PROPOSED PREPROCESSING ALGORITHM FOR SIGNATURES RECOGNITION

Nada Abdullah Rasheed
University of Babylon
College of Basic Education

Abstract

Signatures are used everyday to authorize the transfer of funds of millions of people, Bank checks, credit cards and legal documents, all of which require our signatures.

Different people sign their signatures with different orientation, size, deviation, etc. Even the signatures of the same individual change temporarily in the aforementioned attributes under different circumstances (e.g. the size of the signing space). To minimize the variation in the final results, all signatures are normalized for duration, rotation, position and size. The preprocessing includes some operations such as scaling and rotation to get a fixed size and direction.

Introduction

Since the beginning of the computer industry, users of computers have been forced to modify their behavior to utilize these devices. User interfaces ranged from confusing to downright hostile. May be the computer should be able to read our handwriting [1].

The benefits of an automatic handwriting recognition system have long been known [2]. A handwritten signature is a kind of agreement. Mostly, it is an agreement to the content of a document. It can be the signature on a contract, an application from an official document [3].

Handwritten signatures come in many different forms and there is a great deal of variability even in signatures of people who use the same language. Some people simply write their name while others may have signatures that are only vaguely related to their name, some signatures may be quite complex while others are simple and appear as if they may be forged easily. It is also interesting to note that the signature style of individuals relates to the environment in which the individual developed their signature. It is known that no two genuine signatures of a person are precisely the same and some signature experts note that if two signatures of the same person written on paper were identical they could be considered forgery by tracing. Successive signatures by the same person will differ and may also differ in scale and orientation [4].

However, for several reasons the task of verifying human signatures cannot be considered a trivial pattern recognition problem. It is a difficult problem because signature samples from the same person are similar but not identical. In addition, a person’s signature often changes radically during his life [5].

The description of the stages sequence of the proposed system will be given, which includes some modifications of: **Image Loading, Noise Removal, Image Scaling-1, Image Centralization-1, Image Rotation, Image Trimming, Image Scaling-2, Image Centralization-2.**

The system is divided into a set of procedures, each of which does a specific job; the result is then given to the post procedure.

Step (1): Image Loading

This procedure is responsible for loading the bitmap (BMP) image file into memory. The dimension of the image used in this work is (250x250) pixels.

Step (2): Preprocessing

The signature images require some manipulation before the application of any recognition technique. This process prepares the image and improves its quality in order to eliminate irrelevant information and to enhance the selection of the important features for recognition and to improve the robustness of features to be extracted. Moreover Preprocessing steps are performed in order to reduce noise in the input images, and to remove most of the variability of the handwriting.

Step(2.1): Noise Removal

In this work an algorithm was designed to remove these objects. This is done through the use of windows for testing the noisy objects in the image, which are within the window border, after testing the existence of these objects and checking whether they are isolated from the signatures or not. If they are isolated, then they will be removed, as in the figure (1). The following algorithm represents this procedure:

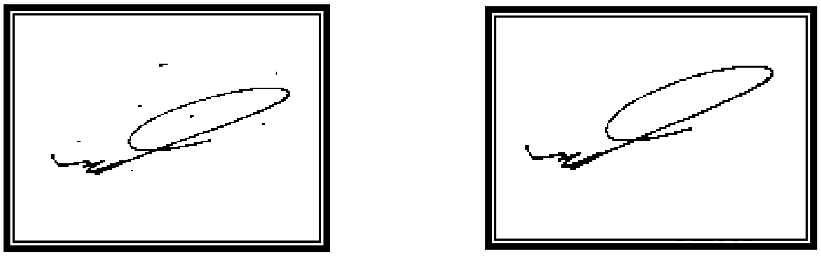


Figure (1): Noise Removal.

Step(2.2) Image Scaling-1

When the rotation of object is done in a limited array and the object is a little smaller than the array size that may cause sometimes the object to cut. The image scaling-1 is therefore used to minimize the object size at a fixed ratio equals (0.7071068) to facilitate the rotation operation and this ratio is calculated as follows.

