Asian Journal of Information Technology 5 (3) : 306-309, 2006

Handwritten Signature Verification Using P-tree

A.K.M. Najmul Islam
ZTE Corporation 80, Park Road, Baridhara, Dhaka-1212, Bangladesh

Abstract: Handwritten signature recognition is a very complex task. At present many implementations are available for recognition of handwritten signatures. But none of them gives the hundred percent performances and their implementation is also very complex. This paper presents the implementation of handwritten signature verification system using the P-tree. The implementation of this system is very straightforward. This system takes a scanned image of the handwritten signature and tries to match it with already stored images in the database. This method uses the concept of P-tree. The signatures are converted into P-trees and then stored into the database. The signature, which needs to be verified, is also converted into a P-tree.

Key words: Image processing, P-tree, Root Count (RC), Quadrant ID (Q-ID)

INTRODUCTION

The increasing number of transactions in bank and offices are needed to be authorized. Signature verification is one of the most acceptable means of verifying check/document for authorization. Since manual verification for a large amount of check /documents is tedious, the methods of automatic signature verification must be developed for electronically verifying a person's identity.

Generally, there are two categories of signature verification system[1,2] according to the acquisition of data:

- On-line and
- Off-line

In this study we have tried to implement the off-line verification system. The off-line signature verification system will process a handwritten document that is scanned by a scanner.

PREVIEW OF THE SYSTEM

There are four main parts of the Automatic Handwritten Verification System (AHVS). These are image processing, feature extraction, database searching and data compression.

Image processing: A scanner is used to scan the signature. The scanner should have good resolution.

There will be some problems in this case. And that is there will be some noise in the scanned image. Various algorithms exist for eliminating the noise. The edge detection is also performed. Another problem is that the quality of the image can become different due the ink of the pen by which the signature is taken. Sometimes it can be thicker and sometimes not so thick. To eliminate this problem the image is first thinned up-to a standard line image. After this we have to scale the image of the signature to a standard size.

Feature extraction: Feature extraction plays an important role in the recognition system. This reduces the number of computations. In this section the image of the signature is converted into a real valued vector form of 0s and 1s. To achieve this an MXN image is reduced to a (M/R1)X(N/R2) size image with the reduction factor s R1 across height and R2 across width. The cell value of the reduced image is calculated using R1XR2 window and calculating the number of black pixels within each window. The algorithm is presented below:

```
FOR I=1 to M/R1
 FOR J=1 to N/R2
 BEGIN
    BlackPixelCount=0.0
 FOR K= (I-1)*R1 to I*R1
  FOR L= (J-1)*R2 to J*R2
 BEGIN
 IF(ReadPixel(K,L)=BlackPixel) then
  BlackPixelCoun=BlackPixelCoun+1
 END
 IF(BlackPixelCoun/(R1*R2)>=0.5) then
```

```
        FeatureMat(I,J)=1
    ELSE
                FeatureMat(I,J)=0
    END
```

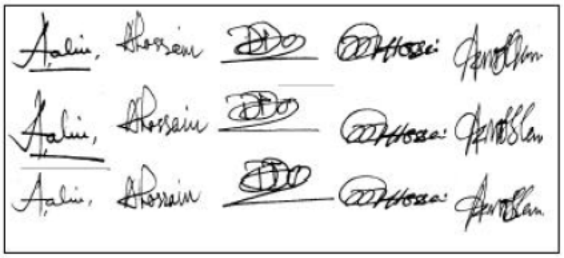

Fig. 1: Sample of the signature

Fig. 2: A scaled image

Database searching: Efficient searching methods must be used to reduce the searching time of the database and improve the performance of the system.

Data compression: The data compression method is another important issue. The compression method should be loss less for better performance.

PROPOSED SYSTEM

Before we introduce the system, some introduction to the P tree is required.

Introducing P-Tree: A P-Tree is a quadrant-based tree. It recursively divides the entire image into quadrants and records the count of 1-bits for each quadrant, thus forming a quadrant count tree.

An example is given in the Fig. 3. It is an 8-row-8-collumn image and its P-tree is drawn in Fig. 4.

