Old Jawi Manuscript: Digital Recognition

ZAIDI RAZAK

THESIS SUBMITTED IN FULFILMENT
OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

FACULTY OF COMPUTER SCIENCE
& INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2016

UNIVERSITI MALAYA
ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: **Zaidi Razak**

Registration/Matric No: **WHA030010**

Name of Degree: **Doctor of Philosophy**

Title of Project Paper/Research Report/Dissertation/Thesis ("this Work"):

Old Jawi Manuscript: Digital Recognition

Field of Study:

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This Work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date

Subscribed and solemnly declared before,

Witness's Signature Date

Name:
Designation:

Abstract

Jawi is an ancient writing style that originated from Arabic characters but has additional characters to suit certain local pronunciation in the Malay spoken language. Jawi has been the lingua franca in the Malay archipelagos long before the introduction of Roman characters in this part of the world. As such, there are thousands of old manuscripts on the socio-economic development, governance and history of the Malay Archipelagos written in Jawi. These old manuscripts are rich sources of knowledge and thus invaluable reference for learning the culture, traditions, and early history of Malaysia. However, these old manuscripts are not easily accessible for learning or research. Moreover, old Jawi manuscripts are not easily understood because the complexity of the writing, the overlapping of characters, physical condition of the manuscripts, and the absence of a consistent writing style. The mentioned problems have provided the motivation to conduct this research. Digitizing the old Jawi manuscripts and converting old Jawi to modern Jawi will provide solution to the problems. In this context, this research will examine Jawi writing style, its characters and the affinity between Jawi and Arabic characters. The literature review in this research looks at works conducted in the past on line and character segmentation as well as recognition systems. The information gathered is useful for the development of a line segmentation algorithm, a new character segmentation algorithm, as well as a recognition algorithm which is based on the use of a unique code and Hamming distance calculation. This research did not use artificial intelligence techniques to achieve the research goal. The algorithms developed in this research were evaluated against similar algorithms for other recognition methods. The evaluation results show that the proposed approach outperforms the algorithms of other recognition methods used for comparison. Future research in this domain could explore the implementation of a hardware-based Jawi recognition system, development of a character recognition technique to handle

different character widths, designing comprehensive manuscript preservation system, and developing a tool for manuscript authentication.

Abstrak

Jawi merupakan satu kaedah penulisan yang menggunakan aksara-aksara Arab dan beberapa aksara tambahan untuk menggambarkan bahasa pertuturan Melayu. Jawi pernah menjadi lingua-franca bagi Kepulauan Melayu sebelum aksara Roman diperkenalkan. Manuskrip-manuskrip yang ditulis dalam Jawi adalah kaya dalam sejarah mengenai Kepulauan Melayu dan juga pengetahuan lain seperti agama, pemerintahan dan budaya. Pada masa ini, kajian berkaitan dengan manuskrip-manuskrip ini tidak begitu rancak disebabkan ketidaksediaan manuskrip-manuskrip secara fizikal dan juga ianya amat sukar untuk dibaca oleh golongan muda walaupun mereka mempunyai kemahiran dalam pembacaan Jawi. Masalah ini berpunca daripada kekompleksan penulisan, kualiti manuskrip dan juga ketiadaan piawaian dalam cara penulisan. Kami dapat menyelesaikan dengan mendigitalkan manuskrip ini untuk dijadikan rujukan bagi generasi sekarang dan yang akan datang. Dalam kajian ini, kami meneliti cara-cara penulisan Jawi, ciri-cirinnya dan juga kesaksamaan penulisan Jawi dengan aksara Arab. Dalam kajian ini juga, kami membuat peninjauan kajian-kajian yang berkaitan dalam bidang pendigitalan manuskrip. Akhir kajian ini, kami dapat menghasilkan satu algoritma baru untuk mengasingkan baris-baris dalam manuskrip dan disamping itu kami juga menghasilkan algoritma untuk pengasingan aksara-aksara. Kami juga berjaya merekabentuk satu algoritma untuk pengestrakan kod-kod unik bagi aksara dan seterusnya menghasilkan satu algoritma pengecaman aksara dengan menggunakan penggiraan perbezaan *Hamming.* Hasil akhir menunjukkan peratusan ketepatan yang tinggi dalam proses pengecaman walaupun tidak melibatkan kepintaran buatan ataupun algoritma pembelajaran. Keputusan kajian ini juga membuka peluang untuk menghasilkan satu produk komersil dalam pendigital manuskrip Jawi.

Acknowledgment

I would like to express my sincere gratitude to my supervisor Assoc. Prof. Dr. Rosli Bin Salleh for his continuous support, invaluable guidance and encouragement during the period of my candidature.

In addition, I would like to thank all people particularly Prof. Dr. Abdullah Bin Gani, who have in one way or other helped me in completing this thesis.

Finally, I am grateful to my beloved family for their patience, encouragement, and sacrifice during course of this research.

Table of Content

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

This chapter presents an overview of digital preservation of old Jawi manuscripts. It also states the motivation for the research and discusses the problem statements, the objectives and the methodology used. The chapter is divided into six sections. Section 1.2 highlights the motivations for the research by explaining the importance of the proposed work and the proposed solution. Section 1.3 summarizes the problem statements by highlighting issues concerned in extracting information from old Jawi manuscripts. Section 1.4 highlights the research objectives while Section 1.5 discusses the methodology used in this research. Section 1.6 summarizes the organization of the thesis.

1.1 Background

Malay manuscripts are defined as handwritten works produced from the 16th century to the 19th century (National Library of Malaysia, 2002). These works were written using Jawi characters which originated from Arabic characters. Before the introduction of the Arabic characters, the Malays used characters called Palava and Kawi (Hashim, 2005). These old characters became obsolete when Islam came to the Malay archipelagos. Muslim missionaries used Arabic texts for teaching and propagating Islam in this region. Malay Muslim scholars soon adopted Arabic in their work. The first evidence of Jawi writing is Batu Bersurat of Terengganu which was found in 1887 (Haji Saidi, 1996). As the Malay language was the lingua franca of the region at that time, all communication media on religious matters, history, tales, and government

documentations were in Jawi. Jawi also became one of the main writing styles in the world during that time. Western countries also learnt this writing in order to communicate with the Malay government of Malacca. These historical manuscripts are well preserved in various locations in the country such as the national archive, national library and several public universities; as well as abroad such as the British Library. Jawi manuscripts contain a lot of information on Malay history during the period thus they have become invaluable treasures to the Malay communities today. Often, these manuscripts can only be accessed upon approval by the relevant authorities. Most of these manuscripts are kept in the national archive and national library to physically preserve them. As a result, this limits access to the manuscripts for study or research. Since Jawi characters originated from Arabic characters, they are identical except that in Jawi, another nine characters have been added to suit Malay pronunciations, hence, making it a 37-character language script. For example, in Malay, silver is called "*perak*" but in Arabic, there is no character that can produce "*p*" sound. If no extra character had been added, it would sound "*berak*", which makes it a word with a different meaning. Thus, one character "*pfa*" ڤ — with a triple dot on top — which resembles "*fpa*" ف in Arabic is added to liken it to the one dot in "*fa*". To acquire basic reading skill in Jawi, one must be familiar with the Arabic characters. However, in order to read the historical manuscripts, it is essential to have knowledge of the Arabic character set and the old vocabularies. Those with such knowledge include mainly the religious scholars, historians and language researchers who are mostly from the older generation of Malays. The younger generation lacks this ability because the current education system emphasises the use of Roman characters for writing. Those who study the manuscripts and have good knowledge of the old vocabularies can greatly benefit from the useful information contained in them. The information has to be transliterated into modern Malay writing system. In the transliteration process, some words which have different

spellings and different meanings are changed to suit the modern writing system (Dahaman, 1991).

1.2 Motivations

The young generation of Malays today show an increasingly poor ability to read Jawi. In a study, Nik Yaacob (2007) found that 70% of young generation of Malays are not able to read Jawi documents. He attributed this mainly to the national education system which stresses on the use of Roman characters even from the early phases of education (Dewan Bahasa dan Pustaka, 1967). It is not surprising that Old Jawi Manuscripts (OJMs) have become unfamiliar to many and thus, underutilized and not used as a source of knowledge. The only way to restore Jawi to its former status is to foster reading ability of Jawi among the Malay community.

To acheive this objective, the federal government introduced the compulsory jQAF (Jawi, Quran, Arabic and Fardu Ain (Compulsory Routine as a Muslim)) programme in schools in 2004 (Malaysia. Ministry of Education, 2004). The introduction of jQAF has improved the ability to read Jawi. However, to read OJMs, knowledge of old Jawi vocabularies is needed. To address this issue, a special programme was initiated by Kang (2008) to promote better reading and understanding of OJMs.

Another way to promote wider usage of OJMs is to convert them into digital copies. Digital copies are not scanned copies but produced using optical character recognition (OCR) technology and thus the text can be edited and manipulated. The manipulated form does not mean that the content has been changed, but it can be considered the first stage of the transliteration process. In the transliteration process, the original content (text) of the OJM which uses the old Jawi script is changed into modern Jawi, which makes the digital transliterated form more easily understandable to those who need it for study or research. The digitized format of the manuscripts can be manipulated using

computers to extract or retrieve useful historical information. The relevant information can be used to enrich the literary and historical resources of the Malays— often needed for research and reference. The digitized format of the OJMs will also ensure secure preservation of these important documents. Researchers also prefer this format as it is more convenient for access and retrieval.

The emergence of various sophisticated image processing tools recently has greatly facilitated conversion of old historical Jawi documents into the digital format. Image processing tools that include those for image enhancement, feature extraction and recognition are also used for optical recognition of handwritten documents. Some of these tools have been reviewed to determine those suitable for character recognition to achieve the aim of this research.

1.3 Statement of Problem

Today, those working with the OJMs use the transliterated format produced by experts in OJMs. A new or novice user of OJMs should initially be guided by a more experienced user. In the past, OJMs were written without conforming to any writing standard. Different writers from different regions of the country adopted different writing styles (Hashim, 2005). Some writers produced the manuscripts based on their own basic knowledge of the Jawi scripts. For example, Malays in the peninsula have their own way of writing and it is different from that of the Malays in Brunei. The different writing styles also cause differences between their manuscripts. Some writers used simple characters but others used fashionable styles which cause overlapping of characters not only between two lines but also between adjacent characters. Overlapped characters between lines can confuse the readers because some overlapped characters spoil or adversely affect other characters below them. Non-uniformity in writing style — where simple characters are used alongside fashionable characters — can cause

difficulty to inexperienced readers. In OJMs, characters in a paragraph are often cramped to save space. The cramped characters in the lines or within the paragraphs leave no gap between words and those who are not familiar with this will have difficulty in reading the manuscripts. For this reason, OJMs in their original format pose some challenges to new users. In addition, the paper used in OJMs has inevitably deteriorated over the years and this causes the characters to be smeared, further reducing the legibility of the text.

1.4 Statement of Objectives

This research is aimed at developing a computerized system for recognizing characters in OJMs. In this context, the following objectives will be pursued to achieve the research aim:

1. To investigate the characteristics of Jawi scripts and identify their differences with Arabic scripts.
2. To review existing character recognition algorithms for Jawi characters and determine the most suitable algorithms to be developed in this research.
3. To propose algorithms for line, and character segmentations as well as Jawi character recognition.
4. To evaluate the effectiveness and efficiency of the proposed approach for handling and processing old Jawi manuscripts. In the context of this research, effectiveness refers to time complexity of algorithm and efficiency refers to accuracy rate of algorithm.

1.5 Research Methodology

In this research, we reviewed the literature to identify pertinent issues on state-of-the-art systems for character recognition in OJMs. The focus will be on the past researches on cursive handwriting as well as Arabic character recognition methods. The research

problem stated in Section 1.3 will be addressed by studying the physical appearance of OJMs. Every Jawi character — in their original form and the mutated form — will be studied. Writing styles which cause problems of overlapping between lines, between characters as well as the difference between group-based and individual-based characters will also be investigated.

In this research, a fast and accurate line segmentation algorithm is proposed to separate overlapped lines to reduce misrepresentation of character pixels. On the other hand, the proposed character segmentation algorithm employs sliding techniques. For character recognition, a simple pattern matching architecture using unique character code and Hamming distance calculation is proposed.

The proposed approach is simulated in Matlab using three different sets of samples. The first sample has very clear images, and no overlapped lines or characters. The second images are of low clarity, no intense overlapping between lines and small number of overlapping characters. The third sample has poor quality images, intense overlapping between lines and also between characters.

The proposed approaches are evaluated by comparing the visually-recognized characters in every sample. The accuracy rate as shown in Equation 1.1 is then calculated by comparing the number of correctly recognized character with the total number of recognized characters.

$$Accuracy\ rate = \frac{number\ of\ correctly\ recognized\ characters}{Total\ number\ of\ characters} \times 100 \qquad 1.1$$

1.6 Thesis Organization

This thesis consists of six chapters, which are illustrated in Figure 1.1. Below is a brief summary of the subsequent chapters:

CHAPTER 2 reviews past researches in the domain as well as other related works on cursive handwriting. Discussion focuses on the crucial processes involved in Jawi character recognition such as line segmentation, word segmentation, character segmentation and recognition. The outcome of the research in this chapter helps in better understanding of the different characteristics of Jawi particularly when it is compared to Arabic character based manuscripts. Moreover, the analysis of different techniques in this chapter facilitates choosing the most suitable algorithm.

CHAPTER 3 presents the proposed approach for line segmentation. It discusses problems in line segmentations, and the proposed approach for line segmentation and the result of evaluation of the new algorithm.

CHAPTER 4 discusses the proposed approach for character segmentations as applied to Jawi scripts. It also discusses the evaluation of the proposed approach and how they perform against other similar algorithms.

CHAPTER 5 describes in detail the proposed approach for Jawi character recognition. It also discusses the evaluation of the proposed recognition approach and how it performs against other similar algorithms.

CHAPTER 6 concludes the research by recapitulating the research objectives, and discusses how they were achieved. It also states the contributions and limitations of the research, and suggests other aspect in the domain for future research.

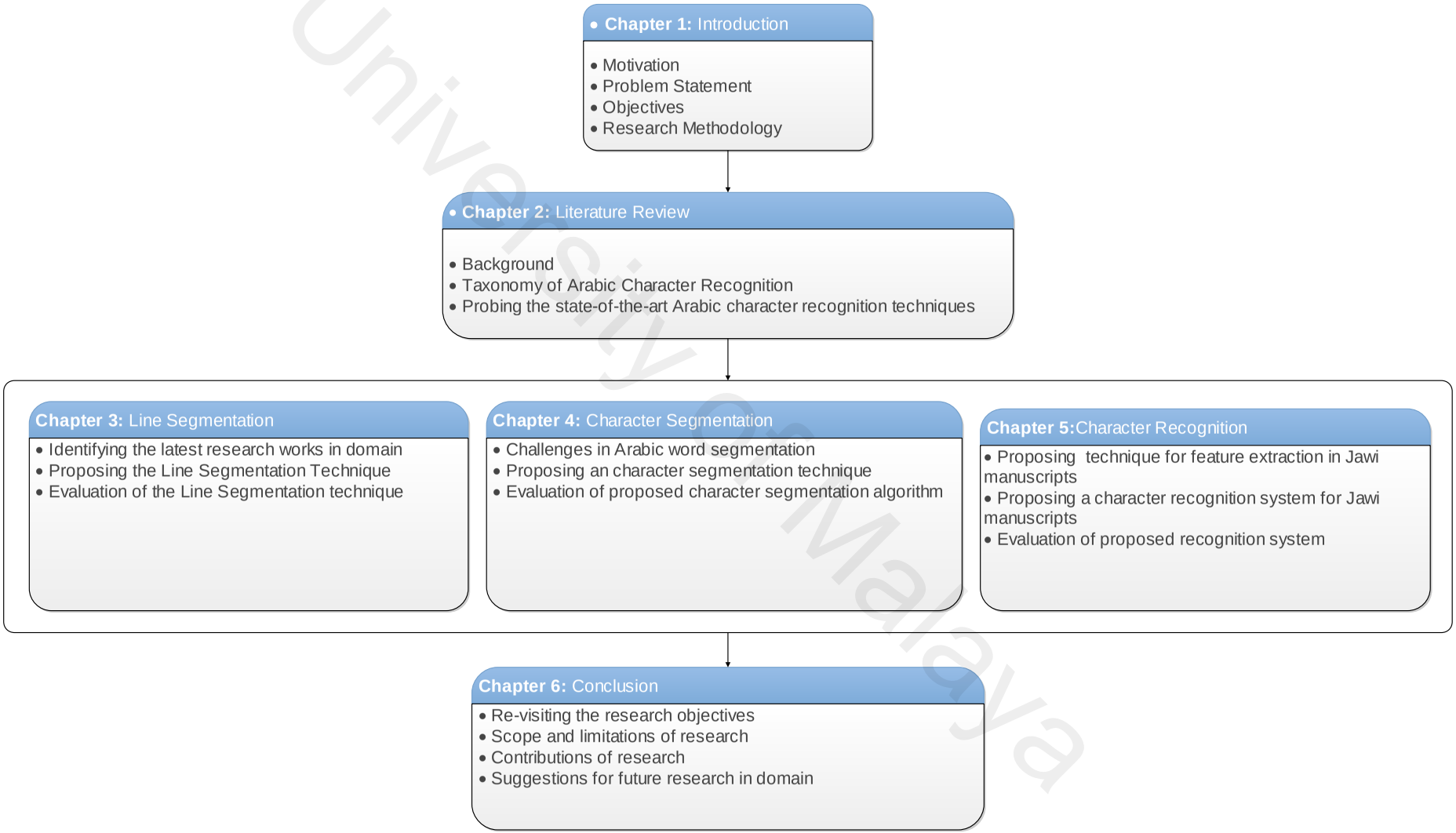

Figure 1.1 Thesis organization

CHAPTER 2

OLD JAWI MANUSCRIPT: AN OVERVIEW

This chapter presents the origin of Jawi, reviews the state-of-the-art techniques and the taxonomy for Arabic-like characters recognition. This chapter is organized into six sections. Section 2.1 discusses the origin of Jawi writing, its role as writing lingua franca; details on the Jawi script, and its writing style. Section 2.2 reviews researches in Arabic character recognition. Section 2.3 presents the role of feature extraction in character recognition methods. Section 2.4 discusses the classification approach based on code matching. Section 2.5 discusses the analytical approach in Arabic-like characters recognition. Section 2.6 provides a summary of the chapter, together with concluding statements.

2.1 Background

This section presents the background on Jawi characters and Jawi writing styles, and the current methods of analyzing Old Jawi Manuscripts (OJM).

2.1.1 Jawi Writing as the Lingua Franca of the Malay Archipelago

Jawi originated from the Arabic script and was introduced to the Malay Archipelago at the same time with Islam. Muslim missionaries introduced the Arabic language and also brought along their documents in Arabic scripts used as teaching materials to propagate Islam. Since then, Malays have adopted the Arabic writing system in their writings and official documents. Initially, Jawi had 28 Arabic characters, but over time, six other characters were added to suit local Malay pronunciations. Modern Jawi has

another three characters which have been added to suit modern Bahasa Malaysia pronunciation (Che Wan Ahmad, Omar, Nasrudin, Murah, & Bakar, 2013) thus making present-day Jawi a 37-character script. The first evidence of Jawi writing is the Batu Bersurat that was found in Terengganu in 1899 (Zabidi Haji Saidi, 1996). Other evidences include Hikayat Hang Tuah, which is a tale about a Malay legendary hero. Figure 2.1 shows a sample of Jawi script used in Hikayat Hang Tuah.

Jawi also played an important role in the political arena when it was used as a communication medium between the westerners and the Malay government. Figure 2.2 shows a letter sent to the westerners by the Royal Malay Ternate (Doa, 2008).

The introduction of Roman characters to the Malay archipelago has adversely affected the ability of the Malays to read and write Jawi (Nik Yaacob, 2007). The Malay community today realizes the need for the young generation to be conversant in reading Jawi (Abdul Hamid & Abdullah, 2009). Despite its low usage among the general public Jawi is still being used in several specific fields such as Islamic studies, mini magazines, and monthly newspapers (Nik Yaacob, 2007). Old Jawi manuscripts are still useful for learning the early history and culture of the Malays. The information they contain include kinship, religious education, diplomatic agreements, trade contracts, government matters and law (Hj Yahaya, 2004). The next section presents more detailed discussion on Jawi characters and the writing styles.

