A New Method for Slant Calculation in Off-Line Handwriting Analysis

Latypov Rustam, Stolov Evgeni
Department for Computer Sciences
Kazan Federal University
Kazan, Russia
Email: Roustam.Latypov@kpfu.ru

***Abstract*—In this paper, we propose a new method for estimating the slant of word in handwritten text. The method allows a researcher to analyze snippets of a picture containing a few words in different lines. The main goal of the research is to present a tool to observe small changes of slant in the text during work. Student check sheets were used as a database for the research. Some changes in slant depending on speed of writing are discovered.**

***Keywords*—handwriting, calculation of slant.**

I. Introduction

Handwriting Analysis or Graphology is among ancient scientific methods used for revealing the characteristic traits of a human being, for example, physical, emotional, or mental states ([1], [2]). Feature extraction is the first step in handwriting analysis; to this end a list of the features to be extracted from a manuscript was developed. Most of these features are qualitative: height of letters, baseline form, slant of word, size of margins, and others ([3], [4], [5], [6]). The result of obtaining the needed traits is not a stable one, so experts known as graphologists are often under fair critic. The problem of acquisition of the human characteristics is very urgent now, but it seems to be rather difficult to reveal a kind of behavioural shift without a direct contact with person. Off-line analysis of changes in style of handwriting might resolve partially this problem.

The slant of words is a very important feature of a handwritten text. Mostly, authors are dealing with a qualitative determination of the slant: extreme left, controlled left, vertical, controlled right and extreme right [5]. There is an idea, known from publications on information theory (see [8], for example) on how to obtain a quantitative value of the slant. Assume that we have a black-white image containing a handwritten text, where pixels, corresponding to letters in the text, possess the value 1 while all other pixels take the value 0. For a given angle t, one creates a straight line which inclines at such angle the $X-$axis. The line moves along the text, and for any position x of the line the sum of the pixels, which are on the line at that time, is stored in an array $Sm(x)$. Then the array is normalized by division by the sum of all elements. In what follows, we call this array ‘array of projections ’. Since all values of the array are non-negative and their sum is 1, the array can be considered as a density of a random value. Such a density has a very important feature known as entropy of the random value.

The entropy is obtained accordingly to (1)

$$Entr(t) = -\sum_k \log(Sm(k,t)) \cdot Sm(k,t) \tag{1}$$

The values of entropy have various interpretations. For our aim, the following interpretation is of the most interest: the greater value of entropy one has, the more the density under research is looking like density of a uniform distribution [7]. If the angle t is close to the slant of the text, then the big values of $Sm(k,t)$ will be concentrated in a relatively small set of points. The latter means that the array of projections is far from the shape of the uniform distribution density, and the entropy value should be small. The value t_0, for which $Entr(t_0)$ is minimal, is declared as the slant of the word in the image. This idea is mentioned by many authors, and various implementations of the algorithm have been proposed, but the final result depends on the details of that implementation. In our paper, we present a new representation of the idea, the developed method allows quick finding the changes in slant.

II. Calculating the Slant

First, we will outline the traditional approach. The first problem that faces us in implementing the idea is how to build the line with a given angle t in digital case. One can easily create a straight line on a raster image if the line is horizontal or vertical. To get rid of such a limitation, some authors suggest to skew the original image by using the angle t and implement afterwards a vertical projection of the performed image [8]. All projections are stored in the array Sm, and entropy is then computed accordingly to (1). Suppose the baseline is known and point P has the coordinates (x, y) where all y-coordinates of points on baseline are equal to zero. For a given angle t the new coordinates of that point after such a transform are

$$x = x - y/\tan(t), \;\; y = y \tag{2}$$

This is the simplest version of the skew transform. Some other transforms based on the same idea are also possible [9]. It follows from (2) that an automatic calculation of the slant in a snippet is possible if all letters have a common baseline.

In this paper, we propose an approach where a direct implementation of the basic idea is presented. The advantage of this approach is that, in this case, the baseline is not used for computation. Of course, as a result, the value of slants evaluated by different methods can produce different values for the same word. However, our goal is to identify changes in this feature in dependence on the mood of a person; so only the differences in the obtained values are of importance.

Let us suppose that a text image has N and M as the width and height, respectively. Let us create a zero picture $Line_t$ of size $L \times M$ where $L = \mathrm{round}(M/\tan(t))$. Now, set

$$Line_t(\mathrm{round}(y/\tan(y)), y) = 1,\ y \in [1, M],\ if\ M > L$$
$$Line_t(x, x \cdot \mathrm{round}(\tan(x)) = 1,\ x \in [1, L], if\ M \leq L.$$

In the first case the $Line$ contains vertically extended rectangles and in the second case the $Line$ consists of horizontally extended areas. Examples of the pictures are given in Fig. 1.

