

USE OF ZERNIKE MOMENTS
IN
HANDWRITTEN CHARACTER RECOGNITION

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering

in

Computer Science & Engineering



Submitted by

Harikesh Singh

(8053208)

Under the supervision of

Dr. R. K. Sharma

Dean, Academic Affairs

and

Mr. Karun Verma

Lecturer, CSED

JUNE 2007

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY, PATIALA-147004, PUNJAB (INDIA)

Certificate

I hereby certify that the work which is being presented in the thesis entitled, “**Use of Zernike Moments in Handwritten Character Recognition**”, in partial fulfillment of the requirements for the award of degree of **Master of Engineering in Computer Science and Engineering** submitted in **Computer Science and Engineering Department** of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Dr. R. K. Sharma and Mr. Karun Verma.

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

(Harikesh Singh)
8053208

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

(Mr. Karun Verma)
Co-supervisor, Lecturer
Computer Science & Engineering Department
Thapar University, Patiala-147004

(Dr. R. K. Sharma)
Supervisor, Dean, Academic Affairs
Thapar University, Patiala-147004

Countersigned by:

(Dr. (Mrs.) Seema Bawa)
Professor and Head,
Computer Science and Engineering Department,
Thapar University, Patiala-147004, Punjab (India)

(Dr. R. K. Sharma)
Dean, Academic Affairs
Thapar University, Patiala-147004

Acknowledgement

First of all, I would like to express my deep gratitude to my supervisor Dr. R. K. Sharma, Dean, Academic Affairs, who really give me a chance to explore my skills in the field of Pattern Recognition and devoted a lot of time in exploring the terms used in my thesis work. He gave suggestions always one step ahead to me so that my research was always in progressive mode.

No words in owe to my co-supervisor Mr. Karun Verma, Lecturer, Computer Science & Engineering Department, who always supervised me in the exact way from valuable corrections and implementations of my thesis work. He always spends sufficient time to understand my problem from their busy schedules.

I am also thankful to Mrs. Shivani Goyal, P.G.Coordinator, Computer Science & Engineering Department, who always cares to my problems related to my thesis work. She always considered problem and take immediate action in the benefits of students.

I am also thankful to Dr. (Mrs.) Seema Bawa, H.O.D., Computer Science & Engineering Department, who always considered my problem and provided me each and every facilities related to my thesis work.

Also, I would like to thank the members of the Computer Science & Engineering Department for their encouragement and support during the preparation of this thesis report.

Most importantly, I would like to thank my parents, my wife and the almighty for showing me the right direction out of the blue, to help me stay calm in oddest of the times and keep moving even at times when there was no hope.

Abstract

There has been a significant amount of research in pattern recognition in different aspects of handwriting based user interfaces including interactive design tools, ink beautification, and handwritten character recognition. In this thesis, we have focused on the recognition of handwritten Hindi numerals that can be used in common applications like type checking, digital signatures, online document recognition and digital library. Challenges in handwritten characters recognition lie in the variation and distortion of online handwritten numerals since different people may use different style of handwriting, and direction to draw the same shape of any numerals.

Handwritten Hindi numerals are imprecise in nature as their corners are not always sharp, lines are not perfectly straight, and curves are not necessarily smooth, unlikely the printed numerals. Furthermore, Hindi numerals can be drawn in different sizes and orientation (the orientation of an arrow depends on its pointing direction), in contrast to handwriting which is often assumed to be written on a baseline in an upright position. Therefore, a robust online handwritten recognition system has to account for all of these factors.

In this thesis work, we are considering a statistical approach to Online Handwritten Character Recognition using Zernike moments. We are using Zernike moments as features in the reconstruction of Hindi numerals with its invariance property. Zernike moments have been used in the Optical Character Recognition and Image Recognition applications with good results, but, this feature has not been explored for use in Online Handwritten Character Recognition especially for Hindi numerals recognition.

Acronyms

Int.	International
DPI	Dots Per Inch
2D	2 Dimensional
ZM	Zernike Moment
PC	Personal Computer
PDA	Personal Digital Assistant
OCR	Optical Character Recognition
MCR	Magnetic Character Recognition
HCR	Handwritten Character Recognition
OHCR	Online Handwritten Character Recognition

Table of Contents

Certificate	i
Acknowledgement	ii
Abstract	iii
Acronyms	iv
Table of Contents	v
List of Figures	vii
List of Tables	viii
Chapter 1 Introduction	1
1.1 Definitions	2
1.2 Motivation	3
1.3 Research Scope	4
1.4 Applications	4
1.5 Thesis Contents	5
Chapter 2 Background Information	6
2.1 Character Recognition	6
2.2 Handwritten Character Recognition	8
2.3 Image Features	9
2.4 Digital Image Representation	10
2.4.1 Image Acquisition	12
2.4.2 Preprocessing	12
2.4.3 Segmentation	12
2.4.4 Representation and Description	12
2.4.5 Recognition and Interpretation	13
2.5 Image Resolutions	13
2.6 Architecture of the OHCR System	14
2.7 Summary	16

Chapter 3 State of the Art in Recognition System	17
3.1 Introduction	17
3.2 Boundary-Based Invariants	18
3.3 Region-Based Invariants	18
3.3.1 Regular moments and Central moments	19
3.3.2 Zernike moments	20
3.4 Algorithm's complexity for moment computation	22
3.5 Summary	22
Chapter 4 Problem Statement	23
4.1 Introduction	23
4.2 Development of an OHCR system	24
4.2.1 Online Handwritten Image Samples	24
4.2.2 How to compute Zernike moments	29
4.2.3 Image Reconstruction using Zernike moments	32
4.2.4 Accuracy of ZM in reconstruction of image	34
4.3 Summary	34
Chapter 5 Research Problem Solving	35
5.1 Computations of ZM and accuracy of reconstructed image	35
5.2 Summary	73
Chapter 6 Conclusions and Future Research	74
6.1 Conclusions	74
6.2 Future Research	75
References	76
Appendices A	80
Appendices B	86

List of Figures

Fig 1.1: Samples of handwritten Hindi numerals drawn by different user	2
Fig 2.1: Different classification of character recognition	6
Fig 2.2: The basic processes of an HCR system	9
Fig 2.3(a): A digital image $f(x, y)$ in x - y coordinate	11
Fig 2.3(b): Fundamental steps in digital image processing	11
Fig 2.4: Image recognition by Feature Extraction Process	15
Fig 2.5: The basic process of an OHCR system	16
Fig 3.1: Discrete pixel sampling in radial-polar form	20
Fig 4.1: Sample of Hindi numeral ० drawing by online handwriting	24
Fig 4.2: Sample of Hindi numeral १ drawing by online handwriting	25
Fig 4.3: Sample of Hindi numeral २ drawing by online handwriting	25
Fig 4.4: Sample of Hindi numeral ३ drawing by online handwriting	25
Fig 4.5: Sample of Hindi numeral ४ drawing by online handwriting	26
Fig 4.6: Sample of Hindi numeral ५ drawing by online handwriting	26
Fig 4.7: Sample of Hindi numeral ६ drawing by online handwriting	27
Fig 4.8: Sample of Hindi numeral ७ drawing by online handwriting	27
Fig 4.9: Sample of Hindi numeral ८ drawing by online handwriting	28
Fig 4.10: Sample of Hindi numeral ९ drawing by online handwriting	28
Fig 4.11: Block diagram of computing Zernike moments	29
Fig 4.12: The computation process of the unit disk mapping for ZM	30
Fig 4.13: (a) $N \times N$ pixels image bitmap	30
Fig 4.13: (b) Unit circles mapped onto $N \times N$ pixels size image	30
Fig 4.14: (a) Original image	32
Fig 4.14: (b) Reconstruction order 23	32
Fig 4.14: (c) Reconstruction order 25	32

List of Tables

TABLE 5.1: The accuracy of reconstructed Hindi numeral image “०”	35
TABLE 5.2: The accuracy of reconstructed Hindi numeral image “१”	36
TABLE 5.3: The accuracy of reconstructed Hindi numeral image “२”	37
TABLE 5.4: The accuracy of reconstructed Hindi numeral image “३”	38
TABLE 5.5: The accuracy of reconstructed Hindi numeral image “४”	39
TABLE 5.6: The accuracy of reconstructed Hindi numeral image “५”	40
TABLE 5.7: The accuracy of reconstructed Hindi numeral image “६”	41
TABLE 5.8: The accuracy of reconstructed Hindi numeral image “७”	42
TABLE 5.9: The accuracy of reconstructed Hindi numeral image “८”	43
TABLE 5.10: The accuracy of reconstructed Hindi numeral image “९”	44
TABLE 5.11: For $n = 23, m = 7$, testing accuracy of reconstructed “०”	45
TABLE 5.12: For $n = 23, m = 7$, testing accuracy of reconstructed “१”	46
TABLE 5.13: For $n = 23, m = 7$, testing accuracy of reconstructed “२”	46
TABLE 5.14: For $n = 23, m = 7$, testing accuracy of reconstructed “३”	47
TABLE 5.15: For $n = 23, m = 7$, testing accuracy of reconstructed “४”	47
TABLE 5.16: For $n = 23, m = 7$, testing accuracy of reconstructed “५”	48
TABLE 5.17: For $n = 23, m = 7$, testing accuracy of reconstructed “६”	48
TABLE 5.18: For $n = 23, m = 7$, testing accuracy of reconstructed “७”	49
TABLE 5.19: For $n = 23, m = 7$, testing accuracy of reconstructed “८”	49
TABLE 5.20: For $n = 23, m = 7$, testing accuracy of reconstructed “९”	50
TABLE 5.21: For $n = 24, m = 10$, testing accuracy of reconstructed “०”	50
TABLE 5.22: For $n = 24, m = 10$, testing accuracy of reconstructed “१”	51
TABLE 5.23: For $n = 24, m = 10$, testing accuracy of reconstructed “२”	51
TABLE 5.24: For $n = 24, m = 10$, testing accuracy of reconstructed “३”	52
TABLE 5.25: For $n = 24, m = 10$, testing accuracy of reconstructed “४”	52
TABLE 5.26: For $n = 24, m = 10$, testing accuracy of reconstructed “५”	53
TABLE 5.27: For $n = 24, m = 10$, testing accuracy of reconstructed “६”	53

TABLE 5.28: For $n = 24, m = 10$, testing accuracy of reconstructed “ \mathcal{G} ”	54
TABLE 5.29: For $n = 24, m = 10$, testing accuracy of reconstructed “ \mathcal{C} ”	54
TABLE 5.30: For $n = 24, m = 10$, testing accuracy of reconstructed “ \mathcal{E} ”	55
TABLE 5.31: For $n = 25, m = 9$, testing accuracy of reconstructed “ \mathcal{O} ”	55
TABLE 5.32: For $n = 25, m = 9$, testing accuracy of reconstructed “ \mathcal{H} ”	56
TABLE 5.33: For $n = 25, m = 9$, testing accuracy of reconstructed “ \mathcal{Q} ”	56
TABLE 5.34: For $n = 25, m = 9$, testing accuracy of reconstructed “ \mathcal{A} ”	57
TABLE 5.35: For $n = 25, m = 9$, testing accuracy of reconstructed “ \mathcal{B} ”	57
TABLE 5.36: For $n = 25, m = 9$, testing accuracy of reconstructed “ \mathcal{Y} ”	58
TABLE 5.37: For $n = 25, m = 9$, testing accuracy of reconstructed “ \mathcal{X} ”	58
TABLE 5.38: For $n = 25, m = 9$, testing accuracy of reconstructed “ \mathcal{G} ”	59
TABLE 5.39: For $n = 25, m = 9$, testing accuracy of reconstructed “ \mathcal{C} ”	59
TABLE 5.40: For $n = 25, m = 9$, testing accuracy of reconstructed “ \mathcal{E} ”	60
TABLE 5.41: For $n = 26, m = 8$, testing accuracy of reconstructed “ \mathcal{O} ”	60
TABLE 5.42: For $n = 26, m = 8$, testing accuracy of reconstructed “ \mathcal{H} ”	61
TABLE 5.43: For $n = 26, m = 8$, testing accuracy of reconstructed “ \mathcal{Q} ”	61
TABLE 5.44: For $n = 26, m = 8$, testing accuracy of reconstructed “ \mathcal{A} ”	62
TABLE 5.45: For $n = 26, m = 8$, testing accuracy of reconstructed “ \mathcal{B} ”	62
TABLE 5.46: For $n = 26, m = 8$, testing accuracy of reconstructed “ \mathcal{Y} ”	63
TABLE 5.47: For $n = 26, m = 8$, testing accuracy of reconstructed “ \mathcal{X} ”	63
TABLE 5.48: For $n = 26, m = 8$, testing accuracy of reconstructed “ \mathcal{G} ”	64
TABLE 5.49: For $n = 26, m = 8$, testing accuracy of reconstructed “ \mathcal{C} ”	64
TABLE 5.50: For $n = 26, m = 8$, testing accuracy of reconstructed “ \mathcal{E} ”	65
TABLE 5.51: For $n = 27, m = 7$, testing accuracy of reconstructed “ \mathcal{O} ”	65
TABLE 5.52: For $n = 27, m = 7$, testing accuracy of reconstructed “ \mathcal{H} ”	66
TABLE 5.53: For $n = 27, m = 7$, testing accuracy of reconstructed “ \mathcal{Q} ”	66
TABLE 5.54: For $n = 27, m = 7$, testing accuracy of reconstructed “ \mathcal{A} ”	67
TABLE 5.55: For $n = 27, m = 7$, testing accuracy of reconstructed “ \mathcal{B} ”	67
TABLE 5.56: For $n = 27, m = 7$, testing accuracy of reconstructed “ \mathcal{Y} ”	68

TABLE 5.57: For $n = 27, m = 7$, testing accuracy of reconstructed “ Ξ ”	68
TABLE 5.58: For $n = 27, m = 7$, testing accuracy of reconstructed “ Γ ”	69
TABLE 5.59: For $n = 27, m = 7$, testing accuracy of reconstructed “ \mathcal{C} ”	69
TABLE 5.60: For $n = 27, m = 7$, testing accuracy of reconstructed “ \mathcal{E} ”	70
TABLE 5.61: Comparison of avg. accuracies of Hindi numerals for $(\mathcal{O} - \mathcal{E})$	71

Chapter 1

INTRODUCTION

The main objective of Handwriting Recognition is interpretation of data which describes handwritten drawing by user. On-line handwriting recognition deals with a data stream which is coming from a transducer an electronic device, tablet PC or PDA's while the user writes, and off-line handwriting recognition deals with a data set which has been obtained from a scanned handwritten document. The goal of handwriting recognition is to interpret the contents of the data and to generate a description of that interpretation in the desired format. The primary mode of data input from a user to a computer is still keyboard and many tasks still exist in the people tend to prefer handwritten input over keyboard entry, e.g., classroom notes producing annotated hand sketches [3].

Several handheld devices are becoming essential part of our life to learn varying interfaces. These devices are too small to have full sized keyboards, or may be too small for any keyboard at all, requiring pen or voice interfaces to enter data. So handwritten input is more efficient in comparison to speech input since it is insensitive to environmental noise, which is an important advantage for many applications as diagrammatical calculations, mathematical proofs and languages recognition.

From the past few researches, it has been done to solve this problem in related areas such as Image Processing, Pattern Recognition, Artificial Intelligence, Cognitive Science etc. Problems are being encountered and solutions to these are broadly targeted to improve accuracy and efficiency. Recognition of Online Handwritten Hindi numerals are a goal of many research efforts in the pattern recognition field. The subproblem of numeral recognition is also seen as important, not only because of advances in it that are expected to lead to advances in the general case, but also because of its immediate applicability to a number of fields, the reading of online data written by different Indian people those are not much educated and they want to explore their soft skills.

The challenges in handwritten digit recognition arise not only from the different ways in which a single numeral can be written, but also from the varying requirements imposed by the specific applications like online digital representation of any data [11][15]. The primary performance measures are classifying accuracy or recognition efficiency of an online handwritten recognition system for reading Hindi numerals from various applications like any data written online consisting Hindi numerals may not be appropriate for reading to those people who have never seen or have not learnt the Hindi numerals [6].

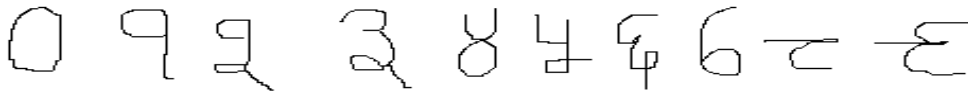


Fig 1.1: Samples of online handwritten Hindi numerals drawn by different user

1.1 Definitions

Handwriting recognition is the ability of a computer to receive intelligible handwritten input. The image of the written text may be sensed “offline” from a piece of paper by optical scanning or the movements of the pen tip may be sensed “online” for example by a pen based computer screen surface. Handwriting Recognition systems is not complete without exploring the Optical Character Recognition (OCR) systems. Optical Character Recognition (OCR) is a problem recognized as being as old as the computer itself [7]. Now a days, researchers focus on online handwritten character recognition. Unfortunately, the success of OCR could not be carried on to Intelligent Character Recognition (ICR), due to the variations in people’s handwriting. As for the recognition of isolated handwritten numerals, the online handwritten character recognition systems include three major parts: Preprocessing Part, Feature Extraction Part and Classification Part. Feature Extraction deals with recognition of input character using a statistical approach, Zernike moment as feature vectors.

Statistical moments have a long history in computer vision since the original work of Hu on moment invariants in the early 1960’s. They are particularly popular due to their

efficient and compact description, their capability to select differing levels of detail and their known performance attributes. The ability to reconstruct an image from its moment description is often exploring as justification for their application. Reconstruction of image using statistical moment is not only recognizing the character even if it computes the accuracy of reconstruction of any character [12] [13].

