Audio Engineering Society
Convention Paper 5535
Presented at the 112th Convention
2002 May 10–13 Munich, Germany

This convention paper has been reproduced from the author's advance manuscript, without editing, corrections, or consideration by the Review Board. The AES takes no responsibility for the contents. Additional papers may be obtained by sending request and remittance to Audio Engineering Society, 60 East 42nd Street, New York, New York 10165-2520, USA; also see `www.aes.org`. *All rights reserved. Reproduction of this paper, or any portion thereof, is not permitted without direct permission from the* Journal of the Audio Engineering Society.

Removal of Long Pulses from Audio Signals Using Two-pass Split-window Filtering

Paulo A. A. Esquef[1], Luiz W. P. Biscainho[2], Vesa Välimäki[3,1], and Matti Karjalainen[1]

[1] *Helsinki University of Technology, Laboratory of Acoustics and Audio Signal Processing, P.O.Box 3000, FIN-02015 HUT, Espoo, Finland*

[2] *Universidade Federal do Rio de Janeiro, LPS-DEL/POLI, Cx. Postal 68564, 21945-970, Rio de Janeiro, RJ, Brazil*

[3] *Tampere University of Technology, Pori School of Technology and Economics, P.O.Box 300, FIN-28101, Pori, Finland*

Correspondence should be addressed to Paulo A. A. Esquef (`esquef@acoustics.hut.fi`)

ABSTRACT

This paper addresses the restoration of audio signals proceeding from old recordings, and focuses on long-pulse removal. We propose a new two-stage method to estimate the waveform of each long pulse from the observed noisy signal. First, an initial estimate for the pulse shape is obtained via a non-linear filtering scheme called two-pass split-window (TPSW) filtering. Then, this estimate is further smoothed through a piecewise polynomial fitting. The degree of smoothness of the estimate can be controlled by adjusting either the TPSW parameters or the length of the segments to be fitted. The proposed method has low computational complexity, it is not constrained by the assumption of shape similarity among pulse waveforms, and can be successfully applied for removing overlapping pulses.

0 INTRODUCTION

One the most annoying degradations associated with old recordings is that produced by severe damage on the disc surface, such as deep scratches or even more extreme cases, e.g., a broken disc whose parts were glued together. These types of degradation are capable of strongly exciting the stylus-arm

reproducing mechanism in such a way that its impulse response is added to the audio signal and yields the so-called long pulses. In this sense, long pulses can be roughly characterized by a short (usually shorter than 2 ms) but strong discontinuity at its very beginning followed by a long (usually longer than 50 ms) and damped transient of low-frequency content. They are also called pops and thumps.

Due to the long duration of the pulses, standard signal interpolation schemes, such as those based on statistical modeling, become prohibitive as a tool to treat the corrupted segment. A possible alternative is to first apply a suitable technique to remove the low-frequency transient of the pulses. After that, the remaining initial impulsive disturbances can be suppressed, for instance, through model-based interpolation. In fact, this strategy is presented in [1] where the authors propose a template matching method to detect and remove the low-frequency transient of long pulses.

The supposition behind the template matching method is that the long pulses are the impulse response of the stylus-arm reproducing system and have therefore nearly identical shapes. Assuming that this hypothesis holds true, the method requires the availability of a clean pulse template. If not directly obtainable from a 'quiet' portion of the recording, one needs a sufficient number of 'noisy' pulses to estimate the clean pulse by averaging out the valuable audio information. However, the pulse shape may vary according to the characteristics of the reproducing apparatus as well as to the type of surface degradation. One way to partly overcome this problem is to use a database with several templates in the matching procedure. Of course, this resort increases the computational cost of the method. Even so, it is still prone to errors when dealing with more general cases such as superimposed pulses.

An alternative method for long pulse removal, which is based on the separation of two autoregressive (AR) processes, is proposed in [2]. In this method each long pulse is modeled as a switched low-order AR process with high-variance excitations associated with the initial discontinuity of the pulse and low-variance excitations associated with its low-frequency transient. On the other hand, the audio signal is modeled as a high-order AR process. The AR-separation method employs block-based processing to cope with the non-stationarity of audio signals. Thus, in the proposed scheme, the signal parameters needed to treat a certain block are estimated from the previously restored block. On the other hand, fixed AR parameters can be used to model the pulse. Notwithstanding, the choice of the variances associated with the pulse model requires some initial experimentation. Moreover, the method is computationally expensive, since it involves a large matrix inversion per block. This burden may be alleviated through an efficient implementation in which the matrix system is solved by means of Kalman filtering [3].

In this paper, a two-stage procedure is proposed to estimate the low-frequency part of long pulses from the corrupted audio signal. First of all, one needs to approximately locate the long pulse occurrences along the signal. This can be accomplished by either a standard model-based inverse filtering, as suggested in [3], or even more simply, by highpass filtering the noisy signal and searching for the locations of the high-amplitude discontinuities which occur at the beginning of the pulses.

For each located pulse, we propose to compute an initial estimate for its shape by first filtering the original noisy signal using a non-linear procedure called two-pass split-window (TPSW) filtering [4]. The TPSW filtering is capable of yielding smooth local-mean estimates along the signal even in cases when the analyzed signal contains spurious spikes.

By observing the typical shape associated with long pulses, one can verify that the oscillations in the low-frequency transient (pulse tail) tend to be faster just after the initial impulsive discontinuities. Such a characteristic shows the importance of controlling the level of smoothness of the pulse estimate along time. This can be achieved by properly adapting the parameters of the TPSW. For instance, by changing the length of the split-window in the TPSW filtering it is possible to control how fast the local-mean estimate can vary. Of course, shortening the window length yields more localized means that are able to follow fast signal variations. On the other hand, this also leads to noisier estimates in the sense that more leakage is observed from the audio signal to the pulse estimate.

