EFFICIENT SIGNAL EXTRAPOLATION BY GRANULATION AND CONVOLUTION WITH VELVET NOISE

Stefano D'Angelo

Independent researcher
Agropoli, Italy
s@dangelo.audio

Leonardo Gabrielli

Università Politecnica delle Marche,
Ancona, Italy
l.gabrielli@univpm.it

ABSTRACT

Several methods are available nowadays to artificially extend the duration of a signal for audio restoration or creative music production purposes. The most common approaches include overlap-and-add (OLA) techniques, FFT-based methods, and linear predictive coding (LPC). In this work we describe a novel OLA algorithm based on convolution with velvet noise, in order to exploit its sparsity and spectrum flatness. The proposed method suppresses spectral coloration and achieves remarkable computational efficiency. Its issues are addressed and some design choices are explored. Experimental results are proposed and compared to a well-known FFT-based method.

1. INTRODUCTION

Several techniques have been devised since the advent of digital signal processing for the creative generation of *textures* and signal *freezing* effects. Some of these methods, or variations thereof, are also employed for audio restoration (see e.g. [1]), as they allow to mimic a given signal and extend its time duration. Several techniques have been proposed [2], among which some of the most used ones are:

- Overlap-and-add (OLA) techniques [3, 4];
- FFT-based methods based on spectral analysis and resynthesis [5];
- Linear Predictive Coding (LPC) schemes ([6, 7]).

OLA techniques constitute the foundation of granular synthesis, which essentially consists in summing together several time-shifted copies of a small number of short and usually windowed signals (grains) to form the output signal. Generally, however, granulation is meant as a creative tool, thus grains are often processed, e.g. with constant or time-varying pitch-shifting. Despite its conceptual simplicity, this synthesis method finds use in a wide variety of applications. For extrapolation and freezing, it is sufficient to employ a single input grain and have a sufficient density of overlapping repetitions. The relative computational efficiency of such algorithms is anyway normally counterbalanced by spectral coloration, modulation effects, and phase-related artifacts, unless countermeasures are taken [4].

Vocoding [8] is a well-known FFT-based method for signal analysis and resynthesis and it has been used for the purpose of freezing or texturing of a signal. Being block-based, it results in nonuniform execution time and/or high implementation complexity, and significant difficulties arise in handling parameter changes.

Finally, LPC methods generally achieve best output performance in terms of timbre quality, and extensions of these methods can also work in the time-frequency domain, thus allowing for accurate modeling of transients [9]. The good output quality normally obtained by LPC methods is however traded for high computational cost due to the adaptive filtering techniques [10] they are based on.

In this work we describe an OLA method for signal extrapolation which, unlike previous approaches [3, 4], is targeted not only for efficiency, but also for maximal spectral flatness, leading to results that are on par with FFT-based techniques.

The outline of the paper follows. In Section 2 we introduce OLA techniques and justify our proposition from a theoretical perspective. Section 3 reports implementation details, experimental and comparative data. Finally, Section 4 concludes the paper and discusses the outcomes of this research.

2. PROPOSED METHOD

Overlap-and-Add methods are widely used in digital signal processing to evaluate the convolution between two signals, one of which has finite length, e.g. a filter kernel, and another that can theoretically be infinitely long. If $s[n]$ is the latter signal, we can decompose it in non-overlapping blocks of length L, i.e.

$$s[n] = \sum_{r=0}^{+\infty} s_r[n - rL]. \tag{1}$$

Thus, the result of the convolution between such running signal and a finite impulse response $h[n]$ can be defined as

$$c[n] = \sum_{r=0}^{+\infty} s_r[n - rL] * h[n] = \sum_{r=0}^{+\infty} c_r[n - rL]. \tag{2}$$

If $h[n]$ has length P, then $c_r[n - rL]$, has length $L + P - 1$. Therefore, each two contiguous blocks c_r and c_{r+1} need to be overlapped and added (hence the name) to obtain the corresponding portion of $c[n]$.

