Segmentation of Offline Printed and Handwritten Mathematical Expressions

Manisha Bharambe
Associate Professor
MES Abasaheb Garware College

ABSTRACT
Mathematical expression recognition is an active research field and it becomes a challenging problem in the field of Optical character recognition.The fundamental problem of mathematical expression recognition system is the Off-line Printed expression recognition. One of the difficulties of handwritten mathematical symbol recognition lies in the variability of the symbols, different fonts in addition to the recognition of other language characters. The segmentation is the most important phase in the recognition of the expression. This paper deals with efficient segmentation technique to segment logical mathematical expressions with subscripts. In this paper, the database of 288 printed expressions and 960 handwritten expressions using logical symbols was developed. The proposed algorithm was tested on the handwritten and the printed expression database and the results are quite promising.

Keywords
Optical Character Recognition, Printed and Handwritten, Logical Mathematical Expressions, Segmentation.

1. INTRODUCTION
Mathematical expression recognition is an important problem about pattern recognition, because mathematical expression (ME) is an essential part of scientific literature [12]. Mathematical symbols set is very huge about 2000 symbols, so commonly used keyboard input is not sufficient. The inputting and editing of mathematical expressions have been difficult due to their non-linear structure. Mathematical symbols can be written beside, above, or below in different sizes, fonts, typefaces, and font sizes used [14]. In recent year research towards recognition of handwritten mathematical symbols and expressions is getting increasing attention. The ME contains more than a single character such as logical symbols, brackets, and English alphabets. The segmentation of ME results in isolated characters. To achieve efficiency of classification, the image should be properly segmented. Accuracy of ME recognizer heavily depends on segmentation phase. Segmentation is a difficult task in the handwritten ME due to the following reasons:ME structure is a 2-Dimentional structure, containing overlapping characters, i.e. the neighboring characters are written such that they share the area of others in a symbol block.Touching characters reduces the rate of segmentation.The characters in the ME have different spatial relationship like subscripts.The broken characters may be present in the ME due to writing style and poor quality of paper.ME may contain characters one inside the other, top and bottom of other character, for example, =, i, j, !. To segment such characters is complex task.The paper proposed a new approach of segmentation of subscripts and character having multiple components.

2. LITERATURE REVIEW
Hans-Jurgen Winkle et al proposed research on online segmentation and recognition in mathematical expressions. Preprocessing is done by removing slant and pre-recognition is done for separating the symbols “Dot", “Minus" and “Fraction" from the remaining symbol of the alphabet which have ambiguity, requires contextual knowledge.
Ahmad Awal et al. (2009) proposed the method for segmentation of online ME using number of strokes. The strokes were traced between pen down and a pen lift. Each symbol contained several strokes. The segmentation was carried out by grouping the number of strokes belonging to the same symbol. The difficulty in this approach was delayed strokes of symbol while writing, also grouping of strokes is a complex problem. They have proposed a framework allowing a simultaneous segmentation, recognition, and interpretation and achieved 84.8% recognition rate. Xue-Dong Tian et al. (2006) described a new efficient method of segmentation for segmenting symbols in ME using projection profile cut as well as connected component labeling. They presented recursive projection profile cutting for segmentation to merge small segments into one symbol block. A recognition threshold is defined to accept the recognition result, and if is not accepted by the recognition threshold, then the symbol block was re-segmented. Then connected component labeling was used to segment the symbol blocks. This method does not work for touching characters. The detailed algorithm was not given, though the outline for the same was stated. The system given by Francisco Alvaro et al.(2013) using recognition of printed mathematical expressions, segmentation was done manually. They have segmented into isolated symbols manually from the scanned images of dataset UW-III. Also they were not considered spatial relationship of symbols and no segmentation method was proposed.

3. DATA COLLECTION
For the proposed work, data is collected from different users by different handwritten style. The database of ME is not publically available, therefore the database is developed by collecting data from different writers. A4 sheets are used for data collection. Data is collected from twenty writers from different fields, each expression is written by each writer 10 times. Ten logical symbols are used to prepare the ME. We have used the logical expressions for the experimentation. The Fig.1 shows the samples of handwritten expressions from the scanned image. Datasheets of 48 handwritten expressions written by 20 writers were scanned and stored as grayscale images. Each expression was manually cropped and stored in jpg format, results in the dataset of 960 expressions. Database of 960 handwritten expressions and 288 printed expressions were collected. The printed expressions were collected from books by scanning the book pages. The Fig 2 shows the samples of printed expressions images. We have used Boolean algebra, preposition logic, number theory, electronics and Discrete mathematics books to collect 288 printed logical expressions. All these expressions were individually cropped and saved as jpg image.

