

DEVELOPMENT OF COMPUTER AIDED THREE-DIMENSIONAL
GEOMETRIC MODELER BASED ON DESIGNER'S IDEA IN TWO-
DIMENSIONAL

(PEMBANGUNAN PEMODEL GEOMETRI TIGA DIMENSI
BERBANTUKAN KOMPUTER BERDASARKAN IDEA
PEREKABENTUK DALAM DUA DIMENSI)

HABIBOLLAH BIN HARON

RESEARCH VOTE NO:
74225

Jabatan Pemodelan dan Pengkomputeran Industri
Fakulti Sains Komputer & Sistem Maklumat
Universiti Teknologi Malaysia

ACKNOWLEDGEMENT

In the name of Allah the Most Gracious and the Most Merciful, I thank Thee with all my heart for granting Thy Servant immeasurable help during the course of this study and peace be upon him, Muhammad, the last messenger of Allah.

I would like to express my gratitude to my research officer Syarul Haniz Subri, assistant researcher namely Faizal, Nasim, Fairuz, Zaki, and Edinorman for your assistance in completing this work.

I am very grateful to Associate Professor Dr Dzulkifli Mohamed and Associate Professor Dr Siti Mariyam Shamsuddin for your comment and ideas, and Professor Dr Ralph Martin for suggesting this research topic.

My sincere thank also goes to UTM, RMC and MOSTI for granting me this grant. I would also like to thank Zahari Supene, the technician, for his help in ensuring my PC always clean from virus and always in good condition.

ABSTRACT

In designing an engineering product during the conceptual stage, freehand sketch plays an important role in conveying engineers' ideas. Solutions on the problems of representing, interpreting and reconstructing the sketch have been of research interest for several decades. The aim of this study is to propose a framework of geometric modeler in interpreting a single line drawing sketch and reconstruct the sketch to produce a two dimensional model. The methodology used in this study is based on the framework of a geometric modeler that contains two phases. In the first phase, two algorithms were proposed. The first is enhanced thinning algorithm that is used to thin the sketch while maintaining its topology and basic shape. The second is the conversion algorithm to convert the irregular line drawing into vertex chain code (VCC) and into modified Freeman chain code (FCC). These algorithm are used to produce chain code series that represented a thinned binary image of the sketch. In the second phase, a new computational corner detection algorithm of FCC and neural network classifier for corner detection of VCC are proposed to analyse the chain code series to produce two-dimensional geometry entities of the sketch in which the irregular line drawing is transformed into regular line drawing. This study has proposed better solutions in terms of an enhanced thinning algorithm, a modified Freeman chain code and Vertex chain code algorithm as picture description language, and a new computational algorithm and neural network classifier for corner detection. Performance comparison of the corner detection algorithms shows that neural network classifier is performed better while FCC is easier in representing thinned binary image.

ABSTRAK

Semasa peringkat konsepsualisasi dalam merekabentuk satu produk kejuruteraan, lukisan lakaran tangan memainkan peranan penting dalam menzahirkan idea jurutera. Penyelesaian kepada masalah mewakili, menterjemah dan membina semula lakaran telah menjadi bidang penyelidikan sekian lama. Matlamat kajian ini ialah mencadangkan mencadangkan kerangka kerja sebuah pemodel geometri menterjemah dan membina semula sebuah lakaran tunggal, dan seterusnya untuk menghasilkan model dua dimensi. Metodologi yang digunakan dalam kajian ialah berdasarkan rangka kerja penterjemah lakaran yang melibatkan tiga fasa. Dalam fasa pertama, dua algoritma dicadangkan. Pertama ialah ialah pembaikan algoritma penipisan yang digunakan untuk menipiskan lakaran dengan mengekalkan falsafah memelihara topologi dan bentuk asasnya. Kedua ialah penciptan algoritma menukar lukisan tak sekata kepada kod rangkaian puncak dan pengubahsuaian algoritma kod rantai Freeman. Algoritma-algoritma ini digunakan untuk menghasilkan siri kod rantai bagi mewakili imej binari ternipis lakaran. Dalam fasa kedua, algoritma pengesanan selekoh pengiraan baru bagi kod rangkaian Freeman dan pengelas rangkaian neural bagi mengesan simpang berasaskan kod rangkaian puncak telah dicadangkan bagi menganalisa siri kod rantai untuk menghasilkan entiti geometri dua dimensi di mana lukisan garisan tak sekata ditukar kepada lukisan garisan sekata. Kajian ini telah mencadangkan penyelesaian yang lebih baik dari segi pembaikan algoritma penipisan, pembaikan kod rantai Freeman and kod rangkaian puncak sebagai bahasa penghuraian gambar, dan algoritma pengesanan selekoh pengiraan baru dan pengelas rangkaian neural bagi pengesanan simpang. Perbandingan pencapaian bagi pengesanan simpang menunjukkan bahawa rangkaian neural adalah lebih baik manakala kod rangkaian Freeman lebih baik sebagai perwakilan data imej binari.

CONTENTS

CHAPTER	SUBJECTS	PAGE
	TITLE	i
	ACKNOWLEDGEMENT	ii
	ABSTRACT	iii
	ABSTRAK	iv
	TABLE OF CONTENTS	v
	LIST OF TABLES	ix
	LIST OF FIGURES	x
	LIST OF ABBREVIATIONS	xiii
	LIST OF APPENDIXES	xiv
I	INTRODUCTION	1
1.1	Definition of Geometric Modeler	1
1.2	Motivation	1
1.3	Objectives	4
1.4	Scope	5
1.5	Assumptions	6
1.6	Report Outline	6
1.7	Contributions of the Research	8

II	LITERATURE REVIEW	11
2.1	Line Drawing Sources	11
2.1.1	Origami World	13
2.1.2	Engineering Sketch	14
2.1.3	Impossible Object	14
2.1.4	Wireframe Object	15
2.2	Boundary Representation	15
2.3	A Geometric Modeler	17
2.3.1	Sketching Process	17
2.3.2	The Sketch Interface	18
2.3.3	The Modeler Engine	22
2.4	Rangkaian Neural	27
2.5	Thinning Process	29
2.5.1	Definition	29
2.5.2	Thinning Algorithm	30
2.6	Picture Description Language	33
2.6.1	Chain Code Scheme	33
2.5.2	Bribiesca Chain Code	37
2.7	Corner Detection Algorithm	39
2.7.1	Feature Extraction	39
2.7.2	Chain Code based	43
2.7.3	Artificial Intelligence based	44
2.8	Summary	44

III	RESEARCH METHODOLOGY	46
3.1	The General Framework of Geometric Modeler	46
3.2	Problem Identification and Classification	50
3.3	Nature of Data	52
3.4	Pre-Processing	52
3.5	Development of Algorithm	53
3.6	Testing and Validation	54
3.7	Implementation	54
IV	THE PROPOSED ALGORITHMS	56
4.1	The New Framework	56
4.2	The Boundary Representation Data Structure	57
4.3	Digitisation Process	61
4.4	The Enhanced Thinning Algorithm	61
	4.4.1 Basic Thinning Constrain	61
	4.4.2 Stage I: Neighbourhood greater than 3	62
	4.4.3 Stage II: Neighbourhood equal to 2 or 3	63
4.5	VCC Conversion Algorithm	65
	4.5.1 The Formation of Rectangle Cell	65
	4.5.2 The Derivation of Vertex Chain Code	66
4.6	FCC Feature Extraction: The Computational Model	68
	4.6.1 The Chain Code Series Derivation	68
	4.6.2 T-junction, Corner, and Line Detection	75
	4.6.3 Region Detection	85
4.7	VCC Corner Detection: The Neural Network Classifier	91
4.8	Summary	96

V	RESULTS AND DISCUSSION	97
5.1	Introduction	97
5.2	The Digitisation of Input Object	99
5.3	The Enhanced Thinning Algorithm	102
5.4	The Conversion Algorithm	103
5.5	Computational Corner Detection	103
5.5.1	The Chain Code Algorithm for T-junction, Corner, and Line Detection	103
5.5.2	The Chain Code Algorithm of Region Detection	104
5.6	Neural Network Classifier	105
5.7	Research Contributions	108
5.8	Discussion	112
VI	CONCLUSIONS AND FUTURE WORKS	113
6.1	Conclusions	113
6.2	Future Works	114
6.3	Summary	116
	REFERENCES	118
	APPENDIXES	
	Appendix A1-A6	125-130

LIST OF TABLES

TABLE NO.	TITLE	PAGE
4.1	Start test location (clockwise direction)	70
4.2	Row and column values for $i = \frac{n+1}{2}, \dots, n-1$	78
4.3	Row and column values for $i = \frac{n-3}{2}, \dots, 0$	79
4.4	Row and column values	81
4.5	The Sum of difference of rows	81
4.6	The Sum of difference of columns	82
4.7	Total of row and column differences	82
4.8	Row and column values	83
4.9	The Sum of difference of rows	83
4.10	The Sum of difference of columns	84
4.11	Total of row and column differences	84
4.12	Start test location (anti-clockwise direction)	86
4.13	The best model of VCC with the parameter	95
5.1	Table of Vertex and Table of Edge (2D)	104
5.2	Table of Face (2D)	105
5.3	Corners detected by the VCC Classifier	107

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
2.1	Example of line drawings	12
2.2	Irregular line drawings and their regular representation	13
2.3	Origami world 'W-folded paper' line drawing	13
2.4	Engineering sketch represents a cube	14
2.5	Line drawings of impossible object	14
2.6	Line drawings of wireframe object	15
2.7	Process flow of a sketch interpreter	18
2.8	Description of neuron	28
2.9	Description of neural network formula	28
2.10	Thinning process	30
2.11	The 2D Freeman chain code	34
2.12	Freeman cubic lattice chain code	35
2.13	Example of Freeman Chain Code	37
2.14	Three different cells	38
2.15	VCC of triangle cell	38
2.16	Five directions and codes of 3D discrete curve	39
2.17	T-junction creation	39
2.18	Irregular line drawing with three corners	40

2.19	Line creation	42
2.20	A region bounded by 4 junctions	42
3.1	The architecture of the DCS system	47
3.2	The new framework of a geometric modeler	49
4.1	The schematic diagram of the proposed algorithm	57
4.2	Constrain of thinning algorithm	62
4.3	Twelve cases of total neighbourhood equal to 2	64
4.4	Twelve cases of total neighbourhood equal to 3	64
4.5	From Thinned Binary Image to Rectangle Cell	65
4.6	The Process of Rectangle Cell Formation	66
4.7	Example of VCC Cell Code	67
4.8	Example of VCC code of the rectangle cell	67
4.9	Chain code & the location	68
4.10	Chain code format and its example	69
4.11	Example of binary image	70
4.12	Example coordinate (2,6)	72
4.13	Example coordinate (6,2)	73
4.14	Example of binary image	73
4.15	Example coordinate (3,5)	74
4.16	Example coordinate (5,3)	75
4.17	T-junction	76
4.18	A corner at coordinate (6,6)	80
4.19	No corner at coordinate (2,4)	83
4.20	Example of binary image (indicated with T-junction)	87

4.21	Direction of chain code for the first region	88
4.22	Direction of chain code for the second region	89
4.23	Direction of chain code for the third region	89
4.24	Example of pole for corner and not corner detection	91
4.25	Chain code Pattern Pole (corner exist)	92
4.26	Chain code Pattern Pole (corner exist)	93
4.27	Iteration during the training process	94
4.28	The graph in train the network	94
5.1	The Adobe interface and the cube	98
5.2	The interface of Feature Extraction Module	99
5.3	A cube	100
5.4	The cube displayed in the interface	100
5.5	The binary image	101
5.6	The thinned binary image	102
5.7	The boundary and internal line chain code	104
5.8	The region chain code	105
5.9	The input of the classifier for Stair	106
5.10	Corners at the Stair Drawing	107
5.11	Output of the Classifier and the Actual Output	107
5.12	The optimised thinning algorithm	109

LIST OF ABBREVIATIONS

2D	Two-dimensional
3D	Three-dimensional
B-Rep	Boundary Representation
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CSG	Constructive Solid Geometry
DCS	Design Capture System
DWG	AutoCAD Drawing File Format
DXF	Data Interchange Format
FCC	Freeman Chain Code
GVA	General View Assumption
PCC	Primitive Chain Code
PDL	Picture Description Language
TIFF	Tag Image File Format
TOE	Table of Edge
TOF	Table of Face
TOV	Table of Vertex
VCC	Vertex Chain Code

LIST OF APPENDIXES

APPENDIX	TITLE	PAGE
A1	Output display of TIFF information (Cube1)	125
A2	Output display of binary image (Cube1)	126
A3	Output file of the binary image (Cube1)	127
A4	Output display of thinned binary image (Cube1)	128
A5	Output file of the thinned binary image (Cube1)	129
A6	Output file of complete chain code series (Cube1)	130

CHAPTER I

INTRODUCTION

1.1 Definition of Geometric Modeler

Geometric Modeler interpreter is a system that allows engineers to design naturally using input devices and automatically interprets the sketch to a three-dimensional object. It consists of two parts namely the interface and the engine. In general, the interface must imitate a natural style of sketching by providing all attributes such as quick response and act as pencil and paper as used by engineers in sketching engineering product. The sketching concept used in Palm Desktop or commercial software such as PaintBrush or Adobe Photoshop software is an example of the interface. On the other hand, the engine of a geometric modeler must be able to produce accurate two-dimensional regular line drawing or three-dimensional object by classifying type of drawing intended and validating the drawing.

1.2 Motivation

In engineering product life cycle, the cycle starts with conceptual design and ends with product marketing. Pencils and papers are common tools used to sketch by engineers. The rapid development of hardware and software enable these tools to be

replaced by digitiser and light pen. The evolution of Computer Aided Design (CAD) initiated by Sutherland (1963) in 1960's had given tremendous effect on the development of Computer Aided Design software. Since then, developments of CAD softwares that cater needs in designing, updating, storing and visualising engineering drawing have rapidly improved. Unfortunately, the failure of conventional interface in CAD software that used mouse, pop-up menu, and icons at conceptualisation stage is a lack of the development of CAD software in manufacturing industries because engineers do not want to lose the naturalness of paper and pencil when conveying their ideas. This fact is supported by research done by a few number of researcher as explained in this section.

Foley (1976) examined man-machine communication from a language viewpoint. He claimed that the best interaction techniques are the most natural to the users. Therefore, better interaction tool is needed because the existing mechanism does not permit actions that are sufficiently natural.

Hwang and Ullman (1990) suggested that icons and menus are barriers between the designer and the design. They are the artificial devices added to enable the designer to instruct the interface of his attentions, rather than aiding the designer to visualise his design. He also pointed out that current interfaces are aimed at accurate graphical presentation, rather than rough sketches required when a designer is working quickly at a conceptual stage.

Goldschmidt (1992) examines architects at work during their design process, and concludes that sketch is not merely an output mechanism to record an idea, but actually a form of dialogue between the designer and the drawing, a form of external representation to aid the brain in visualising a design. He also stressed that the input method must be fast and natural to the user, to prevent the design dialogue from breaking.

Wang (1992) in his survey stated that there are two important research areas in the field of computer vision and artificial intelligence. They are the reconstruction of a three-dimensional object from its two-dimensional projections and its corresponding problem of 3D object recognition. He emphasised the importance of the degree of user interaction that is necessary for correct reconstruction and internal representation used in the reconstruction process.

Jenkins and Martin (1993) emphasised the importance of sketch in conceptual design. In a conceptual design process, a designer cannot tolerate losing ideas because of constraints created by devices such as mouse, icons, and menus. Therefore, pen and pencil are still preferred by designers.

Pham (1994) stated that CAD still falls short of designer's expectation; they still remain to be seen more as a drafting tool than a design tool. Designers cannot use or at best feel very uncomfortable to use these systems during the early stages of design.

Grimstead and Martin (1995) stated that CAD systems are biased towards computer rather than design; this introduces barriers between the designer and the modelling system. The use of natural pencil and paper style interface moves the CAD system near to the designer.

Matsuda et. al. (1997) emphasised the importance of the sketch in conceptual design. He introduced a new method to deal with sketches for inputting geometric model at a workstation using stylus pencil and a tablet.

Liu and Lee (2001) claimed to that current CAD tools cannot directly convert a line drawing into a 3D object, denying mean of input. Therefore, it is highly desirable to develop algorithms that can convert a design sketch into a 3D model.

Ullman (2002) stated that freehand sketches still constitute a fundamental tool for the engineer to express his or her creativity, because CAD systems are still not entirely suitable for conceptual design.

Therefore, the aim of this research is to propose methods that can be used in the development of geometric modeler as commercial softwares, so that it can cater to the needs of natural style of sketch in conceptual stage that is important but yet still not available in market. To support the development, research on the techniques to develop a user-friendly geometric modeler interface and more accurate and faster engine to reconstruct the drawing should be given attention. With the help of existing input tools, such as stylus pencil, tablet and digitiser, the objective to provide tools for engineers in sketching process is possible.

1.3 Objectives

There are three stages in the development of a geometric modeler. The stages are pre-processing stage, two-dimensional feature extraction, and three-dimensional object reconstruction. The objectives of this research can be classified according to the stages involved. The three objectives are listed below:

- i. To develop a new framework in representing, interpreting, and reconstructing a two-dimensional freehand engineering sketch of three-dimensional object that meets the general architecture of geometric modeler.
- ii. To maintain the continuity in representing, interpreting, and reconstructing sketches.
- iii. To develop enhanced algorithms in sketch interpretation that consists of thinning algorithm in the pre-processing stage, chain code algorithm in representing the drawing, new corner detection method in 2D feature extraction.

- iv. To develop algorithm in term of mathematical model to convert the single two-dimensional data into three-dimensional solid object. Due to the budget constraint, this objective cannot be achieved.

1.4 Scope

There are many types of line drawing such as line drawing of impossible object, wireframe object, origami world, and engineering sketch. The research only accepts valid line drawing of engineering sketch that represents a three-dimensional object.

In the first stage, this research proposes improvement of existing thinning algorithm, proposes vertex chain code conversion algorithm, and proposes Freeman chain code and vertex chain code algorithm as a picture description language to represent the irregular line drawing.

In the second stage, this research focuses on the use of vertex and Freeman chain code in feature extraction. There are two algorithms proposed in this stage. First is the corner and region detection algorithm of FCC representation besides the derivation of T-junction and lines. Second is the corner detection using neural network classifier of VCC representation. In this stage, a regular line drawing which is represented by boundary representation (B-Rep) scheme is produced.

In the third stage, the research should focus on the reconstruction of the two-dimensional data into three-dimensional solid object, but due to the budget constraint this research is limited only to the first two stages

1.5 Assumptions

This research has made a few assumptions to simplify the interpretation and reconstruction process. The line drawing of the system is assumed as a freehand sketch line drawing in the form of scanned image, drawn using any sketch software such as Adobe or any CAD software, and in format of Tag Image File Format (TIFF) graphic image.

In the sketch, when a scene is projected onto the view plane, only the edges are drawn and exactly two faces share every edge. For every face, one side of face is occupied with material and the other side is empty. Therefore, texture and scribbles on a surface are not drawn. The sketch is assumed to represent a valid three-dimensional object where all unwanted pixels have been removed and there is no unconnected pixel. The sketch is assumed as a sketch with all informative lines shown. The general viewpoint assumption (GVA) is applied in this research where the observer is not coplanar with any pair of non-collinear edges and the observer is not coplanar with any face. This eliminates degenerate case for example, a plane that is projected by a line.

These assumptions make the reconstruction more logical or otherwise line drawing of a cube projected on a view plane is seen as a square because the projection is parallel to three faces of the cube. In this case, it is impossible to interpret and reconstruct the line drawing as three-dimensional object.

1.5 Report Outline

The report is divided into six chapters. The first chapter gives an overview of the geometric modeler, motivation, objective, scope of this research, assumptions made and report outline. This chapter ends with a brief explanation of contributions made in this research.

The second chapter discusses literature review on seven areas namely the sources of line drawing, background of boundary representation, development of geometric modeler, thinning process, picture description language, and corner detection algorithm. The inter-relationship between these areas and their importance in the research are explained in the summary of the chapter.

The third chapter explains the methodology used in this research. The discussion starts with the explanation on general framework of geometric modeler development cycle and the proposed framework. This is followed by description of six steps involved in the methodology. The steps are problem identification and classification, nature of data, pre-processing, development of algorithms, model formulation, and testing and validation process. The six steps involved in the methodology are based on the general framework explained in the introduction of the chapter.

The fourth chapter explains contributions of the research. First, this chapter gives details of the new framework for geometric modeler development life cycle. Next, two pre-processing stages required by the new framework i.e. (the definition of data structure and the digitisation process) are discussed before the explanation of the other five contributions is preceded. The other four contributions discussed are the enhanced thinning algorithm, vertex chain code conversion algorithm, Freeman chain code feature extraction algorithm, and vertex chain code neural network classifier for corner detection. Summary is given at the end of the chapter.

The fifth chapter discusses on the result of the enhanced algorithms. The discussion is divided based on steps involved in the interpretation and is supported by a sketch of a cube and stair. Research contribution and discussion are given at the end of the chapter. Chapter 6 suggests future works of the research and gives conclusion of the whole thesis.

1.6 Contributions of the Research

There are six contributions of the research. The contributions are the enhanced thinning algorithm, VCC conversion algorithm, FCC closed loop chain code algorithm for T-junction, corner and line detection, FCC closed loop chain code algorithm for region detection, neural network classifier for corner detection of vertex chain code series, and the new framework of geometric modeler in the form of new algorithm for 3D object interpreter.

The proposed algorithm covers all stages in the development of a geometric modeler. The enhanced thinning algorithm utilises image-processing concept in skeleton a sketch. The VCC conversion algorithm presents the automation process in representing thinned binary image as vertex chain code. The FCC closed loop chain code algorithm acts as a language to represent the sketch. The closed loop chain code for corner and region detection algorithm detects the existing of corners and regions that are represented by the chain code series. The neural network classifier of vertex chain code presents a new intelligent corner detection algorithm.

The algorithms and mathematical model presented and designed in this thesis can be used as independent tools in different areas have been designed to act as integrated tools in the proposed framework. The thinning algorithm that has been used as a tool in image processing is enhanced to produce thinned binary image. The thinned binary image then acts as an input of the FCC and VCC algorithm. Chain code representation that has been used as picture description language of line drawing is enhanced and applied in the mathematical model and neural network corner detection algorithm. The integration of these tools produces a new framework of geometric modeler.

The enhanced thinning algorithm has been used to remove unwanted pixel and to simplify the image so that the new image will suit the philosophy of chain code algorithm. The proposed thinning algorithm is an enhancement of the work by Pitas (1995).

The VCC conversion algorithm presents a new algorithm in automatic conversion of a thinned binary image into vertex chain code. The work is based on Bribiesca (1999) chain code scheme. The new algorithm has been tested on three line drawings and should be programmed so that it can perform on any line drawing.

The FCC closed loop chain code algorithm is an improvement of work on Picture Description Language (PDL) by Freeman (1969) with specialization in representing a line drawing. A new scheme in traversing the pixel is proposed by introducing two parameters namely setting the first location of the code to be traversed, and the direction of the traversal.

The extraction of T-junction is based on the intersection of lines by utilising the series of chain code and their Cartesian coordinates. In the corner detection algorithm, computational method has been applied by introducing a new algorithm to detect corners between consecutive T-junctions. Region detection algorithm shows that the combination of Table of Vertex (TOV) and region chain code series can be used to produce the new algorithm.

Neural network classifier has been used to detect corner of the chain code algorithm. This research has proposed a chain code based classifier since not many works has been done on this topic.

Applying these algorithms and models in the proposed algorithm produces an intermediate and final data structure to represent a solid model. The data structure is then saved as boundary representation that is now ready to be viewed using any CAD software.

The proposed algorithm is able to produce a boundary representation of single planar, trihedral model drawn as single isometric, hidden-line removed sketch line drawing. The algorithm is an improvement on general framework of geometric modeler

proposed by Hwang and Ullman (1990). The objective of the algorithm is still similar namely to produce a valid 3D solid model.

CHAPTER II

LITERATURE REVIEW

2.1 Line Drawing Sources

Line drawing is one of the simplest forms of graphic expression in many fields. It represents two-dimensional scenes such as diagrams and plans and three-dimensional scene such as solid model or origami world. Line drawing can be in the form of irregular and regular line drawings. With the help of Computer Aided Design system, irregular line drawing can be improved by using the editing command. The existence of CAD system actually eliminates the naturalness of sketching. The advent of peripheral technology such as high quality scanner, light pen, and digitiser has given alternative to the CAD system. Therefore, an interpreter that interprets the hand sketch must be available so that the sketch can be interpreted and solid model depicted by the drawing can be produced.

The definition of the term line drawing, type of line drawing and sources of the line drawing are given in this section. A line drawing is an abstraction derived from an image that conveys its information solely through the shape of thin lines on a contrasting background. Freeman (1969) classified line drawing into two types namely regular and irregular line drawing. Regular line drawing is a picture which consists of straight lines, arcs or circles or other conic sections, and standard symbols (such as characters). An irregular line drawing is one primarily constructed from irregular curves. Regular line

drawings occur in architectural and technical drawings; while irregular line drawings are found in terrain maps, electrocardiograms, the tracks of animals looking for food, and most other line drawings that are used to describe natural rather than man-made phenomena or objects. Figure 2.1 shows these two types of line drawing.

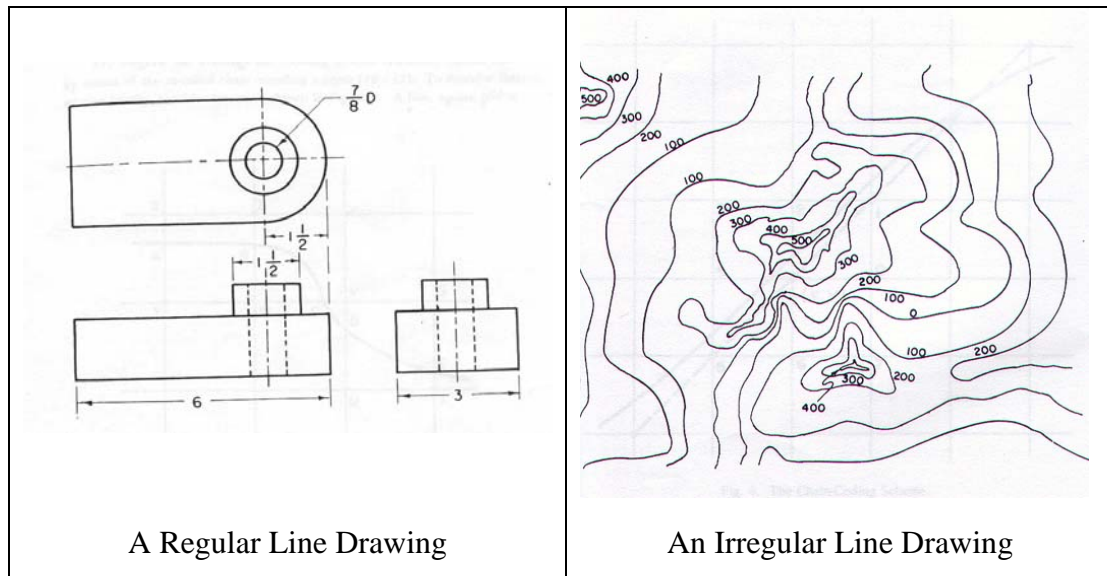


Figure 2.1: Example of line drawings Freeman (1969)

This research focuses on irregular line drawing. Line drawing can be produced from many sources. Four sources of line drawing discussed in this section are origami world, engineering sketch, impossible object, and wireframe model. Given a line drawing with different thickness, it actually represents the same topology. Figure 2.2 (a-c) shows a line drawing formed with three different line thickness that is actually having similar geometry. In order to represent these lines, a single representation is enough. Figure 2.2 (d) is one of the regular line drawings of the freehand sketch in Figure 2.2 (a-c). Note that more than one regular line drawing can be formed based on an irregular line drawing.

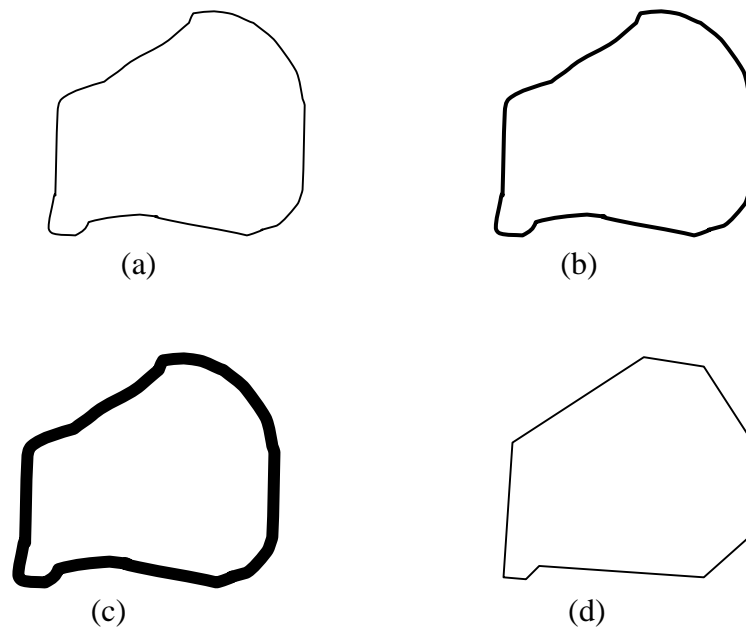


Figure 2.2: Irregular line drawings and their regular representation

2.1.1 Origami World

Started in Japan, origami world is an art of making an object from a sheet of paper. Since the folding of the paper will represent lines, the view of the origami world can be represented through a regular line drawing. Figure 2.3 shows an example of line drawings of origami world that represents a bird. The line drawing of origami world is classified as regular line drawing as folding a sheet of paper creates it.

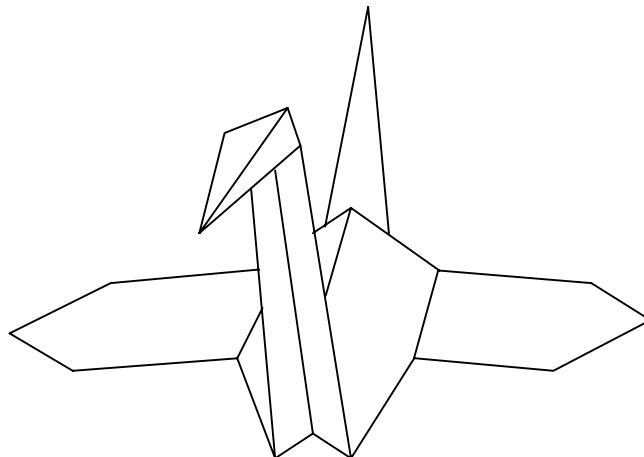


Figure 2.3: Origami world 'W-folded paper' line drawing Kanade (1980)

2.1.2 Engineering Sketch

An engineering sketch is a preliminary design in the life cycle of engineering product design namely in conceptual stage. The sketch must be tidied up and translated into proper engineering design. It is classified as irregular line drawing. Figure 2.4 shows an example of engineering sketch. The drawing represents a cube in three-dimensional world. This research focuses on this type of line drawing.

