AN EFFICIENT METHOD FOR THE REMOVAL OF IMPULSE NOISE FROM SPEECH AND AUDIO SIGNALS

Charu Chandra, Michael S. Moore and Sanjit K. Mitra

Dept. of Electrical and Computer Engineering
University of California, Santa Barbara
Santa Barbara, CA 93106-9560

ABSTRACT

A computationally efficient algorithm is proposed to remove noise impulses from speech and audio signals while retaining its features and tonal quality. The proposed method is based on the SD-ROM (Signal Dependent Rank Order Mean) algorithm. This technique has successfully been used to remove impulse noise from images. It has the advantage of being relatively fast, simple and robust. The algorithm estimates the likelihood the sample under inspection is corrupt relative to the neighboring samples and replaces a sample detected as corrupted by a value based on the neighboring samples. This algorithm also has the advantage of being 'configurable' to the type of noise impulses in the sample, as the thresholds used to detect noise impulses can be varied to suit the signal.

1. INTRODUCTION

Standard global filtering techniques like lowpass filtering do not differentiate between impulse corrupted samples and uncorrupted samples. Median filters and other order static filters that operate on a localized area typically modify uncorrupted samples as the filtering is applied uniformly over the whole signal. The median filter is a highly robust estimator of the signal value in the presence of impulse noise. However, a median filter will eliminate changes in the input signal with a duration less than half the size of the filter window. When the signal is heavily corrupted with impulse noise, a large median window is needed and this leads to more uncorrupted samples being replaced by the median value within the window and more high frequency components being removed by the filter. As a result, conventional techniques for impulse noise removal perform poorly for an acceptable level of feature and tonal quality.

This work was supported in part by a University of California MICRO grant with matching support from Envision Corporation and Xerox Corporation.

The ideal objective would be to replace only the noisy samples, leaving uncorrupted samples unchanged. The technique presented in this paper attempts to do this using a detection-estimation approach. Each impulse is first detected in the sample stream and then is replaced by an estimate based on neighboring samples.

We assume that the noise impulses can take any arbitrary value within the dynamic range with some error probability. Let $\mathbf{s}(n)$ denote the uncorrupted 1-D sequence and $\mathbf{x}(n)$ denote its corrupted version containing some impulse corrupted samples. Then, at time instant n, for an impulse noise model with an error probability p_e, we have

$$x(n) = \begin{cases} s(n), & \text{with probability } 1 - p_e \\ \eta(n), & \text{with probability } p_e \end{cases} \tag{1}$$

where $\eta(n)$ is an identically distributed, independent random process with an arbitrary underlying probability density function.

2. THE SD-ROM ALGORITHM

The SD-ROM algorithm proposed here is a variant of the original algorithm used for filtering images [1]. In image restoration, a 2-D sample window of size 3 by 3 was used. Here we use a 1-D sliding window of size five as shown in Figure 1.

Consider a sample vector $\mathbf{x}(n)$ of size five centered at n. Define $\mathbf{w}(n)$ as the vector of size four that excludes $x(n)$, the center sample under inspection.

$$\begin{aligned} \mathbf{w}(n) &= [w_1(n), w_2(n), w_3(n), w_4(n)] \\ &= [x(n-2), x(n-1), x(n+1), x(n+2)] \end{aligned} \tag{2}$$

X_{n-2}	X_{n-1}	X_n	X_{n+1}	X_{n+2}

Figure 1: Filter window.

The observation samples from $\mathbf{w}(n)$ are sorted,

$$\mathbf{r}(n) = [r_1(n), r_2(n), r_3(n), r_4(n)] \tag{3}$$

where the elements in $\mathbf{r}(n)$ are ordered by rank i.e. $r_1(n) \leq r_2(n) \leq r_3(n) \leq r_4(n)$, so that the elements are arranged in ascending order.