Many signal extrapolation methods work by summing time-shifted copies of the windowed input signal $x_\mathrm{w}[n]$. This is conceptually equivalent to applying the OLA method to compute the convolution between $x_\mathrm{w}[n]$, impersonating the fixed-length signal, and an impulse train $v[n]$ as the running signal. If the impulses in $v[n]$ are equally spaced, as is often done, the operation will inevitably produce spectral coloration. This can be intuitively understood by considering that such a process corresponds to feeding $x_\mathrm{w}[n]$ into a feedback comb filter with unitary gain, thus resulting in significant cancellation of spectral components that cannot be compensated by post-equalization.

For our purposes, we need $v[n]$ not only to have infinite temporal duration, but also a sufficiently flat spectrum. Two well-known signals that have these properties are white noise and dense

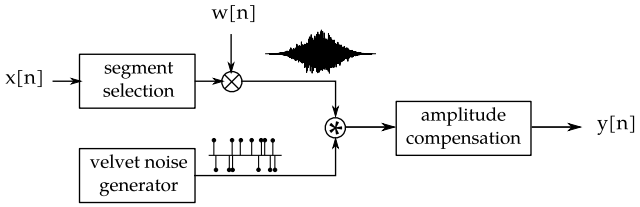

Figure 1: Overview of the proposed system.

velvet noise [11]. Velvet noise, in particular, consists of randomly-spaced unitary bipolar impulses and it has been shown to approximate white noise from a psychoacoustical standpoint when its pulse density rises above a certain threshold [12]. Due to its sparsity and the constant amplitude of impulses, convolution with velvet noise can be efficiently implemented in the time domain by simply summing together multiple randomly time-shifted copies of $x_{\mathrm{w}}[n]$ with random sign. The random nature of $v[n]$ implies random fluctuations of the local energy, requiring, thus, an amplitude compensation mechanism to reduce this undesired phenomenon, later discussed. The overall architecture is shown in Figure 1.

2.1. Issues and Implementation

The proposed method exposes three degrees of freedom in its design and operation: grain length, choice of window function, and velvet noise density.

In granular synthesis, the user can often directly choose which window function is applied as this has noticeable effect on the sound, and especially when using short grains. In our case, we definitely need windowing to eliminate potential discontinuities at the extremes of the input grain, and it would be preferable to pick a function that has high dynamic range and that is easy to compute. However, since grains need to be relatively long to retain low frequency components in the output, we can pragmatically choose the window function based on computational cost alone. The Welch window seems to be a valid choice because it is twice differentiable, except at the extremes, and has an exceptionally low computational cost. In Section 3 a few low cost windows, namely the triangular, half sine, and Welch windows, are compared.

While many musical signal processing devices nowadays are able to perform real-time convolution between a running signal and a long impulse response, the complexity and computational cost of our system can be largely reduced by leveraging the concept of *voices*, as in other forms of synthesis. Each voice is a sample playback engine, triggered randomly and with random sign, thus reducing the convolution operation to a limited number of random memory accesses, sums, and sign changes per output sample. The only potential drawback of this approach is that a finite number of voices needs to be defined beforehand, thus limiting the number of possible simultaneous grain playbacks, which theoretically corresponds to imposing a maximum "instantaneous density" to the velvet noise signal.

Given the suggested implementation approach, we believe it makes most sense to parameterize in terms of simultaneous grains on average, which corresponds to the product of the average velvet noise density (spikes over time) and the grain length. It is probably impossible to determine an optimal density for a given input signal, and especially when the input grain is somehow not sonically uniform (e.g., it contains transients), however we have empirically verified that satisfactory results can be in most cases obtained by employing relatively few voices, usually less than 30. Furthermore, preallocating twice the number of average voices reduces the likelihood of running out of available voices at any instant to at most a few percentage points.

A last issue that needs to be addressed derives from the local energy fluctuations of $v[n]$ that are inherent to its random nature. Those are also found in the output signal and need be compensated for. Significant variations of the amplitude are indeed usually noticeable in our experiments. To attenuate these, at least in a psychoacoustic sense, we propose applying a time-varying gain which depends on the signal volume. We propose employing a simple VU meter-inspired envelope detector for volume estimation, which performs full-wave rectification and conversion to the dB scale (with a lower limit of -120 dB), then applying a one-pole lowpass filter with a rise/fall time of 300 ms for 99% excursion (i.e., $\tau \approx 65.144$ ms). In order to match input and output levels, the same volume estimator can be also applied to the input signal to establish a target level. In any case, the gain factor needs to be limited to avoid the occurrence of loud peaks.