Moments of orthogonal polynomial basis were proposed by Teague [31]. They have proven less sensitive to noise, are natively invariant to linear transformations and can be effectively used for image reconstruction [5]. There are always arises problems due to computational complexity and it becomes a major issue in real-time implementation. Moments of discrete orthogonal basis have been proposed recently [32]. In this thesis work, we are presenting an overview of the most commonly used image moments, Zernike moment as feature in the recognition of online handwritten Hindi numerals. We are developing a high performance recognition system that must increasing recognition accuracy from other systems.

1.2 Motivation

On the basis of literature survey, character recognition leads the thesis to wok on different algorithms used in the recognition of character images. An experimental approach needs to be developed to compare and evaluate the performance of different invariant features of different shape-based Hindi numerals. The investigations of the reconstruction of the region-based Hindi numeral's image are a major motivation for this thesis. The Zernike moment algorithms may perform well for the ideal clean Hindi numerals. However, in the practical image acquisition systems and conditions, noise is a common phenomenon in the acquisition process. The observed numerals are being represented only a degraded version of the original numerals. Recognition of Hindi numerals and patterns that are of various shapes is being a goal of recent research. Evaluating the accuracy of recognition of handwritten Hindi numerals using Zernike moment is of the thesis's interest.

1.3 Research Scope

The research in Character Recognition usually includes two stages, “lower-level” and “upper-level”. Lower level involves extracting features from character images such as the boundary of a character or regions with the same texture. The task of “upper-level is then to recognize these objects with the extracted features. This thesis concerns with both “lower-level” and “upper-level” stages, in particular with finding properties of a character image which are invariant to image transformations including scale, translation and rotation. The idea of this arises from a project of Optical Character Recognition of printed or handwritten English alphabet letters, if they can recognize using these feature descriptors so we can also recognize Hindi numerals also using that feature descriptors.

All OHCR(Online Handwritten Character Recognition) algorithm in terms of classification accuracy, invariance property against different resolutions and noise as well as the computation efficiency will not possible to be included in this thesis. In this thesis, we are containing our research on online recognition of gray level character image samples of Hindi numerals by implementing Zernike moments as feature vectors.

1.4 Applications

Online recognition algorithms need some sort of learning mechanism to train the parameters for a given application. The idea of online writing is to continue the process of learning while using the system on the job. There are various applications of online handwriting systems exist in real life which makes recognition process easier and fast in this competitive world. Some online applications are as follows:

- Real time computation of data consisting Hindi numerals.
- Pen based systems requires recognition engines to support multiple languages and multi language documents.
- Comparison of different languages needs a robust recognition system [14].

1.5 Thesis Contents

Chapter 1 consists of the definition, motivation, research scope, and applications of this thesis.

Chapter 2 is the general background review. It gives a general introduction to Handwritten Character Recognition system, its classifications and history. Then it introduces the concept of character image features and the character recognition system. Other related concepts including image features, image resolutions, and digital image representation are also introduced in this chapter.

Chapter 3 focuses on the state of the art in Recognition System from the theoretical point of view: their definitions, representing invariance properties to scaling, translation, and rotation of image, reconstruction of Hindi numerals.

Chapter 4 describes the problem statement of recognition system, computing Zernike Moments of each character images, reconstruction process of image using Zernike moment, and accuracy of reconstructed Hindi numerals.

Chapter 5 analyses the problem states about recognition using Zernike Moment and accuracy of reconstruction property of Hindi numerals evaluation.

Chapter 6 gives the conclusion, summary of contributions and future research relating to thesis.

Chapter 2

BACKGROUND INFORMATION

2.1 Character Recognition

Character Recognition is a research area in the field of Pattern Recognition that is widely used in Image Processing and Artificial Intelligence applications. It is broadly classified on the basis of its acquisition process of raw data that further implements the process of preprocessing of any recognition system. Therefore, on the basis of that data acquisition process we can classify the character recognition in different modes as follows:

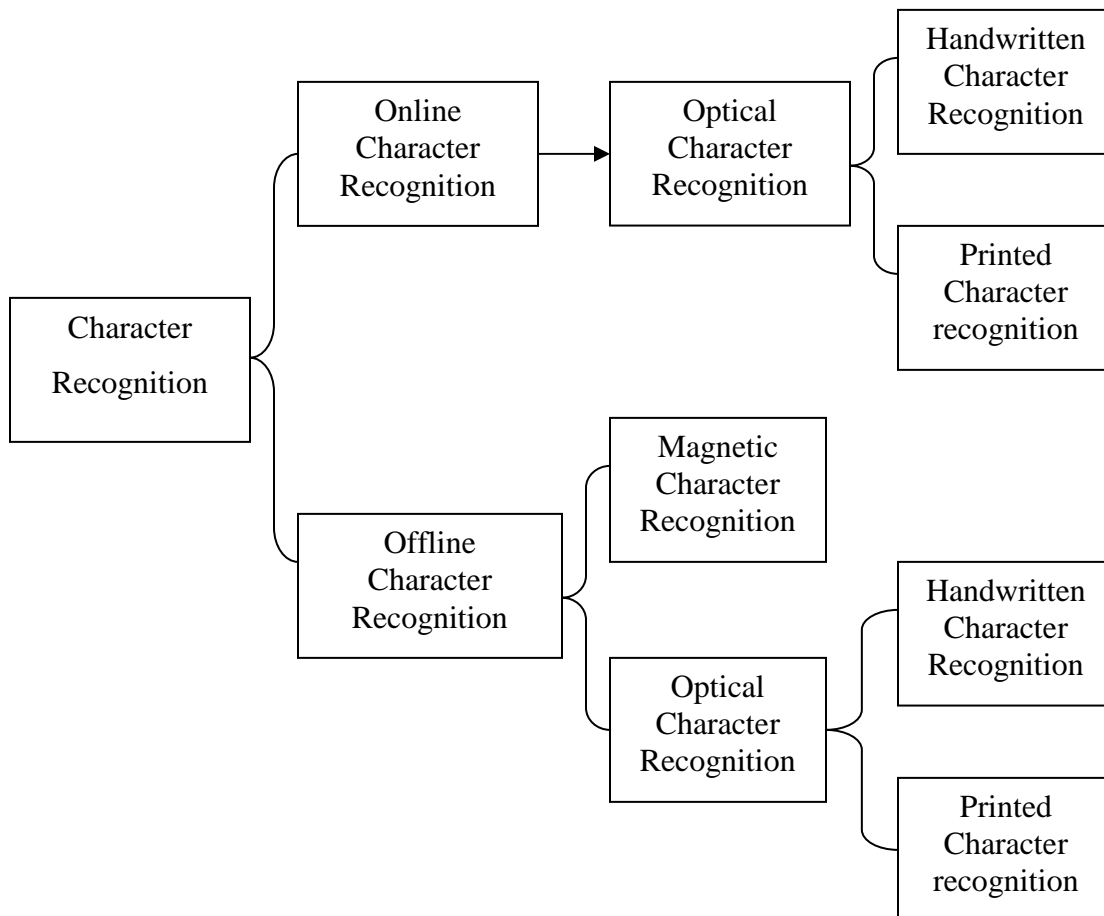


Fig 2.1: Different classification of character recognition

Figure 2.1 shows the different classification of character recognition. Two different classifications are included in the general term of character recognition:

- Online Character Recognition
- Offline Character Recognition

Online Character Recognition always deals with a data stream which comes from a transducer an electronic device while the user is writing. Online handwriting recognition involves the automatic conversion of text as it is written on a special digitizer or PDA, where a sensor picks up the pen-tip movements $x(t)$ and $y(t)$ as well as pen up/pen down switching. Where $x(t)$ is the x -coordinate and $y(t)$ is the y -coordinate of pixel variable t . Such kind of data known as digital ink and can be regarded as a dynamic representation of handwriting. The typical hardware device to collect data is a digitizing tablet which is electromagnetic or pressure sensitive. When the user writes on the electronic tablets, the successive movements of the pen are transformed to a series of electronic digital signals which is memorized and analyzed by the computer [1]. The online handwriting recognition has some distinguish features as:

- It is a real times process.
- It is adaptive in real time.
- It captures the temporal and dynamic information of the pen trajectory.
- Very little preprocessing is required.
- Segmentation is easy in online handwriting.

Offline character recognition is performed after the writing is finished. Offline handwriting recognition involves the automatic conversion of text in an image $f(x, y)$ into letter codes which are usable within computer and text processing applications. The data obtained by this form is regarded as a static representation of handwriting [2]. The major difference between Online and Offline Character Recognition is that Online Character Recognition has real time contextual information but offline data does not [27]. This difference generates a significant divergence in processing architectures and methods. The offline character recognition can be further grouped into:

- Magnetic Character Recognition (MCR)
- Optical Character Recognition (OCR)

In MCR, the characters are printed with magnetic ink. The reading device can recognize the characters according to the unique magnetic field of each character. MCR is mostly used in banks for check authentication. OCR deals with the recognition of characters acquiring by optical means, typically a scanner or a camera. The characters are in the form of pixelized images, and can be either printed or handwritten, of any size, shape, or orientation. The OCR can be subdivide into handwritten character recognition and printed character recognition. Handwritten Character Recognition is more difficult to implement than printed character recognition due to diversify human handwriting styles and customs. In printed character recognition, the images to be processes are in the forms of standard fonts like Times New Roman, Arial, Courier, etc.

2.2 Handwritten Character Recognition

Handwritten Character Recognition (HCR) is an important research area in Pattern Recognition. The objective of an HCR system is to recognize alphabetic letters, numbers, or other characters, which are in the form of digital images, without any human intervention. This is accomplished by searching a match between the features extracted from the given character's image and the library of image models. Ideally, we would like the features to be distinct for different character images so that the computer can extract the correct model from the library without any confusion. Figure 2.2 illustrates the basic processes of an HCR system as follows [26]:

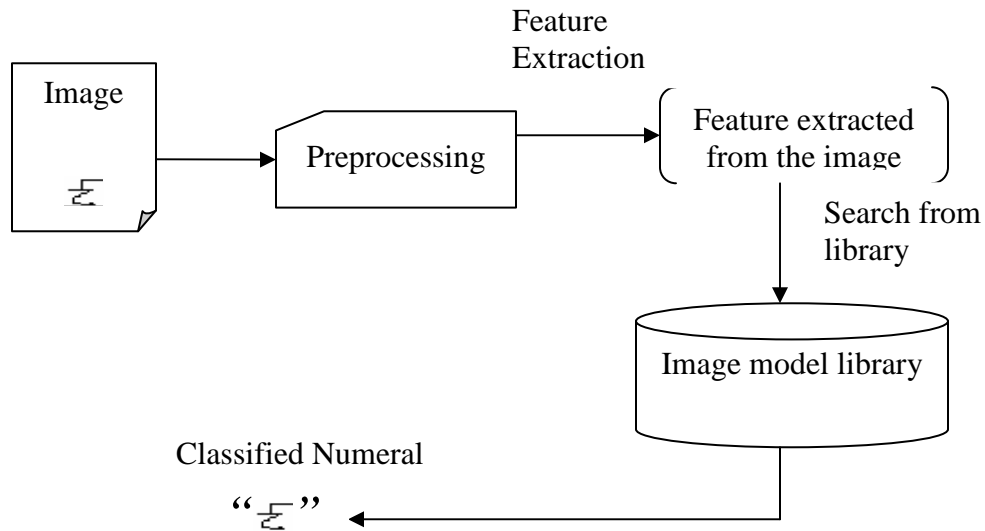


Fig 2.2: The basic processes of an HCR system

This basic process first takes data input from user and leads to preprocessing stage, feature extraction and this extracted image search matching from image model library and then classifying the numerals. As see in the above figure 2.2 user inputs a numeral “६” to recognize by the HCR system and it processes all the stages and finally classifying as Hindi numeral “६”.

2.3 Image Features

Binary images are images that have been quantized to two values, usually denoted 0 and 1, but often with pixel values 0 and 255, representing black and white. Binary images are used in many applications since they are simplest to process, but they are such an impoverished representation of the image information that their use is not always possible. Image features are unique characteristics that can represent a specific image.

Image features are meaningful, detectable parts of the image. Meaningful means the features are associated to interesting elements via the image formation process. Two types of characteristics are usually referred by image features:

1. A global property of a character image, for instance the average gray level of all pixels included in a gray level character image.
2. A part of the character image with special properties, for example the boundary length of a character image.

Sometimes the image features are not directly or obviously associated to any part of the character image, but still reflect particular image properties, like the image moments. Different features are associated with different extraction algorithms that output collections of the feature descriptors. Good character image features should satisfy the following conditions:

1. **Robust to image transformations** – the image features should be as invariant as possible to image transformations including translation, rotation, and scaling, etc.
2. **Robust to degraded noises** – the image features should be robust to noises and various degraded situations.
3. **Feature extraction efficiency** – image features can be computed efficiently.
4. **Feature matching pattern efficiency** – the matching algorithms should only require a reasonable computational cost.

The selection of image features and corresponding extraction methods is probably the most important step in achieving high performance for an OHCR system. At the same time, the image feature and the feature extraction methods also decide the nature and the output of the image-preprocessing step. Some image features and the extraction algorithms work on color images, while others work on gray level or binary images (Appendices A). Numerical features are ideal for statistical classifiers and discrete features are ideal for decision trees.

2.4 Digital Image Representation

A digital image is an image $f(x, y)$ that has been discretized both in spatial coordinates and brightness. A digital image is a binary storage of data that stores in memory and stores in the form of “0” and “1”. A digital image can be considered as a matrix whose

row and column indices identify a point in the character image and the corresponding matrix element value identify the gray level at that point and elements of such a digital array are called image elements, picture elements, or pixels [7].

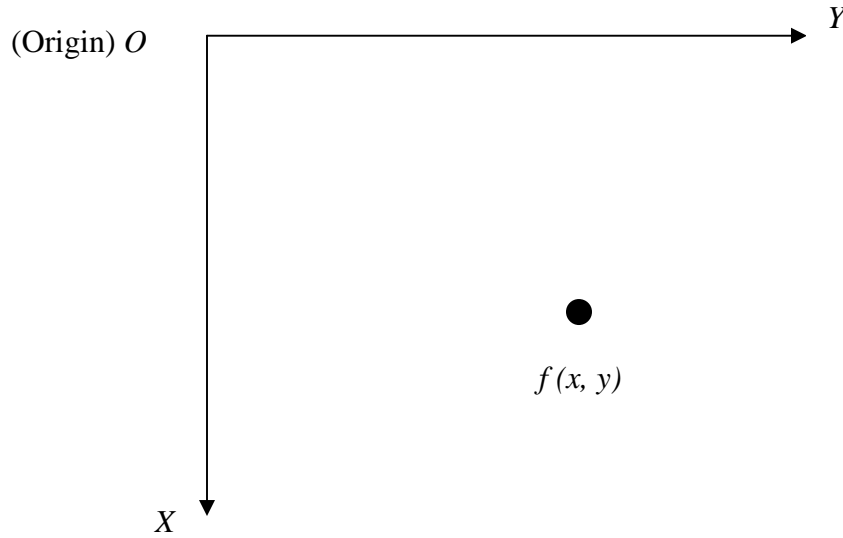


Fig 2.3(a): A digital image $f(x, y)$ in x - y coordinate

In digital image processing method the data should be existed in problem domain. When the data acquisition process follows data stores in the binary format and sends to preprocessing stage, segmentation, representation and recognition process of the data as follows:

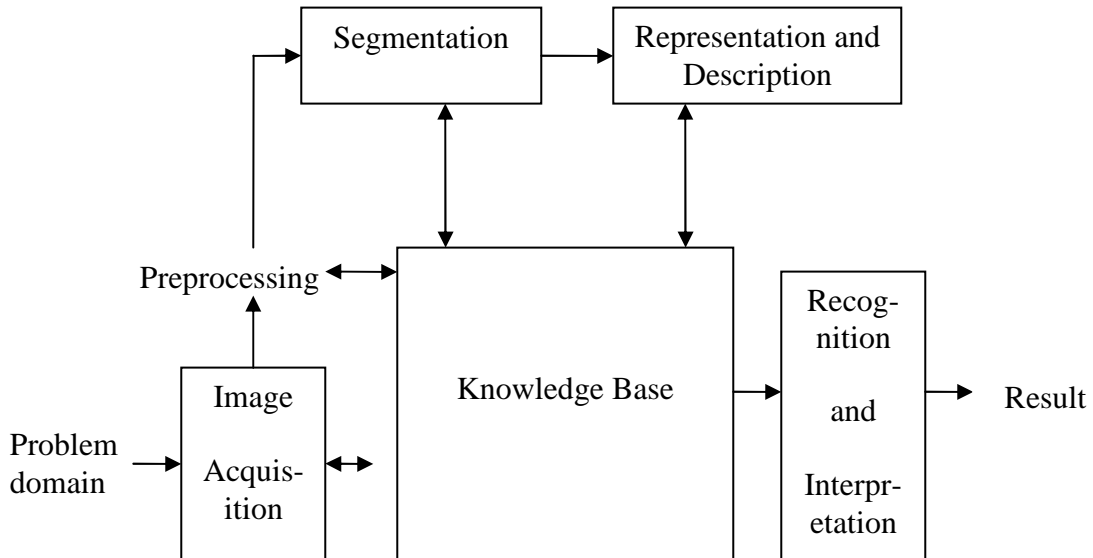


Fig 2.3(b): Fundamental steps in digital image processing

2.4.1 Image Acquisition

From a problem domain, we start an image acquisition process that correlates to knowledge base. There are various types of image acquisition devices as digitizer, PDA's, Transducer, tablet PC and electronic pen. All are stores the image in binary (0 and 1) format. We are mainly focus the image acquisition based on mouse gesture handwriting.

2.4.2 Preprocessing

It deals with technique for enhancing contrast, removing noise and isolating regions whose texture indicate a likelihood of alphanumeric information. In preprocessing stage it is being normalized and removing all redundancy errors from the image and sends to next stage.

