EXTRACTION AND REMOVAL OF PERCUSSIVE SOUNDS FROM MUSICAL RECORDINGS

John Usher

Sound Recording Area
McGill University, Montreal, Canada
jusher@po-box.mcgill.ca

ABSTRACT

Automated removal and extraction (isolation) of percussive sounds embedded in an audio signal is useful for a variety of applications such as speech enhancement and for music processing effects. A novel method is presented to accomplish both extraction and removal of beats, using an adaptive filter based on the LMS algorithm. Empirical evaluation is undertaken using computer generated music with a mix of natural voice and repeating drum, and shows that the efficacy of the system is robust to different sound processing techniques such as non-linear distortion and tempo jitter.

1. INTRODUCTION

Extraction and removal of percussive sounds from musical recordings are two different things; the former requires that only the beats be left after the audio signal processing, and that the user of such a "beat-extractor" wishes the extracted beats to be undistorted relative to those beats used to create the original musical piece. Beat *removal*, on the other hand, implies that the user is not concerned whether the beats themselves are recoverable from the audio mix. In the latter case, it is the non-percussive sounds which are of interest. The motivations for either objective are varied. An example is as an effect for reproduction of pre-recorded music, such as live DJ performances where a percussive beat from one musical piece can be mixed with the non-beat component of a second piece. Another use is for audio engineering to "fix" a recording which has already been mixed to two-channels, yet the engineer wishes to keep only the drum part of the mix, or everything *BUT* the drum part. A third use of such a beat extracting or removal device is for music production using samples from prerecorded music; a very common phenomenon in modern popular music.

2. SYSTEM OVERVIEW

The system architecture for the new beat extractor/ remover is described in figure 1. The musical input signal is first processed by a rhythmic feature analyser. The function of this device is to extract timing data about percussive events in the musical piece, classifying the sound into onsets and time intervals. Analysis of beat for music with strong percussive content relies on finding the local maximum in the power spectra envelope [1, 2], and can be used with measures of either auto-correlation [1] or outputs from resonant filters [3] to determine inter onset interval. A hierarchical representation of rhythmic patterns can be extracted to give the basic pulse-rate (the *tactus*) and higher structural levels such as bar-length [4]. However, the exploratory discussion of the system presented here is concerned only with identification of the basic (foot-tapping) pulse, which in the foregoing experiment was identified manually from a single channel of computer-generated music.

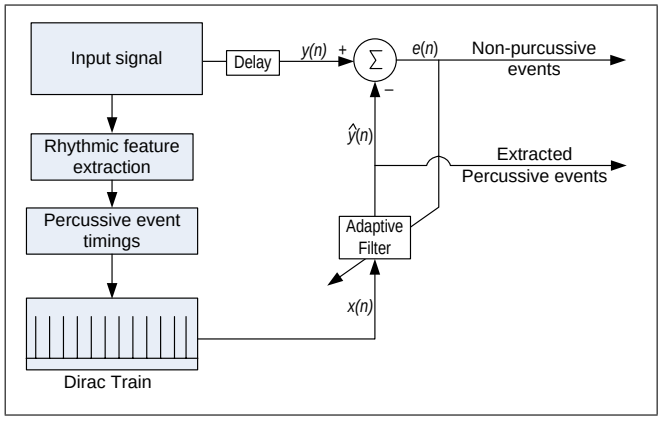


Figure 1: *System overview.*

Once the basic pulse has been identified, a scaled dirac train is created which is defined as a single sample value of unity at the start of each beat. The delay (τ) on the input signal (i.e. to create signal $y(n)$) ensures that the onset time is not critical; the onset estimate could be up to τ samples earlier than the actual onset time: the adaptive filter can compensate for such timing discrepancies with non-minimum phase filter coefficients. Such inaccuracy may be caused by a noisy energy-envelope threshold-based system, or a "true-event" time which occurs *between* samples. Effect of such timing jitter in the estimation of onset time on system performance is investigated later.

