

Slant normalization of handwritten numeral strings

Alceu de S. Britto Jr^{1,4}, Robert Sabourin², Edouard Lethelier¹, Flávio Bortolozzi¹,
Ching Y. Suen³

adesouza, sabourin@livia.etsmtl.ca
suen@cenparmi.concordia.ca
fborto, edouard@ppgia.pucpr.br

1-Pontifícia Univ. Católica do Paraná (PUC-PR)
Rua Imaculada Conceição, 1155 – P. Velho
80215-901 - Curitiba (PR) - Brazil

2-École de Technologie Supérieure (ETS)
1100, Rue Notre Dame - Ouest
H3C 1K3 - Montreal (QC) – Canada

3-Centre for Pattern Recognition and Artificial
Intelligence (CENPARMI)
1550, de Maisonneuve Blvd. - Suite GM 606 -
Concordia University
H3G 1M8 - Montreal (QC) -Canada

4-Univ. Estadual de Ponta Grossa (UEPG)
Praça Santos Andrade, S/N – Centro
84100-000 - Ponta Grossa (PR) - Brazil

Abstract

Two word slant correction methods are evaluated in this work using handwritten numeral strings. A method originally proposed to word slant correction is adapted in order to improve the results for handwritten numeral strings. The original and modified methods are evaluated by means of some interesting analyses on the NIST SD19 database. In these analyses the difference, in terms of slant estimated, between two slant correction techniques of isolated numerals extracted from strings, and the impact of slant correction on the number of overlapping numerals in the strings are evaluated. The results look promising.

Keywords

Slant correction, Numeral Strings, Overlapping Numerals, NIST Database.

Introduction

The character inclination typically found in cursive script is called slant. Formally, it is defined as the slope between the axis referent to the mean direction of the characters and the vertical axis. Even being a writer characteristic, slant may be considered useless. Thus, it is common to have slant correction during the pre-processing stage in writer-independent methods for word recognition. The general objective is to reduce the script variability.

Some specific objectives are dependent on the recognition approach. For example, in explicit segmentation-based methods, the slant correction can be used to improve the results of segmentation techniques based on vertical single-run zones to define character candidates. In these methods, the slant correction allows us to increase the probability of a candidate to be a character. Another example is found in implicit segmentation-based methods, where a vertical strip is used to scan the word image for feature extraction. Here, the slant is corrected in order to minimize the problem when slanted characters overlap adjacent characters, interfering in the columns of pixels extracted from them.

In this study the Bozinovic and Srihari (BS) [1] and the Kimura, Shridhar and Chen (KSC) [4] methods are evaluated using handwritten numeral strings. The last method is adapted in order to improve the results for handwritten numeral strings. In addition, the original and modified KSC methods are compared by means of interesting analyses performed on a data set extracted from NIST SD19 database [3].

For this purpose, this work is organized in 5 sections. Section 1 presents different methods for slant estimation. The BS and KSC methods are evaluated using handwritten numeral strings. In Section 2, the KSC method is adapted to improve the results for handwritten numeral strings. Section 3 presents some analyses regarding slant correction of isolated numerals extracted from strings, and the impact of slant correction on the number of overlapping numerals in the strings. These analyses are also used to compare the KSC and MKSC methods. A discussion is presented in Section 4, and the conclusion in Section 5.

1. Methods of slant estimation

Many methods have been proposed to correct the slant of cursive words. In [2], slant estimation is obtained by placing two horizontal thresholds through the center of the word. After the crossover points have been determined (points where the script crosses the thresholds), the associated slope is calculated from the crossover coordinates. The average of these slopes is used as the script slant.

The BS method [1], as a first step, removes all horizontal lines which contain at least one run of length greater than a parameter *maxrun*. In addition, all horizontal strips with height less than a parameter *stripheight* are removed. The second step divides the horizontal strips by vertical lines into isolated windows. For each window the center of gravity for its upper and lower halves are computed and connected (except windows with an empty half). Finally, the slope of the connecting line defines the slope of the window, and the average for all windows defines the slope β of the word.

