Parameter Estimation of Virtual Musical Instrument Synthesizers

Katsutoshi Itoyama
Kyoto University
itoyama@kuis.kyoto-u.ac.jp

Hiroshi G. Okuno
Kyoto University
okuno@kuis.kyoto-u.ac.jp

ABSTRACT

A method has been developed for estimating the parameters of virtual musical instrument synthesizers to obtain isolated instrument sounds without distortion and noise. First, a number of instrument sounds are generated from randomly generated parameters of a synthesizer. Low-level acoustic features and their delta features are extracted for each time frame and accumulated into statistics. Multiple linear regression is used to model the relationship between the acoustic features and instrument parameters. Experimental evaluations showed that the proposed method estimated parameters with a best case error of 0.004 and signal-to-distortion ratio of 17.35 dB, and reduced noise to smaller distortions in several cases.

1. INTRODUCTION

The demand for active music appreciation [1], which is symbolized by consumer generated media (CGM) and user generated content (UGC), has been increasing. A limited number of people have actively appreciated computer generated music for the past 30-40 years due to its requirements for specific technical knowledge, experience, and equipment. For example, musical composition and arrangement may require knowledge of musical structure and chord progression. A person must have adequate training to enjoy playing an instrument. Typically, only musical experts can actively appreciate music. One of the main CGM activities is imitation and improvement of past work. Sound source separation [2–6] is an important basic technique for CGM. These sound source separation methods separate audio mixtures into sources at a good level of accuracy under limited conditions. However, separated sources are generally distorted and contain noise. These effects degrade the quality of CGM products.

We have developed an alternative way to obtain isolated instrument sounds without distortion from the input sound mixtures by using virtual instrument sound synthesizers. Various virtual instrument sound synthesizers, such as musical instrument digital interface (MIDI) synthesizers and virtual studio technology (VST) instruments, have been developed and used to compose musical pieces. A wide variety of musical instruments have been implemented, e.g., acoustic instruments such as pianofortes, guitars, and violins, and electric and electronic instruments such as analog synthesizers and theremins. If we could collect every virtual instrument sound synthesizer in the world, some would produce sounds sufficiently similar to the original sound sources without any distortion or noise in principle. An overview of the proposed method is shown in Fig. 1.

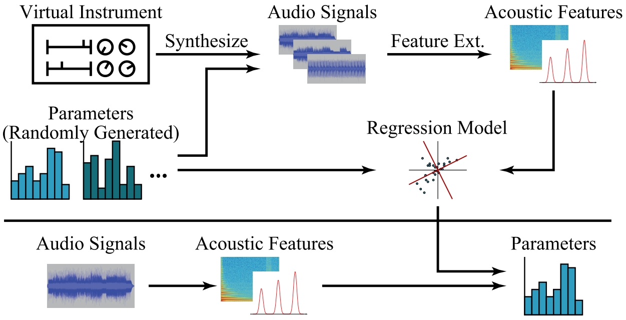

Figure 1. Overview of proposed method.

Related work includes analysis and synthesis methods that use physical modeling of musical instruments, e.g., plucked strings [7, 8] and bowed strings [9]. These methods explicitly model physical phenomena such as string vibration and estimate the physical parameters from input sounds without noise and distortion. Similarly, VocaListener [10, 11] estimates the parameters of Vocaloid, a singing voice synthesizer. Using the relationship between several parameters and the pitch and volume, VocaListener iteratively estimates the optimal parameters for the input singing voice without noise or distortion.

Our method has two unique features.

1. It can deal with arbitrary virtual instrument synthesizers. That is, the relationships between the instrument parameters and the audio signals are unknown.

2. It can estimate the parameters of an instrument's sound without distortion or noise even if the input sounds have distortion due to source separation.

The proposed method has two basic steps.

1. Acoustic feature extraction. The low-level acoustic features are extracted from each time frame, the delta features and gradients of the approximated lines are calculated, and the statistics, including the mean and variation, are calculated for each dimension.