The basic goal of the system is to filter the dirac train so as to minimize the difference signal (measured as the mean-square energy difference) between it and the input audio signal; hence the adaptive filter coefficients are updated according to the Least Means Square (LMS) algorithm [5]. The premise of this is that the percussive musical sounds which occur at the dirac event times are caused by the same musical source. This is valid for computer-generated music such as techno, though to ensure that the timings for the dirac train correspond to percussive events from the same instrument an analysis of the local percussive event must be undertaken; feature extraction methods for accomplishing this, such as using harmonic analysis, are suggested in [6]. A major assumption is that time-varying non-percussive events (such as voice) will be

different on each identified transient, so the filter will not adapt to model these and the filtered dirac train (signal $\hat{y}(n)$) will resemble a train of just the percussive sounds.

The convolution of the dirac train signal $u(n)$ with the M-length adaptive filter $\mathbf{h}$ gives signal $\hat{y}(n)$:

$$\begin{aligned}\hat{y}(n) &= \sum_{k=0}^{M-1} x(n-k)h_k \\ &= \mathbf{x}^T(n)\mathbf{h}.\end{aligned} \tag{1}$$

Where:
$\mathbf{x}(n) = [x(n), x(n-1), ..., x(n-M+1)]^T$,
$\mathbf{h} = [h_0, h_1, ..., h_{M-1}]^T$.
It is this filtered dirac signal $\hat{y}(n)$ which approximates the beat occurring at the same time as the dirac impulse; hence this signal can be considered the "extracted" percussive events.

The delayed input audio signal $y(n)$ is then subtracted from the filtered dirac train $\hat{y}(n)$ to give the error signal $e(n)$:

$$e(n) = y(n) - \hat{y}(n). \tag{2}$$

If the adaptive filter coefficients match the actual percussive event at the time of the dirac pulse, then the percussive event would be totally canceled from the original input signal. Hence, under such optimal filter conditions the error signal can be considered to have the percussive events removed (i.e. the percussive events which have the same onset time as the dirac pulse).

The adaptive filter is adjusted over time so as to decrease the error signal level. This goal is formally expressed as a "performance index" or "cost" scaler J, where for a given filter vector $\mathbf{h}$:

$$J(\mathbf{h}) = E\left\{e^2(n)\right\}, \tag{3}$$

and $E\{\cdot\}$ is the statistical expectation operator. The requirement for the algorithm is to determine the operating conditions for which J attains its minimum value. This state of the adaptive filter is called the "optimal state" [5].

When a filter is in the optimal state, the rate of change in the error signal level (i.e. J) with respect to the filter coefficients $\mathbf{h}$ will be minimal. This rate of change (or gradient operator) is an M-length vector $\boldsymbol{\nabla}$, and applying it to the cost function J gives:

$$\boldsymbol{\nabla} J(\mathbf{h}) = \frac{\partial J(\mathbf{h})}{\partial \mathbf{h}(n)}. \tag{4}$$

The right-hand-side of the last equations are expanded using partial derivatives in terms of the error signal $e(n)$ from equation (3):

$$\frac{\partial J(\mathbf{h})}{\partial \mathbf{h}(n)} = 2E\left\{\frac{\partial e(n)}{\partial \mathbf{h}(n)} e(n)\right\}. \tag{5}$$

Updating the filter vector $\mathbf{h}$ from time sample $(n-1)$ to time (n) is done by multiplying the negative of the gradient operator by a constant scaler and the filter update (i.e. the steepest descent gradient algorithm) is:

$$\mathbf{h}(n) = \mathbf{h}(n-1) + \frac{\alpha}{\delta + \mathbf{x}^T(n)\mathbf{x}(n)}\mathbf{x}(n)e(n) \tag{6}$$

with

$$0 < \alpha < 2.$$

δ is a regularization constant to ensure against computational errors when the power estimate of the input signal is too low (this update version is called the Normalized LMS algorithm [5]).