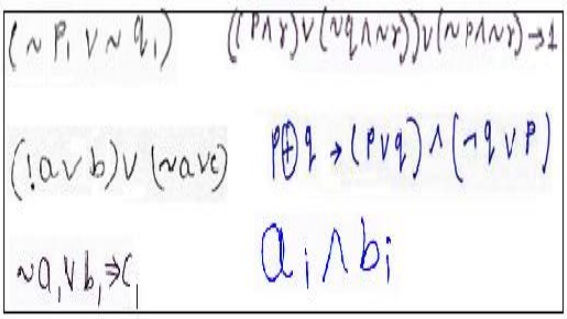

Fig 1: Handwritten Expressions

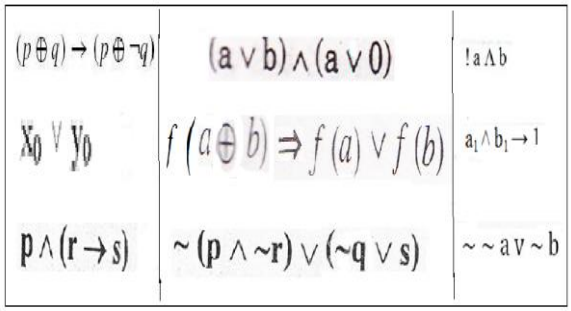

Fig 2: Printed Expressions

4. EXPRESSION PREPROCESSING

Handwritten images of expressions written on A4 sheet are scanned and store in .jpeg format. The image is RGB image. RGB image is converted into grayscale image. Grayscale image is converted into binary image, using a suitable threshold value by Otsu's method. The noise is removed from the image using median filter.The morphological operation image opening is used to separate touching symbols, dilation is used to join unconnected pixels..Image thinning is used to reduce space while storing topological information of an image.

5. SEGMENTATION

In the segmentation stage, the image of a sequence of characters is decomposed into sub-images of individual characters. In the proposed work, the pre-processed input image is segmented into isolated characters and symbols by assigning a number to each character using a labeling process. Each individual character is uniformly resized into 64x64 pixels for extracting its features.The connected component labeling method was used for segmentation. The Label matrix *L* was used to store the output of connecting components and *N* was used to store the number of components. *V* = <*xl*, *yl*, *xw*, *yw*> was used to define the bounding box (smallest rectangle region), where < *xl*, *yl*> specifies the upper left corner of the bounding box, and < *xw*, *yw*> specifies the width of the bounding box (BB) along x-direction and y-direction. The vector $C = < x_c , y_c >$ specifies the center of the bounding box, where x_c =horizontal coordinate of the center and y_c = vertical coordinate of the center as shown in Figure 3.

5.1 Algorithm for Segmentation

The segmentation experimentation was performed using Matlab. The algorithm is given below:

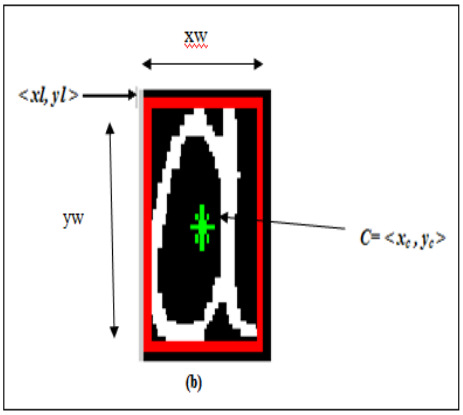

Fig 3: Structure of character

5.1.1 Algorithm : Segmentation 1

Input: Expression image in gray scale.
Output: Isolated characters of an expression.

Method:

1. Read the image of expression and apply median filter to gray scale image to eliminate noise.

2. Binarize the image to obtain foreground as 0 and background as 1.

3. Crop the image, and store height of the image, *r*.

4. Invert the image, to obtain the expression as 1 and background as 0, Apply morphological operation to close the gaps between the pixels.

5. For *i* =1 to *N* (*N* = Number of characters in the expression)

Obtain the connecting components of the image with labeled L.

6. Repeat steps 6 to 11 for N times to segment each character from the expression using BB.

Obtained the values of BB *(xl, yl, xw, yw*) using properties of the image region and store these values in vector *V* as follows:

Properties=regionprops(L,'BoundingBox','Area');

V =cat(1,properties.BoundingBox);

7.Obtained area of the region by ***area=[properties.Area];***

8. Process the image of an expression to merge the multiple connected components using vector ***V*** and ***area*** as in the algorithm '***Segmentation2*'.**

9. Find the Centroid, *C* of each BB.

10: Identify the subscripts of an image by using ***subscript-decision rule***.

11.Label the each component as 'sb' for subscripts of image and 'up' for the component other than the subscripts.

