GENERATION OF NON-REPETITIVE EVERYDAY IMPACT SOUNDS FOR INTERACTIVE APPLICATIONS

Wasim Ahmad, Ahmet M. Kondoz

I-Lab, Centre for Vision, Speech and Signal Processing (CVSSP)
University of Surrey, Guildford, GU2 7XH, United Kingdom
{w.ahmad, a.kondoz}@surrey.ac.uk

ABSTRACT

The use of high quality sound effects is growing rapidly in multimedia, interactive and virtual reality applications. The common source of audio events in these applications is impact sounds. The sound effects in such environments can be pre-recorded or synthesized in real-time as a result of a physical event. However, one of the biggest problems when using pre-recorded sound effects is the monotonous repetition of these sounds which can be tedious to the listener. In this paper, we present a new algorithm which generates non-repetitive impact sound effects using parameters from the physical interaction. Our approach aims to use audio grains to create finely-controlled synthesized sounds which are based on recordings of impact sounds. The proposed algorithm can also be used in a large set of audio data analysis, representation, and compression applications. A subjective test was carried out to evaluate the perceptual quality of the synthesized sounds.

1. INTRODUCTION

Our environment is full of diverse types of impact sounds such as hitting, collision, bumping, dripping, etc. Such impact sounds are generally produced when two or more objects interact with each other. Pre-recorded versions of these sounds are used to generate such sound effects in interactive and virtual reality applications, in real time and offline productions. This method requires a large set of recordings of impact sounds to cover all possible situations which in turn necessitates a very large memory. One possible way to reduce recordings' size is by grouping them by size, material type, etc., but even then many recordings need to be carried out. For this reason, a small set of recordings of impact sounds is generally played back repetitively, and that can be tedious to the listener. Methods have been proposed to improve the realism of sound effects in games, such as the work of Vachon [1]. However, the repetition of sound effects in interactive applications, particularly in game's audio, remains a big challenge for the researcher and audio designer.

Alternatively, impact sounds can be generated automatically using either physics-based interaction of objects, known as physical models, or by imitating the properties of sound as perceived by the listener, known as spectral models. In recent years a number of such synthesis algorithms have been developed and applied to impact sounds synthesis [2, 3, 4, 5, 6, 7, 8, 9]. Physical models [2, 3, 4, 5] are very efficient and accurate in simulating a target sound but the refinement of such models is not always successful because the physical mechanisms of many environmental impact sounds are still not completely understood [10]. Therefore, a limited class of impact sounds has been targeted by this type of models. Furthermore, these models are computationally-intensive and require significant parameter-tuning to achieve realistic results, making it more difficult to use in a game production pipeline. In contrast, spectral models [6, 7, 8, 9] have a broader scope and construct the spectrum as received by the ear. Therefore, their refinement and repurposing is easier than physical models.

In recent years, combinations of sound synthesis models with pre-recorded sound have been used to generate high quality impact sound in interactive applications [11, 12]. Such approaches reduce the effect of the monotonous repetition of recorded sounds, and enhance the quality of synthesized sounds by linking the synthesis parameters to the physics engine. Bonneel *et al.* [11] presented a new frequency-domain method that used both pre-recorded sounds and physical models to generate high quality sounds. In [12], Picard *et al.* proposed a technique where non-repetitive sound events can be synthesized for interactive animations by retargeting the audio grains, extracted from the recorded sounds, based to the parameters received from the physics engine.

In this paper, we propose a similar approach where the pre-recorded impact sounds are represented in the form of a dictionary and synthesis patterns. During the generation phase, the synthesis pattern and corresponding atoms from the dictionary are selected according to the reported synthesis parameters from the physical interaction. During the analysis process, a continuous pre-recorded impact sounds are automatically segmented into individual events and all the events collected from different impact sound sources are decomposed into sound grains, where each grain has energy only at a particular frequency or scale. A dictionary is trained from the extracted sound grains. The recorded impact sound events are projected onto the dictionary which constitutes the synthesis patterns. During synthesis process, these patterns are tuned according to the target sound parameters.

2. SIGNAL REPRESENTATION TECHNIQUES

For many years, a large family of signal analysis techniques have heavily relied on Fourier transform (FT) and short-time Fourier transform (STFT) where the input signal is represented with the superposition of fixed basis functions i.e. sinusoids. The FT and STFT methods are most useful when considering stationary signals but most real-world sound signals are not stationary in time. Therefore, these analysis techniques are inadequate for such signals. Over the last two decades there has been a lot of interest to find alternative signal representation techniques which are adaptive and specialized to the signals under consideration. As a result, a number of basis functions and representation techniques have been developed to represent any input signal in a more compact, efficient, and meaningful way.

2.1. Dictionary-Based Methods

One of these techniques, which have attracted a lot of interest in recent years, is dictionary-based representation as it offers a compact form of the signal, and is highly adaptive. These methods have been used in many signal processing applications including analysis and representation of audio signals [13, 14] and music [15].