A schematic overview of the implemented algorithm is shown in Figure 2 where the amplitude compensation strategy described in Section 2 is detailed.

3. EXPERIMENTAL RESULTS

The algorithm has been implemented as a GNU Octave script to determine the quality of the audio output. The script and sound samples are available at `http://www.dangelo.audio/dafx2018-freeze.html`. A C++ implementation has also been developed for execution on regular desktop computers and on an embedded system running ELK by Mind Music Labs[1]. It was tested on two laptops, an Acer Extensa 5220 (Intel Celeron M 530 1.73 GHz single-core CPU, 1 GB DDR2 RAM) and an Acer Aspire E1-522 (AMD A4-5000 1.5 GHz quad-core CPU, 4 GB DDR3 RAM), both running 64-bit Arch Linux and using an external Focusrite Scarlett 2i4 sound card. In all cases (laptops and embedded system), the CPU load never exceeded 9% for a grain density of 32 simultaneous grains on average, at a sample rate of 44.1 kHz and with different buffering configurations.

3.1. Qualitative Results

Informal listening tests have been conducted with several audio source materials. An example of such experiments is reported in Figure 4, where a small excerpt of a male voice singing an */a/* phoneme tuned to A2 is taken as source. Its spectrogram is shown in Figure 4(a) and its DFT is shown in Figure 4(d). This signal has been extrapolated according to the proposed algorithm yielding a signal of length 5 s. Its spectrogram is shown in Figure 4(b) and its time-domain representation is shown in Figure 4(c). Random fluctuation of the overall amplitude is visible, however within a range of 3 dB maximum. The DFT from the original and the extrapolated signals are depicted in Figure 4(d-e) and show high resemblance, as expected, due to the spectral flatness of velvet noise. Similar experiments have been done with less stationary audio material, such as a guitar chord, polyphonic music and percussive instruments, with similar outcomes, see Figures 3, 8, 7.

[1] `https://www.mindmusiclabs.com/`

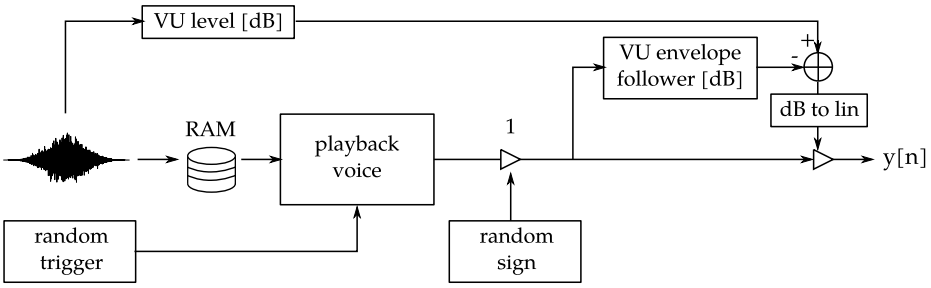

Figure 2: Schematic overview of the implemented algorithm, including the amplitude compensation strategy and exploiting multiple sample playback voices to reduce the computational cost of the convolution.

Figure 5 shows the DFT from the impulse train method. This signal has been synthesized by convolution with an impulse train having the same density as in the previous experiments, i.e. 32 pulses per second. In this case, it is quite evident that the signal has a comb-like pattern, with peaks at multiples of 32 Hz. As discussed previously, since the convolution with an impulse train has the same effect of a comb filter with unitary gain, the peaks are very pronounced, losing the timbre of the original signal.

We have also verified that the output sound quality has little dependency on the choice of the window function when the input grain is sufficiently long. The DFT from signals extrapolated using three window types, triangular, half sine, and Welch, are shown in Figure 6(a),(b) and (c), respectively. The results are almost identical, as expected. Please note that this is also true for any pulse density.

