Implementation and Evaluation of Document Retrieval for the PC Notes Taker (PCNT) Handwriting Device

Master's Thesis

by

Nasir Mahmood

Matrikel-Nr.: 171129

Course: Computational Visualistics

Advanced Multimedia and Security Lab (AMSL)
Institute of Technical and Business Information Systems
Faculty of Computer Science
Otto - von - Guericke - University, Magdeburg

Supervisor 1: Prof. Dr.-Ing. Jana Dittmann
Supervisor 2: Dipl.-Inform. Sascha Schimke

To my wife

Acknowledgments

I deem utmost pleasure to thank my supervisor, Mr. Sascha Schimke, for his encouraging attitude and supervision that enabled me to get this manuscript perfected. I would also like to thank to Prof. Jana Dittmann for her guidance and suggestions during the course of my project.

I don't have words at my command to express my gratitude and admiration to my mother, father, sisters and brother who stood by me mentally and spiritually and always helped me in solving all of my troubles with their love to accomplish my goal.

I wish to thank two of my best friends from Magdeburg, Ayaz Farooq and Kamran Ali, for their time and companionship.

Further thanks go to the secretarial and examination office staff for their support on technical issues.

Nasir Mahmood

Implementation and Evaluation of Document Retrieval for the PC Notes Taker (PCNT) Handwriting Device

Nasir Mahmood, M.Sc.
Otto - von - Guericke - University, Magdeburg,

Supervisor: Prof. Dr.-Ing. Jana Dittmann

In spite of recent technological developments, handwriting is important in contemporary communication methods due to its claims of authenticity, (inter-) mediality and corporeality. Handwriting devices are available to write documents and such documents are easy to manage and search text among them. In this work, we present implementation and benchmarking of a document retrieval system for a handwriting device - PC Notes Taker (PCNT). PCNT device was found to be competitive to the ones we have already tested and benchmarked before with our document retrieval system. We also extended features of the document retrieval system by implementing a subtype of features and compared its results with those of existing features. We found former slightly poor but closer to later.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Handwriting

Handwriting has an important position in the contemporary communication society. It is used for many different practices such as in the form of literary writing, correspondence, advertisement etc. Recently, it has undergone electronic articulation in the form a typewriter or a computer rather than a humand hand. Inspite of these technological developments, handwriting has not lost its importance due to the claims of authenticity, (inter-)mediality and corporeality made by handwriting [20].

1.1.1 Authenticity

Handwriting has traditionally been considered as an autography that guarantee the presence of an individual writer and serves as an un-exchangeable, unique and authentic "signature". It conveys a physical token of identity as an authentic and recognizable expression alongwith information. This claim of authenticity discerns handwriting from its rival, *typed* writing where it is destroyed by the mechanization involved in *typed* writing. The significance of *typed* writing is due to its characteristics of iterability and reproducity and standardization which are perceived its advantages. Handwriting, on the other

hand, may be considered as forgery if one tries to reproduce it and this resistance to reproduction gives real power of authenticity to the handwriting [22].

1.1.2 (Inter-)mediality

In philosophical terms, handwriting is regarded as an almost invisible, immaterial medium to *immediately* depict thought. Immediality of handwriting has been controversial in the disciplines of philosophy and media. This criticism takes an extra and more sever dimension if writing is *typed* writing rather than *hand*writing. Handwriting has *inter*medial character of being unreadable and having a material mode as does both linguisitc writing and visual image.

1.1.3 Corporeality

Handwriting consists of a compound of *hand* and *writing* which asks for analysis of both status of *writing* and status of *hand* writing i.e. of the body. *Hand*writing is written by a hand and is authentic and unique whereas typed writing is associated with a typewriter or a computer which is inanimate machines. In different disciplines, role of body or hand in the process writing has been controversial. In grapho-psychology, the hand movement in a writing is taken as a trace of the character of the subject writing.

1.2 Digital Handwriting

Digital handwriting is information of a user 's handwriting entered using a pen or pointing device through interfaces such as a PDA touch screen, TabletPC touch screen, Smartphone touch screen, Digital Pen, Graphics Tablet, Interactive Whiteboard, and so on [21]. It is a way to convert the written words from the ink on paper to digits that can be stored on a personal computer [6]. Ocassionally, the term digital ink is also used to refer digital handwriting.

1.3 Digital Handwriting Devices

The range of digital handwriting devices available in the market can be put into following three main classes based on their features.

1.3.1 Small Handheld Computers

This pen based class of computers is gaining popularity due to their small size, convenience and effectiveness to enhance communication and documentation. Personal digital assistant (PDA), mobile phones with PDA features, Tablet PCs and newly arrived ultra mobile PCs (UMPC) fall under this category of digital handwriting devices. Use of pen to take handwritten notes directly on the screen, to check the boxes on screen forms or to draw screen diagrams extend in an oppurtunity to collect data accurately and easily by reducing huge amount of paperwork. Such devices are of utmost importance and value for mobile workers to help them save information conveniently while talking with customers, warehouse employees or for nurses examining patients. Pen enabled computers have a touch sensitive screen and input is accepted when a special (digital) pen is pressed against the screen. The screen is able to record drawings and handwriting or accept taps on special areas of it which represent keys or buttons.

Use of pen is more supportive and prefered over keyboard and mouse operation while taking brief notes or making quick drawings on a handheld computer during a meeting or lecture because pen movements are more natural than mousing or typing around. Pen computers come with their own operating system rather than tradition personal computer system [23].

1.3.2 Digital Whiteboards

A digital whiteboard (e.g Xerox Liveboard or mimio Xi) allows to record everthing that is drawn on it for posterity, or transmit it elsewhere [13]. It is

composed of a familiar interface - a board, a pen and an eraser. Every word, line and color written is stored on personal computer automatically. The attractive feature of whiteboard is the ease to use it by just picking up dry erase markers. One can make mistakes, erase words and correct it accordingly. Digital whiteboards are even able to remember the mistakes you made and erased over the course of evolution of your ideas.

Some boards come with a pressure-sensitive surface whereas others use marker pens embedded with a tracking device. When one writes on a board, sensors in or around the board pick up and track the position, movement and even color of the pen. This data is transferred and displayed on the computer. The automatic capturing of all the notes helps to focus on the ideas rather than note-taking [5].

1.3.3 Digital Pen-and-paper

A digital pen is a a battery-operated writing device which is used to register or capture series of the strokes of handwriting when user moves pen over the paper and transfers this information to an application which stores handwriting as a digital handwritten document [8]. A typical digital pen that looks like a regular ball-point pen comes with or without a Universal Serial Bus (USB) to let the user upload the handwritten notes to a personal computer. The components and structure of digital pen also differs with the choice of paper to write on, whether it is touch screen or a digital paper.

Digital paper is a patterned paper which is used with a digital pen to create handwritten digital documents. The printed dot pattern uniquely identifies the position coordinates on the paper. The digital pen uses this pattern to store the handwriting and upload it to a computer. Digital paper is also called an interactive paper [34].

Digital pen, that looks and works like an ordinary pen, captures the strokes with a tiny camera (or sensor) of handwriting and drawings from a normal

paper or a paper overprinted with a dot pattern. The camera may be fitted near the nib of the pen or inside a paper clip which is used to hold paper while writing on it. In case of former, the data is uploaded by docking it in a computer whereas in later data is transfered immediately to the computer through paper clip when you write or draw something on the paper [30]. IBM CrossPad [24], Logitech ioPen [14], and Pegasus PC Notes Taker [18] have pen-and-paper kind of features.

It provides a cost effective solution for tradition paperwork to link to digital world of computers. Such as a digital handwriting system provides an exact image of the handwriting and drawings which can be translated to text. Pen and paper is thought to be an emerging technology and its use will become more widespread as the cost of the pen decreases [1].

1.4 Handwriting Data Acquisition

What you write with adapted writing device i.e. digital pen, handwriting acquisition transforms it into a digital format which can later be processed by computers. In addition to transformation of handwritten text to digital one, handwriting acquisition strategy opens a a range of possibilities from searching for notes to trigerring actions by writing a symbol [33]. There are two main handwriting data acquisition approaches: online handwriting acquisition and offline handwriting acquisition. These approaches are described below in detail.

1.4.1 Offline handwriting Acquistion

Offline data acquisition or input method represents a visual representation of text rather than to any dynamic information about the order or how the character was written. This methodology is used in Optical Character Recognition (OCR) and Intelligent Character Recognition (ICR) applications to read digitally in a scanned or photographed image of printed or handwritten text as

Figure 1.1: Offline handwriting

Figure 1.2: Online Handwriting (image taken from visionobjects.com)

shown in figure 1.1 (taken from visionobjects).

Offline data acquisition is used in applications where information has already been acquired through forms and now it needs to be saved into a computer first by scaning the handwritten or printed forms and then reading data into computer. One of main distvantages is the noise which results from scanning or photographing the text. It introduces lines or patterns on the paper, extra marks from dust or scratches of a printing process which distracts the data recognition or acquisition system from the main images [33].

1.4.2 Online Handwriting Acquisition

In contrast to offline handwriting acquisition, the way a text is written is thought to be important. The ink signal is capture by one of the following techniques which comes with different digital handwriting systems discussed in section 1.3:

- a digital pen on a patterned paper
- a paper-based captuer device
- a pen-sensitive surface such as a touch screen

A digital ink signal composed of a sequence of 2-dimensional points with reference to time is used to mathematically represent the information of strokes and trajectories of a handwriting as shown in figure 1.2 (image taken from visionobjects). Digital ink is not always a 2-dimensional sequence with reference to time but there are devices which provide information about a kind pen pressure, at least binary pressure i.e. pen-down and pen-up, applied during writing process. A few devices measure angle of the pen while one is writing, others have velocity or acceleration sensors.

Online data acquisition is free of optical noise i.e. no dirty paper background, which one needs to remove from the image. Though online data acquisition only captures the information needed i.e. trajectory and strokes to make a clear signal, but the movement signals may have noise in the sense, that for example drawing a line has a small jitter. Even then it is comparatively easier to process data which comes from online acquisition rather than an offline acquisition and it offers broad range of possiblitities of its application.

1.5 Document Retrieval

Give a set D of documents $\{d_1, d_2, d_3, ..., d_n\}$ and a query word q, a document retrieval method finds a list $D\acute{}$ of documents $\{d_1^{'}, d_2^{'}, d_3^{'}, ..., d_n^{'}\}$ out of D where

retrieval query q has one or more occurrences in the documents of $D´$. In each document $D´$, positions of all the occurrences of a query q are given. In a pen based retrieval system, query q and all the elements of D are handwritten or hand-drawn acquired directly with the help of a special handware (See figure 4.1). Our investigation will involve PC Notes Taker (PCNT) device for acquisition of a handwriting or drawing. A user can perform document retrieval operation either by writing or drawing the query or by selecting an occurrence as a query within a document.

The term handwritten documents in this work will refer to the pen-movment data acquisition based approach i.e. online handwritting rather than to the scanned images of sheets of paper as it happens in off-line handwriting systems. The handwritten document data consist of sequences of sampled pen tip positions : x_t, y_t at time t. In this work, time information is not being interpreted for the determination of features used by the document retrieval method under investigation. But one may use it to determine other features like velocities in the direction of x and y axis or track velocity $(v_x(t), v_y(t), v(t))$.

Textual recognition is the most intuitive method that comes into mind to search occurrences of a textual query q with a simple string search function on textual feartures of documents D. There are two major disadvantages of using textual recognition for document retrieval: a) texual recognition often fails in most of the searches and does not work absolutely perfect, b) there could be situations where no text exists at all but hand-drawn images or sketches instead of words and in such cases textual recognition does not come up with any answer. To address these non-trivial problems, a kind of *direct handwriting matching* has already been presented and its description is give in chapter 3.

Thesis is organized as follows: Chapter 2 provides an overview of the related published work in the literature and basics of algorithms needed to understand our algorithm. Chapter 3 explains the extraction of geometric features to be used in searching algorithms. Futhermore a short overview of similar features

from the literature is given. Chapter 4 describes benchmarking of document retrieval system on PCNT device against both square and regular triangle driven Freeman features. Chapter 5 presents results and discussion and chapters 6 concludes this work.

Chapter 2

Literature Review

2.1 Related Work

Plenty of work has already been done in the field of pen-based document retrieval but with different approaches to achieve different goals.

Srihari et al. came up an image feature indexing approach to perform search operation on handwritten documents acquired off-line [31]. Since handwritten data was acquired off-line, the steps it involves to build image features are quite complicated compared to those used by our method. Govindaraju et al. presented a similar approach to enhance the search speed by mamking smaller sets of large lexicons and parallel processing [10].

Landy and David proposed a note-sharing system called NotePals [15]. All the meeting documents e.g. personal notes, minutes, or slides captured through PDAs and paper-based digitizer devices are stored in a central repository through PDA and paper user interface respectively. All the centrally stored information by different members is later avaible to all the group members for browsing and search interface.

In addition to text-based retrieval approaches, people have proposed image-based solutions for pen aided document retrieval in image databases. The image-based document retrieval system developed by Schomaker et al. makes use of pattern recognition and machine learning [29]. Upon a pen

drawn query, it extracts shapes from the images and makes a comparison of those shapes and query.

Fonseca et al. have proposed an indexing and retrieval method for vector graphic files [7]. User needs to present a hand-drawn sketch query to retrieve vector grahics files with similar image(s). The method uses graph matching approach to compare query with vector descriptors of the indexed drawings. Vector descriptors are generated from spectral information topology graphs of the drawings to avoid costly graph-isomorphism computations over large databases.

Jawahar and Balasubramanian presented a synthesis model to improve recognition and retrieval of handwritten data consisting of more complex characters of the Indian languages. The algorithms presented can learn from annotated data and improve their representation with feedback [16].