In dictionary-based methods, a signal is represented as a linear combination of elementary waveforms (atoms) taken from a dictionary. A dictionary is a collection of parameterized waveforms. Let $\mathbf{x}$ be a discrete-time real signal of length N i.e. $\mathbf{x} \in \Re^N$, and $\mathbf{D} = [\delta_1, \delta_2, \ldots, \delta_K]$ be a dictionary, where each column δ_k represents an atom and its length is N i.e. $\mathbf{D} \in \Re^{N \times K}$. The aim is to represent $\mathbf{x}$ as a weighted sum of atoms δ_k which can be written as,

$$\mathbf{x} = \sum_{k=1}^{K} \delta_k \, w_k \tag{1}$$

where $\mathbf{w}$ is a column vector in $\Re^K$ and represents the expansion coefficients or weights. Generally, the dictionary $\mathbf{D}$ is overcomplete i.e. $N < K$, which means the matrix $\mathbf{D}$ is of rank N and the linear system in Eq. (1) is undetermined. In that case, the decomposition vector $\mathbf{w}$ in Eq. (1) is not unique and there may even be an infinite number of possible expansions of the form of Eq. (1). Therefore, one has to introduce some additional constraints to specify a unique or particular decomposition.

2.2. Sparse Representations

Given an overcomplete dictionary $\mathbf{D}$ and the signal $\mathbf{x}$, finding a solution to the underdetermined systems given in Eq. (1) is a nontrivial task. In general, the representation in Eq. (1) is approximated by applying some additional constraints to specify a unique or particular solution. An adequate approximation of the signal $\mathbf{x}$ in Eq. (1) is obtained by selecting few atoms δ_k from dictionary $\mathbf{D}$ corresponding to highest weights w_k. That is, useful representations are the ones where most of the energy of the signal $\mathbf{x}$ is concentrated into a small number of coefficients, hence $\mathbf{x}$ can be approximated using only j atoms from the predefined dictionary as

$$\mathbf{x} = \sum_{k=1}^{j} \delta_k \, w_k + \mathbf{r} \tag{2}$$

or in matrix form

$$\mathbf{x} = \mathbf{D}\mathbf{w} + \mathbf{r} \tag{3}$$

where $j < K$ and $\mathbf{r} \in \Re^N$ is residual. The selection of atoms and their numbers are controlled by limiting the value of approximation error. By applying such criterion, the approximation solution given in Eq. (3) can be redefined as

$$\mathbf{x} \approx \mathbf{D}\mathbf{w} \;\text{ such that }\; \|\mathbf{x} - \mathbf{D}\mathbf{w}\|_2 \leq \epsilon \tag{4}$$

where ϵ is a given small positive number. The solution with the fewer number of atoms and corresponding weights is certainly an appealing representation. Sparse or compact approximation of a signal $\mathbf{x}$ is measured using the ℓ_0 criterion, which counts the number of non-zero entries of the weights vector $\mathbf{w} \in \Re^K$. The problem of finding the optimally sparse representation can be defined as the solution to

$$\min_{\mathbf{w}} \|\mathbf{w}\|_0 \;\text{ such that }\; \|\mathbf{x} - \mathbf{D}\mathbf{w}\|_2 \leq \epsilon \tag{5}$$

where $\|\mathbf{w}\|_0$ is the ℓ_0-norm, which count the number of non-zero coefficients in weight vector $\mathbf{w}$. The problem of finding the optimally sparse representation, i.e., with minimum $\|\mathbf{w}\|_0$, is a combinatorial optimization problem in general. Constraining the solution $\mathbf{w}$ to have the minimum number of nonzero elements creates an NP-hard problem [16] and cannot be solved easily. Therefore, approximation algorithms, such as matching pursuit (MP) [17], orthogonal matching pursuit (OMP) [18], and basis pursuit (BP) [19], are used to find an optimal approximation solution of Eq. (5). The MP and OMP algorithms are classified as greedy methods where a signal approximation is iteratively built up by selecting the atom that maximally improves the representation at each iteration. These algorithms converge rapidly, and exhibit good approximation properties for a given criterion [17, 20].

2.3. Selection of Dictionary

Dictionaries are often constructed from a combination of discretized, scaled, translated, and modulated lowpass functions. An overcomplete dictionary that leads to sparse representations can either be chosen as a prespecified set of functions or designed by adapting its content to fit a given set of signal examples. Choosing a prespecified transform matrix is appealing because it is simpler but there is no guarantee that these bases will lead to a sparse representation of signals under consideration.

The sparse approximation of the Eq. (5) can also be improved by using an appropriate dictionary for the given class of signals. Instead of using predetermined dictionaries, dictionary learning methods [21, 15] can be used to refine them. Such methods adapt an initial dictionary to a set of training samples. Therefore, the aim is to learn a dictionary for which an input signal, taken from a given class of signals, has a sparse approximation.

