Granular Analysis/Synthesis for Simple and Robust Transformations of Complex Sounds

Jung-Suk Lee[1,2,3], François Thibault[4,*], Philippe Depalle[1,2], Gary P. Scavone[1,2]

[1]*Music Technology Area, Schulich School of Music, McGill University, Montréal, Québec, H3A 1E3, Canada*

[2]*Centre for Interdisciplinary Research in Music Media and Technology (CIRMMT), Montréal, Québec, H3A 1E3, Canada*

[3]*Broadcom Corporation, Matawan, New Jersey, 07747, U.S.A.*

[4]*Audiokinetic Inc., Montréal, Québec, H2Y 2P4, Canada*

Correspondence should be addressed to Jung-Suk Lee (jungsuk.lee@mail.mcgill.ca)

ABSTRACT

In this paper, a novel and user-friendly granular analysis/synthesis system particularly geared towards environmental sounds is presented. A granular analysis component and a grain synthesis component were intended to be implemented separately so as to achieve more flexibility. The grain analysis component segments a given sound into many 'grains' that are believed to be microscopic units that define an overall sound. A grain is likely to account for a local sound event generated from a microscopic interaction between objects. Segmentation should be able to successfully isolate these local sound events in a physically or perceptually meaningful way. The second part of the research was focused on the granular synthesis that can easily modify and re-create a given sound. The granular synthesis system would feature flexible time modification with which the user could re-assign the timing of grains and adjust the time-scale. Also, the system would be capable of cross-synthesis given the target sound and the collection of grains obtained through an analysis of sounds that might not include grains from the target one.

1. INTRODUCTION

Nowadays, audio rendering in virtual reality applications, especially games, requires higher standards to meet the users' demands. Conventional ways of sound generation in games, mostly playing pre-recorded samples, are often limited in their lack of ability to deal with variations in sounds, for interactions between objects in games occur in various ways. This problem demands model-based sound synthesis techniques capable of generating many sound instance variations without having to use additional sound samples. Sounds that appear in games are in general non-musical/verbal, often referred to as 'environmental' or 'everyday' sounds. Such sounds are generated mostly either from interactions between objects or environmental background that is given in the virtual space, including bouncing, breaking, scratching, rolling, streaming, etc. It is very important to maintain the quality of such sounds for a feeling of reality. In general, every synthesis technique has its own strength and it differs according to the types of sounds. Therefore it is crucial to choose a synthesis technique appropriately, given the sounds to be dealt with. The granular analysis/synthesis technique is regarded as one of the promising methods to deal with sounds in games since the technique can easily preserve complex sound textures and create variations of the given sound by mosaicking grains. Thus, it would be helpful to develop a novel granular analysis-based synthesis framework that could be used easily by non-signal processing experts to allow parametric synthesis controls to generate many variations of complex sounds. This framework could benefit from granular synthesis to fill the gap between information contained locally in the waveform (specific to a grain) and global information about the sound production process, such as resonance frequencies. It could also use information derived from an understanding of physical processes to control the density and time-distribution of sound grains.

Since Curtis Roads implemented granular synthesis using digital computers [1] for the first time, there have

[*]Now with Nuance Communications Inc., Montréal, Québec, H3A 3S7, Canada

been many works done on granular analysis/synthesis for various applications. Picard *et al.*[2] used a dictionary of short sound with respect to a given target sound by selecting best matched sound segments in a dictionary with time modification. Granular analysis/synthesis environments, provided with GUI have also been developed by several researchers, as found in [3], [4], [5], [6]. They serve as tools that enable users to intuitively and interactively synthesize sounds for various applications using the granular analysis/synthesis framework.

The goal of the research presented in this paper is to flexibly synthesize complex sounds within the framework of granular analysis/synthesis. To this end, not only existing granular analysis/synthesis techniques are enhanced and customized, but also novel features are introduced, culminating in a user-friendly granular analysis/synthesis scheme. In the analysis stage, a measure referred to as the 'stationarity' is proposed to categorize a complex sound into the region where distinctive micro sound events can be identified (non-stationary region) and the region where the boundaries of the micro sound events are too ambiguous to be distinguished from each other (stationary region). This is done to adjust parameters associated with granular analysis so as to achieve more promising granular synthesis. In the synthesis stage, to enable flexible synthesis of complex sounds aiming at various kinds of interaction scenarios, time modification allows for seamless time stretching and shrinking with the aid of proposed gap-filling algorithms and also for grain time remapping by re-ordering the locations of grains, which leads to modifying given complex sounds in a physically meaningful way.

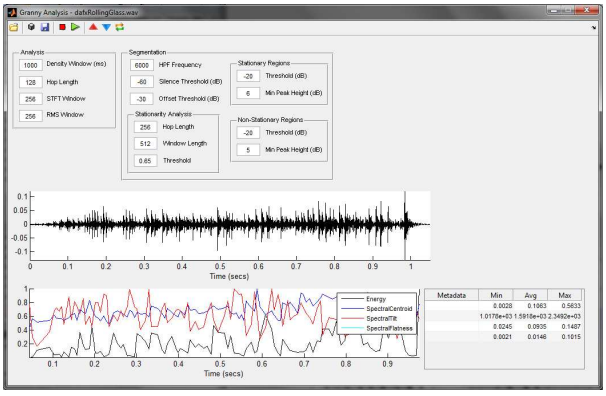

Fig. 1: GUI for granular analysis.