Another approach to search handwritten script captured online has been proposed by Sun et al. in [32]. It is very much similar to the method we have developed but with different features and matching algorithm.

Our approach is intended to reduce the complexity of the system by making the features simple and matching algorithm accurate and efficient.

2.2 String Algorithms and Document Retrieval

String comparison is an area of research where efforts have been made for a long time to develop faster algorithms to solve this problem. Since goal of our research was also to make handwritten document retrieval faster by reducing complexity of data to be compared by an algorithm, therefore string comparison was a unique idea to be used to get this comparison done over a finite alphabet i.e. strings. A similarity measure is needed to get a feeling that how close two strings are and it plays a very important role in the efficiency of the algorithms which are used to make comparison of data. In literature, a range of similarity measures is availble for different kinds of applications but there is no 'one' best to be used for all kinds of applications. In the following,

few of the similarity measures have been discussed briefly.

2.2.1 String Edit Distance

String edit distance is mostly widely used notion for string comparison[26]. The miniumum number of insertions, deletions and substitutions required to transform one string into other is called string edit distance [17]. It is usually preferred to use for character based techniques of string comparison. Edit operations of character insertion, deletion and substition are assigned a cost learned from data. Computational cost of string edit distance operations is quadratic using dynamic programming. It has also been used in its extended form to perform edit operations on a higher level of tokens such as synonyms, abbreviations etc.

Hamming Distance

Hamming distance, a variant of string edit distance, is another similarity measure used for string comparison. It is the number of positions for which the corresponding symbols of the strings are different. In other words, it counts the number of substitutions required to change one into ther other, or the number of errors that transformed one string into the other [35]. For example, the Hamming distance between

1011101 and 1001001 is 2,
2143896 and 2233796 is 3, and
"toned" and "roses" is 3.

Orginally, Hamming distance was proposed by Richard W. Hamming in his paper about *error-correcting codes* [12] and it is very frequently used in telecommunications to the number of flipped bits in a fixed-length binary word as an estimate of error - called **signal distance**.

Damerau-Levenshtein Distance

Damerau-Levenshtein distance, an extension of Levenshtein distance, is the minimal number of insertions, deletions, substitutions and transpositions needed to transform one string to the other [4]. Damerau presented the idea of Damerau-Lenvenshtein distance, with more emphasis on single-character misspellings, in [4] allowing four edit operations. It is worth noting that edit distance proposed by Lenvenshtein in [17] does allow multiple edition operation but no tranposition operation. Though introduction of an addition operation of transpositions sounds simple, in reality it is complicated to calculate Levenshtein edit distance.

Since Damerau-Levenshtein method is able to calculate a restricted edit distance, it plays an important role in natural language processing. In natural language processing, strings are normally short and the number of errors rarely exceed 2. In such cases where restricted and real edit distance difference is very low, the limitation of restricted edit distance does not matter too much.

Dynamic programming techniques with asymptotic complexity time of $O(mn)$, where m and n are lengths of the strings, offer a classical solution for obtaining the edit-distance involving minimal number of operations. The classical edit distance suits well to fix misspellings in word processing. Schimke et al. have successfully adapted it to use for comparison of handwritten signatures for biometric applications [28].

Local Similarity

Another similar approach called basic local alignment search tool (BLAST) was proposed by Altschul et al. in [2]. It uses local similarity measure, the maximal segment pair (MSP) score, by optimizing it to generate approximate alignments of two strings. The algorithm is simple and robust and is being used extensively in the field of bioinformatics for DNA and protein sequence database searches, motif searches, gene identification searches and in analysis

of multiple regions of DNA.

There is yet another variant which is called *approximate string searching*. It computes the edit distance between one string i.e query and all the substrings of another reference string. One can use this method to find all similar occurrences of a short string within a longer one [11]. The method proposed by Schimke et al in [27] is based on the idea of approximate string searching by extending it to develop a search system for on-line handwritten documents. The following section has been dedicated to explain how does real algorithm work and how it has been adapted to be used in relation to a handwritten document retrieval system.

2.3 Approximate String Search and Document Retrieval

Approximate string searching method performs a fuzzy search of a shorter query string q for all its appearances within a longer string d which in our case represents a whole handwritten document to be search in. According to equation 2.1, it can be realized by filling a matrix D of size $(m+1) \times (n+1)$ where m and n are lengths of q and d respectively.

$$D(i,j) = \left\{ \begin{array}{ll} 0 & if\ i = 0, \\ D(i-1,0)+1 & if\ i > 0\ and\ j = 0, \\ min \left\{ \begin{array}{l} D(i,j-1)+1 \\ D(i-1,j)+1 \\ D(i-1,j-1)+\delta(i,j) \end{array} \right\} & else, \end{array} \right\} \tag{2.1}$$

$$\delta(i,j) = \left\{ \begin{array}{ll} 0 & if\ q[i] = d[j], \\ 1 & else, \end{array} \right\} \tag{2.2}$$

In equation 2.1, the function $\delta(i, j)$ depicts cost of a substitution of i^{th} character of query string q by j^{th} character of document string d.

It is obvious from above description that the computational complexity of approximate string searching algorithm is $O(mn)$. The complexity can be further reduced by calculating the matrix D column-wise and by holding only two actual columns. It makes the matrix row $D(m, 0...n)$ contain edit distances between the query string q and a substring of document string d ending at position j of document string d. If a match of query string q exists a certain position j of document string d, the matrix element $D(m, j)$ has to be smaller that threshold $\tau : D(m, j) < \tau$. If τ number is taken smaller, the missing rate increases and the recall rate decreases. On the other hand, by raising it to a larger number, the mismatch rate goes high and ultimately the precision of search is declined. Theoretically, the value of $D(m, j)$ can not exceed m but practically its average is smaller in experiments with random strings q and d (uniformly distributed randomness) over a finite alphabet A. The longer the alphabet is, the greater the normalized averaged edit distance $(D(m, j)/m)$ is. In practice, the mismatch rate rises rapidly if the threshold τ for the maximal allowed edit distance is chosen greater than this averaged value for the perspective alphabet size [27].

Chapter 3

Method

3.1 Features for Document Retrieval

Approximate string search algorithm is considered as one of the potential methods for searching in handwritten documents. One needs to provide the algorithm with some sort of feature data, which represent ink traces of the process, of the handwritten document one is going to search a query in. The feature data should be in a string-like format. Since the features of a handwriting are represented in a string format, that is why they are called string features. One can easily extract string feature data from handwritten documents by taking into account the handwriting signals in terms of discrete x_t, y_t position of the tip of the pen and binary value of pressure p_t over time t.

Schimke et al. have already investigated four different types of string features: *(a)* Freeman grid codes, *(b)* direction based codes, *(c)* curvature based codes, and *(d)* slant based codes [27]. Freeman grid codes were found more robust kind of features and the aim of this work was to further investigate them. Therefore, we will not be discussing the rest of three features in the following sections.

3.1.1 Freeman Grid Codes

Originally the idea of represenation of features of a handwritten document in the form of Freeman grid codes was presented by H. Freeman in 1974. He used this method to encode the line drawings in [9]. It presents an encoding method to convert the x_t, y_t measurements of a handwriting into a form of data that is acceptable to a computer. An encoding is basically a description which involves quantization of a sequence of data i.e. x_t, y_t position over time t. The quantization involves a grid to discretize the sequence of data and to assign a so called Freeman code to each position.

In [27], Schimke et al. performed Freeman codes extaction using square grids and outlooked extraction of the features with triangular grid codes to see what could be the effects of equally spaced six possible sample point directions offered by a triangular grid in contrast to eight possible sample point directions of square grid. The four diagonally placed sample points of a square node are a little bit more far away from those which are placed horizontally and vertically (see figure 3.2 and figure 3.6). Therefore, we tried two different types of grids i.e. square and triangular grids for the quatization of the handwritten data. The range of codes which a position may be assigned depends upon the type of grid we are using. The Freeman grid codes which all x_t, y_t positions are assigned depending upon the grid node they were mapped to are then used by approximate string search algorithm for searching in that particular document. In the following sections, we describe in detail how the Freeman grid codes are generated by using both square girds and regular triangular grids.

Square Grid Freeman Codes

A grid usually refers to two or more infinite sets of evenly-spaced parallel lines at particular angles to each other in a plane, or the intersections of such lines [3]. Square grids, also known as orthogonal grids, consist of two sets of lines perpendicular to each other as shown in figure 3.1. The intersection points are

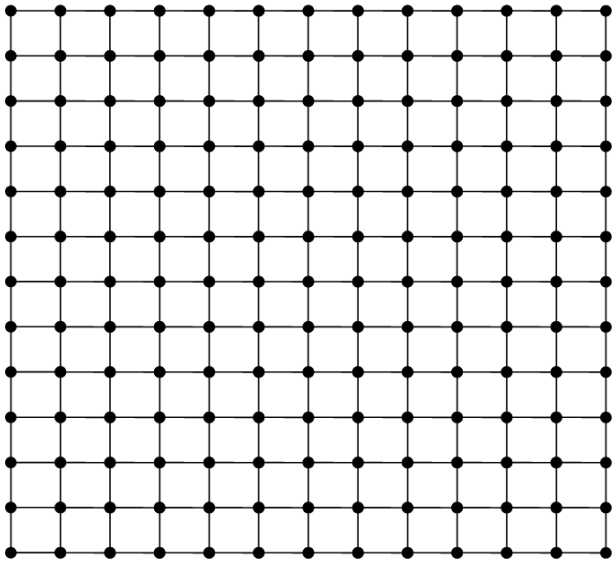


Figure 3.1: Square grid

called grid nodes and are used to map sample points x_t, y_t of a handwritten document. The handwritten input i.e. text or figure is superimposed on the square grid (see figure 3.7) and to each sampling point (dashed lines) the next grid node it will jump to is assigned. Assignment of next node to a sample point is done in terms of grid codes. Each node of square grid has got eight possible neighbours (see figure 3.2) and one of them could be the next node whose code would be assigned to the preceding node of the grid. Therefore one can code the original ink shape, as a sequence of symbols, by taking into account the eight possible neighbours of each grid node which stand for eight possible directions ink shape may happen to shape into. A nineth symbol can be used to encode a gap between segments. In figure 3.7 (left column) the dashed lines drawing of ink shape represents original one whereas the solid lines that of coded shape which computer understands well. The main difference between original ink shape and the coded ink shape is that former is a sequence of sample points x_t, y_t whereas later is a sequence of eight possible symbols which square grid offers.

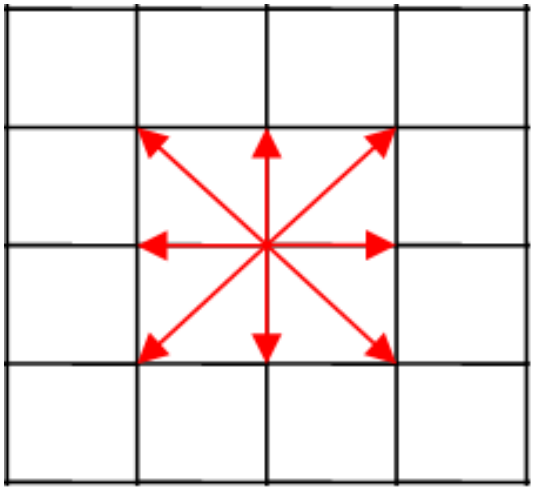


Figure 3.2: Square grid directions

Triangular Grid Freeman Codes

Triangular grid Freeman codes is very much similar to square grid Freeman codes generation scheme except type of grid. It uses a regular triangular grid rather than a sequare grid. A triangular grid, also known as isometric grid, consists of three sets of lines at 60-degree angle to each other [3] or in other words it is composed of a regular tessellation of equilateral triangles. (See figure 3.3).

When it comes to the implementation of triangluar grid for generation of Freeman grid codes, one may think to manipulate the implemenation of square grid so that it should look like a triangular grid. In fact, it is achieveable with two basic operations on a square grid to convert it to a triangular one. First, distance between each row of nodes of the square grid needs to be reduced to the height h of an equilateral triangle according to 3.1.

$$h = a\ sin60^o = \frac{1}{2}\sqrt{3}\ a \tag{3.1}$$

In equation 3.1, a is length of sides of a triangle as shown in figure 3.4. Second, each node of $\frac{i}{2}th$ row is displaced to the right by $\frac{a}{2}$ distance. See figure

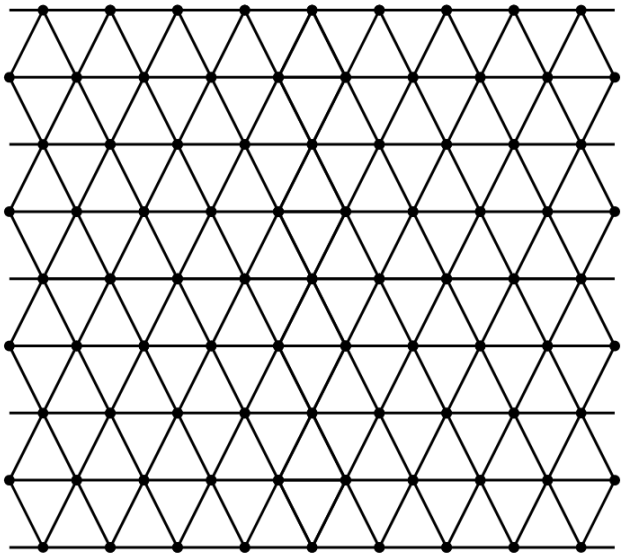


Figure 3.3: Triangular grid

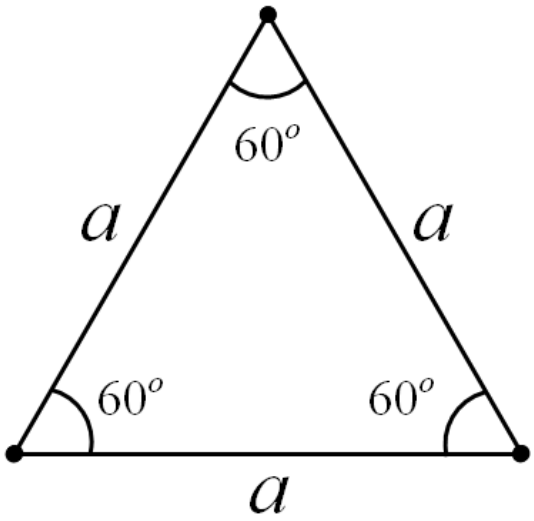


Figure 3.4: A typical equilateral triangle

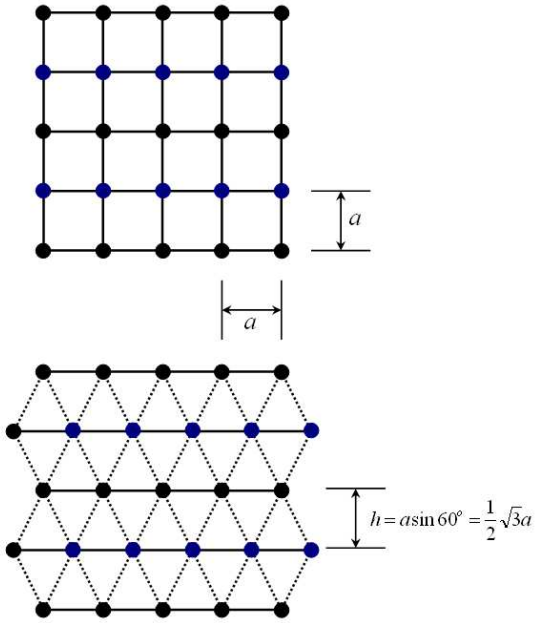


Figure 3.5: Triangular grid construction

3.5.

