Pattern Recognition Letters 7 (1988) 9–12
North-Holland

January 1988

A heuristic noise reduction algorithm applied to handwritten numeric characters

S. RAY

Electronics and Communication Science Unit, Indian Statistical Institute, 203 B.T. Road, Calcutta 700035, India

Received 16 February 1987

Abstract: A new noise reduction algorithm based on window sizes varying from 3 × 3 to 5 × 5 is presented. The algorithm was applied to reduce black ('1'-valued) noise specks from 1000 isolated handwritten numeric characters.

Key words: Noise reduction, character matrix, digitization, binarization, grey-tone image, character recognition.

1. Introduction

Binarization error and spattering of ink sometimes lead to the presence of isolated blocks of pixels of value '1' (black pixels) in the background, or contain '0'-pixels which should ideally be '1', in the representation of a character matrix. Various smoothing techniques have been used in the past in order to reduce the number of noisy pixels. In most of these techniques a 3 × 3 window has been considered and some logical (or averaging) rules applied to the pixel appearing in the middle of the window (e.g., Unger, 1958, 1959; Ullman, 1973). Sometimes a 3 × 3 window is not good enough to eliminate the noise. This is particularly true when higher resolution is used in the process of digitization. In this paper an algorithm, based on windows of different sizes varying from 3 × 3 to 5 × 5, has been developed for the removal of black ('1'-valued) noise specks from a character matrix. Consideration of higher order windows would enable the elimination of bigger noise specks. Moreover, smaller noise specks are detected by lower order windows, and in this case there is no need to go for checking with higher order windows. In practice smaller noise specks are more in number. Therefore the algorithm is expected to be computationally more efficient than the algorithm which considers a fixed window size of order 5 × 5. The algorithm was applied to reduce noise from 1000 isolated handprinted numeric characters, each represented by a matrix of order 60 pixels × 60 pixels.

2. The algorithm

Let $A = (a_{ij})$ denote the character matrix under consideration. In the present case, A is of order 60 × 60. To remove the noise specks from A the following steps are followed:

Step 1. Augment the matrix by adding three rows in the end, two columns in the beginning and two columns in the end such that all the elements of these added rows and columns are 0's. Thus $A = (a_{ij})$ is now a matrix of order 63 × 64 with $i = 1,2,\ldots,63$ and $j = -1,0,1,\ldots,62$.

Step 2. Set all the elements of the first row to 0, i.e., $a_{ij} = 0, j = -1,0,1,\ldots,62$.

Step 3. For each $i = 2,3,\ldots,60$ and $j = 1,2,\ldots,60$ check whether $a_{ij} = 0$ or 1. If $a_{ij} = 0$ then go to next element. If $a_{ij} = 1$ then check conditions (1) to (14) given in Figure 1. If any of these conditions is true

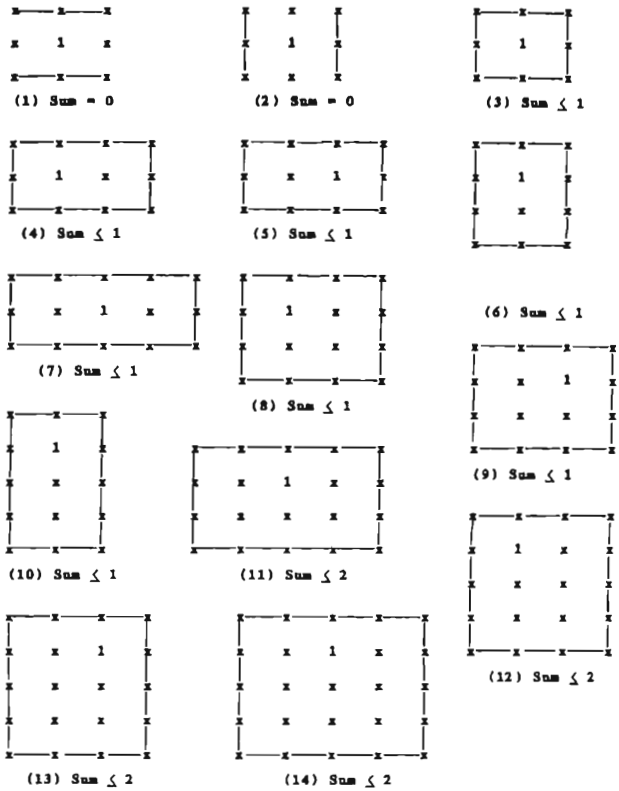

Figure 1. Noise conditions for a black ('1') element (the term 'Sum' stands for the sum of connected elements).

then make $a_{ij} = 0$, otherwise no operation. Go to the next element.