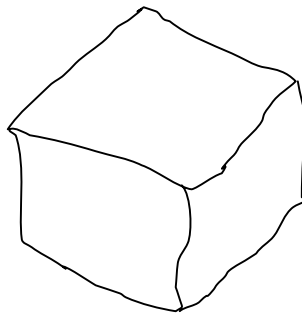
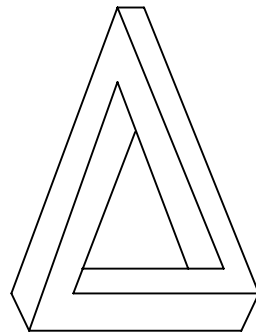


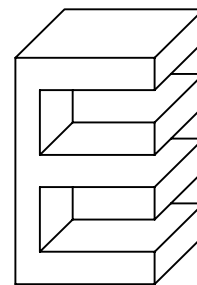
Figure 2.4: Engineering sketch represents a cube

2.1.3 Impossible Object

Impossible object can be represented by a regular line drawing as shown by E block and Penrose tri-bar object in Figure 2.5. The drawing is designed uniquely to test mind. Research, such as that conducted by Cowan (1977), has been done in picture interpretation to determine the correctness of this type of line drawing.



Penrose Tribar



E Character

Figure 2.5: Line drawings of impossible object ^{Huffman (1971)}

2.1.4 Wireframe Object

Wireframe model is a three-dimensional representation of a solid object without faces. It is classified as regular line drawing as it is produced by Computer Aided Design software representing a three-dimensional object. Figure 2.6 shows examples of line drawings of a wireframe object that represent a cube. It shows that ambiguity exists in this type of line drawing.

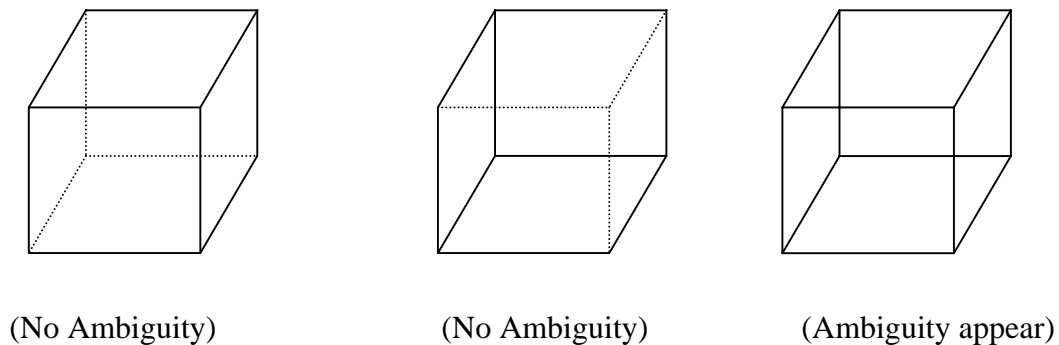


Figure 2.6: Line drawings of wireframe object

2.2 Boundary Representation

Solid modelling is a computer description of three-dimensional objects in sufficient details to render, analyse, or manufacture the model in a realistic manner. Generation of numerical code for machining or analyses of finite element are examples of its use in solid modelling. In computer processing, examples of the representation schemes to record solid models are primitive instancing, sweep representation, boundary representation, feature-based, and constructive solid geometry (CSG). Since methods of recording the solid model in term of hard copy or soft copy are regarded as standard graphics procedures, CAD textbooks such as by Hoffmann (1989) and Lee (1999) can be referred for further details. In this section, three representation schemes discussed are boundary representation.

Boundary representation represents an object in terms of boundary, namely in terms of vertices, edges, and faces. There are several variations of boundary representations such as the winged-edge representations; this particular representation has each edge stored with one pointer for each incident face, two pointers for terminating vertices, and two more pointers for its incident faces. A simplified version of this structure is used in this research, where an edge has pointers to its incident faces, and two pointers to its terminating vertices; no pointers are made to incident edges, but each vertex has pointers to its three incident edges (because of a trihedral solid assumption). Boundary representation may also be used to model objects with curved surfaces, although this thesis limits itself to planar surfaces. Note that boundary representations may not necessarily describe a valid object, solids may self intersect; faces may not be planar or bounded (which would cause problems when a planar model is required). Despite this, boundary representation is used in this research due to ease of representation and manipulation. It is particularly suited to the nature of the data input, and the manipulation to be carried out, which are creation of linear constraint equations and the conversion of the two-dimensional into a three-dimensional solid model.

Data representation used by Marill (1991) and Leclerc and Fischler (1992) is not computer oriented and structured enough even though the usage of table is applied. The list of junctions and lines in a form of table makes it difficult for computer to represent them in any data structure of programming language. Compared to the boundary representation used by Grimstead (1997) and Lipson and Sphitalni (1996), the representation used by Grimstead is more systematic and easier to be represented by structured programming language. For that reason, this research uses the boundary representation in defining the data structure of 2D line drawing and 3D solid model.

2.3 A Geometric Modeler

This section discusses factors that encourage the research development in sketch interpretation. Strictly speaking, it is impossible to uniquely identify a 3D structure by drawing a single view. Line drawing interpretation involves two stages namely feature extraction and feature interpretation. Feature extraction is a lineal description of the image sensed by the scanner and is extracted by means of image processing. Feature interpretation is a process to obtain a set of interpretations of the line drawing as an object in 3D space. This section discusses past works on the two-dimensional sketch interpreter of line drawings that the user wishes to be interpreted as three-dimensional solid models. The discussion of a sketch interpreter is categorised into three issues. The first discussion is on the general process flow of a sketch interpreter. The second discussion is on the development of a computer system that provides interface for on-line sketching. Finally, the third discussion is on geometric modeller that infers the missing depth information from image regularities in the drawing that produce possible three-dimensional solid model.

2.3.1 Sketching Process

A sketch interpreter is a system that interprets and reconstructs a freehand sketch. Figure 2.7 shows the process flow of a sketch interpreter. The process starts during the sketching process where engineers sketch ideas on a sheet of paper or any input device. Draftsmen then translate the sketch into a proper regular line drawing manually or by any sketch interpreter automatically. This drawing is then interpreted in terms of validity by classifying the drawing by its sources. Once the interpreter validates the drawing and represents it in the form of computer representation, the interpreter then extracts the 2D information of the drawing. Based on the 2D data structure, the interpreter interprets the drawing into a three-dimensional object and stores it in as a neutral format that can be accessed by any CAD software for visualisation and manipulation.

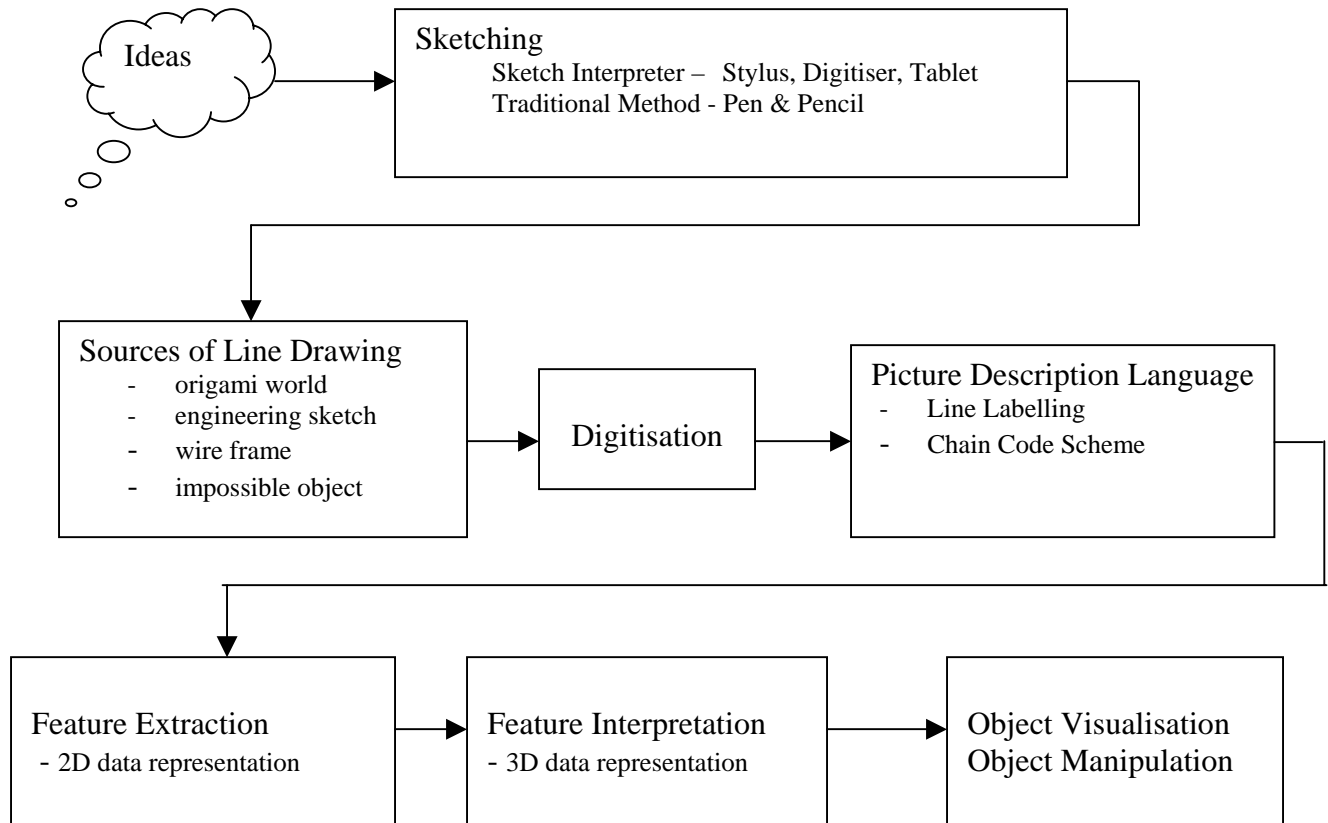


Figure 2.7: Process flow of a geometric modeler

2.3.2 The Sketch Interface

Computer Aided Design started in 1960's. The development of digital computer and computer hardware fastens the growth of study in geometrical modelling. The technology of CAD software has improved in terms of user interface such as higher resolution screen, variety of input methods such as light pen and mouse, icon in windows, faster processor to display and process image, and in dimensioning tool. In conventional design process, one of the earliest processes involved is conceptualisation. This section discusses interfaces that provide a natural interface between computer and engineers during sketching. The term natural interface used in this thesis means the interface of a computer system that combines the features of Graphical User Interface

(GUI) such as icons and menus, and hardware such as digitizer and light pen to imitate the pencil and paper used in traditional way of sketching.

Sutherland (1963) started the evolution of two-dimensional sketch. The work has given tremendous effect in the emergence of CAD software. The system enables users to interactively digitise a three-view drawing, consists of two stylus, and a digitising tablet, one stylus is self-standing, and is positioned over a point in one view, and the second stylus is used to trace the lines in another view. One disadvantage of his work is that it interprets the drawing as two-dimensional objects.

Guzman (1968) developed a system called SEE that considered visual scenes composed by the optical image of a group of bodies. The system looks at three-dimensional objects in a two dimensional scene. The scene itself is not obtained from visual input device, or from an array of intensities or brightness. Rather, it is assumed that a pre processing of some sort has taken place, and the scene to be analysed is available in a symbolic format, in terms of junctions, lines, and faces.

Mackworth (1973) developed a system called POLY that interprets a line drawing as polyhedral scenes. The method is based on general coherence rules that the surfaces and edges must satisfy, thereby avoiding the use of predetermined interpretations of particular categories of picture junctions and corners.

Hwang and Ullman (1990) used an expert system called Design Capture System (DCS) to recognize parts from a known database. The system allows users to sketch directly into the computer (without constraint), and the system tidies the two-dimensional sketches. The limitation is that the system must contain all possible features and parts in its knowledge base before the user can expect a sketch to be correctly interpreted. It shows that the system lacks of general solid features.

Sivaloganathan (1991) developed an interface that allows users to directly sketch a three-dimensional object by first tidying the two-dimensional drawing, and then interpreting it as an isometric projection.

Marill (1991) emulated model-based human interpretation of line drawing by constructing a program called CONSTRUCT. The fundamental principle underlying the program was Minimum Standard Deviation Angle (MSDA) and Necker cube illusion.

Hale et al. (1992) used input cross-sections as features in the sketch interpreter. The user first orientates the axis about which drawing will take place (the three areas forming three planes, called centreline, plan and section). The user sketches the centreline; plan view and several sections, all in two dimensions. A solid model is then built up by fitting a surface through the lines sketched. This is not normally a very natural approach to design an object. This system was intended for the car design industry, and as such is suited extremely well for this industry only. The limitation with his style of input is that the model produced is not fully defined by the user.

Pugh (1992) developed a program called Viking that acts as interactive sketch interpretation. The program allows designers to create a line drawing and the program automatically generates a three-dimensional object description. The constraint solver applied in the program allows the program to reject inconsistent interpretation based on the presence of hidden lines-segment.

Jenkins and Martin (1993) introduced an interface called Easel for conceptual design with the philosophy to replace pencil and paper without losing the naturalness on conveying and throwing ideas in conceptualisation stage. The proposed technique that uses light pen and digitiser was a neater, faster and more accurate tool to design. Without the interface proposed by him, ideas from designers or engineers drafted on paper can also be analysed by scanning the image, storing it in computer and interpreting it so that the intended 3D solid model can be visualised. The system also tidies the hand

sketch so that all lines are connected and there will be no unwanted lines and junction as well as no unbounded region.

Lipson and Sphitalni (1996) provided a natural user interface where the user sketched a design in the form of wireframe. The system interprets the sketch as a three-dimensional solid through an optimisation-based reconstruction. Image regularities in the sketch are used to create an optimised function for producing the resulting solid. One of the image regularities used by Lipson that was used in this research is skewed symmetry.

Zelevnik et. al. (1996) developed an interface for sketching 3D scenes called SKETCH. The program uses simple non-photo realistic rendering and a purely gesture interface based on simplified line drawings of primitives that allows all operations to be specified within the 3D world.

Eggl et. al. (1997) proposed quick modeller for conceptual design called Quick-sketch. It has full two-dimensional sketching interface which tidies the user's input as the sketches are drawn, but has a limited three-dimensional interface. The only mean to produce three-dimensional objects is either by interpolating of a two-dimensional contour or the creation of a B-Spline surface by interpolating between two curves. So the user can sketch additional details (again, by extruding two-dimensional features) on the surface of the resulting three-dimensional object. This is not as natural sketching on a piece of paper since the user must consider the interface in order to provide an object as an input. For example, user must enter a side-on view to provide L-bracket as an input, ready for extrusion, instead of sketching a projection of the bracket directly onto a screen.

Matsuda et. al. (1997) introduced a new method to deal with sketches as inputs to geometric models through a system called Sketch Interpreter. Sketch is drawn on a CRT screen through a stylus pencil and a tablet. Three characteristics of the system are to recognize hand sketches on the screen, to construct additional geometric models, and to

redraw modified geometric models by replacing the sketch lines and applying thinning algorithm.

As a summary, examples of sketch interpreter system can be found in systems called SEE by Guzman (1968), Poly by Mackworth (1973), Design Capture System by Hwang and Ullman (1990), CONSTRUCT by Marill (1991), Viking by Pugh (1992), Easel by Jenkins and Martin (1993), SKETCH by Zeleznik et. al. (1996), Quick-sketch by Egli et. al. (1997), and Sketch Interpreter by Matsuda et. al. (1997). Besides these systems, commercial software are also available such as PaintBrush, Adobe Photoshop, and MacDraw to sketch drawing but they are not intended to suit the need of naturalness to emulate pen and paper.

2.3.3 The Modeler Engine

This section discusses geometric modellers used to interpret and reconstruct a single view line drawing. Recovery of hidden parts or inferring missing depth involves many techniques. This section presents works on recovering hidden parts by classifying them based on approaches used. The discussion focuses on the line drawing of impossible object, origami world, wireframe object, and engineering sketch with hidden-line removed regardless that the source of the line drawing is a freehand sketch or regular line drawing.

Kanade (1980) demonstrated on how the theories and techniques that exploit assumptions about 3D shapes of objects could provide a systematic shape recovery method. Two parts discussed were the application of Origami theory that models the world as a collection of plane surfaces and recovers the possible shape qualitatively, and the techniques of mapping image regularities into shape constraints for recovering the probable shapes quantitatively.

Barrow and Tenenbaum (1981) proposed a computational model that was based on constraints on local surface orientation along external and discontinuity boundaries. Techniques for recovering 3D conformation of a space curve and interpolating smooth surfaces from orientation constraints along extreme boundaries were also discussed.

Fukui (1988) allowed users to draw freehand by which a boundary solid model was defined with a tablet in two dimensions, and attempted to fix lines in three dimensions relative to those previously drawn. This feature works well for a new face that is adjacent to two or more faces as it is fixed in space. Unconnected faces were assumed to be face-on to the user (parallel to the projection plane), and faces connected to single face (one degree of freedom) were rotated such that the three-dimensional face has minimum surface area. This enabled the user to create fairly complex objects, yet still there were constraints placed on the user. Problems also occurred when two separate faces were drawn, which were as yet unconnected as the system will automatically assume that they were the same distance from the viewer. The method limits the naturalness of interface and there was no freedom in constraints.

Hwang and Ullman (1990) presented an algorithm where two-dimensional features were matched against a database of three-dimensional features and corresponding objects (three-dimensional features being projected into two dimensions for comparison with the sketch), finally producing the interpreted object that consisted of the recognized parts.

Sivaloganathan (1991) presented an algorithm by assuming that, in two dimensions, any lines along three major axes are interpreted as lines along the x, y or z-axes in three-dimension. The system interprets straight lines and ellipses, but has the three axes of the isometric projection fixed on-screen, represented as graph paper. This feature limited the naturalness of the sketching, as the user has to orientate the sketch to fit the three axes. The system encounters difficulties if a few of the lines in the sketch are actually parallel to the three isometric axes (as the depth of the object is under

constrained). With these shortcomings in mind, the direct interpretation of isometric axes that it was not suitable and a general solution was preferable.

Hale et. al. (1992) proposed a system that formed an approximate surface to pass through the lines, and introduced error. It would be extremely hard to design an angular polyhedron with this method, but it serves as a quick visualisation tool for approximated curved surfaces. Since this thesis produces an accurate solid model of polyhedron, this technique was insufficient for our purpose.

Marill (1991) presented an algorithm that combined a line representation using integer pair (i, j) , minimum standard deviation of angles, and the Hill-Climbing scheme in seeking the minimum standard deviation of angles. Leclerc and Fischler (1992) improved Marill's approach by improving the objective function that favoured both symmetry and planarity. Improving the MSDA function and using a more competent optimisation technique reduces time taken to reconstruct.

Pugh (1992) presented an algorithm that was fully compatible with the object descriptions used by traditional solid modeller. The proposed three-dimensional sketch helps the user visualise the object it represents using geometric constraints as its simple mechanism in specifying approximate vertex positions.

Wang (1992) applied constraints in the form of the Huffman-Clowes line labelling, and constructive solid geometry concept to interpret a line drawing that represented a polyhedron. The difference with this thesis is in the usage of boundary representation instead of CSG to represent the solid model. Since the work was based on a CSG representation, it is beyond our discussion.

Sugishita et. al. (1994) used perspective drawing to add extra constraints to the sketch, making depth retrieval less ambiguous, but this lead to a more complex system of equations to solve, compared to parallel projection. An additional problem was that the system had to rely on the user to draw an accurate perspective projection freehand,

which was difficult to accomplish with a complex object. This makes the system unusable for quick sketching, due to the precision required. The aim of this research is to support a natural interface, which is robust to drawing errors and quick to use. Hence the usage of perspective projection is rejected, opting for orthographic projection instead which is easier for a designer to sketch.

Akeo et. al. (1994) used cross-section lines of an idea sketch in the form of wire frame drawing as an input data. The proposed system automatically creates surface and finds the relationship between the direction of surface loop and the vector viewpoint to determine whether a surface is the front or back surface.

Grimstead (1997) described a method of reconstructing a boundary representation solid model from a single hidden-line removed sketch view of a 3D object. A line labelling of the sketch, identification of image regularities, creation of linear system, and usage of least square method were the steps involved in the reconstruction step. The method produced a valid 3D interpretation of many cases of valid sketches; produced most plausible interpretations for a subset of sketches, and was quick enough to be considered interactive.

Lipson and Sphitalni (1996) used optimisation-based algorithm for reconstructing a 3D model from a single, inaccurate, 2D edge-vertex graph. This method avoided the usage of line labelling and supported manifold and non-manifold object that contained flat and cylindrical faces.

Zelevnik et. al. (1996) interpreted a series of simple gesture from the user to produce three-dimensional objects; these can then be combined using CSG to create more complex objects. The stumbling block with this system was the amount of gestures the user was required to learn; the user must then spend time using CSG to create the desired object. Again, the user interface required too much consideration by the user to be natural in usage while this research aims to create a system that is completely natural to use.

Egglı et. al. (1997) developed an algorithm that allowed a pen stroke to be interpreted interactively. The interpretation system used graph-based constraint solver to establish the geometric relationship.

Matsuda et. al. (1997) presented an algorithm that can correct geometric models in a computer even though input data of 2D sketches were drawn freehanded. Mathematical parameters were computed and additional geometric models were constructed in the interpretation. Company et. al. (1999) presented an algorithm that accepted large variety of polyhedral, origami, and wireframe objects using optimisation-based approach.

Varley and Martin (2000) converted a 2D sketch with hidden lines removed of a single polyhedral object into a boundary representation by improving Grimstead's system. Image regularities and symmetry elements were used to produce more plausible topological completion of the hidden parts of the object, and to enforce exact constraints on the geometry.

Liu and Lee (2001) focused on surface reconstruction from single 2D line drawing using graph-based method. Face identification was formulated as a maximum weight clique problem. The work not only improved previous work by Lipson's but managed to accept a manifold object.

Langbein et. al. (2001) focused on beautification of reconstructed solid model. The work based on boundary-representation model that required beautification due to noise in the data and the model building. The method sought clusters of similar objects that described properties of faces, loops, edges, and vertices.

Liu and Lee (2002) proposed new method for finding faces from single 2D line drawings representing manifolds. The face identification was formulated based on a property of manifolds: exactly two faces share each edge of a manifold. The two main

steps in the method are searching for cycles from a line drawing, and searching for faces from the cycles.

Vicent et. al. (2003) focused on the skewed mirror symmetry in the 3D reconstruction of polyhedral models from axonometric line drawings. The new approach determines planes of mirror symmetry based on prior detection of their planar faces and any axes of symmetry of these faces.

As a summary, as early as early 80's, works on interpretation and reconstruction of line drawing have started. With the same objective to produce a valid three-dimensional solid object, methods in terms of computer algorithm, mathematical formulation, constraint-based, geometrical-based, and optimisation-based have been proposed. This research proposed a new framework by combining algorithm-based and mathematical framework to achieve the same objective.

2.4 Neural Network

Neural network is a new stream that is getting popular in computer era. There are many techniques in NN such as fuzzy logic, genetic algorithm and neuran network. This research chose neural network as medium in applying the case study.

Biologically, neural network is a model from cognitive process of human brain. Rosa (2001) stated that nodes for neural network are similar with human brain. Neuron consisted of dendrite, acson, sinaps and soma. Figure 2.8 shows the cell contains in the neuron. Norgaard et al. (2000) describes neuron in neural network form as formula as shown in Figure 2.9. Rajagopalan and Rajagopalan (1996) stated that neural network functions as human brain.

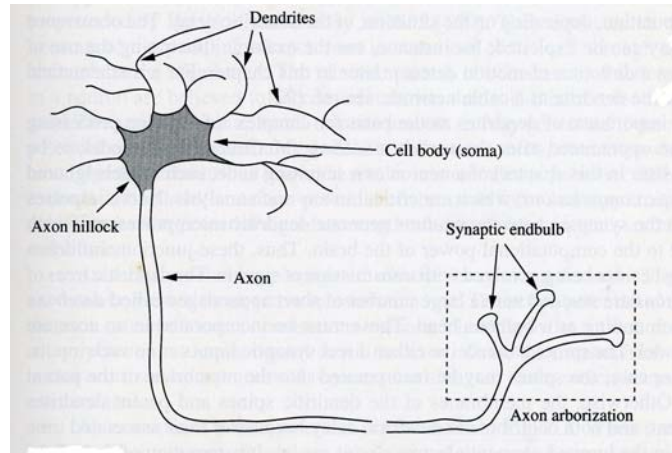


Figure 2.8: Description of neuron

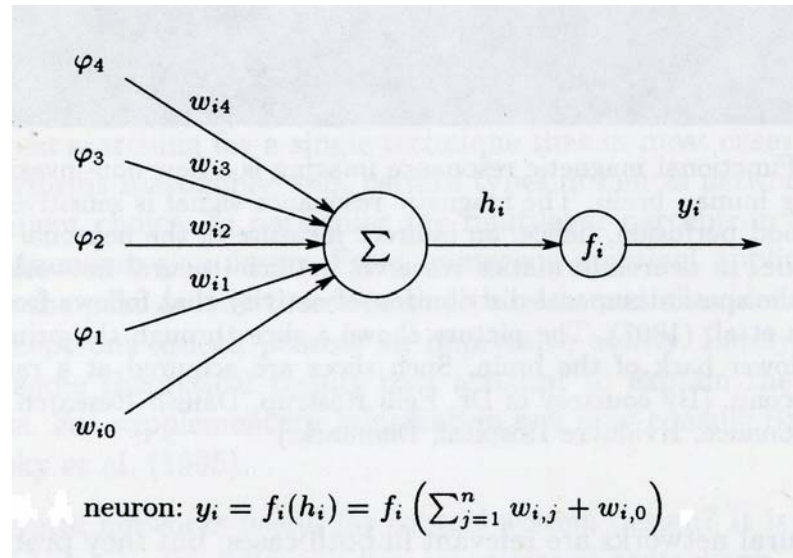


Figure 2.9: Description of neural network formula

Tsai (1997) used back-propagation algorithm in the corner detection algorithm. Terry and Vu (1993) also used back-propagation algorithm in edge detection and used back-propagation to train the input. The classifier produced by Dias et al. (1995) also used back-propagation in the neural network classifier. This shows that back-propagation is very popular in development of neural network classifier.

2.5 Thinning Process

This section reviews on the thinning process used in line drawing interpretation. The discussion is divided into two sections namely the definition on thinning and reviews on the thinning algorithm that leads the researcher to the proposed enhanced thinning algorithm. There are many works done on thinning algorithm but the discussion focuses on the thinning algorithm that is integrated with the use of chain code scheme.

2.5.1 Definition

Thinning is an important pre-processing step for many image analysis operations such as print recognition, optical character recognition, and biomedical systems. Thinning is a process of producing a skeleton of an image by getting rid of all redundant points, but preserving the basic structure and characteristics of the pattern. There are two types of thinning algorithm namely sequential and parallel. This section presents these two types of thinning algorithm for digital images. Figure 2.10 shows example of how thinning process skeletons an image but preserves the basic structure and characteristics of the pattern. Sequential thinning algorithm refers to the deletion process where contour points are examined in a predetermined order. Parallel thinning algorithm refers to the deletion of pixels based on the results of only the previous iteration.

A number of approaches for thinning have been reported in literature. Lam et. al. (1992) identified that there are 138 publications on thinning algorithm. This shows the importance of thinning in pattern recognition and the various approaches to yield skeletons of shapes. Research conducted on sequential and parallel thinning algorithm also shows that thinning algorithm can be used in reducing the amount of pixels to be saved and to be processed.

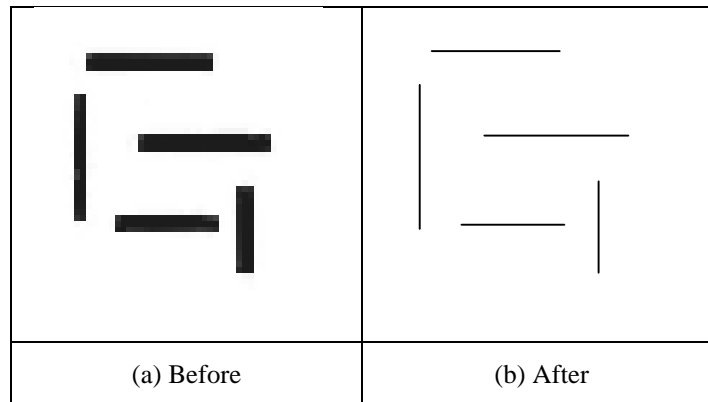


Figure 2.10: Thinning process

2.5.2 Thinning Algorithm

This section discusses work on thinning algorithm. The discussion can be categorised on the proving that the thinning algorithm does not change the connectivity of picture, improvement of existing algorithm, and proposing a new thinning algorithm in different applications. Here, the discussion is based on the year of events and its importance on the evolution of thinning algorithm.

At early stage of development of thinning algorithm, proving that algorithm does not change the connectivity of picture is important. Stefanelli and Rosenfeld (1971) proved this theory by proposing parallel thinning algorithm for digital pictures in the work.

Deutsch (1972) investigated the connectivity criteria on triangular, rectangular, and hexagonal arrays. In the proposed thinning algorithm, he concluded that the algorithm operated in conjunction with the hexagonal array has features that strike a balance between those of the other two arrays.

Zhang and Suen (1984) proposed a fast parallel thinning algorithm to extract a picture by removing all the contour points of the picture except those points that belong to the skeleton. Two sub iterations applied were deleting the southeast boundary points and the northwest corner points; and deleting the northwest boundary points and the southwest corner points.

Lu and Wang (1986) improved the fast parallel algorithm by Zhang and Suen (1984) and Stefanelli and Rosenfeld (1971). Wang et. al. (1986) described a method called contour generation thinning (CGT) that iteratively generated new contour pixels from a queue of coordinates of the pixels.

Kwok (1988) combined the idea of chain code and contour generation. He found that the removal of a pixel could be done implicitly and automatically when considering the multiplicity of a pixel. This led to considerable speed up when compared with other contour generation algorithms. Kwok (1990) again proposed a modified contour generation thinning algorithm. He used three pieces of information to process a contour pixel, compared with four in the case of the original algorithm and nine in the case of typical sequential or parallel algorithm. Kwok (1992) then extended his work by giving relation of thinning algorithm with the chain code scheme and presented two algorithms. Additional data structures were incorporated in the chain code to ensure consistency in the propagation of contour pixels.

Lee and Wang (1994) proposed a new thinning algorithm that was simple yet could maintain the same advantages of other key algorithms. He used only very small set or rules of criteria for deleting pixels, which was faster and easier to implement.

Pitas (1995) defined thinning as heuristically a set of successive erosions of the outermost layers of a shape, until a connected unit-width set of lines (skeleton) was obtained. Thus, thinning algorithms are iterative algorithms that ‘peel off’ border pixels, i.e. pixels lying at $0 \rightarrow 1$ transitions in a binary image. He also stressed that connectivity was an important property that must be preserved in the thinned image. Therefore,

border pixels were detected in such a way that object connectivity was maintained. To suit the chain code algorithm, he concluded that two stages were required in the thinning process.

Zhu et. al. (1995) proposed new line-based thinning algorithm that gives freedom of choosing the deletion width at iteration, and thus significantly reduced the number of iterations. The proposed algorithm was a new alternative method compared to point-based thinning algorithm.

