Comparison of Handwritings

Miroslava Božeková
mirka_boz@yahoo.com
http://sprite.edi.fmph.uniba.sk/∼bozekova
Faculty of Mathematics, Physics and Informatics
Comenius University
Bratislava, Slovak Republic

March 2008

Abstract

The goal of this paper is to contribute to the solution of handwritten text writer verification. Having two scanned images with handwritten text as input, our task is to create methods which determine whether the documents were written by the same person or not. We have proposed our approach consisting of preprocessing, feature vector extraction and combination with graphemes clustering. We made a set of experiments using the IAM Handwriting Database. A working implementation is presented together with experimental results.

Keywords: Handwriting, writer verification, line segmentation, word segmentation, graphemes, clustering

1 Introduction

We contribute to the solution of a following problem. Having two scanned images with handwritten text as input, our task is to create methods which determine whether the documents were written by the same person or not. In our work we report on preprocessing of images with handwritten text, various handwriting features and writer verification. We introduce our implementation which is inspired by recent research about preprocessing and writer verification. We made our experiments on 100 images from 40 different writers. These images were taken from the IAM Handwriting Database which contains 1539 pages of scanned text from 657 writers. In our experiments we achieved 96,5 % accuracy.
The paper is structured as follows. Section 2 gives the survey of related work. Section 3 describes our approach. Section 4 surveys results. Section 5 describes conclusions and discusses future work.

2 Recent research

Several widely acknowledged efforts have been presented in recent years in the writer verification problem. Cha and Srihari [4] model the problem as a two class classification problem: authorship or non-authorship. Feature distance is computed and the dichotomizer takes this feature distance vector as an input and outputs the authorship. They use 12 feature distances and achieved 97% accuracy on 1000 writers with 3 sample documents per writer.
Bensefia et al. [5] use sample set of 88 writers and extract the set of graphemes in two input handwritten documents. A grapheme feature set G is extracted thanks to a particular sequential clustering technique: $G = \{ g_1 , g_2 , g_3 ,..., g_N \}$. Some of these features may occur on the two documents while the others may occur specifically on one single document. Writer verification is based on the mutual information between the grapheme distributions in the two handwritings that are compared.
Schlapbach and Bunke [6] use HMM based recognizers. For each writer, they build an individual recognizer and train it on text lines of that writer. This gives their recognizers that are experts on the handwriting of exactly one writer. In the identification or verification phase, a text line of unknown origin is presented to each of these recognizers and each one returns a transcription that includes the log-likelihood score for the considered input. Using over 8,600 text lines from 120 writers an Equal Error Rate(EER) of about 2.5% is achieved (EER quantifies in a single number the writer verification performance).
A statistical model of the task and a large number of features divided into two categories: macro-features which capture the global characteristics of the writers individual writing habit and style and micro-features which capture finer details at the character/word level are proposed by Srihari et al. [7]. Same writer accuracy was 94.6% and different writer accuracy was 97.6%.
Bulacu and Schomaker 2007 [8] developed new and very effective techniques for automatic writer identification and verification that use probability distribution functions (PDFs) extracted from the handwriting

images to characterize writer individuality. Their methods operate at two levels of analysis: the texture level and the character-shape (allograph) level. The probability distribution of grapheme usage is computed using a common codebook of shapes obtained by clustering. Combining multiple features yields increased writer identification and verification performance. They use data sets from 3 databases Firemaker, IAM and ImUnipen.