A natural heuristic choice when estimating pulse tails via the TPSW filtering is to start using a short split-window just after the initial clicks, and then gradually increase the length of the window towards the end of the pulse.

One way to refine the pulse estimate, especially nearby the impulsive occurences, is to apply a second stage which consists of modeling the TPSW-based pulse estimate by means of a piecewise low-order polynomial fitting scheme. This allows a flexible control over the degree of smoothness of the pulse estimates by varying either the order of the polynomial or the length of the segments to be fitted.

The proposed method has low computational cost and was tested successfully to remove the tail of long pulses in audio signals also in cases of superimposed pulses. It should be noted that it does not remove the initial impulsive disturbance. Nevertheless, this initial click can be suppressed by a declicking algorithm which, as a matter of fact, is usually applied to the whole signal afterwards.

Section 1 of this paper reviews the TPSW filtering and possible strategies to adjust the TPSW length according to the pulse behavior. In Section 2 the polynomial fitting of the TPSW-based pulse estimate is described. Section 3 gives details about the system implementation as well as corresponding examples. Conlusions are drawn in Section 4.

1 TPSW FILTERING

1.1 Basic Definitions

The TPSW filtering is a non-linear procedure originally intended to provide smooth estimates for the background spectrum based on an observed peaky power or magnitude spectrum. Performance comparisons among several filtering strategies for spectral background normalization can be found in [5, 6].

A discrete-time split-window can be defined as

$$h_{\mathrm{SW}}(n) = \begin{cases} 0, & |n| < M \\ 1, & M \leq |n| < N, \end{cases} \tag{1}$$

where M and N are non-negative integers satisfying $1 < M < N$. Hence, the total length of the window is $L = 2N - 1$ and the length of its central gap is $G = 2M - 1$. Note that L and G are odd values just for convenience. As an example, Figure 1 shows $h_{\mathrm{SW}}(n)$ for $N = 10$ and $M = 3$.

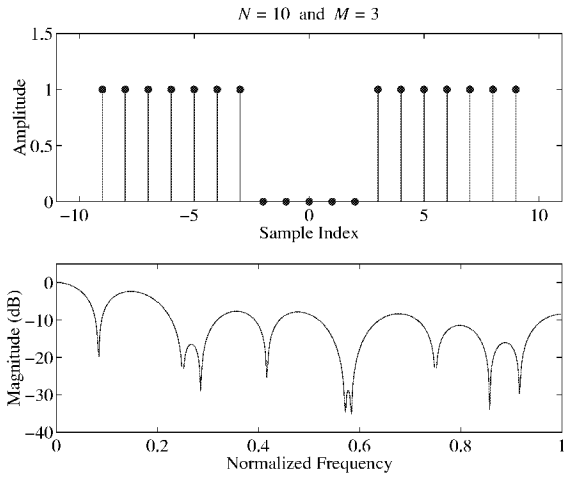

Fig. 1: Split-window for $N = 10$ and $M = 3$.

Consider a positive-valued input sequence $s(n)$, which for background spectrum estimation purposes corresponds to sampling in frequency of a periodogram or magnitude spectrum. A TPSW-based estimate $s_{\mathrm{TPSW}}(n)$ of background spectrum of $s(n)$ can be obtained via the following steps:

1. Compute $\tilde{s}(n)$ by filtering $s(n)$ through $h_{\mathrm{SW}}(n)$, i.e.,

$$\tilde{s}(n) = \frac{1}{2(N-M)} \sum_{k=-N}^{N} h_{\mathrm{SW}}(k) s(n-k) \qquad (2)$$

2. Modify $s(n)$ via the following substitution criterion

$$s'(n) = \begin{cases} s(n), & \text{if } s(n) \leq \alpha \tilde{s}(n) \\ \tilde{s}(n), & \text{if } s(n) > \alpha \tilde{s}(n) \end{cases} \qquad (3)$$

3. Obtain the TPSW-based local-mean estimate $s_{\mathrm{TPSW}}(n)$ by filtering the previously obtained $s'(n)$ through $h_{\mathrm{SW}}(n)$.

The convolution between the magnitude spectrum and the split-window in the first pass works similarly to a moving-average filter. Thus, it is plausible to expect an increase in intermediate local-mean estimate $\tilde{s}(n)$ when a sharp spectral peak is inside the window. However, when the peak coincides with the center gap of the split-window it does not effect the output $\tilde{s}(n)$. Of course, when still inside the window but outside the gap, the peak increases the value of $\tilde{s}(n)$. The substitution criterion and the second pass aim to correct this resulting lateral biasing.

Note that the substitution criterion depends on $\tilde{s}(n)$. As a consequence, for n associated high values of $\tilde{s}(n)$, which corresponds to regions surrounding the peaks, the intermediate sequence $s'(n)$ receives the original input values. Thus, the output of the second pass $s_{\mathrm{TPSW}}(n)$ is free of lateral biasing. The parameter α employed in the substitution criterion controls the reduction of the lateral biasing. In summary, the TPSW filtering estimates the background spectrum by computing local-means which are less sensitive to the presence of peaks along the spectrum.

Regarding the split-window parameters, increasing the value of N yields smoother local-mean estimates. However, it implies a reduced capability of following possible fast variations in the signal. For a fixed value of N the larger the value of M, the noisier the local-mean estimate, since less samples are used in the local-mean computation.

