Intuitive Real-Time Control of Spectral Model Synthesis

Phillip Popp
Oakland, CA 94612
popp.phillip@gmail.com

Matthew Wright
CREATE/MAT
University of California
Santa Barbara, CA 93106
matt@create.ucsb.edu

ABSTRACT

Several methods exist for manipulating spectral models either by applying transformations via higher level features or by providing in-depth offline editing capabilities. In contrast, our system aims for direct, full, intuitive, real-time control without exposing any spectral model features to the user. The system extends upon previous machine learning work in gesture-synthesis mapping by applying it to spectral models; these are a unique and interesting use case in that they are capable of reproducing real world recordings, due to their relatively high data rate and complex, intertwined and synergetic structure. To achieve a direct and intuitive control of a spectral model, a method to extract an individualized mapping between Wacom Pen parameters and Spectral Model Synthesis frames is described and implemented as a standalone application. The method works by capturing tablet parameters as the user pantomimes to synthesized spectral model. A transformation from Wacom Pen parameters to gestures is obtained by extracting features from the pen and then transforming those features using Principal Component Analysis. Then a linear model maps between gestures and higher level features of the spectral model frames while a k-nearest neighbor algorithm maps between gestures and normalized spectral model frames.

Keywords

Spectral Model Synthesis, Gesture Recognition, Synthesis Control, Wacom Tablet, Machine Learning

1. INTRODUCTION

Spectral Model Synthesis (SMS) is a flexible platform capable of generating rich and vivid sounds [11] [9]. It represents a sound's periodic and noisy components as a series of frames, each frame consisting of a set of sinusoidal frequencies and amplitudes plus a spectral envelope for noise. Deriving a compact spectral model from recorded audio captures a veridicality difficult to create using other forms of synthesis. SMS retains the gestalt of the audio while allowing stretching, pitch shifting and other modifications. Despite this flexibility, SMS is difficult to manipulate intuitively in real-time beyond macro control such as volume, pitch, and duration. The number of synthesis parameters in a single SMS frame can be well over 100; choosing how to tie a low-dimensional control device to these is non-trivial.

Recently, machine learning and statistical analysis techniques have been applied to mapping inputs to synthesizer controls [2]. We apply these ideas to SMS and propose a new model to map input gestures to SMS control. We tailor each mapping in a user-directed way by having the user listen as a spectral model is resynthesized and pantomime, in real-time, the gestures that "should" correspond to the sound. Essentially, the user imagines that she is directly controlling the sound with a Wacom Tablet [12]. The system captures these pantomimed input gestures for use as a training set to determine a mapping between the tablet and SMS parameters via machine learning techniques.

The first learning step analyzes the captured input gestures with principal component analysis (PCA) to create a lower dimensional "gesture" space. Linear regression then maps between the (principal components of the) gestures and higher level spectral model frame features, while k-nearest neighbors maps between the gestures and normalized spectral model frames. After the system learns a complete mapping it can synthesize new sounds in response to real-time Wacom gestures. Since the mapping originated from the user's pantomimes to the original spectral model, the control is intuitive. Repeating the example gestures results in approximating the original SMS frames, while deviating from the original pantomimes results in new spectral model frames that did not exist in the original spectral model but make intuitive perceptual sense to the user.

2. RELATED WORKS

There have been several approaches to controlling SMS. One approach focuses on providing software tools to allow users to edit spectral models in an offline manner [5]. A second approach reduces the number of inputs needed to control synthesis by extracting higher level sonic descriptors derived from the spectral model [8], using general purpose dimension reduction techniques [6], or defining generic *a priori* mappings [13]. Instead, our approach allows users to map their personal gestures to aspects of the spectral model rather than simply mapping the model characteristics to a parameter value. Like the second approach, it attempts to control the synthesis in a higher-level and more abstract space, but in contrast it provides a personalizable and potentially more flexible platform because each user can reconfigure the control mapping for each spectral model.