Sample data

1 1	1 1	1 1	0 0
1 1	1 1	1 0	0 0
1 1	1 1	1 1	0 0
1 1	1 1	1 1	1 0
1 1	1 1	1 1	1 1
1 1	1 1	1 1	1 1
1 1	1 1	1 1	1 1
0 1	1 1	1 1	1 1

Fig. 3: 8-row-8-collumm image

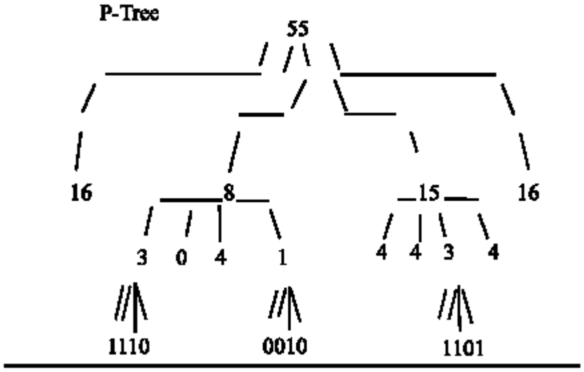

Fig. 4: P-tree of the image

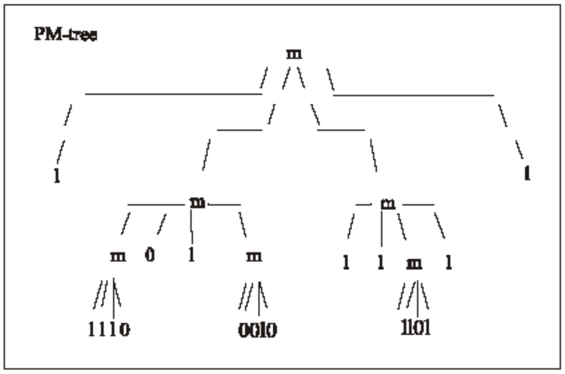

Fig. 5: PM-tree of the image

In this example 55 is the total number of 1's. It is the root of the tree. It is labeled as level 0. it has four children at level 1. These are 16, 8, 15, and 16. These numbers are the 1-bit counts for the four major quadrants in raster order or Z order (upper left, upper right, lower left, lower right). The first and the last quadrants are entirely composed of 1's. These quadrants are called pure-1 quadrants. Here sub trees are not required. Again the quadrants composed of entirely 0's are called pure-0 quadrants. It does not require making sub trees. In other cases the sub trees are made by following the recursive Z ordering.

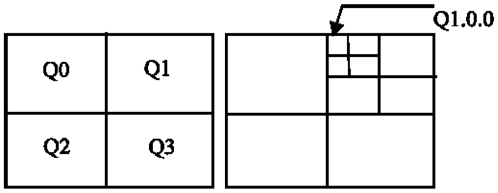

Fig. 6: Quadrant ID of a P-tree

0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0
0	0	1	0	0	1	0	0	0	1	1	0	0	0	0	0
0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
0	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0

Fig. 7: Dividing image into 16 rows and 16 columns

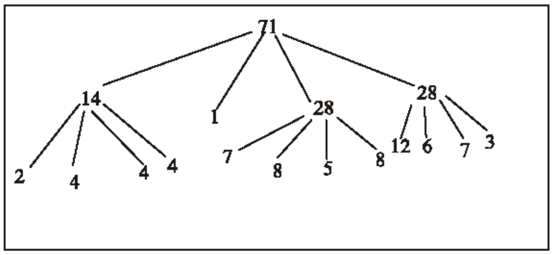

Fig. 8: P-tree for Fig. 5 (3 levels are shown)

Peano Mask tree (PM-tree): A variation of the P-tree data structure, the Peano Mask Tree (PM-tree), is a similar structure in which masks, rather than counts, are used. In a PM-tree, we use three value logic to represent pure-1, pure-0, and mixed quadrants. (A 1 denotes pure-1; 0 denotes pure-0; and m denotes mixed.) The PM-tree for the previous example is also given in Fig. 4. We can easily construct the original P-tree from its PM-tree by calculating the counts from the leaves to the root in a bottom-up fashion. Since a PM-tree is just an alternative implementation for a Peano Count tree, for simplicity we will use the same term, P-tree, for Peano Mask tree.

Root-count and quadrant ID of a P-tree: The total number of 1's in a P-tree is called the Root Count (RC). For example, The P-tree shown above has root count 55.

When we are constructing the P-tree, we divide the original image into four quadrants, the upper left quadrant is called Q0 (0 th), the upper right one is called Q1(1 st), the lower left one is called Q2(2nd), and the lower right one is called Q3(3 rd). Now if we again divide a quadrant (say Qi) into four sub-quadrants then we will call them Qi.0, Qi.1, Qi.2, Qi.3 respectively. So Q1.0.3 means the 3 rd sub quadrant of the 0 th sub-quadrant which is the 1 st quadrant of the P-tree. Thus a quadrant can be of any depth. The notation of a quadrant such as Q1.0.3 is called the quadrant id or Q-ID. Fig. 6: shows the Q-ID of a P-tree.

Constructing the P-tree: At first we clipped and thinned the image to the appropriate size. The image is then converted into black and white image of lines and curves. The whole image is then converted into one bit one band spatial data in raster order so that the whole image can be represented in one P-tree for further processing. First we divide the processed image into equal number of rows and columns. The number of rows and column must multiple of four so that we can construct P-tree with fan-out four. The higher those numbers the better the resolution would be. After that we examine individual pixels whether any line of signature has passed through that pixel or not. If one passing line is found then that pixel is given a value 1 and else its value remains 0. After completing this task we are left with nxn matrix of bit patterns from which we will construct our P-tree.