1.2 Modified TPSW Method for Long-Pulse Estimation

To use the TPSW filtering to estimate the low-frequency transient of long pulses it is necessary to modify the substitution criterion in order to cope with a sequence with positive and negative values. One possibility is to replace step 2 shown in Section 1.1 with the following step:

2. Modify $s(n)$ based on the following substitution criterion

$$s'(n) = \begin{cases} \tilde{s}(n), & \text{if } |s(n) - \tilde{s}(n)| > |1 - \alpha||\tilde{s}(n)| \\ s(n), & \text{otherwise.} \end{cases} \qquad (4)$$

Note that in the modified scheme the split-window parameters play the same role as in the original formulation. Moreover, it is implicitly assumed that the length of the window is chosen sufficiently short to follow the supposedly slow variations of the pulse to be estimated, although sufficiently long not to follow the fast variations of the underlying signal. Thus, $\tilde{s}(n)$ will follow the pulse behavior.

As for the substitution criterion and the second pass, it makes sense that the threshold in the substitution criterion be proportional to the magnitude of the estimated mean $\tilde{s}(n)$, which is the output of the first pass. In fact, the output of the second pass, $s_{\mathrm{TPSW}}(n)$, tends to follow the behavior of $\tilde{s}(n)$. Hence, when $\tilde{s}(n)$ is large, $s_{\mathrm{TPSW}}(n)$ will be less sensitive to large sample values of $s'(n)$. Therefore, a larger threshold can be employed. On the other hand, when $\tilde{s}(n)$ approaches zero, $s_{\mathrm{TPSW}}(n)$ will be easily disturbed by momentary high-amplitude values of $s'(n)$, and to prevent this a smaller threshold should be used.

In order to clarify what happens in the intermediate steps of the modified TPSW filtering Figure 2 shows one example featuring the beginning of a long pulse, in which the intermediate sequences are plotted. Thus, one can observe the lateral biasing in $\tilde{s}(n)$ (thick-line plot) around the click location in Figure 2(a). Note also in Figure 2(b) that the lateral biasing is reduced in the final estimate (thick-line plot). Moreover, the dependence of the substitution criterion on $|\tilde{s}(n)|$ can be clearly verified from Figure 2(b), since the closer $\tilde{s}(n)$ is to zero, the more reduced the threshold substitution becomes and more samples are substituted in $s'(n)$ before the second pass.

To illustrate the use of the modified TPSW filtering in long pulse estimation, it is applied to a test signal. This signal, which will be used in the following experiments, consists of an audio excerpt extracted from a broken 78 rpm disc whose parts were glued together. In this particular case, long and superimposed pulses occur. All signals used in this work are sampled at 44.1 kHz.

As an example, one of the superimposed pulse occurrences in the test signal is taken as an input signal for the TPSW-based pulse estimate. First of all, the influence of the window size in the TPSW estimate is verified. For this specific example the central window gap was fixed at $M = 6$ and the value of α

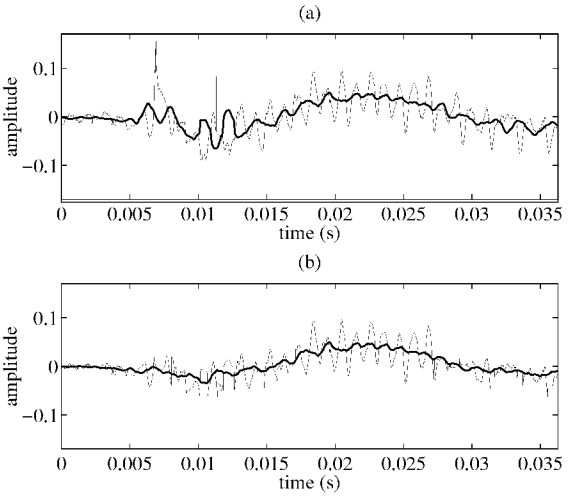

Fig. 2: Intermediate steps of the modified TPSW. (a) Signal corrupted by a long pulse (thin line) and corresponding output of the first pass $\tilde{s}(n)$ (thick line). (b) Intermediate sequence $s'(n)$ obtained after the substitution criterion (thin line) and resulting TPSW estimate $s_{\mathrm{TPSW}}(n)$ (thick line) for $N = 55$, $M = 20$, and $\alpha = 4$.

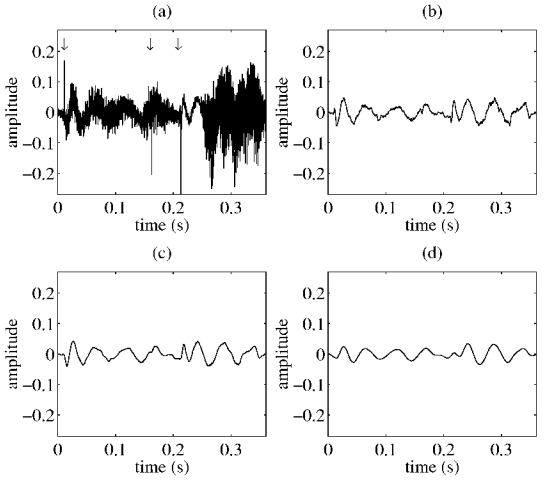

Fig. 3: Example of long pulse corruption and respective TPSW-based pulse estimates. (a) Corrupted signal (the arrows indicate the location of initial clicks). (b), (c), and (d) TPSW-based estimates for $N = 51$, (c) $N = 101$, and (d) $N = 251$, respectively.

was set to 2. Figure 3 shows the original signal and respective TPSW-based pulse estimates for different values of N.