Besides the massive increase in computational efficiency of implementing the filter-update and signal filtering in the frequency domain (requiring 5 FFT's per iteration; i.e. for every M input samples), the performance of the frequency-domain and time-domain NLMS algorithm are equivalent [7]. The overlap-save technique was used (as described in [7]) with an overlap factor of two (performance was not significantly affected by an increase in overlap). In the filter update, the time-domain constraint (to ensure against "wrap-around" errors when M is less than the length of the actual impulse response) was affected so as to weight later coefficients less than early ones; a modification known as the "exponential step" (ES) algorithm [8]. This ensures an exponential decay of the extracted beat. Furthermore, a parallel multi-filter approach was implemented whereby three simultaneous filters ran with different update (α) parameters [9]; this allowed for fast initial convergence (subjectively, performance stabilized after 1 or 2 iterations) and robustness to sudden changes in envelope from new sounds.

3. ELECTRONIC VALIDATION

3.1. Test stimuli

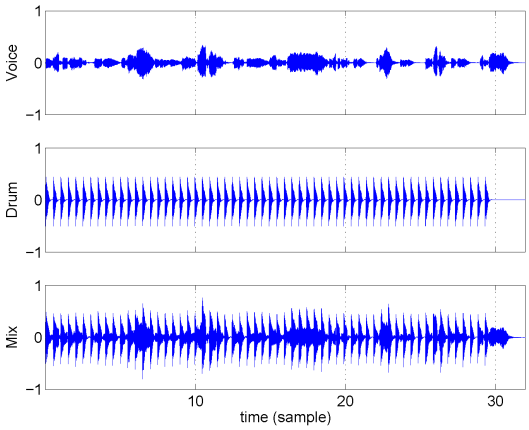
For this exploratory investigation of the proposed system, a simple test stimulus was created by the summation of a sung voice and kick-drum train, as shown in figure 3. The drum sound was a typical decaying techno bass-drum sample (14000 samples long), and was repeated at intervals of 21900 samples (i.e. 121 beats per minute). The voice tempo was synchronized with the drum. As can be seen from the spectral analysis, the two musical instruments overlap in frequency (within 5-10 dB) for nearly two octaves centred about 500 Hz.

To investigate the effect of non-linear distortion on the system response, two common audio-processing techniques were applied to the mono mix; compression and reverberation. The dynamic compression algorithm applied an increasing gain to low-level sounds with a 2:1 ratio (quite an extreme case of compression), with a 20 ms attack time and 40 ms release time. Reverberation was artificially simulated using a commercially available processor with a reverberation time of 2.6 seconds. The RMS-averaged energy for the processed stimuli were matched with the original mono mix. It was expected that any non-linear distortion would reduced the efficacy of the system, creating a mismatch between the adaptive filter and the time-variant percussive sound (i.e. the optimal filter condition would also be time variant).

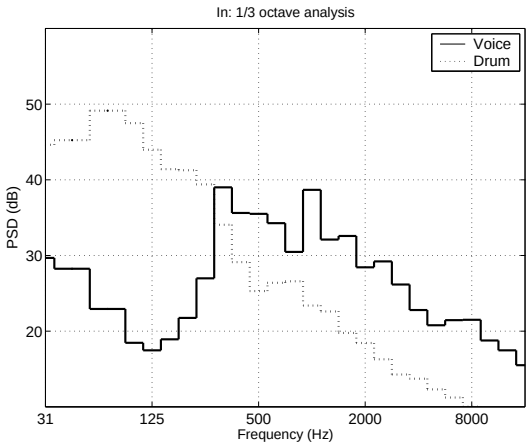
Furthermore, to simulate the effect of inaccuracies in the pulse event timing analysis system, the true beat-onset time (which was, of course, known *a priori* when the beat-track was created) was randomized by adding a gaussian-shaped noise process with a mean of zero and a variance of 0-100 samples (a kind of timing jitter).