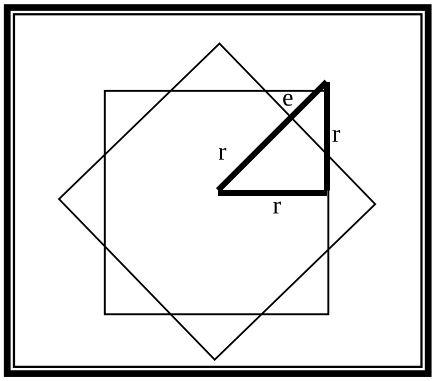


Figure (2): illustrate the ratio that reduced signature size.

As shown in figure (2), the radius (r) divides the side opposite to the center into equal parts. Thus we obtain right triangle. Applying the low of Pythagorean, the hypotenuse can be calculated from equation (1).

$$x^2 = \sqrt{r^2 + r^2} \qquad (1)$$

$$x = \sqrt{2}\,\mathrm{r}$$

Where, (x) is the hypotenuse.

When the image is rotated its sides intersect with the original image angles with a magnitude (e) calculated in (2):

$$\mathrm{e} = \sqrt{2}\,\mathrm{r} - \mathrm{r} \qquad (2)$$

$$\mathrm{e} = \mathrm{r}\,(\sqrt{2} - 1)$$

Thus the image is reduced by the ratio (Sc), as shown in figure (3a,b):

$$\mathrm{Sc} = \frac{\mathrm{r}}{\sqrt{2}\,\mathrm{r}} \qquad (3)$$

$$= \frac{1}{\sqrt{2}}$$

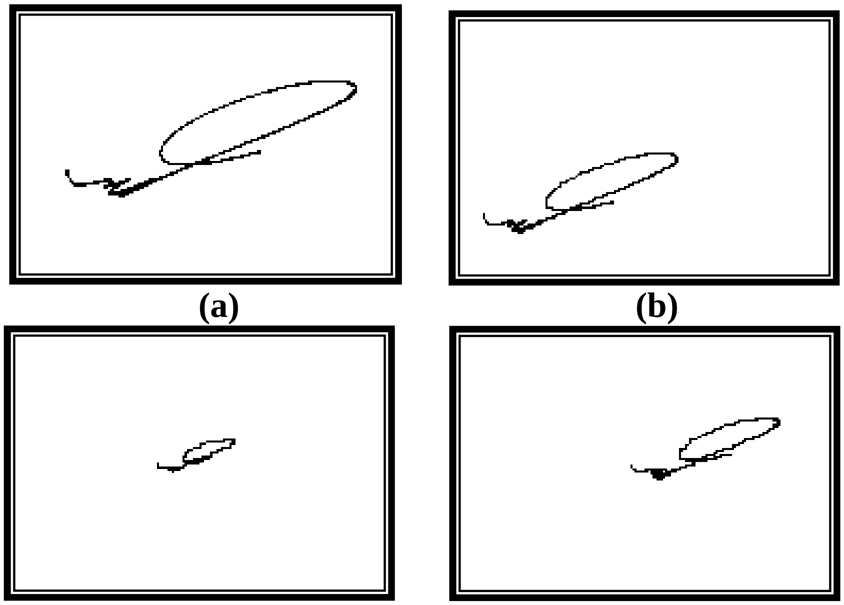
$\mathrm{Sc} = 0.7071068$ Where, (Sc) is the ratio that reduced signature size.

On the other hand as in figure (3c,d) the object sometimes is very small size to rotate. Thus the image scaling-1 is used to maximize the object size at a fixed ratio to facilitate the rotation operation, and this ratio is calculated as follows:

$$\mathrm{Sc} = \frac{50}{\mathrm{R} - \mathrm{L}} \qquad (4)$$

Where, (Sc) is the ratio of signature size magnification

(L, R) are the right and left first on-pixel of the signature body.