Matsuda et. al. (1997) proposed thinning algorithm for interactive line drawing by segment composing, clipping, and segment joining. The purpose of his thinning algorithm was to fix an edge line from freehand sketch lines. He used the thinning algorithm in the sketch interpretation.

Flores (1998) proposed a fast thinning algorithm based on chain code of three directions. He used 3x3 windows in delimitation of the contour of the image. Staunton (2001) developed a fully one-pass parallel hexagonal thinning algorithm. The work compared the algorithm with rectangular thinning algorithm. Mathematical morphology was used as techniques in the algorithm.

Cowell and Hussain (2001) applied thinning algorithm in producing a thinned 1 pixel thick representation to recognize Latin characters by extracting features from that character such as number of strokes, stroke intersections and holes. He used 5x5 windows to prevent tail growth. Kwon et. al. (2001) proposed an enhanced parallel thinning algorithm that connected the pixels to each other and preserved the end point by improving the algorithm proposed by Zhang and Suen (1984), Lu and Wang (1986) and Wang et. al. (1986).

As a summary, previous research on new algorithm mostly focused on 3x3 windows, except those carried out by Deutsch (1972) and Cowell and Hussain (2001). The objective of thinning algorithm is to maintain topology or basic shape of the image.

This research uses point-based serial thinning algorithm as proposed by Pitas (1995), and integrates thinning algorithm with chain code scheme as suggested by Kwok (1992). The application of thinning algorithm by Matsuda et. al. (1997) in sketch interpretation also proves that the thinning algorithm can be applied in sketch interpretation.

2.6 Picture Description Language

Research on PDL started in 1960's to represent line drawing in a mathematical and linguistic approach. Two picture description languages discussed in this thesis are line labelling and chain code scheme. Picture description language is introduced by Clowes (1971) to analyse and recognize letter. A picture can represent thousand meanings. Therefore, to simplify the meaning of a picture and to represent a picture in a computer, a language must be developed. Representing a line drawing in the context of language is a study under cognitive science. Primitive two-dimensional entities such as line, circle, and ellipse can be represented by an equation, but irregular line drawing that cannot be represented by an equation. So ways must be searched to represent the irregular line drawing so that it can be represented by a computer. This section discusses how to represent an irregular line by a computer representation. This section is divided into two sections namely line labelling and chain code representation. Besides representing the line drawing, PDL also used to categorize and to validate a line drawing. For example, does the line drawing represent an impossible object or origami world?; or does the line drawing represent a valid 3D object?

2.6.1 Freeman Chain Code Scheme

This section presents an alternative scheme in representing a picture called chain code scheme. Chain code scheme is widely used because it preserves information and allows considerable data reduction. Chain codes are the standard input format for

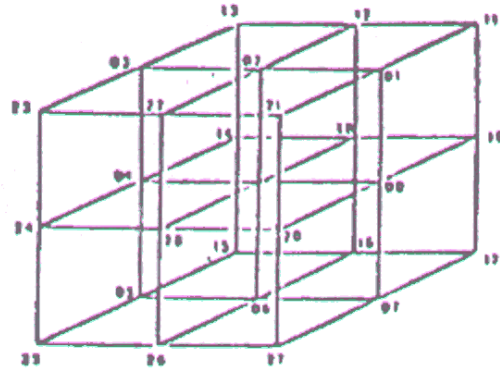


Figure 2.12: Freeman cubic lattice chain code ^{Freeman (1974)}

Minami and Shinohara (1986) proposed multiple encoding schemes called multiple grid chain code (MGCC). The proposed scheme was claimed to have many advantages over the Freeman chain code. O’Gorman (1988) proposed primitives chain code (PCC) to represent image that contains eight-connected lines. PCC was an extension of FCC, but in addition contain chain code connectivity information. There are 185 PCC codeword in total compared to 8 in FCC.

Choo and Freeman (1989) presented an efficient technique for compressing chain-encoded line drawings. The technique constructed a codebook containing vectors of chains that recur frequently in the chain-encoded line drawings.

Maeder (1990) tried animating digital images sequences for visualisation using chain coding scheme. Method for scaling, rotation and elastic deformation of objects based solely on chain code elements was presented. Chen and Lee (1991) introduced a new three-dimensional chain code for encoding 3D curves called C-chain code. The new chain code has extended Freeman chain code to 3D scheme.

Choo and Freeman (1992) presented algorithms for computing features of objects from poly curve-encoded boundaries. Some geometric calculations such as inverse and intersection points of two poly curves were also considered. Nunes et. al. (1997) used MGCC, work proposed by Minami and Shinohara (1986), to present a chain

code based approach to efficiently codes binary shape information of video objects, in the context of object-based video coding.

Bribiesca (1999) proposed a new chain code to represent shapes called vertex chain code (VCC). The VCC was invariant under translation and rotation. Using this concept, it was possible to relate the chain length to the contact perimeter, which corresponds to the sum of the boundaries of neighbouring cells of the shape. Bribiesca (2000) again used chain code representation extensively in representing line drawing of three-dimensional curve. The 3D discrete curve was composed of the constraint straight-line segments of the discrete curve. He proposed to use this chain code for representing solid composed of voxels in a single chain series.

Sun et. al. (2002) used Freeman chain code to recover human posture. The silhouette surrounding the region of body was represented by FCC. Then each part of the body was labelled, and the coordinates of joints among the parts were retrieved.

As a summary, the definition and development of chain code in representing 2D line and 3D object, and the application of chain code have been discussed in this section. This thesis applies FCC to represent 2D line of a thinned binary image. Figure 2.13 (a) shows an example of a line drawing that can be represented by Freeman chain code. Figure 2.13 (b) shows the digitised image of Figure 2.13 (a), and Figure 2.13 (c) is the thinned image of Figure 2.13 (b) shown by the grey pixels. The chain code series of FCC in Figure 2.13 (c) are shown at the bottom of the figure. Note that the thinned binary image is based on Pitas's thinning algorithm. This thesis proposes an enhanced thinning algorithm that is integrated with the FCC. The chain code series then acts as input to any image processing process such as in extracting the two-dimensional geometric features namely junction, line and face.

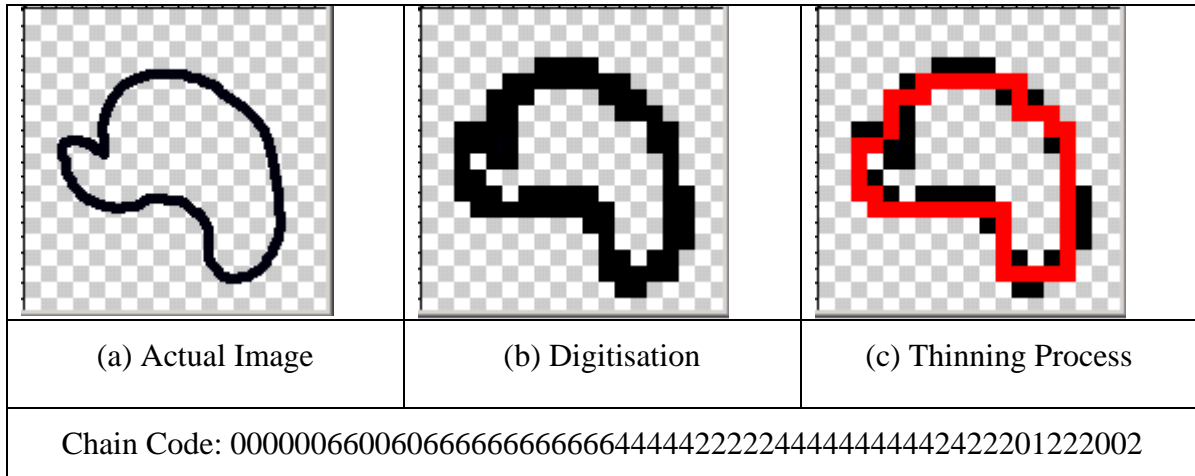


Figure 2.13: Example of Freeman chain code

2.6.2 Bribiesca Chain Code Scheme

This chain code scheme is another chain code scheme proposed by Bribiesca (1999). This chain code is boundary line chain code and based on the vertex cell that bounding edge line of any kind of shape. The chain code scheme is called vertex chain code (VCC). Shape can be translated without refer or represented by Cartesian coordinate. The VCC is not affected by rotation or scaling. Bribiesca (1999) said that using this concept, it is possible to relate chain code length to perimeter where total of cell boundary line adjacent to the shape is related directly to chain code at the vertex based on the adjacent vertex cell.

VCC satisfies three objectives of chain code as mentioned by Freeman (1969).

Nevertheless, VCC has three own characteristics:

- i. VCC is not affected by translation or rotation, and also not affected when mirroring process and starting point created.
- ii. VCC can represent three kinds of cell namely triangle, rectangle and hexagonal.
- iii. Its chain element represent actual value.
- iv. Using the VCC the relation between edge line of boundary and internal shape can be achieved.

VCC has its own way to represent a shape. There are three form of cell namely triangle, rectangle and hexagonal and shown in Figure 2.14. Each shape has its code; triangle represented by 1, 2, 3, 4 and 5, rectangle represented by 1, 2 and 3 and hexagonal represented by 1 and 2. Figure 2.15 shows example of shape represented by rectangle cell. Figure 2.16 shows another version of chain code proposed by Bribiesca (2000) called three-dimensional discrete curve. This scope is beyond the discussion in this research.

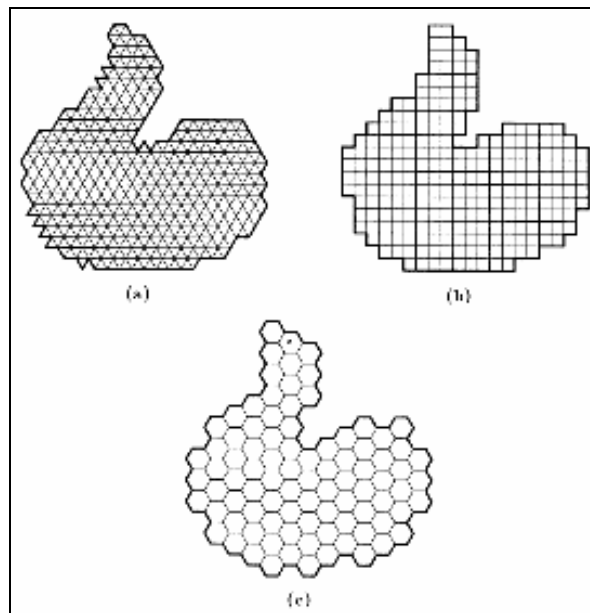


Figure 2.14: Three different cells: (a) triangle (b) rectangle and (c) hexagonal

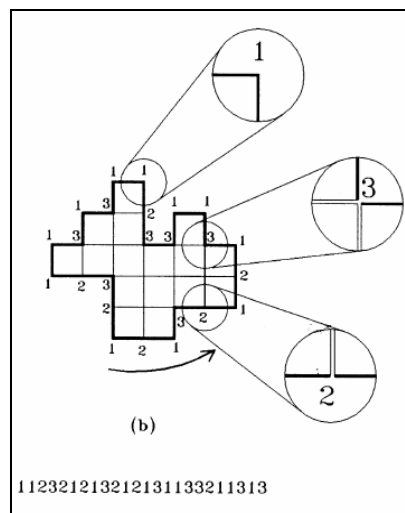


Figure 2.15: VCC of rectangle cell

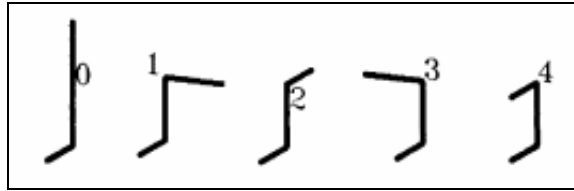


Figure 2.16: Five direction and codes of 3D discrete curve.

2.7 Corner Detection Algorithm

2.7.1 Feature Extraction

Feature extraction of a line drawing involves the process of deriving the four basic entities namely T-junction, corner, line and region. Deriving corner involves a corner detection algorithm, while deriving T-junctions, lines, and regions involve the application of Computer Aided Manufacturing (CAM) geometry concept as used in Automatically Programmed Tool (APT) language and can be found in CAD/CAM book such as by Lee (1999). Note that *T-junction* and *corner* are the terms used to represent a *junction* in 2D space that are related to the term *vertex* used in 3D space. The vertex is produced by intersection of two lines in 2D drawing and intersection of three faces in 3D object. The term *line* and *region* are used in 2D space and is related to the term *edge* and *face* in 3D space respectively. These conventions are used in this thesis.

Junction is made of intersection of two lines as shown in Figure 2.17. This concept is applied in deriving T-junction of line drawing.

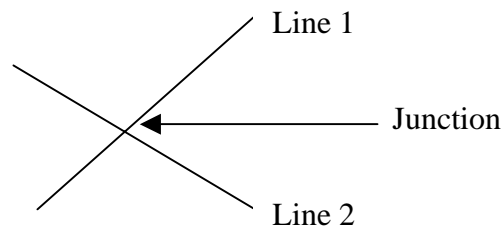


Figure 2.17: T-junction creation

Corner detection is an important aspect in image processing and finds many practical applications. Corner that exists in any irregular line must be detected so that the irregular line can be interpreted to represent actual line. Figure 2.18 shows example of an irregular line drawing consisted of three corners. Based on these corners, a region can be detected. Research on the corner detection has widely been done in many types of image but this discussion focuses on corner detection of an irregular line drawing. This section also presents work on corner detection that uses chain code as input to their algorithms on a thinned binary image.

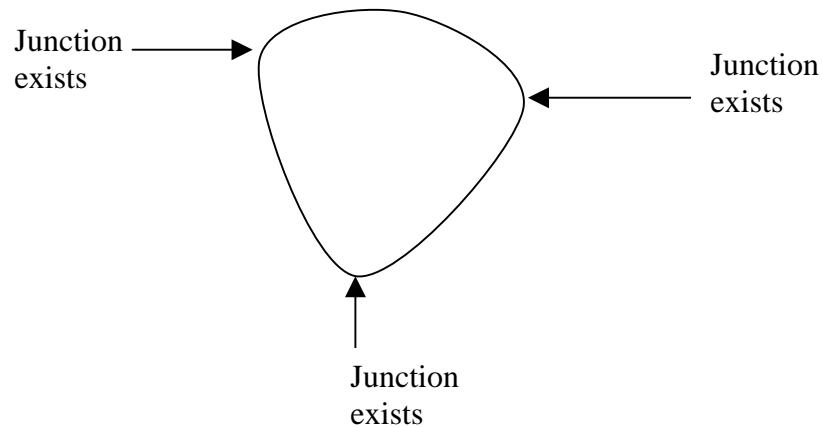


Figure 2.18: Irregular line drawing with three corners

Rajan and Davidson (1989) determined that the main characteristic that a corner detection algorithm should have is that the formula used to determine the corner value must be easy to compute, that it took all parameters of a corner point into account, and that it was less sensitive to noise. They addressed six issues to be considered in corner detection algorithm that were corner angle, corner arm length, corner adjacency, corner sharpness, grey level distribution, and noise level. This research focuses on two of these issues namely the corner arm length and corner adjacency, with the application of corner detection using chain-coded images.

Bachnak and Celenk (1989) presented corner detection that operated on a thinned binary image and used 3x3 windows in two computational phases. They used topological properties of corners and proposed similar domain by using chain code and thinned binary image as the input image to detect corner. The work has similarity with this thesis in term of the input image and the method in representing the curve.

Arrebola et. al. (1997) again proposed a new method for determining the shape of an object by obtaining its dominant points or border corners. The algorithm used the contour chain code as descriptor and calculated the curvature of each border point, and compared two histograms of the chain code.

Arrebola et. al. (1999) extended his work and proposed a new algorithm for representing and characterizing the contours of 2D objects, as well as detecting their corners. The use of curvature estimation techniques is based on circular histograms of the contour chain code. Two histograms were calculated at the neighbourhood of every point, and instead of considering them of linear type and being compared by correlation; they were considered to be circular and compared by a new method which defines a circular mean of variable origin. The usage of histogram is not considered in this research and therefore it is not considered.

Bandera et. al. (2000) extended Arrebola's work by proposing planar shapes on a new curvature function. The function produced calculation parameters that were not constant, but adapted the maximum length of the contour, yielding no significant discontinuities at each of its points in order to calculate the incremental adaptive chain code of the contour at those points.

Reche et. al. (2002) extended Bandera's work for planar shapes based on a new curvature index. They estimated the maximum length of contour yielding no significant discontinuities on the right and left sides of each contour point in order to calculate the contour curvature index at that point. Corners were detected by simply threshold the curve.

Works on corner detection by previous researchers show that chain-coded representation of thinned binary image is one of the types of image that has been investigated. The proposed algorithms by these researchers are too complicated therefore they are rejected. However they are considered in term of the usage of chain code of the thinned image. This thesis proposes a new simple computational method to detect corner but still produces a valid corner along a curve.

In 2D line drawing, one way to define a line is through connection of two junctions as shown in Figure 2.19. This concept is applied in this thesis by assuming corners and T-junctions as junctions.

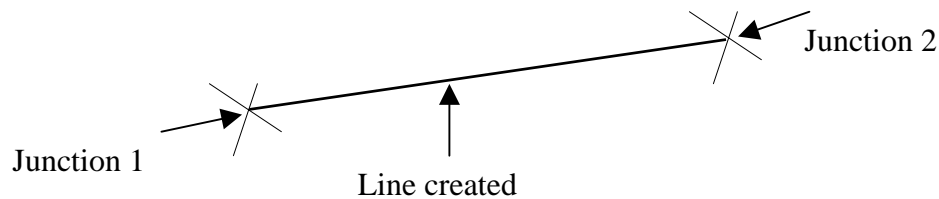


Figure 2.19: Line creation

In 2D line drawing, a region is created by a series of junctions as shown in Figure 2.20. This concept is applied in this thesis to derive region by connecting corners and T-junctions that bound a region.

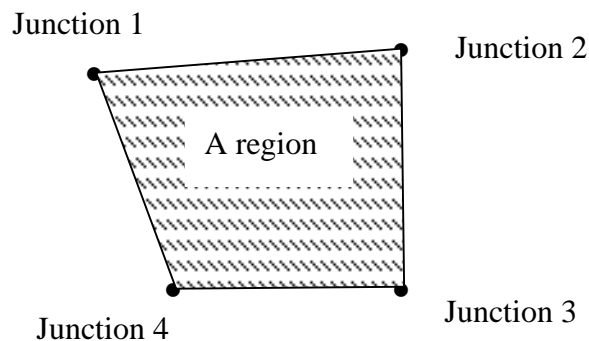


Figure 2.20: A region bounded by 4 junctions

2.7.2 Chain Code based

This section presents work on corner detection of chain code representation. Rosenfeld and Johnston (1973) developed a corner detection using chain code. The works on corner detection as listed i-vi below. The list are work on corner detection based on chain code representation compared by Liu dan Srinath (1990).

- i. Corner detection Rosenfeld-Johnson by Rosenfeld(1973)
- ii. Corner detection Rosenfeld-Weszka by Rosenfeld(1975)
- iii. Corner detection Weighted-K-curvature by Rutkowski (1978)
- iv. Corner detection Medioni-Yasumoto by Medioni (1987)
- v. Corner detection Beus-Tiu by Beus (1987)
- vi. Corner detection Cheng-Hsu by Cheng (1988)

Below is also the list of works on corner detection based on chain code representation.

- i. Corner detector object representaion of 2D image by Bachnak and Celenk (1988)
- ii. Corner detector for curve chain code by Koplowitz (1995)
- iii. Corner detection algorithm by Sanchiz (1996)
- iv. Local Histogram algorithm by Arrebola (1997)
- v. Corner detector *Means Adaptively Estimated Curvature Function* by Bandera et. al (2000)

Literature review shows that works on corner detection mentioned above used Freeman chain code either directly or with modification. It shows that vertex chain code has never been used in corner chain code. Therefore, this research proposed the work on corner detection of vertex chain code.

2.7.3 Artificial Intelligence Based

This section presents works on corner detection that applied artificial intelligence technique namely neural network . Tsai (1997) stated that the application of neural network onto robust detector will produce a better corner detector. Dias et al. (1995) divided the corner detector into two categories; first is boundary extraction and curve maximum point, and second is related to gray levels. Tsai (1997) proposed two neural network model in the work. First is to detect corner that pose high curve, and second is detect tangen point and response curve that is low curve. The work applied back-propagation with one input layer, and two hidden layer and one output layer.

Dias et al. (1995) applied supervised training in the classifier. The training input of the classifier is angle form of 45, 90, 135, 225, 270 and 315 degree. Nevertheless, the angle can be adjusted to be the input in the range 0 to 360. Terry dan Vu (1993) has produced the similar corner detector using neural networ by representing the pattern as +1 or -1.

The literature work shows that corner detector can be developed in many ways using the neural network classifier. Therefore, this research applied the neural network approach in developing the classifier.

2.8 Summary

This section has discussed six areas that are related in the development of a sketch interpreter. The discussion of these literature reviews leads to the selection of specific algorithms and mathematical model described in Chapter 4. The discussion also identified the weaknesses and strengths of each algorithm and followed by the explanation of the selected algorithm used in this research.

Discussion on the geometric Modeler shows that development of better interface to cater the need of engineers and designers in conceptualisation stage is crucial. It shows that, besides the interface, works on the development of geometric modeller as extension of the interface that accept line drawing of engineering design is becoming more important.

Discussion on boundary representation shows that the most suitable representation to represent a line drawing. Since a line drawing is only represented by two dimensional vertices, edges, and face, the usage of table and pointer-orientation in the data structure leads to the selection of boundary representation.

Discussion on thinning algorithm shows how important the thinning process is in reducing the data to be processed. Discussion on its application in sketch interpretation and its relation to chain code scheme by previous researchers shows their practicality in this thesis.

Discussion on the chain code presents the improvement from Freeman chain code to vertex chain code by Bribiesca scheme. Various researchers have introduced various schemes. It shows how the scheme is as good as line labelling to represent a line drawing and to be used as input of feature extraction process in sketch interpretation.

Discussion on the corner detection algorithm shows that besides computational algorithm, neural network classifier is also the option that can be used in corner detection algorithm. The literature review shows that the application of chain code in neural network classifier is very limited.

CHAPTER III

RESEARCH METHODOLOGY

3.1 The General Framework of Geometric Modeler

The methodology of this research was based on the schematic diagram of Design Capture System (DCS) presented by Hwang and Ullman (1990) as shown in Figure 3.1. Three subsystems involved were Freehand Sketching Subsystem, Feature Recognition Subsystem, and Spatial Reasoning Subsystem. The schematic diagram was used as the basis to draw the methodology of this research. In the first subsystem, the aim was to produce a two-dimensional drawing in terms of digital forms. In the second subsystem, the recognition subsystem produced a three dimensional data. In the third subsystem, the aim was to represent the 3D data in the form that can be represented by a computer. Methods available in solid modelling such as feature graph, boundary representation and constructive solid geometry were used in DCS.

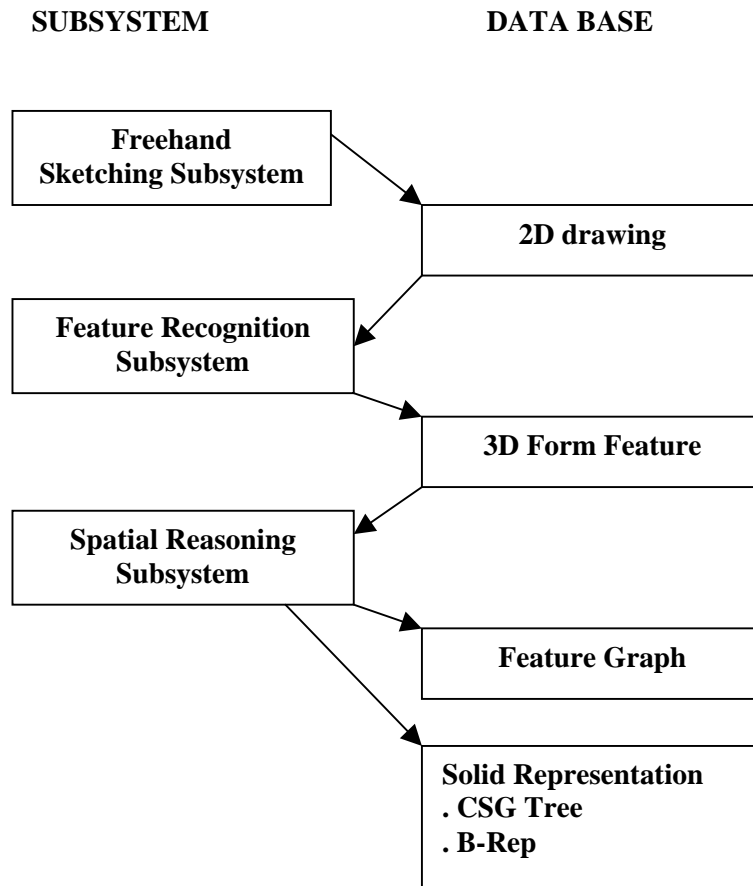


Figure 3.1: The architecture of the DCS system Hwang and Ullman (1990)

The architecture of the DCS has similarity with the architecture proposed in this report in term of the output produced by each subsystem. The detail of the steps of each stage varied due to the selection of algorithms and techniques, source of input drawing and data representation. The three stages in the methodology were still maintained in the proposed methodology but the detail of the steps involved in each stage vary because of different approaches taken. The following steps in each stage outline the detailed methodology employed in this research.

Freehand Sketch Pre-Processing

In the first stage, there were six steps involved. The input of this stage was a line drawing image and the output was the chain code representation of the line drawing.

The steps were as follows:

- i. Identifying the sketch interface used in sketching process.
- ii. Identifying the source of line drawing of sketch interpreter.
- iii. Identifying methods used in representing line drawing.
- iv. Developing thinning algorithm to skeleton an image.
- v. Developing chain code algorithm to represent 2D line drawing of 3D object.
- vi. Testing and comparing the thinning and chain code algorithm with the established methods.

Feature Extraction

In the second stage, there were four steps involved. Input of this stage was a chain code representation of the line drawing and the output was two-dimensional data structure of the line drawing. The steps were as follows:

- i. Identifying methods in extracting features of two-dimensional line drawing.
- ii. Developing algorithm to detect junctions, lines and regions of a chain code representation.
- iii. Creating a B-Rep data structure to represent the extracted features.
- iv. Comparing the feature extraction algorithms with the established model.

Object Reconstruction

In the third stage, there were five steps involved. Based on the two-dimensional data structure passed from the second stage, a three-dimensional data structure of the line drawing was produced. The steps were as follows:

- i. Identifying optimisation methods in 3D reconstruction of a line drawing.
- ii. Identifying image regularities of a 2D drawing that represent 3D solid.
- iii. Developing mathematical models and the solutions.
- iv. Comparing the mathematical model used with the established model.
- v. Drawing conclusions and giving suggestions based on the results of the why not “experiments”?

Figure 3.2 shows the schematic diagram of the proposed methodology that acts as a new framework in sketch interpretation. The explanation of algorithms and formulations in Chapter 4 is based on this schematic diagram. The following sections give detailed explanation of the steps mentioned above.

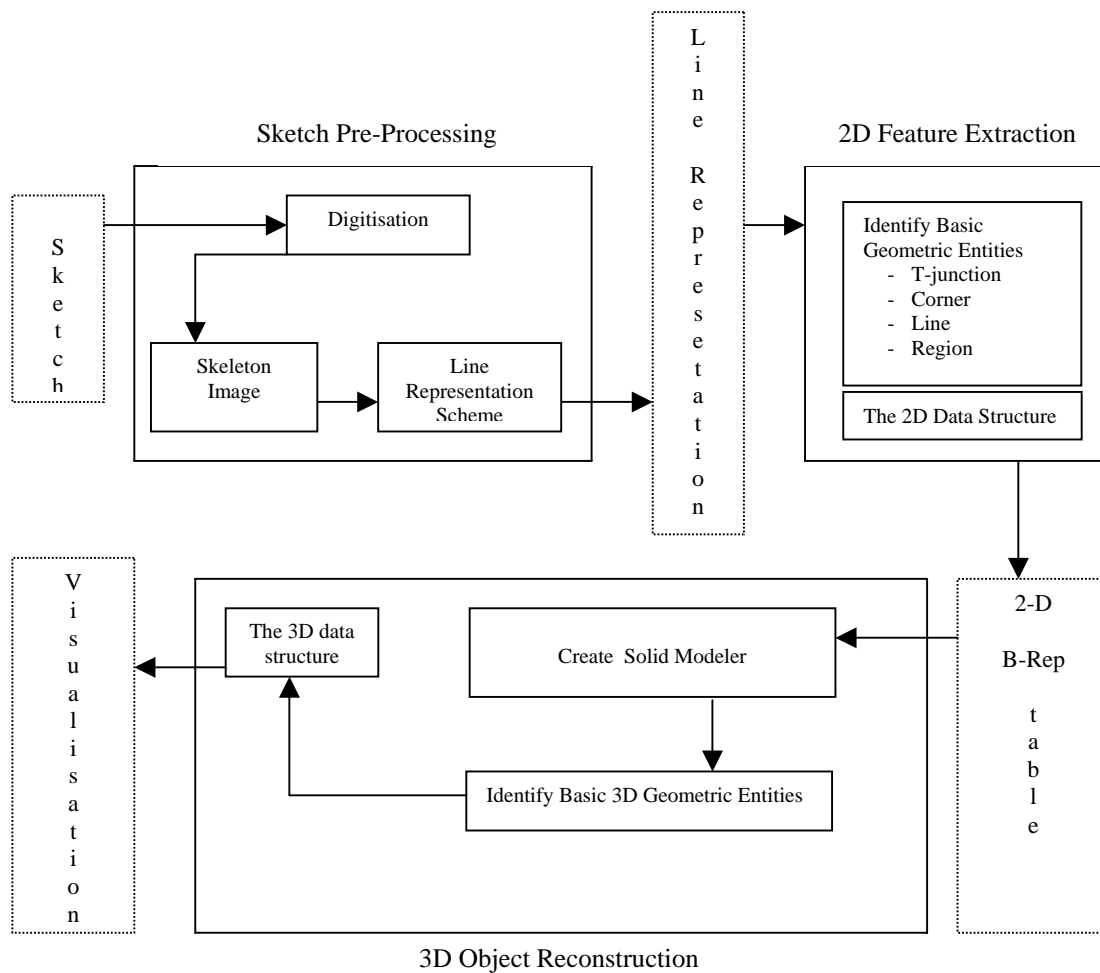


Figure 3.2: The new framework of a geometric modeler

3.2 Problem Identification and Classification

In this step, the existence of problems in sketching, interpreting, representing, and reconstructing a freehand sketch of engineering design were identified and classified, and their natures defined. The identification and classification were based on the stages involved in sketch interpretation as mentioned in Section 3.1 Figure 3.2.

In freehand sketch pre-processing, three elements were identified and classified. First, sketch interface that were proposed and developed by previous researchers were identified and classified. Problems in imitating traditional ways to sketch on-line and the solutions proposed were studied. Besides on-line interface of sketching, the problems in off-line sketch interpreter were also studied. Non-interactive sketch interface shows that pre-processing to digitise and simplify line drawing was needed. Second, source of line drawing interpreted and investigated by previous researchers were identified and classified. The four sources of line drawing identified were origami world, engineering sketch, impossible object, and wireframe model. Third, methods and approaches used in sketch interpretation, and reconstruction were identified and classified. Problems in representing a line drawing in a manner that would be processed and understood by computer were also identified. The identification of this stage led the researcher to propose a non-interactive pre-processing step to represent a line drawing so that extraction of feature would be implemented. The identification also led the researcher to the development of algorithms in simplifying the input image to be represented by a computer.