3.2. System output response

As can be seen from the time-domain signal outputs in figure 3, the extracted drum beat and voice signal were very similar to the original input signals. Subjectively, the extracted bass-drum was distortion free after a single iteration . However, the extracted voice signal (or rather, the input signal with the drum-beat removed) had



(a) *Time-domain plot.*



(b) *1/3 octave frequency analysis.*

Figure 2: *Details of input test stimulus. The mono mix used in the simulations was created by the summation of a voice and a drum track. The drum beat was created electronically from a single sample of a kick drum repeated at regular intervals.*

noticeable distortion artifacts. This was partly due to the exponential window smoothing which forced the adaptive filter coefficients to decay to zero faster than the actual decay of the drum beat (compare the drum envelope in the lower two plots of figure 3). Distortion was also noticeable at repeating intervals related to the block size length caused by high energy narrow-band resonances in the voice. Further work is needed to "fine-tune" the algorithm so that the power-estimate analysis used in the filter update has a memory which can account for such sudden resonances.

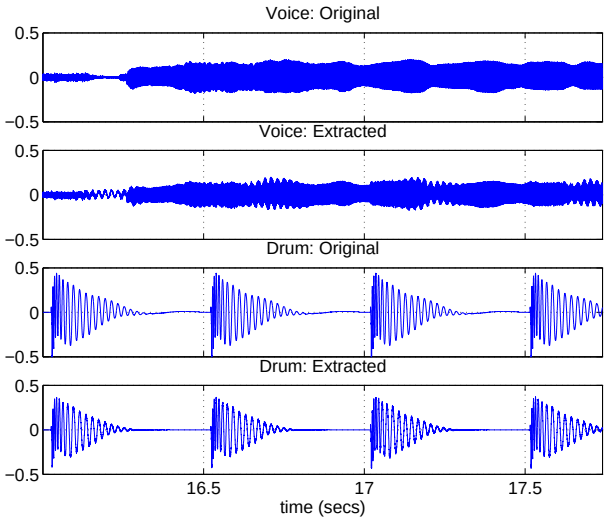


Figure 3: *1.8 second snippet of original voice and drum input signals and extracted output signals.*

Considering an optimal solution set of filter coefficients $\mathbf{h}$ and a current set of filter coefficients $\hat{\mathbf{h}}$ then the magnitude of the difference or mismatch between the two can be expressed as a simple dimensionless quantity ξ called the *misalignment* [10]:

$$\xi = \frac{\left|\left|\mathbf{h} - \hat{\mathbf{h}}\right|\right|}{||\mathbf{h}||}, \tag{7}$$

where $||\cdot||$ denotes the two-norm of a vector. In the new system, $\mathbf{h}$ is really the actual percussive event (i.e. the optimal solution) and $\hat{\mathbf{h}}$ is the adaptive filter coefficients (i.e. the approximated beat). In this study, the optimal solution is know *a priori*- they are the individual beats used to create the drum channel.

As can be seen in figure 4, the non-linear processing of the original mono mix signal reduced the system performance (as measured in terms of misalignment). This was expected, though it should be noted that the subjective degradation in performance was not noticeable for the case when artificial reverb was added, and that for the processing involving compression the extracted drum channel was also subjectively undistorted.

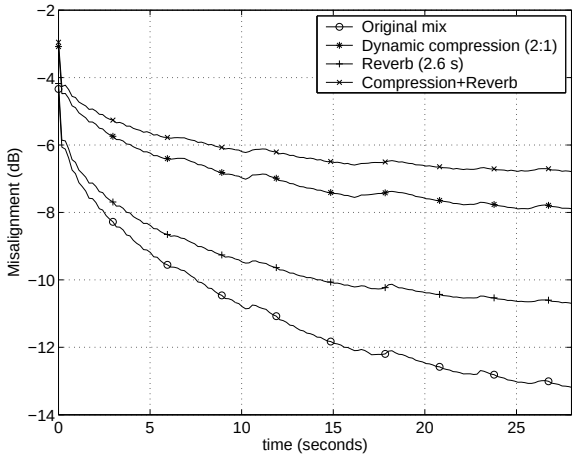


Figure 4: *Misalignment for different non-linear audio processing techniques applied to the original mono mix.*

Figure 5 shows the degree of robustness of the system to a jitter in the estimated beat-onset detector. For a given variance σ, 16% of the event timings (i.e. the sample time of a given dirac pulse) will be later or earlier than the true event time by σ samples. Subjectively, the extracted drum and voice signals were undistorted if this variance was below 50 samples. This is still a relatively low tolerance (a $1/16^{th}$ note inter-event for a 120 BPM beat is 1300 samples), and to accomplish such a necessary beat-tracking accuracy for live (rather than computer-generated) music would be difficult.

