PROCEEDINGS of the
23rd International Congress on Acoustics

9 to 13 September 2019 in Aachen, Germany

Sparse modeling of musical instruments sounds in time-frequency domain

Hiromu Ogi$^{(1)}$, Kohei Yatabe$^{(2)}$, Yasuhiro Oikawa$^{(3)}$, Yusuke Miyagi$^{(4)}$, Koji Oishi$^{(5)}$

$^{(1)}$Department of Intermedia Art and Science, Waseda University, Japan, hiromu.ogi.6626@ruri.waseda.jp

$^{(2)}$Department of Intermedia Art and Science, Waseda University, Japan, k.yatabe@asagi.waseda.jp

$^{(3)}$Department of Intermedia Art and Science, Waseda University, Japan, yoikawa@waseda.jp

$^{(4)}$KORG INC., Japan, miyagi@korg.co.jp

$^{(5)}$KORG INC., Japan, koji@korg.co.jp

Abstract

Sample-based synthesis is one of the most common synthesis methods of the digital synthesizer. It can synthesize musical instrument sounds well because it plays recorded sounds instead of generating sounds such as additive and frequency modulation syntheses. The recorded sounds are usually stored in the time domain. If the recorded sounds are stored in the time-frequency domain, sound processing can be simplified. Users can process audio signals intuitively by operating the time-frequency bins of the spectrograms. However, the representation of sounds in the time-frequency domain usually requires more memory than that in the time domain. In synthesizers, if much data is needed, hardware costs are increased correspondingly. Therefore, a technique to reduce the amount of data is required. In this paper, to reduce the amount of data, we introduce a sparse modeling technique for musical instrument sounds in the time-frequency domain. We propose an optimization algorithm of sparse modeling with four shrinkage operators. Numerical experiments show that the data quantity of the signals reduced by over 95 percent by using the technique.

Keywords: Sample-based synthesis, Spectrogram, Sparse modeling, Alternating direction methods of multipliers (ADMM)

1 INTRODUCTION

There are many synthesis methods of digital synthesizers, such as additive synthesis, subtractive synthesis, wavetable synthesis, frequency modulation synthesis, and sample-based synthesis [1,2]. Especially, sample-based synthesis is a powerful method when synthesizers imitate traditional musical instrument sounds like piano, violin or trumpet. The synthesizers of this type have read-only memory (ROM) in which the recorded sounds are stored, and synthesize sounds by playing them back. Generally, the recorded sounds are stored in the time domain.

If the recorded sounds are stored in the time-frequency domain, sound processing can be simplified. Users can process audio signals intuitively by operating the time-frequency bins of the spectrograms because time-frequency bins have information for a certain frequency band in a certain time. However, the representation of sounds in the time-frequency domain usually requires more memory than that in the time domain. In synthesizers, if much data is needed, hardware costs are increased correspondingly. Therefore, a technique to reduce the amount of data without sacrificing quality is required.

It is expected that the time-frequency bins necessary to represent the musical instrument sounds exist sparsely since sounds have harmonic structure. If it is possible to express the musical instrument sound sparsely, the reduction of the amount of data is realized.

One of the effective tools for providing sparse representation is sparse modeling. This approach is widely used in signal processing, image processing, and machine learning [3]. For effective modeling, it is necessary to consider the characteristics of the signals and its transformation. Thus, we focus on that the window function extends spectral peaks and introduce "social sparsity [4]" to the sparse modeling for musical instrument sounds.

In this paper, we propose some algorithms of sparse modeling for musical instrument sounds in the time-frequency domain. We also found that these algorithms are effective in the viewpoint of the amount of data compared with the usual time domain representation.