Finally, the rank-ordered differences $d_i(n)$ are defined as

$$d_i(n) = \begin{cases} r_i(n) - x(n), & \text{if } x(n) \leq \mu(n) \\ x(n) - r_{4-i}(n), & \text{if } x(n) > \mu(n) \end{cases} \tag{4}$$

where $\mu(n) = [r_2(n) + r_3(n)]/2$ and is called the rank-ordered mean (ROM). For a window of size five, $i = 1, 2$.

The algorithm decides $x(n)$ is a noise impulse if any of the following conditions hold

$$d_i(n) > T_i, \text{ for } i = 1, 2 \tag{5}$$

where T_1 and T_2 are two appropriately chosen threshold values. In our trials these were typically $T_1 = 4$, $T_2 = 12$ for 8-bit PCM (dynamic range between 0-255).

Every detected impulse is replaced by the ROM, $\mu(n)$.

The algorithm can also be applied in a recursive fashion, where previously filtered impulses are considered. So in the case of a size five sliding window (sample No. 3 under consideration) the first two samples (sample No. 1, 2) are not samples from the input signal, but are ones that have already passed through the SD-ROM filter. This recursive approach improves the performance of the algorithm for highly corrupted samples as the number of impulses in the first half of the sliding window will be reduced or eliminated. It should be noted that the recursive approach can decrease performance for lightly corrupted samples.

3. EXPERIMENTAL RESULTS

As an initial implementation of the algorithm, a music sample (8-bit PCM, sampled at 22.05 kHz) was corrupted with 5% random height impulse noise, with the range of the impulses comparable to the range of the music sample. The results shown here are for the non-recursive approach although the recursive application should yield better results. Our trials with different window sizes showed that while larger windows were not necessarily better, increasing the window size can improve the performance when there are more than two noise impulses within the sample window. However, increasing the window size required chosing additional threshold values. Analysis on varying the window size yielded the results summarized in Table 1.

Table 1: Results for varying window sizes.

Window Size	SNR
5	35.781 dB
7	34.416 dB
9	32.771 dB

As Table 1 shows, a window of size five produced the best signal-to-noise ratio. We set the threshold values to $T_1 = 4$, $T_2 = 12$. Our trials with many combinations of thresholds and samples indicate that these values should work well with most inputs, though the algorithm can be 'tuned' by varying the window size and weights to give better results for a particular sound sample. Table 1 also shows the robustness of the SD-ROM algorithm, as varying the window size does not drastically reduce the performance of the filter as in the case of median or weighted median filters. However, increasing the window size does increase the complexity of the design since a larger number of thresholds is required.

Figure 2 shows a section of the signal corrupted with noise impulses. This section of the input signal has a relatively low amplitude making the impulses obvious. Figure 3 shows the results after filtering with a median filter. Note that the output is modified in several sections where no impulses were added. In most cases, these additional changes are a result of clipping off peaks or filling in valleys in the original signal.

Figure 4 shows the results after filtering with the SDROM filter. The SDROM filter made two kinds of errors. A detection error is an impulse that is not detected or a sample that is erroneously detected as an impulse. An estimation error is the substitution of an incorrect value when an impulse is detected. Both kinds of errors can be seen in the example. However, these errors are both fewer and smaller than the errors made by the median filter.

4. EVALUATION

Table 2 gives a comparison of some of the standard techniques for filtering impulse noise. The table shows the superior performance of SD-ROM (the weights for the window of size seven were $T_1 = 6$, $T_2 = 8$, $T_3 = 14$).

There are other methods for removing noise from corrupted music samples such as techniques using local trigonometric bases and wavelet packets [3] that produce superior results at restoring old recordings. But, these techniques address all forms of noise and are computationally more complex.

Table 2: Performance comparisons.

Filter Type	SNR
Median (Win. Size 5) [2]	32.240 dB
Median (Win. Size 7) [2]	28.639 dB
Wt. Median (Win. Size 5) [4]	32.322 dB
Wt. Median (Win. Size 7) [4]	28.639 dB
SD-ROM (Win. Size 5)	35.781 dB
SD-ROM (Win. Size 7)	34.410 dB

5. FINAL REMARKS

Many applications can be foreseen for this technique. One of the original motivations for this research was to restore old gramophone discs. Scratches and static on these discs are essentially modeled as impulse noise, which are detected and removed by our technique. Other applications of this technique could be in telecommunications, where noise both in regular and cellular telephones and could be reduced without sacrificing tonal quality with a fast hardware implementation of the algorithm.