Fiebrink et al. utilize various machine learning and signal processing algorithms to map between input controllers and synthesis controls [2], demonstrating this approach by applying it to score following, physical modeling synthesis, and video manipulation utilizing a "play-along" data gathering method. Our approach differs in both the synthesizer's structure and the aspirations of the control mapping. Particularly, SMS provides a more low-level representation of sound than musical scores or physical models. The data rate

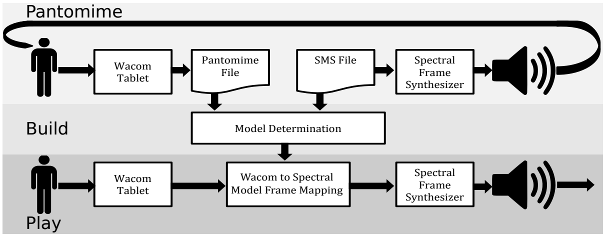
Pantomime
Wacom Tablet
Pantomime File
SMS File
Spectral Frame Synthesizer
Build
Model Determination
Wacom Tablet
Wacom to Spectral Model Frame Mapping
Spectral Frame Synthesizer
Play

Figure 1: Overview of Mapping Generation Steps

and emergent nature of the information in SMS culminate in rich, detailed and life-like sounds while making intuitive control of the model exponentially more difficult. As well as SMS structurally differing from the previously investigated synthesis models, the source and derivation of the model is also conceptually different. Fiebrink et al. utilize scores and/or random permutations of synthesis settings to provide examples for users to pantomime to. Here, the model is derived from a recording, and since spectral models are so flexible, special care must be taken to retain the gestalt of the sound while still offering new spaces for exploration. The spectral model synthesizer in itself is cannot retain the essence of the recording. Instead it is the duty of the the machine learning algorithms to retain certain qualities of the model, while relaxing others in order to provide direct, intuitive control while still being capable of creating unforeseen sounds. To do so we utilize expert knowledge of SMS, pervious work upon motion gesture analysis [1] and machine learning for synthesis control [2], and provide an environment where users can experiment to create new mappings [7].

3. SYSTEM OVERVIEW

Our method uses several steps to learn a mapping from tablet input to SMS control (figure 1). First the user generates examples by pantomiming "control" gestures as a predetermined model plays; the system time-stamps and records the resulting tablet parameters into a *pantomime file.* Then the machine learning engine derives a two-level mapping: from pen parameters to *gestures*, and from these gestures to SMS frames. One can then use this mapping in real-time to control SMS using the tablet. A standalone OSX application guides the user through the entire process, from pantomiming, to building a mapping, to playing.

3.1 Pantomimes

Our software lets the user load a spectral model, preview it, pantomime to it, and see the Wacom Pen's parameters. An example of a pen's parameters in comparison to a spectral model's frequencies can be seen in figure 2. The software provides three mechanisms to aid the user. The first is visual feedback of the recent history of all pen parameters shown as a trail of slowly fading dots upon a white canvas: X and Y position determine dot's position, pen-tip pressure and Z position (pen's height above tablet) determine hue, and the X and Y tilt parameters control the size and shape of the dot. Second, the user can learn the nuances of a chosen spectral model via practice runs listening and pantomiming without recording the results. Third, to help accurately synchronize the pantomime timing to the sound, a stop-light metaphor counts down (via large red, then yellow, then green circles each displayed for one second) to the beginning of audio playback after the user clicks the record button. This also gives the user time to prepare (e.g., picking up the Wacom pen after clicking the record button).

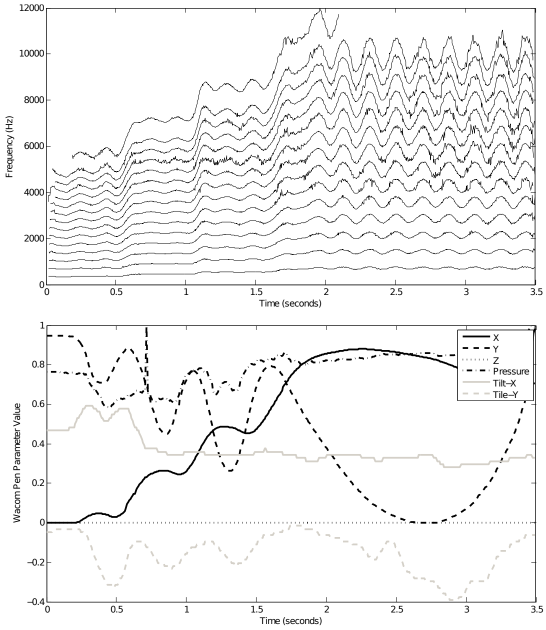

Figure 2: Sinusoidal Tracks of Spectral Model (top) and Wacom Pen Parameters (bottom)