It can be noted from Figure 3 that TPSW estimates with longer window sizes yield smoother estimates. However, pulse

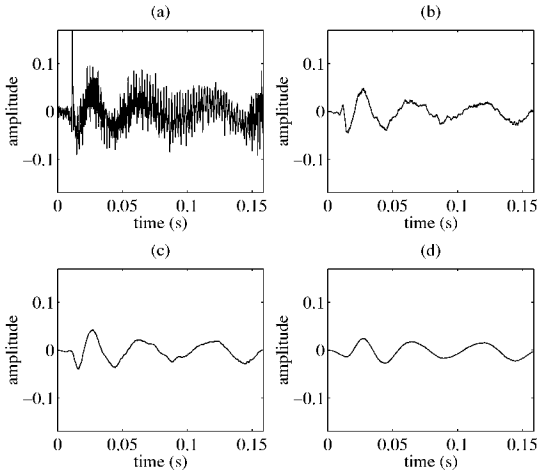

Fig. 4: Detailed view of the beginning of the plots shown in Figure 3.

estimates obtained through long TPSW's may not capture properly the faster pulse variations which occur just after the impulsive disturbance. This behavior can be better observed in Figure 4, which displays the first 16 ms of Figure 3.

1.3 Variable-Length TPSW

The TPSW estimate can be viewed as a local mean estimate which has suitable properties to deal with spurious discontinuities in the signal. For fixed values of the central gap, M, increasing the window length N yields smoother but less localized mean estimates.

As mentioned before, the tail of the long pulses is expected to vary quite smoothly. Therefore, when using the TPSW filtering in its estimation it would be reasonable to employ long split-windows. However, typically, the pulses tend to vary faster just after the occurrence of the initial discontinuity compared with the end of their tails (see Figure 4). This behavior suggests a scheme in which the TPSW length be changed along the estimation procedure may capture properly the initial fast fluctuations. One possibility in this case would be to use a shorter split-window just after the impulsive occurrences and to increase its length towards the end of the pulse. When facing superimposing pulses, the corresponding initial clicks would indicate the regions to temporarily reduce the split-window length and restart the process.

A practical implementation of a variable-length TPSW consists of predefining a template for the behavior of the split-window length starting from the initial click to the end of the pulse. Thus, this would imply changing the length of $h_{\mathrm{SW}}(n)$ accordingly in both passes. For simplicity, instead of changing the length of the split-window, the parameters N and M would stay fixed to certain values and the length of $h_{\mathrm{SW}}(n)$ could be artificially reduced by zeroing the coefficients at the extremities of the window. By adopting this resort it is not necessary to compensate for different net delays, since the length of the delay-line that implements the split-window filter does not change.

The template behavior to guide the variation of the split-window effective length is defined experimentally, for instance, using a signal segment corrupted by a representative pulse. The idea is to obtain a rough estimate of the number of samples that could be associated with the faster initial oscillations, and to set the split-window lengths appropriately during this portion of the signal. In spite of the ad hoc procedure employed, the same window-size template can be used to estimate all the long pulses in the signal.

Another alternative for obtaining pulse estimates using variable-length TPSW's is to compute TPSW estimates for the whole pulse duration using different split-window lengths. Then, the final pulse estimate would be composed by splicing segments from different size TPSW estimates. For simplicity one may restrict the options to three split-window sizes and denote them by N_{small}, N_{medium}, and N_{large}. In this case, the pulse estimate obtained by the large-length TPSW is taken as a basis. However, near the locations of impulsive occurrences this estimate is replaced with those obtained from the small and medium size TPSW's. To avoid discontinuities when splicing the segments a cross-fading scheme must be applied. Again, the lengths of the TPSW as well as the splicing strategy are defined experimentally.

As an example, the result of applying the TPSW splicing procedure to the signal of Figure 4(a) is shown in Figure 5. In this example, three TPSW-based pulse estimates are obtained using split-window sizes of, respectively, $N_{\text{small}} = 25$, $N_{\text{medium}} = 100$, and $N_{\text{large}} = 250$ samples. To compose the pulse estimate the location of the initial click must be known. In this particular case and assuming a sample rate of 44100 Hz, the following strategy is adopted: for the first $T_{\text{small}} = 10$ ms after the click the N_{small} TPSW estimate is used; for the subsequent $T_{\text{medium}} = 30$ ms the N_{medium} TPSW estimate is used; and for the rest of the signal the N_{large} TPSW estimate is used.

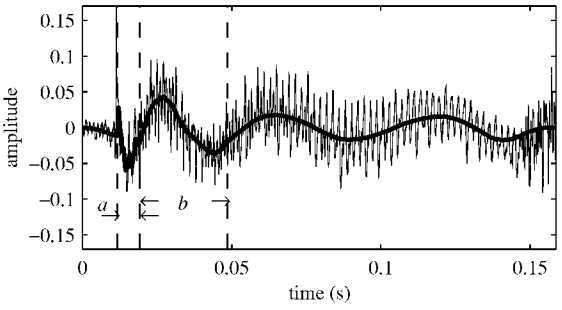

Fig. 5: Corrupted signal (thin line) and pulse estimate (thick line) obtained through splicing TPSW-based estimates of different window sizes. The splicing regions are delimited by vertical dashed lines. The dimension values a and b refer, respectively, to the regions in which TPSW's with N_{small} and N_{medium} are employed.