6. REFERENCES

[1] E. Abreu, M. Lightstone, S. Mitra, and K. Arakawa, "A new efficient approach for the removal of impulse noise from highly corrupted images," *IEEE Trans. on Image Processing*, vol. 5, no. 6, pp. 1012-1025, 1996.

[2] G.R. Arce, N.C. Gallagher, and T. Nodes, "Median filters: Theory and Applications," *Advances in Computer Vision and Image Processing* (T. Huang, ed.), Greenwich, CT: JAI Press, 1986.

[3] J. Berger, R. Coifman, and M. Goldberg, "Removing noise from music using local trigonometric bases and wavelet packets," *J. Audio Eng. Soc.*, vol. 42, no. 10, pp. 808-818, October 1994.

[4] H. Zhou, B. Zeng, and Y. Neuvo, "Weighted FIR median hybrid filters for image processing," *IEEE International Symposium on Circuits and Systems*, vol. 2, pp. 793-796, June 1991.

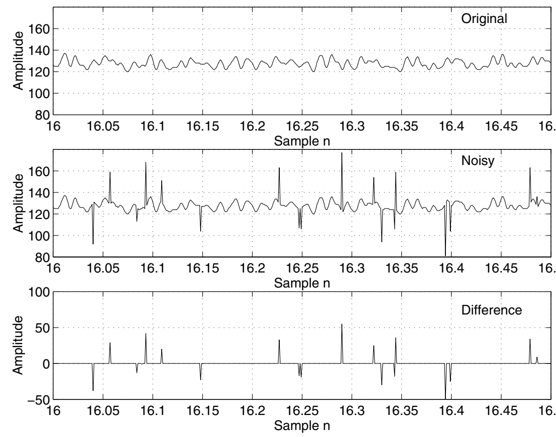

Figure 2: A section of the original and noisy sequences. The third plot is the difference between the sequences and shows the location of the corrupted impulses.

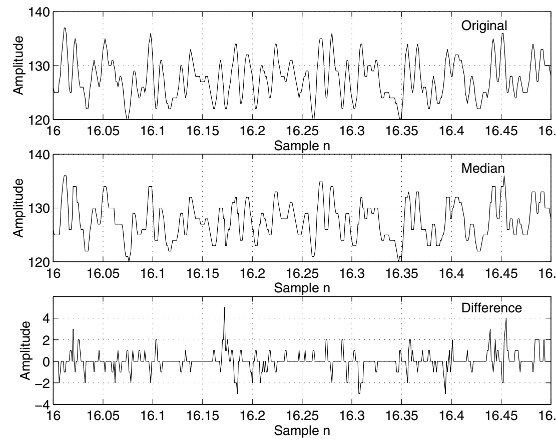

Figure 3: A section of the original and median filtered sequences. The third plot is the difference between the sequences.

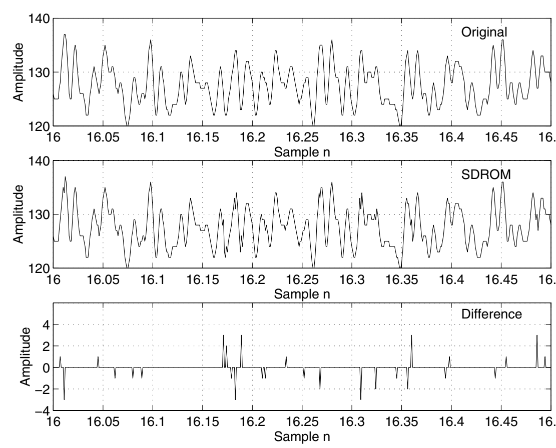

Figure 4: A section of the original and SDROM filtered sequences. The third plot is the difference between the sequences.