Localization, Extraction and Recognition of Text in Telugu Document Images

Atul Negi
Department of CIS
University of Hyderabad
Hyderabad 500046, India
atulcs@uohyd.ernet.in

K. Nikhil Shanker
Department of CSE
Mahatma Gandhi
Institute of Technology
Hyderabad, India
nikhil.shanker@acm.org

Chandra Kanth Chereddi
Department of CSE
College of Engineering
Osmania University
Hyderabad 500007, India
chandra-kanth@ieee.org

Abstract

In this paper we present a system to locate, extract and recognize Telugu *text. The circular nature of* Telugu *script is exploited for segmenting text regions using the* Hough Transform. *First, the* Hough Transform *for circles is performed on the* Sobel *gradient magnitude of the image to locate text. The located circles are filled to yield text regions, followed by* Recursive XY Cuts *to segment the regions into paragraphs, lines and word regions. A region merging process with a bottom-up approach envelopes individual words. Local binarization of the word MBRs yields connected components containing glyphs for recognition. The recognition process first identifies candidate characters by a zoning technique and then constructs structural feature vectors by* cavity analysis. *Finally, if required, crossing count based non-linear normalization and scaling is performed before template matching. The segmentation process succeeds in extracting text from images with complex* Non-Manhattan *layouts. The recognition process gave a character recognition accuracy of 97%-98%.*

1 Introduction

Text isolation and extraction from varied backgrounds is a difficult problem where regional properties of text and non-text regions are used to separate them. Previously Negi et al. [4] presented a template matching based OCR system for Telugu using fringe distances where the difficulties of recognition of Telugu script were introduced. However, they did not address complex layout analysis, which occurs in local magazines and newspapers. Hence, there is a need to develop a complete OCR system to address this concern. In this paper, we present a novel approach exploiting the circular nature of Telugu script to locate and extract Telugu script in a document image. We also present a different approach to recognize the isolated Telugu text, using zoning and structural feature vectors. We define *cavities* as a new set of structural features for recognition, which are commonly seen in Telugu orthography.

The text segmentation system uses the fact that readable text has a reasonable contrast with the background. Here we use the gradient magnitude of the image to extract areas of high contrast. The circular nature of Telugu text allows its isolation using the Hough Transform[1] on the Sobel gradient magnitude[1] of the image and leads to the identification of text regions. One of the difficult aspects of segmenting Telugu script is that of merging connected components of the half consonants, *vatthus* (see Negi et al for introduction to Telugu orthogoraphy) with their respective characters. The present approach yields good grouping of text components into words. The isolated word regions are binarized by the technique proposed by Wu et al.[5].

Template matching as proposed by Negi et al [4] has a recognition complexity linear to the template set size, which is large in the case of Telugu. Instead, here we use a candidate search and elimination technique based on statistical and structural features for recognition. First, the pixel density in different zones is used for a candidate search. The candidates obtained, if found *inconclusive,* are analyzed for the presence of cavities. A successful perfect match concludes the search. If not so, a controlled nonlinear shape normalization stage based on crossing counts is used to produce a glyph for template matching with the L_2 metric (Euclidean distance). The recognized glyph code is returned to the segmentation system from which the word can be rebuilt. An example image for the text extraction with a complex layout is shown in section 6.

2 The Segmentation System

The scheme for segmentation is depicted in figure 1. The document image is the input. The image is analyzed in 2 phases - Top-Down Analysis and Bottom-Up Analysis. In the Top-Down Analysis phase, the main regions of interest

Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003)
0-7695-1960-1/03 $17.00 © 2003 IEEE

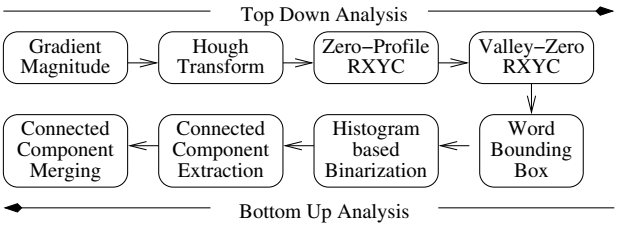


Figure 1. Scheme for Segmentation

are zeroed in. This is followed by the Bottom-Up Analysis phase in which the regions of interest are grown so as to envelope complete words and split into individual characters, which are, along with their associated *vatthus* passed to our OCR. Each phase comprises of 5 unique stages, which operate on the result of the previous stage. In the forthcoming sections, we explain each stage of the system in more detail.