2.4.3 Segmentation

It partitions an input image into its constituent parts or objects. In general autonomous segmentation is one of the most different tasks in digital image processing. In terms of character recognition, the key role of segmentation is to extract individual characters and words from background. The output of segmentation stage usually is raw pixel data, constituting either the boundary of a region or all the points in the region itself.

2.4.4 Representation and Description

The data should be represented as a boundary or as a complete region. Boundary representation is appropriate when the focus is on external shape characteristics such as corners and inflections. Regional representation is appropriate when the focus is on internal properties such as textures or skeletal shape. In some applications like character recognition these representations coexist, which often require algorithm based on boundary shape as well as skeletons and other internal properties. Description also called feature selection, deals with extracting features those results in some quantitative information of interest or features that are basic for differentiating one class of objects from another. In terms of character recognition descriptors such as holes and bays are powerful features that help differentiate one part of the numerals from another.

2.4.5 Recognition and Interpretation

It is the process that assigns a label to an object based on the information provided by its descriptors. Interpretation involves assign meaning to an ensemble of recognized objects. In terms of identifying a character say “ $\overline{2}$ ” requires associating the descriptors for that character with the label “ $\overline{2}$ ”. Interpretation attempts to assign meaning to a set of labeled entities, e.g., a set of “10” numerals represents a mobile number and another set of “4” numerals follows by “7” numerals represents a home telephone number.

2.5 Image Resolution

Digital images are made up of small squares called pixels. Image quality is based on image resolutions. There are two types of image resolutions: Spatial and Output. Spatial resolution is defined in terms of width and height. An image of 640 pixels wide and 480 pixels high has a spatial resolution of 640×480 and contains 307,200 pixels. The higher the spatial resolution, the more pixels are available to create the image. This makes more detail possible and creates a sharper image. If we sub-sample this image by deleting every other row and column from the original 512×512 image, we can get an image of 256×256. The 128×128, 64×64, 32×32 images can be generated by the same way [4].

Output resolution is defined in terms of the number of dots/pixels per inch, or dpi. The idea is to match the resolution of a character image to the resolution of the output device, usually a computer screen or printed document. The higher the dpi of the image (up to the limit of the output device), the better the quality of the printed image. Typical ink-jet and laser printers can have output resolutions of 1200dpi. Typical monitor resolution is 96 dpi. From now on, the word “resolution” used in this thesis is only mean the spatial resolution of the image.

2.6 Architecture of the OHCR System

An OHCR (Online Handwritten Character Recognition) is a system which loads a character image, preprocesses the image, extracts proper image features, computes the “statistical moment” between the extracted image features and the known feature vectors (Zernike moment) stored in the image model library, and recognizes the image according to the degree of similarity between the loaded image and the image models. The preprocessing stage aims to make the image be suitable for different feature extraction algorithms. Some feature extraction algorithms only deal with the contours of the image while some algorithms calculate every pixel of the image [10]. The preprocessing stage which includes binarizing, edge detection, gap filling, and segmentation and so on can make the initial image more suitable for later computation as in figure 2.4 given below.

This OHCR system maintains a larger file databases which are creating by initial users and stores the data in each and every file in the binary form. Different shapes drawn by initial users maintain the image model library that further uses in recognition process to match the pattern of original image drawn by new users else shows the percentage match with similar character if there any. In the recognition process image is reconstructed by a statistical moment, Zernike moment using as a feature vectors.

This featured vector takes some standard order of moment (n) with its repetition of order (m) for calculating the Zernike moment (A_{nm}) of each and every pixel value and reconstruct the image using this Zernike moment and also calculates the accuracy of reconstructed image.

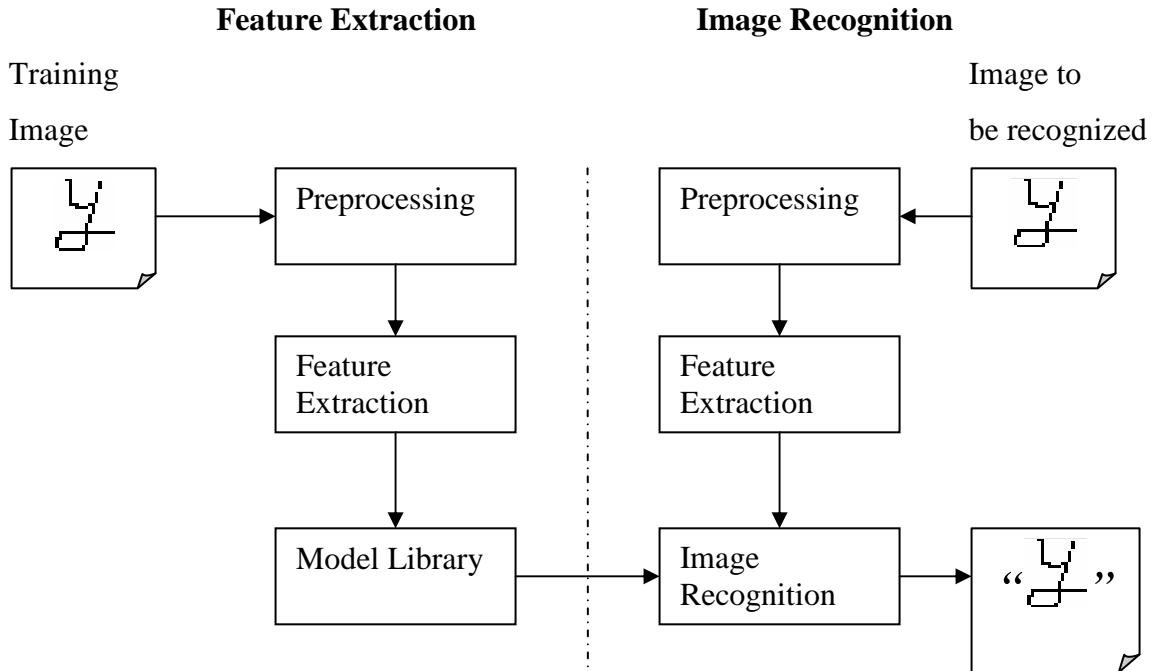


Fig 2.4: Image recognition by Feature Extraction Process

The aim of this thesis is first develop a prototype (an OHCR system) for learning of data, this prototype can be learned by user by thousand numbers of characters .The learning process can be at the run time also. When the system is being learnt by user it goes to be training for different user inputs by the computation of Zernike moment. Zernike moment is calculating for some standard order of moments with their repetition. If the calculation of Zernike moment results correct output then it leads to testing process else process returns back to programming part of the prototype model for error correction. Once the system is being trained then we starts the testing process and obtain maximum accuracy of reconstruction of character image. The basic architecture of OHCR system is describes in the figure 2.5 given below:

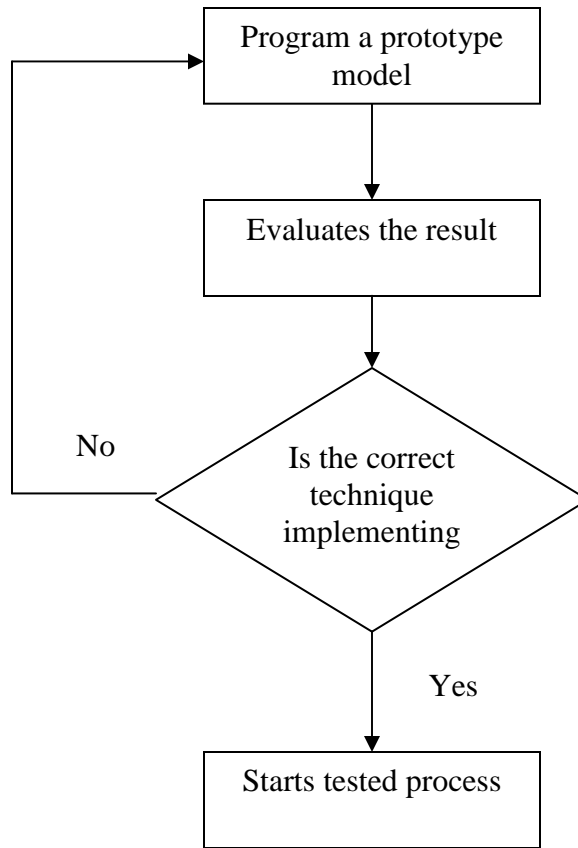


Fig 2.5: The basic process of an OHCR system

2.7 Summary

An Online Handwritten Character Recognition system is a real time process based on character recognition methodology. Character recognition is widely used in Image Processing, Artificial Intelligence applications. It captures the online user data by pen based user interfaces like electronic tablet or PDA's and stores in the file databases of the system in the binary form, i.e., "0" and "1". These are stores different binary format for different shapes drawn by user in the prototype of OHCR system that maintains an image model library that helps to system in the recognition process of any character or numerals using a feature vector Zernike moment, a statistical moment.

Chapter 3

STATE OF THE ART IN RECOGNITION SYSTEM

3.1 Introduction

An important problem in OHCR is the automatic recognition of a character in a scene regardless of its position, size, orientation, etc. In order to recognize different variations of the same character, image features which are invariant to certain transformations need to be used. Image invariants are features which have approximately the same values for samples of the same image which are, for instance, translated, scaled, rotated, skewed, blurred, or noise affected. Image recognition based on these invariants includes three major issues: shape representation, shape similarity measure and shape indexing. Among these issues, shape representation is the most important issue [30]. Various shape representation methods and shape descriptors exist in literatures. These methods can be classified into two categories:

- Boundary-based invariants
- Region-based invariants

In boundary-based invariants, only the contour information of the shape is explored. In region-based invariants, all the pixels within a shape are taken into account to obtain the mathematical representation. The most popular region-based methods include various moment-based invariants such as Hu's seven moment invariants (Appendices B), Zernike moments, complex moments, etc[12][13]. Thus, we are using Zernike moment as a feature vector in recognition of character for reconstructed the image and results the accuracy of reconstruction.

The fundamental element of all these schemes is definition of a set of mathematical functions for image representation and data reduction. Usually additional transformations

requires achieving the desired invariant properties for the image features. This chapter is focus on the theoretical derivation of these shape-based image invariants along with the evaluation of their advantages and disadvantages.

3.2 Boundary-Based Invariants

Boundary is one of the most straightforward and important image features as human beings tend to perceive scenes as being composed of different individual objects, which can be best identified by their boundaries. Meanwhile, as far as the implementation is concerned, boundary is also very simple to calculate. Boundary based invariants are having the features of image boundary in the terms of pixel objects that easily calculate shape boundary of any object. The object consisting larger boundary are having larger pixel resolution as 250×250 has the boundary of 625,00 pixel range and 640×480 has the 307,200 pixel range of boundary. So the boundary based invariants are consists only the contour information of any shape and cannot capture the interior contents of any shape or boundary.

3.3 Region Based Invariants

Region based invariants accounts all of the pixels of the image taken to represent any shape or boundary. Because region-based invariants combine information of an entire image region rather than exploring information just along the boundary pixels, they can capture more information from the image. Moment based invariants are the most common region based image invariants which are using as pattern feature in many applications. One of the statistical moments, Zernike moments that allow independent moment invariants to be reconstructed easily to an arbitrarily high order [16].

Statistical moments are applicable to many different aspects of image processing, ranging from invariant pattern recognition and image encoding to pose estimation. When applies to image, they describe the image content with respect to its axes. They are designed to capture both global and detailed geometric information about the image. In continuous form an image can be consider as a 2-D Cartesian density function $f(x, y)$ with this

assumption, the order of moment (n) with the repetition of order (m), evaluating over the complete image plane ϕ is as follows[29]:

$$M_{nm} = \iint_{\phi} \beta_{nm}(x, y) f(x, y) dx dy, \quad \text{Where } n, m = 0, 1, 2, \dots$$

Where $\beta_{nm}(x, y)$ is the basis function or weighting kernel and produces weighted description of $f(x, y)$ over the entire plane ϕ . These are orthogonal moments that measure the image invariants as follows:

3.3.1 Regular Moments and Central Moments

Regular moments are statistical moments using for continuous image function $f(x, y)$ having order of moments n and their repetitions m , Regular moments (also be referred to as geometric moments) are defined as:

$$m_{nm} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^n y^m f(x, y) dx dy, \quad n, m = 0, 1, 2, \dots$$

Where m_{nm} is the $(n + m)^{th}$ order moment of the continuous image function $f(x, y)$.

The central moments are computed using centroid of the image, which is equivalent to regular moment of an image whose centre has been shifted to coincide with its centroid. Therefore the central moments are invariant to image translations. The central moments of $f(x, y)$ are defined as:

$$\mu_{nm} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \bar{x})^n (y - \bar{y})^m f(x, y) dx dy, \quad n, m = 0, 1, 2, \dots$$

Where $\bar{x} = m_{10}/m_{00}$ and $\bar{y} = m_{01}/m_{00}$, which is the centroid of the image.

For digital images the integral are replaced by summations and m_{nm} becomes:

$$m_{nm} = \sum_x \sum_y x^n y^m f(x, y), \quad n, m = 0, 1, 2, \dots$$

Then the central moments are changed to:

$$\mu_{nm} = \sum_x \sum_y (x - \bar{x})^n (y - \bar{y})^m f(x, y), \quad n, m = 0, 1, 2, \dots$$

3.3.2 Zernike Moments

Zernike function introduced by Frits Zernike, a Dutch Mathematician and Physician, is a set of complex orthogonal functions with a simple rotational property which forms a basis over the class of square integrable function defined over the unit disk. The kernel of Zernike moments is the orthogonal Zernike polynomials defined over polar coordinates inside a unit circle [33] [19]. Zernike introduced a set of complex polynomials $\{V_{nm}(x, y)\}$ which form a complete orthogonal set over the unit disk of $x^2 + y^2 \leq 1$ in polar coordinates [11] [8] [9]. The form of the polynomials is:

$$V_{nm}(x, y) = V_{nm}(\rho, \theta) = R_{nm}(\rho) e^{im\theta}$$

Where n is positive integer or zero, m is integers subject to constraints $n - |m|$ is even, and $|m| \leq n$, ρ is the length of the vector from the origin to the pixel (x, y) , θ is the angle between the vector ρ and x -axis in counter clockwise direction. $R_{nm}(\rho)$ is Radial polynomial defined as:

$$R_{nm}(\rho) = \sum_{s=0}^{(n-|m|)/2} (-1)^s \frac{(n-s)!}{s! \left(\frac{n+|m|}{2} - s\right)! \left(\frac{n-|m|}{2} - s\right)!} \rho^{n-2s}$$

Zernike polynomials are defined only inside a unit circle, $x^2 + y^2 \leq 1$, and therefore the computation of Zernike moments requires a linear coordinate transformation from the image space to the interior of the unit circle, followed by a mapping from the rectangular coordinate system to the polar coordinate system. The values of m, n can be selected to suit the desired sampling frequency [25]. Typically m has a value which is at least $N/2$ and n is at 360° when the image is sampled at one degree intervals:

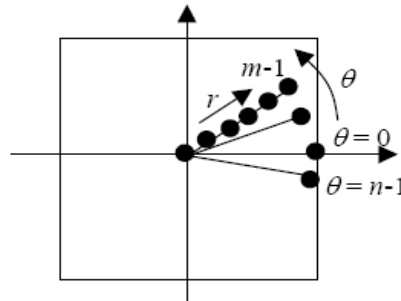


Fig 3.1: Discrete pixel sampling in radial-polar form

The Zernike moment of order n with repetition m for function $f(x, y)$ is defined as:

$$A_{nm} = \frac{n+1}{\pi} \iint_{x^2+y^2 \leq 1} f(x, y) V_{nm}^*(x, y) dx dy$$

Where, $V_{nm}^*(x, y) = V_{n,-m}(x, y)$.

To compute the Zernike moment of a digital character image, we just need to change the integrals with summations:

$$A_{nm} = \frac{n+1}{\pi} \sum_x \sum_y f(x, y) V_{nm}^*(x, y), \quad x^2 + y^2 \leq 1$$

The defined features of Zernike moments themselves are only invariant to rotation. To achieve scale and translation invariance, the image needs to be normalized first by using the regular Zernike moments [23] [24]. The translation invariance is achieved by translating the original image $f(x, y)$ to $f(x + \bar{x}, y + \bar{y})$, where $\bar{x} = \frac{m_{10}}{m_{00}}$ and $\bar{y} = \frac{m_{01}}{m_{00}}$.

In other words, the original image's center is moved to the centroid before the Zernike moment's calculation. Scale invariance is achieved by enlarging or reducing each shape so that the image's 0th regular moment m'_{00} equals to a predetermined value β . For a binary image, m_{00} equals to the total number of shape pixels in the image [20].

According to equation, for a scaled image $f(\alpha x, \alpha y)$, its regular moments $m'_{pq} = \alpha^{p+q+2} m_{pq}$, m_{pq} are the regular moments of $f(x, y)$. Since the objective is to make $m'_{00} = \beta$, we can let $\alpha = \sqrt{\beta/m_{00}}$. By substituting $\alpha = \sqrt{\beta/m_{00}}$ into m'_{00} , we can obtain $m'_{00} = \alpha^2 m_{00} = \beta$.

Suppose we know all Zernike moments A_{nm} of $f(x, y)$ up to order N [22]. Due to the orthogonal property of Zernike moments, we can reconstruct the image based on this set of Zernike moments by:

$$f'(x, y) = \sum_{n=0}^N \sum_m A_{nm} V_{nm}(x, y)$$

3.4 Algorithm's Complexity for moment computation

Since computing complexity of all moment invariants depends almost solely on the computing complexity of geometric moments themselves, we review efficient algorithms for moment calculation in a discrete space. Most of the methods are focused on binary images but there are also a few methods for gray level character images. Basically, moment computation algorithms can be categorized into two groups: decomposition methods and boundary-based methods. The decomposition of the object into simple areas (squares, rectangles, rows, etc.) whose moments can be calculated easily in $O(1)$ time [8].

