A Transient Detection Algorithm for Audio Using Iterative Analysis of STFT.

Balaji Thoshkahna

Lab126

15 PUBLICATIONS **35** CITATIONS

Francois Xavier Nsabimana

Helmut Schmidt University

9 PUBLICATIONS **20** CITATIONS

Kalpathi Ramakrishnan

Indian Institute of Science

142 PUBLICATIONS **2,285** CITATIONS

A TRANSIENT DETECTION ALGORITHM FOR AUDIO USING ITERATIVE ANALYSIS OF STFT

Balaji Thoshkahna
Dept. of Electrical Engineering,
Indian Institute of Science,
Bangalore,India
balajitn@ee.iisc.ernet.in

Francois Xavier Nsabimana
Project Group Hearing,Speech and Audio Technology,
Fraunhofer Institute of Digital Media Technology,
Oldenberg,Germany
nba@fraunhofer.idmt.de

K.R.Ramakrishnan
Dept. of Electrical Engineering,
Indian Institute of Science,
Bangalore,India
krr@ee.iisc.ernet.in

ABSTRACT

We propose an iterative algorithm to detect transient segments in audio signals. Short time Fourier transform (STFT) is used to detect rapid local changes in the audio signal. The algorithm has two steps that iteratively - (a) calculate a function of the STFT and (b) build a transient signal. A dynamic thresholding scheme is used to locate the potential positions of transients in the signal. The iterative procedure ensures that genuine transients are built up while the localised spectral noise are suppressed by using an energy criterion. The extracted transient signal is later compared to a ground truth dataset. The algorithm performed well on two databases. On the EBU-SQAM database of monophonic sounds, the algorithm achieved an F-measure of 90% while on our database of polyphonic audio an F-measure of 91% was achieved. This technique is being used as a preprocessing step for a tempo analysis algorithm and a TSR (Transients + Sines + Residue) decomposition scheme.

1. INTRODUCTION

Transients are portions of audio signals that evolve fast and unpredictably over a short time period [1]. Transients can be classified as attack transients (sound onsets), rapid decay transients (sound offsets), fast transitions (portamentos) and noise/chaotic regimes (sounds like handclaps, rain etc) [2]. Percussive sounds, guitar slaps, stop consonants (uttered during singing) are very good examples of transient signals. Transients generally last for 50ms and display fast changes in amplitude and phase at various frequencies. Transients can be classified as weak or strong based on the strength of the envelope while they can also be charaterized as fast or slow depending on the rate of change of envelope amplitude. Fast transients have sharp amplitude envelopes while slow transients have broad (platykurtic) envelopes. Transient detection is an important problem in many areas of music research like - audio coding (parametric audio coding [3], pre-echo reduction [4] etc), onset detection [5, 6], time-scaling of audio signals [2, 7, 8], note transcription [2], rhythm analysis and percussion transcription [9, 10].

One of the first attempts to detect and model transients was the TMS (Transient Modeling Synthesis) model proposed in [11] as an extension to the popular sinusoidal modeling of McAulay et al. [12] and sine + noise model [13]. The basic idea of the TMS model is the time-frequency duality. The TMS model is also dual to the sinusoidal modeling [12]. That is, by choosing a proper linear transform, a pure sinusoid in time domain appears impulsive in the frequency domain and an impulsive like signal in time domain looks sinusoidal in the frequency domain. Discrete Cosine Transform (DCT) was thus chosen to provide the mapping from the time domain to the frequency domain so that transients in the time domain become sinusoidal in the frequency domain. Energy of the original signal and its residue from signal modeling using DCT is used for transient detection. Masri et al. [5] used the high frequency content feature to detect attack transients for the purposes of audio analysis/synthesis. Abrupt phase changes in a bank of octave spaced filters has been employed to detect transients in [7]. Recently, group delay function has been used to detect transients in monophonic and pitched percussive instruments [14]. In [15, 16] linear prediction followed by thresholding on the residual signal envelope have been used for transient detection and modeling. Roebel used the center of gravity (COG) of a signal to locate transients and use it for onset detection with good results [6]. Torresani et al. [17] have used a concept of “transientness“ to detect transient signals. Two sets of basis functions that have sparse (dense) representations for pure sinusoids and dense (sparse) representations for transients simultaneously are chosen to define the transientness of audio signals. For a more exhaustive survey on transient detection we refer the reader to [18].

