IMAGE ENHANCEMENT FILTERS BASED ON THRESHOLD PATTERNS

Pak-Cheung Lai and Bing Zeng

Department of Electrical and Electronic Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon
Hong Kong Special Administrative Region, China

ABSTRACT

Threshold Boolean filters (TBFs) constitute a large class of non-linear filters that are very effective in removing impulsive noise. However, the required computational complexity is very high because thresholding has to be done at all levels. In this paper, a new type of filters based on threshold operation is proposed. The proposed filters, called threshold pattern filters (TPFs), try to pick up a sample within the window as the output such that the objective function, which is chosen based on the threshold patterns, is minimized. An efficient MAE-based training method for designing TPFs is proposed. Some designed TPFs are applied to enhanced images corrupted by mixed noise and the results are very comparable to that of the corresponding TBFs.

1. INTRODUCTION

Threshold Boolean filters (TBFs) [1] are large class of non-linear filters which are effective in removing impulsive noise. They include the well-known stack filters and median filter. However, one drawback of TBFs is their complexity. To obtain the filter's output, thresholding has to be done at all levels. The binary vector at each level is then filtered by a Boolean function. The outputs of the Boolean functions at all levels are then added up to form the filter's output.

Using the fact that there are only at most $N+1$ distinct binary vectors in a given window of size N, more efficient method to find out filter's output based on sorting are developed [1]. However, the complexity is still high.

In this paper, a new type of filters based on threshold operation called threshold pattern filters (TPFs) is proposed. This type of filters aims at removing impulsive noise in a relatively simple filter structure. To avoid high complexity of thresholding at all levels, a TPF only thresholds the window at the levels equal to sample values in the window. We call the binary vectors obtained "threshold patterns". One of the samples in the window is selected as the filter's output. The selection is based on some criteria of the threshold patterns obtained in the window.

In the next section, the detail structure of the proposed filter is introduced. In Section 3, relation between the proposed filters and the rank-order filters (ROFs) is discussed. An efficient design method for TPFs is developed in Section 4. In Section 5, TPFs are applied to enhance images corrupted by mixed noise. Their performances are compared with those of TBFs and other commonly used techniques. Some concluding remarks are given in Section 6.

2. STRUCTURE OF TPFS

Suppose that X can only take values $0, 1, \cdots, M$, i.e. $(M+1)$-valued. Thresholding X at level m, an integer between 1 and M inclusively, is defined as :

$$I_m(X) = \begin{cases} 1 \ , & \text{if } X \geq m \\ 0 \ , & \text{otherwise} \end{cases} \tag{1}$$

For a vector $\underline{X} = [X_1, \cdots, X_N]$, $I_m(\underline{X}) = [I_m(X_1), \cdots, I_m(X_N)]$ becomes a binary vector of length N. Each of the 2^N distinct vectors is called a *threshold pattern*. We denote a threshold pattern by $\underline{v}_i$, where $\underline{v}_i \in \mathcal{B}^N$ is the binary representation of the decimal number i, $i = 1, 2, \cdots, 2^N - 1$.

Let $f(\cdot)$ be a binary-vector-input real-value-output function, i.e. :

$$f : \mathcal{B}^N \to \mathcal{R} \ . \tag{2}$$

Then the threshold pattern filter (TPF) operation $TPF_f(\cdot)$, where function f is the filter parameter, is defined as :

$$TPF_f(\underline{X}) = \arg\left[\min_{X_1, \cdots, X_N} \left\{ f(I_{X_r}(\underline{X})) \right\}\right] \tag{3}$$

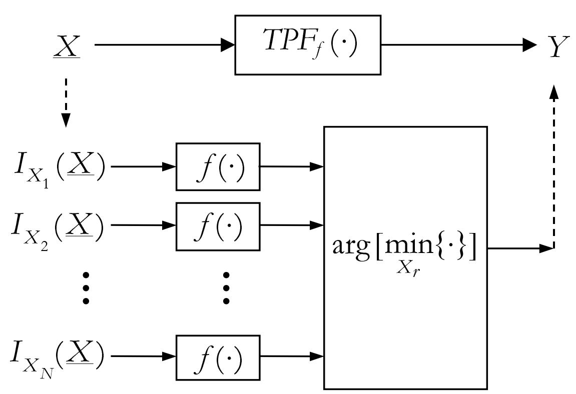

Figure 1: Structure of TPFs

This filter structure, as shown in Figure 1, has an intuitive interpretation — if there is an objective function $f(\cdot)$ which depends on the threshold patterns, the filter tries to pick a sample from the window as output such that the objective function is minimized among the samples in the window.

