Pattern Recognition Letters 11 (1990) 77–80

An efficient algorithm for running window pel gray level ranking in 2-D images

B.B. CHAUDHURI

Electronics & Communication Sciences Unit, Indian Statistical Institute, 203 Barrackore Trunk Road, Calcutta 700035, India

Abstract: An efficient algorithm is proposed for computing the rank orders of pel gray levels over running windows in a 2-D image array. For a $n \times n = M$-pels window, it is shown that the worst case computer complexity is $O(M)$ and it is independent of the image business. Practical results show a factor of 1.5 to 2.5 improvement over the worst case and this algorithm compares favourably with other proposed algorithms. The algorithm may be used for min, max or median filtering as well as for image transformations involving rank orders.

Key words: Fast algorithm, max/min filtering, median filtering, rank-order filtering, gray level thinning.

1. Introduction

Consider a two-dimensional (2-D) array of integers representing the gray levels of pels of an image. For each pel position, called the candidate pel position, consider a sub-array or window of $n \times n$ pels around it. We address here the computational aspect of sorting the gray levels of each window in decreasing or increasing order of magnitudes. If the candidate pels are encountered in a raster-scan mode, the corresponding window may be assumed to slide or run horizontally with the candidates. The term running window is used to denote this process.

The window gray level sorting can be used for image smoothing where for each candidate pel, the median, min or max of the gray levels in the window is found and the candidate pel gray level is replaced by the found value. The process may be termed as median, min or max filtering. More generally, one may think of a k-th order filtering where the gray level of the candidate pel is replaced by the k-th order gray level over the window where $1 \leqslant k \leqslant n^2$. Based on the local constraint on the image, k may vary from candidate to candidate. In such a situation it may be wise to have an efficient algorithm that returns complete ordering at each candidate pel rather than computing median, max or min alone.

Let $n^2 = M$. Sorting by an algorithm like quicksort has an expected $O(M \log M)$ complexity while its worst case complexity is $O(M^2)$. While computing complexity, we consider only the number of comparisons required per window. For a digital image gray level array, there exist two types of redundancy that many be exploited for discovering efficient algorithms in this problem. One type of redundancy is the correlation of neighboring pel gray levels. If the neighboring pel gray levels are well correlated, then it is expected that an order, say the median, for a candidate pel will not differ markedly from that of the neighboring candidate pel. Using this idea Huang et al. (1979) proposed an algorithm with expected complexity $2n + 1.5 + d$ where d varies from picture to picture but usually does not exceed a value of 10. However, this algorithm is suitable to compute only one of the ranks, say the median. The other type of redundancy is the spatial arrangement where part of the computation used for sorting around a candidate pel can also be used for sorting at the neighboring candidate pel. An al-

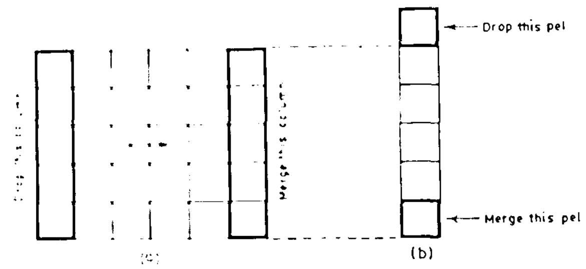

Figure 1. Basic approach of ranking.

gorithm using this concept is reported by Chaudhuri (1983) with results comparable to those of Huang et al. (1979).

In this study, an efficient algorithm is proposed to obtain the complete ranking using the spatial redundancy. The complexity of this algorithm is $O(M)$ and it is not restricted to image data.

2. Algorithm and its complexity

The basic idea of the algorithm is very simple. Let the candidate pels be accessed in a horizontal scanning fashion. When moving from one candidate to the next, the window for the current candidate can be obtained by deleting the leftmost column of the previous window and appending a leading column to the right. While ordering in the current window as well, left-most column pel ranks are dropped from the ranked gray levels of the previous window. In the remaining gray levels, which still remain ordered, the leading column pels are inserted using the principle of mergesort.

