Algorithm to Detect and Segment Gurmukhi Handwritten Text into Lines, Words and Characters

Rajiv Kumar and Amardeep Singh

***Abstract*—The output of a scanner is a non editable scanned text image. Though the text is visible but one can neither edit it nor make any change, if required. This provides a basis for the optical character recognition (OCR) theory. OCR consists of generally three major phase; pre processing after image acquisition, segmentation and recognition. The segmentation process is the most crucial phase. The output of this phase decides the outcome of recognition phase. If this output is right then recognition phase would give the right output otherwise not. In this paper, we provide an algorithm which is used to segment the scanned document image as a lines, words and characters. The coordinates of line detected are used to find the word position present in that line. Finally, these words position coordinates are used to find characters present in the word. To detect lines and words, one module is proposed which is used to find both. For character detection, the reverse engineering is used, i.e. one part is extracted from the word present in the line. This extracted part is checked whether it has some meaningful symbol (as per Gurmukhi script). If it has then the extracted part is marked and written in the file, otherwise the extracted part is readjusted to find the symbol. This overall concept was implemented, and got encouraging results.**

***Index Terms*— OCR, Segmentation, Gurmukhi, Handwritten, Feature, Water Reservoir, Line, Word.**

I. Introduction

Optical character recognition, usually abbreviated to OCR, is electronic translation of scanned images of handwritten, typewritten or printed text into machine-encoded text. Optical character recognition (OCR) is the process of converting scanned images of machine printed or handwritten text into a computer processable format. It involves computer software designed to translate images of text into machine printed editable text, or to translate pictures of characters into a standard encoding scheme representing them in ASCII or Unicode. If anyone scans a text document, one might want to use optical character recognition (OCR) software to translate image into text that can be edited. It is widely used to translate documents, article and books into electronic files, to computerize a record-keeping system in an office, or to publish the text on a website. OCR makes it possible to edit the text, search for a word or phrase, store it more compactly, display or print a copy free of scanning artifacts. In most OCR systems, character recognition performs on individual characters. In OCR process, after capturing the text image, it is passed through various phases. These are generally named as pre processing, segmentation, and then recognition to have a lexical meaning.

The last phase, that is, recognition phase heavily depends upon the out come of segmentation i.e. the correct recognition is possible only if segmentation is correct. Segmentation refers to the process of partitioning a digital image into multiple segments. But to define character segmentation we can say it is a technique, which partitions images of lines or words into individual characters. It is an operation that seeks to decompose an image of a sequence of character into sub-images of individual symbols. According to a survey of vast literature done by Casey. and Lecolinet.[1996], there are three pure strategies for segmentation, plus numerous hybrid approaches that are weighted combination of these three. The elementary strategies are:

The Classical Approach in which segmentations are identified based on character like properties. This process of cutting up the image into meaningful components is called dissection. Recognition Based Segmentation, in which the system searches the image for components that match classes in alphabet. Holistic Methods, in which the system seeks to recognize words as a whole, thus avoiding the need to segment into characters. There are many strategies for segmentation, which are combinations of one or more of above three pure ones. Hybrid methods can be represented as weighted combinations of these lying at points in the intervening space.

Manuscript received March 16, 2011 and revised on May 25, 2011.
Rajiv Kumar, Thapar University,(email: rajiv.patiala@gmail.com)
Amardeep Singh, Pbi University (email: amardeep_dhiman@yahoo.com)

II. Characteristics of Gurmukhi Script

A line of Gurmukhi script can be partitioned into three horizontal zones namely, upper zone, middle zone and lower zone as shown in Figure1. In the middle zone, consonants are present. The upper and lower zones may contain parts of vowel modifiers and diacritical markers.

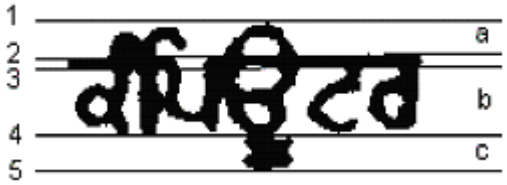

Fig. 1. a) Upper zone from line number 1 to 2, b) Middle Zone from line number 3 to 4, c) lower zone from line number 4 to 5

The upper zone denotes the region above the head line, where vowels reside, while the middle zone represents the area below the head line where the consonants and some sub parts of vowels are present. The lower zone represents the area below middle zone where some of vowels and certain half characters lie in the foot of consonants. The half characters in the lower zone frequently touch the above lying consonants in the above zone. There are many multi component characters in Gurmukhi script. A multi component character is a character that can decompose into isolated parts.Writing style of Gurmukhi script symbol is

from left to right. In Gurmukhi, there is no concept of upper or lowercase characters. There are 41 consonants and 12 vowels in Gurmukhi script alphabet. These are shown in Figure 2 (a) and (b).