If $f(\underline{v}_i)$ is chosen to be the mean absolute error (MAE) of using X_r , where $I_{X_r}(\underline{X}) = \underline{v}_i$, as an estimate of desired signal s, then the filter will choose a sample from window as output such that the MAE is minimized.

It can be seen that the complexity of TPFs is much lower than that of TBFs. For TPFs, thresholding is needed only at N levels that are equal to sample values in the window, instead of all levels from 1 to M as in TBFs. Even compared with the efficient implementation of TBFs in which only N thresholdings are needed [1], TPFs save the computations for sorting, subtractions and additions needed in TBFs with N compare-and-select operations.

3. RELATION WITH RANK-ORDER FILTERS

Threshold patterns provide useful information about the ranks of the samples within the window. For the threshold pattern at level X_r, 0's in the pattern indicate the samples which are smaller than X_r and the 1's indicate the of samples which are greater than or equal to X_r. If the samples in the window are all distinct, the number of ones is in fact the rank of X_r in decending order.

Based on the above fact, it can be shown that a TPF is a supper-set of rank-order filters (ROFs) [2]. Suppose that a ROF chooses the r-th largest sample as output. By setting $f(\underline{v}_{(r)}) < f(\underline{v}_{(r+1)}) < \cdots < f(\underline{v}_{(N)}) < f(\underline{v}_{(m)})$, where $\underline{v}_{(r)}$ is the threshold pattern with r ones and $m < r$, it can be sure that the TPF always select the r-th largest sample as output.

To verify this, let's first assume that the samples in the window are all distinct. Then the threshold pattern at sample value whose rank is r has exactly r ones. By above parameter ordering, this sample is chosen as output. Obviously, the arguement also holds when some of the samples which rank higher or lower than r are not distinct.

The remaining scenario is that the sample that rank r is not distinct, but appears several times in the window. Then, threshold pattern at that level must have r or more ones. Suppose that the threshold pattern at this sample value has k ones, where $k > r$, then, it is impossible to have threshold patterns with number of ones between r and $k-1$ for that window. Therefore, $f(\underline{v}_k)$ is the minimimun value among all the threshold patterns in the window and the filter's output has the same value as the r-th ranked sample. However, note that the reverse does not hold, i.e. if there are k ones in the threshold pattern at level whose rank is r, where $k \geq r$, it is possible to have $k+1, k+2, \cdots, N$ ones in the threshold patterns for that window. But they are produced by thresholding the window at values smaller than the k-th rank sample, and these values should not be chosen as output. It is therefore necessary to set $f(\underline{v}_{(r)}) < f(\underline{v}_{(r+1)}) < \cdots < f(\underline{v}_{(N)}) < f(\underline{v}_{(m)})$, where $m < r$, such that a TPF becomes a r-th rank ROF.

4. AN EFFICIENT DESIGN METHOD

The optimal MAE design of TPFs under training framework can be stated as : Given an observed signal $X(n)$ and a desired signal $s(n)$, find a mapping $f : \mathcal{B}^N \rightarrow \mathcal{R}$, such that the expected absolute error between the desired signal and the signal produced by filtering the observed signal is minimized. Mathematically, it can be formulated as :

$$\underset{f}{minimize} \quad J(f) = E\Big\{ \left| TPF_f(\underline{X}) - s \right| \Big\} \quad (4)$$

where $\underline{X}$ is the input window.

However, the minimization above is very complicated and no practical algorithm has been found to solve the problem at this time. Instead, we adopt an *ad-hoc* approach in designing the filters. As described in Section 2, if the function $f(\underline{v}_i)$ represents the MAE of selecting X_r, which gives rise to the threshold pattern $\underline{v}_i$, then, the filters try to minimize the MAE. With this interpretation, the filter design now changes to a problem of estimating the MAE associated with the threshold pattern $\underline{v}_i$.