The idea is explained by Figure 1(a). It is easy to show that no comparison is required to delete a column of pel orders from the ordered values over a window. Let P denote the order of pel (x,y) in the window. To drop (x,y) out, all ranks greater than P are reduced by 1 and assigned to the corresponding pels.

When n pels are dropped, there are n^2-n pels remaining in the window. Suppose that the pels within the leading column are already ranked among themselves. To merge the n pels of the leading column with the remaining n^2-n pels of the window, n^2-n comparisons are needed at most.

Now, consider the sorting of n pels of the leading column. Suppose that the previous row of candidate pels is already processed and the sorting of all $n\times 1$ columns for those candidate pels are stored in the memory. Of these, the column sorting corresponding to the current leading column is recalled, the top pel rank is dropped and the bottom pel is merged, thus completing the sorting of the leading column for the current candidate. This idea is explained by Figure 1(b). No comparison is required to drop the top pel. To merge the bottom pel in the $n-1$ ranked pels, at most $n-1$ comparisons are needed. The total number of comparisons needed is, therefore,

$$(n^2-n)+(n-1)=n^2-1=M-1$$

which is a $O(M)$ worst case algorithm.

No other reported work on fast algorithms for complete ranking over a window is known to the present author and hence a direct comparative study cannot be made. However, one can use the algorithm due to Huang et al. (1979) to compute all n^2 ranks. By direct analogy, this algorithm is expected to require $\sum_{i=1}^{n^2}(2n+1.5+d_i)$ comparisons where d_i represents the value of d for the i-th rank computation. Then our algorithm will, at least, have an advantage by a factor of

$$\frac{\sum_{i=1}^{n^2}(2n+1.5+d_i)}{n^2-1}=\frac{n^2\,(2n+1.5+\bar{d})}{n^2-1}$$

$$>2n+1.5+\bar{d}$$

where

$$\bar{d}=\frac{1}{n^2}\sum_{i=1}^{n^2}d_i.$$

3. Experimental results and discussion

This algorithm was tested on a highly textured Landsat image and 4 image data (see Figure 2) given in the book by Ganzalez and Wintz (1987). The results for working on different window sizes is presented in Table 1. It is seen that the average number of comparisons required is much less than