Following Fig. 8 shows the P-tree constructed from the signature shown in Fig. 7.

Matching of signatures: After converting the image into P-tree the matching of two signatures is nothing but matching of two P-trees. For matching we will begin form the top level, first compare the root counts of two P-trees. If they match we will go one level down and match four quadrant of the root node. At this we will compare the quadrant ID Q0 , Q1 , Q2 , Q3. If these quadrant match for both P-trees then we will further go one level down to the Q0.0, Q0.1,Q0.2, Q0.3 and then Q1.0, Q1.1,Q1.2, Q1.3 and so forth. If the root counts match all the way to the leaf node we will decide that the two signatures match. But if at any of the above places if two root counts do not matches, then the two signatures are different.

In order to reduce searching time we will not go through the leaf nodes of the tree if certain criterion does not match. And the criterion is that we can define a threshold value. In the case of matching, if two-root counts differ more than the threshold value we will say that they do not match and discontinue our search. But if

Table 1: Empirical result

No. of Persons	No of patterns for each persons	Rate of correct classification (%)	Rate of false acceptance (%)	Rate of rejection (%)
5	10	95	2	3
8	9	93	3	4
12	5	88	3	9
16	5	86	4	10

the difference is less than the value we will continue our search. Thus we can reduce the search space in great deal.

EXPERIMENT RESULTS

For signature verification system, signatures are collected from different persons. For each person, five signatures are collected. Sixteen persons participated in the experiment. The system was trained using five original signatures and two forgeries generated for each person. Original signatures served as positive examples and forgeries and other signatures in the training data are the negative examples. The number of persons taking part in the experiment has been varied to observe the verification accuracy at various levels. Again the number of patterns and forgeries are also varied. The empirical result is shown in the Table 1.

CONCLUSION

As, signature of same person is varied highly for various factor like emotion, pen pressure, environmental factors etc., it is very difficult to verify signature quite accurately. Though it is problematic, if proper feature of the signature through various preprocessing technique is identified, then the signature may be recognized more accurately. Due to the tedious method of preprocessing the input and time constant, only a (16X16) sample image is used in the training and recognition purpose. To improve the recognition ability, the size of the input sample image may be increased or more feature like area of black pixel, maximum horizontal projection etc. may be taken into account for recognition.

REFERENCES

1. Plamondon, R., 1998. The Design of an On-Line Signature erification System: From Theory to Practice, Intl. J. Pattern Recognition and Artificial Intelligence, 8: 795-811.
2. Sabourin, R., G. Genest and F.J. Preteux, 1997. Off-Line Signature Verification by Local Granulometric Size Distributions, IEEE Transactions on Pattern Analysis and Machine Intelligence, 19: 976-988.
3. Le, Cun *et al.,* 1989. Bacpropagation Applied to Handwritten Zip Code Recognition, Neural Computation, 1: 541-551.
4. Kashi, R.J., W.L. Hu, Nelson and W. Turin, 1998. A Hidden Markov Model Approach to On-Line Handwritten Signature Verification, In tl. J. Document Analysis and Recognition, 1: 102-109.
5. Martens, R., L. Claesen, 1998. Incorporating Local Consistency Information into the On-Line Signature Verification Process, Int. J. Document Analysis and Recognition, 1: 110-115.
6. Yang, N., B.K. Widjaja and R. Prasad, 1995. Application of Hidden Markov Models for Signature Verification, Pattern Recognition, 2: 161-170.
7. Rigoll, G. and A. Kosmala, 1998. A Systematic Comparison Between On-Line and Off-Line Methods for Signature Verifications with Hidden Markov Models, Int. Conference on Pattern Recognition, Brisbane, Australia, August. 16-20, pp: 1755-1757.
8. Bose, N.K. and P. Liang, 1996. Neural Networks Fundementals With Graphics; Algorithms and Applications, International Edition, 1996.
9. Jackson, T. and R. Beate, 1994. Neural Computing: An Introduction, Institute of physics Publishing, Britol and Philadelphia, 1994.
10. Looney, Backpropagation Training With Multiple Pattern, Neural Networks, Elsevier. North Holland 1997.
11. Mohammad Kabir Hossain2 and William Perrizo, Automatic Fingerprint Identification System using P-tree (ICCIT-2002).
12. Asaduzzaman, A.O.M., Md. Khademul Islam Molla and Ganjer Ali, Printed Bangla Text Recognition using Artificial Neural Network with Heuristic Method (ICCIT-2002).