3 Top Down Analysis

In the Top-Down Analysis Phase, we try to approach the image as a whole and isolate the regions of interest.

Gradient Magnitude: First, the gradient magnitude of the image is computed to obtain contrasting regions in the image. This is because readable text usually appears with sufficient contrast with the background. The gradient magnitude of the image is binarized by applying a threshold, depending on the clarity of the text in the document. This image is passed through a median filter which is useful in removing noise which occurs in images with half-toned backgrounds.

Hough Transform: Next, the Hough Transform for circles is applied on the gradient magnitude of the image from the previous step to obtain the circular gradients which is a very characteristic feature of Telugu text. The radii of the circles to be detected is generally proportional to the font size. For documents scanned at 300 dpi with 10 point font, a radii range of 5 to 10 was found to be appropriate. Each detected circle is filled to obtain the regions of interest. This filling process also connects close lying characters which are parts of a single word as well as close lying lines, which are part of a paragraph.

Zero-Profile RXYC: Once the regions of text have been identified, an adaptation of the RXYC [3]approach is used. We define this approach as the Zero-Profile RXYC. Here, we segment the regions by using the horizontal and vertical projection profiles of the image obtained in the previous step. We divide the region wherever a zero projection profile is found. This separates all the individual paragraphs, since the previous step joins any text lines which are in close proximity.

Valley-Zero RXYC: In the next stage, we use another variation of the RXYC approach which we call the Valley-Zero RXYC. Here, we segment the regions using RXYC based on projection profiles again. The region is split horizontally if a valley is found in the horizontal profile. This separates all the lines in the paragraph into individual lines. However, the regions are split vertically only at a zero vertical profile. This causes the words within a paragraph to be separated. This also allows us to separate inlaid paragraphs embedded within a paragraph.

Once that is done, a pass of the Vertical Zero-Profile RXYC is applied. It is similar to the Zero-Profile RXYC process described above but here, the regions are split vertically. The horizontal splitting process is not applied. The process is applied on the gradient magnitude to separate any regions which might have been connected during the Hough transform stage but form disconnected gradients. Next, a region cleanup process is performed to eliminate regions which have extremely small or extremely large sizes. The region cleanup process is discussed in a later section. At this point, our Top Down Analysis is complete. The resulting regions found in this phase are the discrete words in the document.

4 Bottom Up Analysis

In this phase, we employ various methods to grow the detected regions so that they completely envelope the words in the document image.

Word Bounding Boxes: In the first stage of this phase, we generate a bounding box around each region of interest using the gradient magnitude of the image. This allows us to capture the exact bounding box of each word.

Binarization: Once the bounding box of each word is obtained, each region in the bounding box is binarized using a simple but effective histogram based binarization method proposed by [5]. This word region is binarized using the computed threshold to yield a binary image of the region thus highlighting the text. This works well for text in reverse video also.

Connected Component Extraction: The resulting binarized regions are again passed through a Vertical Zero-Profile RXYC process which separates any unnecessary elements which might have been brought into the region. This includes lines in a table which lie in close proximity to the text and thus may be enveloped within the text regions during the Top Down Analysis phase. The resulting regions are again passed through the region cleanup process which was applied earlier.

Connected Component Merging: Now, each word region is split into its connected components, using the binarized image, and their bounding boxes are determined. The resulting connected components are checked for close proximity vertically. This is done to cater to the *vatthus* of the

Telugu script, which are completely disconnected with their associated character and occur somewhat lower in the line. Each word is scanned for connected components in a horizontal line running through the center of the word region. When a connected component is found, the area above and below the connected component is examined for any *vatthu*. Finally, the character and its associated *vatthus* are passed to the OCR in a specific order.

