PERCEPTSYNTH: MAPPING PERCEPTUAL MUSICAL FEATURES TO SOUND SYNTHESIS PARAMETERS

Sylvain Le Groux, Paul FMJ Verschure

Laboratory for Synthetic, Perceptive, Emotive & Cognitive Systems
Universitat Pompeu Fabra, Barcelona
slegroux@iua.upf.edu

ABSTRACT

This paper presents a new system that allows for intuitive control of an additive sound synthesis model from perceptually relevant high-level sonic features. We suggest a general framework for the extraction, abstraction, reproduction and transformation of timbral characteristics of a sound analyzed from recordings. We propose a method to train, tune and evaluate our system in an automatic, consistent and reproducible fashion, and show that this system yields various original audio and musical applications.

Index Terms— Sound Synthesis, Feature Extraction, SVM

1. INTRODUCTION

Nowadays, computers are already able to synthesize high quality sounds, and sound synthesis software has become largely accessible. Yet, the use of these tools can be quite intimidating and even counter-intuitive for non-technically oriented users. Most of the time, the relation between a change of synthesis parameter and its effect on perception of the synthesized sound is not predictable. A key issue in order to make rich sound modelling more widely available and easily usable is the control of sound synthesis in a natural, intuitive way.

The past few years have witnessed a growing interest from the Music Information Retrieval (MIR) community and the music industry to find good sound descriptors for retrieval and classification of audio files in large databases. This research effort has provided the community with a set of well-defined sound features. In this paper we aim at bridging the gap between sound description and sound synthesis. We focus on the mapping from perceptual features to sound generation parameters. In our system, the user gives the desired sound descriptors as an input to a synthesizer which will in turn calculate a sound with these desired properties.

2. RELATED WORKS

A few studies have explored the control of audio synthesis from perceptual parameters. We can distinguish three main trends: the machine learning view, where a model of the timbre is learned from audio data, the concatenative view, where the timbre is "constructed" by the juxtaposition of pre-existing sonic grains, and the signal processing view, where the transformations on timbre are model-dependant and direct applications of the signal model affordances. The first illustration of the machine learning point of view is found in [1] where Wessel et al. managed to control an additive synthesis model using artificial neural networks. With concatenative synthesis [2], a sound is defined as a combination of pre-existing samples in a database. These samples are already analyzed, classified and can be retrieved by their audio characteristics. Finally, in [3], Serra et al. proposed a set of timbral transformations based on Spectral Modelling Synthesis (SMS)[4]. By explicitely translating and distorting the spectrum of a sound, they achieved the control of vibrato, tremolo and gender transformation of a voice. Our approach is influenced by the machine learning view, which appears to be more general and independent of the sound analysis/synthesis model.

3. ANALYSIS/SYNTHESIS PARADIGM

We chose the additive synthesis model for its ability to synthesize a large range of sounds. Unlike physical models, it is based on original recorded sounds. It doesn't require having a theoretical representation of the physical properties of each different instrument. Nevertheless, to obtain satisfactory results, additive models require controlling many synthesis parameters, which are not musical, nor intuitive.

3.1. Harmonic Additive Model

A quasi-periodic tone can be decomposed in a sum of sine waves with time-varying amplitudes and frequencies [5].

$$y(n) \simeq \sum_{k=0}^{N-1} a_k(n) \sin(2\pi f_k(n)) \quad (1)$$

We assume the sample $y(n)$ can be reconstructed from the fundamental frequency estimate vector $\mathbf{f_j}$ (dimension NFrames)

and the matrix of partial amplitudes $\mathbf{A}_{i,j}$ (dimension NHarmonics by NFrames). We assume the influence of the phase is not primordial for resynthesis. This basic analysis model proves satisfactory in the context of this paper. We favor expressive control of synthesis over high quality of sound generation.

3.2. Principal Component Synthesis

We want to reduce the dimensionality of the parameter synthesis space by finding a compact representation of the time-varying amplitude matrix. For this purpose, we use the Principal Component Analysis (PCA) technique [6], which compute the most meaningful basis to re-express a noisy data set. We propose a method to synthesize a quasi-harmonic sound from the low-dimensional principal component decomposition of the harmonic amplitude matrix.