3.2. Comparison to FFT analysis-resynthesis

In this section we compare the proposed method with a well-known method based on FFT analysis-resynthesis, dubbed *timbre stamping* in [5]. In general, the quality of such an FFT-based method is rather high if the number of DFT bins is sufficiently large. In Figures 7 and 8 we compare the proposed method and the FFT-based method with a polyphonic music excerpt (trumpet playing a scale and accompanying jazz combo in the background) and a percussive jazz excerpt (containing a double bass note and cymbals) respectively. Both methods retain features of the original spectra. For instance, the polyphonic excerpt overlaps the notes of the scale are contained in the window. The time envelope of the FFT-based method is perceived as smoother for long grain size (1 s or more), however with shorter windows, such as those used in the figures (32 windows per second and window size of 300ms) the FFT method shows periodic repetitions in the output that are easily perceived especially in the presence of transients in the windowed signal. This is even clearer with percussive audio material. In the FFT-based method, transients may result smeared and are repeated periodically. The proposed method shows to have a smoother temporal domain envelope with respect to the FFT-based method, resulting in a less mechanical behavior and a denser output.

4. CONCLUSIONS

This paper described a novel method for signal extrapolation that has a low computational cost and is, thus, easily implemented in real-time applications. The method is mathematically formulated as a convolution problem with spectral flatness as a constraint. Owing from overlap-and-add methods we derived a formulation that ensures maximal spectral flatness. The low computational cost of this method is an additional benefit that allows for real-time implementations with a very low effort, as it processes the signal directly in the time domain and requires no filter adaptation, as in LPC methods. The real-time implementation can take advantage of the sparsity of the velvet noise reducing the convolution to the playback of randomly triggered *voices*. The method requires an additional step of automatic gain control to reduce random fluctuations of the output signal energy. In the current work we describe a mechanism that is widely used in the literature, however, this may be improved upon taking in consideration both the velvet noise density and the fluctuations inherent to the input signal as well. Experimental results are provided showing the preservation of the original spectrum and the minimal effect of the window type, which can be, thus, selected depending on computational constraints. As a future work, subjective listening tests could be performed to compare it to other well-known methods. The quality of these effects is very subjective, thus, some audio semantic descriptors may be employed as well for the evaluation.

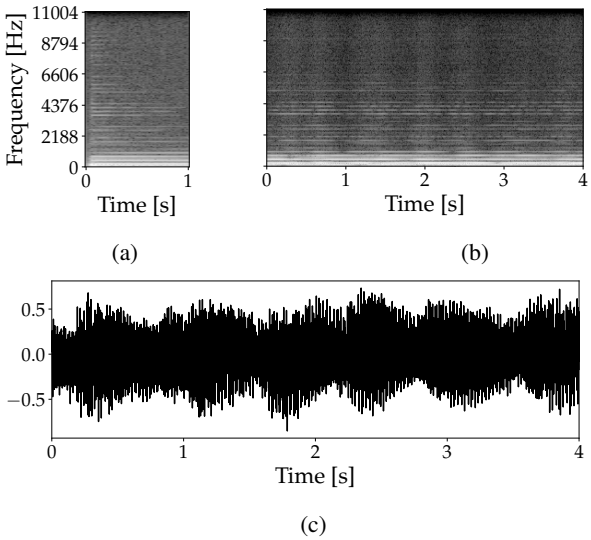

Figure 3: Extrapolation of a guitar chord: spectrogram of the original excerpt (a), spectrogram of the extrapolated audio (b) and waveform (c).

5. REFERENCES

[1] Ismo Kauppinen and Kari Roth, "Audio signal extrapolation–theory and applications," in *the Proc. of the DAFx Conference*, 2002, pp. 105–110.

[2] Diemo Schwarz, "State of the art in sound texture synthesis," in *the Proc. of the DAFx Conference*, 2011.

[3] Jim R. Parker and Brad Behm, "Creating audio textures by example: tiling and stitching," in *the Proc. of the DAFx Conference*, 2004.

[4] Martin Frojd and Andrew Horner, "Fast sound texture synthesis using overlap-add," in *the Proc. of the 2007 International Computer Music Conference, Copenhagen, Denmark*, 2007.

[5] Miller Puckette, *The Theory and Technique of Electronic Music*, World Scientific Press, 2007.