Fig. 2. a) Gurmukhi Consonants

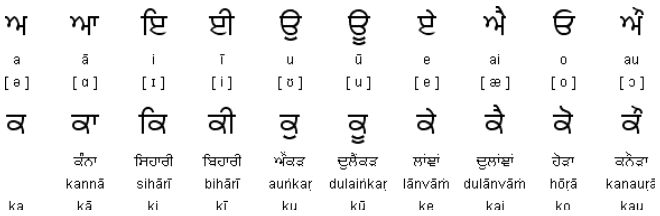

Fig. 2. b) Vowels and diacritics (Laga Matra)

The Gurmukhi script is a two dimensional composition of consonants, vowels and half characters which require segmentation in a vertical as well in horizontal direction. Thus the segmentation of Gurmukhi text calls for a two dimensional analysis instead of commonly used One dimensional analysis for Roman script.

III. Preprocessing

It is a process of representing the scanned image for further processing. The raw data, depending on the data acquisition type, is subjected to a number of preliminary processing steps to make it usable in the descriptive stages of character analysis. Pre processing aims to produce data that are easy for the CR systems to operate accurately. Preprocessing is applied on the input binary document so that the effect of spurious noise can be minimized in the subsequent processing stages. It is supposed that height and width of document can be known easily. The image file is in grey scale. But we require two types of information either zero or one. For that purpose, we calculated the average of intensities of all the pixels present in the document image file. Then the intensity of each pixel is set as per the following rule:

if (pixel intensity < Average intensity) then pixel intensity = 0 else pixel intensity = 1

Scan every pixel of document and compare its intensity with the maximum intensity. If the intensity is equal to maximum intensity, store one in the array at that location, and if it is not equal store zero in the array. The quality of scanned image depends upon the scanner type too and it plays an important role in segmentation. We are using higher end scanner for the scanning purposes. But even if some impurities are introduced due to used paper quality or due to scanner quality then these are taken care by using anti windowing concept. As per this concept, first the area is searched with a window of width dw. If there is only few pixel and nothing is around the window then those intensities of present pixel values are set to zero. The words and characters are handwritten. Between any two lines and any two words there is a definite gap of minimum width. A line is supposed to have different words and the words are made up of one or more characters. The lines consisting of words are generally straight in nature. If there is any skew then present work may not work properly.

IV. Algorithm to Segment Line, Word and Character

Get the window coordinates. A pixel position is defined as a point which is taken as POINT { Int X;Int Y;}. For whole of the window, find areas of no pixel zones, store to some file. For two consecutive no pixel zones – find the minimum and maximum of X coordinates and find the minimum and maximum of Y coordinates. Write the min and max of X and Y in a file. The formal algorithm is given below:

Procedure to find Lines

- Read the Initial values of BMP, it will give the starting point of the file and the size of file. (// this will enable us to find starting and ending point of BMP file, having the handwritten image .)
- LOC = Starting Point of the image
- Do while (Not end of Input file)
  - Call Boundary_ Vertical (LOC as POINT, d as integer, X as integer, MaxY as integer, MinY as integer)
  - Call Horizontal Boundary (LOC as POINT, MaxY as integer, MaxX as integer, MinX as integer)
  - Write in OFILE, MinX,MinY, MinX, MaxY, MaxX, MaxY, MaxX, MinY
  - LOC.x = MinX and LOC.Y = MinY

  End of While

Procedure to find Words

- From the data file, get window coordinates of first two lines store to L1 and L2.
- While (L1 or L2 < > NULL)
  - Check the distance between the windows of L1 and L2. If it is less than a predefined value then merge these two windows of L1 and L2 to one large window (say) L1 and write the coordinates of L1 to the file and replacing L2.
  - If L2 is replaced then Set L2 = next line from file.
  - Set L1= L2 and L2 = next line from file
- End While
- Do while (Not All Lines are processed)
  - L = Get the Line from Input file
  - LOC.X=MinX of line L and LOC.Y = MinY of Line L
  - Do while (Not end of Window of L)
    - Call Boundary_Vertical(LOC, d, Y as integer, MaxX as integer, MinX as Integer)

Call Horizontal Boundary (LOC as POINT, MaxX as integer, MaxY as integer, MinY as integer)
Write the Window coordinates to file along with the coordinates of Line L
LOC.X = MaxX and LOC.Y = MinY
Enddo // end of inner While