A. Calculating the Projection

The next step in the procedure is to calculate the projection of all points in text, which are located on the specified $Line$. It is not a problem to find the number of the pixels on a line, if the line is horizontal or vertical. In general case we employ Discrete Fourier Transform (DFT) to find that number.

Recall some facts related to DFT [10]. Let A, B be two arrays of the same length N and FA and FB the results of their DFTs, respectively. Let FG be an array of the length N defined as follows: $FG(k) = FA(k) \cdot \overline{FB(k)}, k = 0, \ldots, N-1$, where $\bar{z} = \overline{a + jb} = a - ib$ for a complex value z. Let G be the result of inverse DFT applied to FG. Then we have

$$G(k) = \sum_{n=0}^{N-1} A(n)B(ind(n,k)), \qquad (3)$$
$$where\ ind(n,k) \equiv k + n \mod (N).$$

Let us suppose that $B = (b_0, \ldots, b_{m-1}, 0, \ldots, 0)$. Accordingly to (3),

$$G(k) = \sum_{n=k}^{k+m-1} A(n)B(k+n),\ k = 0, 1, \ldots, N-m. \qquad (4)$$

If we are dealing with two-dimensional arrays, then the formulas analogous to (4) also hold, but in this case only the first row of G will be of interest. In our approach, the role of A plays a snippet of the text, the role of B is played by a picture Z having the same size as that of A. Picture Z consists of picture $Line$ padded by zeros. The algorithm is presented below.

Algorithm 1 Calculate entropy

1: *t* ← *Angle*
2: *Im* ← *Image* { black-white image}
3: *N,M* ← size of *Image* { width and height of the image}
4: *Z* ← zeros *(N,M)*
5: *A* ← create $Line_t$
6: *L,M* ← size of *A*
7: *Z(1:L,1:M)* ← *A*{ place A in beginning of Z }
8: *FIm* ← fft2 *(Im)* { Fourier transform}
9: *FZ* ← fft2 *(Z)*
10: *FZC* ← $\overline{(FZ)}$
11: *FG* ← *FIm*FZC* { element-wise multiplication}
12: *Corr* ← ifft2*(FG)* { inverse Fourier transform}
13: *FirstRow* ← *Corr(1,1:N-L+1)* { N-L+1 item of the first row}
14: *Sm* ← Sum*(FirstRow* { sum of all items in FirstRow}
15: *FirstRow* ← *FirstRow/Sm* { element-wise division}
16: *FirstRowLog* ← log*(FirstRow)* { element-wise operation}
17: *Temp* ← *FirstRow* FirstRowLog* { element-wise multiplication}
18: *Entropy* ← -Sum*(Temp)*

There are some remarks concerning the algorithm presented.

1) We use $fft2$ and $ifft2$. i.e., the standard realizations of two-dimensional DFT.
2) In row # 16 we applied log function to the array $FirstRow$. Before the application of that operation, all zero items must be removed from that array.
3) While choosing a snippet for analysis, one have to fulfill the inequality $L < N$, since the number of possible positions of the $Line$ inside the snippet equals $N-L+1$.

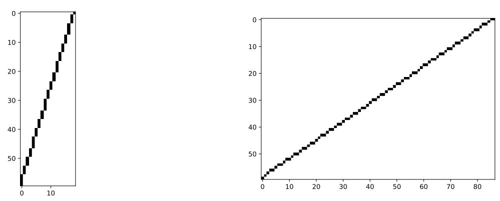
In a real situation, we know just an interval of the angles, to which belongs the slant, but L depends on the angle. To be sure that value of slant obtained is a correct one, the inequality

$$M << N \qquad (5)$$

must be satisfied. Later will be shown that if (5) is violated, then the calculations bring us an incorrect result.

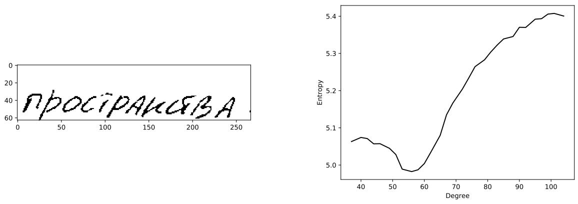
B. Examples of Calculating Slant of Word and That of Letter

Here are a few demonstrations of the work of the algorithm presented above; compare with the algorithm defined by (2). In Fig. 2, there is only single minimum point in the graph of entropy, and in accordance with that graph the slant is about 57 degrees. Of course, this slant is average. Now, select the last a letter in the in Fig. 2 and implement (2). The result is displayed in Fig. 3. In accordance with the expectation, we observe two local minimums, one of those being a global minimum, with close values. Those values correspond to the angles where one of two legs of the letter becomes vertical.