2 PRELIMINARIES

2.1 Time-frequency Representation

The short-time Fourier transform (STFT) is one of the most popular methods of the time-frequency representation [5]. The discrete STFT is given by

$$X[n,m] = \textstyle\sum_t w_a[t-\Delta n]x[t]e^{-\mathrm{i}2\pi(t-\Delta n)m/L} \ , \tag{1}$$

where $x[t]$ is the time domain signal, w_a is an analysis window, L is window length, Δ is a window shifting step parameter, and i is the imaginary unit. Inverse transformation of STFT (iSTFT) is also given by

$$\tilde{x}[t] = \textstyle\sum_m w_s[t-\Delta n]\left(\dfrac{1}{L}\displaystyle\sum_{m=0}^{L-1} X[n,m]e^{\mathrm{i}2\pi(t-\Delta n)m/L}\right) \ , \tag{2}$$

where w_s is a synthesis window [6]. In STFT, one sample point in the time domain is analyzed by $Q = L/\Delta$ windows. As a result, the time-frequency representation has Q times as many time-frequency bins as the sample points of the time domain representation. In other words, STFT usually provides redundant expression. If the signal is expressed redundantly, it is possible to represent it in various ways. Thus, we propose algorithms to find the representation where the number of zeros is maximal (i.e., sparsest) in the time-frequency representation.

2.2 Sparse Modeling

Sparse modeling is an effective method to solve underdetermined problems, and it is possible to reduce the amount of data significantly when signals admit sparse representation [3]. Studies based on the sparsity is widely conducted in acoustic engineering, including sound source separation [7–10], speech enhancement [11, 12] and restoration of the sound field [13–15].

Assuming that the musical instruments sound is $x[t]$ and the observed musical instrument sound including noise is $\hat{x}[t]$, the time-frequency representation of $x[t]$ is given by

$$\text{Find} \quad X \quad \text{s.t.} \quad \|\mathscr{F}^* X - \hat{x}\|_\infty \le \varepsilon \ , \tag{3}$$

where $\|y\|_\infty = \max_i |y_i|$, $\mathscr{F}^*$ is a operator which indicates iSTFT, and ε is a small constant which indicates the tolerance of the error between $x[t]$ and $\hat{x}[t]$. This equation is an underdetermined system, hence there are innumerable solutions to it.

To get the sparsest solution, we should solve the optimization problem:

$$\min_{X} \iota_C(X) + \lambda\Lambda(X) \,, \tag{4}$$

$$\iota_C(X) = \begin{cases} 0 & (X \in C) \\ \infty & (\text{otherwise}) \end{cases} , \quad C = \{X \in \mathbb{C}^{M \times N} \mid \|\mathscr{F}^* X - b\|_\infty \leq \varepsilon\} \,. \tag{5}$$

The ideal function as $\Lambda(X)$ giving a sparse solution is ℓ_0 norm defined as a function representing the number of nonzero elements of X [16]. However, it is difficult to find the optimal solution of this minimization problem because it is combinational optimization of high-dimensional data. Therefore, the ℓ_0 norm is often relaxed with the convex function, ℓ_1 norm, to find the optimal solution [17–20].

2.3 Alternating Direction Methods of Multipliers (ADMM)

There are many algorithms for optimization problems. One of them is the alternating direction methods of multipliers (ADMM) [21]. Many optimization problems including the sparse modeling, can be interpreted as the following minimization problem:

$$\min \; f(x) + g(z) \quad \text{s.t.} \quad Kx + Lz = c \,, \tag{6}$$

where f and g are real-valued cost functions, $x \in \mathbb{R}^N$, $z \in \mathbb{R}^N$, $K \in \mathbb{R}^{S \times N}$, $L \in \mathbb{R}^{S \times N}$, and $c \in \mathbb{R}^S$. ADMM can solve the above minimization problem, and it is written as the following procedure:

$$x^{[n+1]} = \operatorname*{argmin}_{x} \left[f(x) + \frac{\rho}{2} \|Kx + Lz^{[n]} - c + u^{[n]}\|_2^2 \right] \tag{7}$$

$$g^{[n+1]} = \operatorname*{argmin}_{z} \left[g(z) + \frac{\rho}{2} \|Kx^{[n+1]} + Lz - c + u^{[n]}\|_2^2 \right] \tag{8}$$

$$u^{[n+1]} = u^{[n]} + Kx^{[n+1]} + Lz^{[n+1]} - c \,, \tag{9}$$

where ρ is a positive constant. The strengths of the ADMM is that it can be applied even if the cost function is not differentiable. By this fact, we can solve the optimization problem which has non-differentiable functions such as ℓ_1 norm and ℓ_∞ norm. The effectiveness of ADMM has been substantiate by many studies [22, 23].