3. PROPOSED ANALYSIS-SYNTHESIS ALGORITHM

The proposed synthesis algorithm generates the target impact sounds using parametric representation modeled from the recorded impact sounds. This algorithm is divided into three stages i.e. analysis, parameterization, and synthesis, as depicted in Fig. 1. In the analysis phase, the recorded continuous impact sounds are segmented and split into sound grains. During the parameterization phase, the impact sounds are represented by synthesis patterns, and an adaptive dictionary trained from these sound grains. The target sound is generated at the synthesis stage where a pattern is selected and adjusted according to the parameters received from the physical interaction.

4. ANALYSIS OF RECORDED SOUNDS

The aim of the analysis process is to extract the sound grains which characterize the recorded impact sounds. The analysis stage includes the segmentation, peak alignment, and the extraction of sound grains.

4.1. Automatic Segmentation

The first step during the off-line analysis of the impact sound is to segment each recorded sound signal into individual *sound events* or simply *events*. For example, if the input sound is a clapping sound then each clap in the sound sequence is called an *event*,

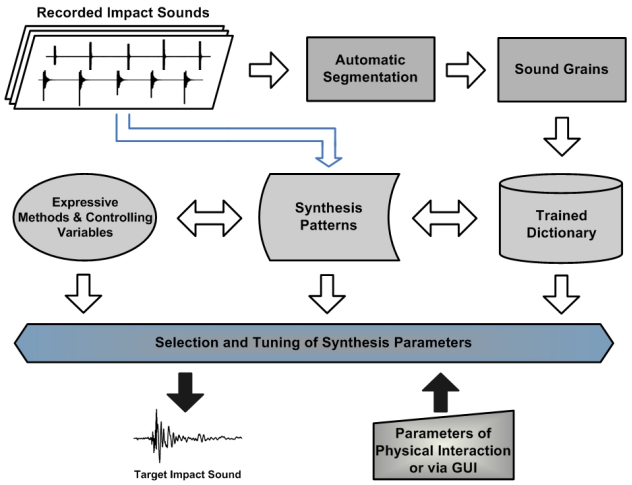

Figure 1: *Overview of the proposed analysis-synthesis algorithm.*

which is represented by $\mathbf{x}$. Each event is isolated by detecting and labeling its onset and offset points.

An impulsive event consists of an attack and a decay parts which are concatenated together. The onset of events is labeled using the energy distribution method proposed by Masri *et al.* [22], which detects the beginning of an impulsive event by observing the suddenness and the increase in energy of the attack transient. Detection is selected as an onset of an event when there is a significant rise in energy along with an increased bias towards the higher frequencies. Short-time energy of the signal is used to locate the offset of each event. Starting from the onset of each event, the short-time energy is calculated with overlapped frames, and compared against a constant threshold to determine the offset. Onset detection methods have been applied to input sounds that are monophonic i.e. only a single melodic line or tune is present, and music notes or events do not overlap [23, 24]. In this paper, we have also assumed that there is no overlapping of the sources in the recorded impact sounds.

Equal or different number of events can be selected from each sound source. Once the events are selected and segmented, they are peak aligned by cross-correlation such that the highest peaks occur at the same point in time. This increases the similarities between the extracted sound grains and improves the dictionary learning process. The set of collected sound events can be represented as a matrix $\mathbf{X}$ i.e.,

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m] \tag{6}$$

where each column represents a sound event and length of each event is n. Zero padding is used for any segmented sound event whose length is less than n.

4.2. Extraction of Sound Grains

The recorded impact sounds from different sources need to be represented in a way that i) the similarities and differences between various impact sounds can be observed and parameterized, ii) this parametric representation can be manipulated in various ways to generate sound effects at synthesis stage. Impact sound belongs to the transient signal family that is non-stationary. Based on the frequency resolution properties of the human auditory system, such signals can be split into layers of grains where the energy of each grain is presented at a particular frequency or scale. The information in each grain and the overall structure of these grains are analyzed based on human auditory system. Such parametric representation can be used to compare the characteristics of different sounds [25]. Furthermore, during the synthesis process, the parameters representing these grains can be manipulated in various ways to control the generated sound.

In the proposed scheme, stationary wavelet transforms (SWT) [26, 27] is used to extract the sound grains from the impact sound events. The SWT is the real-valued extension to the standard discrete wavelet transform (DWT). SWT is preferred over DWT because the latter lacks the property of shift-invariance. The SWT has the ability to underline and represent time-varying spectral properties of the transient signals and offers localization both in time and frequency.