[6] Florian Keiler, Daniel Arfib, and Udo Zölzer, "Efficient linear prediction for digital audio effects," in *the Proc. of the DAFx Conference*, 2000.

[7] Xinglei Zhu and Lonce Wyse, "Sound texture modeling and time-frequency lpc," in *the Proc. of the DAFx Conference*, 2004, vol. 4.

[8] Udo Zölzer, *DAFX-Digital Audio Effects*, John Wiley and Sons, 2011.

[9] Marios Athineos and Daniel PW Ellis, "Sound texture modelling with linear prediction in both time and frequency domains," in *the Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03)*. IEEE, 2003, vol. 5, pp. V–648.

[10] Simon Haykin, *Adaptive Filter Theory*, Prentice Hall, Englewood Cliffs, NJ, USA, second edition, 1991.

[11] Matti Karjalainen and Hanna Järveläinen, "Reverberation modeling using velvet noise," in *the Proc. of the 30th International Conference on Intelligent Audio Environments*, 2007.

[12] Vesa Välimäki, Heidi-Maria Lehtonen, and Marko Takanen, "A perceptual study on velvet noise and its variants at different pulse densities," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 7, pp. 1481–1488, 2013.

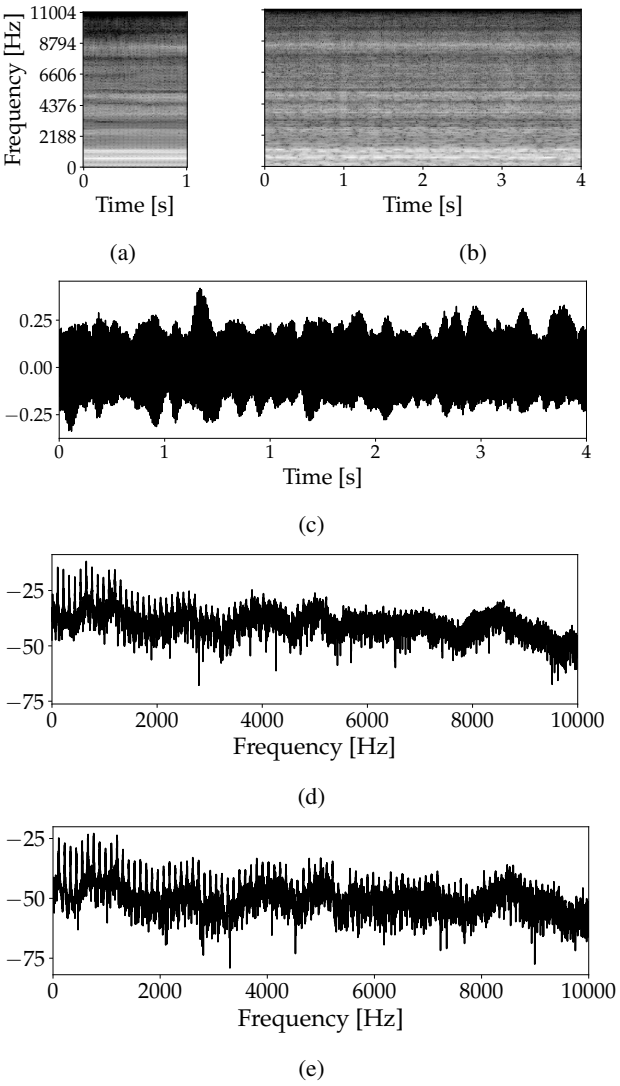

Figure 4: Experiments with voice extrapolation. The input signal is an excerpt of an /a/ phoneme by a male singer tuned to A2. Its spectrogram is shown in (a) and the resulting extrapolated signal is shown in (b), where the impulse density is set to 32 pulses per second. The original phoneme length was 1 s, while the extrapolated signal lasts 5 s. The time domain plot of the extrapolated signal is shown in (c), while the DFTs for the original and the extrapolated signals are respectively shown in (d-e). All signals are sampled at 44100 Hz.

