# SEGMENTATION OF TOUCHING NUMERIC DIGITS

*Thesis submitted in partial fulfillment of the requirement for*

*The award of the degree of*

*Masters of Science*

*In*

**Mathematics and Computing**

*Submitted by*

**Keshwani Sharma**

**Roll no. - 30703010**

**Under**

**the guidance of**

**Mr. Rajiv Kumar**

**JULY 2009**

**School of Mathematics and Computer Applications**

**Thapar University**

**Patiala-147004 (PUNJAB)**

**INDIA**

# CERTIFICATE

*I hereby certify that the work which is being presented in the thesis entitled "**Segmentation of Touching Numeric Digits**" in partial fulfillment of the requirements for the award of degree of Master of Science , School of Mathematics and Computer Applications, Thapar University, Patiala is an authentic record of my own work carried out under the supervision of Mr. Rajiv Kumar.*

*The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.*

Keshwani Sharma

*(Signature of Student)*

*Regd. No. 30703010*

*This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.*

Rajiv Kumar

**(Mr. Rajiv Kumar)**
***Supervisor***
**SMCA, Thapar University**
**Patiala**

**Countersigned by:**

DBhatia 15.7.09

**Dr. S.S. Bhatia**
(Professor & Head)
School of Mathematics & Computer Applications
Thapar University, Patiala.

RK Sharma 22/7

**Dr. R.K. Sharma**
Dean of Academic Affairs
Thapar University
Patiala.

# *Acknowledgement*

*It gives me immense pleasure to acknowledge my sincere gratitude to my academic supervisor Mr. Rajiv Kumar, Lecturer, Department of School of Mathematics and Computer Applications , Thapar University , Patiala, who has helped and guided me for this work. His conversation and encouragement will always be remembered.*

*I would also like to thank Dr. S.S. Bhatia, Head, Department of School of Mathematics and Computer Applications, Thapar University, Patiala, for providing help and necessary facilities in the department for the completion of my thesis.*

*I would also like to pay my gratitude to my parents, for giving me the opportunity to open my eyes in one of the most beautiful planets I have ever known. It will be unfair if I don't thankful to my other family members who inspired, encouraged and cheered me throughout my thesis period.*

*Last but certainly not least, I would like to thank all of my friends, Especially, Antarpreet and Ramanjeet, who are always there for extending their support and suggestions.*

*(Keshwani Sharma)*

# ABSTRACT

In this work, we have discussed the segmentation system for numeric digits. We have studied the existing algorithms used for the segmentation of numeric digits and tried to explain it. As by casey.et.al, there are three approaches for segmentation (classical, recognition based and holistic methods). But under present study we focused on the classical approach of segmentation for numeric digits, which is based on character features alone (i.e. not aided by recognition). Various techniques like Min-Max, Hit and Deflect, Structural feature based technique and drop fall have studied for numeric digit segmentation. The problem of segmenting overlapping or touching character has been studied in detail. Variations of drop fall have also discussed which have advantages over one another. A hybrid heuristic, which is the mixing of all these variations have discussed in detail, which has promising results. This algorithm is used by many OCR designers to have complete OCR system. An improved segmentation algorithm, Extended Drop fall has also been discussed for proper segmentation of problematic cases.

# TABLE OF CONTENTS

# LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

# INTRODUCTION

People have always tried to develop machines which could do the work of a human being. The reason is obvious since for most of the history, man has been very successful in using the machines developed to reduce the amount of physical labor needed to do many tasks. With the invent of computer, it became a possibility that machines could also reduce the amount of mental labor needed for many tasks. Over the past fifty or so years, with the development of computers ranging from ones capable of becoming the world chess champion to ones capable of understanding speech, it has come to seem as though there is no human mental faculty which is beyond the ability of machines.

## 1.0 NATURAL LANGUAGE AND NATURAL LANGUAGE PROCESSING (NLP)

Language is a term most commonly used to refer to so called **"*natural languages*"**- the forms of communication considered to humankind. In linguistics the term is extended to refer the type of human thought process which creates and uses language. Essential to both meanings is the systematic creation and usage of systems of symbols– each referring to linguistic concepts with semantic or logical or otherwise expressive meanings. The most obvious manifestations are spoken languages such as English or Chinese. However, there are also written languages. When discussed more technically as a general phenomenon then, "language" always implies a particular type of human thought which can be present even when communication is not the result, and this way of thinking is also sometimes treated as indistinguishable from language itself. In Western Philosophy for example, language has long been closely associated with reason, which is also a uniquely human way of using symbols. In Ancient Greek philosophical terminology, the same word, logos, was used as a term for both language or speech and reason. Languages live, die, move from place to place, and change with time. Any language that ceases to change or develop is categorized as a dead language. Conversely, any language that is in a continuous state of change is known as a living language or modern language.

**Natural Language Processing is** a field of computer science and linguistics concerned with the interaction between computers and human (natural) languages. Natural language processing (NLP) systems convert information from computer databases in to readable human languages.

Natural language understanding systems convert samples of human language into more formal representations such as parse trees or first order logic that are easier for computer programs to manipulate. Many problems within NLP apply to both generation and understanding; for example, a computer must be able to model morphology (the structure of words) in order to understand an English sentence, and a model of morphology is also needed for producing a grammatically correct English sentence.In theory, natural language processing is a very attractive method of human –computer interaction. Early systems working in restricted "blocks worlds" with restricted vocabularies, worked extremely well, leading researchers to excessive optimism, which was soon lost when the systems were extended to more realistic situations with real world ambiguity and complexity. NLP has significant overlap with the field of computational linguistics, and is often considered a sub-field of artificial intelligence. The term natural language is used to distinguish human languages (such as Spanish, Swedish) from formal or computer languages (such as C++, Java)

## 1.1 NATURAL LANGUAGE VERSUS COMPUTER LANGUAGE

As natural language referred to as human language, where as computer language is a language acceptable to a computer system is called computer language. In natural language, each word has a definite meaning and can be looked up in a dictionary. In the similar manner, all computer languages have a vocabulary of their own. Each word of the vocabulary has a definite unambiguous meaning, which can be looked up in the manual meant for that language. The main difference between a natural language and computer language is that natural language has large vocabulary but most computer languages use a very limited or restricted vocabulary. This is because a programming language, by very its nature and purpose, does not need to say too much. Every problem to be solved by computer has to be broken down into discrete (simple and separate), logical steps, which basically comprise of four fundamental operations-input and output operations, movement of information within the CPU and

memory, and logical or comparisons operations. Each natural language has a systematic method of using the words and symbols of that language, which is defined by the grammar rules of the language, similarly the words and symbols of a computer language must also be used as per set rules, which are known as the syntax rules of the language. People can use poor and incorrect vocabulary and grammar, and still make them understood. However in the case of computer language ,we must stick to the exact syntax rules of the language, if we want to be understood correctly by the computer.Yet,no computer is capable of correcting and deducing meaning from incorrect instruction.So,computer languages are smaller and simpler than natural languages, but they have to be used with great precision.

## 1.2 PROBLEMS IN NATURAL LANGUAGE

As we know, a Natural language is that language that is spoken, signed or written by human being. Also it is easy for the communication. But still there are some problems. When writing has to be done in huge quantum, a lot of time is needed and also the more human effort is required. Also Sometimes people may use poor or incorrect vocabulary and grammar, which others may not be able to understand and resulting in creation of more problems, with reduction in accuracy. Also the pronunciation of words by two different persons can be different from each other with different meaning of the same word. Thus, the word spoken by one person may not be interpreted correctly by the other person due to the difference in pronunciation of the word. Also the written materials on the document some times, are not easily recognized by the others. A lot of time is wasted in doing all these. Due to all these problems, A system is required which can automatically do these works, with less human effort and also with greater accuracy.So,the processing of work automatically with less supervision of human is known as automatic processing.

## 1.3 AUTOMATIC PROCESSING

In this digital day and age, it has become obligatory to have all the available information in a digital form recognized by machine. The main reasons to convert paper in to electronic form are:

1) Electronic document is more economical and useful. So, no need to waste a lot of time and money because everything is done by machine with less human effort.

2) Unlimited copies can be made and small space is required for it. So, redundancy decreases.
3) Information on the documents can be shared by everyone. It is not limited to one person.
4) Documents can be edited as per we require. For example, there is a record of students on that document and want to include record of new student, then that document can be edited.
5) Convert document to other useful formats.
6) Whole file cabinets of paper can be stored on a few dozens CD ROMs or a single hard drive.
7) Publish on the web

Data can scan through image scanner but they have some limitations which are as follows:

1) Since the input document is stored as an image, instead of text, it is not possible to do any word processing of the document because the computer cannot interpret the stored documents as letters, numbers and special characters.

2) The storage required for storing the document as an image is much more than that required for storing the same document as a text. For example: a page of printed text ,having 2000 characters, can be stored as 2000 bytes by using the ASCII representation .A bitmap image representation of the same document will require 10 to 15 times more storage ,depending on the resolution of grid points.

So OCR technology is used to overcome these limitations, the scanner is equipped with a character recognition software(called OCR software) which convert bitmap images of characters to equivalent ASCII codes. That is, the scanner first creates the bitmap images of the document and then the OCR software translates the array of grid points into ASCII text, which the computer can interpret as letters, numbers and special characters.

## 1.4 CHARACTER RECOGNITION

Character recognition can be divided in to two categories:

1) Recognition of machine printed characters

2) Recognition of Handwritten characters.

**Recognition of machine printed characters**

Recognition of machine printed characters is a relatively simpler problem. Machine printed characters are uniform in height, width, and pitch assuming the same font and size are used .Also location of characters on the page is somewhat predictable. Machine printed text includes the materials such as books, newspapers, magazines, documents and various writing units in the video or still image. The problems for fixed-font, multi-font and omni-font character recognition is relatively well understood and solved with little constraint. When the documents are generated on a high quality paper with modern printing technologies, the available systems yield as well as 99% recognition accuracy. However, the recognition rates of the commercially available products are very much dependent on the age of the documents, quality of paper and ink.