Most of the above discussed works use monophonic audio for their results. Daudet et al. [18] conducted a survey of various techniques and their efficiency of transient detection on the popular “glockenspiel” and “trumpet” audio signals. Gnann et al. [14] have used the EBU-SQAM database (monophonic signals) to test their algorithm and we use the same too.

In this paper, we propose to build on Ono et al. [19, 20] by using a much simpler iterative procedure. This algorithm can be used for audio coding, rhythm analysis and percussion transcription amongst the many possible tasks.
This paper is organised as follows. We describe our approach and choice of parameters in section 2. Section 3 presents our experimental setup, databases used and the results along with some advantages of our approach. We conclude in section 4.

2. THE TRANSIENT DETECTION ALGORITHM

We consider percussive sounds (drums, tom-toms etc), guitar slaps and sung consonants as transients. They show up as vertical lines in spectrograms [19] . Our algorithm detects such vertical lines in the spectrogram that have sufficient strength and bandwidth. We intend to detect reasonably fast percussive transients like piano hits, guitar slaps and the various drums while neglecting the slow transient signals like gongs.

Let $x[n]$ be a polyphonic audio signal. The signal is resampled at 16kHz to account for varied sampling rates and recording conditions (The algorithm works at any sampling rate but we choose 16kHz to standardize steps for our TSR algorithm). The signal is normalised such that its maximum value is 1 as follows,

$$x_{norm}[n] = \frac{x[n]}{\max(|x[n]|)}. \tag{1}$$

The normalization step is not necessary for audio coding applications. This signal is now split into frames of 40 ms with an overlap of 30ms. Each frame of the signal is multiplied with a Blackman-Harris window of length, $N = 640$ samples to reduce sidelobes. A STFT of the signal analyses the frequency content of the signal in regular periods. Let $X(i,k)$ denote the STFT of the signal for the i^{th} frame and k^{th} frequency bin. Then,

$$X(i,k) = \sum_{n=0}^{N-1} x[n].w[n-iR].e^{-\mathbf{j}.2\pi.n.k/N}, \tag{2}$$

where $w[n]$ is the windowing function, N is the number of samples in a window and R is the time shift in samples [21].

We now define functions T_- and T_+ that are derived from the magnitude spectrum of the signal as follows:

$$\begin{aligned} T_-(i,k) &= |X(i,k)| - |X(i-1,k)|, & (3)\\ T_+(i,k) &= |X(i,k)| - |X(i+1,k)|. & (4) \end{aligned}$$

The functions T_- and T_+ act as intermediate functions which detect vertical edges in the spectrogram. These derivatives indicate onsets and offsets respectively. Since transients have fast onsets followed by fast offsets, the T_- and T_+ functions should have high values at frames corresponding to transients. We now form a smoothened version of the above functions as follows;

$$\begin{aligned} F(i,j) &= 0.5\{\sum_{k=j-\nu}^{j+\nu} \{1 + sgn(T_-(i,k))\}.T_-(i,k) \\ &\quad +\{1 + sgn(T_+(i,k))\}.T_+(i,k)\}, \end{aligned} \tag{5}$$

where

$$sgn(\theta) = \begin{cases} 1 & \text{if } \theta \geq 0\ , \\ -1 & \text{if } \theta < 0. \end{cases} \tag{6}$$

$F(i,j)$ computes temporal changes in the magnitude spectrum at the frame i. $F(i,j)$ considers half wave rectified positive values of T_+ and T_- functions and adds it across frequency bins $j-\nu$ to $j+\nu$. The half wave rectification in equation (5) ensures that we detect only onsets from T_- and offsets from T_+ respectively. The parameter ν takes into account the spectral spread of the transient, neglecting noisy inflections in the spectrogram.