2. Model training. The coefficients of the multiple linear regression model between the acoustic features and instrument parameters are iteratively estimated under the assumption that the parameters are in the acoustic feature space.

2. ACOUSTIC FEATURES

The extraction of the acoustic features comprises four steps.

1. Framewise feature extraction: calculates low-level features for each time frame from the instrument sounds.
2. Time differentiation: differentiates low-level features and obtains delta features.
3. Accumulation: accumulates the features across frames and obtains the fixed-dimension features for each instrument sound.
4. Compression: reduces the feature's dimensions by using principal component analysis (PCA).

2.1 Extraction of Low-level Features

Acoustic features that represent the timbre of instrument sounds were designed on the basis of previous studies on instrument identification and musical mood detection [12,13]. Input instrument sound signals are segmented into overlapping short-time frames. Features are extracted from the segmented signals, and magnitude spectra are calculated using a short-time Fourier transform. A number of low-level features are extracted.

Root-mean-square (RMS) Overall energy of the signal.

Low energy Degree of energy concentration in the low-frequency band.

Zero-crossing rate Intensity ratio between harmonic and percussive components.

Spectral centroid Centroid of the short-time Fourier transform spectrum.

Spectral width Amplitude weighted average of the differences between the spectral components and the centroid.

Spectral rolloff 95th percentile of the spectral distribution.

Spectral flux 2-norm distance of the frame-to-frame spectral amplitude difference.

Spectral peak The largest amplitude values in the spectrum.

Spectral valley The smallest amplitude values in the spectrum.

Spectral contrast The difference between the peak and valley.

Mel-frequency cepstrum coefficients (MFCCs) Overall timbre of the sounds. We use 12-dimensional MFCCs.

Harmonic amplitudes Timbre of the harmonic components. We use the first to tenth harmonic components. This feature is extracted using PreFEst [14].

The dimension of the low-level feature vectors is 32.

The low-level feature vectors can represent the instantaneous characteristics of the instrument sounds but not their time variation. We use three kinds of time derivatives of the features to represent the time variation: the delta of adjacent frames, the gradient of the line approximated during the last 50 ms, and the gradient of the line during the last 100 ms. Additionally, three second derivatives are calculated in the same way. As a result, we obtain $32 \times (1 + 3 + 3) = 224$ dimensional framewise feature vectors.

2.2 Accumulation and Compression

The set of framewise feature vectors extracted from each instrument sound contains an inconsistent dimension because the sound durations are inconsistent. The dimensions of the feature vectors must be equal to train the regression model. Thus, we accumulate the feature vectors across the time frames into various statistics to make the dimensions uniform.

Twenty-five statistical values are calculated for each dimension of the feature vectors:

1. Summation, mean, variance, skewness, and kurtosis. These statistics represent the characteristics of the distribution of the feature vectors.
2. Minimum, maximum, median, 10th and 90th percentiles, and their positions (time). These statistics represent another characteristic of the distribution of the feature vectors and their temporal structure.
3. Bottom 10 coefficients of discrete cosine transform. These statistics represent the temporal changes of the feature vectors.

The characteristics of the instrument sounds vary in the temporal region, e.g., attack, decay, sustain, and release. We thus calculate the statistics in three temporal regions: the entire interval, during excitation (i.e., MIDI note-on to note-off), and during reverberation (MIDI note-off to silence). In addition, we segment each temporal region into subregions: beginning to end, begining to {20, 40, 60, and 80} percent points, {20, 40, 60, and 80} percent points to end, {200, 400, 600, 800, and 1000} ms from the beginning, and {200, 400, 600, 800, and 1000} ms until end (see Fig. 2).

We thereby obtain $224 \times 3 \times 19 \times 25 = 319,200$ dimensional feature vectors for each instrument sound. Although the regression model can be trained even as it is, we apply PCA to reduce the dimension of the feature vectors and computational costs for estimating regression model

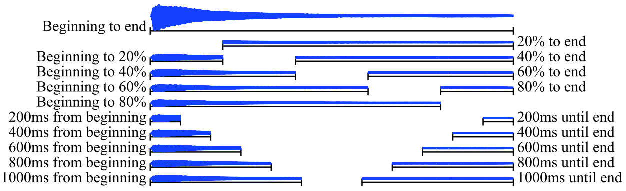

Figure 2. 19 temporal subregions.

parameters. The dimension of the compressed feature vectors depended on the number of parameters of the virtual instrument, which is roughly between 100 and 1000.