**Recognition of Handwritten characters**

Recognition of handwritten characters is a much more difficult problem. Characters are non-uniform and can vary greatly in size and style. Even characters written by the same person can vary considerably. In general, the location of the characters is not predictable, nor is the spacing between them. In an unconstrained system, characters may be written anywhere on the page and may be overlapped or disjoint. Furthermore, there is the possibility of ambiguous or illegible characters. Very little information about the characters is available in unconstrained recognition.

## 1.5 TYPES OF HANDWRITING RECOGNITION SYSTEMS

Handwriting recognition systems can be divided to many different categories. Most important discriminating feature of systems is the time when the text is recognized. Recognition can be done even Offline or Online. Also the different sequences of text which form the units to be recognized as a single entity discriminates problem types: the text can be considered as characters, words or whole sentences. The set of characters, the writing style and the constraints on the writing have also a great influence on recognition methods. The distinctions re described in the following sections in more detail.

**Online System:** An Online system typically operates in real time and recognizes characters as they are being written down. On-line handwriting recognition involves the automatic conversion of text as it is written on a special digitizer, where a sensor picks up the pen-tip movements as well as pen-up/pen-down switching. That kind of data is known as digital ink and can be regarded as a dynamic representation of handwriting. The obtained signal is converted into letter codes which are usable within computer and text-processing applications.

The elements of an on-line handwriting recognition interface typically include:

- a pen or stylus for the user to write with.
- a touch sensitive surface, which may be integrated with, or adjacent to, an output display.
- a software application which interprets the movements of the stylus across the writing surface, translating the resulting curves into digital text.

**Offline System:** The handwriting recognition task heavily depends on the application it is used. First applications were systems that recognized text that was originally written on the paper. In such systems paper sheets are digitized to get two dimensional images. Features for recognition are then first enhanced and then extracted from the bitmap image by means of digital image processing. This type of recognition is called Offline recognition as the text is not recognized same time as it is produced but after the writing task is completed. Offline character recognition is known as *Optical Character Recognition* [1]. *(This will be discussed in next section)*

## 1.6 OPTICAL OR OFFLINE CHARACTER RECOGNITION (OCR)

OCR is the acronym for Optical Character Recognition. This technology allows a machine to automatically recognize characters through an optical mechanism. Human beings recognize many objects in this manner our eyes are the “optical mechanism”. But while the brain “sees” the input, the ability to comprehend these signals varies in each person according to many factors. By reviewing these variables, we can understand the challenges faced by the technologist developing an OCR system.

First, if we read a page in a language other than our own, we may recognize the various characters, but be unable to recognize the words. However, on the same page, we are usually able to interpret numerical statements-the symbols for numbers are

universally used. This explains why many OCR systems recognize numbers only, while relatively few understand the full alphanumeric character range.

Second, there is similarity between many numerical and alphabetical symbol shapes. For example, while examining a string of characters combining letters and numbers, there is very little visible difference between a capital letter “O” and the numeral “0”. As humans, we can read the sentence or entire paragraph to help us determining the accurate meaning. This procedure, however, is much more difficult for a machine.

Third, we rely on contrast to help us recognize characters. We may find it very difficult to read text which appears against a very dark background, or is printed over other words or graphics. Again, programming a system to interpret only the relevant data and disregard the rest is a difficult task for OCR engineers.

Documents are in the form of papers which the human can read and understand but it is not possible for the computer to understand these documents directly. In order to convert these documents in to computer processible forms, OCR systems are developed. OCR is the process of converting scanned images of machine printed or handwritten text, numerals, letters and symbols into a computer processible format such as ASCII. One of the major goals of research has been to make computer easier to communicate with and thus to make their benefits available to a much greater number of people.

## 1.7 VARIOUS STAGES OF OCR

Various processes are same in online and offline system. But the main difference in between them is scanning. Scanning is not required in online system but it is essential in offline system. So here the discussion is about the stages of offline system. There are following major stages in OCR:

1) Scanning
2) Preprocessing
3) Segmentation
4) Features Extraction
5) Classification/Recognition
6) Post-processing

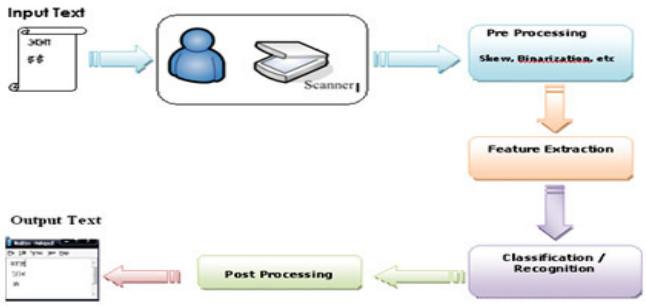


**Figure 1: An offline character Recognition System**

**1) Scanning:** Scanning is the process, to scan the documents by optical means and convert the signal in to digital form. So when the paper document is being scanned, it translates paper document in to an electronic format which can be stored in computer. The input documents may be typed text, pictures, graphics, or even handwritten material.

**2) Pre-Processing:** The raw data, depending on the acquisition type, is subjected to a number of preliminary processing steps to make it usable in the descriptive stages of character analyses. Preprocessing aims to produce data that are easy for the character recognition system to operate accurately. The main objectives of Preprocessing are:

**a) Noise Reduction:** The noise, introduced by the optical scanning device or the writing instrument causes disconnected line segments, gaps in lines and filled loops etc.So,this technique is used to remove these types of noises.

**b) Skew detection and Correction:** During the scanning process, the whole document or a portion of it fed through a loose leaf page scanner. Some pages are not fed straight in to the scanner, however causing skewing of the bitmapped images of these papers and thus requiring identification and rescanning by a quality control operation. Quality control is the slowest process in the system, because while scanning, it is difficult to feed the page exactly straight into the scanner while checking other things like focus,intensity,constraint,missing or extra paper and paper sequencing, so that there is a need to automate the quality assurance process.

**c) Compression:** Compression for character recognition requires space domain techniques for preserving the shape information. Two popular compression techniques are thresholding and thinning.

- Thresholding**:** In order to reduce storage requirements and to increase processing speed, it is often desirable to represent gray scale or color images as binary images by picking a threshold value.
- Thinning**:** While it provides a tremendous reduction in data size, thinning extracts the shape information of the characters.

**3) Segmentation:** The preprocessing stage yields a clean document in the sense that sufficient amount of shape information, high compression and low noise on normalized image is obtained. The next stage is segmenting its subcomponents. Segmentation is an important stage because the extent one can reach in separation of words, lines or characters directly affects the rate of the script.

**There are two types of Segmentation:**

**External Segmentation**, which is the isolation of various writing units, such as paragraphs, sentences or words. External segmentation decomposes the page layout into its logical units. It is the most critical part of the document analysis, which is a necessary step prior to the off-line character recognition.

**Internal Segmentation,** which is the isolation of letters, specially, in cursively written words. It is an operation that seeks to decompose an image of a sequence of characters into sub images of individual symbols.

**4) Features Extraction:** Features extraction is an important step in achieving good performance of OCR systems. However, the other steps also need to be optimized to obtain the best possible performance, and these steps are not independent. The choice of features extraction method limits or dictates the nature and output of the preprocessing step and the decision to use gray-scale versus binary image, filled representation or contour, thinned skeletons versus full-stroke images depends on the nature of the features to be extracted. So, the feature extraction stage analyzes a text segment and selects a set of features that can be used to identify text segment.

**5) Classification/Recognition:** It is the main decision making stage of an OCR system and uses the features extracted in the previous stage to identify the text segment according to preset rules.

**6) Post Processing:** While preprocessing tries to "clean" the document in a certain sense, it may remove important information, since the context information is not available at this stage. The lack of context information during the segmentation stage may cause even more severe and irreversible errors, since it yields meaningless

segmentation boundaries. It is clear that if the semantic information were available to certain extent, it would contribute a lot to the accuracy of the Character Recognition (CR) stages. On the other hand, the entire CR problem is for determining the context of the document image. The review of the recent CR research indicates minor improvements, when only shape recognition of the character is considered. Therefore, the incorporation of context and shape information in all the stages of CR systems is necessary for meaningful improvements in recognition rates. This is done in the post processing stage with a feedback to the early stages of CR.

In this work, the main focus is on the offline handwriting system for numeric digits. In which we will discuss the most important stage of the system which is the segmentation stage.

Numeric digits are usually used for writing Zip codes, postal address, bank check amount etc. The way of writing numerals are different for different languages. Since there are total ten digits in numerals but for all languages like English, Devanagri, Gurmukhi, Bangla etc have their own way of writing the digits. So here in this work, we are going to discuss about English Numeric digits which are from 0 to 9 and written as:

**0,1,2,3,4,5,6,7,8,9.**

These digits make up a number. The main thing occurs here is that how the number has written. It may be well written, overlapping, touching or disjointed. The successfully separating out the individual digits from the number becomes more difficult for the cases which we have mentioned above. A human mind can segment and recognize the digits simultaneously. But it is difficult for the system, to segment and recognize it simultaneously. The most important is the segmentation on which the accuracy depends. So in this work, we have studied existing segmentation techniques for touching numeric digits.

# LITERATURE SURVEY

[1] The material given by Nafiz Arica and Fatos T. Yarman-Vural serves as guide and update for the readers, working in the Character Recognition area. First, an overview of CR systems and their evolution overtime is presented. Then, the available CR techniques with their superiorities and weaknesses are reviewed. Finally, the current status of CR is discussed and directions for future research are suggested.