A different approach is found in the KSC method [4], where the chain code of the word contour is used to slant estimation. The horizontal chain elements (code n_0) are not considered, and the other elements are divided into slant chain segments: n_1 , n_2 and n_3 , respectively 45° , 90° and 135° . The average orientation of these segments is given by:

$$\theta = \tan^{-1} \left(\frac{n_1 + n_2 + n_3}{n_1 - n_3} \right). \quad (1)$$

Simoncini and Kovacs use a similar way to define word slant in [5]. The slant estimation is given by the following expression:

$$\theta = \tan^{-1} \left(\frac{n_1 - n_3 + Kn_2}{n_1 + n_2 + n_3} \right), \quad (2)$$

where again, n_1 , n_2 and n_3 are respectively the number of contour segments of 45° , 90° and 135° , and $K \in [0,1]$ is an adjustment factor.

Yacoubi [7] defines a number of lines with different inclinations (slope p_i), the same origin and regularly spaced from -45° to 45° , using multiples of an elementary angle ($= \pi/80$). Afterwards, a projection histogram H_i is calculated for

each line. The H_i histogram is used to calculate a second histogram G_i , containing the number of segments in H_i for each k possible length. The surface S_i of each histogram G_i is calculated (or, the sum of the segments weighted by their respective lengths). Only segments of length greater than s are used ($s=3/5$ x length of the word body). The mean slope is calculated as follows:

$$\theta = \sum_i c_i p_i \quad (3)$$

where,

$$c_i = \frac{S_i}{\sum_j S_j} \quad (4)$$

Except for the KSC method, the majority of the described methods are strongly based on heuristics. We selected the KSC and BS methods for evaluation using handwritten numeral strings. The BS method was chosen, since it has been often cited in the literature. Some slant-corrected strings using both methods are shown in Table 1. During the experiments, the BS method has shown to be strongly dependent on the parameter *maxrun*, which is dependent on the stroke width (thickness) of the string. The examples *B*, *D* and *E* in Table 1 represent problems found when *maxrun* was fixed in 15. A possible alternative to avoid this kind of problem is to define *maxrun* using a distance function to calculate the mean thickness of the string. However, this may present time-consuming problem.

	Original string	Slant-corrected string BS method	Slant-corrected string KSC method
A			
B			
C			
D			
E			

Table 1 - Slant correction of different lengths numeral strings using the BS and KSC methods

On the other hand, the KSC method presents only one inconvenience. The image contour must be defined. However, no heuristic is used and good results have been obtained (also in Table 1).

2. Modified KSC method

The KSC method was originally proposed for slant correction of cursive words. The slant estimation is performed using the whole word. This is not satisfactory for numeral strings, since these are sets of components (digits or fragments), where each one has its own slant. However, an independent correction of each component is not viable, since this may produce distortions when broken digits are present in the numeral string.

We propose a modified KSC method (MKSC), which consists of estimating the mean slant based on the average of the slant of each string component. The string slant is obtained through a weighted average, where the slant of each string component is weighted by its contour length. Let N be the number of components (digits or parts of digits) in a numeral string; the mean slant can be estimated by the following expression:

$$\bar{\theta} = \frac{\sum_{i=1}^N (def - \theta_i) \times w_i}{\sum_{i=1}^N w_i} \quad (5)$$

where, def is the normal slant (90°),

$$\theta_i = \tan^{-1} \left(\frac{n_{i1} + n_{i2} + n_{i3}}{n_{i1} - n_{i3}} \right) \quad (6)$$

and, w_i is the chain code length of the i^{th} connected component. The objective of using a weighted average is to avoid distortions, which can be caused by small fragments in the string.