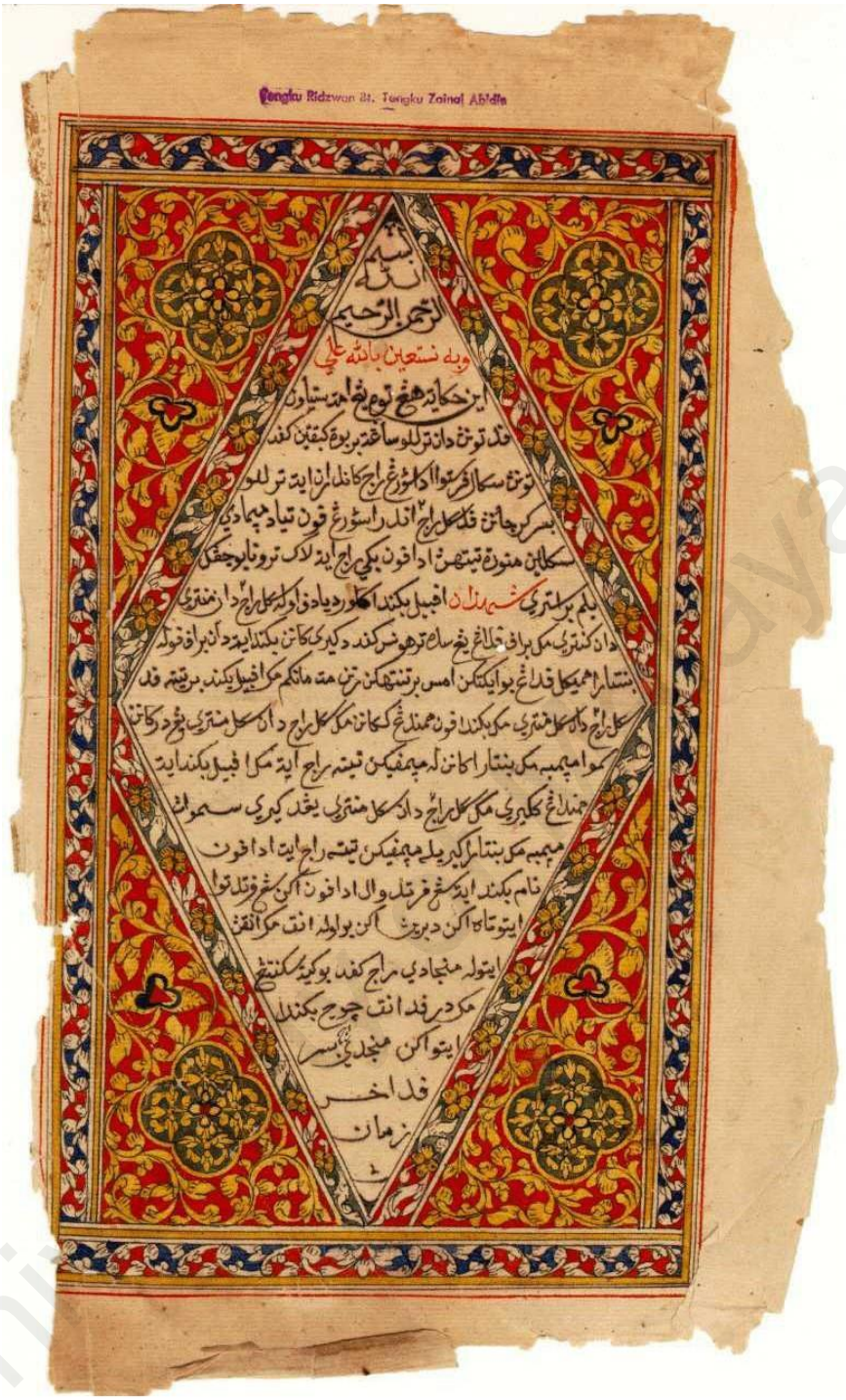

Figure 2.1 Sample of Jawi script in Hikayat Hang Tuah

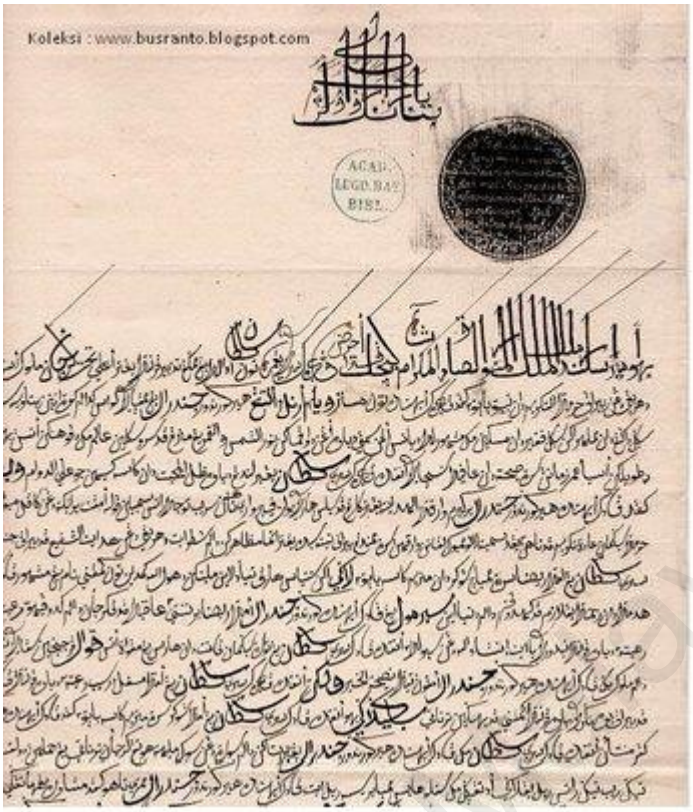

Figure 2.2 Political Communication of Royal Malay Ternate

2.1.2 Jawi Characters Details and Writing Style

From the literature review, a few salient features about the Jawi characters and writing have been noted:

- Jawi is similar to Arabic;
- Jawi is written from right to left;
- There is no upper case or lower case letter in Jawi;
- Characters have different appearances depending on their adjacent characters; and
- Dots in Jawi have significant representation.

Jawi has 37 characters in the isolated form, as illustrated in Table 2.1, compared to Arabic which has only 28 characters. The extra characters in Jawi were added to suit Malay pronunciations of certain words, for example, letters *ca*, *nga*, *pa*, *ga*, *va*, *dan*, and *nya*. Table 2.1 shows additional characters in Jawi, marked by dark shades.

Some Jawi characters have their own unique form but most of them belong to a group, depending on their basic forms, as shown in Table 2.2.

In the Jawi writing style, not all characters can be connected — some characters connect at the beginning, middle, or at the end, but some characters have to be in their original form in a sub-word. For example, character *Alif* can be connected at the beginning of a word but not at the end. If it is connected at the end, then *Alif* will become *Lam* (refer to Table 2.1). Another example is the letter *Ha,* which can be connected on both sides, but its form will change from the original form to other forms (refer to Table 2.1) .

Jawi letters also connect at the baseline, as illustrated in Figure 2.3. The baseline refers to an imaginary line with the highest number of black pixels, as shown in the histogram in Figure 2.4.

Table 2.1 List of Jawi characters

Name	Characters			
	Isolated	Beginning	Middle	Ending
Alif	ا			ـا
Ba	ب	بـ	ـبـ	ـب
Ta	ت	تـ	ـتـ	ـت
Sa	ث	ثـ	ـثـ	ـث
Jim	ج	جـ	ـجـ	ـج
Ha	ح	حـ	ـحـ	ـح
Ca	چ	چـ	ـچـ	ـچ
Kha	خ	خـ	ـخـ	ـخ
Dal	د			ـد
Zal	ذ			ـذ
Ra	ر			ـر
Zai	ز			ـز
Sin	س	سـ	ـسـ	ـس
Syin	ش	شـ	ـشـ	ـش
Sad	ص	صـ	ـصـ	ـص
Dad	ض	ضـ	ـضـ	ـض
Ta	ط	طـ	ـطـ	ـط

Za	ظ	ظ	ـظ	ظ
Ain	ع	ع	ـعـ	ع
Ghain	غ	غ	ـغـ	غ
Nga	ڠ	ڠ	ـڠـ	ڠ
Fa	ف	ف	ـفـ	ف
Pa	ڤ	ڤ	ـڤـ	ڤ
Qaf	ق	ق	ـقـ	ق
Kaf	ك	ک	ـکـ	ك
Ga	گ	گ	ـگـ	گ
Lam	ل	ل	ـلـ	ل
Mim	م	م	ـمـ	م
Nun	ن	ن	ـنـ	ن
Wau	و			و
Va	ۏ			ۏ
Ha	ه	ھ	ـھـ	ه
Ya	ي	ی	ـیـ	ي
Ye	ى			ى
Nya	ڽ	ڽ	ـڽـ	ڽ
Hamzah	ء			ء
ta marbutah	ة			ة

Table 2.2 Group-based and individual letters

Group Based	Individual
ب ت ث	ا
ج چ ح خ	ل
د ذ	ء
ر ز	ه
س ش	
ص ض	
ط ظ	
ع غ ڠ	
ف ڤ ق	
ك گ	
و ۏ	
ن ي	

Figure 2.3 Baseline

Figure 2.4 t is the highest value in the Histogram of Black Pixels

It is essential to have good knowledge of the Jawi writing styles in order to read and understand the contents of OJMs.

2.1.3 Traditional Practice of Reading OJMs

Traditionally, the OJMs are perused with the aim of identifying the morphology of characters, and to understand old Malay vocabularies. Thus, in the past, identification of characters in OJM were identified solely by human visual observation. However, readers encountered problems due to incomplete shape of the written characters, poor legibility of the characters, and degradation of the paper used in OJMs (Hj Yahaya, 2004). The second aspect in perusing OJMs is to identify words used in old Malay vocabularies.

2.1.4 Computer-Based Ways of Reading OJM

Recognition of Jawi characters in OJMs must be done in a systematic manner. It should start with the basic image pre-processing step such as noise removal and enhancement of the image quality. This is followed by performing line segmentation, word segmentation, characters segmentation, and finally recognition of the segmented characters. These processes are based on the sequence of steps for recognition of Arabic characters, as shown in Figure 2.6.

2.2 Arabic Character Recognition Taxonomy

The earliest research on Arabic characters was carried out by Nazif (1975). His research, however, had only focused on the theory and there was little discussion on the implementation. In 1980, Amin, Kaced, Haton, and Mohr (1980) developed the first Interactive Recognition of Arabic Character (IRAC) system which uses the chain-code tracing algorithm, as shown in Figure 2.5, to represent Arabic characters. Their success attracted other researchers to become involved in more complex research based on new theories and new ways of implementing their findings.

Alma'adeed (2006) developed a complete recognition system using neural network. Other researchers such as Snoussi-Maddouri, Amiri, Belaïd, and Choisy (2002) used Transparent Neural Network (TNN) for recognition of handwritten Arabic. Lopresti, Nagy, Seth, and Zhang (2008) suggested a new method of Symbolic Indirect Correlation (SIC) and Style Constrained Classification (SCC) for recognition of handwritten Arabic words.

Nasrudin, Petrou, and Kotoulas (2010) used the trace transformation for character recognition. The achieved results shows that trace signatures relatively improves the recognition rate as compared to affine moment invariant and angular radial transform. Azmi, Omar, Nasrudin, Idrus, and Wan Mohd Ghazali (2013) proposed a method to recognize Arabic/Jawi and Roman digits based on the features from triangle geometry. The proposed algorithm achieves over 90% of accuracy rate for each trained dataset used in evaluation. Salim et al. (2013) mainly focused on baseline detection as one of the important processes in recognition systems. The authors proposed a new framework for baseline detection and straightness for cursive handwritten text on the basis of analysing and extracting the direction features from sub-words in text skeleton. Abandah, Jamour, and Qaralleh (2014) proposed a system for recognition of cursive

handwritten Arabic words. The system uses rule-based segmentation algorithm, performs careful feature extraction during and after segmentation stage and leverages recurrent neural network (RNN) recognition engine to achieve higher rate of accuracy.

Researches on other types of character recognition techniques, especially for Roman characters, did not contribute much to Arabic character recognition, except only in cursive handwriting recognition. In their research on character recognition, Miled, Olivier, Cheriet, and Lecoutier (1997) used the Hidden Markov Model (HMM) which is based on global recognition and an analytical model. Benouareth, Ennaji, and Sellami (2006) added explicit state duration in HMM to enhance performance in the classification of characters. An online character recognition technique was developed by El-Hajj, Likforman-Sulem, and Mokbel (2005) using the Baseline-Dependent Features and HMM.

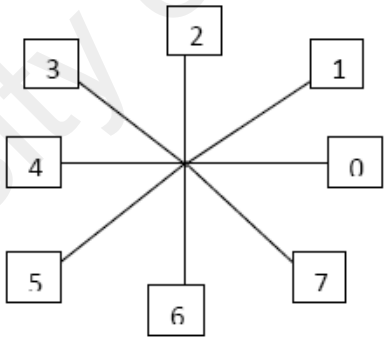

Figure 2.5 8-chain code numbering

In summary, Arabic-like characters recognition research can be categorized into two approaches — analytic approach and global approach.

• **Analytic approach**: it concentrates on the extraction and recognition of an individual character before combining it to form a "closest" word.

• **Global approach**: performs global recognition of the actual written word but not individual character recognition. This approach can overcome the problems

encountered in the segmentation process. It was originally used for speech recognition. The global approach can be further divided into two categories:

i. based on distance of words using dynamic programming; and

ii. based on probability as in the Hidden Markov Model.

2.3 Feature Extraction

Feature extraction involves changing the original domain of a sample into a different representation or domain, commonly known as Domain Transformation. Diversity of writing styles makes the statistical feature extraction ineffective. To overcome this problem, Mohamad, Manaf, Rauf, and Nasruddin (2015) used a numeric code representation technique to represent the range of primitive structure tilt in a zone. The evaluation results demonstrated that the numeric code representation performs best in representing the Jawi sub-word image as compared to three other feature selection techniques. Nasrudin, Omar, Liong, and Zakaria (2010) demonstrated that employing the wrapper method together with a number of ranking evaluation measurements to select the useful features is better than random selection of features or not selecting any feature at all. Discrete Wavelet Transform (DWT) and Fourier Transform (FT) are the most commonly-used techniques for feature extraction.

Yu, Muthukkumarasamy, Verma, and Blumenstein (2003) used FT to extract the features of input samples. In their research, every segmented character is processed using 2D-FT. Phase magnitude can only extract little information, whereas magnitude matrixes perform well for feature extraction. The following Formula 2.1 is used for extraction:

$$(Ff)(u,v) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{x=0}^{N-1} f(x,y)exp\left(-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right) \quad 2.1$$

The magnitude coefficient is obtained using Formula 2.2

$$F'(u,v) = \frac{|F(uv)|^2}{\sqrt{\sum_{u,v(u\neq 0)\vee(v\neq 0)}|F(uv)|^2}} \quad 2.2$$

Sim, Kim, and Oh (2000) used DFT to extract the power spectral image, which is then scale-normalized by threshold frequency (calculated with the power of area in circle). The power spectral image is used to get the texture descriptor. DWT uses translation and scaling on the Wavelet Analyzer which must be in indiscrete form, as shown in the following Formula 2.3:

$$\varphi(x) = \sum_{k=0}^{M-1} c_k \, \varphi(2x-k) \quad 2.3$$

The increment range is defined by a positive value of *M*, where *c* as a constant is wavelet order, and c_k is wavelet coefficient. Formula 2.4 is orthogonal to its own translation by this condition, $(x)\phi\,(x - k)dx = 0$, and also orthogonal to its scale by n $\varphi\,(x)\varphi\,(2x - k)dx = 0$, where function φ can be defined as in Formula 2.4.

$$\varphi(x) = \sum_{k} (-1)^k \, c_{1-k}\varphi(2x-k) \quad 2.4$$

Wavelet transform is getting more popular in recent years for feature extraction in character recognition systems.

2.3.2 Wavelet Transform in Character Recognition

Correia, De Carvalho, and Sabourin (2002) adopted wavelet transform in their Human-Perception handwritten character recognition model. They used wavelet to simulate the multi-resolutional capability of vision and also to extract features as fixation points. Wavelet is also used for image details extraction and this is done in three directions — vertical, horizontal, and diagonal. Other researchers led by Iftekharuddin and Parra (2003) also used wavelet transform for feature extraction. Another kind of wavelet transform - Wavelet Packet Transform – was used by Sasi (1997) for his proposed

handwritten character recognition system. He used this method because of the good features of wavelet packet - time-frequency, localization and compression. The same method was also adopted by Xin, Lijuan, Mou, and Dake (2009) to compare Zernike Moments method with the wavelet packet. They found that the wavelet packet achieved 94.9% recognition accuracy for handwritten characters as compared to 83.8% recognition accuracy rate using Zernike Moments. Zhang, Bui, and Suen (2005) used wavelet for feature extraction in their recognition system for handwritten numerals. They applied two types of discrete wavelet transform - 2D discrete wavelet transform, and 2D complex wavelet transform. The former was used to extract 2D wavelet features and the latter to extract detailed features. This feature extraction approach produces important geometrical features. This hybrid approach extracts directional information of the character image and keeps two important features of the character — the global and endpoints information. In a related research, Jose and Wahi (2013) also used wavelet transform for Tamil handwritten characters recognition. They focused on the use of daubechies wavelet coefficients and adopted two approaches: multiresolution analysis of image to study character image in different frequency bands, and localized basis to obtain localized features of the characters. Using the combined approach, they obtained more distinct traits as features for each character. Another research in the handwritten character domain was conducted by Singh and Budhiraja (2012) on the recognition of Gurmukhi characters, using several types of wavelet transforms. The outputs of these wavelet transformations were fed into the backpropagation neural network for recognizing characters. In his recognition system, El-Fishawy (2004) used orthogonal wavelets to obtain the characters waveforms and he matched these waveforms with the prestored waveforms, and then classified them based on closest match. Wavelet compression was also used to get important coefficients for any input character image. In this research, Euclidean distance is computed for the test images

against the training images and recognition is considered successful if the distance is less than global threshold value of 258. Biorthogonal discrete wavelet transform has also been used in a character recognition system by Kapogiannopoulos and Papadakis (1996). They found that by using biorthogonal discrete wavelet transform, the system can still recognize the character regardless of the writing styles - italic, bold, or fonts. Shelke and Apte (2011) used only single level of wavelet decomposition in their system to recognize Marathi Compound character. They combined the method with Neural Networks. Laine, Schuler, and Girish (1993) used another type of wavelet feature extraction form — orthonormal wavelet — for the recognition of complex annotations in maps and engineering drawings. Primekumar and Idiculla (2011) developed an online recognition system for handwritten Malayalam which also uses wavelet transform as their feature extraction tool. Input coordinates and angle features were used as wavelet transform inputs. These two inputs were combined to form a single feature before it was passed to the Simplified Fuzzy ARTMAP network for recognition. Stockton and Sukthankar (2000) also used wavelet Haar-based decomposition for online recognition of Japanese Kanji characters.

2.4 Unique Code Extraction

Unique code extraction uses a concept that is similar to that for the barcode or Quick Response (QR) code. QR code is a combination of two one-dimensional barcodes (Furht, 2011) which are arranged vertically and horizontally. Feature extraction of characters can form a unique code that represents a simplified version of feature extraction coefficients.

2.4.1 Code Matching Process

In the code matching process, two strings of numbers are compared and their distance is calculated. One of the well-known techniques in code matching is Hamming distance,

which was developed by Richard Hamming (Moon, 2005). This technique is defined by the number of positions of equal length in which the corresponding strings are different. With this number, we can calculate the percentage of the difference against the similarity. This approach is easy to implement if two uniquely produced strings need to be compared.

Another method of determining similarity between two strings of numbers is Euclidean Distance. This approach uses Pythagorean formula and it involves measuring the distance between two points.

2.5 Arabic-like Character Recognition Taxonomy

Arabic-like character recognition involves several sub-processes which include the line, word, and character segmentation processes followed by feature extraction, unique code extraction, and Hamming distance calculation processes. Figure 2.6 shows an overview of this general structure.

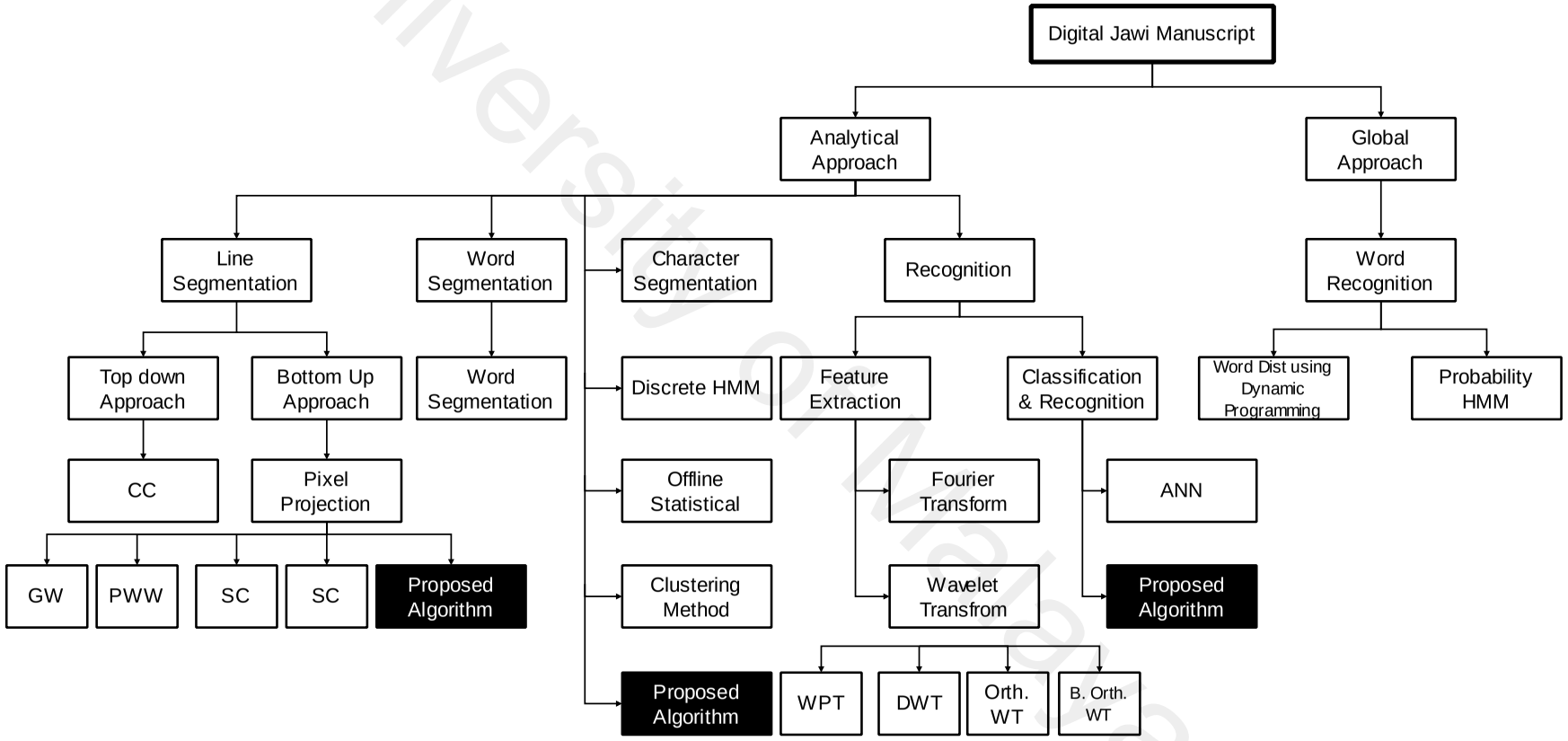

Figure 2.6 Taxonomy of Digital Jawi Manuscript

2.5.1 Line Segmentation

Line segmentation is a crucial process in this research. Any inaccurate segmentation will cause errors to be propagated to the final results. Line segmentation consists of two approaches — top-down and bottom-up approaches. In the top-down approach, segmentation is carried out based on a combination of the components that is achieved through the geometrical similarity between adjacent blocks. The bottom-up approach involves using pixels projection on clear separation between the lines.