The mapping of the sampling points x_t, y_t of a handwritten document to nodes of a triangular grid and assignment of codes is done in the same fashion like those to a square grid (see figure 3.7 (right column)) except number of distinct code are reduced to 6 (from 0 to 5) rather than 8 (from 0 to 7) (see figure 3.6 and pseudocode given below).

Procedure FreemanTriangular
    document = extractDocument
    gridPoints = quantizePositions (*document*, *gridWidth*)
    fillGaps (*gridPoints*) /*Bresenham 's algorithm */
    assignDirections (*gridPoints*)

Procedure quantizePositions
    FOR each *samplingPoint* of *document*
        $x = samplingPoint.x$

```
        y = samplingPoint.y
        time = samplingPoing.time
        IF samplingPoing is a gap THEN
            gridPoint.x = −1
            gridPoint.y = −1
            gridPoint.time = time
            addResult (result, gridPoint)
        ELSE
            xWidth = gridWidth/2
            yWidth = sqrt(3.0) ∗ 0.50 ∗ gridWidth
            IF x is odd THEN
                x = x + gridWidth/2
            END IF
            gridPoint.x = x/gridWidth
            gridPoint.y = y/yWidth
            gridPoint.time = time
            addResult (result, gridPoint)
        END IF
        return result
    END LOOP

Procedure assignDirections
    FOR each gridPoint of feature vector
        dx = diffX (gridPoint, nextGridPoint)
        dy = diffY (gridPoint, nextGridPoint)
        time = gridPoint.time
        IF dx is greater than 0 AND dy is equal to 0
            directon = 0
        END IF
        IF dx is greater than 0 AND dy is less than 0
            direction = 1
```

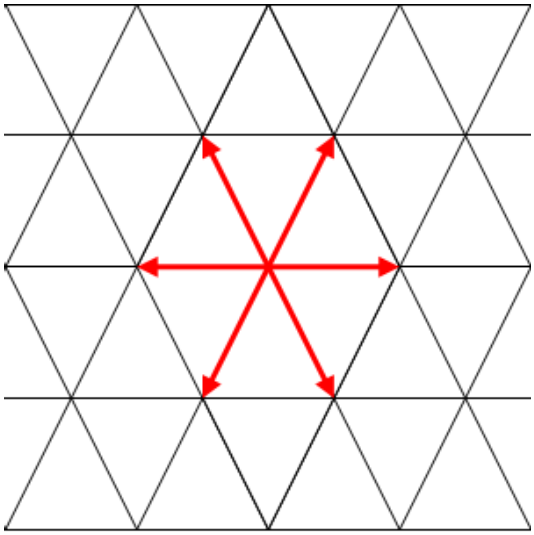


Figure 3.6: Triangular grid directions

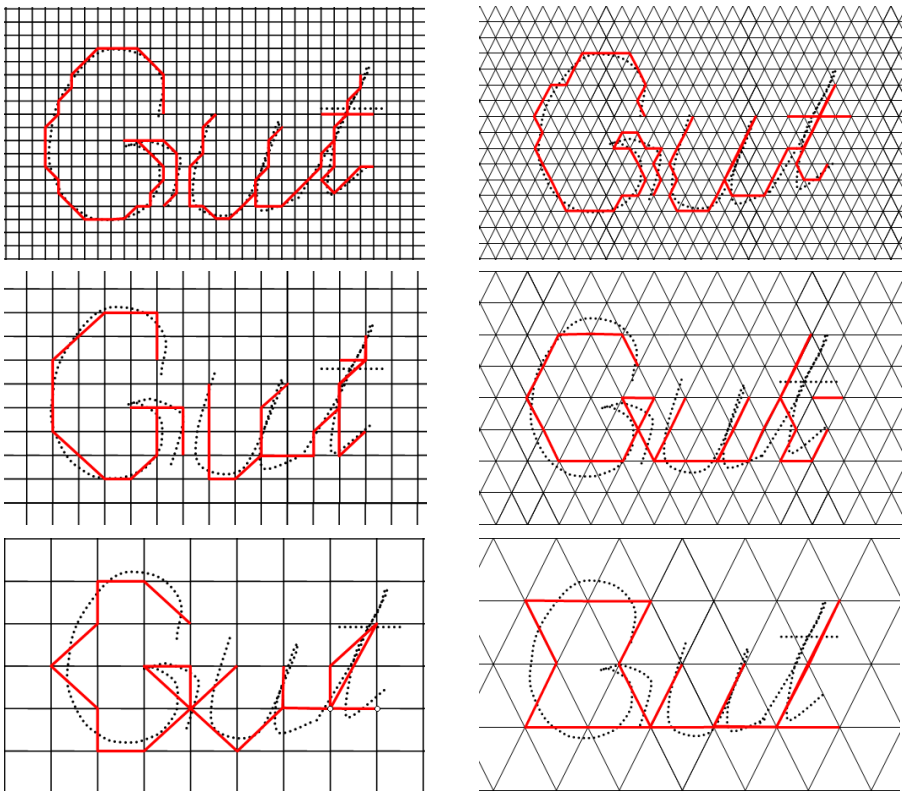
END IF
IF dx is less than 0 AND dy is less than 0
$direction = 2$
END IF
IF dx is less than 0 AND dy is equal to 0
$direction = 3$
END IF
IF dx is less than 0 AND dy is greater than 0
$direction = 4$
END IF
IF dx is greater than 0 AND dy is greater than 0
$direction = 5$
END IF
addResult (d, t, FREEMAN3);
END LOOP

The quality of generated code sequences depends upon the size of the

quantization square or quantization triangle. The larger the size of the tessellated geometric shapes (i.e. triangle or square) is the less accurate and short is the coded sequence. The smaller size of geometric shapes results in more precise but longer sequence of codes. In chapter 5, we have tested Freeman codes generated with grids of different sizes of geometric shapes.

Though there is always a tradeoff between the size of geometric shapes a grid is composed of and accuracy of the generated sequence of codes, the use of a size, which is too small, could lead to a coding of not only the user intended handwriting data but even of the involuntary noise of the hand movement, which would influence the retrieval performance negatively.

The two sequences of the same word written by two different persons are dissimilar in most of the cases due to different stroke order and writing style and the direction based nature of the features. That is why, it is not practically possible to get good results for a simultaneous retrieval in documents of different persons. Therefore, our search system is intended for an individual to search in his own documents only.



(a) Square grid superpositions (b) Triangular grid superpositions

Figure 3.7: Superposition of text on (square and triangular) grids of differnt sizes

Chapter 4

Testing and Performance Evaluation

In addition to introduction of triangular grid based Freeman code features, one of the main objectives of this study was to test and evaluate its performance against a different pen device called Pegasus PC Notes Taker shown in figure 4.1 [18]. Pen device specifications alongwith test environment, dataset collection and performance measures are explained in detail in the following sections.

4.1 *Pegasus* PC Notes Taker (PCNT)

PC Notes Taker (PCNT) captures handwriting online while it is being written on a simple paper of common use and stores it onto a PC in real time. Handwriting is also being displayed on the screen in real time while it is being stored on the PC. PCNT has an electron pen with which user writes his notes, memos or drawings. PC Notes Taker works with both PCs and notebooks by making an installation of a software provided with it. The provider of PCNT device has also made available a software development kit (SDK) written in Microsoft Visual C++. One can use this SDK to capture data from PCNT device and to further process it to be used for your application. In our case,

Figure 4.1: Pegasus PC Notes Taker device
[18]

we made use of SDK to capture from device and convert it into a format readable in our document retrieval system. Figure 4.2 shows how our system was provided with the handwritten documents acquired with PC Notes Taker device.

4.1.1 Features and Speciations

PC Notes Taker package comes with a cordless electronic pen including standard refill and batteries and a detachable base unit with USB cable. PC Notes Taker requires Microsoft Windows 98SE/ME/2000/XP operating system to work with. Its coverage area is upto A4 sheet of paper and resolution of 1200 DPI [18].

4.2 Data Collection

For testing and performance evaluation of our system, we made our own collection of testset documents written in two different language scripts i.e. Urdu and English. Since there was no suitable testset database in the community,

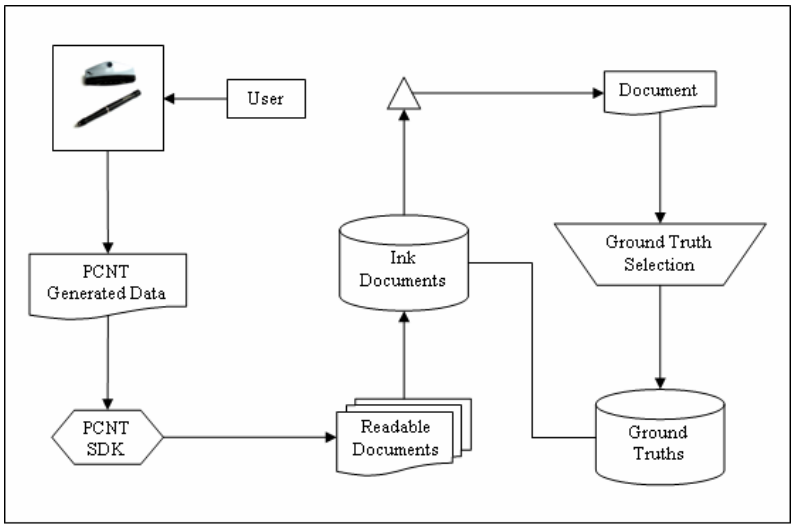


Figure 4.2: Benchmarking data collection flowchart

we built it our own by including a different Asian script of Urdu which has is quite similar to Arabic script in appearance. The available databases contain only off-line handwritten documents. Such documents contain no text but only symbols and characters of the scanned images of handwritten texts.

Pegasus PC Notes Taker (PCNT) device was used to collect all the handwriting documents. Figure 4.1 shows PC Notes Taker device. Its electronic cordless pen has ability to write on any kind of paper/surface fixed with detachable base unit show in figure 4.1. A sensor in base unit reads horizontal and vertical *(x, y)* position of the pen as it moves on the surface of paper. Through USB cable connected to PC, the position data is displayed on the PC screen and is also stored as an image on the PC for future use. Since we were interested in *x, y* positions rather than image of the handwriting, therefore we read this information by making use of PCNT software development kit (SDK) and stored in a format readable to our system. Figure 4.2 depicts how a user 's (handwritten) documents were made available in a database readable for our system.

We collected 80 documents from eight people, including both males and females of Asian origin. They were good in writing both Urdu and English

language scripts. Each writer was requested to write five documents in English and five in Urdu. A document consisted of an A4 sheet of paper and its contents had some text with repetitive words to make search of those words possible during evaulation step of our document retrieval system. Figure 4.3 depicts an example of the documents collected.

The sample point coordinates are located at the rate of 1200 units per inch of the surface vertically and horizontally. The sample rate varies and can go up to 50Hz. Whereas grid sizes are given in terms of points as integrals of the basis unit of pen coordinates. The basis unit comes from the sampling device. It means 10 points grid size correponds to 0.21 mm.

From our database of 80 documents we manually selected and tagged a set of 29 queries i.e. words and phrases and the positions of their 804 representative repetitions as expected true matches (see figure 4.4).