Granular analysis parameters		
Spectral Flux Analysis	Density window Hop length STFT window RMS window	
Segmentation	HPF frequency Silence threshold Offset threshold	
	Stationary Analysis	Hop length Window length Threshold
	Stationary regions	Threshold Min peak height
	Non-stationary regions	Threshold Min peak height

Table 1: Granular analysis parameters.

2. GRANULAR ANALYSIS SYSTEM

The granular analysis system involves the decomposition of a sound into short snippets, termed as the 'grains', on the assumption that a sound is generated from numerous micro-interactions between physical objects. In the proposed granular analysis system, analysis is based on a process similar to onset/transient detection for segmenting a sound composed of grains that have percussive/impulsive characteristics. The grain analysis system is implemented in MATLAB with a GUI where one can set all the parameters (Fig. 1). The parameters used for analysis are listed in the Table.1.

2.1. Grain Analysis

In order to perform the task of granular analysis, it is essential to transform an audio signal into a form that reveals and emphasizes the transients in the audio signal, referred to as the *detection function* [7]. Among many detection functions that have been devised so far, no dominant detection function that outperforms other detection functions exists, so a detection function is chosen and used depending on the nature of the given audio signal and the purpose of the analysis. We have chosen to use a detection function that measures differences in the spectral content of transient and non-transient parts of the signal.

The well-known short-time Fourier transform (STFT) [8] is used for frame-by-frame analysis, enabling comparison of spectral content between neighboring short portions of the signal sequentially. The STFT of $x(n)$ is

given as

$$X_k(n) = \sum_{m=0}^{N-1} x(hn+m)w(m)e^{-2j\pi mk/N}, \quad n = 0,1,2,\cdots \tag{1}$$

where h ('Hop Length' in Fig. 1 and Table.1) is the hop length and w is the window of length N ('STFT Window' in Fig. 1 and Table.1). $X_k(n)$ is the kth discrete Fourier transform (DFT) coefficient of the nth frame. In order to detect transients, the Spectral Flux (SF) was chosen, which is defined as below [7],

$$S(n) = \sum_{k=0}^{N-1} \{H(|X_k(n)| - |X_k(n-1)|)\}^2 \tag{2}$$

where $S(n)$ is the value of the SF at the nth frame. $H(x) = \frac{(x+|x|)}{2}$ is a half-wave rectifier employed to put an emphasis on only the positive changes. The SF first measures the square of the Euclidian distance between magnitude spectra of successive frames and takes into account only the energy increases in frequencies. Depending on the nature of the signal to be analyzed, high-pass filtering can be conducted to reveal transients more vividly [9]. The parameter referred to as the 'HPF frequency' (Fig. 1 and Table.1) actually defines the cut-off frequency of the high-pass filter.

2.2. Grain Segmentation

2.2.1. Peak Detection

Since a typical impact sound begins with a transient, broadband energy and then continues with decaying resonances, peaks in the SF are likely to appear around the beginning point of a local impulsive event. A peak in the SF, $S(n_{peak})$, is defined as

$$S(n_{peak}-1) \le S(n_{peak}) \ge S(n_{peak}+1) \tag{3}$$

where n_{peak} is a sample index on which the peak is located. A valley in the SF, $S(n_{valley})$, is defined as,

$$S(n_{valley}) \le S(n_{valley}-1) \quad \text{and} \quad S(n_{valley}) \le S(n_{valley}+1) \tag{4}$$

where n_{valley} is a sample index on which the valley is located. Prior to conducting peak selection, noise components in the SF that may be confused as meaningful peaks are first discarded by partitioning the overall signal into silent/non-silent regions. In order to do this, we propose a parameter referred to as the 'silent threshold' (given in the 'Segmentation' category as in Fig. 1, Table.1) and calculate the frame-based short time root mean square (RMS) of the overall signal as

$$RMS(n) = \sqrt{\frac{1}{N_{rms}} \sum_{m=0}^{N_{rms}-1} |x(h_{rms}\cdot n+m)|^2} \tag{5}$$

where N_{rms}, h_{rms} is the length of the frame and the hop length used, respectively (the 'RMS window' and the 'Hop Length' in the 'Analysis' category as in Fig. 1 and Table.1). Regions where $RMS(n)$ are smaller than the silent threshold could be labeled silent regions and peaks only in the non-silent regions are considered to reduce the chances of including unnecessary noise components. Also, a parameter called the 'Peak Threshold Height' (given as the 'Threshold' in the categories of 'Non-Stationary Regions' and 'Stationary Regions' (Fig. 1 and Table.1) is defined to ignore peaks whose heights appear to be too small, depending on the nature of the given signal. By using the peak detection method proposed in [10], peaks that satisfy a certain condition are picked to determine the grain segmentation. That condition is associated with a ratio γ in such a way that:

$$\gamma < \frac{S(n_{peak})}{(S(n^l_{valley}) + S(n^r_{valley}))/2} \tag{6}$$

where $S(n_{peak})$ is the SF value at the peak location and $S(n^l_{valley})$ and $S(n^r_{valley})$ are the SF values at the neighboring valleys to the left and right sides of the peak $S(n_{peak})$. The ratio γ is referred to as the 'Minimum Peak Height' (given as 'Min Peak Height' in the categories of 'Non-Stationary Regions' and the 'Stationary Regions' (Fig. 1 and Table.1)). Only when the ratio of the peak and the valleys is larger than the minimum peak height, is $S(n_{peak})$ chosen as the grain segmentation boundary. A grain segmentation boundary is set in such a way that the beginning point of a grain is set at $n_{peak} - \frac{h}{2}$, a sample index ahead of the peak location by half of the hop length, to consider rising time in the attack phase of an impulsive event.

2.2.2. Stationarity Analysis

A 'stationary sound' is regarded as a sound that would convey more consistent and regular impressions to listeners, e.g. that of a gentle brook, in terms of texture. The stationary sounds usually consist of numerous sound events of very short durations heavily blended with each

other so that an individual sound event is scarcely identifiable in the overall sound, thus the previously introduced grain segmentation method would tend not to result in meaningful sound events. Therefore it is desirable to be able to adjust criteria for grain segmentation according to the 'stationarity' of a given sound. To achieve this, we first propose a measure to detect which part of the signal is stationary or non-stationary. Here we assume that the stationarity is closely related to how a signal looks in the time domain, in such a way that a stationary part would look statistically flat while a non-stationary signal would look rather 'bumpy'. The measure proposed is referred to as the 'stationarity measure' and is defined as,

$$sm(n) = \frac{\sqrt[N_{sm}]{\prod_{m=0}^{N_{sm}-1} |x(h_{sm} \cdot n + m)|}}{\frac{1}{N_{sm}} \sum_{m=0}^{N_{sm}-1} |x(h_{sm} \cdot n + m)|} \tag{7}$$

where h_{sm} is the hop length and N_{sm} (given as the 'Hop Length' and the 'Window Length' in the 'Stationary analysis' category in Fig. 1 and Table.1) is the frame size. The numerator and the denominator are the geometric mean and the arithmetic mean of the absolute values of the samples contained in the nth frame. This can be viewed as the time domain version of the 'spectral flatness measure' [11]. The spectral flatness measure corresponds to the extent that a signal is bumpy in the time domain, as indicated by $sm(n)$. Once stationary/non-stationary parts are partitioned, we can individually set the parameters associated with peak detection, so that there are tighter conditions for non-stationary parts and more relaxed ones for stationary parts, as separated into the 'Non-stationary' and the 'Stationary' categories in Fig. 1 and Table.1. Fig. 2 shows an example of partitioning a signal into stationary/non-stationary regions on the basis of the stationarity measure. The signal in the top pane consists of two types of applause sounds. The one on the left of the blue dashed vertical line in the middle is applause by a large number of people, while the one to the right of the blue dashed vertical line is by a small number of people. The middle pane shows the $sm(n)$, and it is obvious that the applause by the large audience has a higher and consistent $sm(n)$, whereas the one by the small audience has a lower and varying $sm(n)$. The bottom pane shows that, on the basis of the stationarity measure, the parameters for peak detection can be separately set and yield different grain segmentation results accordingly.

2.3. Meta Data

For further applications, such as feature matching-based synthesis [4], meta data associated with a grain $g_k(n)$ (kth grain in the dictionary), features widely used in music information retrieval (MIR) and psychoacoustics research, are extracted as auxiliary information. Selected features that constitute meta data are the following (they are all normalized between 0 to 1).

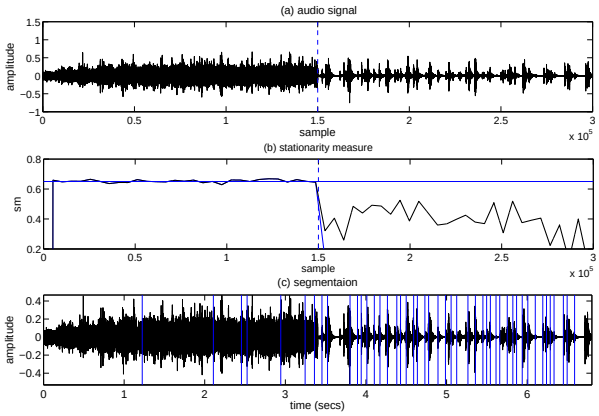

Fig. 2: Comparison of stationarity measure depending on the nature of a signal. (a) Original signal. The signal consists of two types of applause sounds. The one on the left of the blue dashed vertical line in the middle is applause by a large audience, while the one to the right of the blue dashed vertical line is by a small audience. (b) Stationarity measure of the signal in (a). The blue horizontal line is the silent threshold, set as 0.65. The hop length and the window length used are 6144 and 1024, respectively. (c) The result of grain segmentation with respect to two different sets of parameters. For the stationary part, the left side, the peak height threshold is -45dB and the minimum peak height is 11dB, and those for the non-stationary part, the right side, are respectively -25dB, 3dB.