3 PROPOSED METHODS

3.1 Proposed Algorithm

Applying ADMM to Eq. (4), it is formulated as

$$f(x) = \iota_C(x), \quad g(z) = \lambda\Lambda(z), \quad K = I, \quad L = -I, \quad c = 0 \,, \tag{10}$$

where I is the identity matrix. Using Eq. (7) to Eq. (9), an algorithm for the sparse modeling is derived, and it is summarized in Algorithm 1. Here, sign(y) is the signum function,

$$\text{sign}(y) = \begin{cases} y/|y| & (y \neq 0) \\ 0 & (y = 0) \end{cases} , \tag{11}$$

Algorithm 1 Proposed Algorithm

```
Initialization: $u^{[0]} \in \mathbb{C}^{M \times N}$,
repeat
    $x' \leftarrow \mathscr{F}^{*}(z^{[k]} - u^{[k]})$ ;
    $e \leftarrow x' - b$ ;
    $e \leftarrow \mathrm{sign}(e) \min\{|e|,\ \epsilon\}$;
    $x^{[k+1]} \leftarrow (z^{[k]} - u^{[k]}) + \mathscr{F}(b + e - x')$ ;
    $z^{[k+1]} \leftarrow \mathcal{P}.(x^{[k]} + u^{[k]})$ ;
    $u^{[k+1]} \leftarrow u^{[k]} + x^{[k+1]} - z^{[k+1]}$ ;
    $k \leftarrow k + 1$ ;
until convergence;
```

$\mathscr{F}$ is a operator which indicates STFT, and $\mathcal{P}.(v)$ indicates the proximity operator of $\lambda\Lambda(X)$. The proximity operator [24] is defined by

$$\mathcal{P}.(v) = \mathrm{prox}_{\lambda\Lambda}(v) = \underset{z}{\mathrm{argmin}} \left[\lambda\Lambda(z) + \frac{1}{2}\|z - v\|_2^2 \right] . \tag{12}$$

3.2 Thresholding and Shrinkage

3.2.1 Soft-Thresholding

Let $\Lambda(X)$ be a ℓ_1 norm, $\mathcal{P}_{\ell_1}$ is given by

$$[\mathcal{P}_{\ell_1}(z)][n] = \max\left\{1 - \lambda/|z[n]|\ ,\ 0\right\} z[n]\ , \tag{13}$$

and it is known as a soft-thresholding operator. ℓ_1 norm is used in various optimization problem since it is a convex function [17–20].

3.2.2 Log-Thresholding

When $\Lambda(X)$ is ℓ_1 norm, if there are large time-frequency bins in X, it is expected that these are reduced by the optimization problem Eq. (4). For this reason, it has been proposed relaxing the effect of ℓ_1 norm using a logarithmic function [25, 26].

Let $\Lambda(X)$ be

$$\Lambda(X) = \log(\delta + |X|)\ , \tag{14}$$

where δ is a small positive constant, $\mathcal{P}_{\log}$ is given by

$$[\mathcal{P}_{\log}(z)][n] = \begin{cases} \frac{1}{2}\left((z[n] - \delta) + \sqrt{(z[n] + \delta)^2 - 2\lambda}\right) & (z[n] > \sqrt{2\lambda} - \delta) \\ \frac{1}{2}\left((z[n] + \delta) - \sqrt{(z[n] + \delta)^2 - 2\lambda}\right) & (z[n] < \sqrt{2\lambda} - \delta) \\ 0 & (\text{otherwise}) \end{cases} . \tag{15}$$