3. REGRESSION MODEL

3.1 Parameters of Virtual Instrument

Virtual musical instruments, such as MIDI synthesizers and VST instruments, have various parameters that are both dependent and independent of the instruments. Each parameter is treated as a scalar value within a given range, such as 0–127 (MIDI) and 0–1 (VST). We assume that the ranges of all the instrument parameters are normalized to 0–1 in this paper.

The parameters are divided into two types:

1. Continuous parameters. These parameters continuously affect the generated instrument sounds, e.g., the volume and reverberation.

2. Selective parameters. These parameters have a discrete effect on the sounds, such as the kind of wave oscillation (sinusoidal, triangle, sawtooth, square, etc.). The range of a parameter is segmented into sub-ranges to enable a discrete value to be specified from the set.

We assume that the instrument parameters have a linear relationship with the acoustic features, but the selective parameters cannot be treated in a linear model. Therefore, we convert the selective parameters to extended ones that are suitable for a linear regression model.

Original parameters to extended ones Increase the dimensions of the parameters to the number of parameter options. Each option can then be described as a 1-of-K representation.

Extended parameters to original ones The original parameter corresponds to the maximum value of the extended parameters. For example, $(1, 0, 0, 0)$ is converted to sinusoidal wave oscillation, and $(0.3, 0.5, 0.8, 0.2)$ is converted to sawtooth wave oscillation.

3.2 Model Training

A multiple linear regression model is used to represent the relationship between the extended instrument parameters and the acoustic features. Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ be the acoustic features, and let $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$ be the extended parameters. A matrix of regression coefficients A and intercept vector $\boldsymbol{b}$ are used to define the relationship

$$\boldsymbol{y} = A\boldsymbol{x} + \boldsymbol{b}. \qquad (1)$$

The parameters should be orthogonal in the acoustic feature space for precise parameter control. This orthogonality is achieved by minimizing the sum of the dot products between each pair of the row vectors of A.

Optimal coefficient matrix and vector are obtained by minimizing the objective

$$\sum_{i=1}^{n} \|\boldsymbol{y}_n - A\boldsymbol{x}_n - \boldsymbol{b}\|^2 + \lambda \sum_{i \neq j} \boldsymbol{a}_i \cdot \boldsymbol{a}_j, \qquad (2)$$

where $\|\boldsymbol{x}\|^2$ and $\boldsymbol{x} \cdot \boldsymbol{y}$ represent the Frobenius norm and dot product, respectively, λ is a constant that represents the effect of the orthogonality, and $\boldsymbol{a}_i$ is the i-th row vector of A.

By minimizing the objective function for each row vector, we obtain the update equations

$$a_{km} = \frac{\sum_n y_{nk} x_{nm} - \sum_{m' \neq m} a_{km'} \sum_n x_{nm} x_{nm'}}{\sum_n x_{nm}^2 + \lambda \sum_{k' \neq k} a_{k'm}} \quad \text{and} \qquad (3)$$

$$b_m = \frac{\sum_n x_{nm} - \sum_{m' \neq m} b_{m'} \sum_n x_{nm} x_{nm'}}{\sum_n x_{nm}^2}. \qquad (4)$$

The optimal coefficients are calculated by iteratively applying the update equations.

4. EXPERIMENTAL EVALUATION

We conducted two experiments to evaluate the proposed method. In the first experiment, the effect of the number of parameters on the accuracy of parameter estimation was examined. The number of parameters to be estimated was chosen between one and ten. The unselected parameters remained at their default values. If an instrument has less than 10 parameters and the number of parameters to be estimated exceeds it, the estimation procedure is omitted. In the second experiment, the robustness of the proposed method against noise was evaluated. We added white noise to the sounds. The signal-to-noise ratio (SNR) was chosen between -20 and 20 dB with 10 dB increments. The number of parameters to be estimated was fixed to one. The size of training data was chosen from 1000, 2000, . . . , and 10000 for each experiment. The fundamental frequency and duration of the sounds were fixed to 440 Hz and 0.8 s.

