International Journal of Information Technology and Knowledge Management
July-December 2012, Volume 5, No. 2, pp. 239-243

THE COMPUTATIONAL FACETS OF NORMALIZATION STRATEGIES FOR THE RECOGNITION OF A LANGUAGE

Indu Chhabra*

ABSTRACT: Normalization of a character image is a crucial preprocessing stage in the development of robust text recognizers. It helps in the standardization of size, position as well as font variants of a same character shape so as to reduce the differences between the images of the same character class. Comparison between the ratio-based normalization and scaling based normalization is made to analyze their affects in the process of standardization with respect to shape degradation and deformity, they introduce in the original shape. The effect of each normalization method on a character image is evaluated. In fixed aspect ratio mapping the normalized image plane is mapped to a square. For inherently long-shaped or wide-shaped characters, transforming those to square may yield excessive distortion and will deteriorate the recognition performance. An Aspect Ratio Adaptive Normalization is analyzed that yields better recognition performance as the aspect ratio of the input image is partially preserved with adaptation in such a way that the original aspect ratio can be derived from the normalized aspect ratio and vice-versa.

Keywords: Character Recognition, Preprocessing, Linear Normalization, Aspect Ratio, Deformity and Distortion.

1. INTRODUCTION

A text may exhibit different size, shape and orientation in the image space. The performance of handwritten or printed character recognition is largely dependent on normalization of character shapes as it regulates the variations in size, position and shape of the character images of the same class. Normalization is a process of relating pieces of data together, instead of storing the same data over and over again. It is an important pre-processing factor in character recognition [1, 2] because many feature detection techniques require an image of standard size for feature derivation, scale normalization and proper classification [3,4]. In character recognition a number of evaluation studies are accomplished for normalization. Casey [5] proposed moment based and Yamada et al. [6] introduced non-linear method for normalizing hand-printed characters. But after analyzing the affects of aspect ratio on the recognition performance, C. L. Liu et al. [7] proposed an Aspect Ratio Adaptive Normalization (ARAN) strategy to improve the accuracy. To preserve the character shape, Shi [8] proposed a method that aligns both centroid and character boundary. In normalization, the direct scaling of all the different sizes to an identical size results in a significant deformation in the character shape [9, 10]. It is expected from the normalization technique that the deformation of characters should be restored so as to reduce the within-class shape variations.

This paper focuses on the implementation details of various linear normalization applied to the original character size. The aim is to diversify the aspect ratio mapping function that improves the recognition performance. The normalized image $g(x', y')$ is generated by pixel value and coordinate interpolation of original image $f(x, y)$. The mapped coordinates (x', y') are discretized and interpolated to generate the image $g(x', y') = f(x, y)$.

2. LINEAR NORMALIZATION

In linear dimension based methods, the actual image width and height is linearly mapped onto a standard plane of fixed x-y dimensions by interpolation or extrapolation. It aligns the physical boundaries (e.g. ends of stroke projections) of input image to the boundaries of normalized image. So the size and position of character is altered while filling the normalized x-y dimensions that leads to the changes in the original character aspect ratio. The forward coordinate mapping of linear function is derived with $x' = \alpha x$ and $y' = \beta y$ where α and β are ratios of transformation with $\alpha = W_2/W_1$ and $\beta = H_2/H_1$.

3. PERFORMANCE OF LINEAR NORMALIZATION TECHNIQUES

In order to govern the size and shape of the normalized image, the width and height of the normalized image is determined according to the varying aspect ratio mapping function. In the present implementation, the coordinates are mapped forwardly from binary input image to normalized image and are discretized to generate a binary normalized image with R_1 and R_2 denoting the aspect ratio of original and normalized images respectively.

$$R_1 = \begin{bmatrix} W_1 / H_1, \text{ if } W_1 < H_1 \\ H_1 / W_1, \text{ Otherwise} \end{bmatrix} \text{ and } R_2 = \begin{bmatrix} W_2 / H_2, \text{ if } W_2 < H_2 \\ H_2 / W_2, \text{ Otherwise} \end{bmatrix}$$