[2] Character Segmentation has long been a critical area of the OCR process. The higher recognition rates for isolated characters versus those obtained for words and connected character strings well illustrate this fact. A good part of recent progress in reading unconstrained printed for words and written text may be ascribed to more insightful handling of segmentation. R.G.Casey and E.Lecolinet describes a review of these advances. The aim is to provide an appreciation for the range of techniques that have been developed, rather than to simply list sources. Segmentation methods are listed under fore main headings. What may be termed the “classical” approach consists of methods that partition the input image in to sub images, which are then classified. The operation of attempting to decompose the image into classifiable units is called “dissection”. The second class of methods avoids dissection, and segments the image either explicitly, by classification of prespecified windows, or implicitly by classification of subsets of spatial features collected from the image as whole. The third strategy is a hybrid of the first two, employing dissection together with recombination rules to define potential segments, but using classification to select from the range of admissible segmentation possibilities offered by these sub images. Finally, holistic approaches that avoid segmentation by recognizing entire character strings as units are described.

[3] G .Congedo, G.Dimauro, S.Impedovo and G.Pirlo describes a complete procedure for the segmentation of handwritten numeric strings. The procedure uses a hypothesis then verification strategy in which multiple segmentation algorithms based on contiguous row partition work sequentially on the binary image until an acceptable segmentation is obtained. At this purpose a new set of algorithms simulating a Drop falling process is introduced.

[4] Punnose J. describes a segmentation-based recognition system for reading handwritten numeral on bank checks. An improved segmentation algorithm, Extended Drop Fall, has been incorporated into the system for proper segmentation of problematic cases. An improved method of choosing between the different segmentation algorithms has been introduced in order to utilize the Extended Drop Fall algorithm. Other modifications have also been incorporated into both the segmentation and recognition modules to increase performance and produce a more robust system.

[5] Salman Amin Khan describes the system for Automatic check amount verification. A segmentation workbench is developed for the purpose of studying current and potential techniques for numerical digit segmentation. The problem of segmenting overlapping or touching characters is studied in detail. Also, a hybrid heuristic involving a combination and variations of the "drop-falling" algorithm is developed which has promising results. This algorithm is then incorporated into a complete optical character recognition system.

[6] U.Pal, A.Belaid and Choisy C. describe a new scheme for automatic segmentation of unconstrained handwritten connected numerals. The scheme is mainly based on features obtained from a new concept based on water reservoir. A reservoir is a metaphor to illustrate the region where numerals touch. Reservoir is obtained by considering accumulation of water poured from the top or from the bottom of the numerals. At first, considering reservoir location and size, touching positions (top, middle and bottom) are decided. Next, analyzing the reservoir boundary, touching position and topological features of the touching pattern, the best cutting point is determined. Finally, combined with morphological structural features the cutting path for segmentation is generated.

[7] Segmented connected characters are an essential preprocessing step in character recognition applications. Traditional drop fall algorithm has proved to be an efficient segmenting method due to its simpleness and effectiveness. But it is subject to small convexness on the contour of characters. Xiujuan Wang, Kangfeng Zheng, and Jun Guo presents Inertial drop fall algorithm and Big drop fall algorithm to overcome this defect.

[8] Luiz E.Soares de Oliveira, Lethelier E. and Bortolozzi F. deals with a new segmentation approach applied to unconstrained handwritten digits. The novelty of the proposed algorithm is based on the combination of two types structural features in order to provide the best segmentation path between connected entities. This method was developed to be applied in a segmentation based recognition system. In this method, firstly the features used to generate basic segmentation points. Then, define segmentation paths depending on the encountered configurations with only few heuristic rules.

[9] Yi-Kai Chen and Jhing-Fa Wang describe a new approach of segmenting handwritten connected numeral string. Most algorithms for segmented connected digits mainly focus on the analysis of foreground pixels. Some others concentrated on the background pixels only. In this background and foreground analysis is used to segment handwritten connected numeral string. Thinning of both foreground and background pixels are first processed on the image of numeral string and the feature points on foreground and background skeletons are extracted separately.

[10] Brijesh Verma presents a novel contour code feature in conjunction with a rule based segmentation for cursive handwriting recognition. A heuristic segmentation algorithm is initially used to over segment each word. Then the prospective segmentation points are passed through the rule based module to discard the incorrect segmentation points and include any missing segmentation points.

[11] Dipankar Das and Rubaiyat Yasmin give an approach for the segmentation and recognition of unconstrained offline Bangla handwritten numerals. The projection profile based heuristic technique is used to segment handwritten numerals

[12] Rafael Palacios, Amar Gupta and Patrick S.Wang, in this paper, examines the issue of reading the numerical amount field. In the case of checks, the segmentation of unconstrained strings into individual digits is a challenging task because of connected and overlapping digits, broken digits, and digits that are physically connected to pieces of strokes from neighboring digits. The proposed architecture involves four stages: segmentation of the string in to individual digits, normalization, and recognition of each character.

[13] V.K. Madasu, M.H.M. Yusof, M.Hanmandlu and K.Kubik proposed an innovative approach for extracting signatures from bank check images and other documents which is based on the integration of the crop method with the sliding window technique. The idea is to estimate the approximate area in which the signature lies using the sliding window technique. In this approach, a window of adaptable height and width is moved over the image; one pixel at a time and the density of pixels within the window is calculated.

[14] Most algorithms for segmenting connected handwritten digit strings are based on the analysis of the foreground pixel distribution and the features on the upper/lower contour of the image. Z.Lu, Z.Chi, W.Siu and P.Shi present a new approach to segment connected handwritten two digit strings based on the thinning of the background regions. The algorithm first locates several feature points on the background skeleton and then constructed several possible segmentation paths by matching these feature points.

[15] U.Pal and Sagarika Datta proposed a robust scheme to segment unconstrained handwritten Bangla texts into lines, words, and characters. Then they used a concept based on water reservoir principle for the purpose.

[16] S. Ouchtati, B. Mouldi and A. Lachouri present an off line system for the recognition of the handwritten numeric chains. In this work is divided in two big parts. The first part is the realization of a recognition system of the isolated handwritten digits. The second part is the extension of system for the reading of the handwritten numeric chains constituted of a variable number of digits. The vertical projection is used to segment the numeric chain at isolated digits and every digit (or segment) will be presented separately to the entry of the system achieved in the first part (recognition system of the isolated handwritten digits).

[17] R. Palacois, and A. Gupta, describes a system for processing handwritten bank checks automatically. The system describes in this paper uses the scanned image of a bank check to read the check. There are four main stages in the system that focus on: the detection of courtesy amount block(CAB) within the image; the segmentation of string into characters; and the recognition of isolated characters; and the post processing process that ensure correct recognition.

# CHAPTER 2

# SEGMENTATION OF NUMERIC DIGITS

# SEGMENTATION OF NUMERIC DIGITS

## 2.0 INTRODUCTION

Segmentation is the process of separating out the individual characters which constitute a written character string. This is a straightforward process when the characters are well spaced and written with some degree of neatness. The problem becomes much more difficult when characters are touching, overlap, or disjointed.

In OCR system firstly, the preprocessing stage yields a clean document in the sense that sufficient amount of shape information, high compression and low noise on normalized image is obtained. The next stage is segmenting its subcomponents. Segmentation is an important stage because the extent one can reach in separation of words, lines or characters directly affects the rate of the script.

**TYPES OF SEGMENTATION**

There are two types of segmentation:

1) External Segmentation

2) Internal Segmentation

**External Segmentation**, which is the isolation of various writing units, such as paragraphs, sentences or words. External segmentation decomposes the page layout into its logical units. It is the most critical part of the document analysis, which is a necessary step prior to the off-line character recognition.

**Internal Segmentation,** which is the isolation of letters, specially, in cursively written words. It is an operation that seeks to decompose an image of a sequence of characters into sub images of individual symbols.

## 2.1 ROLE OF SEGMENTATION IN RECOGNITION SYSTEM

Segmentation is an operation that seeks to decompose an image of a sequence of characters into sub images of individual symbols. It is one of the decision processes in a system for optical character recognition. Its decision, that a pattern isolated from the image is that of a character, can be wrong. Errors prove segmentation often to make a major contribution to the error rate of the system. It is therefore very important to

minimize the error in character segmentation process, so as to reduce the error rate of character recognition process, as the success of recognition is based on that of segmentation.

Segmentation is the initial step in a three step procedure:

When a starting point is given in a document image, then

1) First step is to find the next character image.
2) Second step is to extract distinguish attributes of the character image.
3) Third step is to find the member of a given symbol set whose attributes best match those of the input, and output its identity.

The above sequence of steps repeated again and again, until no additional character images are found. So, a character is a pattern that resembles one of the symbols the system is designed to recognize .But to determine such a resemblance the pattern must be segmented from the document image. Each stage depends on the other and in complex cases it is paradoxical to seek a pattern that will match a member of the system's recognition alphabet of symbols without incorporating detailed knowledge of the structure of these symbols into the process. The segmentation decision is not a local decision, independent of previous and subsequent decisions. It is necessary to produce a good match to a library symbol, but not sufficient for reliable recognition. That is, a poor match on a later pattern can cast doubt on the correctness of the current segmentation/recognition result. Even a series of satisfactorily pattern matches can be judged incorrect if contextual requirements on the system output are not satisfied. Segmentation decision is interdependent with local decisions regarding shape similarity, and with global decisions regarding contextual acceptability.

Researchers in the 1960s and 1970's observed that segmentation caused more errors than shape distortions in reading unconstrained characters, whether hand or machine –printed. The problem was often masked in experimental work by the use of databases of well segmented patterns, or by scanning character strings with extra spacing. By the beginning of the 1980', workers were beginning to encourage renewed research interest to permit extension of OCR to less constrained documents.

## 2.2 HOW TO ORGANIZE THE METHODS?

A major problem in discussing segmentation is how to classify methods. The division according to use or non use of recognition in the process fails to make clear the

fundamental distinction among present day approaches. For example, it is not uncommon in text recognition to use a spelling corrector as a post –processor. This stage may propose the substitution of two letters for a single letter output by the classifier.