The SWT is applied to each event, $\mathbf{x}_i$, which decomposes it into two sets of wavelet coefficient vectors: the approximation coefficients $\mathbf{ca_1}$ and the detail coefficients $\mathbf{cd_1}$, where the subscript represents the level of decomposition. The approximation coefficients vector $\mathbf{ca_1}$ is further split into two parts, $\mathbf{ca_2}$ and $\mathbf{cd_2}$, using the scheme shown in Fig. 2(a). This decomposition process continues up to L^{th} level which produces the following set of coefficient vectors: $[\mathbf{cd}_1, \mathbf{cd}_2, \ldots, \mathbf{cd}_L, \mathbf{ca}_L]$. The approximation coefficients represent the low-frequency components, whereas the detail coefficients represent the high-frequency components. To construct the sound grains from coefficients vectors, the inverse SWT is applied to each coefficient vector individually by setting all others to zero which produces the following bandlimited sound grains: $[\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_{L+1}]$. Each grain contains unique information from the sound event, retains the size of the the sound event. The block diagram of the process of extraction of sound grains from a coefficient vector is shown in Fig. 2(c). The entire sound event matrix $\mathbf{X}$ is split into sound grains which produce the grain matrix $\mathbf{G} = [\mathbf{g}_i : i = 1, 2, \ldots, p]$, where $\mathbf{g}_i$ form the columns of the grain matrix and the number of total grains are $p = m \times (L + 1)$.

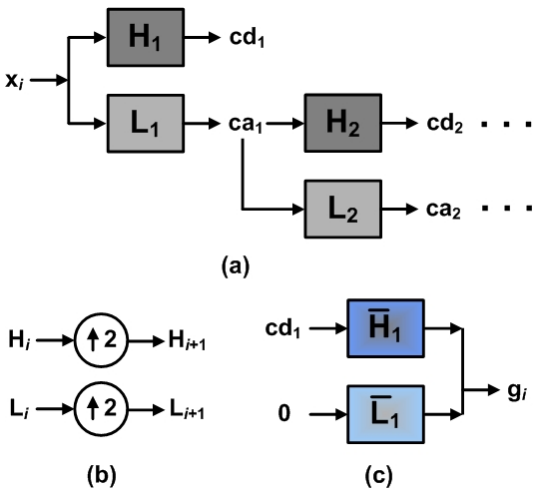

Figure 2: (a) Decomposition tree of SWT, (b) SWT filters, (c) construction of a sound grain.

The selection of wavelet type from the family of wavelets (i.e. Haar, Daubechies, etc.) and their decomposition level depend on

the input sound signal, application area, and the representation model. This is an iterative process where the best wavelet type and optimum decomposition level are obtained by evaluating the perceived quality of the synthesized sounds generated from the different wavelet types and decomposition levels.

5. PARAMETERIZATION

The proper parameterization of the sound features extracted from the analysis part is an essential element of the synthesis systems. In this paper, a dictionary-based approach is used to create a parametric representation of the recorded sounds. The similarities and differences of the sound grains, as well as their relationships to the input sounds are preserved and reflected in the presented parametric representation. One key advantage of dictionary-based signal representation methods is the adaptivity of the composing atoms. This gives the user the ability to make a decomposition suited to specific structures in a signal. Therefore, one can select a dictionary either from a pre-specified set of bases functions, such as wavelets, wavelet packets, Gabor, cosine packets, chirplets, warplets etc., or design one by adapting its content to fit a given set of signals, such as dictionary of instrument-specific harmonic atoms [15].

5.1. Dictionary Learning

Choosing a set of prespecified basis functions is appealing because of its simplicity but there is no guarantee that these basis functions will lead to a compact representation of given signals. The success of such dictionaries in practice depends on how suitable they are to sparsely describe the signals in question. However, there are many potential application areas, such as transient and complex music sound signals, where fixed basis expansions are not well suited to model this type of sound signals. A compact decomposition is best achieved when the elements of the dictionary have strong similarities with the signal under consideration. In this case, a fewer set of more specialized basis functions in the dictionary is needed to describe the significant characteristics of the signal [15, 28, 29]. Ideally, the basis itself should be adapted to the specific class of signals which are used to compose the original signal. As we are dealing with a specific class of sound signals, we believe that it is more appropriate to consider designing learning-based dictionaries.

Given training impact sounds and using adaptive training process, we seek a dictionary that yields compact representations of the sound event matrix $\mathbf{X}$. The K-SVD algorithm [21] is such a technique for training a dictionary from given example signals. It is a highly effective method, and has been successfully applied to several image processing tasks [30, 31]. The K-SVD algorithm consists of an iterative process of optimization to produce a sparse representation of the given samples based on the current dictionary, and an update of the atoms that best represent the samples. The update of the dictionary columns is done along with an update of the sparse representation coefficients related to it, resulting in accelerated convergence.