Line segmentation of handwritten documents is an important process to handle the problem of writing fragmentation that results from the connectivity between characters or letters. Moreover, the problem becomes more challenging when it involves Jawi characters which have many diacritical marks. Li, Zheng, Doermann, and Jaeger (2006) used the threshold value method in their projection of horizontal pixels histogram. They changed a binary image into a gray-scale image, and used Gaussian windows to determine the segmentation points. Lines of text are extracted using level set (refer to Figure 2.7). In their research, different types of texts were used and were successfully segmented.

Black pixels projection had been used for Arabic characters (Al Abodi & Li, 2014; Ismail & Abdullah, 2012). Arivazhagan, Srinivasan, and Srihari (2007) presented the steps involved in their algorithm, where the initial set of lines were extracted using Projection Piece Wise, and all overlapped characters were disconnected by following the rules below:

i. Gaussian Bivariate model of these lines; and
ii. Calculate the probability values using matrix distance.

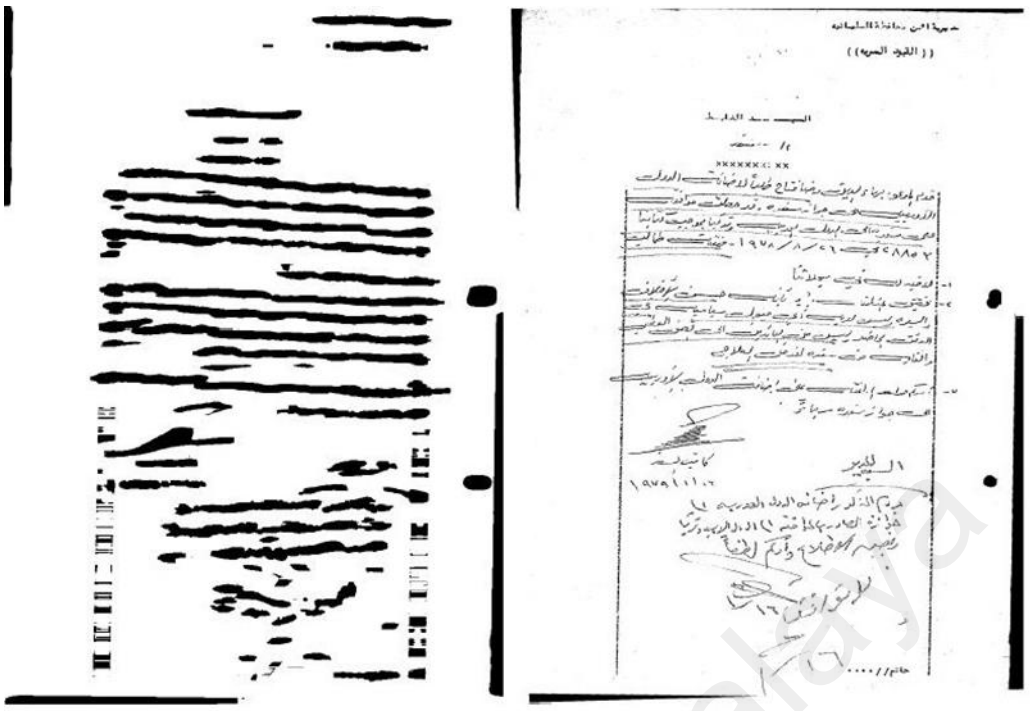

Figure 2.7 Level Set

They applied their method on 720 documents including handwritten Romanised texts, which consist of 11,581 lines. They achieved 97.31% accuracy for correct line segmentation as shown in Figure 2.8, Figure 2.9 and Figure 2.10.

Abuhaiba, Datta, and Holt (1995) proposed a method that involves using strokes (Refer to Figure 2.11). In the method, the main stroke must first be extracted and must follow the same direction. The secondary strokes are then extracted and are combined with the main strokes to form the complete text groups.

Öztop, Mülayim, Atalay, and Yarman-Vural (1999) used the Repulsive Attractive Network on old Ottoman documents and handwritten Latin text to extract the baselines. They succeeded in extracting most of the baselines.

Figure 2.8 Original

Figure 2.9 Output

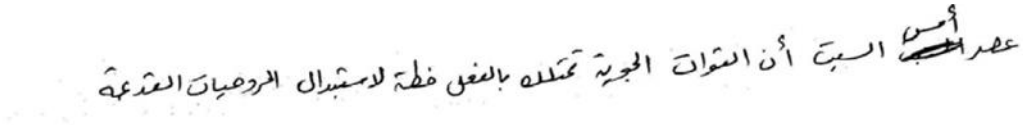

Figure 2.10 A line of words after the rotation process (Diacritical marks remained)

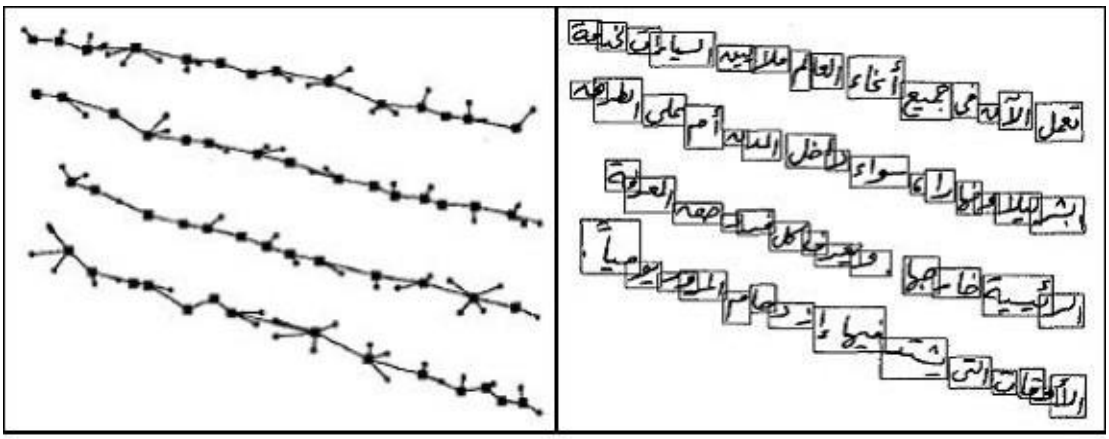

Figure 2.11 Secondary Strokes on Primary Stroke

The outcomes from the researches were very encouraging, as shown in Table 2.3 However, none of the studies focused on overlapping characters between two lines.

Table 2.3 Researches Conducted on Line Segmentation

Researcher (s)	Sample	Experiment	Accuracy	Comment
(Arivazhagan et al., 2007)	11,581lines in 720 documents include English, Arabic and kid's handouts	Accuracy of *cut-through* matching with component.	97.31%	Normal component covers two or more lines or in between two lines of texts
(Li et al., 2006)	More than 10,000 documents in Arabic, Hindi, and Chinese in a variety of handwriting styles.	Line segmentation point detected if *ground-truth* matches with corresponding lines for 90% of their pixels	2,691 *ground-truth* lines, 2,303 (85.6%) lines segmented.	14.4% errors because of two overlapped lines and noise caused by scanner.
(Öztop et al., 1999)	Gray-level image of Ottoman old manuscripts and Latin handwriting	Document feeds into Repulsive Attractive Network to extract baseline.	Successful baseline extraction although noise is present, and overlapped characters in lines.	No discussion
(Zahour, Taconet, Mercy, & Ramdane, 2001)	More than 100 tales (1,000 lines) of Arabic handwriting of different writers	Implemented in C on 200MHz processor	97% accuracy rate in line detection	Error of segmentation caused by different baseline orientations, overlapped characters and diacritical marks

As mentioned in Section 1.2, old Jawi manuscripts were written with no baselines, but overlapping lines are evident throughout the documents, and these cause inaccurate line segmentation. This segmentation problem will be propagated and reflected in the final result on recognition accuracy.

2.5.2 Word Segmentation

Words in Arabic handwriting are separated by spaces (Al-Badr & Mahmoud, 1995), although some of the words have sub-word(s). In addition, there are also spaces between sub-words which further complicate the word separation process. Besides, spaces in Arabic writing are too narrow, hence, it is difficult to distinguish two adjacent words.

Romeo-Pakker, Ameur, Olivier, and Lecourtier (1995) proposed the following vertical projection Formula 2.5 for word segmentation,

$$v(j) = \sum_{iL} g(i,j) \qquad 2.5$$

where iL refers to the column in the detected lines. The distance between words is bigger than the distance between sub-words, as shown in Figure 2.12 .

Kandil and El-Bialy (2004) proposed a word segmentation technique which is not dependent on baselines, and uses vertical peak identifier, as shown in Figure 2.15 .

Farooq, Govindaraju, and Perrone (2005) found it easier to separate words by using a collection of components. The result is shown in Figure 2.14 . Table 2.4 summarizes the results of all the studies mentioned above.

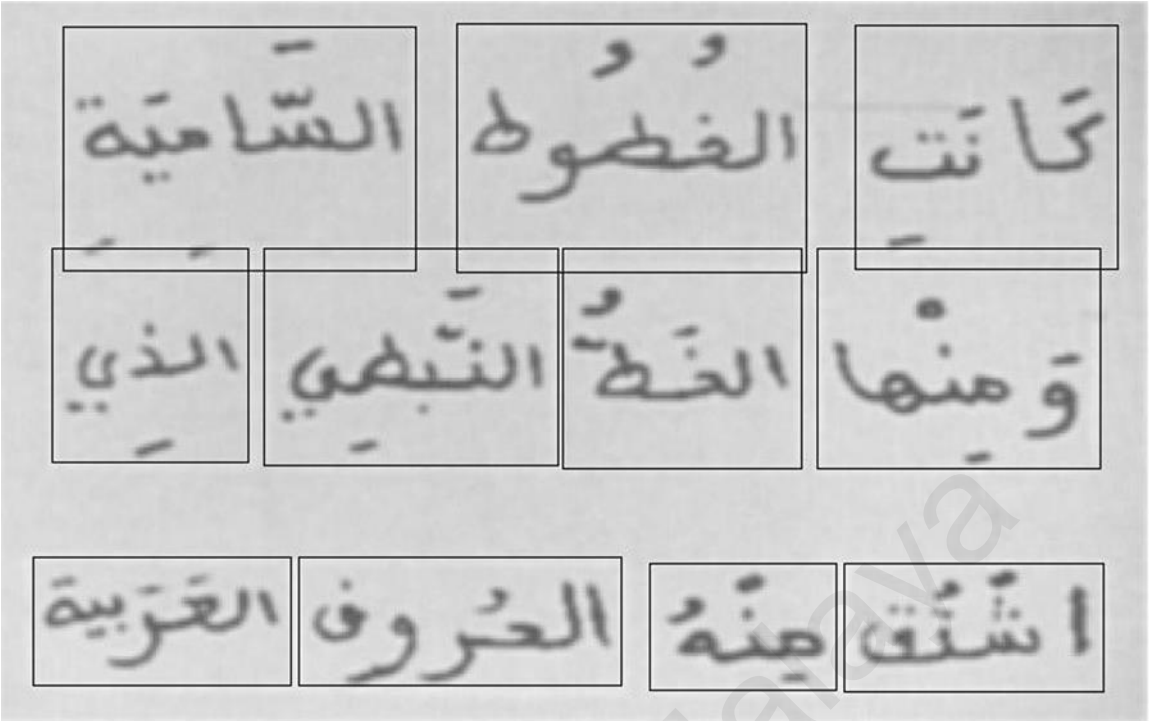

Figure 2.12 Words and Sub-Words Segmentation

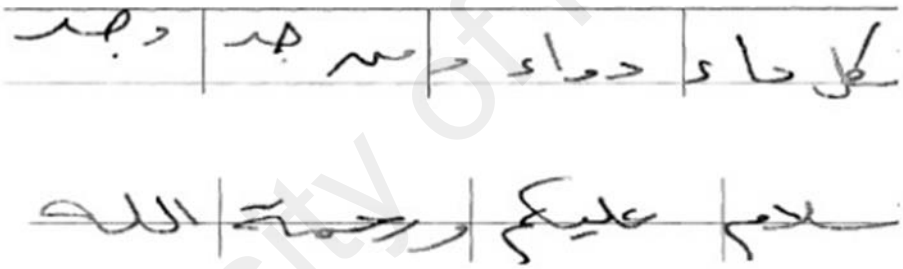

Figure 2.13 Word Segmentation of Handwritten Arabic Words

Figure 2.14 Word Segmentation of Handwritten Arabic Words

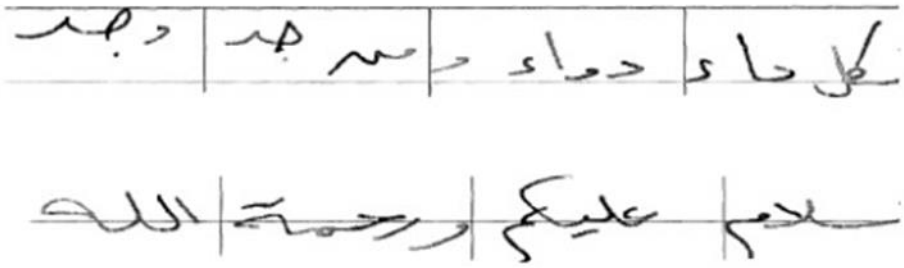

Figure 2.15 Word Segmentation of Handwritten Arabic Words

Table 2.4 Summary of Researches on Word Segmentation

Research	Method	Sample	Experiment	Result	Error
(Romeo-Pakker, Ameur, et al., 1995)	Vertical projection	Arabic handwriting	Sample set of handwriting to execute algorithm	63.5% accuracy	23.4% of words were wrongly segmented
(Kandil & El-Bialy, 2004)	Vertical projection	Arabic handwriting	Maximum vertical projection detection	No discussion	No discussion
(Farooq et al., 2005)	Components collection	Arabic handwriting	Component labeled and convex hull detected	No discussion	No discussion

2.5.3 Character Segmentation

Gouda and Rashwan (2004) studied the use of Discrete Hidden Markov Modelling on printed uniform Arabic characters. They succeeded in segmenting 99% of the characters.

Goraine, Usher, and Al-Emami (1992) used offline statistical analysis on printed characters and easily segmented them because of the uniformity of the characters. Syiam, Nazmy, Fahmy, Fathi, and Ali (2006) adopted the clustering method which uses K-Algorithm for histogram projection of pixels. They achieved 90% accuracy in segmentation, but they failed for two types of characters that are found especially at the end of the word.

2.6 Conclusion

There was no standard writing style in old Jawi manuscripts. As a result, it is common to find irregularities such as overlapping between lines, insufficient space between words or sub-words, and no standard size for each character. In considering the preservation of digitized copies of old Jawi manuscripts, six major issues — line segmentation; word segmentation; character segmentation; feature extraction; unique code extraction; and string matching process — must be addressed.

In line segmentation, baseline detection is a commonly-used technique, and it is also used for further segmentation process. Vertical histogram projection is widely used in word segmentation and involves empty space or minimum separation point or space for detecting segmentation points between words. The character segmentation process involves further analysis such as contour analysis, and statistical analysis. Both approaches are concerned with details on pixel distribution.

Wavelet and Fourier transformation are the best-known signal transformation techniques. However, wavelet transform has received more attention in recent years. Transformation of characters pixel distribution can be used to determine the uniqueness of pixel distribution, and to represent it in numbers. Unique code extraction is used to translate feature extraction output into a simple representation of numerical strings. These unique strings cause the matching algorithm to perform faster and more efficiently.

The literature review has provided relevant information for the development of the most suitable algorithms to fulfil the main aim and objectives of this research. CHAPTER 3 discusses the development of these algorithms.

CHAPTER 3

LINE SEGMENTATION PROBLEMS

Overlapped characters between upper and lower lines is a major problem in line segmentation of cursive characters such as Jawi which is similar to Arabic characters. This chapter discusses the proposed fast approach using a tangent value to find a separation point (SP) for accurate line segmentation without any data loss. This approach can be adopted for designing a dedicated Jawi character recognition chip which will produce a high level of accuracy in character segmentation. The rest of this chapter is organized as follows: The next section presents a general introduction to Jawi. Section 3.2 reviews a number of related works on line segmentation. Section 3.3 discusses an effective approach for line segmentation in handwritten Jawi text. Section 3.4 discusses the analysis, evaluation, and results of the proposed approach. Section 3.5 concludes the chapter.

3.1 Introduction

The arrival of Islam in the Nusantara archipelago along with Arabic script had greatly enriched Indonesian literature (Pudjiastuti, 2006). During that time, some sections of the Nusantara society had begun to express their thoughts through a new writing system that had been modified from Arabic script to suit the local pronunciations. One of the Arabic script forms which was modified to suit local requirements emerged as the Jawi script. Jawi consists of 28 characters which originated from Arabic characters, and six characters which are unique to the Jawi script. Malay manuscripts are defined as handwritten documents in Jawi which were written from the beginning of the 14th

century to early 20th century when the increasing influence of the Western countries and the introduction of the printing machines stopped further production of the manuscripts. The oldest evidence of Jawi script usage is the Terengganu inscription dated in 1303 (Omar, 2001).

Although the Jawi script has generally been replaced by the Roman script today, Jawi documents are still a rich source of information on South East Asian history and culture (Toru, 2005). In Malaysia, the Jawi script is still widely used for various purposes such as in inscriptions, chronicles, genealogies, tales, religious texts on Islam, diplomatic treaties, legal documents, contracts, petitions and other administrative documents, newspapers and magazines. Therefore, Jawi character recognition system is necessary in order to digitize and preserve Jawi writings for learning and research.

In almost all character recognition systems, especially those for cursive character recognition, segmentation is the most important issue. Inaccuracy in segmentation - line segmentation, word/sub-word segmentation, and character segmentation - will result in recognition errors. In this research, the many overlapped characters over the lines in old Malay manuscripts is the main problem to resolve to achieve our research aim. In this context, it is crucial to plan and design a very accurate method in the first stage of segmentation — line segmentation. Artificial intelligence is not used for our analysis because it involves high computational complexity. In our approach, the image is not analyzed using morphological tracing, as had been done by Ymin and Aoki (1996); there is no calculation of pixel averages as had been done by other researchers (Adnan, 1991; Cheung, Bennamoun, & Bergmann, 1998). Our technique uses histogram projection regardless of the character orientation or line skew. The histogram is normalized to omit false local minimum segmentation point (SP). Our algorithm was developed in order to improve accuracy and speed, while maintaining the quality of the

segmented text lines. The next section reviews some previous works on text line segmentation of handwritten documents.

3.2 Previous Works on Text Line Segmentation

Text line segmentation of handwritten documents can generally be categorized into bottom-up segmentation or top-down segmentation. In the bottom-up approach, the connected components-based methods merge neighbouring connected components using simple rules on the geometric relationship between neighbouring blocks (Feldbach & Tonnies, 2001; Likforman-Sulem & Faure, 1994; Likforman-Sulem, Hanimyan, & Faure, 1995; Louloudis, Gatos, Pratikakis, & Halatsis, 2006; Nicolas, Paquet, & Heutte, 2004). In the top-down approach, the projection-based methods might be the most suitable algorithm for machine-printed documents since the gap between two neighbouring text lines in machine-printed documents is usually wide, hence, the text lines are easily separable (Arivazhagan et al., 2007; Sesh Kumar, Namboodiri, & Jawahar, 2006; Timar, Karacs, & Rekeczky, 2002; Tripathy & Pal, 2004; Yanikoglu & Sandon, 1998; Zahour et al., 2001). However, these projection-based methods cannot be directly used on handwritten documents, unless there are wide gaps between the lines or the handwritten lines are straight.

Likforman-Sulem and Faure (1994) proposed an approach based on perceptual grouping of connected components of black pixels. Text lines are iteratively constructed by grouping the neighbouring connected components based on certain perceptual criteria such as similarity, continuity, and proximity. Therefore, local constraints on the neighbouring components are combined with global quality measures. To handle conflicts, the technique merges with a refinement procedure to perform both global analysis and local analysis. According to the authors, the proposed technique cannot be used on degraded or poorly structured documents.

Nicolas et al. (2004) viewed the text line extraction problem from the artificial intelligence aspect. The aim is to cluster connected components in a document into homogeneous sets, corresponding to the text lines of the document. To resolve this problem, a search is applied over the graph that is defined by the connected components as vertices, and the distances between them as edges.

Feldbach and Tonnies (2001) proposed a method for line detection and segmentation in historical church registers. This method is based on local minima detection of connected components and is applied on a chain code representation of the connected components. The idea is to gradually construct line segments until a unique text line is formed. This algorithm is able to segment text lines close to each other, touching text lines, and fluctuating text lines.

Likforman-Sulem et al. (1995) proposed an iterative hypothesis validation strategy based on Hough transform. The skew orientation of handwritten text lines is produced by applying the Hough transform to the centre of gravity of each connected component in the document image. If the most nearest neighbours of the components in the alignment string belong to the group of components forming the alignment, the alignment has both properties of direction continuity and proximity, and it will be accepted as a text line. This enables several text line hypotheses to be generated. A validation is then performed to eliminate incorrect alignments between connected components using contextual information such as proximity and direction continuity criteria. The authors stated that this technique is able to detect text lines in handwritten documents, which may contain lines oriented in different directions, with erasures, and with annotations between the main lines.

Louloudis et al. (2006) presented a text line detection method for unconstrained handwritten documents based on a strategy that involves three distinct steps. The first

step involves preprocessing for image enhancement, connected component extraction, and average character height estimation. In the second step, a block-based Hough transform is applied for the detection of potential text lines, while in the third step, any possible false alarm is corrected. A grouping method of the remaining connected components uses the centres of gravity of the corresponding blocks. The performance of the proposed method is evaluated by using a consistent and robust technique that compares the text line detection results with the corresponding ground truth annotation.

Another detection technique divides the text image into columns. Zahour et al. (2001) performed a partial projection on each column. A partial contour method is then used to detect the separating lines in the direction and opposite direction of the writing. Tripathy and Pal (2004) used a projection-based method in each column, and combined the results obtained for adjacent columns into a longer text line. In this way, the document is divided into vertical stripes. The width of a stripe is calculated by analyzing the heights of the histograms obtained from different components of the document. Stripe-wise horizontal histograms are then computed and the relationship of the peak-valley points of the histograms is used for line segmentation.

In the algorithm proposed by Arivazhagan et al. (2007), an initial set of candidate lines from the piece-wise projection profile of the document is first obtained. The lines go around any obstructing handwritten-connected component by associating it to the line above or below. A decision on associating such a component is made by: (i) modeling the lines as bivariate Gaussian densities and evaluating the probability of the component under each Gaussian statistical distribution, or (ii) basing on the probability obtained from a distance metric. The proposed method is robust enough for handling skewed documents and touching lines.