The distinction between methods is based on how segmentation and classification interact in the overall process. Segmentation is done in two stages, one before and one after image classification. Basically an unacceptable recognition result is re-examined and modified by a (implied) resegmentation. This is a rather "loose" coupling of segmentation and classification.

A more deeply interaction between the two aspects of recognition occurs when a classifier is invoked to select the segments from set of possibilities. In this family of approaches, segmentation and classification are integrated. To some observers even if appears that the classifier performs segmentation since, conceptually at least, it could select the desired segments by exhaustive evaluation of all possible sets of sub images of the input image.

According to a survey of vast literatures done by [2], there are three "pure" strategies for segmentation, plus numerous hybrid approaches that are weighted combination of these three. So that three elementary strategies are:

a) The classical approach, in which segments are identified based on "character-like" properties. This process of cutting up the image into meaningful components is given a special name, "dissection".

b) Recognition-based segmentation, in which the system searches the image for components that match classes in its alphabet.

c) Holistic methods, in which the system seeks to, recognize words as a whole, thus, avoiding the need to segment into characters.

There are many strategies for segmentation which are combination of one or more of above three pure ones. We can show these three strategies to occupy orthogonal axes. Hybrid methods can be represented as weighted combination of these lying at points in the intervening space (Fig 2.1)

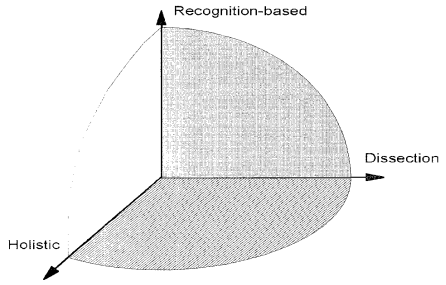


**Fig 2.1: A three dimensional representing the strategies of segmentation**

Out of the three strategies mentioned above, we will use the classical approach for the segmentation of numeric digits i.e., separating out the individual digits which make up a number. As we have mentioned at the starting of the chapter, that segmentation is easy for the cases where the numbers are well written or machine printed, but it is the most difficult task if the digits which constitute the number are touching , overlapping or disjointed. For example, handwritten amount on checks, it may be well written, touching or disjointed, because of the wide variations in handwriting styles. Here are some figures (2.2, 2.2a to 2.2c) which shows well written, touching, overlapping and disjointed numbers:

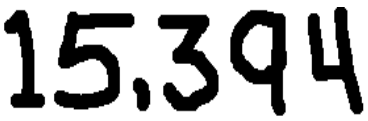


**Figure2.2: Example of well written number**


**Figure2.2: (a) Example of touching**


**(b) Example of disjointed**


**(c) Example of overlapping**

There are several steps commonly employed for segmenting, an image of numerical string, before segmenting any string of symbols, the first step is to identify the connected components in the image of the string. Connected pixels are usually grouped together using some sort of search algorithm (which will be discussed further). Once the connected components are found, then they have to be classified according to whether they represent part of a character, a whole character, or two or more characters. This is commonly done using a variety of heuristics which make use of the components aspect ratios and positions. If a connected component is classified as being made up of more than one character, then another 'splitting' heuristic is used.

For example, a common algorithm used today is the min-max algorithm which separates the connected characters by splitting the connected region from a minimum on the upper contour to a maximum on the lower contour. The heuristics tend to be effective in general, but still fail on a significant portion of cases with many connected or overlapping digits

Now, the basic processes common to any system dedicated to handwritten numeral segmentation and the assumption s on which they are based will be explored in this chapter.

## 2.3 THRESHOLDING OF THE NUMERAL IMAGE

Before any segmentation or processing on a numerical image can be done, it is very useful to convert the image from grayscale or color to a black and white bitmap. This is a very simple process which entails just scanning the image and changing each pixel value to either black or white depending on whether they are above or below a given threshold. The threshold chosen should be a function of the intensity range of the pixels in the image.

The threshold-determining function can be made arbitrarily complex to optimize the conversion process, but setting the threshold to the average of the minimum and maximum threshold values is usually sufficient. This heuristic will break down only when the average of the minimum and maximum pixel values is either less than many of the pixels meant to represent black or greater than many of the pixels meant to represent white. This could happen if, say, most of the pixels values fall in the 150-255 range, but there is one pixel with a value of zero. Since this is unlikely to happen (devices that scan the images are very unlikely to produce images with random pixel values so far from the range of most of the other pixels), it is usually not necessary to worry about these special cases. Alternate heuristics which could take care of the above-mentioned problem (but have their own cases on which they fail) include basing the threshold on the weighted average of all the pixels or determining white or black pixels based on changes in contrast (for example, if two adjacent pixel values differ from each other significantly, than set the lighter one to white and the darker one to black.

## 2.4 EXTRACTING CONNECTED COMPONENTS

The first step in actually segmenting any numeral image is to extract out each of the connected components. In order to do so, it is necessary to find a black pixel and to then use a search algorithm to find all of the pixels which are connected to it. Once all of those pixels are recognized, they should be grouped together into one connected component. Then another, as yet not looked- at, pixel needs to be found on which the process of searching for all of the connected components is repeated. The overall process is repeated again and again until every black pixel in the image is classified as being part of a connected component.

Two pixels are connected to each other if they are adjacent. Since the characters which are being segmented are at least several pixels wide, it is unusual for two pixels which are only diagonal with respect to each other (they share no other adjacent neighbors) to be part of the same character. This is why, in searching for connected pixels to a given pixel, it only necessary to take into consideration the pixels above, below, to the right, and to the left of that pixel. If the digits in the numeral image are well separated when they are written, then each connected component would represent a whole character in the numeral string and the task of segmentation would be done. Unfortunately, this is seldom the case. If any of the digits are connected or disjointed, then any of the connected components could represent a single character, more than one character, or part of a character. So, there is need to classify the connected components.

## 2.5 CLASSIFICATION OF CONNECTED COMPONENTS

Most segmentation methods today classify the connected components using information on their sizes and positions. The ratio of a connected component's bounding box height to its width is the most common indicator of what fraction of characters the component is composed of. If the width is greater than a certain multiple of the height, then it is very unlikely that the component is only one character. More likely, it is either more than one character or part of a character that is a horizontal line.

Although using the width-to-height ratio is useful for the more standard cases, it is not difficult to find cases on which it breaks down. For example, very often the numbers 2, 3, 5, or7 are written so that they are unusually wide. Moreover, a 1 and 7 could be

connected together and the resulting connected components would have a height to width ratio identical to the 7 alone. Because of this, it is a nontrivial problem to robustly determine how many characters a connected component is composed of.

Using a connected component's width-to-height ratio and position is much more useful in identifying whether or not the component is a separated segment of a larger character. The most common case of this happening is when the top stroke of a five is separated from the rest of it. In this case the top stroke, which is an almost completely horizontal line, would have a very high width to height ratio (much higher than even a connected component composed of many characters). Not only that, but the height of the component will be significantly less than the heights of the other connected components in the image and its bounding box will be contained in the upper half of the image. Based on these assumptions, it is not difficult to find the upper stroke of a five (the only characters that it could conceivably be confused with are the commas and decimal points, but those are located in the lower half of the image and would not have as high of a width to height ratio.

So, classifying the components is not an easy thing to do. More complicated heuristics based on features other than the aspect ratios and positioning of the components will have to be developed in order to more accurately classify them. These heuristics could potentially make use of min-max points on the upper and lower contours and come type of complexity measure of the pixels making up the component, in conjunction with the aspect ratio and positioning, to make a better guess at properly classifying the component.

## 2.6 CURRENT SEGMENTATION TECHNIQUES

Many different methods have been developed for the purpose of separating two connected characters. Despite the wide variety in the methods developed, they can be divided in to two groups:

1) Those that are based only on character features

2) Those that make use of some type of pre recognition process

So, here main focus would be on developing a heuristic which falls under first of the two categories.

Once a connected component is classified as being composed of more than one character all that is left to be done is to determine how many characters the

component is composed of and how to optimally separate the characters. Most of these components will be composed of two characters, but in unusual cases they can be composed of three or more. Heuristics employed to determine that how many characters make up the component, are not perfect and can lead to misclassification in special cases. So, here that cases would be discussed where these heuristics can involve (i.e. where there are two connected characters) and then extend the heuristics developed for that problem which contains three or more connected characters.

## 2.7 BASIC METHODS

The most simple and straight forward segmentation algorithm is a vertical scan. The algorithm binarizes the image into black and white pixels and simply looks for unbroken columns of white pixels. This work well for machine printed characters or handwritten characters in which a prescribed amount of white space is guaranteed. Unfortunately, normal handwritten characters are often slanted at different angles so vertical scans or scans at any other set angle will fail.

A more robust technique is to isolate regions of connected black pixels. This method separates the black pixels into sets in which each black pixel is adjacent to another black pixel in the set. This method works very well for digits that are not overlapped, touching or disjoint. However, two characters that are overlapped or touching will be treated as a single mass of pixels and a character that is disjoint (that is, not all pixels are adjacent) will be treated as two or more characters. While it is straightforward to combine two disjoint segments, it is a much more difficult problem to separate connected digits. All the algorithms presented in this section use isolation of connected pixels as the first step in processing and then attempt to separate connected digits.

So, the algorithms to be discussed here can be classified under the categories of Hit and Deflect, Min-Max, Drop Fall and Structural Feature based segmentation. Collectively, they constitute the major types of non recognition based segmentation algorithms .Drop fall based algorithms will be studied in further detail in Chapter 3 as they provide the basis for the hybrid drop fall algorithm and Extended drop fall algorithm .