When dealing with superimposing pulses a similar splicing strategy may be employed. Again, the location associated with the beginning of each pulse present in the segment must be know beforehand. But it is not necessary to treat each pulse separately. On the contrary, the large-size TPSW estimate is used as a basis for the whole segment and the replacements with smaller-size TPSW estimates are performed at appropriate regions.

It can be seen from Figure 5 that the previous scheme is effective to produce a pulse estimate that captures its initial fast variations. The side effect of this approach is to yield noisier estimates close to the click locations.

It must be noted that the TPSW filtering works similarly to a lowpass filtering procedure. In this sense, one could argue in favor of using a properly designed lowpass filtering instead of the TPSW procedure. However, the TPSW filtering has appealing characteristics, such as low sensitivity to spurious spikes in the signal, low computational cost, and simple implementation of variable-length filtering schemes.

1.4 Efficient Implementation of the Split Window Filter

The impulse response of the split window filter defined in Eq. (1) consists of two square pulses, as shown in Fig. 1. Each square pulse contains $N - M$ samples and, when used as a digital filter, corresponds to the following transfer function:

$$H_1(z) = 1 + z^{-1} + z^{-2} + \cdots + z^{-N+M+1} = \frac{1 - z^{-N+M}}{1 - z^{-1}} \quad (5)$$

The latter form suggests an implementation with a cascade of a non-recursive comb filter and a digital integrator. This structure is called a recursive running sum, and it has been known in digital signal processing for a long time [7].

An impulse response with two identical parts can be produced by adding the output of the system with its delayed copy. In the case of the split window, the second square pulse starts $N + M - 1$ samples after the first one. This is realized with the transfer function:

$$H_2(z) = 1 + z^{-N-M+1} \quad (6)$$

The complete split window filter $H_{\text{SW}}(z)$ then consists of the cascade of these two filters, that is, $H_{\text{SW}}(z) = H_1(z)H_2(z)$.

Figure 6 shows the block diagram of the split window filter implementation based on the above transfer functions and a scaling by multiplying coefficient $g = 0.5(N - M)$. It is seen that the filtering requires 3 additions and 1 multiplication per output sample. $2N - 1$ words of sample memory are needed for the delay lines. Note that the computational complexity does not depend on N or M. The memory consumption is a function of N and not of M, because as M is changed, one of the delay lines gets longer while the other one gets shorter by the same amount (see Fig. 6).

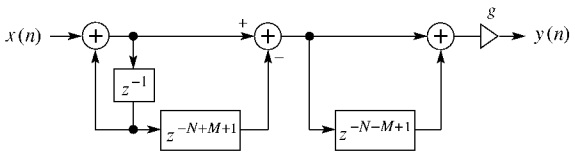

Fig. 6: Efficient implementation of the split window filter as a cascade of a recursive-running-sum structure (left-hand side) and a non-recursive comb filter (right-hand side).

2 PIECEWISE POLYNOMIAL FITTING

The TPSW estimates obtained in the first stage can be considered initial estimates for the long pulses. Usually they still contain valuable signal information, and this indicates that the estimates are not smooth enough. To further smooth them, each TPSW-based pulse estimate is submitted to a second stage. In this stage the final pulse estimates result from a piecewise polynomial fitting over the previously obtained TPSW-based pulse estimates.

For each previously obtained TPSW-based pulse estimate s_{TPSW} the piecewise polynomial fitting procedure consists of the following steps:

- Segmenting $s_{\mathrm{TPSW}}(n)$ in overlapping short frames;
- Fitting a low-order polynomial to each frame, for instance, using the least-squares criterion;
- Windowing the fitted curves, for instance, using a Hanning window;
- Summing up the windowed curves as in an overlap-and-add scheme.

A schematic diagram which shows the steps used in the piecewise polynomial fitting is depicted in Figure 7.

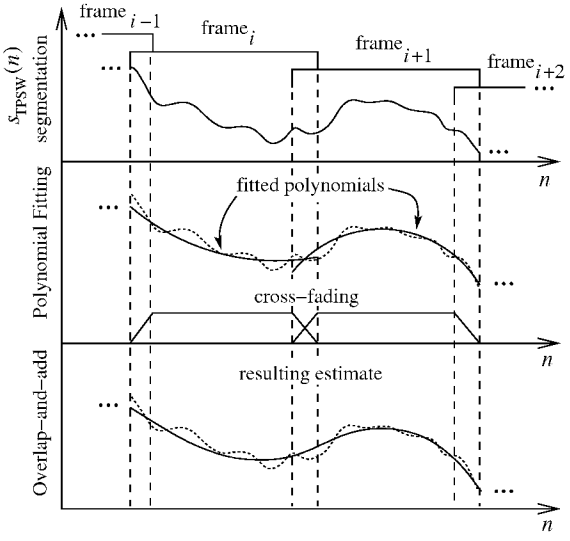

Fig. 7: Illustrative scheme of the piecewise polynomial fitting.

Here, the overlap-and-add acts as a cross-fading scheme to avoid discontinuities between successive frames. It should be noted that the windowing must not introduce amplitude distortion to the resulting pulse estimates, except at their extremities. Thus, if for example Hanning windows are employed, an overlap factor of 50% or 75% must be adopted. Moreover, the piecewise polynomial fitting allows controlling the degree of smoothness along the signal by varying either the length of the frames or the polynomial order.

