IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-6, June 2018 E-ISSN: 2347-2693

Enhancing Skew Detection and Correction methods to Improve Optical Braille Recognition

Vishwanath Venkatesh Murthy [1*], M. Hanumanthappa [2]

[1] Department of Computer Science and Applications, Bangalore University, Bengaluru, India
[2] Department of Computer Science and Applications, Bangalore University, Bengaluru, India

Available online at: www.ijcseonline.org

Accepted: 23/Jun/2018, Published: 30/Jun/2018

Abstract—Visually impaired persons use Braille language as they can't access digital or print media. Braille document is mostly embossed on Braille plate and are captured and processed using Optical Braille Recognition (OBR) system. OBR converts the Braille document into natural text. The quality of converted documents largely affected by the skew generated during the document scanning. Skew angle detection is one of the most important requirements in preprocessing stage of OBR. The introduction of skew is unavoidable in digitized document. Skew may get introduced while scanning the Braille plate due to human error. In this paper, two different algorithms based on Sobel edge detector, Gaussian filter method and another histogram and blob properties are used to estimate skew angle and its correction. The preprocessing, skew detection and skew correction algorithms are evaluated on MATLAB. The obtained results are compared to identify the efficient algorithm

Keywords—Braille, Gaussian Filter, Histogram Equalization, image segmentation, Linear regression, Skew angle, OBR, preprocessing, sobel, skew detection

I. Introduction

The Braille writing language used primarily used by blind. The Braille cell pattern consists of a collection of six raised dots that represent one cell that identifies one text character or one word. The Braille plates are exist with either one or double sided embossing [6]. To avoid these plates from getting deteriorate, they are scanned processed and converted to natural text using OBR system. During scanning the image produced may introduce noise due to uneven light, irregular pixel, image skew or slant, deprived dots, irregular spacing or uninvited dots [7].

Skew refers to the tilt in the resultant Braille image introduced due to human mistakes during scanning. Any segmentation methods that are applied on OBR require image with correct angle. [6] Hence detecting image skew is a very important step. Skew angle is detected using several methods like 1. Hough Transform, 2. Gaussian Filter 3.Nearest neighbor, 4.Projection profile, 5.Principle component analysis and 6.Sobel edge operator. Many different skew detection techniques have been discussed in literature survey [6].

The main aim of this study was to present an efficient method which can accurately detect the skew angle in the scanned Braille plate image and apply required rotation to derive its corrected image that can further used as input for the cell segmentation process. The accuracy of skew detection and correction of the algorithms are compared as a conclusion of the paper [8].

This research paper is structured in 4 segments. Section-I gives the introduction of the Braille document, preprocessing and segmentation modules of OBR system. Section-II gives the literature survey of author who applied the different methods of skew detection and correction. In Section-III, the application and results of skew detection and skew correction is discussed. In section IV, the obtained results are compared. Finally the section-V concludes the paper.

II. Literature survey

Padmavathi. S [7] in her paper has used line fitting algorithm for skew angle detection. The author uses the Xmin and Xmax co-ordinates of the cell dots and corresponding values of y given as y (Xmin) and y (Xmax). The conditions of equation-1 and equation-2 are used to identify the clockwise or anti-clockwise angle [7].

$$y(\text{Xmin}) < y(\text{Xmax}) \qquad \text{-- (3)}$$

$$y(\text{Xmin}) > y(\text{Xmax}) \qquad \text{-- (3)}$$

The straight line is drawn between these selected dots. The selection of dots is made to reduce the square deviation of observed co-ordinates. The sum of squared error is given in equation-3 [7].

$$\sum ei2 = \sum (yi - y'i) \quad \text{-- (3)}$$

Where, $y'i = a + b.xi$, yi same as y and xi depend on x value at ith position. The 'b' represents the regression slope and 'a' is constant point where line crosses Y axis for X=0. The angle generated by the line along with X-axis is taken as the skew angle [7].

Jie Li in his paper has used the manually drawn rectangle of 15x10mm using the Braille cell properties [4]. These rectangles are positioned with scanned document to transform them to common reference point. Tilt angle is then estimated using the equation-4 and 5.

$$x_2 = \cos(\theta)(x_1 - x_0) - \sin(\theta)(y_1 - y_0) + x_0 \quad \text{-- (4)}$$

$$y_2 = \sin(\theta)(x_1 - x_0) + \sin(\theta)(y_1 - y_0) + y_0 \quad \text{-- (5)}$$

Here (x0, y0) represents centre point of rotation, whereas (x1, y1) are the original co-ordinate values and (x2, y2) are the new co-ordinate values. Rotation angle is given by θ with positive clockwise angle.