### 2.7.1 Hit and Deflect Algorithms

Hit and Deflect algorithms attempt to find an optimal path for cutting a connected component by literally hitting and deflecting their way through the connected component. [2] define Hit and Deflect algorithms to be those "able to compute a curved segmentation path by iteratively moving a fixed point". When it 'hits' the black pixels in the image it 'deflects' and changes its direction of motion. The algorithm follows a set of rules that maximizes the chances that it will hit and deflect its way to an accurate path. The rules could be designed so as to mimic other segmentation heuristics such as min-max or drop fall (to be mentioned in the next section) methods.

In the earliest examples, when a hit and deflect strategy in to a segmentation system was developed, the system first tried to separate out characters by a performing a vertical scan near the middle of an image. If the scan resulted in zero or two black-to-white or white-to-black transitions, then it was assumed that the characters were either not connected or simply connected (connected at only one point). In this case, the scan line would provide a suitable segmentation path. On the other hand, if scan line crossed more than two transitions, the assumption was made that the characters are slanted. This is when a Hit and Deflect-type algorithm was used. Their algorithm starts from a peak in the lower contour of a connected component. It then attempts to move upwards according to a series of rules. These rules are designed to minimize the number of cuts which have to be made though the component. The algorithm would finish once it reaches the top of the image.

**Figure 2.3: A Hit and Deflect split**

## 2.7.2 Drop Fall Algorithms

Drop fall algorithms attempt to build a segmentation path by mimicking an object such as a marble or a drop of water falling or rolling in between the two characters which make up a connected component (figure 2.4).This class of algorithm was first proposed by [4]. The object falls downward until it gets stuck in a "well", at which point it cuts through the contour and continuous downward .There are four primary types of drop fall algorithm which differ on the direction and the starting point of the drop fall. These are top-left (or left descending), top- right (or right descending), bottom-left (or left ascending) and bottom–right (or right ascending) (figure2.5).Although similar in techniques, each of these variations has their own strong and weak point. Since the implementation of a drop fall algorithm usually involves a set of rules which dictate the next pixel in the path of the fall which are based on the current position, drop fall algorithm could be seen as a specific type of Hit and Deflect algorithm.

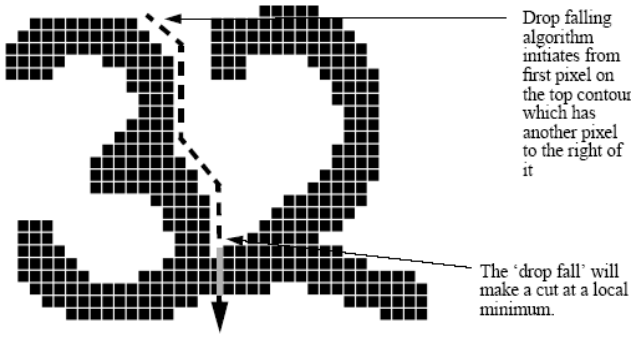


**Figure 2.4: Example of the path of a 'drop-falling' algorithm**

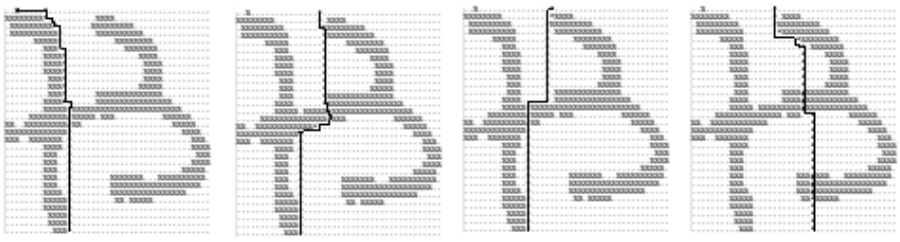

**Figure 2.5: Drop falling algorithm (a) top left (b) bottom left (c) top right (d) bottom right**

## 2.7.3 Structural Feature based Segmentation

Another major type of non recognition based heuristic involves basing the segmentation of a connected component on structural features in the component itself. (that is uses the structures properties of the characters to determine possible

segmentation paths). These algorithms analyze the contour slope or the presence of local maxima and minima to segment the digits. Abrupt changes in contour slope and pairs of relatively nearby maxima and minima often indicate points through which the character should be cut.

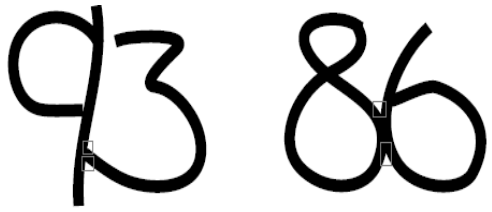

**Figure 2.6: Structural features which could be entry or exit points of intersecting strokes**

The algorithm is designed to find the start and end point of pen strokes. The algorithm seeks terminal points of both normal strokes and strokes that are overlapped or touching other characters (see figure 2.6). This is done by identifying points that are either:

a) corner points
b) local minima
c) local maxima or
d) a potential exit point of a straight stroke created by extending a contour through a concave corner ( figure 2.7).

These points are referred to as a Significant Contour Points. The algorithm selects a pair of points from the set of SCPs which could be the entry and exit points and cuts the segment along the line joining two points. Segments are selected using the criteria listed below:

- Whether one is a minima and the other is a maxima
- Corner points are preferred
- The closer the points are to each other the better
- The sharpness of the concavity at SCPs
- The degree to which a minima is directly above a maxima
- The distance of a minima/maxima to the left side of the image

**Figure 2.7: Significant contour points (SCPs)**

### 2.7.4 Min-Max Algorithms

Min-Max algorithms are some of the most common employed method in segmentation systems and can be considered a subclass of structural feature -based methods. They are bases on the premise that most touching or overlapped characters are joined where the distance from upper contour to the lower contour of the connected component is a minimum. The algorithm finds this minimum by examining the extrema on the upper and lower contours. In particular, it identifies the minima in the upper contour and the maxima in the lower contour. Then, a straight line path is picked between upper-contour minima and lower contour maxima. The decision as to which minima and maxima to connect is usually based on values such as the distance of the cut from the center of the image ,the distance between the extrema, and the number of white-black or black-white transitions encountered along the cut.

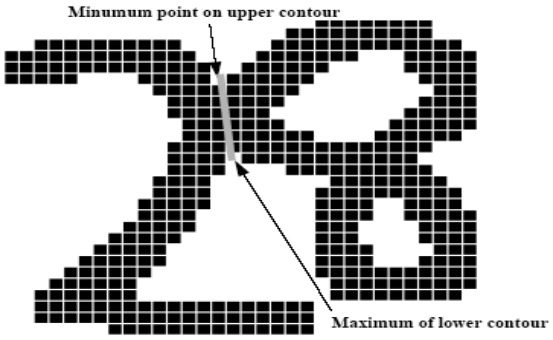


**Figure 2.8: Example of a cut produced by a min-max based heuristic**

## 2.8 The need for developing more sophisticated Segmentation methods

The heuristics described in this chapter work reasonably well for a large number of cases, but they also fail badly for a lot of other cases (Figure 2.9). Also, the 'global' approaches such as structural feature-based segmentation tend to be extremely computationally expensive as opposed to the more local "Hit and Deflect" and drop

fall approaches. Because of this, it is useful to explore ways of modifying and combining these heuristics so as to produce better results more often. The hybrid algorithm developed is the subject of the next chapter.

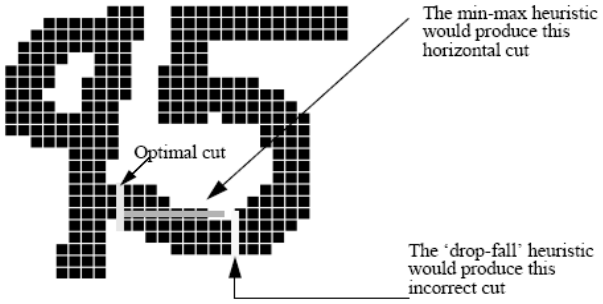


**Figure 2.9: Both 'drop-fall' and min-max heuristics fail on the above example**

# CHAPTER 3

# THE HYBRID DROP FALLING TECHNIQUE

# THE HYBRID DROP FALLING TECHNIQUE

## 3.0 INTRODUCTION

Drop falling algorithm is based on the principle of letting a hypothetical marble fall in between two connected characters. The marble starts falling from the top of the first character and make fairly optimal cut. This algorithm is a simple algorithm that offers remarkably good results. The hybrid Drop fall concept augments the algorithm by attempting Drop Fall in four different orientations. The Extended Drop Fall algorithm developed in this chapter is also based on the Hybrid Drop Fall algorithm and operates in essentially the same way. So, a detailed explanation of Drop Fall and Hybrid Fall is necessary to explain the differences in Extended Drop Fall algorithm.

## 3.1 THE TRADITIONAL DROP -FALLING HEURISTIC

As mentioned above, the drop falling heuristic is based on the principle that a fairly optimal cut between two connected characters can be made if one were to role a hypothetical marble off the top of the first character and make the cut where the marble falls. Despite its apparent simplicity, the algorithm has proven itself to be quite useful.

Based on this simple description of the method, the main issue comes out in its implementation is from where to drop the marble because if the algorithm starts in the wrong place ,the marble could easily roll down the left side of the first digit or second digit and thus ,would be completely ineffective. So, the first and the most necessary step is to decide from where to drop the marble.