In the light of statistical analysis, one row of the partial amplitude matrix $\mathbf{A}_{i,j}$ is a variable (a particular partial amplitude trajectory), and one column is a measurement (a particular spectrum) for the j-th analysis frame.

PCA tells us that it is possible to get an approximation $\hat{\mathbf{A}}$ of the original matrix $\mathbf{A}$ from the low dimensional principal component space by multiplication of the Principal Component (PC) bases matrix and the time-varying envelope matrix. $\hat{\mathbf{A}} \approx \mathbf{FD}$, where the time-varying envelopes $\mathbf{D_i}$ are trajectories in the new PC basis $\mathbf{F}$. The new PC bases represent the "principal spectra", and a time-varying partial amplitude is a time-varying linear combination of those basis spectra. With this technique, we are able to synthesize a sound from a reduced set of PC synthesis parameters.

We now want to be able to predict those synthesis parameters (output or target) from fundamental frequency, loudness and timbral features (input or controls).

4. FEATURE EXTRACTION

4.1. Controls/Inputs

The features are obtained from short-term additive analysis based on the SMS model [4]. The data is organized in SDIF time-tagged frames [7]. Each frame is composed of the frequencies, amplitudes and phases of the quasi-harmonic components. The SMS analysis window (BlackmanHarris92) size is 1024 samples. Hop size is 256 samples. The sampling rate 44100Hz. We kept 80 harmonics trajectories ($dim(\mathbf{A}) = 80 * NFrames$).

Fundamental frequency and loudness are the main continuous features (independent from timbre) we extract and use as controls for the synthesis. Fundamental frequency is crucial for the quality of the resynthesis. We chose to use the robust Yin algorithm [8] to extract an accurate fundamental frequency from the audio signal. Additionally, we use a median filter and interpolates pitch during unvoiced regions to produce a smooth pitch curve.

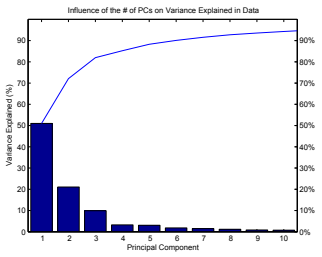

Fig. 1. Pareto chart of the number of PCs against percentage of variance expressed

We extract a loudness estimate by applying a loudness summation model like those proposed by Zwicker and Scharf [9]: $I(n) = \sum_{k=1}^{N} a_k(n)$.

The exploration of fine timbral control is also possible. Our system is designed in such a way that it allows to "shape" the synthesis of a sound using any kind of continuous feature extracted from audio as a controller. For instance, we can obtain a measure of brightness by computing the centroid for each spectral frame [10]:$C(n) = \frac{\sum_{k=1}^{N} a_k . f_k}{\sum_{k=1}^{N} a_k}(n)$.

Our high-level input is of dimension two (loudness and fundamental frequency) by NFrames.

4.2. Targets/ Outputs

We have found a way to reduce the dimension of the target space with PCA, and are able to resynthesize an estimate of the original sound from those parameters without any significant perceptual loss. It is possible to choose the number of Principal Components (NPCs) by entering the percentage of variance of the data we wish to take into account. In Fig. 1 we can see than most of the variance of our database is explained by only 3 PC. From now on, we assume that our synthesis parameters live in a 3D space.

The dimension of the target vector is now NPCs by NFrames. In addition to dimension reduction, PCA allows the decorrelation of the new target vector, which is a good preprocessing practice for the machine learning algorithm used for mapping. A problem of PCA decomposition is that the most significant PC parameters don't necessarily have an obvious relation to human perception. We need another layer to map perceptual, intuitive, musically relevant controls to this lower dimensional synthesis parameter space. This task is handled by the Support Vector Regression (SVR) algorithm.

The following section deals with the practical aspects of training the SVR learner that maps sonic percepts to sound generation parameters, taking into account our PCA synthesis model.