As can be seen in Figure 1, the function in red (dashes) is with smoothing along the vertical direction (vertical neighbours $\nu = 3$) and the function in blue (dashes and dots) is without smoothing ($\nu = 0$). The smoothing operation ensures that only genuine vertical edges in the spectrogram are accentuated and spurious changes (due to inflections in vocals/instrumentation) are suppressed.

2.1 Proposed iteration steps

For the extraction of transients, we now use X and F in an iterative framework as described below. The main algorithm consists of 3 iterative steps. In the first step, dynamic thresholds are computed.In the second step the transient signal updates are obtained. In the third step functions dependent on $X(i,k)$ are updated. We now use the F to detect the presence of transients in the audio signal.

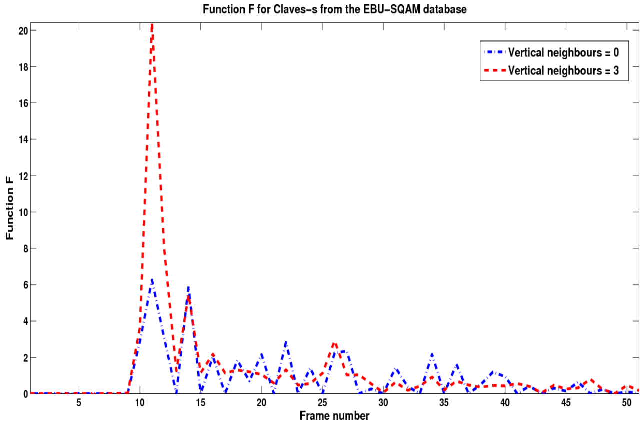

Figure 1. Function F at bin number 2kHz for Claves-s signal from EBU-SQAM database. The transient regions get accentuated more with vertical neighbours, $\nu = 3$ compared to $\nu = 0$, while the local ringing noise is suppressed.

2.1.1 Step I: Computing dynamic thresholds

An adaptive threshold for the detection function $F(i, j)$ is derived. Let $\lambda(i, j)$ represent the desired threshold. Then,

$$\lambda(i, j) = \beta \times \frac{\sum_{l=i-\tau}^{i+\tau} F(l, j)}{2\tau + 1}, \tag{7}$$

where β is a parameter to control the strength of transients that are to be extracted. Equation (7) calculates a time varying threshold for every time-frequency bin (i^{th} frame and j^{th} frequency bin). A flag is set if the value of F at the bin j is greater than the threshold $\lambda(i, j)$. That is,

$$\Gamma(i, j) = \begin{cases} 1 & \text{if } F(i, j) > \lambda(i, j) \ , \\ 0 & \text{if } F(i, j) \leq \lambda(i, j). \end{cases} \tag{8}$$

Summing the flag function Γ along the frequency bins (represented by Σ_Γ) indicates the number of frequency bins in a single frame that have more significant energy than their neighbours and may reveal the presence or absence of a transient. That is,

$$\Sigma_\Gamma(i) = \sum_{j=0}^{N-1} \Gamma(i, j). \tag{9}$$

2.1.2 Step II: Extraction of the transient portion and update of X

If the Σ_Γ is greater than a threshold λ_{Thr}, the corresponding frame is declared transient frame and a small fraction δ of the magnitude spectrum is subtracted from that frame and added to the function P to build transients as follows,

$$P(i, j) = \begin{cases} P(i, j) & \text{if } \Sigma_\Gamma < \lambda_{Thr} \ , \\ P(i, j) + \delta.X(i, j) & \text{if } \Sigma_\Gamma \geq \lambda_{Thr}, \end{cases} \tag{10}$$

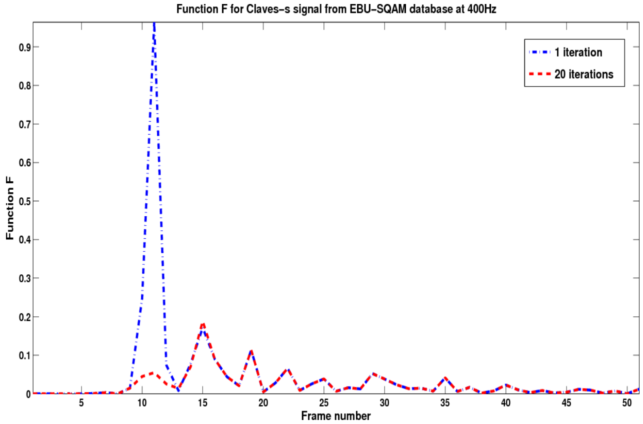

Figure 2. The function F - initial value and value after 20 iterations at 400Hz for Claves-s signal from EBU-SQAM database.

where j varies from 0 to $N - 1$.