In two-dimensional feature extraction, problems and solution methods in detecting T-junction, corner, line and region of an irregular line drawing were identified. The classification of the algorithms and type of line to be processed were classified. The classification led to the selection of computational method in the development of corner detection algorithm, iterative-based method in region detection, and geometry-based on T-junction and line detection. The chain code representation was chosen as the line representation after comparisons were made between chain code and other

representation scheme used by the feature extraction module. The output of this stage was kept in the chosen boundary representation data structure.

Feature extraction specifically corner detection has been chosen to applied with artificial intelligence technique based on the works by Tsai (1997), Dias et al. (1995) and Terry dan Vu (1993). These three research works has demonstrated how to identify techniques and parameters to be used in applying AI techniques. All these three works used neural network as medium in producing corner detection based on AI technique.

After NN chosen as AI technique for the corner detection, next problem is to determine the representation scheme that is suitable to represent thinned binary image, and the suitable input and output in the development of NN model. Chain code scheme has been chosen based on literature review that shows that few attempts on using chain code as the input of the corner detection algorithm, but none has applied chain code as the input of the neural network model. The literature review also shows that only Freeman chain code ever used in corner detection algorithm but not the chain code proposed by Bribiesca. .

In each stage, experimental objects were tested, comparisons to the established methods were made, and conclusions were drawn. This research was targeted to produce algorithms and mathematical model that interpret a freehand sketch and produce valid three-dimensional objects. It was hoped that these algorithms and models could be integrated into the development of CAD/CAM software so that sketching process by engineers could be done naturally and not constrained by mouse, menus and bars. As research in sketch interpreter was ongoing, the proposed solutions could be used to produce a complete sketch interpreter.

3.3 Nature of Data

The data used in this research was based on the classification of sources of line drawing, classification of single view and multiple views, and assumptions made on the drawing. Four sources of line drawing that represent origami world object, engineering sketch, wireframe object, and impossible object were identified. For single views, two types of sketch identified were hidden line removal or wireframe object. For multiple views, two types of line drawings identified were three-views and two-views that were depicted from engineering drawing of CAD software. Assumptions made on the drawing were identified as it gave constraints on the development of algorithms and formulation. The General View Assumptions (GVA) given in Section 1.5 were applied on the set of data.

As a summary, the data chosen was a line drawing of engineering sketch, drawn as a single view sketch with hidden-line removed, and applied the GVA. The experimental data was based on the freehand sketch taken from object tested by Grimstead (1997). Although the data was to imitate a sketch in natural manner by engineers, simplification and clearance of unwanted pixel and lines such as dangled line had been made to suit the algorithm developed.

3.4 Pre-processing

Pre-processing of the image was required because non-interactive concept was employed in the sketch interpreter. Two stages required were image acquisition and digitisation process. In image acquisition, there were two ways used. First, image was acquired by sketching the drawing using any image editor such as Adobe Photoshop and then the image was saved in a file as TIFF format. Second, using a sheet of paper and a pencil or pen, a drawing was sketched. Then, it was scanned and saved as TIFF image.

None of these methods require programming functions. The TIFF image obtained acts as the input of the pre-processing stage that contains line drawing.

In digitisation process, C functions that read the TIFF image were required. By reading the image, pixel values of each location were extracted. The grey level values were extracted and converted into a raw file format by choosing threshold value 200. The selection of the threshold value was based on experimentation. The output of the pre-processing stage was a two-dimensional array of the pixel value of raw file format that consisted of binary values that represent the line drawing.

3.5 Development of Algorithm

Three algorithms developed in the research were thinning algorithm, VCC conversion algorithm, FCC conversion algorithm, feature extraction algorithm of the FCC, and VCC neural network classifier of corner detection.

The development of thinning algorithm was based on the review done on Pitas (1995). He stated that the development philosophy of thinning algorithm was to ensure that the constraint of thinning algorithm was maintained.

The VCC conversion algorithm was based on Bribiesca chain code by applying the chain code scheme to represent a thinned binary image. The algorithm convert the binary image into the chain code that can be used in any feature extraction process.

The development of FCC chain code algorithm was based on the Freeman chain code algorithm by applying it in a different context. The development of the algorithm was implemented so that the closed boundary of the thinned binary image could be represented and analysed by the feature detection algorithm.

The feature detection algorithm detected four entities namely the T-junctions, corners, lines, and regions. The algorithm used chain code as its input. Three approaches were applied in the detection namely computational method, iterative-based, and geometry-based. The computational model was developed in the corner detection

while geometry concept used in Automatically Programmed Tool (APT) language is applied in the detection of T-junctions, lines and regions.

The neural network classifier is implemented on the vertex chain code. The experiment is conducted is the development of the classifier. Nine steps involved in development is followed. The steps is taken from common steps of neural network classifier.

3.6 Testing and Validation

The input freehand sketch image was acquired, digitised, and then the 2D data were extracted. All the pseudo codes and procedures used in the thinning, chain code and feature extraction algorithm were translated into working programs written in C language utilising Borland C 5.02 Compiler and Visual Basic version 6.0. The algorithms developed were tested on the experimental data as separate entity but in a continuous manner where the output of the thinning process acted as input of the chain code algorithm, and the output of chain code algorithm acted as input of the feature extraction algorithm. The output of the feature extraction algorithm was kept in a data structure of two-dimensional boundary representation table.

3.7 Implementation

Hardware platforms employed by the researcher were a 1 Gigabyte Pentium III processor with 256MB of memory and running Windows 2000. It should be noted that the scope of this research did not include establishing necessary conditions to hardware requirements.

Results of the study were implemented by means of C functions that process and display the output of the extraction, manual calculation to derive the set equations of the image regularities, and Matlab functions to solve the linear system and to display the

output. The algorithms and models developed were synthesised in the form of enhanced algorithms. The algorithms and model could be applied separately in each stage for different source of input drawing.

CHAPTER IV

THE PROPOSED ALGORITHMS

4.1 The New Framework

In Chapter 3, the methodology in the development of a geometric modeler that consisted of three parts has been explained. It shows that there are many algorithms and mathematical models used in every stage of the development of a geometric modeler. This research has selected certain algorithms and mathematical models with the same objective that is to interpret line drawing. Figure 4.1 shows the algorithms and mathematical model selected in each stage. Figure 4.1 was actually a refinement of Figure 3.2 where only the selected algorithms and model was shown.

In pre-processing stage, enhanced thinning algorithm and closed loop chain code algorithm was selected to skeleton and represent the line drawing. Chain code was selected as picture description language to be the input to the feature extraction stage. In the Feature Extraction Module, junctions, lines, and regions were detected based on the chain code derived. These entities were then passed on to the third stage in the form of boundary representation table with depth values of zero.

There are four steps involved in the third stage were identification of image regularities, creation of linear system, the usage of total least square to recover the depth of each junction, and the derivation of the edges and faces of the three-dimensional object. The third stage updated the boundary representation by adding the values of

depth in the vertex table. The third stage is beyond our discussion and the research because of budget constraint.

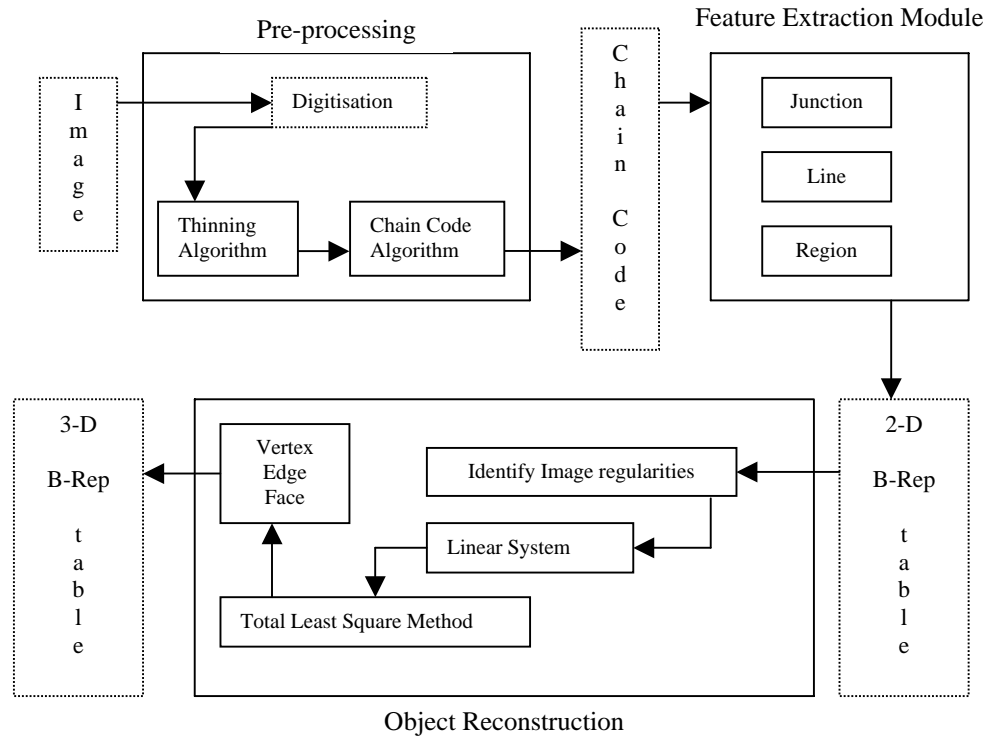


Figure 4.1: The schematic diagram of the proposed algorithms

4.2 The Boundary Representation Data Structure

This section explains data structure used by the algorithm in deriving junctions, lines and regions of the irregular line drawing. An array to represent the coordinate and pixel value was created to hold the binary image and the thinned binary image. Two intermediate records needed for the purpose were boundary and internal line chain code series, and region chain code series. Three basic entities of boundary representations were vertex (T-junction and corner), edge (line), and face (region). Therefore, three records were needed to hold the information of the object. This section describes six records needed by the proposed algorithm.

Array of Image

Array of *image* is a two-dimensional array with type `char`. It keeps *pixel_value* of the thinned binary image that consists of two values namely 0 and 1, and the coordinate *row* and *column* of the pixel. The record structure was arranged as shown below:

```

record {
    integer row[100], column[100];
    integer pixel_value[100];
} image;

```

Table of Boundary and Internal Chain Code

The record is used to keep information of chain code and its Cartesian coordinate of every boundary and internal line of the image. It consists of three fields namely *start_row*, *start_column* and integer array of *chain_code*. *Start_row* and *start_column* represent position of the start coordinate of the code in the image. *Chain_code* array represents integer value ranged from zero (0) to seven (7) that represent the code in the series. If the content of the array is -1, it indicates that there are no more pixels in the list. For boundary list, the row and column of the last record must be adjacent to the first record because it keeps all record of pixels bounding the drawing. The record structure was arranged as shown below:

```

record {
    integer start_row, start_column;
    integer chain_code[200];
} bdry_line_ccode;

```


Table of Region Chain Code

The record is used to keep the code and Cartesian coordinate of every region of the image. It consists of three fields namely *start_row*, *start_column* and integer array of *chain_code*. *Start_row* and *start_column* represent position of the start coordinate of the code in the image. *Chain_code* array represents integer value ranged from zero (0) to seven (7) to represent the code in the series. If the content of the array is -1, it indicates that there are no more pixels in the list. The first record contains junction on boundary and the last record of the list is adjacent to the first record. The list is used to derive the list of junctions that bound the region. The record structure was arranged as shown below:

```

record {
    integer start_row, start_column;
    integer chain_code[200];
} region_ccode;

```

Table of Vertex (Junction)

Table of Vertex consists of five fields namely *vertex_number*, *row_y*, *column_x*, *depth_z* and *junction_type*. *Vertex_number* is an index of the table. *Row_y*, *column_x*, and *depth_z* represent the coordinate of the junction in two-dimensional (2D) and vertex in three-dimensional. For 2D data, the *depth_z* field is always zero. After guessing the depth, this field is updated so that the *depth_z* field represents the depth derived. *Junction_type* is a code that represents the junction type in 2D. The codes are 2, 3 and 99 that represents corner, T-junction on boundary and T-junction on internal line respectively. The record structure was arranged as shown below:

```

record {
    integer vertex_number;
    integer row_y, column_x, depth_z;
    integer junction_type;
} vertex;

```

Table of Edge (Line)

Table of Edge (TOE) consists of three fields namely *Edge_No*, *StartVertex*, and *EndVertex*. *Edge_No* represents index of the edge. *StartVertex* and *EndVertex* are a vertex number, which is an index of the Table of Vertex. The record structure was arranged as shown below:

```

record {
    integer edge_number;
    integer startvertex, endvertex;
} edge;

```

Table of Face (Region)

Table of Face (TOF) consists of two fields namely *face_number* and *Vertex_list*. *Face_number* is an index of the table. A face is bounded by a series of vertex. *Vertex_list* is an array of integer that is used to keep the list of vertex numbers that bound the region. The vertex number is the number found in Table of Vertex. The record structure was arranged as shown below:

```

record {
    integer face_number;
    integer vertex_list[50];
} face;

```

4.3 Digitisation Process

Digitisation process is a pre-processing stage to digitise an image that is in the form of colour mode or grey level into a binary image. The process read the TIFF image, converted the original image into binary image, and kept the pixel values and their coordinates in the form of binary image file format or array of image as defined in Section 4.2. This section does not discuss the digitisation process in detail since it is a pre-requisite function in any image processing works.

4.4 The Enhanced Thinning Algorithm

This section describes in detail the enhanced thinning algorithm used in thinning the binary image of the line drawing. The discussion started with the basic thinning constraint used in the algorithm. The discussion then followed with the explanation on two stages required by the algorithm.

4.4.1 Basic Thinning Constraint

Pitas (1995) stated that the thinning algorithms must satisfy the following constraints:

1. In every iteration of thinning algorithm, connectivity must be maintained and border pixels must not be removed because this may cause discontinuities.
2. The thinning algorithm does not shorten the end of thinned shape limbs. Therefore, border pixels were not removed.

Figure 4.2 (a) and (b) show these constraints. Figure 4.2 (a) shows the removal process of the current pixel indicated by **1** (bold pixel) that can cause discontinuities. Figure 4.2 (b) shows the removal process of the current pixel indicated by the **1** (bold

pixel) that can shorten an object limb. Therefore, these constraints must be observed in preserving the drawing topology and line connectivity. The following sections discuss two stages required by the proposed algorithm.

0	0	0
1	1	0
0	0	1

(a)

0	0	0
1	1	0
0	0	0

(b)

Figure 4.2: Constrain of thinning algorithm

4.4.2 Stage I: Neighbourhood greater than 3

The first stage reads the binary image pixel by pixel from coordinate (0,0). Pixel 1 was replaced by pixel 0 when the total neighbourhood pixel of the current pixel was greater than 3. In this stage, the second constraint, shortening the shape limbs was ignored due to the assumption that the image was representing 3D object. Steps of the first stage are shown in Algorithm 4.1.

```

For every row
  For every column
    Select 3x3 windows at the current pixel
    Count the total number of pixel 1 of the neighbourhood, tot_3
    If tot_3 is greater than 3, replace the pixel 1 with number 0
  endfor
endfor

```

Algorithm 4.1: Thinning process of stage I

4.4.3 Stage II: Neighbourhood equal to 2 or 3

In the second stage, starting from coordinate (0,0) the algorithm read thinned binary image that had passed the first stage. The algorithm minimized the number of pixels by concentrating on pixel 1 of neighbourhood pixel in which the total was 2 and 3. The algorithm still preserved the topology of the drawing and met the first constraint to maintain the connectivity of the line. Steps of the second stage were shown in Algorithm 4.2.

```

For every row
  For every column
    Select 3x3 windows at the current pixel.
    Count the number of pixel 1 of the neighbourhood, tot_pixel.
    If tot_pixel is equal to 2, and match one of the templates in Figure 4.3
      Replace the pixel with number 0
    endif
    If tot_pixel is equal to 3, and match one of the templates in Figure 4.4
      Replace the pixel with number 0
    endif
  endfor
endfor

```

Algorithm 4.2: Thinning process of stage II

For pixel 1 of neighbourhood pixel in which the total number of pixel 1 was 2, Figure 4.3 showed all twelve cases that the current pixel had to be replaced with pixel 0. For example, in Figure 4.3 (a), the removal of the current pixel did not cause discontinuities but it does shorten an object limb. Therefore, the first constraint is satisfied.

For total pixel 1 of the neighbourhood, that was equal to 3, there were twelve cases to be considered as shown in Figure 4.4. Since the removal of the current pixel would not cause discontinuities it will shorten an object limb, the first constraint is satisfied. The final thinned binary image derived was used as input by the chain code algorithm to detect T-junction, corner, line and region.

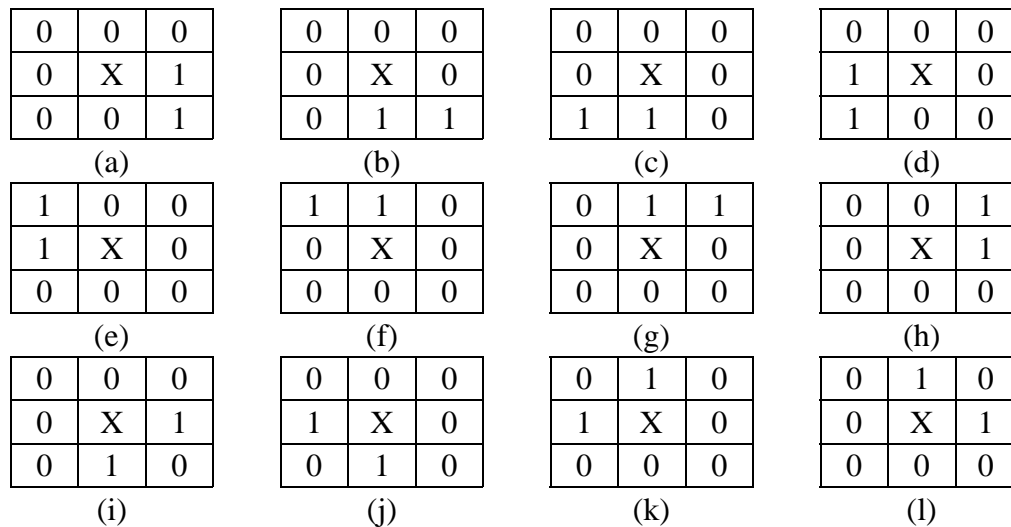


Figure 4.3: Twelve cases of total neighbourhood equal to 2

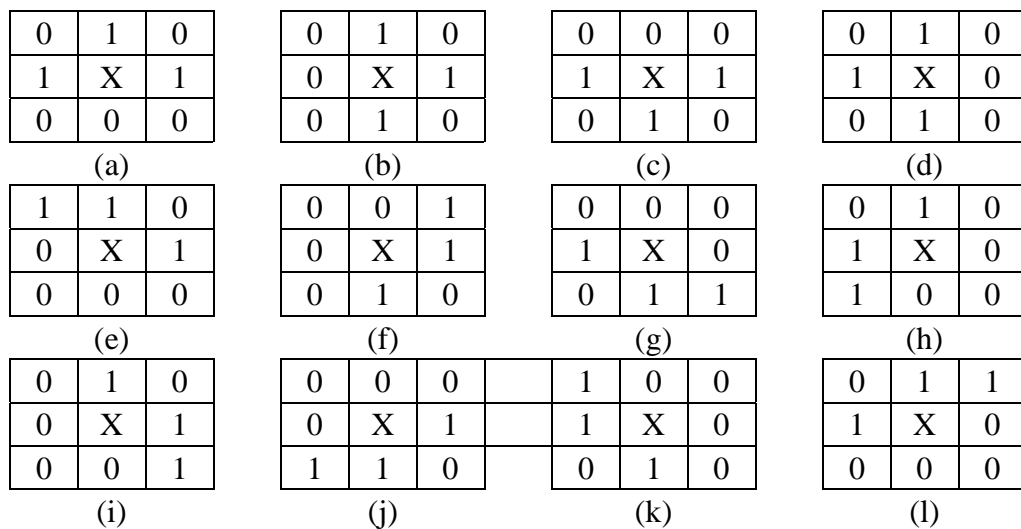


Figure 4.4: Twelve cases of total neighbourhood equal to 3

4.5 VCC Conversion Algorithm

Vertex chain code is different from Freeman chain code not only in the code scheme but also in the way the codes derived. FCC can only be interpreted after the thinned binary image produced but for VCC, one step before the chain code can be derived must be executed namely the cell forming process. Since this research only focused on rectangle cell, therefore the thinned binary image will only through the rectangular cell forming.

4.5.1 The Formation of Rectangle Cell

Each code 1 in the thinned binary image represents each form of the rectangle cell. The direction of code 1 that adjacent to to a code 1 leads to the formation of next rectangle. Figure 4.5 shows an example of the formation of rectangle cell from binary image. When each code 1 in the binary image is interpreted, a line drawing that consisted of rectangle cell will be created. Figure 4.6 shows example of a line drawing and the rectangle cell created that represents the line drawing. After the formation process completed, the new image will be used in deriving the vertex chain code.

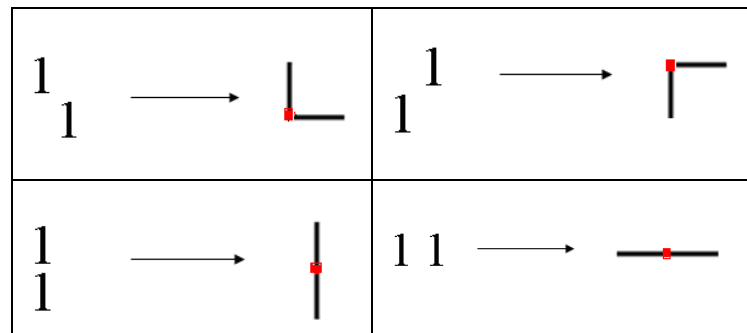


Figure 4.5: From Thinned Binary Image to Rectangle Cell of VCC

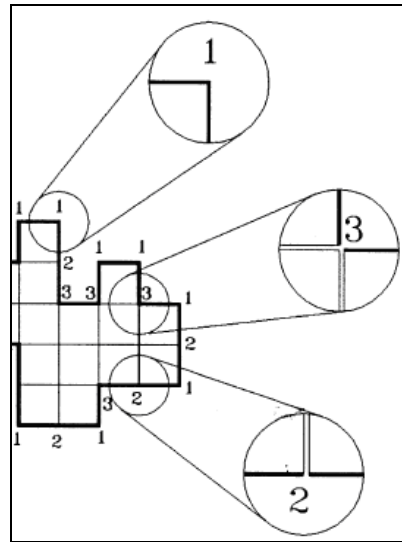


Figure 4.7: Example of VCC Rectangle Cell Code

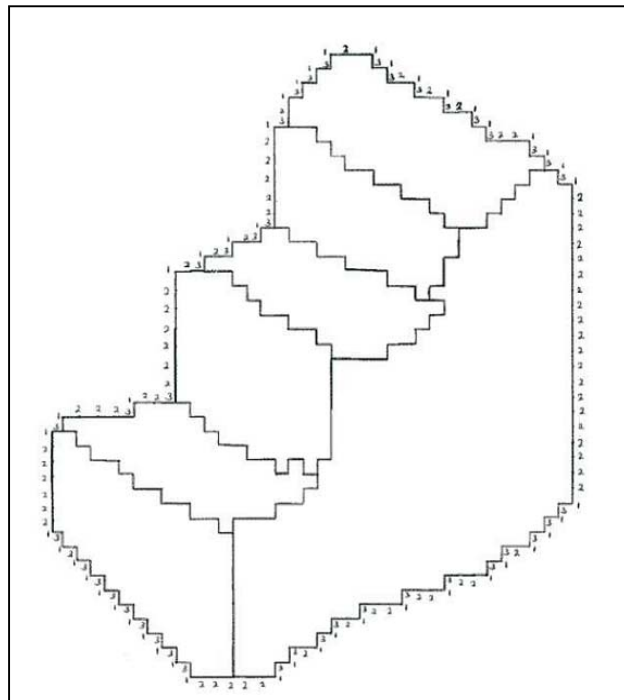


Figure 4.8 : Example of VCC code of the rectangle cell

The process of converting the thinned binary image into rectangle cell and to convert the rectangle cell into VCC code has been carried out manually due to the time constraint. Therefore, automation process of these two steps must be explored as no works has been done in this process yet.

4.6 FCC Feature Extraction: The Computational Model

This section presents the algorithm that is used in deriving the closed loop chain code series of boundary and internal line of a thinned binary image. The discussion is divided into the introduction to the closed loop chain code, algorithm to derive boundary and internal line series, and followed by description on the algorithm to detect T-junction, corner and line. The algorithm started with deriving boundary series and followed by internal series. Using these chain code series, T-junctions were detected and followed by corner. Lastly, lines were detected based on the chain code series and Table of Vertex derived.

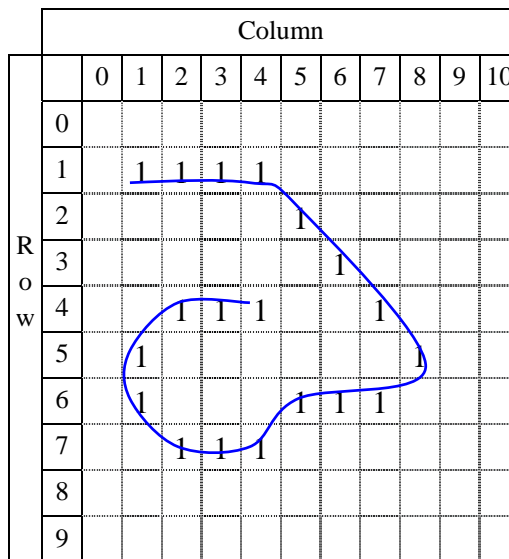
4.6.1 The Chain Code Series Derivation

Chain code is a list of code range from zero to seven in clockwise direction. The codes represent the direction of the next pixel from current pixel connected in 3x3 windows as shown in Figure 4.9.

	column-1	column	column+1
row-1	5	6	7
row	4	(current pixel)	0
row+1	3	2	1

Figure 4.9: Chain code & the location

There were eight locations of pixel surrounding the current pixel that are represented by row and column. For example, in code 0, the coordinate is (row, column+1). This code range is a modified Freeman chain code in terms of the direction of traverse. This algorithm can only be applied to single closed loop drawing. Each series are represented by the start coordinate (row, column) and followed by code series ranged from 0, 1, 2, 3, 4, 5, 6 and 7. The series must be ended with -1 to act as terminator of the series. Figure 4.10 shows an example of chain code and its irregular line.



Format: (<start_row>,<start_col>) <series of code>-1

Example: (1,1) 000111134434456700-1

Figure 4.10: Chain code format and its example

Boundary Series

This section explains the detail of algorithm in deriving the chain code of the boundary. For better understanding, the explanation is supported with an example of thinned binary image shown in Figure 4.11. Table 4.1 shows the start test location based on the current chain code. The derivation of series must be referred to Table 4.1 to ensure that the series represent a boundary. Algorithm 4.3 shows the algorithm to derive

the boundary chain code. To determine the start test location in the window, the current code needs to be checked. For the first pixel in the series, it is assumed that the previous pixel is from location 0 but the code is not added in the series. For the rest of the pixels, start test location is based on the current code.

		Column												
		0	1	2	3	4	5	6	7	8	9	10		
R o w	0													
	1					1	1							
	2				1			1						
	3			1			1		1					
	4		1			1				1				
	5		1		1				1		1			
	6			1		1	1	1			1			
	7			1						1				
	8				1	1	1	1	1					
	9													

Figure 4.11: Example of binary image

Table 4.1: Start test location (clockwise direction)

Current Code	0	1	2	3	4	5	6	7
Start Test Location	5	6	7	0	1	2	3	4

1. Starting from coordinate (0,0), locate the first pixel 1 in the image started from column and followed by row.
2. If found
 - Assign the coordinate as current_location.
 - Replace pixel with 2 to indicate that the coordinate is already traversed
 - Select 3x3 window that surrounds the pixel.
- else
 - Stop.
- endif
3. At current_location, starting at code location 0 and in clockwise direction, locate the first pixel 1.
4. If found,
 - If current_location is adjacent to the first pixel found, go to step 8.
 - Print the current_location in the series.
 - Add the current_code in the series.
 - Assign current_location with the new location.
 - Assign current_code with the new code.
- else
 - Go to Step 8.
- endif
5. At current_location, select 3x3 window that surrounds the pixel. Replace pixel with 2.
6. Referring to Table 4.1 and in clockwise direction, locate the first pixel 1.
7. If found,
 - Assign current_location with the new location.
 - Assign current_code with the new code.
 - Print/Add current_code in the series.
 - Repeat Step 5.
- else
 - Go to Step 8.
- endif
8. Add -1 in the series to indicate the termination of the series.

Algorithm 4.3: Chain code algorithm: Boundary and internal line

Figure 4.12 shows 3x3 windows of coordinate (2,6) of image in Figure 4.11. The previous connected pixel was from coordinate (1,5) and the code was 1. There are two pixels connected to the current pixel namely at locations 1 and 3. Based on Table 4.1, the start of search location for current_code 1 was at location 6 and in clockwise direction. Since there was no pixel at location 6, 7 and 0, the first pixel '1' was found at location 1. Therefore, the current_location was at coordinate (3,7) and the current_code was equal to 1.

		Column												
		0	1	2	3	4	5	6	7	8	9	10		
R o w	0													
	1					2	2							
	2				1		1	1						
	3			1			1		1					
	4		1			1					1			
	5		1		1					1		1		
	6			1		1	1	1				1		
	7			1							1			
	8				1	1	1	1	1					
	9													

Figure 4.12 : Example coordinate (2,6)

Figure 4.13 shows current_location at coordinate (6,2) of image in Figure 4.11 and the current_code is equal to 6 that was from coordinate (7,2). There were two pixels in the window located at coordinate (5,1) and coordinate (5,3) and code was 5 and 7 respectively. Based on Table 4.1, if current_code was equal to 6, the first start test location was at location 3. Clockwise direction started from location 3 gives location 5 as the next connected pixel. Therefore, current_location was changed to coordinate (5,1) and the current_code was changed from 6 to 5.

		Column										
		0	1	2	3	4	5	6	7	8	9	10
R o w	0											
	1					2	2					
	2				1			2				
	3			1			1		2			
	4		1			1				2		
	5		1	1					1		2	
	6		1	1		1	1	1			2	
	7			2							2	
	8				2	2	2	2	2			
	9											

Figure 4.13: Example coordinate (6,2)

Deriving the boundary chain code is terminated when the current pixel is the first pixel found in the array. Figure 4.14 shows pixel values after the boundary chain code derived. The pixels with number **2** indicate that the pixels are already traversed as a boundary pixel. The boundary chain code represented by curve in Figure 4.14 was given as (1,4) 0111123344445656777-1.

