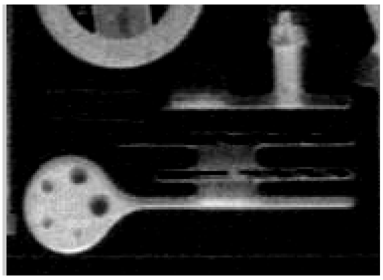


Figure 2:25 Subtracted image

After a few operations have been done to the image, the Area Opening operation will be proposed in this research where the grains (unwanted objects) will be removed and only the required objects or shapes will be kept (Haralick, Sternberg, & Zhuang, 1987; Scott & Mukherjee, 2000). The Area Opening operation will remain the portion of interest based on the conditions given, for example, in choosing the area of the object pixel wisely (SDC, 2011).

In addition, Close Holes operator can be also used in order to get fine extracted objects in the image. Close Holes is one of the morphological operators that function as closing the holes of the image or filling the holes in every connected component of an image. It is also resulted from the improvement of the closing operator. This is very useful for granulometry function analysis of the size of the objects in the image because it will entirely produce the shape of objects in the image without changing their contour (Ruberto & Dempster, 2000). Figure 2:26 below shows the flow chart of the basic and complete object extracting processes as have been discussed above.

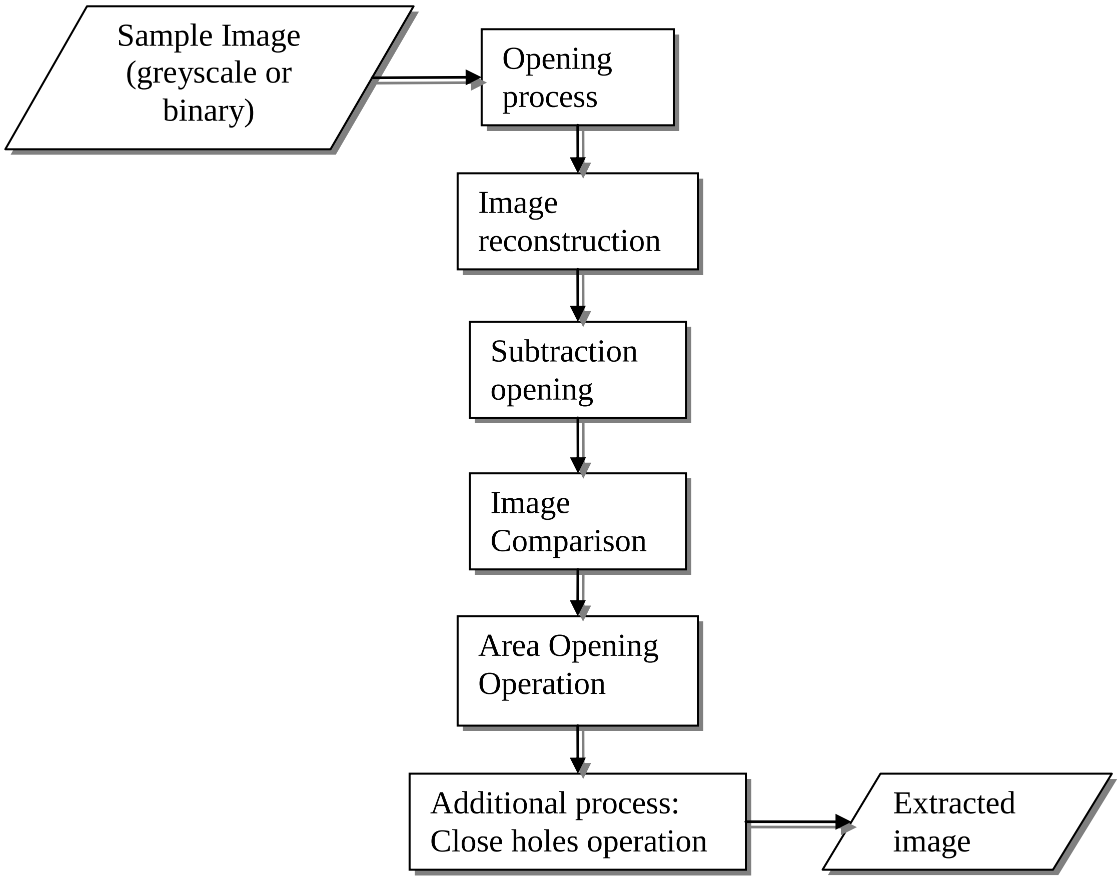


Figure 2:26 Object extracting process Flow Chart

2.6 Over-lapping Objects Extraction Review

This review is extended from the previous review (Section 2.5), which is about extracting the over-lapping objects from the image. There are certain processes for this review that are similar with the subject-matter discussed above, and the processes include thresholding, marking, and Geodesic ‘Skeleton by Influence Zones’ (SKIZ) which will be discussed later.

Generally, thresholding is a segmentation process of the objects in the image from the rest of image area and this process produces segmented objects, which we can distinguish from the other areas (Gonzalez & Woods, 2002). In order to get smooth segmented objects, an opening process is required for the image. Consequently, the distance transformation (Shih & Mitchell, 1992) and dilation operation can be applied to these segmented objects in order to obtain marked objects.

As stated above, this section is about extracting the over-lapping objects from the image. The separation of the over-lapping objects can be done by Geodesic SKIZ Approach (Vincent & Dougherty, 1994), a process which has a watershed operation. The process of watershed is based and limited by the markers image and applied to the negated distance function image. Furthermore, to get fully objects separation, a subtraction process can be applied to the Geodesic SKIZ processed image of the segmented image. This operation will separate the objects. See Figure 2:27 that shows the process flow chart, and Figure 2:28 shows the simulated image (224×186) pixels, which runs under the process where the image has over-lapping objects in it.

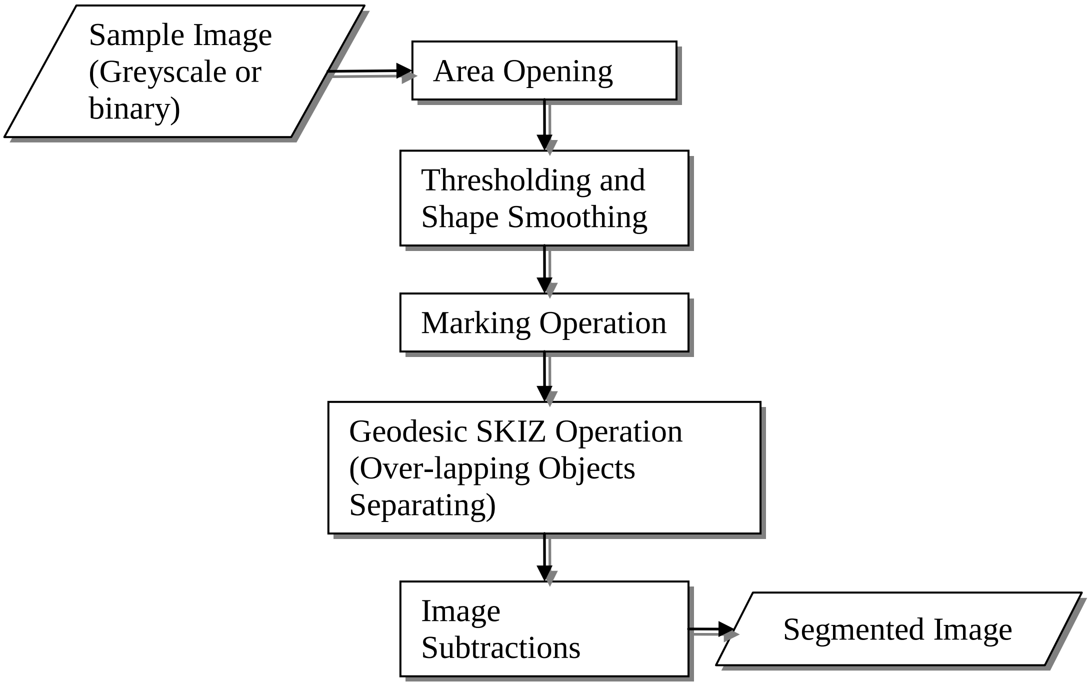


Figure 2:27 Over-lapping objects separating process

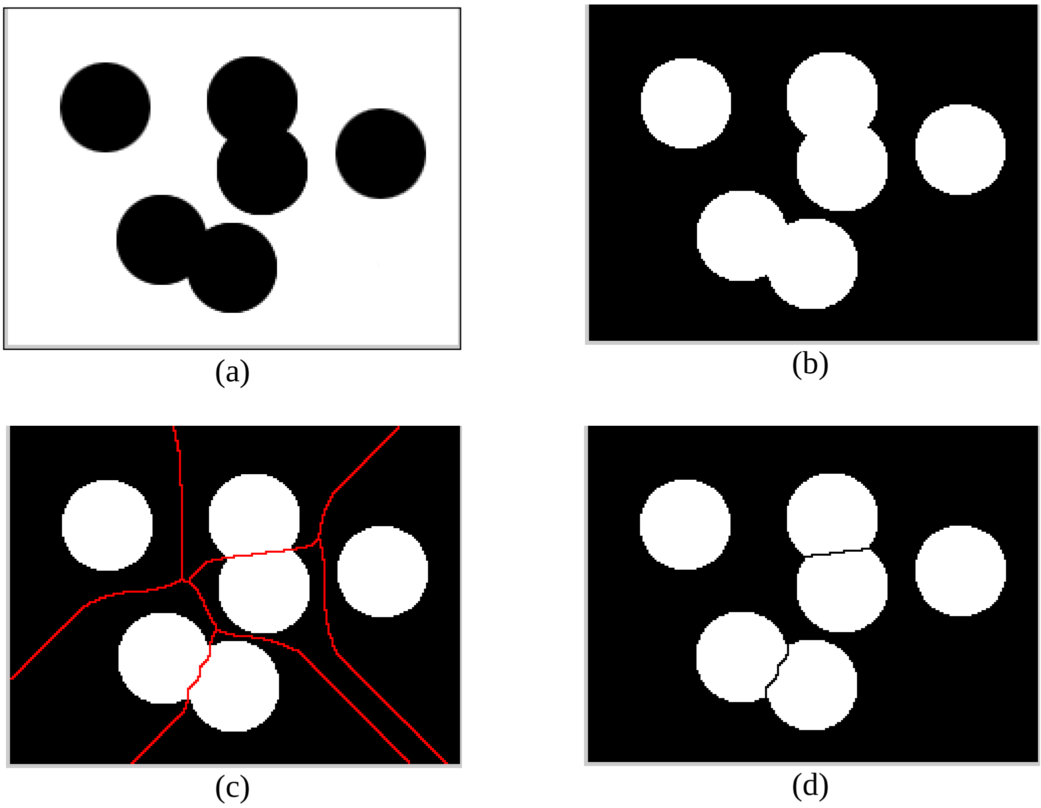


Figure 2:28 Images of Over-lapping extraction process; (a) Original image, (b) Thresholded image, (c) Image after Geodesic SKIZ process, and (d) Over-lapping objects segmented image

2.7 Measurement of the Objects/Grains, (Granulometry)

The objects that have been extracted and segmented as previously discussed can be used for measurement analysis, which is also known as Granulometry Analysis (Vincent, 1994b). Granulometries are programmed sets of morphological openings and closings operator that can be applied as detail image removals depending on its structuring elements' size and characteristic. Because of this reason, it becomes one of our tasks in this research to discover the toolbox capabilities' on Granulometries. Based on the SDC Morphology Toolbox's demonstrations and documentations as our references, we need to perform some Granulometries tasks to our images that have been extracted and segmented (the objects) as above. For other references on recent granulometry research, please refer to Ves, Benavent, Ayala, & Domingo (2006).

2.8 Nuclear Track Formation Mechanism and Its Shape

The formation of the nuclear track could occur when the radiation particle penetrated the film detector (nuclear track detector). The detectors are basically insulated by electrically solid materials where the passage of heavily charged particles creates trails along their paths and these damage zones on an atomic scale are called latent tracks. The treatment to these latent tracks using chemical or electrochemical etching allows their visualisation under optical microscopes.

When the detectors are exposed to the radiation source, the ionising particle interacts with matter and transfers a part of its energy to the electrons of the medium with some rate of energy loss. When the energy loss is above a certain critical value, local structure transformations or etched track are formed. These transformations in the case of polymers can be explained by splits in the molecular chains and formations of new components chemically very reactive along the particle trajectory. Typically the latent track formed has diameters from 1 to 10 nm (Fleischer, Price, & Walker, 1975). Figure 2:29 gives a schematic illustration of the effect caused by the passage of charged particles. Figure 2:30 shows a schematic of the film detector with latent track before and after the etching process.

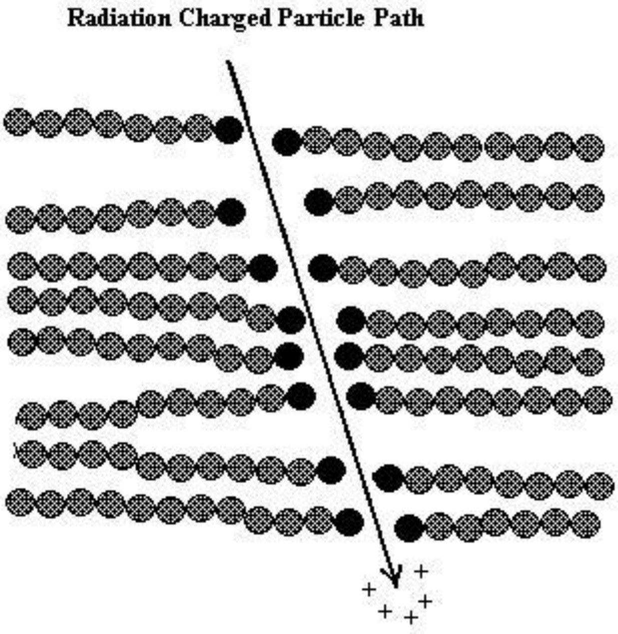


Figure 2:29 The schematic illustration of effect caused by the passage of charged particle

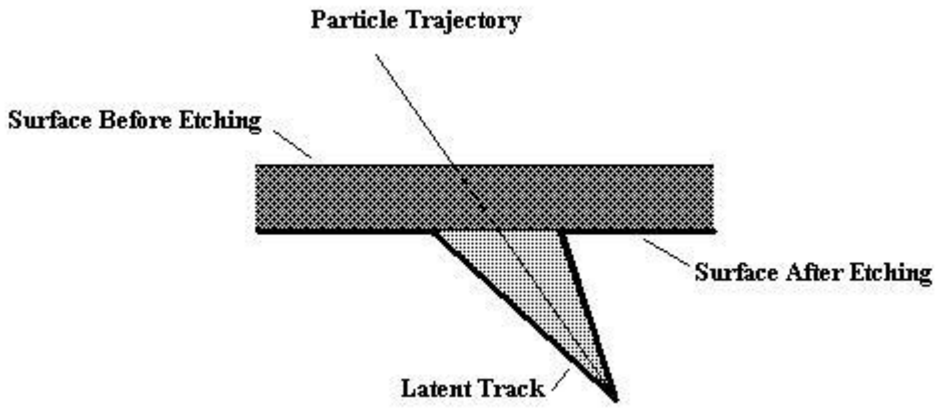


Figure 2:30 The schematic of the film detector with latent track before and after the etching process

The shapes of the nuclear tracks were not always regularly circular (circle) when viewed from above, there can also be oval in shape (ellipse), this depends upon the angle of incidence on the detector surface. Larger incidence angles will make nuclear track's shape more elliptical (oval) (Khan & Khan, 1989), see Figure 2:31. Some radiation particles make diamond shaped nuclear tracks; this could be seen from above view of image of a single-crystal mica template formed by nuclear track etching (Sun & Hao, 2005), see Figure 2:32.

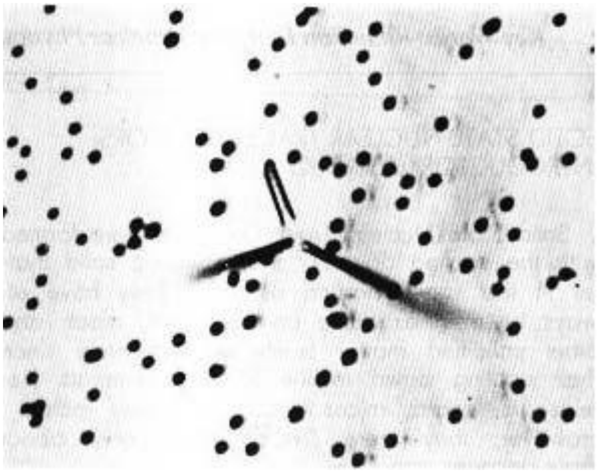


Figure 2:31 The top view of nuclear tracks ellipse shape (Khan & Khan, 1989)

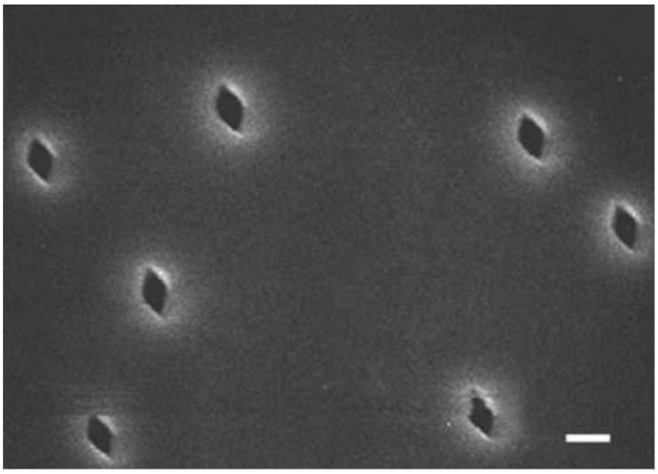


Figure 2:32 The top view of nuclear tracks diamond shape (Sun & Hao, 2005)

2.9 Chapter Summary

The effect of these MIP operations with their characteristics of structuring elements to the images could produce a variety of results depending on the parameters of structuring elements. The MIP has two important basic operators, namely the Erosion and Dilation, which could expand the procedure of the MIP. For instance, there are the Opening and Closing operations. The MIP that has a built-in SDC Morphology Toolbox is a complete tool of MIP applications, along with the MATLAB system.

There are 3 processing procedures that will be used with regard to this research; they are Object Extraction, Over-lapping Objects Extraction and Granulometry. All of them are related to MIP. It means that these procedures will also be utilised to test SDC Morphology toolbox and Image Processing toolbox capabilities.

CHAPTER 3

THE MATLAB TOOLBOX FOR IMAGE PROCESSING

3.1 MATLAB Image Processing

Fundamentally, image processing is basically to perform operation on images. In general terms, image processing refers to the manipulation and analysis of two dimensional visual images. It is any operation that acts to improve, correct, analyse, or change an image in some ways to produce better images as the objective requires (Forsyth & Ponce, 2003; Horn, 1986; Jain, Kasturi, & Schunck, 1995). Figure 3:1 shows the differences between image processing, image analysis and image understanding.

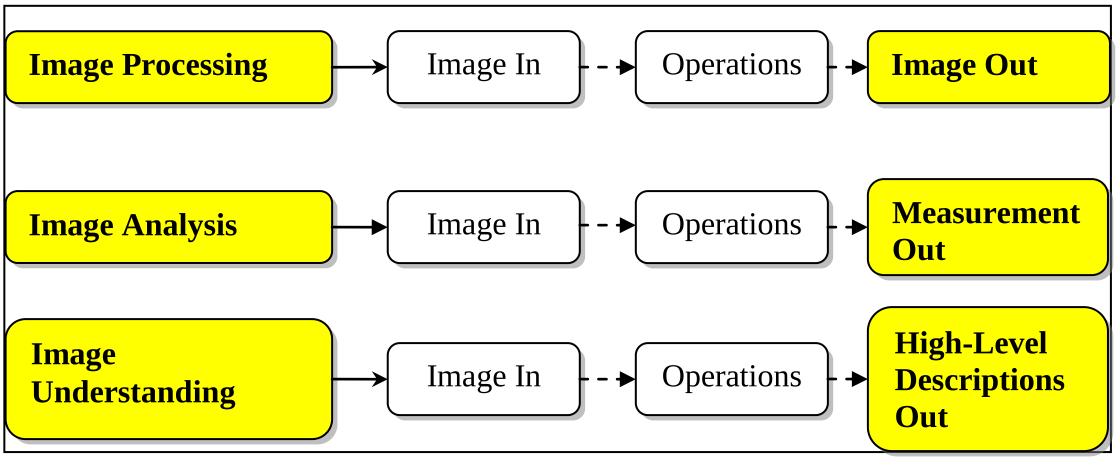


Figure 3:1 Three manipulations of image

Image processing operations can be generally divided into three major categories. They are image compression, image enhancement and restoration, and measurement extraction. Image compression involves reducing the amount of memory needed to represent a digital image (Matheron, 1988; User's Guide, 2001; Using MATLAB, 1999). The purpose is to compress the size of a file (picture or image) without reducing the quality of the picture or image itself. For instance, Joint Photographic Experts Group (JPEG) is one of the most popular and comprehensive continuous tone (as opposed to binary), still frame compression standard format. Further information related to image compression, other images or graphics file format and JPEG 2000 can be found in Gonzalez and Woods (2002), and Miano (1999).

A digital image can be composed (*consists*) of a set of points which may be defined as a two-dimensional function, $f(x, y)$; where x and y are *spatial* (plane) coordinates, and the amplitude of f at any pair of coordinates (x, y) has its own brightness which is called the ***intensity*** or ***grey level*** of the image at that point, see Figure 3:2. These points are called ‘pixels’ (*pixel stand for picture element*). The number of pixels within a unit length divided by area is called ‘resolution’, the level of detail measured in units per inch.

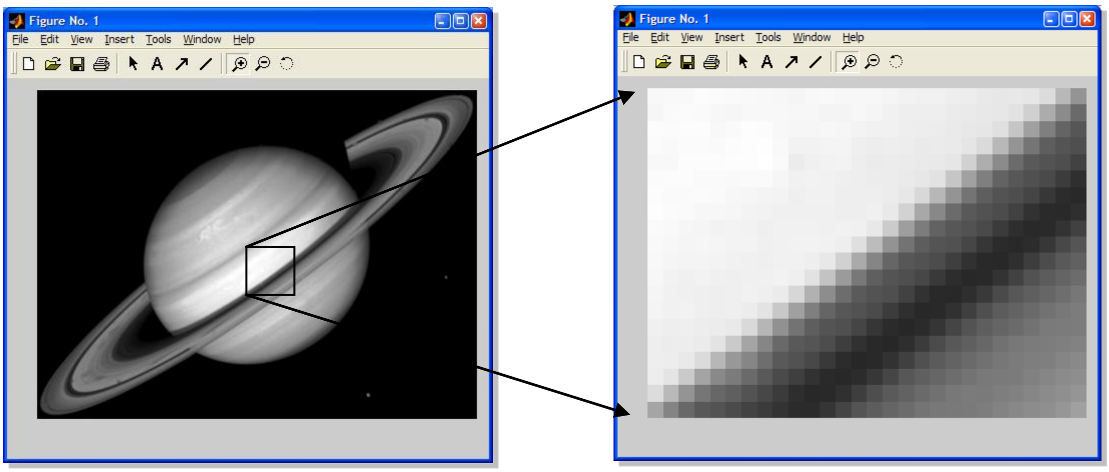


Figure 3:2 Pixels

Various operations may be performed on the pixels of the original image to produce a new image. These operations may be performed on each pixel in isolation or relative to other pixels in the image (Matheron, 1975). The purpose of these operations is to improve the visual appearance of images, or to prepare images for the measurement of features and structures present. More specifically, image processing may be used to clean up an image by removing the noise (noise reduction), improve the contrast of the image and highlight elements with certain characteristics. It may also overcome distortions such as blurring or warping of an image caused by a camera or scanner, and then compensate for uneven illumination which existed in the image.

The basic of image processing operations may also be carried out by **filtering or mask processing operation**, which attempts to highlight significant features and suppresses insignificant detail based on some neighbourhood operations. This is done by working with the values of the image pixels in the neighbourhood and the corresponding values of a sub-image (*filter*, *mask*, *kernel*, *template*, or *window*) that has the same dimensions as the neighbourhood.

Edge detectors operation (detecting the edge of the objects in the image) highlights the significant transitions of the object in the image. This will specify the boundaries within the image. Edges are places in the image with strong intensity contrast. Since edges often occur at image locations representing object boundaries, edge detection is extensively used in image segmentation when we want to divide the image into areas corresponding to different objects. Representing an image by its edges has the further advantage that the amount of data is reduced significantly while retaining most of the image information. Since edges consist of mainly high frequencies, theoretically, edge detection can be done by applying a high-pass *frequency filter* in the Fourier domain or by *convolving* the image with an appropriate *kernel* in the spatial domain. Practically, edge detection is performed based on spatial domain, is more computational and gives better results. Figure 3:3, Figure 3:4 and Figure 3:5 show the differences of Saturn edge detected images using Sobel and Canny operators. This processing procedure was produced by using MATLAB script, see Appendix 3A.

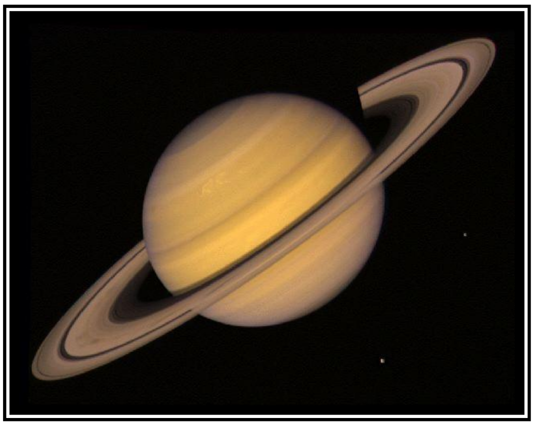


Figure 3:3 Original image of Saturn.

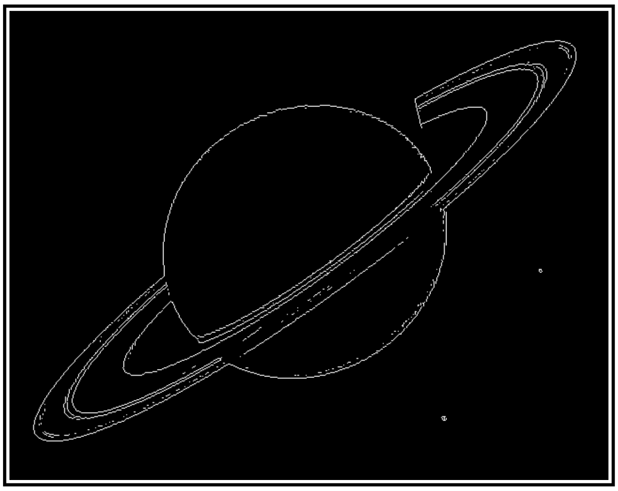


Figure 3:4 Edge detection by Sobel operator.