Zhenfei Tai [8] in his paper had applied Randon Transform with an angle bound of 0◦ to 90◦ with Δθ step. This gives an energy intensity image in all possible degrees [8]. The output image gives the approximate lines indicating the skew angle. Considering I_R (θ, *l*) as randon transform image with 'θ' being an angle and '*l*' representing offset from origin [8]. The power sum *sum_S* of I_R for every angle is defined in equation-6

$$\text{sum\_}S(\theta) = \sum_{l}^{h} || I_R(\theta, l)||^2 \quad \text{-- (6)}$$

sum_$S(\theta)$, will have high values of skew angle since integral of high intensity shows at most of the dots. Finally the projected skew angle $\hat{\theta}$ is presented in equation-7 [8].

$$\hat{\theta} = arg_{\theta}^{max} sum\_S(\theta) \quad \text{-- (7)}$$

After calculating $\hat{\theta}$ the author simply removes the skew by rotating the image by -$\hat{\theta}$.

Huaxun Zhang [1] in his paper has used the linear regression method which first extracts the image edge using segmentation that comes same as to straight line. He had obtained the unitary linear regression [1] model for this edge points as shown in eqation-8.

$$y = \beta_0 + \beta_1 x + \epsilon, \ E_{\epsilon} = 0, \ E_{\epsilon} = \sigma^2 < 1 \quad \text{-- (8)}$$

Further the author uses the least square method to find the angleβ_{τ}. The slope of the line is calculated with the equation-9. This calculated tilt angle is used for correcting the image [1].

$$\alpha_1 = arctan\beta_{\tau} \quad \text{-- (9)}$$

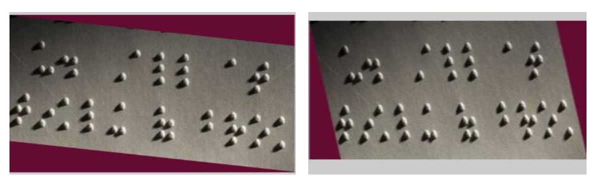
Joko Subur [2] in his paper applies the segmentation method that is designed to identify the area from where Braille cell and be read and processed. Formula used to find the area segmentation is given in equation-10, where Aseg represents segmentation area [2].

$$Aseg_i = \sum_{b=1}^{b} \sum_{k=1}^{k} [i = i + 1] \left(\sum_{m=b}^{b+3} . \sum_{n=k}^{k+2} [aXn][aYm]\right) \quad \text{-- (10)}$$

In the above equation, b and k represents the row and column whereas 'i' represents the Braille character counter value. aX and aY represents average values of X and Y coordinates. [2] The author has done the testing with pictures having different tilt angles and accuracy rate of recognition of characters is 89% for 1^0 tilt to 100% for 0.5^0 tilt [2].

III. SKEW DETECTION AND CORRECTION

Skew can be in clockwise or anticlockwise direction [4]. Image skew can be introduced due to human errors for two reasons: 1. Improper placement of Braille plate while scanning, 2. Braille cells itself are punched with some angle on to the plate [5].



(a) Clockwise skew (b) Anticlockwise skew

Figure-1: Types of Braille image skewness

We have applied two algorithms on skew detection and correction. The algorithm-1 uses the Sobel edge detector and Gaussian filter method. Gradient edge detection uses the high intensity peak values with steep change and marks it as edge pixel. It uses two kernel as x-direction and y-direction kernel [4].

Algorithm-1 for skew detection and removal

1. Input the color image
2. Obtain the Gray scale image
3. Compute Sobel Edge Detector on horizontal directions

4. Compute Sobel Edge Detector on vertical directions
5. Apply Gaussian Filter on calculated values
6. Plot all corner points
7. Apply Edge detection
8. Calculate the rotation angle
9. Rotate the image using this angle

The algorithm starts by reading the color image. The image is first converted into Grayscale. Grayscale image can be obtained using the MATLAB function as: grayImg = rgb2gray (newImg) or we can use the formula given in the equation-11.

$$\mathrm{gray} = (0.2989 * \mathrm{double}(\mathrm{im}(:,:,1)) + 0.5870 * \mathrm{double}(\mathrm{im}(:,:,2)) + 0.1140 * \mathrm{double}(\mathrm{im}(:,:,3)))/255; \quad \text{-- (11)}$$

The original color image is shown in figure-2 and the figure-3 shows the resultant gray scale image obtained by applying eutin-11.