3.5 Summary

The object moment is then given as a sum of moments of all regions. The boundary based methods calculate object moments just from the boundary, or similar technique. In the discrete case, the integral in the moment definition must be replaced by a summation. Moments are applicable to many different aspects of image processing, ranging from invariant pattern recognition and image encoding to pose estimation. When applied to images, they describe the image content with respect to its axes. They are designed to capture both global and detailed geometric information about the image.

In discrete form of an image can be considered as a 2D Cartesian density distribution function $f(x, y)$ with this assumption, the general form of a moment of order n with repetition m , evaluating over the complete image is as follows:

$$m_{nm} = \sum_{x=1}^N \sum_{y=1}^N x^n y^m f(x, y)$$

Where N is the size of the character image and $f(x, y)$ is the grey levels of individual pixels. m_{nm} is the moment of any discrete image.

Chapter 4

PROBLEM STATEMENT

4.1 Introduction

On basis of literature survey of character recognition algorithms, one question to be answered is: which algorithm is the best choice for a given application? This question leads the thesis to characterize the available algorithms so that the most efficient methods can be sorted out for different applications. An experimental approach needs to be developed to compare and evaluate the performance of different invariants of shape-based Hindi numeral. Meanwhile, as the region-based character image invariants take the whole image area into account, the total number of pixels contained in the image will be a critical factor for the algorithms' computation cost.

An image with 1280×800 pixels will certainly take much longer time to compute than a 32×32 image. The investigations of the reconstruction of the region-based Hindi numeral's image are a major motivation for this thesis. The Zernike moment algorithms may perform well for the ideal clean Hindi numerals. However, in the practical image acquisition systems and conditions, shape distortion is common processes in OHCR system because of different people handwriting have different shape of numerals.

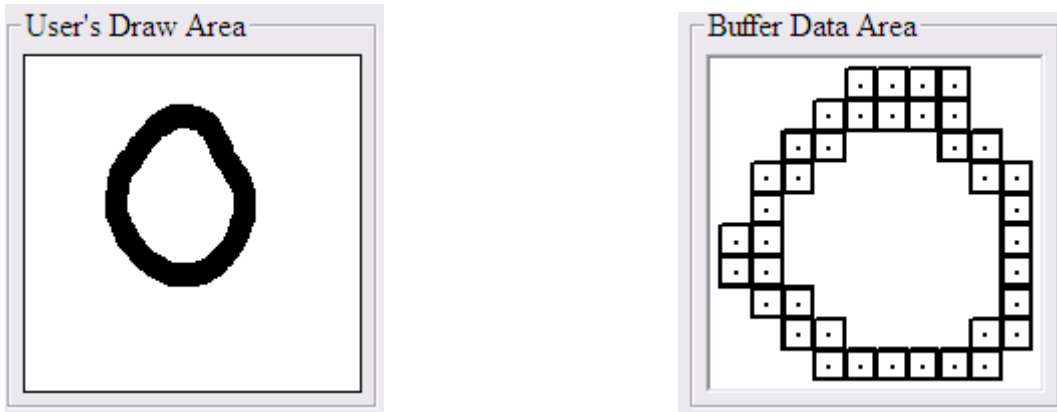
The observed character image is being representing only a degraded version of the original character image. Recognition of Hindi numerals and patterns that are of various shapes is being a goal of recent research. We have recognized the Hindi numerals using the feature vector, Zernike moment on a standard order of moment with standard repetition of order. Computation of the Zernike moment and evaluation of accuracy of reconstruction of handwritten Hindi numerals using Zernike moment is also of this thesis's interest.

4.2 Development of an OHCR System

The problem defines in the acquisition process of an OHCR system can be justified by computing Zernike moment using as a features vector in reconstruction of Hindi numerals. First of all, we taught our system by online handwritten different shapes of Hindi numerals. On the basis of this image model database we match our pattern and classify the reconstructed image. The OHCR system is developing as follows:

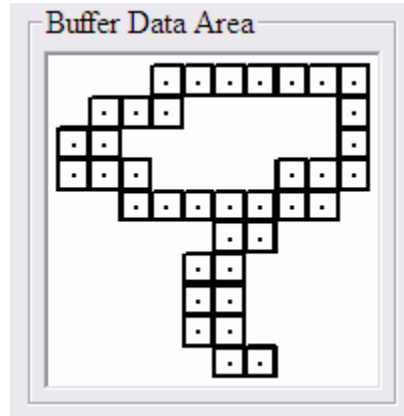
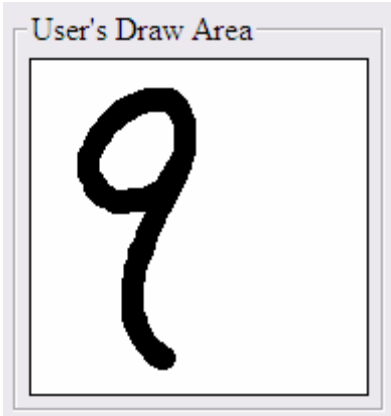
4.2.1 Online Handwritten Image Samples

These are the original image drawn by user by free handwriting that stores in a file databases. This file database makes an image model library in which we are storing different types of binary images drawn by different user using different styles of handwriting. These are the samples of Hindi numerals from (० - ९) drawing by simple handwriting as follows:



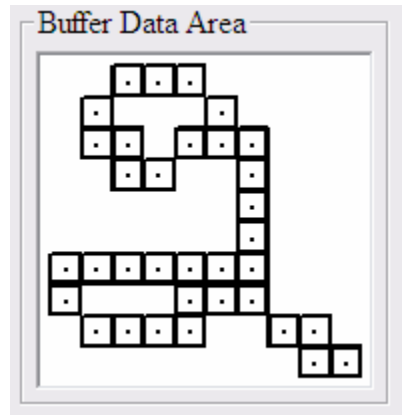
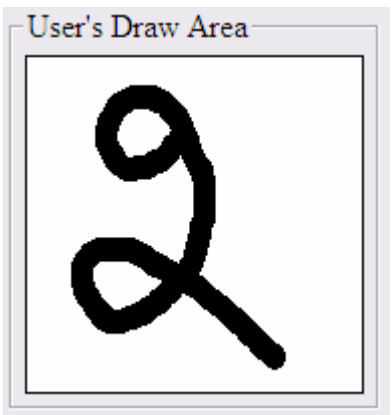
```
"0, 11110000001100000000100011111010011111100011111110  
0011111110001111111000111111100001111110100000000"
```

Fig 4.1: Sample of Hindi numeral 0 drawing by online handwriting and its buffer storage



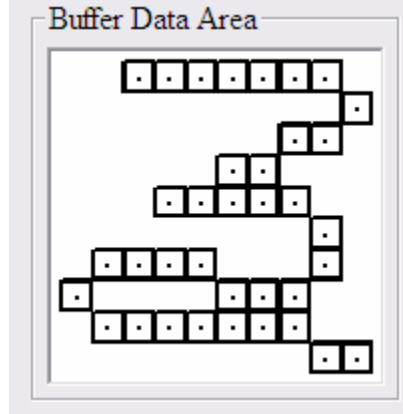
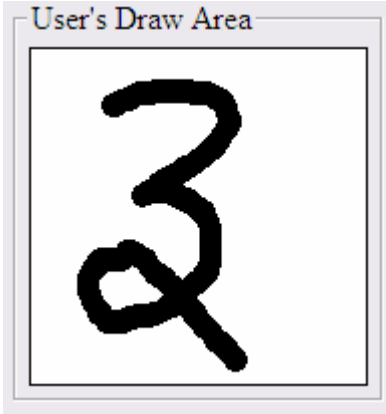
"९, 11000111111001101111101110111101111011110111101111011110111
 101111011111000000000111111110111111111111111111111"

Fig 4.2: Sample of Hindi numeral ९ drawing by online handwriting and its buffer storage



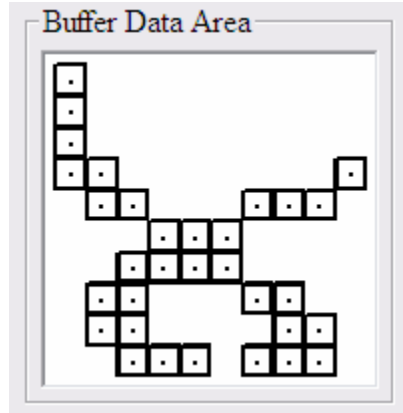
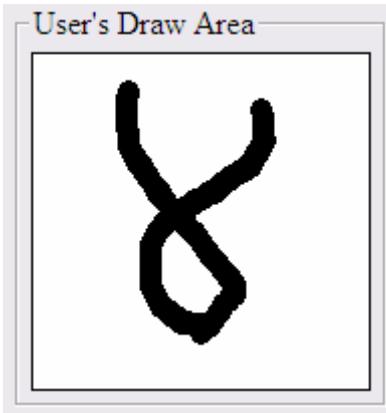
"२, 1111110011100010000100101011010110101011011010101
 1000000011110000000111111110111111111001111111110"

Fig 4.3: Sample of Hindi numeral २ drawing by online handwriting and its buffer storage



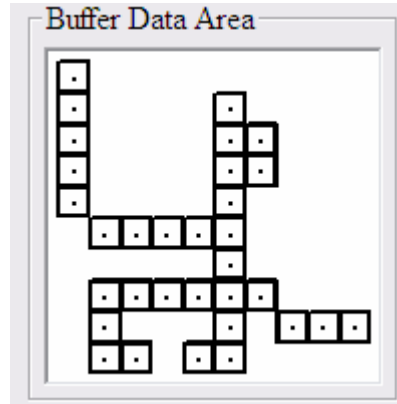
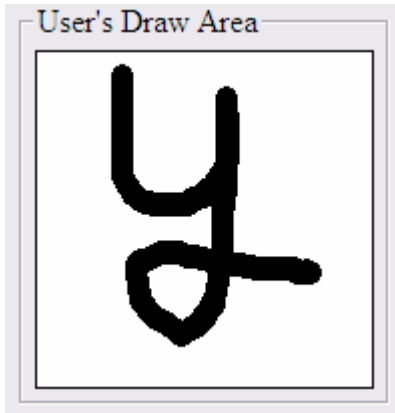
"३, 111111111011111100101110110010111011101011101110
 1 01100110010100011001100101100111111000101111111110"

Fig 4.4: Sample of Hindi numeral ३ drawing by online handwriting and its buffer storage



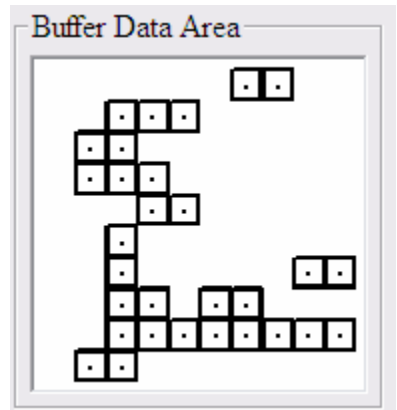
"४, 11111100000001100000000110111011000011101110001110
 11110011101111100110111100001011100100000000011001"

Fig 4.5: Sample of Hindi numeral ४ drawing by online handwriting and its buffer storage



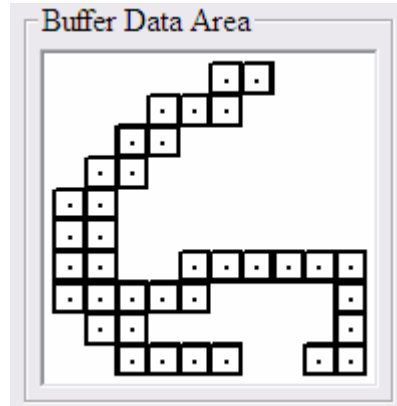
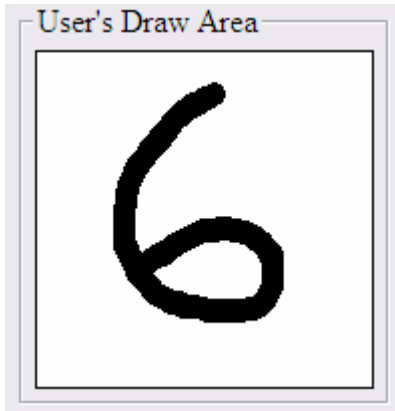
"५, 0000111111000001110011110110101111011010111001101
000000000001111110111111110111111100111111110111"

Fig 4.6: Sample of Hindi numeral ५ drawing by online handwriting and its buffer storage



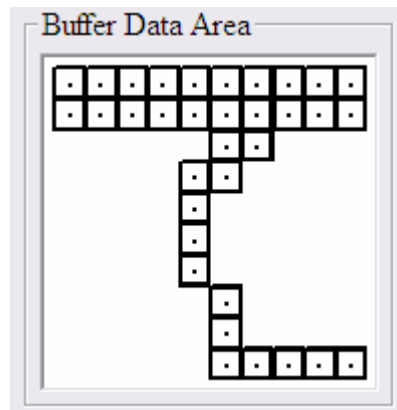
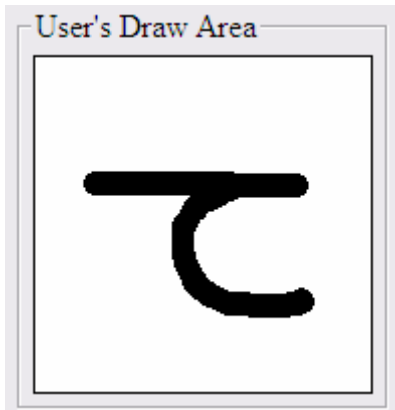
"६, 1100111111100100011110100000110110111011011011101
0 0110110000111110001111110101111111010111111100011"

Fig 4.7: Sample of Hindi numeral ६ drawing by online handwriting and its buffer storage



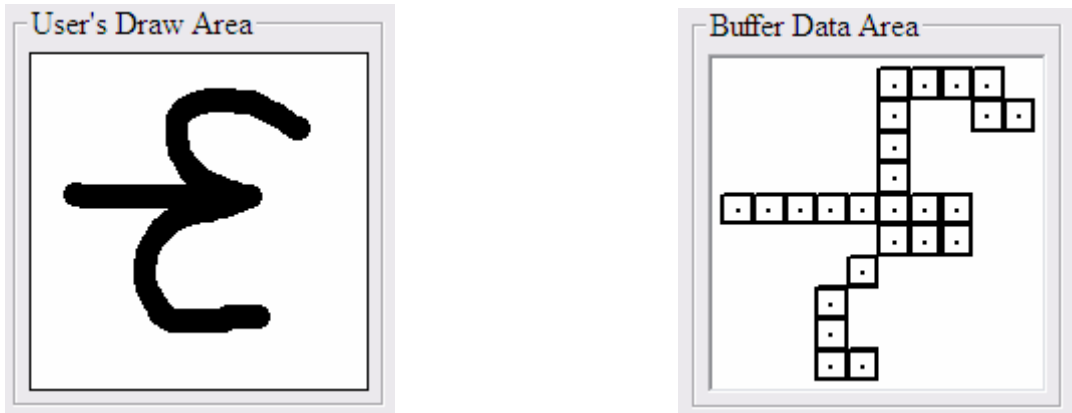
"6, 1100000111100111001100111100010111101100111101111
01111011110111101111011110111101111001101111110000"

Fig 4.8: Sample of Hindi numeral 6 drawing by online handwriting and its buffer storage



"7, 0011111111001111111100111111110011000001000011110
0000111110000011111000011111100011111100011111110"

Fig 4.9: Sample of Hindi numeral 7 drawing by online handwriting and its buffer storage



"३, 11111011111111101111111101111111101111111000000
11100000001001100110101110111000111011100011101110"

Fig 4.10: Sample of Hindi numeral ३ drawing by online handwriting and its buffer storage

4.2.2 How to compute Zernike Moments

Zernike Moments are due to Zernike polynomials introduced in 1934 by Zernike. Zernike polynomials are one of infinite set of polynomials that are orthogonal over the unit circle.

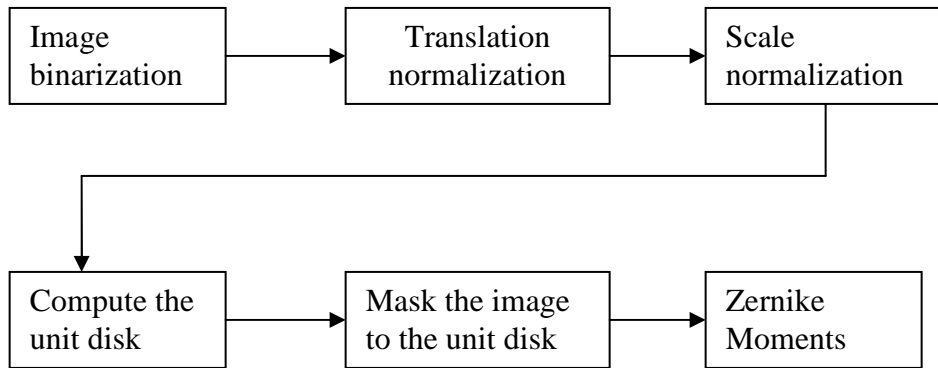


Fig 4.11: Block diagram of computing Zernike moments

Figure 4.11 is the block diagram of Zernike Moment's computation. The major reason for this is the image normalization process. The Zernike moments, however, are only

invariant to image rotation for them. To achieve translation and scale invariance, extra normalization processes are required. The translation normalization can be achieved by moving the image center to the image centroid.

The scale normalization can be achieved by set the image's 0^{th} order regular moment m_{00} to a predetermined value. Because m_{00} is the total number of white pixels for binary image, we use interpolation to set m_{00} to the predetermined value. The regular moments which employ the summation within a square range of pixels, Zernike polynomials take the unit disk $x^2 + y^2 \leq 1$ as their computation domain.