- *Energy*

$$en(k) = \sum_{m=0}^{l_k} |g_k(m)|^2 \tag{8}$$

l_k : kth grain's length

- *Spectral Centroid*

$$sc(k) = \frac{\sum_{m=0}^{N-1} m|G_k(m)|}{\sum_{m=0}^{N-1} |G_k(m)|} \tag{9}$$

$G_k(m)$: mth DFT coefficient of g_k

N : DFT length

- *Spectral Tilt*

$$st(k) = \frac{N\sum_{m=0}^{N-1} m|G_k(m)| - \sum_{m=0}^{N-1} m \cdot \sum_{m'=0}^{N-1} |G_k(m')|}{N\sum_{m=0}^{N-1} m^2 - \left(\sum_{m=0}^{N-1} m\right)^2} \tag{10}$$

- *Spectral Flatness*

$$sfl(k) = \frac{\sqrt[N]{\prod_{m=0}^{N-1} |G_k(m)|}}{\frac{1}{N}\sum_{m=0}^{N-1} |G_k(m)|} \tag{11}$$

2.4. Grain Dictionary

All segmented grains are separately stored in a grain dictionary together with the side information. In many cases, a grain has a long tail with small amplitude. We can set the parameter referred to as the 'Offset threshold' (Fig. 1 and Table.1), which defines the amplitude threshold below which the tail is discarded, to efficiently compress the size of the grain wave data. In the grain dictionary, an element that represents a grain contains the following: *grain wave data, the starting point and the end point in the original signal, meta data, sampling rate.* As will be explained later, the starting point and the end point data enable time modification at the synthesis stage.

3. GRANULAR SYNTHESIS

With the granular synthesis system we have developed, the user can flexibly manipulate the temporal aspects of a sound dictionary. With a sound dictionary given, a user can perform time-scaling (stretching/shrinking) and can shuffle grains at will. To this end, algorithms that can fill gaps which inevitably arise when grain timings are modified have been devised. As shown in Fig. 3, the grain synthesis system consists of three components. In the grain dictionary component, the user can load target and corpus grain dictionaries from which grains are selected for the synthesis. Once the grain dictionary is loaded, the user can manipulate the temporal length of the given sound by stretching or shrinking intervals between grains by adjusting parameters that belong to the time stretching component. The component of time scrub allows for more flexible time modification in conjunction with time stretching, by enabling users to scrub the original order of the grain sequence in the dictionary.

3.1. Grain Dictionaries: Target and Corpus

The granular synthesis system requires two types of dictionaries for synthesis. One is the target dictionary and the other is the corpus dictionary. In our granular analysis/synthesis system, the target dictionary provides the time position information of grains as a target reference. Let $i(k)$ denote the starting time sample index of the kth grain, g_k, in the target dictionary. The initial operation of the granular synthesis system is to shift the time positions of grains in the corpus dictionary, making the starting time positions of the kth grain in the corpus dictionary become $i(k)$. With the time modification processes that will be explained below, synthesis with respect to the target sound is achieved in flexible ways.

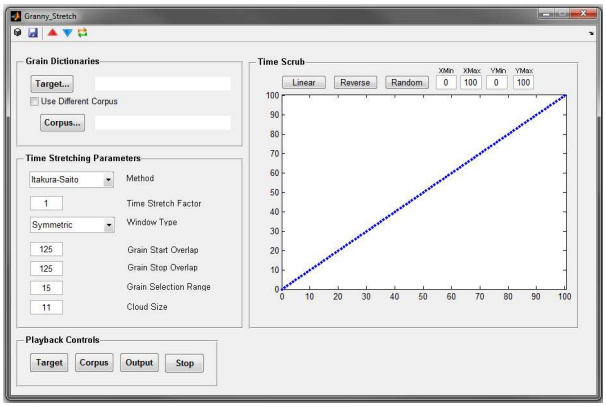

Fig. 3: GUI for synthesis.

Time Stretching Parameters	
Method	Grain Extrapolation Itakura-Saito MFCC
Time Stretch Factor	α (> 0)
Window Type	Symmetric One-sided
Grain Start Overlap	Number of samples overlapped
Grain Stop Overlap	Number of samples overlapped
Grain Selection range	Number of grains
Cloud Size	Number of grains

Table 2: Time stretching parameters.