### 3.1.1 Starting point for the Construction of Path

One of the most important aspects of Drop Fall is the point at which the algorithm begins to construct a path. The starting point should be above the image and toward the left side because the simulated marble falls downward and to the right. If the starting point is selected too far to the left, the marble can slide off the left side of the first character and produce no segmentation. Similarly if the starting point is chosen too far to the right, the marble can slide off the right side of the second character. (see figure 3.1)

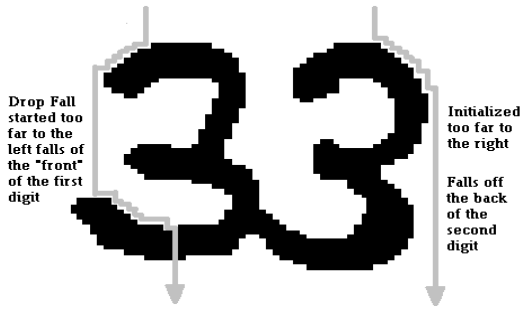


**Figure 3.1: Bad starting points for Drop Fall Segmentation**

The staring point can be any position between the connected characters and above the connection (area to be cut). It will be the best to start as close as possible to the point where two characters are connected. Further more, an appropriate starting point should have black pixels on either side of it. The system scans the image row by row starting at the top, until a back boundary pixel with another black boundary black pixel to the right of it is detected, where the two pixels are separated by only white space. This pixel is then used as the point from which to start the drop –fall. (see figure 3.2).

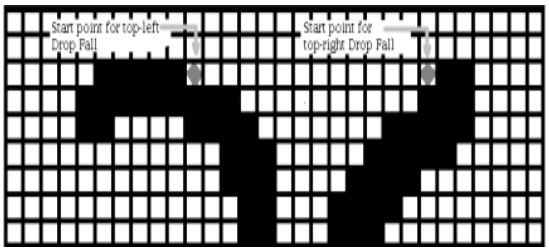


**Figure 3.2: Selecting a good point for Drop Fall Segmentation**

## 3.1.2 Movement Rules

When an appropriate point start point is selected, then the algorithm "drops the marble". The algorithm follows a set of rules that advances the marble one pixel at a time until the bottom of the image is reached. The algorithm considers only five adjacent pixels: the three pixels below the current pixel and the pixels to the left and right. Upward moves are not considered, because the rules are meant to mimic a falling motion. The marble will not bounce back. Characters are treated as solid objects, the marble will not cut through the boundaries (i.e., enter a black pixel) unless all other options are precluded. The movement rules are depicted graphically in figure 3.3.

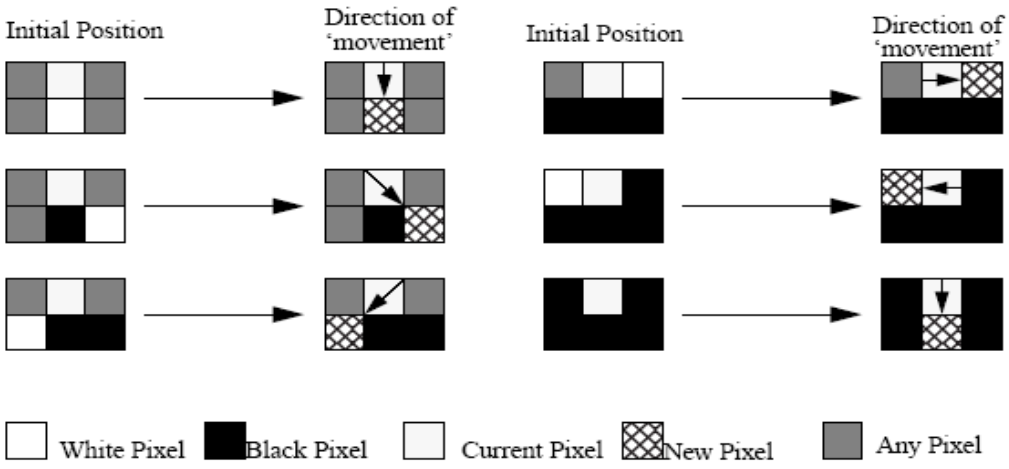


**Figure 3.3: stepwise movement of the Drop Fall Algorithm**

The marble will always move directly downward if that pixel is empty (white). If the pixel is occupied then the marble will move down and to the right. If both the pixels directly downward and down and right are occupied the marble will fall down and to the left. If all three pixels at the lower level are occupied the marble move directly to the right. If all pixels except for the one directly left of the marble are occupied then the marble will move directly left. Finally if all pixels around the current point are occupied the marble will do move downward, thereby cutting the contour.

**Pseudo code for Movement Rules**

IF (WHITE (down))

THEN {GO (downright)}

IF (WHITE (downLeft))

THEN {GO (downLeft)}

IF (WHITE (right))

THEN {GO (right)}

IF (WHITE (left))

THEN {GO (left)}

ELSE GO (down);

There is an additional restriction that the marble cannot move back to a position it has already occupied. If the above rules specify a location that has already occupied, the marble will cut downward instead. This prevents the algorithm from getting stuck in a cycle when it reaches a flat plane.

## 3.2 VARIATIONS OF DROP FALLING

The standard version of the drop fall algorithm described above falls down and to the right from the top left of the connected component. Three more variations are possible and all four are used in the Hybrid Drop Fall algorithm. Drop fall with other three variants is similar except that they will not start near the top left. Two of the variants start at the bottom of the image and fall upward.

**TOP-RIGHT DROP FALL:** This algorithm is identical to the top-left orientation (standard drop fall) except that it initiates from the top of the image toward the right side (i.e., top right of the connected component rather than the top left (3.4b)). The marble still falls downward but favors the left side instead of the right. The standard drop fall can be used if the input image is 'flipped' vertically (3.4a). The resulting segmentation path P can be obtained through the transformation of equation (1) where Pinvx is a vector of the x co ordinates of the segmentation path resulting from the standard drop fall algorithm being performed on the vertically inverted images and w is the width of the image.

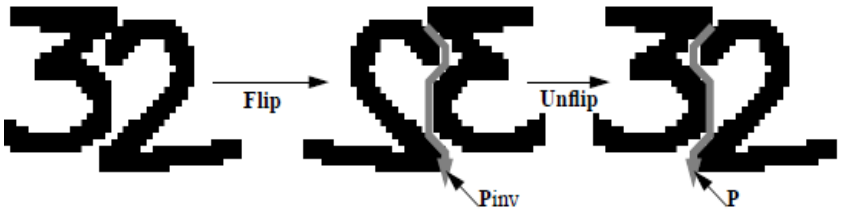


**Figure 3.4: (a) standard drop fall to an inverted image (b) top-right drop fall**

$$\mathbf{P}_x^i = w - \mathbf{P}\text{inv}_x^i$$

For i=1, 2... n where n is the length of **P**inv (1)

$$\mathbf{P}_y^i = \mathbf{P}\text{inv}_y^i$$

**BOTTOM-LEFT DROP FALL:** This algorithm is identical in principle to the original drop-fall algorithm except that it starts from a pixel at the bottom left of the image and falls up and to the right (figure 3.5b). The standard drop fall algorithm can be used here if the input image is flipped horizontally (figure 3.5b). The bottom left drop fall path can then be obtained using equation (2) where h is the height of the image.

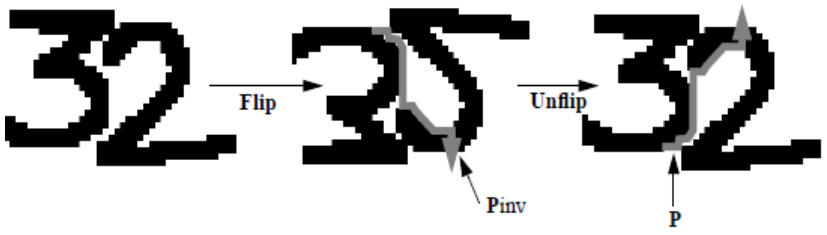


**Figure 3.5: a) standard drop fall to an inverted image b) bottom left drop fall**

$$\mathbf{P}_x^i = \mathbf{P}\text{inv}_x^i$$

For i=1, 2,…. n where n is the length of **P**inv (2)

$$\mathbf{P}_y^i = h - \mathbf{P}\text{inv}_y^i$$

**BOTTOM-RIGHT DROP FALL:** This heuristic is identical to the previous three variations except that in this case, it starts at the bottom –right corner. Like Bottom – left drop fall this algorithm falls upwards and then leftward direction. It can be viewed as the exact opposite of the standard top-left drop fall. As before, a bottom drop fall (figure 3.6b) can be implemented by performing a standard top left fall after appropriately transforming the input image(figure 3.6b) This time, the image must be flipped in both the horizontal and vertical directions. The bottom right path P can be obtained from the transformation of Pinv given in equation (3).


**Figure 3.6: a) standard drop fall to an inverted image b) bottom-right drop fall**

## 3.3 EVALUATION OF THE VARIATIONS OF DROP FALLING

Despite the apparent similarity between the variations of the drop fall heuristics discussed above, they do very often provide very different segmentation paths for various examples. It is easy to find the cases where these variations of the drop fall work well and also where they don't work well. It is much more difficult to find the cases where they all of them don't work well. This principle will provide the basis for the construction of a hybrid heuristic which makes use of all of the aforementioned drop fall variations. Now, we will look at the kinds of examples where each of the drop falls tends to succeed or fail. Standard drop fall seems to be very robust method

for segmenting two connected characters (figure4.7) ,the top left drop fall tends to work well in cases when the point at which two characters are joined are a minimum point.

**Figure 3.7: Examples for which the top left drop fall heuristic succeeds**

But when the point at which two characters are joined are not a minimum point then the top left drop fall heuristic will fail**.** Figure 3.8 shows that this heuristic will usually fail when a minimum point does not occur at the point of connection or overlap of the two characters (as is the case with the '95' and '73') or the point of initiation (where the drop fall starts from) is not close enough to the point where the two characters touch (for example, the '46' in figure 3.8).

**Figure 3.8 Examples for which the top left drop fall segmentation heuristic fails**

The second of the two failures can be corrected by formulating more advanced heuristics for deciding where to initiate the drop fall from. It is much harder to make corrections for failures of the first type which tends to happen anytime when the second character is a 3 or 5 since both of these numbers have a minimum on their lower strokes which are accessible to the drop fall algorithm.