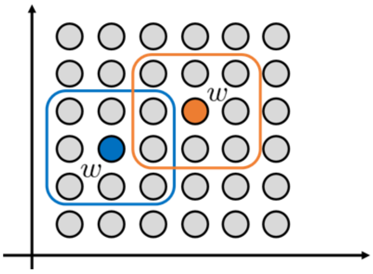

Figure 1. Schematic diagram of social shrinkage

3.2.3 Social Shrinkage

In the time-frequency representation of instrumental sound, it is considered that energy is concentrated in the grouped part of the spectrograms because the window function extends spectral peaks. Hence, we propose to combine social sparsity [4, 27, 28]. This is sparse modeling in which the information of surrounding time-frequency bins is added as a weight to take a decision: keeping or discarding the bin. The effects of this modeling are expected that the time-frequency bins of the spectrogram become grouped and noise which is not related to the signal is reduced. A schematic diagram of social sparsity is shown in Fig. 1. Introducing the social sparsity, $\mathcal{P}_{\text{social}}$ is given by

$$[\mathcal{P}_{\text{social}}](z)[n] = \max\{1 - \lambda / V^{w}[n],\ 0\}\ z[n]\ , \tag{16}$$

$$V^{w}[n] = \sqrt{w * |z[n]|^2}\ , \tag{17}$$

where w is a non-negative kernel, and $*$ indicates the convolution operation. Compared to the Eq. (13), Eq. (16) can be regarded as a kind of soft-thresholding.

3.2.4 Social-Log Shrinkage

Finally, we propose combining social sparsity and log-thresholding. In Eq. (15), $\mathcal{P}_{\text{social-log}}$ can be determined by replacing $z[n]$ with $V^{w}[n]$, and $\mathcal{P}_{\text{social-log}}$ is given by

$$[\mathcal{P}_{\text{social-log}}(z)][n] = \begin{cases} \frac{1}{2}\left((V^{w}[n] - \delta) + \sqrt{(V^{w}[n] + \delta)^2 - 2\lambda}\right) & (V^{w}[n] > \sqrt{2\lambda} - \delta) \\ \frac{1}{2}\left((V^{w}[n] + \delta) - \sqrt{(V^{w}[n] + \delta)^2 - 2\lambda}\right) & (V^{w}[n] < \sqrt{2\lambda} - \delta) \\ 0 & (\text{otherwise}) \end{cases} . \tag{18}$$

By combining social sparsity and log-thresholding, it is expected that non-zero elements of spectrograms can be grouped while suppressing the effects of bias.