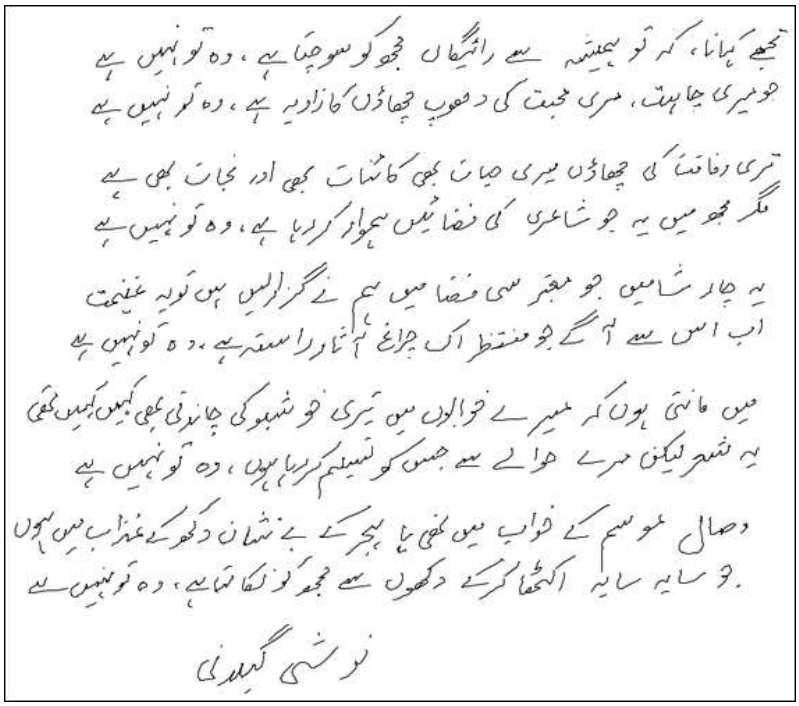
We calculated scores of our performance measures, explained in following section, using our set of queries and different settings of parameters i.e. grid type, grid size and threshold. The results of the set of parameters tested are listed and explained in chapter 5.

All the tests were conducted on a Dell notebook computer equipped with 1.80 GHz Centrino Duo processor, 1 GB RAM and Genuine Windows XP Media Center Edition 2005 operating system. Our document retrieval system is implemented in Java JRE 1.5.

4.3 Performance Measures

When a search operation is performed against a query, it results in a certain number of correct matches, mismatches and missed instances which ultimately contributes in the calculation of classical retrival measures i.e. *precision*, *recall rate* [19] and F_1-*Measure* [36]. See equations 4.1, 4.2 and 4.3 for measures of precision, recall rate and F_1-*Measure* respectively.

$$precision = \frac{matches}{matches + mismatches} \tag{4.1}$$



تجھے گمان، کہ تو ہمیشہ سے رائیگاں مجھ کو سوچتا ہے، وہ تو نہیں ہے
جو میری چاہت، مری محبت کی دھوپ چھاؤں کا زاویہ ہے، وہ تو نہیں ہے

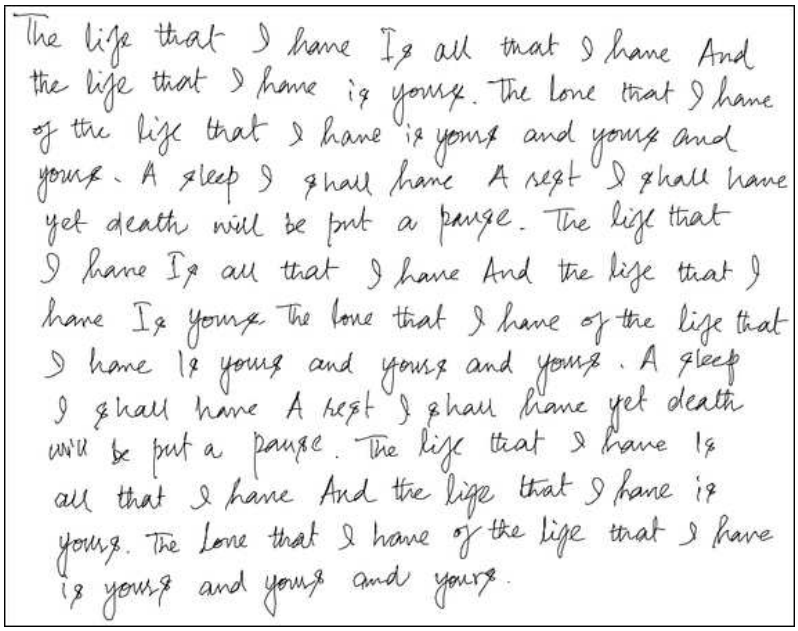
تری رفاقت کی چھاؤں میری حیات بھی کائنات بھی اور نجات بھی ہے
مگر مجھ میں یہ جو شاعری کی فضائیں ہموار کر رہا ہے، وہ تو نہیں ہے

یہ چار شامیں جو معتبر سی فضا میں ہم نے گزار لیں ہیں تو یہ غنیمت
اب اس سے آگے جو منتظر اک چراغ تاریک راستہ ہے، وہ تو نہیں ہے

میں مانتی ہوں کہ میرے خوابوں میں تیری خوشبو کی چاندنی بھی کہیں کہیں تھی
یہ شعر لیکن مرے حوالے سے جس کو تسلیم کر رہا ہوں، وہ تو نہیں ہے

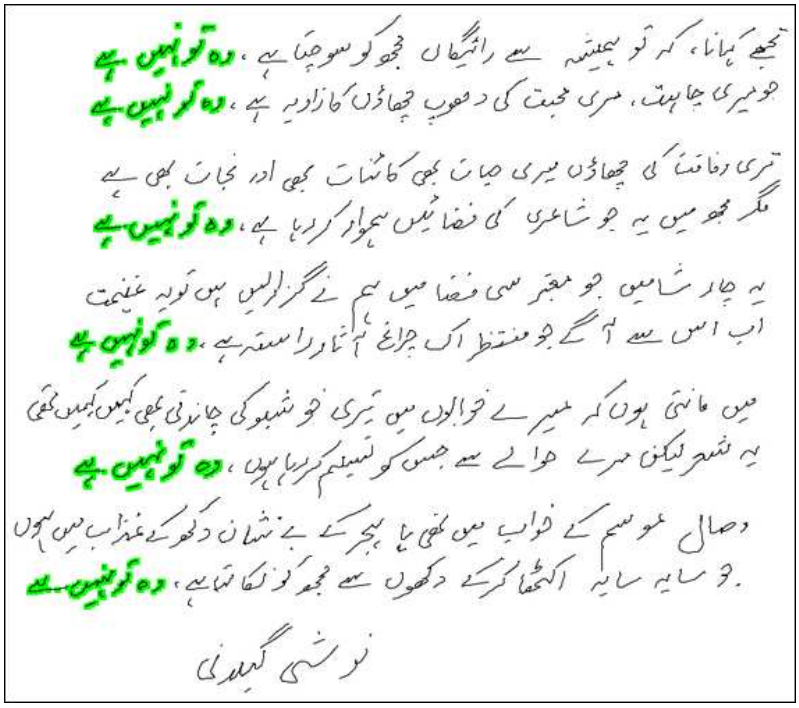
وصال موسم کے خواب میں بھی یا ہجر کے بے نشان دکھ کے عذاب میں ہوں
جو سایہ سایہ اکٹھا کر کے دکھوں سے مجھ کو نکالتا ہے، وہ تو نہیں ہے

نوشی گیلانی



The life that I have Is all that I have And the life that I have is yours. The love that I have of the life that I have is yours and yours and yours. A sleep I shall have A rest I shall have yet death will be put a pause. The life that I have Is all that I have And the life that I have Is yours The love that I have of the life that I have Is yours and yours and yours. A sleep I shall have A rest I shall have yet death will be put a pause. The life that I have Is all that I have And the life that I have is yours. The love that I have of the life that I have is yours and yours and yours.

Figure 4.3: Documents acquired with PCNT.



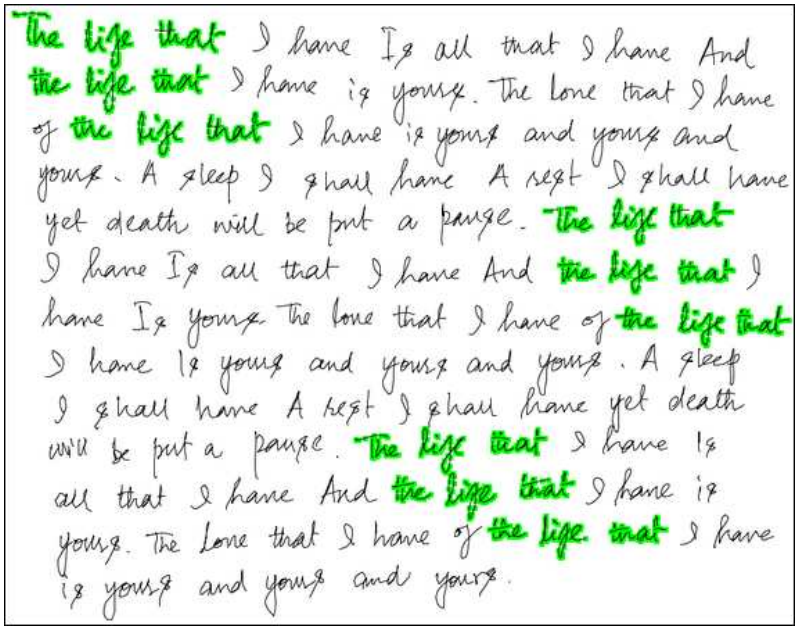


Figure 4.4: Selection of a query and its repetitions.

Table 4.1: English handwriting groundtruths used for benchmarking.

GT Nr.	**Description**	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
1	*God*	-	-	4	4	4	-	-	-
2	*line*	11	9	12	11	11	-	12	10
3	*lift*	-	5	5	5	5	-	5	5
4	*right*	-	6	5	5	5	-	5	5
5	*between*	3	3	3	3	3	-	3	3
6	*bloody*	4	4	4	4	-	-	5	4
7	*wanting*	-	-	-	-	6	-	-	-
8	*offering*	2	2	2	3	-	-	3	3
9	*She said*	-	6	5	5	5	-	5	5
10	*the american*	-	-	4	-	5	-	-	-
11	*the life*	-	9	-	8	-	-	9	9
12	*the life that*	-	8	-	8	-	-	9	9
13	*as soon as*	-	3	3	3	-	-	3	3
14	*that I have*	-	13	-	13	-	-	13	14
15	*you wait for about*	-	3	3	3	-	-	3	3
16	*the life that I have*	-	9	-	9	-	-	9	9

Table 4.2: Urdu handwriting groudtruths used for benchmarking

GT Nr.	**Description**	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
1	*duk*	-	13	13	-	13	11	-	11
2	*nadaan*	-	-	-	11	-	-	12	-
3	*daikho*	-	13	-	13	-	-	13	-
4	*mohabat*	-	5	6	6	-	-	6	6
5	*ka dukh*	-	5	6	-	6	5	-	6
6	*ahista ahista*	-	11	11	11	-	-	11	-
7	*kia khabar*	-	5	-	-	5	5	-	5
8	*rahi hai*	-	-	-	-	4	4	-	-
9	*karain tu kia*	-	-	-	12	-	-	12	-
10	*koon karta hai*	-	-	-	-	7	7	-	7
11	*wo tu nahi hai*	-	-	-	-	6	6	-	6
12	*ab nahi ho gi*	-	5	6	6	-	-	6	6
13	*tu ankhain beigh jati hain*	-	-	-	-	8	8	-	-

$$recall = \frac{matches}{matches + missings} \tag{4.2}$$

$$F_1 = \frac{2 \times precision \times recall}{precision + recall} \tag{4.3}$$

Measures of precision and recall rate have earlier been used to measure the performance of information retrieval and information extraction systems. Precision is the ratio of correct matches by the document retrieval system divided by the total number of the matches found by the system i.e. including matches and mismatches (See figure 4.5). Recall is defined to be the ration of correct assignments by the document retrieval system divided by the total number of correct assignments. F_1-*Measure* was initially introduced by van Rijsbergen [25] and it combines recall and precision with an equal weight as show in equation 4.3

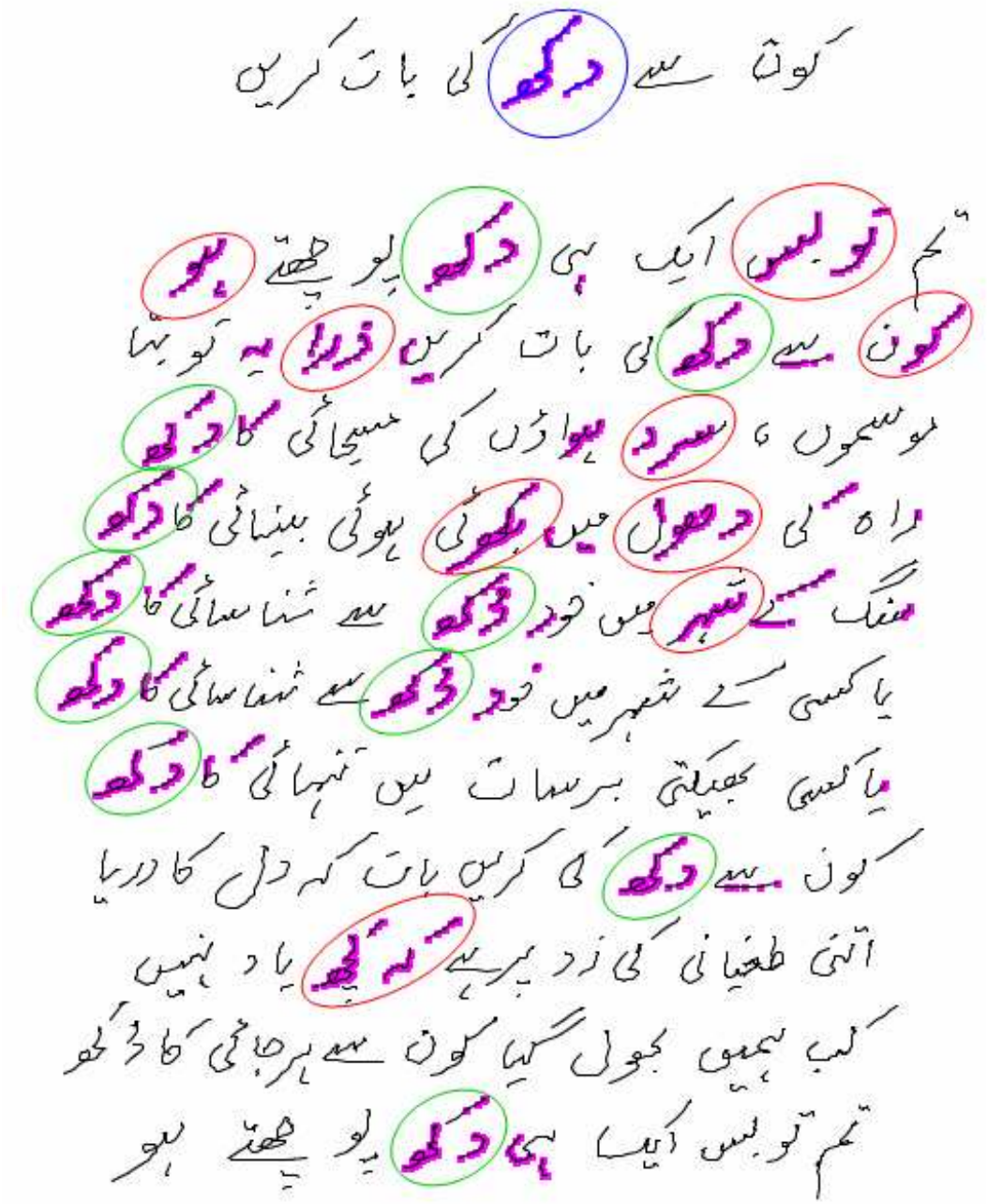


Figure 4.5: Illustration of the search process for a query word. Query word is marked with a blue circle, its correct matches with green and mismatches with red circles.