To compute the Zernike moments of a digital image, the range of the image should be mapped to the unit circle first with its origin at the image's center. The pixels falling outside the unit circle are discarding in the computation process. In our implementation of Zernike moments, we use binary images with spatial resolution of 64×64 . All of these binary images are normalizes into a unit circle with fixed radius of 32 pixels. Figure 4.12 shows the process.

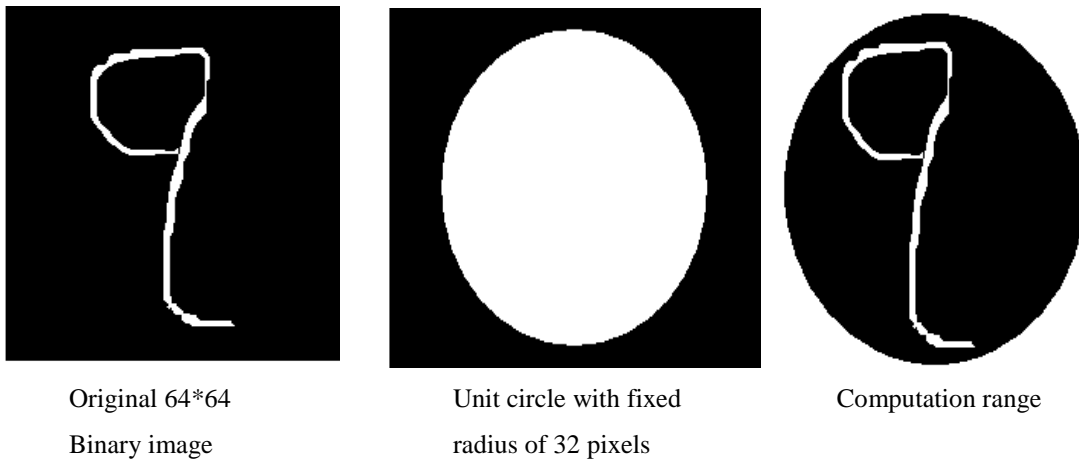


Fig 4.12: The computation process of the unit disk mapping for Zernike moments

To reconstruct any handwritten character image (Hindi numerals) computation of Zernike moments is using in the reconstruction process. So following steps are necessary to implement over any character image to compute the Zernike moments [17]:

- (1) First of all converts gray-scale image into the binary character image.
- (2) To map over a unit disc image must be convert into polar coordinate.

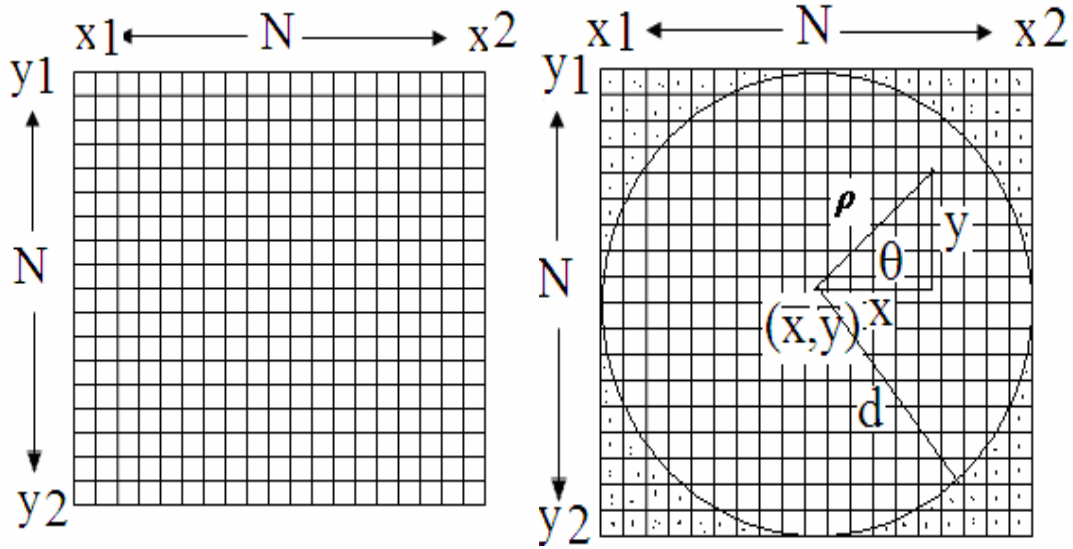


Fig 4.13: (a) $N \times N$ pixels image bitmap b) Unit circles mapped onto $N \times N$ pixels size image

In the above figure 4.13 the center of the image and disk is same. Where $x1, x2$ are X-axis dimensions and $y1, y2$ are Y-axis dimensions of the pixel rectangle, (\bar{x}, \bar{y}) is the center of the unit disk, ρ is polar value and θ is polar angle. Now the image is mapping into polar coordinate and onto unit circle as:

(a) Compute the distance (d) in figure 4.13(b) as $d = \sqrt{(x2 - \bar{x})^2 + (y2 - \bar{y})^2} / 2$

(b) Compute the distance vector ρ and angle θ , for any (x, y) pixel

in $f(x, y)$, in polar coordinate as $\rho = \sqrt{(x - \bar{x})^2 + (y - \bar{y})^2} / d$

and $\theta = \tan^{-1} \left(\frac{y - \bar{y}}{x - \bar{x}} \right)$.

(c) This step maps pixel coordinates $(x1, x2)$ to $(-1, +1)$ and $(y1, y2)$ to $(-1, +1)$ in polar. In this way almost all the pixels in image bound box as given in figure 4.13(b) are inside the unit circle except some foreground pixels.

(3) When the image is mapped onto unit disc, take desired value of order of moment, i.e., n and compute real and imaginary parts of the Zernike moment using Radial

polynomial,
$$R_{nm}(\rho) = \sum_{s=0}^{(n-|m|)/2} (-1)^s \frac{(n-s)!}{s! \left(\frac{n+|m|}{2} - s\right)! \left(\frac{n-|m|}{2} - s\right)!} \rho^{n-2s}$$

(4) Compute Complex Zernike moments from real and imaginary parts and take the magnitudes of Complex Zernike moments. This is rotation invariant Zernike moment feature vector of the given character image.

4.2.3 Image Reconstruction using Zernike Moments

It is a well-recognized property of moments that they can be used to reconstruct the original function, i.e., none of the original image information is lost in the projection of the image on to the moment basis functions, assuming an ‘infinite’ number of moments are calculated. For non-orthogonal moments, the reconstruction is not straight forward and requires a moment-matching technique [28]. In the case of orthogonal moments like Zernike, the reconstruction is simple, by virtue of the orthogonal of the basis functions [18]. When we have all moments of image $f(x, y)$ up to order n_{\max} , then it is possible to compute a discrete image function $f'(x, y)$, whose Zernike moments are exactly same as that of $f(x, y)$ up to order n_{\max} . As n_{\max} approaches infinity, $f'(x, y)$ will become $f(x, y)$ as follows:

$$f'(x, y) = \sum_{n=0}^{n_{\max}} \sum_m A_{nm} V_{nm}(\rho, \theta)$$

It is worth noting that this reconstruction formula gives a discrete approximation to a continuous function, i.e., while the values of x and y are discrete, the values of $f'(x, y)$ are from a continuous range. Another way of considering this point is that if the original image is binary, then the reconstructed function, $f'(x, y)$ will not simply take the values 0 and 1 . The reconstruction effectively gives the simplest (smoothest) function whose moments match the given set.

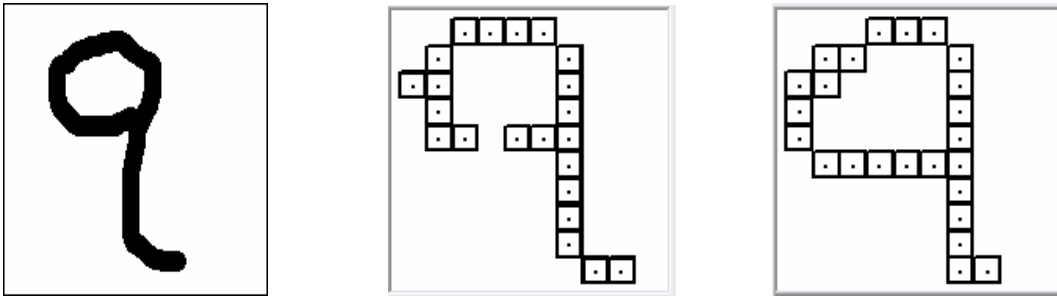


Fig4.14: (a) original image, (b) reconstruction order 23, (c) Reconstruction order 25

Therefore, when reconstructed binary images such as those in Figure 4.14, the reconstructed image needs to be threshold to reproduce a binary image. An appropriate threshold, used in, would appear to be the mid-point between the maxima and minima of the reconstructed function, but there appears to be little discussion in the literature on threshold reconstructions. Using an example of a 64×64 binary letter ‘9’, Figure 4.14 illustrates how increasing the number of moments in the reconstruction improves the resulting image (which are shown here after threshold as described above). It is quite clear from Figure 4.14 that the letter ‘9’ is recognizable from the reconstruction up to and including order 25. However, it is also clear that the reconstructed image does not match the original. Thus, we see that recognition requires fewer moments than reconstruction. Most previous work on reconstruction from moments has concentrated on recognition rather than accuracy, but here we wish to concentrate on the accuracy alone.

4.2.4 Accuracy of Zernike Moment in reconstruction of image

The accuracy of Zernike moment for reconstructed the image shows the performance of the OHCR system, i.e., how much efficient to recognize any Hindi numerals. For any Hindi numerals image the accuracy of reconstructed image (in %) is as follows:

$$Accuracy = \frac{\sum_x \sum_y (f(x, y) - f'(x, y))^2}{\sum_x \sum_y f(x, y)^2} \times 100$$

Where $f(x, y)$ is original image of any Hindi numerals and $f'(x, y)$ is reconstructed image of same Hindi numerals. Using different order of moment(n) with different repetition(m), we can be obtained the accuracy of the reconstructed image [21].

4.3 Summary

Zernike moments themselves are not invariant to translation and scale change. The images need to be preprocessed first to achieve the property of translation and scaling invariance. The advantage of Zernike moments lays in the image reconstruction priority. We define the problem statement of this thesis in this chapter and describe the implementation of the Zernike moment in the reconstruction of image. Furthermore, Zernike moments also help to find out the accuracy of the reconstructed Hindi numerals drawn by different users. For this reason, we focused our research on their accuracy of reconstruction ability rather than their sensitivity property to different spatial resolutions.

Chapter 5

RESEARCH PROBLEM SOLVING

5.1 Computation of Zernike moment and Accuracy of reconstructed image using Zernike moment

As we define the Zernike moment (A_{mn}) and Zernike Polynomial (V_{mn}) in the previous chapter, using these values we are calculating the accuracy of reconstructed image (in %) as follows:

$$Accuracy = \frac{\sum_x \sum_y (f(x, y) - f'(x, y))^2}{\sum_x \sum_y f(x, y)^2} \times 100$$

Where $f'(x, y) = \sum_{n=0}^{n_{max}} \sum_m A_{nm} V_{nm}(\rho, \theta)$, ρ is polar value and θ is polar angle and $f(x, y)$ is original image of any Hindi numerals and $f'(x, y)$ is reconstructed image of same Hindi numerals.

We have followed these steps to obtain best accuracy of reconstructed image from the OHCR system. First of all, we have trained our OHCR system by different order of moment with different repetitions and measured the accuracy of reconstructed image using Zernike moment. In the next step, we have checked accuracy for each and every Hindi numeral from “०-९” at the order of moment and respective repetition of order obtained from previous step. As well as we have checked accuracy of reconstructed Hindi numerals handwritten by 10 different users so that we can obtain best accuracy of reconstructed image at a standard order of moment (n) and repetition of order (m).

In the training step we checked accuracy of reconstructed image from OHCR system at different order of moment (n) and repetition of order (m). The observed data for Hindi numerals “०-९” are showing in the tables as follows:

TABLE 5.1: Accuracy of reconstructed Hindi numeral “०” by OHCR system

Order of moment (n)	Repetition of order (m)	Accuracy of reconstructed image(%)	Average Accuracy
18	4	99.4228	98.3228
18	6	87.5957	
20	6	95.5031	
21	3	96.4845	
21	5	99.6375	
21	7	98.7282	
22	6	99.9612	
22	8	99.6916	
22	10	87.7945	
23	5	99.8601	
23	7	99.9998	
23	9	99.9710	
23	11	99.5404	
24	6	99.9758	
24	8	99.9994	
24	10	99.9994	
25	5	98.8599	
25	7	99.9990	
25	9	99.9999	
26	6	99.9177	
26	8	99.9999	
27	7	99.9988	
28	6	98.4845	

From the above table 5.1, we obtained the average accuracy of reconstructed Hindi numeral ० as 98.3228% from OHCR system at different order of moment and repetition of order.

TABLE 5.2: Accuracy of reconstructed Hindi numeral image “१” by OHCR system

Order of moment (n)	Repetition of order (m)	Accuracy of reconstructed image(%)	Average Accuracy
18	4	91.6485	96.9210
18	6	81.1887	
20	6	92.4590	
21	3	93.0148	
21	5	99.1242	
21	7	98.0465	
22	6	99.9269	
22	8	99.5224	
22	10	82.0405	
23	5	99.6431	
23	7	99.9998	
23	9	99.9539	
23	11	99.2795	
24	6	99.9455	
24	8	99.9989	
24	10	99.9990	
25	5	97.0031	
25	7	99.9980	
25	9	99.9999	
26	6	99.8159	
26	8	99.9999	
27	7	99.9976	
28	6	96.5783	

From the above Table 5.2, we obtained the average accuracy of reconstructed Hindi numeral १ as 96.9210% from OHCR system at different order of moment and repetition of order.

TABLE 5.3: Accuracy of reconstructed Hindi numeral image “२” by OHCR system

Order of moment (n)	Repetition of order (m)	Accuracy of reconstructed image(%)	Average Accuracy
18	4	94.5271	97.4632
18	6	83.6661	
20	6	93.6653	
21	3	94.3544	
21	5	99.3123	
21	7	98.3256	
22	6	99.9404	
22	8	99.5928	
22	10	84.4005	
23	5	99.7226	
23	7	99.9998	
23	9	99.9611	
23	11	99.3895	
24	6	99.9572	
24	8	99.9991	
24	10	99.9992	
25	5	97.6807	
25	7	99.9985	
25	9	99.9999	
26	6	99.8551	
26	8	99.9999	
27	7	99.9981	
28	6	97.3082	

From the above table 5.3, we obtained the average accuracy of reconstructed Hindi numeral २ as 97.4632% from OHCR system at different order of moment and repetition of order.

TABLE 5.4: Accuracy of reconstructed Hindi numeral image “३” by OHCR system

Order of moment (n)	Repetition of order (m)	Accuracy of reconstructed image(%)	Average Accuracy
18	4	90.9672	97.2144
18	6	83.6544	
20	6	93.3978	
21	3	93.3483	
21	5	99.1832	
21	7	98.3413	
22	6	99.9343	
22	8	99.5943	
22	10	84.7873	
23	5	99.6639	
23	7	99.9998	
23	9	99.9606	
23	11	99.3859	
24	6	99.9489	
24	8	99.9991	
24	10	99.9992	
25	5	97.1526	
25	7	99.9982	
25	9	99.9999	
26	6	99.8278	
26	8	99.9999	
27	7	99.9976	
28	6	96.7889	

From the above table 5.4, we obtained the average accuracy of reconstructed Hindi numeral ३ as 97.2144% from OHCR system at different order of moment and repetition of order.

TABLE 5.5: Accuracy of reconstructed Hindi numeral image “३” by OHCR system

Order of moment(n)	Repetition of Order (m)	Accuracy of reconstructed image(%)	Average Accuracy
18	4	96.4080	98.6440
18	6	91.0449	
20	6	96.7550	
21	3	97.3225	
21	5	99.6578	
21	7	99.1346	
22	6	99.9706	
22	8	99.7923	
22	10	91.7311	
23	5	99.8629	
23	7	99.9998	
23	9	99.9805	
23	11	99.6923	
24	6	99.9794	
24	8	99.9995	
24	10	99.9996	
25	5	98.8543	
25	7	99.9992	
25	9	99.9999	
26	6	99.9301	
26	8	99.9999	
27	7	99.9971	
28	6	98.6960	

The above table 5.5 gives the average accuracy of reconstructed Hindi numeral ३ as 98.6440% from OHCR system at different order of moment and repetition of order.

TABLE 5.6: Accuracy of reconstructed Hindi numeral image “५” by OHCR system

Order of moment(n)	Repetition of Order (m)	Accuracy of reconstructed image(%)	Average Accuracy
18	4	81.5340	98.6850
18	6	97.4881	
20	6	98.6179	
21	3	97.9412	
21	5	99.6489	
21	7	99.7529	
22	6	99.9817	
22	8	99.9372	
22	10	97.8971	
23	5	99.8437	
23	7	99.9999	
23	9	99.9931	
23	11	99.8986	
24	6	99.9799	
24	8	99.9998	
24	10	99.9998	
25	5	98.6053	
25	7	99.9994	
25	9	99.9999	
26	6	99.9327	
26	8	99.9999	
27	7	99.9993	
28	6	98.7086	

From this table 5.6, we have obtained the average accuracy of reconstructed Hindi numeral ५ as 98.6850% using the proposed OHCR system at different order of moment and repetition of order.

TABLE 5.7: Accuracy of reconstructed Hindi numeral image “५” by OHCR system

Order of moment(n)	Repetition of Order (m)	Accuracy of reconstructed image(%)	Average Accuracy
18	4	93.7600	97.0470
18	6	81.4404	
20	6	92.5813	
21	3	92.8475	
21	5	99.1613	
21	7	98.0751	
22	6	99.9281	
22	8	99.5291	
22	10	82.3068	
23	5	99.6589	
23	7	99.9998	
23	9	99.9545	
23	11	99.2891	
24	6	99.9467	
24	8	99.9989	
24	10	99.9991	
25	5	97.1351	
25	7	99.9981	
25	9	99.9999	
26	6	99.8199	
26	8	99.9999	
27	7	99.9977	
28	6	96.6566	

From the above table 5.7, we have obtained the average accuracy of reconstructed Hindi numeral ५ as 97.0470% from OHCR system at different order of moment and repetition of order.