In the proposed scheme, the K-SVD algorithm is used to train an adaptive dictionary $\mathbf{D}$ which determines the best possible representation of a given impact sounds. The K-SVD algorithm takes the sound grains matrix $\mathbf{G}$, as initial dictionary $\mathbf{D}_0$, a number of iterations j, and a set of training signals, i.e sound event matrix $\mathbf{X}$. The algorithm aims to iteratively improve the dictionary to achieve sparser representations of the sound events in $\mathbf{X}$, by solving the optimization problem

$$\min_{\mathbf{w}_i} \|\mathbf{x}_i - \mathbf{D}\mathbf{w}_i\|_2^2 \;\text{ such that }\; \forall i \; \|\mathbf{w}_i\|_0 \leq T_0 \tag{7}$$

where T_0 is the number of non-zero entries in $\mathbf{w}_i$. The iteration of K-SVD algorithms is performed in two basic steps: i) given the current dictionary, the sound events in $\mathbf{X}$ are sparse-coded which produce the sparse representations matrix $\mathbf{W}$, and ii) using this current sparse representations, the dictionary atoms are updated. The dictionary update is performed one atom at a time, optimizing the target function for each atom individually while keeping the other atoms fixed.

5.2. Synthesis Pattern

The OMP is used to find the synthesis patterns of the input impact sound events over the dictionary. The OMP is a greedy step-wise regression algorithm. The aim of OMP algorithm is to approximate the solution of the sparsity-constrained sparse coding problem given in Eq. (7), where the dictionary atoms have been normalized. At each stage, this algorithm selects the dictionary atom with the maximal projection onto the residual signal. Once the atom is selected, the signal is orthogonally projected to the span of the selected atoms, the residual is recomputed, and the process is repeated. The algorithm stops after a predetermined number of steps, selecting a fixed number of atoms T_0 in every iteration. At this stage, the impact sound matrix $\mathbf{X}$ can be fully represented as a dictionary matrix $\mathbf{D}$ and synthesis patterns matrix $\mathbf{W}$. The information about the impact sound sources is labeled onto synthesis pattern $\mathbf{W}$ for future reference and for possible use during the synthesis process.

6. GENERATION OF TARGET SOUND

To synthesize the target impact sound, the controlling variables are employed to select the best sound parameters. During the synthesis process, an impact sound event from the sound matrix $\mathbf{X}$ is synthesized by selecting the decomposition pattern $\mathbf{w}_i$ and then adding the corresponding weighted dictionary atoms, which can be written as,

$$\hat{\mathbf{x}}_i \cong \sum_{j \in J} \delta_j \; \mathbf{w}_i(j) \tag{8}$$

where J contains the T_0 number of indices of the non-zero entries in $\mathbf{w}_i$. The perceptual quality of the synthesized impact sound event $\hat{\mathbf{x}}_i$ is directly related to the number of non-zero entries in $\mathbf{w}_i$. The quality of synthesized impact sound event $\hat{\mathbf{x}}_i$ improves sharply for the first few atoms but become imperceptible after a particular value of T_0.

6.1. Expressive Synthesis Method

Two sound events generated consecutively by the same sound source will be similar but not identical. For example, when a person claps twice in the same way with the same applied force, the generated clapping sounds will be similar but not identical. The proposed algorithm can synthesize example impact sounds approximately from the represented parameters, i.e. synthesis pattern $\mathbf{W}$ and dictionary atoms $\mathbf{D}$. A limited number of impact sound events sequence can be generated from this representation as the number of synthesis pattern vectors is limited and fixed. Therefore, the same set of impact sounds will be repeated during long impact

sound sequences, which will make it perceptually artificial in the ears of the listeners.

To generate more natural and customized sounds, the proposed method modifies the synthesis process given in Eq. (8). This equation uses the represented parameters, $\mathbf{W}$ and $\mathbf{D}$, to synthesize an impact sound event. Every time Eq. (8) is executed to synthesize an impact sound event $\hat{\mathbf{x}}_i$, a synthesis pattern $\mathbf{w}_i$ is used to combine the dictionary atoms. For expressive synthesis, when an impact sound event $\hat{\mathbf{x}}_i$ is generated, a small random vector ψ is added to the selected synthesis pattern $\mathbf{w}_i$ such that the overall time-varying spectrum of the impact sound is unchanged. The value of ψ is generated randomly in a sphere of radius R with the origin at the synthesis pattern of the generated impact sound. A different vector ψ is generated for every event of impact sound and the length of ψ is equal to T_0 because only non-zero entries in $\mathbf{w}_i$ are changed. Hence, The synthesis equation given in Eq. (8) is modified for the expressive synthesis process and can be rewritten as,

$$\hat{\mathbf{x}}_i \cong \sum_{j \in J} \delta_j \, [\mathbf{w}_i + \psi](j). \tag{9}$$

The impact sound sequence generated using Eq. (9) will be similar but not identical, and they will also not be exact copies of the sound events matrix $\mathbf{X}$.

7. SUBJECTIVE EVALUATION OF SYNTHESIS SOUND QUALITY

Subjective tests have been used to accurately assess the quality of the sound events generated by the proposed algorithm.