Chapter 5

Results and Discussions

There were two main objectives of our work: 1) to implement and benchmark triangular grid driven Freeman codes, a subtype of string features used by our document retrieval approach to extract features of a handwritten document , 2) to implement and evaluate the performance of our method against PC Notes Taker, a device which is used to collect handwritting data of users. We used most widely used measures i.e. *recall*, *precision* and F_1-*Measure* to evaluate the peformance in both of the above cases we were interested in. Chapter 4 has been provided with description of performance measures of *recall*, *precision* and F_1-*Measure*. See equations 4.2, 4.1 and 4.3.

We computed scores of *recall*, *precision* and F_1-*Measure* for a number of parameter settings i.e grid type (square and triangular), grid width and threshold. The results along with detailed description of parameters and their effects are given in the following sections of chapter.

5.1 Freeman Grid Codes

Freeman codes are one of the most promising string features which have been tested in past with our document retrieval approach by Schimke et al in [27]. There are two main differences of our current tests from the previous ones: 1)

in previous tests Freeman codes were generated on the basis of square grids whereas in current tests with both square and triangular grids to evaulate and compare triangular and square grid driven Freeman codes, 2) we used PCNT device to collect handwritings whereas in previous tests they used ioPen device of Logitech [14].

In the following we have shown the results that how Freeman codes perform when they are generated with two different kinds of grids alongwith comparison of their performance.

5.1.1 Square Grid based Freeman Codes

To evaluate performance of square grid generated Freeman features, all the ground truths were benchmarked against a range of threshold values (i.e. 0.40, 0.41, 0.41, 0.42, 0.43, 0.44, 0.45, 0.46, 0.47, 0.48, 0.49, 0.50) and grid widths (5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16) of the square grid used to generate Freeman codes. For each possible set of parameters, the number of correct matches, incorrect matches and missed instances of the search queries were counted in order to calculate precision, recall and F_1-Measure. In addition to these three measures of performance, average time duration to retrieve occurence for a query was also calculated. For selected threshold values and grid width, the scores of the three measures are given in table 5.1. Tables A.1, A.2, A.3 and A.4 in appendix are provided with scores of precision, recall F_1-Measure for all threshold (i.e. 0.40 - 0.50) and grid widths (i.e. 5 - 16) of square grid used for Freeman code generation. A graphical representation of precision and recall rate is give in figure 5.1.

In best case, we got precision of 79.47% at recall rate of 73.10% and their combined score, $F_1 - Measure$, of 0.76 with average time duration of 2007 milliseconds required for retrieval of the matches of a query. The optimal threshold and grid width were found to be 0.46 and 12 respectively.

The relation between retrieval efficiency of the system and grid width of a square grid is quite obvious from the plot. At a larger grid width, the representative Freeman code sequence does not provide true representation of the query and potential matches of that query but it is rather coarse representation. That is why, it results in making wrong matches due to absence of true feature representation and ultimately gives poor scores of precision and recall. In case of smaller grid width, the problem becomes two fold. On one hand, Freeman code sequence becomes longer which leads to more time for retrival and on the other hand it matches only to the exactly similar occurences of query and most of the others which slightly differ in shape are missed. F_1-Measure which combines precision and recall has been plotted against threshold in figure 5.2. It clearly shows that value $F_1 - Measure$ decreases both at lower and higher thresholds. When the threshold is lower, system finds more matches but most of them are false matches whereas at higher threshold sytem finds only few but correct matches to the query.

5.1.2 Triangular Grid based Freeman Codes

The second type of features which was tested differs from the one in previous section in terms of geometric shape (i.e. triangle rather than square) used to generate Freeman codes. To evaluate performance of triangular geometric shapes for feature extraction, the document retrieval system was benchmarked against different settings of parameters and ground truths used to evaluate square grid generated Freeman features in previous sections. For each possible set of parameters, all of its matches, mismatches and missing instances and time duration were counted for all ground truths to finally calculated precision,recall rate and $F_1 - Measure$ and average time for that settings of parameters. Numerical data of the benchmark for selected parameter settings of threshold and grid width is given in table 5.2. For complete set of used parameter settings, the data has been given in tables A.1, A.2, A.3 and A.4 in the appendix. Figure 5.3 shows graphical presentation of precision and recall

Table 5.1: Precision, Recall Rate, F_1-Measure and Average Time per Document for Square Grid Freeman Codes using Different Parameters of Grid Widths and Thresholds.

Size	**Threshold**	**Precision %**	**Recall %**	F_1	(Time ms)
6	0.44	86.06	64.74	0.74	8141
	0.45	81.77	71.97	0.77	8312
	0.46	76.51	78.78	0.78	8458
	0.47	69.69	85.34	0.77	8611
	0.48	62.07	89.26	0.73	8828
8	0.44	87.56	61.84	0.72	4446
	0.45	83.26	69.89	0.76	4536
	0.46	78.68	76.97	0.78	4644
	0.47	71.97	82.44	0.77	4726
	0.48	64.65	87.99	0.75	4859
10	0.44	87.84	58.94	0.71	2698
	0.45	84.08	67.24	0.75	2757
	0.46	78.98	74.80	0.77	2810
	0.47	73.13	81.23	0.77	2869
	0.48	66.59	85.88	0.75	2923
12	0.44	87.42	56.70	0.69	1929
	0.45	84.06	65.47	0.74	1970
	0.46	79.47	73.10	0.76	2007
	0.47	73.66	80.15	0.77	2052
	0.48	66.80	86.02	0.75	2090
16	0.44	88.65	51.94	0.66	1283
	0.45	85.11	59.97	0.70	1298
	0.46	81.49	67.74	0.74	1326
	0.47	75.61	75.40	0.76	1349
	0.48	68.74	81.76	0.75	1374

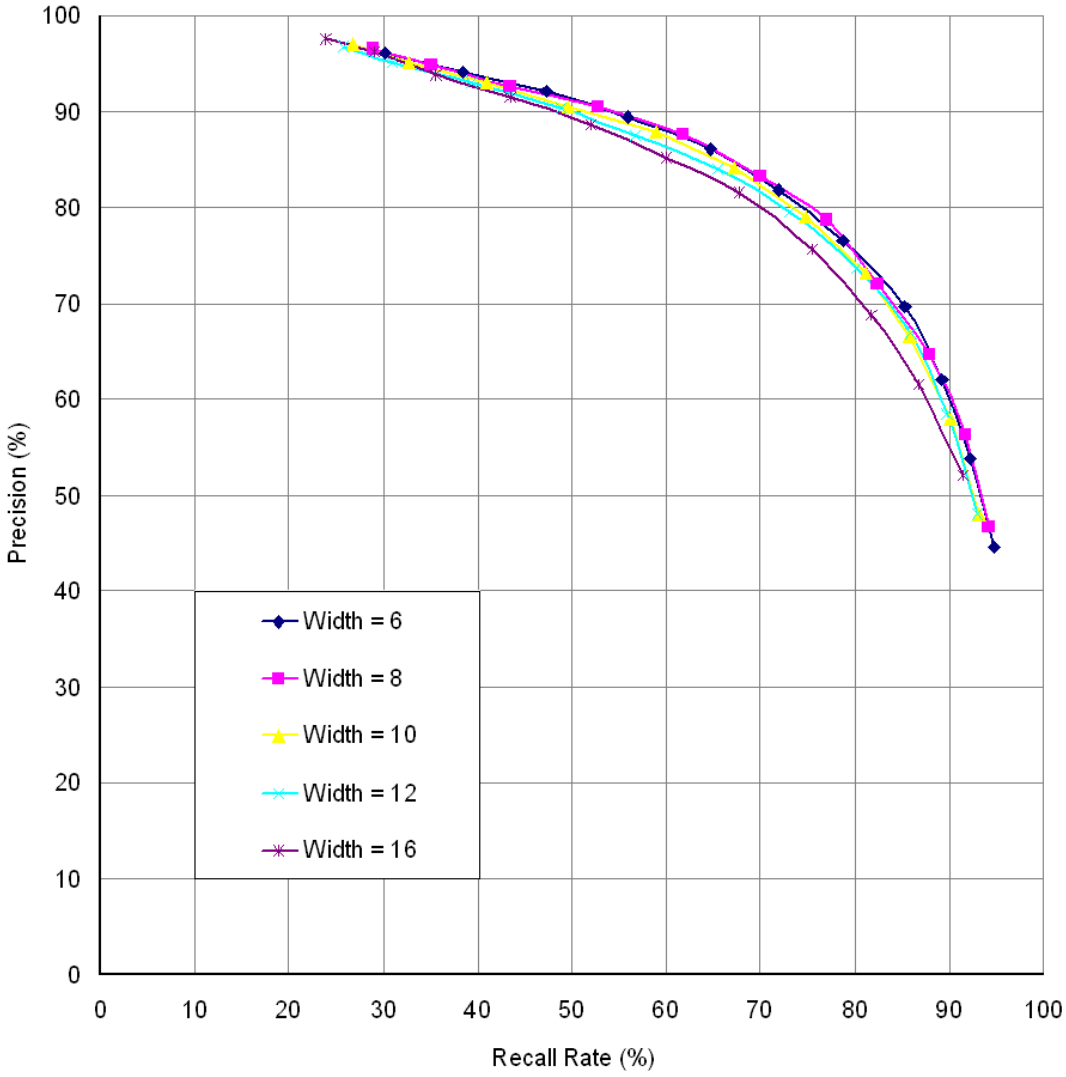


Figure 5.1: ROC (receiver operation characteristics) curve, showing the precision and recall for the Freeman features generated with square grid using different width sizes of the grid.

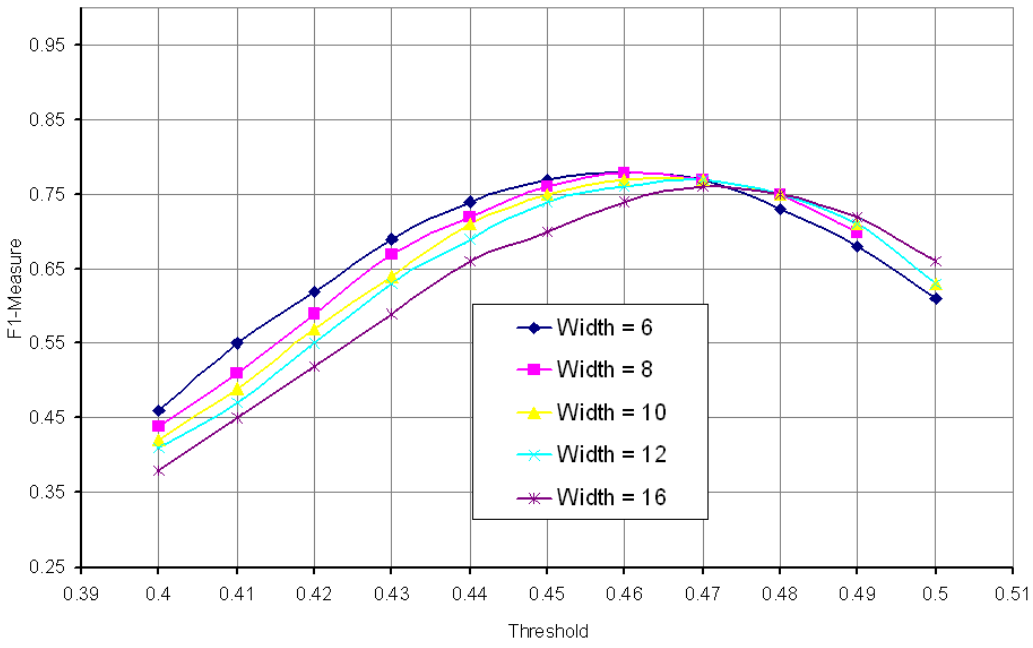


Figure 5.2: F_1-Measure of Freeman features plotted against threshold again using different width sizes of a square grid.

rates for grid widths of 6, 8, 10, 12 and 16. Additionaly, figure 5.4 shows behaviour of $F_1 - Measure$, a combined score, plotted against threshold for grid widths of 6, 8, 10, 12 and 16.

In best case, we got 68.82% precision at recall rate of 73.64% with corresponding $F_1 - Measure$ of 0.71. The former best case was found at grid width of 12. It can be seen from the numerical and graphical data that performance falls in both cases of bigger and smaller grid sizes. Threshold value of 0.45 was found to be the optimal one at various grid widths. $F_1 - Measure$ falls below and above threshold of 0.45.