It was found experimentally that adopting second order polynomials and segmenting the signal in frames of 150 samples

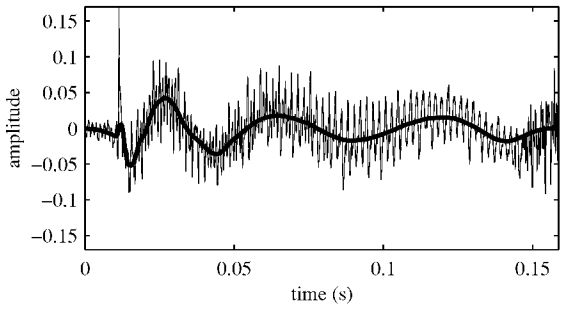

Fig. 8: Corrupted signal and pulse estimate obtained through piecewise polynomial fitting over the TPSW-based pulse estimate of Fig. 5.

yields satisfactory results. For cross-fading successive frames 150-sample Hanning windows were used and the overlap factor was set to 50%. Figure 8 shows the result of the piecewise polynomial fitting applied to the TPSW-based pulse estimate shown in Figure 5. It can be seen that although the pulse estimate is smoother, the faster oscillations which characterize the beginning of the pulse are still captured.

3 PERFORMANCE ASSESSMENT AND RESULTS

The performance of the TPSW-based pulse estimation is evaluated by means of a set of case studies. In addition, the TPSW-based pulse estimates are compared with those obtained by the AR-separation method proposed in [2]. The signals under study are sampled at 44.1 kHz in all examples shown within this section. The processing parameters used in the following case studies are given in Tables 1 and 2.

3.1 Case Study 1: Artificial Single Pulse

An artificial pulse can be formed by concatenating a pre-pulse part that has zero amplitude, a high amplitude click, and a low-frequency transient or tail part. The tail part can be for example imitated by

$$p_{\mathrm{tail}}(n) = e^{\frac{n}{\tau_{\mathrm{e}}}} \sin\left(2\pi n f_n - \frac{\pi}{4}\right), \tag{7}$$

with

$$f_n = (f_{\max} - f_{\min}) e^{\frac{n}{\tau_{\mathrm{f}}}} + f_{\min}, \tag{8}$$

where

- τ_{e} is the time-constant in seconds associated with the envelope decay of the pulse;
- $f_{\max}$ and $f_{\min}$ are, respectively, the maximum and minimum oscillation frequencies in Hz allowed in the pulse's tail part;
- τ_f is the time-constant in seconds associated with the frequency decay of the pulse.

An example of an artificially generated pulse is displayed in Figure 9(a). In this example, the pre-pulse part has 500 samples and the high-amplitude click is a 23-samples long impulsive disturbance taken from the beginning of a real pulse. Parameters associated with the tail part are: $\tau_{\mathrm{e}} = 0.07$ s,

$\tau_f = 0.015$ s, $f_{\max} = 60$ Hz, and $f_{\min} = 20$ Hz. Figure 9(c) shows an example in which the generated pulse is added to the clean audio segment plotted in Figure 9(b).

Now, the signal shown in Figure 9(c) is applied to the TPSW-based method and to that based on AR-separation.

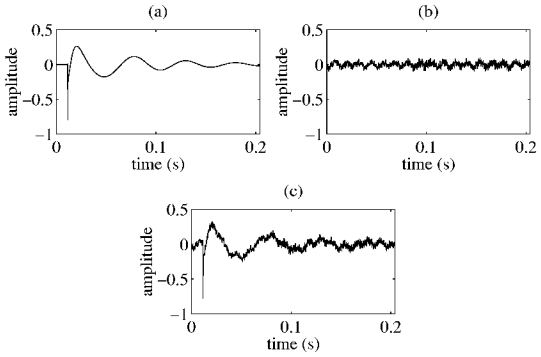

Fig. 9: Case study 1: (a) Synthetically generated long pulse, (b) clean audio segment, and (c) artificially corrupted signal by summing the previous two.

Table 1: Values of the parameters used in the TPSW-based method.

TPSW	
N_{large}	250 samples
N_{medium}	50 samples
N_{small}	25 samples
T_{medium}	30 ms
T_{small}	10 ms
M	5 samples
α	2
Piecewise Polynomial Fitting	
Frame size	150 samples
Window type	Hanning
Overlap factor	50%
Polynomial order	2

Table 2: Values of the parameters used in the AR-separation method (see [3] for more details).

AR-separation Method	
Block size	1000 samples
Overlap	200 samples
Model-order (signal)	60
Model-order (pulse)	2
Pulse model	$x(n) = 2x(n-1) - x(n-2)$
σ^2 (click part)	0.05
σ^2 (pre-pulse part)	1×10^{-20}
σ^2 (tail part)	1.5×10^{-8}
σ^2 decay (tail part)	1.4 per block

The obtained results are depicted in Figure 10, from where it can be noticed that the methods yield similar results, except for the initial click, which is not removed by the TPSW-based method. Moreover, by comparing Figure 9(b) and Figures 10 (e) and (f), it can be noticed that the restored signals suffer from localized low-frequency distortion, especially in the portion that follows the click occurrence (note the absence of the low-frequency undulations in the restored signals but which are present in the uncorrupted signal).