3 Our approach

In this section we describe the whole method. First step is preprocessing of input images. Next, the handwriting features are extracted from the images. For each image we create feature vector consisting of the set of numbers which represent features. Then a new vector is created by the subtraction of two feature vectors and its coordinates are changed into absolute values. Result is given by the comparation of this new vector and vector of our thresholds (obtained from experiments). The first system, solving our task, is based on this feature vector.
Next, graphemes which were obtained from preprocessing are clustered using multiple algorithmic solutions, namely Kohonen's Self-Organizing Map (SOM) and hierarchical clustering. Result from each clustering is a measure which determines how input images are similar. This measure is realized by amount of clusters which contain graphemes coming from the same image. We join results from the first system and results from SOM to the second system. The third system is based on results from first system and results from hierarchical clustering. Result of 2nd and 3rd system is given by the comparation of measure and two thresholds. These three systems are compared experimentally.
All mentioned steps are described in detail in the following subsections. We also introduce the IAM Handwriting Database which is used for our experiments.

3.1 Input images

Input data are scanned images in PNG from the IAM Handwriting Database [1]. We decided for this database because it's a huge database suitable for our experiments. It contains forms of handwritten English text which can be used to train and test handwritten text recognizers and to perform writer identification and verification experiments. The database contains forms of unconstrained handwritten text, which were scanned at a resolution of 300 dpi and saved as PNG images with 256 gray levels.

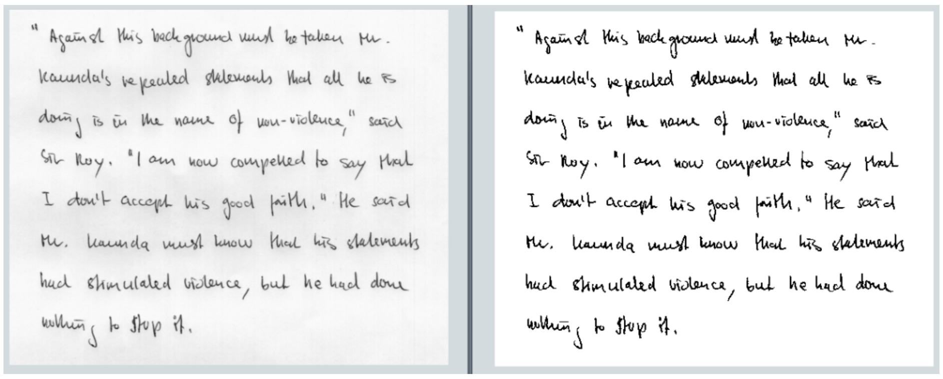
3.2 Preprocessing

The whole process is preceded by several preprocessing steps. Our goal is to get input images into the form which is suitable for future work and for obtaining graphemes which are inputs for grapheme clustering. Preprocessing steps contain:

- Thresholding (binarization of image)
- Line segmentation
- Slant correction
- Word segmentation
- Grapheme segmentation and normalization

Thresholding
We use Otsu's thresholding algorithm [2], which chooses the optimal threshold value by maximizing the between-class variance. If the gray level of pixels on an image ranges in L [1, 2,..., L] and the number of pixels at level i is denoted n_i, the total number of pixel can be calculated by equation: N=n_1+n_2+n_3+…+n_i+…+n_L. For bi-level thresholding, the pixels sets [1, 2,..., L] are divided into two classes; C_1 = [1, 2,..., t] and C_2= [t+1,..., L]. Otsu algorithm determines the optimal threshold value (t) that maximizes the between-class variance, $\sigma_B^2 = \omega_1\omega_2(\mu_2 - \mu_1)^2$
where $\omega_1(t) = \sum_{i=1}^{t}(p_i)$, $\omega_2(t) = \sum_{i=t+1}^{L}(p_i)$, $\mu_1(t) = \sum_{i=1}^{t}(ip_i/\omega_1(t))$, $\mu_1(t) = \sum_{i=1}^{t}(ip_i/\omega_2(t))$, and $p_i = n_i/N$ [10].
This method has been often used for images with handwritten text in recent research (see Figure 1).