12. Obtained segmented characters.

13. Obtain label structure of each character with three fields: (class label, sb/up, index)

Subscripts decision rule:
If yl>= r/2±2 and yw<=r/2 ± 2 and y_c >=r/2, then it is a subscript.
Where, r = height of the expression image.
If subscript, then label of sub-image of subscript is 'sb' else 'up'.

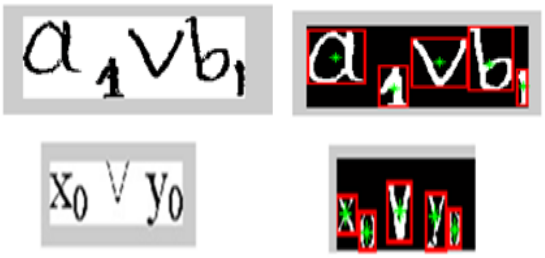

Fig 4: Segmentation of handwritten and printed images with subscripts

5.1.2 Algorithm: Segmentation2

This algorithm is used to merge the characters with multiple connected components such as i, j and ! as shown in the Fig 5 and Fig 6.

Input: BB of each character of the image.
Output: BB after merger.
Method:
1. Read each component of the image till the end of the expression.
2. If (abs (*xl* of i^{th} *component* - *xl* of $(i+1)^{th}$ *component*) < 30 and *area of* $(i+1)^{th}$ *component < 60 then*
xl = xl of i^{th} *component*
yl = yl of $(i+1)^{th}$ *component*
xw = xw of i^{th} *component* + *xw* of $(i+1)^{th}$ *component*
yw = yw of i^{th} *component* + yw of $(i+1)^{th}$ *component*+ 20
3. Store the values *(xl, yl, xw, yw)* of new BB for segmentation.

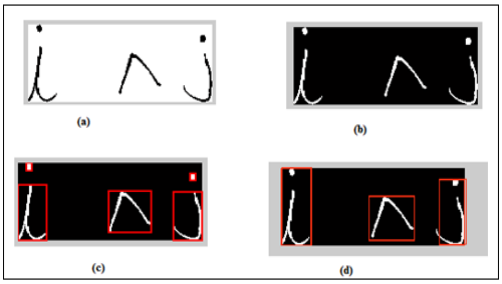

Fig 5: Segmentation of the character having multiple component a) Binary image b) Inverted image c) BB of 5 components d) BB after merging

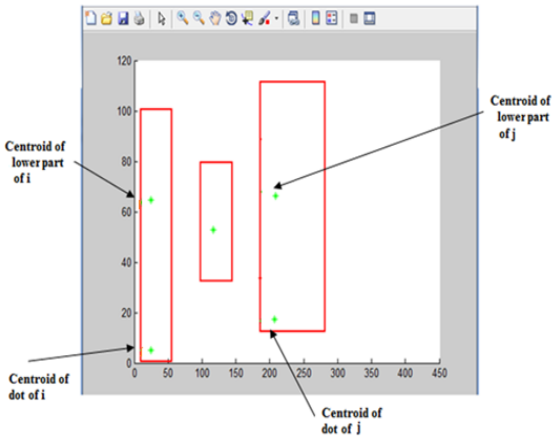

Fig 6: BB and centroid of expression i^ j

6. EXPERIMENTAL RESULT

In the handwritten ME segmentation, the segmentation accuracy depends on the writing of the writer, i.e. proper space between the characters. An experiment was carried out on 35 printed expressions. The number of occurrences of each expression varies from 5 to 15 times with different fonts, results in total 288 printed expressions. An experiment was carried out on 48 handwritten expressions. Each expression was written by 20 different writers, results in 960 expressions. The segmentation rate for handwritten expressions is shown in the Table 1. The average segmentation rate (SR) is the percentage of correctly segmented characters in the expressions. It is computed as,

$$SR = \frac{correctly\ segmented\ characters}{total\ number\ of characters\ (t)} * 100$$

where t= number of characters in expression number of occurrences of expression*

The Expression Segmentation Rate (ESR) is the percentage of correctly segmented expressions from the total number of expressions.

It was observed that the characters !, i, and j are correctly segmented and result of segmentation was improved by using the proposed segmentation method. The handwritten character $\Rightarrow$ reduces the segmentation rate relative to that of in printed expressions. The segmentation rate for printed expressions is shown in the Table 2.

7. CONCLUSION

From 960 handwritten expressions, 105 expressions were incorrectly segmented. From 288 printed expressions, 20 expressions were incorrectly segmented. The average segmentation rate of handwritten expressions and printed expressions was **89.06%** and **93.05%** respectively as shown in the Fig 7. The average segmentation rate of the printed expressions was higher relative to that of in handwritten expressions. The segmentation rate was reduced due to overlapping characters and touching characters. These segmented characters are inputted to the feature extraction phase, to recognize the character. The proposed method of segmentation was used to improve the segmentation rate, which in turn improve the recognition rate of the ME significantly.