The experimental procedures are as follows. First, the set of the parameters to be estimated was randomly extracted for each instrument. Instrument sounds were synthesized from randomly generated parameters and divided into ten subsets for ten-fold cross-validation. The regression models were trained using the subsets. The parameters were estimated for each sound of the remaining subset, and the sounds were re-synthesized from the estimated parameters. Finally the estimation error of the parameters and signal-to-distortion ratio (SDR) [15] between the original and re-synthesized sounds was calculated.

Table 2. Classification of parameters.

Description	# of param.
Volume	20
Envelope (attack, decay, sustain, and release)	47
F0 (vibrato and modulation)	31
Filter and equalizer (e.g., cutoff freq.)	26
Reverberation and delay	23
Effects (e.g., chorus and distortion)	34
Low frequency oscillator	32
Others (e.g., type of oscillators)	69

The estimation error of the parameters is defined as:

$$e = \frac{e_{\mathrm{c}} + e_{\mathrm{s}}}{\text{number of parameters}},$$

$$e_{\mathrm{c}} = \sum_i |p_{\mathrm{est},i} - p_{\mathrm{ref},i}|, \quad \text{and}$$

$$e_{\mathrm{s}} = \sum_i \begin{cases} 0 & \text{if estimated parameter was correct} \\ 1 & \text{otherwise} \end{cases},$$

where e_{c} and e_{s} mean the estimation errors for continuous and selective parameters, respectively, because they must be calculated for each way. The $p_{\mathrm{est},i}$ and $p_{\mathrm{ref},i}$ are the estimated and randomly chosen parameters, respectively.

4.1 Training Data

The virtual instruments listed in Table 1 were used in the first experiment. In the second experiment, 4Front R-Piano, DSK Strings, and Synth1 were chosen from Table 1 because of the limitation of computational resources. Table 2 shows the classification of the parameters.

4.2 Results

The results of the first experiment are shown in Figure 3. We discuss several noted facts.

1. Increasing the size of the training data reduces the estimation error of the parameters and improved the SDR. This suggests the estimation error can be used as the timbre similarity of instrument sounds.

2. Increasing the size of the number of parameters degrades the estimation accuracy and SDR.

3. The accuracy has a large gap between the case of five parameters and the case of six parameters. On the other hand, the SDR has large gaps between one and two parameters, and between eight and nine parameters. It could be caused by diffusion of type of the parameters to be estimated.

The results of the second experiment are shown in Figure 4. They show that increasing the noise ratio decreases the estimation accuracy and SDR. However, except for the case of 20 dB of SNR, the SDRs of the re-synthesized sounds increased compared to the SNRs of the original ones.

Next, we discuss objective criteria of the parameter estimation error. The value range of the MIDI synthesizer parameters is generally 0 to 127 7-bit digits. By assuming that VSTi synthesizers operate in this way, any errors in the estimated parameters of less than 0.008, i.e., $1/128$, can be treated as zero. For example, the SDR in the best case was 0.030, i.e., $3.8/128$, which can be regarded as sufficiently accurate for practical purposes.

5. CONCLUSION

This paper describes a method for estimating the parameters of a virtual musical instrument synthesizer. Multiple linear regression is used to model the relationship between the acoustic features and instrument parameters. In the experimental evaluation, our method estimated accurate parameters under several conditions. Our future work includes further evaluation using other virtual instruments and other kinds of noise and distortions and achievement of noise robustness.

Acknowledgments

This study was partially supported by a Grant-in-Aid for Young Scientists (B) (No. 24700168) and a Grant-in-Aid for Scientific Research (S) (No. 24220006).