7.1. Impact Sound Database

A sample of commonly heard everyday impact sounds were used to evaluate the perceptual quality of sounds synthesized using the proposed analysis-synthesis algorithm. The group contains six impact sounds which include: bumping sounds of a tennis ball, a football and a basketball on laminate floor; a finger knocking sound on a wooden table; and male and female clapping sounds. The recordings of these sounds were made in an acoustical booth ($T_{60} < 100$ ms) at a sampling rate of 44.1 kHz.

To record the bumping sounds, each ball was dropped on laminated floor from a fixed height of 80 cm with no applied force. After each bump, the ball is lifted up to the same height and dropped again. A microphone was placed vertically close to the floor level and horizontally about 100 cm away from the potential point of contact at the floor. The experimenter knocked the centre of the wooden table[1] top with his right hand index finger with a constant force. To capture this sound, the microphone was placed at the level of table top and about 100 cm away horizontally from the centre of the table. The recording of the clapping sounds was made with one male and one female subjects[2], both between the age of 25 and 35. Each subject was seated in the acoustical booth alone and a microphone was placed about 100 cm away from their hands. Subjects were asked to clap at their most comfortable or natural rate using his or her conventional clapping style. A set of sequences was recorded for each sound source where each sequence contains series of event.

[1] The size of the table was 20.5 cm width, 39.5 cm length, and 28.5 cm height.

[2] The male clapper was the author and the female clapper was a research fellow at I-Lab.

7.2. Synthesis Model and Stimuli

The purpose of this listening test is to compare the quality of the synthesized impact sounds with the original recorded sounds. The set of sequences of six impact sounds from the recorded database were segmented into individual sound events. An equal number of sound events, i.e. 30, were taken from each impact sound type. The segmented sound events were peak aligned and put into a matrix form i.e. $\mathbf{S} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m]$, where $m = 6 \times 30 = 180$ was the number of collected events and the length of each event was $n = 2048$. To extract the sound grains from the collected event matrix $\mathbf{X}$, the SWT was applied to each event up to the 5^{th} level. That produced the sound grains matrix $\mathbf{G} = [\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_p]$, where $p = 180 \times 6 = 1080$ represented the number of sound grains and the length of each grain was equal to $n = 2048$. The parametric representation of the input impact sounds was done by training an adaptive dictionary using the extracted sound grains. Given the sound grains matrix $\mathbf{G}$ as an initial dictionary $\mathbf{D_0}$, and the impact sound events matrix $\mathbf{X}$ as training signals, K-SVD algorithm was used to train a final adaptive dictionary $\mathbf{D} = [\delta_1, \delta_2, \ldots, \delta_K]$, with a number of atoms equal to $K = 108$. To find the decomposition patterns $\mathbf{W}$, OMP was used to project recorded sound events over the dictionary $\mathbf{D}$. Hence, the decomposition patterns $\mathbf{W}$ and adaptive dictionary $\mathbf{D}$ fully described the parametric representation of the input impact sounds.

Three groups of stimuli were synthesized from the represented model of the recorded sounds $\mathbf{W}$ and $\mathbf{D}$. The first group of stimuli was synthesized using seven atoms from the represented model, while in the second and third groups, they were synthesized using fourteen and twenty one atoms respectively from the represented model. Furthermore, each group contains twelve stimuli, where two stimuli are used from each sound source: male clapper, female clapper, tennis ball, football, basketball, and one finger and table. During the synthesis process, one event of the target sound was generated from the represented model using seven, fourteen, and twenty one atoms. However, when this event was used as a stimulus, the same event was repeated three times with 0.5 seconds interval. Similarly, the corresponding reference sound (the original recorded event) was also repeated three times with 0.5 seconds interval. During each experiment, one reference stimulus and a corresponding synthesized stimulus from each group were presented to the subjects simultaneously. The subjects listened to the reference and synthesized stimuli and graded the quality of the synthesized sounds. The subjects' responses were collected using the graphical user interface (GUI).

7.3. Subjects and Evaluation Setup

A group of 10 subjects (8 male and 2 female), between the age of 26 and 40, participated in the subjective evaluation. The subjects included PhD students and staff from the I-Lab centre with no reported hearing impairment. The subjects were trained before the evaluation session and can be considered to be expert listeners.

For the evaluation experiment, the subjects were seated in an isolated multimedia room. The experimental setup consisted of one Dell Inspiron 630m laptop and one Sennheiser HD 500 headphone. Every subject was familiarized with the evaluation process by undertaking a training session. A GUI was built in MATLAB which was used for the training and sound quality evaluation processes.