5.1.3 Comparison of Square and Triangular Grid Driven Freeman Features

The motivation to implement triangular grid based Freeman features was to see if it performs better than those extracted using square grids. Therefore a comparison of both subtypes of Freeman features was made to elaborate difference in their performance. Table 5.3 shows precision, recall rate, $F_1 - Measure$ and time duration calculated for few of the parameter settings of both of square and triangular based Freeman features. For complete table for all parameter settings we have benchmarked system with, tables A.1, A.2, A.3 and A.4 has been given in the appendix. A graphical representation to make comparison relatively easy is given in figures 5.5 and 5.6 which depict the behaviour of measures of precision plotted against recall rate and $F_1 - Measure$ plotted against threshold repsectively for different grid widths of both square and triangular grids.

In best cases, the precision and recall rate of square driven features were 79.47% and 73.10% respectively whereas the precision and recall rate of triangle driven features were found to be 68.82% and 73.64% respectively. The best combined score i.e. $F_1 - Measure$ for square and triangle driven fea-

Table 5.2: Precision, Recall Rate, F_1-Measure and Average Time per Document for Triangular Grid Freeman Codes using Different Grid Widths and Thresholds.

Size	**Threshold**	**Precision %**	**Recall %**	F_1	**Time (*ms*)**
6	0.44	77.21	58.94	0.67	11778
	0.45	71.34	67.70	0.69	12063
	0.46	63.15	75.67	0.69	12293
	0.47	54.81	82.33	0.66	12541
	0.48	45.51	87.46	0.60	12787
8	0.44	72.05	61.56	0.66	7337
	0.45	65.42	70.82	0.68	7496
	0.46	58.17	78.80	0.67	7653
	0.47	49.33	84.96	0.62	7803
	0.48	40.41	89.91	0.56	7961
10	0.44	74.84	61.62	0.68	4422
	0.45	68.25	70.89	0.70	4529
	0.46	60.93	78.85	0.69	4612
	0.47	51.09	84.79	0.64	4734
	0.48	41.87	90.13	0.57	4804
12	0.44	75.96	63.94	0.69	3042
	0.45	68.82	73.64	0.71	3099
	0.46	61.06	80.20	0.69	3158
	0.47	52.66	86.35	0.65	3229
	0.48	42.33	90.70	0.58	3309
16	0.44	72.88	64.94	0.69	1744
	0.45	65.95	73.14	0.69	1776
	0.46	57.80	79.61	0.67	1816
	0.47	48.37	85.18	0.62	1858
	0.48	38.89	90.85	0.54	1899

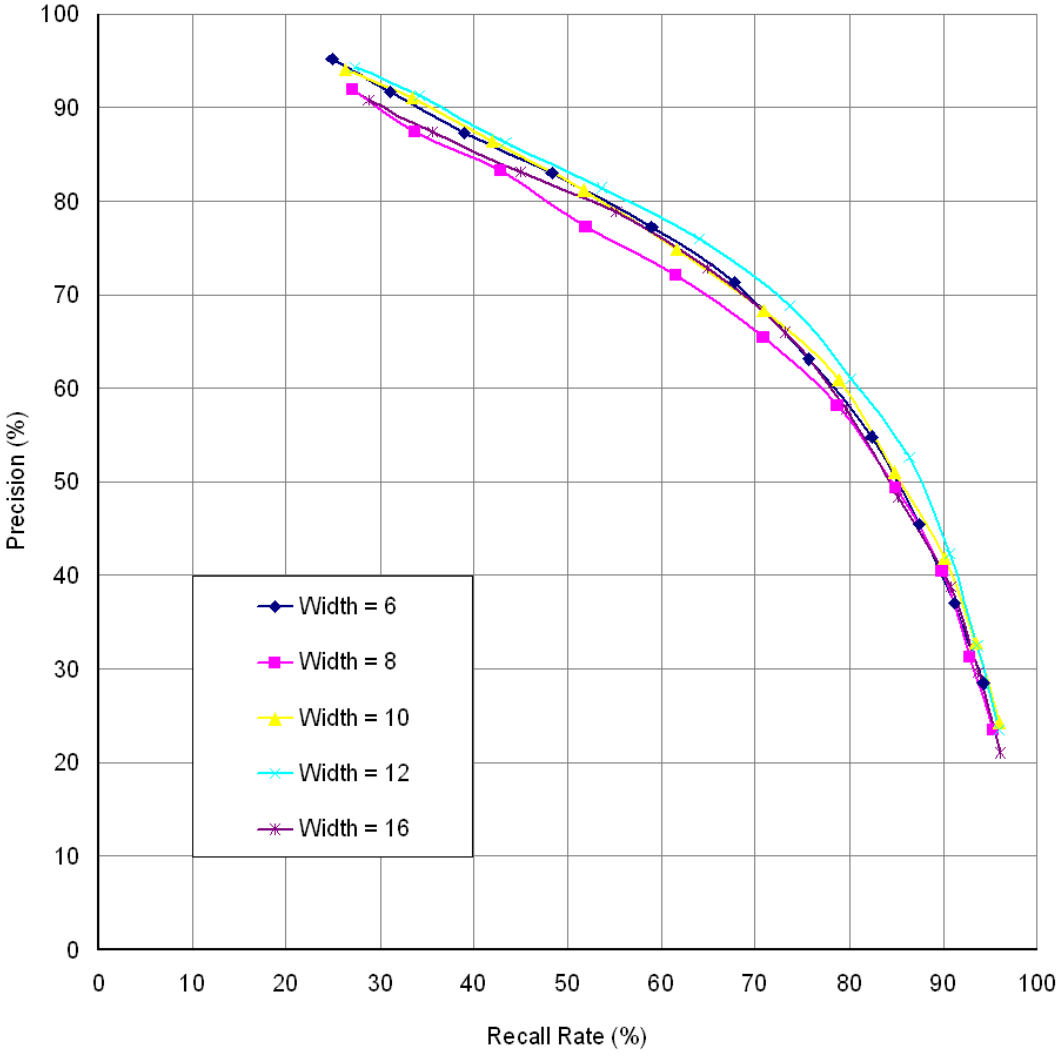


Figure 5.3: ROC curve, showing the precision and recall for the Freeman features generated with triangular grid using different width sizes of the grid.

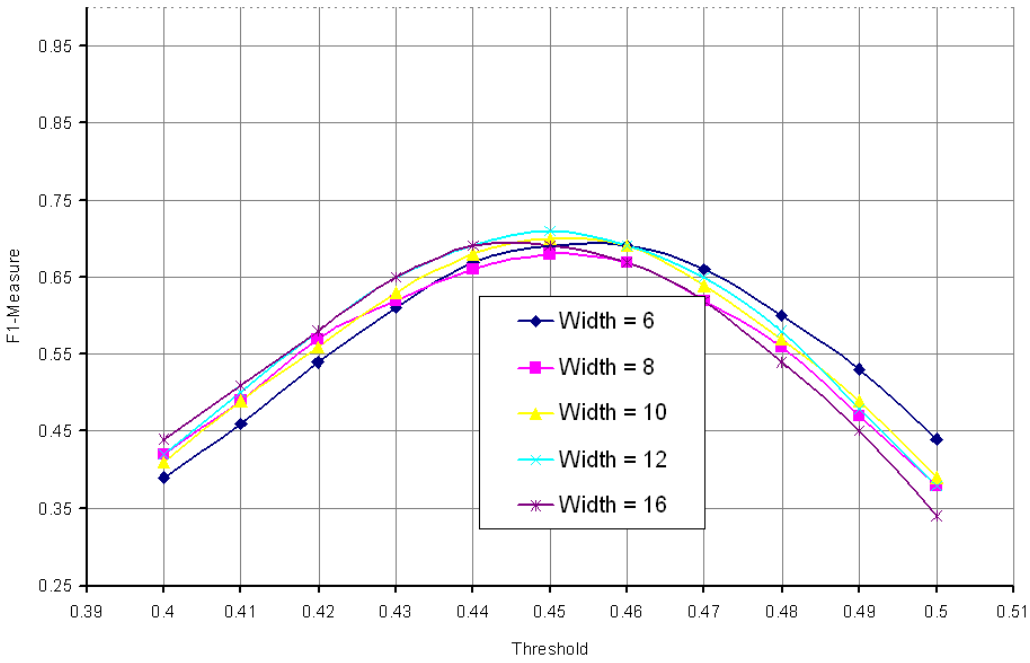


Figure 5.4: F_1-Measure of Freeman features plotted against threshold again using different width sizes of a triangular grid.

tures was 0.76 and 0.71 respectively. The best scores of precision and recall rate and their combined score of $F_1 - Measure$ of triangular and square grid driven features were obtained with grid width of 12. In best cases, the optimal thresholds for triangular and square grid driven features were found to be 0.45 and 0.46 respectively.

A noticeable difference between both kinds of features was found regarding time duration required for retrieval of document. In case of square driven feaures, the best scores of all the three measures of performance (i.e. precision, recall rate and $F_1 - Measure$ were found with average time of 2007 milliseconds whereas in case of triangle driven features, the best scores of three measures took 3099 milliseconds.

According to given numerical and graphical results, there is an obvious performance dominance of square grid based Freeman features over triangular grid ones. If we see gap of performance of two features in terms of measures of performance, there exists a difference of 10.65%, 0.54% and 0.05 of precision, recall rate and $F_1 - Measure$ respectively between the best cases of two types of features.

It is also noticeable from the combined score of $F_1 - Measure$ of both of the features plotted against threshold in figure 5.6 that the score is much closer at lower thresholds from 0.40 - 0.44 but the gap becomes wider beyond threshold of 0.44. It must be mentioned here that at lower thresholds retrieval rate is poor because system looks for exactly same instances of query text and at very high thresholds it matches query to the instances coarsely and returns a lot of wrong matches most of the time. If we see in the context of time duration required to retrieve matches, lower threshold leads to longer time and vice versa. The behaviour of time factor relative to threshold can also be seen from tabular data of both features in table 5.3 and the tables given in appendix.

Table 5.3: Precision, Recall Rate, F_1-Measure and Average Time per Document for Square and Triangular Grid Freeman Codes using Different Grid Widths and Thresholds.

		Square Grid				**Triangular Grid**			
Size	**Th**	**P %**	**R %**	F_1	**T (ms)**	**P %**	**R %**	F_1	**T (ms)**
6	0.44	86.06	64.74	0.74	8141	77.21	58.94	0.67	11778
	0.45	81.77	71.97	0.77	8312	71.34	67.7	0.69	12063
	0.46	76.51	78.78	0.78	8458	63.15	75.67	0.69	12293
	0.47	69.69	85.34	0.77	8611	54.81	82.33	0.66	12541
	0.48	62.07	89.26	0.73	8828	45.51	87.46	0.60	12787
8	0.44	87.56	61.84	0.72	4446	72.05	61.56	0.66	7337
	0.45	83.26	69.89	0.76	4536	65.42	70.82	0.68	7496
	0.46	78.68	76.97	0.78	4644	58.17	78.80	0.67	7653
	0.47	71.97	82.44	0.77	4726	49.33	84.96	0.62	7803
	0.48	64.65	87.99	0.75	4859	40.41	89.91	0.56	7961
10	0.44	87.84	58.94	0.71	2698	74.84	61.62	0.68	4422
	0.45	84.08	67.24	0.75	2757	68.25	70.89	0.70	4529
	0.46	78.98	74.80	0.77	2810	60.93	78.85	0.69	4612
	0.47	73.13	81.23	0.77	2869	51.09	84.79	0.64	4734
	0.48	66.59	85.88	0.75	2923	41.87	90.13	0.57	4804
12	0.44	87.42	56.70	0.69	1929	75.96	63.94	0.69	3042
	0.45	84.06	65.47	0.74	1970	68.82	73.64	0.71	3099
	0.46	79.47	73.10	0.76	2007	61.06	80.20	0.69	3158
	0.47	73.66	80.15	0.77	2052	52.66	86.35	0.65	3229
	0.48	66.80	86.02	0.75	2090	42.33	90.70	0.58	3309
16	0.44	88.65	51.94	0.66	1283	72.88	64.94	0.69	1744
	0.45	85.11	59.97	0.70	1298	65.95	73.14	0.69	1776
	0.46	81.49	67.74	0.74	1326	57.80	79.61	0.67	1816
	0.47	75.61	75.40	0.76	1349	48.37	85.18	0.62	1858
	0.48	68.74	81.76	0.75	1374	38.89	90.85	0.54	1899