		Column										
		0	1	2	3	4	5	6	7	8	9	10
R o w	0											
	1					2	2					
	2			2				2				
	3		2			1			2			
	4		2			1				2		
	5		2		1				1		2	
	6		2			1	1	1			2	
	7			2							2	
	8				2	2	2	2	2			
	9											

Figure 4.14: Example of binary image

Internal Line Series

Derivation of internal line series is a continuation of the Section 4.5.2. Using the modified array of image shown in Figure 4.14, the search for of internal line chain code was proceeded based on algorithm in Algorithm 4.3. There are two differences between internal line chain codes compared to boundary line processing algorithm. First, the array of image for internal line is a modified array that passed after derivation of the boundary line. Second difference is in terms of how the traversal is terminated. The traversal of boundary line is terminated when pixel coordinate is back to the first pixel found while for internal line, the traversal is terminated when the first pixel was equal to 2 found. The similarity between internal and boundary line algorithm is that, for internal line, the first search pixel is the first pixel equal to 1 that is not traversed yet which is indicated by pixel value 1. Therefore, there may be more than one internal line chain code.

From the modified array of image in Figure 4.14, the traversal locates that coordinate (3,5), as shown in Figure 4.15, is the first pixel 1 found. Applying algorithm in Algorithm 4.3, the derivation of internal line is preceded. The next current_location is at coordinate (4,4).

		Column												
		0	1	2	3	4	5	6	7	8	9	10		
R o w	0													
	1					2	2							
	2				2	2	2							
	3			2		1	2		2					
	4		2			1	2			2				
	5		2		1				1		2			
	6			2		1	1	1			2			
	7			2							2			
	8				2	2	2	2	2					
	9													

Figure 4.15: Example coordinate (3,5)

Let us assume that `current_location` is at coordinate (5,3) as shown in Figure 4.16. There are two pixels 2 in the neighbouring window. Note that pixel 2 at coordinate (4,4) belonged to internal line. In contrast, pixel 2 at coordinate (6,2) belonged to the boundary. Since the `current_code` was 3 or from coordinate (4,4), therefore the first start test location was at location 0. Therefore, clockwise traverse gave location 1 as the first connected pixel. It should be noted that during clockwise checking, if pixel 2 was found then the search for the current chain code was terminated. Then, the search was preceded until no more pixels 1 in the thinned binary image. The internal chain code of Figure 4.7 indicated with curve in Figure 4.16 was given as (3,5) 331007-1.

		Column										
		0	1	2	3	4	5	6	7	8	9	10
R o w	0											
	1					2	2					
	2				2			2				
	3			2			2		2			
	4		2	2	2	2	2			2		
	5		2	2	1	1	1		1	2		
	6			2	2	1	1	1			2	
	7			2						2		
	8				2	2	2	2	2			
	9											

Figure 4.16: Example coordinate (5,3)

4.6.2 T-junction, Corner and Line Detection

T-junction Detection

Intersection of two lines creates a junction. This geometrical fact was applied in this section. There were two types of T-junction; one was located on the boundary line and the other on the internal line. The T-junction on boundary was derived first and then followed by junction on internal line. The Table of Vertex mentioned in Section 4.2.4

was referred to in this discussion. Figure 4.17 shows the difference between T-junction on boundary line (indicated by blue curve) and internal lines (indicated by red curve).

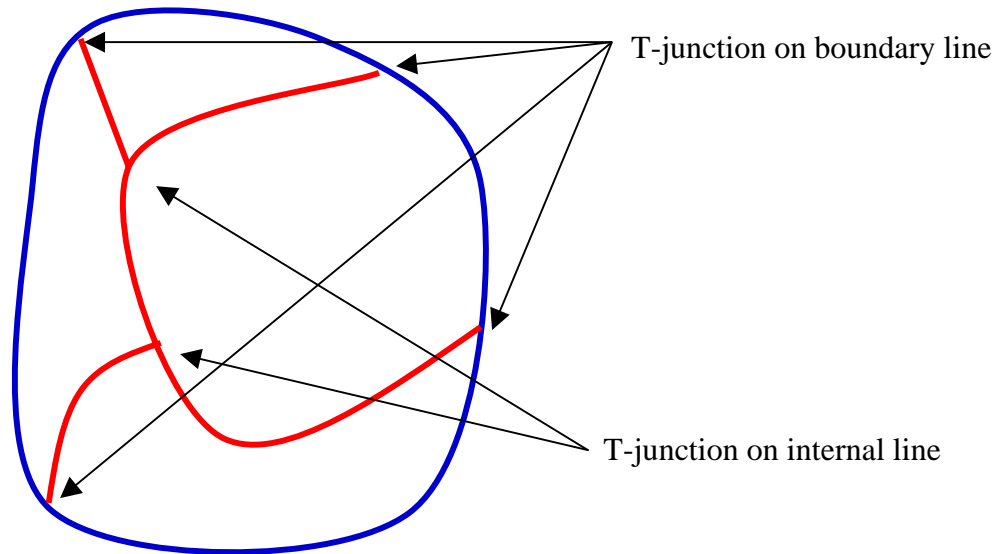


Figure 4.17: T-junction

T-junction on boundary line

For every coordinate in the boundary chain code, they were compared to the first and last coordinate of all the internal line chain code. If current boundary coordinate were adjacent to those records, then the boundary coordinate were assigned as T-junction of boundary line.

T-junction on internal line

Next, by comparing all coordinates in the current internal line chain code with the first and the last coordinates of other internal lines derives another type of T-junction was derived. These coordinates were compared to the first and last records of the rest of internal line chain code. If adjacent coordinate was found, then the T-junction

of internal line was assigned to the coordinate. The comparison was done on all internal lines. Table of Vertex was updated when the process completed.

Corner Detection

This section explains in detail the corner detection algorithm. The Table of Vertex mentioned in Section 4.2.4 is referred in this discussion. The boundary and internal line chain code were still referred in deriving corner. For every chain code series, the coordinates of consecutive T-junctions on the chain code were identified. Codes between these coordinates were copied into a temporary chain code. For boundary chain code, since the first pixel was not necessarily a T-junction, the last record of the list was linked to the first record of the list. The algorithm explained below was based on the temporary chain code which consists of *nine* codes between two coordinates of T-junction.

Examples were used to assist the explanation. In general, four phases were involved. The first phase states that on the series of chain code and their starting coordinate, the algorithm segmented the chain code based on a pre-defined *length parameter*. Secondly, values of all rows and columns associated with each code in the segment were derived. Thirdly, these values were analysed and a *weighted parameter* was produced. Finally, the weighted parameter produced was then compared to the pre-defined *corner parameter* to determine the existence of a corner at the testing coordinate. The detailed steps involved in detecting a corner are as follows:

1. Set the values of length parameter, n where $n = 3, 5, 7, 9, 11, \dots$
2. Set the corner parameter, t where $t > 3$.
3. Create a segment consisted of n chain codes where the testing location is at location $i = \frac{n-1}{2}$.

4. From the testing location $\frac{n-1}{2}$, and for $i = \frac{n+1}{2}, \dots, n-1$, calculate row and column values of each code using equation 4.1 and 4.2 respectively.

$$Row_i = \begin{cases} Row_{i-1} - 1 & \text{for } code_{i-1} = 5,6,7 \\ Row_{i-1} & \text{for } code_{i-1} = 0,4 \\ Row_{i-1} + 1 & \text{for } code_{i-1} = 1,2,3 \end{cases} \quad (4.1)$$

$$Col_i = \begin{cases} Col_{i-1} - 1 & \text{for } code_{i-1} = 3,4,5 \\ Col_{i-1} & \text{for } code_{i-1} = 2,6 \\ Col_{i-1} + 1 & \text{for } code_{i-1} = 1,0,7 \end{cases} \quad (4.2)$$

Equations 4.1 and 4.2 are summarized in Table 4.2.

Table 4.2: Row and column values for $i = \frac{n+1}{2}, \dots, n-1$

	col _{i-1}	col _i	col _{i+1}
row _{i-1}	5	6	7
row _i	4	code _{i-1}	0
row _{i+1}	3	2	1

5. From the testing location $\frac{n-1}{2}$, and for $i = \frac{n-3}{2}, \dots, 0$ calculate row and column values of each code using equations 4.3 and 4.4 respectively.

$$Row_i = \begin{cases} Row_{i+1} - 1 & \text{for } code_i = 1,2,3 \\ Row_{i+1} & \text{for } code_i = 0,4 \\ Row_{i+1} + 1 & \text{for } code_i = 5,6,7 \end{cases} \quad (4.3)$$

$$Col_i = \begin{cases} Col_{i+1} - 1 & \text{for } code_i = 1,0,7 \\ Col_{i+1} & \text{for } code_i = 2,6 \\ Col_{i+1} + 1 & \text{for } code_i = 3,4,5 \end{cases} \quad (4.4)$$

Equation 4.3 and 4.4 are summarized in Table 4.3.

Table 4.3: Row and column values for $i = \frac{n-3}{2}, \dots, 0$

	col _{i-1}	col _i	col _{i+1}
row _{i-1}	1	2	3
row _i	0	code _i	4
row _{i+1}	7	6	5

6. Using equation 4.5, calculate the sum of difference of row.

$$DiffRow = R_1 - R_2$$

where

$$R_1 = \sum_{i=0}^{\frac{n-3}{2}} [Row_{i+1} - Row_i]$$

$$R_2 = \sum_{i=\frac{n-1}{2}}^{n-2} [Row_{i+1} - Row_i]$$

(4.5)

Row_i is row value at location i

n is the length parameter

7. Using equation 4.6, calculate the sum of difference of column.

$$DiffCol = C_1 - C_2$$

where

$$C_1 = \sum_{i=0}^{\frac{n-3}{2}} [Col_{i+1} - Col_i]$$

$$C_2 = \sum_{i=\frac{n-1}{2}}^{n-2} [Col_{i+1} - Col_i]$$

(4.6)

Col_i is column value at location i

n is length parameter

8. Using equation 4.7, calculate the total of the absolute value of the row differences and absolute value of the column differences.

$$Total = |DiffRow| + |DiffCol|$$

(4.7)

9. If weighted parameter, *Total* is greater than threshold value, *t*, then the testing location is a corner; otherwise the current pixel is not a corner.

Figure 4.18 shows an example of thinned binary image with *n* set to 11. There were 14 chain codes between the coordinates namely 0021212343544-1 and the starting junction is (1,2), therefore the process has to be repeated four ($14-n+1=14-11+1=4$) times. The process created three segments which consisted of *n* codes and *n* values of rows and columns. For the first segment, the start, the last and the testing coordinates were at coordinate (1,2), (8,3) and (4,5) respectively. For the last or the third segment, the start, the last and the testing coordinates were at coordinate (2,4), (7,5), and (7,0) respectively.

		Column												
		0	1	2	3	4	5	6	7	8	9			
R o w	0													
	1			1	1	1								
	2					1								
	3						1							
	4						1							
	5							1						
	6								1					
	7	1	1	1		1	1							
	8				1									
	9													

Figure 4.18: A corner at coordinate (6,6)

For example in Figure 4.18, the existence of corner at coordinate (6,6) was going to be determined. Therefore, given $n = 11$, a segment of series of chain code where the start, the last and the testing coordinates were at coordinates (1,4), (7,1) and (6,6) respectively was derived as 21212343544. Row and column values of each code in the series except at the testing coordinate where $i = 5$ were calculated. For instance, at

location $i = 3$, the chain code of the location, $code_3$ was equal to 1. Using equation 4.3 and 4.4, the row and column values when $i = 3$, Row_i and Col_i , are given as:

$$Row_3 = Row_4 - 1 = 5 - 1 = 4$$

$$Col_3 = Col_4 - 1 = 6 - 1 = 5$$

The row and column values of all locations in the segment were calculated using equation 4.1, 4.2, 4.3 and 4.4, and the results are shown in Table 4.4.

Table 4.4: Row and column values

i	$Code_i$	Row_i	Col_i
0	2	1	4
1	1	2	4
2	2	3	5
3	1	4	5
4	2	5	6
5	3	6	6
6	4	7	5
7	3	7	4
8	5	8	3
9	4	7	2
10	4	7	1

← Testing Coordinate (6,6)

Based on the values in Table 4.4 and by using equation 4.5, the sum of the difference of rows is calculated and the results are shown in Table 4.5.

Table 4.5: The Sum of difference of rows

i	Row_{i+1}	Row_i	Diff
0	2	1	1
1	3	2	1
2	4	3	1
3	5	4	1
4	6	5	1
$R_1 = \sum Diff = 5$			

i	Row_{i+1}	Row_i	Diff
5	7	6	1
6	7	7	0
7	8	7	0
8	7	8	-1
9	7	7	0
$R_2 = \sum Diff = 0$			

Based on the values in Table 4.4 and using equation 4.6, the sum of the difference of columns was calculated and the results are shown in Table 4.6.

Table 4.6: The Sum of difference of columns

i	Col_{i+1}	Col_i	Diff
0	4	4	0
1	5	4	1
2	5	5	0
3	6	5	1
4	6	6	0
$C_1 = \sum Diff = 2$			

i	Col_{i+1}	Col_i	Diff
5	5	6	-1
6	6	5	1
7	3	4	-1
8	2	3	-1
9	1	2	-1
$C_2 = \sum Diff = -3$			

The total of the row and column differences were calculated as shown in Table 4.7.

Table 4.7: Total of row and column differences

$DiffRow$	$= R_1 - R_2 = 5 - 0 = 5$
$DiffCol$	$= C_1 - C_2 = 2 - (-3) = 5$
$Total$	$= DiffRow + DiffCol $
	$= 5 + 5 = 10$

Since the weighted parameter, $Total$ is greater than the corner parameter, 3, then the location at row 6 and column 6 is a corner. It should be noted that for any irregular line, there was a possibility of two consecutive corners that were adjacent derived from the calculation. Verification to determine the final corner between these two corners was done by taking the newest corner that was adjacent to the previous corner.

Figure 4.19 shows another example of a chain code started at coordinate (0,0) with the series of chain code as 100171700. Let us determine the existence of corner at coordinate (2,4). Given $n = 9$, and the start, the last and the testing coordinate were at (0,0) and (1,8) and (2,4) respectively, the segment from coordinate (2,4) was derived as 100171700. Using equation 4.1 and 4.2, the row and column values of codes for $i = 5, 6, 7, 8$ were derived. Next, using the equation 4.3 and 4.4, the row and column values of

the code for $i = 3, 2, 1, 0$ are derived. Table 4.8 shows the row and column values derived for each code.

		Column									
		0	1	2	3	4	5	6	7	8	9
R o w	0	1									
	1		1	1	1		1		1	1	1
	2					1		1			
	3										

Figure 4.19: No corner at coordinate (2,4)

Table 4.8: Row and column values

i	$Code_i$	Row_i	Col_i	
0	1	0	0	
1	0	1	1	
2	0	1	2	
3	1	1	3	
4	7	2	4	← Testing Coordinate (2,4)
5	1	1	5	
6	7	2	6	
7	0	1	7	
8	0	1	8	

Based on the values in Table 4.8 above and by using equation 4.5, the sum of the difference of rows was calculated and the results are shown in Table 4.9.

Table 4.9: The Sum of difference of rows

i	Row_{i+1}	Row_i	Diff
0	1	0	1
1	1	1	0
2	2	1	1
3	2	2	0
$R_1 = \sum Diff = 2$			

i	Row_{i+1}	Row_i	Diff
4	1	2	-1
5	2	1	1
6	1	2	-1
7	1	1	0
$R_2 = \sum Diff = 1$			

Based on the values in Table 4.8 and by using equation 4.6, the sum of the difference of columns was calculated and the results are shown in Table 4.10.

Table 4.10: The Sum of difference of columns

i	Col_{i+1}	Col_i	Diff
0	1	0	1
1	2	1	1
2	3	2	1
3	4	3	1
$C_1 = \sum Diff = 4$			

i	Col_{i+1}	Col_i	Diff
4	5	4	1
5	6	5	1
6	7	6	1
7	8	7	1
$C_2 = \sum Diff = 4$			

The total of the row and column differences was calculated as shown in Table 4.11.

Table 4.11: Total of row and column differences

$DiffRow$	$= R_1 - R_2 = 2 - 1 = 1$
$DiffCol$	$= C_1 - C_2 = 4 - 4 = 0$
$Total$	$= DiffRow + DiffCol $
	$= 1 + 0 = 1$

Since the weighted parameter, $Total$ is less than the corner parameter, 3, then the coordinate at row 2 and column 4 was not a corner. Note that the length and corner parameter set in this thesis was 9 and 3 respectively. These values were taken from experimentation and were suitable for image sized 100 x 100 and 50 x 50 pixels.

Line Detection

This section discusses the derivation of lines. The discussion was boundary and internal line chain code series, and associated with the Table of Vertex derived in Section 4.5.4 and Section 4.5.5.

For boundary chain code, coordinate for each record in the chain was compared to the row and column in Table of Vertex. The first coordinate that matched the coordinate in Table of Vertex was assigned as start junction of the current line. After the first junction was founded, the next junction in the list was searched. The next junction found acted as the current line and start of the next line. The chain code was continuously traversed and every junction founded was assigned as the end and start junction of the current line and next line respectively. When the list terminator was found, and then the first junction found in the list was assigned as end of the current line and this indicated the end of line search of boundary chain code.

The next step was to find line of internal line series. There are two differences between line searching on boundary and internal line chain code. First, coordinate of junction in boundary line can be found exactly as in the Table of Vertex while the coordinate of junction in internal line may be adjacent to the coordinate in the table. Therefore, to find the junction in the internal line, checking must be done to find coordinate that was adjacent to coordinate in the table. Secondly, the last coordinate in internal line chain code was the end of current line.

For each internal line, start coordinate and end coordinate of the list was a start and end of the first and last line in the list. The junction found between these junctions would be the end and the start junction of the current line and the next line in the list respectively. The lines found were kept in Table of Line.

4.6.3 Region Detection

This section discusses the algorithm to detect region based on region chain code series. The algorithm started with the creation of region chain code series and followed by the algorithm to detect the region.

Region Series

This section discusses how to derive region chain code series. The derivation reinitialised the thinned binary image so that it contained only pixel 1 and 0 as the original thinned image. There were three differences between the searching of boundary and internal line chain code and region chain code. Firstly, region series uses different table as start test location. Table 4.12 shows the current code and the start test location. Secondly, direction of pixel searching for region series is in anti-clockwise direction. Thirdly, there should be no replacement of pixel from 1 to 2 since a coordinate can be shared with two regions.

Table 4.12: Start test location (anti-clockwise direction)

Current Code	0	1	2	3	4	5	6	7
Start Test Location	3	4	5	6	7	0	1	2

The process of deriving region series needs to refer to junctions that exist along the boundary in Table of Vertex. In this discussion, junction of boundary line is assumed already derived. The discussion of algorithm to derive junction along the boundary is given in Section 4.5.4. Image in Figure 4.11 is used as a basis in this discussion. In the image, there are three T-junctions along the boundary namely at coordinate (2,6), (4,8) and (6,2). The location of this junction is highlighted in Figure 4.20. Since there are three T-junctions, there are three regions in the image. Algorithm 4.4 shows the algorithm of region detection.

		Column										
		0	1	2	3	4	5	6	7	8	9	10
R o w	0											
	1					1	1					
	2				1			1				
	3			1			1		1			
	4		1			1				1		
	5		1		1				1		1	
	6			1		1	1	1			1	
	7			1						1		
	8				1	1	1	1	1			
	9											

Figure 4.20: Example of binary image (indicated with T-junction)

1. For every T-junction of boundary line, add region chain code with the boundary chain code from the current_T_junction till the next_T_junction. Replace the traversed pixel with 2. Print the current_T_junction as the start of the region series.
2. Assign current_location equal to location of the next_T_junction.
3. At current_location, select 3x3 window that surrounding the pixel.
4. Referring to Table 4.12 and in anti-clockwise direction, locate the first pixel=1.
5. If found,
 - Assign current_location with the new location.
 - Assign current_code with the new code.
 - Print/Add current_code in the series.
 - Repeat Step 3.
- else
 - Go to Step 6
6. Add -1 in the series to indicate the termination of the series.
7. Repeat Step 1 for the next T-junction of boundary line.

Algorithm 4.4: Chain code algorithm: Region line

Based on Figure 4.21, starting from the code of the first T-junction on boundary located at coordinate (2,6), and the chain codes of the shaded box were copied into region code until next T-junction at coordinate (4,8). The chain code series copied was (2,6) 11. At coordinate (4,8), the current_code was 1. Using Table 4.12, the start test location was 4 and moved in anti-clockwise direction. The testing gave location 3 as the next connected pixel. The process continued until the search pixel returned to the start pixel at coordinate (2,6) and a first region chain code of the image was created. Figure 4.21 showed the traverse direction of the first region. The complete region series of the first region was (2,6) 113344577-1

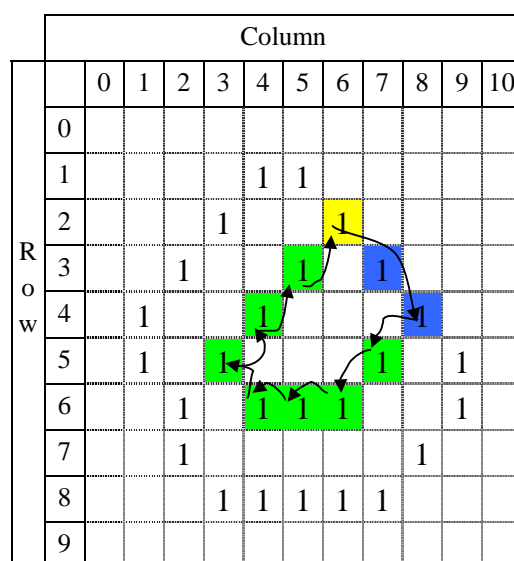


Figure 4.21: Direction of chain code for the first region

Figure 4.22 shows directions of the code to produce the second region. The current T-junction was at coordinate (4,8). From the junction, the next T-junction on the boundary was at coordinate (6,2) and thus gave the chain code as (4,8) 1233444456. From the coordinate (6,2) and using Table 4.12 to determine the next pixel, the second region was found when the traversal return to coordinate (4,8) giving the complete series of the second region as (4,8) 123344445671007-1.

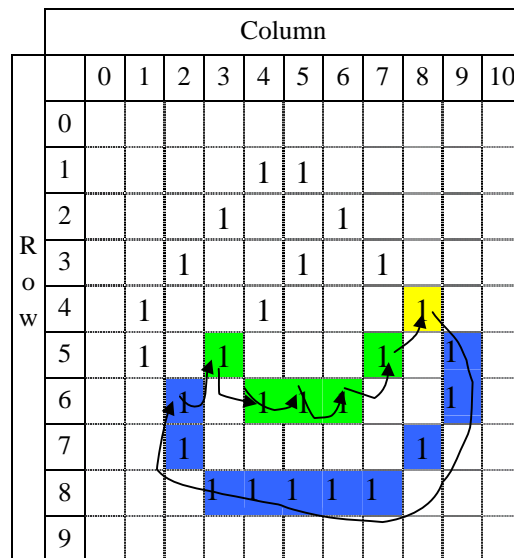


Figure 4.22: Direction of chain code for the second region

Figure 4.23 shows directions of the chain code to produce the third region. The current T-junction was at coordinate (6,2). From the junction, the next T-junction on the boundary was at coordinate (2,6) giving series (6,2) 5677701. From the coordinate (2,6) and using Table 4.12 to determine the next pixel, the third region was found when the traversal returns to coordinate (6,2) and gave complete series (6,2) 5677701333-1. The total number of region of Figure 4.11 was three.

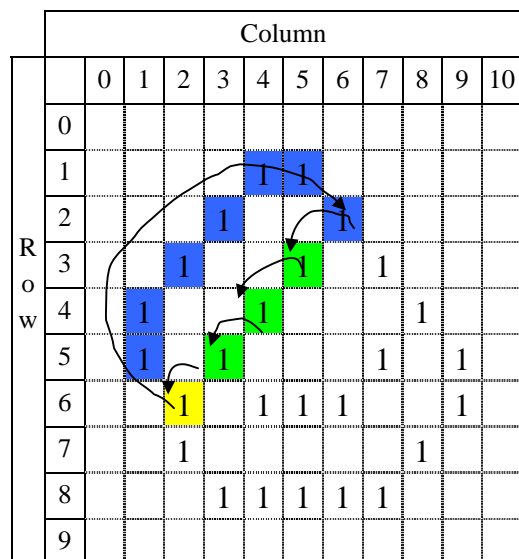


Figure 4.23: Direction of chain code for the third region

Region Creation

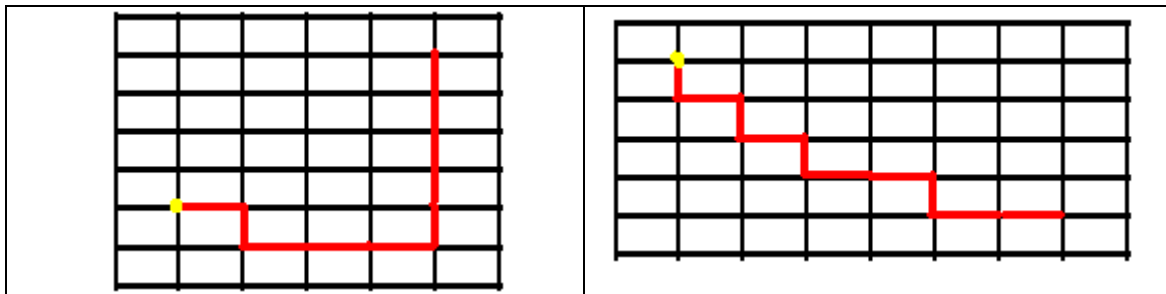
Given region chain code series, regions were detected. The Table of Face mentioned in Section 4.2.6 was referred in this explanation. There was more than one region that could be derived from an image. A region was made of series of junctions. The purpose of this algorithm was to create a table that was made of series of junction that existed in Table of Face. Therefore, the coordinate of codes in the region chain code series belongs to one of the junctions in the table. Comparison is made to every coordinate in the region chain code series with the Table of Vertex. If the comparison matches then the junction is added into the list of junction that bound the region. The iteration is repeated for every region chain code series. Algorithm 4.5 shows the algorithm to create region.

```
for every region chain code series
    for every coordinate in the chain code series
        if the coordinate match with the coordinate in junction table, add the
            junction in the list of junction that bound the region.
        endif
    endfor
endfor
```

Algorithm 4.5: Region creation algorithm

4.7 VCC Corner Detection: The Neural Network Classifier

This section presents the development of VCC neural network corner detection classifier. The input of the classifier is the VCC code consisted of series of code 1, 2 and 3. Pole of corner and not corner must be determined first. Series of code that represented corner and not corner are determined based on actual translation of the line drawing. Figure 4.24 (a) and (b) shows pattern and code translation of the code to determine the existence of corner. Figure 4.25 shows pole of VCC that is corner and Figure 4.26 is the opposite.



Kod: 132232222

Figure 4.24 (a): Example of pole and Corner Detection

Kod: 313132132

Figure 4.24 (b): Example of Pole and No Corner Detection

1	1	2	2	2	2	2	2	1	3	1	2	1	3	1	3	1	3	2	2
3	2	2	2	2	2	3	1	2	1	3	2	3	1	3	1	3	1	2	2
2	1	2	2	2	2	1	3	2	2	1	2	1	3	2	2	2	2	2	2
3	3	3	1	2	2	3	1	1	3	3	2	3	1	1	3	2	2	2	2
1	1	1	3	1	3	1	3	2	1	2	1	2	2	2	3	1	1	3	3
3	2	3	1	3	1	2	2	3	3	3	3	3	1	3	1	1	3	3	1
2	2	2	2	1	3	2	2	1	2	1	1	1	3	1	3	2	2	1	3
2	2	3	1	2	2	3	1	3	2	2	2	2	2	2	2	2	2	2	2
2	2	1	3	1	3	1	3	1	2	2	3	3	1	3	1	2	2	1	3
1	3	2	1	2	2	2	1	2	3	2	2	2	1	3	1	1	3	2	2
3	1	2	3	2	2	2	3	3	1	2	2	2	2	1	3	2	2	2	2
1	3	2	1	2	2	2	1	1	3	2	2	2	1	3	2	1	3	1	3
2	2	2	3	2	1	3	3	3	1	2	2	2	2	3	1	3	1	3	1
3	1	3	1	1	3	1	2	1	2	1	1	1	3	2	2	1	1	2	3
1	3	3	2	3	1	3	3	2	1	2	2	2	2	1	3	2	1	1	3
2	2	1	1	1	2	2	1	2	3	2	2	2	2	2	3	2	2	1	3
2	2	2	3	3	2	2	3	3	2	2	2	2	2	2	2	2	1	3	2
2	2	2	1	1	2	2	1	1	2	2	2	2	2	2	2	2	2	1	3

Figure 4.25: Chain Code Pattern Pole to Determine the Corner for Classifier

Training Input

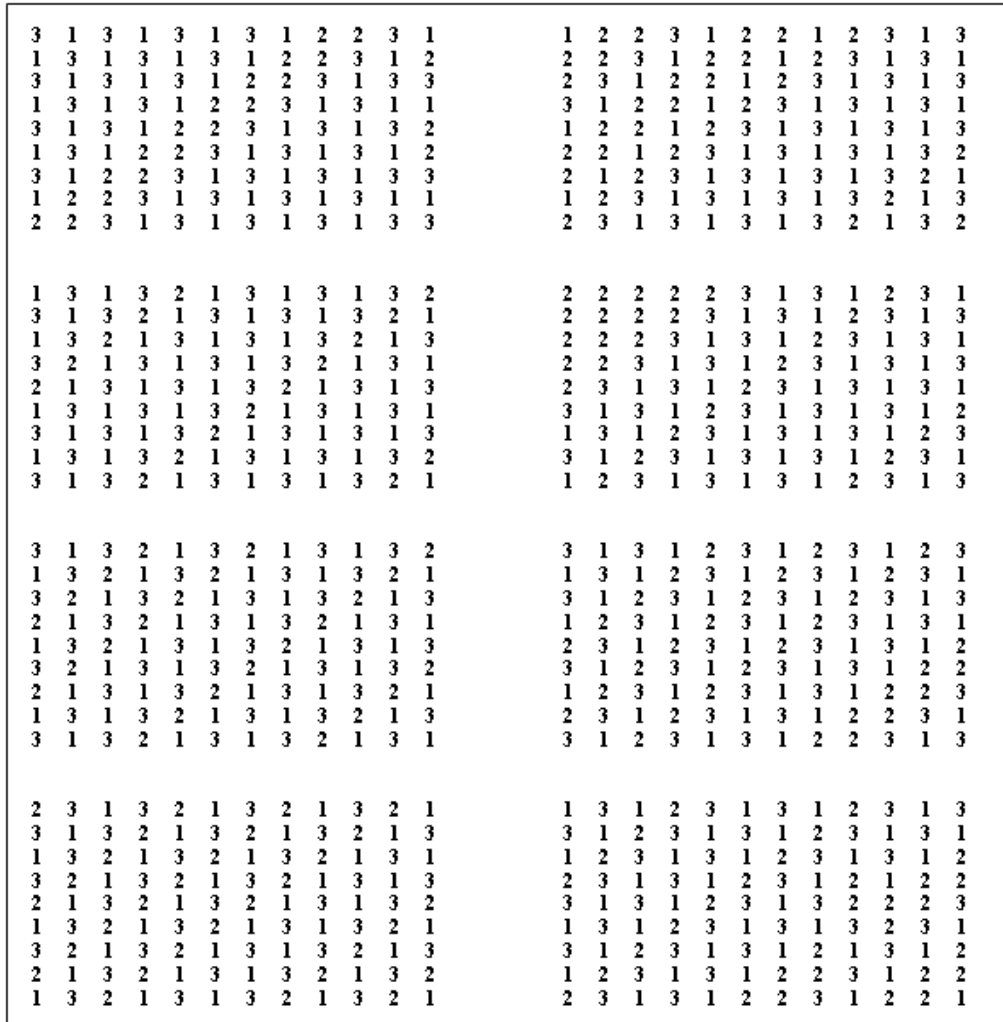


Figure 4.26: Chain Code Pattern Pole to Determine the Not Corner for Classifier Training Input

In the development of the VCC classifier, there are 363 network model has been trained to determine the best model. All these model used different parameters to obtain the best output. Only one model has been chosen to the classifier. Figure 4.27 shows the iteration of the best model to train the network. Figure 4.28 shows the graph of the training process.