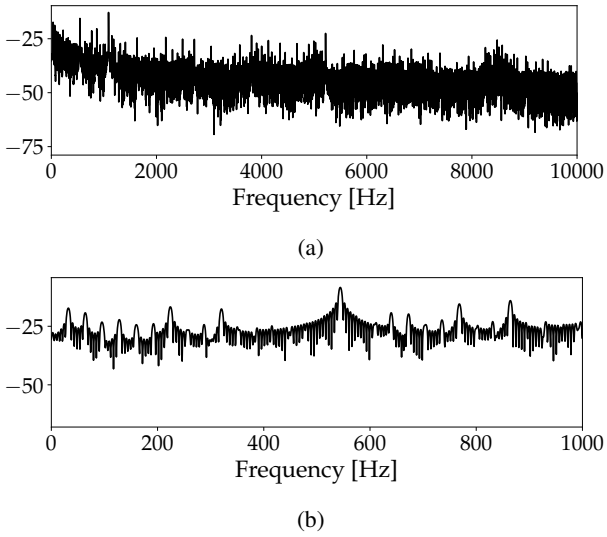

Figure 5: Repeating the experiment of Figure 4 with an impulse train instead of velvet noise with impulse density 32. The DFT is shown in (a). A detailed view shows that the periodicity can be clearly seen by the peaks emerging at multiples of 32 Hz, reducing the effect to a comb filter with unitary gain. The sampling rate is 4100 Hz.

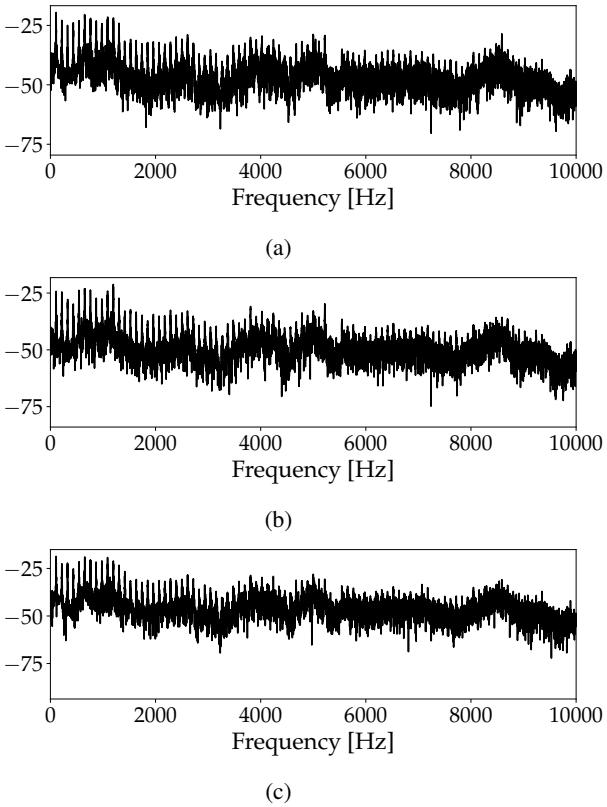

Figure 6: Comparison between signals extrapolated from the vocal signal in Figure 4(a) with window duration of 0.3 s and different window types: triangular (a), half sine (b) and Welch (c). All signals were generated using a grain density of 32 simultaneous grains on average. All signals are sampled at 44100 Hz.