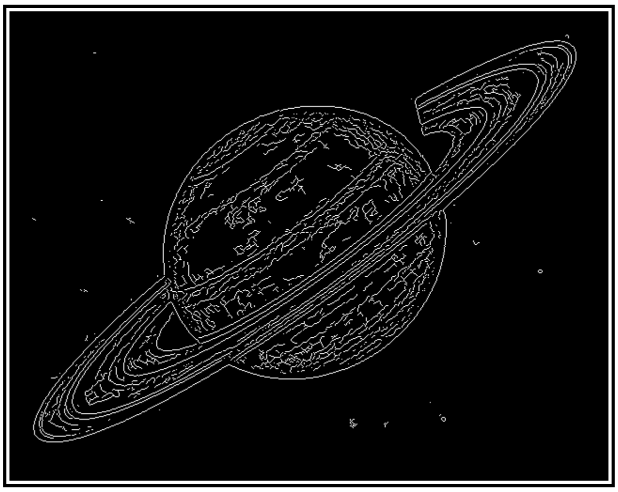


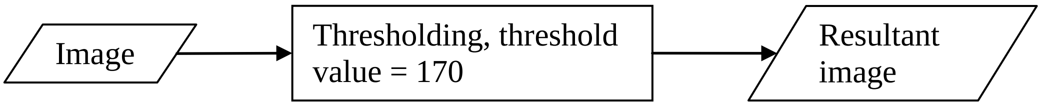
Figure 3:5 Edge detection by Canny operator.

Texture analysis involves identifying variation in brightness from one pixel to the next or within a small region in the image. Then, it can be defined as the pattern of the Grey Level Distribution (GLD) within the boundary of the field of view (Beddow, 1997). Initially the GLD is obtained from the grey levels at each coordinate (x, y) point of the image. The next step is to clarify the appropriate boundary function, called the Grey Level Boundary Function.

Thresholding and segmentation procedure partitions an image into its constituent parts or objects (Gonzalez & Woods, 2002). In other words it attempts to separate an object in image from its background. Thresholding is a well known technique for image segmentation (Haralick & Shapiro, 1991). It is the operation of converting a multi-level image into a binary image. In a binary image, each pixel value is represented by a single binary digit. In its simplest form, see equation 3:1; thresholding is a point-based operation that assigns the values of 0 or 1 to each pixel of an image based on a comparison with some global threshold value *T*.

$$f(x,y) = \begin{cases} 1, \text{ if } f(x,y) \geq T \\ 0, \text{ if } f(x,y) < T \end{cases} \qquad 3:1$$

Besides that, thresholding is an attractive early processing step as it leads to significant reduction in data storage and results in binary images that are simpler to analyse. Binary images permit the use of powerful morphological operators to shape and structure-base the analysis of image content. Appendix 3B gives an example of image segmentation using MATLAB. The input of a thresholding operation is typically a greyscale or colour image. In an uncomplicated implementation, the output is a binary image representing the segmentation. Black pixels correspond to background and white pixels correspond to foreground or vice versa. In other implementation, the segmentation is determined by a single parameter known as the *intensity threshold*. In a single pass, each pixel in the image is compared with this threshold. If the pixel's intensity is higher than the threshold, the pixel is set to white colour in the output. If it is less than the threshold, it is set to black colour (Fisher & Perkins, 2003). A simple flow chart that illustrates the effect of thresholding on an image is given in Figure 3:6 and Figure 3:7. Figure 3:6 shows the original image, *sample1.jpg* and Figure 3:7 shows an image after thresholding processes with a global threshold value, *T* of 170.



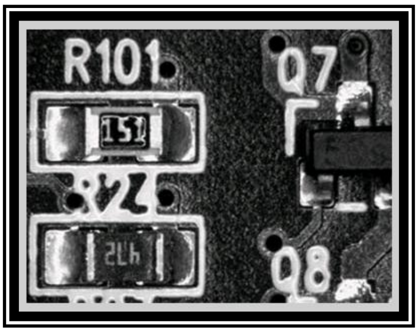


Figure 3:6 Original image, sample1.jpg

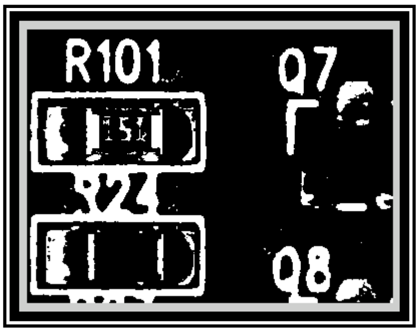


Figure 3:7 Threshold image

Thinning, skeletonizing and distance transforms are important approaches to represent the structural shape of a plane region. This is to reduce it to a graph where its converting dominant shape features economical, simplified representations (Gonzalez & Woods, 2002). The skeleton of the region may be defined as the Medial Axis Transformation (MAT), which can be divided into two main methods. The first method is the process of morphological thinning that erodes away pixels from the boundary (while preserving the end points of line segments) until no more thinning is possible. An alternative method is to first calculate the *distance transform* of the image and then the skeleton lies along the *singularities* that is curvature discontinuities in the distance transform. This approach is more suitable to calculate the MAT since the MAT is similar to the distance transform but with all points off the skeleton suppressed to zero. Figure 3:8 and Figure 3:9 show us the binary images and Figure 3:10 shows the images after skeletonizing are processed. The images are processed and presented by flow chart as shown below.

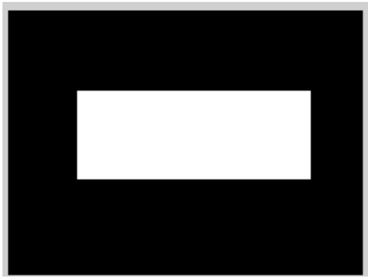


Figure 3:8 Binary image

Figure 3:9 Binary images for Skeletonizing operation

Figure 3:10 Skeletonized images

Histogram equalization processing, also known as histogram modelling technique is an approximation of a uniform histogram for an image. This method improves the range of dynamic and contrast of an image. This is done by altering the image based on its intensity which the histogram shaped. This technique is utilised in image comparison processes because of its effectiveness in detail enhancement and improvement of non-linear effects produced by a digitiser or display system (Fisher & Perkins, 2003). Figure 3:11, shows the usage of functional Image Processing Toolbox for MATLAB to perform histogram equalization. The process of adjusting intensity values can be done automatically by the *histeq* function. Example, Figure 3:12 show the histograms of intensity distributions for the two images.

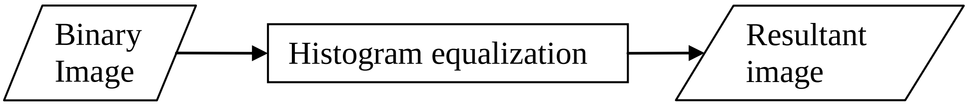


Figure 3:11 The image before (left) and after Histogram Equalization (right).
(*the image captured by camera digital laboratory*)

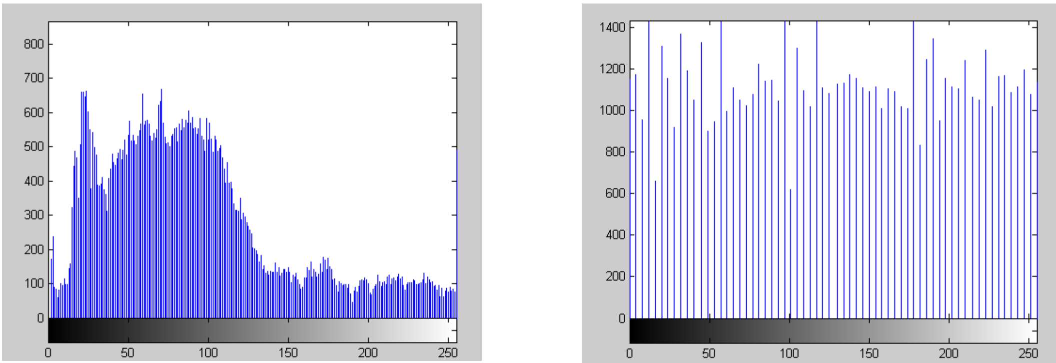


Figure 3:12 The histogram before Equalization (left) and after Equalization (right)

3.2 MATLAB Applications

MATLAB is a software that we choose to use as a tool platform to conduct our research. MATLAB stands for '***MATrix LABoratory***'. It is one of the programming languages used for technical computing. It produces a flexible environment for technical computing, specifically for mathematical computing, visualization and command language. It was previously developed based on the concept of open architecture, which makes it easier to use. It has specific companion products for specific usage (Using MATLAB, 1999). Figure 3:13 shows the snapshot of MATLAB version 6.5.

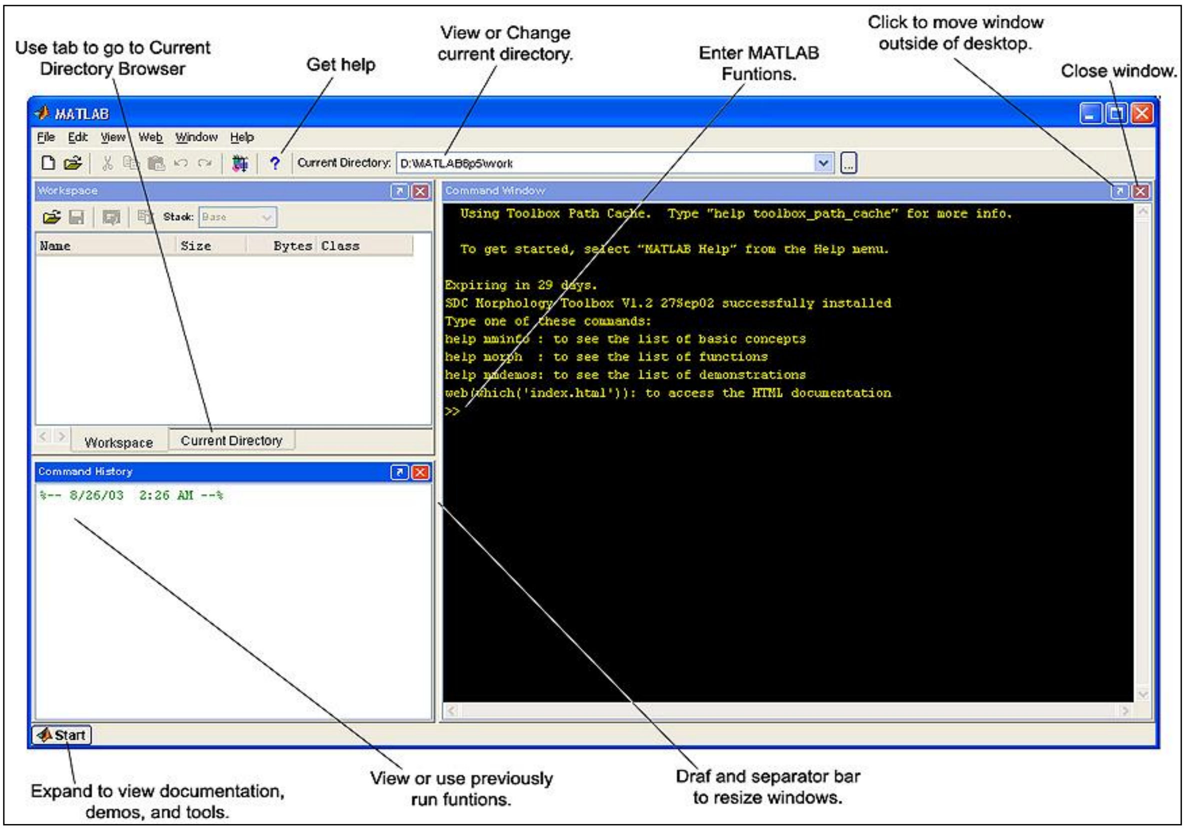


Figure 3:13 MATLAB desktop snapshot

MATLAB was initially written to provide easy access to matrix software developed by the LINPACK and EISPACK projects (Forsyth & Ponce, 2003). Nowadays, MATLAB engines exist as a corporation of the LAPACK and BLAS libraries (LAPACK, 2010). Many references gave detail explanations and informations about MATLAB applications, for instance Stanoyevitch (2004) and Lyshevski (2003).

MATLAB features are completed with additional application specific solutions, called ***toolboxes***. These toolboxes are important to the users who engaged in advance research. These comprehensive collections of MATLAB functions (M-files) extend the MATLAB capability to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others (Forsyth & Ponce, 2003). The Image Processing toolbox version 3.2 and the SDC Morphology toolbox version 1.3 are used extensively in these studies.

3.2.1 Image Processing Toolbox

The Image Processing Toolbox is one of the toolboxes developed by MathWorks, Inc. It is a compilation of image processing functions that extend the capability of the MATLAB® numeric computing environment. The toolbox supports most of the range of image processing operations such as Spatial Image Transformations, Neighbourhood and Block Operations, Linear Filtering and Filter Design, Image Analysis and Enhancement, and many more. For more detail, please refer to MathWorks (2010).

The Image Processing toolbox includes the Morphology operations feature, which is a special technique of image processing based on shapes. This technique is the main method that we used in this research, which is also provided by SDC Morphology Toolbox. Because of the similarity of the image processing operations, we conclude that it would be excellent to use it as a comparison or benchmark in determining how far the reliability and capability between these two toolboxes. Only then would the portion of Image Processing application can be seen and studied, especially in determining the number of object in the image (User's Guide, 2001). The SDC Morphology Toolbox will be discussed in detail in the next section, see Figure 3:14. It shows the position of image processing toolbox and SDC Morphology Toolbox in MATLAB program.

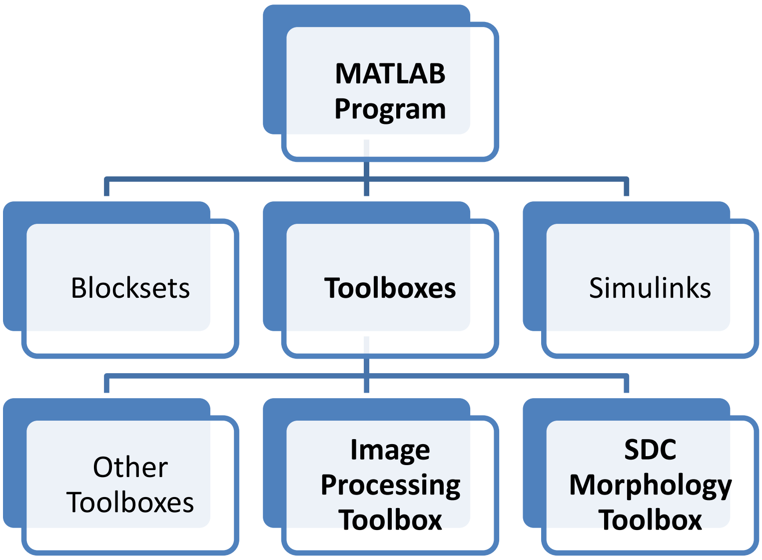


Figure 3:14 Diagram of Image Processing Toolbox in MATLAB Program

3.2.2 Images in MATLAB and the Image Processing Toolbox

The structure of data in MATLAB is an *array* of ordered set in real or complex elements. This structure will be utilised as the representation of *images*, real valued ordered sets of colour or intensity data. Basically, the images are stored in 2-D matrix arrays in which each of the elements of the matrix corresponds to a single pixel in the displayed image. For example, an image that is generated by 400 rows and 500 columns of different coloured dots is stored in MATLAB as a 400-by-500 matrix.

The 3D array image is also known as Red, Green, and Blue (RGB) image composed by 3 matrix layers. The first plane represents the red pixel intensities, the second plane represents the green pixel intensities, and the third plane represents the blue pixel intensities, refer to Figure 3:15, which shows a three-dimensional illustration of the RGB image.

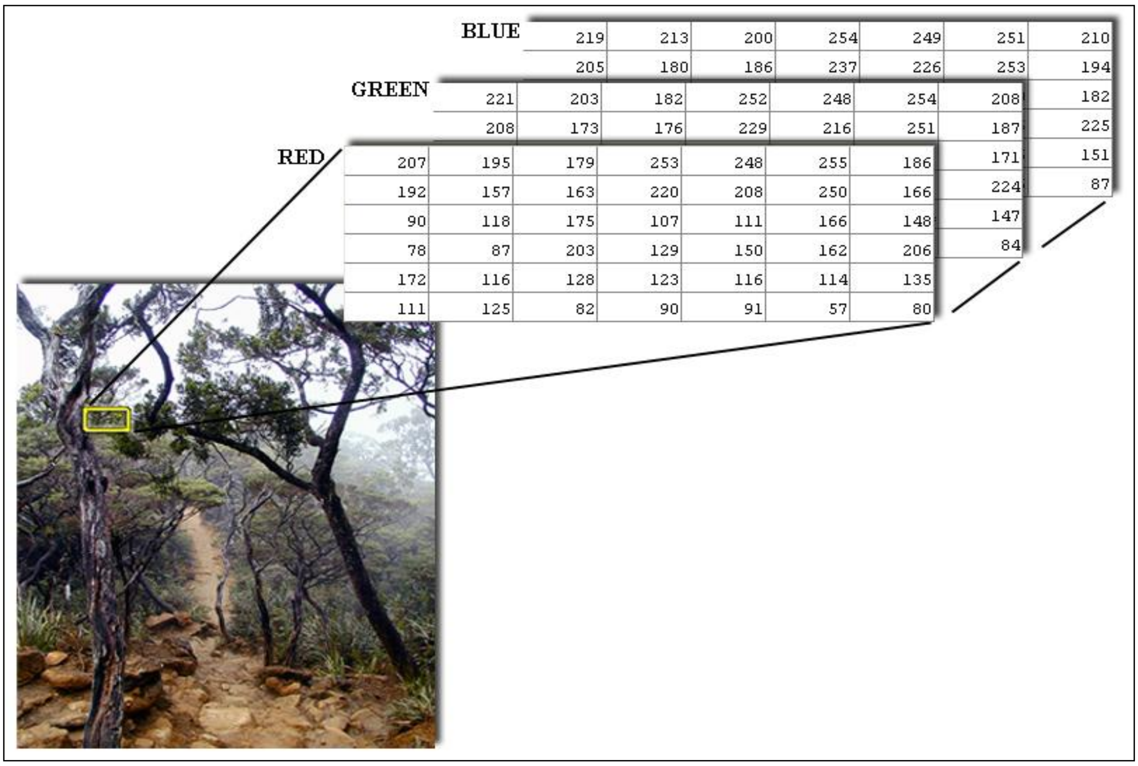


Figure 3:15 Illustration of RGB image

Consequently, the concept of matrix represents an image display that works with images in MATLAB. This is similar to operating with any other types of matrix data, and makes the full potential of MATLAB available for image processing applications. For example, selection can be generated from an image matrix to any single pixel, using normal matrix subscripting. For instance, the command;

```
I(5, 4)
```

Returns the value of the pixel at row 5, and column 4 of the image I, see the example below, by using the MATLAB script in its command window;

The variable of I is an image conversion of a random 10-by-7 matrix to unsigned 8-bit integers.

```
>> I = uint8(rand(10,7)*100)

I =

   74   44   62   13   18   13   16
   34   86    5   49   50    6   48
   88   39   54    3   45   37   49
   34   25   45   22   32   37   84
    5   35   86   32   38   48   80
   71   74   85   89   88   96   85
   95   65   47   31   76   34   60
   15   93   78   25   88   25   56
   41   83   65   43   45   58   61
    9   46    0   84   79   52   10

>> I(5, 4)

ans =

   32
```

The return value of image I at the pixel row 5, column 4.

3.3 SDC Morphology Toolbox

The SDC Morphology Toolbox for MATLAB was developed by SDC Information Systems. It is a collection of greyscale morphological tools that can be applied to image segmentation, non linear filtering, pattern recognition, and image analysis (SDC, 2011). The SDC Morphology Toolbox for MATLAB 6.5 is a joint program of Image Analysis and Signal Processing which is composed of discrete non linear filters based on lattice operations. These filters, called morphological operators, are useful for restoration, segmentation and quantitative analysis of images. The operators include the classical morphological filters, which are used for restoration and shape description, and modern connected filters and watersheds, which are used for image segmentation.

The SDC Morphology Toolbox deals with greyscale and binary images or signals and it is data type oriented. Therefore, most operators perform both greyscale and binary image processing and the choice of the appropriate (binary or greyscale) algorithm is automatic. The images may be represented by the following formats:

binary, 8-bit greyscale and 16-bit greyscale, where each pixel is represented; respectively by a *logical uint8*, a *uint8* and *uint16* data type.

Basically, the Morphological operators are developed by the improvement of two elementary operators, named dilation and erosion. Several operators are implemented by special fast algorithms to increase their efficiency (SDC, 2011). Some examples of these operators are Distance Transform, Watershed, Reconstruction, Labelling and Area Opening.

Dilations and erosions are parameterised by sub-images, called structuring elements. These structuring elements may be flatted as binary images or non-flat as greyscale images. The SDC Morphology Toolbox supports both kinds of structuring elements and represents them in a decomposed form, which increases the performance of the corresponding dilation and erosion.

3.3.1 SDC Morphology Toolbox System

SDC Morphology Toolbox system consists of three main parts. They are Basic Concept, Demonstrations, and Functions. All of the operators that belong to the SDC Morphology Toolbox start with '***mm***'. This will make us recognise the operators by name, which is different from the others using MATLAB (User's Guide, 2001).

The Basic Concept is an important item that a user is required to know well in order to use this toolbox. The basic concept of SDC Morphology Toolbox can be further subcategorised by three categories, which are Software Model and Conventions, Installation and Software Evolution, and 3D Processing.

The following part of SDC Morphology Toolbox is known as Demonstrations, which gives a full list of demonstration scripts or a set of instructions. This part is like a tutorial for the user who wants to learn the toolbox. Refer to Appendix 3C for the list of the demonstration scripts and its descriptions.

The last one is called Functions, which is the most important part in SDC Morphology Toolbox system. This part contains all the functions in the system (refer to Appendix 3D). Every function has its own description or help manual, which can be read by typing;

help *function name*

in the MATLAB Command Window.

CHAPTER 4

RESEARCH METHODOLOGY

4.1 Chapter Overview

This chapter is about the research methodologies that includes the image sample preparations in Detector Etching process. There are also flow charts supported methodologies of morphological image processing, which utilised the MATLAB and toolboxes.

4.2 SDC Morphology Toolbox Installation

The SDC Morphology Toolbox is designed to be used with MATLAB, but it is not a part of it. As a third party product, the SDC Morphology Toolbox must be obtained from another company, known as SDC Information System at www.mmorph.com. The Toolbox is available for Windows systems as well as for UNIX systems. After downloading it, the installation is done by MATLAB script file. To use the toolbox’s functions as a free evaluation, a serial code is needed, which must be obtained from the SDC homepage after the installation is completed. The SDC Morphology Toolbox operates on MATLAB Version 5 and above. One of the most significant factors about SDC Morphology Toolbox, compared to other Image Processing toolboxes, is that it does not depend on any other toolboxes (SDC, 2011).

4.3 Detector Etching and the Sample Images of Nuclear Track

The LR 115 detectors (LR 115 film, Type 2) used in this research was purchased from KODAK, Malaysia. The LR 115 is a film which consists of a layer of cellulose nitrate on a 100 μm clear polyester base substrate. The LR 115 detectors were cut to the size of about 2.25 cm^2.

For us to get detectors with latent track, the LR 115 detectors were exposed to a radiation source. The radiation source was Monazite (reddish brown colour phosphate). It is in the form of granule and contains the elements thorium, lanthanum, and cerium. The LR 115 films were exposed to monazite up to 43 hours.

After the detectors were exposed to monazite in detecting radiation particles such as alpha particles, the detectors were then etched in a 2.5 N aqueous solution of NaOH at 60°C. The NaOH solution was placed in a water bath in which the temperature was kept to within ±1°C. The LR 115 was etched for 1 hour.

After the etching period, the detectors were removed from the etchant and rinsed with distilled water for 15 minutes. For the drying processes, the images are fanned for 10 minutes. After drying, each portion of the detectors was placed between the transparent plastic slides (Figure 4:1). These slides with detectors within them were observed using optical microscope (Model Axioskop, Carl Zeiss MicroImaging Inc.). Then, the images of the nuclear track in each detector were captured after the best image of the nuclear track was seen through the microscope.

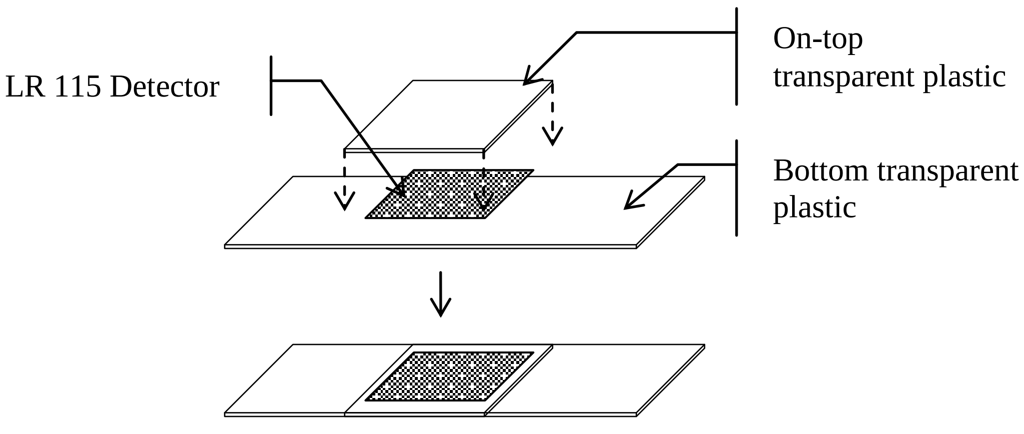


Figure 4:1 The slide of LR 115 detector

For each detector with exposed time differences, 10 sample images of nuclear track were captured from different areas of the sample using the microscope supported with digital camera (Model Canon, Power Shot G5 with 5 megapixel CCD). This means that there were 100 sample images of nuclear track taken using the system with the combination of microscope, digital camera and computer. Figure 4:2 showed the sketch diagram of the equipments used to capture the image of nuclear track. These images were focused through the microscope and directly captured from a computer software (Canon Utilities RemoteCapture, Canon Inc.) via digital camera.

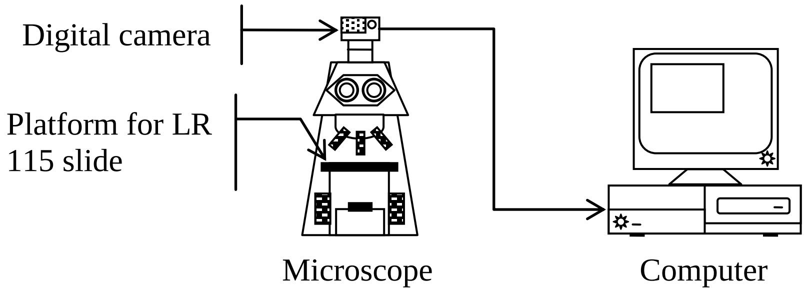


Figure 4:2 The sketch diagram of the equipments to capture image

While the nuclear tracks were focused, the nuclear track images were magnified up to 10 times magnification for eyepiece lens, 20 times for nosepiece lens and 1.3 times for camera magnification. This implies that the images were magnified up to 260 times magnification. During image capturing, all of the size of images was set to (640×480) pixels and saved as RGB mode. Then, these images were converted to greyscale mode in preparing for the next process. This will be covered in section 4.5.