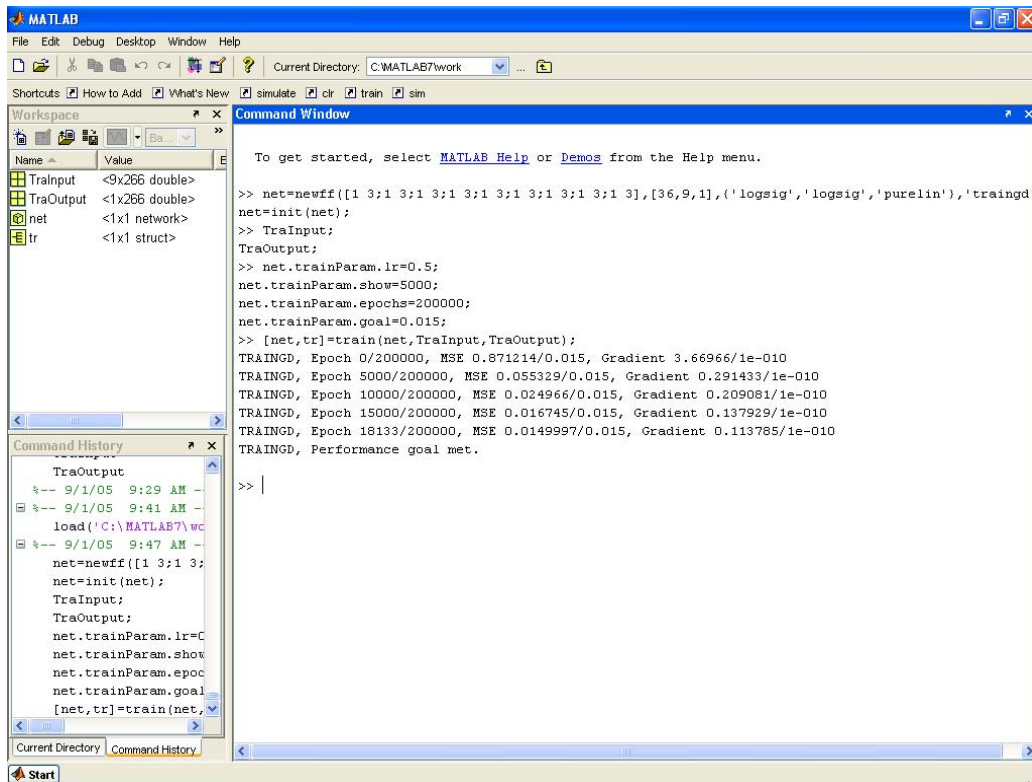


Figure 4.27: Iteration during the training process

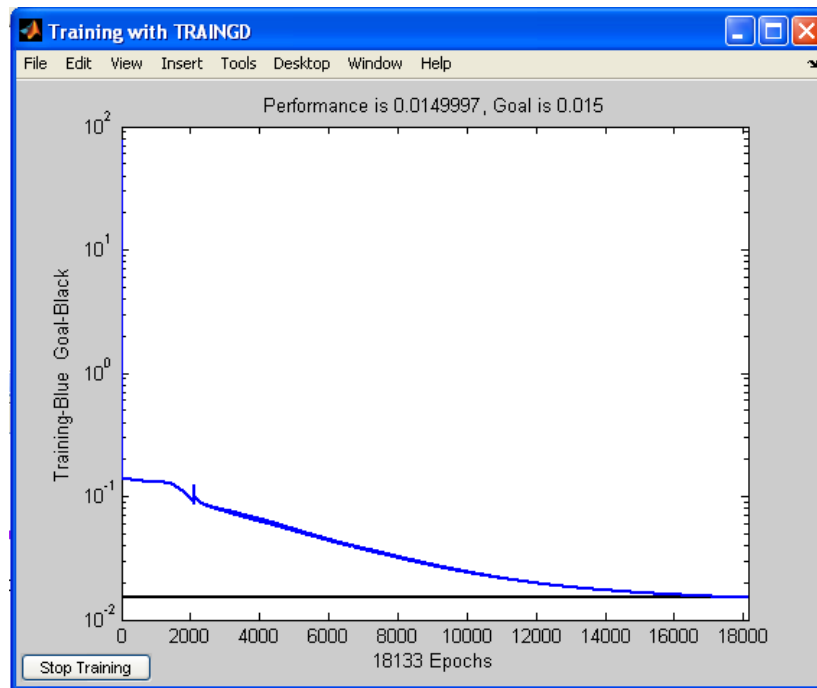


Figure 4.28: The graph in train the network

Out of 363 trained network model, three best models and training function has been chosen. Table 4.13 shows the best model of the training function. The parameter of the table are:

- i. **Training Function:** - The training function used are Traingdx, Traingd atau Traingdm
- ii. **Input** - Input is set to 9
- iii. **Hidden 1 dan 2** – Hidden node for level 1 and use sigmoid function (logsig) at hidden node 2.
- iv. **Output** - Output of the network model and is set to 1 and use linear function (purelin)
- v. **α** – learning rate
- vi. **β** – momentum rate
- vii. **Goal** – Target value
- viii. **Iteration** – No of iteration
- ix. **MSE Output** –MSE value produced from the network

Nevertheless, in order to obtain the VCC, only one model is chosen from the three models. The second model has been chosen because it has Traingd training function. The model has accuracy 92.25% in determining corner. This selection is also based on the smallest MSE value produced by the model compared to other models. Table 4.13 shows the parameters used.

Table 4.13: The best model of VCC with the parameter

Fungsi Latihan	Input	Hidden 1	Hidden 2	Output	α	β	Goal	Lelaran	Ketepatan (%)	MSE Output
Traingdx	9	36 Logsig	9 Logsig	1 Purelin	0.1	0.1	0.015	10576	91.51	0.07468
Traingd	9	36 Logsig	9 Logsig	1 Purelin	0.5	nil	0.015	8043	92.25	0.0725
Traingdm	9	27 Logsig	9 Logsig	1 Purelin	0.3	0.25	0.015	15772	92.25	0.08502

4.8 Summary

The components of the proposed algorithm to interpret line drawing that represents a 3D object have been explained in this chapter. The explanation began with introduction on the creation of data structure to represent the data and the digitisation process. Next, explanation is given on the detail two algorithms used in 2D feature extraction that are enhanced thinning algorithm and closed loop chain code algorithm. The last part of the discussion was explanation on steps taken in process of guessing the depth values of junctions. Two steps involved were deriving the image regularities of the linear system, and solving the linear system using Matlab.

CHAPTER V

RESULTS AND DISCUSSION

5.1 Introduction

Results of processing a sketch of a cube by the proposed algorithm are presented, showing a cube that has been used by Grimstead (1997). The processing of the sketch of cube is presented in detail. The experiment has been carried out in four different ways. They are the sketching process, the development of C functions, manual calculation, and creation of Matlab functions.

Firstly, the sketch was drawn using Adobe Photoshop and the image was saved as TIFF format as grey level image. Figure 5.1 shows the Adobe interface with the cube that has been drawn using the software.

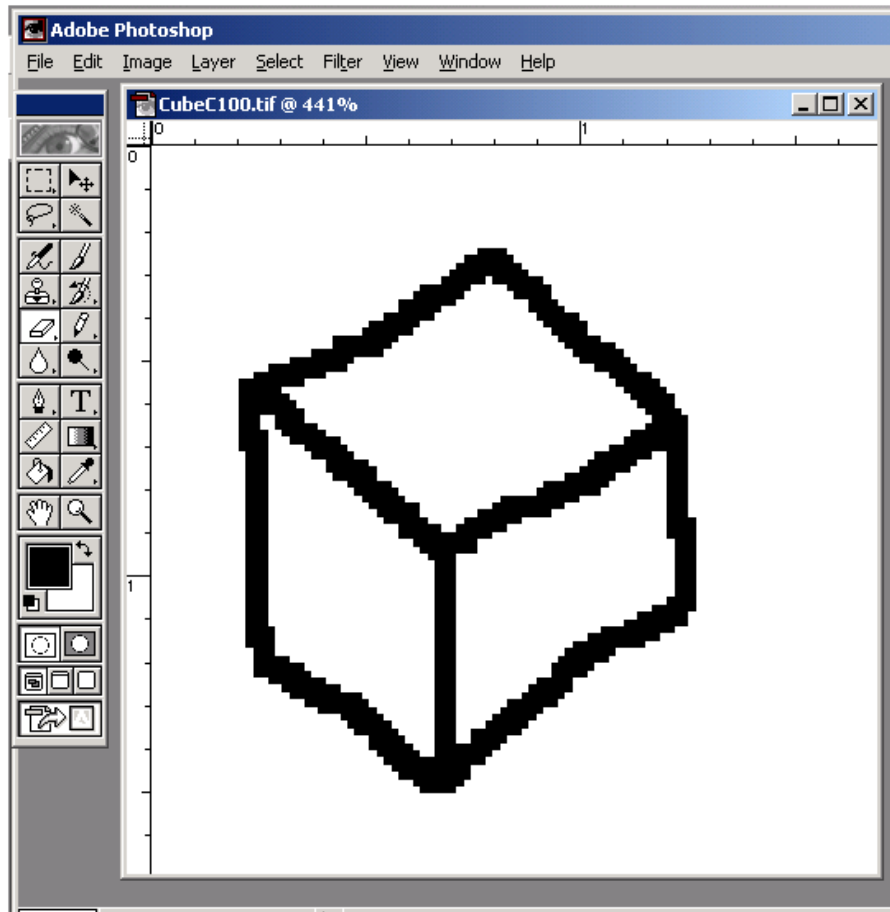


Figure 5.1: The Adobe interface and the cube

Secondly, the image was then digitised and thinned by the thinning module to produce the thinned binary image. The thinned binary image was kept as an array of image and used by the chain code module to produce the two-dimensional boundary representation data structure. Figure 5.2 shows the interface of the Feature Extraction Module that was used to select input image, digitise the image, thin the binary image, produce chain codes, interpret the image, and produce the two-dimensional data. The functions were developed using C functions but the interface was developed using Visual Basic 6.0.

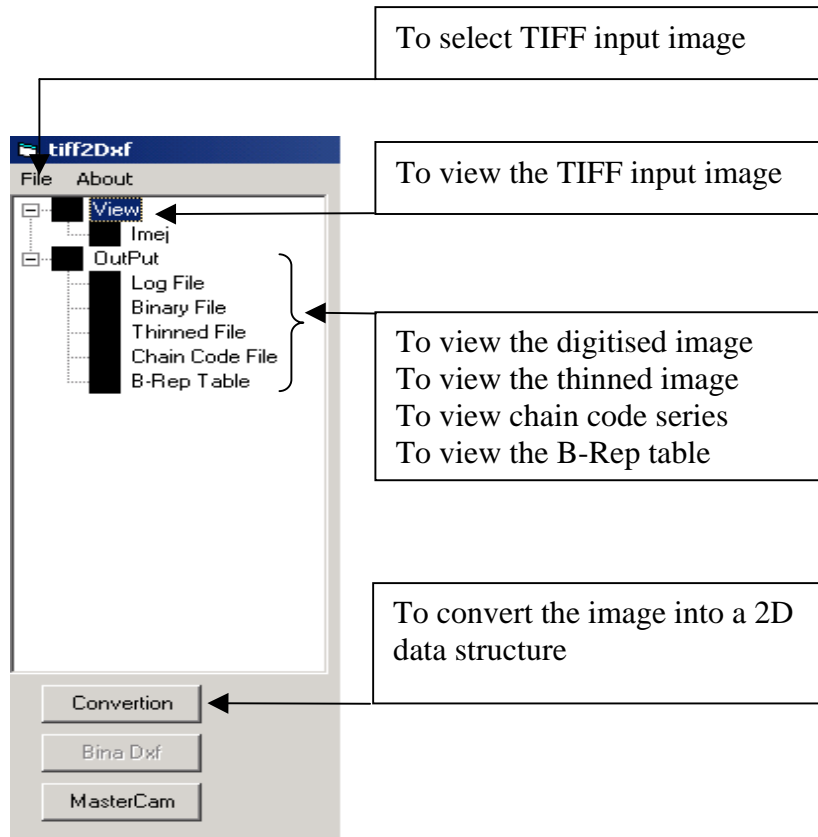


Figure 5.2: The interface of Feature Extraction Module

Thirdly, a set of equations of the spatial structure and the gradient estimates of each region were derived. The set of equations were then organized in the form of linear system that acted as input to the Matlab input file.

5.2 The Digitisation of Input Object

Figure 5.4 shows a cube with 7 vertices, 9 edges and 3 faces, which has been tested on the proposed algorithm. The sketch has been refined by removing all unwanted pixels using Adobe Photoshop and saved in TIFF format.

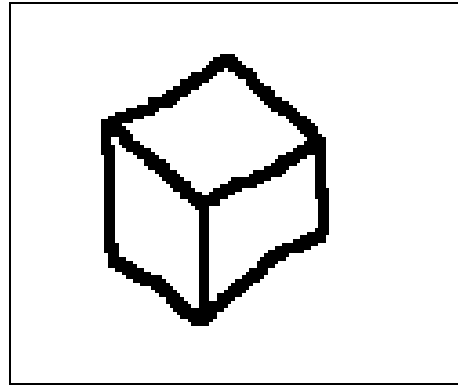


Figure 5.3: A cube

The TIFF image file was then scanned and converted into binary image. Figure 5.5 shows the cube image retrieved by the Feature Extraction Module. The View menu function was used to view the image. **APPENDIX A1** shows part of the grey level values of the image as extracted by the digitisation function. The grey level image was then converted into binary image based on the pre-defined threshold value that is 200.

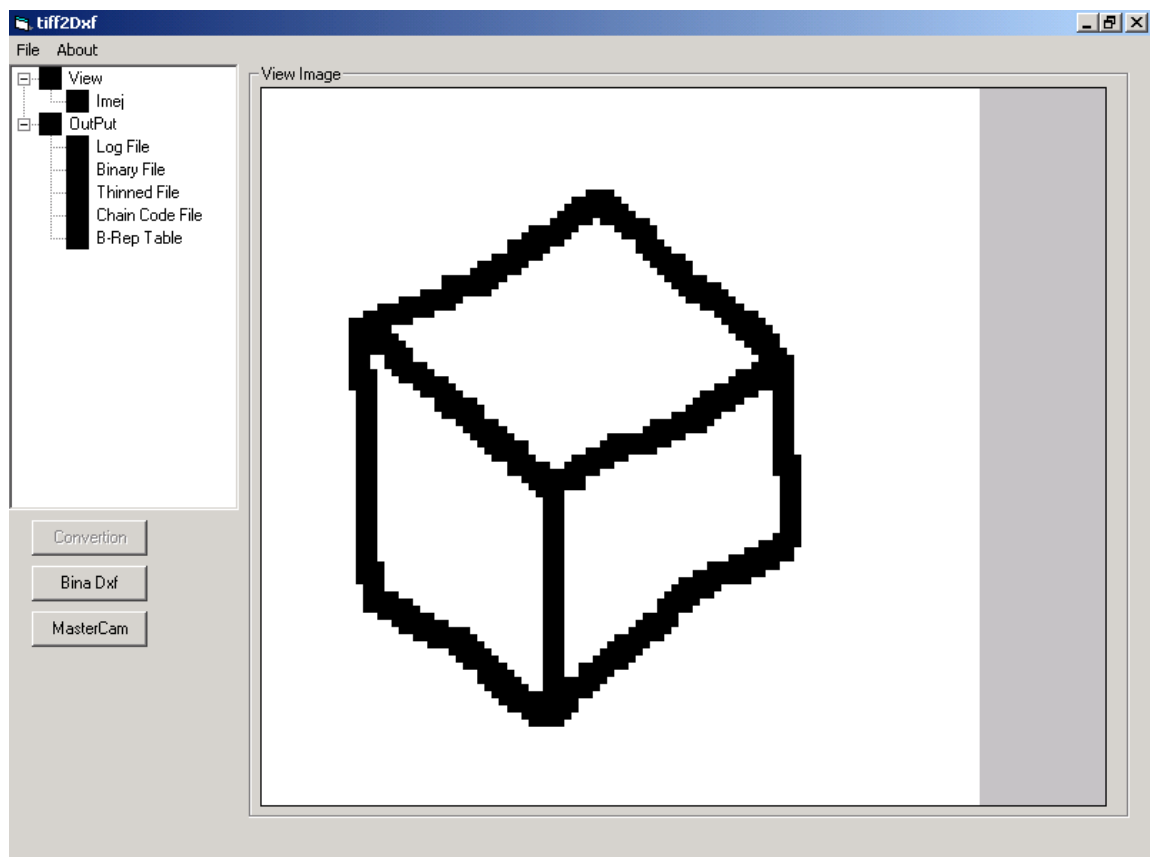


Figure 5.4: The cube displayed in the interface

5.3 The Enhanced Thinning Algorithm

This module reads the array of the binary image produced by the digitisation process discussed in Section 5.2.1. **APPENDIX A4** shows the output display of thinned binary image while **APPENDIX A5** shows the output file of thinned binary image file, but the figure has been added with rows and columns to assist in searching the location of each pixel. Figure 5.7 shows the thinned binary image obtained where the locations indicated with symbol ‘.’ indicate that the pixel have been replaced with value 0 by the thinning algorithm.

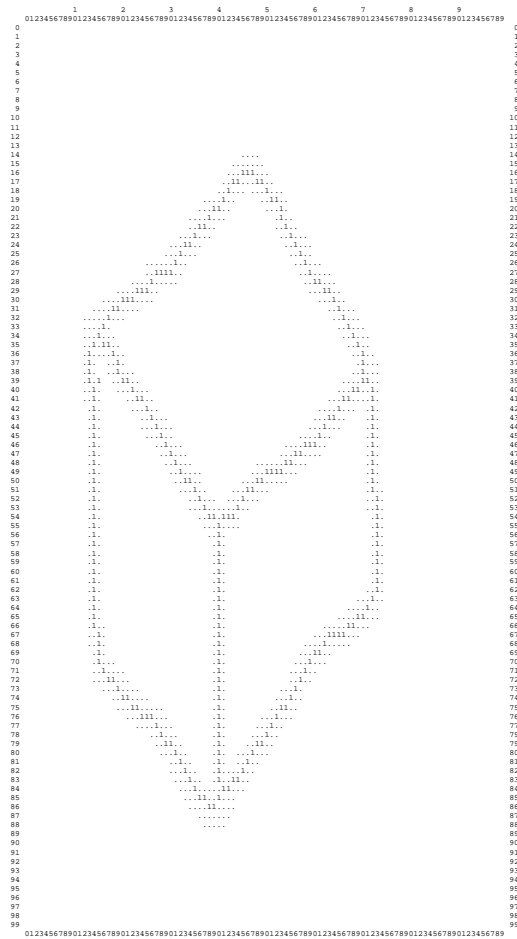


Figure 5.6: The thinned binary image

5.4 The VCC Conversion Algorithm

The algorithm has been created in a manual manner where the thinned binary image in the form of a graph paper has been converted into rectangle cell manually and then the vertex chain code is derived manually. These two steps has been done manually. Section 4.5 described in detail the process and presents example of the line drawing that has been converted into VCC.

- Step 1: Read the thinned binary image.
- Step 2: For each pixel '1' in the binary image
 - Determine the rectangle form of the pixel based on Figure 4.5.
 - The formation of rectangle form is based on the neighbouring pixel.
- Step 3. For each rectangle cell,
 - Determine the VCC code either 1, 2 or 3 based on the scheme in Figure 4.7.
- Step 4: End.

5.5 Computational Corner Detection

5.5.1 The Chain Code Algorithm for T-Junction, Corner and Line Detection

This section discusses the derivation of boundary and internal line to detect T-junctions, corners, and lines of chain code representation. Given a thinned binary image, a chain code series was derived. Figure 5.7 shows the three series of chain code derived from the thinned binary image of Figure 5.6. **APPENDIX A6** partly shows the detail of location of each code of boundary and internal line series.

<p>Columns 1 through 15</p> <pre> 1 3 1 3 2 2 1 3 1 3 1 3 1 2 2 3 1 3 2 2 1 3 1 3 1 3 1 2 2 2 1 3 2 2 1 3 1 3 1 3 1 2 2 2 2 3 2 2 1 3 1 3 1 3 1 2 2 2 2 2 2 2 1 3 1 3 1 3 1 2 2 2 2 2 2 2 1 3 1 3 1 3 1 2 2 2 2 2 2 2 1 3 1 3 1 3 1 2 2 2 2 2 2 2 2 3 1 3 1 3 1 2 2 2 2 2 2 2 2 2 1 3 1 3 1 2 2 2 2 2 2 2 2 2 2 </pre>	<p>Columns 91 through 105</p> <pre> 3 1 3 1 2 2 2 2 2 2 2 1 3 1 2 2 1 3 1 2 2 2 2 2 2 2 2 1 3 1 2 2 3 1 2 2 2 2 2 2 2 2 1 3 1 2 2 3 1 2 2 2 2 2 2 2 2 1 3 1 2 2 3 1 2 2 2 2 2 2 2 1 3 1 2 2 2 3 1 2 2 2 2 2 2 2 1 3 1 2 2 2 3 1 2 2 2 2 2 2 1 3 1 2 2 2 3 1 2 2 3 2 2 2 2 1 3 1 2 2 2 3 1 2 2 3 2 2 </pre>
<p>Columns 16 through 30</p> <pre> 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 3 2 2 2 2 2 2 2 2 2 2 2 2 2 1 3 1 2 2 2 2 2 2 2 2 2 2 2 2 1 3 1 3 </pre>	<p>Columns 106 through 120</p> <pre> 2 3 1 2 2 3 2 2 2 2 2 2 2 2 1 3 1 2 2 3 2 2 2 2 2 2 2 2 2 1 2 1 2 2 3 2 2 2 2 2 2 2 2 2 1 2 3 2 2 3 2 2 2 2 2 2 2 2 2 1 2 3 1 2 3 2 2 2 2 2 2 2 2 2 1 2 3 1 2 3 2 2 2 2 2 2 2 2 2 1 2 3 1 2 3 2 2 2 2 2 2 2 2 2 2 1 2 3 1 2 3 1 2 2 2 2 2 2 2 2 2 1 2 3 1 2 3 1 2 2 2 2 2 2 2 2 2 1 2 3 1 2 3 1 2 3 </pre>
<p>Columns 31 through 45</p> <pre> 2 2 2 2 1 3 1 3 1 3 1 2 3 1 3 2 2 2 1 3 1 3 1 3 1 2 3 1 3 1 2 2 1 3 1 3 1 3 1 2 3 1 3 1 2 2 1 3 1 3 1 3 1 2 3 1 3 1 2 2 1 3 1 3 1 3 1 2 3 1 3 1 2 2 3 3 1 3 1 3 1 2 3 1 3 1 2 2 3 1 1 3 1 3 1 2 3 1 3 1 2 2 3 1 2 3 1 3 1 2 3 1 3 1 2 2 3 1 2 2 1 3 1 2 3 1 3 1 2 2 3 1 2 2 3 </pre>	<p>Columns 121 through 135</p> <pre> 2 3 1 2 3 1 2 3 1 3 2 2 2 2 2 3 1 2 3 1 2 3 1 3 2 2 2 2 2 2 1 2 3 1 2 3 1 3 2 2 2 2 2 2 1 2 3 1 2 3 1 3 2 2 2 2 2 2 1 3 3 1 2 3 1 3 2 2 2 2 2 2 1 3 2 1 2 3 1 3 2 2 2 2 2 2 2 1 3 2 1 2 3 1 3 2 2 2 2 2 2 2 1 3 2 1 3 3 1 3 2 2 2 2 2 2 2 1 3 2 1 3 1 1 3 2 2 2 2 2 2 2 1 3 2 1 3 1 3 </pre>
<p>Columns 46 through 60</p> <pre> 1 2 2 3 1 2 2 3 1 2 2 3 1 2 3 2 2 3 1 2 2 3 1 2 2 3 1 2 3 1 2 3 1 2 2 3 1 2 2 3 1 2 3 1 3 3 1 2 2 3 1 2 2 3 1 2 3 1 3 1 1 2 2 3 1 2 2 3 1 2 3 1 3 1 2 2 2 3 1 2 2 3 1 2 3 1 3 1 2 3 2 3 1 2 2 3 1 2 3 1 3 1 2 3 1 3 1 2 2 3 1 2 3 1 3 1 2 3 1 3 1 2 2 3 1 2 3 1 3 1 2 3 1 3 1 </pre>	<p>Columns 136 through 150</p> <pre> 2 1 3 2 1 3 1 3 1 2 1 3 2 1 3 2 1 3 1 3 1 3 1 2 1 3 2 3 3 2 1 3 1 3 1 3 1 2 1 3 2 3 2 2 1 3 1 3 1 3 1 2 1 3 2 3 2 1 1 3 1 3 1 3 1 2 1 3 1 3 2 1 3 3 1 3 1 3 1 2 1 3 1 3 2 1 3 2 1 3 1 3 1 2 1 3 1 3 2 1 3 2 1 3 3 1 3 1 2 1 3 1 3 2 1 3 2 1 3 3 1 3 1 2 1 3 1 3 2 1 3 2 1 3 2 </pre>
<p>Columns 61 through 75</p> <pre> 1 3 1 2 3 1 3 1 2 2 2 2 2 1 3 3 1 2 3 1 3 1 2 2 2 2 2 1 3 1 1 2 3 1 3 1 2 2 2 2 2 1 3 1 3 2 3 1 3 1 2 2 2 2 2 1 3 1 3 1 3 1 3 1 2 2 2 2 2 1 3 1 3 1 3 1 3 1 2 2 2 2 2 1 3 1 3 1 3 1 3 1 2 2 2 2 2 1 3 1 3 1 3 1 3 1 2 2 2 2 2 1 3 1 3 1 3 1 3 1 2 2 2 2 2 1 3 1 3 1 3 1 3 1 3 </pre>	<p>Columns 151 through 165</p> <pre> 3 2 1 3 2 1 3 2 2 1 3 2 1 3 2 1 1 3 2 1 3 2 1 3 3 2 1 3 2 1 3 1 2 1 3 2 1 3 1 3 1 3 2 1 3 1 3 2 3 2 1 3 1 3 2 2 2 1 3 1 3 2 2 1 1 3 1 3 2 2 1 3 </pre>
<p>Columns 76 through 90</p> <pre> 1 3 1 2 1 3 1 3 1 3 1 3 1 3 1 3 1 2 2 3 1 3 1 3 1 3 1 3 1 3 1 2 2 2 1 3 1 3 1 3 1 3 1 3 1 2 2 2 2 </pre>	

Figure 5.9: The input of the classifier for Stair

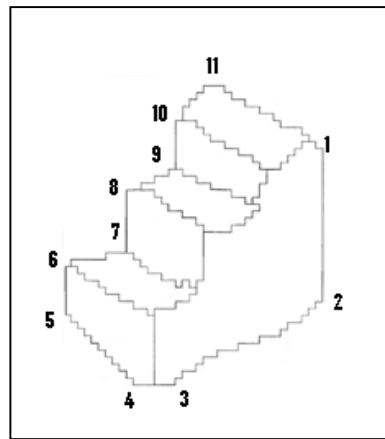


Figure 5.10: Corners at the Stair drawing

Table 5.3: Corners detected by the VCC Classifier

Method	No of Corners detected	Location of Corner
VCC NN Classifier	9	1, 2, 3, 4, 5, 6, 8, 9 and 11

<p>Columns 1 through 15</p> <p>0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0</p> <p>Columns 16 through 30</p> <p>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</p> <p>Columns 31 through 45</p> <p>1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</p> <p>Columns 46 through 60</p> <p>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</p> <p>Columns 61 through 75</p> <p>0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0</p> <p>Columns 76 through 90</p> <p>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1</p> <p>Columns 91 through 105</p> <p>0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0</p> <p>Columns 106 through 120</p> <p>0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0</p> <p>Columns 121 through 135</p> <p>0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0</p> <p>Columns 136 through 150</p> <p>0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0</p> <p>Columns 151 through 158</p> <p>0 0 0 0 0 0 0 0 0</p>	<p>Columns 1 through 15</p> <p>0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0</p> <p>Columns 16 through 30</p> <p>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</p> <p>Columns 31 through 45</p> <p>1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</p> <p>Columns 46 through 60</p> <p>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</p> <p>Columns 61 through 75</p> <p>0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0</p> <p>Columns 76 through 90</p> <p>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1</p> <p>Columns 91 through 105</p> <p>0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0</p> <p>Columns 106 through 120</p> <p>0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0</p> <p>Columns 121 through 135</p> <p>0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0</p> <p>Columns 136 through 150</p> <p>0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0</p> <p>Columns 151 through 158</p> <p>0 0 0 0 0 0 0 0 0</p>
a) Output NN Classifier	b) Actual Output

Figure 5.11: Ouput of the Classifier and the Actual Output

5.7 Research Contributions

This section discusses the results of the proposed algorithm. The major success of the system is to produce a solid model from irregular line drawing as very few solutions can be met in interpreting an irregular line drawing. The discussions on research contributions can be done by making direct comparisons to other algorithms and models used by other sketch interpreters, and are classified on few issues as listed below.

New Framework in Sketch Interpretation

In general, a new framework in sketch interpretation as shown in Figures 3.2 and 4.1 proposed that synthesises from digitising the image until producing the solid model. Although the proposed algorithm is based from the DCS by Hwang and Ullman (1990), methods used in three stages of sketch interpretations are different. This is due to three factors namely the General View Assumptions, sources of the line drawing, and the mathematical formulation used. Among the five improvements proposed are the usages of thinning algorithm and chain code in line drawing interpretation that are only used by Matsuda et. al. (1997).

Shum (1999) stated that most work on reconstruction accepts multiple views of single drawing for reconstruction. Very few considered single line drawing to be reconstructed, mainly for interpretation. Therefore, this work has contributed to one more alternative in reconstructing 3D solid model of a single line drawing rather than interpreting it.

The Enhanced Thinning Algorithm

In the enhanced thinning algorithm, besides detecting total pixel '1' of the neighbourhood is 1, this work has extended works by Pitas (1995) in terms of detecting that the total pixel '1' of the neighbourhood is 2 or 3. In addition, this algorithm has maintained the basic shape and size of the line drawing. Figure 5.15 shows the difference of output of Pitas work and this research. In the 3x3 windows of Figure 5.15 (a), Pitas algorithm does not delete the pixel indicated with the bold character while this work optimises the deletion by deleting the pixels as shown in 3x3 window of Figure 5.15 (b).