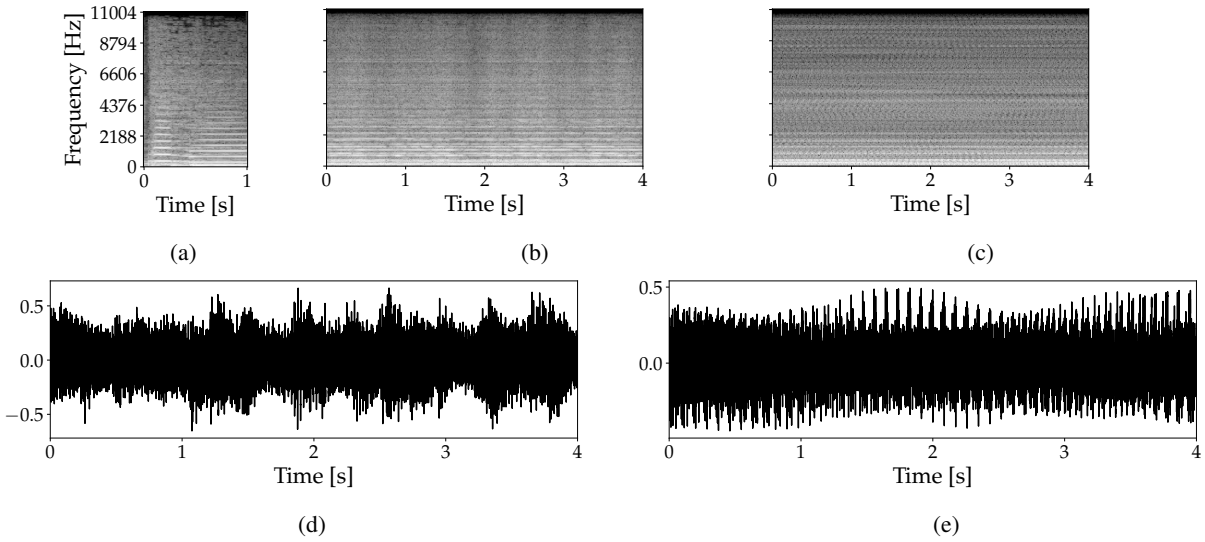

Figure 7: Extrapolation of jazz polyphonic music: spectrogram of the original excerpt (a), spectrogram of the extrapolated audio using the proposed method (b) and a FFT-based method (c). The time waveform are the one from the proposed method (d) and from the FFT-based method (e). The FFT-based method and the proposed extrapolation method use same window size. All signals are sampled at 44100 Hz.

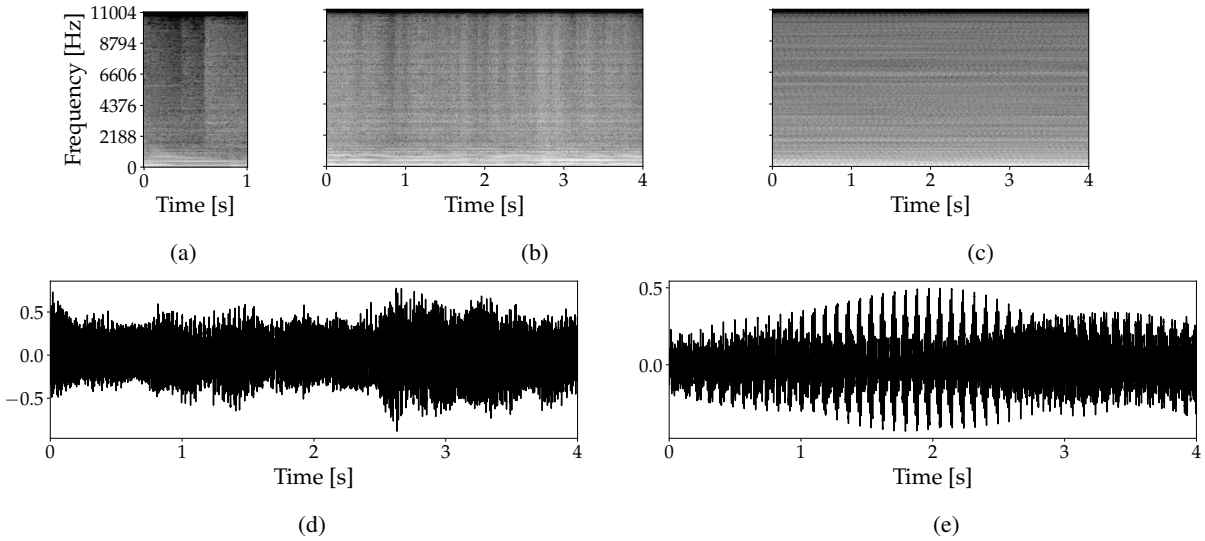

Figure 8: Extrapolation of a jazz excerpt with double bass and cymbals: spectrogram of the original excerpt (a), spectrogram of the extrapolated audio using the proposed method (b) and a FFT-based method (c). The time waveform are the one from the proposed method (d) and from the FFT-based method (e). The FFT-based method and the proposed extrapolation method use same window size. All signals are sampled at 44100 Hz.