In case of detected transients, the magnitude spectrum is modified as follows,

$$X(i, j) = \begin{cases} X(i, j) & \text{if } \Sigma_\Gamma < \lambda_{Thr} \ , \\ (1 - \delta).X(i, j) & \text{if } \Sigma_\Gamma \geq \lambda_{Thr}, \end{cases} \tag{11}$$

where j varies from 0 to $N - 1$.

2.1.3 Step III: Update of functions dependent on X

The functions F, λ, Γ and Σ_Γ are updated using the X obtained from 2.1.2.

We iterate over steps I, II and III for M times. Figure 2 shows the changes in F at a particular frequency bin after various iterations. As can be seen from Figure 2, F decreases at places of transients and increases in the adjacent frames. This is due to the definition of F, since it considers a contribution from T_- and T_+ only if they are positive. If after a particular iteration, say $T_-(i, j)$ becomes positive because $|X(i - 1, j)|$ reduced from the previous iteration (see update equations in Algorithm.1), then $F(i, j)$ can be greater than its value in the previous iteration.

The function P at the end of M iterations represents the spectrogram of the transient signal. The same steps are presented as follows. From now on all variables and functions used for the algorithm are superscribed with (n) (only if their values depend on the iteration) to represent the n^{th} iteration.

We begin by initialising P to 0. The values for the functions X, F, Γ, λ and Σ_Γ, calculated from the original signal, are used for the initial values of the algorithm.

We thus have two parameters that control both the strength of the extracted transient (controlled by β) and its spread in frequency (controlled by λ_{Thr}). We have used $\tau = 3$ and $\delta = 0.1$ in our implementation.

```
Input: Initialise $P^{(1)}$ to 0, $X^{(1)}$ to $X$, $F^{(1)}$ to $F$, $\lambda^{(1)}$
       to $\lambda$, $\Gamma^{(1)}$ to $\Gamma$, $\Sigma_\Gamma^{(1)}$ to $\Sigma_\Gamma$
Output: Transient signal P extracted from X
foreach $n$ = 1 to $M$ do
    (I, II) if $\Sigma_\Gamma^{(n)}(i) \geq \lambda_{Thr}$ then
        (i)$|X^{(n+1)}(i, 0:N-1)| =$
        $(1-\delta) \times |X^{(n)}(i, 0:N-1)|$
        (ii)$P^{(n+1)}(i, 0:N-1) =$
        $P^{(n)}(i, 0:N-1) + \delta \times |X^{(n)}(i, 0:N-1)|$
        else
        (i)$|X^{(n+1)}(i, 0:N-1)| = |X^{(n)}(i, 0:N-1)|$
        (ii)$P^{(n+1)}(i, 0:N-1) = P^{(n)}(i, 0:N-1)$
    end
    (III) Calculate $F^{(n+1)}$, $\lambda^{(n+1)}$, $\Gamma^{(n+1)}$, and
    $\Sigma_\Gamma^{(n+1)}$ using $X^{(n+1)}$
end
```

Algorithm 1: Flow for updating equations of the algorithm

The iterative procedure is used instead of a single step transient detection since this algorithm is a part of a TSR decomposition algorithm we are currently developing. The iterative procedure helps to uncover slightly masked and weak transients at later steps as has been revealed in our preliminary experiments. The TSR decomposition algorithm works by alternatively identifying and extracting transients and sinusoids until we are left with a residue signal. Even without the sinusoidal extraction steps here, the algorithm does detect partially masked and low energy transients. As can be seen from Figure 3, the transient signal builds slowly over iterations.