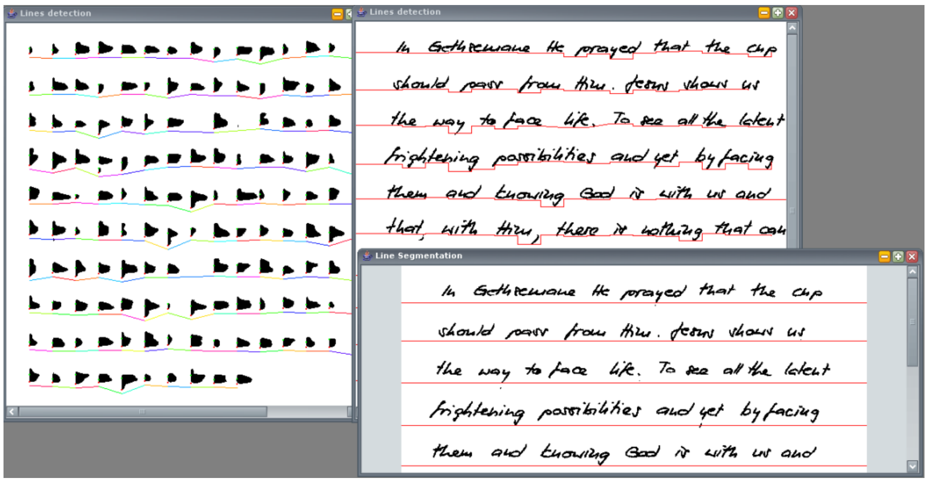
Line segmentation
The document is divided across its width into chunks, each represents 5%. Then the projection profile


" Against this background must be taken Mr.
Kaunda's repeated statements that all he is
doing is in the name of non-violence," said
Sir Roy. "I am now compelled to say that
I don't accept his good faith." He said
Mr. Kaunda must know that his statements
had stimulated violence, but he had done
nothing to stop it.

" Against this background must be taken Mr.
Kaunda's repeated statements that all he is
doing is in the name of non-violence," said
Sir Roy. "I am now compelled to say that
I don't accept his good faith." He said
Mr. Kaunda must know that his statements
had stimulated violence, but he had done
nothing to stop it.

Figure 1: Otsu's thresholding: on the left side is an input greyscale image and on the right side is result of Otsu's thresholding - a binary image. Background has white color and foreground has black color. Result is good if background doesn't contain black pixels and foreground doesn't contain white pixels.

(which counts number of black pixels for every row) and averaging filter are applied for every chunk. The averaging filter smooths projection profile. This filter performs spatial filtering on each individual pixel in an image using the grey level values in a square or rectangular window surrounding each pixel. Afterwards the image is converted to binary form. Algorithm sequentially traverses the chunks in horizontal direction and the result are red dashes which separate single lines (see Figure 2). We inspired by the article [3].



In Gethsemane He prayed that the cup
should pass from him. Jesus shows us
the way to face life. To see all the latent
frightening possibilities and yet by facing
them and knowing God is with us and
that, with him, there is nothing that can

Figure 2: Line Segmentation: on the left side is figured image with chunks after applying projection profile, averaging filter and thresholding on the image. Coloured lines are the result of sequential scanning (one color for one chunk).

Slant detection and correction

Handwriting has tendency to slant to the right or left side or it can be vertical, too. Our algorithm works on lines of handwriting. We determine 20 different slants. For every slant, line is corrected. Then the vertical projection profile (which counts number of black pixels for every column) is applied. From these 20 tries, one is chosen as the best slant for particular line. The choosing criteria involves the fact that the projection profile has to contain the largest amount of the highest peaks and also maximum gaps (see Figure 3).

Word segmentation

Word segmentation is based on the idea that distances between words are bigger than distances between characters. We extract contours from image (Moore's algorithm [11]) and then we count distances between horizontally neighbouring contours for every line. If the distance between contours is bigger than

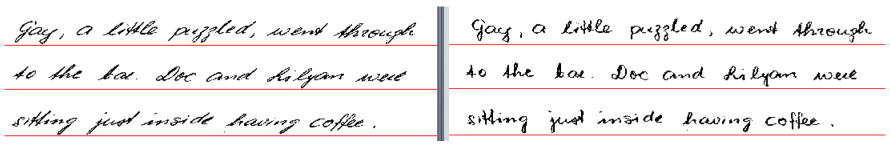


Figure 3: Slant correction: on the left side is input image and on the right side is the result of slant correction.

threshold (which is obtained by analyzing all distances), we get a gap between words.

