

**COLOUR IMAGE MAGNIFICATION**

By

Lim Boon Yong

A PROPOSAL

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONS)

INFORMATION SYSTEMS ENGINEERING

Faculty of Information and Communication Technology

(Perak Campus)

JAN 2011

# DECLARATION OF ORIGINALITY

I declare that this report entitled “**Colour Image Magnification System**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : \_\_\_\_\_

Name : \_\_\_\_\_

Date : \_\_\_\_\_

# ABSTRACTS

Digital contents, such as computer images in any form and type, are usually large and require substantial amount of server spaces to store it permanently. The increasing rate at which the volume of digitized images occupies an organization's server is overwhelming, and hence, may potentially use up all the available storage spaces quickly. Therefore, there is a need for an image interpolation technique to be developed to solve the problem and help to improve content management in any organization that needs to keep large amount of digital images. However, there have been many image magnification techniques introduced in the past, but only until recent decade has seen new proposed methods being developed to deal with colour image interpolation. Most of the proposed algorithms seem to work well to solve aliasing effect on computer graphics. However, many of the proposed methods do not consider minimizing edge blurring effect on the magnified image. This study proposed a hybridized interpolation method that combines the proposed interpolation algorithm with the edge detection method. The proposed algorithm consists of three stages. An image is replicated to two replicas. The first stage is to interpolate the first replica to two, four, or eight times whichever user likes using the proposed interpolation algorithm. In the next stage, the second replica will be converted into 3 bit or 4 bit colour image whichever user choose, it is for easy edge detection work to be performed. The detected edges are maintained while the similar pixels within the edge regions are removed. The proposed Geometrical Pattern Classification approach will be conducted to form the maintained pixels along the edges with correctly chosen colours in the positions where jaggies normally occur due to magnification process. This is also performed to minimize the smoothing effect caused by the interpolation algorithm taking the weighted average of the surrounding pixels. The final stage is to map the outline of the detected edges into the first replica, and then a classifier will be adopted to refine the final result. The magnified image with clear clarity is expected to be produced in the project. The initial result was evaluated by conducting the satisfaction survey, which is based on these factors such as clarity, jagged edge problem and sharpness of the magnified image.

# Table of Contents

<b>TITLE</b>	<b>i</b>
<b>DECLARATION OF ORIGINALITY</b>	<b>ii</b>
<b>ABSTRACT</b>	<b>iii</b>
<b>TABLE OF CONTENTS</b>	<b>iv</b>
<b>LIST OF FIGURES</b>	<b>vii</b>
<b>LIST OF TABLES</b>	<b>viii</b>
<b>CHAPTER 1 INTRODUCTION</b>	
1.1 Introduction	1
1.2 Problem Statement	1
1.3 Scope	2
1.4 Objectives	3
1.5 Contribution	4
<b>CHAPTER 2 LITERATURE REVIEW</b>	
2.1 Literature Review	5
2.1.1 Non-Adaptive Algorithms	5
2.1.2 Strengths and Weaknesses of Non-Adaptive Algorithms	8
2.1.3 Adaptive Algorithms	8
2.1.4 Strengths and Weaknesses of Adaptive Algorithms	12
2.2 Critical Remark of the previous works	14
2.2.1 Similarities between Adaptive and	



4.1.3	3bit and 4bit Colour Palette	32
4.1.4	The Algorithm for Preserving Edges	33
4.1.5	The Proposed Geometrical Pattern Classification Approach	35
4.1.6	Final Stage of the Proposed Algorithm	35
4.2	Testing	37
4.3	Verification	39
<b>CHAPTER 5 CONCLUSION</b>		
5.1	Discussion and Conclusion	40
<b>BIBLIOGRAHPY</b>		41
<b>APPENDIX A FIGURES AND TABLES</b>		A-1
A.1	Figures	A-2
A.2	Tables	A-18
<b>APPENDIX B SURVEY FORM AND SLIDE</b>		B-1
B.1	Slide	B-2
B.2	Survey Form	B-13
B.3	Survey Answer	B-15

# LIST OF FIGURES

- Figure 2.1 Linear Interpolation
- Figure 2.2 Cubic Interpolation
- Figure 2.3 Bilinear Interpolation
- Figure 2.4 Bicubic Interpolation
- Figure 2.5 Least Directional Differences Based on Linear and Cubic
- Figure 3.1 Work Procedure Using Prototyping
- Figure 3.2 The Proposed Interpolation Algorithm
- Figure 3-3 3bit Colour Classification
- Figure 3.4 4bit Colour Classification
- Figure 3.5 The Algorithm for Preserving Edges
- Figure 3.6 Geometrical Pattern Classification Approach
- Figure 3.7 The Final Stage of the Proposed Algorithm
- Figure 3.8 The Flow Diagram of the Proposed Algorithm
- Figure 3.9 The Proposed Interpolation Algorithm
- Figure 3.10 3bit Colour Palette
- Figure 3.11 The Flow Chart of the 3bit Colour Classification
- Figure 3.12 4bit Colour Palette
- Figure 3.13 The Flow Chart of the 4bit Colour Classification
- Figure 3.14 Pixel Array Structure
- Figure 3.15 The Predefined Geometrical Patterns
- Figure 3.16 The Process of the Geometrical Pattern Classification
- Figure 3.17 The Classifier

Figure 4.1 The Pseudocode of Read Bitmap Function

Figure 4.2 The Pseudocode of the Proposed Interpolation Algorithm

Figure 4.3 The Pseudocode of the Preserving Edge Algorithm

Figure 4.4 The Pseudocode of the Proposed Geometrical Pattern Classification Approach

Figure 4.5 The Pseudocode of the Final Mapping Function

Figure 4.6 The Pseudocode of the Classifier Function



# **LIST OF TABLES**

Table 4.1 The Header Format of 24bit BMP Image

Table 4.2 The Header Format of JPEG Image

Table 4.3 The Total Score of Each Method

## **Chapter 1**

### **1.1 Introduction**

In the 21st century, computer technology is rapidly advancing in the world. Digital contents, such as computer images in any form and type, are usually large and require substantial amount of server spaces to store it permanently. The increasing rate at which the volume of digitized images occupies an organization's server is overwhelming, and hence, may potentially use up all the available storage spaces quickly. Therefore, the images will normally have to be scaled down to an ideal size to save space before storage for better content management. Reduction of image size through scaling down brings about a common problem, which is that the clarity of the image is reduced and the quality decreases significantly when magnified. Therefore, there is a need for an image interpolation technique to be developed to solve the problem and help improve content management in any organization that needs to keep large amount of digital images.

In addition, high technologies of digital and optical zooming have been widely applied and embedded in digital camera and mobile communication devices. However, these devices are being restricted due to the limitation on its optical zooming capabilities. Hence, there is still demand for image magnification to enlarge captured images to desired sizes before printing [1][9].

### **1.2 Problem Statement**

Image magnifying is a common and prime problem in image processing. The traditional interpolations techniques, such as linear, cubic, bilinear and bicubic, have been introduced in the past. But, these methods are mostly requiring interpolation to read between the neighbourhood pixels, which seemed to work well to minimize the aliasing effect on computer graphics and produce a smooth interpolated image. However, these methods do not consider the edge information during the interpolation process [2][3][7]. As a result, the interpolated image will appear smooth but rather blurry in general. The reason for the enlarged image to appear blurry in its entirety is mainly due to the

interpolator assigns colours into the undefined pixels inside the interpolation region based on a weighted average model instead of taking its original colour. Therefore, there is a need to improve the interpolation process in order to produce smooth and clear colour images.

### 1.3 Scope

This project proposed a hybridized interpolation method that combines the proposed image interpolation technique with the edge detection method to achieve the desired output. The proposed image interpolation is expected to minimize the jagged effect on the edge of the magnified image, but it doesn't consider the smoothing. Thus, the proposed edge detection method is used to preserve the edge's information in order to increase the sharpness of magnified image.

By adopting the proposed method, the image processing consists of three stages. First stage, the image is replicated to two copies. First replica is used as original image to be interpolated to two times, four times or eight times (whichever user likes) its size using the proposed interpolation algorithm. By the time, the initial interpolated image is produced. In the second stage, the proposed edge detection algorithm will convert the second replica into 3-bit colour image or 4-bit colour image (whichever user like) for easy edge detection work to be performed. Then, an iterative loop will use to flush the similar pixels in the 3 or 4 bit converted image for preserving the edges. The final task in second stage is to trim the edges based on the proposed Geometrical Pattern Classification approach. By the time, the edge information such as the position of the pixels along the edges is retrieved. Final Stage, the edge information is used to compare to the first replica for touch-up work to be applied to colour the pixels with correctly chosen colours in areas where jaggies normally occur due to magnification process. Then, a classifier will be used to classify all the weighted average between edges back to its own weighted value within its same colour region itself. This is performed the minimization of the blurring effect caused by the image interpolation algorithm taking the weighted average of the surrounding pixels in the first stage. In the end of the project, a

software prototype with basic interface will be developed based on the proposed algorithm. The prototype is expected that works well to deal with 24bit Bitmap Images (BMP). The capability of magnifying others image format such as JPEG and PNG will be taken as the further improvement in future works.

### 1.4 Objective

The aim of the project is to contribute a new interpolation algorithm that works well to improve the clarity of images, which the visual output of the produced image is expected to be clearer and sharper. The problems such as aliasing effect and jagged edge are focused in the project.

The proposed algorithm is an innovative algorithm that combines the proposed image interpolation technique with the edge detection method. The purpose of combining these two methods is because both methods are aimed to solve different specific problems. The image interpolation algorithm is developed to improve the visual effect of the target image, but it doesn't consider the smoothing effect on edge regions. [7] Therefore, the edge detection technique is adopted to solve the problems that have been neglected in the first stage. The proposed edge detection method is concentrated to gather the accurate edge information for the touch up work to be done in the final stage. Normally, the colour pixels along the maintained edge always occurs jaggies. Therefore, the trimming job is to be conducted by forming the correct colour into the targeted pixels where the jaggies occurred. By combining these two methods, the two outputs obtained from image interpolation and edge detection are composed by mapping the correctly chosen colour on the initial interpolated image based on the edge template. The final output of image will be clearer and sharper than initial interpolated image. Moreover, the focused problems could be minimized.

In this project, the software prototype is only applied in computers, and this is not covered any application or firmware that are embedded into digital camera or mobile device.

## 1.5 Contribution

The main contribution of the project is to develop the image magnifying software to help those of the people who encounter with the problem of low resolution images because of poor hardware capability of cameras used to take photos. The proposed algorithm is able to help them to transform the low resolution image to higher resolution. This software is trying to benefits them by giving the satisfactory image. [7]

However, there have many problems in image processing. The algorithm is only focusing to solve the jagged edge and aliasing effect of the magnified image. The development of digital zooming is being foreseen that contains higher potentiality than optical zooming in future. By comparing digital zooming and optical zooming, the capability of digital zooming is always based on the optical device its size. Higher degree of zoom requires the larger size of the optical device. In other words, the optical zooming capability can be known as equal to the physical size of the optical device itself. Hence, the capability of optical zooming is being restricted, for example: People can't or not convenient to carry a very big camera to any place as theirs like, although it has high optical zoom capability. Otherwise, the digital image magnifying capability is more and more advance today. The capability is derived from the efficiency of the algorithm. Hence, it is only concerned on how does the problem solved by human. [1][8]

## Chapter 2

### 2.1 Literature Review

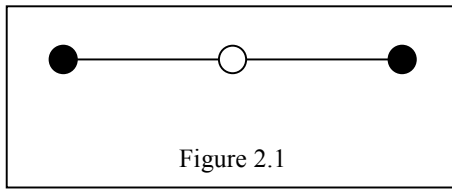
Image interpolation seems has been widely used nowadays, it occurs anytime when resizing, resampling and magnifying the digital image from one pixel grid to another. In recent decade, most of the proposed algorithms and methods are dealing with the image interpolation to reduce image magnifying problem. Commonly, these algorithms are categorized into two types, such as adaptive and non-adaptive. Adaptive algorithm are the algorithm that interpolating the colour pixels depending on image perspective such as edge information and colour direction, whereas non-adaptive methods are interpolating the colour pixels by reading in between the neighbour pixels. [13][7]

#### 2.1.1 Non-Adaptive Algorithms

Non-adaptive algorithms have been known as the traditional methods in image interpolation, these algorithms include nearest neighbour, linear, cubic, bilinear, bicubic and others. These algorithms are depending on the calculation to calculate the colour value from the adjacent pixels, and interpolating the weighted average colour pixels to magnify the image. In other words, non-adaptive algorithms are only used to treat all the pixels equally. Therefore, traditional non-adaptive algorithms produce the smooth image without considering the edges. However, some researchers have tried to solve the problem of the traditional algorithms based on modifying the mathematic equation. For example: Spline, Sinc and others. [13]

##### *2.1.1.1 Linear interpolation*

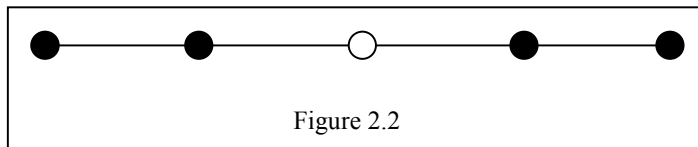
It is a simple form of interpolation. Linear interpolation algorithm considers the closest 2 neighbour pixels where at the both side of the interpolating pixel.



Two ● are the known pixels, ○ is the interpolated pixel. The colour value of ○ is the average of these two ● values. [7]

*2.1.1.2 Cubic interpolation*

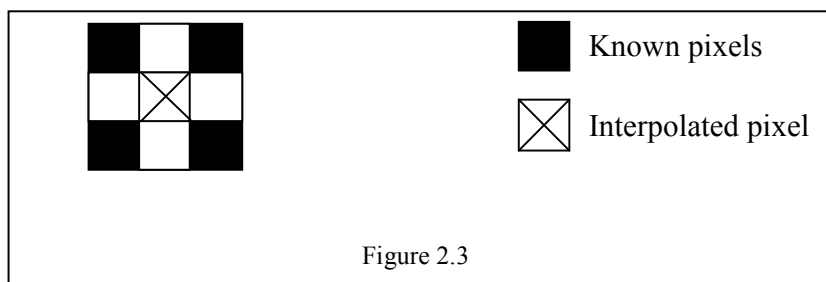
It is more accurate than linear interpolation, because the consideration of the neighbour known pixels is greater than linear interpolation. It considers the closest 4 neighbour pixels at the both side of the interpolating pixel.





Four ● are the known pixels, ○ is the interpolated pixel. The colour value of ○ is the average of these four ● values. [7]

*2.1.1.3 Bilinear interpolation*

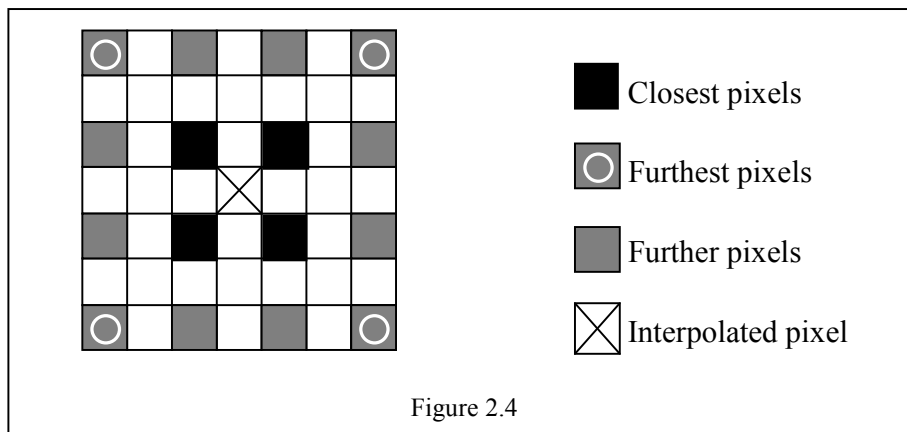
Bilinear interpolation is an extension of linear interpolation. It considers the closest 2x2 neighbourhood of known pixel values that surrounding the interpolating pixel.



The value of  is the average of these four  values. The result of this algorithm is much better than linear interpolation. [13]

#### 2.1.1.4 Bicubic interpolation

Bicubic interpolation is an extension of cubic interpolation. It considers the closest 4x4 neighbourhood (for a total of 16 pixels) of known pixels. The closer pixels are given a higher weighting in the calculation.



The image produced by Bicubic is better than bilinear interpolation, because of the larger amount of neighbour pixels are considered. [13]

#### 2.1.1.5 Nearest neighbour interpolation

It is the most fundamental interpolation. It only considers one pixel (the closest one to the interpolating pixel). This has the effect of simply making each pixel bigger [13]

#### 2.1.1.6 Higher order non-adaptive interpolation

There are many higher order interpolation algorithms which take more surrounding pixels into consideration. These algorithms include Spline and Sinc. The



result of the output image is definitely better than the traditional methods (bilinear and bicubic), but the higher computation power is crucially required. [13]

### 2.1.2 Strengths and Weaknesses of the Non-Adaptive Algorithms

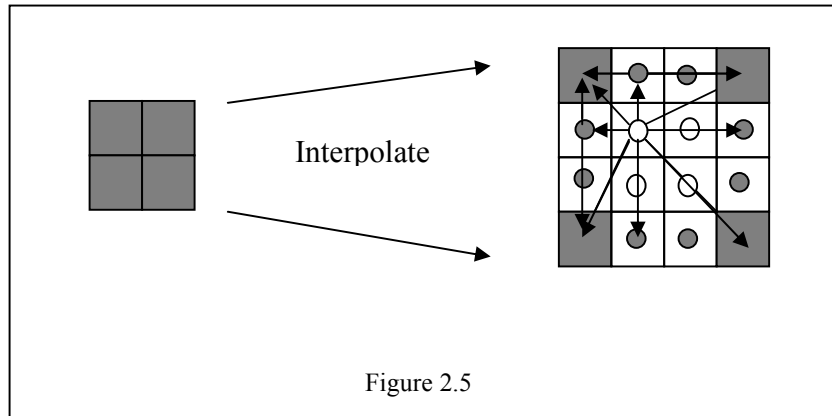
Nearest neighbour, linear, cubic, bilinear and bicubic have the strengths that the algorithms are simple to implement and low complexity of the algorithm. However, the weakness is the produced image exist the obvious problems such as jagged edge and smoothing effect, although the visual effect of the image looks good. The strength of Spline and Sinc is that it's able to produce the better result of magnified image, but the weakness is required higher processing load.

### 2.1.3 Adaptive Algorithms

Adaptive algorithms are more rely on detecting the presence of the edge. Generally, the aim of these algorithms is to minimize the unsightly interpolation artifacts such as edge blur and jaggies. Nowadays, there have many adaptive algorithms been introduced, most of the algorithm are more preferable to be applied in mathematic way to detect the edges. This kind of algorithm is relying on the mathematic equations to calculate the position of pixels along edges rather than studying the pattern of the edges in the original image to retrieve the edge information.

#### *2.1.3.1 Least Directional Differences based on Linear and Cubic*

This method is more sophisticated and produces smoother edges than both bilinear and bicubic interpolations. The difference between this algorithm and traditional linear and cubic interpolation is the consideration of known pixels. [7]



is the original pixel. The four original pixels are interpolated by using linear interpolation.  and  are the interpolating pixel. The arrows indicate the known pixels that need to consider. The pixel  is interpolating with only consider the two original pixels that beside the interpolating pixel. The pixel  is interpolating with consider the directional eight pixels (special indicated pixel).