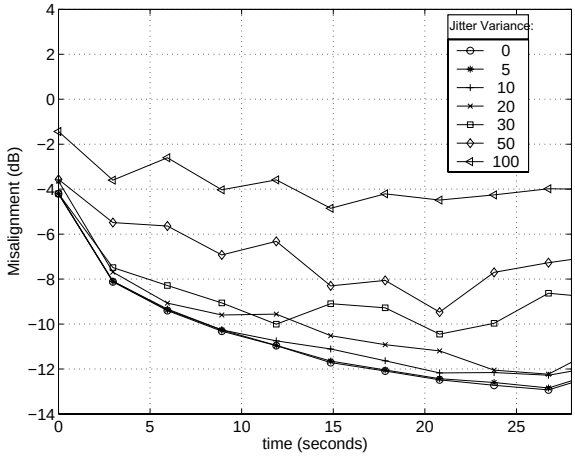


Figure 5: *Effect of event timing jitter on misalignment. Different curves show magnitude of jitter: the variance (in samples) for a normal distribution centred about the actual event timing.*

4. CONCLUSION

A novel application of adaptive filters for the purpose of extraction and removal of percussive sounds from music recordings has been presented. The proposed system relies on a rhythmic analysis device which extracts event timings for the percussive sounds and creates a dirac train with the pulse located at the percussive event onset time. The dirac train is then filtered with an adaptive filter updated according to the NLMS algorithm and the filtered signal approximates the percussive event. The system was tested with a computer-generated musical signal created from a repeating bass-drum and a sung voice. Subjective audition and electronic measurement of the extracted drum signal shows very promising effectiveness. However, distortion artifacts are larger in the non-percussive audio output channel which suggests the system is better suited for beat *extraction* rather than beat *removal.*

5. REFERENCES

[1] M. Goto, "An audio-based real-time beat tracking system for music with or without drum sounds," *Journal of New Music Research*, vol. 30, no. 2, pp. 159–171, 2001.

[2] C. Duxbury, M. Davies, and M. Sandler, "Separation of transient information in musical audio using multiresolution analysis techniques," in *Proc. of the 4th Int. Conference on Digital Audio Effects*, Limerick, Ireland, 2001.

[3] E. D. Scheirer, "Tempo and beat analysis of acoustic musical signals," *Journal of the Acoustical Society of America*, vol. 103, no. 1, pp. 588–601, 1998.

[4] C. Uhle and J. Herre, "Estimation of tempo, micro time and time signature from percussive music," in *Proc. of the 6th Int. Conference on Digital Audio Effects*, London, UK, 2003.

[5] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, Englewood Cliffs, N. J., 4th edition, 2001.

[6] E. D. Scheirer, *Music-Listening Systems*, Ph.D. thesis, Massachusetts Institute of Technology, 2000.

[7] P. C. W. Sommen, P. J. VanGerwen, H. J. Kotmans, and A. J. E. M. Janssen, "Convergence analysis of a frequency-domain adaptive filter with exponential power averaging and generalized window function," *IEEE Trans. on Circuits and systems*, vol. 34, no. 7, pp. 788–798, 1987.

[8] S. Makino and Y. Kaneda, "Acoustic echo canceller algorithm based on the variation characteristics of a room impulse response," in *Proc. ICASSP*, 1990, pp. 1133–1136.

[9] J. Usher, J. Cooperstock, and W. Woszczyk, "A multi-filter approach to acoustic echo cancelation for teleconferencing," in *Proceedings of the 147th Meeting of the Acoustical Society of America*, New York, 2004.

[10] J. Benesty, T. Gänsler, and P. Eneroth, "Multi-channel sound, acoustic echo cancellation, and multi-channel time-domain adaptive filtering," in *Acoustic Signal Processing for Telecommunication*, S. L. Gay and J. Benesty, Eds., chapter 6, pp. 101–120. Kluwer Academic Publishers, 2000.