3.2. Time Stretching/Shrinking

One of the main time modification schemes used in the granular synthesis system is time stretching and shrinking. Once grains in the given corpus are rearranged with respect to $i(k)$, the time modification, either stretching or shrinking, is conducted by controlling intervals between

$i(k)$. The time stretch factor α ($\alpha > 0$) controls the time modification. The new sample index of the starting time of grains in the corpus dictionary after time modification is given as

$$\begin{aligned} i'(k) &= \text{round}(\alpha \cdot i(k)) \\ 0 < \alpha < 1 &: \text{shrinking} \\ \alpha > 1 &: \text{stretching} \end{aligned} \tag{12}$$

where 'round' denotes the rounding operation. Time modification inevitably gives rise to unnecessary gaps between grains, as in

$$i'(k+1) - i'(k) > l_k \tag{13}$$

where l_k is the length of the kth grain in the corpus dictionary. Not only time stretching but also time shrinking could possibly create gaps since the lengths of the grains in the target dictionary are not the same as those in the corpus dictionary. Fig. 4 shows how time modification creates gaps. These gaps give rise to audible artifacts associated with signal discontinuities. It is essential to devise a way to fill gaps to remove the audible artifacts. In the present granular synthesis system, two different approaches for gap filling are proposed.

3.3. Gap Filling Strategies

3.3.1. Gap Filling with Grain Extension Method

One way to fill a gap is to extend the grain placed right ahead of the gap. The idea of grain extension is inspired by the audio signal interpolation/extrapolation that have been studied and developed for application to signal restoration for disturbed and missing data [12] [13] [14]. The grain extension algorithm used here is based on the Linear prediction (LP), using samples at the end of the grain to be extended as initial data for the LP. In [12], an audio extrapolation technique based on the Burg algorithm-based LP is proposed, which has been adopted for this grain extension. Estimated LP coefficients allow for extrapolation of a grain in such a way that past samples are filtered with the FIR filter whose coefficients are the LP coefficients. In order first to estimate the LP coefficients to use for grain extension, we begin with a linear prediction of the last sample of a grain $g(n)$ of length L,

$$\hat{g}(L) = \sum_{m=1}^{p} a_m g(L-m) \tag{14}$$

where a_m are the LP coefficients estimated using the last p samples of g, and $\hat{g}(L)$ is the estimate of the last sample of the grain $g(L)$. p is the LP order. Given $g(L)$, a_m

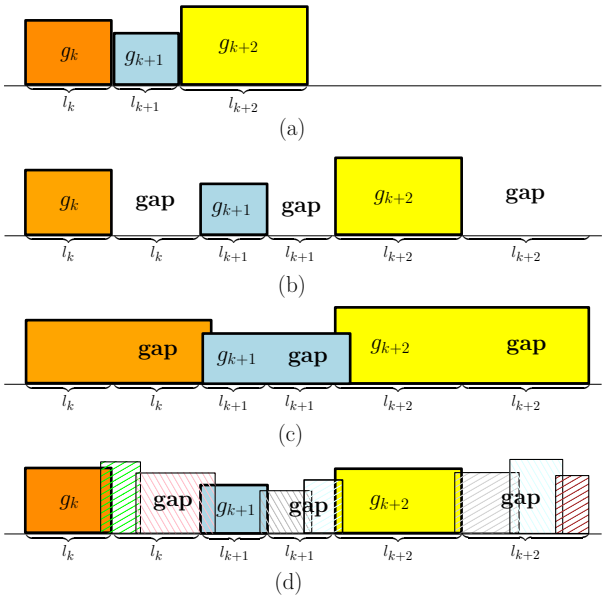

Fig. 4: Time stretching and gap filling. (a) original sequence of grains g_k, g_{k+1}, g_{k+2} of length l_k, l_{k+1}, l_{k+2}, respectively. (b) time stretched with the time stretch factor $\alpha = 2$. (c) Gap filling with grain extension (d) Gap filling with additional grains.

are estimated using Burg's method. Once the LP coefficients are estimated, they are used to extrapolate $g(n)$ by predicting the future samples. The first extrapolated sample, $\hat{g}(L+1)$ is obtained as

$$\hat{g}(L+1) = \sum_{m=1}^{p} a_m g(L-m+1) = \mathbf{A}\mathbf{g_1} \tag{15}$$

where $\mathbf{A}$ and $\mathbf{g_1}$ are

$$\mathbf{A} = [a_1 \; a_2 \; \cdots \; a_{p-1} \; a_p] \tag{16}$$

$$\mathbf{g_1} = [g(L) \; g(L-1) \; \cdots \; g(L-p+1)]^T \tag{17}$$

In the same way, we can proceed to produce further samples just by updating $\mathbf{g_1}$ with newly extrapolated samples. For example, in order to produce $\hat{g}(L+r)$, which is calculated as $\hat{g}(L+r) = \mathbf{A}\mathbf{g_r}$, $\mathbf{g_r}$ should be given as

$$\mathbf{g_r} = \begin{cases} [\hat{g}(L+r-1) \; \cdots \; \hat{g}(L+r-p)]^T & \text{if } r > p \\ [\underbrace{\hat{g}(L+r-1) \cdots \hat{g}(L+1)}_{r-1} \cdots \\ \underbrace{g(L) \; \cdots \; g(L-p+r)}_{p-(r-1)}]^T & \text{otherwise.} \end{cases} \tag{18}$$

The number of samples to be extrapolated is determined by the sum of the length of the gap and the parameter 'Grain Stop Overlap' (Table.2), which specifies how many samples are overlapped between neighboring grains. The LP order p can be arbitrarily chosen as long as $L > p$. In general, the more samples used to estimate the LP coefficients, the more accurate the estimate of the LP coefficients becomes [15].

