TechFest 2006

Character Recognition Software

It is based on the following two techniques:

- Position based matching (implemented)
- Shape Transformation Feature Extraction / Recognition (going on)

Position based matching (implemented)

Training Phase:

(a) First read the character drawn as an image (pixel by pixel) and store it in a 2-D pixel array (150 x 150 = pixel dimensions of draw area). Image consists of Black & White pixels.
(b) **Uniform the array**: This is important, because depending on the speed of drawing, the number of pixels registered may vary. At places where you draw fast, fewer pixels are there and vice versa.
(c) **Scaling**: Enclose the character in a rectangle and scale it to a size comparable to characters in library (scale to be used is 10 i.e. each image data will be scaled to a pixel array of 10 x 10). The main use of scaling function is to keep the font size of each written character to be same. This helps in position based matching.
(d) In this way keep an off-line data or library for each character. The data may consist of 20-30 samples for each character.

Main Algorithm:

(a) Follow steps a-c of the training phase.
(b) Compare the two arrays element by element and increase the count whenever there is a match. Do this for all the stored patterns for a character and for all the characters.
(c) Finally take the max of all the counts.
(d) The more the count the more is the percentage match.

Future Improvements:

To increase the efficiency relative angle between the two corresponding pixels can be recorded. A lot of filtering can also be done initially based on some parameters, such as no. of strokes in character, no. of pixels (big characters will have large no. of pixels and vice versa) etc.

Shape Transformation Feature Extraction / Recognition (on going)

References:

[1] N.Liolios, E.Kavallieratou, N.Fakotakis and G. Kokkinakis
"A New Shape Transformation Approach to Handwritten Character Recognition"

[2] T. Kohonen, J. Kangas, J. Laaksonen, K. Torkkola.
"LVQ PAK: A program package for the correct application of Learning Vector Quantization algorithms"

This is a simple algorithm, based on dynamic programming for handwritten character recognition, with improved accuracy. The Shape Transform (ST) approach is based on the calculation of the cost of transforming the image of a given character into that of another, thus taking into account local geometrical similarities and differences.

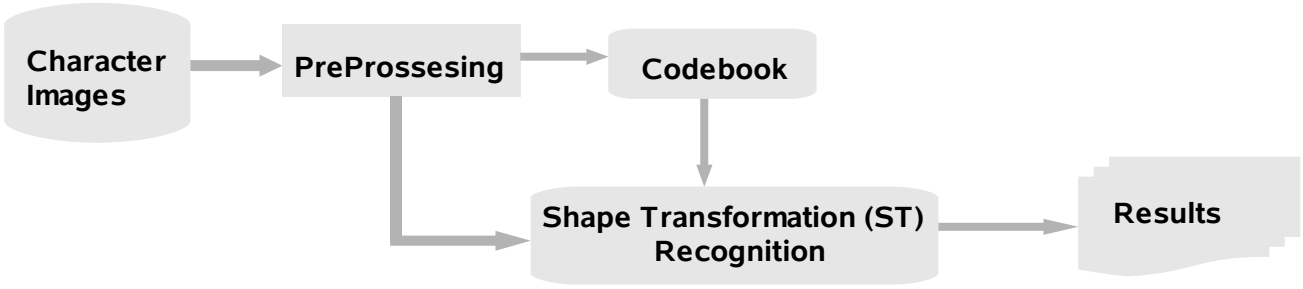

Figure: The shape Transformation Approach

Feature Extraction

After size normalization the image x is a square image of size NxN. The feature extraction process here is as simple dump of the 2D image as a series of 0's and 1's. In the training phase, a label follows the data, while during recognition; the whole image is passed to the recognition stage. The codebook as well as the test ensemble are therefore sets of vectors of length N^2 where each vector x = (x11, x21, …,xN1, x12, … ,xN2, …, xNN) is formed by concatenating the rows of the image. The Learning vector quantization **LVQ [2]** method is used to create the codebook during training.

Shape Transformation Recognition

It is based on the simple idea that the location of each and every pixel is a feature and it cannot be ignored. It resembles, to some extend, the dynamic programming technique, used for lexicographical corrections, as a post-processing step in OCR.

We start from the simple fact that the binary character image x can be transformed into image y by:

a) Leaving some pixels intact (substitution)
b) Moving some pixels of x into locations that correspond to their respective closest pixels in y and
c) Deleting some pixels if necessary.

We attribute no cost to substitutions, as this is the equivalent of leaving them in place because they are common to both images (characters). The cost of moving (translating) a pixel to another location is set equal to the Euclidean distance between the source and destination locations. As each pixel in x is either substituted or translated, we consider the corresponding pixels in x and y associated. Pixels that are associated are not used again.

After all substitutions and translations we may be faced with one of two possible situations:

a) There are pixels remaining in x
b) There are pixels in y, which are not associated.

If there are pixels still remaining in x, we delete them. The cost for deleting a pixel in image x, is the cost of moving the pixel outside (distance from) the character frame, in a straight-line path, following the closest route available.

For each non-associated pixel in y a pixel is added to x. The cost for each such addition is equal to the cost of moving the pixel (distance) from the character frame, to it's corresponding location, in a straight-line path, following the closest route available.

The algorithm below describes in detail the shape transformation, as well as the transformation cost calculation for two character images x and y.

```
Function cxy = ST_Cost(x, y)
 xa = x.y             // intersection of x,y
 xm = x   xa          // not common set of pixels in x
 ym = y   xa          // not common set of pixels in y
 cxy = 0              // Cost of transforming x to y
 dmax = 32².32²    // 32x32 character's size

For each pixel xij in xm
      dmin = dmax
      For each pixel ykl in ym
         d = Euclidean_Distance(xij, ykj)
         If d < dmin then
            dmin = d
         End If
      End For
      cxy = cxy + dmin
End For
Return cxy
```

The ST Recognition algorithm below describes how successive calls to the ST_Cost() function above are made to identify the image in the codebook that the unknown transforms to, with the least transformation cost.

```
Algorithm ST Recognition
//for image x to be recognized
  dmax = 32².32²
  cmin = 1024.dmax
  For each image y in the codebook
     cxy = ST_Cost(x, y)
     If cxy < cmin then
        lx = Label(y)
        cmin = cxy
     End If
  End For
```

After execution of the algorithm above, the label for the image to be recognized is found in lx and it is the codebook label, for which the corresponding image, had the lowest transformation cost cmin.

Prekshu Ajmera
IIT Bombay