After the M steps, the transient signal that has been generated is converted back into time domain by an inverse Discrete Fourier transform (IDFT). The phase of the original signal is used for the IDFT procedure. Frames of the transient signal that have less than 5% of the maximum energy are discarded to retain only significant transients. The locations of the transients are compared with the ground truth data for evaluation.

Figure 4 shows the glockenspiel signal from EBU-SQAM database and its extracted transient. As can be seen, the transient signal is well extracted.

3. EXPERIMENTS AND EVALUATIONS

A database of 33 clips averaging 10 seconds each was prepared by selecting audio from various possible genres (pop, rock, R&B etc). Each clip was converted to '.wav' format from CDs, and resampled at 16kHz. Each clip was manually annotated for percussive transients after multiple listening, using the gating procedure [22]. Two people annotated the database independently. The common transient segments from both the annotators were chosen for our final ground truth set. The database has a total of 1308 transient segments. The database was split into 2 non-overlapping sets consisting of 10 clips for the training dataset having 406 transient segments and 23 clips for the test dataset with 902 transients respectively [1] .

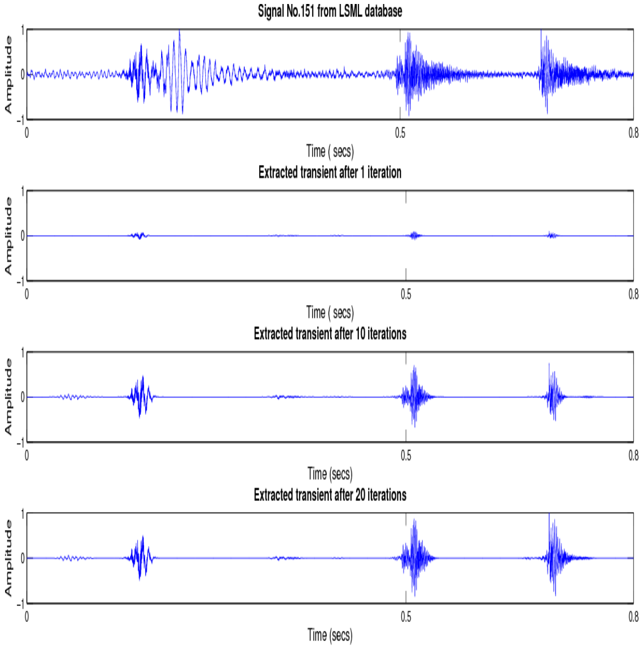

Figure 3. The original signal and the extracted transient signal after 1, 5 and 20 iterations. The strength and time duration of the transient signal increases with iterations

The parameters β and λ_{Thr} were optimised using the training set consisting of 10 clips. β was varied in steps of 0.1 from 1.25 to 2.5 while λ_{Thr} was varied as a fraction of frame length N (i.e $N/10$,$N/9$,$N/8$...). A transient was declared to have been found if the extracted transient overlapped with the ground truth segment. We got optimal performance for $\beta = 2$ and $\lambda_{Thr} = N/6$. λ_{Thr} parameter selects only significantly long vertical lines in the spectrogram while β parameter evaluates the strength of the transients. We used $M = 20$ iterations for the algorithm. This way if a transient exists, approximately 90% of the magnitude can be extracted in the iterative steps if the Σ_Γ satisfies the threshold conditions for all the 20 iterations.

These parameters were used to test the remaining 23 songs for their performance. The algorithm was able to correctly detect (CD) 808 (90%) transients with 65 (7%) false posi-

[1] This is denoted as the LSML database. We intend to make this a freely available database for research soon