### 2.1.3.2 Progressive Refinement Approach

It is a method that doubling the size of images in which the sharpness of the edges is preserved without introducing distinct artifacts in the magnified images. The algorithm consists of two steps; first step is the generation of an initial magnified image, then, second step is the progressively refining the magnified image to produce a high quality image. The main contribution of this approach is the image refinement. This refinement is performed progressively by transforming the value of each pixel, producing a new magnified image that owns the colours closer to its input image. The authors mentioned the limitation of the algorithm is the method used to initial magnified image does not guarantee the smooth effect of the edges in the magnified image. Hence, jagged edge may still appear in the final magnified image. [12]

### *2.1.3.3 Vector Operators*

It is a vector filtering framework for colour image magnification. Popular vector filters include vector median filter, the basic vector directional filter and data-adaptive vector filter. Vector median filter is most basic filter in these three; it used the minimization concept to ensure the outputting of colour vector which is the most similar as well as under the aggregated distance criterion. The basic vector directional filter has inherited the functionality from vector median filter and improved the minimization approach that able to be useful for directional data such colour data. Vector data-adaptive filter is advance than the previous two filters. It has inherited the functionalities from vector median filter and basic vector directional filter. Thus, vector data-adaptive filter can operate either on magnitude or direction of the vectorial inputs. Respectively, produce different weights and the filtering scheme will result a clearer image than the previous two filters. [9]

### *2.1.3.4 Image Sharpening Algorithm*

It is an algorithm that is inspired by the usage of gradient profile prior, but the author utilizes it to sharpen an image directly instead of using it as regularization for reconstruction. The main idea of this algorithm is to find out the gradient profile of each pixel and change the sharpness of the gradient profile accordingly. It used bicubic interpolation to compute the intensity of those pixels which locations are not aligned with the image grid. The extracted gradient profile is segmented to obtain a single mode that contains the center pixels. The center pixels are the boundaries between two colour domains, which are also the edges of the image. [3]

### *2.1.3.5 Edge-Forming Method*

It is a method that based on non-convex and non-linear partial differential equations. The equations are carefully discretized and incorporating numerical schemes of anisotropic diffusion to able to form reliable edges satisfactorily. Anisotropic diffusion

is a technique that focuses on reducing image noise without removing significant parts of the image content. This algorithm is trying to remove or reduce image blur and checkerboard effect arising in image interpolation for colour images and to retrieve the accurate colour data that along edges. The non-linear partial differential equations result the minimization of a non-convex functional of the image gradient. The authors have employed an efficient simulation model such as alternating direction implicit (ADI) method which is known to be very efficient in solving diffusion equations defined on rectangular domains. [6]

### *2.1.3.6 Error-Amended Sharp Edge (EASE) Scheme*

The EASE scheme is based on the bilinear interpolation method and modified it to be able to amend the interpolation error. In EASE, each resampling evaluation requires at most 12 pixel values of the low resolution image for the estimation of the edge orientation and at most 7 pixel values for the resampling. The authors emphasized on implement the algorithm with ease and improve the ability of the algorithm to be better preserve edges than those conventional linear interpolations. [5]

### *2.1.3.7 Edge-Directed Interpolation*

It is an adaptive method that also focuses on edges to improve image magnification. The algorithm consists of two phases: rendering and correction. The rendering phase is edge-directed which is based on bilinear interpolation modified to prevent the interpolation across edges. The correction phase is to modify the mesh values on which the rendering is based to account for the disparity between the true low resolutions data, and that predicted by a sensor model operating on the high resolution output of the rendering phase. The overall process is repeated iteratively until the satisfactory image is obtained. [11]

### *2.1.3.8 New Edge-Directed Interpolation*

The basic idea of this approach is to first estimate local covariance coefficient from a low resolution image and then uses these covariance estimates to adapt the interpolation at a higher resolution based on the geometric duality between the low resolution covariance and high resolution covariance. It is a hybrid approach that switching between bilinear interpolation and covariance based adaptive interpolation that is proposed to reduce the overall computational complexity. Therefore, it is demonstrated to significant improvements over bilinear interpolation by the correction of bilinear interpolation errors. [10]

### *2.1.3.9 Adaptive Interpolation by Pixel Level Data-Dependent Geometrical Shapes*

It is a method that classify interpolation region in the form of geometrical shapes and then assign proper intensity values inside interpolation region to the undefined pixels while preserving the sharpness and smoothness at the same time. The algorithm includes four phases. In the first phase, the input image is expanded by inserting the undefined pixels in between each pixel. The second phase is to assign the colour values to the undefined pixels by pixel level data dependent geometrical shapes. Third phase is for scanning the magnified image line by line and looking for those pixels which left undefined in the previous phase. In the fourth phase, the median of the neighbouring defined pixels of the undefined pixel is calculated and then assign this calculated value to the undefined pixel. The authors have mentioned the algorithm has its own limitation that it is not always applicable to every image, but it is excellent for gray scale image. The algorithm is simple to implement and has lower computational complexity. [2]

### 2.1.4 Strengths and Weaknesses of Adaptive Algorithms

Least Directional Differences Linear and Cubic is a simple algorithm that modified the interpolation process of traditional method to improve the algorithm. It makes the algorithm able to investigate the colour values depending on the directional

structure. The algorithm has not emphasized on solving the smoothing effect on edges. But, it does well in solving the aliasing effect and produce a smooth image. [7]

Progressive Refinement and Edge-Directed are the new approach that differs with previous. These two methods have the similar functionality which is repeating the process iteratively to produce the clearer image. The progressive refinement approach is progressively refining the image to make it clearer. Edge-Directed is repeating the correction to correct the interpolation error. These two algorithms are simple and low complexity. The weakness of these two methods is the algorithm need to iteratively refine or correct the image many times until the expected result is produced. Therefore, it causes the longer processing time, even though the algorithm is low complexity. [12][11]

Vector Operator, New Edge-Directed are the different techniques for colour image magnification. Vector Operator is using vector filter to filtering image. The New Edge-Directed method has inherited the characteristic of the previous Edge-Directed method which is also the correction of interpolation error. But, the New Edge-Directed method has not adopted the iterative correction. It uses switching between bilinear interpolation and its own covariance based adaptive interpolation. Based on the experiment result, these two methods are proved that are able to produce the satisfactory results, but they have the limitation due to theirs zooming factor. [9][10]

Image Sharpening and Pixel Level Data-Dependent Geometrical Approach are the methods that use alternative ways to preserve the edges. Image Sharpening is using the gradient profile to find out center pixels, these center pixels are recognized as the pixels that along the edges. Pixel Level Data-Dependent Geometrical Approach is based on the pattern of geometrical shapes to form the colour into the correct position. These two approaches are doing well in reducing the smoothing effect on edge, but only efficient in gray scale image. [3]

Edge-Forming and Error-Amended Sharp Edge are the pure mathematical approaches that done well in predicting the edges in any circumstance. These algorithms are relying on calculating the edge position mathematically rather than studying the

pattern of edge in original image. Thus, it is able to handle the image which has serious jaggies existed before zooming. [5][6]

## **2.2 Critical Remark of the previous works**

### 2.2.1 Similarities between Adaptive and Non-Adaptive Algorithms

There have many similarities between the adaptive algorithms and non-adaptive algorithms in those algorithms that are used through mathematical equations to retrieve the most accurate colour values. The similarities can be categorized accordingly.

#### *2.2.1.1 Complexity*

Most of the algorithms that believe the larger consideration of known pixels, the more accurate pixel values can be retrieved. Thus, the algorithms are expected that able to process the large amount of the input colour values. Generally, the input colour values are calculated through mathematical equation to find out the weighted average of colour value. Therefore, larger considerations of known pixels will increase the complexity of the algorithms as well as the image processing time. These kinds of algorithms involve adaptive and non-adaptive algorithms. For non-adaptive algorithms include bicubic, Spline and Sinc interpolations. Adaptive algorithms include Least Directional Differences Linear and Cubic, Error Amended Sharp Edge (EASE). [13][7][5]

#### *2.2.1.2 Characteristic*

Adaptive and non-adaptive algorithms have the similar characteristic that is the same objective to obtain the most accurate colour values. For non-adaptive algorithms such as nearest neighbour, linear, cubic, bilinear, bicubic that use the weighted average to calculate the colour pixel values. For adaptive algorithms such as Least Directional Differences Linear and Cubic and Edge-Forming Method that are also using its own equations to retrieve the colour values. Otherwise, some of the approaches do not focus on obtaining the accurate colour values to produce the clearer magnified image. These

approaches are using alternative ways such as vector operator, geometrical classification, iterative refinement, error correction and gradient profile. [5][6][7][12]

### 2.2.2 Relationship between Adaptive and Non-Adaptive Algorithms

Adaptive and non-adaptive algorithms have a delicate relationship, because of the adaptive algorithms are designed to solve the problems that are occurred in non-adaptive algorithms. Some of the non-adaptive algorithms such as nearest neighbour, linear, cubic, bilinear, bicubic occur the jagged edge and smoothing effect due to the algorithm is interpolating without considering the edge information. Thus, there have many adaptive algorithms been introduced to dealing with this kind of problem. The initial objective for designing the adaptive algorithms is reducing the edge blur and smoothing effect to improve the clarity of image. [7][13]

### 2.2.3 Comparison between the Previous Methods and the Proposed Method

Some of the previous methods are doing well to improve the visual effect of the image but the jagged edge issue still appear obviously. Some of the previous methods are doing well in resolving the jagged edge issue but the originality of the produced image is suspected. However, the strengths of the previous method have always been taken as the inspiration in designing the algorithm in the project. [6][7][13]

The proposed method is a hybridize method that adopted the advantages of interpolation algorithm and edge-detection method. The weakness of the proposed method is the algorithm is efficient in magnifying gray scale and clip-art images. For colour photo image, it has not guaranty that it works well in any circumstance.



## Chapter 3

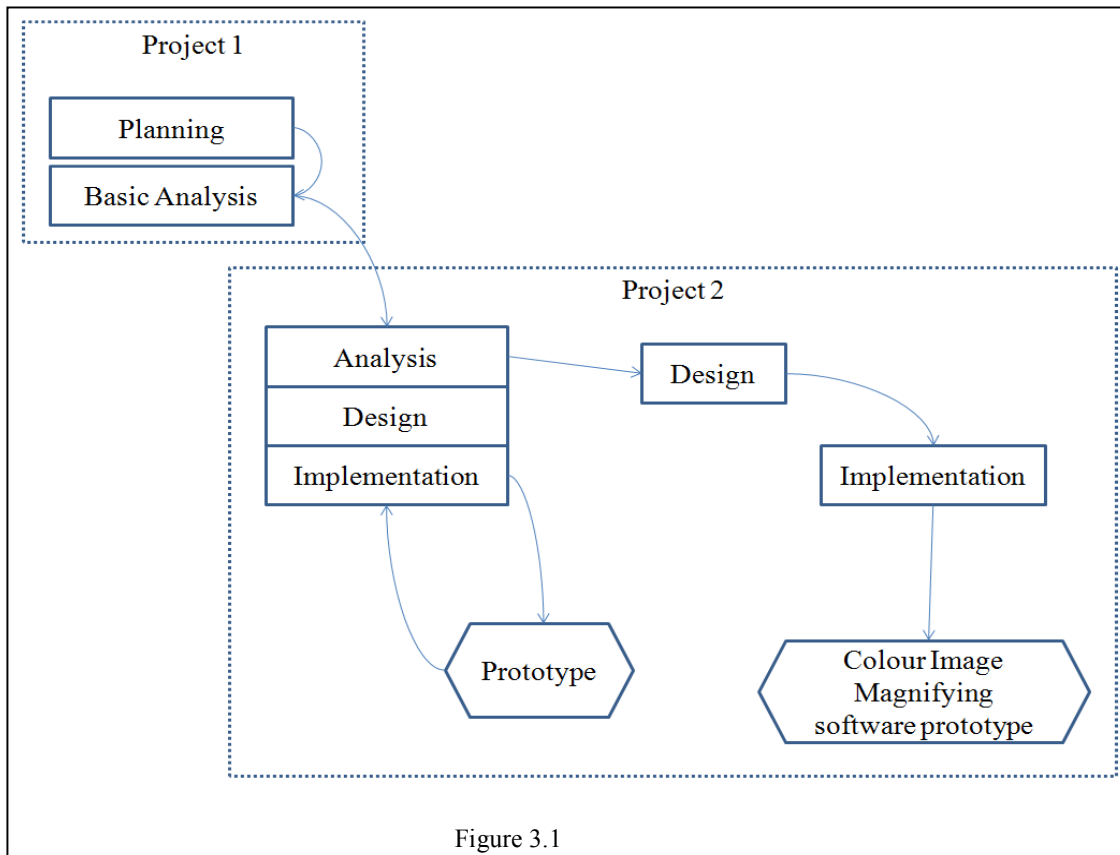
### 3.1 Methodology and Tools

According to my supervisor's suggestion, I adopt Throw-away Prototyping as the methodology and C++ Language as the software development language. At the end of the project, an image magnifying software prototype will be developed that fulfils customer's (user's) requirements. At this time, the targeted customers are involved supervisor and moderator who known as the expert user. The reason for choosing the C++ as the development language is because C++ is low level programming language that helps programmer in performing more feasible manipulation among functions and codes. With my objective, the output of the project should be efficient software which is able to produce the good quality of magnified image and do not consume longer processing time.

Throw-away Prototyping is a rapid prototyping that commonly used to reduce the risks when collecting inaccurate requirements. The prototype is developed from initial specification, and it is delivered for experiment then discarded. The throw-away prototype should not be considered as the final complete system.

By adopting Throw-away Prototyping, planning and basic analysis are finished in Project 1, then, analysis, design and implementation will be done in Project 2 and then develop the prototype for customer. The purpose of developing prototype is to investigate and gather the customer's requirements. Then, the initial prototype will be thrown away. The process of analysis, design and implementation will repeat until the sufficient data and information are gathered. Lastly, throw all the prototypes away and then start to develop the complete version of application. Thus, the process of gather user's requirements such as questionnaire survey and interview is skipped, it enable the final complete software prototype can be rapidly developed in the shortest time.

The figure shows the work procedure of Project 1 and 2. According to the project's scope, the final output of Project 2 should be the colour image magnifying software prototype with basic interface.

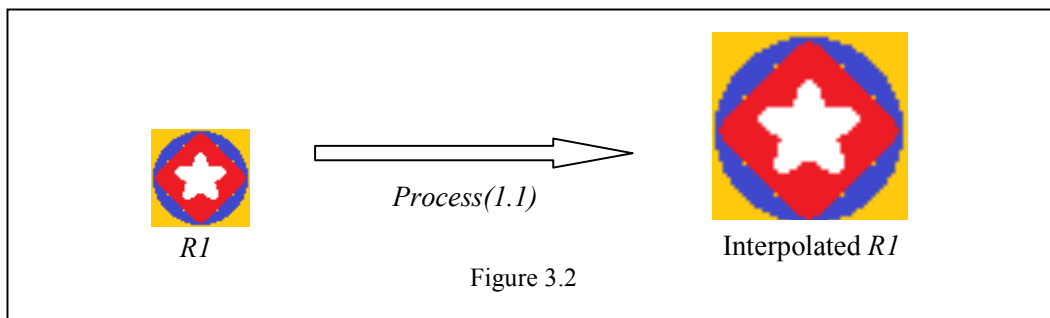


### 3.2 Design

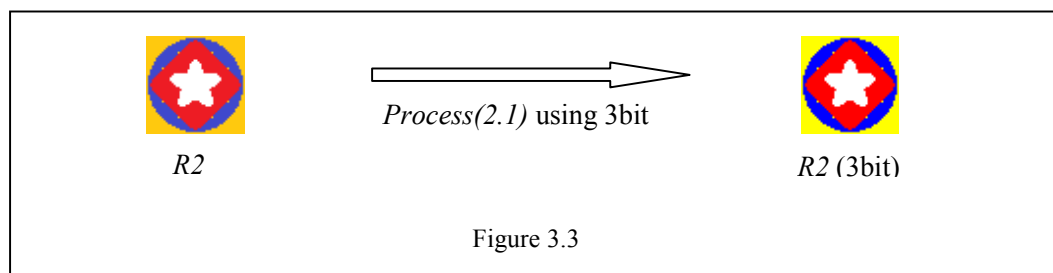
The proposed algorithm consists of three stages. First stage (*S(1st)*) involves the proposed interpolation method (*Process(1.1)*). Second Stage (*S(2nd)*) involves the 3 and 4 bit colour palette technique(*Process(2.1)*), the algorithm that flushing the similar pixels for edge-detection (*Process(2.2)*) and the proposed geometrical pattern classification approach (*Process(2.3)*). In the third stage (*S(3rd)*), it has a simple code that performs mapping the outline of detected edge to initial interpolated image (*Process(3.1)*) and a classifier (*Process(3.2)*) will be used for classifying the weighted average of the pixel between two edges based on the highest weighted value in its own 8 surrounding pixels.

#### 3.2.1 The Design of the Proposed Algorithm

First stage (*S(1st)*), an image is replicated to two copies (first replica (*R1*), second replica (*R2*)). *R1* is interpolated using the proposed interpolation algorithm (*Process(1.1)*).



Second stage (*S(2nd)*), *R2* is converted using 3 bit or 4 bit colour palette (*Process(2.1)*) whichever user likes. The algorithm (*Process(2.2)*) that able to flush the similar pixels for edge-detection is used to process *R2*. Then, *R2* will be expanded 2 times for the geometrical pattern classification approach (*Process(2.3)*) to be conducted.



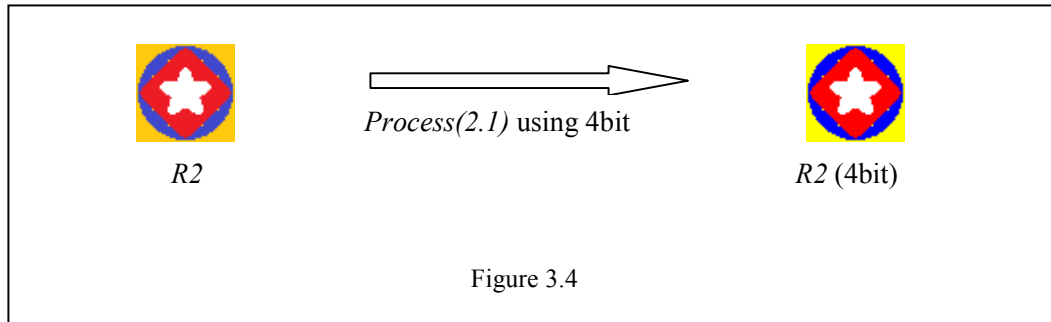


Figure 3.4

*\*For this image, 3bit and 4 bit have not different*

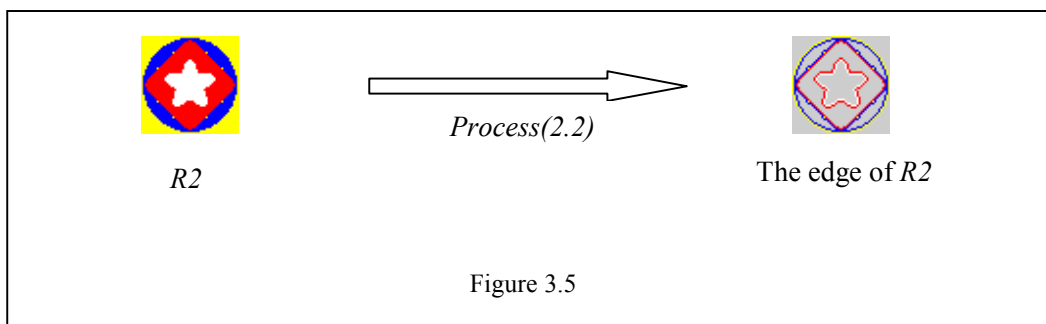


Figure 3.5

*\*The grey colour (205,205,205) is the default colour of undefined pixel set by Microsoft*

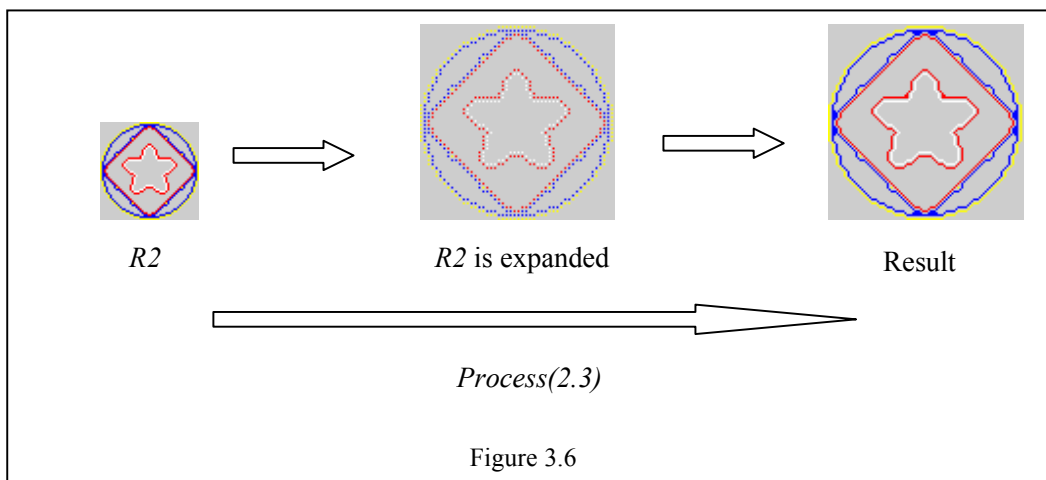
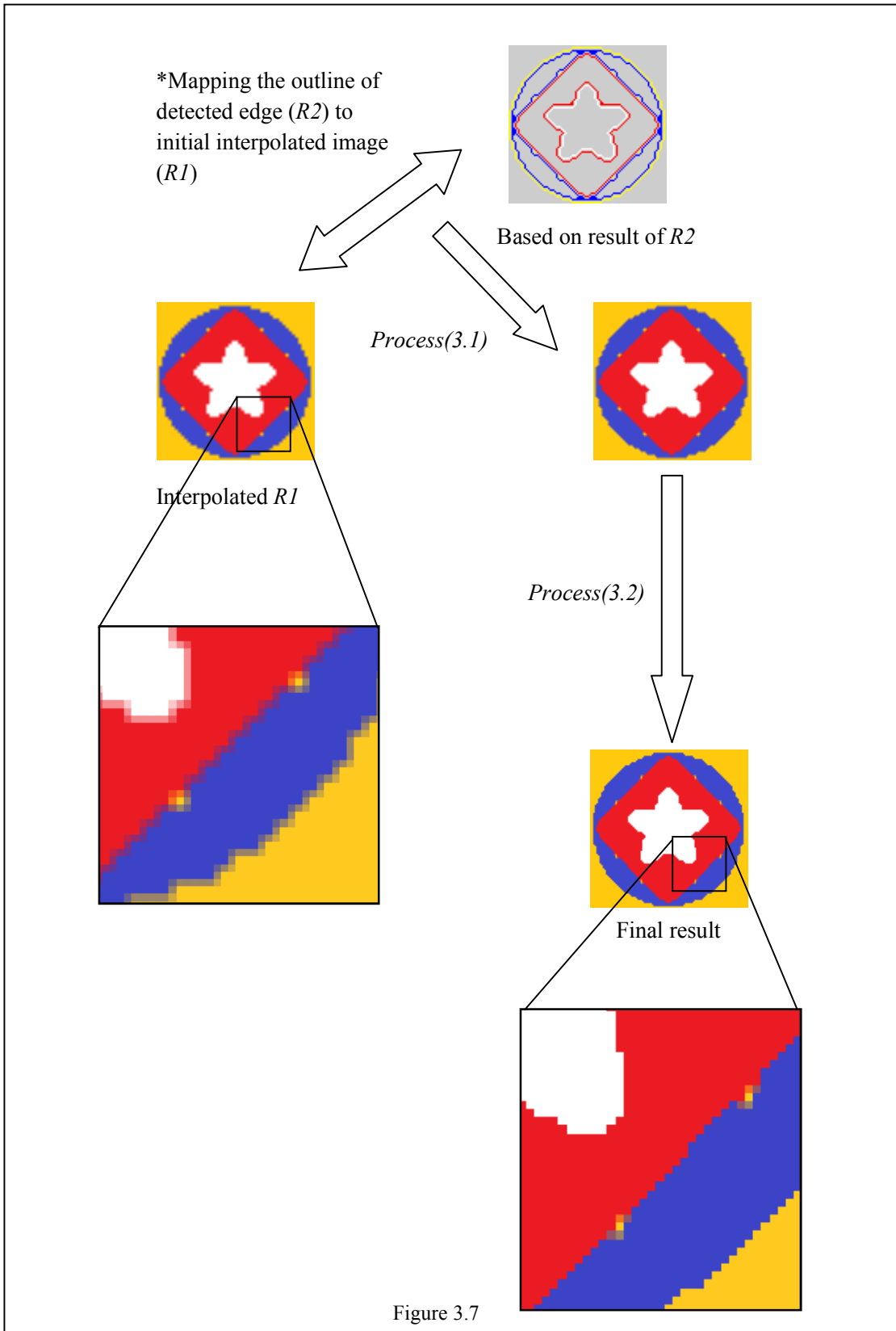


Figure 3.6

Third stage (*S(3rd)*), a code(*Process(3.1)*) that form the outline back to the interpolated *R1* based on the detected edge position in *R2*. Then, the classifier (*Process(3.2)*) will be used for classifying the weighted average of the pixel between two edges based on the highest weighted value in its own 8 surrounding pixels.