Original string	Original string
960941	87

Slant estimated for each component (KSC)		Slant estimated for each component (KSC)	
9	-45°	8	-26.6°
6	-45°	7	-45°
0	-26.6°	-	-
9	-26.6°	-	-
4	-45°	-	-
1	-45°	-	-

String slant		String slant	
KSC method	MKSC method	KSC method	MKSC method
-45°	-39°	-26.6°	-35.8°

String after slant correction		String after slant correction	
KSC method	MKSC method	KSC method	MKSC method
960941	960941	87	87

a) b)
Table 2 – Two comparisons between KSC and MKSC methods

Table 2 shows two comparisons between the KSC and MKSC methods. In both, the slant obtained from the modified method (-39° and -35.8° respectively) is better than that obtained from the original method (-45° and -26.6° respectively), since those are more uniformly distanced from the slant of each string component. Even visually, it is possible to verify a noticeable improvement. Table 3 shows different length numeral strings slant-corrected using both methods.

Original Image	Slant-corrected image KSC method	Slant-corrected image MKSC method
67	67	67
31	31	31
40	40	40
701	701	701
488	488	488
4234	4234	4234
419219	419219	419219
109334	109334	109334
0123456789	0123456789	0123456789

Table 3 - Numeral strings of different lengths slant-corrected using the KSC and MKSC methods

3. Analyses

We perform two interesting analyses using a data set extracted from NIST SD19 database. In the first, we evaluate the difference of using or not contextual information during slant correction of isolated numerals extracted from strings. The second analysis consists of evaluating the impact of slant correction on the number of overlapping numerals in the strings.

Figure 1 shows the process used to extract 44,256 good strings and 197,784 well-segmented numerals from the NIST database, which are used in these analyses. A handwritten numeral string is considered a “good string” when its components are recognized as isolated digits. Moreover, the recognition result of this string, from its digits, must correspond to that labeled by NIST. In order to implement this process, we have used a digit classifier proposed in [6].

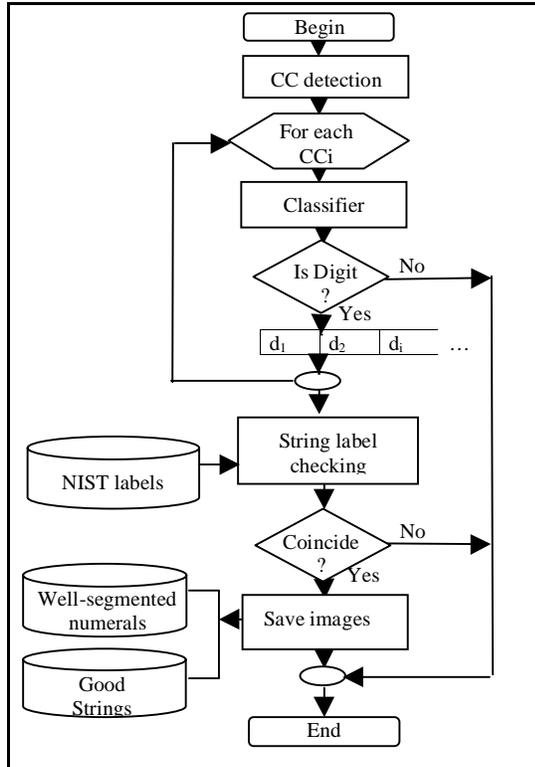


Figure 1 - Process to extract good strings and well-segmented numerals from NIST

3.1 Slant correction with and without contextual information

We use the KSC and MKSC methods to show the difference between correcting the slant of isolated numerals extracted from strings using two techniques: slant correction with contextual information, where the slant is estimated from the original string; and slant correction without contextual information, where the slant is estimated for each isolated numeral in particular, without taking into account the origin. The objective of this analysis is to evaluate the difference between both techniques in terms of the slope estimated. The well-segmented numerals extracted from the good string samples are used in this analysis, since they have a link to the original strings. For each numeral image $\Delta\theta$ is calculated, as $\Delta\theta = |\theta_1 - \theta_2|$, where θ_1 is the slant of a numeral, which is estimated from the single numeral; θ_2 is

the slant of a numeral, which is estimated from the original string. Figure 2 presents the frequency of each $\Delta\theta$ in our set of 197,784 well-segmented numerals, using the KSC and MKSC slant correction methods.