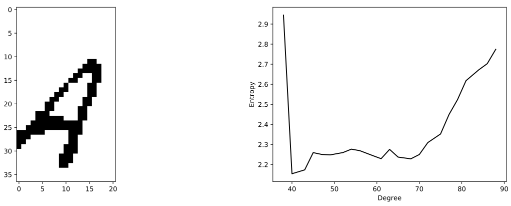
(a) (b)

Fig. 1. Examples of Line picture: (a) $M > L$,(b) $M \leq L$..



(a) (b)

Fig. 2. An example of calculation: (a) original picture, (b) entropy.



(a) (b)

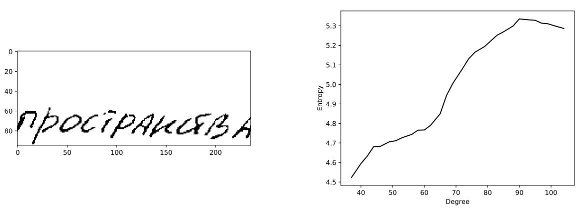
Fig. 3. Slant of a single letter: (a) the A letter, (b) entropy.

Our method cannot be implemented for investigation of slant of single letters, since (5) is not fulfilled in this case. For example, in Fig.4 there is depicted the situation where the condition (5) fails.

There is no local minimum of entropy for this rectangle. Let us explain the observed phenomenon.

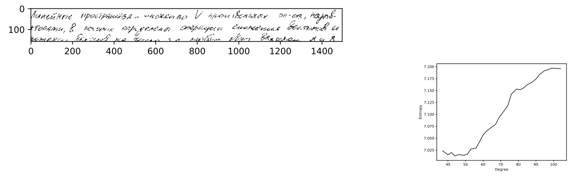
It follows from Fig. 1 that for a small angle the width of $Line$ can be significant, and, as a result, we have that the number of possible positions $N - L + 1$ is small. That is the reason of absence of a local minimum of the entropy function in that case.

On the other hand, our approach (as soon as an appropriate size of the rectangle is used) makes it possible to evaluate slant of words even if the rectangle contains a few rows (see Fig. 5).



(a) (b)

Fig. 4. Incorrect selected rectangle: (a) original picture, (b) entropy.



(a) (b)

Fig. 5. Slant of a few rows: (a) original picture, (b) entropy.

III. Experiments

We used the presented technique for investigating the dependence of slant in words while students fulfill their examination tasks. The dataset consists of images of student works. Each examination ticket consisted of three tasks from math:

1) proof of a theorem,
2) standard calculation,
3) theoretical problem.

and the time of the test was restricted to 1.5 hour. All answers where placed on a white sheet of paper of A4 size without any line. Note that, as a rule, the first task is not difficult, and students are not in a hurry while writing the proof. The second task is slightly more difficult, it requires significant calculation and takes much time. But, in answering to the second question, student uses many formulas and the corresponding area of the check sheet contains just few words. The most interesting task is the third one. It is a theoretical claim, and a student has to state whether it is a correct statement or not. It is not enough to say Yes or No and he/she must justify his/her decision. The student thus considers the tasks one after one, and to the time to deal with the third task the student possesses a very short interval to end of the test. On the other hand, the answer to the third question assumes usage of many words, that accelerates tempo of writing and leads to change of the slant. The main hypothesis of the authors of the paper is as follows: during the transition from slow writing to fast one changes occur in slant of words, which can be measured by suggested technique. An example of a check sheet filled in is presented in Fig. 6. The total number of the processed check sheets equals 90. The possibility of exact measuring the slant of letters brings some interesting facts. Let us define a basic slant as a word placed in top of the sheet of paper. It is supposed that the word is written in slow tempo. What is more, as a rule, the first line of the text is parallel to the upper bound of the sheet of paper. The distribution of the basic slant is depicted in Fig. 7. One can evaluate the mean value $Mean$ of the distribution, it is close to 75 degrees. The second measuring relates to a