3.2.2 Summary of the Proposed Algorithm

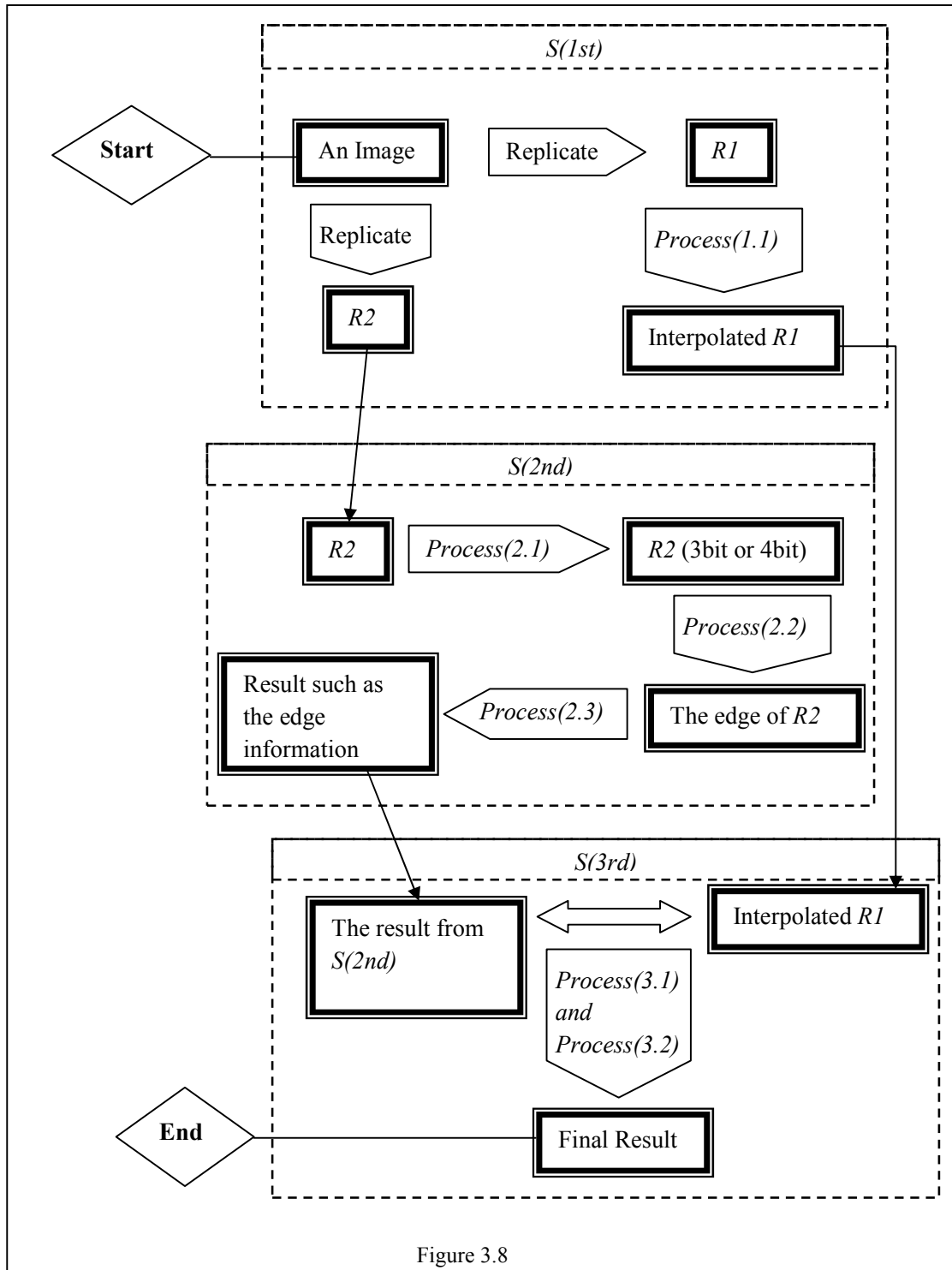
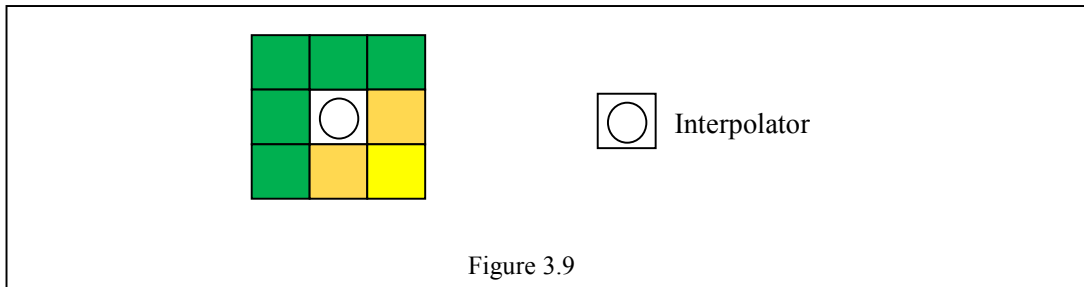


Figure 3.8

Figure 3.8 shows the process flow of the proposed algorithm

### 3.2.3 The Proposed Interpolation Algorithm (*Process(1.1)*)

The proposed interpolation algorithm (*Process(1.1)*) has slightly differed to the traditional interpolation algorithm such as Bilinear. Bilinear interpolator only considers the 4 surrounding pixels (refer to Figure 2.3 in Chapter 2). The proposed interpolation algorithm considers 8 surrounding pixels, as shown in Figure 3.9.



The value of the interpolator is the average of the 8 surrounding pixels. In Figure 3.9, it shows that ■ takes highest weighted value than others; ■ takes two, and ■ only one. The calculation is shown at below.

$$\text{The value of the Interpolator} = 5(\text{■}) + 2(\text{■}) + 1(\text{■}) / 8$$

### 3.2.4 The Edge-detection Method

The edge-detection method involves 3bit and 4bit colour palette (*Process(2.1)*), the algorithm for flushing similar pixels (*Process(2.2)*), and the proposed geometrical pattern classification approach (*Process(2.3)*).

#### 3.2.4.1 3bit and 4bit Colour Palette (*Process(2.1)*)

Colour palette can classify the entire colour pixels in the targeted image into only the specific colour category. 3bit colour palette consists of 8 colours (as shown in Figure 3.10). 4bit colour palette consists of 16 colours (as shown in Figure 3.12).

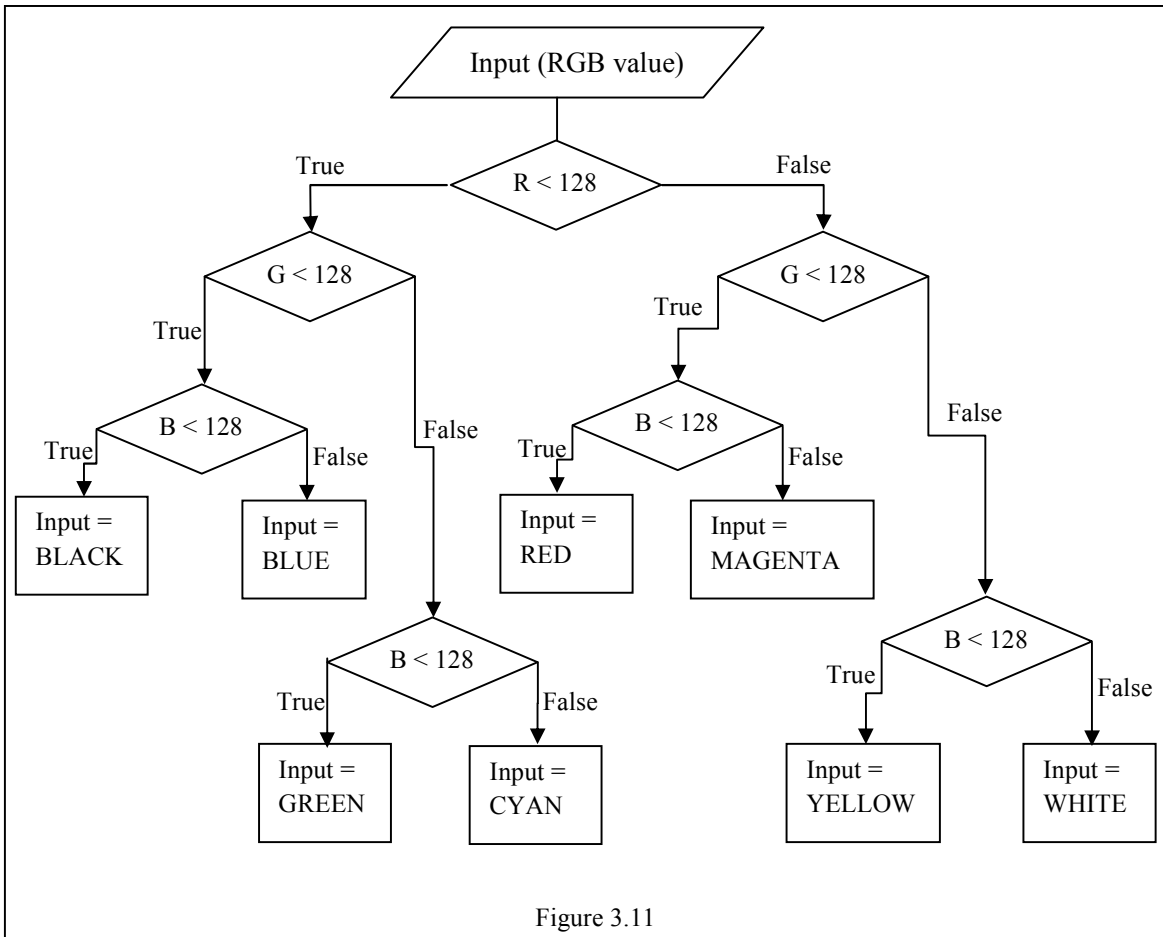
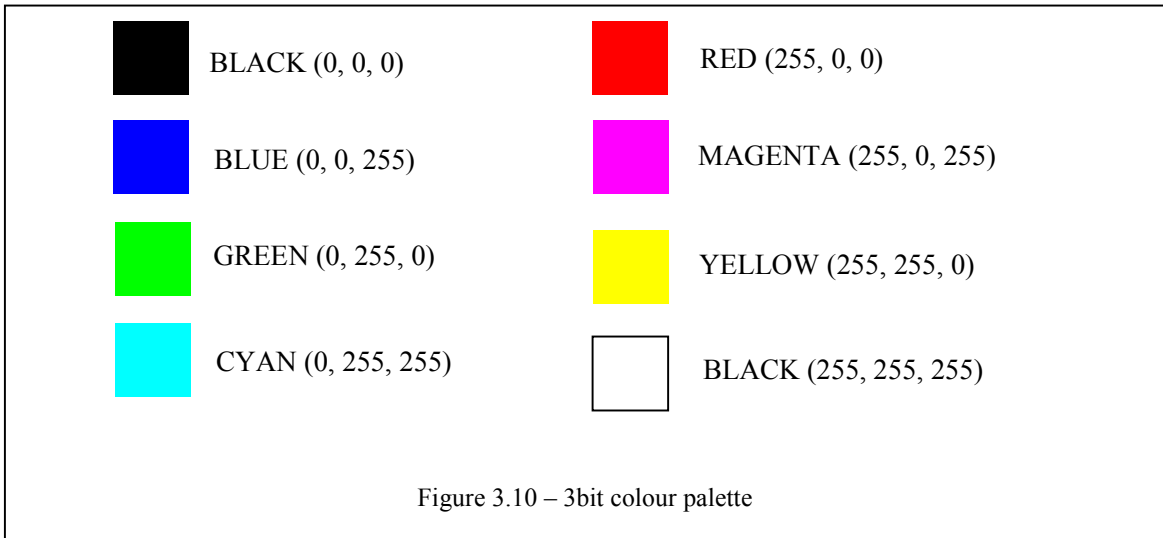


Figure 3.11 shows how does the 3bit colour classification process in the flow of If-Else statements.



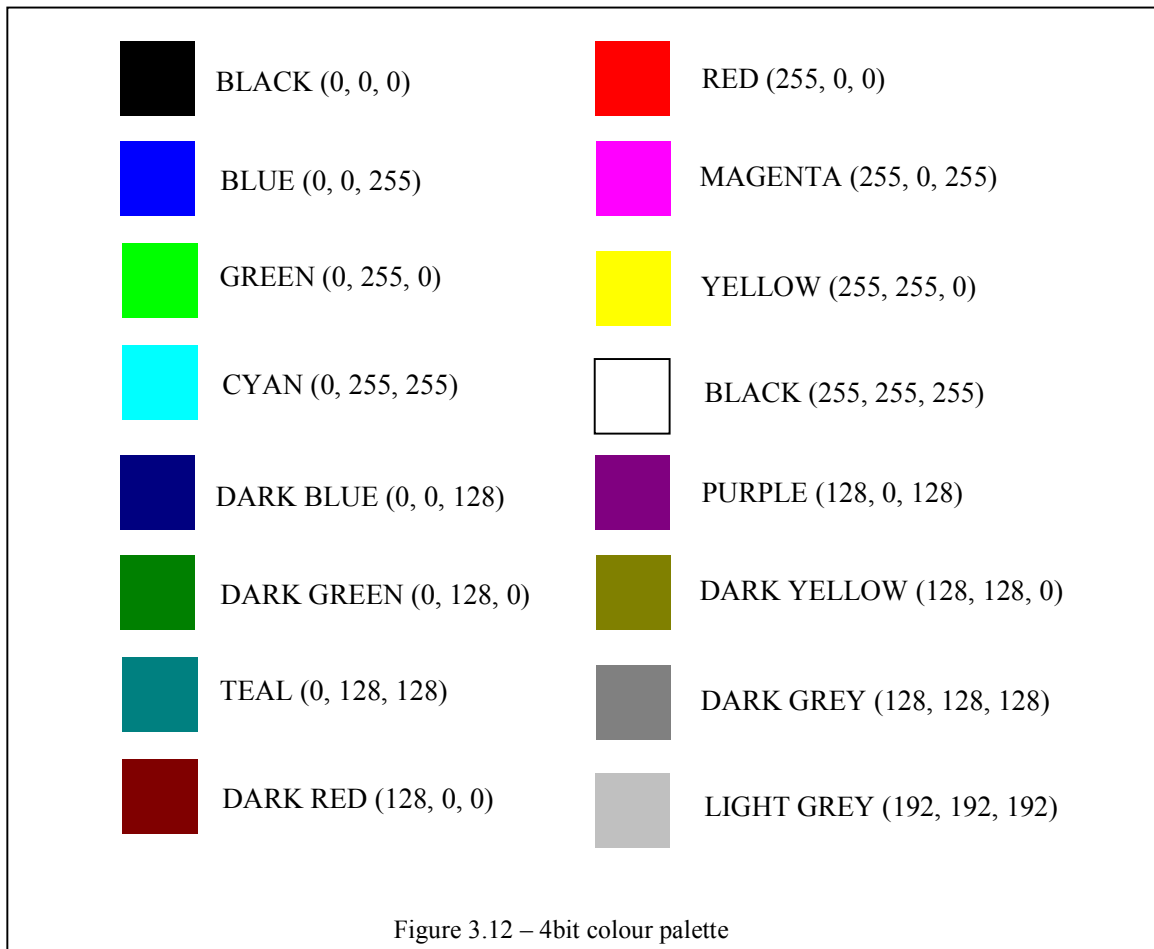


Figure 3.13 shows how does the 4bit colour classification process in the flow of If-Else statements.

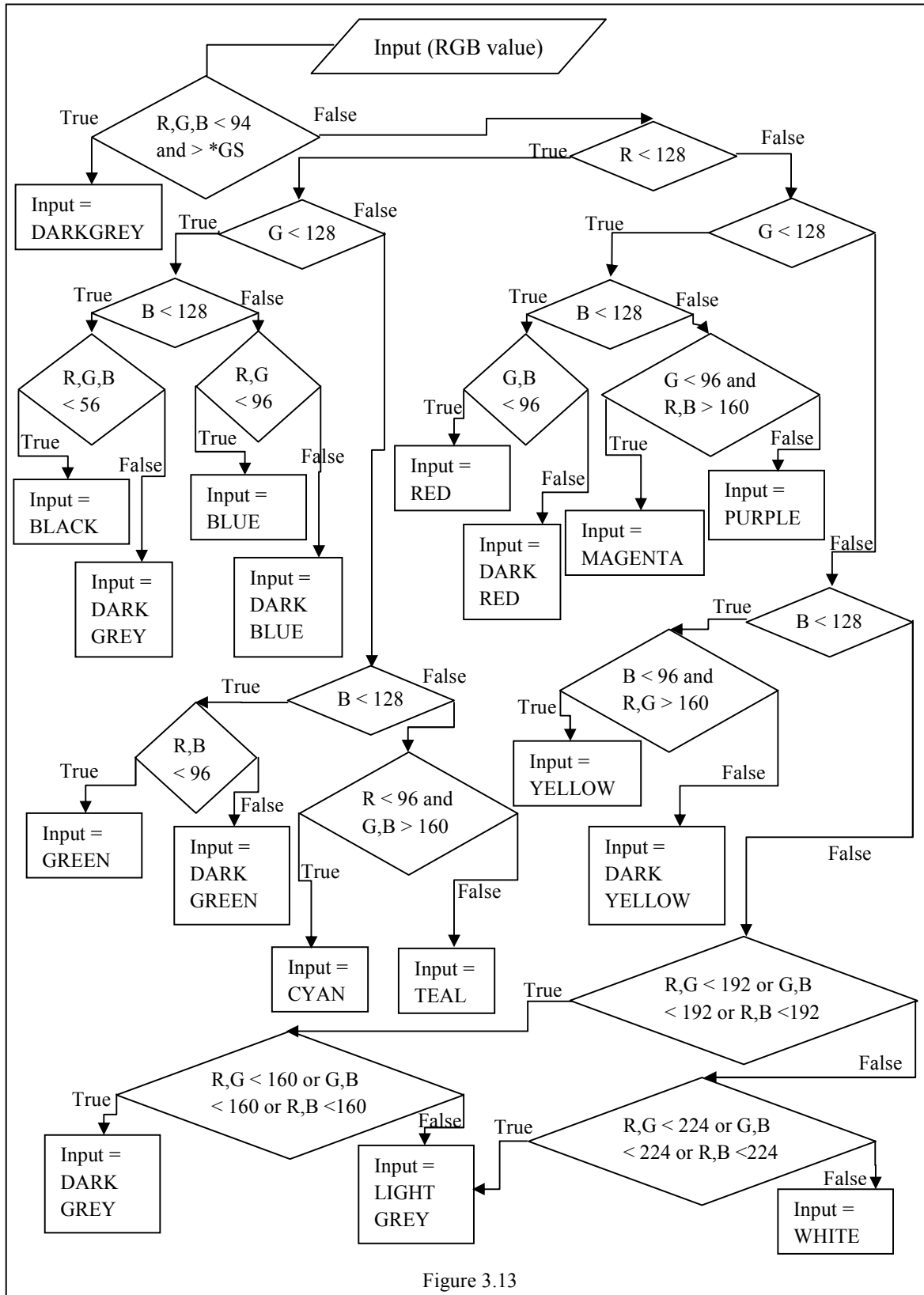
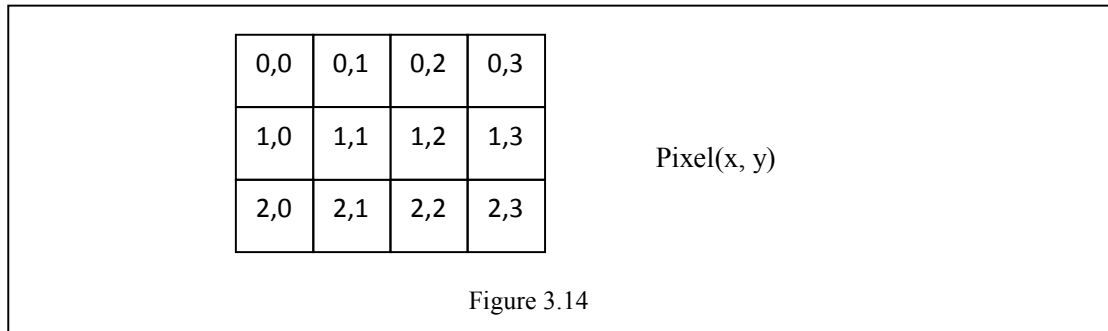


Figure 3.13

\*GS stands for Grey scale level. It is adjustable by user.

3.2.4.2 Algorithm for flushing similar pixels (*Process(2.2)*)

It is an algorithm that able to flush the similar pixels within an edge region for edge-detection. The algorithm is processing the pixels in a linear line. The following has shown the demo of the algorithm.



If process Pixel(0, 3), Pixel(1, 3) and Pixel(2, 3). The algorithm will compare the Pixel(x-1, 3) whether equals to UNDEFINED\_PIXEL or not. If is true, then Pixel(x, 3) is UNDEFINED\_PIXEL. Else, it compares Pixel(x-1, 3) whether equals to Pixel(x, 3) or not. If is true, then Pixel(x, 3) is UNDEFINED\_PIXEL.

If process Pixel(0, 0), Pixel(1, 0) and Pixel(2, 0). The algorithm will compare the Pixel(x, 0) whether equals to Pixel(x+1, 0) or not. If is true, then Pixel(x, 0) is UNDEFINED\_PIXEL.