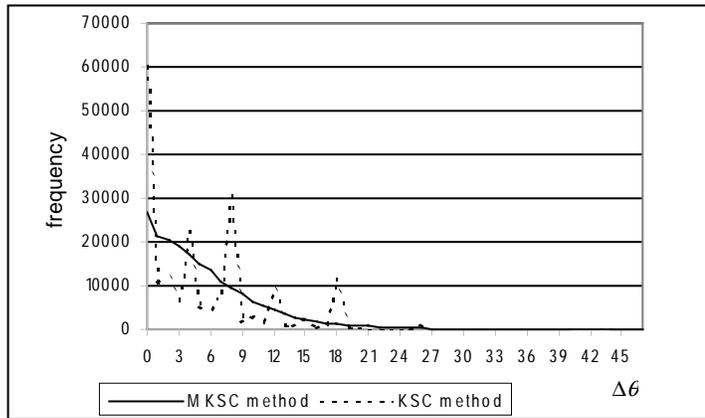


Figure 2 - Frequency of $\Delta\theta$

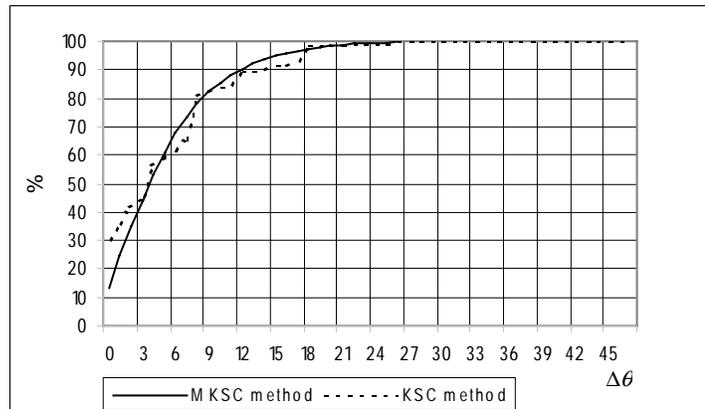


Figure 3 – Cumulative distribution of $\Delta\theta$

In addition, we have calculated the $\Delta\theta$ mean and the dispersion for each numeral class. The results are shown in Table 4 for both slant correction methods.

Class	KSC method		MKSC method	
	$\mu_{\Delta\theta}$	$\vartheta_{\Delta\theta}$	$\mu_{\Delta\theta}$	$\vartheta_{\Delta\theta}$
0	4.76	4.82	5.05	4.20
1	6.30	6.89	6.94	6.35
2	6.08	5.51	5.95	4.76
3	6.40	5.51	6.15	4.94
4	4.74	5.79	5.18	4.74
5	6.85	5.75	6.51	5.18
6	5.45	5.89	5.93	4.90
7	7.11	7.11	7.40	6.10
8	4.73	4.55	4.92	3.98
9	4.23	5.05	4.70	4.12
All samples	5.66	5.83	5.88	5.07

Table 4 - $\Delta\theta$ mean and dispersion using KSC and MKSC methods

We can see that the algorithms for slant correction show similar results. However, the MKSC method provides a smaller dispersion than that obtained by using the KSC method for all the numeral classes.

The mean variation is 5.88° with a dispersion of 5.07° (using MKSC method), with the largest mean variation occurring in the numeral class seven (7.40°). Figure 3 shows that there is a significant number of cases where $\Delta\theta$ is greater than 10.95° ($\mu_{\Delta\theta} + \vartheta_{\Delta\theta}$). This number represents 11.74% of all the analyzed images (23,223 numeral images) using MKSC method, and 15.23% of all the analyzed images (30,313 numeral images) using the KSC method. Again, the modified method shows better results, since the slant estimated from the string is closer to that estimated from each isolated numeral. However, even using MKSC method the difference is still significant. In Tables 5 and 6 we can see some significant differences, where $\Delta\theta > \mu_{\Delta\theta} + 2 \times \vartheta_{\Delta\theta}$ based on the use of MKSC method.