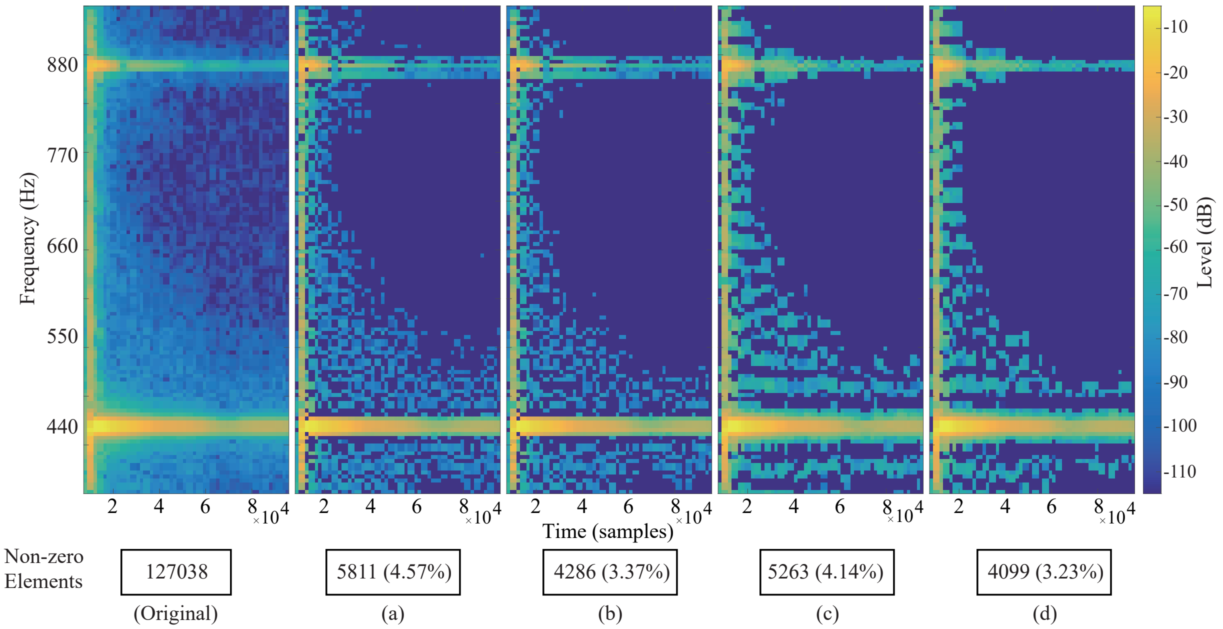

Figure 2. Amplitude spectrograms and numbers of non-zero elements

4 EXPERIMENTS

Using the proposed method, we modeled the recorded piano sound to be a sparse representation in the time-frequency domain. In the simulation, the target sound was piano sound A4 (442 Hz), the window function was Hanning window, L was 2^{12} sample, Δ was 2^{11} sample, λ was 0.1, ε was 3.92×10^{-7}, w was a Gaussian kernel with a size of 7 rows and 3 columns, and iteration number was 3000. For comparison, we performed sparse modeling in four cases: (a) soft-thresholding, (b) log-thresholding, (c) social shrinkage, and (d) social-log shrinkage. An expanded view of the amplitude spectrograms and the number of non-zero elements of the spectrograms are shown in Fig. 2.

From the results, it was confirmed that a large amount of data was reduced by sparse modeling. In (a) and (b), non-zero elements were scattered, while in (c) and (d), non-zero elements were collectively present due to the effect of social sparsity. In addition, we found that log-thresholding is able to reduce more non-zero elements.

5 CONCLUSION

In this paper, we propose a method to represent instrument sounds sparsely in the time-frequency domain. The simulation shows that it is possible to reduce most of the data by sparse modeling, and the combination of log-thresholding and social sparsity is effective. In future works, we plan to consider the implementation of high-speed iSTFT using sparsity so that the synthesizer can sound in real time even if sounds are contained as time-frequency representation.

REFERENCES

[1] M. V. Mathews, F. R. Moore, and J. C. Risset, "Computers and future music," Science **183**, 263–268 (1974).

[2] H. F. Olson and H. Belar, "Electronic music synthesizer," The Journal of the Acoustical Society of America **27**, 595–612 (1955).

[3] A. Bruckstein, D. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," SIAM Review **51**, 34–81 (2009).

[4] M. Kowalski, K. Siedenburg, and M. Dörfler, "Social sparsity! neighborhood systems enrich structured shrinkage operators," IEEE Trans. Signal Processing **61**, 2498–2511 (2013).

[5] K. Yatabe, Y. Masuyama, T. Kusano, and Y. Oikawa, "Representation of complex spectrogram via phase conversion," Acoustical Science and Technology **40**, 170–177 (2019).

[6] T. Kusano, Y. Masuyama, K. Yatabe, and Y. Oikawa, "Designing nearly tight window for improving time-frequency masking," arXiv e-prints arXiv:1811.08783 (2018).

[7] T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," IEEE Transactions on Audio, Speech, and Language Processing **15**, 1066–1074 (2007).