(a) **(b)**

(c) **(d)**

Before **After**

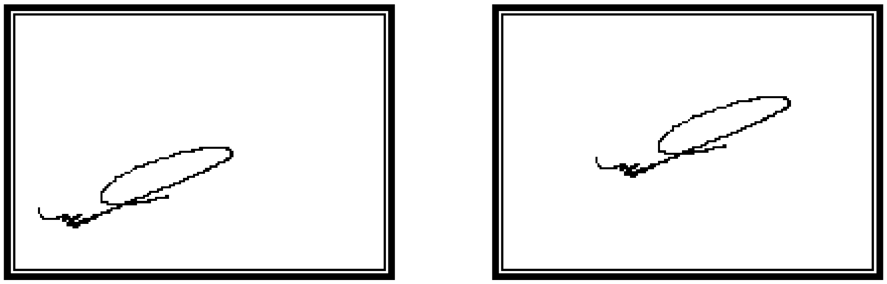
Figure (3): Image Scaling-1.

<u>**Scaling-1 Algorithm1**</u>

```
  Input:
 $X^{old}$ , $Y^{old}$   The pixel which will be scaled
 L, R, U, D   The boundaries of the signature body.
  Output:
 $X^{new}$, $Y^{new}$  The scaled pixel.
 Program body:
  Step 1: Compute the difference between left and right pixel of the signature
      body to scale by minimizing or maximizing ratio procedure
          if   R- L <50  then  scale should be maximizing ratio
  Step 2 :  Compute Sc= 50/ (R-L)                                (5)
  Step2.1: for  i=0 … n
   Step2.2: for j=0 … m
          $X^{new}$ = round ( $X^{old}$ * Sc)                    (6)
          $Y^{new}$ = round ( $Y^{old}$ * Sc)                    (7)
   end for
   Step2.3: scaled pixel at [$X^{new}$ , $Y^{new}$ ]
  Else scald should be minimizing ratio
        Step 3: Sc= 0.7071068 as explained before
 Step3.1:  for i=0 … n
 Step3.2:  for j=0 … m
          using equations (6) and (7) to get $X^{new}$ , $Y^{new}$ .
 Step3.3: scaled pixel at [$X^{new}$ , $Y^{new}$ ]
      end for
```

Step(2.3) Image Centralization-1

This procedure is important to the next step (image rotation), so that the image is rotated exactly, as in figure (4). When image scaling-1 algorithm is done, the position of the signature is changed therefore, this procedure is necessary to drag the signature to the center, which facilitates the image rotation around the center of the image. Here the centralization is done according to X-axis and Y-axis. Therefore the rotation operation is very dependent on image centralization.



Before **After**

Figure (4): Image Centralization-1.

Centralization-1 Algorithm

```
  Input:
  $X^{old}$ , $Y^{old}$    The pixel which will be centralized
  Sx, Sy        The ratio that centralization image according to X-axis
                and Y-axis.
 Output:
  $X^{new}$, $Y^{new}$   The centralized pixel.
 Program body:
  Step1: for i=0…n
   Step2: for j=0 …m
         if   ($X^{old}$ + Sx , $Y^{old}$ + Sy) between boundaries of the image

            $X^{new}$ = round ( $X^{old}$ + Sx)                    (8)
            $Y^{new}$ = round ( $Y^{old}$ + Sy)                    (9)
 Step3: centralized pixel at [$X^{new}$ , $Y^{new}$ ]
    end for
```

Step(2.4) Image Rotation

It is well known that a person's situation differs in each signature from time to time or at the same time. There are many reasons behind this, such as psychology, health, or others. This leads to changes in inclination angles of the same person's signatures. Hence, the rotation algorithm must be used to unify signature orientation in a horizontal manner to overcome this problem.

It is important to compute the angle (θ), which is used in the rotation operation. The rotation of an image requires the calculation of a new position for each point of the image after the transformation. Each image point is rotated through an angle (θ) about the origin, which varies from one signature to other and can be calculated according to the inclination angle. The following Algorithm is used for this purpose.