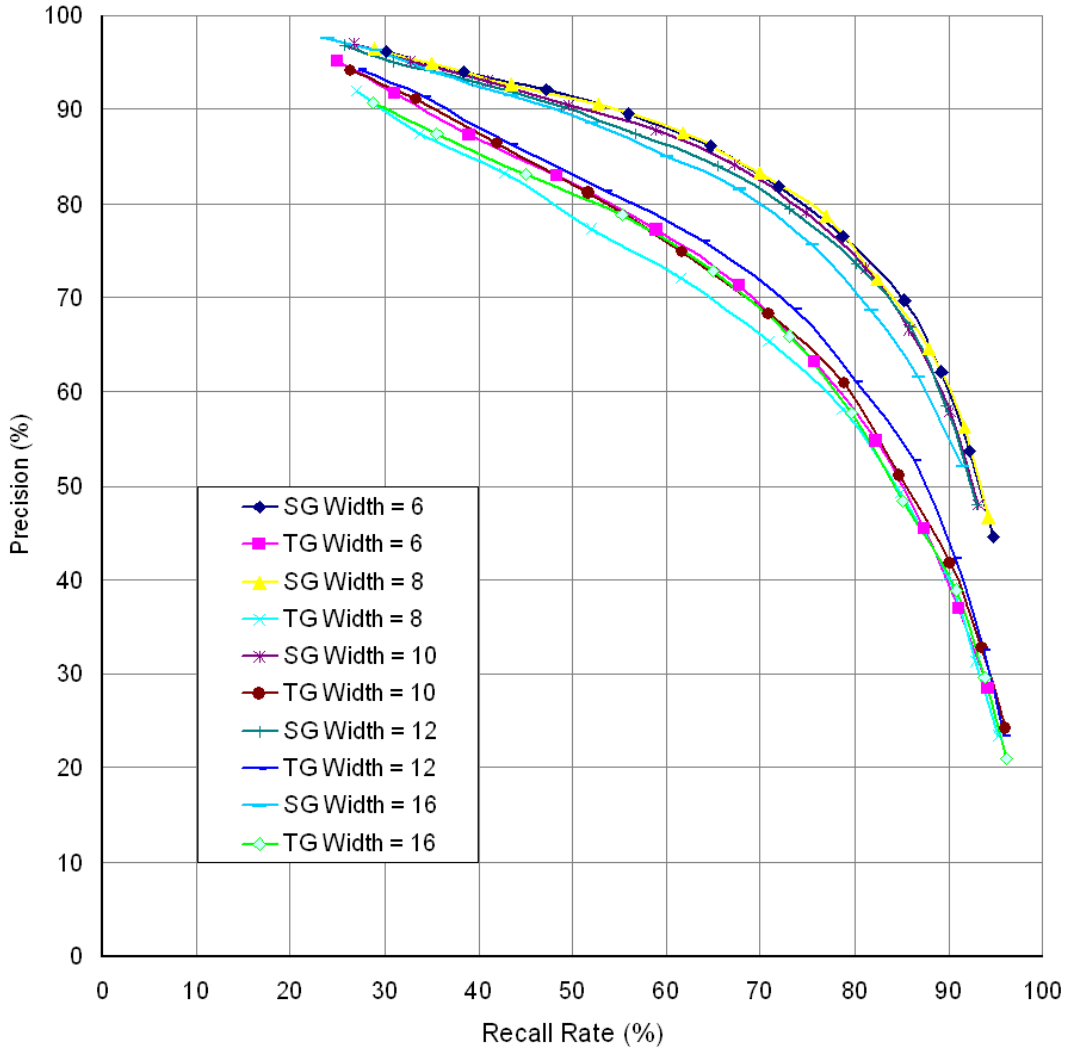


Figure 5.5: ROC curve, showing the precision and recall for the Freeman features generated with both square andvtriangular grid using different width sizes.

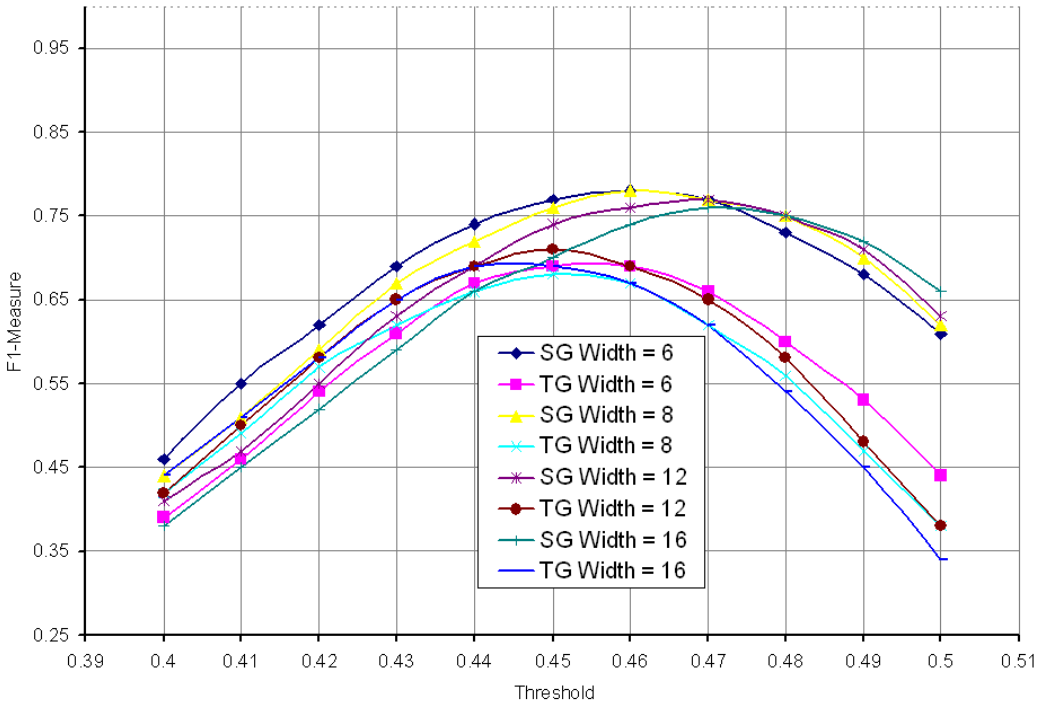


Figure 5.6: ROC curve, showing the precision and recall for the Freeman features generated with both square andvtriangular grid using different width sizes.

5.2 Performance with PC Notes Taker Device (PCNT)

One of the objectives of this work was to evaluate the performance of document retrieval system with PC Notes Taker (PCNT) device and compare it with that of already tested. Schimke et al. have already evaluated the retrieval system with ioPen device of Logitech in earlier work [27]. Since triangular driven features are new and have not yet been tested with ioPen device, we were only able to make a comparision on the basis of square grid driven features used for document retrieval within the documents acquired through ioPen and PCNT devices in earlier and this work respectively.

In table 5.4, data of precision, recall rate, $F_1 - Measure$ and time duration for document retrieval is given for both ioPen and PCNT devices. One can say in figures of performance measures that there is not much difference in the performance of document retrieval system with two devices. If we see the score of combined measure of $F_1 - Measure$ at grid width of 12, optimal grid width, it is 0.76 and 0.75 for PCNT and ioPen devices respectively.

There is a noticeable difference between the time duration for the retrieval of a document i.e. 2007 and 451 milliseconds for PCNT and ioPen devices respectively. But the comparison and performance of device cannot be concluded and discussed in the context of time duration because the evaluation of both devices were done on the systems of different computing capacity. To do such an analysis, one needs to run a benchmark simultaneously under the same computing environment.

Table 5.4: Precision (P), Recall Rate (R), F_1-Measure and Average Time (T) per Document obtained with ioPen and PCNT Device using Square grid driven Freeman Code Features at Different Grid Widths and Thresholds (Th).

	PCNT Device				**ioPen Device**			
Size	**P (%)**	**R (%)**	F_1	**T (ms)**	**P (%)**	**R (%)**	F_1	**T (ms)**
6	76.51	78.78	0.78	8458	81.50	81.50	0.81	1555
8	78.68	76.97	0.78	4644	82.30	78.90	0.80	1607
10	78.98	74.80	0.77	2810	78.30	78.80	0.78	572
12	79.47	73.10	0.76	2007	77.10	73.90	0.75	451
16	81.49	67.74	0.74	1326	73.80	71.60	0.72	284

Chapter 6

Conclusion

In this work we tested our document retrieval system with PC Notes Taker (PCNT) handwriting device and introduced a new subtype of features to the retrieval system.

The approach of our document retrieval system is distinguished from others in the sense that it does not involve any kind of textual recognition but a different technique of approximate string search from the field of bioinformatics where it is used to find similar gene sequences within a database of genome sequence of an organism. The system has the ability to work with any kind of text from any language and even with figures and sketches. Freeman features are used to convert handwriting signals into a string of codes. Freeman codes string represents a sequence of directions relative to time interval. This string is later used by the approximate string search algorithm to find the instances of query text in the handwriting.

To evaluate PCNT device with our system, we built a database with documents acquired through PCNT device. The documents were written in two different language scripts (i.e. English and Urdu). During evaluation, precsion and recall rate remained 79.47% and 73.10% respectively in contrast to previously achieved precsion and recall rate of 77.10% and 73.90% for ioPen of

Logitech. It must be noted that this comparision does not carry much worth and weight because both tests were conducted with different databases.

We introduced a new subtype of features with which Freeman features are extracted using triangular grid in addition to square grid. Triangular grid offers 6 equi-distant directions to the neighbouring nodes. In contrast, square grid gives 8 possible directions to the neighbours and 4 of them which are directed to the diagonally placed neigbouring nodes are placed a little bit far away in comparsion to each of two horizontally and vertically placed neighbour nodes. With triangle grid driven features, precision and recall rate were found to be 68.80% and 73.64% respectively whereas with square grid driven features, percision and recall rate remained 79.47% and 73.10% respectively. The little difference in performance of both subtypes of Freeman features is thought to be due to comparatively less number of available directions in triangular grid than those avaiable in square grid. In future, one may think of a composite type of features by combining both triangular and square grid driven features. Such a feature could complement weakness of each of two features by providing enough directions to neighbours and most of them placed at equal distant from the origin node.

Bibliography

[1] Accenture. Digital pen and paper - a point of view. *http://www.accenture.com/*. [Online; accessed 25-December-2006].

[2] S. F. Altshul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment tool. *Journal of Molecular Biology*, (215):403–410, 1990.

[3] M. W. Bern, J. E. Flaherty, and M. Luskin. *Grid Generation and Adaptive Algorithms*. New York: Springer-Verlag, 1999.

[4] F. J. Damerau. A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 1964.

[5] 3M Visual Systems Division. Capture ideas not notes. *http://www.3m.com/meetingnetwork/files/meetingguide_idea.pdf*, 2006. [Online; accessed 24-December-2006].

[6] DCLnews Editorial. Are digital pens something to write home about? *http://www.dclab.com/digitalpens.asp*, 2006. [Online; accessed December 17, 2006].

[7] M. Fonseca, B. Barroso, P. Ribeiro, and J. Jorge. Sketch-based retrieval of clipart drawings. In *Proceedings of the Working Conference on Advance Visual Interfaces*, pages 429–432, 2004.

[8] Nokia for business. White paper - nokia digital pen. *http://www.nokia.com/NOKIA_COM_1/About_Nokia/Press/White_*

Papers/pdf_files/whitepaper_nokiadigitalpen.pdf, 2006. [Online; accessed December 17, 200].

[9] H. Freeman. Computer processing of line-drawing images. *Computer Serveys*, 6(1):57–97, 1974.

[10] V. Govindaraju and H. Xue. Fast handwriting recognition for indexing historical documents. In *Proceedings of the First International Workshop on Document Image Analysis for Libraries (DIAL'04)*, pages 314–320, 2004.

[11] D. Gusfield. Algorithms on strings, trees, and sequences. *Cambridge University Press*, 1997.

[12] R. W. Hamming. Error detecting and error correcting codes. *Bell system technical journal*, (26(2)):147–160, 1950.

[13] Virtual Ink in Australia. Virtual ink mimio. *http://www.dansdata.com/mimio.htm*, 2000. [Online; accessed 24-December-2006].

[14] Logitech Inc. Logitech io digital pen. *http://www.logitech.com/*. [Online; accessed 25-December-2006].

[15] J. A. Janday and R. C. Davis. Making sharing pervasive: Ubiquitous computing for shared note taking. *IBM Systems Journal*, 38(4):531–550, 1999.

[16] C. V. Jawahar and A. Balasubramanian. Synthesis of online handwriting in indian languages. *http://hal.inria.fr/inria-00105121*, 2007. [Online; accessed 24-January-2007].

[17] V. I. Levenshtein. Binary codes capable of correcting deleltion, insertions and reversal. *Soviet Physics Doklady*, 10(8):707–710, 1966.

[18] Pegasus Technologies Ltd. Pegasus - digital pens. *http://www.pegatech.com/*. [Online; accessed 25-December-2006].

[19] J. Makhoul, F. Kubala, R. Schwartz, and R. Weischedel. Performance measures for information extraction. In *Proceedings of DARPA Broadcast News Workshop*, pages 249–252, Feb. 1999.

[20] S. Neef, J. V. Dijck, and E. Ketelaar. *Sign Here! Handwriting in the Age of New Media*. Amsterdam University Press, 2006.

[21] Vision Objects. Glossary. *http://www.visionobjects.com/handwriting-recognition/glossary/*, 2006. [Online; accessed December 17, 2006].

[22] J. M. Owen. Whose writing is this? authenticity and reproduction in the digital world. *http://eprints.rclis.org/archive/00001176/01/Whose-writing-is-this.pdf*, 2006.

[23] J. T. Jr. Phillips. Pens, pencils, and computers. *http://www.findarticles.com/p/articles/mi_qa3691/is_199404/ai_n8712863*, 1994. [Online; accessed 24-December-2006].

[24] IBM Research. Pen technologies. *http://www.research.ibm.com/electricInk/*. [Online; accessed 25-December-2006].

[25] C. J. Rijsbergen. *Information retrieval*. Butterworths, London, 1979.

[26] E. S. Ristad and P. N. Yianilos. Learning string edit distance. *Research report CS-TR-532-96*. Technical report, Department of Computer Science, Princeton University, Princeton, NJ., 1997.

[27] S. Schimke and C. Vielhauer. Document retrieval in pen-based media data. *Proceedings of the 2nd International Conference on Automated Production of Cross Media Content for Multi-channel Distribution*, pages 186–190, 2006.

[28] S. Schimke, C. Vielhauer, and J. Dittmann. Using adapted levenshtein distance for on-line signature authentication. *Proceedings of 17th International Conference on Pattern Recognition*, 2:931–934, 2004.

[29] L. Schomaker, E. de Leau, and L. Vuurpijl. Using pen-based outlines for object-based annotation and image-based queries. In *Lectures Notes in Computer Science*, volume 1614, pages 585–592. 1999.

[30] Digital Field Solutions. Digital pen and paper. *http://www.digitalfieldsolutions.com/digitalpen/index.html*, 2006. [Online; accessed 25-December-2006].