Grapheme segmentation and normalization
We extract lower and upper contours from the image (see Figure 4). Then we find local minimums which occur in places where the upper contour is close to the lower contour. It means that the distance between lower and upper contour is width of line of handwriting. Words are segmented into graphemes at these local minimums by vertical lines (see Figure 5). After segmentation, graphemes are normalized to 30x30 pixels. We inspired by the article [8].

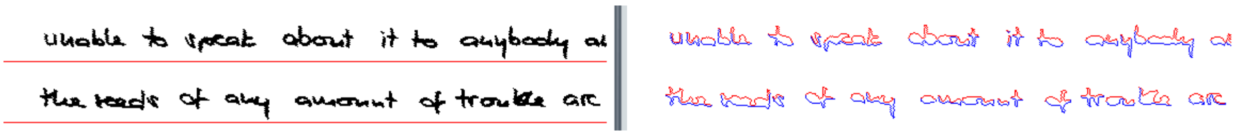


Figure 4: Lower and upper contours: on the left side is the original image and on the right side are extracted contours. Upper contours are illustrated by red curves and lower contours by blue curves.

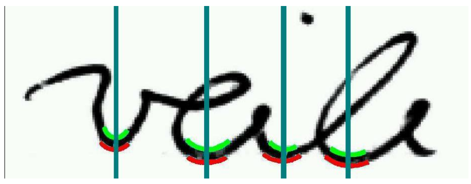


Figure 5: Grapheme segmentation: upper contours are illustrated by green curves and lower contours by red curves. Word 'veile' is divided into graphemes by vertical blue lines. Original figure comes from [8].

3.3 Extraction of features

Individuality of handwriting is based on the idea that the variation within a person's handwriting is less than the variation between the handwriting of two different people [12]. We use the following features:

Proportion
Four baselines are computed for every line (see Figure 6) [9]:

- Ascender line passes through the topmost point of the line (Asc)
- Descender line passes through the bottom point of the line (Des)
- Upper baseline goes through the top of the lower case characters (Upp)
- Lower baseline goes through the bottom of the lower case characters (Low)

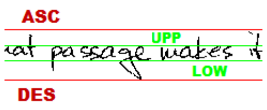


Figure 6: Four baselines

Then the following ratio is computed: (Upp - Asc) : (Low - Upp) : (Des - Low)
Children in the elementary schools are taught to write in a way that yields a 1:1:1 proportion. But later on, when each of us forms his/hers writing, the proportion is changed and that is characteristic for our handwriting. Baselines are a basic feature in handwriting and more of them can be found in [9].

Height of handwriting and distance between lines
Height is computed for every line as absolute value of (Des - Asc). Distances between lines are obtained by line segmentation. Next, the ratio of distance between lines to height is computed for each couple and the result is an average value of all ratios (see Figure 7).

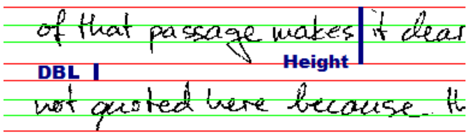


Figure 7: Image figures distance between lines (DBL) and height of handwriting.

Slant
Slant is obtained by slant detection which is part of preprocessing.

Density
Number of black pixels is computed for every line and the result is average of the numbers.

Block letters
Nowadays people often use block letters because of its readibility. They sometimes mix both types, block letters and court hand. We define so-called 'measure of using block letters'. This measure is determined by the next criteria: the more gaps are in the words the more block letters writer use. Result is again an average value of measures of all lines.