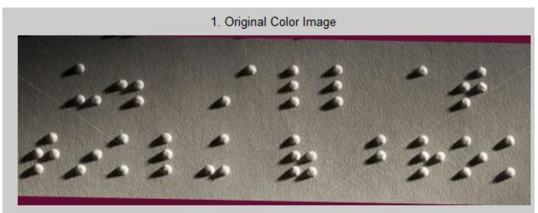


Figure-2: Original color image

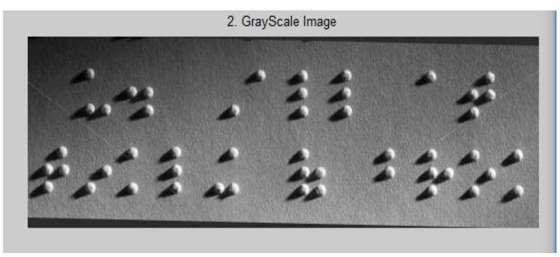


Figure-3: Gray scale image

The Sobel edge filters used on horizontal and vertical direction are given as, in x-direction xDir = [-1 0 1;-1 0 1;-1 0 1]; and in y-direction yDir = [1 1 1; 0 0 0;-1 -1 -1]; [5] The Gaussian filter on the above Sobel edge detected values are applied using the equation-12.

$$h = \text{ fspecial } ('\text{gaussian}', [7\ 7], 2); \quad \text{-- (12)}$$

This calculated 'h' values used in computations of edges [8]. The corner points are marked with the function corner (img, 100), where 100 represents the total corner plots that will be detected. The image obtained after corner point detection is shown in figure-4.

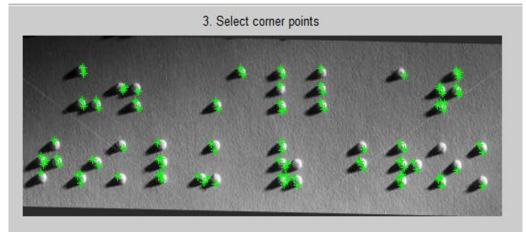


Figure-4: Corner point detection

Next the image is applied with edge detection method, the result of which is shown in figure-5. The edge detection function in MATLAB is edge () that uses a Sobel operator.

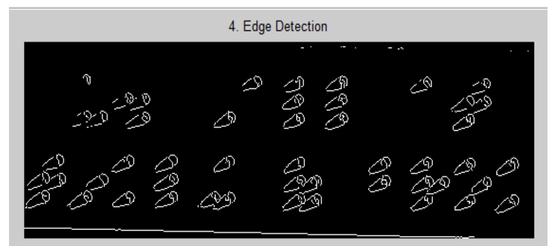


Figure-5: Edge detection

Finally by calculating PCA of detected corner point image we find the tilt angle. This tilt angle is used to rotate the image that finally removes the skew angle. [9] The resultant rotated image is shown in figure-6. The imrotate () function of MATLAB is used to rotate the image in clockwise or anticlockwise direction with calculated tilt angle.

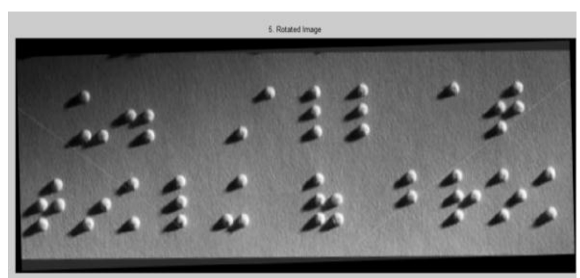


Figure-6: Rotated image (after skew correction)

The algorithm-2 uses the blob detection technique. The blob detection method detects the image regions that differ in properties like brightness and color [9].

The algorithm-2

1. Read the color image
2. Convert to 8-bit Gray scale image
3. Binarize the image by taking Threshold
4. Obtain histogram
5. Get blob properties.
2. Extract just one orientation.
3. Calculate the rotation angle
4. Rotate the image using this angle

We apply the algorithm-2 on the same input colored image of figure-2. The output grayscale image obtained after step-2 of algorithm is also shown in figure-3. In step3, the binary image is obtained using the equation-13 that uses the imfill () method.

$$\text{im} = \text{grayImg} > 128;$$
$$\text{binImage} = \text{imfill}(\text{im}, '\text{holes}'); \quad \text{-- (13)}$$

Where 'im' is a binarized image obtained by comparing with the value 128. We generally obtain an image consisting of bits.The resultant binarized image of step3 of algorithm is shown in figure-7.