If process Pixel(0, 1), Pixel(1, 1) and Pixel(2, 1). The algorithm will compare the Pixel(x, 1) whether equals to Pixel(x+1, 1) or not. If is true, then it compares the Pixel(x, 0) whether equals to UNDEFINED\_PIXEL or not. If is true, it will check whether the Pixel(x-1, 1) and Pixel(x+1, 1) equal to UNDEFINED\_PIXEL or not. If is true, then Pixel(x, 1) is UNDEFINED\_PIXEL.

3.2.4.3 Geometrical Pattern Classification Approach (*Process(2.3)*)

It is an algorithm that studies the geometrical pattern of an image to form and trim the edges by assigning the pixels to the correct position. In Figure 3.15 and Figure 3.16, it shows how are the patterns to be formed using the proposed geometrical pattern classification approach.

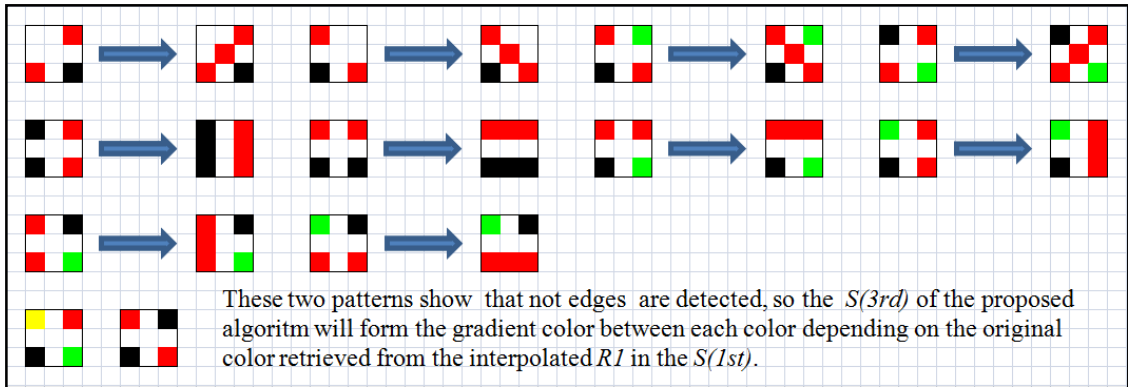


Figure 3.15

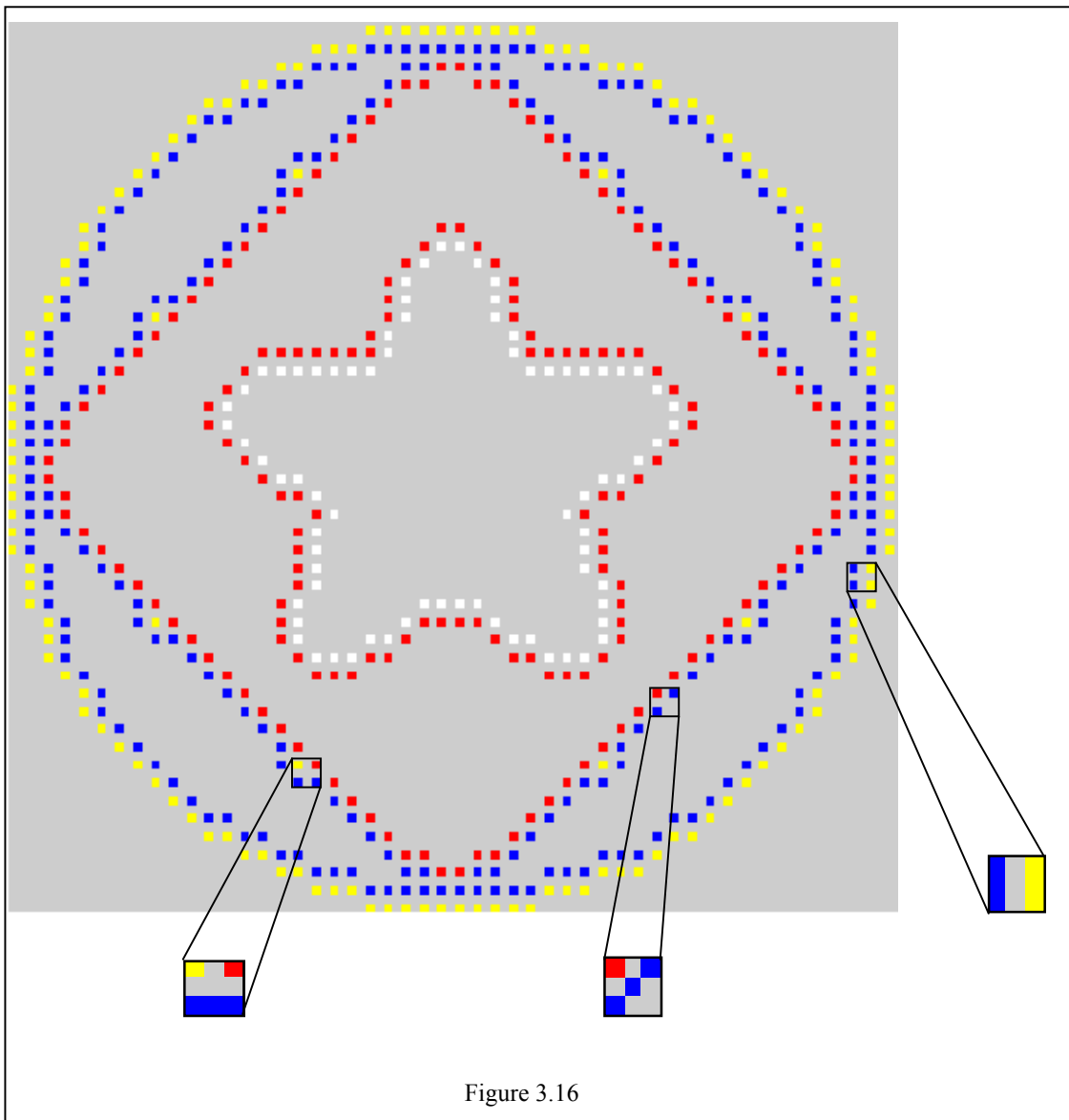
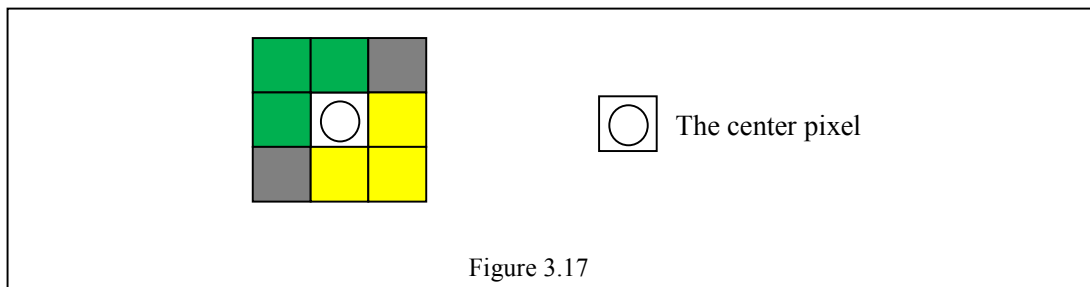


Figure 3.16

3.2.5 The Classifier (*Process(3.2)*)

There have the empty pixels (UNDEFINED\_PIXEL) occur in the between the edge regions, after the geometrical pattern classification processed. It is because the center pixel between the edge regions has the ambiguity to justify whether the colour is belongs the left colour region or right colour region. Therefore, the classifier (*Process(3.2)*) is chosen to apply the fair and reasonable classification.



In Figure 3.17, the grey colour pixels are UNDEFINED\_PIXEL. In this scenario, the classifier is based on the position of the center pixel to justify it is belongs to green or yellow. If the position X is smaller than the image width divided by 2 and Y is smaller than the image height divided by 2, then the center pixel will be assigned the average value of the three yellow values.

## Chapter 4

### 4.1 Implementation

In the implementation of the proposed algorithm, the tasks include reading the Bitmap Info Header and pixel value from Bitmap image, implement the proposed interpolation algorithm, classify using 3bit or 4bit colour palette, flush the similar pixels within the edge region for preserving edge, launch the proposed geometrical pattern classification approach, mapping the outline of the retrieved edge positions to initial interpolated image, and use the classifier to refine the final result. Pseudocode is adopted in the elaboration of each program.

#### 4.1.1 Reading Bitmap Images

As mentioned in Chapter 1, the project focused on the magnification of 24bit Bitmap images only. Initially, the prototype must be able to read the information header of Bitmap images. The following shows the format of Bitmap info header. The total size of Bitmap Info Header is 54 bytes.

Offset	Size	Description
0	2	Signature, must be 4D42 hex
2	4	Size of BMP file in bytes (unreliable)
6	2	Reserved, must be zero
8	2	Reserved, must be zero
10	4	Offset to start of image data in bytes
14	4	Size of Bitmap Info Header structure, must be 40
18	4	Image width in pixels
22	4	Image height in pixels
26	2	Number of planes in the image, must be 1
28	2	Number of bits per pixel (1, 4, 8 or 24)
30	4	Compression type (0=none, 1=RLE-8, 2=RLK-4)
34	4	Size of image data in bytes (including padding)

CHAPTER 4 IMPLEMENTATION

38	4	Horizontal resolution in pixels per meter (unreliable)
42	4	Vertical resolution in pixels per meter (unreliable)
46	4	Number of colours in image, or zero
50	4	Number of important colours, or zero

Table 4.1 the header format of 24bit Bitmap image

The pixel RGB values are stored after 54 bytes. In fact, the sequence of storing RGB values is B, G, then R. It is stored reversely. For 24bit Bitmap, each RGB values consume 8 bits (8 bits are equal to 1 byte), so 1 pixel consumes 3 bytes. Because of each RGB values have 8 bits space, so it is able to handle the value from 0 to 255. 24bit Bitmap is already a true colour photo image; it consists of 16777216 colours (256\*256\*256 or  $2^{24}$ ). However, there are many unusable properties in the Bitmap Info Header such as Offset no.2, no.38, no.42, and so on. The following is Header format of Jpeg Image.

Offset	Size	Description
0	2	Signature. Jpeg SOI marker (FFD8 hex)
2	2	Width in pixel
4	2	Height in pixel
6	1	Number of components (1 = greyscale; 2,3=RGB)
7	1	Sampling factors for component 1
8	1	Sampling factors for component 2
9	1	Sampling factors for component 3

Table 4.2 the header format of Jpeg image

Jpeg image had a great improvement in designing the header structure. It eliminated the useless properties to scale down the size of the header. Jpeg image is a compressed image that consumes lesser storage than Bitmap image, but the original content and the clear clarify after zooming are lost. Therefore, Bitmap is adopted in this project. The pseudocode of the read Bitmap function is shown in Figure 4.1 at below.

1. Read the Bitmap image as binary file.
2. Read the header information byte by byte.
3. IF image signature is not Bitmap and not 24 bit
  - a. THEN exit
4. After successfully read the image
5. FOR each rows of the image
  - a. FOR each columns of the image
    - i. Read the B, G, R value byte by byte to an array
  - b. FOR each last two columns of the image
    - i. Read the value from the padded rows to the array
6. Finally, the array contains the RGB values of the image is obtained.

Figure 4.1

#### 4.1.2 The Proposed Interpolation Algorithm

In Chapter 2 Literature Review, we realized that the larger consideration of surrounding pixels the more accurate weighted average value will be obtained. Thus, the proposed interpolation algorithm is calculating the weighted average value based on the 8 surrounding pixel values. The pseudocode is shown in Figure 4.2 at below.

1. FOR each rows until the height of the image multiple by 2
  - a. FOR each columns until the width of the image multiple by 2
    - i. Copying the value from original image to enlarged image
2. FOR each rows of the enlarged image
  - a. FOR each columns of the enlarged image
    - i. Horizontally, interpolating the value based the pixel value at both left and right side
    - ii. Vertically, interpolating the value based on the pixel value at both top and bottom side.
    - iii. Lastly, interpolating the value based on the four targeted pixels
3. Finally, the enlarge image in pixel array is obtained.

Figure 4.2

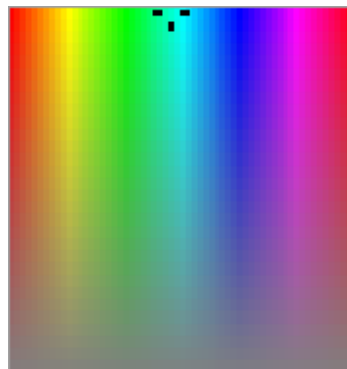


## CHAPTER 4 IMPLEMENTATION

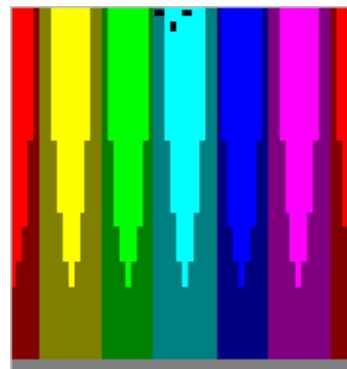
### 4.1.3 3bit and 4bit Colour Palette

As mentioned in previous section, the 3bit and 4bit colour classification is processed in a set of If-Else statements (refer to Figure 3.11 and Figure 3.13). 3bit colour classification is the fundamental method which only consists of 8 colours. However, 4bit colour classification is getting more complex than 3 bit classification. The adjustable value such as grey scale level is allowed to user to customize as their likes in order to retrieve the desire result. It has 5 values allowed user to select, such as -2, -1, 0, 1, and 2.

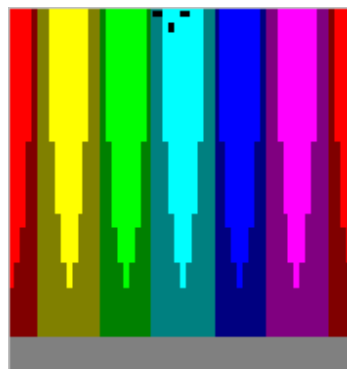
The different levels of grey scale produced results are shown at below.



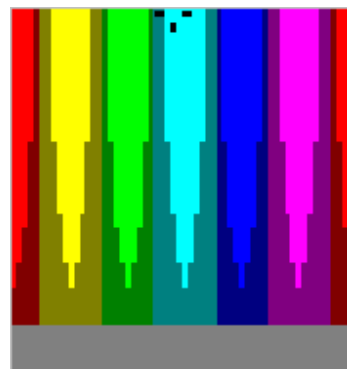
Original image



“case -2”

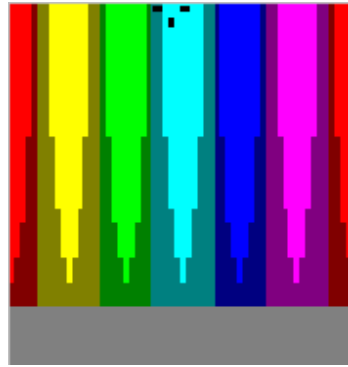


“case -1”

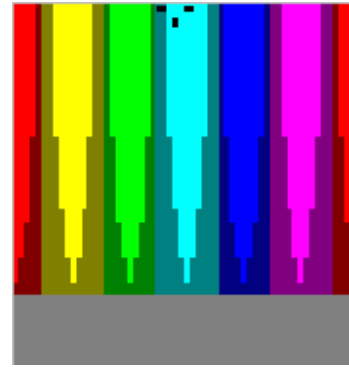


“case 0”

## CHAPTER 4 IMPLEMENTATION



“case 1”



“case 2”

Different level of grey scale is suitable for different kind of images, so that the customization function helps user to get their desired output.



The result using Microsoft Paint

Compare to Microsoft Paint, the proposed 4bit classification is more accurate than it.

### 4.1.4 The Algorithm for Preserving Edges

It is the algorithm that able to flush out the similar pixels within the specific edge region for preserving edge. The following is the pseudocode of this algorithm.

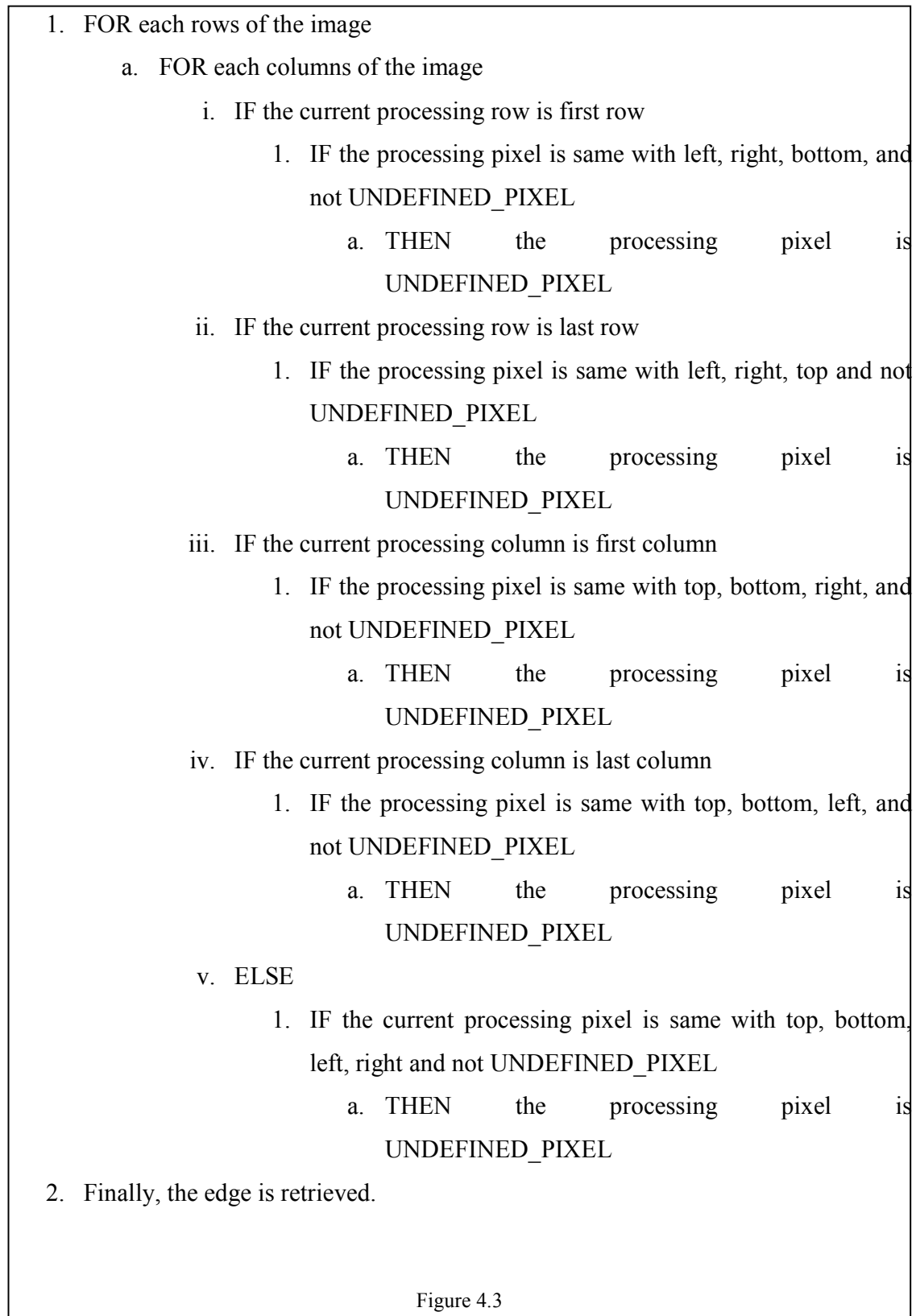


Figure 4.3

## CHAPTER 4 IMPLEMENTATION

### 4.1.5 The Proposed Geometrical Pattern Classification Approach

In the implementation of the proposed geometrical pattern classification approach, the entire predefined patterns have to be coded manually. Then, the If-Else statement will be used to compare the current processing pattern with the predefined patterns. The pseudocode is shown in Figure 4.4 at below.

1. FOR each rows of the enlarged image
  - a. FOR each columns of the enlarged image
    - i. IF the current processing pattern is same with the specific predefined pattern
      1. THEN the processing pattern will be formed based on the specific predefined pattern
2. Finally, geometrical pattern classification completed

Figure 4.4

### 4.1.6 Final Stage of the Proposed Algorithm

It consists of two parts, such as the final mapping and the classifier. The pseudocode of the final mapping is shown in Figure 4.5, and the classifier is shown in Figure 4.6.

1. FOR each rows of the enlarged image
  - a. FOR each columns of the enlarged image
    - i. IF the current processing position is on the edge based on the retrieved geometrical pattern
      1. THEN form the average colour of the two pixels at both side based on the position of the geometrical pattern
2. Finally, the smoothing effect is minimized, and then passes it to classifier.

Figure 4.5

1. FOR each rows of the enlarged image
  - a. FOR each columns of the enlarged image
    - i. IF the current processing position is the pixel between two edge regions based on the geometrical pattern
      1. IF the 8 surrounding pixel have the highest weighting
        - a. THEN the current processing pixel value is assigned the average value of those pixels which have highest weighting. The lower weighted pixels are not considered.
      2. IF two same weighting occur
        - a. THEN the average value will depend on the current processing position to justify.
2. Finally, the final result is produced.

Figure 4.6

## 4.2 Testing

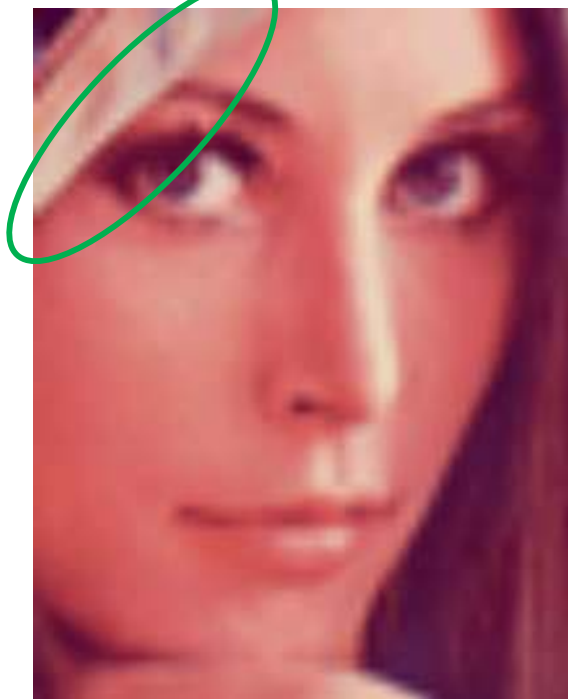
The standard test images are chosen for testing in this section. These images are downloaded from (<http://www.hlevkin.com/default.html#testimages>).