5. TRAINING AND RESULTS

5.1. Support Vector Mapping Machine

Support Vector Machines (SVMs) and kernel methods have become increasingly popular tools. We chose to apply this technique to the perceptual mapping problem based on good results the SVM algorithms obtained in various surveys from the machine learning community [11], and on the improvement it could bring to previous related works. Compared to other approaches, SVMs exhibit a lot of interesting properties. Namely, we can build highly non-linear regression function without getting stuck in a local minima. Moreover, there's only a few model parameters to pick, and the final results are stable and reproducible, unlike neural networks models where the result depends on the initial starting point. Finally, SVMs have been proved to be robust to noise in many applications [11].

5.2. Database

We normalize our dataset so that we have zero mean and unity standard deviation. We end up with training examples of two-dimensional (fundamental and loudness) real-valued vector as input and three-dimensional (3 PCs) real-valued vector as output. Our training set is extracted from an audio file of 2 minutes of solo saxophone with as much variability as possible in the attack, dynamics, range, etc...We split our data into a training set and a testing set (roughly 2/3, 1/3).

The support vector regression algorithm as described in [12] works for only one output. It doesn't work as is for multiple outputs. Thus we have to split our problem into n distinct (and supposedly independent) function estimation problems, considering each time a different "PC trajectory" as output, n being the number of PC necessary for resynthesizing a sound without significant perceptual loss.

Once the controls and targets are defined, computed and standardized, we can start the training of the supervised machine learning algorithm.

5.3. Model selection

Resampling approaches, commonly used for SVM, are very expensive in terms of computational costs and data requirements. The approach we used for parameter selection is based on a work by [13]. They propose a practical analytical approach to SVM regression parameter setting based directlly on the training data.

We use Cherkassky et al. prescription for the regularization parameter $C = max(|\bar{y} + 3\sigma_y|, |\bar{y} - 3\sigma_y|)$, where $\bar{y}$ is the mean of the training responses, and $\sigma_y$is the standard deviation of the training response.

The value of ε is chosen to be proportional to the input noise level and we assume that the standard deviation of noise σ can be estimated from data. We used the prescription in [13] for noise variance estimation via k-nearest neighbor's method: $\hat{\sigma}^2 = 1.5\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)$ where n is the number of training data, y_i the training response and $\hat{y}_i$ the fit by the k-nn regression.

After empirical comparisons, [13] proposed the dependency equation: $\varepsilon = \tau\sigma\sqrt{\frac{\ln n}{n}}$with $\tau = 3$ giving a good performance. The kernel width is selected to roughly reflect the input range of the training/test data.

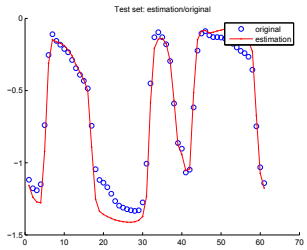

Fig. 2. Target and estimated first PC trajectories of the test set

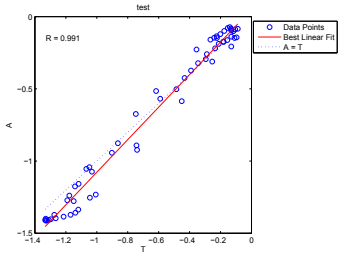

Fig. 3. Target/estimate correlation on the test set

5.4. Evaluation Measures

For evaluating the performance of our system, we use two functional measures on the test dataset. The first one is the RMS error, which gives a measure of the overall distance between two trajectories and the second measure is the correlation score, that indicates similarity of shape and synchrony between two trajectories. Finally, the resynthesis from the estimated output also give us a perceptual, subjective equivalent of those functional measures.

5.5. Results

The results of our experiment on the data we described in section 7.1 are summarized in table 1. As can be seen in Fig.3, the estimation of the principal component trajectory on the test data (which controls have not been learned by the support vector machine before) is good and the system generalizes well. The corresponding audio resynthesis examples are available on the *PerceptSynth* website[1].

[1] http://www.iua.upf.edu/~slegroux/perceptsynth

PC trajectory	RMS error	Correlation score
First PC	0.009	0.991
Second PC	0.0159	0.99
Third PC	0.0534	0.973

Table 1. Results with a 2 min saxophone solo

6. APPLICATIONS

One of the most powerful feature of this system is its ability to change the original melody contour or loudness control while preserving the musical identity of the audio sequence on which the machine learning algorithm has been trained.