The top right fall tends to succeed and fail on the same cases as the standard top left algorithm. But also some special cases are there where one of them will succeed and other will fails. But in general, both are equivalent in terms of effectiveness.

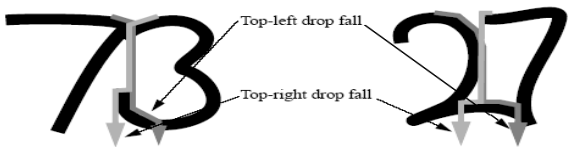


**Figure 3.9: Examples of when top left and top right drop fall differ in results**

Bottom left and Bottom right tend to succeed in most of the cases where the top based drop falls fail (see figure 3.8), also both of them fail on some cases where top based drop falls succeed. So ,the top based and bottom based drop falls complement each other, then it is clear that a heuristic which makes use of both of them could potentially be significantly better than any of the drop fall heuristic.

## 3.4 HYBRID DROP FALL HEURISTIC

Based on the evaluation of the four variations of the drop fall, it is clear that each of the four variations of the drop fall have advantages over another. In particular, top based and bottom based drop falls can each successfully segment cases that other can't. Keep this in mind, there is need to design a hybrid drop fall heuristic which combines the best of both types of elements.

The algorithm will initially attempt a top left drop fall on the image, and then it will attempt to determine if the algorithm performed a successful segmentation. To cover as many cases as possible, the hybrid algorithm needs to be able to recognize when one variety of the drop fall fails in order to know when to use one of the other variations. Here a character recognizer could be used at this point, to determine if each of the resulting character are legible numeral. This however would be computationally expensive and would not necessarily even give a good indication of the quality of the segmentation. As see in figure 3.9 and figure 3.10 the top left drop fall tends to fail when the second character has a local minimum in its lower stroke.

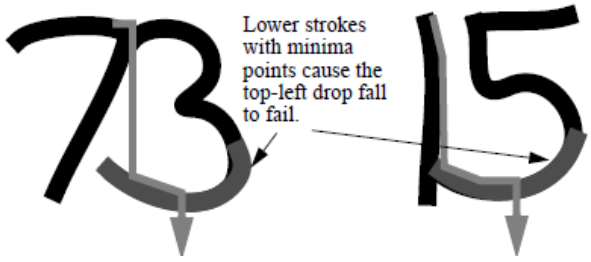


**Figure 3.10: lower strokes on characters like '3' and '5' cause the top-left drop-fall algorithm to fail.**

This is true for all threes and fives. When it is known priory whether the second digit is 3 or 5, so some methods will have to be used ascertain whether the character is 3 or 5. So for recognizing the second digit, one must look at the minima of lower strokes of characters versus minima at the junction point between two characters. There will be one fundamental difference between minima of 3 or 5 than others. In case of 3 or

5, the minima at a lower stroke of three or five will be very gradual and flat, while for others, the minima at the junction point between two characters will be very sharp and steep. (see figure 3.11)

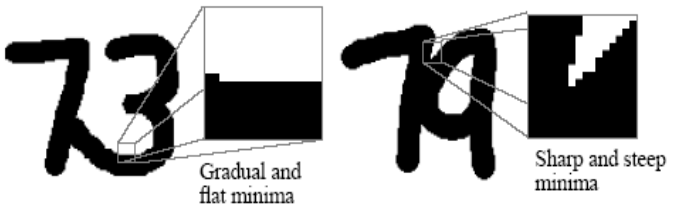


**Figure 3.11: Differences between minima on the lower stroke of characters and minima at the junction of two characters**

Based on this observation, all that the algorithm would need to do is recognize whether the minima at which the top left drop fall begins to cut into the characters is sharp and steep or flat and gradual. If it is sharp and steep, then the top left drop fall is most probably sufficient. On the other hand, if the minima is flat, then the top left drop fall probably failed by cutting through the lower stroke of a three or five. So in this case, bottom right drop fall will be applied, because the bottom right drop fall usually properly segments cases where a top left drop fall would fail. A best method for recognizing steep versus maxima would be to travel along the contour of the characters a given number of pixels to the left and to the right. If the contour rises in either direction more than a certain threshold then it would be classified as being steep otherwise it would be flat. (figure 3.12). Between 3 and 5 pixels should be travelled in each direction along the contour, depending on the size of the image. The threshold should be half the number of pixels travelled. Based on this, if the slope in either direction is greater than ½, then it is steep. Otherwise, it is flat.

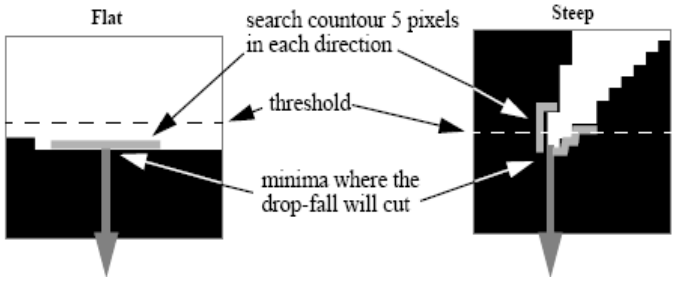


**Figure 3.12a: Using the slope of the neighboring contour to determine if the minima is at a 'steep' or a 'flat' point**

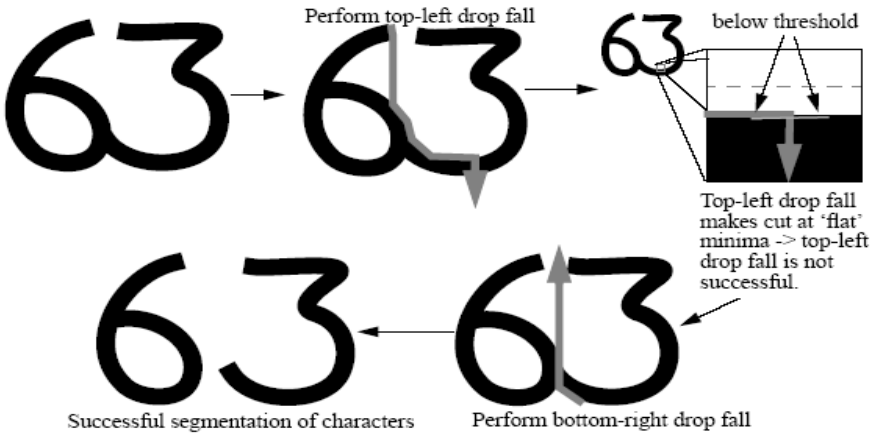


**Figure 3.12b: Performing bottom-right drop fall when top-left drop-fall is recognized to fail**

## 3.5 PROBLEMS IN SEGMENTING CONNECTED ZEROS BY DROP FALL ALGORITHMS

This approach causes error in certain cases such as connected zeros. The two zeros are frequently connected or overlapping. Infact, 80-85% of successive zeros are connected. Further more 45- 55% of all connected digits consists of a pair of zeros. When system used as Hybrid Drop Fall algorithm to perform segmentation, although relatively the system was accurate in many cases but it showed relatively poor performance on connected zeros. The Hybrid Drop Fall was able to perform segmentation for 25% of the connected zeros in test cases presented by [4].Because in case of connected zeros, a large number of overlapping pixels are there and a long cut is required in order to separate the characters. But HDF give the short cuts, so it fails to segment the connected zeros correctly.

Connected zeros are problematic because of the structural properties of the joint between the digits. The joint between connected zeros tends to be longer than the width of a pen stroke and slanted at an angle. For most other cases the joint is no longer than the width of a stroke and can be separated by cutting down a straight vertical line at the joint. When rules of Drop fall is applying for segmenting the connected zeros, these rules tend to cut the connected zeros in to the loop of one of the Zeros. Once the path cuts into the loop, it will also cut out leaving an open arc on one side of the path, and a digit with a small piece of the broken zero on the other. This is a particularly distressing problem because it leads to false recognition. An open arc on the right hand side can look very much like a '7' and in some cases it can

look like ‘2’. Similarly an open fragment on the left side can look like a ‘6’. Because of this error a frequently occurring string ‘00’ can be mistaken for ‘07’,’02’,or ‘06’.

This type of problem is not unique to zeros; a long junction near a loop can cause problems for other looping digits. Similar problems can occur with digits such as ‘9’,’8’,’6’ and ‘2’depending on the writer and the nearby digits. In order to avoid these problems, the algorithm needs to make a long cut through a diagonal joint. So to overcome these problems, the system has been modified to include a new segmentation algorithm, referred to as Extended Drop Fall. This algorithm is also a variant of Drop Fall and is based on the Hybrid Drop Fall segmentation presented by [5]**.** However, in the general case, a long cut through a segment is a bad one.

## 3.6 EXTENDED DROP FALL ALGORITHM

In order to avoid such problems, segmentation paths need to cut along the junction and not enter the loop. Drop fall move downward on a binarized image by seeking white pixels in order to avoid the contour. So, when a cut is made in to a black pixel, the algorithm continues to seek white and minimizes the length of a cut. But for connected zeros and for other cases, the cut should be extended. So, Extended Drop Fall will seek black pixels rather than white once a cut has been initiated. (i.e., it is currently on a black pixel). This results in longer cuts that follow the interior of the contour and are more likely to be correct for connected zeros.

In order to use the Extended Drop fall algorithm effectively, firstly that cases must be identified in which it is appropriate. In the general case, the EDF algorithm produces a path that is less likely to be correct one because this algorithm is useful only for special cases where other Drop fall algorithms tend to fail. The system needs to be able to choose between different paths generated using different segmentation algorithms. The HDF algorithm generates a set of paths, using standard drop fall from four different orientations. The best segmentation path is selected based on a heuristic analysis of each path. The EDF algorithm also generates multiple paths using the four different orientations of the DF algorithm, but it cannot rely on heuristic analysis to select the correct path because the algorithm is designed for exceptional cases.