Sesh Kumar et al. (2006) presented a graph cut-based framework using a swap algorithm to segment document images containing complex scripts such as those in Indian languages. The text block is first segmented into lines using the projection profile approach. The framework enables learning of the spatial distribution of the components of a specific script, and can be adapted to a specific document such as a book. Moreover, they can use both corrections made by the user as well as any segmentation quality metric to improve segmentation quality.

Yanikoglu and Sandon (1998) first searched for the handwriting text line boundaries and then processed each text line in turn. The boundary between two text lines is not a straight line if the text lines are touching or overlapping. To find the exact boundary between two text lines, they searched for the rough boundary location by analyzing the horizontal pixel density histogram of the line. They then applied a contour-following algorithm within that zone to find the exact boundary. The contour-following algorithm is modified to operate within a rectangular zone. It is also changed to force a cut at the half-line, when necessary, in order to separate text lines that are touching and cannot be separated otherwise.

Timar et al. (2002) used their algorithm to localize lines by computing the horizontal histograms for the entire image at a number of relevant skew angles; the angle and position where the histograms have local minima are then selected as the location between lines. Calculation of the horizontal histograms was done using the traditional histogram, executable on DSPs. They refined the line-finding algorithm by using a method of blurring the words without affecting their locations. They computed the pseudo-convex hull of each word using the HOLLOW template. The horizontal histogram computed on the pseudo-convex hulls is smoothed further using a sliding window. The local maximum of the histogram can then be located since it corresponds to the location of the lines. Thresholds are specified to associate all maxima with a line.

Weliwitage, Harvey, and Jennings (2005) used a Cut Text Minimization (CTM) method for text lines segmentation. The CTM method finds a path or cut line in between the text lines to be separated, and this minimizes the text line pixels cut by the segmentation line, especially the descenders from the upper line and the ascenders from the lower line. The method attempts to track around ascenders or descenders to avoid cutting them. If the deviation is too great, the segmenter aborts and continues its forward path. A rough estimate of text line separations is first obtained using vertical projection histograms.

Li et al. (2006) proposed an approach for text line detection by adopting a state-of-the-art image segmentation technique. They first convert a binary image into a gray-scale image using a Gaussian window to enhance the text line structures. Text lines are extracted by evolving an initial estimate using the level set method. Results from the preliminary experiments show that their method is more robust compared to a bottom-up connected component-based approach. Moreover, the method is script independent, and this had been qualitatively confirmed by testing it on handwritten documents of different languages, such as Arabic, English, Chinese, Hindi, and Korean. Statistical results show that the algorithm performs consistently under reasonable variation of skew angles, character sizes, and noise.

Abuhaiba et al. (1995) proposed a method based on the shortest spanning tree search. The method involves building a graph of main strokes of the document image and searching for the shortest spanning tree of this graph. In this method, it is assumed that the distance between the words in a text line is less than the distance between two adjacent text lines.

Almost all text line segmentation approaches assume that text lines are straight. However, projection-based methods can be extended to deal with curved text lines.

Generally, it produces better result than the simple projection methods, but the merging of detected line segments of adjacent columns may cause ambiguity. As such, it is still difficult to obtain a reasonably good result.

3.3 Proposed Line Segmentation Approach

The line segmentation process is crucial in a character recognition system and hence must be performed correctly in order to prevent errors from being propagated to the next process. In our approach, the tangent in the graph will be calculated to find other representations — tangent representation of either 1 or -1. Our approach is aimed at eliminating false local minima that might exist in our graph, by collecting the number of 0s (black pixels) of corresponding rows using Equation 3.1 and Equation 3.2. These values will be stored temporarily in one array.

$$h_r(k) = \sum_{c=0}^{N-1} I_k(r, c) \tag{3.1}$$

$$h'_r(k) = N - h_k(r) \tag{3.2}$$

where N = number of columns, r = row, c = column, $h_k(r)$ = horizontal projection of 1s, and $h'_k(r)$ = horizontal projection of 0s. Values in the array (histogram) will be used to calculate the new value (1 or -1) using Equation 3.3

$$f(x) = \begin{cases} 1 & m > 0 \\ -1 & m < 0 \end{cases} \tag{3.3}$$

where m is $h'_{k+1} - h'_k$ and k is the corresponding row. These values will be used to form another graph. The value of $f(x)$ will be tested and will be given a new value if it fulfils the following rule:

Rule 1: if the number of adjacent -1 values does not exceed the constant value k, then it will be replaced by a new value, that is 1.

The purpose of Rule 1, above, is to eliminate false local minima that could disrupt the line segmentation process (with $k = 3$, which means that false local minima will not exceed value 3 in tangent tracing based on the results of our experiment). The final step is to trace the changes of tangent in our new value and this will point to the location of the valid separation point. Figure 3.1 shows the raw image of an old manuscript.

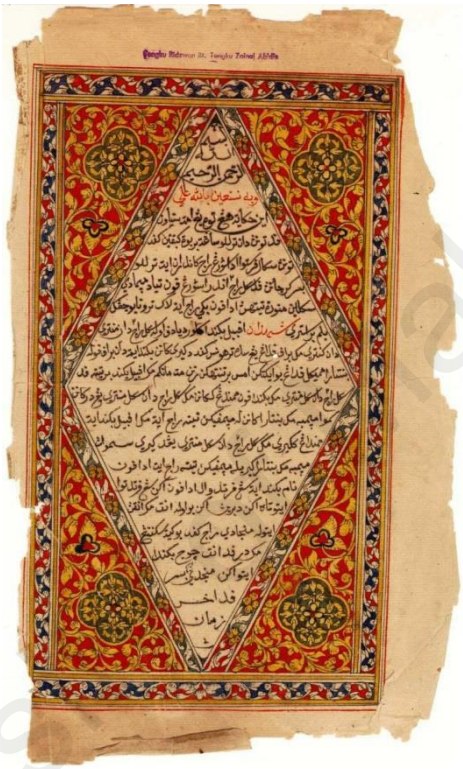

Figure 3.1 Old Jawi manuscript with size 1277 x 774 pixels

The first step in our experiment is to crop out our Region of Interest (ROI) from the image shown in Figure 3.1 and convert it into a binary image, as shown in Figure 3.2 (a) The ROI clearly shows that the presence of overlapped characters between adjacent lines is due to the way the line is written. Figure 3.2 (b) also shows that there are two false local minima (circled) between lines 100 to 120. Figure 3.3 shows the output when the graph in Figure 3.2 (b) is converted into tangent representation. The figure shows that there is an active change of value from 1 to -1, and vice versa. These active changes might produce inaccurate vertical segmentation point if the conversion had been done without eliminating the false local minimum. If the elimination had been done, the graph is more stable, as shown in Figure 3.4. The results of full line segmentation are

shown in Figure 3.5. The pseudocode for the overall line segmentation procedure is listed in Algorithm 3.1.

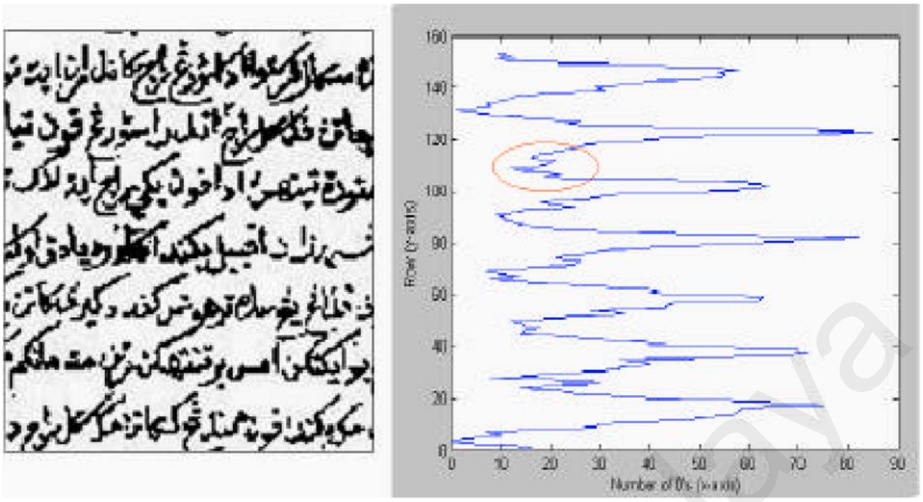

Figure 3.2 (a) Binary image of ROI of old manuscript (b) Graph row versus number of 0's

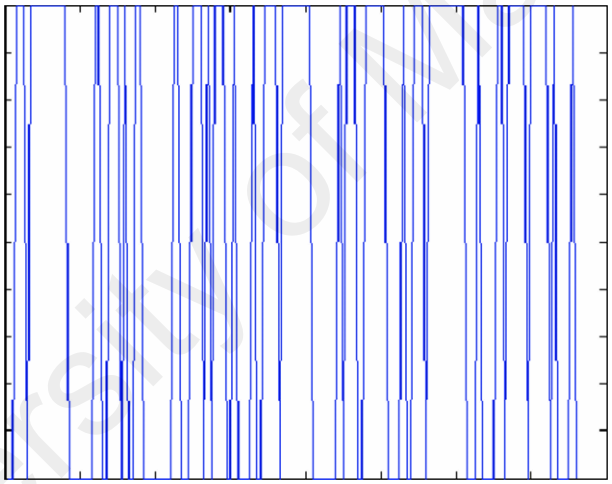

Figure 3.3 New representation of tangent versus row (before elimination of false local minimum)

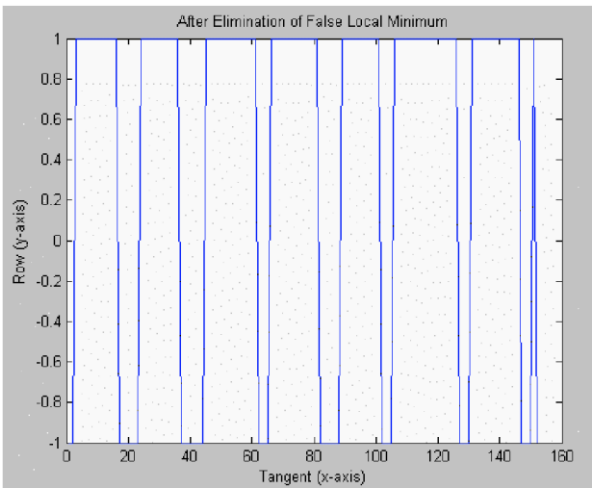

Figure 3.4 Tangent versus row (after elimination of false local minimum)

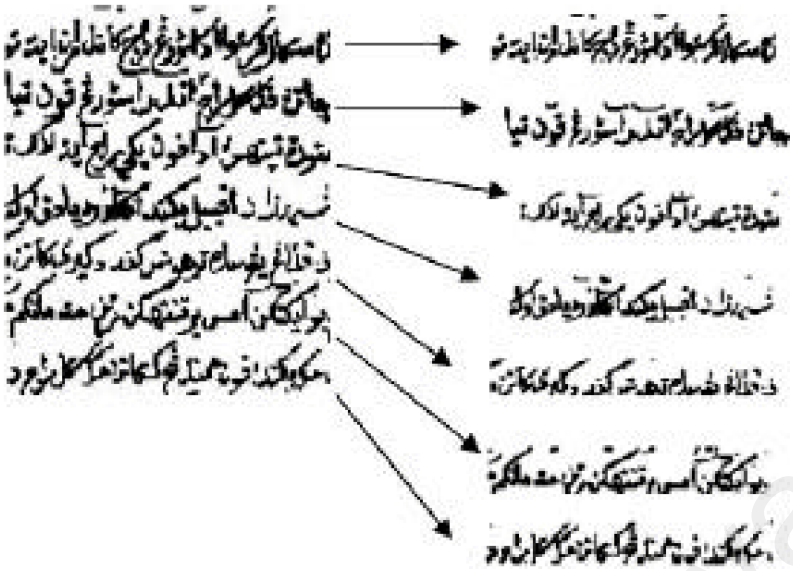

Figure 3.5 Results of full line segmentation

Algorithm 3.1 Line Segmentation

```
1.  Image (Max_Row, Max_Column)=Read_Image(Source)
2.  For each row in Image                    //procedure for generating the linear histogram
3.      For each column in Image
4.          If Image(row, column)=0 Then
5.              Increment Histogram(row) // Linear Histogram
6.          End If
7.      End For
8.  End For
9.  For i=1 to Max_Row   //procedure for calculating the gradient of histogram graph
10.     If Histogram (i+1)>Histogram(i)
11.         Histogram(i)=+1
12.     Else    Histogram(i)=-1
13.     End If
14. End For
15.
16. Set Threshold=k; Count=0
17. For i=1 to Max_row                  // procedure for eliminating false local minima
18.     If Histogram(i)=-1
19.         Increment Count
20.         Add i to Negative_List
21.     End IF
22.     If  Histogram(i)=+1
23.         If Count<k
24.             For each j in Negative_List
25.                 Histogram(j)=+1
26.             End For
27.         End If
28.     Count=0; Clear Negative_List
29.     End If
30. End For
31. For i=1 to Max_Row                  //  procedure for extracting lines
32.     If  histogram (i)+histogram(i+1)=0
33.         Add i to Seperation_Point_list
34.     Then
35. End For
36. For each point  in  Seperation_Point_list
37.         Extract the corresponding line
.38. End For
```

3.4 Evaluation Results for Line Segmentation

The algorithm proposed by Zahour et al. (2001) was tested on 100 samples of texts that consist of 1,000 lines. The results show 97% accuracy rate in correct line segmentation. The small number of errors was caused by baseline-skew variability, overlaps between characters, and the presence of diacritical marks in the first column. The association of the diacritical mark to a text line also causes serious errors when the mark is distant from the separating border line.

Tripathy and Pal (2004) used 1,627 text lines in single-column pages that also have different writing styles. Text lines segmentation accuracy rate is calculated by drawing boundary lines between two consecutive text lines. The line segmentation accuracy rate was then manually calculated by viewing the text line displayed on the computer screen. If all text lines are extracted correctly, segmentation accuracy rate is considered 100%. Of the 1,627 lines tested, 984 lines were segmented correctly.

Arivazhagan et al. (2007) tested their algorithm on 11,581 lines contained in 720 documents which include English, Arabic, and children's handwriting. Their results show 97.31% accuracy in line segmentation. In the experiment involving over 200 handwritten images with 78,902 connected components, the results show that 98.81% of the components are associated to the correct lines. Experiments were also conducted on 300 exam essays written on ruled-lined paper to test the robustness of the algorithm. The results show 96.3% accuracy in line segmentation. Furthermore, the results show that the segmentation algorithm is language independent. A 98.62% accuracy rate was obtained when test was conducted on 120 handwritten Arabic images. The proposed algorithm is also able to associate most of the dots above or below a word to the correct lines.

Sesh Kumar et al. (2006) tested their algorithm on 256 scanned pages from a Telugu book entitled “Aadarsam" which was printed in 1973. They calculated line segmentation accuracy rate using a segmentation quality metric. Hardly any segmentation correction was needed after the first 44 pages, as line segmentation was done correctly for the remaining 212 pages.

Weliwitage et al. (2005) tested their method on 30 images from the NIST special database which contains data in 34 text boxes from 2,100 forms scanned at a resolution of 300 pixels per inch and saved in binary format. Text boxes in the form which correspond to a text paragraph of 52 words were extracted for text line segmentation. The test results show that 183 text segmenting lines and 213 text lines in the images were correctly segmented into 176 lines, giving an accuracy rate of 96%.

Likforman-Sulem and Faure (1994) tested their method on handwritten documents. They detected fluctuating text lines, sloped annotations, or annotations added between main lines. Line fluctuation combined with proximity of text lines may cause merged lines. Timar et al. (2002) tested their algorithm on 10 pages of a handwriting database containing 7,000 words from the LOB (Lancaster-Oslo / Bergen) corpus written by a single writer. The experiments were conducted using the MatCNN simulator in the Matlab software. The line segmentation algorithm correctly segmented each line in every page.

Li et al. (2006) tested their algorithm on more than 10,000 diverse handwritten documents in different scripts such as Arabic, Hindi, and Chinese. During testing, if a ground-truth line and the corresponding detected line share at least 90% of the pixels, a text line is considered to be detected correctly. From a total of 2,691 ground-truth lines, their method correctly detected 2,303 (85.6%) lines. At the text line level, the connected component-based method performs much worse as only 951 (35.3%) text lines were

detected correctly. The errors were attributed to signatures, extensive overlapping of two adjacent text lines, correction in the gap between two lines, and the severe noise introduced during scanning.

Louloudis et al. (2006) proposed a text line detection method which uses a Block-based Hough transform approach to test unconstrained handwritten Greek documents containing 20 images, from the historical archives of the University of Athens. The corresponding text line detection ground truth was manually created. They detected 450 text lines in the images reflecting an accuracy rate of 96.87%. The difficulties encountered during the extraction of text lines were due to the variety of accent marks appearing above or below the body of the text line, and the small difference of the skew angle in the text lines.

Nicolas et al. (2004) reported that overlapping text lines, and text lines where the interline distance is smaller than the intraline distance, will cause segmentation errors. Only well separated text lines can be correctly segmented. Weliwitage et al. (2005) tested their algorithm - which used a chain code representation - on the text in church registers which span 300 years. The algorithm was applied to images from 61 paragraphs contained in seven pages. The 61 paragraphs contain 300 lines. The algorithm produced good results with consistent values even for different handwriting styles. The results show that 222 lines (90%) of 246 lines in 49 paragraphs containing six different handwritings, were reconstructed correctly.

The experimental results are summarized in APPENDIX A, Table A.1, APPENDIX B Table B.1 and Table B.2. Experiments using the proposed line segmentation algorithm were conducted in the Matlab environment for simulation purpose, as shown in the last row in Table A.1. Segmentation is done much faster using the proposed approach when compared to other techniques. This is because the proposed approach is able to dispense

with the contour and boundary tracing procedures. Hence, there is increased productivity and performance because of the reduction in processing overhead. Also, the algorithm is lightweight because only simple mathematical calculations are involved.

3.4.1 Time Complexity Analysis

To assess the computational complexity of the proposed line segmentation algorithm a time complexity analysis based on the Big O notation were conducted. The details and result of the analysis are shown in Table 3.1.

3.5 Conclusion

Our algorithm compares favourably against other techniques for line segmentation. It performs segmentation faster and more accurately although there are a few cut-off characters which can be eliminated through local analysis of each line. On-chip implementation of the algorithm is feasible while ensuring fast processing in view of the simplicity of our approach.

Table 3.1 Time complexity analysis for line segmentation algorithm

	Line Segmentation **(m: number of rows, n: number of columns)**	**Number of Iterations**
1.	Image (Max_Row, Max_Column)=Read_Image(Source)	
2.	**For** *each row* in Image //procedure for generating the linear histogram	***m×n***
3.	**For** *each column* in Image	
4.	**If** *Image(row, column)*=0 **Then**	
5.	*Increment Histogram(row)* // Linear Histogram	
6.	**End If**	
7.	**End For**	
8.	**End For**	
9.	**For** *i=1 to Max_Row* //procedure for calculating the gradient of histogram graph	***m***
10.	**If** *Histogram (i+1)>Histogram(i)*	
11.	*Histogram(i)=+1*	
12.	**Else** Histogram(i)=-1	
13.	**End If**	
14.	**End For**	
15.		
16.	*Set Threshold=k; Count=0*	
17.	***For*** *i=1* ***to*** *Max_row* // procedure for eliminating false local minima	***m***
18.	**If** *Histogram(i)=-1*	
19.	*Increment Count*	
20.	*Add i to Negative_List*	
21.	**End IF**	
22.	**If** *Histogram(i)=+1*	
23.	**If** *Count<k*	
24.	**For** each *j* in *Negative_List*	
25.	*Histogram(j)*=+1	
26.	**End For**	
27.	**End If**	
28.	Count=0; Clear *Negative_List*	
29.	**End If**	
30.	**End For**	
31.	**For** *i=1 to Max_Row* // procedure for extracting lines	***m***
32.	**If** *histogram (i)+histogram(i+1)=0*	
33.	Add *i* to Seperation_Point_list	
34.	**Then**	
35.	**End For**	
36.	**For** *each point* in *Seperation_Point_list*	
37.	*Extract the corresponding line*	
.38.	**End For**	
	TIME COMPLEXITY CALCULATION	
	O(m×n+m+m+m)=O(m×n)	

CHAPTER 4

CHARACTER SEGMENTATION

This chapter mainly focuses on the segmentation of character, which when dealing with Old Jawi Manuscripts can be a most challenging and intricate task. The proposed approach for handling character segmentation encompasses several techniques such as histogram generation, histogram gradient sign normalization, and sliding window. The rest of this chapter is organized as follows: Section 4.1 introduces the sliding windows technique in character recognition, based on information gathered from the literature review. Section 4.3 presents details of the proposed character segmentation approach outlining the different steps involved. Section 4.4 provides the justification for the proposed approach. Section 4.5 presents the results of experiments using the proposed approach. Finally, Section 4.5.1 concludes this chapter.

4.1 Introduction

In our approach, histogram normalization and the sliding window technique were used for Jawi character segmentation. The projection histogram technique is used for horizontal and vertical segmentation in character and word recognition systems (Casey & Lecolinet, 1996). The vertical projection histogram method segments the document vertically by detecting the space between the different characters. This method can also locate vertical strokes in printed documents or any region of the lines in handwritings (Pal & Datta, 2003). Zeki (2005) stated that the projection histogram analysis of text lines has been used as the basic method for segmenting non-recursive writing. However,

it is not suitable for segmenting slanted characters, which are commonly found in handwritten documents.

4.2 Related Works

Several methods have been proposed to segment Arabic characters, especially those methods which use vertical projection histogram. Gouda and Rashwan (2004) found that each word can be segmented into individual characters based on the baseline and the use of a vertical histogram. Syiam et al. (2006) implemented a clustering technique (k-means algorithm) on the vertical histogram. This improves the performance of the histogram technique when applied in the recognition of handwritten characters. The characters are clustered to identify similarities among the characters. Romeo-Pakker, Miled, and Lecourtier (1995) proposed two methods for segmenting Arabic characters. The first method detects the junction in each connected character, and the second method detects the upper contour of each word. These methods were also used by Zahour et al. (2001) and Omidyeganeh, Nayebi, Azmi, and Javadtalab (2005).