6. REFERENCES

[1] M. Goto, "Active music listening intefaces based on signal processing," in ICASSP2007, 2007, pp. 1441–1444.

[2] M. A. Casey and A. Westner, "Separation of mixed audio sources by independent subspace analysis," in ICMC2000, 2000, pp. 154–161.

[3] T. Virtanen and A. Klapuri, "Separation of harmonic sounds using linear models for the overtone series," in ICASSP2002, 2002, pp. 1757–1760.

[4] M. R. Every and J. E. Szymanski, "A spectral-filtering approach to music signal separation," in DAFx-04, 2004, pp. 197–200.

[5] J. Woodruff, B. Pardo, and R. Dannenberg, "Remixing stereo music with score-informed source separation," in ISMIR2006, 2006, pp. 314–319.

[6] H. Viste and G. Evangelista, "A method for separation of overlapping partials based on similarity of temporal envelopes in multichannel mixtures," IEEE Trans. Audio, Speech and Lang. Process., vol. 14, no. 3, pp. 1051–1061, May 2006.

[7] A. W. Y. Su and S.-F. Liang, "A class of physical modeling recurrent networks for analysis/synthesis of plucked string instruments," IEEE Trans. Neural Netw., vol. 13, no. 5, pp. 1137–1148, Sep. 2002.

[8] J. Riionheimo and V. Välimäki, "Parameter estimation of a plucked string synthesis model using

Table 1. Virtual instruments used for evaluation.

Name	Instrument	# of param.	URL
Balthor's Grand	Pianoforte	6	`http://vstantenna.info/dev/4/Balthor_Grand_piano_plugin`
Delay Lama	Chorus	4	`http://www.audionerdz.com/`
DSK Brass	Brass	79	`http://www.dskmusic.com/dsk-brass/`
DVS Guitar	Electric guitar	8	`http://www.dreamvortex.co.uk/instruments/`
FAMISYNTH-II	Chiptune	16	`http://www.geocities.jp/mu_station/vstlabo/famisynth.html`
GTG FM4	FM synthesizer	45	`http://www.gtgsynths.com/`
MrTramp2	Electric piano	5	`http://www.genuinesoundware.com/?a=showproduct&b=40`
Neon	Subtractive synthesizer	14	`http://japan.steinberg.net/jp/support/unsupported_products/vst_classics_vol_2.html`
Phat Bass	Bass guitar	9	`http://www.dreamvortex.co.uk/instruments/`
Spicy Guitar	Guitar	22	`http://www.spicyguitar.com/`
Synth1	Subtractive synthesizer	99	`http://www.geocities.jp/daichi1969/softsynth/`
VB-1	Bass guitar	6	`http://japan.steinberg.net/jp/support/unsupported_products/vst_classics_vol_1.html`

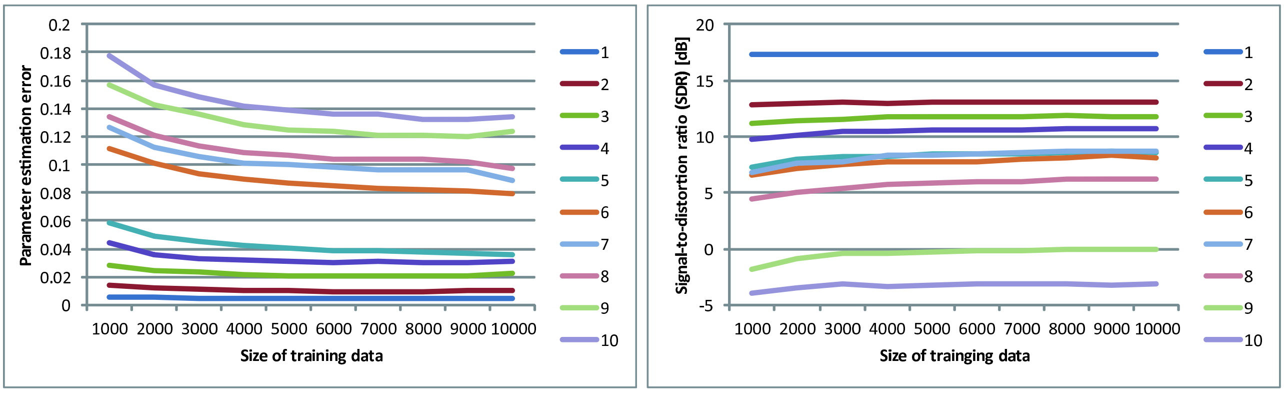

Figure 3. Result of the first experiment. Left and right figures show parameter estimation error and SDR, respectively. Each line indicates the number of parameters to be estimated.