- Enddo

Procedure to find the characters

- Get the Line window from Data File
- Repeat the following (immediate two) steps for all the lines written in the file.
- Get the window coordinates of word W_i of current line.
- Repeat the following for W_i
  - Moving along with horizontal line of word to get a region (say) R, enclosed between minimum pixels.
  - Send this region to classifier module to check whether it has some meaningful symbol or not.
  - If classifier module returns true then note down coordinates of region R to the output file along with the coordinates of line and word W_i , else readjust the coordinates of region R so that classifier module can send true.
- W_i = next word of current line.
- Stop

```
Boundary_Vertical (LOC, Dw, Xp, MaxY, MinY)
{
Set MaxY = LOC.Y
Set MinY = -9999
For J = LOC.X to X
        Yp = LOC.Y
            Do Until ((MaxY – Yp) > Dw)
                    If (Pixel = 1)
                       If(CCP=Noted PixelPosition)
                         If(MinY=-9999)Set MinY=CCP.Y
                If ((CCP.Y – MinY)<0) Set MinY = CCP.Y
                    If(CCP.Y – MaxY)>0 AND (CCP.Y –
MaxY) <Dw) Set MaxY = CCP.Y)
                    End if
                Yp = Yp + 1
            End of Until
NEXT J
}
```

Input parameter
SLOC is of POINT type and it would be MinX and MinY
Output parameter
MaxY is of Integer type
MinY is of Integer type
d can be a predefined value
or can be calculated as d = First no Pixel Zone Area – Second no pixel zone area
X Horizontal width of the image // can be obtained from the file information.

```
Boundary_Horizontal (LOC, MaxY, MaxX, MinX)
{
Set MinX = -9999
For I = LOC.Y to MaxY
        Xp = LOC.X
         For K = LOC.X to X
              If (Pixel = 1)
                 If (CRP = Noted PixelPosition)
                   If(MinX=-9999)Set MinX=CRP.X
                   ELSEIf ((MinX-CRP.X)>0) Set MinX=CRP.X
                     If((MaxxX – MinX )>0) Set MaxX = CRP.X)
                   End if
        Next K
NEXT I
}
```

V. Implementation and Result

The above idea is implemented using Visual Basic 6.0. The authors are good in Programming with Visual Basic 6.0. The file header as well as Info header, used to read the BMP file, is given below:

```
Private Type BITMAPFILEHEADER
    bfType          As Integer
    bfSize          As Long
    bfReserved1     As Integer
    bfReserved2     As Integer
    bfOhFileBits    As Long
End Type
```

```
Private Type BITMAPINFOHEADER
biSize              As Long
biWidth             As Long
biHeight            As Long
biplanes            As Integer
biBitCount          As Integer
biCompression       As Long
biSizeImage         As Long
biXPelsPerMeter     As Long
biYPelsPerMeter     As Long
biClrUsed           As Long
biClrImportant      As Long
End Type '40 bytes
```

After successfully implementing the concepts we got the outputs. Here is the sample of scanned image of document and processed document.

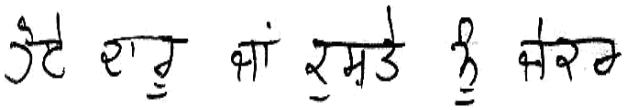

Fig. 3. Scanned Image of a Document

Fig. 4. Processed Document

The results obtained are summarized in the following tables:

TABLE I. ACCURACY FOR LINE SEGMENTATION

Document	No of Lines	Correctly Detected	Inaccurate segmentation	Accuracy
Document1	13	12	1	92.3%
Document2	16	15	1	93.75%
Document3	18	16	2	88.89%
Document4	30	28	2	93.33%

TABLE 2: ACCURACY FOR WORD SEGMENTATION

Document	No of Words	Correctly Detected	Inaccurate segmentation	Accuracy
Document1	68	62	6	91.17%
Document2	76	68	8	89.47%
Document3	95	83	16	87.36%
Document4	120	101	19	84.16%

TABLE 3: ACCURACY FOR CHARACTER SEGMENTATION

Document	No of Words	Correctly Detected	Inaccurate segmentation	Accuracy
Document1	89	84	5	94.38%
Document2	148	138	10	93.24%
Document3	223	209	14	93.72%
Document4	278	255	23	91.72%

VI. Conclusion

After implementing the concept, it was tested on different documents, the results obtained were encouraging. The lines

were detected with a great accuracy. The lines, which were having some characters in the lower zone, were interpreted almost correctly. To get the character, the coordinates of detected lines and words are used. For character segmentation process was divided in two part, (i) to get the segmented region R (ii) to check, if R has a meaningful symbol or not. This can be a reverse approach to ensure correct segmentation, i.e. if R does not have a meaningful symbol then R is readjusted. As per the data shown in the Table 3, there is certain incorrect segmentation too. After close analysis, we found that this is due to the shapes of characters. Certain characters in Gurmukhi script are combined in nature. But overall, results were good and encouraging.