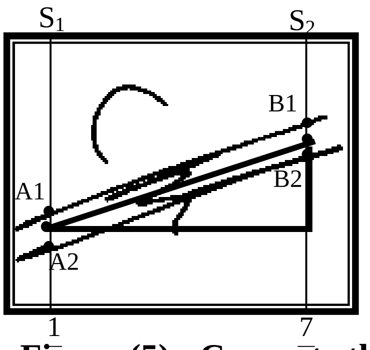


Figure (5): Compute the angle (θ).

Step1: Divide the image vertically into (8) equal sections. Consider the first and last section (S_1) and (S_2).

Step2: Calculate the first point in the signature from top to the base in (S_1) and (S_2).

Step3: Calculate the average of the two side points:

$$C_1 = \frac{A_1 + A_2}{2} \quad (10)$$

$$C_2 = \frac{B_1 + B_2}{2} \quad (11)$$

Step4: Connect C_1 and C_2.

Step5: Calculate the lengths of the triangle sides which is formed from the points coordinates C_1 (x_1,y_1) and C_2 (x_2,y_2) by using :

$$\theta = \tan^{-1}\left(\frac{y_2 - y_1}{x_2 - x_1}\right) \quad (12)$$

as shown in figure (6), The signature is rotated through an angle (θ).

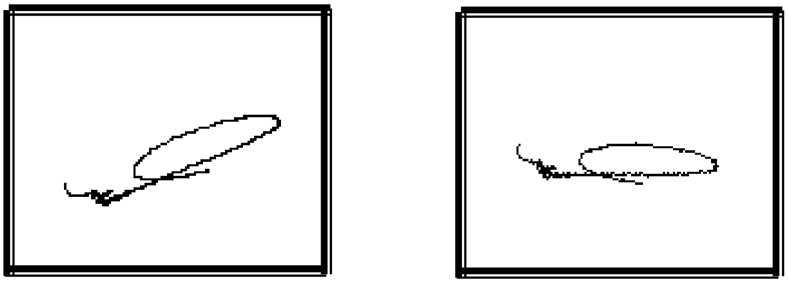
```
Rotation Algorithm:
Input:
 $X^{old}$ , $Y^{old}$          The pixel which will be Rotated
L, R, D, U                      The boundaries of image

Output:
$X^{new}$, $Y^{new}$            The Rotated pixel
$\theta$                        is the angle of rotation
Program body:
Step1 : Calculating the angle ($\theta$)
Step1.1: Calculating ($S_1$, $S_2$) which are represented the number of
            columns in the left and the right of object to cutting
            the signature in the first On-pixel for up and down.

         $S_1=L+(R -L +1.0) /8.0$                                (13)
         $S_2=L+7.0*(R -L+1.0) /8.0$                             (14)
 Step1.2: Calculating the averages between first on-pixel for up and
          down the cutting points
          $C_1$, $C_2$
 Step1.3:

          Using the equation (12)
 Step2:
 Step2.1: for i=0 … n
 Step2.2: for j=0 … m
     $X^{new} = X^{old} * Cos(\theta) - Y^{old} * Sin(\theta)$   (15)

     $Y^{new} = X^{old} * Sin(\theta) + Y^{old} * Cos(\theta)$   (16)
 Step3:  rotated pixel at [$X^{new}$ , $Y^{new}$ ]
 end for
```

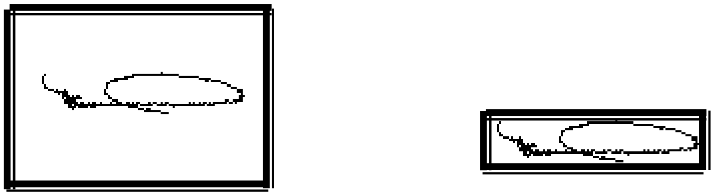


Before **After**

Figure (6): Image Rotation.

Step(2.5) Image Trimming

Scanned signature file is an area that contains signature image. Usually the scanning image for any signature may consist of the area of the signature and its surrounding. Thus the image may include an additional empty lines and columns that have no data (space lines). These empty lines should be eliminated by tracing from outside margins towards inside and stopped at the first occurrence of on-pixel at each side of the four edges, as in figure (7).