Let $\underline{X}(n)$ be the input window at time n. Let $\varepsilon(\underline{v}_i)$ be the accumulative absolute error for the pattern $\underline{v}_i$ and $\rho(\underline{v}_i)$ be the occurrence of the pattern $\underline{v}_i$. Define $\mathcal{M}(\underline{v}_i) = \{(n,r) : I_{X_r}(\underline{X}(n)) = \underline{v}_i\}$. Then,

$$\varepsilon(\underline{v}_i) = \sum_{(n,r)\in\mathcal{M}(\underline{v}_i)} \mid X_r(n) - s(n) \mid \tag{5}$$

for $i = 1, 2, \cdots, 2^N - 1$, and

$$\rho(\underline{v}_i) = Card\{\mathcal{M}(\underline{v}_i)\} \tag{6}$$

for $i = 1, 2, \cdots, 2^N - 1$.

The MAE associated with the threshold pattern $\underline{v}_i$, i.e. $f(\underline{v}_i)$, is estimated by :

$$f(\underline{v}_i) = \frac{\varepsilon(\underline{v}_i)}{\rho(\underline{v}_i)} \tag{7}$$

for $i = 1, 2, \cdots, 2^N - 1$.

With this mapping, a TPF is then defined. Note that the filters obtained by this method are not necessarily optimal. But we will show by simulations that this design method yields promising results in practical cases in the next section.

The complexity of designing TPFs as proposed above is lower than that of TBFs under the sum of micro MAE (SMMAE) criterion [1]. Again, the reason is that the number of thresholding needed is much reduced.

5. SIMULATIONS

5.1. Enhancing a Noisy Edge

In this part of simulation, we try to find out the filter's response to a noisy edge. An ideal 1-D edge of height 100 is shown in Figure 2(a). It is corrupted by additive mixed noise having the following distribution :

$$\eta \sim (1-\lambda)\,\mathcal{N}(0,\sigma) + \lambda\,\mathcal{N}(0, \frac{\sigma}{\lambda}) \tag{8}$$

where $\sigma = 10$, $\lambda = 0.1$.

The noisy signal, as shown in Figure 2(b), and the clear signal are used to train TPFs with window size $N = 5$ by method proposed in Section 4. TBFs are also trained, under the sum of micro MAE criterion, using the same set of signals.

The resulting filters are then applied to filter the noisy signal. Their outputs are shown in Figure 2(c) and 2(d) respectively. Broken lines in the figures indicate the shape of the ideal edge. For comparison, the results for median filter and averaging filter are also included in Figure 2. The MAEs of the filtered signals deviated from the ideal edge are listed in Table 1.

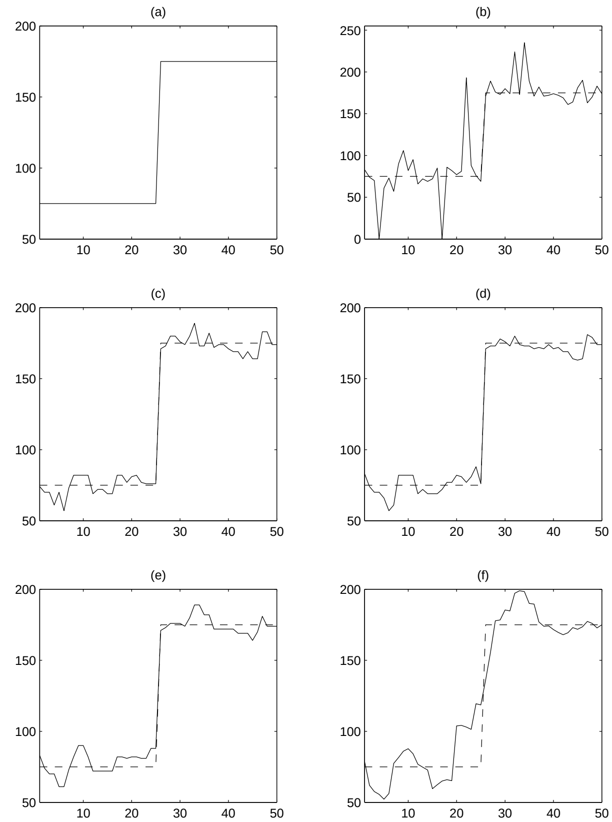

Figure 2: (a) An ideal edge; (b) the corrupted edge; (c) filtered by TPF; (d) filtered by TBF; (e) filtered by median filter; (f) filtered by averaging filter.