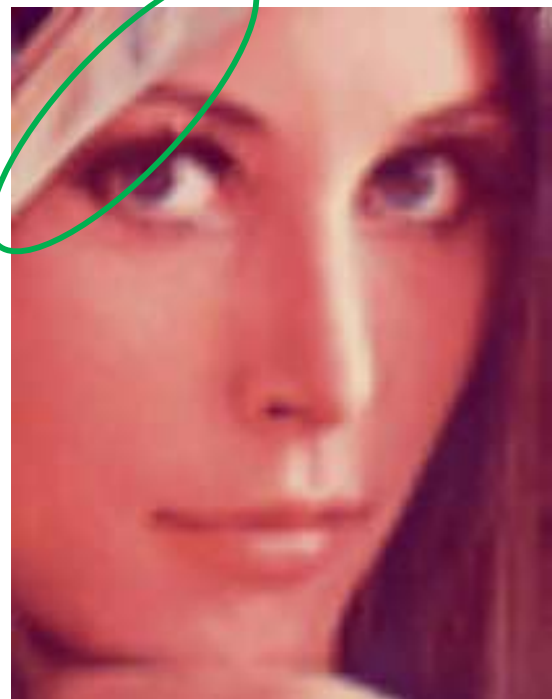
Original image (200x200)



Cropped (52x67)

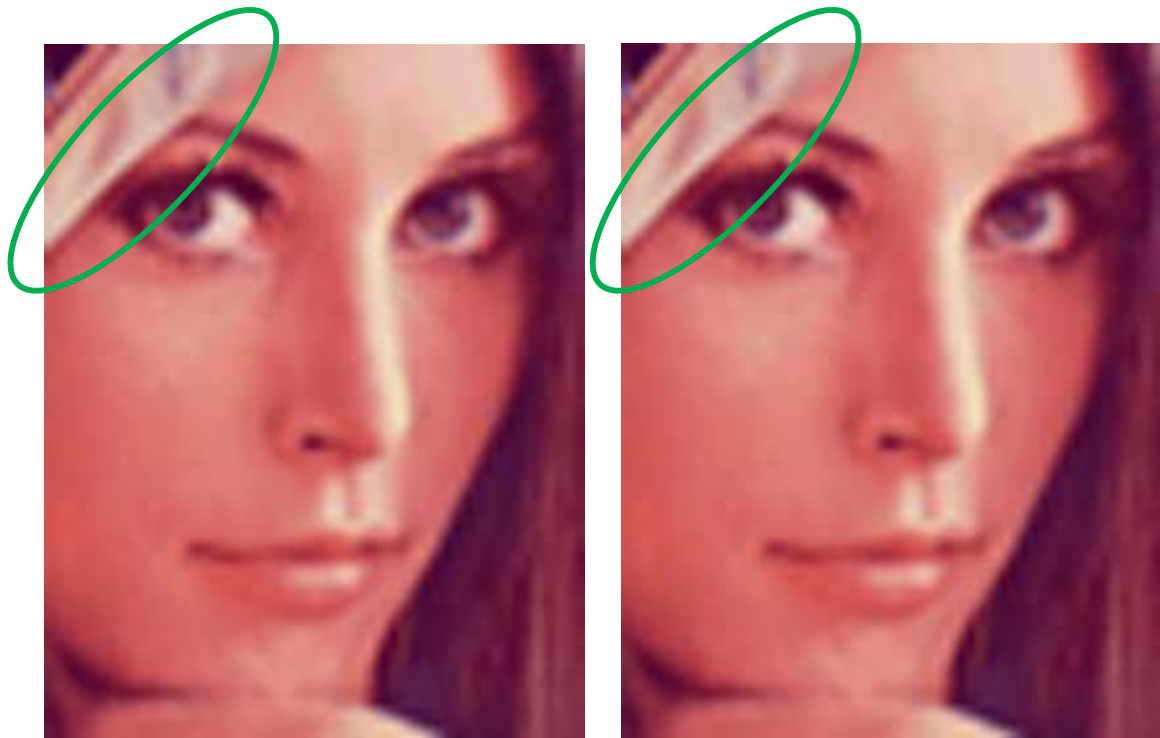


Bilinear (208x268)



The proposed algorithm (208x268)

## CHAPTER 4 IMPLEMENTATION



Bicubic (208x268)

PhotoShop (208x268)

The bicubic enlarged image is generated by using OpenCV. In fact, PhotoShop are also using Bicubic algorithm for image resizing. But, PhotoShop officially announced that the Bicubic they implemented is the modified and enhanced version of Bicubic.

Overall, these four enlarged images are still blurring. But, if you focus on the green circles on each images. It shows that the proposed algorithm is sharper than others where the jaggies along the edge are reduced.

From this testing, it shows the proposed algorithm has the strength that is the capability to reduce the jaggies where the edge is a straight line but not curve. In contrast, the weakness is not handling well to reduce the jaggies where the edge is curve shape.

\*There have many tested images are attached as appendices.

### 4.3 Verification

A Microsoft Power Point slide contains 10 categories of images was used to conduct a survey to evaluate the proposed algorithm. Each category contain the 4 magnified images that using 4 different interpolation algorithms such as the proposed algorithm, bilinear, bicubic (OpenCV), and the enhanced bicubic (PhotoShop). The positions of the 4 magnified images are randomly placed in each slide.

A form is passed to the tester for them to write down the ranking of each image. If they are not sure or cannot differentiate the differences, they are allowed to leave it blank or give "X". It is to avoid tester simply give the ranking.

There have 4 person involved in this evaluation survey. They are all students which from different Faculties. I have assisted them by giving some hints for them to easier differentiate the images during the survey.

The following table shows the total score of each method in percentage.

Methods	Total score in Percentage (%)
Proposed Algorithm	$90/140 = 64.28\%$
Bicubic (OpenCV)	$119/148 = 80.40\%$
Bilinear	$52/128 = 40.62\%$
Bicubic (PhotoShop)	$107/146 = 73.28\%$

Table 4.3 The total score of each method

From the result, it shows that the result of the proposed algorithm is average. The OpenCV Bicubic is the winner in the evaluation survey, which stronger than the PhotoShop. Bilinear gets the lowest ranking in the evaluation.



## Chapter 5

### 5.1 Discussion and Conclusion

The requirements of image magnification techniques have been seen since the users have faced the problems of low resolution images and the problem in storing larger resolution. Thus, many users have started to focus more attention on the image magnifying techniques. Moreover, the image magnification is also helping those company and enterprise in maintaining the content management system. Nowadays, those companies have concentrated on reducing the consumption of server storage. They are seeking for the efficient image magnification technique to assist them in developing it. The efficient image magnification system is able to help them to magnify the images which are scaled down when it was storing into the server. The scaled down images can help them to save the server storage, and the image magnification technique can help them to retrieve the image that still maintain the originality.

Since the proposed algorithm is still young and fresh that only able to produce the better result than the bilinear but not able to compete with bicubic. The strength of the proposed algorithm is that it performs well to reduce the jaggies and smoothing effect on the image where the edges are straight line. It is the best situation. In contrast, the worst situation is the edges are curve shape. Others method such as bicubic (OpenCV) and bicubic (PhotoShop) are also not able to do well on it. Hence, to enhance the proposed algorithm that able to handle the jaggies problem where the edges are curve shape is the future works of the project.

## BIBLIOGRAPHY

### **Bibliography**

- [1] R. Lukac, K. Martin, and K. N. Plataniotis, "Digital Camera Zooming based on Unified CFA Image Processing Steps", *IEEE Transactions on Consumer Electronics*, 50(1), 2004.
- [2] M, Sajjad. N. Khattak, N.Jafri, "Image Magnification Using Adaptive Interpolation by Pixel Level Data-Dependent Geometrical Shapes", *World Academy of Science, Engineering and Technology*, vol 25 Nov 2007.
- [3] X.Ning, T.K,Yeong, "An Image Sharpening Algorithm for High Magnification Image Zooming", *Digital Media Solutions Lab, Samsung Information Systems America, Inc.*
- [4] B. S. Morse, D. Schwartzwald, "Image Magnification Using Level-Set Reconstruction", *Department of Computer Science, Brigham Young University, 3361 TMCB.*
- [5] Y. J. Cha, S. J. Kim, "The Error-Amended Sharp Edge (EASE) Scheme for Image Zooming", *IEEE Transactions on Image Processing*, vol 16, no. 6 June 2007.
- [6] Y. J. Cha, S. J. Kim, "Edge-Forming Methods for Color Image Zooming", *IEEE Transactions on Image Processing*, vol 15, no 8 August 2006.
- [7] M. Hajizadeh, M. S. Helfroush, A. Tashk, "Improvement of Image Zooming Using Least Directional Differences based on Linear and Cubic Interpolation", *Department of Electrical Engineering, Shiraz University of Technology, Iran.*
- [8] M. Sheng, B. Su, W.Hu, "Adaptive Image Zooming Method Based on Hermite-type Interpolation Spline with Local Control Parameters". *Anqing Teachers College.*

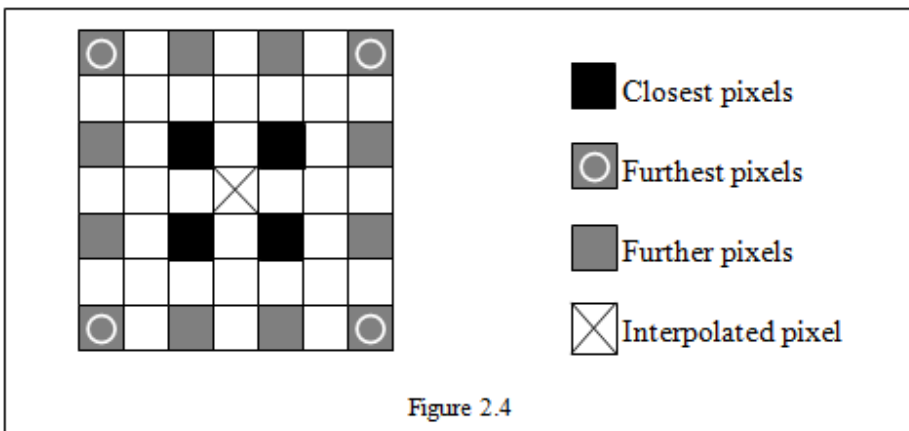
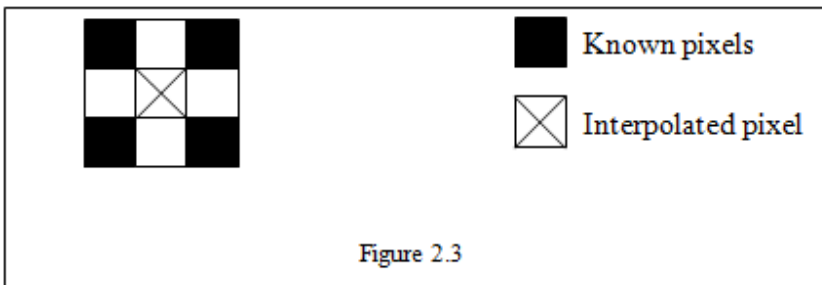
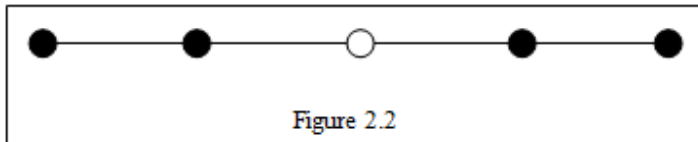
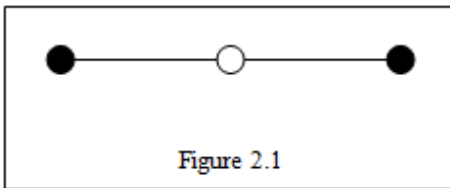
## BIBLIOGRAPHY

- [9] R. Lukac, K. N. Plataniotis, B. Smolka, A. N. Venetsanopoulos, “Vector Operators for Colour Image Zooming”, *IEEE ISIE 2005*.
- [10] L. Xin, M. T. Orchard, “New Edge-Directed interpolation”, *IEEE Transaction on Image processing, vol 10, no 10, October 2001*.
- [11] J. Allebach, P. W. Wong, “Edge-Directed Interpolation”, *School of Electrical and Computer Engineering, Purdue University*.
- [12] H. Johan, T. Nishita, “A Progressive Refinement Approach for the Image Magnification”, *The University of Tokyo*.
- [13] “Cambridge in Colour: A Learning Community of Photography”, available: <http://www.cambridgeincolour.com/tutorials/image-interpolation.htm>

***Appendix A: Figures and Tables***

## APPENDIX A: FIGURES AND TABLES

### A.1 Figures



APPENDIX A: FIGURES AND TABLES

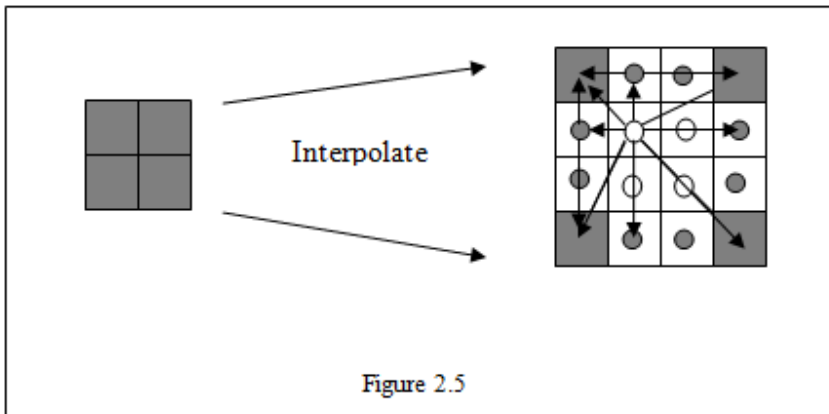


Figure 2.5

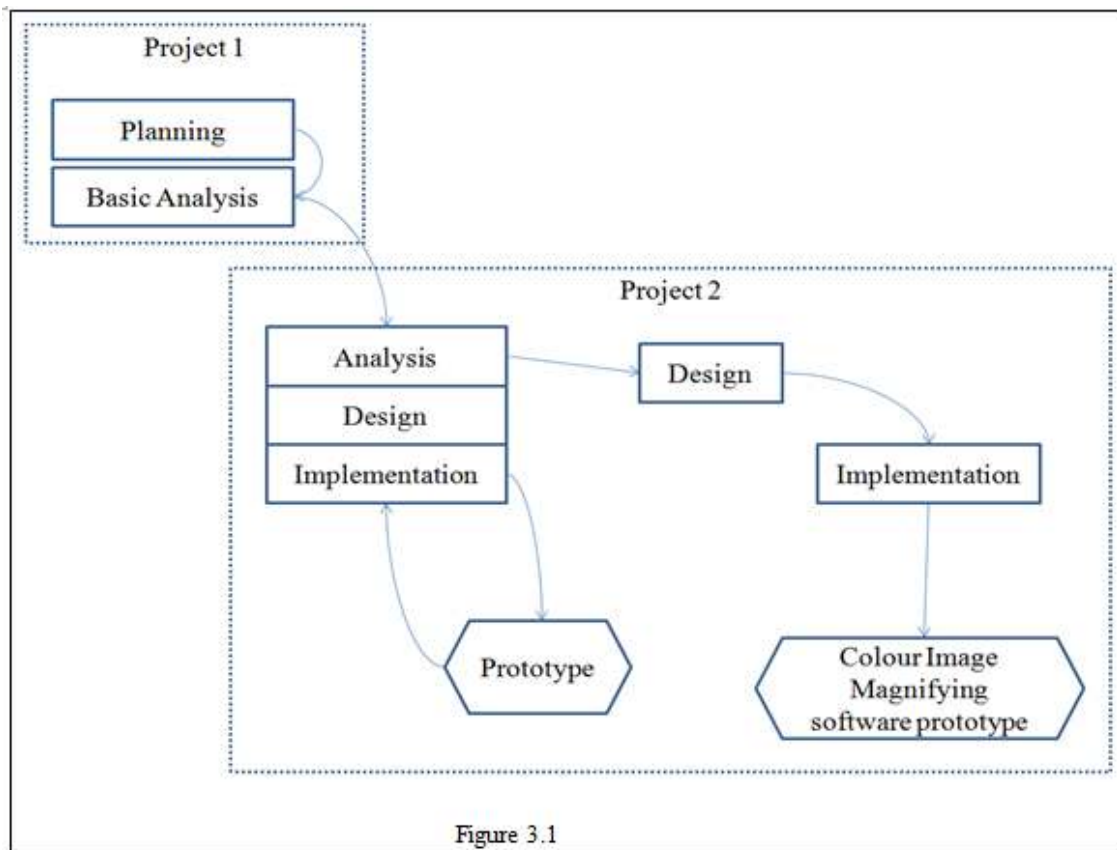
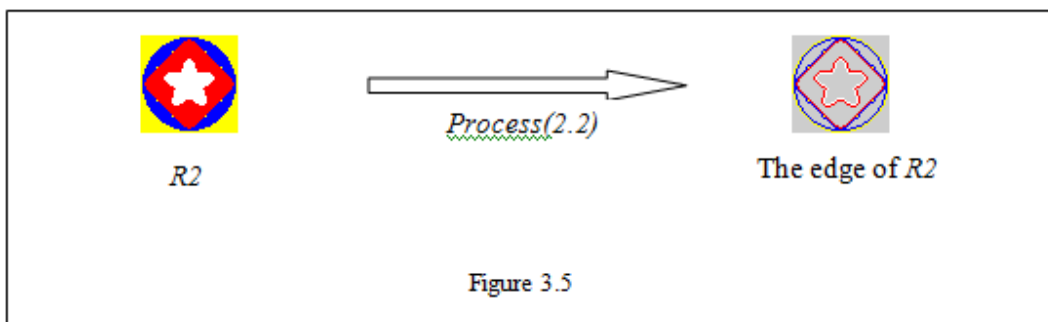
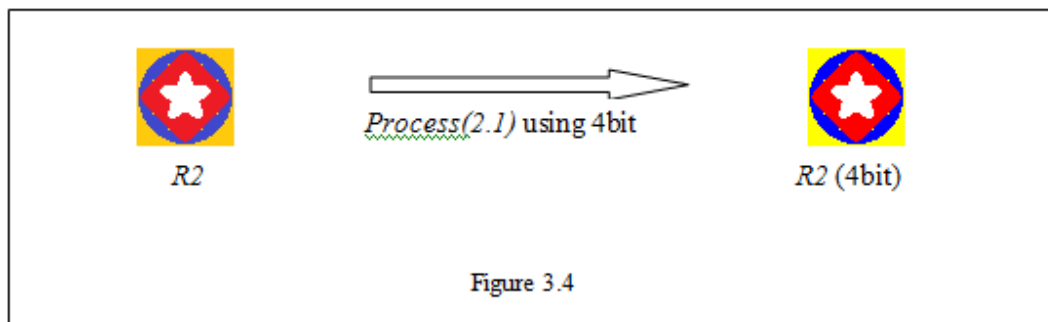
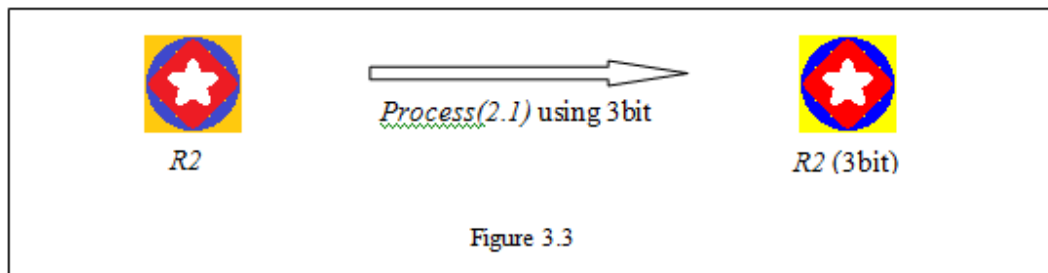
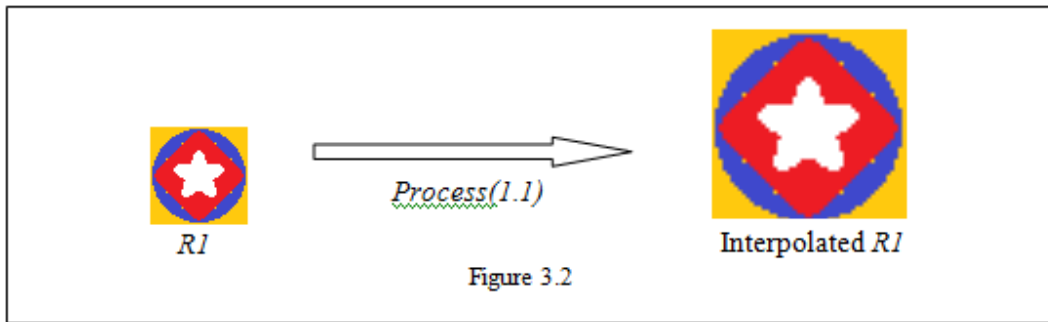
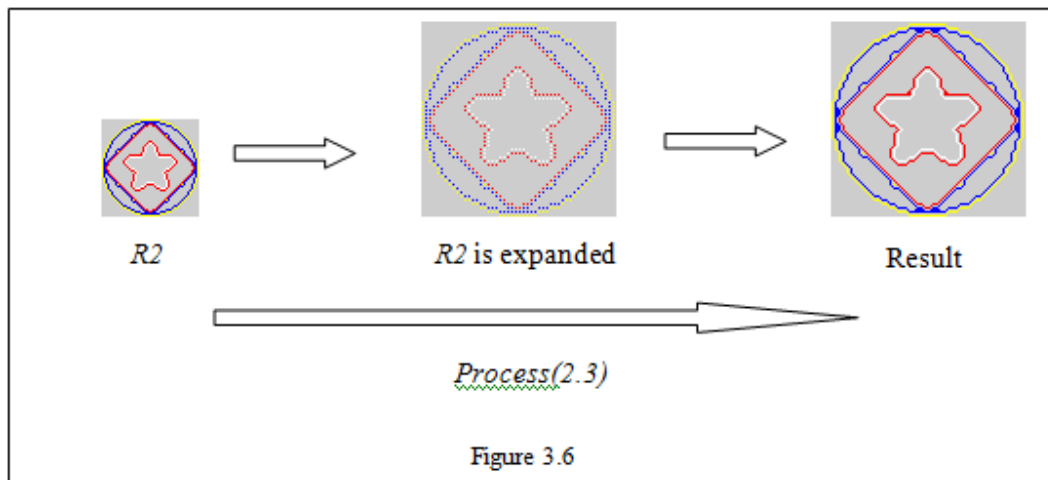