* Associate Professor, Deptt. of Computer Science, Panjab University, Chandigarh, *E-mail: chhabra_i@rediffmail.com*

For the case study, we have implemented these techniques for the 14th most spoken Gurumukhi language in the world. The various normalization functions with varying aspect ratio mapping functions are:

N_0: Linear Normalization with Fixed Aspect Ratio

This technique is an aspect ratio adaptive normalization technique where the aspect ratios of the original and the normalized images are not same but the width and height of the original image can be derived from its normalized version and vice-versa. In linear normalization the original image of any size is normalized in such a way that the size of normalized plane is considered to be fixed and square with

$$R_2 = 1 \quad \text{so} \quad \alpha R_1 = \beta$$

Observations and Results: This following table lists the scaling factors, width and height of the normalized images corresponding to varying scaling factors and sizes of original images.

Table 1
Implementation of N_0

W_1	H_1	R_1	α	β	R_2	W_2	H_2
26	**26**	**1**	**1.07**	1.07	1	28	28
23	**26**	**0.884**	**1.31**	1.15	1	30	30
16	**28**	**0.571**	**2.00**	1.14	1	32	32
27	**19**	**0.703**	1.52	1.06	1	29	29

The Figure 1 depicts the effects of N_0 normalization function on the original shape.

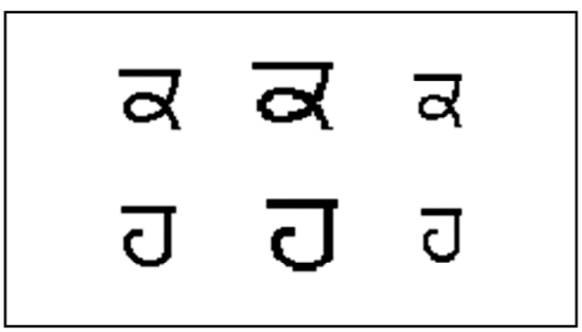

Figure 1: (*a*) Original (*b*) Normalized with Scaled-up (*c*) Normalized with Scaled-down with N_0 Function

N_1: Linear Normalization with Aspect Ratio Preserved

The dimensions of original are increased by scaling factors α and β in such a way that the aspect ratio remains same. This function preserves the aspect ratio by assigning

$$R_2 = R_1 \quad \text{so} \quad \alpha W_1/\beta H_1 = W_1/H_1$$

Observations and Results

Table 2
Implementation of N_1 Function

W_1	H_1	R_1	α	β	R_2	W_2	H_2
23	26	0.884	1.20	1.20	0.884	27.6	31.2
11	28	0.392	0.81	0.81	0.392	8.91	23
16	28	0.571	2.01	2.01	0.571	32	56
27	19	0.703	1.52	1.52	0.703	41	29

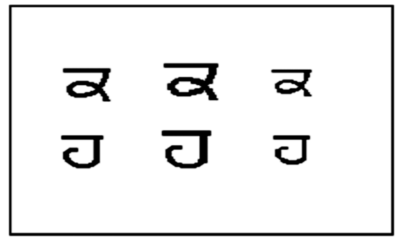

Figure 2: (*a*) Original (*b*) Scaled-up Normalization (*c*) Scaled-down Normalization with N_1 Function

N2: Linear Normalization with Square Root of Aspect Ratio: This technique is an aspect ratio adaptive normalization (ARAN) technique where the normalized aspect ratios is got by having the square root of the original aspect ratio, as given by: $R_2 = \sqrt{R_1}$ so $\alpha^2 W_1^2 / \beta^2 H_1^2 = W_1 / H_1$

Observations and Results: The Table 3 details the aspect parameters of the normalized images.

Table 3
Implementation of N_2 Function

W_1	H_1	R_1	α	β	R_2	W_2	H_2
32	**23**	**0.718**	0.90	0.76	0.85	21	24.5
11	**28**	**0.392**	1.03	0.65	0.63	11.3	18
23	**26**	**0.884**	1.20	1.13	0.94	27.6	29.3
9	**28**	0.321	2.00	1.13	0.567	18	32

The column R_2 shows the adaptability of column R_1.