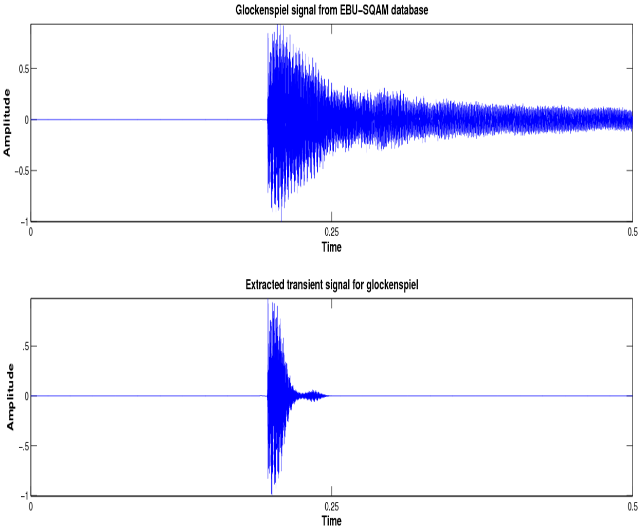

Figure 4. Glockenspiel signal from EBU-SQAM database and the extracted transient signal

Database Name	Total	CD	FP	FN	DD
LSML	902	808	65	94	2
EBU-SQAM	276	237	11	39	0

Table 1. Performance of the transient extraction algorithm

tives (FP). This is equivalent to a Precision (P) of 0.92 and Recall (R) of 0.89. The F measure is thus 0.91.

We have also tested our algorithm on the EBU-SQAM percussive monophonic database [14]. The testing procedure followed in [14] was used for testing our algorithm on this database. The EBU-SQAM database has 276 percussive transients in 24 files. Our algorithm correctly detected 237 transients correctly while 11 transients were detected as FP. This gives our algorithm an F-measure of 0.90. This compares very well with the results from [14], where an F-measure of 0.92 is achieved on the same dataset of EBU-SQAM database. While the parameters in [14] are optimized for the EBU-SQAM database, we use the same parameters that are optimized for our LSML polyphonic music database.

For the EBU-SQAM database we observed that we got false positives during the slow transient portions of the signal or for signals with heavy ringing in the decay tail. Also, a shortcoming that we observed with our algorithm was the sensitivity to signal continuity. The EBU-SQAM database has signal discontinuity in 2 files and those portions were detected as transients.

The performance of our algorithm is tabulated in Table 1. Since the algorithm acts as a pre-processing stage for a tempo analysis algorithm and a TSR decomposition algorithm, the false positives do not harm much except when they have sufficient energies. Figure 5 shows the audio signal and extracted transient locations in comparison with the groundtruth locations for a polyphonic piece from LSML database.

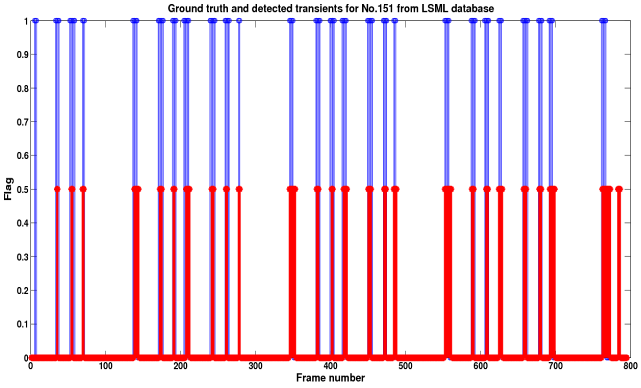

Figure 5. The locations of the extracted transients are shown w.r.t the ground truth.The blue lines indicate the extracted transient locations and the red lines the ground truth

Parameter	Numeric value
Frame size	640
ν	3
δ	0.1
τ	3
β	2
M	20
F_{thr}	106

Table 2. Used parameters and their values

The values of the parameters used by us in our implementation is given in Table 2.

4. CONCLUSIONS AND FUTURE WORK

We have discussed a simple iterative procedure for detecting transients from polyphonic audio signals. The method is used in a TSR decomposition algorithm. This algorithm is also currently acting as a pre-processing step for a tempo analysis algorithm. We are also looking at using generalised TEF (Teager energy functions) type of functions to improve our transient detection accuracy.

5. REFERENCES

[1] J. P. Bello and L. Daudet and S. Abdallah and C. Duxbury and M. Davies and M. B. Sandler: ”A Tutorial on Onset Detection in Music Signals”, *IEEE Transactions on Speech and Audio Processing,*, Vol-13,No.5,September,2005.