The sliding window technique is a common localization technique in the signal processing domain (Su, Zhang, Guan, & Huang, 2009). Since the character string is written in a certain direction, a shifting window, called sliding window, which follows the same order, can be used to draw a zone of interest from which features are extracted. Generally, the height of the sliding window is the same as that of the text line. The width of the sliding window and the shift step are assigned by the researchers or determined through experiments.

Bushofa and Spann (1997) used sliding windows to find the angle that is formed by joined characters. Although this method produces promising results, its success rate in finding the correct angle is affected by the noise in the image. As an alternative, they used a segmentation algorithm which is more reliable than the histogram method. The

correct position of segmentation is selected and indicated based on the angle that is formed by each pair of joined characters.

Pechwitz and Margner (2002) proposed a method that is based on detecting the character baseline. The baseline estimation is implemented in the feature extraction module. A sliding window is used to extract the character features.

4.3 Proposed Character Segmentation Approach

The proposed character segmentation algorithm can handle the following processes: histogram generation, histogram gradient sign normalization; and character segmentation. The flowchart for character segmentation of the proposed approach is shown in Figure 4.1

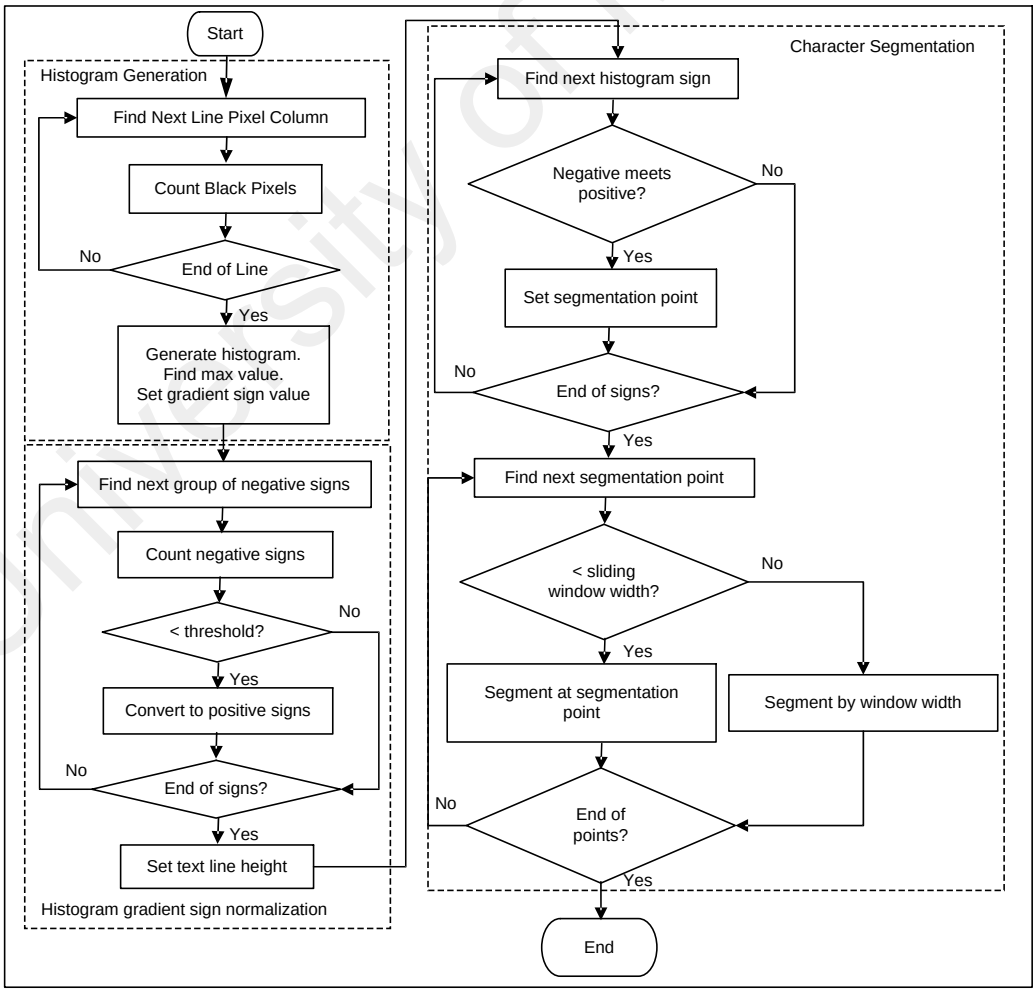

Figure 4.1 Character segmentation flow chart

4.3.1 Histogram generation

A tangent value is used to find the separation point (SP) accurately and without any data loss (Razak, Zulkiflee, Salleh, Yaacob, & Tamil, 2007). The tangent in the graph is calculated to find alternative representations (i.e. tangent representation with value either 1 or -1). False local minima in the graph can be detected and eliminated by collecting the number of 0s (black pixels) in the corresponding rows. These values will be stored temporarily in one array.

After the completion of the line segmentation process, the text image is divided into a number of separate text lines. A histogram representing the number of black pixels in each column is then generated for every text line. The maximum value in the histogram is used as the threshold value. Positive and negative signs are assigned accordingly to the histogram gradient values.

4.3.2 Histogram gradient sign normalization

Consecutive negative sign values in the histogram gradient graph for each group of consecutive negative sign values are counted. If the number of negative sign values is less than the threshold value, all negative sign values in the group are converted to positive sign values.

4.3.3 Character segmentation

The text line height is assumed to be the sliding window width. In the histogram, the pixel count, where the negative sign gradients meet the positive sign gradients, is set as the character segmentation point. If the length between neighbouring segmentation points is less than the sliding window width, then the particular segmentation points are used for segmentation. On the other hand, if the length between neighbouring segmentation points is more than the sliding window width, then the sliding window length is used for segmentation.

4.3.4 Pseudo code

The pseudo code of Algorithm 4.1 to implement the character segmentation process of the proposed character segmentation algorithm, handles histogram generation, histogram gradient sign normalization, and the character segmentation processes.

Algorithm 4.1 Character Segmentation

```
1.  For each pixel column in the text line
2.      Count the number of black pixels
3.  End For
4.  Generate histogram representing the number of black pixels in each column
5.  Find the maximum value in the histogram
6.  Set the threshold using the maximum value in the histogram
7.  Assign positive and negative signs to the histogram gradient values.
8.  For each group of consecutive negative sign values in the histogram
9.      Count the negative sign values
10.     If the number of negative values is less than the threshold
11.     Convert all the negative sign values to positive
12.     End If
13. End for
14. Set the text line height as the sliding window width
15. For each histogram value
16.     If negative sign gradients meets positive sign gradients
17.         Set pixel count as character segmentation point
18. For each character segmentation point
19.     Find next segmentation point
20.     If length between the segmentation points is less than the sliding window
        width
21.         Segment based on the segmentation points
22.     End If
23.     Else
24.         Segment based on the sliding window width
25.     End Else
26. End For
```

4.4 Justification for Proposed Approach

The proposed approach dispenses with the need for pre-processing such as edge detection, contour tracing, and use of Artificial Neural Network (ANN). This is because 100% accuracy is not crucial in our approach. This also means that less computational time is needed. For feature extraction, a sliding window is used to find the pixel distribution. The proposed approach focuses on analyzing rather than pre-processing and post-processing.

4.5 Evaluation Results

Compared to other techniques, our algorithm performs character segmentation faster and more accurately although there are a few cut-off characters. These cut-off characters can be eliminated through a local analysis of each line. In our experiments, we used Jawi manuscripts that contain a lot of noise. The character segmentation results obtained when using the proposed approach are shown in Figure 4.2.

Inaccuracies in segmentation are caused by overlapping of characters which is common in Jawi writing. Character overlapping will affect the black pixel distribution histogram and cause segmentation errors. A comparison of the evaluation results of the proposed approach with other approaches is shown in APPENDIX A., Table A.2, APPENDIX B, Table B.1 and Table B.3.

The performance of the proposed approach was compared with other approaches such as the algorithms proposed by Syiam et al. (2006) and (Romeo-Pakker, Miled, et al. (1995)) for segmenting Arabic handwriting. Evaluation involves comparison of the accuracy rate of segmentation achieved by each method — expressed as percentage of correctly segmented characters in the whole text. Segmentation error rate reflects the percentage of incorrectly segmented characters. In the evaluation, a scanned copy of "Hikayat Hang Tuah" manuscript from the 17th century was used.

The results of the evaluation show that the proposed approach achieved an accuracy rate of 98%, indicating that it outperforms the approaches of Syiam et al. (2006) and (Romeo-Pakker, Miled, et al. (1995)), which achieved 90% and 93.5% accuracy rates, respectively.

It is important to note that this 98% accuracy rate was achieved for segmentation done in the identified ROI. Only the overlapping characters that are common in Jawi

manuscripts prevents the proposed approach from achieving full (100%) segmentation accuracy

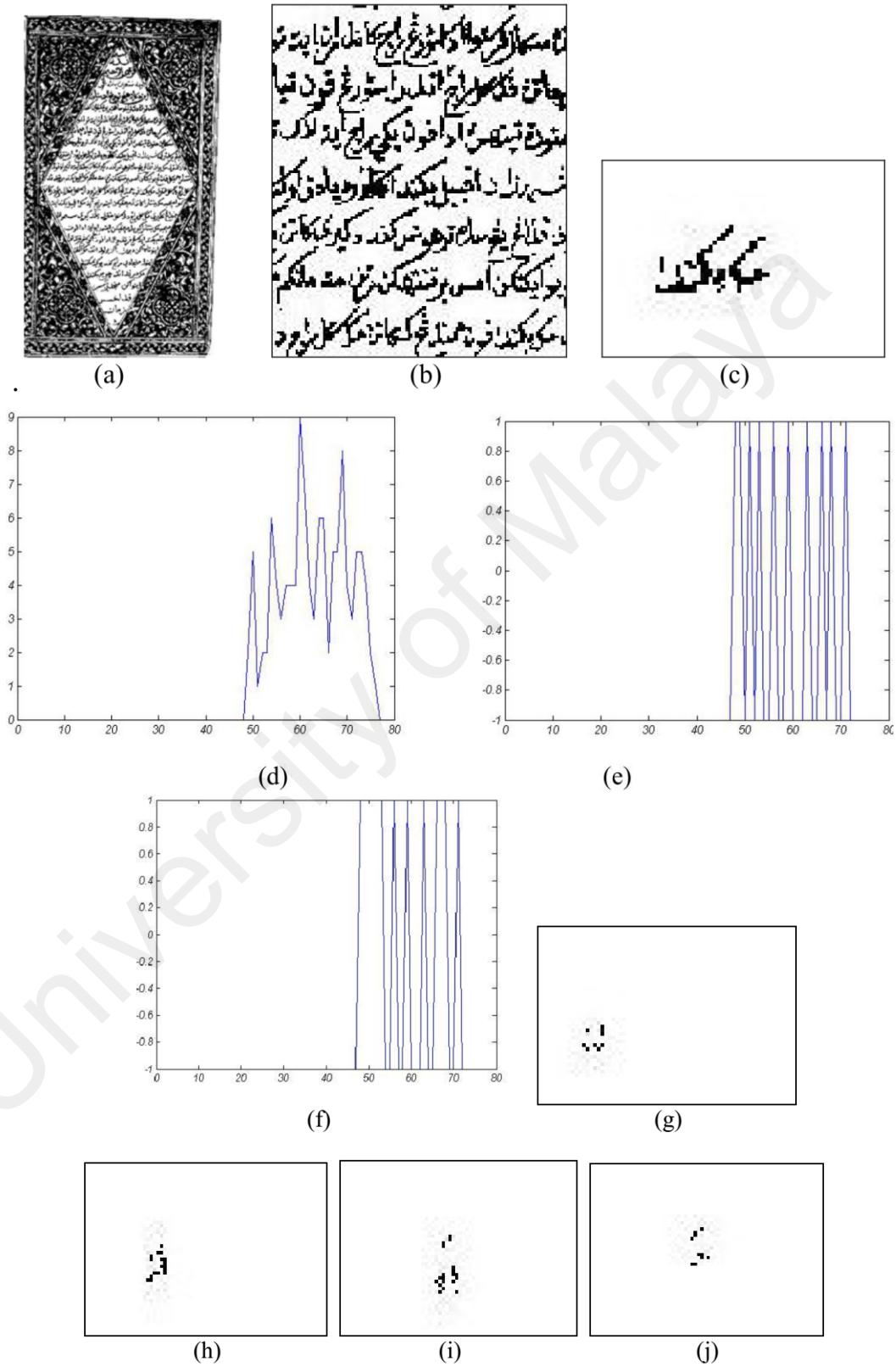

Figure 4.2 Character segmentation results. (a) Original Jawi manuscript image, (b) Binary image of the Region of Interest (ROI) (c) Segmented word, (d) Black pixel histogram, (e) Histogram gradient graph, (f) Normalized histogram gradient sign, (g) – (j) sample segmented characters

4.5.1 Time Complexity Analysis

To assess the computational complexity of the proposed character segmentation algorithm a time complexity analysis based on the Big O notation were conducted. The details and result of the analysis are shown in Table 4.1.

Table 4.1 Time Complexity Analysis for Character Segmentation Algorithm

Character Segmentation **(n: number of pixel columns)**	**Number of Iterations**
For *each pixel column in the text line* *Count the number of black pixels* **End For**	***n***
Generate histogram representing the number of black pixels in each column	
Find the maximum value in the histogram	***n***
Set the threshold using the maximum value in the histogram	
Assign positive and negative signs to the histogram gradient values.	***n***

For *each group of consecutive negative sign values in the histogram*
Count the negative sign values
If *the number of negative values is less than the threshold*
Convert all the negative sign values to positive
End If
End for
	n

Set the text line height as the sliding window width	
For *each histogram value* **If** *negative sign gradients meets positive sign gradients* *Set pixel count as character segmentation point*	***n***

For *each character segmentation point*
Find next segmentation point
If *length between the segmentation points is less than the sliding window width*
Segment based on the segmentation points
End If
Else
Segment based on the sliding window width
End Else
End For
	n

TIME COMPLEXITY CALCULATION	
O(n+n+n+n+n+n)=O(n)	

4.6 Conclusion

This chapter presents a novel approach for character segmentation in old Jawi manuscripts. The proposed approach uses vertical histogram projection, together with histogram gradient sign normalization to determine the maximum line height. A sliding window with length equal to the line height was then used to detect the segmentation point. Experiments were conducted to evaluate the performance of the proposed approach against two other methods for character segmentation. The experimental results show that the proposed approach outperforms the other two methods in terms of accuracy rate in segmentation. Only the presence of overlapping characters in the sample manuscript (i.e. Hikayat Hang Tuah) prevents the proposed approach from achieving full accuracy. Further research must be undertaken to find ways of eliminating the inaccuracies. In addition, efforts must also be made to analyze the horizontal histogram for each character segment as well as the entities.

CHAPTER 5

CHARACTER RECOGNITION

Character recognition is a process of mapping the already segmented characters into their equivalent digital form. This chapter presents an evaluation on the proposed characters recognition approach, which incorporates the good features of other techniques. The remainder of this chapter is organized as follows: Section 5.1 presents a brief introduction on the characteristics of Jawi characters, and discusses the challenges they pose in the effort to develop an efficient and flexible system for automatic Jawi character recognition. Section 5.2 presents an overview of existing methods for line and character segmentation that are relevant for developing a character recognition method. Section 5.3 discusses to the feature extraction process, particularly the unique code extraction process. Section 5.4 describes the classification method used. Section 5.5 presents the experimental results obtained for character recognition using the proposed approach. Section 5.6 concludes this chapter.

5.1 Introduction

Character recognition is a process for identifying non-digital characters from printed scripts, and mapping the collection of character shapes into the digital format. Jawi script consists of 37 characters that can be divided into two categories: grouped form, and individual form. Every Jawi character also has three different shapes based on their position in a connected Jawi word - at the beginning, middle, or end of the word. In Jawi character recognition research, there has been more emphasis on the writing styles,

because some of the Jawi characters have similar shapes at the middle and at the end of the word.

The wide usage of Jawi scripts in inscriptions, Islamic religious texts, petitions, newspapers and magazine, has provided the motivation to undertake this research. It is aimed at developing an efficient, cheap, flexible and user-friendly system that uses the computer for processing, storing and retrieval of Jawi manuscripts. This will greatly facilitate the use of Jawi in the office for various administrative purposes, and by the general public. The digitized Jawi scripts can also assist historians in studying old Jawi manuscripts, and the digitized content is easily and securely preserved for future use. Without the aid of information technology and full-fledged digitized Jawi system, the contents of these old and yet invaluable manuscripts are vulnerable to serious or permanent damage or lost for use by future generation. The proposed approach uses Hamming distance classifiers for off-line handwritten Jawi character recognition.

5.2 Line and Character Segmentation

The segmentation process simplifies or changes the representation of an image so that it can be analyzed easily. It is usually used to locate objects and boundaries such as the lines and curves in images. Line segmentation is important for analyzing the arrangement and separating the upper and lower lines, while character segmentation is concerned with the task of separating the word into its component characters. These two processes should be done accurately to prevent any segmentation errors from being propagated to other processes such as feature extraction.

5.2.1 Line segmentation

In their research, Razak et al. (2007) studied old manuscripts which have many overlapping characters over the lines. Overlapping characters present a serious problem in the first segmentation process — the line segmentation process. In this research, no

artificial intelligence approach is used in analysis and no morphological tracing is used unlike in the study by Ymin and Aoki (1996), and no calculation of pixel averages is carried out unlike in the studies by Cheung et al. (1998) and Adnan (1991). In this study, histogram projection is used regardless of the character orientation and the line skew. The false local minimum SP is omitted during normalization of the histogram. With such strategies, the algorithm can perform faster, achieve higher rate of accuracy, and maintain the quality of the segmented text lines. The results obtained using the algorithm in the line segmentation process are shown in Figure 5.1 to Figure 5.5.

Louloudis et al. (2006) presented a text line detection technique for off-line unconstrained handwriting based on a three-step strategy. The first step involves image enhancing pre-processing, connected components extraction, and average character height estimation. In the second step, a block-based Hough transform is applied for potential text lines detection, while the third step involves correcting possible false detections. The performance of the proposed technique was evaluated by comparing the text line detection result with the corresponding ground truth annotation.

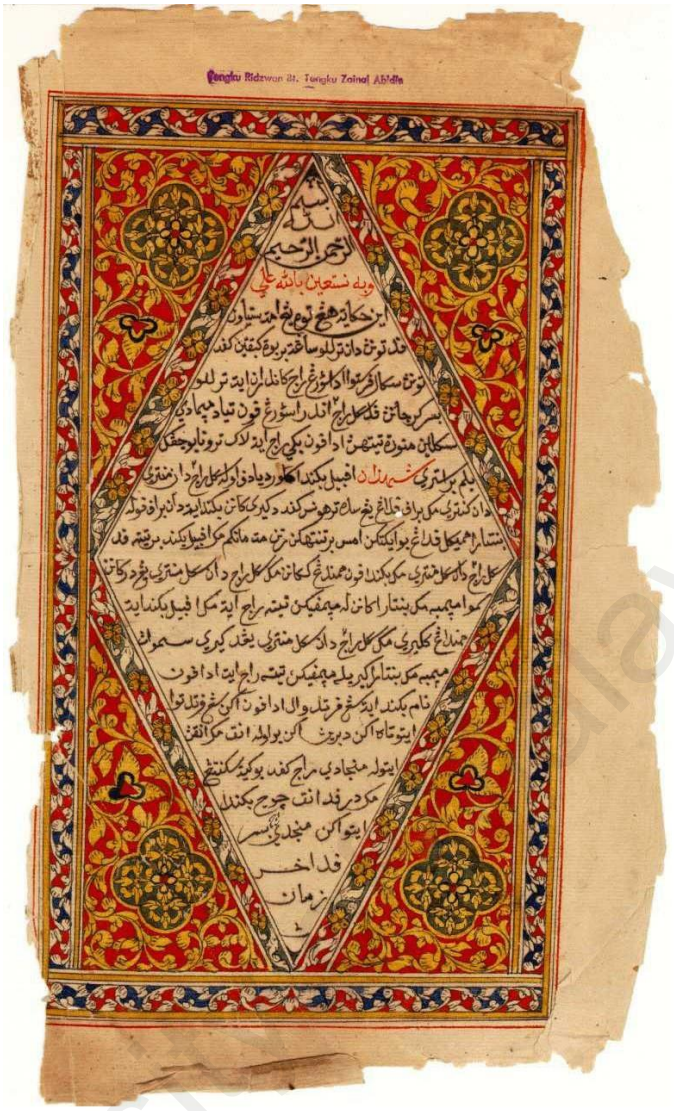

Figure 5.1 Old Jawi manuscript with size 12777*774

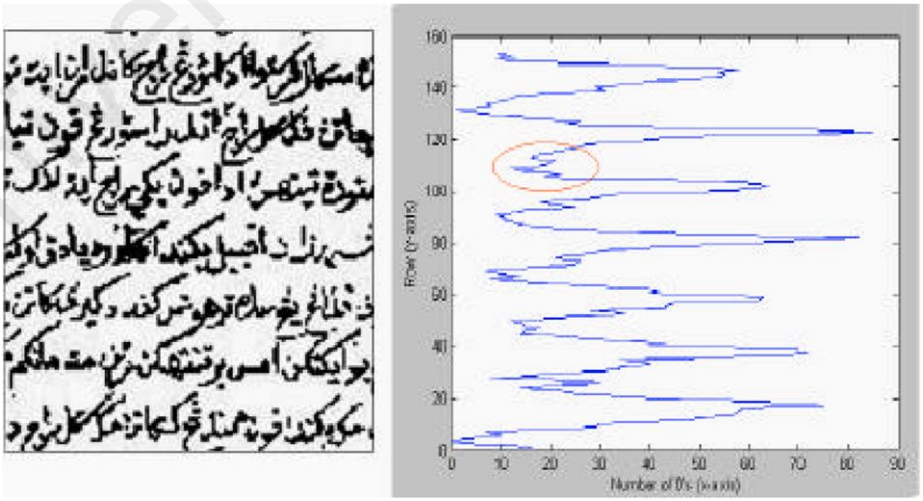

Figure 5.2 (a) Binary image of ROI of old Jawi manuscript Figure 5.2 (b) Graph row versus number of 0's

