Global Journal of Computer Science and Technology: F
Graphics & Vision
Volume 15 Issue 2 Version 1.0 Year 2015
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

A Study on Preprocessing and Feature Extraction in offline Handwritten Signatures

By Tansin Jahan, Md. Shahriar Anwar & S. M. Abdullah Al-Mamun
Ahsanullah University of Science and Technology, India

Abstract- In offline handwritten signature verification process, preprocessing of the signature is the very fast and most essential part. In some cases the raw signature can include extra pixel known as noises or may not be in proper form where preprocessing is mandatory. If a signature is preprocessed correctly, it leads to a better result for both signature matching and forgery detection. Pre-processing includes binarization, noise removal, thinning, orientation etc. Many experiments and techniques have already been proposed for implementing these processes and some of them have shown exclusive and spectacular results. Regarding to this situation we have studied several pre-processing steps, signature features, feature detectors and also implemented some of them using MATLAB software. We have studied several image processing algorithms, and proposed an algorithm to correct the alignment of the input signature which can be used at the preprocessing stage to achieve better results in the signature detection process. We have tried to find a baseline of the handwritten signature and align it with respect to the baseline.

Keywords: image preprocessing, edge detection, feature extraction, orientation.

GJCST-F Classification: I.3.2

A Study on Preprocessing and Feature Extraction in offline Handwritten Signatures

Tansin Jahan α, Md. Shahriar Anwar σ & S. M. Abdullah Al-Mamun ρ

Abstract- In offline handwritten signature verification process, preprocessing of the signature is the very fast and most essential part. In some cases the raw signature can include extra pixel known as noises or may not be in proper form where preprocessing is mandatory. If a signature is preprocessed correctly, it leads to a better result for both signature matching and forgery detection. Pre-processing includes binarization, noise removal, thinning, orientation etc. Many experiments and techniques have already been proposed for implementing these processes and some of them have shown exclusive and spectacular results. Regarding to this situation we have studied several pre-processing steps, signature features, feature detectors and also implemented some of them using MATLAB software. We have studied several image processing algorithms, and proposed an algorithm to correct the alignment of the input signature which can be used at the preprocessing stage to achieve better results in the signature detection process. We have tried to find a baseline of the handwritten signature and align it with respect to the baseline. Though there are some limitations involved but the experimentations have shown very promising results.

Keywords: image preprocessing, edge detection, feature extraction, orientation.

I. Introduction

An offline signature consists of some specific characteristics of any individuals which need to be verified for forgery detection. But before that the signature must have gone through some preliminary steps like simplification of the signature, feature extraction, classification etc.

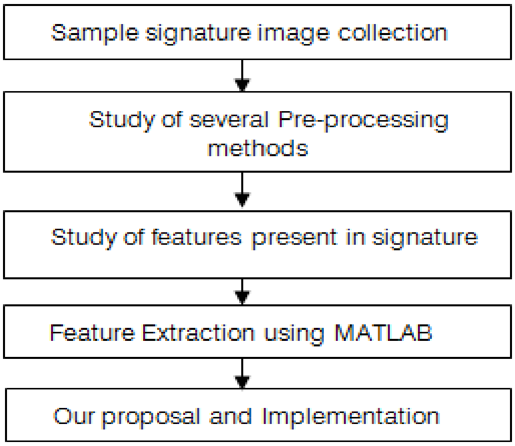
Authenticating an individual using his or her handwritten signature is a biometric process [1], [2] because signature could differ on the basis of movement, position and behavioral state of that person. So it is necessary to look upon these facts while performing forgery detection of any signature.

Matching signature is complicated and time consuming process. So we would like to propose a way to narrow down the possible number of sample input signatures that can probably match with the reference signature present in database. In order to perform similarity checking between the test signature and the reference signature present in database, some procedures needs to be followed-

a. Collect sample signatures
b. Pre-processing
c. Feature extraction
d. Matching and Verification

Pre-processing is the first step for signature matching process where the input signature is simplified to match with referenced one. After the simplification of signature, it is necessary to find which features are present inside the signature. Then extract those features using proper method and classifiers for verification. Features can be divided broadly into three categories - a) Global features b) Local features c) Geometric features [1].