Original Image	Slant-corrected Image (MKSC)	θ_2	
		-42.1°	
Isolated Numerals			
Original Image	Slant-corrected Image	θ_1	$\Delta\theta$
		-44.9°	2.8°
		-44.9°	2.8°
		-44.9°	2.8°
		-24.5°	17.6°

Table 5 – Example of numeral string with significant $\Delta\theta$ (17.6°)

Original Image	Slant-corrected Image (MKSC)	θ_2	
		-35.6°	
Isolated Numerals			
Original Image	Slant-corrected image	θ_1	$\Delta\theta$
		-18.4°	17.2°
		-44.9°	9.3°
		-44.9°	9.3°
		-44.9°	9.3°
		-44.9°	9.3°
		-14.0°	21.6°

Table 6 – Example of numeral string with two significant $\Delta\theta$ (s) (17.2° and 21.6°)

3.2 Impact of slant correction on the number of overlapping numerals

The objective of this analysis consists of estimating the impact of correcting the string slant on the number of overlapping between adjacent numerals. The KSC and MKSC methods are also used in this analysis.

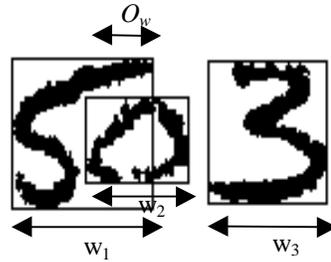


Figure 4 – Overlap estimation

The overlap between adjacent numerals is estimated based on the overlap between adjacent bounding boxes. For example, given the overlapping numerals in Figure 4, the algorithm calculates for each bounding box involved, its overlapping percentage (O_{pi}), as follows:

$$O_{pi} = O_w / w_i * 100 \quad (7)$$

In this example the overlapping (O_w) represents for the first bounding box (numeral 5) 47.6% of its width, and for the second one (numeral 0) 65.2% of its width. Figure 5 shows the impact of slant correction on the number of overlapping numerals in the 44,256 numeral strings analyzed.

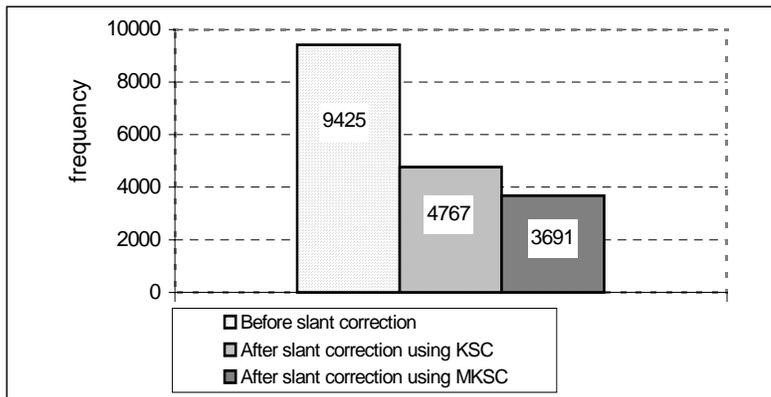


Figure 5 – Number of overlapping numerals before and after slant correction

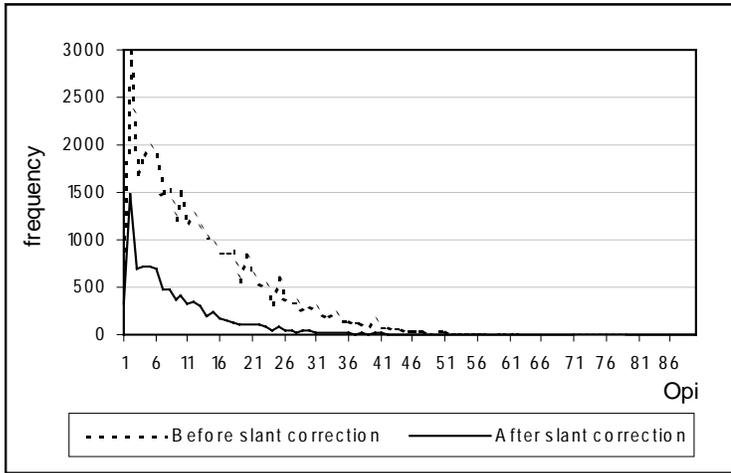


Figure 6- Frequency of overlapping percentages (Op) before and after slant correction using MKSC