During each experiment, the subjects were presented with one reference stimulus (the original recording) and three test stimuli

(one for each group) from the same sound source. Since there were twelve experiments, the total number of synthesized sounds evaluated by each subject was equal to 36. The subjects' task was to listen to the reference and the three test sounds, and then rate the quality of the test sounds in comparison to the reference stimulus. The subjects can replay the reference and test sounds as many times as they wished. To register a rating for each test, the subjects were asked to move the slider on a scale ranging from 0 to 100. The 0 to 100 scale is divided into five equal quality steps: Excellent (81-100), Good (61-80), Fair (41-60), Poor (21-40), and Bad (0-20). Once subjects completed all test sounds within a particular experiment, they could move to the next one by clicking the "save and proceed" button, which stored the rating and presented them with the following set of tests. Each subject took about 15 minutes to evaluate all the experiments.

7.4. Results

Fig. 3 shows the mean evaluation ratings from all the subjects as well as the bars 95% confidence intervals of the mean ratings. It can be observed that the higher the number of atoms used in the synthesis process, the better the perceived quality. Furthermore, the relationship between the perceived quality of the synthesised sound and the number of atoms is linear. This result is due to the fact that as the number of atoms increases in the synthesized pattern, the synthesized sound event approximate more closely to the original signal. The figure also shows that even with a small number of atoms, $T_0 = 7$ out of the size of the dictionary $K = 108$, the mean subjects' rating of the quality was "Good". This indicates that our method achieved a perceptually acceptable level of sound quality with only few number of atoms, hence a more compact form. When increasing the number of atoms to $T_0 = 21$, the mean quality rating improved to "Excellent". These results highlight the efficiency of the parameterization technique used, and the advantages of using an adaptive dictionary trained from sound grains that are extracted from the input signal.

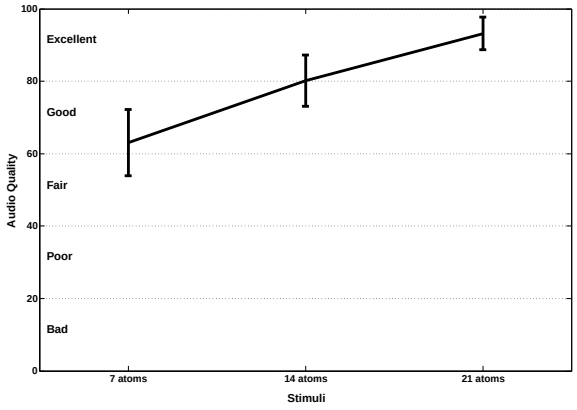

Figure 3: Mean synthesis quality of the synthesized sounds averaged across all subjects. The bars show 95% confidence intervals of the mean ratings.

8. CONCLUSIONS

We presented a new algorithm, which can synthesize any impact sound by analyzing and representing the recorded sound as a set of atoms and synthesis patterns. The atoms of the dictionary were first adaptively trained from the input sound using K-SVD algorithm, and then the synthesis patterns were generated by projecting the sound events over the trained dictionary. The target sound was synthesized by selecting and tuning the synthesis pattern and their corresponding atoms from the dictionary. In addition, an expressive synthesis method was presented which can generate non-repetitive and customized impact sounds. Subjective tests were carried out to evaluate the perceptual quality of the synthesis model. The tests' results showed that it is possible to achieve a satisfactory level of perceived sound quality using the compact representation of a given impact sound with a small number of atoms ($T_0 = 7$) from the trained dictionary. An approximation sound with $T_0 = 21$ was sufficient to yield an "Excellent" quality average rating.

As part of future work, we will further investigate the expressive synthesis model and analyze the distribution of the synthesis patterns of real life sound events and their possible statistical modeling. The quality and realism of the synthesized impact sounds generated from expressive synthesis model will be evaluated using subjective tests.

9. REFERENCES

[1] Jean-Frederic Vachon, "Avoiding tedium: Fighting repetition in game audio," in *Proc. of Int. Audio Engineering Society Conference: Audio for Games*, London, UK, Feb 2009.

[2] Kees Van Den Doel, P G Kry, and D K Pai, "Foleyautomatic: Physically-based sound effects for interactive simulation andanimation," in *Proc. of ACM Int. Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH-01)*, Los Angeles, USA, Aug 2001, pp. 537–544.

[3] Matthias Rath, Davide Rocchesso, and Federico Avanzini, "Physically based real-time modeling of contact sounds," in *Proc. of Int. Computer Music Conf. (ICMC-2002)*, Goteborg, Sweden, Sep 2002.

[4] Leevi Peltola, Cumhur Erkut, Perry R Cook, and Vesa Välimäki, "Synthesis of hand clapping sounds," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 3, pp. 1021–1029, Mar 2007.

[5] Niels Böttcher and Stefania Serafin, "Design and evaluation of physically inspired models of sound effects in computer games," in *Proc. of Int. Audio Engineering Society Conference: Audio for Games*, London, UK, Feb 2009.

[6] Mitsuko Aramaki and Richard Kronland-Martinet, "Analysis-synthesis of impact sounds by real-time dynamic filtering," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 2, pp. 695–705, Mar 2006.