Figure 3.1

APPENDIX A: FIGURES AND TABLES

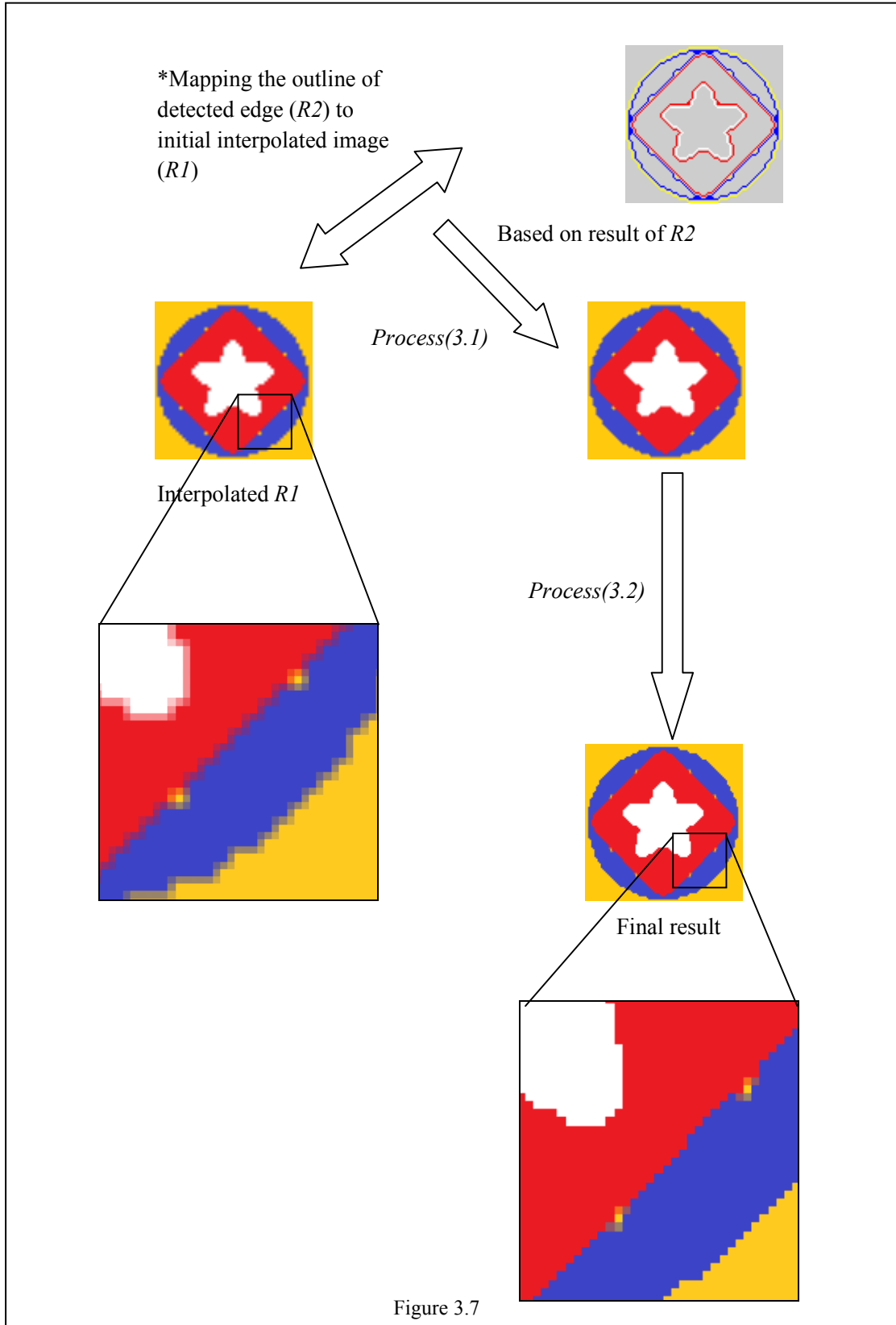


**APPENDIX A: FIGURES AND TABLES**





**APPENDIX A: FIGURES AND TABLES**



APPENDIX A: FIGURES AND TABLES

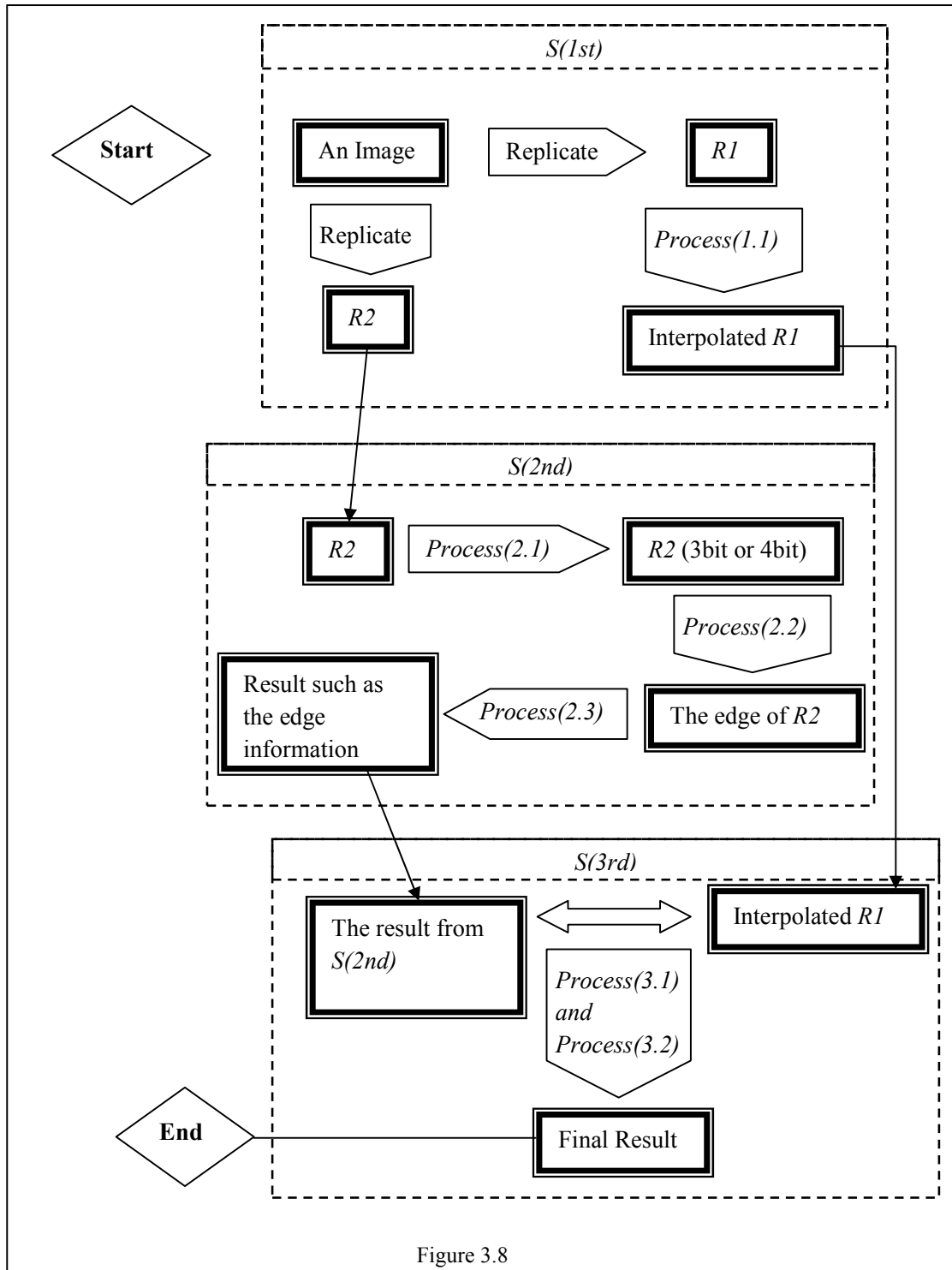
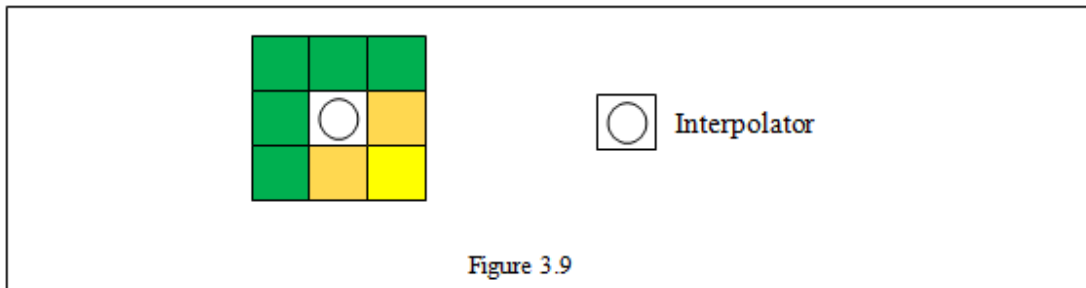


Figure 3.8

## APPENDIX A: FIGURES AND TABLES











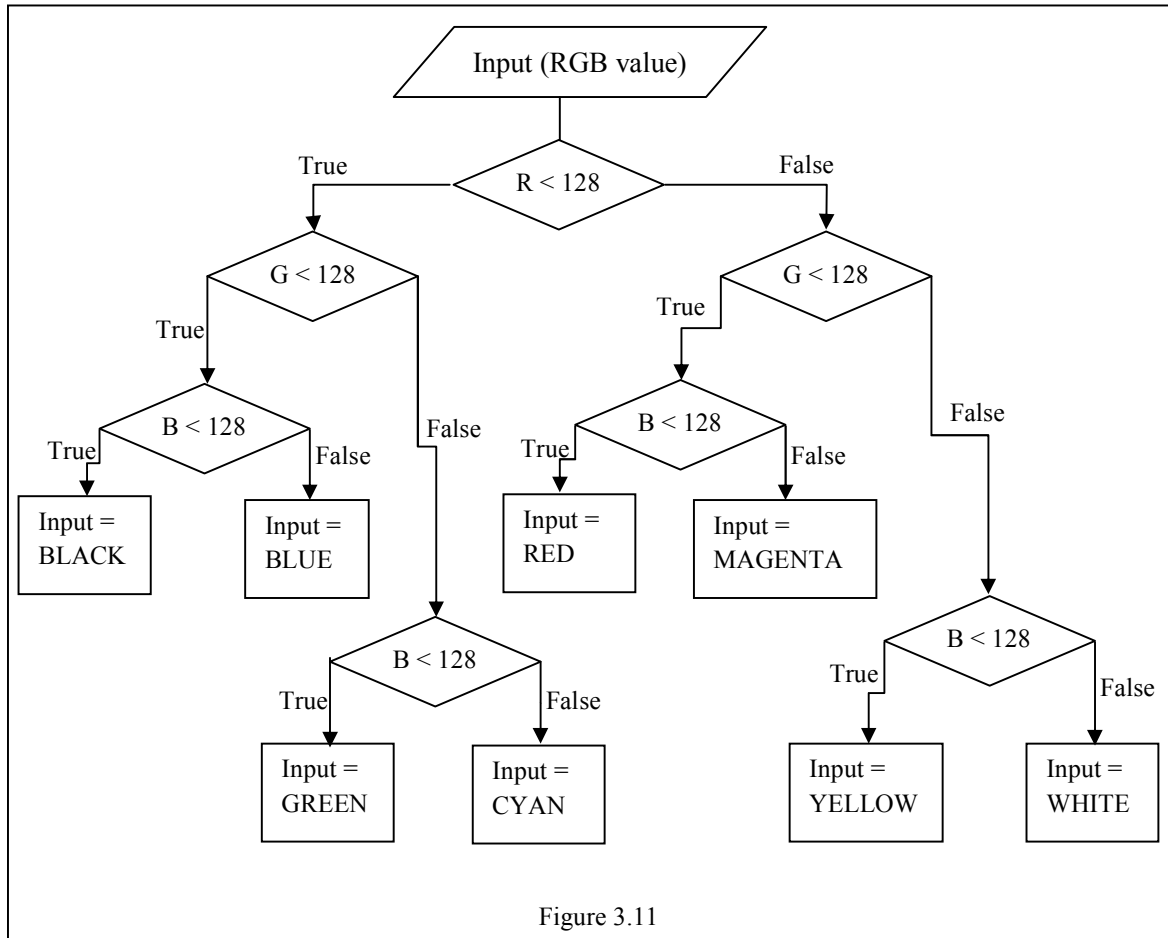
	BLACK (0, 0, 0)		RED (255, 0, 0)
	BLUE (0, 0, 255)		MAGENTA (255, 0, 255)
	GREEN (0, 255, 0)		YELLOW (255, 255, 0)
	CYAN (0, 255, 255)		BLACK (255, 255, 255)

Figure 3.10 – 3bit colour palette

APPENDIX A: FIGURES AND TABLES



## APPENDIX A: FIGURES AND TABLES



APPENDIX A: FIGURES AND TABLES

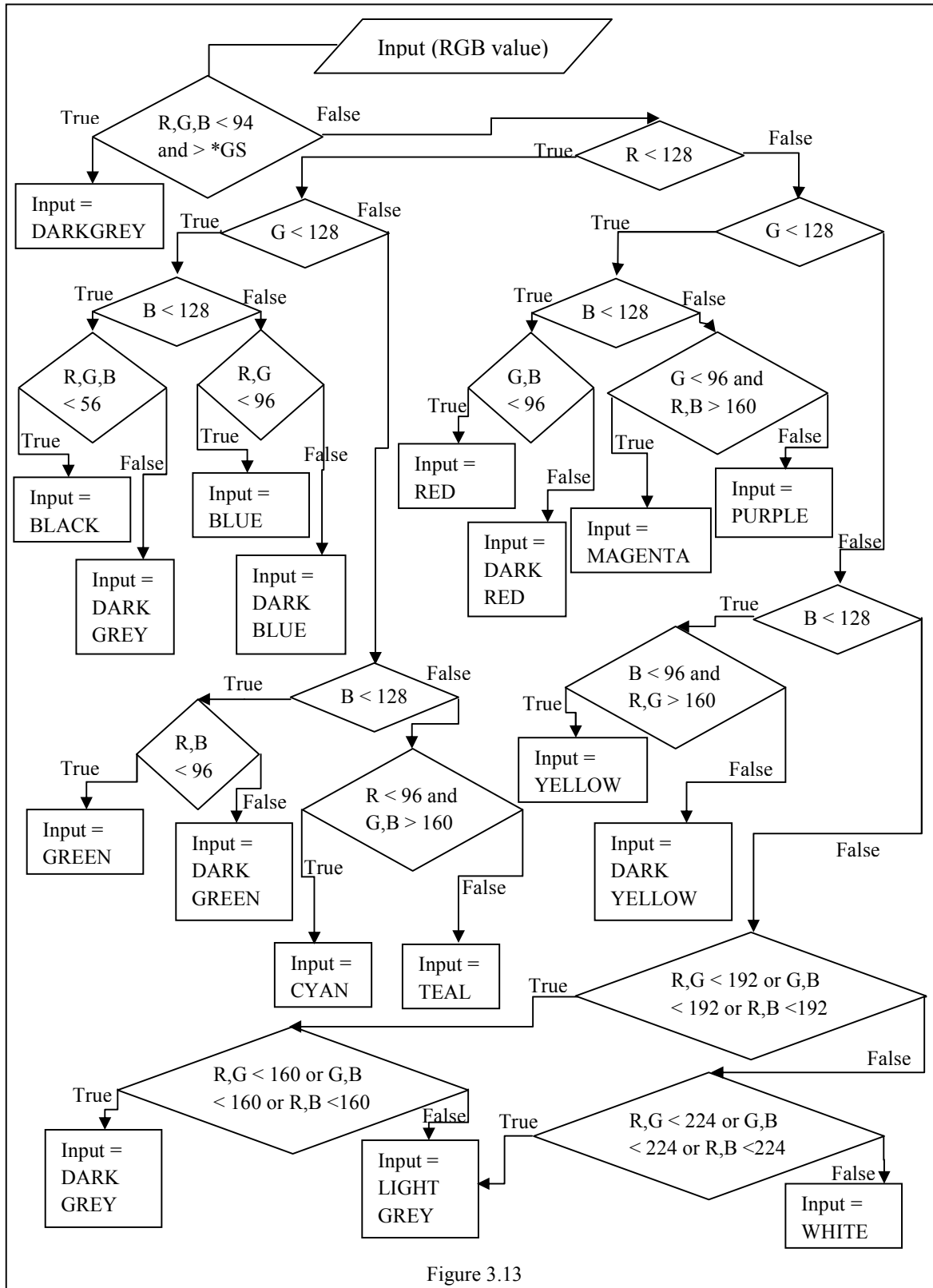


Figure 3.13

**APPENDIX A: FIGURES AND TABLES**

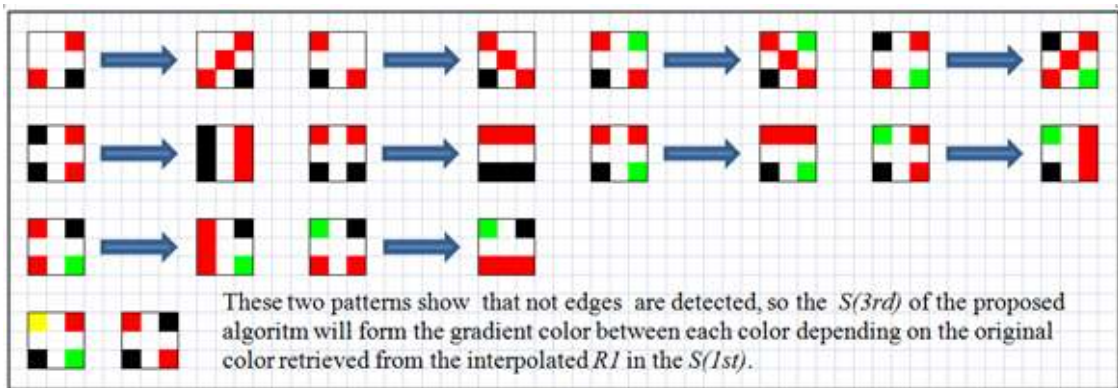
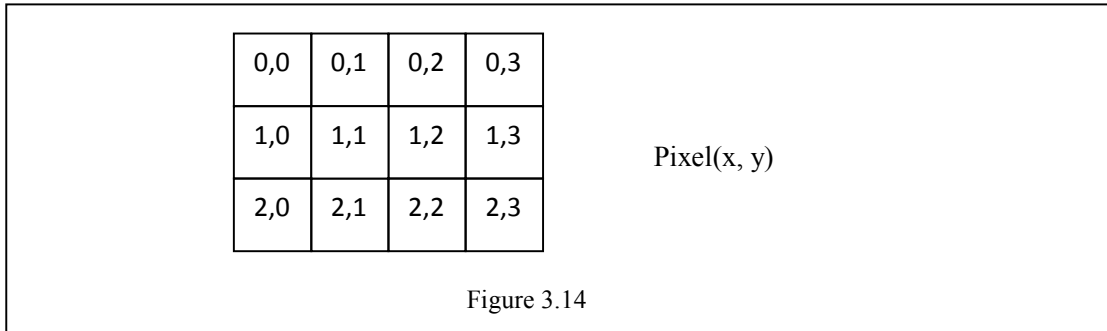
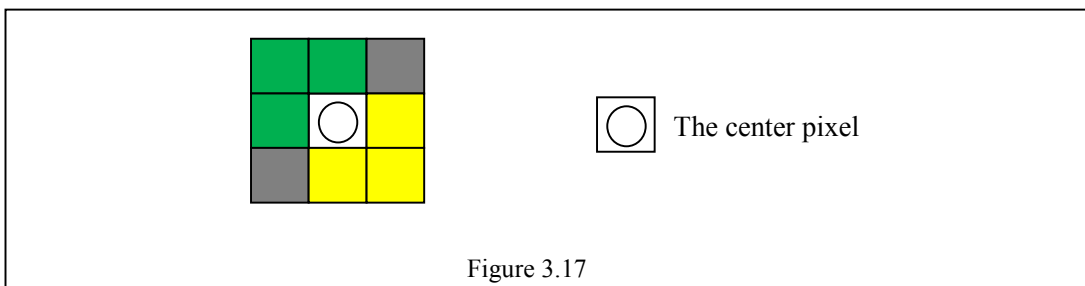
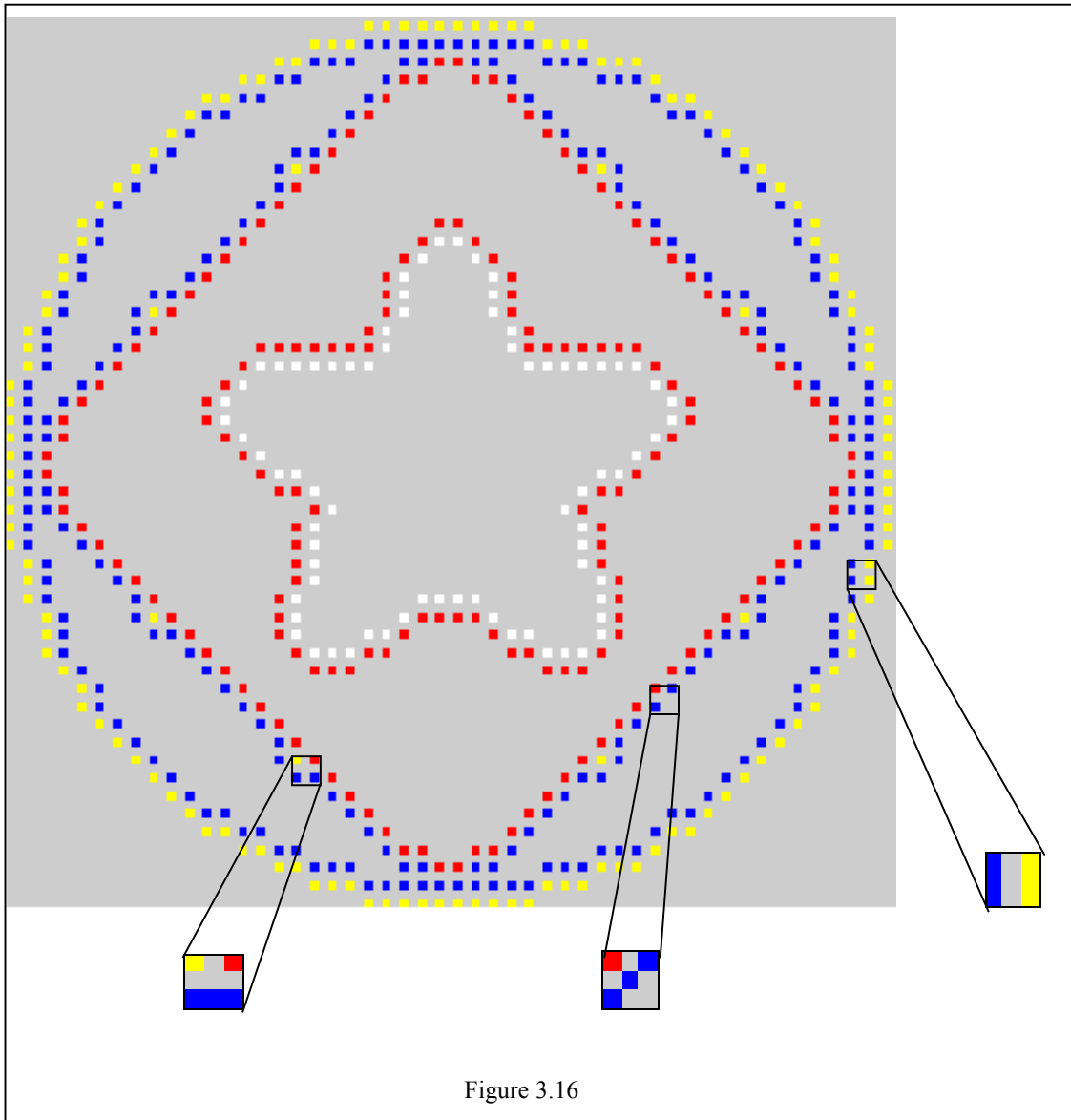