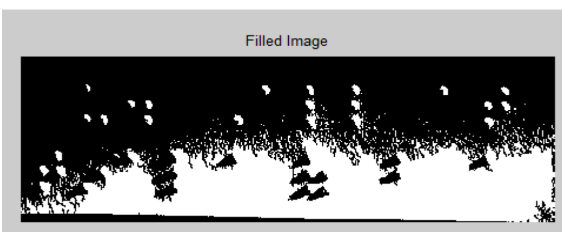


Figure-7: Binarized image

The rotated image using this algorithm is shown in the figure-8.

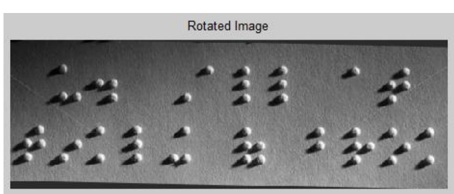


Figure-8: Rotated image (after skew correction)

Results presented in this paper are evaluated for the input image with 5^0 tilt angle for both the algorithms. The experiment is repeated for the images having various tilt angles. The results are presented in table-1.

IV. RESULTS AND DISCUSSION

The accuracy of skew detection and correction of both the algorithm are tested on 5 different images having some percentage of tilt angles are used as the data base using MATLAB. The Skew angles are in the range of 0^0 to 10^0. The accuracy results are shown in Table-1. The results are compared to consider the efficient algorithm to be used for further in Braille to text conversion processing.

Table-1: Result of accuracy rate of skew detection and correction

Tilt angle in degrees	Algorithm-1 Accuracy	Algorithm-2 Accuracy
1^0	100 %	100 %
3^0	99 %	92 %
5^0	75 %	63 %
8^0	51 %	30 %
10^0	2 %	0 %

V. CONCLUSION AND FUTURE WORK

We compared two algorithms to verify efficiency in skew detection and correction. Both algorithms are applied and evaluated on 5 dataset with certain amount of introduced skew. The comparison of the results is shown in table-1. The comparison proved that the algorithm that uses sobel edge detector and Gaussian filter method is more efficient in determining the skew angle and effectively de-skew the document. This work can be extended further by applying the different image segmentation methods which will improve the quality of feature extraction. We will concentrate our study to find out the best segmentation method by comparing the final results after evaluating on MATLAB.

References

[1] H. Zhang, J. Li , "A Research on Paper-Mediated Braille Automatic Extraction Method," *2010 International Conference on Intelligent Computation Technology and Automation*, Changsha, China, 2010, pp. 328-331. doi:10.1109/ ICICTA.2010.145.

[2] J. Subur, T. A. Sardjono and R. Mardiyanto, "Braille character recognition using find contour method," *2015 International Conference on Electrical Engineering and Informatics (ICEEI)*, Denpasar, 2015, pp. 699-703. doi: 10.1109/ ICEEI. 2015. 7352588.

[3] J. Mennens, L. Van Tichelen, G. Francois and J. J. Engelen, "Optical recognition of Braille writing using standard equipment," in *IEEE Transactions on Rehabilitation Engineering*, vol. 2, no. 4, pp. 207-212, Dec 1994. doi: 10.1109/86.340878.

[4] Jie Li and Xiaoguang Yan, "Optical Braille character recognition with Support-Vector Machine classifier," *2010 International*

Conference on Computer Application and System Modeling (ICCASM 2010), Taiyuan, 2010, pp. V12-219-V12-222. doi: 10.1109/ICCASM.2010.5622245

[5] L. Nian-feng and W. Li-rong, "A kind of Braille paper automatic marking system," *2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC)*, Jilin, 2011, pp. 664-667. doi: 10.1109/MEC.2011.6025553.

[6] M. Hanumanthappa and V. V. Murthy, "Optical Braille recognition and its correspondence in the conversion of Braille script to text — A literature review," *2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*, Bangalore, 2016, pp. 297-301. doi: 10.1109/CSITSS.2016.7779374

[7] Padmavathi.S, Aishwarya.A.V, Iswaryah,. "Geometric Correction for Braille Document Images.", *2016 International Conference on Pedagogic Advances in Computer Science (ICPACS-2016)*, Computer Science & Information Technology (CS & IT), Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, India. (2016).

[8] T. Shreekanth and V. Udayashankara, "A two stage Braille Character segmentation approach for embossed double sided Hindi Devanagari Braille documents," *2014 International Conference on Contemporary Computing and Informatics (IC3I)*, Mysore, 2014, pp. 533-538. doi: 10.1109/IC3I.2014.7019590.

[9] Wang Changjie, Nian Hua. "Algorithm of remote sensing image matching based on corner-point", 2017 Int ernat ional Workshop on Remote Sensing with Intelligent Processing (RSIP), 2017, doi:10.1109/RSIP.2017.79588