[31] S. Srihari, C. Huang, and H. Srinivasan. A search engine for handwritten documents. In *Proceedings of SPIET-IS&T Electronic Imaging*, pages 66–75, 2005.

[32] K. Sun and J. Wang. Similarity based matching method for handwriting retrieval. In *Proceedings of SPIE 2003 - Document Recognition and Retrieval X*, pages 156–163, 2003.

[33] VisionObjects. What is handwriting recognition? *http://www.visionobjects.com/handwriting-recognition/*. [Online; accessed 26-December-2006].

[34] Wikipedia. Digital paper-wikipedia, the free encyclopedia. *http://en.wikipedia.org/w/index.php?title=Digital_paper&oldid=96163545*, 2006. [Online; accessed 24-December-2006].

[35] Wikipedia. Hamming distance - wikipedia, the free encyclopedia. *http://en.wikipedia.org/w/index.php?title=Hamming_distance&oldid=103144064*, 2007. [Online; accessed 26-January-2007].

[36] Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings of the 22nd Annual International AMC SIGIR Conference on Research and Development in Information Retrieval*, pages 42–49, 1999.

Appendix A

Table A.1: Precision (P), Recall Rate (R), $F_1 - Measure$ and Average Time (T) per Document at Different Thresholds (Th) with Grid Widths 5-7.

		Square Grid				**Triangular Grid**			
Size	**Th**	**P %**	**R %**	F_1	**T (ms)**	**P %**	**R %**	F_1	**T (ms)**
5	0.40	95.72	29.46	0.45	11089	93.78	27.45	0.42	19798
	0.41	93.95	37.66	0.54	11182	89.77	33.15	0.48	20253
	0.42	92.09	46.59	0.62	11323	85.43	40.89	0.55	20733
	0.43	89.50	55.24	0.68	11623	80.14	49.11	0.61	21162
	0.44	86.30	63.71	0.73	12031	75.02	58.87	0.66	21696
	0.45	82.17	71.69	0.77	12373	68.38	67.33	0.68	22123
	0.46	76.67	78.27	0.77	12741	60.62	75.49	0.67	22640
	0.47	70.08	84.32	0.77	12598	52.01	82.34	0.64	23082
	0.48	62.85	88.46	0.73	12946	44.26	87.15	0.59	23507
	0.49	54.46	91.94	0.68	13318	35.75	90.97	0.51	23951
	0.50	45.12	94.29	0.61	13384	27.77	93.55	0.43	24336
6	0.40	96.18	30.24	0.46	7463	95.16	24.87	0.39	10833
	0.41	94.01	38.46	0.55	7620	91.72	31.04	0.46	11131
	0.42	92.16	47.25	0.62	7811	87.29	38.96	0.54	11293
	0.43	89.46	55.96	0.69	7963	83.00	48.34	0.61	11556
	0.44	86.06	64.74	0.74	8141	77.21	58.94	0.67	11778
	0.45	81.77	71.97	0.77	8312	71.34	67.70	0.69	12063
	0.46	76.51	78.78	0.78	8458	63.15	75.67	0.69	12293
	0.47	69.69	85.34	0.77	8611	54.81	82.33	0.66	12541
	0.48	62.07	89.26	0.73	8828	45.51	87.46	0.60	12787
	0.49	53.79	92.25	0.68	8978	36.97	91.13	0.53	13000
	0.50	44.58	94.74	0.61	9163	28.41	94.25	0.44	13334
7	0.40	96.74	28.66	0.44	5110	94.84	25.72	0.40	7350
	0.41	94.41	36.89	0.53	5205	91.57	32.51	0.48	7526
	0.42	92.75	44.81	0.60	5332	87.20	41.06	0.56	7687
	0.43	90.22	53.97	0.68	5417	82.18	49.72	0.62	7841
	0.44	86.83	62.67	0.73	5545	76.94	59.89	0.67	8020
	0.45	82.48	69.66	0.76	5661	69.38	69.52	0.69	8187
	0.46	77.04	77.23	0.77	5784	62.81	77.98	0.70	8354
	0.47	70.95	83.68	0.77	5895	54.29	85.33	0.66	8528
	0.48	63.80	88.45	0.74	5998	44.56	89.90	0.60	8705
	0.49	56.37	91.77	0.70	6124	36.10	93.11	0.52	8859
	0.50	46.24	94.41	0.62	6245	27.72	95.44	0.43	9049

Table A.2: Precision (P), Recall Rate (R), $F_1 - Measure$ and Average Time (T) per Document at Different Thresholds (Th) with Grid Widths 8-10.

		Square Grid				**Triangular Grid**			
Size	**Th**	**P %**	**R %**	F_1	**T (ms)**	**P %**	**R %**	F_1	**T (ms)**
8	0.40	96.49	28.82	0.44	4080	91.96	27.08	0.42	6723
	0.41	94.86	35.07	0.51	4189	87.44	33.69	0.49	6879
	0.42	92.67	43.40	0.59	4261	83.26	42.84	0.57	7033
	0.43	90.50	52.75	0.67	4352	77.32	51.95	0.62	7184
	0.44	87.56	61.84	0.72	4446	72.05	61.56	0.66	7337
	0.45	83.26	69.89	0.76	4536	65.42	70.82	0.68	7496
	0.46	78.68	76.97	0.78	4644	58.17	78.8	0.67	7653
	0.47	71.97	82.44	0.77	4726	49.33	84.96	0.62	7803
	0.48	64.65	87.99	0.75	4859	40.41	89.91	0.56	7961
	0.49	56.21	91.67	0.70	4929	31.36	92.93	0.47	8140
	0.50	46.67	94.26	0.62	5008	23.45	95.28	0.38	8289
9	0.40	97.02	27.84	0.43	3020	92.69	26.69	0.41	5157
	0.41	95.19	34.13	0.50	3081	89.47	33.10	0.48	5276
	0.42	92.65	42.13	0.58	3157	85.27	42.01	0.56	5376
	0.43	90.03	51.69	0.66	3222	80.14	51.05	0.62	5503
	0.44	86.98	60.61	0.71	3289	73.98	61.50	0.67	5639
	0.45	83.19	68.50	0.75	3357	67.77	71.08	0.69	5762
	0.46	78.13	75.95	0.77	3421	59.90	78.91	0.68	5906
	0.47	71.58	82.59	0.77	3495	51.50	85.26	0.64	5992
	0.48	64.49	87.68	0.74	3563	41.65	89.90	0.57	6097
	0.49	56.04	90.94	0.69	3627	32.44	93.19	0.48	6196
	0.50	46.94	94.05	0.63	3714	24.07	95.78	0.38	6870
10	0.40	97.05	26.76	0.42	2476	94.04	26.32	0.41	4065
	0.41	95.14	32.62	0.49	2528	91.04	33.37	0.49	4145
	0.42	93.03	40.93	0.57	2588	86.33	41.96	0.56	4259
	0.43	90.44	49.58	0.64	2635	81.15	51.73	0.63	4343
	0.44	87.84	58.94	0.71	2698	74.84	61.62	0.68	4422
	0.45	84.08	67.24	0.75	2757	68.25	70.89	0.70	4529
	0.46	78.98	74.80	0.77	2810	60.93	78.85	0.69	4612
	0.47	73.13	81.23	0.77	2869	51.09	84.79	0.64	4734
	0.48	66.59	85.88	0.75	2923	41.87	90.13	0.57	4804
	0.49	57.91	90.15	0.71	2989	32.76	93.51	0.49	4895
	0.50	48.00	93.21	0.63	3038	24.26	96.00	0.39	4984

Table A.3: Precision (P), Recall Rate (R), $F_1 - Measure$ and Average Time (T) per Document at Different Thresholds (Th) with Grid Widths 11-13.

		Square Grid				**Triangular Grid**			
Size	**Th**	**P %**	**R %**	F_1	**T (ms)**	**P %**	**R %**	F_1	**T (ms)**
11	0.40	97.36	25.51	0.40	2065	93.37	25.85	0.40	3199
	0.41	95.61	31.3	7 0.47	2114	89.68	33.13	0.48	3278
	0.42	93.17	39.86	0.56	2163	85.20	42.43	0.57	3350
	0.43	91.05	48.41	0.63	2210	79.60	52.45	0.63	3415
	0.44	87.81	57.04	0.69	2257	73.92	62.01	0.67	3488
	0.45	84.60	64.90	0.73	2305	68.12	71.62	0.70	3571
	0.46	80.35	73.36	0.77	2348	61.10	78.81	0.69	3631
	0.47	74.11	79.94	0.77	2403	52.58	85.32	0.65	3717
	0.48	66.41	86.01	0.75	2453	42.88	90.05	0.58	3791
	0.49	58.68	90.13	0.71	2494	33.02	93.86	0.49	3863
	0.50	48.83	93.06	0.64	2544	24.26	96.26	0.39	3946
12	0.40	96.81	25.73	0.41	1773	94.25	27.34	0.42	2799
	0.41	95.05	30.92	0.47	1814	91.34	34.25	0.50	2853
	0.42	93.02	39.38	0.55	1846	86.22	43.41	0.58	2914
	0.43	90.31	48.78	0.63	1888	81.37	53.60	0.65	2973
	0.44	87.42	56.70	0.69	1929	75.96	63.94	0.69	3042
	0.45	84.06	65.47	0.74	1970	68.82	73.64	0.71	3099
	0.46	79.47	73.10	0.76	2007	61.06	80.20	0.69	3158
	0.47	73.66	80.15	0.77	2052	52.66	86.35	0.65	3229
	0.48	66.80	86.02	0.75	2090	42.33	90.70	0.58	3309
	0.49	58.43	89.73	0.71	2132	32.53	93.65	0.48	3373
	0.50	47.96	93.02	0.63	2180	23.45	95.94	0.38	3438
13	0.40	96.88	25.17	0.40	1528	93.12	27.76	0.43	2230
	0.41	95.75	30.82	0.47	1570	89.65	34.30	0.50	2283
	0.42	93.42	37.63	0.54	1604	85.37	43.54	0.58	2339
	0.43	91.20	46.28	0.61	1631	79.94	54.70	0.65	2386
	0.44	88.16	55.07	0.68	1671	74.25	64.72	0.69	2438
	0.45	84.87	63.32	0.73	1708	67.59	73.71	0.71	2486
	0.46	80.72	71.89	0.76	1742	59.79	80.80	0.69	2544
	0.47	74.46	78.55	0.76	1778	51.41	87.03	0.65	2589
	0.48	67.10	84.17	0.75	1812	41.13	91.03	0.57	2646
	0.49	59.17	88.96	0.71	1848	31.82	93.91	0.48	2697
	0.50	48.91	92.48	0.64	1881	22.83	96.37	0.37	2751

Table A.4: Precision (P), Recall Rate (R), $F_1 - Measure$ and Average Time (T) per Document at Different Thresholds (Th) with Grid Widths 14-16

		Square Grid				**Triangular Grid**			
Size	**Th**	**P %**	**R %**	F_1	**T (ms)**	**P %**	**R %**	F_1	**T (ms)**
14	0.40	36.55	08.70	0.14	516	91.66	29.41	0.45	2393
	0.41	35.89	10.98	0.17	529	88.22	36.49	0.52	2028
	0.42	35.07	14.42	0.20	539	83.68	46.19	0.60	2074
	0.43	34.07	18.02	0.24	555	78.53	55.68	0.65	2157
	0.44	32.85	21.40	0.26	565	72.54	64.65	0.68	2421
	0.45	31.44	24.83	0.28	577	66.34	73.79	0.70	2197
	0.46	29.62	27.65	0.29	585	58.45	81.33	0.68	2267
	0.47	27.28	30.19	0.29	600	49.85	87.48	0.64	2317
	0.48	25.25	32.15	0.28	613	39.88	90.89	0.55	2348
	0.49	22.39	33.63	0.27	624	30.34	94.07	0.46	2391
	0.50	18.82	34.85	0.24	640	21.71	96.11	0.35	2436
15	0.40	97.43	24.00	0.39	1617	89.44	32.29	0.47	1875
	0.41	95.93	28.76	0.44	1329	85.94	40.12	0.55	1915
	0.42	93.82	36.09	0.52	1360	80.81	49.84	0.62	1966
	0.43	90.73	43.69	0.59	1396	74.92	59.41	0.66	2003
	0.44	87.87	53.00	0.66	1429	69.09	68.34	0.69	2045
	0.45	84.63	61.46	0.71	1456	61.87	76.48	0.68	2091
	0.46	80.42	69.43	0.75	1479	53.69	83.04	0.65	2137
	0.47	74.47	75.98	0.75	1514	45.25	88.38	0.60	2190
	0.48	67.84	83.04	0.75	1543	35.64	91.37	0.51	2228
	0.49	60.81	88.21	0.72	1577	27.08	94.34	0.42	2276
	0.50	50.66	92.03	0.65	1613	18.24	96.39	0.31	2324
16	0.40	97.51	23.83	0.38	1178	90.76	28.79	0.44	1595
	0.41	96.32	29.04	0.45	1214	87.46	35.57	0.51	1633
	0.42	93.82	35.61	0.52	1224	83.12	44.97	0.58	1669
	0.43	91.47	43.41	0.59	1262	78.81	55.23	0.65	1710
	0.44	88.65	51.94	0.66	1283	72.88	64.94	0.69	1744
	0.45	85.11	59.97	0.70	1298	65.95	73.14	0.69	1776
	0.46	81.49	67.74	0.74	1326	57.80	79.61	0.67	1816
	0.47	75.61	75.40	0.76	1349	48.37	85.18	0.62	1858
	0.48	68.74	81.76	0.75	1374	38.89	90.85	0.54	1899
	0.49	61.56	86.68	0.72	1421	29.67	93.81	0.45	1933
	0.50	52.04	91.53	0.66	1456	21.01	96.15	0.34	1977