[8] H. Kameoka, M. Nakano, K. Ochiai, Y. Imoto, K. Kashino, and S. Sagayama, "Constrained and regularized variants of non-negative matrix factorization incorporating music-specific constraints," in "2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)," (2012), pp. 5365–5368.

[9] Y. Masuyama, K. Yatabe, and Y. Oikawa, "Phase-aware harmonic/percussive source separation via convex optimization," in "ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)," (2019), pp. 985–989.

[10] A. Hiruma, K. Yatabe, and Y. Oikawa, "Separating stereo audio mixture having no phase difference by convex clustering and disjointness map," in "International Workshop on Acoustic Signal Enhancement (IWAENC)," (2018), pp. 266–270.

[11] C. D. Sigg, T. Dikk, and J. M. Buhmann, "Speech enhancement with sparse coding in learned dictionaries," in "2010 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)," (2010), pp. 4758–4761.

[12] J. F. Gemmeke, T. Virtanen, and A. Hurmalainen, "Exemplar-based sparse representations for noise robust automatic speech recognition," IEEE Transactions on Audio, Speech, and Language Processing **19**, 2067–2080 (2011).

[13] K. Yatabe and Y. Oikawa, "Optically visualized sound field reconstruction based on sparse selection of point sound sources," in "2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)," (2015), pp. 504–508.

[14] T. Tachikawa, K. Yatabe, and Y. Oikawa, “3d sound source localization based on coherence-adjusted monopole dictionary and modified convex clusterin,” Applied Acoustics **139**, 267–281 (2018).

[15] K. Kobayashi, D. Takeuchi, M. Iwamoto, K. Yatabe, and Y. Oikawa, “Parametric approximation of piano sound based on kautz model with sparse linear prediction,” in “2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),” (2018), pp. 626–630.

[16] G. Su, J. Jin, Y. Gu, and J. Wang, “Performance analysis of l_0 norm constraint least mean square algorithm,” IEEE Transactions on Signal Processing **60**, 2223–2235 (2012).

[17] S. Chen and D. Donoho, “Basis pursuit,” in “Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers,” , vol. 1 (1994), vol. 1, pp. 41–44 vol.1.

[18] S. Chen, D. Donoho, and M. Saunders, “Atomic decomposition by basis pursuit,” SIAM Review **43**, 129–159 (2001).

[19] D. L. Donoho, “Compressed sensing,” IEEE Transactions on Information Theory **52**, 1289–1306 (2006).

[20] E. J. Candes and T. Tao, “Near-optimal signal recovery from random projections: Universal encoding strategies?” IEEE Transactions on Information Theory **52**, 5406–5425 (2006).

[21] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” Found. Trends Mach. Learn. **3**, 1–122 (2011).

[22] S. Diamond, R. Takapoui, and S. Boyd, “A general system for heuristic minimization of convex functions over non-convex sets,” Optimization Methods and Software **33**, 165–193 (2018).

[23] Y. Masuyama, K. Yatabe, and Y. Oikawa, “Griffin–Lim like phase recovery via alternating direction method of multipliers,” IEEE Signal Processing Letters **26**, 184–188 (2019).

[24] N. Parikh and S. Boyd, “Proximal algorithms,” Found. Trends Optim. **1**, 127–239 (2014).

[25] M. Fazel, H. Hindi, and S. P. Boyd, “A rank minimization heuristic with application to minimum order system approximation,” in “Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148),” , vol. 6 (2001), vol. 6, pp. 4734–4739 vol.6.

[26] D. Malioutov and A. Aravkin, “Iterative log thresholding,” in “2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),” (2014), pp. 7198–7202.

[27] M. Kowalski and B. Torrésani, “Structured sparsity: from mixed norms to structured shrinkage,” Proceeding of Signal Processing with Adaptive Sparse Structured Representations (SPARS) (2009).

[28] K. Siedenburg, M. Kowalski, and M. Dörfler, “Audio declipping with social sparsity,” in “2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),” (2014), pp. 1577–1581.