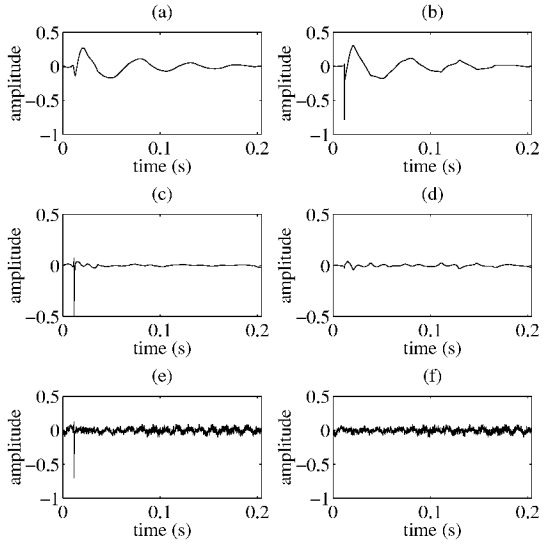

Fig. 10: Case study 1: performance comparison between the TPSW-based (first column) and AR-separation (second column) methods. (a) and (b) estimated pulses; (c) and (d) errors in pulse estimation; (e) and (f) restored signals.

3.2 Case Study 2: Artificial Superimposed Pulses

In this case study, the TPSW-based and AR-separation methods are evaluated for treating an artificially generated superimposing pulse. This pulse is obtained by adding the artificial pulse generated in Section 3.1 with a delayed copy of the same pulse. A between-pulse delay of 4000 samples was chosen in this case.

Figure 11 shows plots of the devised pulse, the uncorrupted audio, and its corresponding corrupted version obtained after adding the pulse. The corrupted signal is then submitted to the TPSW-based method as well as to that based on AR separation.

The obtained results for this case are plotted in Figure 12. Again, except for the initial click, which is not removed by the TPSW-based method, the results are similar, and some level of low-frequency distortion is observed in both the restored versions shown in Figures 12 (e) and (f), especially right after the click locations.

3.3 Case Study 3: Real Superimposed Pulses

The aim of this section is to evaluate the performance of TPSW-based pulse estimation when dealing with real occurrences of superimposed pulses. The test segment used along this section is extracted from a broken 78 rpm disc whose

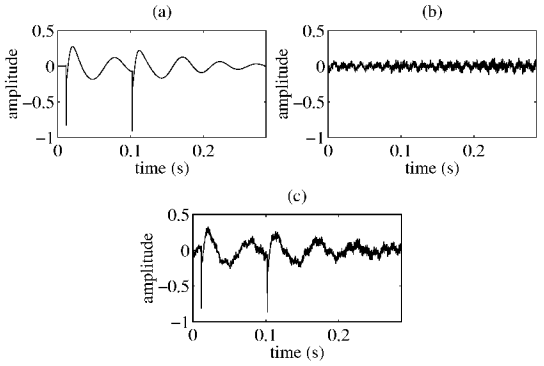

Fig. 11: Case study 2: (a) Synthetically generated superimposed pulses, (b) clean audio segment, and (c) artificially corrupted signal resulted from adding the previous signals.

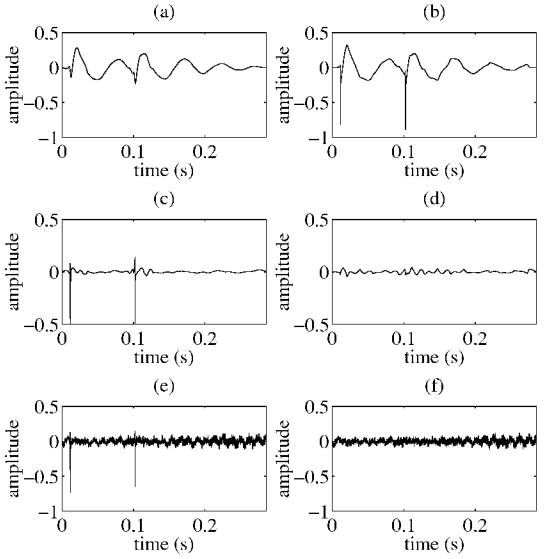

Fig. 12: Case study 2: performance comparison between the TPSW-based (first column) and AR-separation (second column) methods. (a) and (b) estimated pulses; (c) and (d) errors in pulse estimation; (e) and (f) restored signals.

parts were glued together. An example of the long pulse produced by such surface degradation is plotted in Figure 13 (a). The remaining plots in this Figure refer to the pulse estimates obtained via the TPSW-based and AR-separation methods and the respective restored versions of the test signal.

Now the restored version depicted in Figure 13 (d) does not contain residual clicks because after the pulse removal a declicking algorithm was applied to the signal. In these examples, the model-based declicking algorithm described in [8] was employed. In fact, as the remaining clicks have short duration, any good impulsive noise removal technique may be used to suppress them [2, 1]. Indeed, they could be left to be removed together with all other corrupting clicks present in the signal during the declicking stage that is usually included in a typical processing chain of audio restoration.

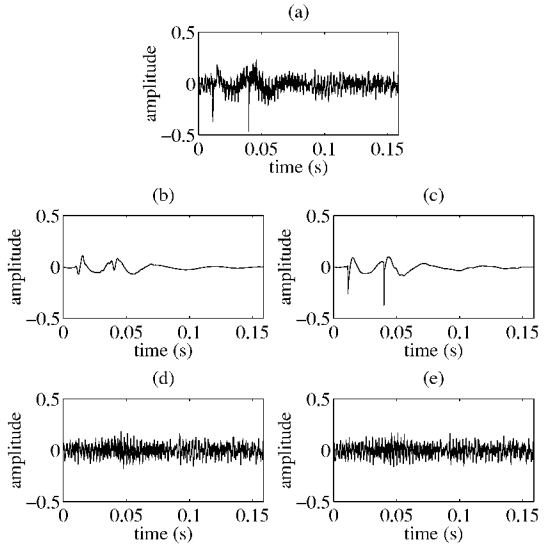

Fig. 13: Case study 3: performance comparison between the TPSW-based (first column) and AR-separation (second column) methods. (a) real example of corrupted signal; (b) and (c) estimated pulses; (d) and (e) restored signals.