The matrix augmentation made in Step 1 is required for checking the conditions mentioned in Step 3. In Step 2 all the elements of the first row are made equal to 0. This is based on the assumption that while writing a character the author usually leaves some blank space on the top. In the case of doubt about the validity of this assumption it can be easily avoided by adding a '0'-row in the beginning of the matrix A. In Step 3 the 14 conditions regarding the sum of boundary elements in windows of different sixes are checked. Figure 1 is more or less self-explanatory. The '1' inside a window represents the element under consideration. The boundary elements denoted by 'x' and connected by straight line segments are summed up to decide whether '1' forms a part of the character or it is a noise. Elements marked with 'x' in Figure 1 could be either '0' or '1'. In condition (1) it is checked if all the six elements in the previous and the follow-

ing rows are '0'. In condition (2) similar check is made for columns. In conditions (3) to (10) it is checked if at the most one of the surrounding 'x'-elements is '1'. In conditions (11) to (14) it is checked if the sum of the surrounding 'x'-elements is less than or equal to 2. If any of the above fourteen conditions holds then it is highly unlikely that the element under consideration constitutes a part of the character. In this case, therefore, it is replaced by '0'.

3. Experimental results and conclusions

Data set

The data set consisted of 1000 isolated handprinted numerals written by 10 members of the Department of Electrical Engineering, Imperial College of Science and Technology, London. Each member wrote 10 repetitions of each of the 10 numerals 0, 1, 2, ..., 9. The writing was done on transparent sheets using a black inked felt pen. The only restriction imposed on writing was to put each character in a square of size 12 mm × 12 mm. The data were digitized by using a scandig 3 scanner controlled by a nova 3 computer. For digitization the scan increment option of 200 microns and density resolution of 1 part in 256 were used. This means that each numeral was represented by a matrix of 60 pixels × 60 pixels, each of the pixels assuming a value in the grey level range of 0 to 255 (Ray, 1984).

Unlike some image processing problems, for the purpose of character recognition it was not necessary to have the detailed grey level differences of the pixels. Binary representation was thought to be ideal. Multi-level representations (grey-tone images) were, therefore, converted into binary representations (two-tone images) of the numerals. Thus, each of the numerals was represented by a matrix of order 60 pixels × 60 pixels, each pixel assuming either a value '0' (background pixel) or a value '1' (pixel on the locus of the character).

Removal of noise

From a visual inspection of the character matrices it was seen that there were a lot of '1'-valued pixels in the background which should ideally be '0'. Numerals '3' and '4' shown in Figure 2 illustrate this.

The representations of the numerals after the application of the noise reduction algorithm described above are given in Figure 3.

Concluding remarks

As expected, the algorithm successfully removed black noise specks from all the 1000 characters. This, together with the fact that the algorithm is computationally efficient, suggests the applicability of the algorithm.

Acknowledgement

The author gratefully acknowledges the financial support given by the Commonwealth Commission to carry out the research, the study leave granted by the Indian Statistical Institute to stay abroad and the research facilities offered by the Imperial College of Science and Technology, London. Typing of the manuscript by Sri. J. Gupta is also gratefully acknowledged.

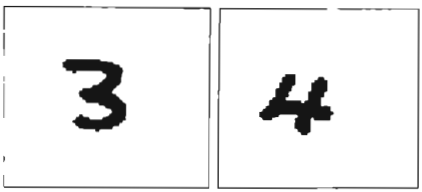

Figure 2. Binary representation of numerals '3' and '4' before noise reduction.

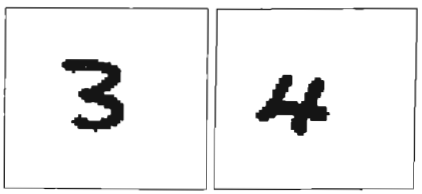

Figure 3. Binary representation of numerals '3' and '4' after noise reduction.

References

[1] Unger, S.H. (1958). A computer oriented toward spatial problems. *Proc. IRE*, 46, 1744–1750.

[2] Unger, S.H. (1959). Pattern detection and recognition. *Proc. IRE*, 47, 1737–1752.

[3] Ullman, J.R. (1973). *Pattern Recognition Techniques*. Butterworths, London.

[4] Ray, S. (1984). The effectiveness of features in pattern recognition. Ph. D. Thesis, University of London.