[2] H. Thornburg: ”Detection and modeling of transient audio signals with prior information”, *PhD Thesis, Stanford University*,2005.

[3] B. Edler and H. Purnhagen: ”Parametric Audio Coding”, *International Conference on Signal Processing (ICSP)*,2000.

[4] R. Vafin and R. Heusdens and S. van de Par and W. B. Kleijn: ”Improved modeling of audio signals by modifying transient locations ”, *Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA01))*,2001.

[5] P. Masri and A. Bateman: ”Improved Modelling of Attack Transients in Music Analysis-Resynthesis”, *Proc. of the Intl.Computer Music Conference(ICMC)*,pg.100-103,1996.

[6] A. Roebel: ”Onset detection by means of transient peak classification”, *http://www.music-ir.org/mirex/abstracts/2010/AR4.pdf*, MIREX-2010.

[7] C. Duxbury and M. Davies and M. Sandler: ”Separation of transient information in musical audio using multiresolution analysis techniques”, *Proc. of the Conference on Digital Audio Effect(DAFx))*,2001.

[8] F. X. Nsabimana and U. Zolzer: ”Audio Signal Decomposition for Pitch and Time Scaling”, *International Symposium on Communications, Control Signal Processing (ISCCSP)*,2008.

[9] J. Sillanpaa and A. Klapuri and J. Seppanen and T. Virtanen: ”Recognition of acoustic noise mixtures by combined bottom-up and top-down processing”, *Proceedings of the European Signal Processing Conference(EUSIPCO)*,2000.

[10] J. Uscher: ”Extraction and removal of percussive sounds from musical recordings”, *Intl Conference on Digital Audio Effects(DAFx)*,2006.

[11] T. S. Verma and S. N. Levine and T. H. Y. Meng: ”Transient modeling synthesis:a flexible analysis/synthesis tool for transient signals”, *Intl Computer Music Conference (ICMC)*,1997.

[12] R. McAulay and T. F. Quatieri: ”Speech Analysis/Synthesis based on a sinusoidal representation”, *IEEE Transactions on Acoustics,Speech and Signal Processing*,Vol 34,pp-744-754,1990.

[13] X. Serra and J. O. Smith: ”Spectral Modeling Synthesis:A sound analysis/synthesis system based on a deterministic plus stochastic decomposition”, *Computer Music Journal*,Vol 14(4),pp-14-24,1990.

[14] V. Gnann and M. Spiertz: ”Transient detection with absolute discrete group delay,”, *IEEE International Workshop on Intelligent Signal Processing and Communication Systems (ISPACS)*,2009.

[15] F. X. Nsabimana and U. Zolzer: ”A transients/sinsoids/residual approach for audio signal decomposition”, *Proc of DAGA’08*,2008.

[16] F. X. Nsabimana and U. Zolzer: ”Transient encoding of audio signals using dyadic approximations”, *Proc of 10th Intl Conference on Digital Audio Effects*,2007.

[17] S. Molla and B. Torresani: ”Determining local transientness in audio signals”, *IEEE Signal Processing Letters*,Vol-11(7),pp. 625-628,2004.

[18] L. Daudet: ”A review of techniques for the extraction of transients in musical signals”, *Proc. of the Computer Music Modeling and Retrieval(CMMR)*,2005.

[19] N. Ono and K. Miyamoto and J. Le Roux and H. Kameoka and S. Sagayama: ”Separation of a monaural audio signal into harmonic/percussive components by complementary diffusion on spectrogram”, *European signal Processing Conference(EUSIPCO))*,2008.

[20] N. Ono and K. Miyamoto and H. Kameoka and S. Sagayama: ”A real-time equalizer of harmonic and percussive components in music signals”, *Intl Society of Music Information Retrieval Conference)*,2008.

[21] Lawrence Rabiner and Ronald Schafer: ”Chap 6,Pages:251-252, Digital Processing of Speech Signals”, *Pearson Education(Indian Edition)*,1993.

[22] D. J. Hermes: ”Vowel Onset Detection”, *Journal of Acoustical. Society. of America*,Vol-87(2), 866-873,1990.