5 Region Cleanup

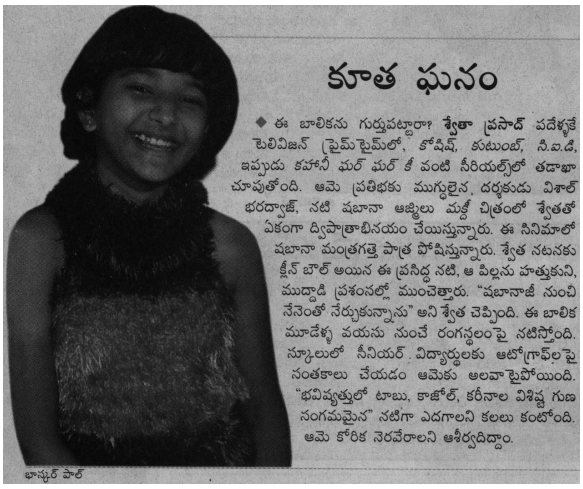
The region cleanup process, which is used as a post processing stage for some major stages, is performed by eliminating regions of very small or very large dimensions. This constraint can be applied by considering the sizes of all regions and computing the mean (μ) and standard deviation (σ) of their dimensions to arrive at a size criterion. Any region is eliminated if its height or width is outside the bounds of the size criterion.

6 The OCR System

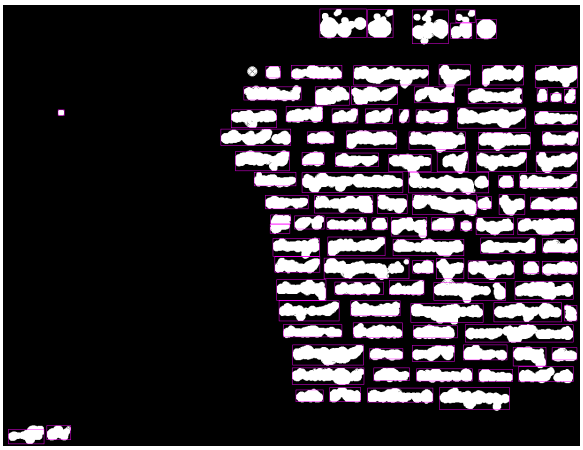
The OCR system employs a candidate search and elimination technique. The system uses both statistical features as well as structural features for recognition. This system has 3 different stages. The first stage uses the density of pixels in different zones for a candidate search. The candidates obtained from this stage, if found *inconclusive,* are passed to the cavity analysis stage where the presence and the position of cavities are analyzed. If the search is still inconclusive a template match based on the L_2 metric is performed, preceded by controlled nonlinear shape normalization based on crossing counts. At any stage, if a single perfect candidate match is found, its corresponding glyph code is returned and the OCR proceeds to recognize the next glyph.

6.1 Zoning

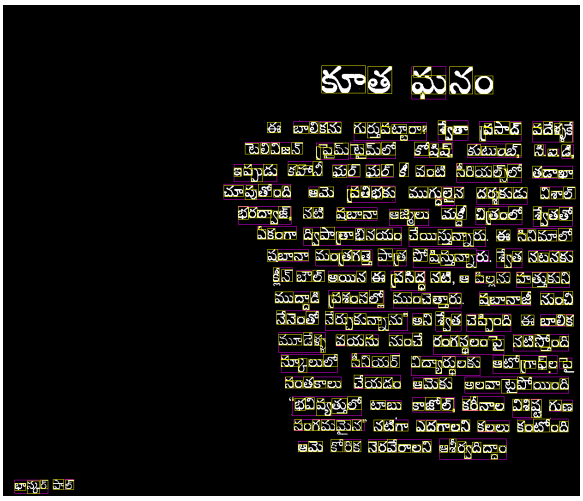
A 4 x 4 grid is superimposed on the input glyph. Then the percentage of the number of foreground pixels is computed to obtain a 16 (4x4) dimensional feature vector (figure 3) represented as (i_1,i_2,...,i_{16}) corresponding to grids from top left to bottom right in that order. A codebook of this zoning feature vector is precomputed from the training set. The feature vector of the input glyph is computed and searched for in the codebook to obtain *k=5* nearest neighbors based on Euclidean distance. After the best *k* candidate distances are computed, we analyze them to to prune the search. Glyphs which have a distance greater than 2.5 times the nearest neighbor distance (value determined empirically) are eliminated. The search concludes if only a



(a)



(b)



(c)

Figure 2. (a) The Original Image (b) Regions identified by the Hough Transform (c) Isolated text connected components