Before correcting the string slant, there were 9,425 numeral strings with overlapping numerals, which represents 21.29% of all analyzed strings. After correcting the string slant using KSC method, the number of strings with overlapping numerals was reduced to 4,767, which represents 10.77% of all analyzed strings. The number of overlapping numerals was reduced by 49.42%. On the other hand, using the MKSC method the reduction was more significant. After correcting the string slant using MKSC method, the number of strings with overlapping numerals was reduced to 3,691, which represents 8.34% of all analyzed strings. The number of overlapping numerals was reduced by 60.83%. Figure 6 shows the Op distribution before and after slant correction using MKSC method.

4. Discussion

The first analysis shows that there is a representative part of the analyzed data, referring to those cases where $\Delta\theta > (\mu_{\Delta\theta} + \sigma_{\Delta\theta})$, in which the slants estimated with and without contextual information present a significant difference. This justifies an investigation of the real contribution of using contextual information during slant correction in terms of recognition performance of handwritten numeral strings. In addition, this analysis shows that the MKSC method provides a better $\Delta\theta$

distribution than that provided by KSC method. Moreover, using MKSC we can reduce the number of cases where $\Delta\theta$ is greater than $(\mu_{\Delta\theta} + \sigma_{\Delta\theta})$. This means that we can approximate the slant estimated from the string to the slant estimated from each isolated numeral in the string.

The second analysis shows that the slant correction reduces significantly the number of overlaps between adjacent numerals in the strings. In addition, we can see that MKSC method achieves a more significant reduction (60.83%) than that provided by KSC method (49.42%).

5. Conclusion

In this work we have evaluated two word slant correction methods using handwritten numeral strings. The method proposed in [4] is adapted in order to improve the results for handwritten numeral strings. The original and modified methods are evaluated by means of some interesting analyses on the NIST SD19 database. In these analyses the difference in terms of slant estimated between two slant correction techniques of isolated numerals extracted from strings, and the impact of slant correction on the number of overlapping numerals in the strings are evaluated. The results of MKSC method have been very encouraging.

6. References

- [1] Bozinovic R.M. & Srihari S. Off-line Cursive Script Word Recognition. IEEE Trans. On Pattern Analysis and Machine Intelligence, Vol. 11, No. 1, pp. 68-83, Jan. 1989.
- [2] Brown, M.K. and Ganapathy. Preprocessing Techniques for Cursive Word Recognition. Pattern Recognition, Vol 16, No. 5, pp-447-458, 1983.
- [3] Grother, P.J. NIST Special Database 19 - Handprinted Forms and Characters Database. National Institute of Standards and Technology (NIST), March, 1995.
- [4] Kimura F., Shridhar M. and Chen Z. Improvements of a Lexicon Directed Algorithm for Recognition of Unconstrained Handwritten Words. International Conference on Document Analysis and recognition 93, pp. 18-22, 1993.

- [5] Simoncini L. and Kovacs V. A system for Reading USA Census 90 Handwritten Fields. International Conference on Document Analysis and Recognition (ICDAR). Vol. 1, pp. 86-91, 1995.
- [6] Suen, C.Y., Liu K. and Strathy N.W. Sorting and Recognizing Cheques and Financial Documents. Proceedings of Third IAPR Workshop on Document Analysis Systems (DAS'98), pp. 1-18, 1998.
- [7] Yacoubi A. E. Modélisation Markovienne de l'écriture manuscrite Application à la reconnaissance des adresses postales. (These doctorate). Université de Rennes 1, 1996.