3.3.2. Gap Filling with Additional Grain-Based Method

Since the gap part would likely be similar to the parts where grains are present, natural gap filling could be achieved by placing the most similar grains in the parts where there are grains in the gap, until the gap is completely filled. The optimal grains for the gap are determined on the basis of how similar they are to the grain placed ahead of the gap. Rather than extrapolating existing grains to fill gaps, these optimal additional grains are chosen from the grain dictionary and placed in the gap. This strategy gives a different kind of audible sensation to the listeners. In order to preserve the natural perception when filling gaps in this way, it is essential to choose grains appropriately. To keep the feeling of continuity with neighboring grains, additional grains that are to be filled into gaps are selected according to the similarity to the existing grain.

As the measures for representing the similarity, we use two features that are based on the spectral distance. One is the Itakura-Saito (IS) distance [16]. The IS distance is a measure of the perceptual difference between two spectra, defined as follows,

$$D_{IS}(k,k') = \frac{1}{2\pi}\left[\int_{-\pi}^{\pi} \frac{P_k(\omega)}{P_{k'}(\omega)} - log\frac{P_k(\omega)}{P_{k'}(\omega)} - 1\right] d\omega \quad (19)$$

where $P_k(\omega), P_{k'}(\omega)$ are the two spectra to be compared. The other is the distance based on Mel Frequency Cepstral Coefficients (MFCC). The MFCC are a perceptually based spectral feature widely used in speech recognition and music information retrieval [17]. The MFCC distance between the two grains is given as

$$D_{MFCC}(k,k') = \frac{\mathbf{C}_k \cdot \mathbf{C}_{k'}}{|\mathbf{C}_k||\mathbf{C}_{k'}|} \quad (20)$$

where $\mathbf{C}_k$ and $\mathbf{C}_{k'}$ are kth and k'th grains' MFCC vectors.

3.3.3. Grain Selection Range and Cloud Size

The simplest way to select an additional grain would be to select the grain that has the smallest measurable distance from the preceding grain ahead of the gap. In principle, the methods based on additional grains are supposed to compare the target grain, the existing one already given ahead of the gap, with all the remaining grains in the corpus dictionary. This often requires heavy computation when the size of the dictionary is huge. In particular, if the target sound is relatively homogeneous, then searching through the entire grain dictionary would be excessive in the extreme as it would be highly likely that all the grains in the dictionary are spectrally similar. In order to let users adjust the tradeoff between the computation load and the extent to which the target grain and the chosen grain are similar to each other, another parameter, referred to as the 'Grain Selection Range' (Fig. 3, Table.2), is proposed. The grain selection range determines a pool of grains in which a search for an additional grain is conducted. If the grain selection range is given n_{gsr} and the kth grain is the target grain, the candidate grains are defined by their orders in the dictionary, k', as

$$\{k - n_{gsr} \leq k' \leq k + n_{gsr},\ k' \neq k\} \quad (21)$$

As it is highly likely that the same grain would be repeatedly chosen, especially when the grains in the grain selection range are alike in terms of spectral content, audible artifacts could often be found in the resulting synthesized sound. To prevent this, instead of selecting the very best matched grain, an additional grain is randomly chosen from among a pool of best-matched grains. The number of grains in this pool is referred to as the 'Cloud Size' (Fig. 3, Table.2). The larger the cloud size, the more random the selection. If the cloud size is set to 1, then the best-matched grain is selected. Note that if the cloud size is given as n_{cs}, then it should satisfy the condition $n_{cl} \leq n_{gsr}$. Once an additional grain is selected, the amplitude of that grain is normalized to the average power of the target grain preceding the gap and the grain succeeding the gap.

3.4. Windowing

As grains are overlapped and added, it is likely that audible artifacts occur at the joints of grains as a result of abrupt change of amplitude. To remedy this situation, a grain is first weighted with a window function to taper either the end side or the beginning side, or both sides. The shape of the window is determined by the length of a grain and the values of the 'Grain Start Overlap' (Fig. 3, Table.2) and the 'Grain Stop Overlap' parameters. The Grain Start Overlap is the number of samples at the beginning of the grain to be tapered, and the Grain Stop

Overlap is the number of samples at the end of the grain to be tapered, respectively. Let n_{start}, n_{stop} be the values of the Grain Start Overlap and the Grain Stop Overlap and L be the length of the grain in concern, then using the Hann window, the window is defined as

$$w(n) = \begin{cases} 0.5(1 - cos(\pi \frac{n}{n_{start}})). \\ \quad \text{for} \quad 0 \leq n \leq n_{start} \\ 1. \\ \quad \text{for} \quad n_{start} < n < L - n_{stop} \\ 0.5(1 + cos(\pi \frac{n - (L - n_{stop})}{n_{stop}})). \\ \quad \text{for} \quad L - n_{stop} \leq n \leq L \end{cases} \tag{22}$$

Depending on values of n_{start} and n_{stop}, one can make a window either double-sided or one-sided. In general, a one-sided window with no tapering at the beginning is used to preserve the attack transient of a grain. This is often the case when using the grain extension method. On the other hand, a double-sided window could be used to smooth both sides of a grain used for bridging two grains into the gap when using the additional grain method.