8. ACKNOWLEDGMENT
The author is grateful to Dr. M. S. Prasad for his contributive support and encouragement during this work.

9. REFERENCES
[1] Ahmad Moritaser Awal, Harold Mouchere, Christian Viard Gaudin. Towards Handwritten Mathematical Expression recognition IEEE 978-07095, 2009

[2] Ahmad Montaser Awal, Harold Mouchere, Christian Viard Gaudin. The problem of Handwritten mathematical expression recognition. ISBN, 978-0-7695-4221-8,2010.

[3] Bharambe Manisha, "Recognition of Offline Handwritten Mathematical Expressions", International Journal of Computer Applications ISSN: 0975–8887,Vol. 108-No. 2 April 2014

[4] Bharambe Manisha, "Logical Symbol Recognition using Normalized Chain code and Density Features", International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Vol. 3, Issue-12, December 2014, pp: 619-62

[5] Hans Jurgen Winkler and Manfred Lang. On-Line Symbol segmentation and recognition inHandwritten mathematical expressions, 0-8186-7919-0/97,IEEE

[6] His-Jian Lee And J. Wang. Design of a mathematical expression recognition system, 0-8186-7128-9/95,IEEE

[7] Kang kim, Taik Rhee, Jae LEE. Utilizing consistency context for handwritten mathematical expression recognition. 978-0-7695-3725-2/2009 IEEE

[8] Kazuki Ashida, Masayuki Okamoto, Hiroki Imai, Performance Evaluation of a Mathematical Formula Recognition System with a large scale of printed formula images, Proceedings of the Second International Conference on Document Image Analysis for Libraries (DIAL'06),0-7695-2531- 8/06, 2006 IEEE

[9] Xue-Dong Tian, Hai-Yan Li, Xin-Fu Li. Research on symbol recognition for mathematical expressions. 0-7695- 2616-0/2006 ,IEEE.

[10] Xie,Xiaofang. On the recognition of handwritten mathematical symbols. Proquest NR39341,2008

[11] Francisco Álvaro, Richard Zanibbi, A Shape-Based Layout Descriptor for Classifying Spatial Relationships in Handwritten Math, 2013 ACM 978-1-4503-1789/4/13/09

[12] Qi Xiangwei Pan Weimin Yusup Wang Yang, The study of structure analysis strategy in handwritten recognition of general mathematical expression, International Forum on Information Technology and Applications, 978-0-7695-3600-2/09, 2009 IEEE

[13] Sanjay S. Garde, Pallavi V. Baviskar, K. P. Adhiya, Identification of Handwritten Simple Mathematical Equation Based on SVM and Projection Histogram, International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-3, Issue-2, May 2013.

[14] Anita Jindal,Renu Dhir,Rajneesh Rani, Diagonal Features and SVM classifier for Handwritten Gurumukhi Character recognition, International Journal of Advance Reasearch in Computer science and software engineering, Vol 2, Issue 5, May 2012. ITRPPR, 2010.

[15] Taik HeonRhee, JinHyungKim , Efficient search strategy in structural analysis for handwritten mathematical expression recognition, patternrecognition (ScienceDirect)0031-32,2009 Elsevier

10. APPENDIX

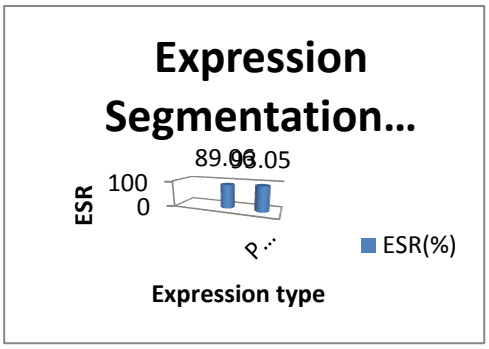

Fig 7: Expression Segmentation Rate

Table 1: Average Segmentation rate of handwritten and printed characters

Exp type	#exp	#characters	Correctly segmented Characters	Wrong segmented Characters	SR(%)
Handwritten without subscripts	660	6560	6267	293	95.54
Handwritten with subscripts	300	2080	1916	164	92.13
Printed	288	3559	3448	111	96.88

Table 2. Average Segmentation rate of handwritten and printed expressions

Exp type	#exp	#characters	Expressions segmented Correctly	Expressions segmented Wrongly	ESR (%)
Handwritten	960	8640	855	105	89.06
Printed	288	3559	268	20	93.05