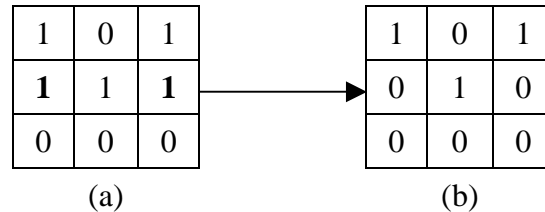


Figure 5.12: The optimised thinning algorithm

The Pitas algorithm has been extended so that the thinned binary image leaves only 1 pixel connected to the current pixel. This kind of image is useful in deriving Freeman chain code to interpret a line drawing in terms of ensuring that the T-junctions and corners derived are the optimised one because this philosophy is important in interpreting engineering sketch.

Studies by Lam et. al. (1992) show that there are many thinning algorithms developed. Therefore, the integration of sequential and parallel thinning algorithm for two-dimensional drawing of 3D solid should be incorporated with the existing sketch interface such as the system proposed by Jenkins and Martins (1993).

In the thinning algorithm, since the traversal started from left to right, therefore the elimination of the pixel is left-oriented. As a result, the thinned binary image is not really representing actual topology of the drawing. It is suggested that instead of 3x3

window, bigger window such as 5x5 window proposed by Cowell and Hussain (2001) can be considered so that more surrounding pixels will be taken in consideration in the thinning process.

Proses Pembentukan sel segiempat bagi (VCC)

The algorithm to form the rectangle cell and to derive VCC of the rectangle cell is a contribution in the chain code areas since the manual algorithm should be programmed so that it can be done by computerprogram and automatically produced VCC based on the thinned binary image. Besides Salem et al. (2005) and Bribiesca (1999), this work has added more values to the vertex chain code.

Chain Code as PDL in Object Reconstruction

Since introduced by Freeman (1969), works on the use of chain code to represent a line drawing of 3D object are very few compared to the use of line labelling as means to represent line drawing. This work presents the use of chain code as an alternative besides line labelling in representing a line drawing in 3D reconstruction.

The Freeman chain code has been integrated with the proposed thinning algorithm in deriving chain code of a closed area that contains more than one region. The algorithm is also capable of processing a two-dimensional line drawing that would represent three-dimensional object.

The 2D Freeman chain code representation can be integrated with the vertex chain code (VCC) proposed by Bribiesca (1999) in representing line drawing. Furthermore, the boundary representation used in representing the solid model can be replaced by the three-dimensional chain code scheme as proposed by Freeman (1974) or Bribiesca (2000). This extension will improve the application of chain code representation in sketch interpreter.

Feature Extraction of Chain Code Representation

This work shows that chain code can be used to derive T-junction by applying the principle saying that intersection of two lines will produce T-junction. This work also shows the effect of direction in the traversal of FCC codes in identifying boundary, internal line and region of a closed area. Two algorithms have been produced to show the significance of the direction of traversal in deriving the chain code. The start of test location as shown in Table 4.1 and Table 4.12 proposed in this thesis is capable of having a complete traversal without missing any region, and deriving all lines of the drawing.

The searching of corners between T-junctions must be carried out so that the region produces correct shape while maintaining the original image. In finding corners, the proposed method has utilised the derived chain codes. This work has used the chain code as input in computational method in corner detection algorithm. By combining the derived chain code and through simple calculations, corners can be detected. Experiment results show that length parameter, 9 and corner parameter, 3 are suitable for image size 50x100 pixels and 100x100 pixels.

VCC Corner Detection Algorithm using Neural Network Model

The vertex chain code and neural network has been used to produced the corner detection algorithm. Three components has been integrated namely VCC, neural network and Matlab software tools namely NN toolbox. Before the corner detected, the corner detection algorithm needs pre-processing stage nemaly to produce the line drawing to the VCC.

5.8 Discussion

This research has successfully proposed a new framework in sketch interpretation of a single drawing by enhancing and integrating the thinning and chain code algorithm, and proposing a new computational method and artificial intelligence technique to find corner of irregular line drawing.

CHAPTER VI

CONCLUSIONS AND FUTURE WORKS

6.1 Conclusions

The general area of this research is sketch interpretation that interprets a two-dimensional line drawing and produces a three-dimensional model that involved three stages namely pre-processing, feature extraction and object recognition. These three stages are required by framework of a general architecture of a sketch interpreter as introduced by Hwang and Ullman (1990). This study develops a new framework as shown in Figure 3.2 that consisted of different techniques and model in each stage but with the same objective. The framework has fulfilled the first objective of the research that is to develop new framework of a sketch interpreter.

As mentioned in Section 1.4, the scope of this research is to investigate all three stages of a sketch interpreter life cycle. Since there are many types of line drawing sources as discussed in Section 2.1, simplification and assumptions on the input type are required so that the interpretation can be carried out. The assumptions on the line drawing have been made and mentioned in Section 1.5. Based on the scope and assumptions made, the development of algorithms and mathematical model is preceded with the philosophy to meet the second objective that is to maintain the continuity of the interpretation. Based on the philosophy, reviews on existing techniques proposed by previous researchers in all three stages that cover thinning algorithm, chain code algorithm, corner detection, and mathematical model are carried out.

The review leads to the development of the proposed algorithm consisted of enhanced thinning algorithm in pre-processing stage, closed loop chain code algorithm in representing the line drawing, corner detection in feature extraction, and bisection method of total least square approximation. The proposed algorithm meets the third objective of the research. Generally thinning algorithm is used in image processing and chain code algorithm is used in representing shape. This thesis manages to show the applicability of these algorithms in interpreting 2D drawing of 3D object. Besides showing the applicability of these algorithms, the usage of total least square in solving a linear system and reconstructing 3D model are also proved in this thesis.

Technically, although the proposed algorithm has been implemented on different platform that is by using C and Matlab software, a complete system can actually be developed by converting the Matlab function into C codes which is allowed by Matlab compiler.

6.2 Future Works

The proposed algorithm can accept input of a single line drawing sketched in isometric view that represents a trihedral solid object with hidden lines removed. Clearly, effort should be spent in improving the proposed algorithm so that the complete sketching interpreter for general purpose can be realised so that it can be incorporated with existing commercial CAD software.

Integration of Thinning and Chain Code Algorithm in Sketch Interpretation

The threshold value chosen in the digitisation process is 200. This value is based on experiment that it is suitable enough to convert grey level values range from 0 to 255 to binary values 0 and 1. Studies on the selection of the value can improve the accuracy of the digitisation process especially when dealing with dangling and unconnected lines.

The cube used in the experiment has yielded junctions, lines and regions that are representing the geometry of the drawing. This work shows that the integration of thinning and chain code algorithm can be used in sketch interpretation. The output of the algorithm can be used as input to the geometric modeller in off-line manner such as the system proposed by Grimstead (1997), and in line labelling process proposed by Huffman (1977) and Mackworth (1973).

Input sketch consisting of unconnected line has not been tested in the algorithm. The proposed algorithm can be modified so that it can accept input of unconnected lines. In fact, if there is an unconnected line, the algorithm should be able to suggest and connect the line to the potential junction as proposed by Grimstead (1997).

Enhanced Corner Detection Algorithm of Chain Code Representation

The computational corner detection method that is used to analyse the chain code series can be improved by selecting length parameter and corner parameter automatically. Currently through experimentation, number 9 and 3 have been used as length parameters. The parameters chosen in this study is may still be optimised. Extensive research can be done to formulate algorithm that can automatically determine the optimum parameter values based on the image size. Therefore, instead of value 9 and 3, these numbers should be assigned automatically by the system. The application of artificial intelligence approach may be suggested in improving the corner detection of a chain coded representation.

General Type of Input and Output File

The issue of either the drawing is regular or irregular line drawing can be overcome by applying the works on sketch interface such as by Jenkins and Martins (1993) and Lipson and Shpitalni (1996). They gave solution to this problem. As an option, input file in terms of scanned image can be used in the interpretation. Regular line drawing is not considered because it conflicts with the natural philosophy introduced by Jenkins and Martins (1993) and Hwang and Ullman (1990).

In terms of the output file of sketch interpreter, the possibility of using computer CAD/CAM software to view the solid object can be implemented by keeping the 3D boundary representation data structure in neutral CAD file format such as DWG and Drawing Interchange Format (DXF). Therefore, commercial CAD software such as AutoCAD or CAM software such as MasterCam can be used to view and manipulate the object

The usage of neural network is hoped to be used in more research related to the development of geometric modeler. The application of NN not only improve the CAD system but avoid the usage of computational method in CAD development. The neural network can also be applied in the work by Lee and Park (1995), and Lam and Lee (1992). In addition, works by Pahor and Carrato (1999) can be considered to be integrated in Picture Description Language.

6.3 Summary

As a summary, the proposed algorithm works well for simple objects, producing results in reasonable time. Many systems require an accurate line drawing as input and therefore cannot cope with drawing errors. When compared to other systems for input of solid models, the proposed algorithm has advantage of using a simple mechanism in interpreting a freehand sketch. The study shows that the most natural way is an important part of the design process.

The extension of the system beyond assumptions has been discussed along with other possible enhancements. An algorithm that is a synthesis process of thinning algorithm, chain code algorithm, corner detection, identification of image regularities, and total least square in the 3D reconstruction of irregular line drawing has been proposed. The importance of the number of image regularities, optimisation techniques, the variety of input line drawing, and picture description language in producing three-dimensional object has also been discussed. As a conclusion, the proposed algorithm is an alternative in a sketch interpreter system not just to interpret a single sketch of line drawing but also to reconstruct the sketch into three-dimensional line drawing. As conclusion, neural network classifier based on VCC is seen as an alternative in geometric modeler development.

REFERENCES

- Akeo, M., Hashimoto, H., Kobayashi, T. and Shibusawa, T. (1994), Computer Graphics System for Reproducing Three-Dimensional Shape from Idea Sketch, *Proceedings of Eurographics '94*, Vol. 13, No. 3, pp. 477-488.
- Arrebola, F., Bandera, A., Camacho, P. and Sandoval, F. (1997), Corner Detection by Local Histograms of Contour Chain Code, *Electronics Letters*, Vol. 33 (21), pp. 1769-1771.
- Arrebola, F., Bandera, A., Camacho, P. and Sandoval, F. (1999), Corner Detection and Curve Representation by Circular Histograms of Contour Chain Code, *Electronic Letters*, Vol. 35 (13), pp. 1065-1067.
- Bachnak, R. and Celenk, M. (1989), A Corner Detection-based Object Representation Technique For 2D Images, *Proc. IEEE International Symp. On Intelligent Control*, USA, pp. 186-190.
- Bandera, A., Urdiales, C., Arrebola, F. and Sandoval, F. (2000), Corner Detection by Means of Adaptively Estimated Curvature Function, *Electronic Letters*, Vol. 36, Issue 2, pp. 124-126.
- Barrow, H.G. and Tenenbaum, J.M. (1981), Interpreting Line drawing as Three-Dimensional Surfaces, *Artificial Intelligence*, 17 (1-3), pp. 75-116.
- Bribiesca, E. (1999), A New Chain Code, *Pattern Recognition*, Vol. 32, pp. 235-251.
- Bribiesca, E. (2000), A Chain Code for Representing 3D Curves, *Pattern Recognition*, Vol. 33, pp. 755-765.
- Chen, Y.W. and Lee, S. (1991), The C-Chain Code - A New Method For Coding 3D Curves, *Proc. Of Asimolav Conf. On Signals, Systems & Computers*, Part 1 (of 2), USA, pp. 472-476.
- Choo, C.Y. and Freeman, H. (1989), Computation of Features of 2-D Polycurve-Encoded Boundaries, *Proc. IEEE Int. Conf. On Systems, Man and Cybernetics*, Vol. 3, pp. 1041-1047.

- Choo, C.Y. and Freeman, H. (1992), An Efficient Technique for Compressing Chain-Coded Line Drawing Images, *Proc. Conf. Record of The Twenty-Sixth Asimolav on Signals, Systems and Computers*, Vol. 2, pp. 717-720.
- Clowes, M.B. (1971), On Seeing Things, *Artificial Intelligence*, Vol. 2, pp. 79-116.
- Company, P., Gomis, J.M. and Contero, M. (1999), Geometrical Reconstruction from Single Line Drawings Using Optimisation-Based Approaches, *Proc. Of The WSCG 99*, Vol. II, pp. 361-368.
- Cowan, T.M. (1977), Organizing the Properties of Impossible Figures, *Perception*, Vol. 6, pp. 41-56.
- Cowell, J. and Hussain, F. (2001), Thinning Arabic Characters for Feature Extraction, *Proceedings Fifth International Conference on Information Visualisation*, pp. 181-185.
- Deutsch, E.S. (1972), Thinning Algorithms on Rectangular, Hexagonal, and Triangular Arrays, *ACM*, Vol. 15, Issue 9, pp. 827-837.
- Dias, P.G.T., Kassim, A.A. dan Srinivasan, V. (1995). *Neural Network Classifier for Detecting Corners in 2D Images*. Conf. Systems, Man and Cybernetics, Intelligent for The 21st Century Vol. 1, 661-666.
- Eggli, L., Hsu, C., Bruderlin, B. D. and Elbert, G. (1997), Inferring 3D Model from Freehand Sketches and Constraints, *Computer Aided Design*, Vol. 29, No. 2, pp. 101-112.
- Flores, E.L. (1998), A Fast Thinning Algorithm, *Proc. SBT/IEE Int. Telecommunications Symposium*, Vol. 2, pp. 594-599.
- Foley, J.D. (1976), Graphical Output and Input Capabilities, In Siegfried Treu, editor. *User-Oriented Design on Interactive Graphics Systems*, ACM/SIGGRAPH Workshop, Pittsburg, PA 1976, pp. 119-120.
- Freeman, H. (1961), On the Encoding of Arbitrary Geometric Configurations, *IRE Trans.*, EC-10 (2), pp. 260-268.
- Freeman, H. (1969), On the Quantization of Line Drawing Data, *IEEE Trans. System Science and Cybern.*, SSC-5 (1), pp. 70-79.
- Freeman, H. (1974), Computer Processing of Line-Drawing Images, *ACM Computing Surveys*, Vo. 6, No. 1, pp. 57-97.
- Fukui, Y. (1988), Input Method of Boundary Solid by Sketching, *Computer Aided Design*, Vol. 20, No. 8, pp. 434-440.

- Goldschmidt, G. (1992), Serial Sketching: Visual Problem Solving in Designing, *Cybernetics and Systems: An International Journal*, Vol. 23, pp. 191-219.
- Grimstead, I.J. and Martin, R.R. (1995), Creating Solid Model From Single 2D Sketches, *Proceedings Third Symp. On Solid Modelling Applications, ACM SIGGRAPH*, pp. 323-337.
- Grimstead, I.J. (1997), *Interactive Sketch Input of Boundary Representation Solid Models*, PhD Thesis, Univ. of Cardiff, UK.
- Guzman, A. (1968), Decomposition of a visual scene into three-dimensional bodies, *AFIPS (American Federation of Information Processing Societies) Proceedings Fall Joint Computer Conference*, Vol. 33, pp. 291-304.
- Hale, B.J., Burton, R.P., Olsen, D.R. and Stout, W.D. (1992), A Three-Dimensional Sketching Environment Using Two-Dimensional Perspective Input, *Journal of Imaging Science and Technology*, Vol. 36, No. 2, pp. 188-195.
- Hoffman, C. M. (1989), *Geometric and solid modelling: An introduction*, Morgan Kaufman, San Mateo, California.
- Hwang, T.S. and Ullman, D.G. (1990), The Design Capture System: Capturing Back-of-the Envelope Sketches, *Journal of Engineering Design*, Vol. 1 No. 4, pp. 339-353.
- Jenkins, D.L. and Martin, R.R. (1993), The Importance of Free-hand Sketching in Conceptual Design: Automatic Sketch Input, In T.K. Hight and L.A. Stauffer, editors, *Design Theory and Methodology DTM93*, Vol. DE-53, ASME, pp. 115-128.
- Kanade, T. (1980), A Theory of Origami World, *Artificial Intelligence*, Vol. 13, pp. 279-311.
- Koplowitz, J. dan Plante, S. (1995). *Corner Detection for Chain Coded Curves*. Science Direct Pattern Recognition, Vol. 28 Issue 6, 843-852.
- Kwok, P.C.K. (1988), A Thinning Algorithm by Contour Generation, *Communications of the ACM*, Vol. 31, No. 7, pp. 1314-1324.
- Kwok, P.C.K. (1990), A Modified Contour Generation Thinning Algorithm, *Proc. IEEE Conf. Region 10 Conf. On Computer and Communication Systems TENCON*, Vol. 2, pp. 491-495.
- Kwok, P.C.K. (1992), Non-Recursive Thinning Algorithms using Chain Codes, *Proc. 11th IAPR Intern. Conf. On Image, Speech and Signal Analysis, Pattern Recognition Vol. III Conf. C*, pp. 369-372.

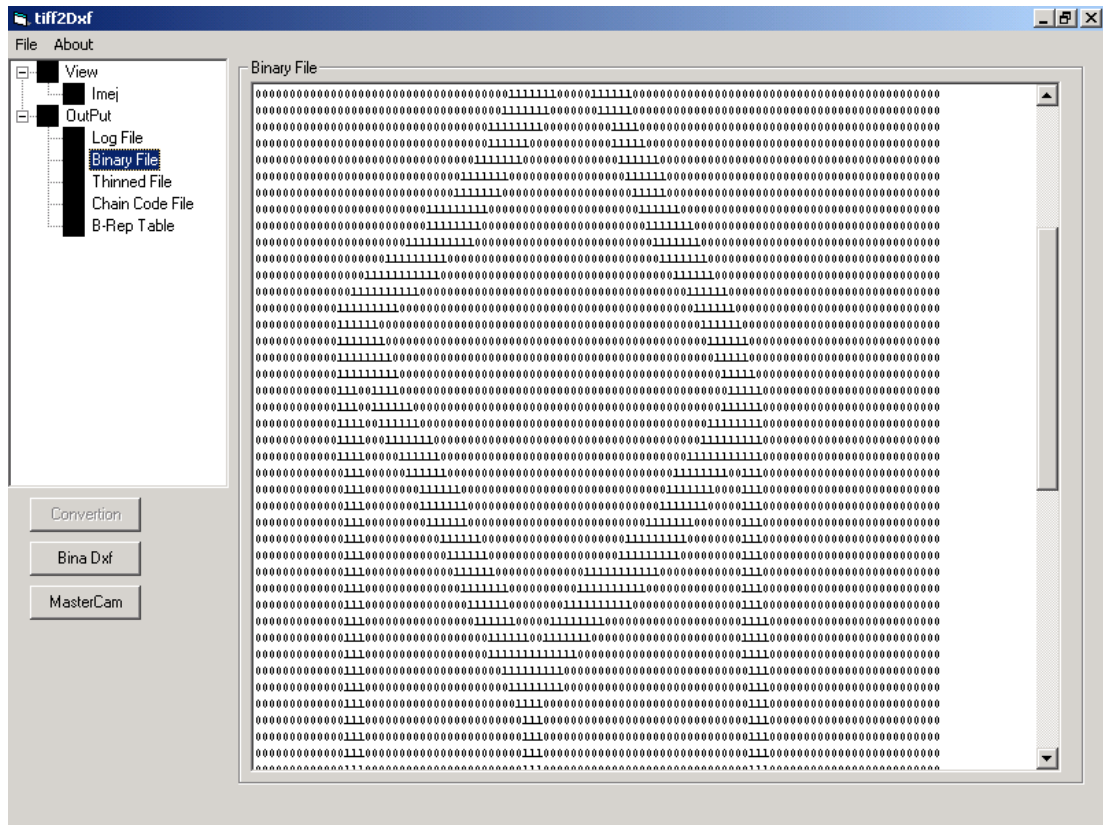
- Kwon, J.k., Gi, J.W. and Kang, E.K. (2001), An Enhanced Thinning Algorithm Using Parallel Processing, *Proc. Intern. Conf. In Image Processing*, Vol. 3, pp. 752-755.
- Lam, L., Lee, W.W. and Suen, C.Y. (1992), Thinning Methodologies - A Comprehensive Survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Issue 9, pp. 869-885.
- Langbein, F.C., Mills, B.I., Marshall, A.D. and Martin, R.R. (2001), Recognizing Geometric Patterns for Beautification of Reconstructed Solid Models, *Proc. Inter. Conf. On Shape Modelling and Applications*, pp. 10-19.
- Leclerc, Y.G. and Fischler, M.A. (1992), An Optimisation-Based Approach to the Interpretation of Single Line Drawing as 3D Frames, *International Journal of Computer Vision*, Vol. 9, No. 2, pp. 113-136.
- Lee, C.L. and Wang, P.S.P. (1994), A New Thinning Algorithm, Proc. 12th IAPR Int. Conf. On Comp. Vision & Image Processing, *Pattern Recognition*, Vol. 1, pp. 546-548.
- Lee, K. (1999), *Principles of CAD/CAM/CAE Systems*, Addison-Wesley.
- Lee, J-H and Park, D-S. (1995). *Modular Neural Net Architecture for One-step Thinning Algorithm Neural Networks*. Proceedings IEEE International Conference, Vol. 4, 1719-1723.
- Lipson, H. and Shpitalni, M. (1996), Optimisation-based Reconstruction of a 3D Object from a Single Freehand Line Drawing, *Computer Aided Design*, Vol. 28, No. 8, pp. 651-663.
- Liu, J. and Lee, Y.T. (2001), A Graph-Based Method for Face Identification from A Single 2D Line Drawing, *IEEE Trans. On Pattern Analysis and Machine Intelligence*, Vol. 23 (10), pp. 1106-1119.
- Liu, J. and Lee, Y.T. (2002), Identifying Faces in a 2D Line Drawing Representing a Manifold Object, *IEEE Trans. On Pattern Analysis and Machine Intelligence*, Vol. 24 (12), pp. 1579-1593.
- Liu, H-C. dan Srinath, M. D. (1990). *Corner Detection from Chain Code*. Pattern Recognition, Vol. 23, Issue 1-2, 51-68.
- Lu, H.E. and Wang, P.S.P. (1986), A Comment on A Fast Parallel Algorithm for Thinning Digital Patterns, *Communications of the ACM*, Vol. 29, Issue 3, pp. 239-242.

- Mackworth, A.K. (1973), Interpreting Pictures of Polyhedral Scenes, *Artificial Intelligence*, Vol. 4, pp. 121-137.
- Maeder, A.J. (1990), Animation Techniques for Chain-Coded Objects, *Proc. First IEEE Conf. On Visualization*, pp. 67-73.
- Marill, T. (1991), Emulating the Human Interpretation of Line Drawing as Three-Dimensional Objects, *International Journal of Computer Vision*, Vol. 6 No. 2, pp. 147-161.
- Matsuda, K., Sugishita, S., Zheng Xu, Kondo, K., Sato, H. and Shimada, S. (1997), Freehand Sketch System for 3D Geometric Modelling, *Proceedings 1997 Int. Conf. On Shape Modelling and Applications*, Japan, pp. 55-62.
- Minami, T. and Shinohara, K. (1986), Encoding of Line Drawings with a Multiple Grid Chain Code, *IEEE Trans. On Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 2, pp. 269-276.
- Norgaard, M., Ravn, O., Poulsen, N. K. dan Hansen, L. K. (2000). *Neural Networks for Modelling and Control of Dynamic Systems*, London: Springer.
- Nunes, P., Pereira, F. and Marques, F. (1997), Multi-Grid Chain Coding of Binary Shapes, *Proc. Int. Conf. On Image Processing*, Vol. 3, pp. 114-117.
- O' Gorman, L. O. (1988), Primitive Chain Code, *Proc. Intern. Conf. On Accoustic, Speech and Signal Proc.*, ICASSP-88, New York, Vol. 2, pp. 792-795.
- Pahor, V. dan Carrato, S. (1999). *A Fuzzy Approach to Mouth Corner Detection. Image Processing*, ICIP 99, Vol. 1, 667-671.
- Pham, B. (1994), From CAD to Automation of Creation Design Process, *Proceedings of Conf. On Computer Aided Geometric Design*, Malaysia, pp. 1-7.
- Pitas, I. (1995), *Digital Image Processing Algorithm*, University Press Cambridge, Prentice Hall.
- Pugh, D. (1992), Designing Solid Objects using Interactive Sketch Interpretation, *ACM Proc. Symp. On Interactive 3D Techniques*, pp. 117-126.
- Rajagopalan, R. dan Rajagopalan, P. (1996). *Applications of Neural Network in Manufacturing*. Proceedings of the Twenty-Ninth Hawaii Int. Conf., Vol. 2, 447-453.
- Rajan, P.K. and Davidson, J.M. (1989), Evaluation of Corner Detection Algorithm, *Proc. Twenty-First Southeastern Symp. On System Theory*, USA, pp. 29-33.

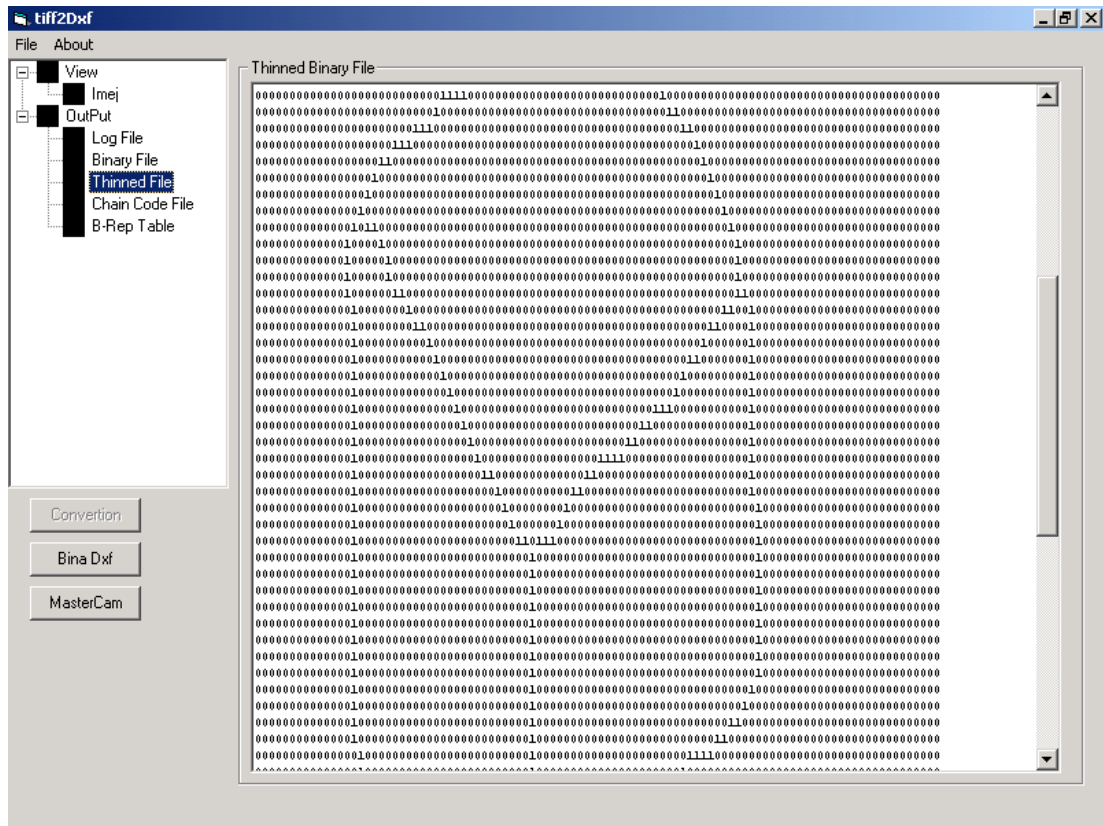
- Rosa, J.L.G. (2001) *An Artificial Neural Network Model Based on Neuroscience: Looking Closely at the Brain*. Proceeding of the Int. Conf. Artificial Neural Nets and Genetic Alg., Vol 1, 138-141.
- Reche, P., Uridales, C., Bandera, A., Trazegnies, C. and Sandoval, F. (2002), Corner Detection by Means of Contour Local Vectors, *Electronic Letters*, Vol. 38 (14), pp. 699-701.
- Rosenfeld, A. dan Johnston, E. (1973). *Angle detection on Digital Curves*. IEEE Trans. Comput. Vol. 22, 875-878.
- Salem, A.-B. M., Sewisy, A. A. dan Elyan U. A. (2005). *A Vertex Chain Code Approach for Image Recognition*. ICGST International Journal on Graphics, Vision and Image Processing, Vol. 2, 1-8.
- Sanchiz, J. M., Inesta, J.M. dan Pla, F. (1996). *A Neural Network-Based Algorithm to Detect Dominant Points from the Chain-Code of a Contour*. Vol. 4, 325-329.
- Shum, Simon Siu-pun (1999), *Solid Reconstruction from Orthographic Projected Line Drawings*, PhD Thesis, Dept. of Manufacturing Engineering, Hong Kong Polytechnic University.
- Sivaloganathan, S. (1991), *Sketching Input for Computer Aided Engineering*, PhD Thesis, City University, Dept. Of Mechanical Engineering and Aeronautics.
- Staunton, R.C. (2001), One-pass parallel hexagonal thinning algorithm, *IEE Proc. Vis. Image Signal Process*, Vol. 148, No. 1, pp. 45-53.
- Stefanelli, R. and Rosenfeld, A. (1971), Some Parallel Thinning Algorithms for Digital Pictures, *Journal of the ACM*, Vol. 18 Issue 2, pp. 255-264.
- Sugihara, K. (1978), Picture Language for Skeletal Polyhedra, *Computer Graphics and Image Processing*, Vol. 8, pp. 382-405.
- Sugishita, S., Kondo, K., Sato, H. and Shimada, S. (1994), Sketch Interpreter for Geometric Modelling, *Proceedings of Conf. On Computer Aided Geometric Design*, Malaysia, pp. 1-8.
- Sun, Y., Li, H-H., Hu, J-S. and Wang, E-L. (2002), 2D Recovery of Human Posture, *IEEE Proceedings of the First International Conference on Machine Learning and Cybernetics*, Beijing, pp. 1638-1640.
- Sutherland, I.E. (1963), Sketchpad A Man-Machine Graphical Communication System, *Proceedings AFIPS Conf. Spring Joint Computer Conference*, Vol. 23, pp. 329-346.