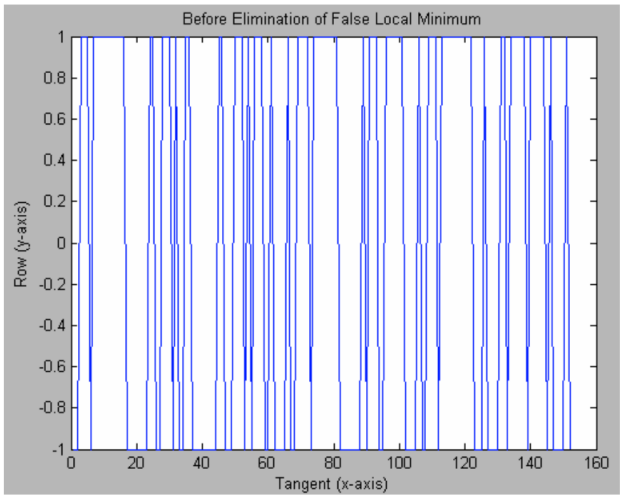

Figure 5.3 New representation of tangent versus row (before elimination of false local minimum)

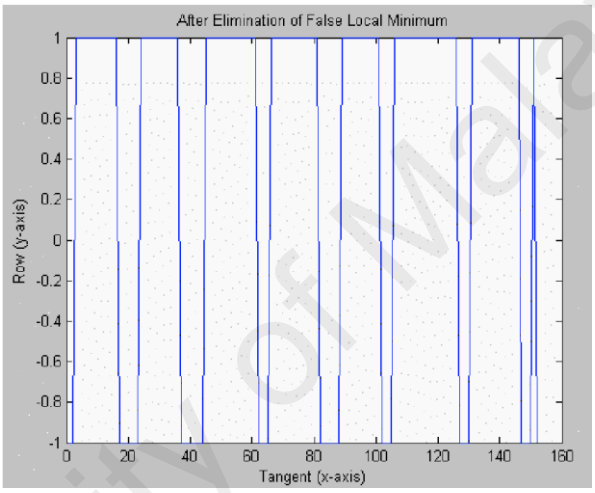

Figure 5.4 Tangent versus row (after elimination of false local minimum)

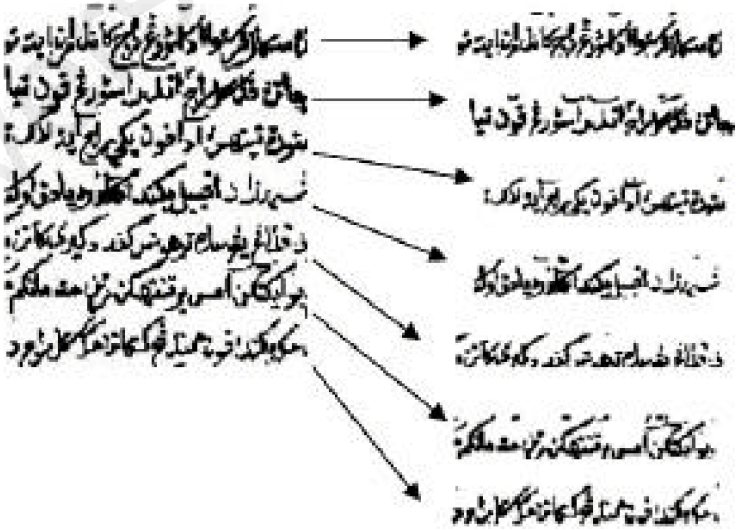

Figure 5.5 Result of line segmentation

El-Hajj et al. (2005) used their proposed method for detecting text lines of handwritten documents. These text lines include lines oriented in various directions, erasures, and annotations between the main lines. The method is based on a hypothesis-validation

strategy which is iteratively activated until segmentation is completed. At each stage of the process, the best text-line hypothesis is generated in the Hough domain, while it also takes into consideration the fluctuations of the text-line components. The validity of the line is then checked in the image domain using proximity criteria which analyze the context that is perceived as the alignment hypothesis. Any ambiguous components which might belong to several text lines are also marked.

Character segmentation: character segmentation of handwritten Jawi text is a much more challenging task because of the cursive writing and various writing styles.

In this study, histogram normalization and sliding windows are also used for the character segmentation process. This method has been used in many character recognition systems to segment the words and characters horizontally or vertically (Casey & Lecolinet, 1996). The vertical projection histogram will segment the document vertically. It can also detect the space between each character and specify the location of the vertical strokes in printed documents or in any line of handwriting.

Gouda and Rashwan (2004) segmented a word into many basic characters based on the baseline using vertical histogram. Al-Yousefi and Udpa (1992) segmented the character by using horizontal and vertical projection histograms, into primary and secondary parts. Syiam et al. (2006) implemented a clustering technique (k-means algorithm) on the vertical histogram. This approach improves the performance of histogram technique for recognition of handwritten characters. The character is clustered to identify similarities among the characters. This algorithm has a simple design and does not require much computational resources to run.

Figure 5.6 is a sample result of character segmentation. Pre-processing such as edge detection or contour tracing is not performed since it is not crucial for our approach to achieve full (100%) accuracy.

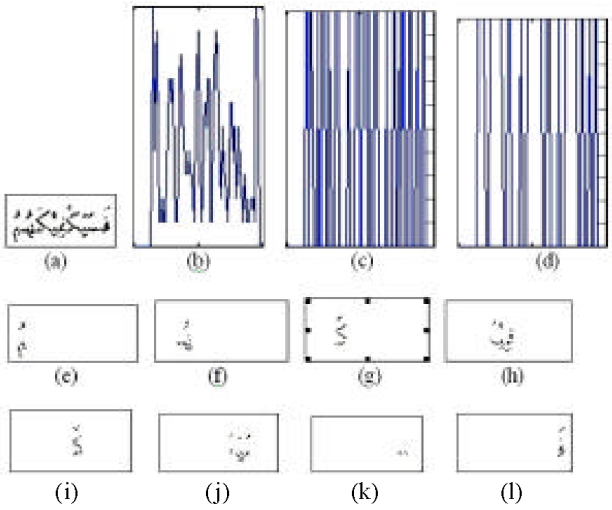

Figure 5.6 Character segmentation results

The text line height is assumed to be the sliding window width. In the histogram, the pixel counts where the negative sign gradients meet the positive sign gradients are set as the character segmentation points (Figure 5.6). If the length between neighbouring segmentation points is less than the sliding window width then the particular segmentation point is used for segmentation. On the other hand, if the length between neighbouring segmentation points is more than the sliding window width, then the sliding window length is used for segmentation.

5.3 Features Extraction

The Discrete Wavelet Transform (DWT) process uses the pyramid algorithm developed by Mallat (1989) to decompose various efficiency resolutions. This decomposition process operates on a signal — in this research it is the value of a pixel — and it can also be applied to image processing.

Mowlaei, Faez, and Haghighat (2002) developed a system for recognition of handwritten Farsi/Arabic characters and numerals. They used DWT to produce the wavelet coefficients, which are used for classification. Haar wavelet is used for feature extraction of Farsi/Arabic handwritten postal addresses which contain the names and postal codes of cities taken from a database of 579 postal addresses in Iran.

Discrete Wavelet Transform (DWT) is chosen for this research because it can provide the information to describe the position of pixels and also the density of pixels in character representation which has been segmented (Razak, Salleh, & Yaacob, 2005). It can also synthesize pixels to DWT coefficient, quickly. By using the Mallat algorithm (Mallat, 1999), DWT can decompose sub-band signal (value of pixel) which can be divided into components of high frequency but low resolution. The location of this frequency and the structure of various resolutions are some of the main traits of DWT, making it suitable for image compression, and getting the best representation for the sowing of character pixels that will be processed.

5.3.1 Unique code extraction process

The unique code is obtained by scanning the rows and columns of DWT coefficients which represent a Jawi character. This scanning is done by using Equation. 5.1:

$$R_{b,l} = \begin{cases} 1 & k > t \\ 0 & k < t \end{cases} \quad 5.1$$

where, R is the unique code, b is row, l is column, k is DWT coefficient, and t is the threshold value. If the threshold value is exceeded, then the value is set to 1, otherwise, it will be set to 0. This would produce 22 strings of value 1 or 0 for both rows and columns. Then, both 22-bit strings will be combined to produce a 44-bit string value which is known as the unique code of the character. The Hamming distance will then be calculated to get the reference value for the letter *Alif*. *Alif* is used as a reference by other characters.

There are two issues to consider in determining the value of the threshold - duplicate and class. Duplicate means that the unique code which has been produced by using the threshold value is similar to the unique code of other characters. Class means that every character will be divided into their classes according to its shape. For example, character *ba* (ب), *ta* (ت) and *tha* (ث) are in the same class since they have similar shape

and can only be differentiated by the position of the dots. Therefore, if the threshold value causes duplication of the unique code, and they are not from the same class, then the threshold value will be ignored. This process will continue until one threshold value does not cause any duplication. If there is any duplication, the character should be in the same class (Figure 5.7). The list of unique codes for each character after the threshold value is shown in Table 5.1 to Table 5.4.

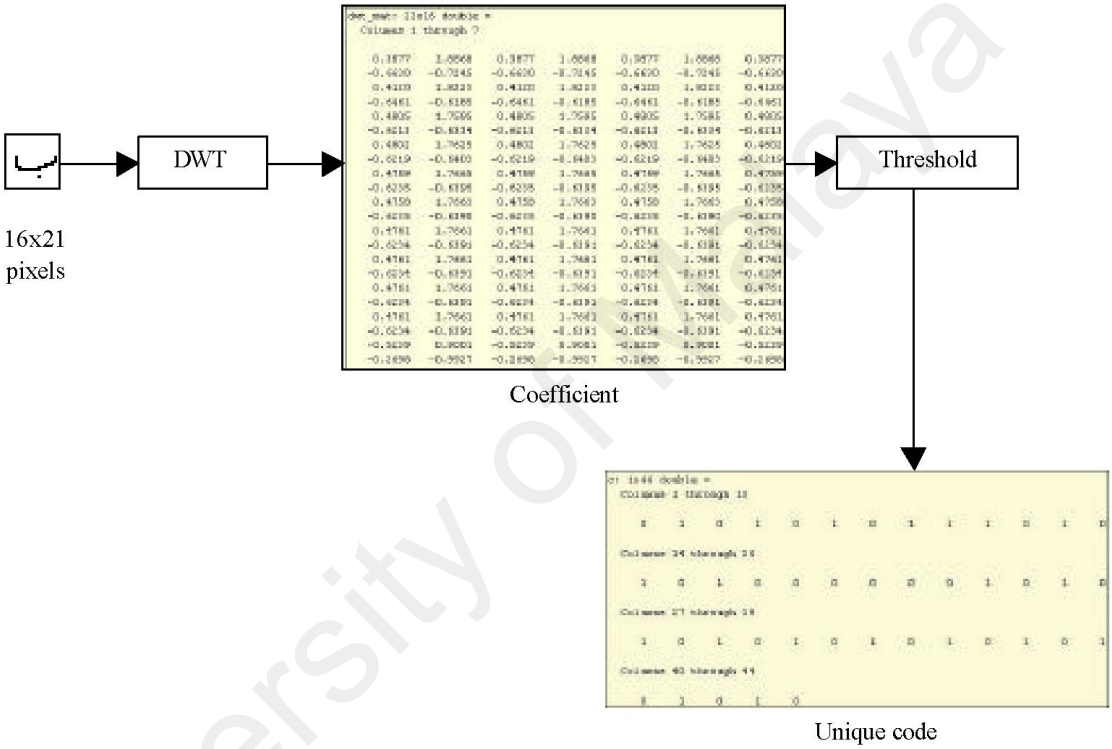

Figure 5.7 Summary of feature extraction and threshold process

Table 5.1 List of unique codes for isolated characters

Characters	**Unique code**
Alif	01010101110101010000001010101010101010101010
Ba	01110101111111110000001010101010101110101010
Ta	01010111110111110000001010101010101010101010
Tha	11111111111111110000001010101010101010111010
Jim	11111111111111010000001010101010101010101010
Ha	01111111111111110000001010101010101010111010
Kha	01111111111111110000001010101010101010101010
Dal	01111111101110100000001010101010101010101010
Dzal	01010111011111010000001010101010101010101010
Ra	11111101111101010000001010101010101010101010
Zai	01011101111111010000001010101010101010101010
Sin	11110111110111110000001010101010101010101010
Shin	01111101111111110000001010101010101010101010
Sod	11111101111111110000001010101010111010101010
Dhod	11011111111111010000001010101010101011101010
Tho	01111111011111010000001010101010101011101010
Dzo	01110111011111010000001010101010101011101110
Ain	01011111011111010000001010101010101010101110
Ghain	11110101111101110000001010101010101010101010
Fa	01111111111111110000001010101010101011101010
Kaf	01111111010111010000001010101110101010101010
Qaf	01010101110111010000001010101110101010101010
Lam	01010101010111010000001010101010101010101110
Mini	01010111110101010000001010101010101010101010
Nun	01010101110111110000001010101010101010101110
Wau	01010111111101010000001010101010101010101010
He	01011111111111010000001010101010101010101010
Ya	01111111111111010000001010101010101010111010
Cha	11111111111111110000001010101010101010101010
Ga	11011101111111010000001010101010101010101010
Nga	01111111110101010000001010101010101010101010
Nya	01010111111111110000001010101010101010101110
Pa	01011111011111010000001010101010101010101010
Va	01010111111111010000001010101010101010101010

Table 5.2 List of unique codes for characters at the beginning of a word

Character	Unique code
Alif	0101010111010101000000101010101010101010101010
Ba	0111111111111101000000101010101010101010101010
Ta	1111111111110101000000101010101010101010101010
Tha	0101010111110101000000101010101010101010101010
Jim	1101111111111111000000101010101010101010101010
Ha	1101111111010101000000101010101010101010101010
Kha	0111011111110101000000101010101110101010101010
Dal	None
Dzal	None
Ra	None
Zai	None
Sill	0111111111111101000000101010101010111010101010
Shin	0111111111110111000000101010101010101010101010
Sod	0111111111111111000000101010101010111010101010
Dhod	0101111101111111000000101010101011101010101010
Tho	0111011111011111000000101010101010101010101010
Dzo	0111010111011111000000101010101010101010101010
Ain	0111011111110101000000101010101010101010101010
Chain	1111111111111101000000101010111010101010101010
Fa	0101010111111101000000101010101010101010101010
Kaf	0101111111111111000000101010101010101010101010
Qaf	1111111111111101000000101010101010101011101010
Lam	0111111101110101000000101010101010101010101010
Mini	1111011101011111000000101010101010101010101010
Nun	1101011111111101000000101010101010101010101010
Wau	None
He	1101111111111101000000101010101010101010101010
Ya	1101011111111111000000101010101010101010101010
Cha	1111111111111101000000101010101010101010101010
Ga	0101111111111101000000101010101010101010101010
Nga	0111111111111101000000101010101010101010101010
Nya	None
Pa	0101111101010101000000101010111010101010101010
Va	None

Table 5.3 List of unique codes for characters at the middle of a word

Character	Unique code
Alif	None
Ba	11111111110101010000001010101010101010101010
Ta	01010111010101110000001010101010101110101010
Tha	01110111010101110000001010101010101110101010
Jim	11111111111111010000001010101010101010101010
Ha	11111111110111110000001010101010111110101010
Kha	11111111111111110000001010101010111010101010
Dal	None
Dzal	None
Ra	None
Zai	None
Sin	01111111110101010000001010101010101011101010
Shin	01111111110101010000001010101010101011101010
Sod	11010101111111110000001010101010111010101010
Dhod	11110111111101110000001010101010101010101010
Tho	01111111011111110000001010101010101010101010
Dzo	01011101110101010000001010101010101010101010
Ain	01011111011111110000001010101010101110101010
Ghain	11111111110101010000001010101011111010101010
Fa	11111111111111010000001010101010101010101010
Kaf	11010101111111110000001010101010101010101010
Qaf	01011111110101110000001010101010101010101010
Lam	01010101111111110000001010101010101010101110
Mini	11111101111111110000001010101011101010101010
Nun	11111111010101110000001010101010101110101010
Wan	None
He	01010111110101010000001010101010101010101010
Ya	11111111111111110000001010101010101011111010
Cha	11110111111111010000001010101010101010101010
Ga	11111111110111110000001010101010101010111010
Nga	11111111011010100000010101010101010101010101
Nya	11111101111111110000001010101010101010101010
Pa	01010111111111110000001010101010101110101010
Va	None

Table 5.4 List of unique codes for characters at the end of a word

Character	Unique code
Alif	01010101111111110000001010101010101010101010
Ba	11111111011111110000001010101010111010101010
Ta	11111111110111110000001010101010101111101010
Tha	11011111011111110000001010101010101010101010
Jim	11011111110111010000001010101010101010101010
Ha	11111111111111110000001010101010101010101010
Kha	11111111111111010000001011101010101010101010
Dal	11111111111111010000001010101110101011101010
Dzal	01010111111101110000001010101010101010101010
Ra	01011111010111110000001010101010101010101010
Zai	01010111111111110000001010101010101010101010
Sin	11010111111101110000001010101010101010101010
Shin	01010101111111110000001010101010111010101010
Sod	11011101010101110000001010101010101010101010
Dhod	11110111111101110000001010101010101010101010
Tho	01111111011101010000001010101010101010101010
Dzo	01111101011111010000001010101010101010101010
Ain	01111111110101110000001010101010101010101010
Ghain	01111111111101010000001010101010101010101010
Fa	11111111111101110000001010101010101010111010
Kaf	01011101010111010000001010101010101010101010
Qaf	01110101111111110000001010101010101010101010
Lam	01010101010111010000001010101010101010101010
Mini	11011101111111110000001010101010101010101010
Nun	01110111011111010000001010101010101010101010
Wau	01010111111101010000001010101010101010101110
He	01010111111101010000001010101010101010101010
Ya	None
Cha	11111111110111010000001010101010101010101010
Ga	01011111011101010000001010101010101010101110
Nga	11010111010101110000001010101010101010101010
Nya	01110111111111110000001010101010101010101010
Pa	11111101110101110000001010101010101010101010
Va	01011111011101010000001010101010101010101010

5.4 Classification

The Hamming distance gives a measure of the number of bits that are different between two bit patterns. Using the Hamming distance of two bit patterns, we can deduce whether the two patterns are generated from different Jawi characters or from one character. For binary strings *a* and *b*, the Hamming distance is equivalent to the number of 1s in $a\ xor\ b$.

In comparing the bit patterns *X* and *Y*, the Hamming distance, *HD*, is defined as the sum of disagreeing bits (sum of the exclusive-OR between *X* and *Y*) over *N*, the total number of bits in the bit pattern, as shown in the Equation 5.2:

$$HD = \frac{1}{N}\sum_{j=1}^{N} X_j \,(XOR) Y_i \qquad 5.2$$

The most preferred distance measure for binary features is the Hamming distance. Two approaches can be adopted to further improve the performance: (i) Weights can be applied to features (Cha, Yoon, & Tappert, 2005) and optimized using techniques such as genetic algorithms (Guoxing, Bingxue, & Wei, 1998), and (ii) Use a similarity measure that gives full credit to features present in both patterns, less credit to those features not present in either pattern, and no credit to those features present in only one of the patterns to be matched (Guoxing et al., 1998). Both approaches have been reported to perform better than the simple Hamming distance approach. Cha et al. (2005) suggested a new measure that combines these two approaches, and the experimental results confirmed that it performs better than other measures.

A compact smart current mode Hamming neural network has been developed for classifying complex patterns such as totally unconstrained handwritten digits. It is based on multi-threshold template matching, multi-stage matching, and k-WTA (k-Winner-Takes-All), some of which are different from the general Hamming neural network. The neural classifier consists of two templates — a binary template, and a multi-value programmable template — each having its own threshold and realized in MOS current mirrors, and the current mode k-WTA, which is reconfigurable. The second stage matching templates are programmable from outside the chip. This mixed analog-digital Hamming neural classifier can be fabricated using standard digital CMOS technology.

Pouliquen, Andreou, and Strohbehn (1997) used the basic building blocks to design an associative processor for bit-pattern classification; and a high-density memory-based neuromorphic processor. When operating in parallel, the single chip system can determine the closest match, based on the Hamming distance between an input bit pattern and multiple stored bit templates. Ties are broken, arbitrarily.

The Hamming distance algorithm for matching Jawi characters can be applied for:

- each pair of identity matrices *A* and *B*; and
- each matrix position.

If the matrix value in identity matrix *A* does not match the matrix value in identity matrix *B*, increase the distance between identity matrices *A* and *B* by 1.

5.5 Experimental Result

Table 5.5 shows the results of Hamming distance for all isolated Jawi characters. It is observed that the character *mim* (م) has the lowest percentage of 2.2727%, and the *HD* value is 1.

Table 5.6 shows the results of Hamming distance and the different percentages for characters at the beginning of a word. The lowest percentage is for character *tha* (ثـ) with Hamming distance equal to 1.

Table 5.7 shows the results for Jawi characters at the middle of a word. In the table, the lowest percentage is *dzo* (ظـ) and *he*(ھـ), with both having the same percentage of 2.27%.

Table 5.8 shows the results for Jawi character at the end of a word. Three characters — *alif* (ا), *dzal* (ذ), and *kaf* (ک) — have the lowest percentage of 6.82%, all having Hamming distance value of 3.