REFERENCES

[1] R.G. Casey, and E. Lecolinet, "A Survey of Methods and Strategies in Character Segmentation". *IEEE Transactions onPattern Analysis and Machine Intelligence*, vol. 18, no.8, pp.690-706, 1996.

[2] S. Liang,, M. Shridhar, and M. Ahmadi, "Segmentation of Touching Characters in Printed Document Recognition". *Pattern Recognition*, vol.27, no.6, pp.825-840, 1994.

[3] Rajiv K. Sharma and Amardeep S. Dhiman, "Challenges in Segmentation of Text in Handwritten Gurmukhi Script". *Proceedings in BAIP 2010, CCIS 70*, Springer-Verlag Berlin Heidelberg, pp. 388–392, 2010

[4] Rajiv K. Sharma and Amardeep Singh, "Segmentation of Handwritten Text in Gurmukhi Script". *International Journal of Image Processing*, vol .2, no. 3, 2008.

[5] Rajiv K. Sharma and Amardeep Singh, "Detection and Segmentation of Handwritten Text in Gurmukhi Script using Flexible Windowing". *IJCTE*, vol 2, no. 3, pp. 329 – 332, 2010.

[6] Rajiv K. Sharma and Amardeep Singh, "Detection and Segmentation of Lines and Words in Gurmukhi Handwritten Text", *Proceedings in IEEE 2nd International Advance Computing Conference*, IEEE Xplore, 2010.

[7] *Antarpreet Kaur, Rajiv K. Sharma, and Amardeep Singh*, "A Hybrid Approach to Classify Gurmukhi Script Characters". *International Journal of Recent Trends in Engineering,*, vol 3, no. 3, pp 103-105, 2010.

[8] M. K. Jindal, G. S. Lehal, and R. K. Sharma. "Segmentation Problems and Solutions in Printed Degraded Gurmukhi Script". *IJSP*, vol 2, no. 4, 2005.

[9] G. S .Lehal and Chandan Singh. "Text segmentation of machine printed Gurmukhi script". *Document Recognition and Retrieval VIII, Proceedings SPIE*, USA, vol. 4307: 223-231, 2001

[10] Veena Bansal and R.M.K. Sinha. "Segmentation of touching and Fused Devanagari characters". *Pattern recognition*, vol. 35: 875-893, 2002.

[11] Giovanni Seni and Edward Cohen. "External word segmentation of off – line handwritten text lines". *Pattern Recognition*, vol. 27, no. 1, pp. 41-52, 1994.

[12] U. Pal and Sagarika Datta. "Segmentation of Bangla Unconstrained Handwritten Text". *Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR)*, 2003.

[13] Devasar, N. M., Madan, S., and Singh, H., "A Hybrid Approach to Character Segmentation of Gurmukhi Script Characters". *Proceedings of the 32nd Applied Imagery Pattern Recognition Workshop (AIPR'03)*, 2003.

[14] Perminder Singh, "A Technique for Preprocessing and Segmentation of Printed Text in Gurmukhi Script". *M.Tech.thesis submitted to Deptt. of Comp. Sc. & Engg., Punjabi University, Patiala.*, 1997.

Rajiv K. Sharma obtained his M.Tech (CSE) and is currently doing Ph.D. in Faculty of Engg. from the University College of Engineering, Punjabi University, Patiala, Punjab, India. At present, he is the Assistant Professor in the School of Mathematics and Computer Applications, Thapar University, Patiala. He has published several articles in international journals and conferences.

Amardeep Singh obtained his M.Tech (CSE) from the Punjabi University, Patiala, Punjab and Ph. D. from Thapar University, Patiala. He is Reader, in Department of Computer Engineering, University College of Engineering, Punjabi University, Patiala. He has published several articles in journals and conferences.

Rajiv K. Sharma obtained his M.Tech (CSE) and is currently doing Ph.D. in Faculty of Engg. from the University College of Engineering, Punjabi University, Patiala, Punjab, India. At present, he is the Assistant Professor in the School of Mathematics and Computer Applications, Thapar University, Patiala. He has published several articles in international journals and conferences.

Amardeep Singh obtained his M.Tech (CSE) from the Punjabi University, Patiala, Punjab and Ph. D. from Thapar University, Patiala. He is Reader, in Department of Computer Engineering, University College of Engineering, Punjabi University, Patiala. He has published several articles in journals and conferences.