Figure 3: (*a*) Original (*b*) Normalized-up (*c*) Normalized-down with N_2 Function

N_3: Linear Normalization with Piecewise Linear Aspect Ratio

In order to preserve the original characteristics, it is always better to work on sub-parts rather than on the whole [2]. The N_3 function performs a piece-wise analysis of original image to get the normalized R_2 as defined by: $R_2 = 0.25 + 1.5 \times R_1$ hence $\alpha = \beta \times (1.5 + 0.25 / (W_1/H_1))$

Observations and Results: Table 4 lists the width and height of normalized images.

Table 4
Implementation of N_3 Function

W_1	H_1	R_1	α	β	R_2	W_2	H_2
26	26	1.000	1.15	1.15	1.00	30	30
11	28	0.392	2.39	1.12	0.84	26	31
27	19	0.321	1.41	0.62	0.73	12.5	17
9	28	0.321	1.41	0.62	0.73	12.5	17

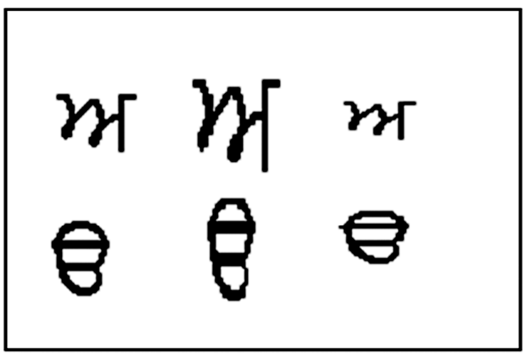

Figure 4: (*a*) Original (*b*) Normalized-up (*c*) Normalized-down with N_3 Function

N_4: Linear Normalization with Continuous Function

In N_4 the aspect ratio R_2 of normalized image is a continuous function of the original aspect ratio R_1 such that $R_2 = \sqrt{\sin(\pi/2 \times R_1)}$ hence $\beta = (\alpha \times R_1) / \sqrt{\sin(\pi/2 \times R_1)}$

Observations and Results: The Table 5 lists the aspect ratio of normalized images.

Table 5
Implementation of N_4 Continuous Function

W_1	H_1	R_1	α	β	R_2	W_2	H_2
23	26	0.884	0.70	0.63	0.98	16	16.22
16	28	0.571	1.07	0.78	0.78	17	21.81
27	19	0.703	0.90	0.70	0.89	17	18.94
11	28	0.392	2.00	1.35	0.58	22	38