Table 5.5 Different percentages of isolated Jawi characters compared with character Alif

Image	Unicode	Isolated Char	HD	Percentage
ا	627	*alif*	0	0.0000
ب	628	*ba*	5	11.3636
ت	062A	*ta*	3	6.8182
ث	062B	*tha*	8	18.1818
ج	062C	*jim*	6	13.6364
ح	062D	*ha*	7	15.9091
خ	062E	*kha*	6	13.6364
د	062F	*dal*	4	9.0909
ذ	630	*dzal*	4	9.0909
ر	631	*ra*	4	9.0909
ز	632	*zai*	3	6.8182
س	633	*sin*	5	11.3636
ش	634	*shin*	5	11.3636
ص	635	*sod*	7	15.9091
ض	636	*dhod*	5	11.3636
ط	637	*tho*	7	15.9091
ظ	629	*dzo*	7	15.9091
ع	630	*ain*	7	15.9091
غ	062A	*ghain*	4	9.0909
ف	062B	*fa*	7	15.9091
ک	062C	*kaf*	5	11.3636
ق	062D	*qaf*	2	4.5455
ل	062E	*lam*	3	6.8182
م	062F	*mim*	1	2.2727
ن	638	*nun*	3	6.8182
و	639	*wau*	2	4.5455
ه	640	*he*	4	9.0909
ي	629	*ya*	6	13.6364
چ	630	*cha*	7	15.9091
ڬ	062A	*ga*	4	9.0909
ڠ	062B	*nga*	3	6.8182
ڽ	062C	*nya*	5	11.3636
ڤ	062D	*pa*	5	11.3636
ڤ	062E	*va*	3	6.8182

Table 5.6 Different percentages of Jawi characters at the beginning of word

Image	Unicode	Initial Char	HD	Percentage
ا	627	*alif*	0	0.00000
بـ	628	*ba*	6	13.6364
تـ	062A	*ta*	5	11.3636
ثـ	062B	*tha*	1	2.2727
جـ	062C	*jim*	6	13.6364
حـ	062D	*ha*	3	6.8182
خـ	062E	*kha*	4	9.0909
د	062F	*dal*	None	None
ذ	630	*dzal*	None	None
ر	631	*ra*	None	None
ز	632	*zai*	None	None
سـ	633	*sin*	6	13.6364
شـ	634	*shin*	5	11.3636

Image	Unicode	Initial Char	HD	Percentage
صـ	635	*sod*	7	15.9091
ضـ	636	*dhod*	7	15.9091
طـ	637	*tho*	4	9.0909
ظـ	629	*dzo*	3	6.8182
عـ	630	*a in*	3	6.8182
غـ	062A	*ghain*	7	15.9091
فـ	062B	*fa*	2	4.5455
كـ	062C	*kaf*	5	11.3636
قـ	062D	*qaf*	7	15.9091
لـ	062E	*lam*	5	11.3636
مـ	062F	*mim*	6	13.6364
نـ	638	*nun*	4	9.0909
و	639	*wau*	None	11.3636
هـ	640	*he*	5	11.3636
يـ	629	*ya*	7	15.9091
چـ	630	*cha*	6	13.6364
ݢـ	062A	*ga*	5	11.3636
ڠـ	062B	*nga*	5	11.3636
None	062C	*nya*	None	None
ڤـ	062D	*pa*	4	9.0909
None	062E	*va*	None	None

Table 5.7 Different percentages of Jawi characters at the middle of a word

Image	Unicode	Middle Char	HD	Percentage
None	627	*alif*	none	0
ـبـ	628	*ba*	4	9.0909
ـتـ	062A	*ta*	4	9.0909
ـثـ	062B	*tha*	5	11.3636
ـجـ	062C	*jim*	6	13.6364
ـحـ	062D	*ha*	8	18.1818
ـخـ	062E	*kha*	8	18.1818
None	062F	*dal*	None	None
None	630	*dzal*	None	None
None	631	*ra*	None	None
None	632	*zai*	None	None
ـسـ	633	*sin*	4	9.0909
ـشـ	634	*shin*	4	9.0909
ـصـ	635	*sod*	5	11.3636
ـضـ	636	*dhod*	5	11.3636
ـطـ	637	*tho*	7	15.9091
ـظـ	629	*dzo*	1	2.2727
ـعـ	630	*ain*	7	15.9091
ـغـ	062A	*ghain*	5	11.3636
None	062B	*fa*	5	11.3636
ـكـ	062C	*kaf*	4	9.0909
ـقـ	062D	*qaf*	3	6.8182
ـلـ	062E	*lam*	4	9.0909
ـمـ	062F	*mim*	7	15.9091
ـنـ	638	*nun*	7	15.9091
None	639	*wau*	None	None
ـهـ	640	*he*	1	2.2727
ـيـ	629	*ya*	9	20.4545
ـچـ	630	*cha*	5	11.3636
ـݢـ	062A	*ga*	7	15.9091
ـڠـ	062B	*nga*	7	15.9091
ـڽـ	062C	*nya*	6	13.6364
ـڤـ	062D	*pa*	5	11.3636
None	062E	*va*	None	None

Table 5.8 Different percentages of Jawi characters at the end of a word

Image	Unicode	End Char	HD	Percentage
ـا	627	*alif*	3	6.8182
ـب	628	*ba*	9	20.4545
ـت	062A	*ta*	8	18.1818
ـث	062B	*tha*	7	15.9091
ـج	062C	*jim*	4	9.0909
ـح	062D	*ha*	7	15.9091
ـخ	062E	*kha*	6	13.6364
ـد	062F	*dal*	8	18.1818
ـذ	630	*dzal*	3	6.8182
ـر	631	*ra*	5	11.3636
ـز	632	*zai*	4	9.0909
ـس	633	*sin*	4	9.0909
ـش	634	*shin*	4	9.0909
ـص	635	*sod*	4	9.0909
ـض	636	*dhod*	5	11.3636
ـط	637	*tho*	5	11.3636
ـظ	629	*dzo*	5	11.3636
ـع	630	*ain*	4	9.0909
ـغ	062A	*ghain*	4	9.0909
ـف	062B	*fa*	7	15.9091
ـک	062C	*kaf*	3	6.8182
ـق	062D	*qaf*	4	9.0909
ـل	062E	*lam*	2	4.5455
ـم	062F	*mim*	5	11.3636
ـن	638	*nun*	5	11.3636
ـو	639	*wau*	3	6.8182
ـه	640	*he*	4	9.0909
ـي	629	*ya*	None	None
ـچ	630	*cha*	5	11.3636
ـگ	062A	*ga*	5	11.3636
ـڠ	062B	*nga*	4	9.0909
ـڽ	062C	*nya*	5	11.3636
ـڤ	062D	*pa*	4	9.0909
ـۏ	062E	*va*	4	9.0909

Table 5.9 shows a sample of the result for classification using Hamming distance following the character segmentation process, as shown in Figures 5.8 (a)-(f).

Experiments were conducted to evaluate the performance of the proposed recognition method against other related techniques, from the aspect of recognition accuracy. The results of the evaluation are shown in APPENDIX A, Table A.3, APPENDIX B, Table B.1 and Table B.4.

Based on the results, the proposed approach for Jawi character recognition achieves 97% accuracy rate, which is much higher than that achieved by other techniques. However, the proposed approach did not achieve full accuracy because of the different widths of the characters.

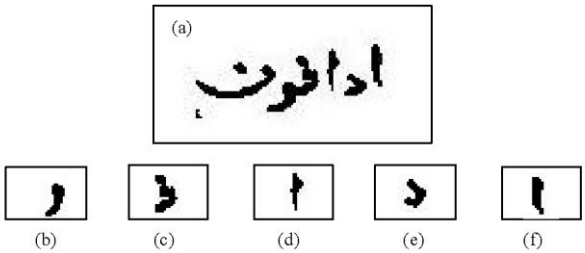

Figure 5.8 (a) Image before segmentation process; (b)-(f) image after character segmentation process

Table 5.9 Results of Hamming distance and percentage of errors for image in Figure 5.8

Character	Image	HD	Error (%)
Alif		1	2.2727
Dal		4	9.0909
Alif		0	0
Fa		1	2.2727
Wau		2	4.5455

5.6 Conclusion

This chapter presents the development of a system that uses histogram projection for line segmentation, histogram normalization for character segmentation, Discrete Wavelet Transform (DWT) for feature extraction, and Hamming distance algorithm for classification of Jawi characters. Different widths of the characters can affect the result, hence it is important to ensure that the characters are of the same width to overcome this problem.

CHAPTER 6

CONCLUSION

Old Jawi manuscripts have been an invaluable treasure of knowledge for sociologists, historians and academicians. However, a number of factors hinder access to and usage of these documents. The present research was initiated in an effort to use today's information technology tools and software to convert the old Jawi manuscripts into their equivalent digital copies, which would then greatly facilitate their usage. The digital formats of manuscripts can easily be manipulated by computers, and be used for transliteration. In the transliteration process, an old Jawi manuscript with its inherent old vocabularies are transformed to modern Jawi, which is more easily understood by scholars, researchers, and other interested Jawi learners and enthusiasts. The remainder of this chapter is organized as follows: Section 6.1 recapitulates the research objectives, and explains the tasks that were undertaken to fulfil those objectives. Section 6.2 states the contribution of this research. Section 0 highlights the limitations of this study vis-à-vis its scope. Section 6.4 suggests future works that could be undertaken in the domain of this thesis.

6.1 Revisiting the Research Objectives

This section recapitulates the objectives of this research as stated in Section 1.4, and discusses the tasks that had been undertaken to achieve each objective.

- **Objective 1:** To investigate the characteristics of Jawi manuscripts and identify their differences with Arabic manuscripts:

An extensive study was conducted to know more about the various characteristics and unique features of Jawi manuscripts. In particular, we made a detailed comparison between the Jawi script and Arabic script. From the study, we found some major similarities as well as differences, and this information facilitated the establishment of a framework for the development of a Jawi recognition method that is closely based on existing Arabic characters recognition techniques.

- **Objective 2:** To review existing character recognition algorithms for Jawi characters and determine the most suitable algorithms to be developed in this research.

We carried out an analytical review of different existing methods for Arabic character recognition. The review provides a clearer picture of the taxonomy of Arabic character recognition and the different stages involved such as: line, word and character segmentation, and character and word recognition. This taxonomy also classifies the current methods for Arabic word recognition into two broad approaches – the analytical approach which concentrates on individual character recognition, and global approach which is mainly concerned with recognition of an actual written word, as a whole.

- **Objective 3**: To propose algorithms for line, and character segmentations, as well as Jawi character recognition.

To achieve this objective, we carried out the following tasks:

a) An effective line segmentation algorithm was developed to handle handwritten Jawi texts. A major contribution of this approach is its ability to successfully eliminate the false local optima, found in horizontal projection histogram graph, that are caused by overlapping between adjacent lines.

b) A novel character segmentation method that uses a sliding window technique was developed to segment the Jawi characters, accurately.

c) A Jawi characters recognition system was successfully developed. This system uses Discrete Wavelet Transform (DWT) for feature extraction and Hamming distance concept for classification.

- **Objective 4:** To evaluate the effectiveness and efficiency of the proposed approach for handling and processing old Jawi manuscripts:

Experiments were conducted to thoroughly evaluate the performance of the proposed line segmentation, character segmentation and character recognition algorithms. In the evaluation, the performance of the proposed approaches was compared to the performance of other similar methods within the domain. The results of the evaluation show that the proposed approaches outperform the other methods.

6.2 Contributions of the Research

The line segmentation, character segmentation and character recognition algorithms developed in this research, have contributed positively to their specific domain. In summary, the novelty of the proposed approach in this research can be listed as:

- Detection and elimination of false local minima to improve the accuracy of line segmentation process.
- Robustness of the segmentation since the algorithm does not involve contour tracing
- The proposed approach address the time complexity issue.

The contribution of each algorithm is discussed in the sections below.

6.2.1 Line segmentation

Overlapped characters between two adjacent lines in Jawi scripts can cause inaccurate line segmentation. To overcome this problem, we developed a line segmentation algorithm that is based on the calculation of tangent from the histogram. This approach was chosen to eliminate the false local minima in the graph. Based on the evaluation results presented in Table A.1, the proposed approach achieved an accuracy rate of 97% in the segmentation of 1,000 lines. The unique feature of the proposed approach that is absent in other related techniques, is its ability to avoid the contour and boundary tracings. As a result, the algorithm performs better because of the less computational time needed. The other notable strength of the proposed approach, as compared to other techniques, is its computational simplicity.

6.2.2 Character segmentation

The character segmentation algorithm for Jawi manuscript developed in this research exploits the sliding window technique and the tangent normalization technique to segment the characters accurately. In this algorithm, the text line height is assumed to be the sliding window width. A vertical histogram is generated for each line. In the histogram, the trial segmentation point is the location where the negative sign gradient meets the positive sign gradient. If the distance between two neighbouring segmentation points is less than the width of the sliding window, then that particular segmentation point would be the end of the character, otherwise, the end of sliding window determines the end of the current character. The results of the evaluation, as presented in Table A.2, clearly show that the proposed character segmentation algorithm outperforms similar methods proposed by Syiam et al. (2006) and (Romeo-Pakker, Miled, et al. (1995)) used for comparison. The algorithm achieves 98% segmentation accuracy when applied to the scanned manuscript of “Hikayat Hang Tuah”.

6.2.3 Character recognition

The character recognition algorithm for Jawi manuscript developed in this research used the Discrete Wavelet Transform (DWT) method for feature extraction and unique code extraction. The Jawi characters were classified based on visual observation to find similarities in their shapes. For each character in a group, a corresponding unique code is generated by choosing 44 coefficients from the DWT matrix. A threshold-based technique was then used to convert a unique code string into the binary form. When the unique binary codes have been generated for all the characters (including isolated characters, characters at the beginning, middle or end of a word), the Hamming distance - a common classification technique - is used to determine the difference between any two patterns. The results of the evaluation, presented in Table A.3, show that the proposed character recognition algorithm outperforms other similar existing character recognition methods proposed by other researchers (Abuhaiba, Mahmoud, & Green, 1994; Al-Yousefi & Udpa, 1992; El-Hajj et al., 2005; Elgammal & Ismail, 2001; Fakir, Hassani, & Sodeyama, 2000; National Library of Malaysia, 2002; Sarfraz, Nawaz, & Al-Khuraidly, 2003). The algorithm achieved 97% accuracy rate in recognition of characters in old Jawi manuscripts. The results also show that character width can adversely affect the accuracy of the proposed character recognition algorithm.

6.3 Scope and Limitations

This research was primarily aimed at studying, designing and implementing a recognition system for old handwritten Jawi manuscripts. Although Jawi and its writing style are analogous to Arabic in many aspects, there are still some important and unique characteristics about Jawi which make it distinguishable from Arabic or other similar Arabic-like languages. These additional characteristics pose serious challenges which have to be properly addressed in the development of a Jawi character recognition system. One of the main problems encountered in the writing of old Jawi manuscripts is

the overlapping of characters that is commonly observed in these manuscripts. The extensive character overlapping inevitably leads to poor rate of accuracy in recognizing the characters in old Jawi manuscripts. Moreover, the different widths of Jawi characters also reduce the level of accuracy in character recognition. With a better understanding of the potential problems to be encountered, this study can be regarded as a preliminary step towards hardware implementation of a Jawi recognition system. However, hardware implementation has its limitation due to the high computational complexity demanded of recognition algorithm, thus making it impractical to be implemented. On the other hand, improving the performance of the recognition algorithm might compromise the rate of recognition accuracy. Therefore, a proper tradeoff is to balance between the need for efficiency against the need for accuracy.

6.4 Future work

This research had focused on converting old Jawi manuscripts into digital copies, to make them more suitable for present-day use. Future research in this domain can be extended to explore a number of other perspectives of Jawi, such as:

- Implementing a hardware-based Jawi recognition system;
- Developing an efficient character recognition technique that can handle different character widths, efficiently;
- Designing a comprehensive manuscript preservation systems; and
- Developing a tool for manuscript authentication.

REFERENCES

Abandah, G. A., Jamour, F. T., & Qaralleh, E. A. (2014). Recognizing handwritten Arabic words using grapheme segmentation and recurrent neural networks. *International Journal on Document Analysis and Recognition (IJDAR), 17*(3), 275-291. doi: 10.1007/s10032-014-0218-7

Abdul Hamid, F., & Abdullah, N. (2009). Penguasaan tulisan jawi di kalangan mahasiswa pengajian Islam : kajian di Institut Pengajian Tinggi Awam (IPTA) tempatan. *Journal of Al-Tamaddun, 4*, 145-156.

Abuhaiba, I. S. I., Datta, S., & Holt, M. J. J. (1995, 14-16 Aug 1995). *Line extraction and stroke ordering of text pages.* Paper presented at the Third International Conference on Document Analysis and Recognition, Montreal, Quebec.

Abuhaiba, I. S. I., Mahmoud, S. A., & Green, R. J. (1994). Recognition of handwritten cursive Arabic characters. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 16*(6), 664-672.

Adnan, A. (1991). Recognition of Arabic handprinted mathematical formulae. *Arabian Journal for Science and Engineering, 16*, 531-542.

Al-Badr, B., & Mahmoud, S. A. (1995). Survey and bibliography of Arabic optical text recognition. *Signal Processing, 41*(1), 49-77.

Al-Yousefi, H., & Udpa, S. S. (1992). Recognition of Arabic characters. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 14*(8), 853-857. doi: 10.1109/34.149585

Al Abodi, J., & Li, X. (2014). An effective approach to offline Arabic handwriting recognition. *Computers & Electrical Engineering, 40*(6), 1883-1901. doi: http://dx.doi.org/10.1016/j.compeleceng.2014.04.014

Alma'adeed, S. (2006, 16-18 Aug. 1993). *Recognition of off-line handwritten Arabic words using neural network.* Paper presented at the Geometric Modeling and Imaging--New Trends, London, England.

Amin, A., Kaced, A., Haton, J., & Mohr, R. (1980). *Handwritten Arabic character recognition by the IRAC system.* Paper presented at the Proceeding of 5th International Conference on Pattern Recognition, Florida, USA.

Arivazhagan, M., Srinivasan, H., & Srihari, S. (2007). *A statistical approach to line segmentation in handwritten documents*. Paper presented at the SPIE Document Recognition and Retrieval XIV, San Jose, CA, USA.

Azmi, M. S., Omar, K., Nasrudin, M. F., Idrus, B., & Wan Mohd Ghazali, K. (2013). *Digit recognition for Arabic/Jawi and Roman using features from triangle geometry*. Paper presented at the 20th National Symposium on Mathematical Sciences: Research in Mathematical Sciences: a Catalyst for Creativity and Innovation, Palm Garden Hotel, Putrajaya, Malaysia. http://scitation.aip.org/content/aip/proceeding/aipcp/10.1063/1.4801171

Benouareth, A., Ennaji, A., & Sellami, M. (2006). *HMMs with explicit state duration applied to handwritten Arabic word recognition*. Paper presented at the 18th International Conference on Pattern Recognition (ICPR 2006), Hong Kong.

Bushofa, B. M. F., & Spann, M. (1997, 2-4 Jul 1997). *Segmentation of Arabic characters using their contour information.* Paper presented at the 13th International Conference on Digital Signal Processing, DSP 97.

Casey, R. G., & Lecolinet, E. (1996). A survey of methods and strategies in character segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 18*(7), 690-706.

Cha, S.-H., Yoon, S., & Tappert, C. C. (2005). *On binary similarity measures for handwritten character recognition*. Paper presented at the Eighth International Conference on Document Analysis and Recognition, Seoul, Korea.

Che Wan Ahmad, C. W. S. B., Omar, K., Nasrudin, M. F., Murah, M. Z., & Bakar, J. A. (2013). *Rule Based For Old Jawi*. Paper presented at the Artificial Intelligence in Computer Science and ICT, Langkawi, Malaysia.

Cheung, A., Bennamoun, M., & Bergmann, N. (1998). *A recognition-based Arabic optical character recognition system*. Paper presented at the IEEE International Conference on Systems, Man, and Cybernetics San Diego, California, USA.

Correia, S. E. N., De Carvalho, J. M., & Sabourin, R. (2002). *Human-perception handwritten character recognition using wavelets*. Paper presented at the The XV Brazilian Symposium on Computer Graphics and Image Processing, Fortaleza-CE, Brazil.

Dahaman, I. (1991). *Pedoman ejaan jawi yang disempurnakan (1986)*. Paper presented at the Konvensyen Tulisan Jawi, Kuala Lumpur.

Dewan Bahasa dan Pustaka. (1967). *Malaysia. Akta Bahasa Kebangsaan 1963/67*. Retrieved from http://eseminar.dbp.gov.my/dokumen/akta_bahasa_kebangsaan_1963.pdf.

Doa, B. A. (2008). Ternate Language Retrieved 01 January 2015, 2015, from https://reyhan07.wordpress.com/2008/09/02/ternate-language/

El-Fishawy, A. S. (2004). *Analysis and design of programmable digital filters*. Paper presented at the STCEX Conference, Riyadh, KSA.

El-Hajj, R., Likforman-Sulem, L., & Mokbel, C. (2005). *Arabic handwriting recognition using baseline dependant features and hidden markov modeling*. Paper presented at the Eighth International Conference on Document Analysis and Recognition Seoul, Korea.

Elgammal, A. M., & Ismail, M. A. (2001). *A graph-based segmentation and feature extraction framework for Arabic text recognition*. Paper presented at the Sixth International Conference on Document Analysis and Recognition, Seattle, Washington, USA.

Fakir, M., Hassani, M., & Sodeyama, C. (2000). On the recognition of Arabic characters using Hough transform technique. *Malaysian Journal of Computer Science, 13*(2), 39-47.

Farooq, F., Govindaraju, V., & Perrone, M. (2005). *Pre-processing methods for handwritten Arabic documents*. Paper presented at the Eighth International Conference on Document Analysis and Recognition, Seoul, Korea.

Feldbach, M., & Tonnies, K. D. (2001). *Line detection and segmentation in historical church registers*. Paper presented at the Sixth International Conference on Document Analysis and Recognition, Seattle, Washington, USA.

Furht, B. (2011). *Handbook of augmented reality*: Springer Science & Business Media.

Goraine, H., Usher, M., & Al-Emami, S. (1992). Off-Line Arabic character recognition. *IEEE Computer Journal, 25*, 71-74.

Gouda, A. M., & Rashwan, M. (2004). *Segmentation of connected Arabic characters using hidden markov models.* Paper presented at the IEEE International Conference on Computational Intelligence for Measurement Systems and Application (CIMSA), Boston, MA, USA.

Guoxing, L., Bingxue, S., & Wei, L. (1998). *A modified current mode hamming neural network for totally unconstrained handwritten numeral recognition*. Paper presented

at the IEEE International Joint Conference on Neural Networks & IEEE World Congress on Computational Intelligence, Anchorage, Alaska, USA.

Haji Saidi, M. Z. (1996). *Hulu Terengganu menuju keemasan*: Kemaman Busara Teguh.

Hashim, M. (2005). Peranan tulisan Jawi dalam perkembangan Islam di Malaysia. *Jurnal Pengajian Melayu, 16*, 86-115.

Hj Yahaya, M. (2004). *Penulisan manuskrip Melayu-Islam: Tatakaedah dan kepentingannya dalam masyarakat Melayu Nusantara*. Paper presented at the Manuskrip Islam: Kepentingan dan Penyebaran, Brunei.