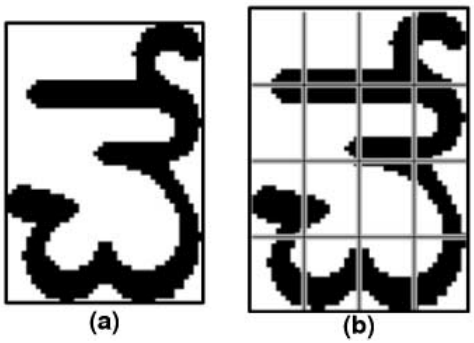


Figure 3. (a) Input glyph (b) Grid superimposed on glyph for zoning

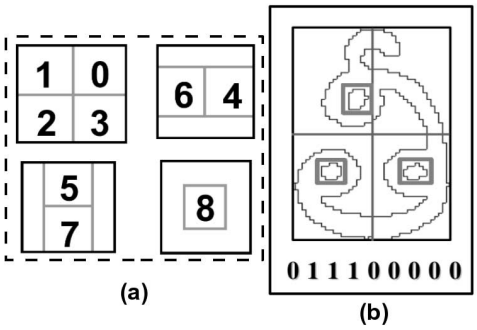


Figure 4. (a) Placement of overlapping quadrants in a glyph (b) Contour image and its cavity vector

single candidate survives the pruning. This stage is invariant to linear scaling. Rotational invariance is not sought as it is incompatible with the Telugu orthography.

6.2 Cavity analysis

Cavity analysis is a structural analysis stage. Many Telugu characters have "cavities" in them. The existence and position of these cavities is a structurally distinguishing feature. We are encouraged to use cavities as they allow greater discrimination between glyphs which otherwise would appear to be similar.

Cavities are detected by generating a contour of the linearly scaled glyph and performing a connected component separation on the contour image. This is because cavities get disconnected from the outer boundary in a contour image. The minimum bounding box of the cavity contour should have an area between threshold low and threshold high of the total glyph area, else it is discarded as being too small or too large (figure 4(b)). To determine the position of the cavity we divide the total glyph area into 9 overlapping quadrants. These are numbered sequentially from 0 to 8 (figure 4(a)). The existence of a cavity in these quadrants is shown by a boolean value generating a 9 bit vector, since more than one cavity in a quadrant is not possible.

For any input glyph, its bit vector is matched against the pre-computed bit vectors of the pruned candidates obtained from the previous stage. If there exists only one match then the search concludes else the results are passed onto the next stage. When no exact matches are found (which is possible if the input image is of poor quality or has breaks) a match with a relaxed condition is performed. In the relaxed condition a bit vector such as, (0 1 1 1 0 0 0 0 0) is considered as (x 1 1 1 x x x x x) where “x” is a don’t care condition. This is searching for the mere presence of a cavity and its absence is being ignored. Under very rare circumstances, if such a relaxed match also fails, we pass the pruned candidates from the zoning stage to the final stage.

6.3 Template Matching

The final stage of the OCR is a template matching stage. This stage has two steps internally. First is the nonlinear normalization stage where image scaling is performed. This enhances the favorable features of the input image. We use a modification to the crossing count based technique for normalization as proposed by Lee and Park[2].

Nonlinearity Control in Nonlinear Shape Normalization: Following the notation in [2] we normalize the input image $f(i,j)$ to produce $G(m,n)$ which is of a desired output size (64 x 64 pixels) using the projection of the crossing count statistic. We use the formulae $H(i) = \sum_{j=1}^{J} \overline{f(i,j-1)}.f(i,j) + a_H$ to calculate the horizontal crossing count and $V(j) = \sum_{i=1}^{I} \overline{f(i-1,j)}.f(i,j) + a_V$ to calculate the vertical crossing count. Here $f(i,0) = f(0,j) = 0$ and a_{H} and a_{V} are constants which affect the linearity of the statistical data.

Lee and Park use equal values of a_{H} and a_{V} to control nonlinearity when it becomes too strong. Their suggestion failed to produce proper results for the Telugu orthography. The solution is to find different values for a_{H} and a_{V} which are calculated dynamically based on the input image data.

In order to obtain favorable values for a_{H} and a_{V} we first calculate the standard deviation(σ) and mean(μ) of the $H(i)$ and $V(j)$ data assuming a_{H}= a_{V}= 0 . The ratio of standard deviation to mean is used as a measure of linearity. An upper limit is fixed for this ratio, represented by η. The a_{H} and a_{V} values are calculated based on this maximum allowable nonlinearity value (η).

$$a_H = [\,(\frac{\sigma_H}{\eta}) - \mu_H\,] \quad ; \quad a_V = [\,(\frac{\sigma_V}{\eta}) - \mu_V\,]$$