In our study we have focused on pre-processing part of offline handwritten signature verification process. We observed that if the test input signature is not oriented properly then sometimes expected result cannot be achieved. So we tried to find the baseline of the test signature as a part of pre-processing step and proposed an algorithm to make it horizontally aligned before further processing. So the whole study can be expressed by the following flowchart –


Sample signature image collection
Study of several Pre-processing methods
Study of features present in signature
Feature Extraction using MATLAB
Our proposal and Implementation

II. Preprocessing

Preprocessing is necessary to remove background noises and to achieve better result [3]. Some common operations of pre-processing is known as binarization, thinning, smoothing, noise removal,

Author α: Computer Science and Engineering Department, Daffodil International University. e-mail: tansin.jahan@gmail.com
Author σ: Computer Science and Engineering Department from Ahsanullah University of Science and Technology, Bangladesh. e-mail: anwarshahriar@gmail.com
Author ρ: Computer Science and Engineering Department from Ahsanullah University of Science and Technology, Bangladesh. e-mail: almamun@aust.edu

slant normalization, skeletonization, orientation, size normalization [1][3][4] etc.

a) Thresholding

In MATLAB , the input signature must be in gray scale or binary otherwise there will not be any output. A binary image has only two possible values 1 and 0 (respectively black and white) [1].To get the binary image of any signature, a threshold must be declared. Depending on various method, threshold value can be between 0 and 1 [4]. For our experiment we assumed that the threshold point is at 0.5 level and converted the following RGB signature image into binary image.

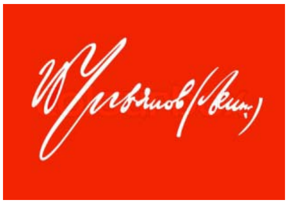


Figure 1 : RGB signature

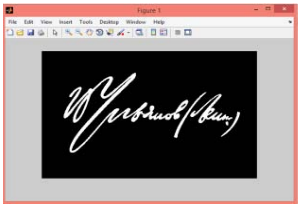


Figure 2 : Binary signature using threshold 0.5

b) Edge Thinning

Here morphological operators which are connected to the set theory of mathematics [4][5] are useful to thin edges of any binary pre processed signature. So we applied morphological operator where operation can be 'thin', 'branchpoint', 'endpoint', 'bridge' etc. Now binary image of Fig-2 can be thinned using 'thin' morphological operation [5] , which removes pixels so that an object without holes shrinks to a minimally connected stroke.

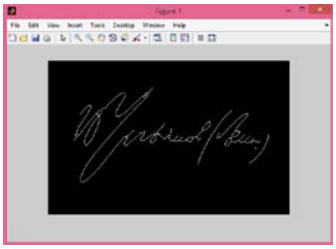


Figure 3 : Thinned signature

c) Noise Removal

Noise is another important factor of offline handwritten signature pre-processing which can be removed using different filtering like median filtering, adaptive filtering and so on. Our signature can have luminance noise or color noise which must be removed before feature extraction and further processing to avoid unwanted result [6]. Image restoration therefore tries to recover an image by removing unnecessary pixels and noises [7]. Several types of noises can be - i) Gaussian noise ii) Sault and pepper noise etc.[5][7].

d) Noise Removal with adaptive filtering

We used wiener2 function to remove noises. wiener2 lowpass filters an intense image that has been degraded by constant power additive noise.

As we can see Gaussian noise can be detected in Fig-4 which is later removed using wiener2 (varargin) function.

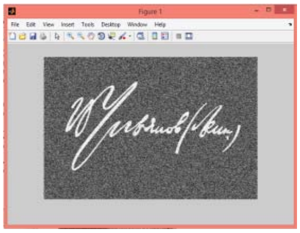


Figure 4 : Signature with Gaussian noise

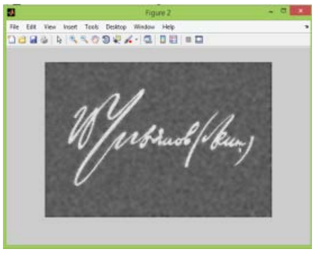


Figure 5 : Signature without noise

III. Feature Extraction

a) Loops present in signature

Loops which is known as a writing pattern of any handwritten signature presents 'holes' and can be classified as natural loop or artificial loop[8]. bwboundaries() function in MATLAB, traces the exterior boundaries of objects, as well as boundaries of holes inside these objects, in the binary image. We can also specify an optional value ('holes' or 'no holes') where 'holes' represents searching of both object (parent and child) and hole boundaries and it is default. On the other hand 'no holes' can search only for object boundaries.

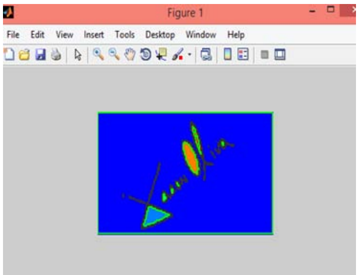


Figure 6 : loops in signature is highlighted

b) Pixel transformation of signature into binary values

As we know, a binary image can be represented by black and white pixel. So to transform a binary image into a text file we specified white pixel as 1 and black pixel as 0. This kind of pixel transformation is needed to create a 2D array of signature from which expected mean [9] of an average data set can be calculated using formula.