3.4 Case Study 4: Corrupting Pulses in Audio Segments with Prominent Low-Frequency Content

A critical situation for long pulse estimation occurs when the pulse location coincides with a signal segment that has strong low-frequency content to be preserved. In this case, the TPSW-based estimator may introduce localized spectral distortion at low frequencies, especially close to the initial impulsive disturbances. One example of such a situation is shown in Figure 14. It can be noted that the first pulse is properly estimated. However, the following pulse occurs almost simultaneously with a low-frequency event around 0.2 s. As a consequence, one can see that the resulting pulse estimate shown in Figure 14 (b) contains valuable signal information which will be removed together with the pulse. As in the previous case, the restored version plotted in (d) is a declicked version.

The proposed pulse estimation scheme is not completely automated. In fact, due to the great variability of pulse shapes one may find in practice, it is recommended that first a representative segment of the signal containing a pulse be taken and, based on its characteristics, the estimator parameters be calibrated as to yield a satisfactory pulse estimate. Then, with the previously selected parameters the pulse estimator is applied for all located long pulses in the signal. Sound examples are available at URL:
http://www.acoustics.hut.fi/publications/papers/aes112-LP/

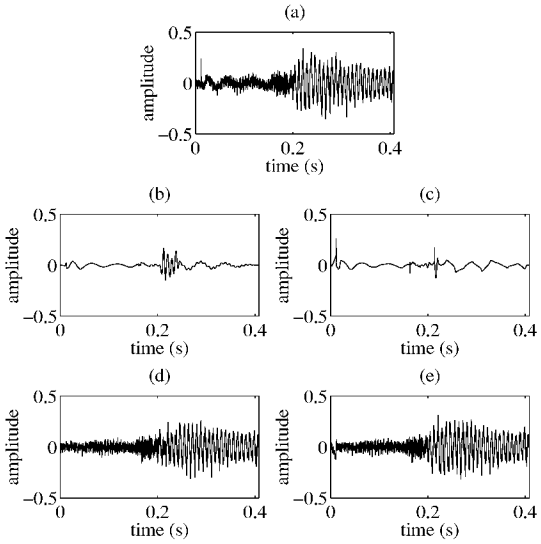

Fig. 14: Case study 4: performance comparison between the TPSW-based (left column) and AR-separation (right column) methods. (a) real example of corrupted signal; (b) and (c) estimated pulses; (d) and (e) restored signals.

4 CONCLUSIONS

This paper presented a new method to estimate and remove the low-frequency transient of long pulses in the context of digital restoration of old gramophone recordings. The pulse estimation is carried out in two stages. The first stage employs a non-linear filtering technique called two-pass split-window, which in this work was adapted to the problem at hand. The computational cost of this filtering is very low. To cope with the characteristics related to long pulses, a scheme was devised where an intermediate pulse estimate is composed by splicing estimates obtained through the adoption of split-windows with different temporal resolutions. The second stage smooths the estimate obtained in the first stage by means of a piecewise polynomial fitting.

The resulting estimates obtained with the proposed method were found to be comparable to those attained by the model-based AR separation method, which is computationally more costly. Contrarily to this method, the proposed one does not remove the initial impulsive disturbances of the pulses. However, they can be treated by a standard declicking algorithm, which usually is needed as one stage within the processing chain that audio restoration involves.

5 ACKNOWLEDGMENTS

The work of Paulo Esquef has been financed by the Brazilian National Council for Scientific and Technological Development (CNPq-Brazil) and by the Academy of Finland project "Technology for Audio and Speech Processing".

REFERENCES

[1] S. V. Vaseghi and R. Frayling-Cork, "Restoration of Old Gramophone Recordings," *J. Audio Eng. Soc.*, vol. 40, no. 10, pp. 791–801, Oct. 1992.

[2] S. J. Godsill and P. J. W. Rayner, *Digital Audio Restoration — A Statistical Model Based Approach*. London, UK: Springer-Verlag, 1998.

[3] S. J. Godsill and C. H. Tan, "Removal of Low Frequency Transient Noise from Old Recordings Using Model-Based Signal Separation Techniques," in *Proc. IEEE WASPAA'97*, (New Paltz, New York), 1997.

[4] R. O. Nielsen, *Sonar Signal Processing (Acoustic Library)*. Artech House, 1991.

[5] W. A. Struzinski and E. D. Lowe, "A Performance Comparison of Four Noise Background Normalization Schemes for Signal Detection Systems," *J. Acoust. Soc. Am.*, vol. 76, no. 6, pp. 1738–1742, Dec. 1984.

[6] W. A. Struzinski and E. D. Lowe, "The Effect of Improper Normalization on the Performance of an Automated Energy Detector," *J. Acoust. Soc. Am.*, vol. 78, no. 3, pp. 936–941, Sept. 1985.

[7] L. R. Rabiner and B. Gold, *Theory and Applications of Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975.

[8] P. A. A. Esquef, L. W. P. Biscainho, P. S. R. Diniz, and F. P. Freeland, "A Double-Threshold-Based Approach to Impulsive Noise Detection in Audio Signals," in *Proc. EUSIPCO'00*, (Tampere, Finland), pp. 2041–2044, Sep. 2000.