Additional features
Design of our system enables us to add another features or to change actual features (e.g. addition of distances between words).

3.4 Feature vector

Each feature is represented by a number (apart from proportion which has 3 numbers). So we obtain a vector of numbers for every image. We have two images as an input. For every image a feature vector is computed. Then a new vector is created by the subtraction of two vectors and its coordinates are changed into absolute values. Then each coordinate is compared with a threshold which was obtained from our experiments (we have unique threshold for each coordinate - for proportion 10, 10, 10, ratio of distance between lines to height 0.5, slant 0.2, density 300, block letters 0.6). If coordinate is smaller than threshold, it's a sign of similarity, if it is bigger, it's a sign of dissimilarity. If we have at least one coordinate bigger than threshold, two input images are marked as written by the different writers. Otherwise the images are marked as written by the same writer. It means that the only one dissimilar feature is enough to determine different writers.

3.5 Graphemes clustering

The features mentioned above represent only a half of the whole solution. An important role is played by the individual graphemes and their clustering. Graphemes are segmented from each image and clustered afterwards.

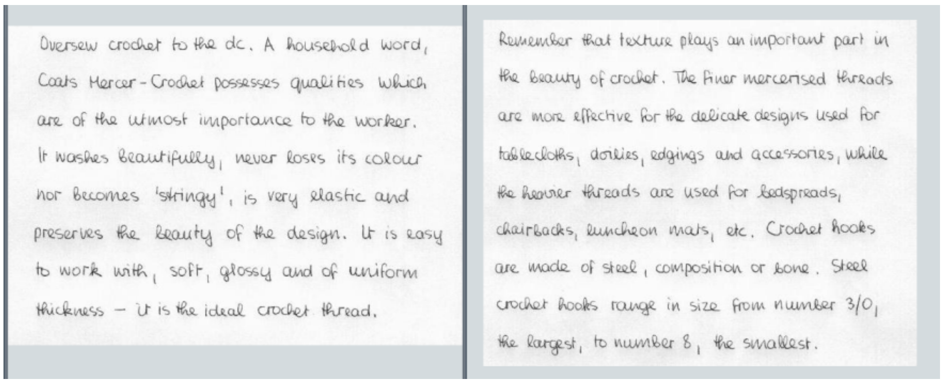
The second system
Inspired by [8], we use Kohonen's Self-Organizing Map (SOM). SOM groups graphemes from both images according to their similarity into the clusters. Similarity of two graphemes is determined by their

Euclidean distance (see Figure 8 a 9). We look at each cluster and find out whether it contains graphemes from both images or only from one image. Result is a number which determines how input images are similar. The more clusters with 'mixed' graphemes exist, the more similar images are. The second system is based on results from first system and results from SOM.

The third system

The second clustering method which we use is hierarchical clustering (HC). As the input we use graphemes from both images. First step is finding for each grapheme the 'closest' pair (using Euclidean distance). Hence we have grapheme pairs (clusters). Second step is analogue with the first, for each cluster we search the closest pair. We produce 2 numbers representing amount of clusters which contain graphemes coming from the same image. First number is obtained after the first step and second after the second step. The third system is based on results from first system and results from HC.

Each resulting number from clustering is compared with two unique thresholds. We got thresholds experimentally, for SOM the thresholds are 55 and 70, for HC 55 and 65 for the first resulting number and 20 and 30 for the second number. If the result is smaller than the first threshold, it's a sign of similarity, if it is bigger than the second threshold, it's a sign of dissimilarity and if it fits between first and second threshold, it's uncertain and we rely on another method. Therefore SOM resp. hierarchical clustering is combined with the first system (which is based on features). If the result of clustering fits between first and second threshold, we take the result from the first system. Otherwise, result from SOM resp. hierarchical clustering is taken.