We have followed some steps while performing the transformation of signatures –

1. An input of a gray scale image.
2. Detection of edges of the input image using canny algorithm.
3. Conversion of this binary image to a text file.
4. Calculation of the number of '1' in every column of image matrix.

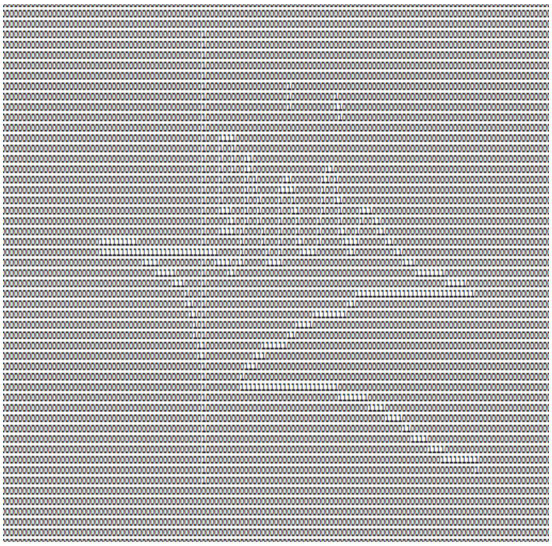
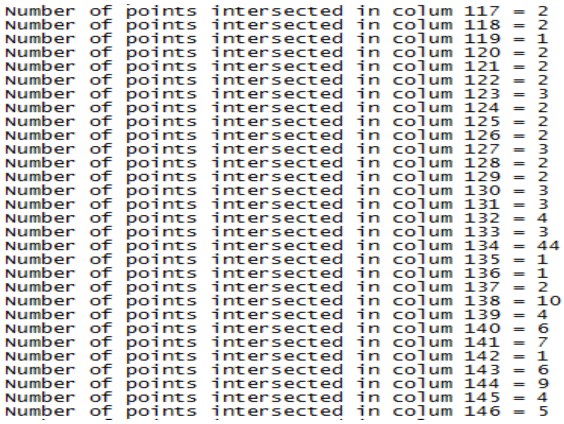


Figure 7 : Binary image normalized to 0's and 1's



```
Number of points intersected in colum 117 = 2
Number of points intersected in colum 118 = 2
Number of points intersected in colum 119 = 1
Number of points intersected in colum 120 = 2
Number of points intersected in colum 121 = 2
Number of points intersected in colum 122 = 2
Number of points intersected in colum 123 = 3
Number of points intersected in colum 124 = 2
Number of points intersected in colum 125 = 2
Number of points intersected in colum 126 = 2
Number of points intersected in colum 127 = 3
Number of points intersected in colum 128 = 2
Number of points intersected in colum 129 = 2
Number of points intersected in colum 130 = 3
Number of points intersected in colum 131 = 3
Number of points intersected in colum 132 = 4
Number of points intersected in colum 133 = 3
Number of points intersected in colum 134 = 44
Number of points intersected in colum 135 = 1
Number of points intersected in colum 136 = 1
Number of points intersected in colum 137 = 2
Number of points intersected in colum 138 = 10
Number of points intersected in colum 139 = 4
Number of points intersected in colum 140 = 6
Number of points intersected in colum 141 = 7
Number of points intersected in colum 142 = 1
Number of points intersected in colum 143 = 6
Number of points intersected in colum 144 = 9
Number of points intersected in colum 145 = 4
Number of points intersected in colum 146 = 5
```

Figure 8 : Number of non-zero values in each column of the matrix

IV. Our Proposal

Anoise free perfectly oriented signature sample can help any signature matching algorithm to perform better. There are several function in MATLAB based on each connected component of binary image such as regionprops(), to measure orientation properties of signature region [10].

Orientation of a leaf is easy to determine, because there is a long axes in the middle of every leaf that makes a standard form of a leaf. But in context of a signature there is no standard way to know whether the signature is written vertically or horizontally or in any angle.