Before **After**

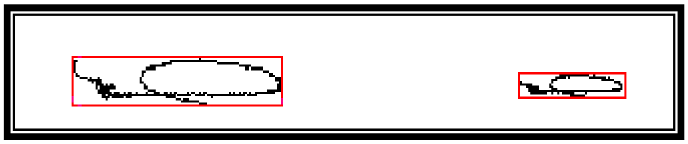
Figure (7): Image Trimming.

Step(2.6) Image Scaling-2

Different people sign their signatures with different size, even the signatures of the same individual change temporarily in the aforementioned attributes under different circumstances. To minimize the variation in the final results, all signatures are scaled at fixed size. The following algorithm clarifies this procedure and figure (8) shows the above algorithm:

```
Scaling-2 Algorithm
  Input:
  $X^{old}$ , $Y^{old}$          The pixel which will be scaled.
  Output:
  $X^{new}$, $Y^{new}$          The scaled pixel.

  Program body:
  Step1: $Sc= 50/ R-1$
  Step2: for  $i=0 \dots n$
  Step3: for $j=0 \dots m$
          using equations (6), (7) to get $X^{new}$ , $Y^{new}$ .
  Step5:
          using the same equations above to get $N^{new}$ , $M^{new}$ the new
          image dimensions
```



Before **After**

Figure (8): Image Scaling-2.

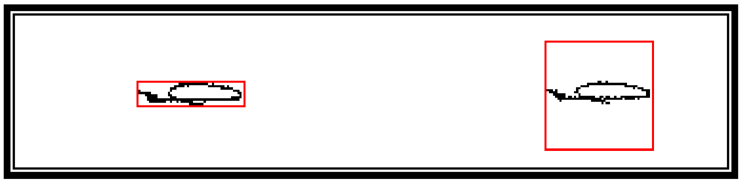
Step(2.7) Image Centralization-2

The purpose of this procedure is to keep the image dimensions fixed after the scaling operation is finished. This procedure is done according to y-axis. The following algorithm clarifies this procedure and figure (9) shows the above algorithm:

```
Centralization-2 Algorithm
 Input:
The centralization is done according to Y-axis only.
 L, R, D, U The boundaries of signature body
 Output:
$Y^{new}$   The centralized pixel

Program body:
Step1: $Sc = 50 / r-l$
          $Sy = (50 - U - D)* Sc /4$                                  (22)
Step2: for i=0 … n
Step3: for j=0 … m
         if  $Y^{old}$ + Sy between boundaries of the image
         using equation (9) to get $Y^{new}$ .

Step4: centralized pixel at [X , $Y^{new}$ ]
   end for
```



Before **After**

Figure (9): Image Centralization-2.

Overview

In the available literature, the handwritten signature recognition problem has been approached in various ways.

In general, handwritten recognition is a difficult task because of the variation of writing styles even with the same writer; therefore, great attentions must be taken in designing a recognition system.

The current research presents satisfactory results in the recognition of handwritten signatures using Backpropagation Neural Network.

Conclusions

The SRS given a high proportion of achievement in the recognition, amounts to (98%), (100) test signatures were chosen out of (50) signatures belonging to persons whose signatures are existents including the database and other (50) signatures not included in the database.

The rotation algorithm must be used to unify signature orientation in a horizontal manner to overcome a person's situation that differs in each signature from time to time or at the same time. The use of the SRS without Rotation operation provides a proportion of achievement in recognition, which amounts to (20%).

References

1- Mohr, D., Pino, D. and Saxe, D. "**Comparison of Methods for Digit Recognition** ", Wilmingtion, DE, USA, 1999.

2- Wilson, A. T., **"Off-line handwriting Recognition Using Artificial Neural Networks"**, University of Minnesota, 2000.

3- Scheibelhofer, K., thesis in **"Signing XML Documents and the concept of What you See Is What You Sign"**, Graz University of Technology, P.11, 2001.

4- McCabe, A., **"A Review of Dynamic Handwritten Signature Verification"**, James Cook University, P.7, 1998.

5- Abbas, R., thesis in **"Backpropagation Networks Prototype for Off-line Signature Verification"** 1995.