TABLE 5.8: Accuracy of reconstructed Hindi numeral image “६” by OHCR system

Order of moment(n)	Repetition of Order (m)	Accuracy of reconstructed image(%)	Average Accuracy
18	4	97.7154	97.3170
18	6	81.5355	
20	6	92.8359	
21	3	93.8754	
21	5	99.3183	
22	6	99.9344	
22	8	99.5219	
22	10	81.7633	
23	5	99.7296	
23	7	99.9997	
23	9	99.9543	
23	11	99.2806	
24	6	99.9554	
24	8	99.9990	
24	10	99.9991	
25	5	97.7639	
25	7	99.9983	
25	9	99.9999	
26	6	99.8489	
26	8	99.9999	
27	7	99.9979	
28	6	97.2094	

From the above table 5.8, we have obtained the average accuracy of reconstructed Hindi numeral ६ as 97.3170% from OHCR system at different order of moment and repetition of order.

TABLE 5.9: Accuracy of reconstructed Hindi numeral image “८” by OHCR system

Order of moment(n)	Repetition of Order (m)	Accuracy of reconstructed image(%)	Average Accuracy
18	4	84.4306	98.8970
18	6	97.9343	
20	6	98.8569	
21	3	98.2351	
21	5	99.7088	
21	7	99.7979	
22	6	99.9848	
22	8	99.9485	
22	10	98.2837	
23	5	99.8702	
23	7	99.9999	
23	9	99.9943	
23	11	99.9167	
24	6	99.9831	
24	8	99.9998	
24	10	99.9998	
25	5	98.8381	
25	7	99.9995	
25	9	99.9999	
26	6	99.9436	
26	8	99.9999	
27	7	99.9993	
28	6	98.9170	

From the above table 5.9, we have obtained the average accuracy of reconstructed Hindi numeral ८ as 98.8970% from OHCR system at different order of moment and repetition of order.

TABLE 5.10: Accuracy of reconstructed Hindi numeral image“५”by OHCR system

Order of moment(n)	Repetition of Order (m)	Accuracy of reconstructed image(%)	Average Accuracy
18	4	89.0512	96.4160
18	6	79.1678	
20	6	91.3712	
21	3	91.0601	
21	5	98.9305	
21	7	97.8318	
22	6	99.9129	
22	8	99.4667	
22	10	80.3743	
23	5	99.5590	
23	7	99.9998	
23	9	99.9478	
23	11	99.1891	
24	6	99.9320	
24	8	99.9988	
24	10	99.9989	
25	5	96.2678	
25	7	99.9976	
25	9	99.9999	
26	6	99.7708	
26	8	99.9999	
27	7	99.9971	
28	6	95.7391	

From the above table 5.10, we found the average accuracy of reconstructed Hindi numeral ५ as 96.4160% using the OHCR system at different order of moment and repetition of order

It can be observed from the above table 5.1 to table 5.10, we have obtained the reconstructed accuracy of the original numerals greater than 99.9900% on following order of moment (n) and repetition of order (m) as given below:

1. $n = 23, m = 7$
2. $n = 24, m = 10$
3. $n = 25, m = 9$
4. $n = 26, m = 8$
5. $n = 27, m = 7$

We have now tested the accuracy of reconstructed image by these orders of moment (n) and their repetition (m) for handwriting of 10 different users (P1, P2, P3, P4, P5, P6, P7, P8, P9, P10) so that we can decide for the best accuracy of reconstruction for Hindi numerals “०-५” on any standard order of moment (n) and repetition of order (m).

For $n = 23, m = 7$, we have tested the accuracy of reconstructed Hindi numerals “०-५” by OHCR system and the results are given in the following tables.

TABLE 5.11: For $n = 23, m = 7$, testing accuracy of reconstructed numeral “०”

USER	Accuracy of reconstructed image (%) for $n = 23, m = 7$	Average Accuracy for $n = 23, m = 7$
P1	99.9997	99.9997
P2	99.9997	
P3	99.9998	
P4	99.9997	
P5	99.9997	
P6	99.9997	
P7	99.9997	
P8	99.9997	
P9	99.9997	
P10	99.9997	

TABLE 5.12: For $n = 23, m = 7$, testing accuracy of reconstructed numeral “7”

USER	Accuracy of reconstructed image (%) for $n = 23, m = 7$	Average Accuracy for $n = 23, m = 7$
P1	99.9998	99.9998
P2	99.9998	
P3	99.9997	
P4	99.9998	
P5	99.9998	
P6	99.9998	
P7	99.9998	
P8	99.9998	
P9	99.9998	
P10	99.9998	

TABLE 5.13: For $n = 23, m = 7$, testing accuracy of reconstructed numeral “8”

USER	Accuracy of reconstructed image (%) for $n = 23, m = 7$	Average Accuracy for $n = 23, m = 7$
P1	99.9998	99.9998
P2	99.9998	
P3	99.9997	
P4	99.9997	
P5	99.9999	
P6	99.9997	
P7	99.9998	
P8	99.9998	
P9	99.9998	
P10	99.9997	

TABLE 5.14: For $n = 23, m = 7$, testing accuracy of reconstructed numeral “3”

USER	Accuracy of reconstructed image (%) for $n = 23, m = 7$	Average Accuracy for $n = 23, m = 7$
P1	99.9998	99.9998
P2	99.9998	
P3	99.9999	
P4	99.9997	
P5	99.9998	
P6	99.9997	
P7	99.9998	
P8	99.9998	
P9	99.9998	
P10	99.9997	

TABLE 5.15: For $n = 23, m = 7$, testing accuracy of reconstructed numeral “8”

USER	Accuracy of reconstructed image (%) for $n = 23, m = 7$	Average Accuracy for $n = 23, m = 7$
P1	99.9998	99.9998
P2	99.9999	
P3	99.9999	
P4	99.9999	
P5	99.9998	
P6	99.9998	
P7	99.9998	
P8	99.9997	
P9	99.9998	
P10	99.9999	

TABLE 5.16: For $n = 23, m = 7$, testing accuracy of reconstructed numeral “ $\underline{4}$ ”

USER	Accuracy of reconstructed image (%) for $n = 23, m = 7$	Average Accuracy for $n = 23, m = 7$
P1	99.9999	99.9998
P2	99.9998	
P3	99.9999	
P4	99.9998	
P5	99.9998	
P6	99.9998	
P7	99.9998	
P8	99.9999	
P9	99.9998	
P10	99.9998	

TABLE 5.17: For $n = 23, m = 7$, testing accuracy of reconstructed numeral “ $\underline{5}$ ”

USER	Accuracy of reconstructed image (%) for $n = 23, m = 7$	Average Accuracy for $n = 23, m = 7$
P1	99.9998	99.9998
P2	99.9998	
P3	99.9998	
P4	99.9998	
P5	99.9998	
P6	99.9997	
P7	99.9998	
P8	99.9998	
P9	99.9998	
P10	99.9998	

TABLE 5.18: For $n = 23, m = 7$, testing accuracy of reconstructed numeral “6”

USER	Accuracy of reconstructed image (%) for $n = 23, m = 7$	Average Accuracy for $n = 23, m = 7$
P1	99.9997	99.9997
P2	99.9997	
P3	99.9997	
P4	99.9998	
P5	99.9997	
P6	99.9997	
P7	99.9997	
P8	99.9997	
P9	99.9997	
P10	99.9997	

TABLE 5.19: For $n = 23, m = 7$, testing accuracy of reconstructed numeral “7”

USER	Accuracy of reconstructed image (%) for $n = 23, m = 7$	Average Accuracy for $n = 23, m = 7$
P1	99.9999	99.9998
P2	99.9999	
P3	99.9998	
P4	99.9998	
P5	99.9998	
P6	99.9998	
P7	99.9997	
P8	99.9998	
P9	99.9998	
P10	99.9998	

TABLE 5.20: For $n = 23, m = 7$, testing accuracy of reconstructed numeral “५”

USER	Accuracy of reconstructed image (%) for $n = 23, m = 7$	Average Accuracy for $n = 23, m = 7$
P1	99.9999	99.9998
P2	99.9998	
P3	99.9998	
P4	99.9998	
P5	99.9998	
P6	99.9998	
P7	99.9998	
P8	99.9999	
P9	99.9998	
P10	99.9998	

For $n = 24, m = 10$, we have again tested the accuracy of reconstructed Hindi numerals “०-९” by OHCR system and the results are being shown in the tables given below:

TABLE 5.21: For $n = 24, m = 10$, testing accuracy of reconstructed numeral “०”

USER	Accuracy of reconstructed image (%)for $n = 24, m = 10$	Average Accuracy for $n = 24, m = 10$
P1	99.9994	99.9992
P2	99.9989	
P3	99.9996	
P4	99.9994	
P5	99.9992	
P6	99.9992	
P7	99.9992	
P8	99.9992	
P9	99.9990	
P10	99.9989	

TABLE 5.22: For $n = 24, m = 10$, testing accuracy of reconstructed numeral “7”

USER	Accuracy of reconstructed image (%)for $n = 24, m = 10$	Average Accuracy for $n = 24, m = 10$
P1	99.9990	99.9992
P2	99.9993	
P3	99.9990	
P4	99.9991	
P5	99.9991	
P6	99.9991	
P7	99.9993	
P8	99.9996	
P9	99.9991	
P10	99.9991	

TABLE 5.23: For $n = 24, m = 10$, testing accuracy of reconstructed numeral “8”

USER	Accuracy of reconstructed image (%)for $n = 24, m = 10$	Average Accuracy for $n = 24, m = 10$
P1	99.9992	99.9991
P2	99.9992	
P3	99.9991	
P4	99.9994	
P5	99.9993	
P6	99.9991	
P7	99.9989	
P8	99.9989	
P9	99.9991	
P10	99.9989	

TABLE 5.24: For $n = 24, m = 10$, testing accuracy of reconstructed numeral “ \mathfrak{a} ”

USER	Accuracy of reconstructed image (%)for $n = 24, m = 10$	Average Accuracy for $n = 24, m = 10$
P1	99.9992	99.9991
P2	99.9993	
P3	99.9995	
P4	99.9991	
P5	99.9988	
P6	99.9989	
P7	99.9992	
P8	99.9989	
P9	99.9990	
P10	99.9990	

TABLE 5.25: For $n = 24, m = 10$, testing accuracy of reconstructed numeral “ \mathfrak{b} ”

USER	Accuracy of reconstructed image (%)for $n = 24, m = 10$	Average Accuracy for $n = 24, m = 10$
P1	99.9996	99.9994
P2	99.9996	
P3	99.9996	
P4	99.9998	
P5	99.9994	
P6	99.9992	
P7	99.9993	
P8	99.9990	
P9	99.9990	
P10	99.9994	

TABLE 5.26: For $n = 24, m = 10$, testing accuracy of reconstructed numeral “4”

USER	Accuracy of reconstructed image (%)for $n = 24, m = 10$	Average Accuracy for $n = 24, m = 10$
P1	99.9998	99.9992
P2	99.9992	
P3	99.9998	
P4	99.9988	
P5	99.9989	
P6	99.9992	
P7	99.9990	
P8	99.9996	
P9	99.9992	
P10	99.9990	

TABLE 5.27: For $n = 24, m = 10$, testing accuracy of reconstructed numeral “8”

USER	Accuracy of reconstructed image (%)for $n = 24, m = 10$	Average Accuracy for $n = 24, m = 10$
P1	99.9990	99.9990
P2	99.9990	
P3	99.9990	
P4	99.9989	
P5	99.9993	
P6	99.9989	
P7	99.9989	
P8	99.9989	
P9	99.9990	
P10	99.9990	

TABLE 5.28: For $n = 24, m = 10$, testing accuracy of reconstructed numeral “6”

USER	Accuracy of reconstructed image (%)for $n = 24, m = 10$	Average Accuracy for $n = 24, m = 10$
P1	99.9990	99.9990
P2	99.9990	
P3	99.9989	
P4	99.9989	
P5	99.9991	
P6	99.9990	
P7	99.9990	
P8	99.9989	
P9	99.9991	
P10	99.9989	

TABLE 5.29: For $n = 24, m = 10$, testing accuracy of reconstructed numeral “7”

USER	Accuracy of reconstructed image (%)for $n = 24, m = 10$	Average Accuracy for $n = 24, m = 10$
P1	99.9998	99.9991
P2	99.9996	
P3	99.9989	
P4	99.9988	
P5	99.9993	
P6	99.9988	
P7	99.9991	
P8	99.9988	
P9	99.9991	
P10	99.9992	

TABLE 5.30: For $n = 24, m = 10$, testing accuracy of reconstructed numeral “ $\frac{5}{2}$ ”

USER	Accuracy of reconstructed image (%) for $n = 24, m = 10$	Average Accuracy for $n = 24, m = 10$
P1	99.9989	99.9990
P2	99.9989	
P3	99.9989	
P4	99.9990	
P5	99.9990	
P6	99.9989	
P7	99.9989	
P8	99.9995	
P9	99.9988	
P10	99.9988	

We have now tested the accuracy of the system for $n = 25, m = 9$ using the OHCR system. The results for this are given in the tables as below.

TABLE 5.31: For $n = 25, m = 9$, testing accuracy of reconstructed numeral “0”

USER	Accuracy of reconstructed image (%) for $n = 25, m = 9$	Average Accuracy for $n = 25, m = 9$
P1	99.9999	99.9999
P2	99.9999	
P3	99.9999	
P4	99.9999	
P5	99.9999	
P6	99.9999	
P7	99.9999	
P8	99.9999	
P9	99.9999	
P10	99.9999	

TABLE 5.32: For $n = 25, m = 9$, testing accuracy of reconstructed numeral “7”

USER	Accuracy of reconstructed image (%) for $n = 25, m = 9$	Average Accuracy for $n = 25, m = 9$
P1	99.9999	99.9999
P2	99.9999	
P3	99.9999	
P4	99.9999	
P5	99.9999	
P6	99.9999	
P7	99.9999	
P8	99.9999	
P9	99.9999	
P10	99.9999	

TABLE 5.33: For $n = 25, m = 9$, testing accuracy of reconstructed numeral “2”

USER	Accuracy of reconstructed image (%) for $n = 25, m = 9$	Average Accuracy for $n = 25, m = 9$
P1	99.9999	99.9999
P2	99.9999	
P3	99.9999	
P4	99.9999	
P5	99.9999	
P6	99.9999	
P7	99.9999	
P8	99.9999	
P9	99.9999	
P10	99.9999	

TABLE 5.34: For $n = 25, m = 9$, testing accuracy of reconstructed numeral “2”

USER	Accuracy of reconstructed image (%) for $n = 25, m = 9$	Average Accuracy for $n = 25, m = 9$
P1	99.9999	99.9999
P2	99.9999	
P3	99.9999	
P4	99.9999	
P5	99.9999	
P6	99.9999	
P7	99.9999	
P8	99.9999	
P9	99.9999	
P10	99.9999	

TABLE 5.35: For $n = 25, m = 9$, testing accuracy of reconstructed numeral “8”

USER	Accuracy of reconstructed image (%) for $n = 25, m = 9$	Average Accuracy for $n = 25, m = 9$
P1	99.9999	99.9999
P2	99.9999	
P3	99.9999	
P4	99.9999	
P5	99.9999	
P6	99.9999	
P7	99.9999	
P8	99.9999	
P9	99.9999	
P10	99.9999	

TABLE 5.36: For $n = 25, m = 9$, testing accuracy of reconstructed numeral “ $\underline{4}$ ”

USER	Accuracy of reconstructed image (%) for $n = 25, m = 9$	Average Accuracy for $n = 25, m = 9$
P1	99.9999	99.9999
P2	99.9999	
P3	99.9999	
P4	99.9999	
P5	99.9999	
P6	99.9999	
P7	99.9999	
P8	99.9999	
P9	99.9999	
P10	99.9999	

TABLE 5.37: For $n = 25, m = 9$, testing accuracy of reconstructed numeral “ $\underline{5}$ ”

USER	Accuracy of reconstructed image (%) for $n = 25, m = 9$	Average Accuracy for $n = 25, m = 9$
P1	99.9999	99.9999
P2	99.9999	
P3	99.9999	
P4	99.9999	
P5	99.9999	
P6	99.9999	
P7	99.9999	
P8	99.9999	
P9	99.9999	
P10	99.9999	

TABLE 5.38: For $n = 25, m = 9$, testing accuracy of reconstructed numeral “6”

USER	Accuracy of reconstructed image (%) for $n = 25, m = 9$	Average Accuracy for $n = 25, m = 9$
P1	99.9999	99.9999
P2	99.9999	
P3	99.9999	
P4	99.9999	
P5	99.9999	
P6	99.9999	
P7	99.9999	
P8	99.9999	
P9	99.9999	
P10	99.9999	

TABLE 5.39: For $n = 25, m = 9$, testing accuracy of reconstructed numeral “c”

USER	Accuracy of reconstructed image (%) for $n = 25, m = 9$	Average Accuracy for $n = 25, m = 9$
P1	99.9999	99.9999
P2	99.9999	
P3	99.9999	
P4	99.9999	
P5	99.9999	
P6	99.9999	
P7	99.9999	
P8	99.9999	
P9	99.9999	
P10	99.9999	

TABLE 5.40: For $n = 25, m = 9$, testing accuracy of reconstructed numeral “५”

USER	Accuracy of reconstructed image (%) for $n = 25, m = 9$	Average accuracy for $n = 25, m = 9$
P1	99.9999	99.9999
P2	99.9999	
P3	99.9999	
P4	99.9999	
P5	99.9999	
P6	99.9999	
P7	99.9999	
P8	99.9999	
P9	99.9999	
P10	99.9999	

For $n = 26, m = 8$, again we have tested the accuracy of reconstructed Hindi numerals “०-५” by OHCR system. The results of this are shown in the following tables.