Oversew crochet to the dc. A household word, Coats Mercer-Crochet possesses qualities which are of the utmost importance to the worker. It washes beautifully, never loses its colour nor becomes 'stringy', is very elastic and preserves the beauty of the design. It is easy to work with, soft, glossy and of uniform thickness – it is the ideal crochet thread.

Remember that texture plays an important part in the beauty of crochet. The finer mercerised threads are more effective for the delicate designs used for tablecloths, doilies, edgings and accessories, while the heavier threads are used for bedspreads, chairbacks, luncheon mats, etc. Crochet hooks are made of steel, composition or bone. Steel crochet hooks range in size from number 3/0, the largest, to number 8, the smallest.

Figure 8: Two different images from the same author

4 Results

4.1 Experiments

We performed our experiments on 100 images from 40 different writers. We made 100 experiments on 2 different images from the same writer and 100 experiments on 2 images from 2 different writers. We tested 3 mentioned systems (see Table 1).

System	writer	NOE	NOCR	NOIR	accuracy	average
1.	same	100	85	15	85%	91%
	different	100	97	3	97%	
2.	same	100	86	14	86%	92%
	different	100	98	2	98%	
3.	same	100	93	7	93%	96,5%
	different	100	100	0	100%	

Table 1: Results of our experiments: signification of shortcuts - NOE (number of experiments), NOCR (number of correct results), NOIR (number of incorrect results). As seen in the table, the best results gives the third system. The first system gives good results in respect of its complexity. However the second system doesn't bring expected enhancement.

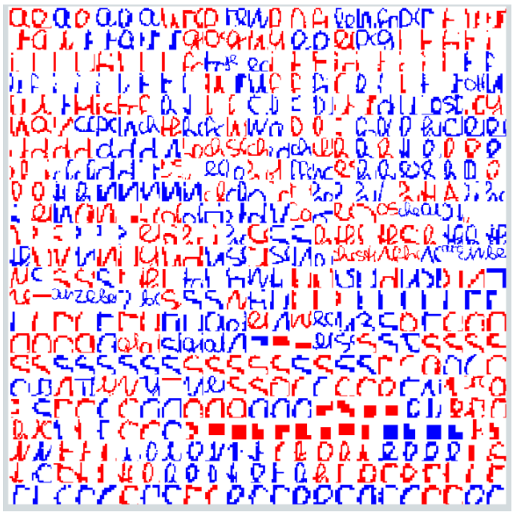


Figure 9: The result of SOM of two images on previous figure. The red graphemes come from grapheme segmentation from the first input image and the blue graphemes come from the second input image. SOM grouped the similar graphemes to the same cluster. All clusters are placed abreast on the image. You can see that SOM places similar graphemes abreast.

4.2 Sources of errors

Results strongly depend on preprocessing. In some cases preprocessing can give incorrect output. For example line segmentation fails if lines are too close and touch each other. Also word segmentation has problem with image where distances between words and distances between characters are similar. If the slant has more directions in the line, for example at the beginning of the line handwriting has tendency to slant to left and at the end to right, it can be problem for our algorithm. Now we explain source of errors for each system:

1. the first system
   errors in experiments on images from 2 different writers: system marked images as written by the same writer because it didn't find any dissimilarity. It is not available to detect differences. We can solve it by adding another features.
   errors in experiments on images from 2 same writers: this errors are more serious because of their quantity. It can be caused by choosing an unsuitable threshold. Another problem is failure of some algorithms (slant detection, proportion of handwriting) which give bad result for particular input.

2. the second and third system
   SOM and hierarchical clustering tend to cluster some particular graphemes that are not significantly similar with respect to the goals of our effort. Errors can be also caused by choosing unsuitable thresholds. When result of SOM resp. hierarchical clustering fits between these thresholds, systems take the results from the first system. So in this case the second resp. third system takes over errors from the first system. We have no errors in experiments on images from 2 different writers in the third system, but we should do more experiments to detect defection of the system.