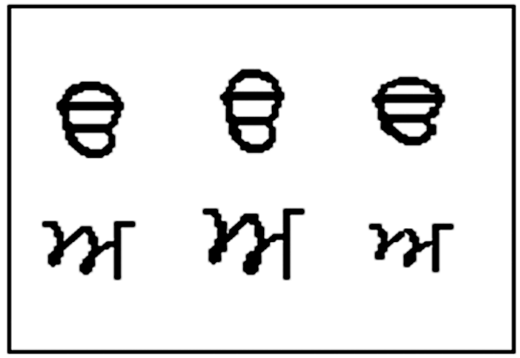

Figure 5: (*a*) Original (*b*) Scaled-up (*c*) Scaled-down with N_4 Function

4. RESULTS AND DISCUSSION

The N_0 normalization function results in widthwise deformation as exposed from scaled-up and scaled-down versions [Figure 1]. The change in height is proportional to the original but the change in the width is more as proportion to height. Hence the image has deformed widthwise. In N_1 as the scaling factors in *x* and *y* directions are same so width and height is increased equally, which gives only slight variation to the original image [Figure 2]. On applying N_2 in Figure 3 height change is not proportional to the original height whereas width change is proportional. Hence the normalized images have height deformation. From Figure 4, it is clear that N_3 is mainly useful for odd shaped characters that are the characters that are inherently long-shaped or wide-shaped. The other above mentioned techniques normalize the complete image by interpolation or extrapolation that can cause excessive distortion in case of these odd-shaped images. Hence, this technique takes care of the deformation by normalizing the image piecewise, if it has an odd shape otherwise it behaves like N_0. It is obvious from Figure 5 that the N_4 continuous function is able to adapt the original dimensions while introducing little deformation. Hence the image deformation is moderate to accommodate the various with-in-class differences. The normalized functions N_2 and N_4 are performing best in dimension based normalization because these are capable of restoring the deformation of original character. The N_1 is giving inferior performance since it preserves the aspect ratio. Among these seven functions N_0, N_3 and N_4 are being tested by Liu [11] and N_3 and N_4 are shown to outperform N_0.

5. CONCLUSION

The pre-processing of an image is an important step in the recognition process as it removes the various noises, skew and size degradation by separating the pattern of interest from the background. Normalization is an important pre-processing factor to achieve scale invariance. The comparative analysis of seven normalization strategies based on linear and moment normalization is presented to analyze their affects with respect to shape degradation and deformity they introduce in the original shape. The aim is to choose the best normalization function that can introduce the sufficient shape deformity while preserving the original aspect ratio in order to classify the various scaled versions of a character into same class.

REFERENCES

[1] Rothe H. Susse K. Voss, "The Method of Normalization to Determine Invariants", *IEEE Transactions Pattern Anal. Mach. Intel.*, **18**, pp. 366-375, 1996.

[2] Indu Chhabra and Chandan Singh, "Describing Character Object with Invariant Features", *Journal of Computer Society of India*, India, **36**, pp. 23-28, December 2006.

[3] B. Tudu and R. Bandyopadhyay, "Comparison of Multivariate Normalization Techniques", *Proceedings of 3rd International Conference on Sensing Technology,* Kolkata, pp. 254-258, 2008.

[4] U. Pal and B.B. Chadhuri, "Indian Script Character Recognition-A Survey", *Pattern Recognition*, **37**, pp. 1887-1898, 2004.

[5] Casey, R, "Moment Normalization of Hand Printed Characters", *IBM Journal of Research Development*, pp. 548-57, 1987.

[6] H. Yamada, K. Yamamoto, T. Saito, "A Nonlinear Normalization Method for Hand Printed Kanji Character Recognition", *Pattern Recognition*, **23**, pp. 1023-1029, 1990.

[7] C.L. Liu, H. Fujisawa, "Aspect Ratio Adaptive Normalization for Handwritten Character Recognition", *Lecture Notes in Computer Science*, **1948**, Springer, pp. 418-425, 2000.

[8] M. Shi, T. Wakabayashi, F. Kimura, "Handwritten Numeral Recognition Using Gradient and Curvature of Gray Scale Image", *Pattern Recognition*, **35**, pp. 2051-2059, 2002.

[9] C.L. Liu, H. Sako, H. Fujisawa, "Handwritten Digit Recognition Investigation of Normalization and Feature Extraction Techniques", *Pattern Recognition*, **37**, pp. 265-279, 2004.

[10] Indu Chhabra and Chandan Singh, "Shape Extraction for the Recognition of Gurmukhi Script", *International Journal of Technology, Knowledge and Society*, University Press, USA, Jan. 2007.

[11] C.L. Liu, R.W. Dai, "Normalization-Cooperated Gradient Feature Extraction For Handwritten Character Recognition", *IEEE Transactions on PAMI*, **29**, pp. 1465-1469, 2007.

[12] C.L. Liu, C.Y. Suen, "A New Benchmark on the Recognition of Handwritten Bangla and Farsi Numeral Characters", *Pattern Recognition*, **42**, pp. 3287-3295, 2009.

[13] M. Mori, M. Sawaki and J. Yamato, "Robust Character Recognition Using Adaptive Feature Extraction Method", *IEICE Transactions on Information and Systems*, **E93**, pp. 125-133, 2010.