### 3.6.1 Movement Rules for Extended Drop Fall

In order to correctly segment connected zeros and other problematic cases, the Extended Drop fall approach will extend cuts diagonally through a contour. The algorithm considers the pixel it is on, as well as the five pixels surrounding it from the sides and below. If the current pixel is white, the algorithm follows the movement rules specified in Hybrid Drop Fall. However, if the current pixel is black it follows the rules described below:

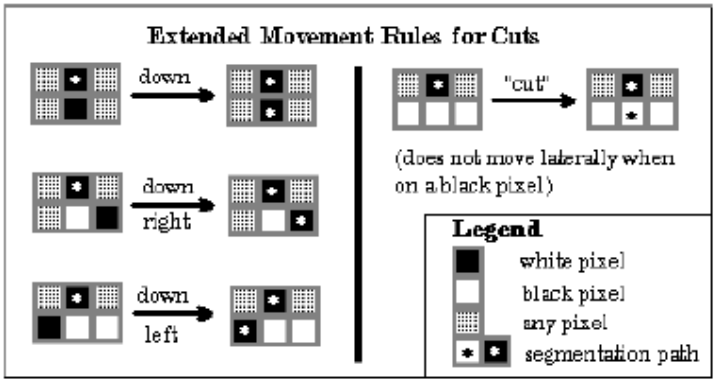


**Figure 3.13: Movement rules for Extended Drop Fall**

**Pseudo code for Extended Drop Fall Movement Rules:**

```
IF (ON WHITE) THEN
{
    IF (WHITE (down))          THEN  {GO (down)       }
    IF (WHITE (downRight))     THEN {GO (downRight)}
    IF (WHITE (downLeft))      THEN {GO (downLeft)   }
    IF (WHITE (right))         THEN {GO (right)      }
    IF (WHITE (left))          THEN {GO (left)       }
    ELSE                                {GO (down)     }
}
ELSE
{
    IF (BLACK (down))          THEN {GO (down)       }
    IF (BLACK (downRight))     THEN {GO (downRight)}
```

```
        IF (BLACK (downLeft))      THEN {GO (downLeft)}
        ELSE                                {GO (down)      }
    }
```

The standard algorithm attempts to move the marble onto white pixels at all times.EDF will seek black pixels when it is already on a black pixel. When these rules are in effect, the "marble" will fall directly downward if there is a black pixel below. It will fall down and to the left or right if there is no black pixel directly below (preference to the right.) However, the pixel will always move downward; unlike the "white seeking" rules that apply when the marble is on white, the "black seeking" rules do not move directly left or right. The intent is to extend a diagonal cut, not to follow the contour. If all three pixel below the current pixel are white, then the marble moves directly down and terminates the cut.

**(a)** **(b)**

**Figure 3.14: Segmenting double zeros (a) best results with Hybrid Drop Fall (b) Extended Drop Fall**

Figure 3.14 illustrates the difference between the two algorithms. Figure 3.14a shows the best result using hybrid drop fall (top-right orientation in this case.) The other three orientations produced similar results. The Extend Drop Fall approach correctly segments the digits by following the diagonal junction all the way through (depicted in Figure 3.14b).

# CONCLUSION

Segmentation is the most important stage in OCR system because recognition accuracy of characters depends on it. Segmentation of character string is relatively simple concept, we have seen that it is extremely difficult task and far is from being perfected. Segmentation of offline handwritten characters is much harder problem than machine printed or well written. The existing algorithms which we have studied in this work made for the purpose of improving the performance of segmentation systems which has the broader impact of improving the quality of automatic handwriting recognition.

Handwritten character recognition, and segmentation in particular, is a process that we humans take for granted. When a human reads a character string, in human mind, the segmentation and recognition are performed simultaneously and in conjunction with each other. A human is able to use knowledge of known words and the context of the writing to further recognize written text. So, keep these factors in mind, there is a need to build a technology which can model the human mind and gives 100% accurate results. Since we currently no where near that point, the best we can currently do is study the patterns and features of handwriting and attempt to develop segmentation heuristics.

In this work we have focused on the segmentation system of offline handwritten numeric digits. As we know, numeric digits are usually used as writing zip code, postal address, bank check amount etc. Segmentation is easy for the cases when number is well written, problems occur on those cases where touching, overlapping like cases occurs. So, segmentation becomes difficult for these types of cases. The segmentation process described in this consists of three steps:

1) Identification of connected component
2) Classification of connected component
3) Finally, splitting the touching character

So, first step (discussed in second chapter) is easy when the digits in the numeral image are well separated, so connected components can be extracted easily, but difficult for the cases when any of the digits are connected or disjointed. But classification of connected component and splitting touching character is not easy things to do. The width to height ratio is used for the classification of connected

components. This method is much more useful for identifying whether or not the component is a separated segment of a large character. But this method fails on cases like 2, 3, 5, or 7 which are written unusually wide by others. Also when 1 and 7 are connected, then its width to height ratio is identical to 7 alone. So, making an entirely new, more robust heuristic for this purpose would do a lot in terms of improving current segmentation systems. The last step in the segmentation process, namely splitting touching characters has been worked on by many researchers, but it too has not been perfected. We have mentioned the current methods (min-max, hit and deflect, and drop fall) in chapter 2. These methods have tried to make use of knowledge of the point where the two characters potentially touch in order to make good 'splitting' decision. As these heuristics described, work reasonably well for a large number of cases, but they also fail badly for lots of cases. Also the global approaches such as structural feature based segmentation tend to be extremely computationally expensive as opposed to the more local Hit and Deflect and drop fall approaches. In case of touching five with nine, Min max gives the horizontal cut where as drop fall gives vertical cut but both are splitting the digit '5' in spite of splitting the connected part between them. They were not giving the optimal cut. We have also discussed all the variations which are based on drop fall. Each variation has advantages over another. Also these variations succeed and fail on many cases which we have already been mentioned in chapter 3. To overcome these failures, a Hybrid drop fall algorithm has been introduced, in which both top based and bottom based drop fall has successfully segmented the cases. Hybrid algorithm gives short cuts on segmenting the touching characters. This algorithm has promising results. But this creates problem in case of connected zeros, because it requires long cut for segmentation. This problem has been discussed in chapter 3. So**,** for solving this type of problem, a new algorithm Extended drop fall has been studied, which is also the variant of Drop fall and is based on the Hybrid Drop fall segmentation algorithm. This algorithm works well for case in which the correct path should make a long cut through the connected characters. It works particularly well for connected zeros and for others problematic cases which occurs frequently and are routinely separated incorrectly. So, these hybrid techniques have proved almost successful on all the cases. Although, reading of all types of handwritten material with this system is still far from perfect, but one has come closer to the goal of imitating human ability to read characters!

# REFERENCES


[1] N.Arica and F.T. Yarman-Vural (2001) "An Overview of Character Recognition Focussed on Offline Handwriting" *IEEE Transaction on system ,Man , and Cybernetics-Part C ;Applications and Reviews* 3(2),216-233 .

[2] R.G.Casey, E.Lecolinet (1996) "A Survey of Methods and Strategies in Character Segmentation" *IEEE transactions on pattern analysis and machine* intelligence, 18(7), pp.690

[3] G.Congedo, G.Dimauro, S.Impedovo, G.Pirlo (1995) "Segmentation of Numeric Strings". *Proceedings of the 3rd International Conference on Document Analysis and Recognition, Vol. 2 pp.1038-1041.*

[4] J.Punnose (1999) "An Improved Segmentation Module for Identification of Handwritten Numerals" Master Thesis, Massachusetts Institute of Technology.

[5] S.Khan (1998) "Character Segmentation Heuristics for Check Amount Verification" Master Thesis, Massachusetts Institute of Technology.

[6] U.Pal, A.Belaid and Ch. Choisy (2003) "Touching Numeral Segmentation using Water Reservoir Concept". *Pattern Recognition Lett. 24, pp.261-*272

[7] X.Wang, K.Zheng and J.Guo (2006) "Inertial and Big Drop Fall Algorithm" *International Journal of Information Technology, vol.12, No.4.*

[8] L.S. Oliveria, E.Lethelier, F.Bortolozzi, and R.Sabourin (2000)"A New Approach to Segment Handwritten Digits" *In proc. of the Int. Workshop on Frontiers in Handwriting Recognition, pp. 577-582*

[9] Y.K.Chen and J.F. Wang (2000) "Segmentation of Handwritten Numeral String using background and foreground analysis". *Proc. 15th ICPR, pp.598-601.*

[10] B.Verma (2003) "A Contour Code Feature based Segmentation for Handwriting Recognition". Proc. 7th ICDAR.

[11] D.Das and R.Yasmin (2006) "Segmentation and Recognition of Unconstrained Bangla Handwritten Numerals". *Asian Journal of Information Technology 5(2), pp.155-159.*

[12] R.Palacios, P.Wang and A.Gupta (2004) "Handwritten Bank Check Recognition of Courtesy Amounts". *International Journal of Image and Graphics, Vol.4, No.2.*

[13] V.K. Madasu, M.H.M. Yusof, M.Hanmandlu and K.Kubik (2003) "Automatic Extraction of Signatures from Bank Checks and other documents". *Proceedings of DICTA, Vol.2, pp.591-600.*

[14] Z.Lu, Z.Chi, W.Siu and P.Shi (1999) "A background thinning based approach for Separating Connected Handwritten Digit Strings". *Pattern Recognition, Vol.32, pp.921-933.*

[15] U.Pal and S.Datta (2003) "Segmentation of Bangla Unconstrained Handwritten Text". *Proc. 7th ICDAR, pp.1128-1132.*

[16] S. Ouchtati, B.Mouldi and A.Lachouri "Segmentation and Recognition of Handwritten Numeric Chains". *International Journal of Information Technology, Vol 4, No.1*

[17] R.Palacios and A.Gupta (2002) "A System for processing handwritten bank checks automatically".