Table 1: MAEs for noisy and filtered signals

Signal	MAE
Noisy	14.36
TPF	5.24
TBF	5.16
Median	5.90
Averaging	12.39

It can be observed from Figure 2 that TPF, TBF and median filter can effectively suppress the noise and preserve the edge, but averaging filter fails to achieve these. In terms of noise suppression in MAE sense, TBF performs the best. TPF is close to TBF and significant better than the other two filters.

Table 2: MAEs for noisy and enhanced images (Lena)

Images	MAE (Training region)	MAE (Whole image)
Noisy	14.0345	13.8716
TPF	4.6703	4.8588
TBF	4.6541	4.8340
Median	-	4.8431
Averaging	-	7.4885

Table 3: MAEs for noisy and enhanced images (Harbor)

Images	MAE (Training region)	MAE (Whole image)
Noisy	14.0241	13.9255
TPF	6.8553	7.8670
TBF	6.4705	7.2545
Median	-	9.0996
Averaging	-	11.8483

5.2. Enhancing Noisy Images

Enhancing images corrupted by mixed noise using TPFs are simulated in this sub-section. The first image is "Lena" (512×512) and the corrupted image is produced by adding noise given by (8).

The top left quarters of the noise-free "Lena" and noisy "Lena" are used to estimate the parameters of TPFs under the MAE criterion. After estimation, the resulting filter is applied to the whole noisy image. To evaluate TPF's performance, the noisy image is also enhanced by TBF, median and averaging filter repectively. The windows used in the simulations are all 3×3 square windows. Table 2 shows the MAEs of the noisy and enhanced images.

The above experiment is repeated with another test image "Harbor" (512×512) which contains many fine structures. The original, noisy and enhanced images are shown in Figure 3 and the resulting MAEs are listed in Table 3.

As shown in the tables, the performance of TPF is almost the same as that of TBF and median filter, and much better than averaging filter for the image "Lena". For "Harbor", although TPF is not as effective as TBF, it is still considerably better and much better than median filter and averaging filter respectively.

The visual quality of images enhanced by TPFs is almost identical to those enhanced by TBFs. Median filter performs well for the image "Lena" but relatively poorly for the image "Harbor". Some fine structures in "Harbor" are blurred by median filtering. Averaging filter causes blurring in both cases and the images still look noisy.

In light of the small differences in their MAEs and visual quality, it can be concluded that the noise suppression capability and detail preserving capability of TPFs are comparable to those of TBFs.

6. CONCLUSIONS

In this paper, a new type of filters called threshold pattern filters (TPFs) is proposed. TPFs only utilize the threshold patterns at the levels that are equal to sample values in the window. The complexity is hence much reduced as compared with TBFs. TPFs select a sample in the window such that the objective function is minimized.

Simulations of using TPFs to enhance mixed-noise corrupted images are carried out and compared with TBFs, median filter and averaging filter. It is found that TPFs are effective in removing impulse noise and have comparable performance as TBFs, but at lower complexity.

There are some issues about TPFs which remain for further studies. For example, the design method proposed in this paper is efficient but not necessarily optimal. Thus, research is still going on to optimize TPFs. Analysis of the filter behaviour is another research area. One way to analyze this type of non-linear filter is through breakdown probability, rank-selection and sample-selection probability [3].

7. REFERENCES

[1] K. D. Lee and Y. H. Lee, "Threshold Boolean Filters", *IEEE Trans. on Signal Processing*, vol. 42, pp.2202-2036, Aug. 1994.

[2] L. Yin, R. Yang, M. Gabbouj and Y. Neuvo, "Weighted Median Filters: A Tutorial", *IEEE Trans. on Circuits and Systems*, vol. 43, pp.157-192, March 1996.