Besides, our model is well-suited for cross-synthesis, where the control parameters of one instrument can be used in combination with the "support vector timbre model" of another instrument. Another possibility of cross-synthesis is at the level of the PC synthesis, where the PCs, or basis spectra, of one instrument can be replaced by those of another instrument.

Due to the characteristics of the PC synthesis, our model has an interesting property of scalability. The number of useful PC bases can be chosen depending on the quality of the sound required at the moment. This behavior is particularly interesting in networked applications, where the available bandwidth is variable.

In this paper we have mostly studied the control of a synthesis model from loudness and fundamental frequency parameters, letting the machine learning component take care of the overall timbre generation. But we have also managed to control directly the properties of an instrument timbre such as brightness. This property, allowing direct generation of sound from continuous timbral features, is extremely interesting, and suggests an analysis-by-synthesis type of applications. For instance, in the Music Information Retrieval community, the problem of finding relevant descriptors and judging their perceptual influence is still open. Our tool allowing feature-based synthesis would prove useful.

7. CONCLUSION

We have proposed a system that allows for flexible and intuitive control of sound generation from high-level sonic percepts. It provides the user with continuous control over a sound directly analyzed from recordings. We devised an analysis/synthesis paradigm well-suited for a machine learning approach to this non-linear mapping problem, and found a way to reduce the dimensionality of the synthesis parameter space, while preserving the auditory quality of the resynthesis. We described a framework to train, tune and evaluate our system in an automatic, consistent and reproducible fashion. This system yields various original audio and musical applications.

In the future, we would like to extend our experiments on larger datasets with as much variance as possible, take into account temporal dependencies, and since the synthesis models presented are causal, realize a real-time implementation of this perceptual synthesizer.

8. REFERENCES

[1] David Wessel, Cyril Drame, and Matthew Wright, "Removing the time axis from spectral model analysis-based additive synthesis: Neural networks versus memory-based machine learning.," in *Proc. ICMC*, 1998.

[2] Diemo Schwarz, *Data-Driven Concatenative Sound Synthesis*, Ph.D. thesis, Ircam - Centre Pompidou, Paris, France, January 2004.

[3] X. Serra and J. Bonada, "Sound Transformations based on the SMS High Level Attributes," in *Proceedings of the Digital Audio Effects Workshop*, 1998.

[4] X. Serra and J. Smith, "Spectral Modeling Synthesis: a Sound Analysis/Synthesis System Based on a Deterministic plus Stochastic Decomposition," *Computer Music Journal*, vol. 14, no. 4, pp. 12–24, 1990.

[5] R.J. McAulay and Th.F. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Trans. on Acoust., Speech and Signal Proc.*, vol. 34, pp. 744–754, 1986.

[6] J. Edward Jackson, *A Users Guide to Principal Components*, John Wiley & Sons, 2003.

[7] M. Wright et al., "New Applications of the Sound Description Interchange Format," in *Proc. ICMC*, 1998.

[8] A. de Cheveigné and H. Kawahara, "Yin, a fundamental frequency estimator for speech and music," *J. Acoust. Soc. Am.*, vol. 111, pp. 1917–1930, 2002.

[9] E. Zwicker and B. Scharf, "A model of loudness summation," *Psychological Review*, vol. 72, pp. 3–26, 1965.

[10] Wolfe J. Schubert, E. and A Tarnopolsky, "Spectral centroid and timbre in complex, multiple instrumental textures," in *Proc. ICMPC*, 2004.

[11] Kristin P. Bennett and Colin Campbell, "Support vector machines: Hype or hallelujah?," *SIGKDD Explorations*, vol. 2, no. 2, pp. 1–13, 2000.

[12] Chih-Chung Chang and Chih-Jen Lin, *LIBSVM: a library for support vector machines*, 2001.

[13] V. Cherkassky and Y. Ma, "Practical selection of SVM parameters and noise estimation for SVM regression," *Neural Networks*, vol. 17, no. 1, pp. 113–126, January 2004.