4.4 Segmentation Limit of Overlapping Objects in Image

The ideal tracks would be isolated circular or elliptical shapes distributed evenly throughout the image. Red images of those tracks may have jagged edges and in some cases overlapping tracks. To ascertain the effectiveness of the software to separate overlapping tracks, a series of artificial images were used as test images. The images prepared were in various shapes with varying degrees of overlap.

The procedure starts with the artificial images being generated using Adobe Photoshop. The characteristic of the image has been set so that the image contains 2 similar shape objects, which were horizontally arranged and have a gap. The objects were considered as first object for the left position and second object for the right position. Besides that, the background of the generated image is set to 87% of grey level and the foreground is set to 15% of grey level (pixels value of the objects). In the first case, there were 2 similar circles sizes (75 pixels of diameter). Please refer to Figure 4:3. Then, these objects were programmed to get closer to each other, until they overlapped with each other by pixels. The image was (320×240) pixels generated from 0 pixel that were overlapping, a few before and after those objects cannot be segmented. The objects or overlapping objects in the image were segmented and their areas were determined.

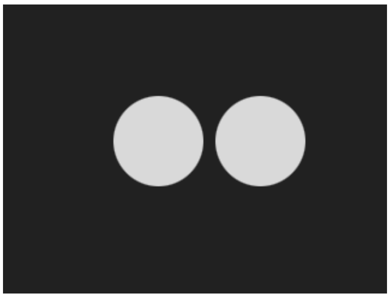
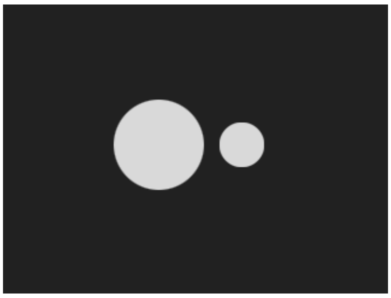
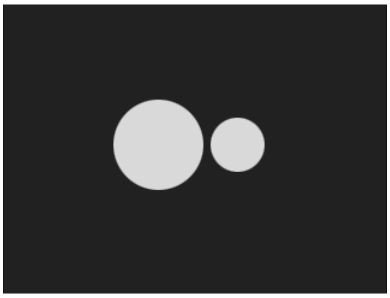


Figure 4:3 The image sample of two circles for Overlapping limitation studies

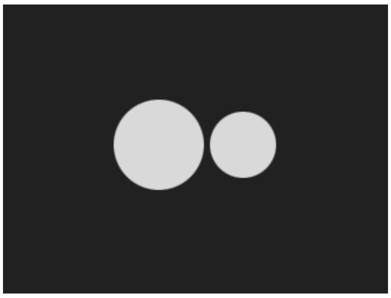
There were 2 circles with different sizes (75 pixels and 37, 45, 55, and 65 pixels of diameter, see Figure 4:4), 2 shapes of ellipses with the same sizes (75 pixels major axis and 45 pixels minor axis, see Figure 4:5), and 2 shapes of square with the same sizes pixels (64×64) and horizontal diagonally arranged pixels, see Figure 4:6.



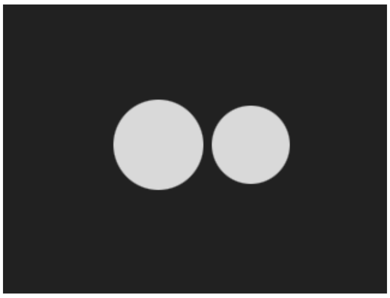
(a)



(b)



(c)



(d)

Figure 4:4 Test images with circles of different diameters

(a) 75 pixels and 37 pixels (b) 75 pixels and 45 pixels

(c) 75 pixels and 55 pixels(d) 75 pixels and 65 pixels

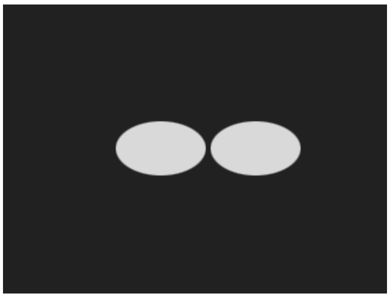


Figure 4:5 Test images of two ellipses with same size

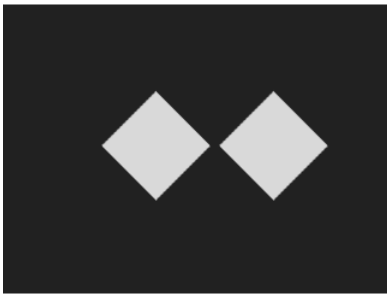


Figure 4:6 Test image with two squares horizontally and diagonally arranged

4.5 Grain Counting and Granulometry to the Nuclear Track Image

All of the 100 sample images of the nuclear track that we had captured in Section 4.3 were processed based on what we had discussed in sections 2.5, 2.6 and 2.7. Each of the image nuclear tracks was processed based on the flow chart on Figure 4:7, Figure 4:8 and Figure 4:9. At the beginning, the image was extracted through the extraction operation (Figure 4:7) then the image was segmented through the segmentation operation (Figure 4:8).

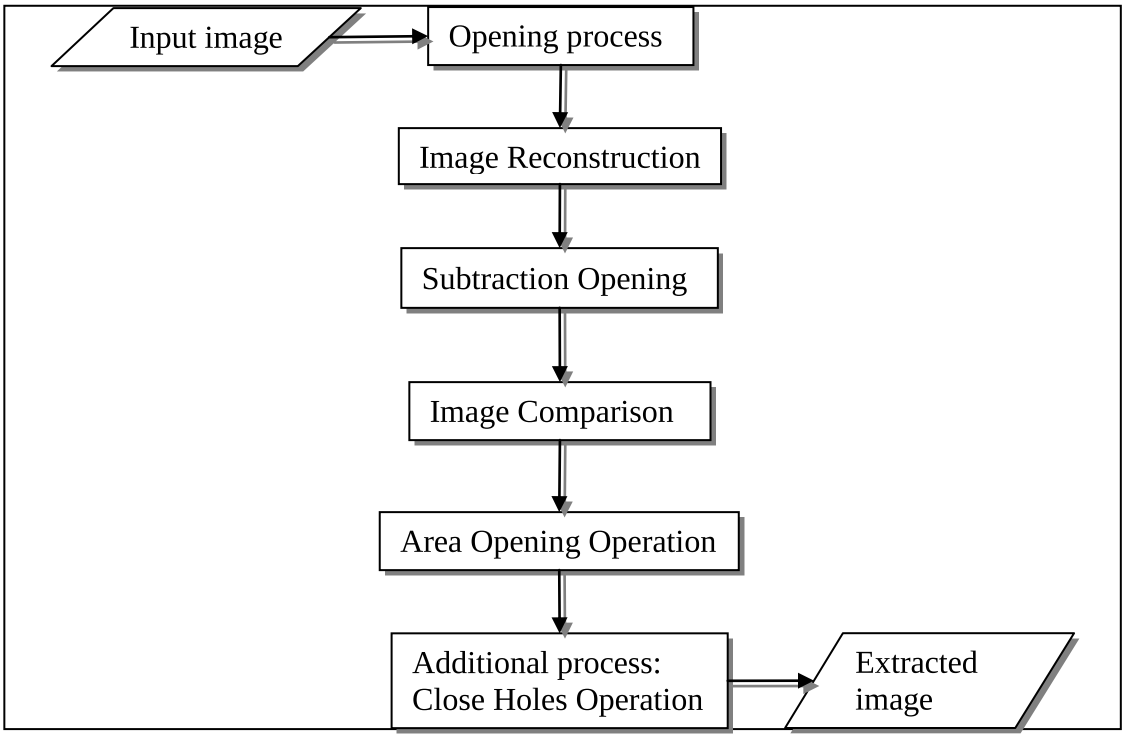


Figure 4:7 Extraction Operations

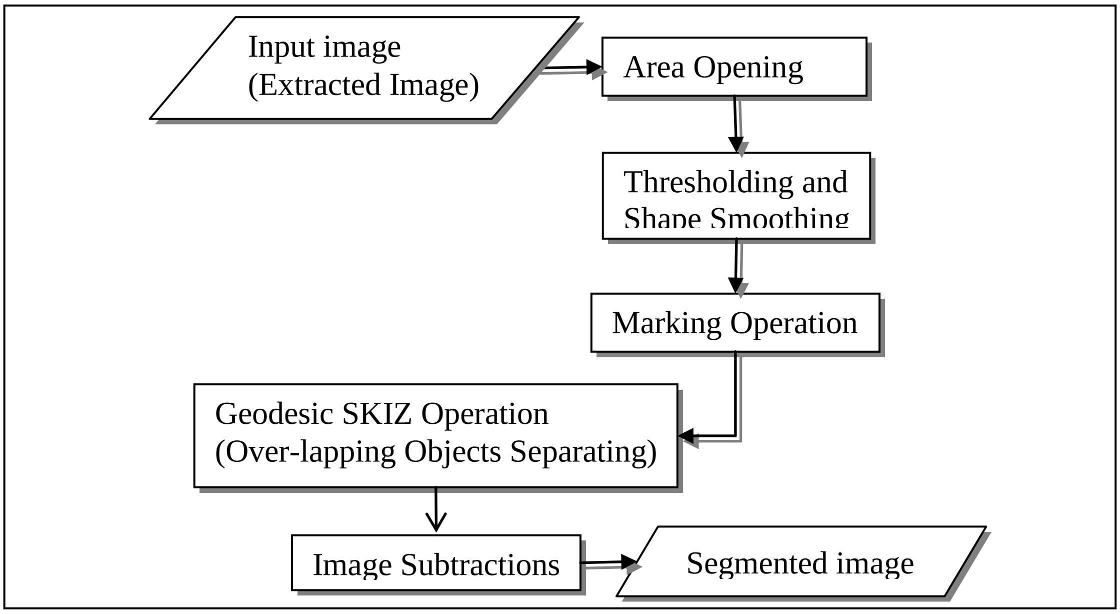


Figure 4:8 Segmentation Operations

After the extraction and segmentation operation, the granulometry part begins when the images were operated with MATLAB's functions, this is to get the information of the number of nuclear tracks, sizes (minimum and maximum) and their mean size. Later on, the images were applied onto these functions, in which the data was extracted. Please refer the data extracting flow chart as follow, Figure 4:9.

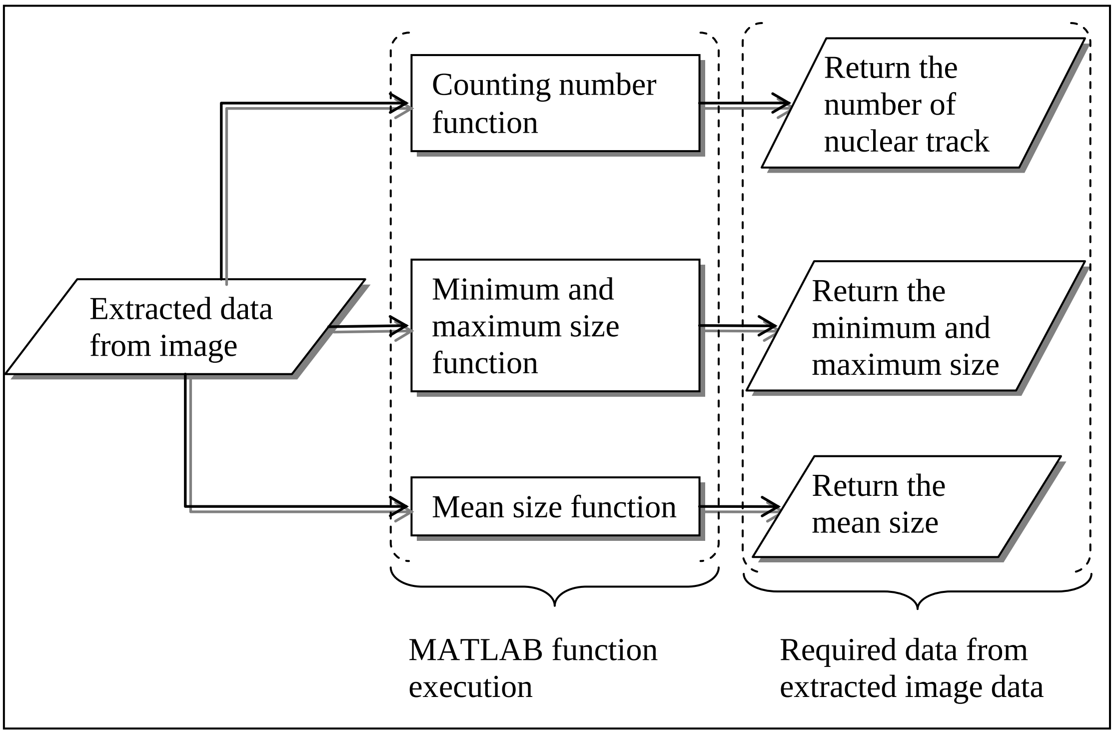


Figure 4:9 The flow chart of required data from extracted image data.

With the aim to have more result evaluations for each toolbox's result, the nuclear tracks in the images were counted manually. These manual results were then used as the standard results to evaluate the toolboxes' results.

This procedure was repeated to both of the toolboxes separately in order to get the information of the grain counting and the granulometry from each toolbox.

4.6 Nuclear Tracks Counting Simulation

Based on section 1.4, two assumptions have been made that the number of nuclear tracks on the nuclear tracks detector will constantly increase due to time and the emission of the radiation particles from the Monazite radiation source to the nuclear track detector was a random emission. This means, that the distribution of the nuclear track position on the nuclear track detector was a normal distribution. Based on these two assumptions, a simulation of the nuclear tracks counting was done, Figure 4:10 shows the simulation flow chart, which '*i*' number is the increment number, which was less than or equal to 20×20, which is equal to 400 and '*n*' is the number that represent the number of the nuclear tracks. The simulation was iterated to 100 times.

The simulated data such as the number of nuclear tracks and exposed time was obtained. The graph of the simulated number of nuclear tracks versus exposed time was plotted and was compared to linear rate of the nuclear tracks.

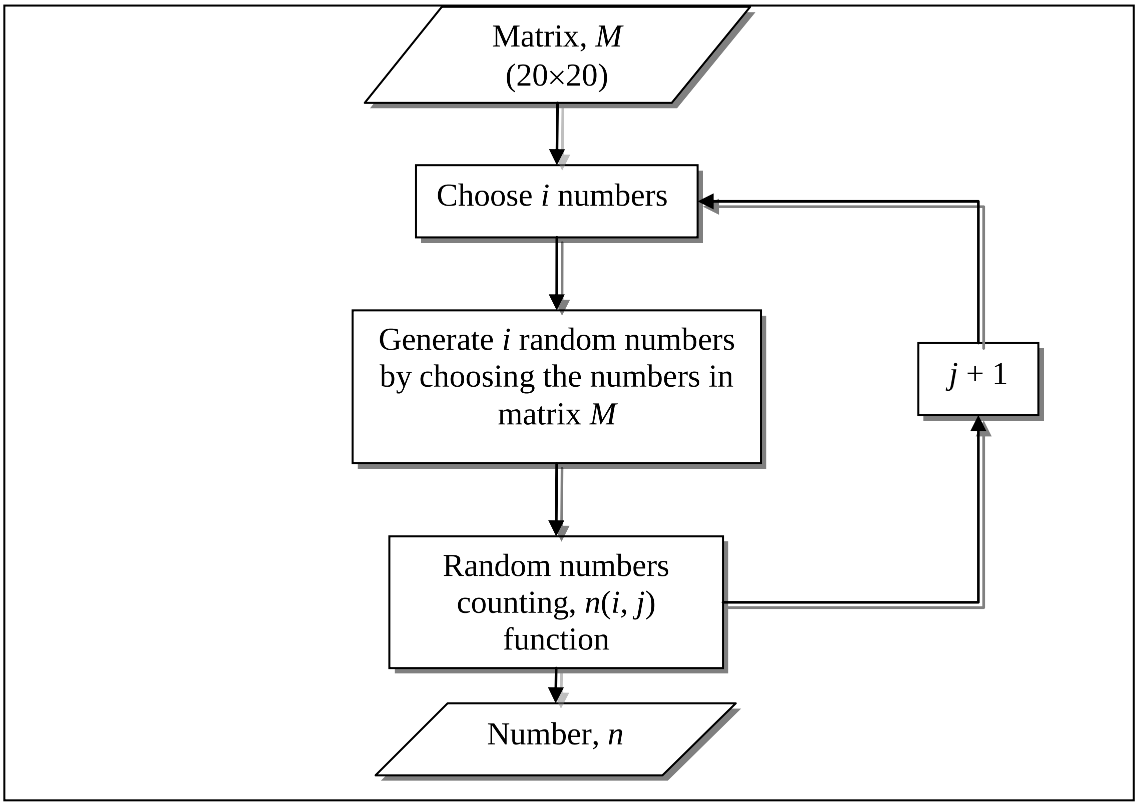


Figure 4:10 The flow chart of the nuclear tracks counting simulation

CHAPTER 5

RESULTS DISCUSSIONS

5.1 Chapter Overview

In this chapter, the analysis and result of the research findings are presented. Basically, this research focuses in finding the limitation of over-lapping objects using different shapes. The research also concentrates on grain counting operation, simulation of the emitted radiation particle to the same track position and granulometry. In granulometry the SDC Morphology Toolbox and Image Processing Toolbox were applied to the research images in order to discover their capabilities and compared them to the manual method of counting. For these purposes, the numbers of nuclear tracks are obtained as research data and are compared to standard data that acts as reference.

5.2 Images of Nuclear Track

For this research, we took images from a few image sources and used it as image samples. Firstly, the microscopic images from our Radiation Laboratory were captured by a special microscope. These images are the image of special plastic films (LR 115 Type 2), which has been exposed to nuclear radiation due to time. The images are also known as nuclear track images. The image samples of radiation exposed film were captured using a high capability microscope and digital camera, in which we considered the radiation exposed time that ranges from 0 second to 43 hours. Below are 2 images from our image samples; Figure 5:1 which shows, 1 hour radiation exposed image and Figure 5:2 which shows, 43 hours radiation exposed image.

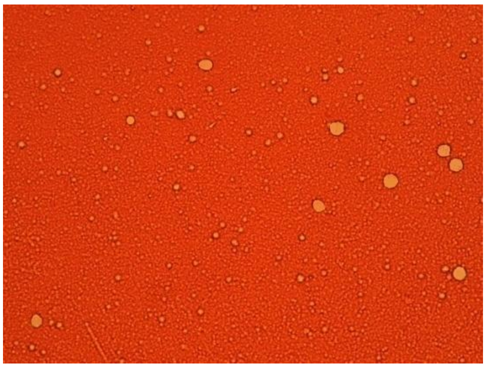


Figure 5:1 The Image after 1 hour radiation exposed

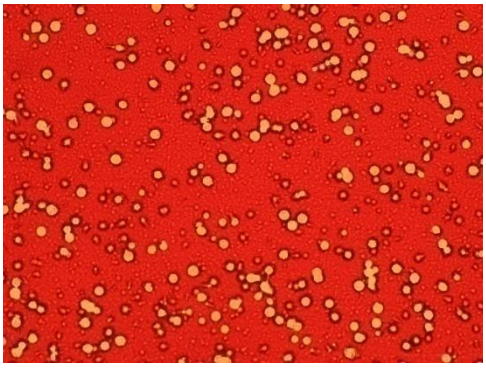


Figure 5:2 The Image after 43 hours radiation exposed

In order to get more image samples, we have also created a generated image or artificial image using image editing software, such as Adobe Photoshop and Paint. This source gives us the options to generate a certain image with its own problem characteristic, where this image will be used as the image sample to test our research techniques. These tasks will enable us to find out about our research techniques capabilities in facing different image processing problems.

After the detector etching process is completed, the images of the nuclear tracks are captured using laboratory microscope, digital camera and a computer. As stated in Section 4.3, the images were taken based on exposed time to the radiation source called Monazite. The following Figure 5:3 shows us the first images of the nuclear track from 10 nuclear track images with different exposure time. The rest of the images can be seen in Appendix 4A.

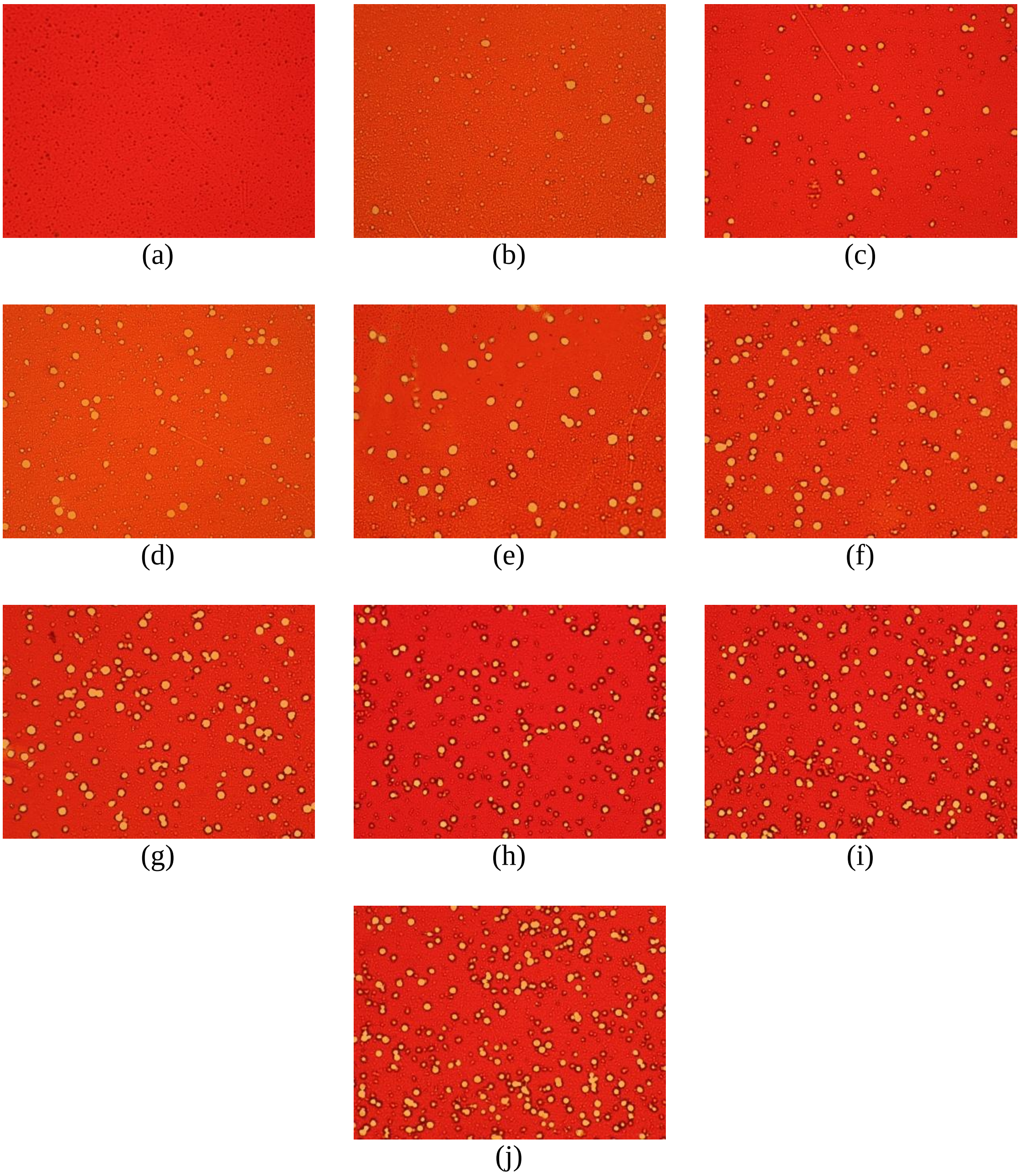


Figure 5:3 The exposed times, (a) 0 hour, (b) 1 hour. (c) 3 hours, (d) 5 hours, (e) 7 hours, (f) 10 hours, (g) 14 hours, (h) 20 hours, (i) 31 hours and (j) 43 hours

5.3 Limitation of Segmented Over-Lapping Objects

In section 4.4 the data that was extracted was divided based on the shapes of the objects that over-lapped that are categorized as Over-lapped Pixel, over-lapping area of the Objects (First Object (FO) and Second Object (SO)). If the over-lapping objects cannot be segmented anymore, one of the object data (second object) will result in zero object area; where the second object is considered as non-existent. Then, the other object data (first object) makes the object area bigger than the original one.

Some portion of the data results in linear distribution which conforms to the linear equation. The best straight line fitting to the data can be obtained by using a tool based on least-squares method and this can be determined by using the Basic Fitting tool in MATLAB.

5.3.1 Over-Lapping of the Two Circles

The data of the over-lapping process of two circles (75 diameter pixel) are presented in the following graphs. Figure 5:4 shows the graph where the segmentation of the objects ends when the objects were over-lapping with more than 60 pixels. This means that the limit of segmentation for these objects is 60 pixels. While the image segmentation was processing in the over-lapping range of between 0 pixel to 60 pixels, we found that both of the objects' area decrease (decay) with 28.62 pixels for one over-lapping pixel, see Figure 5:5.