TABLE 5.41: For $n = 26, m = 8$, testing accuracy of reconstructed numeral “०”

USER	Accuracy of reconstructed image (%) for $n = 26, m = 8$	Average accuracy for $n = 26, m = 8$
P1	99.9999	99.9999
P2	99.9999	
P3	99.9999	
P4	99.9999	
P5	99.9999	
P6	99.9999	
P7	99.9999	
P8	99.9999	
P9	99.9999	
P10	99.9999	

TABLE 5.42: For $n = 26, m = 8$, testing accuracy of reconstructed numeral “7”

USER	Accuracy of reconstructed image (%) for $n = 26, m = 8$	Average accuracy for $n = 26, m = 8$
P1	99.9999	99.9999
P2	99.9999	
P3	99.9999	
P4	99.9999	
P5	99.9999	
P6	99.9999	
P7	99.9999	
P8	99.9999	
P9	99.9999	
P10	99.9999	

TABLE 5.43: For $n = 26, m = 8$, testing accuracy of reconstructed numeral “2”

USER	Accuracy of reconstructed image (%) for $n = 26, m = 8$	Average accuracy for $n = 26, m = 8$
P1	99.9999	99.9999
P2	99.9999	
P3	99.9999	
P4	99.9999	
P5	99.9999	
P6	99.9999	
P7	99.9999	
P8	99.9999	
P9	99.9999	
P10	99.9999	

TABLE 5.44: For $n = 26, m = 8$, testing accuracy of reconstructed numeral “3”

USER	Accuracy of reconstructed image (%) for $n = 26, m = 8$	Average accuracy for $n = 26, m = 8$
P1	99.9999	99.9999
P2	99.9999	
P3	99.9999	
P4	99.9999	
P5	99.9999	
P6	99.9999	
P7	99.9999	
P8	99.9999	
P9	99.9999	
P10	99.9999	

TABLE 5.45: For $n = 26, m = 8$, testing accuracy of reconstructed numeral “3”

USER	Accuracy of reconstructed image (%) for $n = 26, m = 8$	Average accuracy for $n = 26, m = 8$
P1	99.9999	99.9999
P2	99.9999	
P3	99.9999	
P4	99.9999	
P5	99.9999	
P6	99.9999	
P7	99.9999	
P8	99.9999	
P9	99.9999	
P10	99.9999	

TABLE 5.46: For $n = 26, m = 8$, testing accuracy of reconstructed numeral “ $\underline{4}$ ”

USER	Accuracy of reconstructed image (%) for $n = 26, m = 8$	Average accuracy for $n = 26, m = 8$
P1	99.9999	99.9999
P2	99.9999	
P3	99.9999	
P4	99.9999	
P5	99.9999	
P6	99.9999	
P7	99.9999	
P8	99.9999	
P9	99.9999	
P10	99.9999	

TABLE 5.47: For $n = 26, m = 8$, testing accuracy of reconstructed numeral “ $\underline{5}$ ”

USER	Accuracy of reconstructed image (%) for $n = 26, m = 8$	Average accuracy for $n = 26, m = 8$
P1	99.9999	99.9999
P2	99.9999	
P3	99.9999	
P4	99.9999	
P5	99.9999	
P6	99.9999	
P7	99.9999	
P8	99.9999	
P9	99.9999	
P10	99.9999	

TABLE 5.48: For $n = 26, m = 8$, testing accuracy of reconstructed numeral “6”

USER	Accuracy of reconstructed image (%) for $n = 26, m = 8$	Average accuracy for $n = 26, m = 8$
P1	99.9999	99.9999
P2	99.9999	
P3	99.9999	
P4	99.9999	
P5	99.9999	
P6	99.9999	
P7	99.9999	
P8	99.9999	
P9	99.9999	
P10	99.9999	

TABLE 5.49: For $n = 26, m = 8$, testing accuracy of reconstructed numeral “c”

USER	Accuracy of reconstructed image (%) for $n = 26, m = 8$	Average accuracy for $n = 26, m = 8$
P1	99.9999	99.9999
P2	99.9999	
P3	99.9999	
P4	99.9999	
P5	99.9999	
P6	99.9999	
P7	99.9999	
P8	99.9999	
P9	99.9999	
P10	99.9999	

TABLE 5.50: For $n = 26, m = 8$, testing accuracy of reconstructed numeral “५”

USER	Accuracy of reconstructed image (%) for $n = 26, m = 8$	Average accuracy for $n = 26, m = 8$
P1	99.9999	99.9999
P2	99.9999	
P3	99.9999	
P4	99.9999	
P5	99.9999	
P6	99.9999	
P7	99.9999	
P8	99.9999	
P9	99.9999	
P10	99.9999	

For $n = 27, m = 7$, we have tested the accuracy of reconstructed Hindi numerals “०-५” by OHCR system are showing in the tables as follows:

TABLE 5.51: For $n = 27, m = 7$, testing accuracy of reconstructed numeral “०”

USER	Accuracy of reconstructed image (%) for $n = 27, m = 7$	Average accuracy for $n = 27, m = 7$
P1	99.9988	99.9983
P2	99.9975	
P3	99.9994	
P4	99.9987	
P5	99.9983	
P6	99.9983	
P7	99.9983	
P8	99.9985	
P9	99.9977	
P10	99.9973	

TABLE 5.52: For $n = 27, m = 7$, testing accuracy of reconstructed numeral “7”

USER	Accuracy of reconstructed image (%) for $n = 27, m = 7$	Average accuracy for $n = 27, m = 7$
P1	99.9976	99.9980
P2	99.9984	
P3	99.9976	
P4	99.9976	
P5	99.9976	
P6	99.9977	
P7	99.9983	
P8	99.9991	
P9	99.9978	
P10	99.9978	

TABLE 5.53: For $n = 27, m = 7$, testing accuracy of reconstructed numeral “2”

USER	Accuracy of reconstructed image (%) for $n = 27, m = 7$	Average accuracy for $n = 27, m = 7$
P1	99.9981	99.9978
P2	99.9980	
P3	99.9978	
P4	99.9987	
P5	99.9980	
P6	99.9980	
P7	99.9971	
P8	99.9971	
P9	99.9978	
P10	99.9974	

TABLE 5.54: For $n = 27, m = 7$, testing accuracy of reconstructed numeral “3”

USER	Accuracy of reconstructed image (%) for $n = 27, m = 7$	Average accuracy for $n = 27, m = 7$
P1	99.9978	99.9977
P2	99.9982	
P3	99.9986	
P4	99.9979	
P5	99.9967	
P6	99.9975	
P7	99.9979	
P8	99.9972	
P9	99.9976	
P10	99.9976	

TABLE 5.55: For $n = 27, m = 7$, testing accuracy of reconstructed numeral “8”

USER	Accuracy of reconstructed image (%) for $n = 27, m = 7$	Average accuracy for $n = 27, m = 7$
P1	99.9991	99.9984
P2	99.9988	
P3	99.9987	
P4	99.9990	
P5	99.9986	
P6	99.9978	
P7	99.9982	
P8	99.9979	
P9	99.9976	
P10	99.9984	

TABLE 5.56: For $n = 27, m = 7$, testing accuracy of reconstructed numeral “ $\underline{4}$ ”

USER	Accuracy of reconstructed image (%) for $n = 27, m = 7$	Average accuracy for $n = 27, m = 7$
P1	99.9992	99.9980
P2	99.9980	
P3	99.9991	
P4	99.9970	
P5	99.9970	
P6	99.9980	
P7	99.9973	
P8	99.9987	
P9	99.9978	
P10	99.9976	

TABLE 5.57: For $n = 27, m = 7$, testing accuracy of reconstructed numeral “ $\underline{5}$ ”

USER	Accuracy of reconstructed image (%) for $n = 27, m = 7$	Average accuracy for $n = 27, m = 7$
P1	99.9977	99.9975
P2	99.9974	
P3	99.9977	
P4	99.9971	
P5	99.9985	
P6	99.9973	
P7	99.9974	
P8	99.9971	
P9	99.9976	
P10	99.9975	

TABLE 5.58: For $n = 27, m = 7$, testing accuracy of reconstructed numeral “6”

USER	Accuracy of reconstructed image (%) for $n = 27, m = 7$	Average accuracy for $n = 27, m = 7$
P1	99.9979	99.9977
P2	99.9979	
P3	99.9974	
P4	99.9971	
P5	99.9980	
P6	99.9977	
P7	99.9978	
P8	99.9975	
P9	99.9979	
P10	99.9973	

TABLE 5.59: For $n = 27, m = 7$, testing accuracy of reconstructed numeral “7”

USER	Accuracy of reconstructed image (%) for $n = 27, m = 7$	Average accuracy for $n = 27, m = 7$
P1	99.9993	99.9978
P2	99.9987	
P3	99.9971	
P4	99.9968	
P5	99.9982	
P6	99.9970	
P7	99.9981	
P8	99.9969	
P9	99.9976	
P10	99.9979	

TABLE 5.60: For $n = 27, m = 7$, testing accuracy of reconstructed numeral “५”

USER	Accuracy of reconstructed image (%) for $n = 27, m = 7$	Average accuracy for $n = 27, m = 7$
P1	99.9971	99.9973
P2	99.9970	
P3	99.9974	
P4	99.9973	
P5	99.9975	
P6	99.9972	
P7	99.9972	
P8	99.9987	
P9	99.9967	
P10	99.9968	

We have obtained the average accuracies from the above tables’ data for the different order of moment and repetition of order. We have testing the accuracy of Hindi numerals written by 10 different users (P1, P2, P3, P4, P5... P10) for each and every order of moment (n) with repetition of order (m). Each and every user could draw different shape based Hindi numerals that caused the variation in accuracy of reconstruction but due to Zernike moment’s invariance property we could obtain the maximum accuracy of reconstruction at some higher order of moment and repetition of order.

The comparison of average accuracies of Hindi numerals “०-५” with different order of moment (n) and repetition of order (m) is given in the table below:

TABLE 5.61: Comparison of average accuracies of Hindi numerals for “०-९”

Order of moment	Repetition of order	०	१	२	३	४
[n]	[m]	(%)	(%)	(%)	(%)	(%)
23	7	99.9997	99.9998	99.9998	99.9998	99.9998
24	10	99.9992	99.9992	99.9991	99.9991	99.9994
25	9	99.9999	99.9999	99.9999	99.9999	99.9999
26	8	99.9999	99.9999	99.9999	99.9999	99.9999
27	7	99.9983	99.9980	99.9978	99.9977	99.9984

Order of moment	Repetition of order	५	६	७	८	९
[n]	[m]	(%)	(%)	(%)	(%)	(%)
23	7	99.9998	99.9998	99.9997	99.9998	99.9998
24	10	99.9992	99.9990	99.9990	99.9991	99.9990
25	9	99.9999	99.9999	99.9999	99.9999	99.9999
26	8	99.9999	99.9999	99.9999	99.9999	99.9999
27	7	99.9980	99.9975	99.9977	99.9978	99.9973

From the above table, one can observe that the maximum accuracy of reconstruction is achieved when order of moment (n) is 25 and repetition of order (m) is 9. As such, we can make the use of this moment and repetition order, that gives 99.9999% accuracy, for Online Handwritten Character Recognition system for Hindi numerals from “०-९”.

5.2 Summary

In this chapter, we have trained our OHCR system by different order of moment (n) with different repetitions (m) and measure the reconstruction accuracy of reconstructed image using Zernike moment for Hindi numerals “०-५”. The order of moments and repetition of orders that gives the accuracy more than 99.99% are given below:

(a) $n = 23, m = 7$

(b) $n = 24, m = 10$

(c) $n = 25, m = 9$

(d) $n = 26, m = 8$

(e) $n = 27, m = 7$

In the next step, we have decided the accuracy of reconstruction of the Hindi numerals using 10 different users (P1, P2, P3, P4, P5... P10), at each order of moment and repetition of order as given above and checked the best reconstructed order of moment at which we have obtained maximum accuracy.

We have now decided upon the best value of order of moment (n) and repetition of order (m) for the different users and it has been noted that standard order of moment (n) = 25 and repetition of order (m) = 9 gives the maximum accuracy of reconstruction 99.9999% for the proposed Online Handwritten Character Recognition system for Hindi numerals from “०-५”.

Chapter 6

CONCLUSIONS AND FUTURE RESEARCH

In this chapter, we first summarize the study carried out in this thesis. Then, the concluding remarks are given and the future research directions are further discussed.

6.1 CONCLUSIONS

Online handwriting recognition is a difficult problem, not only because of the great amount of variations in human handwriting, but also, because of the overlapping of the neighboring characters. Recognition strategies heavily depend on the nature of the data to be recognized. Since handwritten characters could be of various shapes and size, the recognition process needs to be much efficient and accurate to recognize the characters drawn by different online users.

This thesis is an attempt towards the evaluation of the performance of the online handwritten Hindi numerals character recognition system from the perspective of their invariance property to image transformations including scale, translation and rotation using Zernike Moments. The invariance property of Zernike moment was used in the recognition of online handwritten Hindi numerals. Various image samples were collected from different users online and their Zernike moments were calculated. The accuracy of reconstructing the image for Hindi numerals from “ \odot - $\frac{F}{Z}$ ” at various order of moments (n) and repetition of order (m) was calculated and found to be maximum at order of moment (n) = 25 with repetition of order (m) = 9. The accuracy of reconstruction was found to be 99.9999 %.

Therefore, an order of moment (n) = 25 with repetition of order (m) = 9 could be proposed as the best combination for reconstruction of Hindi numerals from “ \odot - $\frac{F}{Z}$ ”.

6.2 Future Research

In this thesis, the main focus of research was on the Hindi numerals' image from the view of the character image transformations of scale, translation and rotation. The accuracy of reconstruction have been tested and obtained as 99.9999% for 10 different user's handwritten Hindi numerals, so for obtaining higher accuracy more number of test users can be taken. The evaluation of the invariance property of Hindi numerals under the transformation at real time will be of great interest for the future research, because it is very common that the angle of the handwritten strokes can be changed during the character image acquisition. Computation of the angle range of the handwritten stroke to keep the property of the Hindi numerals invariant will make the two dimensional model based character object recognition system more flexible and reliable.

Another research interest will be on the character images degraded or blurred by various reasons. In this thesis, we researched different shapes of handwritten numerals by different users, but still the character image acquisition process could be affected by other factors such as looping of shapes, gesture of hands and atmospheric surrounding. This feature of Zernike moment can be used in Hindi characters recognition and also in multilingual character recognition. Exploration of the blur-invariant Hindi numerals image features based on these character image invariants or some other algorithms will make the recognition systems more reliable and robust.

REFERENCES

- [1] Abu-Mostafa, Y. S. and D. Psaltis, "Image normalization by complex moments," IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. PAMI-7, No.1, pp.46-55, January 1985.
- [2] Africa, Nafiz, Student Member, IEEE and Fatos T. Yarman-Vural, Senior member, IEEE," An overview of character recognition focused on Offline handwriting ", C 99-06-C-203.
- [3] Agarwal, Dr. Suneeta, Kumar, Vikas, "Online Character Recognition", Third Int. conference on Information Technology and application(ICITA'05).
- [4] Al-Omari, Faruq, "Handwritten Indian Numerals Recognition System using Template Matching Approaches", 0-7695-1165-1/01, 2001 IEEE.
- [5] Andreadis, I., Kotoulas, L.,"Image Analysis using moments", Laboratory of Electronics, Democritus University of Thrace, Greece.
- [6] Bajaj, Reena, Dey, Lipika and Chaudhury, "Devnagari numeral recognition by combining decision of multiple connectionist classifiers", Vol.27, part 1, February 2002, pp.59-72, India.
- [7] Capar, Abdulkerim, Tasdemir, Kadim, Ozlem, Kilic, Muhittin GOKMEN, "A Turkish Handprint Character Recognition System".
- [8] Flusser, Jan, "Moment Invariants in Image Analysis", Trans. on Engineering, Computing and Technology V11 February 2006 ISSN, 1305-5313.
- [9] Gonzalez, R. and Woods, R. E., Digital Image Processing, Prentice Hall, 2002.

[10] Hong, J.I. and Landay, J.A. SATIN: A Toolkit for Informal Ink-based Applications. CHI Letters: ACM Symposium on UIST, 2 (2), 63-72.

[11] Hse, Heloise and Newton, A. Richard “Sketched Symbol Recognition using Zernike Moments”, ICPR’2004, proc. of 17th Int. conference in Pattern Recognition, 1051-4651

[12] Hu, J. “HMM Based On-Line Handwriting Recognition”, IEEE Trans. On Pattern Analysis and Machine Intelligence, vol. 18, pp. 1039-1045, 1996.

[13] Hu, M. K., “Visual pattern recognition by moment invariants,” IRE Trans. Inform. Theory, Vol.IT-8, pp.179-187, February 1962.

[14] Jaeger, S., M. Nakagawa, C. L. Liu, “Comparing Online Recognition of Japanese and Western Script in Preparation for Recognizing Multi Language Documents”, 8th Int. workshop on Frontiers in Handwriting Recognition, Niagara-on-the-Lake, pp 84-89, 2002.

[15] Jain, Anil K., Fellow, IEEE, and Douglas Zongker, ”Representation and Recognition of Handwritten Digits Using Deformable Templates”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.19, No.12 December 1997.

[16] Jia-Xiong, P., Bin, Ye (2002),”Invariance Analysis of Improved Zernike Moments”, Journal of Optics A: Pure and Applied Optics, vol. 4, pp. 606-614.