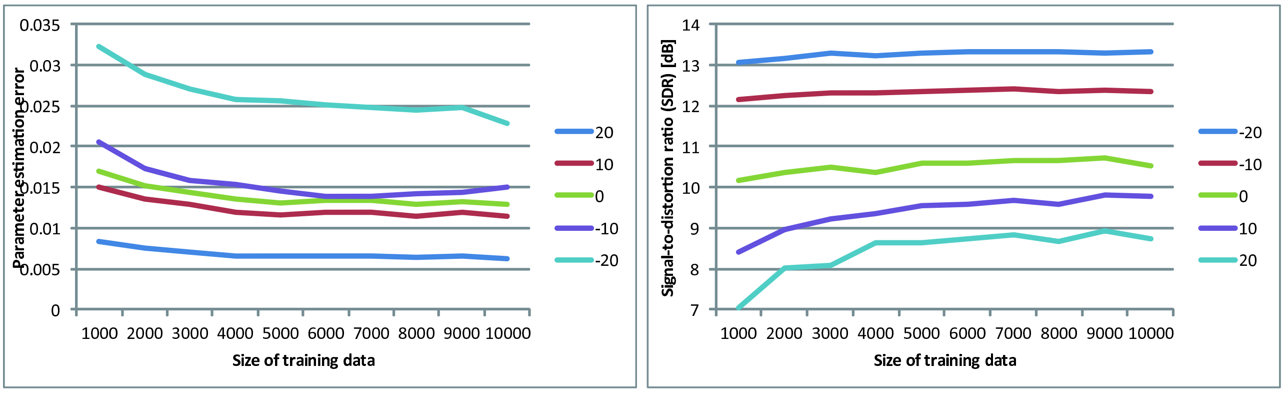

Figure 4. Result of the second experiment. Left and right figures show parameter estimation error and SDR, respectively, Each line indicates the signal-to-noise ratio.

a genetic algorithm with perceptual fitness calculation," EURASIP J. Adv. Signal Process., vol. 2003, no. 8, pp. 791–805, 2003. [Online]. Available: http://asp.eurasipjournals.com/content/2003/8/758284

[9] M. Sterling and M. Bocko, "Empirical physical modeling for bowed string instruments," in ICASSP2010, 2010, pp. 433–436.

[10] T. Nakano and M. Goto, "VocaListener: A singing-to-singing synthesis system based on iterative parameter estimation," in SMC2009, 2009, pp. 343–348.

[11] M. Goto, T. Nakano, S. Kajita, Y. Matsusaka, S. Nakaoka, and K. Yokoi, "VocaListener and VocaWatcher: Imitating a human singer by using signal processing," in ICASSP2012, 2012, pp. 5393–5396.

[12] T. Kitahara, "Computational musical instrument recognition and its application to content-based music information retrieval," Ph.D. dissertation, Kyoto University, 2007.

[13] L. Lu, D. Liu, and H.-J. Zhang, "Automatic mood detection and tracking of music audio signals," IEEE Trans. Audio, Speech and Lang. Process., vol. 14, no. 1, pp. 5–18, Jan. 2006.

[14] M. Goto, "A real-time music-scene-analysis system: Predominant-F0 estimation for detecting melody and bass lines in real-world audio signals," Speech Communication, vol. 43, no. 4, pp. 311–329, Sep. 2004.

[15] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," IEEE Trans. Audio, Speech and Lang. Process., vol. 14, no. 4, pp. 1462–1469, Jul. 2006.