- Terry, P.J. dan Vu, D. (1993). *Edge Detection Using Neural Networks*”, Signals, Sys. and Com. Vol. 1, 391-395.
- Tsai, D-M (1997). *Boundary-based Corner Detection Using Neural Networks, Pattern Recognition*. Vol. 30, Issue 1, 85-97.
- Ullman, D.G. (2002), Toward the Ideal Mechanical Engineering Design Support System, *Research in Engineering Design*, Vol. 13, pp. 55-64.
- Varley, P.A.C. and Martin, R.R. (2000), A System for Constructing Boundary Representation Solid Models from a Two-Dimensional Sketch, *IEEE Proc. Geometric Modelling and Processing*, pp. 13-32.
- Vicent, A.P., Calleja, P.C. and Martin, R.R. (2003), Skewed Mirror Symmetry in the 3D Reconstruction of Polyhedral Models, *Journal of WSCG*, Vol. 11, No. 1, pp. 1-8.
- Wang, P.S.P., Hui, L. and Fleming, Jr. T. (1986), Further Improved Fast Parallel Thinning for Digital Patterns, In *Computer Vision, Image processing and Communication Systems and Application* edited by P.S.P. Wang, pp. 37-40.
- Wang, W. (1992), *On the Automatic Reconstruction of a 3D Objects Constructive Solid Geometry Representation from Its 2D Projection Line Drawing*, PhD Thesis, University of Massachusetts-Lowell, Order No. 9224569, Department of Computer Science, University of Massachusetts-Lowell.
- Zeleznik, R.C., Herndon, K.P. and Hughes, J.K. (1996), SKETCH: An Interface for Sketching 2D Scenes, In *Proceedings of SIGGRAPH*, New Orleans, pp. 163-169.
- Zhang, T.Y. and Suen, C.Y. (1984), A Fast Parallel Algorithm for Thinning Digital Patterns, *Comm. Of the ACM-IPCV*, Vol. 27, No. 3, pp. 236-239.
- Zhu, Y., Seneviratne, L.D. and Earles, S.W.E. (1995), New Line-based Thinning Algorithm, *Proc. IEE on Vision, Image and Signal Processing*, Vol. 142, pp. 351-358.

APPENDIX A2: Output display of binary image (Cube1)



APPENDIX A4: Output display of thinned binary image (Cube1)



APPENDIX A6: Output file of complete chain code series (Cubel)

BOUNDARY	84 42 3	List of	REGION	54 43 0	35 17 0	20 53 1
AND	84 41 4	Direction 1:	LIST	53 44 7	36 18 1	21 53 2
INTERNAL	85 40 3	Row Col Code		52 45 7	37 19 1	22 54 1
LIST	86 39 3	35 16 -1	Region No 0	51 46 7	38 19 2	23 55 1
	86 38 4	35 17 0		51 47 0	39 20 1	24 56 1
	85 37 5	36 18 1		50 48 7	39 21 0	25 57 1
List of	85 36 4	37 19 1	Row Col Code	50 49 0	40 22 1	26 58 1
Direction 0:	84 35 5	38 19 2	38 70 -1	49 50 7	41 23 1	27 59 1
(Boundary)	83 34 5	39 20 1	39 71 1	49 51 0	41 24 0	28 60 1
	82 33 5	39 21 0	40 72 1	49 52 0	42 25 1	28 61 0
Row Col Code	81 32 5	40 22 1	41 72 2	49 53 0	43 26 1	29 62 1
16 45 -1	80 31 5	41 23 1	42 72 2	48 54 7	44 27 1	29 63 0
16 46 0	79 30 5	41 24 0	43 72 2	48 55 0	45 28 1	30 64 1
16 47 0	79 29 4	42 25 1	44 72 2	47 56 7	46 29 1	31 65 1
17 48 1	78 28 5	43 26 1	45 72 2	47 57 0	47 30 1	32 66 1
17 49 0	77 27 5	44 27 1	46 72 2	46 58 7	48 31 1	33 67 1
18 50 1	76 26 5	45 28 1	47 72 2	46 59 0	49 32 1	34 68 1
19 51 1	76 25 4	46 29 1	48 72 2	46 60 0	50 33 1	35 69 1
19 52 0	76 24 4	47 30 1	49 72 2	45 61 7	50 34 0	36 70 1
20 53 1	75 23 5	48 31 1	50 72 2	44 62 7	51 35 1	37 70 2
21 53 2	75 22 4	49 32 1	51 72 2	43 63 7	52 36 1	38 70 2
22 54 1	74 21 5	50 33 1	52 73 1	43 64 0	53 37 1	39 70 2
23 55 1	74 20 4	50 34 0	53 73 2	42 65 7	54 38 1	40 69 3
24 56 1	73 19 5	51 35 1	54 73 2	41 66 7	54 39 0	40 68 4
25 57 1	72 18 5	52 36 1	55 73 2	41 67 0	55 40 1	41 67 3
26 58 1	72 17 4	53 37 1	56 73 2	40 68 7	56 40 2	41 66 4
27 59 1	71 16 5	54 38 1	57 73 2	40 69 0	57 40 2	42 65 3
28 60 1	70 15 5	54 39 0	58 73 2	39 70 7	58 40 2	43 64 3
28 61 0	69 15 6	55 40 1	59 73 2		59 40 2	43 63 4
29 62 1	68 15 6	54 41 7	60 73 2	Region No 1	60 40 2	44 62 3
29 63 0	67 15 6	54 42 0	61 73 2	Row Col Code	61 40 2	45 61 3
30 64 1	66 14 5	54 43 0	62 73 2	84 41 -1	62 40 2	46 60 3
31 65 1	65 14 6	53 44 7	63 72 3	85 40 3	63 40 2	46 59 4
32 66 1	64 14 6	52 45 7	64 71 3	86 39 3	64 40 2	46 58 4
	63 14 6	51 46 7	65 70 3	86 38 4	65 40 2	47 57 3
33 67 1	62 14 6	51 47 0	65 69 4	85 37 5	66 40 2	47 56 4
34 68 1	61 14 6	50 48 7	66 68 3	85 36 4	67 40 2	48 55 3
35 69 1	60 14 6	50 49 0	66 67 4	84 35 5	68 40 2	48 54 4
36 70 1	59 14 6	49 50 7	67 66 3	83 34 5	69 40 2	49 53 3
37 70 2	58 14 6	49 51 0	67 65 4	82 33 5	70 40 2	49 52 4
38 70 2	57 14 6	49 52 0	67 64 4	81 32 5	71 40 2	49 51 4
39 71 1	56 14 6	49 53 0	67 63 4	80 31 5	72 40 2	49 50 4
40 72 1	55 14 6	48 54 7	68 62 3	79 30 5	73 40 2	50 49 3
41 72 2	54 14 6	48 55 0	69 61 3	79 29 4	74 40 2	50 48 4
42 72 2	53 14 6	47 56 7	69 60 4	78 28 5	75 40 2	51 47 3
43 72 2	52 14 6	47 57 0	70 59 3	77 27 5	76 40 2	51 46 4
44 72 2	51 14 6	46 58 7	71 58 3	76 26 5	77 40 2	52 45 3
45 72 2	50 14 6	46 59 0	72 57 3	76 25 4	78 40 2	53 44 3
46 72 2	49 14 6	46 60 0	73 56 3	76 24 4	79 40 2	54 43 3
47 72 2	48 14 6	45 61 7	74 55 3	75 23 5	80 40 2	54 42 4
48 72 2	47 14 6	44 62 7	75 54 3	75 22 4	81 40 2	54 41 4
49 72 2	46 14 6	43 63 7	75 53 4	74 21 5	82 40 2	55 40 3
50 72 2	45 14 6	43 64 0	76 52 3	74 20 4	83 40 2	54 39 5
51 72 2	44 14 6	42 65 7	77 51 3	73 19 5		54 38 4
52 73 1	43 14 6	41 66 7	78 50 3	72 18 5	Region No 2	53 37 5
53 73 2	42 14 6	41 67 0	79 49 3	72 17 4		52 36 5
54 73 2	41 14 6	40 68 7	79 48 4	71 16 5	Row Col Code	51 35 5
55 73 2	40 14 6	40 69 0	80 47 3	70 15 5	34 15 -1	50 34 5
56 73 2	39 13 5	39 70 7	81 46 3	69 15 6	33 16 7	50 33 4
57 73 2	38 13 6		82 45 3	68 15 6	32 17 7	49 32 5
58 73 2	37 13 6	List of	83 44 3	67 15 6	31 18 7	48 31 5
59 73 2	36 13 6	Direction 2:	83 43 4	66 14 5	31 19 0	47 30 5
60 73 2	35 14 7	Row Col Code	84 42 3	65 14 6	30 20 7	46 29 5
61 73 2	34 15 7	56 40 -1	84 41 4	64 14 6	30 21 0	45 28 5
62 73 2	33 16 7	57 40 2	83 40 5	63 14 6	30 22 0	44 27 5
63 72 3	32 17 7	58 40 2	82 40 6	62 14 6	29 23 7	43 26 5
64 71 3	31 18 7	59 40 2	81 40 6	61 14 6	29 24 0	42 25 5
65 70 3	31 19 0	60 40 2	80 40 6	60 14 6	29 25 0	41 24 5
65 69 4	30 20 7	61 40 2	79 40 6	59 14 6	28 26 7	41 23 4
66 68 3	30 21 0	62 40 2	78 40 6	58 14 6	27 27 7	40 22 5
66 67 4	30 22 0	63 40 2	77 40 6	57 14 6	27 28 0	39 21 5
67 66 3	29 23 7	64 40 2	76 40 6	56 14 6	27 29 0	39 20 4
67 65 4	29 24 0	65 40 2	75 40 6	55 14 6	27 30 0	38 19 5
67 64 4	29 25 0	66 40 2	74 40 6	54 14 6	26 31 7	37 19 6
67 63 4	28 26 7	67 40 2	73 40 6	53 14 6	25 32 7	36 18 5
68 62 3	27 27 7	68 40 2	72 40 6	52 14 6	24 33 7	35 17 5
69 61 3	27 28 0	69 40 2	71 40 6	51 14 6	24 34 0	35 16 4
69 60 4	27 29 0	70 40 2	70 40 6	50 14 6	23 35 7	
70 59 3	27 30 0	71 40 2	69 40 6	49 14 6	22 36 7	
71 58 3	26 31 7	72 40 2	68 40 6	48 14 6	22 37 0	
72 57 3	25 32 7	73 40 2	67 40 6	47 14 6	21 38 7	
73 56 3	24 33 7	74 40 2	66 40 6	46 14 6	20 39 7	
74 55 3	24 34 0	75 40 2	65 40 6	45 14 6	20 40 0	
75 54 3	23 35 7	76 40 2	64 40 6	44 14 6	19 41 7	
75 53 4	22 36 7	77 40 2	63 40 6	43 14 6	18 42 7	
76 52 3	22 37 0	78 40 2	62 40 6	42 14 6	17 43 7	
77 51 3	21 38 7	79 40 2	61 40 6	41 14 6	17 44 0	
78 50 3	20 39 7	80 40 2	60 40 6	40 14 6	16 45 -1	
79 49 3	20 40 0	81 40 2	59 40 6	39 13 5	16 46 0	
79 48 4	19 41 7	82 40 2	58 40 6	38 13 6	16 47 0	
80 47 3	18 42 7	83 40 2	57 40 6	37 13 6	17 48 1	
81 46 3	17 43 7		56 40 6	36 13 6	17 49 0	
82 45 3	17 44 0		55 40 6	35 14 7	18 50 1	
83 44 3			54 41 7	34 15 7	19 51 1	
83 43 4			54 42 0	35 16 1	19 52 0	

End of Project Report Guidelines

A. Purpose

The purpose of the End of Project is to allow the IRPA Panels and their supporting group of experts to assess the results of research projects and the technology transfer actions to be taken.

B. Information Required

The following Information is required in the End of Project Report :

- Project summary for the Annual MPKSN Report;
- Extent of achievement of the original project objectives;
- Technology transfer and commercialisation approach;
- Benefits of the project, particularly project outputs and organisational outcomes; and
- Assessment of the project team, research approach, project schedule and project costs.

C. Responsibility

The End of Project Report should be completed by the Project Leader of the IRPA-funded project.

D. Timing

The End of Project Report should be submitted within three months of the completion of the research project.

E. Submission Procedure

One copy of the End of Project is to be mailed to :

IRPA Secretariat
Ministry of Science, Technology and the Environment
14th Floor, Wisma Sime Darby
Jalan Raja Laut
55662 Kuala Lumpur

End of Project Report

A. Project number : 04-02-06-0083-EA 001

Project title : DEVELOPMENT OF COMPUTER AIDED THREE-DIMENSIONAL GEOMETRIC
MODELER BASED ON DESIGNER'S IDEA IN TWO-DIMENSIONAL

Project leader: Dr Habibollah bin Haron

Tel: 07-5532012

Fax: 07-5565044

B. Summary for the MPKSN Report (for publication in the Annual MPKSN Report, please summarise the project objectives, significant results achieved, research approach and team structure)

The objective:

This research is to produce a new framework in development of geometric modeler. Five algorithm/model proposed are the enhanced thinning algorithm, vertex chain code conversion (VCC) algorithm, Freeman chain code (FCC) modified algorithm, feature extraction algorithms based on FCC, and neural network classifier for corner detection based on VCC.

Results achieved:

Development of geometric modeler involved three phases namely pre-processing, two-dimensional feature extraction, and three-dimensional object reconstruction. The research successfully propose a framework of geometric modeler in interpreting a single line drawing sketch and reconstruct the sketch to produce a two dimensional model. The methodology used in this study is based on the framework of a geometric modeler that contains two phases. In the first phase, two algorithms were proposed. The first is enhanced thinning algorithm that is used to thin the sketch while maintaining its topology and basic shape. The second is the conversion algorithm to convert the irregular line drawing into vertex chain code (VCC) and into modified Freeman chain code (FCC). These algorithm are used to produce chain code series that represented a thinned binary image of the sketch. In the second phase, a new computational corner detection algorithm of FCC and neural network classifier for corner detection of VCC are proposed to analyse the chain code series to produce two-dimensional geometry entities of the sketch in which the irregular line drawing is transformed into regular line drawing. This study has proposed better solutions in terms of an enhanced thinning algorithm, a modified Freeman chain code and Vertex chain code algorithm as picture description language, and a new computational algorithm and neural network classifier for corner detection.

Research approach:

1. Sketching: The irregular line drawing is sketched using Adobe Photoshop and saved as TIFF format.
2. Digitization process: The .tiff file is digitised into binary file.
3. Thinning process: The binary file is then thinned to produced thinned binary image.
4. Chain Code Representation:
 - 4.1 The thinned binary image is represented as vertex chain code. The VCC conversion algorithm is applied.
 - 4.2 The thinned binary image is represented as Freeman chain code.
The modified closed-loop chain code is applied.
5. Two-dimensional Feature Extraction
 - 5.1 The Freeman chain code is the extracted to produce 2D entities namely junctions, lines, and region.
A geometry concept and computational method is applied in the feature extraction.
 - 5.2 The vertex chain code is the extracted to produce corner only.
A methodology in development of neural network model is applied to produce the classifier .

Research Team:

1. Habibollah Haron (Project Leader)
2. Dzulkifli Mohamed (Researcher)
3. Siti Mariyam Shamsuddin (Researcher)
4. Mohamad Shukor Talib (Researcher)
5. Syarul Haniz Subri (MSc Student)

C. Objectives achievement

- **Original project objectives** (Please state the specific project objectives as described in Section II of the Application Form)

1. To analyze & determine an effective approach of analyzing the 2D representation of the product and produce the 3D model.
2. To design techniques for modeling that allows engineers to sketch ideas, manipulate and view the design automatically in 3D format by incorporating linguistic approach, mathematical formulation, optimization technique and model representation.
3. To develop a prototype system that is able to convert a 2D model to 3D model.

- **Objectives Achieved** (Please state the extent to which the project objectives were achieved)

1. The effective approach has been delivered in term of a new framework of geometric modeler development.
2. There are five algorithms created.

- **Objectives not achieved** (Please identify the objectives that were not achieved and give reasons)

The third objective has not been achieved because of budget constraint. The proposal required two RA but the budget only can fund one RA.

D. Technology Transfer/Commercialisation Approach (Please describe the approach planned to transfer/commercialise the results of the project)

1. Publication in national and international journals.
2. Results and findings have been diffused in the national seminar.

Publications

1. Syarul Haniz Subri, **Habibollah Haron**, Roselina Sallehuddin, Neural Network Corner Detection of Vertex Chain Code, ICGST International Journal on Artificial Intelligence and Machine Learning (AIML), January 2006, Vol (6) Issue(1), 37-43.
2. **Habibollah Haron**, Siti Mariyam Shamsuddin, dan Dzulkifli Mohamed, Corner Detection of Chain-Coded Representation of Thinned Binary Image, International Journal of Computer Mathematics, U.K., Vol. 82 No 8, August 2005, 941-950.
3. Syarul Haniz Subri, Roselina Sallehuddin, and **Habibollah Haron**, Neural Network In Corner Detection of Chain Code Series, Proc. The International Arab Conference on Information Technology ACIT, Algeria Dec 2004, Vol I page 446-451.
4. Syarul Haniz Subri, and **Habibollah Haron**, A New Framework In Extracting Features Of Irregular Line Drawing, Proc. Inter. Conf. On Comp. Graphic, Imaging and Visualization CGIV 04, page 83-87.
5. **Habibollah Haron**, Dzulkifli Mohamad, and Siti Mariyam Shamsuddin (2003), Enhancement of Thinning Algorithm For 2D Line Drawing of 3D objects, Journal of Mathematics & Computer Sciences (Computer Science Series), India, Dec. 2003, pages 169-174.
6. **Habibollah Haron**, Dzulkifli Mohamad and Siti Mariyam Shamsuddin (2003), Extraction of Junctions, Lines and Regions of Irregular Line Drawing: The Chain Code Processing Algorithm, Jurnal Teknologi, No. 38 (Siri D), June 2003, Penerbit UTM, pages 1-28
7. **Habibollah Haron**, Dzulkifli Mohamad and Siti Mariyam Shamsuddin (2003), Applying chain code algorithm in extracting geometric entities of line drawing, GMAG 03 16-18 July 2003, Book of Abstract, UK, page 6.
8. **Habibollah Haron** and Dzulkifli Mohamad (2002), Penerbitan Maklumat Tiga Dimensi Sebuah Lukisan Garisan Menggunakan Kaedah Jumlah Kuasa Dua Terkecil, Jurnal Teknologi No. 37 (Siri D), Dis. 2002, Penerbit UTM, UTMJB, page 93-116
9. **Habibollah Haron** and Dzulkifli Mohamad (2001), Pendekatan Pengoptimuman Dalam Penghasilan Objek Tiga Dimensi daripada Lukisan Garisan Dua Dimensi, Jurnal Teknologi, No. 34 (Siri C), Jun 2001, Penerbit UTM, UTMJB, page 53-62.

E. Benefits of the Project (Please identify the actual benefits arising from the project as defined in Section III of the Application Form. For examples of outputs, organisational outcomes and sectoral/national impacts, please refer to Section III of the Guidelines for the Application of R&D Funding under IRPA)

- **Outputs of the project and potential beneficiaries** (Please describe as specifically as possible the outputs achieved and provide an assessment of their significance to users)

Output of the project:

1. New framework in development of geomatric modeler.
2. Five algorithms:
 - enhanced thinning algorithm; vertex chain code vonversion algorithm;
 - closed-loop Freeman chain code algorithm for 2D feature extraction;
 - new computational corner detection of Freeman chain code;
 - VCC neural network classifier for corner detection.

Potential beneficiaries:

The newframework and the algorithms can be applied in the development of CAD related software

- **Organisational Outcomes** (Please describe as specifically as possible the organisational benefits arising from the project and provide an assessment of their significance)

Expertise development in the area of computer aided geometric modeler (CAGD) – MSc (1)

- **National Impacts** (If known at this point in time, please describes specifically as possible the potential sectoral/national benefits arising from the project and provide an assessment of their significance)

F. Assessment of project structure

- **Project Team** (Please provide an assessment of how the project team performed and highlight any significant departures from plan in either structure or actual man-days utilised)

The project team consisted of project leader, 3 researchers, 2 Research Assistant. Based on the approved funding from MOSTI, the project team was revised and the project was carried out by the project leader, 3 researchers and 1 RA.

- **Collaborations** (Please describe the nature of collaborations with other research organisations and/or industry)
-nil-

G. Assessment of Research Approach (Please highlight the main steps actually performed and indicate any major departure from the planned approach or any major difficulty encountered)

Part 1: Pre-processing

Sketching

Digitization process

Part 2: FCC Feature Extraction Module

Derivation of Freeman chain code closed-loop algorithm.

Computational method in corner detection of Freeman chain code.

Part 3: Neural Network Classifier Development

Derivation of vertex chain code of thinned binary image.

VCC Neural network classifier for corner detection

G. Assessment of the Project Schedule (Please make any relevant comment regarding the actual duration of the project and highlight any significant variation from plan)

The project was schedule in 2 years and was completed on-time with revision on the objective due to the budget constrain.

I. Assessment of Project Costs (Please comment on the appropriateness of the original budget and highlight any major departure from the planned budget)

Project Cost Component	Total Approved	Expenditure
J400	48,000.00	47,037.90
J500	10,000.00	8,216.99
J600		0.00
J700	6,000.00	5,017.60
J800	0.00	389.00
J900	10,000.00	9,986.00
Balance	74,000.00	70,647.49

J. Additional Project Funding Obtained (In case of involvement of other funding sources, please indicate the source and total funding provided)

None

K. Other Remarks (Please include any other comment which you feel is relevant for the evaluation of this project)

Date :

Signature :

Benefits Report Guidelines

A. Purpose

The purpose of the Benefits Report is to allow the IRPA Panels and their supporting experts to assess the benefits derived from IRPA-funded research projects.

B. Information Required

The Project Leader is required to provide information on the results of the research project, specifically in the following areas:

- Direct outputs of the project;
- Organisational outcomes of the project; and
- Sectoral/national impacts of the project.

C. Responsibility

The Benefits Report should be completed by the Project Leader of the IRPA-funded project.

D. Timing

The Benefits Report is to be completed within three months of notification by the IRPA Secretariat. Only IRPA-funded projects identified by MPKSN are subject to this review. Generally, the Secretariat will notify Project Leaders of selected projects within 18 months of project completion.

E. Submissin Procedure

One copy of this report is to be mailed to :

IRPA Secretariat
Ministry of Science, Technology and the Environment
14th, Floor, Wisma Sime Darby
Jalan Raja Laut
55662 Kuala Lumpur

Benefit Report

1. Description of the Project

A. Project identification

1. **Project number : 04-02-06-0083-EA 001**
2. **Project title : Development of Computer Aided Three-Dimensional Geometric Modeler Based on Designer's Idea in Two-Dimensional**
3. **Project leader : Dr Habibollah bin Haron**

B. Type of research

Indicate the type of research of the project (Please see definitions in the Guidelines for completing the Application Form)

Scientific research (fundamental research)

Technology development (applied research)

Product/process development (design and engineering)

Social/policy research

C. Objectives of the project

1. Socio-economic objectives

Which socio-economic objectives are addressed by the project? (Please identify the sector, SEO Category and SEO Group under which the project falls. Refer to the Malaysian R&D Classification System brochure for the SEO Group code)

Sector : Services & Information Technology

SEO Category : Information and Communications Services

SEO Group and Code : S20901 Computer software and services

2. Fields of research

Which are the two main FOR Categories, FOR Groups, and FOR Areas of your project? (Please refer to the Malaysia R&D Classification System brochure for the FOR Group Code)

a. Primary field of research

FOR Category : Information, Computer and Communication Technologies

FOR Group and Code : F10503 Software

FOR Area : Computer Aided System

b. Secondary field of research

FOR Category : Applied Sciences and Technologies

FOR Group and Code : F10602 Manufacturing and process technologies and engineering

FOR Area : CAD/CAM Systems

D. Project duration

What was the duration of the project ?

24 Months

E. Project manpower

How many man-months did the project involve?

60 Man-months

F. Project costs

What were the total project expenses of the project?

RM74,000.00

G. Project funding

Which were the funding sources for the project?

Funding sources

Total Allocation (RM)

IRPA

74,000.00

II. Direct Outputs of the Project

A. Technical contribution of the project

1. What was the achieved direct output of the project :

For scientific (fundamental) research projects?

Algorithm

Structure

Data

Other, please specify : _____

For technology development (applied research) projects :

Method/technique

Demonstrator/prototype

Other, please specify : _____

For product/process development (design and engineering) projects:

Product/component

Process

Software

Other, please specify : _____

2. How would you characterise the quality of this output?

Significant breakthrough

Major improvement

Minor improvement

B. Contribution of the project to knowledge

1. How has the output of the project been documented?

- Detailed project report
- Product/process specification documents
- Other, please specify : _____

2. Did the project create an intellectual property stock?

- Patent obtained
- Patent pending
- Patent application will be filed
- Copyright

3. What publications are available?

- Articles (s) in scientific publications How Many: 6
- Papers(s) delivered at conferences/seminars How Many: 3
- Book
- Other, please specify : _____

4. How significant are citations of the results?

- Citations in national publications How Many:
- Citations in international publications How Many:
- None yet
- Not known

III. Organisational Outcomes of the Project

A. Contribution of the project to expertise development

1. How did the project contribute to expertise?

PhD degrees How Many: _____

MSc degrees How Many: 1

Research staff with new specialty How Many: _____

Other, please specify: _____

2. How significant is this expertise?

One of the key areas of priority for Malaysia

An important area, but not a priority one

B. Economic contribution of the project?

1. How has the economic contribution of the project materialised?

Sales of manufactured product/equipment

Royalties from licensing

Cost savings

Time savings

Other, please specify : _____

2. How important is this economic contribution ?

High economic contribution Value: RM_____

Medium economic contribution Value: RM_____

Low economic contribution Value: RM_____

3. When has this economic contribution materialised?

- Already materialised
- Within months of project completion
- Within three years of project completion
- Expected in three years or more
- Unknown

C Infrastructural contribution of the project

1. What infrastructural contribution has the project had?

- New equipment Value: RM _____
- New/improved facility Investment : RM _____
- New information networks
- Other, please specify: _____

2. How significant is this infrastructural contribution for the organisation?

- Not significant/does not leverage other projects
- Moderately significant
- Very significant/significantly leverages other projects

D. Contribution of the project to the organisation's reputation

1. How has the project contributed to increasing the reputation of the organisation

- Recognition as a Centre of Excellence
- National award
- International award
- Demand for advisory services
- Invitations to give speeches on conferences
- Visits from other organisations
- Other, please specify: _____

2. How important is the project's contribution to the organisation's reputation ?

Not significant

Moderately significant

Very significant

IV. National Impacts of the Project

A. Contribution of the project to organisational linkages

1. Which kinds of linkages did the project create?

- Domestic industry linkages
- International industry linkages
- Linkages with domestic research institutions, universities
- Linkages with international research institutions, universities

2. What is the nature of the linkages?

- Staff exchanges
- Inter-organisational project team
- Research contract with a commercial client
- Informal consultation
- Other, please specify: _____

B. Social-economic contribution of the project

1. Who are the direct customer/beneficiaries of the project output?

Customers/beneficiaries:	Number:
Software House	1
_____	_____
_____	_____

2. How has/will the socio-economic contribution of the project materialised ?

- Improvements in health
- Improvements in safety
- Improvements in the environment
- Improvements in energy consumption/supply
- Improvements in international relations
- Other, please specify: _____

3. How important is this socio-economic contribution?

High social contribution

Medium social contribution

Low social contribution

4. When has/will this social contribution materialised?

Already materialised

Within three years of project completion

Expected in three years or more

Unknown

Date:

Signature:

UNIVERSITI TEKNOLOGI MALAYSIA
Research Management Centre

PRELIMINARY IP SCREENING & TECHNOLOGY ASSESSMENT FORM

(To be completed by Project Leader submission of Final Report to RMC or whenever IP protection arrangement is required)

1. PROJECT TITLE IDENTIFICATION :

DEVELOPMENT OF COMPUTER AIDED THREE-DIMENSIONAL GEOMETRIC MODELER
BASED ON DESIGNER'S IDEA IN TWO-DIMENSIONAL

Vote No: 74225

2. PROJECT LEADER :

Name : HABIBOLLAH BIN HARON

Address : JABATAN PPI, FSKSM, UTMJB

Tel : 07-5532012 Fax : 07-5565044 e-mail : habib@fsksm.utm.my

3. DIRECT OUTPUT OF PROJECT *(Please tick where applicable)*

Scientific Research	Applied Research	Product/Process Development
<input checked="" type="checkbox"/> / Algorithm	<input checked="" type="checkbox"/> / Method/Technique	<input type="checkbox"/> Product / Component
<input type="checkbox"/> Structure	<input type="checkbox"/> Demonstration / Prototype	<input type="checkbox"/> Process
<input type="checkbox"/> Data		<input checked="" type="checkbox"/> / Software
<input type="checkbox"/> Other, please specify _____	<input type="checkbox"/> Other, please specify _____	<input type="checkbox"/> Other, please specify _____
_____	_____	_____
_____	_____	_____

4. INTELLECTUAL PROPERTY *(Please tick where applicable)*

- | | |
|--|--|
| <input checked="" type="checkbox"/> / Not patentable | <input type="checkbox"/> Technology protected by patents |
| <input type="checkbox"/> Patent search required | <input type="checkbox"/> Patent pending |
| <input type="checkbox"/> Patent search completed and clean | <input type="checkbox"/> Monograph available |
| <input type="checkbox"/> Invention remains confidential | <input type="checkbox"/> Inventor technology champion No |
| <input type="checkbox"/> publications pending | <input type="checkbox"/> Inventor team player |
| <input type="checkbox"/> No prior claims to the technology | <input type="checkbox"/> Industrial partner indentified |

5. LIST OF EQUIPMENT BOUGHT USING THIS VOT

- 1. Digitizer
- 2. Notebook (1 unit)
- 3. Personal Computer (2 unit)

6. STATEMENT OF ACCOUNT

a)	APPROVED FUNDING	RM : 74,000.00
b)	TOTAL SPENDING	RM : 74,000.00
c)	BALANCE	RM :

7. TECHNICAL DESCRIPTION AND PERSPECTIVE

Please tick an executive summary of the new technology product, process, etc., describing how it works. Include brief analysis that compares it with competitive technology and signals the one that it may replace. Identify potential technology user group and the strategic means for exploitation.

a) Technology Description

This research project proposes a new framework in development of geometric modeler That consisted of five modified algorithm/model namely the enhanced thinning algorithm, vertex chain code conversion algorithm, Freeman chain code modified algorithm, computational corner detection algorithm of FCC, VCC neural network classifier for corner detection.

b) Market Potential

The result of this research is useful for software houses that develop a CAD related software. The vertex and Freeman chain code is useful in representing irregularline drawing to be applied in development of CAD software.

.....

Signature and stamp of
JKPP Chairman

Name :

Date :

c) Commercialisation Strategies

Technology

Futher development steps can be taken towards commercialization of the algorithms proposed by the new framework.

8. RESEARCH PERFORMANCE EVALUATION

a) FACULTY RESEARCH COORDINATOR

Research Status	()	()	()	()	()	()
Spending	()	()	()	()	()	()
Overall Status	()	()	()	()	()	()
	Excellent	Very Good	Good	Satisfactory	Fair	Weak

Comment/Recommendations :

.....
 Signature and stamp of
 JKPP Chairman

Name :
 Date :

.....
 Signature and stamp of
 JKPP Chairman

Name :
 Date :

b) RMC EVALUATION

Research Status	()	()	()	()	()	()
Spending	()	()	()	()	()	()
Overall Status	()	()	()	()	()	()
	Excellent	Very Good	Good	Satisfactory	Fair	Weak

Comments :-

Recommendations :

- Needs further research
- Patent application recommended
- Market without patent
- No tangible product. Report to be filed as reference

.....
 Signature and Stamp of Dean / Deputy Dean
 Research Management Centre

Name :
 Date :