So, we observed many input signatures and drew number of regions vs. number of pixels graphs to find a standard which is mostly common in every signature. After a long time observation we've found a pattern that is common to the most of the input signatures. Here's the few sample images we've used to work with our algorithm:

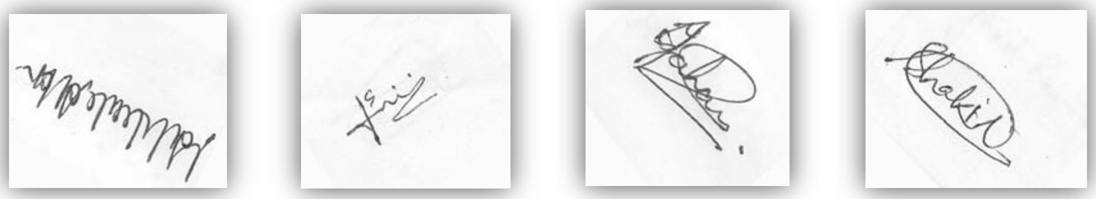


Figure 8 : Sample signatures used in our algorithm

a) Our proposed algorithm

We are proposing a simple algorithm to make the sample signatures horizontally oriented. This algorithm can be used in the preprocessing stage of the signature.

This algorithm finds a baseline of the given signature, and then it rotates the image to set it in parallel with x- axis. We have made some assumptions to make the algorithm work correctly, which are:

1. Signature must be written from left to right or right to left manner

2. Background noises should be removed as much as possible
3. Input sample must be a gray scale image

Now the working procedure of the proposed algorithm is given below:

1. It takes an input signature as a gray scaled image.
2. It converts the input to a binary image applying canny edge detection algorithm.
3. It rotates the signature image in steps until a full 360 degree rotation occurs with a user defined angle in each step and each time does the following:

a) Finds the top, right, bottom and left boundary of the signature

b) Finds the current height of the signature

c) Divide the signature equally to a predefined number of rectangles

d) For each rectangle it counts the image pixels in the rectangle and remembers the rectangle with the highest number of pixels and current image orientation.

4. Finds the orientation with the maximum number of pixels in a rectangle.

b) Experimental result

We have experimented with a set of 100 signatures to find out a common pattern in those. After running our algorithm for the given input signature in MATLAB, we have found some interesting result. Fig.9. depicts these resulted signature and we could easily distinguish that all resulting signatures has rotated horizontally.

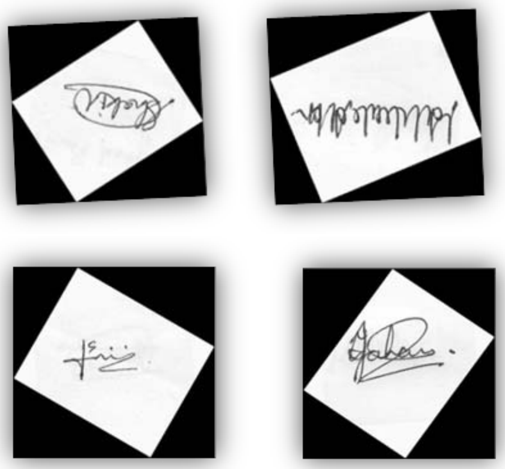


Figure 9 : After applying the Rotation Fix algorithm on the sample images

After cropping the resulting images, we've found some nicely cropped horizontally oriented signature samples as we can observe in the following figure –

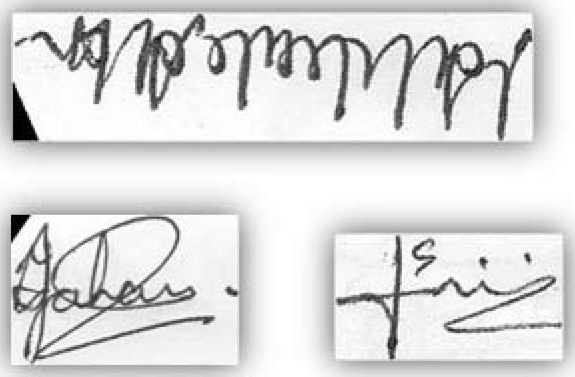


Figure 10 : Cropped signature images after correcting the orientation

From our experiment we could say that, if our input signature image is given in any orientation then by running the algorithm mentioned above we could easily make it horizontally oriented on the basis of the maximum number of pixel could be found when the signature is horizontal.