3.5. Grain Extension Method vs. Additional Grain-Based Method

One thing to take note of is the characteristics of synthesized sounds in accordance with the proposed gap-filling methods. The principal difference of the grain extension method and the additional grain-based method is the grain density after synthesis. The grain density of the sound synthesized with the grain extension method varies proportionally with the time stretch factor, whereas that of the sound synthesized with the additional grain-based method is invariant with respect to the time stretch factor. Fig. 5 shows an example of synthesis with time stretching. Depending on the nature of the given sound and the user's purpose, either method could be preferred. For example, one can create two different kinds of clap sounds based on time modification. Fig. 6 shows the difference of the time stretched synthesis due to the choice of gap-filling methods[1]. The synthesis using the grain extension method results in a decrease of the grain density inversely proportional to the time stretch factor; on the other hand the synthesis based on the additional grain-based method keeps the grain density of the original clap sound. This aspect actually provides users with another option in synthesis, allowing for synthesized sounds that are sparse in terms of the grain density. In this case, the grain extension method plays the role of polishing each grain to avoid incurring audible artifacts due to the abrupt ends of grains.

[1] Sound examples are available at `http://www.music.mcgill.ca/~lee/AES49`

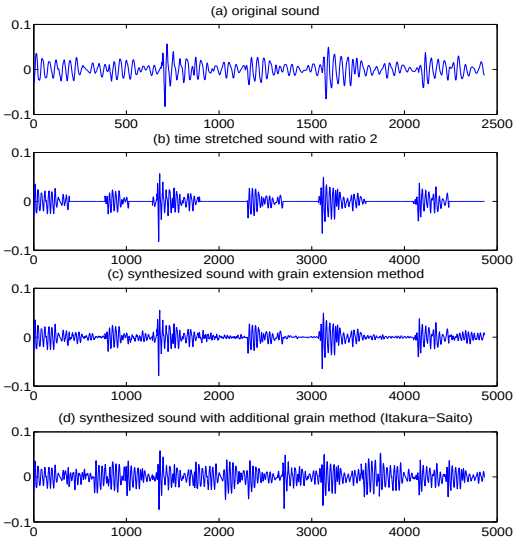

Fig. 5: (a) Original sound. (b) Original sound stretched with the time stretch factor $\alpha = 2$. (c) Gap filling with the grain extension method. (d) Gap filling with the additional grain method.

3.6. Grain Time Remapping

The concept of grain time remapping allows for more variations. Since all the grains have their own time positions representing locations of grains on the time axis, grain time remapping often allows for creating different scenarios for sound generations in the same environment. This can result in many interesting effects. For example, grain time remapping of the rolling ball sound would provide listeners with a variety of acoustic sensations since the time sequence of the grains has to do with the trajectory of the rolling ball, as mentioned in the previous chapter. Thus adjusting the time sequence actually have the effect of changing the trajectory of the ball. For example, if the time sequence of the grains is reversed (Fig. 7-(b)), the resulting synthesis will sound as if the ball were rolling backward along the trajectory of the original sound.

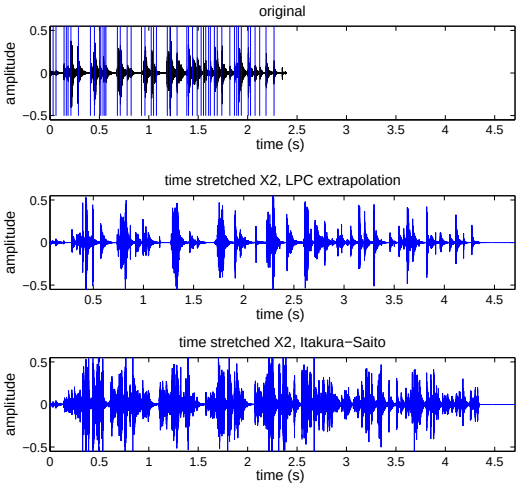

Fig. 6: Time stretched clap sounds. a) Original sound. Blue vertical bars denote the grain boundaries. (b) Time stretched sound by a factor $\alpha = 2$, with the grain extension method. (c) Time stretched sound by a factor $\alpha = 2$, with the additional grain-based method (Itakura-Saito).

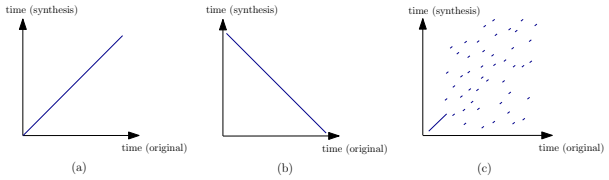

Fig. 7: grain time remapping examples. (a) no grain time remapping. (b) grain time remapping in reverse order. (c) random grain time remapping.