[7] Ananya Misra, Perry R Cook, and Ge Wang, "Musical tapestries: Re-composing natural sounds," in *Proc. of Int. Computer Music Conf. (ICMC-06)*, New Orleans, U.S, Nov 2006.

[8] Wasim Ahmad, Huseyin Hacıhabiboğlu, and Ahmet M. Kondoz, "Analysis-synthesis model for transient impact sounds

by stationary wavelet transform and singular value decomposition," in *Proc. of Int. Computer Music Conf. (ICMC-08)*, Belfast, Northern Ireland, Aug 2008, pp. 49–56.

[9] Wasim Ahmad, Huseyion Hacıhabiboğlu, and Ahmet M. Kondoz, "Morphing of transient sounds based on shift-invariant discrete wavelet transform and singular value decomposition," in *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP-09)*, Taipei, Taiwan, Apr 2009, pp. 297–300.

[10] Perry R. Cook, "Physically informed sonic modeling (phism): Synthesis of percussive sounds," *Computer Music Journal*, vol. 21, no. 3, pp. pp. 38–49, Autumn 1997.

[11] Nicolas Bonneel, George Drettakis, Nicolas Tsingos, Isabelle Viaud-Delmon, and Doug James, "Fast modal sounds with scalable frequency-domain synthesis," *ACM Transactions on Graphics (SIGGRAPH Conference Proceedings)*, vol. 27, no. 3, August 2008.

[12] Cécile Picard, Nicolas Tsingos, and Francois Faure, "Retargetting example sounds to interactive physics-driven animations," in *Proc. of Int. Audio Engineering Society Conference: Audio for Games*, London, UK, Feb 2009.

[13] Remi Gribonval and Emmanuel Bacry, "Harmonic decomposition of audio signals with matching pursuits," *IEEE Transactions on Signal Processing*, vol. 51, no. 1, pp. 101–111, Jan 2003.

[14] Bob L. Sturm, Curtis Roads, Aaron McLeran, and John J. Shynk, "Analysis, visualization, and transformation of audio signals using dictionary-based methods," in *Proc. of Int. Computer Music Conf. (ICMC-2008)*, Belfast, Northern Ireland, Aug 2008.

[15] P. Leveau, E. Vincent, G. Richard, and L. Daudet, "Instrument-specific harmonic atoms for mid-level music representation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, no. 1, pp. 116–128, Jan 2008.

[16] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM Journal on Computing*, vol. 24, no. 2, pp. 227–234, Apr 1995.

[17] Stephane G. Mallat, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, Dec 1993.

[18] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. of Asilomar Conf. on Signals, Systems and Computers*, Nov 1993, vol. 1, pp. 40–44.

[19] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, Dec 1998.

[20] Laurent Daudet, "Audio sparse decompositions in parallel : Let the greed be shared," *IEEE Signal Processing Magazine*, vol. 27, no. 2, pp. 90–96, Mar 2010.

[21] Michal Aharon, Michael Elad, and Alfred Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, Nov 2006.

[22] Paul Masri and Andrew Bateman, "Improved modeling of attack transients in music analysis-resynthesis," in *Proc. of Int. Computer Music Conf. (ICMC-96)*, Hong Kong, China, Aug 1996, pp. 100–103.

[23] Xavier Rodet and Florent JailletRodet:2001, "Detection and modeling of fast attack transients," in *Proc. of Int. Computer Music Conf. (ICMC-01)*, Havana, Cuba, 2001, pp. 1–4.

[24] Juan P. Bello and Mark Sandler, "Phase-based note onset detection for music signals," in *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP-03)*, Apr 2003, vol. 5, pp. 441–444.

[25] Simon Tucker and Guy J. Brown, "Classification of transient sonar sounds using perceptually motivated features," *IEEE Journal of Oceanic Engineering*, vol. 30, no. 3, pp. 588–600, Jul 2005.

[26] M. Holschneider, R. Kronland-Martinet, J. Morlet, and P. Tchamitchian, "A real-time algorithm for signal analysis with the help of the wavelet transform," in *Wavelets: Time-Frequency Methods and Phase Space*, Jean-Michel Combes, Alexander Grossmann, and Philippe Tchamitchian, Eds. 1989, pp. 289–297, Springer-Verlag.

[27] G. P. Nason and B.W. Silverman, "The stationary wavelet transform and some statistical applications," in *Lecture Notes in Statistics, 103*, pp. 281–299. 1995.

[28] M. D. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M. E. Davies, "Sparse representations in audio and music: From coding to source separation," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 995–1005, Jun 2010.

[29] Michael S. Lewicki, Terrence J. Sejnowski, and Howard Hughes, "Learning overcomplete representations," *Neural Computation*, vol. 12, no. 2, pp. 337–365, Feb 2000.

[30] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736 –3745, Dec 2006.

[31] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Transactions on Image Processing,*, vol. 17, no. 1, pp. 53 –69, Jan 2008.