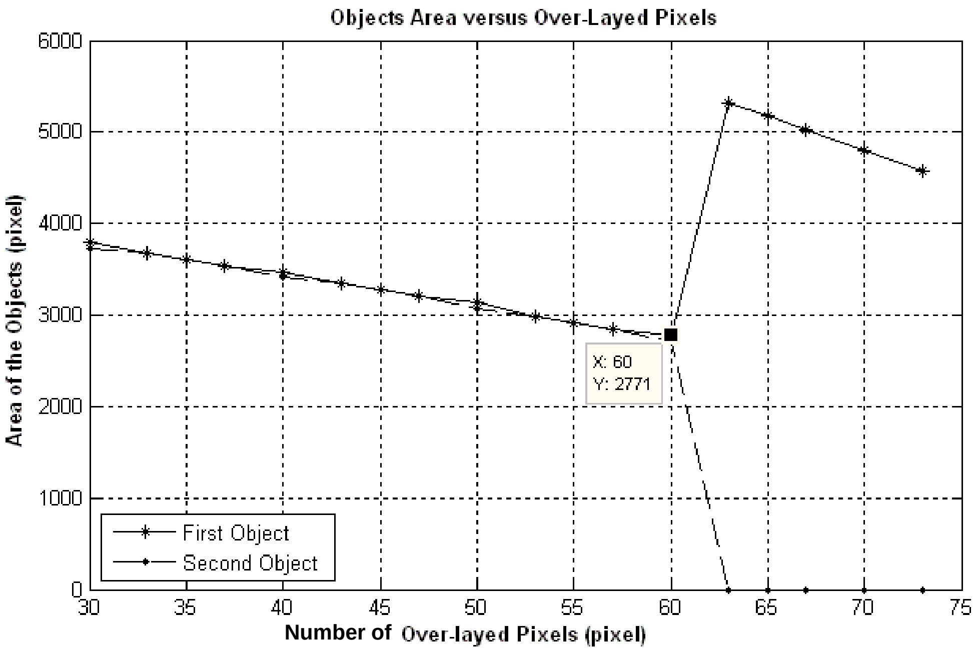


Figure 5:4 Over-lapping Process of two circles with 75 pixels diameter

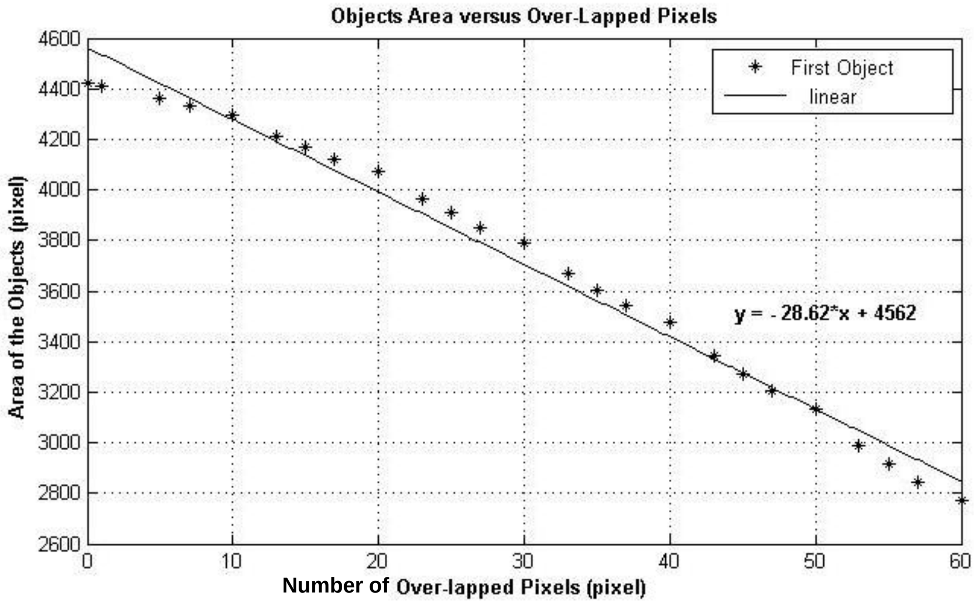


Figure 5:5 Over-lapping Process with the objects that can still be segmented

For the other case study where the second object is smaller than the first object, as discussed in Section 4.4, the data are represented in a single graph, see Figure 5:6. The graph shows that the limits of segmentation are increasing due to the size of the second object. Then, we found that the limit of segmentation is increasing with each 1.13 pixels over 1 pixel of diameter, as seen in Figure 5:7.

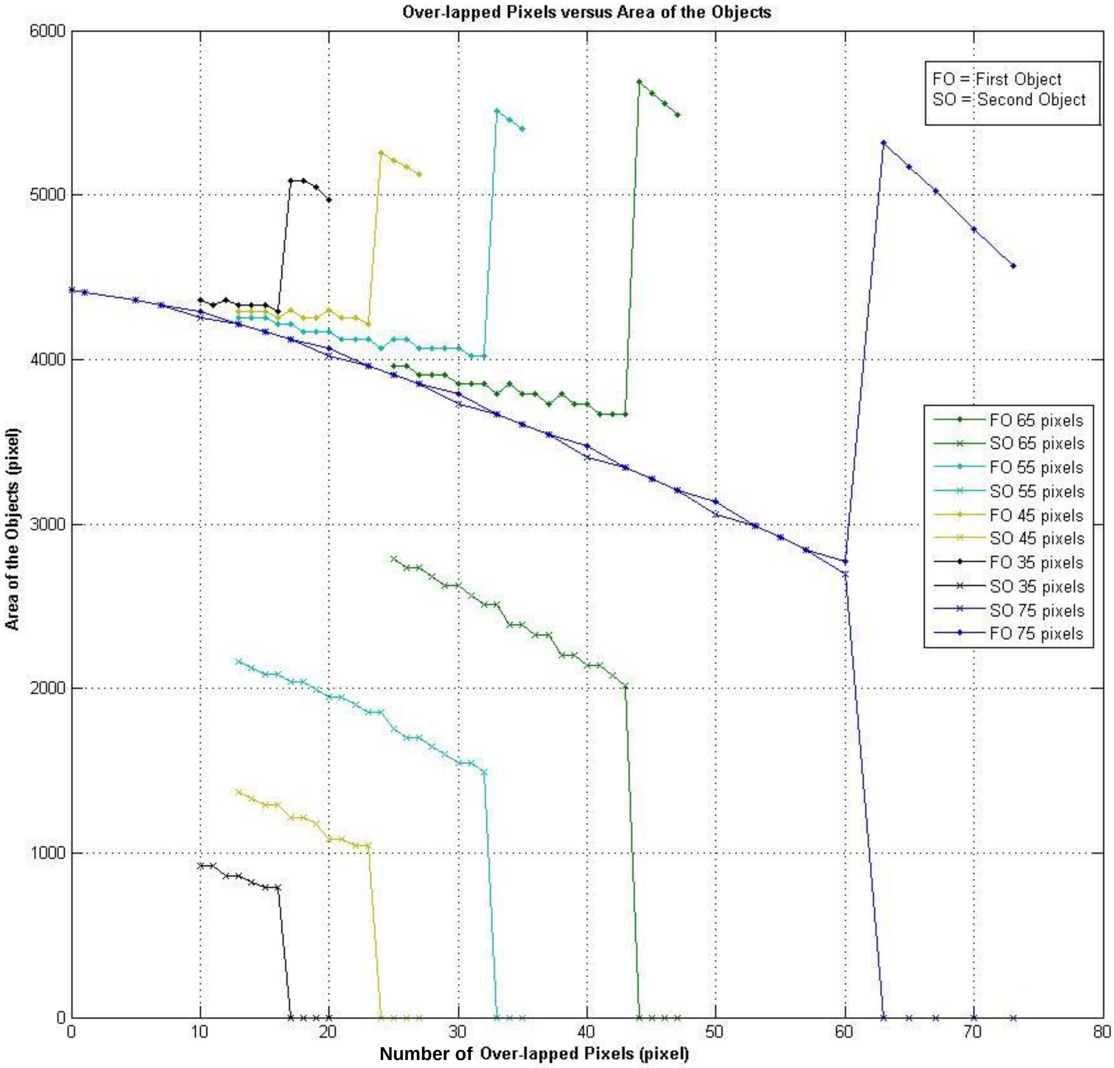


Figure 5:6 Over-lapping Process with two circles of various diameters

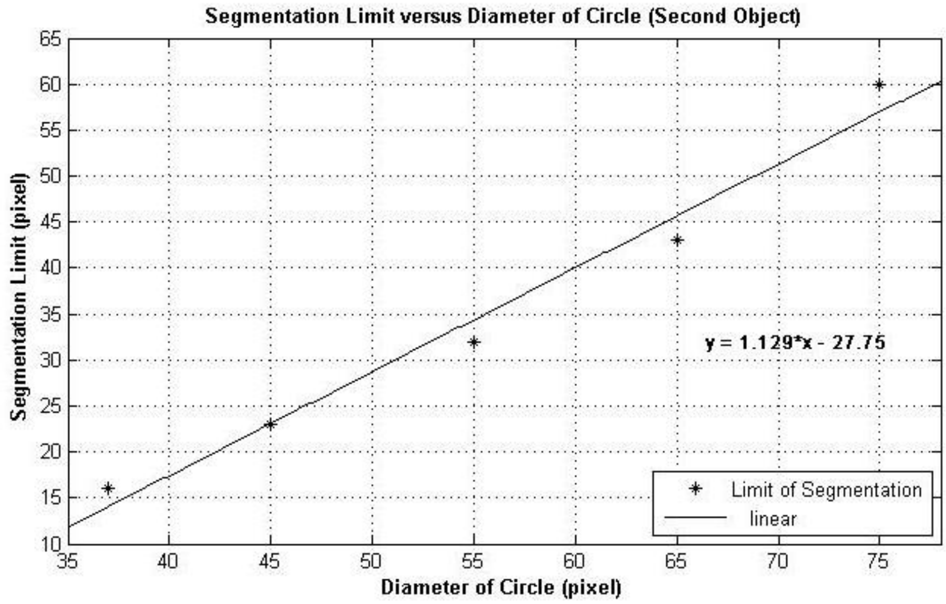


Figure 5:7 Limitation of Segmentation based on differences of circle size (diameter)

As seen in Figure 5:6, the decay in the first object area correspond to over-lapped pixels. Figure 5:8, Figure 5:9, Figure 5:10 and Figure 5:11 show the graph for each case and provide the decay value based on linear Basic Fitting. As summary results, which is the gradient of the graphs present the decay rate of the first object area, which are shown in Table 5:1. (page 68).

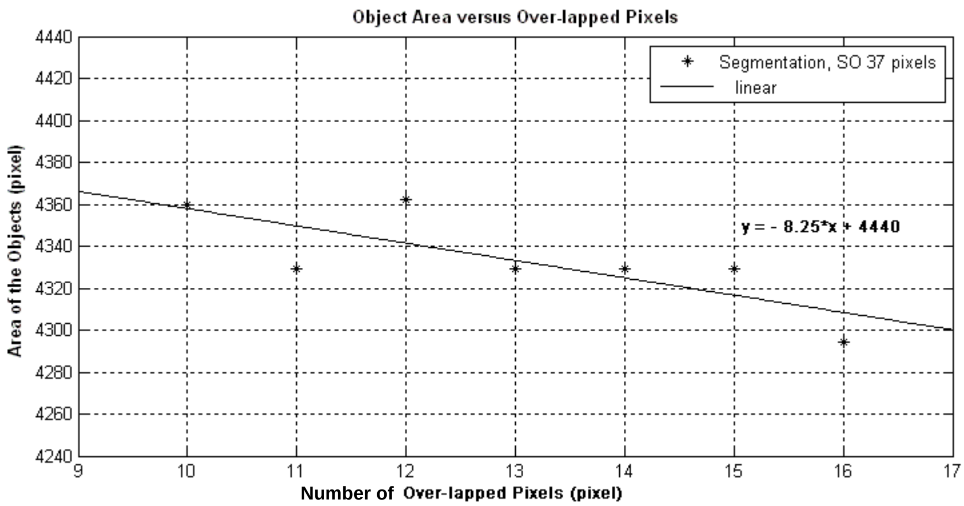


Figure 5:8 Segmentation process, diameter FO = 75 pixels and SO = 37 pixels

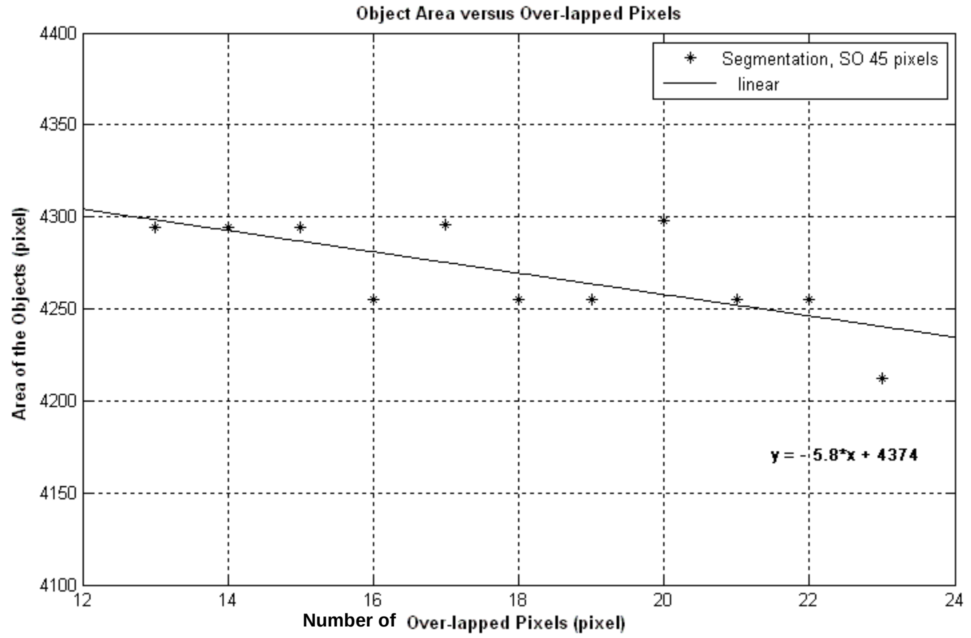


Figure 5:9 Segmentation Process, diameter FO = 75 pixels and SO = 45 pixels

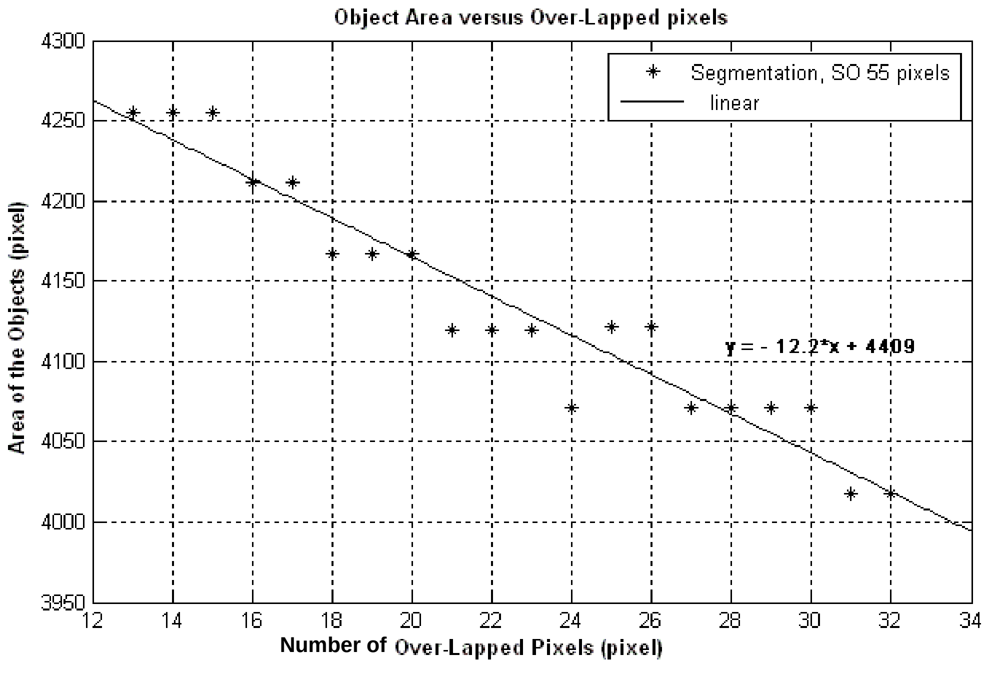


Figure 5:10 Segmentation Process, diameter FO = 75 pixels and SO = 55 pixels

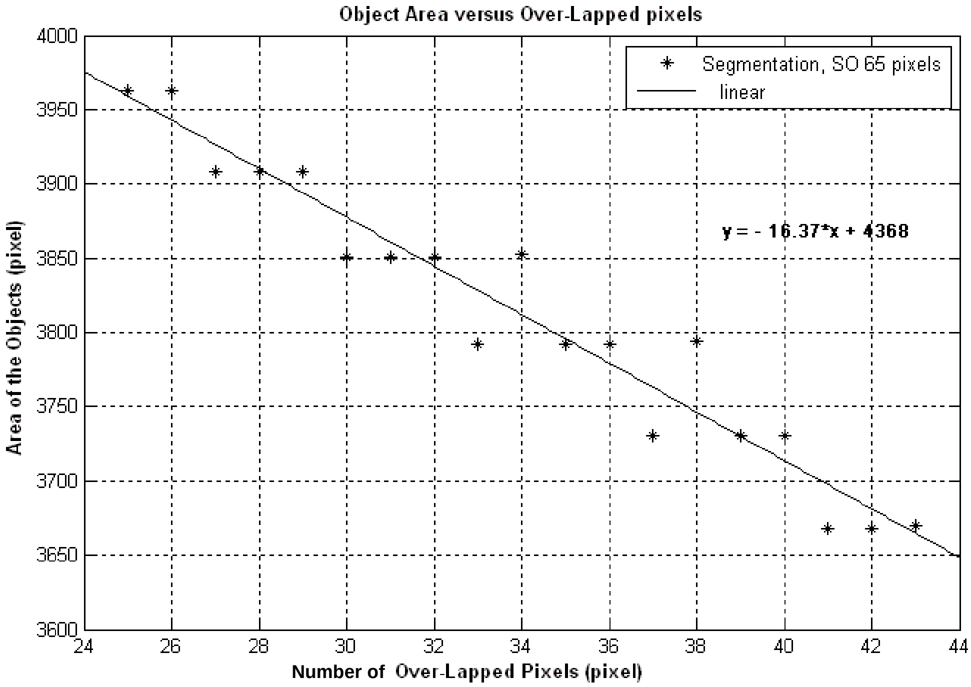


Figure 5:11 Segmentation Process, diameter FO = 75 pixels and SO = 65 pixels

Table 5:1 Decay rate of the first object area based on the second object sizes

Diameter (pixel)		Decay rate of Object Area (pixel/over-lapped pixel)
First Object	Second Object	
75	37	-8.3
75	45	-5.8
75	55	-12.2
75	65	-16.4
75	75	-28.6

5.3.2 Over-Lapping of the Two Ellipses (Oval Shapes)

The segmentation data of the over-lapping of two ellipses is illustrated in Figure 5:12. The graph shows that the limit of segmentation for these two ellipses is 55 over-lapping pixels. Figure 5:13 shows the area decay rate while segmentation is 16.58 pixels over 1 over-lapping pixel.

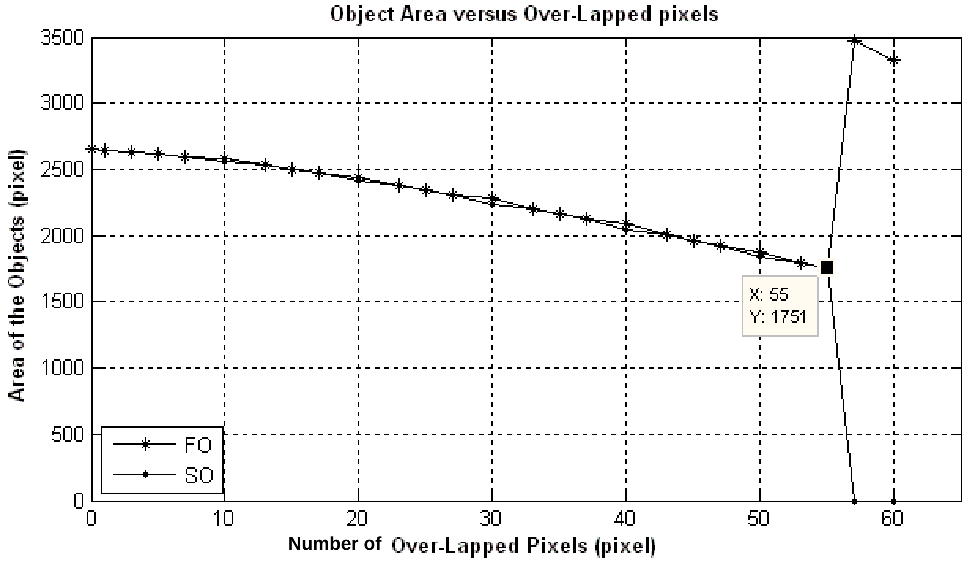


Figure 5:12 Over-lapping process of two ellipses

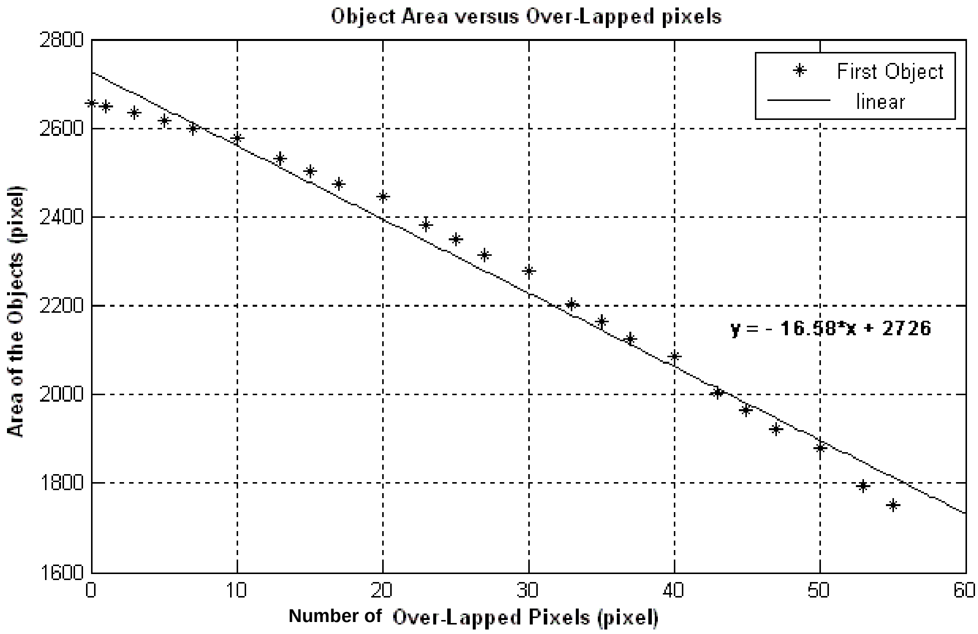


Figure 5:13 Segmentation process of two ellipses

5.3.3 Over-Lapping of the Two Squares Diagonally Arranged

The segmentation data of the two over-lapping diagonally arranged squares are illustrated in Figure 5:14. The graph shows the limit of segmentation for these two squares is 68 over-lapping pixels. Figure 5:15 shows the area decay rate, while segmentation is 30.99 pixels over 1 over-lapping pixel.

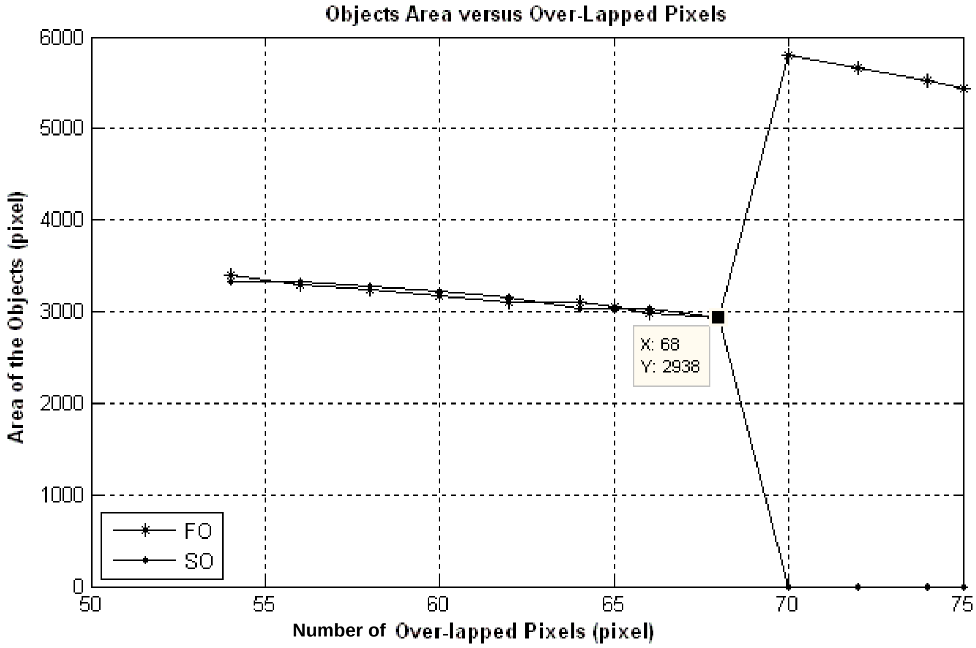


Figure 5:14 Over-lapping Process of two squares diagonally arranged

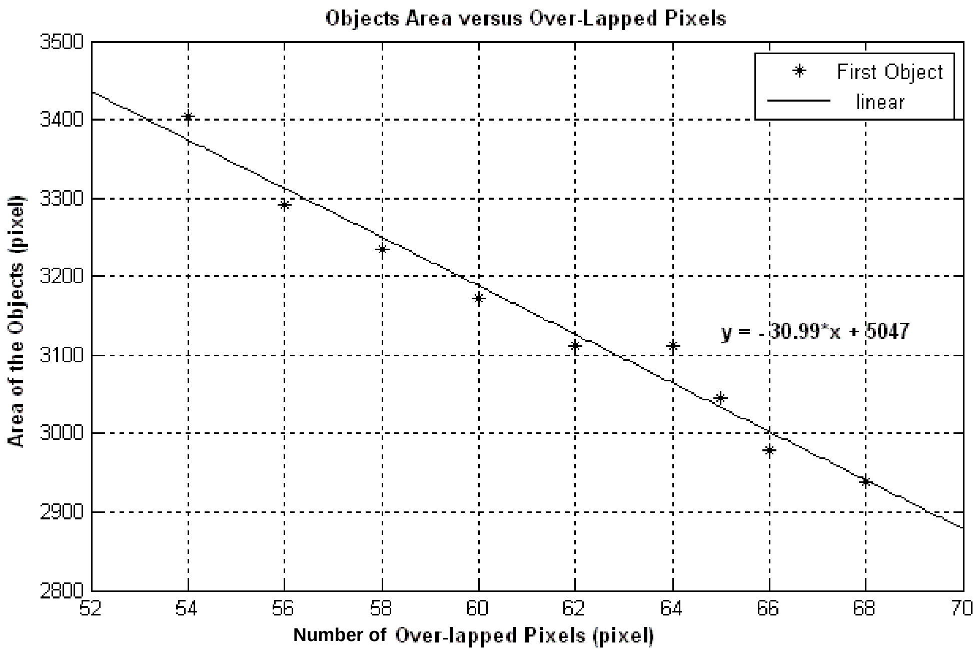


Figure 5:15 Segmentation Process on two squares diagonally arranged

5.4 Limitations of Segmented Over-Lapping Objects Outcome Analysis

Based on section 5.3, we can conclude that the limitation of segmented over-lapping object can give effect to the nuclear track counting. We assume that all the shapes of the nuclear tracks images are circular shape and the radiation activity of the radiation source are random activity. This condition will allow the nuclear track to be randomly emitted onto the nuclear track detector. As a result, there were possibilities that there was nuclear tracks image over-lapping more than segmentations limit. If these occur it will affect the result of the nuclear track counting by decreasing the number of counting; even though it cannot be detected by manual counting.

5.5 Nuclear Track Counting and Granulometry

The results of the nuclear track counting using the toolboxes and manual method seems to have a similar pattern; see Figure 5:16. However, there are slight differences between the three procedures, especially after 10 hours of exposed time.