4. CONCLUSION

The outcome of the current research consists of two components. One is granular analysis and the other is granular synthesis. Both components were implemented in MATLAB and managed through GUIs. The granular analysis system is designed to detect onset-like events so that it can segment a given sound into grains. The granular analysis system is also able to discern stationary/non-stationary regions in the given sound and apply different segmentation parameters for each region, which enables users to apply different criteria for defining the grain in each region. In addition, useful audio features are also calculated for each grain for potential use in synthesis. Segmented grains are tagged with timing information and audio-features are stored in a dictionary. In conjunction with the granular analysis system, a novel, user-friendly granular synthesis system is presented, whereby the user can modify the temporal aspect of the sound in various ways with not only conventional time-scaling (stretching/shrinking) but also user-defined grain time remapping functions.

Future work will include several research tasks that could potentially enhance the current research outcomes. One would be finding an efficient way for grain compression other than using the 'Offset Threshold' parameter, taking advantage of the fact it is likely that redundant grains exist in a dictionary. Another problem to think about is how to figure out the inherent rhythmic aspect of a given sound. If we could analyze the rhythm of environmental sounds, though it might be hard to do so compared to those of speech and music, it would be beneficial in terms of the flexibility of the synthesis system.

5. ACKNOWLEDGMENTS

The authors are grateful to Audiokinetic Inc. and MITACS, who provided support for this research.

6. REFERENCES

[1] C. Roads, "Introduction to granular synthesis," *Computer Music Journal*, vol. 12, no. 2, pp. 27-34, 1988.

[2] C. Picard, N. Tsingos, and F. Faure, "Retargetting example sounds to interactive physics-driven animations," in *Proc. of AES 35th International Conference: Audio for Games*, (London, UK), February 2009.

[3] A. Lazier and P. R. Cook, "MOSIEVIUS: Feature driven interactive audio mosaicing," in *Proc. of the International Conference on Digital Audio Effects (DAFx-03)*, (London, U.K.), pp. 323-326, Sept. 2003.

[4] B. L. Sturm, "MATCONCAT : an application for exploring concatenative sound synthesis using matlab," in *Proc. of the International Conference on Digital Audio Effects (DAFx-04)*, (Naples, Italy), pp. 323-326, Oct. 2004.

[5] D. Schwarz, R. Cahen, and S. Britton, "Principles and applications of interactive corpus-based concatenative synthesis," in *Journées d'Informatique Musicale (JIM)*, (GMEA, Albi, France), March 2008.

[6] D. López, F. Martí, and E. Resina, "Vocem: An application for real-time granular synthesis," in *Proc. of International Conference on Digital Audio Effects (DAFx-98)*, 1998.

[7] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler, "A tutorial on onset detection in music signals," *IEEE Transactions of speech and audio processing*, vol. 13, no. 5, pp. 1035-1047, 2005.

[8] A. V. Oppenheim and R. W. Schafer, *Discrete-time signal processing*. Prentice-Hall, second ed., 1998.

[9] J. Lee, P. Depalle, and G. Scavone, "Analysis/synthesis of rolling sounds using a source-filter approach," in *Proc. of International Conference on Digital Audio Effects (DAFx-10)*, (Graz, Austria), 2010.

[10] X. Serra, *A system for sound analysis/transformation/synthesis based on a deterministic plus stochastic decomposition*. PhD thesis, CCRMA, Stanford University, Stanford, CA, 1989.

[11] J. D. Johnston, "Transform coding of audio signals using perceptual noise criteria," *IEEE journal on selected areas in communication*, vol. 6, no. 2, 1988.

[12] I. Kauppinen and K. Roth, "Audio signal extrapolation : Theory and applications," in *Proc. of the 5th Int. Conf. on Digital Audio Effects (DAFx- 02)*, (Hamburg, Germany), September 2002.

[13] W. Etter, "Restoration of a discrete-time signal segment by interpolation based on the left-sided and right-sided autoregressive parameters," *IEEE transactions on signal processing*, vol. 44, no. 5, pp. 1124-1135, 1996.

[14] R. C. Maher, "A method of extrapolation of missing digital audio data," *Journal of Audio Engineering Society*, vol. 42, no. 12, pp. 350-357, 1994.

[15] S. M. Kay and S. L. Marple Jr, "Spectrum analysis : a modern perspective," *Proceedings of the IEEE,*vol. 69, pp. 1380-1419, May 1981.

[16] F. Itakura and S. Saito, "An analysis-synthesis telephony based on the maximum likelihood method," in *Proc. of 6th International Congress of Acoustics*, (Tokyo, Japan), 1968.

[17] S. Sigurdsson, K. B. Petersen, and T. Lehn-Schiøler, "Mel frequency cepstral coefficients: An evaluation of robustness MP3 encoded music," in *Proc. of the 7th International Society for Music Information Retrieval Conference (ISMIR 06)*, (Victoria, Canada), 2006.