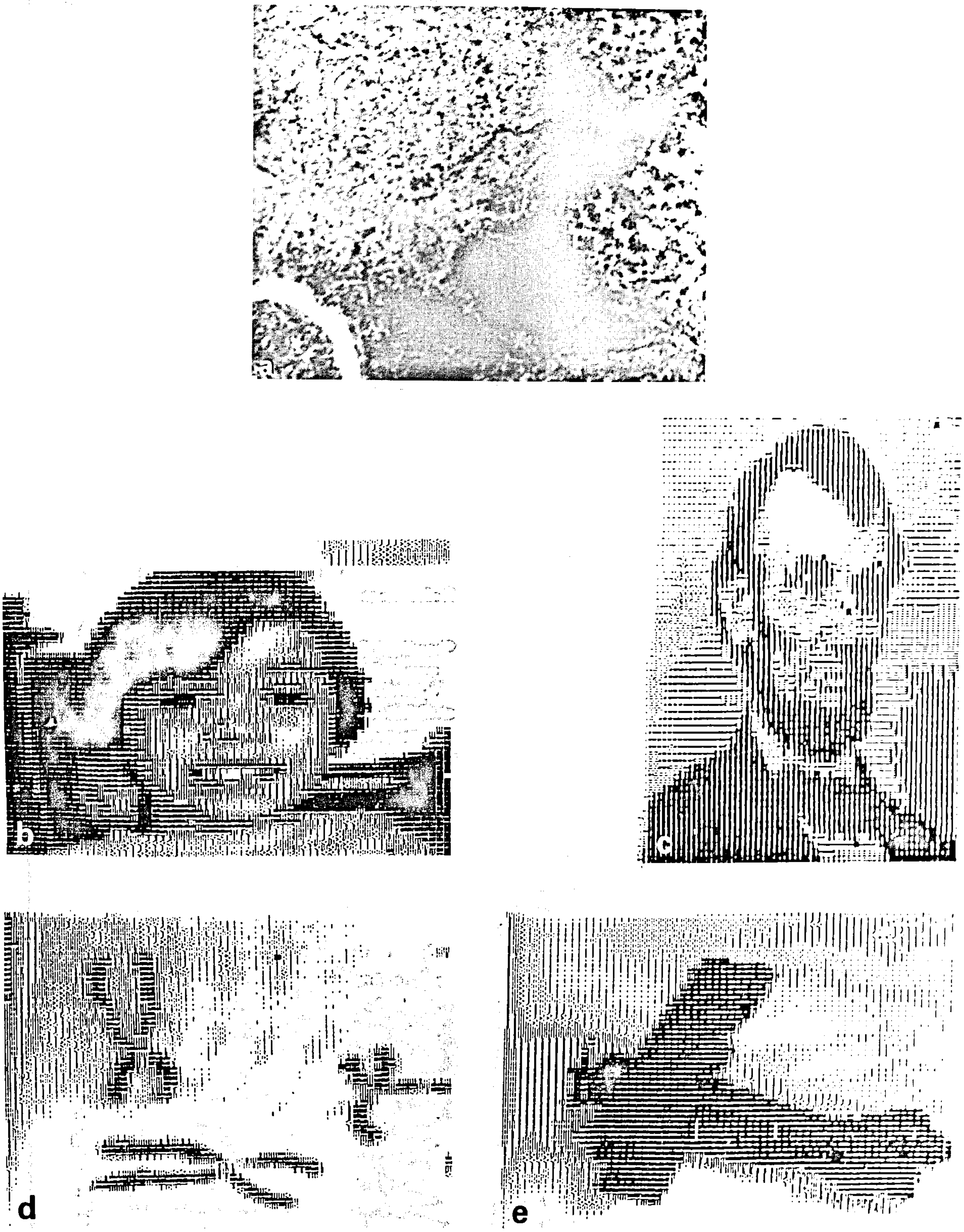

Figure 2. Test images: (a) Texture, (b) Boy, (c) Lincoln, (d) Chromosomes, (e) Biplane.

Table 1
Average number of comparisons actually required

Image name	Window size			
	3 × 3	5 × 5	7 × 7	9 × 9
	No. of comparisons in worst case			
	8	24	48	80
Texture	5.21	17.75	38.05	67.69
Boy	3.45	12.87	29.55	52.58
Lincoln	4.03	14.41	31.56	55.20
Biplane	3.57	13.07	32.27	57.15
Chromosomes	2.96	11.08	25.92	48.74

the worst case and an improvement by a factor of 1.5 to 2.5 is encountered. The performance of *Texture* image is nearest to the worst case because it is the busiest image. *Chromosomes* on the other hand, is the smoothest image and it needs the minimum number of comparisons for windows of any size.

If this algorithm is used to find only one rank, say the median, then for $n = 3, 5$ its performance is not worse than, say, the algorithm due to Huang et al. (1979) that requires nearly $2n + 10$ comparisons.

An attempt was made to improve the algorithm by exploiting the correlation of neighbouring pel gray values. It was expected that the neighboring pels would be nearer to each other in ranks. A rank merging algorithm which started with comparing the neighboring pels was tested on the image data of Figure 2 but no convincing improvement was obtained for the window sizes considered. This result indicates that the ranks of the neighboring pels are not so well correlated that further improvement in computation is possible.

Other applications of window ranking may include thinning and skeletonization of gray-tone images. The topic is being studied and useful results will be reported in a future correspondence. Also, extension of the algorithm for different parallel architecture systems is being investigated.

Acknowledgement

Encouragement in the work by Prof. D. Dutta Majumder and help rendered by Mr. V.S. Prakash, Mr. J. Gupta and Mr. S. Chakraborti in preparing the manuscript are acknowledged with thanks.

References

Chaudhuri, B.B. (1983). A note on fast algorithms for spatial domain techniques in image processing. *IEEE Trans. Syst. Man Cybernet.* 13, 1166–1169.

Gonzalez, R.C. and P. Wintz (1987). *Digital Image Processing.* Addison-Wesley, Tokyo, 2nd ed.

Huang, T.S., G.T. Yang and G.Y. Tang (1979). *IEEE Trans. Acoust. Speech Signal Process.* 27, 13–18.