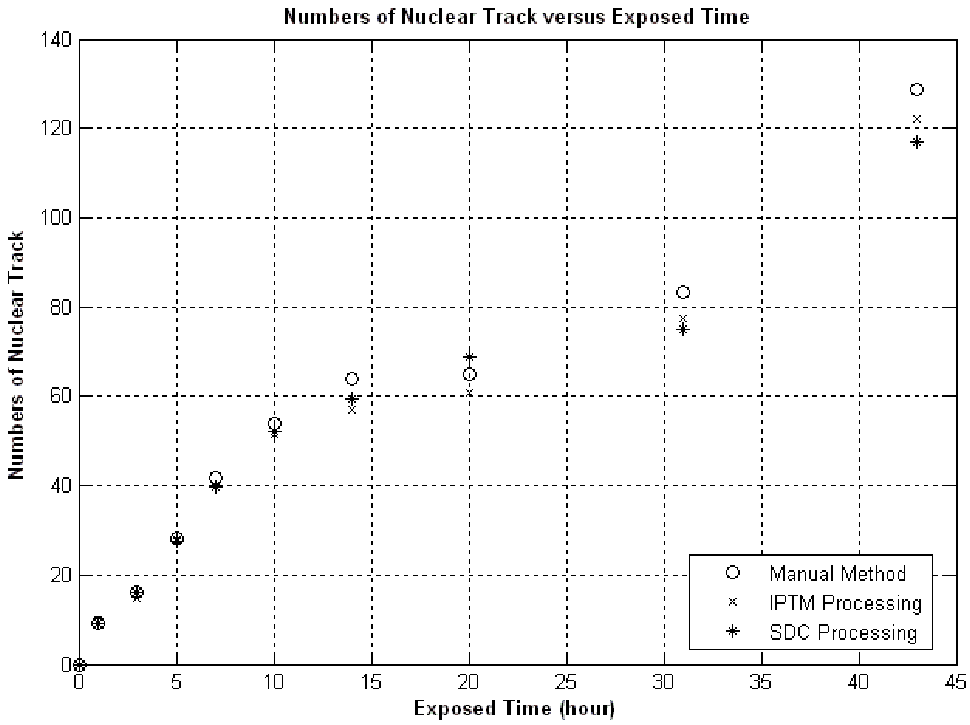


Figure 5:16 The graph of Numbers of Nuclear Track versus Exposed Time

From the graph, the Manual Method, IPTM Processing and SDC Processing illustrate that starting from 0 hour to 10 hours the data form linear rate of number of nuclear tracks over exposed time (nuclear tracks rate). After 10 hours of exposed time, the nuclear tracks counting were broken down till the final data, 43 hours of exposed time. Hence, linear Basic Fitting (MATLAB feature) was fitted to each graph at the linear portion in order to see the differences between nuclear tracks rate of each method and the number of nuclear tracks with exposed time equal to 0 second or recognised as error number of nuclear tracks, see Figure 5:17, Figure 5:18 and Figure 5:19. The final results are shown in Table 5:2.

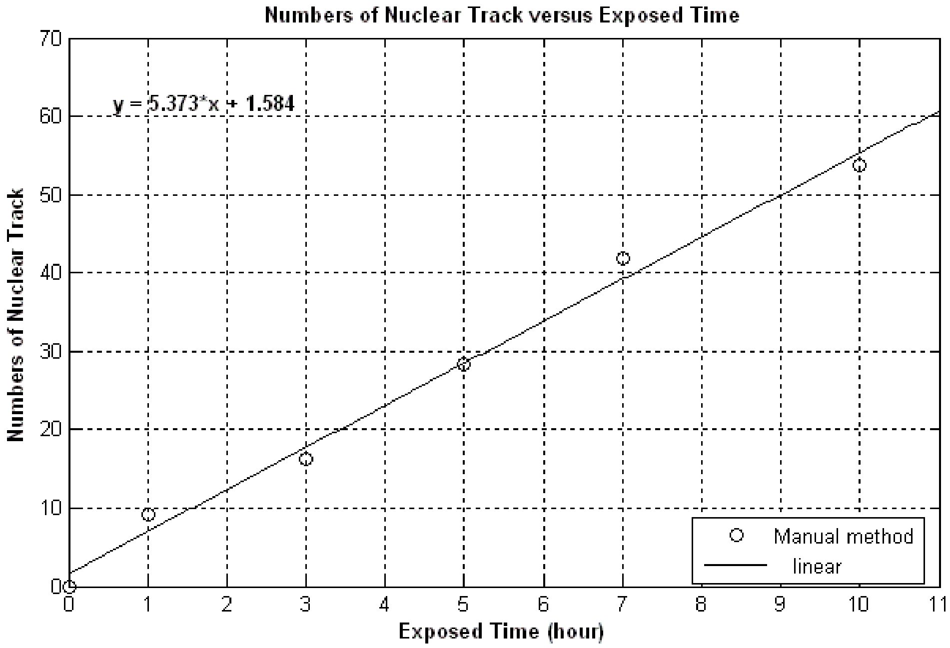


Figure 5:17 Manual Counting Method to the number of nuclear track

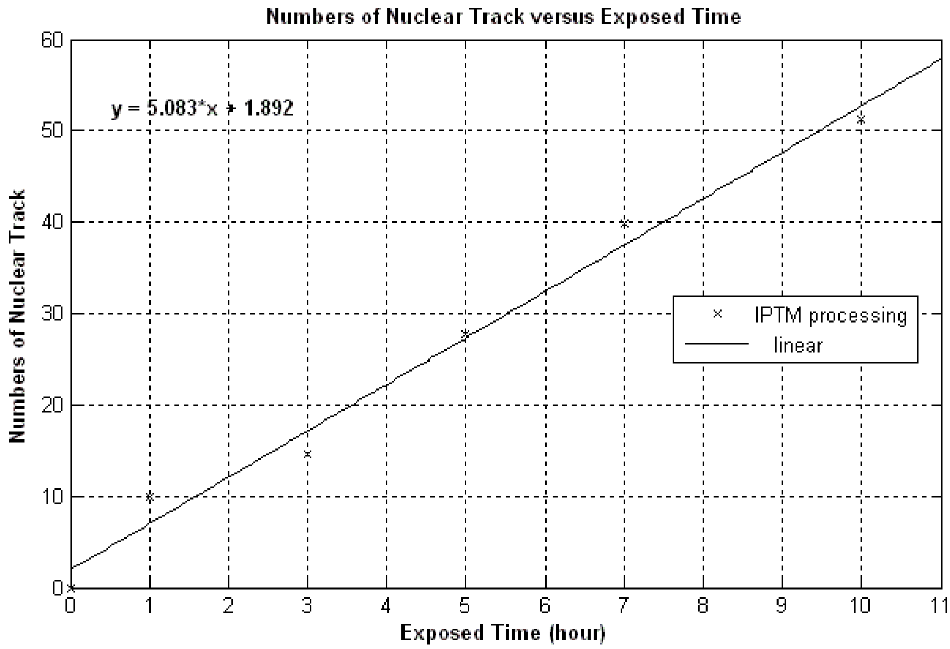


Figure 5:18 IPTM Processing to the number of nuclear track

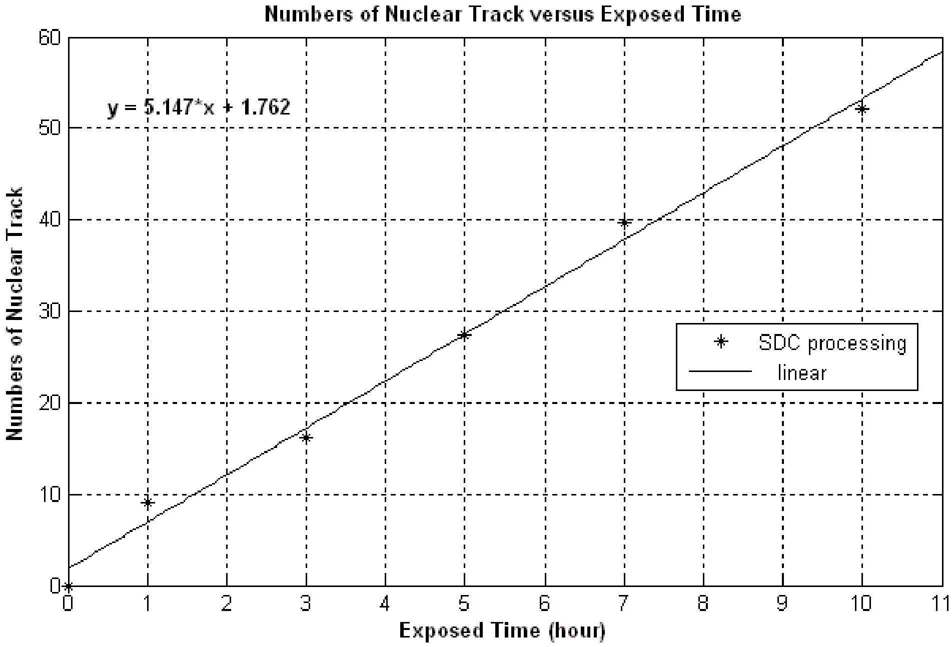


Figure 5:19 SDC Processing to the number of nuclear track

Table 5:2 The results of counting methods

Counting Method	Nuclear tracks rate (number/hour)	Percentage difference to the Manual method (%)	Error number of nuclear tracks
Manual	5.373	0	1.584
IPTM	5.083	5.4	1.892
SDC	5.147	4.2	1.762

Based on the results in Table 5:2, the toolboxes generate almost the same rate of the number of nuclear tracks. The IPTM and SDC processing generate less nuclear tracks rate than the manual method with the percentage difference of 5.4 and 4.2 nuclear track rate respectively. If the Manual method is considered as accurate data, this means that SDC processing generates improved result than the IPTM processing. This result is also supported by the error number of the nuclear tracks, which the SDC processing produces the closest number to the manual method of counting with the difference of 0.18 nuclear tracks. Then, the IPTM processing gives the difference of 0.31 nuclear tracks respective to the manual method.

5.5.1 Simulation of Nuclear Track Counting

Figure 5:20 shows the graph of simulated nuclear tracks counting versus exposed time resulting from the nuclear tracks counting simulation after 100 times iteration. Figure 5:21 shows the graph of its average. Both of the graphs depict the profile that the distribution of the nuclear track number due to time was non-linear and its rate decreasing. Based on this simulation, we can say that there were radiation particles penetrating on the track detector at the same place or position. The probability that radiation particles penetrate the same position also increases due to time.

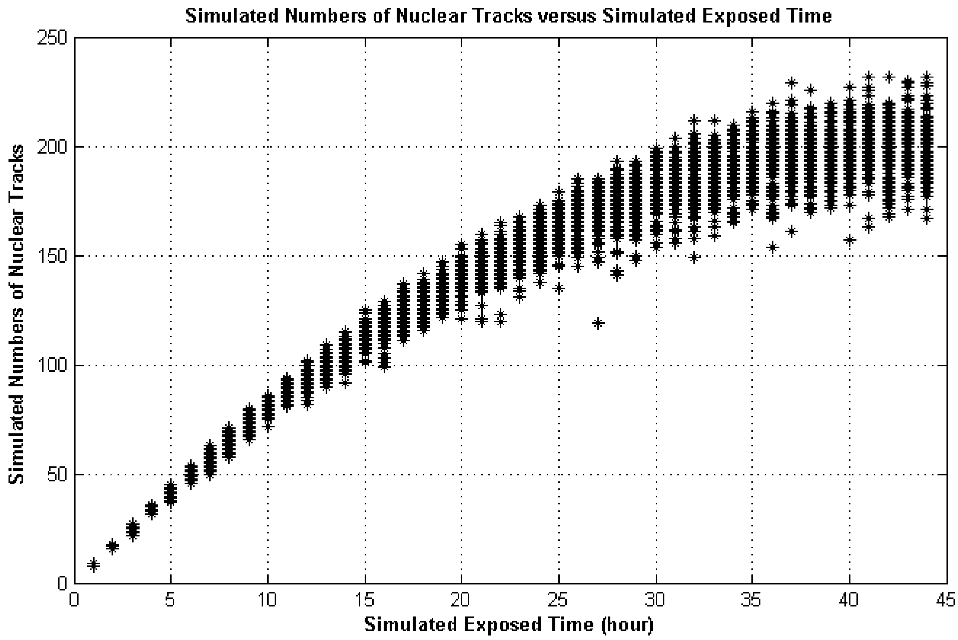


Figure 5:20 Simulated data graph after 100 times iterations

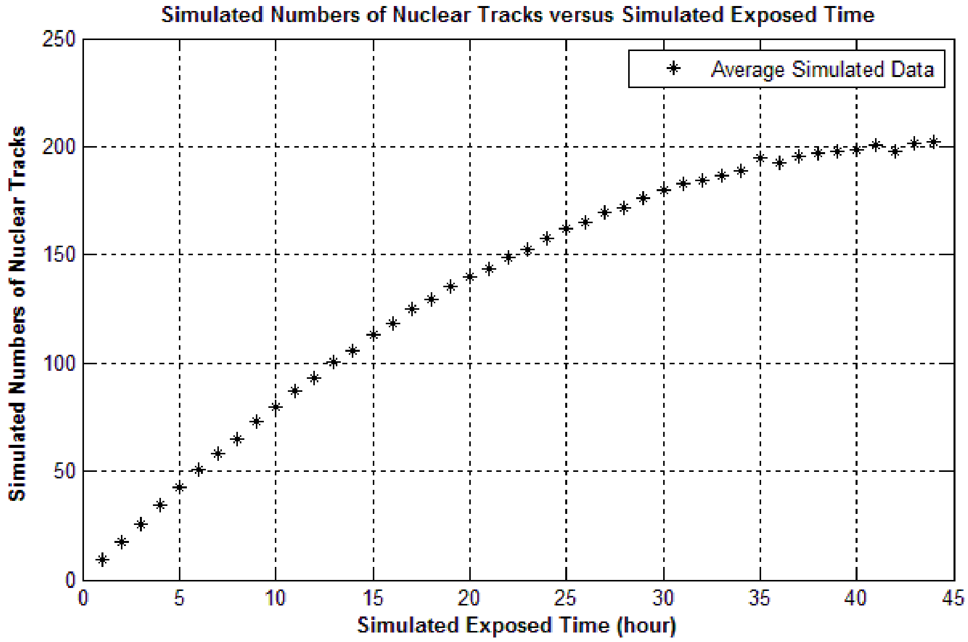


Figure 5:21 Average simulated data graph

5.5.2 Simulated Data, Linear Extrapolation and Nuclear Track Counting

Linear extrapolation of the graph was derived from the first data in the manual method of nuclear track counting, which 9.1 nuclear tracks are rounded to 9 nuclear tracks over 1 hour. Then, linear extrapolation is plotted with rate of 9 nuclear tracks per 1 hour. Figure 5:22 shows the graph simulation of nuclear track counting data and linear extrapolation, which the simulated data were different from the linear extrapolation. In other words, the difference ranging between these data was getting wider due to time. This means that the number of radiation particles penetrating the same position increases exponentially due to time. Unfortunately, because these radiation particles left no tracks behind, as a result it cannot be counted by our nuclear tracks counting scripts, which could affect the nuclear tracks counting results.

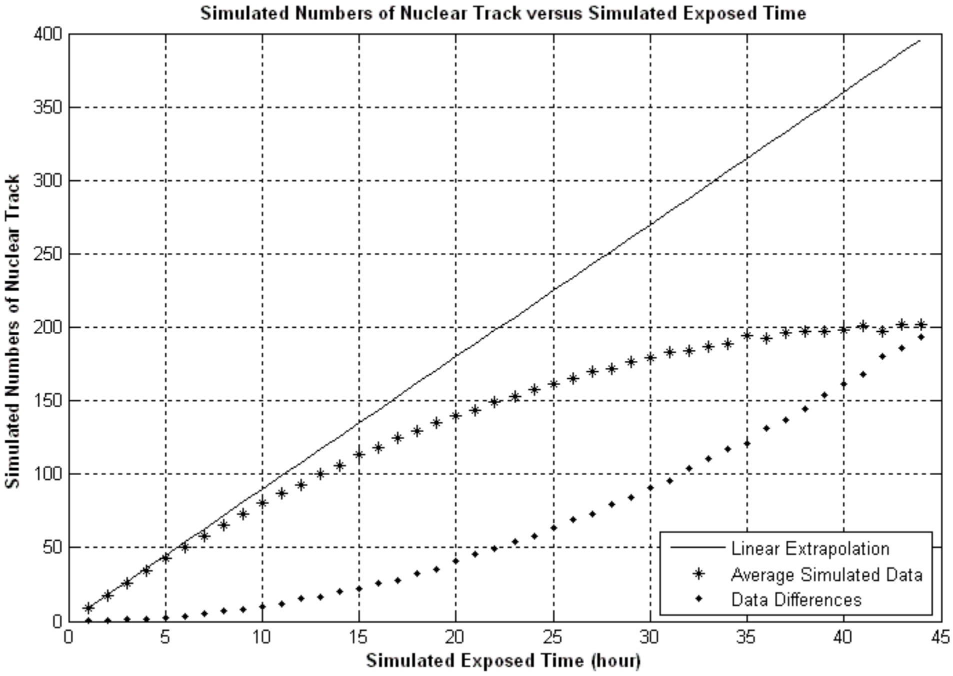


Figure 5:22 Simulated data and linear extrapolation graph with the data differences

Figure 5:23 shows the nuclear tracks counting results from the Manual method, IPTM processing and SDC processing which gave similar data distribution profile to the simulated data, which the slope of the graph represents the rate of nuclear tracks counting decaying due to time. However, there were still significant differences between the distributions of results. There was a range gap between the simulated data and the nuclear tracks counting data distributions. The range gap was recognized as the number of uncounted nuclear tracks by nuclear tracks counting. As a result, the rate of the nuclear tracks from the Manual method, IPTM and SDC processing were already compensated by this uncounted number of nuclear tracks.

From this finding and based on Section 5.4, we can say that these uncounted numbers of nuclear tracks are actually the number of nuclear tracks that over-laps onto another nuclear track. For instance, if two nuclear tracks were over-lapping to each other more than limit of segmentation, the nuclear tracks counting will be counting it as 1 nuclear track and the other one will be considered as uncounted nuclear track.

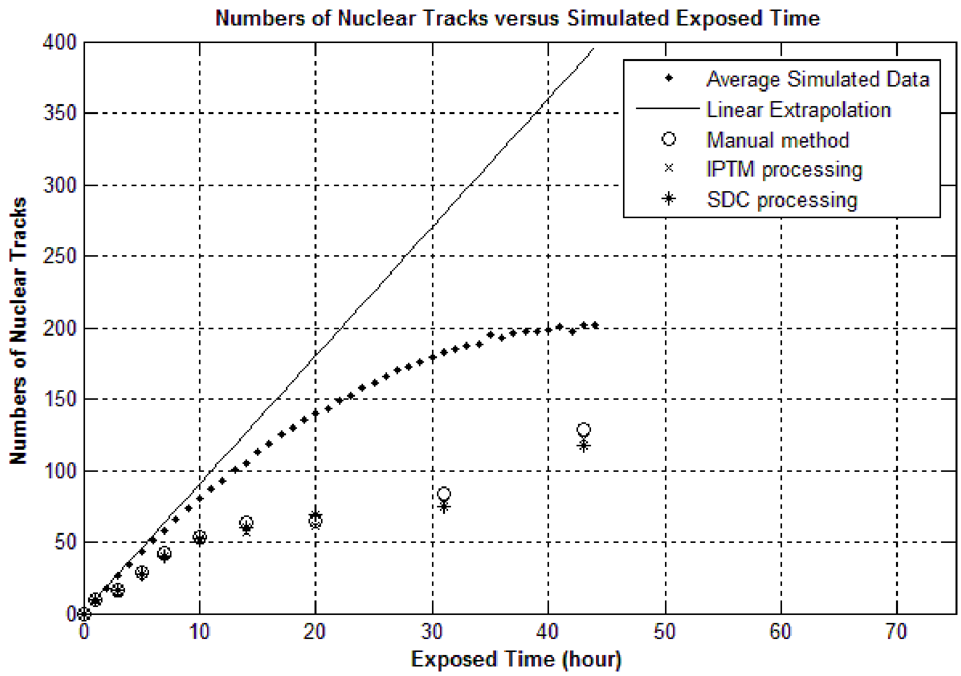


Figure 5:23 Simulated data, Linear extrapolation and Nuclear track counting graph

5.6 Granulometry, the Area Measurement

The analysis starts with an assumption that the area parameter of the nuclear tracks in the images is unvarying and approximate to the average area value. Thus, the final results would be the average of mean area of the nuclear tracks, coupled with its data uncertainty for each toolbox.

Figure 5:24 shows the graph of the data of mean area of nuclear tracks against the exposed time to the radiation source. The results show that the data distribution pattern of the nuclear tracks mean area from both processing methods whether its IPTM or SDC has the same pattern of data distribution.

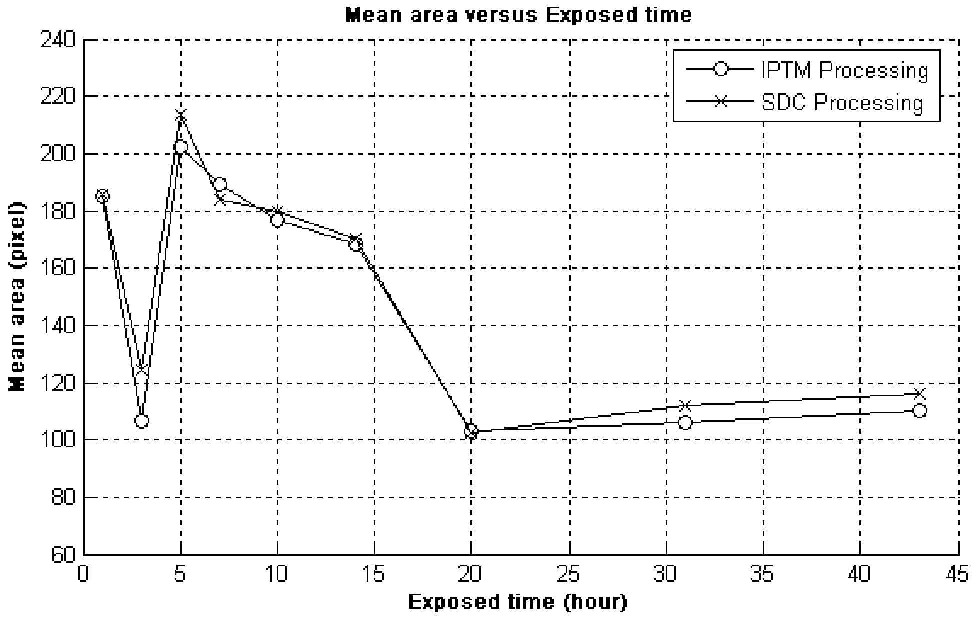


Figure 5:24 Mean area of nuclear tracks

Additionally, for more information refer to Figure 5:25. From the results, we can safely say that areas are in the standard deviation range and there is only one point where it is situated outside of the standard deviation range that is the point at 3 hours exposed time. This means, that both of the processing methods can generate the results that is in the range of their data standard deviations. In other words we can establish that both toolboxes have the capability to produce results that is slightly in the same range of precision in measuring the area of nuclear tracks. Their average standard deviations also show that both toolboxes generate data deviations almost equal in determining the mean area of nuclear tracks, see Table 5:3.

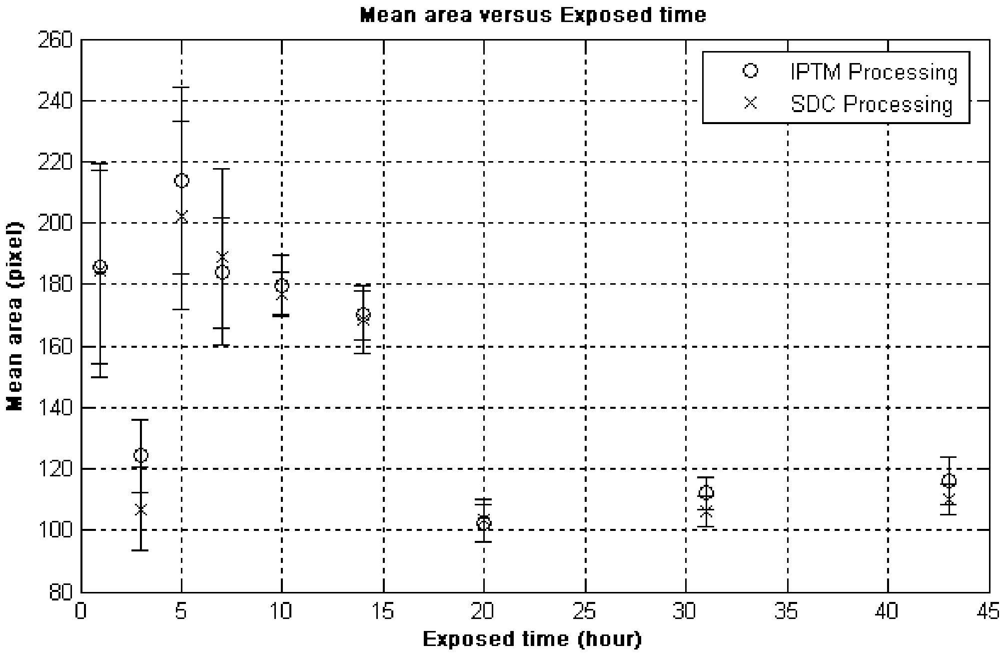


Figure 5:25 Mean area of nuclear tracks with standard deviation error bars

Table 5:3 The results of mean area of the nuclear tracks

Method	Average of mean area ($pixel^2$)	Average of standard deviations ($pixel^2$)
IPTM	149.80	42.04
SDC	154.25	40.51

5.6.1 Nuclear Track Area Analysis

For a more detail analysis, starting with an assumption to the nuclear tracks shapes that were circulars. As a result, the roundness of the nuclear tracks area is related to the expression 5:1, where *A* is the area of the nuclear tracks and *d* is the diameter of the nuclear tracks.

$$A = \pi \left(\frac{d}{2}\right)^2 \qquad 5:1$$

$$d = \sqrt{4A/_{\pi}} \qquad 5:2$$

The value of *d* can be obtained by using the information from the Table 5:3 and applied to the equation 5:2. Unfortunately, the obtained *d* was still in pixel unit, which was required to convert to SI unit. Therefore, in order to convert *d* in SI unit, the value of *d* was divided by image resolution, 180 pixels per inches (PPI or DPI) into inches unit. Then it was converted to micrometre unit. In order to get the actual diameter, the image magnification was reduced by 260 magnifications (Section 4.3, page 51) (see Appendix 4B), as a result, see Table 5:4.