[17] Kumar, Satish and Singh, Chandan, “A study of Zernike Moments and its use in Devnagari Handwritten Character Recognition”, Proc. of Int. conf. on Cognition and Recognition, pp. 514-520.

[18] Khotanzad, A. and Hong, Y.H. (1990), “Invariant Image Recognition by Zernike Moments”, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 12(5).

- [19] Khotanzad, A. and Hong, Y.H., "Rotation Invariant Image Recognition using Features Selected via a Systematic Method", *Pattern Recognition*, 23 (10), 1089-1101.
- [20] Liao, S. and Pawlak, M. (1998), "A Study of Zernike Moment Computing", *Lecture Notes in Computer Science*, pp. 394-401.
- [21] Liao, S. and Pawlak, M. (1998), "On the accuracy of Zernike Moment for Image Analysis", *IEEE Trans. On Pattern Analysis and Machine Intelligence*, Vol. 20, No. 12, December.
- [22] Liao, S. and Pawlak, M. (1997), "Moment Invariants for Image analysis with Application to Image Reconstruction", *IEEE Conference on Communications, Power, and Computing*, pp. 157-162.
- [23] Mukundan, R., Raveendran, P. and Chong, C-W. (2003), "A Comparative Analysis of Algorithms for Fast Computation of Zernike Moments", *Pattern Recognition*, vol. 36, pp. 731-742.
- [24] Mukundan, R., P. Raveendran and C-W. Chong (2003), "Translation Invariants of Zernike Moments", *Pattern Recognition*, vol. 36, pp. 1765-1773.
- [25] Mukundan, Ramakrishnan, "A New Class of Rotational Invariance Using Discrete Orthogonal Moments", *Proceedings of 6th ISATED Int. conference, SIGNAL AND IMAGE PROCESSING*, Aug 23-25, 2004, Honolulu, Hawaii, USA.
- [26] Petriu, Emil M., and Chen, Qing, "Optical Character Recognition for Model based Object Recognition Applications", *Univ. of Ottawa, ON, Canada, IEEE'2003*.

- [27] Plamondon, R. and S. N. Srihari (2000), "On-Line and Off- Line Handwriting Recognition: A Comprehensive Survey", IEEE trans. Pattern Analysis and Machine Intelligence, vol. 22 (1), pp. 63-84.
- [28] Prismall, Stuart P., Nixon, Mark S. and Carter, John N., "On Moving Object Reconstruction By Moments", Image, Speech and Intelligent Systems, Dep't. Of E & C, Univ. of Southampton, SO17 1BJ, BMVC 2002.
- [29] Prismall, Stuart P., Nixon, Mark S. and Carter, John N., "Accurate Object Reconstruction by Statistical Moments", Univ. of Southampton, UK.
- [30] Suen, C.Y., S. Mori and M. Berthod (1980), "Automatic Recognition of Hand printed Characters -The State of the Art", "Proceedings of the IEEE, vol. 68(4).
- [31] Teague, M.R., "Image Analysis via the General Theory of Moments", Journal of the Optical Society of America, 70 (8), 920-930.
- [32] Teh, C. and Chin, R.T. (1988), "On Image Analysis by Methods of Moments", IEEE Transaction on Pattern Analysis and Machine intelligence vol.10 (11).
- [33] Wee, Chong-Yaw, Paramesran, Raveendram, Takeda, Fumiaki,"New computational methods for full and subset Zernike moments", Information Sciences 159(2004) 203-220.

Appendices A

BINARY IMAGES

Binary images are images that have been quantized to two values, usually denoted 0 and 1, but often with pixel values 0 and 255, representing black and white. Binary images are used in many applications since they are the simplest to process, but they are such an impoverished representation of the image information that their use is not always possible. However, they are useful where all the information you need can be provided by the outline of the object and when you can obtain the outline of that object easily.

Some sample application domains include

- Identifying objects on a conveyor, for example, sorting chocolates,
- Identifying orientations of objects, and
- Interpreting text.

Sometimes the output of other image processing techniques is represented in the form of a binary image, for example, the output of edge detection can be a binary image (edge points and non-edge points). Binary image processing techniques can be useful for subsequent processing of the output images. Binary images are typically obtained by threshold a grey level image.

Pixels with a grey level above the threshold are set to 1 (equivalently 255), while the rest are set to 0. This produces a white object on a black background (or vice versa, depending on the relative grey values of the object and the background). Of course, the 'negative' of a binary image is also a binary image, simply one in which the pixel values have been reversed.

Analysis of Binary Images

We wish to determine various attributes of the objects in the scene with the aim of using these to identify the objects and to determine their position and orientation. We define the characteristic function of an object in an image to be

$$b(x, y) = \begin{cases} =1 & \text{for points on the object} \\ =0 & \text{for background points} \end{cases}$$

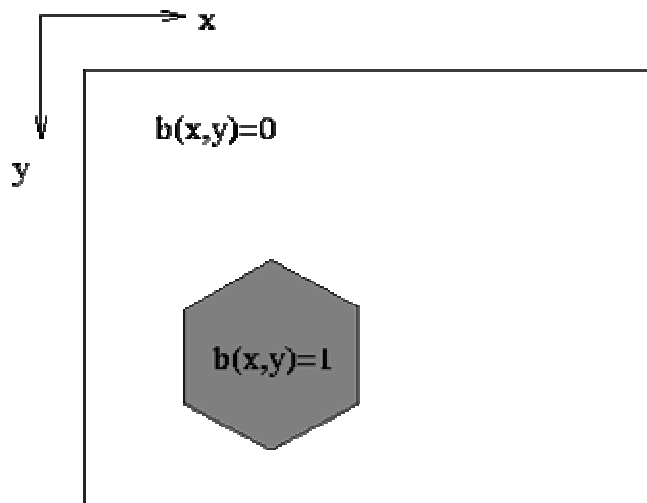


Figure 4: A binary image

Now consider the image to be continuous (that is, to have infinite resolution). The *area* is given by the 0th moment of the object:

$$A = \iint b(x, y) \, dx dy$$

The *centre of mass*, denoted by (\bar{x}, \bar{y}) is given by the 1st moments of the object:

$$\bar{x} = \frac{\iint x b(x, y) \, dx dy}{\iint b(x, y) \, dx dy} \quad \text{and} \quad \bar{y} = \frac{\iint y b(x, y) \, dx dy}{\iint b(x, y) \, dx dy}$$

Generally we use the axis of minimum inertia of the object to identify the orientation. This is the axis of least 2nd moment.

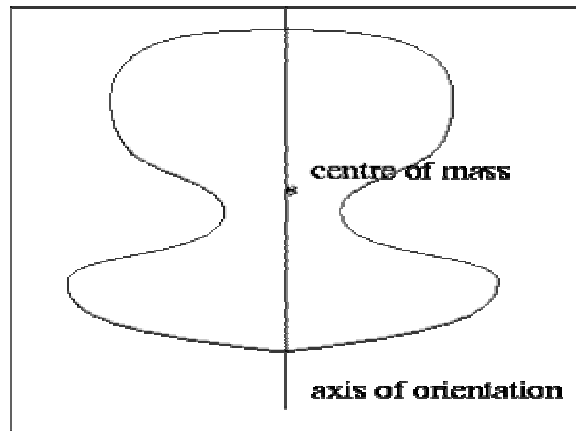


Figure 5: The 0th moment is the area of the object; the 1st moment gives the centre of mass; and the 2nd moments give the axes of orientation.

We want to find the line for which the integral

$$I = \iint r^2 b(x, y) dx dy$$

is a minimum, where r is the perpendicular distance from (x, y) to the line we want to find (see figure 6). The solution for the equation of this line is made more convenient if we parameterize the line using $x \sin \theta - y \cos \theta + \rho = 0$

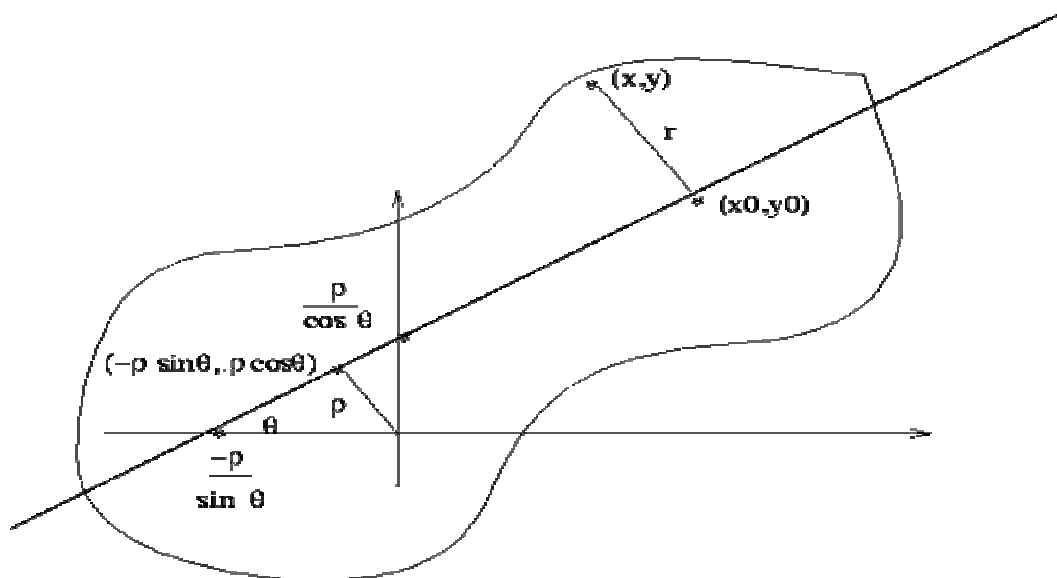


Figure 6: Parametrising a line by its angle with the x -axis and its distance from the origin

Thus we have to solve for the θ and ρ that will give the axis of least second moment. Why do we specify the line like this? Given this equation we can write parametric equations for points on the line as follows:

$$x_0 = -\rho \sin \theta + t \cos \theta$$

and

$$y_0 = \rho \cos \theta + t \sin \theta$$

Where t is the distance along the line from the closest point to the origin. Given a point (x, y) we need to find the closest point on the line so that we can calculate the distance r , which is given by (see figure 6) $r^2 = (x - x_0)^2 + (y - y_0)^2$. We substitute for x_0 and y_0 to get $r^2 = x^2 + y^2 + \rho^2 + 2\rho(x \sin \theta - y \cos \theta) - 2t(x \cos \theta + y \sin \theta) + t^2$. What value of t minimizes this expression?

Differentiating with respect to t and equating the result to 0 gives $t = x \cos \theta + y \sin \theta$. So we substitute this back into a parametric equation for x_0 and y_0 . This gives

$$\begin{aligned} x - x_0 &= x - (-\rho \sin \theta + (x \cos \theta + y \sin \theta) \cos \theta) \\ &= x + \rho \sin \theta - x \cos^2 \theta - y \sin \theta \cos \theta \\ &= x(1 - \cos^2 \theta) + \rho \sin \theta - y \sin \theta \cos \theta \\ &= x \sin^2 \theta + \rho \sin \theta - y \sin \theta \cos \theta \\ &= \sin \theta (x \sin \theta - y \cos \theta + \rho), \end{aligned}$$

and likewise for $y - y_0$. Thus r^2 can be written as $r^2 = (x \sin \theta - y \cos \theta + \rho)^2$. This line is the locus of points for which $r = 0$. Hence, by Parametrising the line in this manner we can obtain the distance from the line directly.

Solving for ρ and θ

We now want to find the ρ and θ that describe the line that minimizes distances of points in the object to that line, that is, that minimizes $I = \iint (x \sin \theta - y \cos \theta + \rho)^2 b(x, y) dx dy$.

We differentiate with respect to ρ and set the resulting expression to 0, giving

$$I = \iint (x \sin \theta - y \cos \theta + \rho)^2 b(x, y) dx dy = 0,$$

$$\text{Which is just } \iint x \sin \theta b(x, y) dx dy - \iint y \cos \theta b(x, y) dx dy + \iint \rho b(x, y) dx dy = 0$$

Multiplying and dividing by $A = \iint b(x, y) dx dy$ gives

$$A \left(\frac{\iint x \sin \theta b(x, y) dx dy}{\iint b(x, y) dx dy} - \frac{\iint y \cos \theta b(x, y) dx dy}{\iint b(x, y) dx dy} + \frac{\iint \rho b(x, y) dx dy}{\iint b(x, y) dx dy} \right) = 0$$

and this is just $A(\bar{x} \sin \theta - \bar{y} \cos \theta + \rho) = 0$, where (\bar{x}, \bar{y}) is the centre of mass. That is, the axis of minimum 2nd moment passes through the centre of mass.

We now have to find the value of θ that minimizes I . To do this, we change coordinates so that we are now working with respect to the centre of mass:

$$\begin{aligned} x' &= x - \bar{x} \\ y' &= y - \bar{y} \end{aligned}$$

This simplifies the equation of our line to $x \sin \theta - y \cos \theta + \rho = x' \sin \theta - y' \cos \theta$, and so

$$I = \iint (x' \sin \theta - y' \cos \theta)^2 b(x', y') dx' dy' \Rightarrow I = a \sin^2 \theta - b \sin \theta \cos \theta + c \cos^2 \theta,$$

Where

$$\begin{aligned} a &= \iint x'^2 b(x', y') dx' dy' \\ b &= 2 \iint x' y' b(x', y') dx' dy' \\ c &= \iint y'^2 b(x', y') dx' dy' \end{aligned}$$

The constants a , b , and c are called the *second moments*. If we use the substitutions

$$b \sin \theta \cos \theta = \frac{b}{2} \sin 2\theta \text{ and } \cos 2\theta = 2 \cos^2 \theta - 1, \text{ then}$$

$$\begin{aligned} I &= a(1 - \cos^2 \theta) + c \cos^2 \theta - \frac{b}{2} \sin 2\theta \\ &= a + (c - a) \cos^2 \theta - \frac{b}{2} \sin 2\theta \\ &= a + \frac{(c - a)}{2} \cos 2\theta + \frac{(c - a)}{2} - \frac{b}{2} \sin 2\theta \end{aligned}$$

Thus $I = \frac{1}{2}(c + a) - \frac{1}{2}(a - c) \cos 2\theta - \frac{b}{2} \sin 2\theta$. We differentiate this expression with

respect to θ and set the result to 0, obtaining $(a - c) \sin 2\theta - b \cos 2\theta = 0$, which implies

$$\tan 2\theta = \frac{b}{a - c}, \text{ unless } b = 0 \text{ and } a = c. \text{ Thus } \frac{\sin^2 2\theta}{\cos^2 2\theta} = \frac{b^2}{(a - c)^2}, \text{ which implies that}$$

$$\frac{\sin^2 2\theta}{1 - \sin^2 2\theta} = \frac{b^2}{(a - c)^2}. \text{ This is just a quadratic in } \sin^2 2\theta, \text{ so we see that it has solution}$$

$$\sin 2\theta = \frac{\pm b}{\sqrt{b^2 + (a - c)^2}}, \text{ and } \cos 2\theta = \frac{\pm(a - c)}{\sqrt{b^2 + (a - c)^2}}. \text{ When one chooses the positive}$$

solution, this minimizes I ; when one chooses the negative solution, this maximizes I . In the case that $b=0$ and $a=c$ we see that I is unaffected by the direction of axis of orientation, that is, our object is rotationally symmetric.

The ratio $\frac{I_{\min}}{I_{\max}}$ gives us some idea of how rounded the object is. This ratio will be 0 for a line and 1 for a circle.

Appendices B

HU INVARIANT SET

The non-orthogonal centralized moments are translation invariant and can be normalized with respect to changes in scale. However, to enable invariance to rotation they require reformulation. Hu described two different methods for producing rotation invariant moments. The first used a method called principal axes, however it was noted that this method can break down when images do not have unique principal axes. Such images are described as being rotationally symmetric. The second method Hu described is the method of absolute moment invariants and is discussed here.

Hu derived these expressions from algebraic invariants applied to the moment generating function under a rotation transformation. They consist of groups of nonlinear centralized moment expressions. The result is a set of absolute orthogonal (i.e. rotation) moment invariants, which can be used for scale, position, and rotation invariant pattern identification. These were used in a simple pattern recognition experiment to successfully identify various typed characters. They are computed from normalized centralized moments up to order three and are shown below I_n , n^{th} Hu invariant moment:

$$I_1 = m_{20} + m_{02}$$

$$I_2 = (m_{20} - m_{02})^2 + 4m_{11}^2$$

$$I_3 = (m_{30} - 3m_{12})^2 + (3m_{21} - m_{03})^2$$

$$I_4 = (m_{30} + m_{12})^2 + (m_{21} + m_{03})^2$$

$$I_5 = (m_{30} - 3m_{12})(m_{30} + m_{12}) \left[(m_{30} + m_{12})^2 - 3(m_{21} + m_{03})^2 \right] + \\ (3m_{21} - m_{03})(m_{21} + m_{03}) \left[3(m_{30} + m_{12})^2 - (m_{21} + m_{03})^2 \right]$$

$$I_6 = (m_{20} - m_{02}) \left[(m_{30} + m_{12})^2 - (m_{21} + m_{03})^2 + 4m_{11}(m_{30} + m_{12})(m_{21} + m_{03}) \right]$$

Finally a skew invariant, to help distinguish mirror images, is:

$$I_7 = (3m_{21} - m_{03})(m_{30} + m_{12}) \left[(m_{30} + m_{12})^2 - 3(m_{21} + m_{03})^2 \right] + \\ (m_{30} - 3m_{12})(m_{21} + m_{03}) \left[3(m_{30} + m_{12})^2 - (m_{21} + m_{03})^2 \right]$$

These moments are of finite order, therefore, unlike the centralized moments they do not comprise a complete set of image descriptors. However, higher order invariants can be derived. It should be noted that this method also breaks down, as with the method based on the principal axis for images which are rotationally symmetric as the seven invariant moments will be zero.