[3] M. K. Prasad and Y. H. Lee, "Stack Filters and Selection Probabilities", *IEEE Trans. on Signal Processing*, vol. 42, no. 10, pp.2628-2643, Oct. 1994.

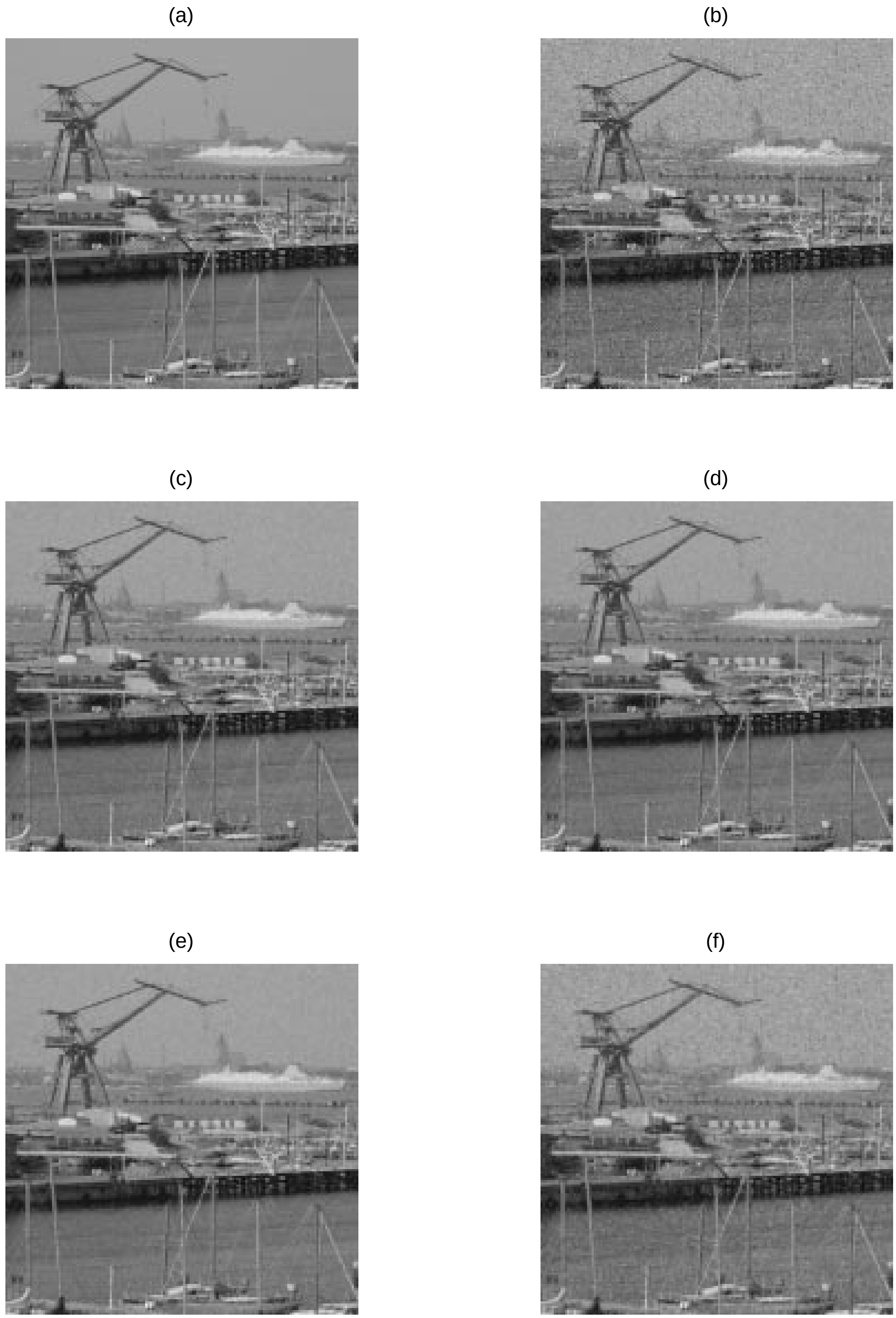

Figure 3: (a) Original image; (b) noisy image; (c) enhanced by TPF; (d) enhanced by TBF; (e) enhanced by median filter; (f) enhanced by averaging filter.