Table 5:4 The results of mean area and diameter of the nuclear tracks

Method	Average of mean area ($pixel^2$)	Diameter, *d*		
		(pixel)	(inches)	(µm)
IPTM	149.80	42.04	0.0384	7.50
SDC	154.25	40.51	0.0389	7.61

The results that gave the diameter of the nuclear track from the IPTM processing and SDC processing were 7.50 µm and 7.61 µm respectively. These nuclear track diameters results were accepted as satisfied results, which they were in the ranges of 4.0 µm and 8.0 µm diameter of nuclear track (Eghan, Buah-Bassuah, & Oppon, 2007).

5.7 Chapter Conclusions

Based on the tasks that have been done, we could safely say that we succeeded in achieving the objectives despite the inferior quality of the original images. MIP is used as the basic method for this study in conjunction with MATLAB system. SDC Morphology Toolbox and Image Processing Toolbox are capable in characterizing one or more of the region (object size) in the image. The usages of the toolboxes are computationally easy and fast. The toolboxes produce slightly different results in determining the nuclear tracks counting and also its accuracy and the precision range. This is due to imaging errors and noise but also to the fact that the ratio object size to size of structuring element must be large enough, sufficient to make the morphological filtering more effective (Rautio & Silvén, 1998).

There is the limitation in using the Segmentation process in processing the over-lapping objects in the image. If the segmentation succeeds, then the objects are recognised as two different objects but if it fails, the objects are recognised as one object. On the other hand, the limitation is dependent on the size, shape of the objects and the over-lapping area of the two objects. The segmentation also makes the segmented objects smaller than the original one, in other words the segmentation process erodes the objects.

The result proved that the MATLAB and SDC toolboxes and the manual counting method have failed to count the nuclear tracks in linearly distribution after 10 hours of exposed time. This condition verified that these nuclear track counting method detected nonlinear growth after 10 hours of exposed time. This means, the effectiveness of LR 115 film track detector decrease in detecting radiation emission from monazite in linear distribution in range up to 10 hours of exposed time.

The outcome for the nuclear tracks counting analysis based on the linear output results, please refer Table 5:2 (page 73), which give the percentage difference between the IPTM and SDC processing and the manual method results. Generally, the percentage differences showed that the SDC processing gave a better result than IPTM processing. This means, in 10 hours of exposed time to the radiation, SDC processing resulted the nuclear track rate was more precise than IPTM processing and it also same goes to the error number of nuclear track.

The conclusions for the whole result of the nuclear track counting; please refer on the graphs in Figure 5:23 (page 77). The experimental graphs were not linear as resulted by Linear Extrapolation graph and also were not even similar to the curve of Simulated Data graph. There were two reasons that affected the experimental graphs, firstly the nuclear tracks that were at the same position as previous nuclear tracks were not counted and the limit of segmentation to the over-lapping nuclear tracks image, which two over-lapping nuclear tracks will be counted as one nuclear track. In other words, these two reasons reduced the number of counting from the Manual method, IPTM processing and SDC processing counting.

Granulometry of the mean area nuclear tracks can be over and done with that both toolboxes have the capability to produce results that is slightly in the same range of precision in measuring the area of nuclear tracks. Their average standard deviations proved that both toolboxes generated almost equal deviations data in determining the mean area of nuclear tracks, please refer Table 5:3. This is also supported by their diameter area, which the toolboxes give the results with 0.09 pixels different, given in Table 5:4.

CHAPTER 6

CONCLUSIONS

6.1 Research Conclusions

A method of producing and developing the nuclear tracks images from the LR115 radiation detector film can be completed successfully in laboratory. The preparation of the images is easy but full of concentration required for each step in order to obtain fine nuclear track slides. This also goes to the image capturing process. During the process, the light that is used to focus the nuclear tracks slide must be always in fixed intensity (level number 7 at the microscope scale). This light is able to affect and distort the structure of the objects in the image if its brightness too high and low, in other words the image becomes saturated.

The over-lapping objects in an image can be segmented to two objects but there is a limitation in separating the objects. This limitation depends on the shape of object and the over-lapping area of the two objects. For example, two identical circle objects (75 pixels diameter) that are over-lapping to each other can be segmented if the over-lapping diameter (intersection, d_i) is less than 80 percent of the circle diameter, d, see Figure 6:1. Besides that, the segmentation process makes the over-lapping objects getting smaller; it erodes the shape of over-lapping objects.

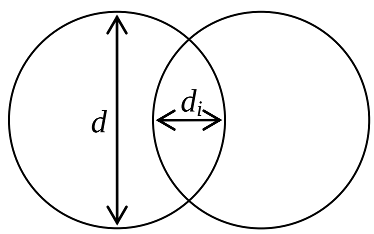


Figure 6:1 Intersection diameter, d_i of the over-lapping two circles

The conclusion for this studies especially the toolboxes usage part can be understood that each toolbox that has been applied has an advantage and also disadvantage in processing the image respectively. We concluded that SDC Morphology toolbox has high precision and accuracy in processing and determining the data from the image (object counting and granulometry) but it starts to gain an error if the exposed time is more than 30 hours. However, the Image Processing toolbox also has high precision and accuracy but its precision is less than SDC Morphology toolbox's. This also proved that SDC Morphology toolbox which was developed based on MIP is improved toolbox in image processing field because precision is superior to the Image Processing toolbox. Besides that, the SDC Morphology toolbox is easy to prepare and we can get it from SDC Information System website as free trial software in 30 days period. Figure 6:2, Figure 6:3 and Figure 6:4 below show the research results based on toolboxes usage.

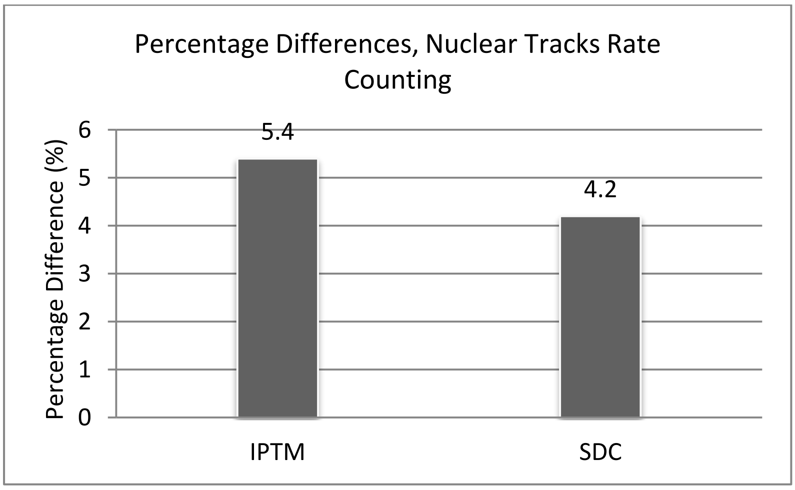


Figure 6:2 Percentage difference of the nuclear tracks rate counting

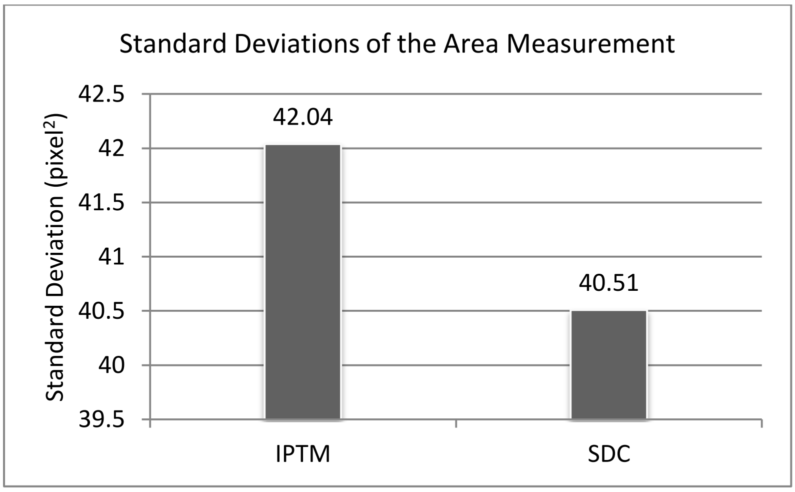


Figure 6:3 Standard Deviations of the area measurement

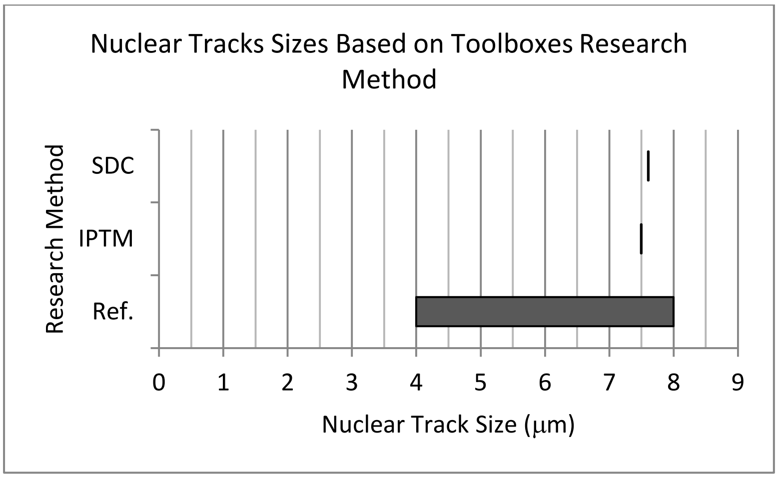


Figure 6:4 The results of the nuclear track sizes based on toolboxes compared to Eghan et al. (2007) as reference (Ref.) result

Image imperfections, which are caused by the distortion of light intensity, can be reconstructed in order to improve the resultant initial image. A few of the nuclear track regions in the image are over-lapping to each other but this situation can be partially corrected by subtracting the over-lapping regions using watershed lines and recognized as separated regions. Watershed operation creates the output image by detecting the domain of the catchments basins of input image, according to the connectivity defined by structuring element.

The counting of the nuclear track regions on the track detector is a non-linear filtering process and this research introduced the method to extract the regions of interest based on morphological opening, reconstruction and the subtraction operations.

6.2 Future Recommendations

For the future recommendations research, this research can be extended to more comparison with variety of the image processing toolboxes, such as DIPimage toolbox, image processing tool with Mathematica platform, image processing tool with LabVIEW platform and many more. By using the same sample images, we propose that the results from all image processing tools will be analysed and studied how diverse their results to each other and to the true result or value. Limitation of the segmentation over-lapping studies also can be extended to the variety and complex shape objects, for instance the shape of triangle, polygon and rod shape.

This research deals with binary and greyscale images. It is logical to extend these techniques to the colour images. However, this will need some work in improving the Mathematical Morphology algorithms before it can be applied to the colour images.

However, based on the research, the Mathematical Morphology concept not just only works with digital image. Therefore, for the last future recommendation research, the Mathematical Morphology algorithms also could be more improved in order to make it works with motion picture or movie.

Bibliography

Ballard, D., & Brown, C. (1982). *Computer Vision.* Prentice-Hall.

Beddow, J. K. (1997). Texture Analysis. In *Image Analysis Sourcebook* (pp. 240-246). Santa Barbara, CA, USA: Publisher, American Universities Science and Technology Press.

Dougherty, E. R. (1992). *An Introduction to Morphological Image Processing* . SPIE-International Society for Optical Engine .

Eghan, M. J., Buah-Bassuah, P. K., & Oppon, O. C. (2007). Track analysis of laser-illuminated etched track detectors using an opto-digital imaging system. *Measurement Science and Technology , 18*.

Fisher, R., & Perkins, S. (2003). *HIPR2*. Retrieved 2010, from Hypermedia Image Processing Reference: http://homepages.inf.ed.ac.uk/rbf/HIPR2/

Fleischer, R. L., Price, P. B., & Walker, R. M. (1975). *Nuclear tracks in solids: principles and applications.* University of California Press.

Forsyth, D. A., & Ponce, J. (2003). *Computer Vision A Modern Approach.* Upper Saddle River, New Jersey: Prentice Hall.

Gatos, B., & Perantonis, S. J. (2004). FAST IMPLEMENTATION OF MORPHOLOGICAL OPERATIONS USING BINARY IMAGE BLOCK DECOMPOSITION. *International Journal of Image and Graphics , 4*, 183-202.

Gonzalez, R. C., & Woods, R. E. (2002). *Digital Image Processing* . Prentice Hall, Inc.

Haralick, R. M., & Shapiro, L. G. (1991). *Computer and Robot Vision* (Vol. I). Addison-Wesley.

Haralick, R. M., Sternberg, S. R., & Zhuang, X. (1987). Image analysis using mathematical morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence , 9* (4), 532-550.

Heijmans, H. (1999, April). *ERCIM News*. Retrieved 2010, from Mathematical Morphology and Image Processing: http://www.ercim.eu/publication/Ercim_News/enw37/heijmans.html

Heijmans, H. (1994). Mathematical morphology as a tool for shape description. *Shape in Picture: Mathematical Description of Shape in Grey-level Images* , 147-176.

Horn, B. K. (1986). *Robot vision.* MIT Press.

Jain, R., Kasturi, R., & Schunck, B. G. (1995). *Machine Vision.* McGraw-Hill Science/Engineering/Math.

Khan, H. A., & Khan, N. A. (1989). Solid State Nuclear Track Detection (Ssntd): A Useful Scientific Tool For Basic And Applied Research. *Journal of Islamic Academy of Sciences 2:4* , 303-312.

Kukielka, G., & Woznicki, J. (2001). Hierarchical Method of Digital Image Segmentation Using Multidimensional Mathematical Morphology. *Computer Analysis of Images and Patterns, 9th International Conference, CAIP 2001 Warsaw, Poland, Proceedings , 2124*, 581-588.

LAPACK. (2010). (U. o. Tennessee, B. Univ. of California, & U. o. Denver, Producers) Retrieved 2010, from LAPACK — Linear Algebra PACKage: http://www.netlib.org/lapack/

Lyshevski, S. E. (2003). *Engineering and Scientific Computations Using MATLAB.* Wiley-Interscience .

Maragos, P. (2005). Lattice Image Processing: A Unification of Morphological and Fuzzy Algebraic Systems. In *Journal of Mathematical Imaging and Vision* (pp. 333-353). Springer Science + Business Media, Inc. Netherlands.

Marshall, J. T. (1992). A Matlab/TMS320/TMS340 image processing environment. *Circuits and Systems, 1992. ISCAS '92. Proceedings., 1992 IEEE International Symposium on , 5*, 2505-2508.

Matheron, G. (1988). Filters and lattices. (J. Serra, Ed.) *Image Analysis and Mathematical Morphology , II*, 13-35.

Matheron, G. (1975). *Random sets and integral geometry.* John Wiley & Sons.

MathWorks. (2010). *Image Processing Toolbox Page*. Retrieved 2010, from The MathWorks, Inc: http://www.mathworks.com/index.html

MathWorks. (2011). *Image Processing Toolbox, Examples and Webinars, Detecting a Cell Using Image Segmentation.* Retrieved 2011, from The MathWorks, Inc Web Site: http://www.mathworks.com/products/image/demos.html?file=/products/demos/shipping/images/ipexcell.html

Miano, J. (1999). *Compressed Image File Formats: JPEG, PNG, GIF, XBM, BMP.* Addison-Wesley Professional .

Michielsen, K., & Raedt, H. d. (2001). Integral-Geometry Morphological Image Analysis. *Physics Reports 347* , 461-538.

Niblack, W. (1986). *An Introduction to Digital Image Processing* (First English Edition ed.). Prentice Hall.

Nikezic, D., & Yu, K. N. (2004). Formation and growth of tracks in nuclear track materials. *Materials Science and Engineering R 46. Reports: A Review Journal* , 51-123.

Pratt, W. K. (2001). *Digital Image Processing: PIKS Inside* (3rd Edition ed.). John Wiley & Sons, Inc.

Pritchard, D. (2002). Vocal Tract Visualisation. *EECE593 Winter 2002 Project Report* .

Quadrades, S., & Sacristán, A. (2001). Automated Extraction Of Microcalcifications Bi-Rads Numbers In Mammograms. *Image Processing, 2001. Proceedings. 2001 International Conference on , 2,* 289-292.

Rautio, H., & Silvén, O. (1998, November). Average Grain Size Determination using Mathematical Morphology and Texture Analysis. *Procceding IAPR Workshop on Machine Vision Applications (MVA '98)* , 506-509.

Roerdink, J. B. (1993). Mathematical morphology with non-commutative symmetry groups. In *In Mathematical Morphology in Image Processing* (pp. 205-254). Marcel Dekker, New York.

Rosenfeld, A. (1969). *Picture Processing by Computer*. New York: Academic Press.

Ruberto, C. D., & Dempster, A. G. (2000). Segmentation of Blood Images Using Morphological Operators. *International Conference on Pattern Recognition Proceeding , 3*, 397-400.

Russ, J. C. (2008). *The Image Processing Handbook,* (5th Edition ed.). CRC Press, Taylor and Francis Group.

Santaló, L. A. (2004). *Integral Geometry and Geometric Probability*. Cambridge University Press.

Scott, T. A., & Mukherjee, D. P. (2000). Scale space classification using area morphology. *IEEE Transactions on Image Processing , 9* (4), 623-635.

SDC, I. S. (2011). *SDC Morphology Toolbox for MATLAB.* Retrieved 2011, from SDC Information Systems: http://www.mmorph.com/

Serra, J. (1982). *Image Analysis and Mathematical Morphology* (Vol. 1). Academic Press.

Serra, J. (1983). *Image Analysis and Mathematical Morphology*. Orlando, FL, USA: Academic Press, Inc.

Serra, J. (1988). *Image Analysis and Mathematical Morphology* (Vol. 2: Theoretical Advances by Jean Serra). Academic Press.

Serra, J., & Soille, P. (1994). Mathematical Morphology and Its Applications to Image Processing. *proceedings of the 2nd international symposium on mathematical morphology (ISMM'94)* , 369-374.

Shih, F. Y., & Mitchell, O. R. (1992). A mathematical morphology approach to Euclidean distance transformation. *IEEE Transactions on Image Processing* , *I* (2), 197 - 204 .

Stanoyevitch, A. (2004). *Introduction to MATLAB with Numerical Preliminaries.* John Wiley & Sons, Ltd.

Stoyan, D., Kendall, W. S., & Mecke, J. (1996). *Stochastic Geometry and Its Applications* (2nd Edition ed.). John Wiley & Sons Ltd.

Sun, L., & Hao, Y. (2005). Tuning the properties of magnetic nanowires. *IBM Journal of Research and Development* , 79-102.

User's Guide. (2001, June). *Image Processing Toolbox™ User's Guide , Version 3.1*. The MathWorks, Inc.

Using MATLAB. (1999). *MATLAB® The Language of Technical Computing* . The MathWorks, Inc.

Ves, E. d., Benavent, X., Ayala, G., & Domingo, J. (2006, May). Selecting the structuring element for morphological texture classification. *Pattern Analysis and Applications* , 48-57.

Vincent, L. (1994a, September). Fast Grayscale Granulometry Algorithms. *Proc. EURASIP Workshop ISMM'94, Mathematical Morphology and its Applications to Image Processing* , 265-272.

Vincent, L. (1994b). Fast opening functions and morphological granulometries. (E. R. Dougherty, P. D. Gader, & M. Schmitt, Eds.) *Image Algebra and Morphological Image Processing V. Proc. SPIE , 2300*, 253-267.

Vincent, L. (1993). Morphological Grayscale Reconstruction in Image Analysis: Applications and Efficient Algorithms. *IEEE Transactions on Image Processing , 2*, 176-201.

Vincent, L., & Dougherty, E. R. (1994). Morphological Segmentation for Textures and Particles. In E. R. Dougherty (Ed.), *Digital Image Processing Methods* (pp. 43-102). New York: Marcel-Dekker.

APPENDIX 2A:

Infimum and Supremum

1. Infimum (inf)

The infimum (inf) is the greatest lower bound of a set *S*, defined as a quantity *m* such that no member of the set is less than *m*, but if $\in$ is any positive quantity, however small, there is always one member that is less than $m + \in$. When it exists (which is not required by this definition, e.g., $\inf \mathbf{R}$ does not exist), the infimum is denoted $\inf S$ or $\inf_{x \in s} x$.

More formally, the infimum $\inf S$ for *S* a (nonempty) subset of the extended reals $\overline{\mathbf{R}} = \mathbf{R} \cup \{\pm\infty\}$ is the largest value $y \in \overline{\mathbf{R}}$ such that for all $x \in S$ we have $x \geq y$. Using this definition, $\inf S$ always exists and, in particular, $\inf \mathbf{R} = -\infty$. Whenever an infimum exists, its value is unique.

2. Supremum (sup)

The supremum (sup) is the least upper bound of a set *S*, defined as a quantity *M* such that no member of the set exceeds *M*, but if $\in$ is any positive quantity, however small; there is a member that exceeds $M - \in$. When it exists (which is not required by this definition, e.g., $\sup \mathbf{R}$ does not exist), it is denoted $\sup_s$ or $\sup_{x \in s}$.

More formally, the supremum $\sup S$ for *S* a (nonempty) subset of the extended reals $\overline{\mathbf{R}} = \mathbf{R} \cup \{\pm\infty\}$ is the smallest value $y \in \overline{\mathbf{R}}$ such that for all $x \in S$ we have $x \leq y$. Using this definition, $\sup S$ always exists and in particular of $\sup \mathbf{R} = \infty$.

Whenever a supremum exists, its value is unique. On the real line, the supremum of a set is the same as the supremum of its closure.

APPENDIX 2B:

Examples and Full Description of Interval Functions.

1. mmendpoints - Interval to detect end-points.

Synopsis : `[Iab] = mmendpoints (OPTION)`

Input : `OPTION`: String. 'LOOP' or 'HOMOTOPIC' Default: "LOOP".

Output : `Iab`: Interval.

Description : *mmendpoints* creates an interval that is useful to detect end-points of curves (i.e., one pixel thick connected components) in binary images. It can be used to prune skeletons and to mark objects transforming them in a single pixel or closed loops if they have holes. There are two options available: LOOP, deletes all points but preserves loops if used in *mmthin*; HOMOTOPIC, deletes all points but preserves the last single point or loops.

Examples:

Interval visualization;

```
>> mmintershow(mmendpoints)
ans =
.  .  .
0  1  0
0  0  0
```

Function to visualize an interval.

mmendpoints interval

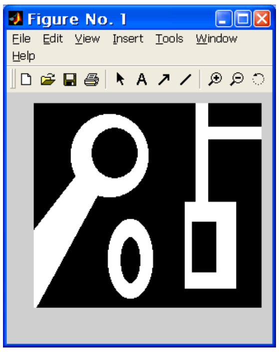
```
>> mmintershow(mmendpoints('HOMOTOPIC'))
ans =
.  1  .
0  1  0
0  0  0
```

Pruning;

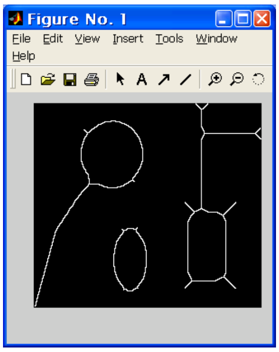
```
>> i = imread('image05.tif');
>> mmshow(i);
>> i1 = mmthin(i);
>> mmshow(i1);
>> i2 = mmthin(i1, mmendpoints, 10);
>> mmshow(i2);
```

Function for image transformation by thinning.

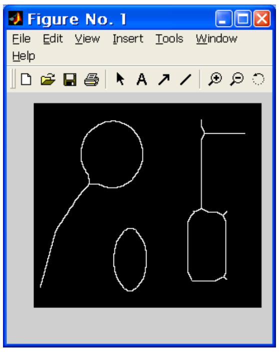
Pruning 10 pixels



(i)



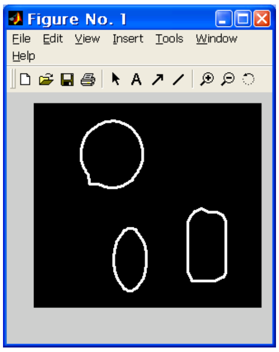
(i1)



(i2)

Homotopic marking;

```
>> in = mmthin(i1,mmendpoints('HOMOTOPIC'));
>> mmshow(mmdil(in));
```



(mmdil (in))

<u>Equation;</u>

$$I_{A,B} : A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

2. **mmhomothick** - Interval for homotopic thickening.

Synopsis : `[Iab] = mmhomothick`

Output : `Iab`: Interval.

Description : *mmhomothick* creates an interval that is useful for the homotopic (i.e., that conserves the relation between objects and holes) thickening of binary images.

Examples:

Interval visualization;

```
>> mmintershow(mmhomothick)
ans =
1  1  1
.  0  .
0  0  0
```

Equation;

$$I_{A,B} : A = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

3. **mmhomothin** - Interval for homotopic thining.

Synopsis : `[Iab] = mmhomothin`

Output : `Iab`: Interval.

Description : *mmhomothin* creates an interval that is useful for the homotopic (i.e., that conserves the relation between objects and holes) thinning of binary images.

Examples:

Interval visualization;

```
>> mmintershow(mmhomothin)
ans =
0  0  0
.  1  .
1  1  1
```

Equation;

$$I_{A,B} : A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

4. mminterot - Rotate the Interval.

Synopsis : `[Irot] = mminterot( Iab, theta, DIRECTION )`

Input : `Iab`: Interval.

`theta`: Degrees of rotation. Available values are multiple of 45 degrees. **Default:** 45.

`DIRECTION`: String. 'CLOCKWISE' or 'ANTI-CLOCKWISE'. **Default:** 'CLOCKWISE'.

Output : `Irot`: Interval.

Description : *mminterot* rotates the interval `Iab` by an angle `theta`.

Limitations : The rotation angles allowed are multiples of 45 degrees.

Examples:

```
>> in1 = mmendpoints;
>> in2 = mminterot(in1);
>> mmintershow(in1)
ans =
.  .  .
0  1  0
0  0  0
>> mmintershow(in2)
ans =
0  .  .
0  1  .
0  0  0
```

Equation;

$$I_{A_\theta, B_\theta}$$

5. **mmintershow** -Visualize an interval.

Synopsis : `[s] = mmintershow( Iab )`

Input : *Iab* : Interval.

Output : *s* : String. (representation of the interval).

Description : *mmintershow* creates a representation for an interval using 0, 1 and . (don't care).

Examples:

```
>> mmintershow(mmhomothin)
ans =
0  0  0
0  1  .
0  0  0
```

6. **mmse2hmt** - Create a Hit-or-Miss Template (or interval) from a pair of structuring elements.

Synopsis : `[Iab] = mmse2hmt( A, Bc )`

Input : *A* : Structuring element. Left extremity.

Bc: Structuring element. Complement of the right extremity.

Output : *Iab*: Interval.

Description : *mmse2hmt* creates the Hit-or-Miss Template (HMT), also called interval [*A*, *Bc*] from the structuring elements *A* and *Bc* such that *A* is included in the complement of *Bc*. The only difference between this function and *mmse2interval* is that here the second structuring element is the complement of the one used in the other function. The advantage of this function over *mmse2interval* is that this one is more flexible in the use of the structuring elements as they are not required to have the same size.