4.3 Performance

We made experiments using Intel Core 2 CPU 1,66 GHz with 1 GB RAM. Average elapsed times: preprocessing 5 seconds, extraction of features less than 1 second, hierarchical clustering 14 seconds and SOM 52 seconds. Average elapsed times of three systems is shown in table 2 and compared in respect of average accuracy.

5 Conclusions and future work

We presented a combination of particular steps that creates a workflow for identifying and deciding whether two documents were written by the same author or not. Our experiments were made on 100 samples from the IAM database. We tested 3 systems described above and the best result was achieved by the third system (96,5 % accuracy). The presented results show the efficiency of our method. Our future

System	average elapsed time	average accuracy
1.	13 sec	91%
2.	67 sec	92%
3.	27 sec	96,5%

Table 2: Elapsed time vs. accuracy: Experiments show that the second system (with SOM) computes 5 times slower than the first and it brings only 1 % enhancement. Therefore it seems to be the least appropriate. The third system computes 2 times slower than the first and it brings 5,5 % enhancement. So this is our winning system.

tasks are: to bring a better preprocessing, include more handwriting features, accelerate our systems, implement another clustering techniques, compare the results for different thresholds in each system and make more experiments. This process can be utilised into the problem with more than two input images.

6 Acknowledgments

Sincere thanks to my thesis supervisor doc. RNDr. Milan Ftáčnik, CSc., my seminary supervisor doc. RNDr. Andrej Ferko, PhD., Mgr. Matej Novotný, my friend Tomáš Sako and the others for their regulations, suggestions and helpful comments. I am grateful to my family and friends for encouragement through my all studies.

References

[1] IAM Handwriting Database *http://iamwww.unibe.ch/~fkiwww/iamDB*

[2] N. Otsu. *A threshold selection method from gray level histograms.* IEEE Trans. Systems, Man and Cybernetics. 1979.

[3] M. Arivazhagan, H. Srinivasan and S. Srihari. *A Statistical approach to line segmentation in handwritten documents.* Center of Excellence for Document Analysis and Recognition (CEDAR) University at Bualo, State University of New York. 2007.

[4] Sung-Hyuk Cha and Sargur N. Srihari. *Multiple feature integration for writer verification.* Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition. 2000.

[5] A. Bensefia, T. Paquet, L. Heutte. *Grapheme Based Writer Verification.* In 11th Conference of the International Graphonomics Society, IGS'2003, Scottsdale, Arizona. 2003.

[6] Andreas Schlapbach and Horst Bunke. *Using HMM Based Recognizers for Writer Identification and Verification.* Proceedings of the 9th Int'l Workshop on Frontiers in Handwriting Recognition. 2004.

[7] Sargur N. Srihari et al.. *A Statistical Model For Writer Verification.* Proceedings of the Eighth International Conference on Document Analysis and Recognition. 2005.

[8] Marius Bulacu and Lambert Schomaker. *Text-Independent Writer Identification and Verification Using Textural and Allographic Features.* IEEE Transactions on Pattern Analysis and Machine Intelligence. 2007.

[9] Brijesh Verma. *A Contour Code Feature Based Segmentation For Handwriting Recognition.* Proceedings of the Seventh International Conference on Document Analysis and Recognition. 2003.

[10] V.V. Nabíyev, C. Köse, S. Bayrak. *An artificial neural network approach for sign language vowels recognition.* Department of Computer Engineering, Faculty of Engineering, Karadeniz Technical University. 2006.

[11] Moore's algorithm
www.imageprocessingplace.com/downloads_V3/root_downloads/tutorials/contour_tracing_Abeer_George_Ghuneim

[12] Sargur N. Srihari, Sung-Hyuk Cha, Hina Arora, Sangjik Lee. *Individuality of Handwriting.* Center of Excellence for Document Analysis and Recognition (CEDAR), University at Buffalo, State University of New York. 2001.