c) Limitations of our Algorithm

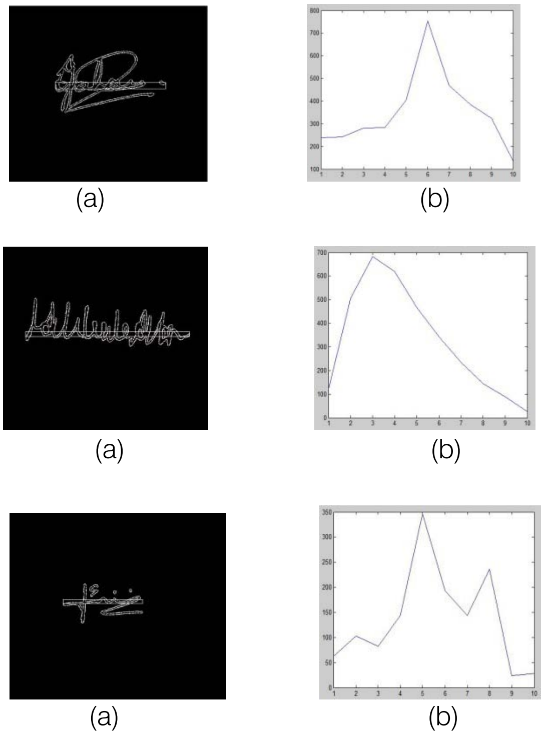
Although this algorithm returns fair results in most of the cases, it has some limitations which are given below:

1. In the process finding correct rotation angle of the signature image, it may go upside down.
2. The image actually may not be fully horizontally oriented, although it may turn out to be very close to that.
3. It is sensitive to background noise.

d) Idea behind our proposal

After thorough observation over a set of 100 signatures we've found that every signature has a pattern. Precisely saying that in every signature there is a dense region or most occupied region which contains more pixels than any other region of that signature.

As a matter of fact every signature is different from the other, and so it is very hard to find a single line as a base line of a signature. That is the reason we have taken the entire region as a base line which contains the most of the pixels of the signature. We have drawn graphs for every image and reached to the above conclusion. Some of our experimental data are given below:



(a) (b)

(a) (b)

(a) (b)

Figure 11 : (a) Most occupied region (here we've divided the image into 10 regions)

(b) Number of regions vs. number of pixels graph

V. Conclusion

We have tried to give a brief description above about our study related to offline signature pre-processing and feature extraction. We have mentioned our specific field of interest and the objectives of the study.

Offline handwritten signature includes many features and patterns that are very important for purposes like forgery detection. In our study we have tried to focus on some of these features such as loops in signatures, pixels transformations of signature into binary image etc.

Another main part of our study was the pre-processing of handwritten signatures which makes it easier to find the similarities among signatures of the same person. We have also proposed a new algorithm for orienting any signature based on its dense region and implemented it. We enjoyed our work very much and hope to have opportunity for further study and improvement for the mentioning algorithm including complexity and many other factors.

References Références Referencias

1. Madasu, Vamsi Krishna & Lovell, Brian C. (2008).An Automatic Offline Signature Verification and Forgery Detection System (chapter 4). In Brijesh Verma and Michael Blumenstein (Ed.), Pattern recognition technologies and applications: Recent advances (pp. 63-89) Hersey, PA: Information Science Reference.
2. Gulzar A. Khuwaja & Mohammad S. Laghari. Offline Handwritten Signature Recognition. World Academy of Science, Engineering and Technology,Vol:5 2011-11-25.
3. Dakshina Ranjan Kisku, Phalguni Gupta & Jamuna Kanta Sing(July,2010). Offline Signature Identification by Fusion of Multiple Classifiers using Statistical Learning Theory. International Journal of Security and Its Applications , Vol. 4, No. 3.
4. Bassam Al-Mahadeen, Mokhled S. AlTarawneh and Islam H. AlTarawneh (March, 2010).Signature Region of Interest using Auto cropping. IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 2, No 4.
5. Ravneet Gill, Maninder Singh (December,2012). Statistical Features Based Off Line Signature Verification System using Image Processing Techniques. International Journal of Science and Research (IJSR), Vol.1, Issue 3.
6. Priya Metri & Ashwinder Kaur. Handwritten Signature Verification using Instance Based Learning. International Journal of Computer Trends and Technology- March to April Issue 2011. ISSN:2231-2803.
7. Dr. Philippe Cattin (April 2012). Image Restoration: Introduction to Signal and Image Processing. MIAC, University of Basel.
8. Tal Steinherz, David Doermann, Ehud Rivlin and Nathan Intrator (February 2009). Offline Loop Investigation for Handwriting Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 31, No. 2.
9. Debnath Bhattacharyya, Samir Kumar Bandyopadhyay, Poulami Das, Debashis Ganguly and Swarnendu Mukherjee. Statistical Approach for Offline Handwritten Signature Verification. Journal of Computer Science 4 (3): 181-185, 2008. ISSN 1549-3636.
10. Suhail Odeh and Manal Khalil. Apply Multi-Layer Perceptrons Neural Network for Off-line signature verification and recognition. IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 6, No. 2, November 2011. ISSN (Online): 1694-0814.

Global Journals Inc. (US) Guidelines Handbook 2015

www.GlobalJournals.org