Iftekharuddin, K. M., & Parra, C. (2003). *Multiresolution-fractal feature extraction and tumor detection: analytical modeling and implementation.* Paper presented at the SPIE 5207, Wavelets: Applications in Signal and Image Processing X.

Ismail, S. M., & Abdullah, S. N. H. S. (2012). Online Arabic handwritten character recognition based on a rule based approach. *Journal of Computer Science, 8*(11), 1859-1868. doi: 10.3844/jcssp.2012.1859.1868

Jose, T. M., & Wahi, A. (2013). Recognition of Tamil handwritten characters using Daubechies Wavelet Transforms and Feed-forward Back Propagation Network. *International Journal of Computer Applications, 64*(8), 26-29.

Kandil, A. H., & El-Bialy, A. (2004). *Arabic OCR: a centerline independent segmentation technique*. Paper presented at the International Conference on Electrical, Electronic and Computer Engineering (ICEEC '04), Cairo, Egypt.

Kang, K. S. (2008). *Pengajaran dan pembelajaran Bahasa Jawi*. Paper presented at the Seminar Kebangsaan Tulisan Jawi, Universiti Pendidikan Sultan Idris (UPSI).

Kapogiannopoulos, G. S., & Papadakis, M. (1996). *Character recognition using a biorthogonal discrete wavelet transform.* Paper presented at the Wavelet Applications in Signal and Image Processing IV.

Laine, A., Schuler, S., & Girish, V. (1993). Orthonormal wavelet representations for recognizing complex annotations. *Machine Vision and Applications, 6*(2-3), 110-123. doi: 10.1007/BF01211935

Li, Y., Zheng, Y., Doermann, D., & Jaeger, S. (2006). *A new algorithm for detecting text line in handwritten documents*. Paper presented at the International Workshop on Frontiers in Handwriting Recognition, La Baule, France.

Likforman-Sulem, L., & Faure, C. (1994). *Extracting text lines in handwritten documents by perceptual grouping*. Paris: Europia.

Likforman-Sulem, L., Hanimyan, A., & Faure, C. (1995). *A Hough based algorithm for extracting text lines in handwritten documents*. Paper presented at the Third International Conference on Document Analysis and Recognition, Montreal, Quebec.

Lopresti, D., Nagy, G., Seth, S., & Zhang, X. (2008). *Multi-character field recognition for Arabic and Chinese handwriting*. Paper presented at the Arabic and Chinese handwriting recognition, College Park, MD, USA.

Louloudis, G., Gatos, B., Pratikakis, I., & Halatsis, K. (2006). *A block-based Hough transform mapping for text line detection in handwritten documents*. Paper presented at the Tenth International Workshop on Frontiers in Handwriting Recognition, La Baule, France.

Malaysia. Ministry of Education. (2004). *Perlaksanaan Program j-QAF Disekolah Surat Pekeliling Ikhtisas Bil 13*. Putrajaya: Ministry of Education Malaysia. Retrieved from http://www.moe.gov.my/v/pekeliling-ikhtisas-view?id=607.

Mallat, S. (1989). A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 11*(7), 674-693.

Mallat, S. (1999). *A wavelet tour of signal processing*: Academic Press.

Miled, H., Olivier, C., Cheriet, M., & Lecoutier, Y. (1997). *Coupling observation/letter for a Markovian modelisation applied to the recognition of Arabic handwriting*. Paper presented at the Fourth International Conference on Document Analysis and Recognition, Ulm, Germany.

Mohamad, R., Manaf, M., Rauf, R. H. A., & Nasruddin, M. F. (2015). Main Structure of Handwritten Jawi Sub-word Representation Using Numeric Code. In W. M. Berry, A. Hj. Mohamed & W. B. Yap (Eds.), *Soft Computing in Data Science* (pp. 208-217). Singapore: Springer Singapore.

Moon, T. K. (2005). *Error correction coding: mathematical methods and algorithms* (1st ed.): Wiley-Interscience.

Mowlaei, A., Faez, K., & Haghighat, A. T. (2002). *Feature extraction with wavelet transform for recognition of isolated handwritten Farsi/Arabic characters and numerals*. Paper presented at the 14th International Conference on Digital Signal Processing (DSP 2002), Santorini, Hellas, Greece.

Nasrudin, M. F., Omar, K., Liong, C.-Y., & Zakaria, M. S. (2010). Object signature features selection for handwritten Jawi recognition *Distributed Computing and Artificial Intelligence* (pp. 689-698). Berlin, Heidelberg: Springer Berlin Heidelberg.

Nasrudin, M. F., Petrou, M., & Kotoulas, L. (2010). *Jawi character recognition using the trace transform*. Paper presented at the Seventh International Conference on Computer Graphics, Imaging and Visualization (CGIV), Sydney, Australia.

National Library of Malaysia. (2002). *Warisan Manuskrip Melayu*. Kuala Lumpur: Perpustakaan Negara Malaysia.

Nazif, A. (1975). *A system for the recognition of the printed Arabic characters.* (Master's Thesis), Cairo University, Cairo, Egypt.

Nicolas, S., Paquet, T., & Heutte, L. (2004). *Text line segmentation in handwritten document using a production system*. Paper presented at the Ninth International Workshop on Frontiers in Handwriting Recognition, Tokyo, Japan.

Nik Yaacob, N. R. (2007). Penguasaan Jawi dan hubungannya dengan minat dan pencapaian pelajar dalam pendidikan Islam. *Jurnal Pendidik dan Pendidikan, 22*, 161-172.

Omar, S. M. S. (2001). Preservation of Malay manuscripts as a national documentary heritage: Issues and recommendations for regional cooperation *Sekitar Perputakaan* (33), 5-11.

Omidyeganeh, M., Nayebi, K., Azmi, R., & Javadtalab, A. (2005). *A new segmentation technique for multi font Farsi/Arabic texts*. Paper presented at the IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP '05), Philadelphia, Pennsylvania, USA.

Öztop, E., Mülayim, A. Y., Atalay, V., & Yarman-Vural, F. (1999). Repulsive attractive network for baseline extraction on document images. *Signal Processing, 75*(1), 1-10.

Pal, U., & Datta, S. (2003). *Segmentation of Bangla unconstrained handwritten text*. Paper presented at the Seventh International Conference on Document Analysis and Recognition, ICDAR2003, Edinburgh, Scotland.

Pechwitz, M., & Margner, V. (2002). *Baseline estimation for Arabic handwritten words*. Paper presented at the Eighth International Workshop on Frontiers in Handwriting Recognition, Ontario, Canada.

Pouliquen, P. O., Andreou, A. G., & Strohbehn, K. (1997). Winner-takes-all associative memory: A hamming distance vector quantizer. *Analog Integrated Circuits and Signal Processing, 13*(1-2), 211-222.

Primekumar, K. P., & Idiculla, S. M. (2011). *On-line Malayalam handwritten character recognition using wavelet transform and SFAM*. Paper presented at the 3rd International Conference on Electronics Computer Technology (ICECT), Kanyakumari, India.

Pudjiastuti, T. (2006). Looking at Palembang through its manuscripts. *Indonesia and the Malay World, 34*(100), 383-393.

Razak, Z., Salleh, R., & Yaacob, M. (2005). *Hardware design of on-line Jawi character recognition chip using discrete wavelet transform*. Paper presented at the Eighth International Conference on Document Analysis and Recognition Seoul, Korea.

Razak, Z., Zulkiflee, K., Salleh, R., Yaacob, M., & Tamil, E. M. (2007). A real-time line segmentation algorithm for an offline overlapped handwritten Jawi character recognition chip. *Malaysian Journal of Computer Science, 20*(2), 69-80.

Romeo-Pakker, K., Ameur, A., Olivier, C., & Lecourtier, Y. (1995). Structural analysis of Arabic handwriting: segmentation and recognition. *Machine Vision and Applications, 8*(4), 232-240.

Romeo-Pakker, K., Miled, H., & Lecourtier, Y. (1995). *A new approach for Latin/Arabic character segmentation*. Paper presented at the Third International Conference on Document Analysis and Recognition Montreal, Quebec.

Salim, J., Ismail, M., Suwarno, I., Abu-Ain, T., Abdullah, S. N. H. S., Bataineh, B., . . . Omar, K. (2013). Text normalization framework for handwritten cursive languages by detection and straightness the writing baseline. *Procedia Technology, 11*, 666-671. doi: http://dx.doi.org/10.1016/j.protcy.2013.12.243

Sarfraz, M., Nawaz, S. N., & Al-Khuraidly, A. (2003). *Offline Arabic text recognition system*. Paper presented at the International Conference on Geometric Modeling and Graphics, London, England.

Sasi, S. (1997). *Handwritten character recognition using fuzzy logic, neuro-fuzzy, and wavelet transform approaches.* Wayne State University.

Sesh Kumar, K. S., Namboodiri, A., & Jawahar, C. V. (2006). Learning segmentation of documents with complex scripts. In P. Kalra & S. Peleg (Eds.), *Computer Vision, Graphics and Image Processing* (Vol. 4338, pp. 749-760): Springer Berlin Heidelberg.

Shelke, S., & Apte, S. (2011). A multistage handwritten Marathi compound character recognition scheme using neural networks and wavelet features. *International Journal of Signal Processing, Image Processing and Pattern Recognition, 4*(1), 81-94.

Sim, D.-G., Kim, H.-K., & Oh, D.-I. (2000). *Translation, scale, and rotation invariant texture descriptor for texture-based image retrieval*. Paper presented at the International Conference on Image Processing, Vancouver, British Columbia, Canada.

Singh, P., & Budhiraja, S. (2012). Handwritten Gurmukhi Character Recognition Using Wavelet Transforms. *International Journal of Electronics, Communication & Instrumentation Engineering Research and Development (IJECIERD), 2*(3), 27-37.

Snoussi-Maddouri, S., Amiri, H., Belaïd, A., & Choisy, C. (2002). *Combination of local and global vision modelling for Arabic handwritten words recognition.* Paper presented at the Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR'02), Ontario, Canada.

Stockton, R., & Sukthankar, R. (2000). *JKanji: wavelet-based interactive kanji completion*. Paper presented at the 15th International Conference on Pattern Recognition, Barcelona.

Su, T.-H., Zhang, T.-W., Guan, D.-J., & Huang, H.-J. (2009). Off-line recognition of realistic Chinese handwriting using segmentation-free strategy. *Pattern Recognition, 42*(1), 167-182. doi: http://dx.doi.org/10.1016/j.patcog.2008.05.012

Syiam, M., Nazmy, T., Fahmy, A. E., Fathi, H., & Ali, K. (2006). *Histogram clustering and hybrid classifier for handwritten Arabic characters recognition.* Paper presented at the International Conference on Signal Processing, Pattern Recognition, and Applications (SPPRA), Innsbruck, Austria.

Timar, G., Karacs, K., & Rekeczky, C. (2002). *Analogic preprocessing and segmentation algorithms for off-line handwriting recognition*. Paper presented at the 7th IEEE International Workshop on Cellular Neural Networks and Their Applications, Frankfurt, Germany.

Toru, A. (2005). Jawi Study Group (Islamic Area Studies in Japan). *Annals of Japan Association for Middle East Studies (AJAMES), 2*(20), 399-404.

Tripathy, N., & Pal, U. (2004). *Handwriting segmentation of unconstrained Oriya text*. Paper presented at the Ninth International Workshop on Frontiers in Handwriting Recognition (IWFHR-9), Kokubunji, Tokyo, Japan.

Weliwitage, C., Harvey, A. L., & Jennings, A. B. (2005). *Handwritten document offline text line segmentation*. Paper presented at the Digital Image Computing: Techniques and Applications, Queensland, Australia.

Xin, Y., Lijuan, C., Mou, C., & Dake, Z. (2009). *Classification and recognition of character using WP decomposition, Zernike Moments and fuzzy integral*. Paper presented at the Sixth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD '09), Tianjin, China.

Yanikoglu, B., & Sandon, P. A. (1998). Segmentation of off-Line cursive handwriting using linear programming. *Pattern Recognition, 31*(12), 1825-1833. doi: http://dx.doi.org/10.1016/S0031-3203(98)00081-8

Ymin, A., & Aoki, Y. (1996). *On the segmentation of multi-font printed Uygur scripts*. Paper presented at the 13th International Conference on Pattern Recognition, Vienna, Austria.

Yu, T., Muthukkumarasamy, V., Verma, B., & Blumenstein, M. (2003). *A texture extraction technique using 2D-DFT and Hamming distance*. Paper presented at the Fifth International Conference on Computational Intelligence and Multimedia Applications, Xi'an, China.

Zabidi Haji Saidi, M. (1996). *Hulu Terengganu menuju keemasan*. Kemaman: Busara Teguh.

Zahour, A., Taconet, B., Mercy, P., & Ramdane, S. (2001). *Arabic hand-written text-line extraction*. Paper presented at the Sixth International Conference on Document Analysis and Recognition Seattle, Washington, USA.

Zeki, A. M. (2005). *The segmentation problem in Arabic character recognition The state of the art*. Paper presented at the First International Conference on Information and Communication Technologies, Karachi, Pakistan.

Zhang, P., Bui, T. D., & Suen, C. Y. (2005). *Hybrid feature extraction and feature selection for improving recognition accuracy of handwritten numerals*. Paper presented at the Eighth International Conference on Document Analysis and Recognition, Seoul, Korea.

LIST OF PUBLICATIONS

- Razak, Z., Zulkiflee, K., Salleh, R., Yaacob, M., & Tamil, E. M. (2007). A real-time line segmentation algorithm for an offline overlapped handwritten Jawi character recognition chip. *Malaysian Journal of Computer Science*, 20(2), 69-80.
- Razak, Z., Zulkiflee, K., Noor, N. M., Salleh, R., & Yaacob, M. (2009). Off-line handwritten Jawi character segmentation using histogram normalization and sliding window approach for hardware implementation. *Malaysian Journal of Computer Science*, 22(1), 34-43.
- Othman, Z., Abdullah, N., Razak, Z., & Mohd- Yusoff, M. Y. (2014). Speech to Text Engine for Jawi Language. *The International Arab Journal of Information* Technology, 11(5), 507-513.

APPENDIX A

DETAILED EVALUATION RESULTS

Table A.1 Experimental results for the line segmentation

Author	Experiment data	Experiments	Accuracy	Segmentation errors
(Zahour et al., 2001)	1000 lines in 100 samples	Implemented in C++ language on a 200MHz PC	97%	Baseline-skew variability, and overlaps between characters and diacritical marks in the first column. Association of diacritic symbols distant from the separating border line, to a text line.
(Tripathy & Pal, 2004)	1,627 lines in single column pages with different writing styles	Draw boundary lines between two consecutive text lines. Then, from the computer display, line segmentation accuracy was calculated manually.	60%	When two consecutive words touch, or distance between two consecutive words is very small.
(Arivazhagan et al., 2007)	11,581 lines in 720 documents of and children's handwriting in English or Arabic	The cut-through accuracy corresponds to the proportion of components correctly classified as overlapping components or cut- through error.	97.31%	A normal component, spanning across two or more lines or lying in between two lines of text
(Sesh Kumar et al., 2006)	Documents scanned from a Telugu book titled “Aadarsam", printed in 1973 containing 256 pages.	The performance of segmentation is calculated using a segmentation quality metric	Few or no corrections required on a new page after adapting to the first 44 pages. The remaining 212 pages were segmented correctly.	Not stated

(Weliwitage et al., 2005)	30 images from NIST database in 34 text boxes from 2100 forms	Text boxes in the form correspond to a text paragraph of 52 words were extracted for text line segmentation.	96%	Very short text lines not longer than one word and text lines not starting from left margin of the image and unclear separation of text lines with merging ascenders and descenders might be detected as an extra text line.
(Likforman-Sulem & Faure, 1994)	Unconstrained handwritten rough drafts, address blocks, letters and manuscripts.	Alignments found as text lines are crossed by a line, components belonging to the same line share the same identification number inscribed above their enclosing rectangles. Alignments found in the Hough domain, but invalidated in a second stage are crossed by a dashed line. Ambiguous components are inscribed in dashed rectangles.	Not stated	When fluctuation is combined with proximity of text lines, merging lines may appear.
(Timar et al., 2002)	10 pages of a handwriting database containing 7,000 words from the LOB (Lancaster-Oslo / Bergen) corpus written by a single writer	Conducted using the MatCNN simulator in Matlab software	Correctly segmented each line in every page	Not stated
(Li et al., 2006)	More than 10,000 diverse handwritten documents in Arabic, Hindi, and Chinese script	If a ground-truth line and the corresponding detected line share at least 90% of pixels, a text line is considered to be detected correctly.	From 2,691 ground-truth lines, correctly detected 2,303 (85.6%) lines. At the text line level, only 951 (35.3%) text lines were detected correctly.	Errors were caused by two adjacent text lines overlapping significantly, signatures, the correction in the gap between two lines, and the severe noise introduced during scanning.
(Louloudis et al., 2006)	Unconstrained handwritten Greek documents using 20 document images taken from the historical archives of the University of Athens	Used Block-based Hough Transform approach for which corresponding text line detection ground truth was manually created	450 text lines were detected in the images with 96.87% accuracy	Various accents above or under the text line body and the small difference in the skew angle in the text lines
(Nicolas et al., 2004)	Collection of drafts of handwritten French novelist Gustave Flaubert		Only well-separated text lines were correctly segmented.	Overlapped text lines and text lines where the interline distance is smaller than the intra line distance will cause errors.

(Feldbach & Tonnies, 2001)	Text in church registers from 61 paragraphs consisting of 300 lines in 7 pages spanning 300 years.	Chain code representation	222 lines (90%) of 246 lines in 49 paragraphs containing six different handwritings were reconstructed correctly.	Serious errors occur when the difference between the reconstructed baseline and centre lines and the real lines was higher than the script size, if a text line was not found, or if an extra line was found at a wrong place.

Table A.2 Evaluation results for character segmentation

Authors	Methods	Experiment Data	Accuracy	Segmentation Errors
Syiam et al. (2006)	K-mean clustering algorithm (clustering technique) and applied to vertical histogram	Arabic handwriting	90% accuracy in segmentation correctness. Achieved recognition accuracy rate of 91.5%	The grouping of two characters at the end of the sub-word. Over-segmentation of لا or س
(Romeo-Pakker, Miled, et al. (1995))	Divide the segmentation process into two methods (detection of characters junction and segment based on the upper contour)	Arabic/ Latin handwriting	The character junction detection method achieved 93.5% accuracy in segmentation. The upper contour segmentation method achieved 99.3% accuracy in segmentation	In the character junction detection method, 2% of the characters were not segmented. 4.5% of characters were segmented unnecessarily. In the upper contour segmentation method, 0.3% of the characters were not segmented. 0.4% of characters were segmented unnecessarily.

Table A.3 Evaluation result for Jawi character recognition

Authors	Method	Experiment data	Accuracy	Errors
Al-Yousefi and Udpa (1992)	Off-line statistical method for feature extraction and Bayesian Classifier for recognition system	Isolated, handwritten and printed Arabic text	Classification rate with 85.5% accuracy using linear discriminant analysis. By using quadratic discriminant, the classification rate is 99.5%	Classification rate for handwritten character is not as good as printed characters with mixed fonts and sizes
Abuhaiba et al. (1994)	Statistics from moments of horizontal and vertical projections for feature extraction. Clustering for classification	Arabic handwriting	Classification accuracy rate of between 73.6-100%. Proved to be flexible	The system is believed to be accurate enough. The only limitation is speed

	process			
Fakir et al. (2000)	Feature extraction using Hough Transform technique. Uses dynamic programming matching technique for classification	Applied to a set of 300 words in a handwritten Arabic text	Classification accuracy rate of 95%	Substitution errors are the most common error in this system, and occur during the thinning process
Sarfraz et al. (2003)	Using moment invariant technique for features extraction and Artificial Neural Network for classification	Printed Arabic text which uses Naskh font	Recognition accuracy rate of 73%	The extracted features deviate from the respective result in the training data, because of the differences in resolution. It affects the geometric moments of the image
Elgammal and Ismail (2001)	Strokes, loops and feature points. Template matching technique for classification	Printed Arabic documents	Classification accuracy rate of 95.2% for the first set, and 94.1% for the second set.	The main error is the result of characters touching each other in irregular positions due to bad printing and scanning
(El-Hajj et al., 2005)	Analytical approach by extracting pixels densities, density transitions and concavity configurations along frame with respect to baseline and using character HMM for the recognition process	Offline handwritten Arabic	The accuracy is quite satisfactory. The recognition rate ranges from 85.45% to 87.2%	The diacritical marks are often not in the exact position, and generation of a letter can be extended to be under one or more letters of the same words

APPENDIX B

EVALUATION OF RELATED ALGORITHMS AGAINT OJM IMAGES

Table B.1 Experimental image data specification

Characteristic	Details
Type	BMP
Size	Average per file : 20 Kbyte
Number of files	Good : 30 Moderate: 30 Worse: 30
Number of lines	1000
Number of words	More than 8000
Number of characters	More than 32000

Table B.2 Comparison of related line segmentation algorithms against OJM images

Authors	Accuracy (Average)	Comments
(Zahour et al., 2001)	80%	Accuracy rate is decrease on worse quality image but improve when tested on higher quality
(Tripathy & Pal, 2004)	50%	Overlapped line caused accuracy dropped
(Arivazhagan et al., 2007)	90%	Accuracy is quite high but due to massive overlapped lines and worst quality of documents, the accuracy dropped
Proposed Approach	96%	Inaccuracies occur due to severe overlapping between lines

Table B.3 Comparison of related character segmentation algorithms against OJM images

Authors	Accuracy	Comments
Syiam et al. (2006)	90%	Overlapped caused the accuracy drop and also by quality of images
(Romeo-Pakker, Miled, et al. (1995))	91%	Oversegmented on overlapped character and unnecessarily segmentations occur at the end of line.
Proposed Approach	98%	Massive overlapped character decrease accuracy

Table B.4 Comparison of related character recognition algorithms against OJM images

Authors	Accuracy	Comments
Elgammal and Ismail (2001)	50%	Performs better only on printed character but not handwritten character
(El-Hajj et al., 2005)	85%	Diacritical marks give problem and can't perform well in low quality images and poor segmentation
Sarfraz et al. (2003)	60%	Better accuracy if segmentation is solid and accurate with no foreign pixels
Proposed Approach	97%	Different sizes of character width give different unique codes