Figure 3.15

APPENDIX A: FIGURES AND TABLES





## APPENDIX A: FIGURES AND TABLES

1. Read the Bitmap image as binary file.
2. Read the header information byte by byte.
3. IF image signature is not Bitmap and not 24 bit
  - a. THEN exit
4. After successfully read the image
5. FOR each rows of the image
  - a. FOR each columns of the image
    - i. Read the B, G, R value byte by byte to an array
  - b. FOR each last two columns of the image
    - i. Read the value from the padded rows to the array
6. Finally, the array contains the RGB values of the image is obtained.

Figure 4.1

1. FOR each rows until the height of the image multiple by 2
  - a. FOR each columns until the width of the image multiple by 2
    - i. Copying the value from original image to enlarged image
2. FOR each rows of the enlarged image
  - a. FOR each columns of the enlarged image
    - i. Horizontally, interpolating the value based the pixel value at both left and right side
    - ii. Vertically, interpolating the value based on the pixel value at both top and bottom side.
    - iii. Lastly, interpolating the value based on the four targeted pixels
3. Finally, the enlarge image in pixel array is obtained.

Figure 4.2

## APPENDIX A: FIGURES AND TABLES

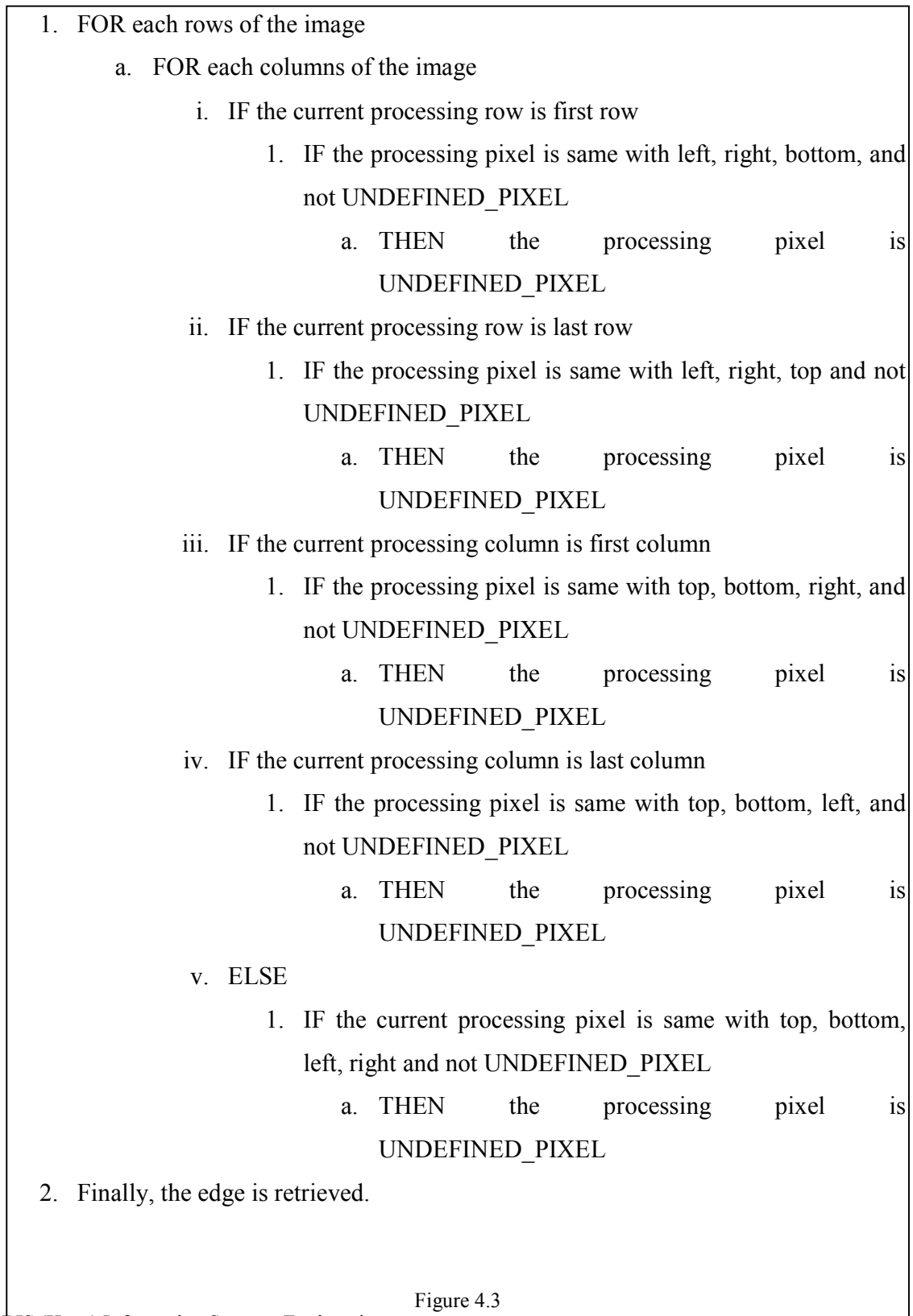


Figure 4.3

## APPENDIX A: FIGURES AND TABLES

1. FOR each rows of the enlarged image
  - a. FOR each columns of the enlarged image
    - i. IF the current processing pattern is same with the specific predefined pattern
      1. THEN the processing pattern will be formed based on the specific predefined pattern
2. Finally, geometrical pattern classification completed

Figure 4.4

1. FOR each rows of the enlarged image
  - a. FOR each columns of the enlarged image
    - i. IF the current processing position is on the edge based on the retrieved geometrical pattern
      1. THEN form the average colour of the two pixels at both side based on the position of the geometrical pattern
2. Finally, the smoothing effect is minimized, and then passes it to classifier.

Figure 4.5

## APPENDIX A: FIGURES AND TABLES

1. FOR each rows of the enlarged image
  - a. FOR each columns of the enlarged image
    - i. IF the current processing position is the pixel between two edge regions based on the geometrical pattern
      1. IF the 8 surrounding pixel have the highest weighting
        - a. THEN the current processing pixel value is assigned the average value of those pixels which have highest weighting. The lower weighted pixels are not considered.
      2. IF two same weighting occur
        - a. THEN the average value will depend on the current processing position to justify.
2. Finally, the final result is produced.

Figure 4.6

## APPENDIX A: FIGURES AND TABLES

### A.2 Tables

Offset	Size	Description
0	2	Signature, must be 4D42 hex
2	4	Size of BMP file in bytes (unreliable)
6	2	Reserved, must be zero
8	2	Reserved, must be zero
10	4	Offset to start of image data in bytes
14	4	Size of Bitmap Info Header structure, must be 40
18	4	Image width in pixels
22	4	Image height in pixels
26	2	Number of planes in the image, must be 1
28	2	Number of bits per pixel (1, 4, 8 or 24)
30	4	Compression type (0=none, 1=RLE-8, 2=RLK-4)
34	4	Size of image data in bytes (including padding)
38	4	Horizontal resolution in pixels per meter (unreliable)
42	4	Vertical resolution in pixels per meter (unreliable)
46	4	Number of colours in image, or zero
50	4	Number of important colours, or zero

Table 4.1 the header format of 24bit Bitmap image

## APPENDIX A: FIGURES AND TABLES

Offset	Size	Description
0	2	Signature. Jpeg SOI marker (FFD8 hex)
2	2	Width in pixel
4	2	Height in pixel
6	1	Number of components (1 = greyscale; 2,3=RGB)
7	1	Sampling factors for component 1
8	1	Sampling factors for component 2
9	1	Sampling factors for component 3

Table 4.2 the header format of Jpeg image

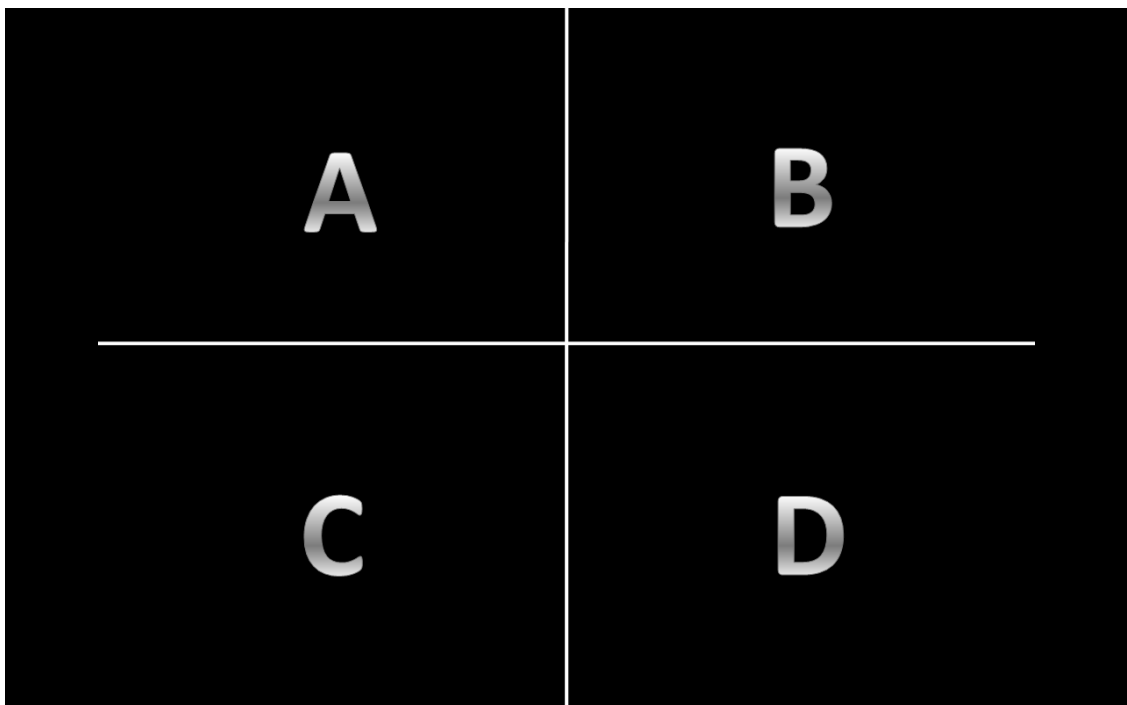
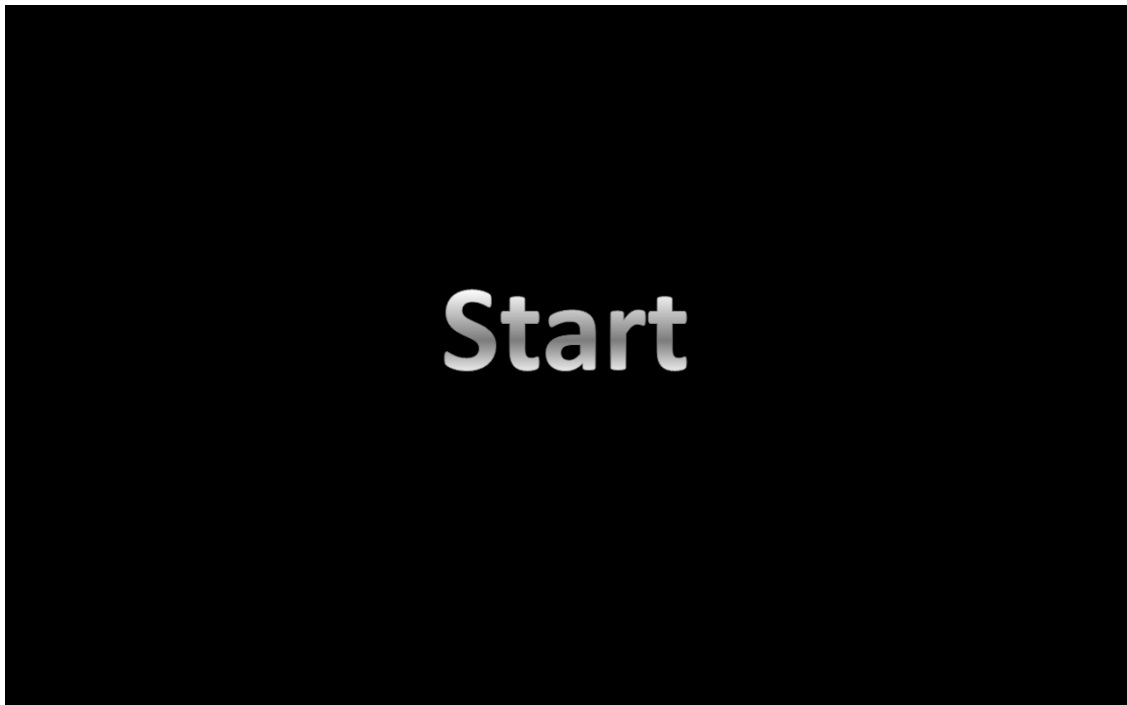
Methods	Total score in Percentage(%)
Proposed Algorithm	$90/140 = 64.28\%$
Bicubic (OpenCV)	$119/148 = 80.40\%$
Bilinear	$52/128 = 40.62\%$
Bicubic (PhotoShop)	$107/146 = 73.28\%$

Table 4.3 The total score of each method

***Appendix B: Survey Form and Slide***

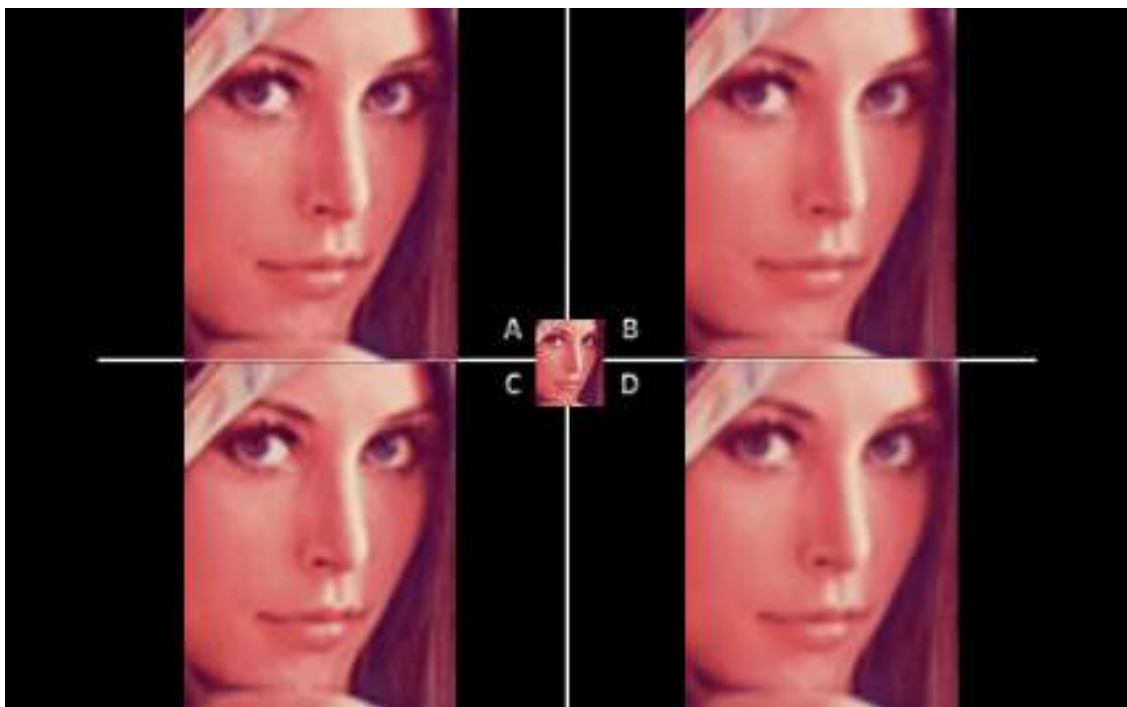
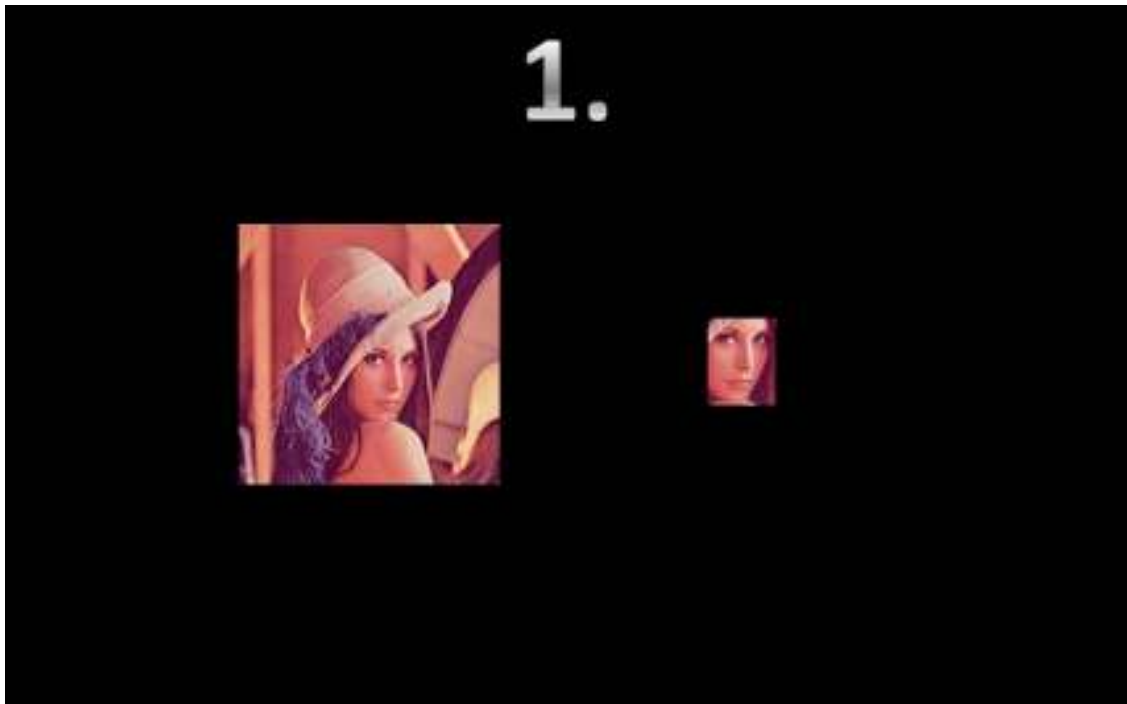
## APPENDIX B: SURVEY FORM AND SLIDE