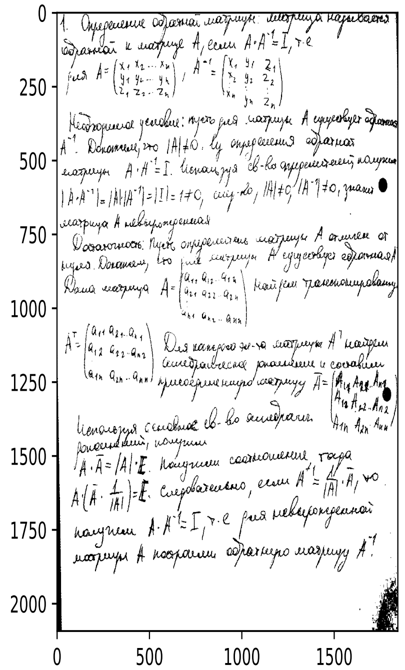


Fig. 6. Example of check sheet

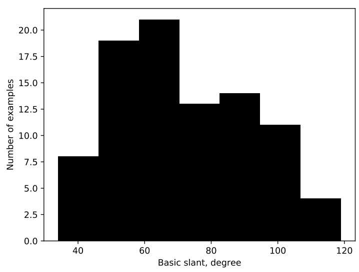


Fig. 7. Distribution of basic slant

word in the bottom of the sheet of paper. At that time, the student is in a hurry, because the time is over. Most of the lines have significant slope which affects the obtained value of the slant. The unexpected feature revealed by the research is the direction where the obtained slant is shifting to. In Fig. 8, the difference between shifted slant and basic slant depending on basic slant is displayed. If the basic slant is less than $Mean$, then the shift is positive, otherwise it is negative. Of course, this statement has just statistical sense, and some points in the figure do not meet this condition.

IV. Conclusion

The suggested method for calculation of slant in written text provides a fast procedure for obtaining the average values

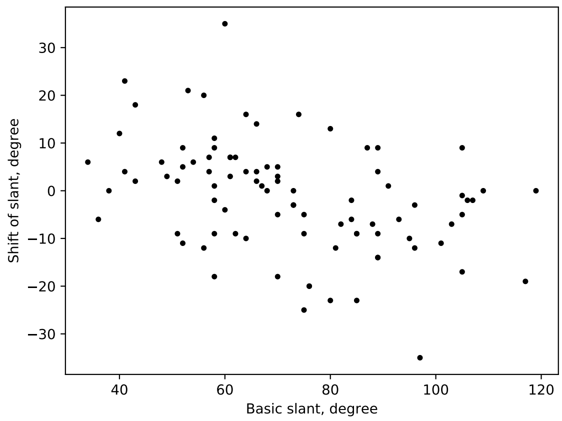


Fig. 8. Shift of slant depending on basic slant

of slant in snippets of text. The precision of the measure depends on the parameters of the procedure. An evaluation for the average slant of words is found, which turned out to be close to 75 degrees. The basic slant is defined as the slant of text written in low speed. It is shown that shift of the slant, when person accelerates writing tempo, depends on the value of his/her basic slant.

References

[1] S. V. Kedar, D. S. Bormane, A. Dhadwal, S. Alone, R. Agarwal, “Automatic Emotion Recognition through Handwriting Analysis: A Review ” in *Proc. Int. Conf. on Computing Communication Control and Automation (ICCUBEA)*,Pune, India, 2015, pp 811–816.

[2] H. N. Champa, K. R. AnandaKumar, “Automated Human Behavior Prediction through Handwriting Analysis ” in *Proc. First Int. Conf. on Integrated Intelligent Computing*,Bangalore, India,2010, pp 160–165.

[3] A. Varshney, S. Puri, “A survey on human personality identification on the basis of handwriting using ANN ” in *Proc. 2017 Int. Conf. on Inventive Systems and Control (ICISC)*, Coimbatore, India, 2017, pp 1–6.

[4] G. Vamvakas, B. Gatos, S. Petridis, N. Stamatopoulos, “An Efficient Feature Extraction and Dimensionality Reduction Scheme for Isolated Greek Handwritten Character Recognition ” in *Proc. 9th Int. Conf. on Document Analysis and Recognition (ICDAR 2007)*,Parana, Brazil,2007, vol. 2, pp 1073–1077.

[5] N. Mogharreban, S. Rahimi, M. Sabhanval, “A combined crisp and fuzzy approach for handwriting analysis ” in *NAFIPS 2004. Annual Meeting of the North American Fuzzy Information Processing Society*,Banff, Alberta, Canada, 2004,pp. 351–356.

[6] T. Bluch, H. Ney, C. Kermorvant, “Feature Extraction with Convolutional Neural Networks for Handwritten Word Recognition ” in *Proc. 12th Int. Conf. on Document Analysis and Recognition*, Washington,DC, 2013, pp 285–289.

[7] D. MacKay *Information Theory , Inference, and Learning Algorithms*. Cambridge University Press, 2003, pp. 32–38.

[8] Z. Q. Liu, J. H. Cai, R. Buse, *Handwriting Recognition: Soft Computing and Probabilistic Approaches*. Springer, 2003, pp. 156–157.

[9] H. Teulings, et al,“An on-line handwriting-recognition system based on unreliable modules. ” in *Computer Processing of Handwriting* R. Plamondon, G. Leedham Eds.,Singapore: World Scientific, 1990, pp. 167–185.

[10] A. Jain,*Fundamentals of Digital Image Processing*, Englewood Cliffs, NJ:Prenice-Hall, 1989, pp 141–146.