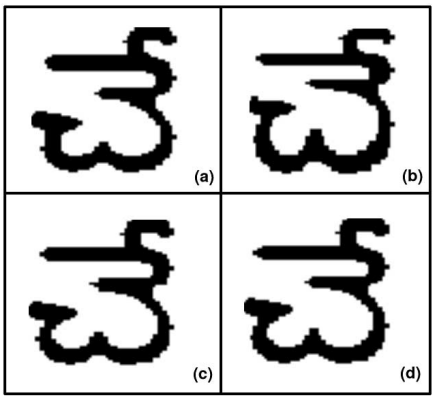


Figure 5. (a) Linear scaling (b) Nonlinear scaling (c) Nonlinear scaling η=0.04 (d) Nonlinear scaling η=0.06

The values of a_{H} and a_{V} are thus obtained by calculating the (σ) and (μ) of $H(i)$ and $V(j)$. These values are added to the existing $H(i)$ and $V(j)$ data. The input image $f(i,j)$ is then re-sampled based on the $H(i)$ and $V(j)$ statistics to obtain the output image $G(m,n)$. Re-sampling is performed using the same formula as proposed by Lee and Park.

The value of parameter η is critical. This value should be chosen very carefully based on experimental evaluation. It has a direct effect on the performance of the OCR system (figure 5). We have chosen to use the value η=0.06 as this gave the best overall result.

The scaled input (64x64) is now matched against the templates (normalized similarly) of the pruned results (typically less than 4) passed by the cavity analysis stage.

6.4 Minkowski (L_2) Distance Template Matching

We use the well known L_2(Euclidean) distance to measure similarity between templates and the input glyph. We call a *forward* distance as the sum of distances from *each* foreground pixel of the scaled input image to the *nearest* foreground pixel in the template. Similarly a *backward* distance is computed in the same way except its computed from the template to the input glyph. The summation of the forward L_2 distance and backward L_2 distance is taken as the final comparison metric, least being the best match. The template matching stage is used only when the previous stages are inconclusive. Experimentally it is observed that very few input glyphs reach this stage, most are recognized in the previous stages.

7 Results and Future Work

The segmentation process was found to be tolerant to skew of about 5 degrees. This can be attributed to the Recursive XY Cuts based approach. Also, it was found to succeed in segmenting complex "non-Manhattan layouts" such as those examples shown in the Pink Panther system [6]. Such complex layouts are commonly found in Telugu magazines.

The OCR gave a good accuracy in the range of 97%-98%. In a test set of 1500 glyphs taken from a magazine (*India Today*, in Telugu) the OCR gave correct results for about 1463 glyphs. In majority of the mismatches the OCR had the correct match in the top 3 candidates. In another experiment using a different training set a recognition rate of 94% was obtained on the same test set. The larger errors here were because of the complete change in the structural style of a few glyphs (like non existence of a few cavities). The OCR design is such that it facilitates the use of a multi font training set. A more extensive analysis of the OCR is being performed to further fine tune all the parameters.

The true challenge for the Telugu OCR would be for it to work in a multi-font environment. In Telugu orthography, the styles suffer a large variation in normal typesetting, further decorative fonts exhibit a very large variation. Future systems can be trained to keep this in view. Layout restoration is one of the extensions that is being planned for the future. Here we would like to point out that picture and photographic image regions are being ignored in the present system. In future these could be identified and processed separately.

Acknowledgments: Atul Negi acknowledges the support of the Resource Center for Indian Language Technology Solutions (Telugu), Ministry of Communications and Information Technology, Govt. of India, New Delhi.

References

[1] R. Gonzalez and R. Woods. Digital image processing, Addison-Wesley, Reading. *MA*, 92:1992.

[2] S. Lee and J. Park. Nonlinear shape normalization methods for the recognition of large-set handwritten characters. *Pattern Recognition*, 27(7):895–902, July 1994.

[3] G. Nagy and S. Seth. Hierarchical representation of optically scanned documents. *Proceedings 7th ICPR*, 84:347–349.

[4] A. Negi, C. Bhagavati, and B. Krishna. An OCR System for Telugu. In *Proceedings Sixth ICDAR, Seattle USA 2001*, 2001.

[5] V. Wu, R. Manmatha, and E. M. Riseman. Textfinder: An automatic system to detect and recognize text in images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1224–1229, 1999.

[6] B. Yanikoglu and L. Vincent. Pink panther: A complete environment for ground truthing and benchmarking document page segmentation. *Pattern Recognition*, 31(9):1191–1204, September 1998.