### B.1 Slide

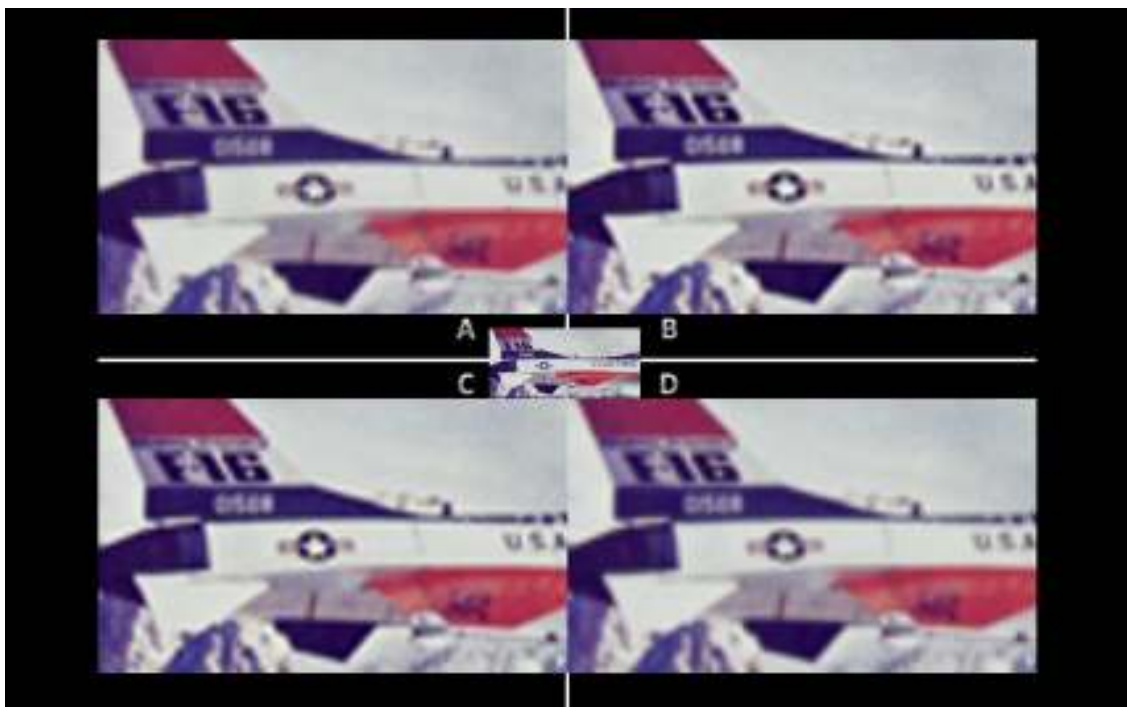
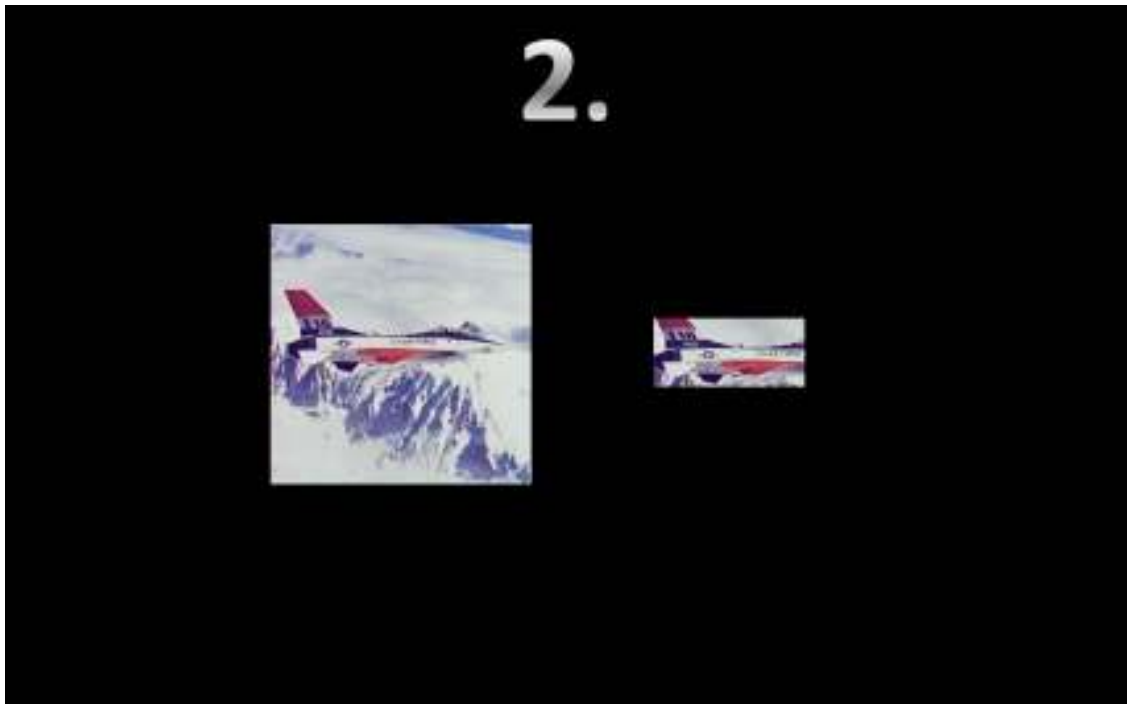




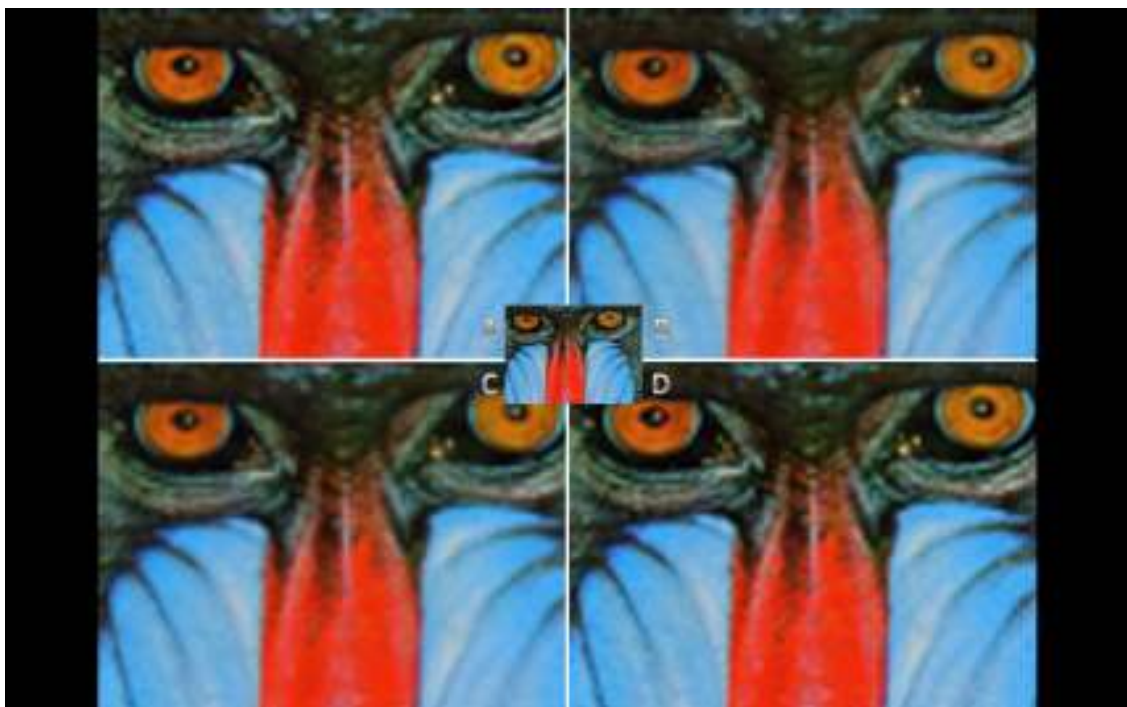
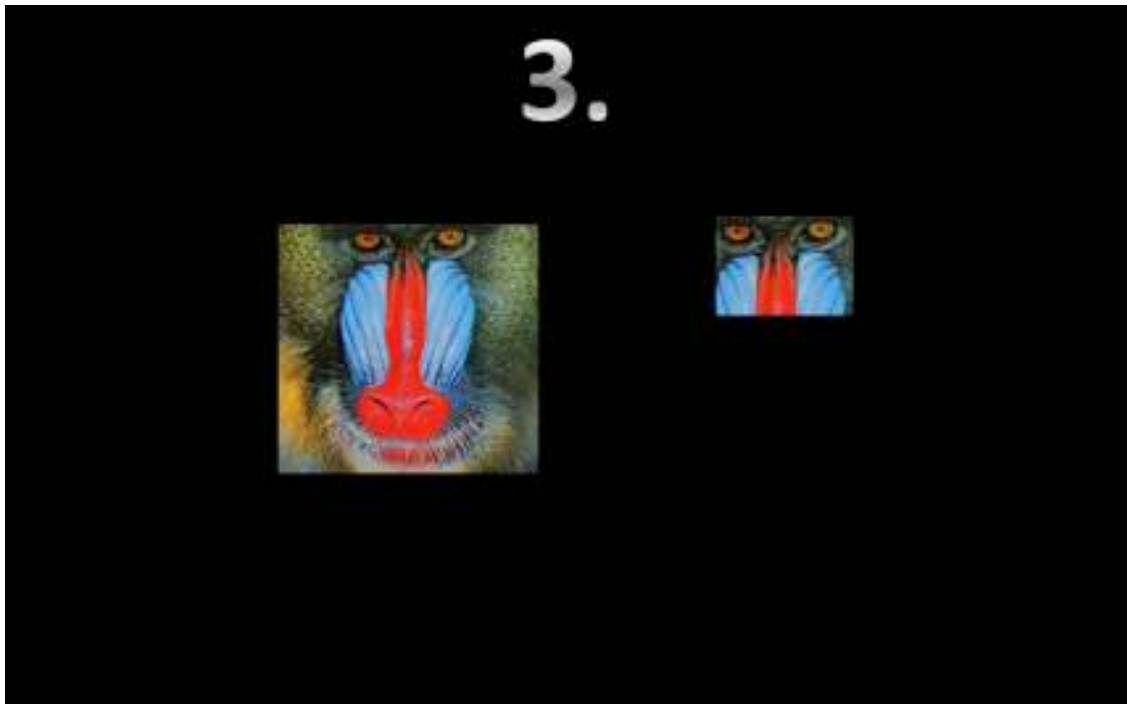
**APPENDIX B: SURVEY FORM AND SLIDE**



APPENDIX B: SURVEY FORM AND SLIDE



APPENDIX B: SURVEY FORM AND SLIDE



**APPENDIX B: SURVEY FORM AND SLIDE**

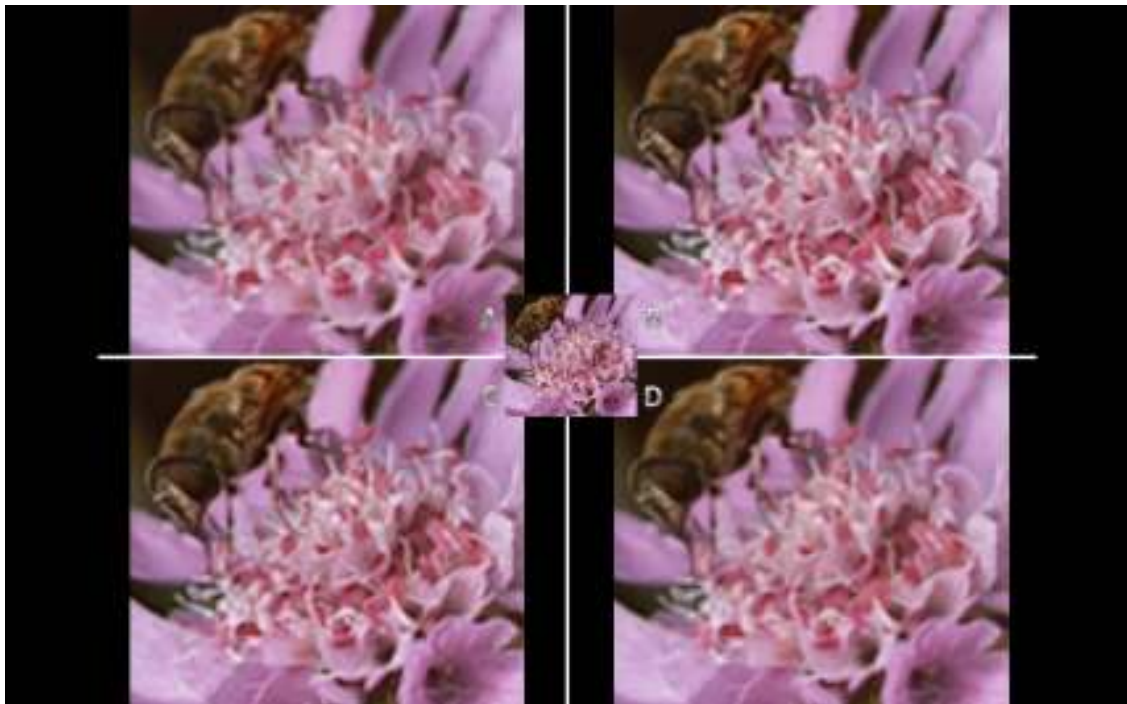
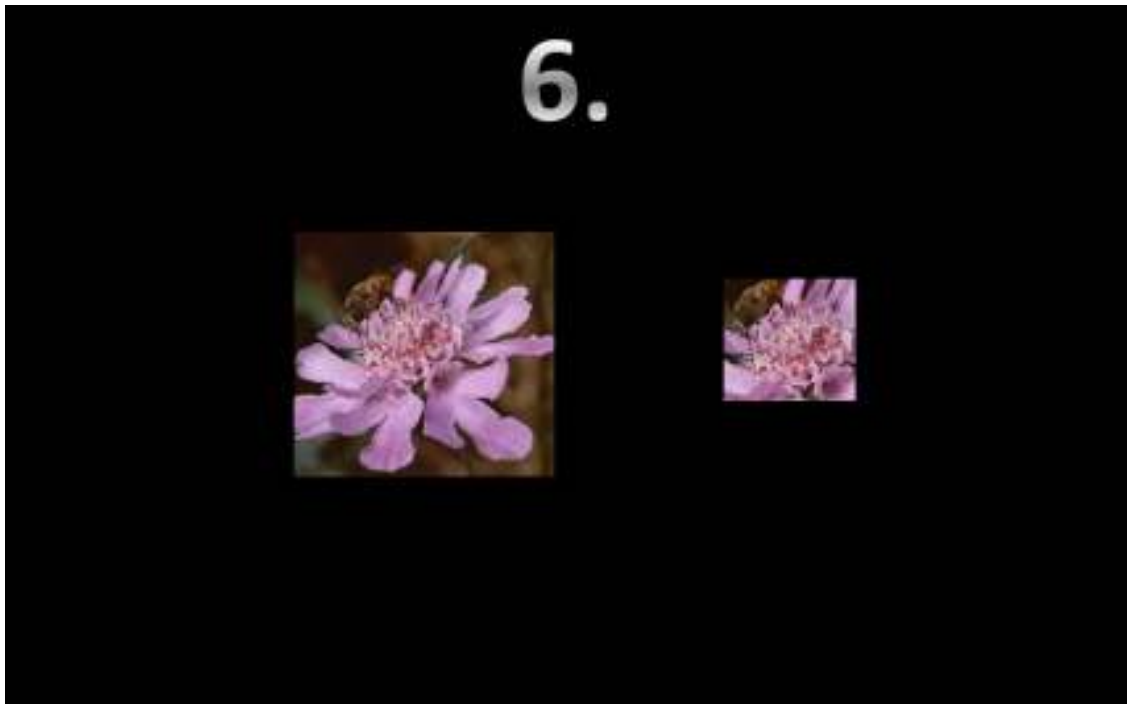


APPENDIX B: SURVEY FORM AND SLIDE

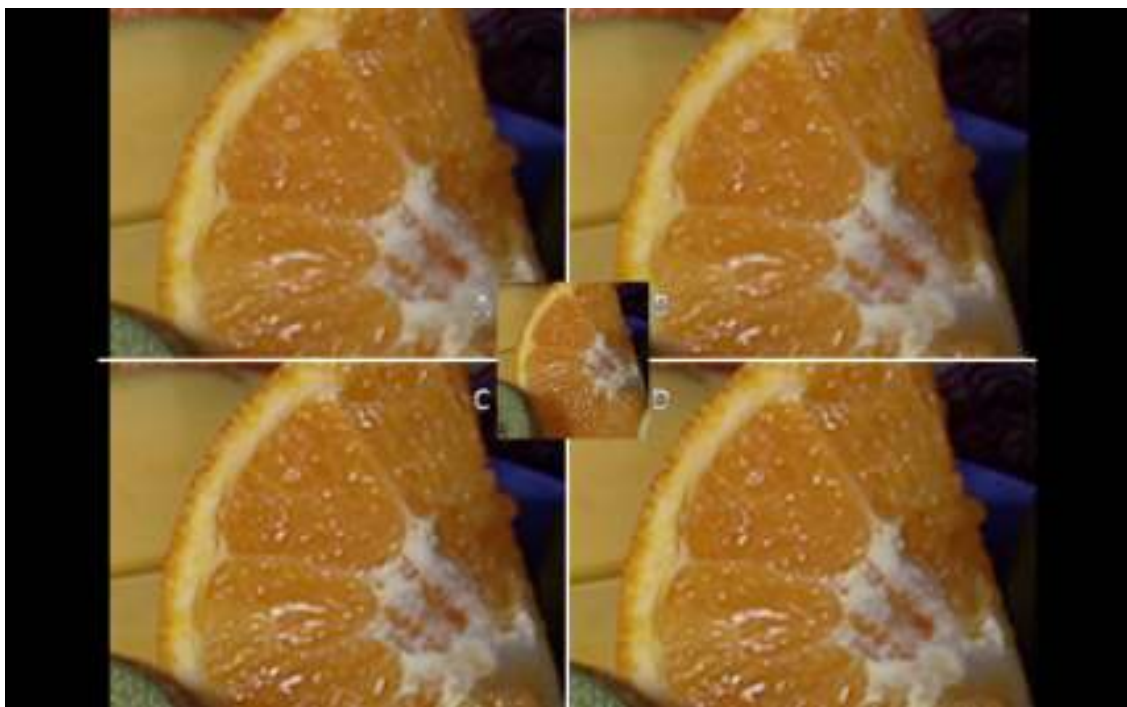
5.



**APPENDIX B: SURVEY FORM AND SLIDE**



**APPENDIX B: SURVEY FORM AND SLIDE**



APPENDIX B: SURVEY FORM AND SLIDE

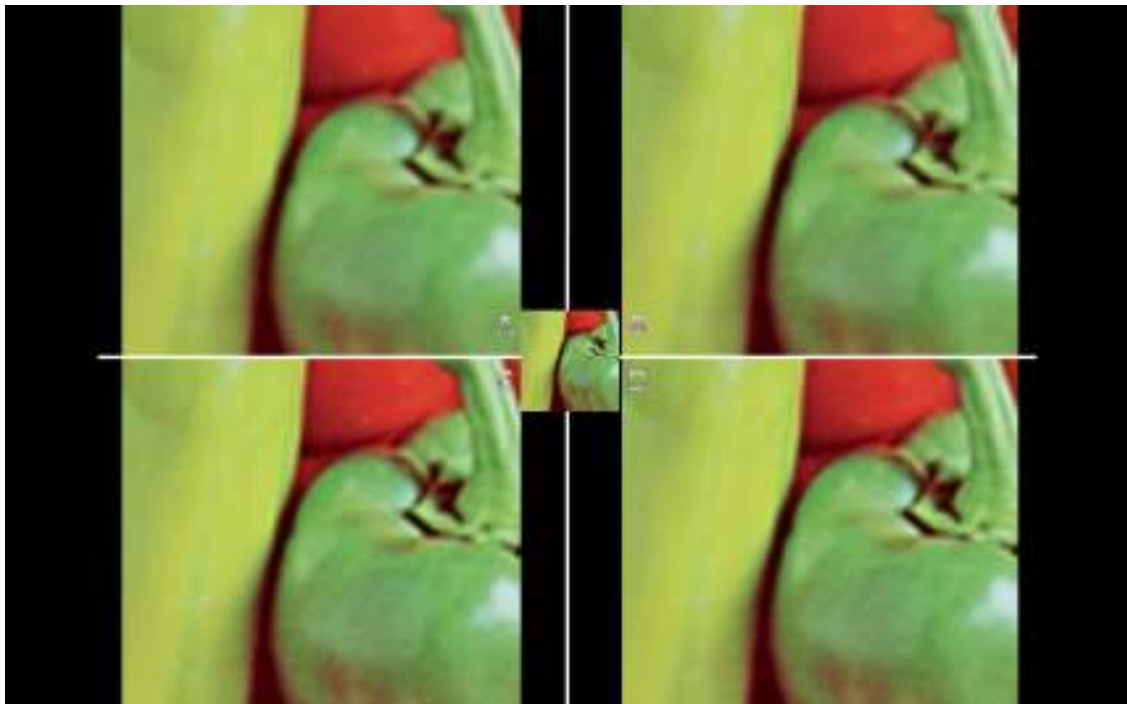




**APPENDIX B: SURVEY FORM AND SLIDE**



**APPENDIX B: SURVEY FORM AND SLIDE**



## APPENDIX B: SURVEY FORM AND SLIDE

### B.2 Survey Form

Name: \_\_\_\_\_ (Student / Lecturer)

Eg,

A	4	B	3
C	2	D	1

Eg,

A	4	B	X
C	X	D	1

\*If you cannot differentiate B ,C

1.

A	B
C	D

2.

A	B
C	D

3.

A	B
C	D

4.

A	B
C	D

5.

A	B
C	D

6.

A	B
C	D

7.

A	B
C	D

8.

A	B
C	D

**APPENDIX B: SURVEY FORM AND SLIDE**

9.

A	B
C	D

10.

A	B
C	D

## APPENDIX B: SURVEY FORM AND SLIDE

### B.3 Survey Answer

\*Answer

1.	A-Bicubic	B-Proposed
	C-PhotoShop	D-Bilinear
2.	A-Proposed	B-Bicubic
	C-PhotoShop	D-Bilinear
3.	A-PhotoShop	B-Proposed
	C-Bilinear	D_Bicubic
4.	A-Bilinear	B-Bicubic
	C-Proposed	D-PhotoShop
5.	A-Proposed	B-Bicubic
	C-PhotoShop	D-Bilinear
6.	A-Bilinear	B-PhotoShop
	C-Bicubic	D-Proposed
7.	A-Bilinear	B-Proposed
	C-Bicubic	D-PhotoShop
8.	A-PhotoShop	B-Bicubic
	C-Proposed	D-Bilinear
9.	A-Proposed	B-PhotoShop
	C-Bilinear	D-Bicubic
10.	A-Bilinear	B-Proposed
	C-PhotoShop	D-Bicubic