Examples:

```
>> B1img = logical(uint8([0 1 0; 1 1 0; 0 0 1]))
B1img =
    0    1    0
    1    1    0
    0    0    1
>> B2img = logical(uint8([1 0 0; 0 0 0; 0 0 0]))
B2img =
    1    0    0
    0    0    0
    0    0    0
>> B1 = mmimg2se(B1img);
>> B2 = mmimg2se(B2img);
>> i = mmse2hmt(B1,B2);
>> mmintershow(i)
ans =
 0   1   .
 1   1   .
 .   .   1
```

Equation;

Flat Interval;

$$I_{A,Bc} = \left\{ X : A \subset X \subset Bc^{c} \right\}$$

7. **mmse2interval - Create an interval from a pair of structuring elements.**

Synopsis : `Iab  = mmse2interval(a, b)`

Input : `a` : Structuring element. Left extremity.

`b` : Structuring element. Right extremity.

Output : `Iab`: Interval.

Description : `mmse2interval` creates the interval `[a, b]` from the structuring elements `a` and `b` such that `a` is less or equal `b`.

Examples:

```
>> i = mmse2interval(mmsecross, mmsebox);
>> mmintershow(i)
ans =
 .  1  .
 1  1  1
 .  1  .
```

Equation;

Flat interval,

$$I_{A,B} = \{ X \subset W : A \subset X \subset B \}$$

Where W is a finite rectangle.

Non-flat interval,

$$I_{a,b} = \{ f \in W^{\kappa} : a \leq f \leq b \}$$

APPENDIX 2C:

The mmconv - Conventions used in the SDC Morphology Toolbox.

Synopsis : `mmconv`

Description : All operators of the SDC Morphology Toolbox start with '*mm*'. This is a simple rule to check if an operator belongs or not to the SDC Morphology Toolbox. All operators of the SDC Morphology Toolbox obey the following rules of parameter uses:

- Return a single data structure.
- The parameters are position and type dependent. For example, if the third parameter of an operator is a structuring element, then the third parameter can only be a structuring element for that operator. Note that this rule is somewhat different from most MATLAB operators.
- The definition of optional parameters depends on the parameters order in the parameter list. One parameter that has a parameter on its right that is not optional can not be optional either. For example, the `mmwatershed` operator has 3 parameters: `f`, `Bc` and `LINEREG`, with the last two being optional. Therefore, the only ways that `mmwatershed` can be invoked are: `mmwatershed(f)`, `mmwatershed(f, Bc)` or `mmwatershed(f, Bc, LINEREG)`.
- When a default parameter is a structuring element, the elementary cross (*mmsecross*) is used as the default value.

All operators of the SDC Morphology Toolbox, with two image parameters, such as *mmaddm*, obey the following rules of parameter uses:

- The two images must have the same size.

One of the images (normally the rightmost) can be represented by a constant. In this case, the constant is treated as a constant image, with the same size and type of the other image.

APPENDIX 2D:

The mmunion - Union of images.

Synopsis : `y = mmunion( f1, f2, f3, f4, f5)`

Input :

f1 : Gray-scale (*uint8* or *uint16*) or binary image (logical *uint8*).

f2 : Gray-scale (*uint8* or *uint16*) or binary image (logical *uint8*). Or constant.

f3 : Gray-scale (*uint8* or *uint16*) or binary image (logical *uint8*). Or constant. **Default:** No parameter.

f4 : Gray-scale (*uint8* or *uint16*) or binary image (logical *uint8*). Or constant. **Default:** No parameter.

f5 : Gray-scale (*uint8* or *uint16*) or binary image (logical *uint8*). Or constant. **Default:** No parameter.

Obstacle: *f1, f2, f3, f4,* and *f5* must have the same data type.

Output : *y*: Gray-scale (*uint8* or *uint16*) or binary image (logical *uint8*).

Description : *mmunion* creates the image *y* by taking the pixel-wise maximum between the images *f1, f2, f3, f4*, and *f5*. When *f1, f2, f3, f4*, and *f5* are binary images, *y* represents the union of them.

Example:

Numerical script example;

```
f = uint8([255   255    0   10    0   255   250]);
g = uint8([0    40   80   140   250    10    30]);
y1 = mmunion(f, g)
 y1 =
   255   255    80   140   250   255   250
y2 = mmunion(f, uint8(255))
 y2 =
   255   255   255   255   255   255   255
```

Equations:

Union: $(f_1 \vee f_2)(x) = \max\{f_1(x), f_2(x)\}$

Generalized Union: $\vee\{f_i : i \in I\} = (((f_1 \vee f_2) \vee ...) \vee f_n)$

APPENDIX 3A:

Source Code of MATLAB

1. Sobel and Canny edge detection MATLAB script:

```
i = imread('saturn.jpg');% read the image
i = rgb2gray(i);% convert the image to greyscale image
ed1 = edge(i,'sobel');% Sobel edge detection processing
ed2 = edge(i,'canny');% Canny edge detection processing
imshow(ed1);% display Sobel edge detected image
figure
imshow(ed2);% display Canny edge detected image
figure, imshow(i);% display original image
```

APPENDIX 3B:

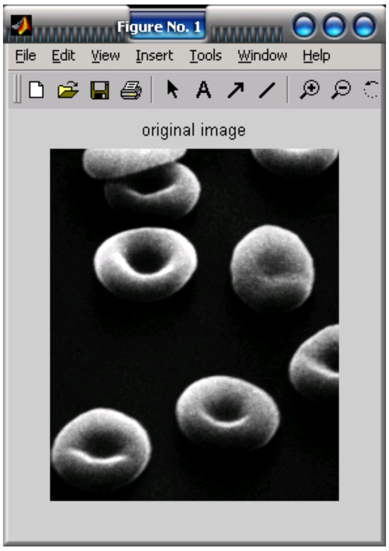
Detecting a Cell Using Image Segmentation

An object can be easily detected in an image if the object has sufficient contrast from the background. We use edge detection and basic morphology tools to detect the blood cells.

Step 1: Read Image

Read in `'cell.tif'`, which is an image of the blood cell:

```
I = imread('cell.tif');

figure, imshow(I), title('original image');
```



Step 2: Rescale the Image

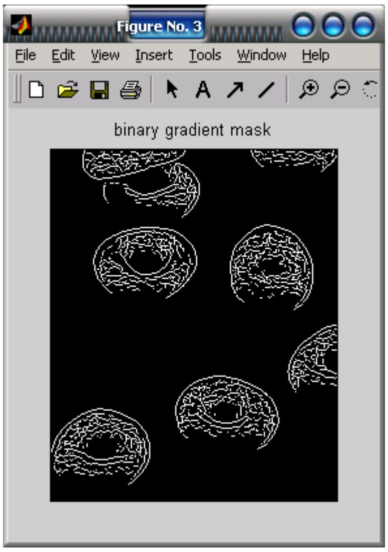
We use the `imadjust` function to rescale the image so that it covers the entire dynamic range ([0, 1]).

```
DI = imadjust(I, [], [0 1]);

figure, imshow(DI), title('scaled image');
```

Step 3: Detect Entire Cells

Seven cells are present in this image, but only one cell can be seen in its entirety. We will detect this cell. Another word for object detection is *segmentation*. The object to be segmented differs greatly in contrast from the background image. Changes in contrast can be detected by operators that calculate the gradient of an image. One way to calculate the gradient of an image is the Sobel operator, which creates a binary mask using a user-specified threshold value. We determine a threshold value using the `graythresh` function. To create the binary gradient mask, we use the `edge` function.

```
BWs = edge(DI, 'sobel', (graythresh(DI) * .1));

figure, imshow(BWs), title('binary gradient mask');
```



Step 4: Fill Gaps

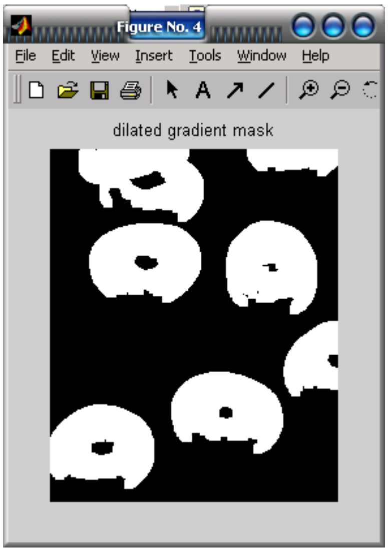
The binary gradient mask shows lines of high contrast in the image. These lines do not quite delineate the outline of the object of interest. Compared to the original image, you can see gaps in the lines surrounding the object in the gradient mask. These linear gaps will disappear if the Sobel image is dilated using linear structuring elements, which we can create with the `strel` function.

```
se90 = strel('line', 3, 90);

se0 = strel('line', 3, 0);
```

Step 5: Dilate the Image

The binary gradient mask is dilated using the vertical structuring element followed by the horizontal structuring element. The `imdilate` function dilates the image.

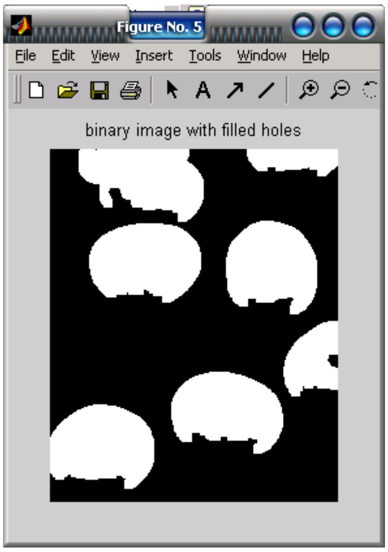
```
BWsdil = imdilate(BWs, [se90 se0]);

figure, imshow(BWsdil), title('dilated gradient mask');
```



Step 6: Fill Interior Gaps

The dilated gradient mask shows the outline of the cell quite nicely, but there are still holes in the interior of the cell. To fill these holes we use the `imfill` function.

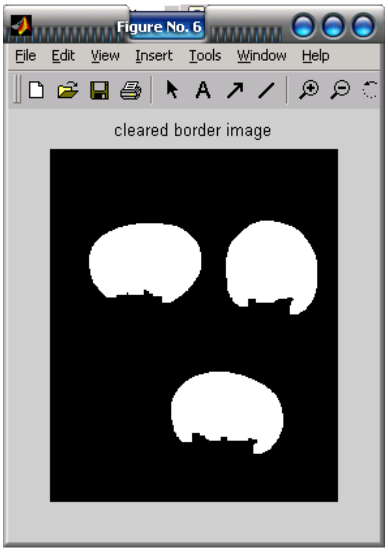
```
BWdfill = imfill(BWsdil,'holes');

figure, imshow(BWdfill);

title('binary image with filled holes');
```



Step 7: Remove Connected Objects on Border

The cell of interest has been successfully segmented, but it is not the only object that has been found. Any objects that are connected to the border of the image can be removed using the `imclearborder` function. The connectivity in the `imclearborder` function was set to 4 to remove diagonal connections.

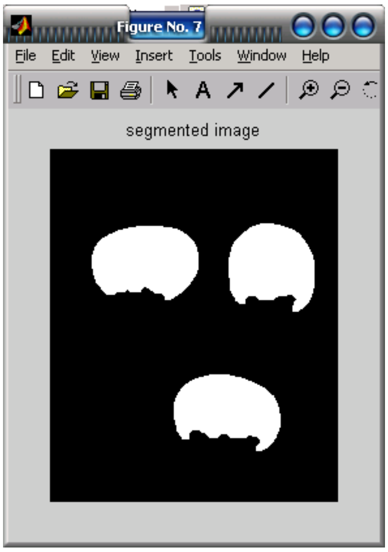
```
BWnobord = imclearborder(BWdfill, 4);

figure, imshow(BWnobord), title('cleared border image');
```



Step 8: Smooth the Object

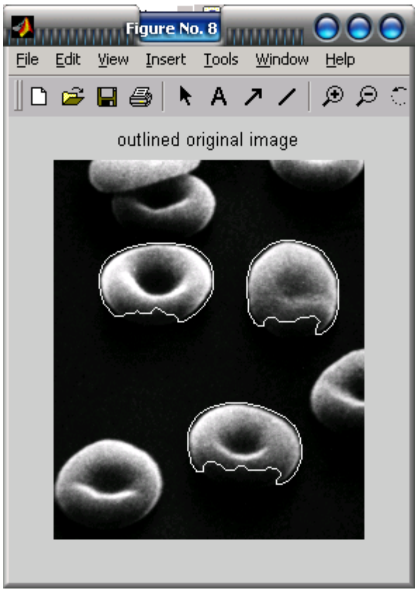
Finally, in order to make the segmented object look natural, we smooth the object by eroding the image twice with a diamond structuring element. We create the diamond structuring element using the `strel` function.

```
seD = strel('diamond',1);

BWfinal = imerode(BWnobord,seD);

BWfinal = imerode(BWfinal,seD);

figure, imshow(BWfinal), title('segmented image');
```



An alternate method for displaying the segmented object would be to place an outline around the segmented cell. The outline is created by the `bwperim` function.

```
BWoutline = bwperim(BWfinal);

Segout = imadd(I, immultiply(BWoutline, 255));

figure, imshow(Segout), title('outlined original image');
```



APPENDIX 3C:

SDC Morphology Toolbox Functions

The SDC Morphology Toolbox Functions can be classified by 17 categories. Each category has its own classification and suitability when it will be applied to solve the problems of image processing. According to documentation site in SDC Information Systems official website and manual reference, below are the lists of the SDC Morphology Toolbox Functions.

1. Data Type Conversion

mmbinary : Convert a gray scale image into a binary image.

mmfreedom : Control automatic data type conversion.

mmgray : Convert a binary image into a gray scale image.

2. Image Creation

mmdrawv : Superpose points, rectangles and lines on an image. (replaces *mmdraw*)

mmframe : Create a frame image.

mmtext : Create a binary image of a text.

3. Image file I/O

mmreadgray : Read an image from a commercial file format and stores it as a gray scale image.

mmwrite : Write a gray-scale image into a commercial file format.

4. Relations

mmcmp : Compare two images pixel wisely.

mmis : Verify if a relationship among images is true or false.

5. Operations

mmaddm : Addition of two images, with saturation.

mmintersec : Intersection of images.

mmneg : Negate an image.

mmsubm : Subtraction of two images, with saturation.

mmsymdif : Symmetric difference between two images.

mmtoggle : Image contrast enhancement or classification by the toggle operator.

mmunion : Union of images.

6. Structuring Elements

mmimg2se : Create a structuring element from a pair of images.

mmsebox : Create a box structuring element.

mmsecross : Cross structuring element.

mmsedil : Dilate one structuring element by another

mmsedisk : Create a disk or a semi sphere structuring element.

mmsedomain : Control implicit finite or infinite domain for image Minkowski operations.

mmseline : Create a line structuring element.

mmsereflect : Reflect a structuring element

mmserot : Rotate a structuring element.

mmseshow : Display a structuring element as an image.

mmsesum : N - 1 iterative Minkowski additions

mmsetrans : Translate a structuring element

mmseunion : Union of structuring elements

7. Dilations and Erosions

mmcdil : Dilate an image conditionally.

mmcero : Erode an image conditionally.

mmdil : Dilate an image by a structuring element.

mmero : Erode an image by a structuring element.

8. Morphological Filters

mmasf : Alternating Sequential Filtering.

mmcenter : Center filter.

mmclose : Morphological closing.

mmopen : Morphological opening.

9. Image Transforms

mmdist : Distance transform.

mmgdist : Geodesic Distance Transform.

mmopentransf : Open transform.

10. Connected Operators

mmareaclose : Area closing

mmareaopen : Area opening.

mmasfrec : Alternating Sequential Filtering by reconstruction

mmclohole : Close holes of binary and gray scale images.

mmcloserec : Closing by reconstruction.

mmflood : Flooding filter h, v, a-basin and dynamics (depth, area, volume).

mmhbasin : Remove basins with contrast smaller than h.

mmhdome : Remove peaks with contrast smaller than h.

mminfrec : Inf reconstruction.

mminpos : Minima imposition.

mmopenrec : Opening by reconstruction.

mmregmax : Regional Maximum.

mmregmin : Regional Minimum (with generalized dynamics).

mmsuprec : Sup reconstruction.

mmvbasin : Remove basins with volume smaller than v.

mmvdome : Remove domes with volume smaller than v.

11. Residues

mmcbisector : N Conditional bisector.

mmcloserecth : Close by Reconstruction Top Hat.

mmcloseth : Closing Top Hat.

mmedgeoff : Eliminate the objects that hit the image frame.

mmgradm : Morphological gradient.

mmlastero : Last erosion.

mmopenrecth : Open by Reconstruction Top Hat.

mmopenth : Opening Top Hat.

mmskelm : Morphological skeleton (Medial Axis Transform).

mmskelmrec : Morphological skeleton reconstruction (Inverse Medial Axis Transform).

12. Intervals (hit-or-miss templates)

mmendpoints	:	Interval to detect end points.
mmhomothick	:	Interval for homotopic thickening.
mmhomothin	:	Interval for homotopic thinning.
mminterot	:	Rotate an interval.
mmintershow	:	Visualize an interval.
mmse2hmt	:	Create a Hit or Miss Template (or interval) from a pair of structuring elements.
mmse2interval	:	Create an interval from a pair of structuring elements.

13. Sup-generating and Inf-generating

mminfcanon	:	Intersection of inf generating operators.
mminfgen	:	Inf generating.
mmsupcanon	:	Union of sup generating or hit miss operators.
mmsupgen	:	Sup generating (hit miss).
mmthreshad	:	Threshold (adaptive).

14. Thinning and Thickening

mmcthick	:	Image transformation by conditional thickening.
mmcthin	:	Image transformation by conditional thinning.
mmcwatershed	:	Detection of watershed from markers.
mmiwatershed	:	Interactive watershed from markers.
mmskiz	:	Skeleton of Influence Zone also know as Generalized Voronoi Diagram
mmswatershed	:	Detection of similarity based watershed from markers.
mmthick	:	Image transformation by thickening.
mmthin	:	Image transformation by thinning.
mmwatershed	:	Watershed detection.

15. Measurements

mmblob	:	Blob measurements from a labeled image.
mmfractal	:	Compute the fractal dimension of a binary image using Minkowski sausage model.
mmgrain	:	Scale statistics for each labeled region.
mmhistogram	:	Find the histogram of the image f.
mmlabel	:	Label a binary image.

mmlabelflat : Label the flat zones of gray scale images.

mmpatspec : Pattern spectrum (also known as granulometric size density).

mmstats : Find global image statistics.

16. Visualization

mmdtshow : Display a distance transform image with an iso-line colour table.

mmgdtshow : Display a distance transform image with an iso-line colour table.

mmglblshow : Apply a random colour table to a gray-scale image.

mmgshow : Apply binary overlays as colour layers on a binary or gray-scale image

mmlblshow : Display a labeled image assigning a random colour for each label.

mmshow : Display binary or gray scale images and optionally overlay it with binary images.

mmsurf : Generate a shaded visualization image of a gray scale image as a topographic model.

mmtruesize : Make the image display true size to screen pixels.

17. Obsolete

mmdraw : Superpose rectangles and lines on an image. Obsolete, use *mmdrawv*.

mmisbinary : Check for binary image. Obsolete, use *mmis*.

mmisequal : Verify if two images are equal. Obsolete, use *mmis*.

mmislesseq : Verify if one image is less or equal another. Obsolete, use *mmis*.

APPENDIX 3D:

The SDC Morphology Toolbox Demonstrations Scripts and Its Descriptions.

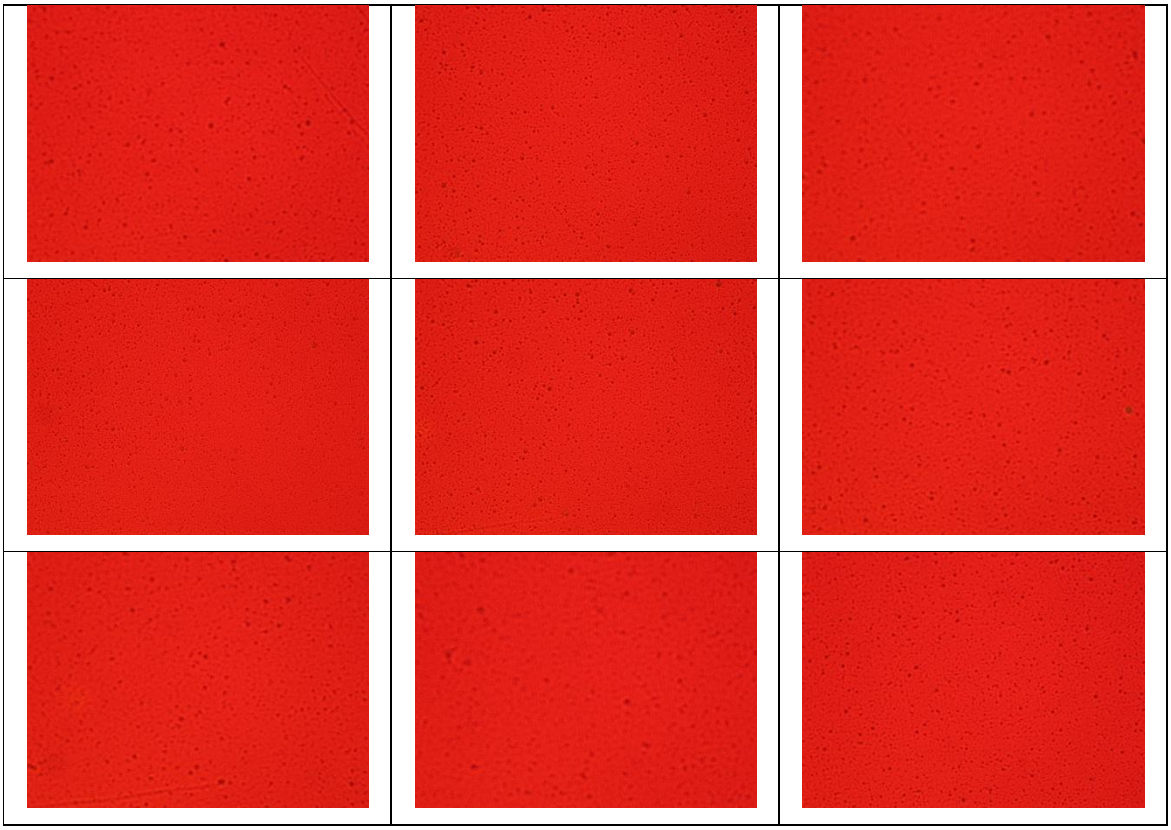
mmdairport	:	Detecting runways in satellite airport imagery.
mmdarea	:	Remove objects with small areas in binary images.
mmdasp	:	Detect the missing aspirin tablets in a card of aspirin tablets.
mmdbeef	:	Detect the lean meat region in a beef steak image.
mmdblob	:	Demonstrate blob measurements and display.
mmdbrain	:	Extract the lateral ventricle from an MRI image of the brain.
mmdcalc	:	Extract the keys of a calculator.
mmdcells	:	Extract blood cells and separate them.
mmdchickparts	:	Classify chicken parts in breast, legs, thighs and wings
mmdconcrete	:	Aggregate and anhydrous phase extraction from a concrete section observed by a SEM image.
mmdcookies	:	Detect broken rounded biscuits.
mmdcornea	:	Cornea cells marking.
mmdfabric	:	Detection of vertical weave in fabrics.
mmdfila	:	Detect Filarial Worms.
mmdflatzone	:	Flat zone image simplification by connected filtering.
mmdflow	:	Detect water in a static image of an oil water flow experiment.
mmdgear	:	Detect the teeth of a gear.
mmdholecenter	:	Hole center misalignment in PCB.
mmdlabeltext	:	Segmenting letters, words and paragraphs.

mmdleaf	:	Segment a leaf from the background.
mmdlith	:	Detect defects in a microelectronic circuit.
mmdpcb	:	Decompose a printed circuit board in its main parts.
mmdpieces	:	Classify two dimensional pieces.
mmdpotatoes	:	Grade potato quality by shape and skin spots.
mmdrobotop	:	Detect marks on a robot.
mmdruler	:	Detect defects in a ruler.
mmdsoil	:	Detect fractures in soil.

APPENDIX 4A:

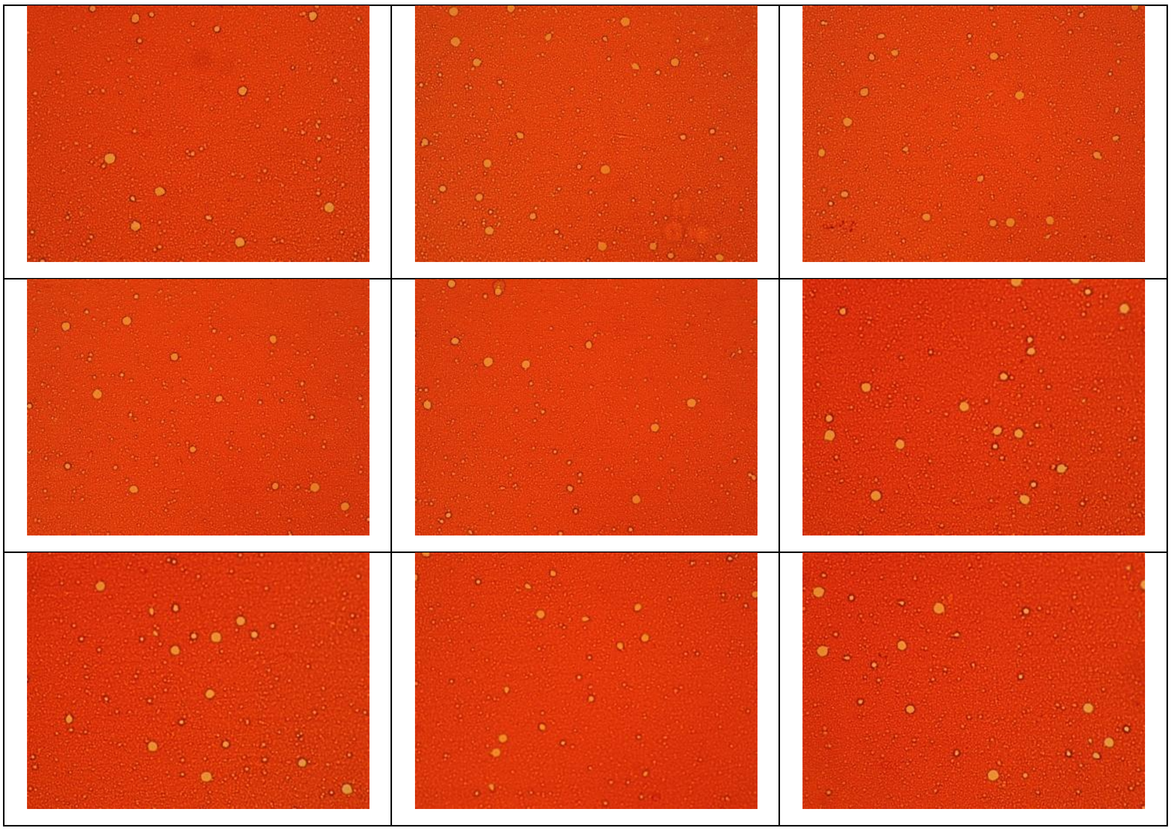
Sample Images;

Exposed time = 0 hour

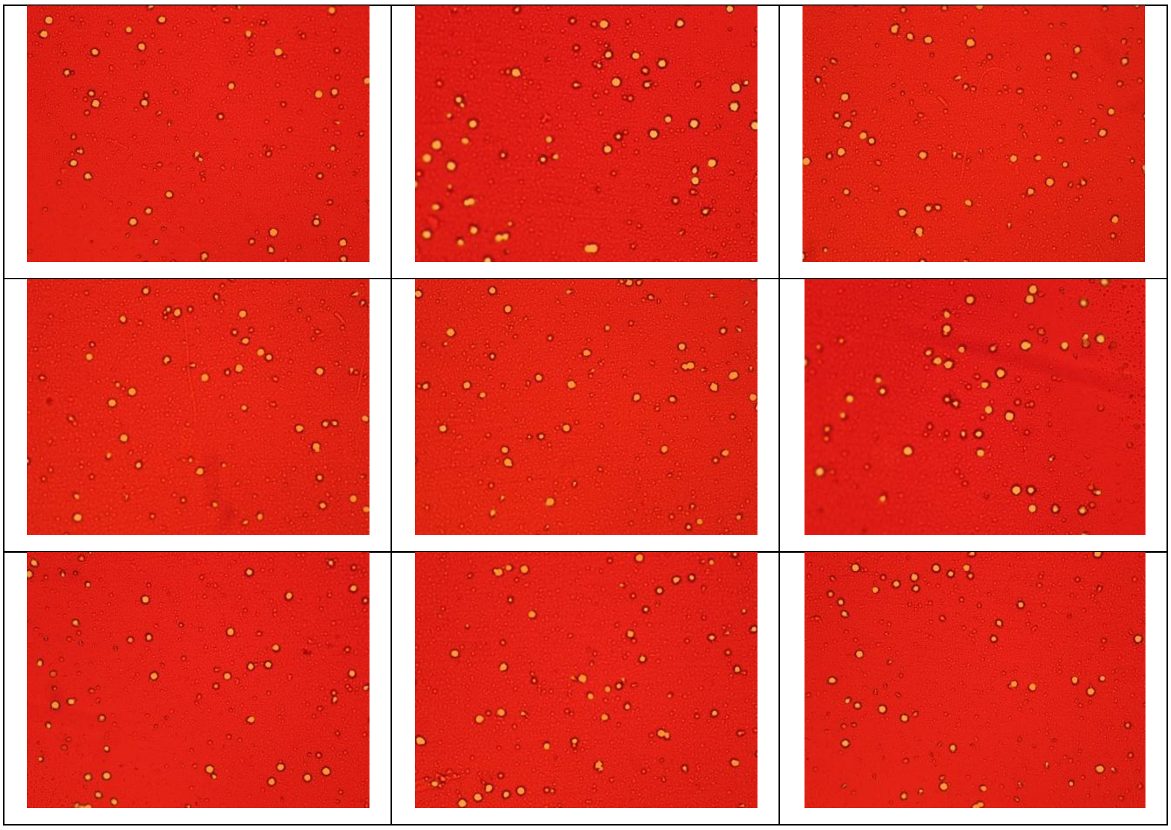


Appendix 4A:

Exposed time = 1 hour

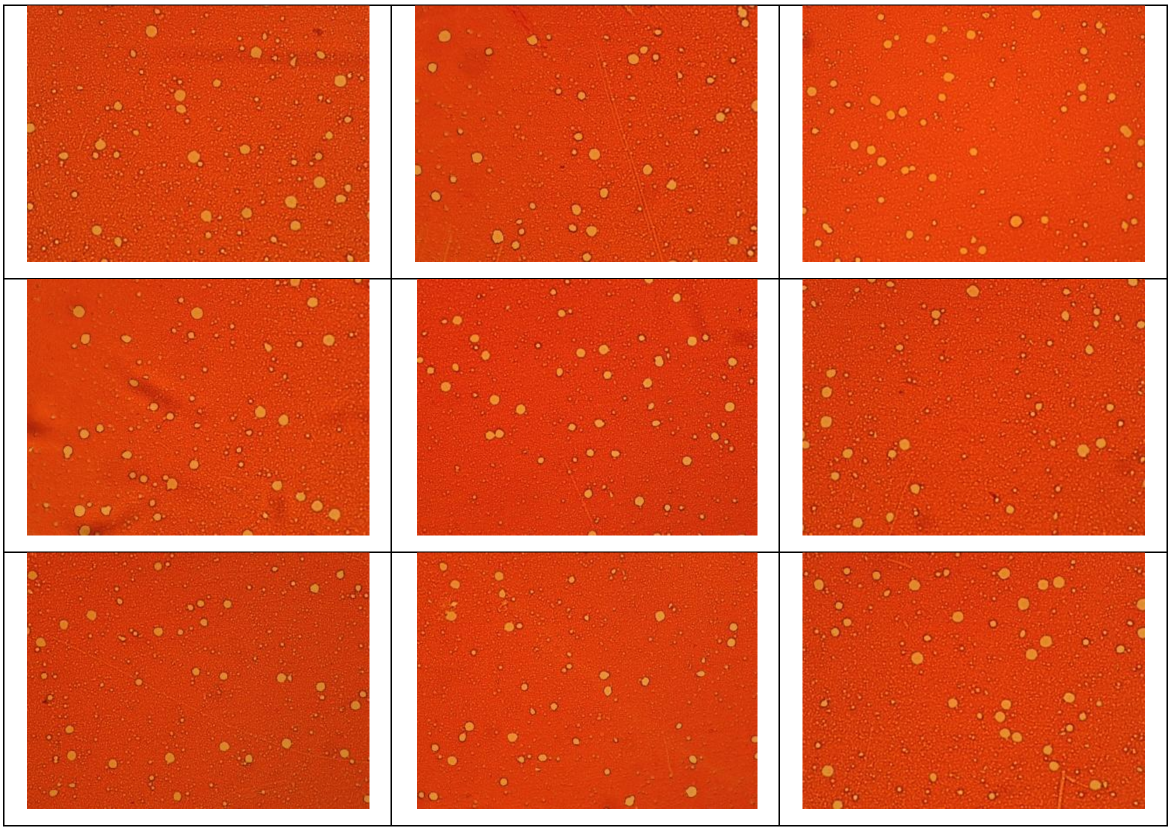


Exposed time = 3 hours

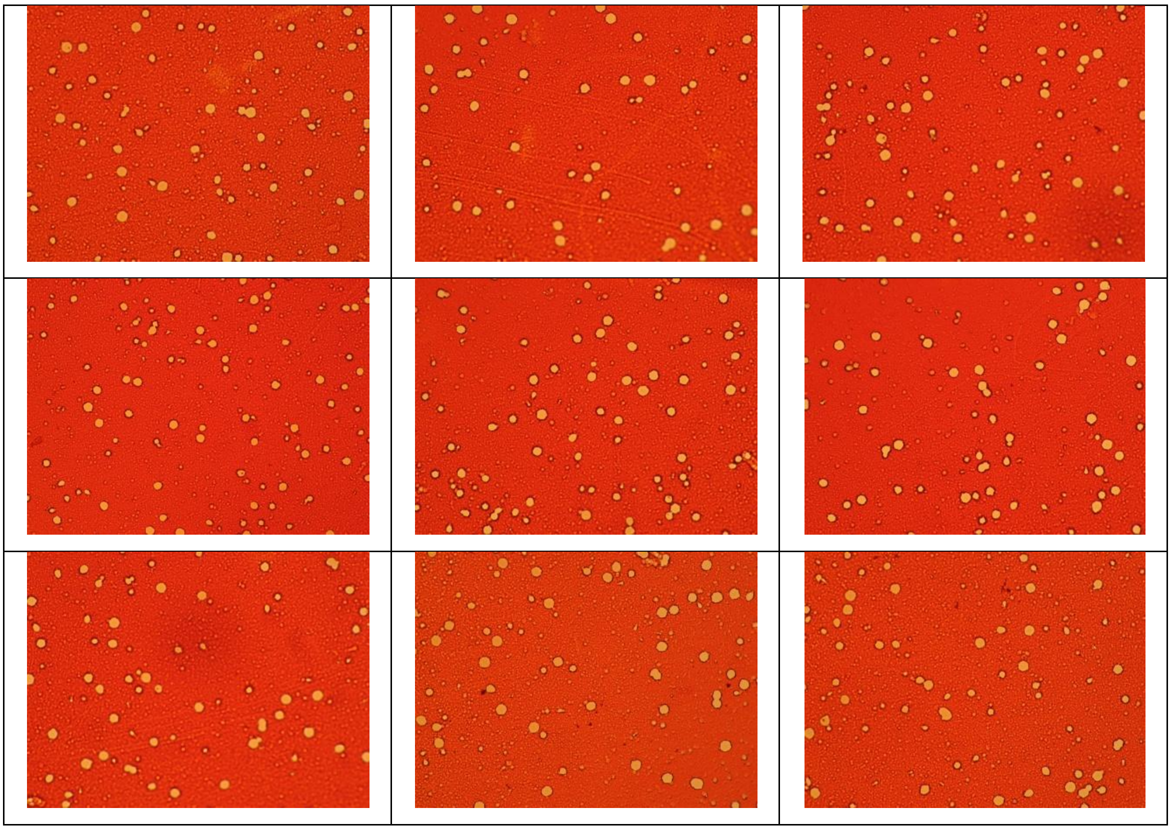


Appendix 4A:

Exposed time = 5 hours

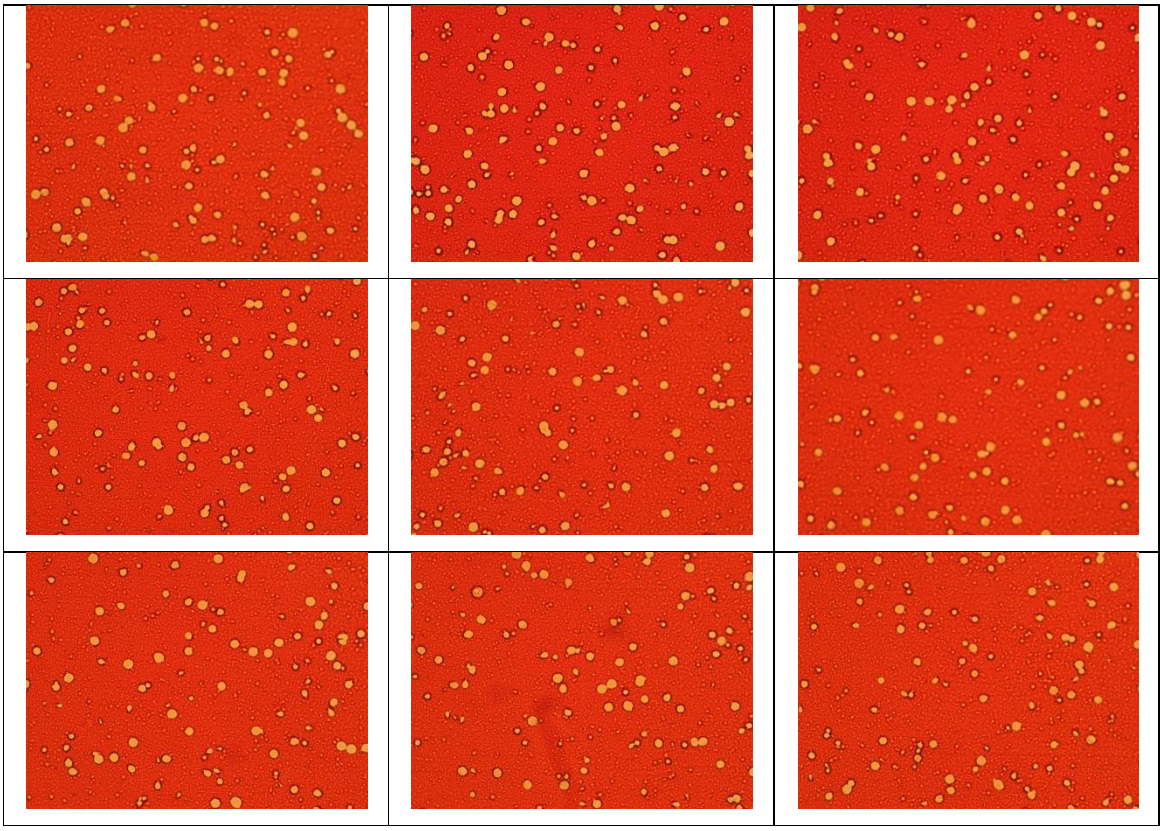


Exposed time = 7 hours

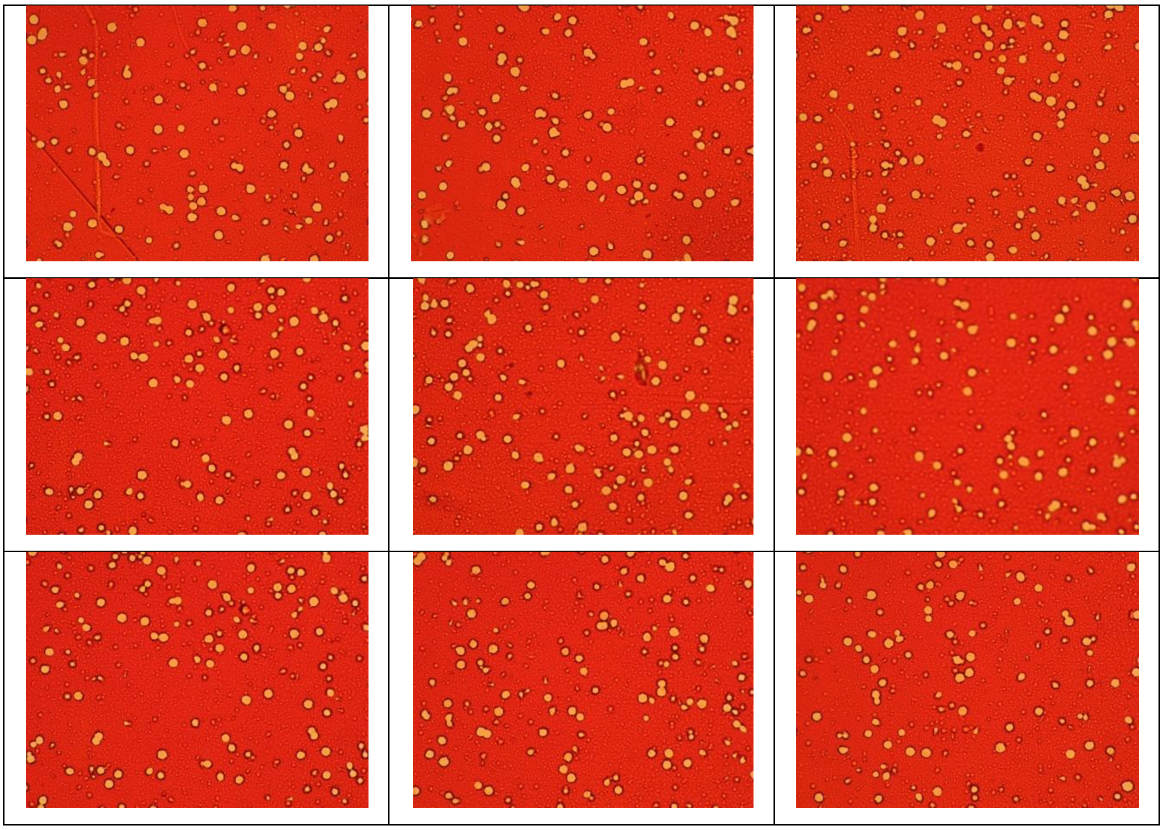


Appendix 4A:

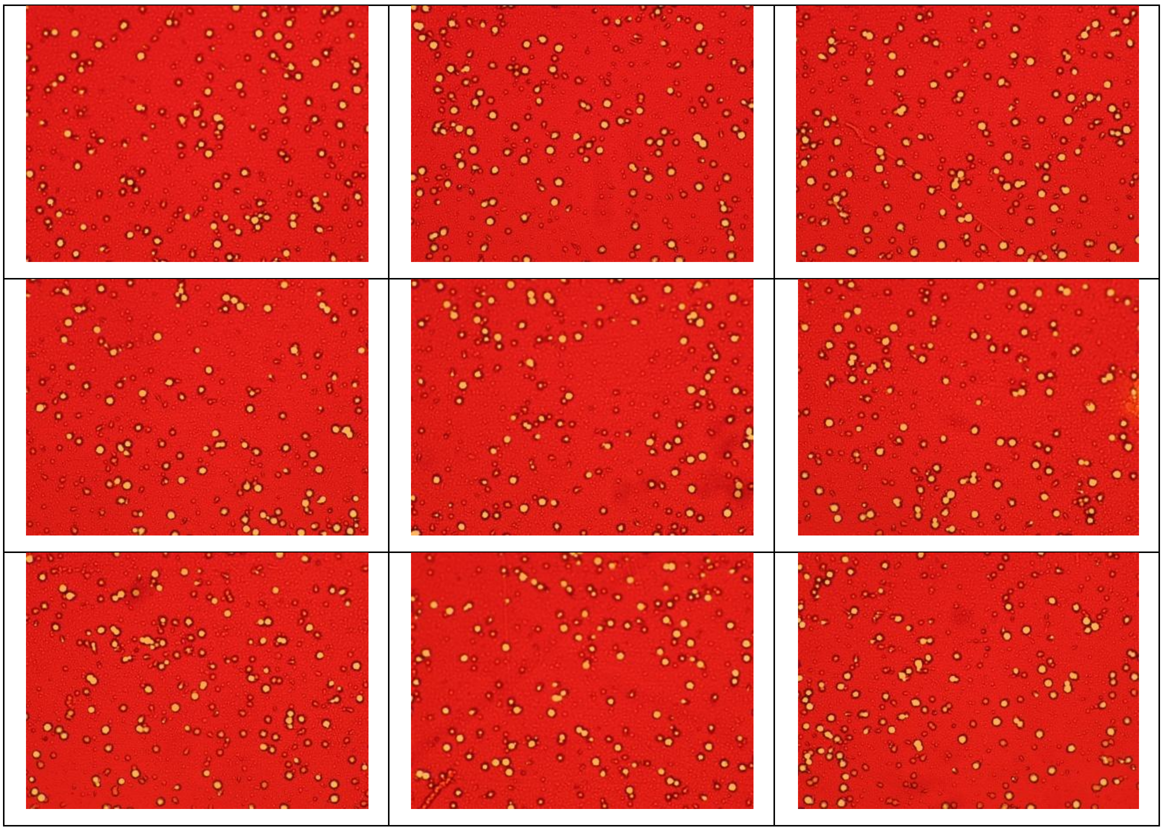
Exposed time = 10 hours



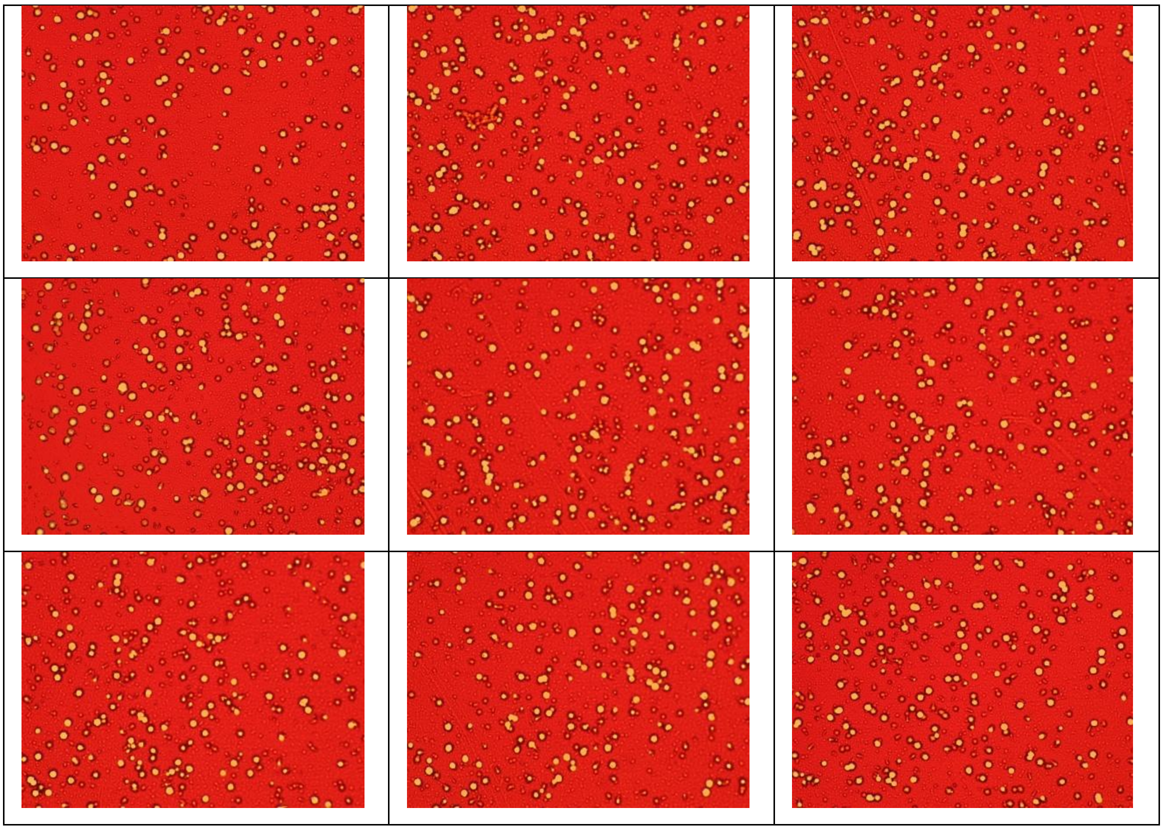
Exposed time = 14 hours



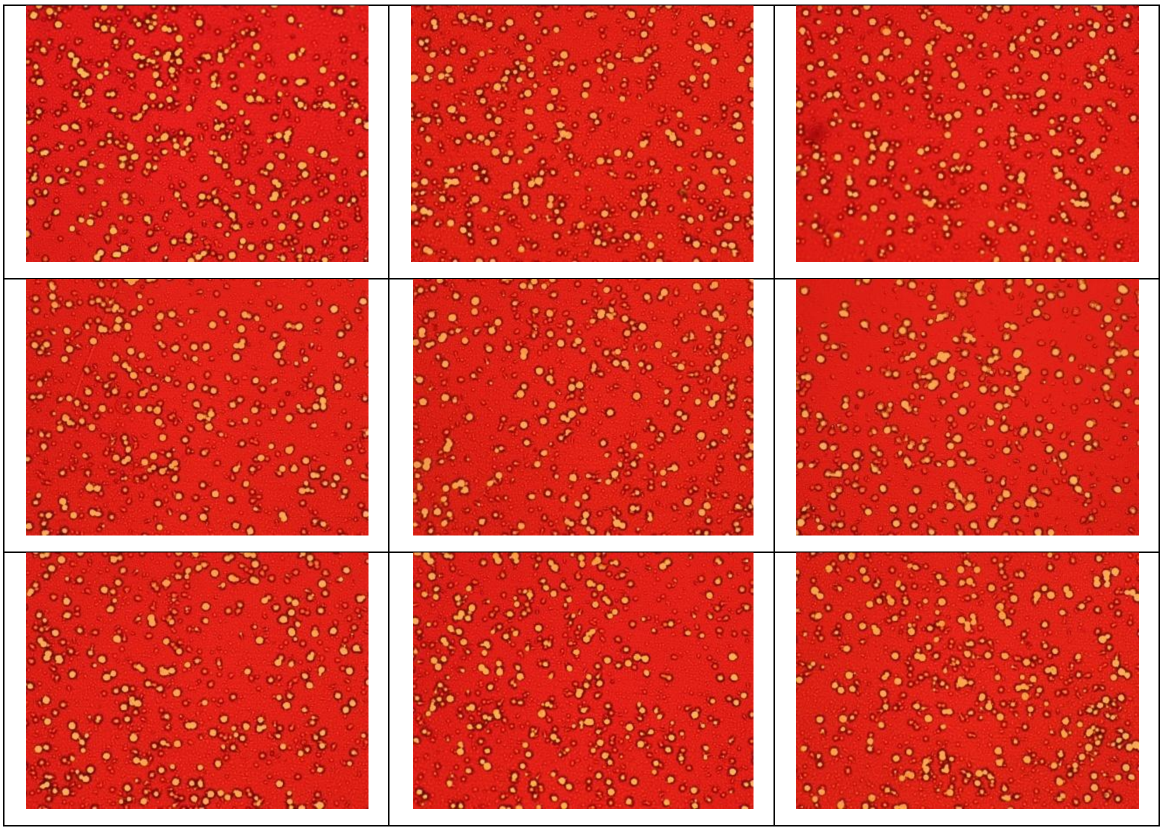
Exposed time = 20 hours



Exposed time = 31 hours



Exposed time = 43 hours



APPENDIX 4B:

Diameter of Nuclear Track Area Conversion to micrometre unit;

The shape of Nuclear track was presumed as circular (circle), A in pixel^2 unit and d is diameter;

$\therefore A = \frac{\pi d^2}{4}$ pixel^2

$=> d = \sqrt{\frac{4A}{\pi}}$ pixel

Note that, the image resolution is equal to 180 pixels per inch, and 1 inch is equal to 2.54 cm.

$$=> d = \left[\left(\sqrt{\frac{4A}{\pi}}\right)\left(\frac{1}{180}\right)\right] \text{ inch}$$

$$=> d = \left[\left(\sqrt{\frac{4A}{\pi}}\right)\left(\frac{1}{180}\right)\left(\frac{2.54}{1}\right)\right] \text{ cm}$$

The images were taken at 260 magnifications, hence the real value of d can be required by reducing its magnifications.

$$=> d = \left[\left(\sqrt{\frac{4A}{\pi}}\right)\left(\frac{2.54}{180}\right)\left(\frac{1}{260}\right)\right] \text{ cm}$$

$$=> d = \left[\left(\sqrt{\frac{4A}{\pi}}\right)\left(\frac{2.54}{180}\right)\left(\frac{1}{260}\right)(10000)\right] \text{ µm}$$

$$=> d = \left[\left(\sqrt{\frac{4A}{\pi}}\right)\left(\frac{25400}{46800}\right)\right] \text{ µm}$$

If A_{IPTM} = 149.80 pixel^2 and A_{SDC} =154.25 pixel^2, (Table 5:3);

$$\therefore d_{\text{IPTM}} = \left[\left(\sqrt{\frac{4(149.80)}{\pi}}\right)\left(\frac{25400}{46800}\right)\right] \text{ µm}$$

$$d_{\text{IPTM}} = 7.50 \text{ µm}$$

$$\therefore\ d_{\mathrm{SDC}} = \left[\left(\sqrt{\frac{4(154.25)}{\pi}}\right)\left(\frac{25400}{46800}\right)\right]\ \mu\mathrm{m}$$

$$d_{\mathrm{SDC}} = 7.61\ \mu\mathrm{m}$$