3.2 Gesture Language Extraction and Transformation

Given the opportunity to pantomime to SMS resynthesis, each user will assuredly perform different gestures to the same audio. Likewise, the same user will usually perform widely differing gestures when pantomiming to different SMS models [3]. These gestures should reflect the way a user would intuitively control the sound if they were producing it. In order to create individualized and intuitive control of SMS parameters, features likely to express musical intention are extracted from the captured Wacom Tablet parameters. We assume that features containing relatively high energy amongst the set of captured input device parameters encapsulate a high expressive potential, according to the principle of a "correspondence between the "size" of a control gesture and the acoustic result" [10]. To this end, we define our gesture language as several linear combinations of features with high expressive potential and derive a transformation between features and the gesture language using Principal Component Analysis (PCA). This transformation emphasizes features with high expressive potential and deemphasizes those with lower expressive potential, as well as reducing the dimensionality of the input to the learning algorithms in later stages of this mapping algorithm.

3.2.1 Feature Extraction

To extend our mapping algorithm's ability to encapsulate gestures, we estimate the instantaneous velocity (first derivative) and acceleration (the second derivative) of the tablet parameters using a five-point stencil. This preprocessing step is primarily to capture non-linear motion information. We define the set of six parameters (x, y, and z position, pressure, x-tilt, y-tilt) from the Wacom Pen as follows:

$$\mathbf{p}(n) = [w_x(n), w_y(n), w_z(n), w_p(n), w_\theta(n), w_\phi(n)]. \quad (1)$$

Each output frame is the concatenation of the original pen parameters with the first and second derivatives:

$$\mathbf{f}(n) = [w_x(n), w'_x(n), w''_x(n), ..., w_\phi(n), w'_\phi(n), w''_\phi(n)]^T \quad (2)$$

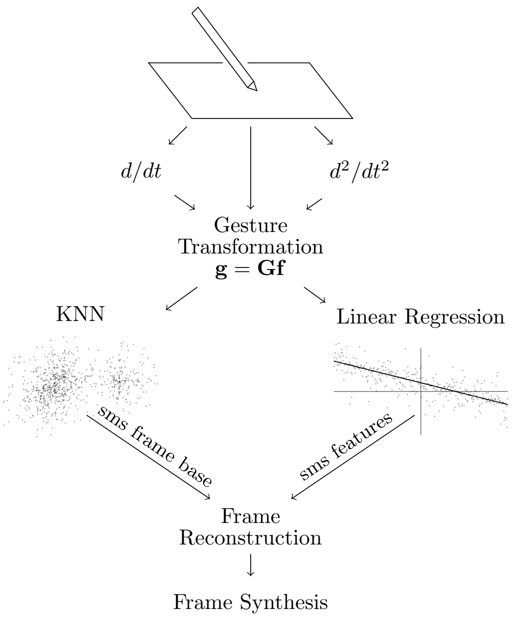

Figure 3: Mapping From a Wacom Tablet to Spectral Model Frames

3.2.2 Gesture Language

We aim to determine a gesture language that distinguishes and accentuates the features that show high potential expressivity for a particular user, as well as a means to transform the tablet features into the gesture language in a real-time fashion. Principal Component Analysis (PCA) provides a suitable means to determine a transformation to a gesture language of M continuous values (where $0 < M \leq 18$). If we associate energy with expressivity, PCA results in a transformation where the first column of the transformation captures the most expressivity, the second column contains the second most expressivity, the third column the third most expressivity, and so on. Additionally, we emphasize/deemphasize the output of the transformation by weighting each of the M columns by their respective eigenvalues. In addition to accentuating Wacom Pen features used for expression, the gesture transformation also reduces the dimensionality of inputs to the next mapping stage, making them more robust against the various pitfalls associated with the curse of dimensionality [4]. Equation 4 describes the transformation matrix between Wacom Pen features and gestures where λ_i is the eigenvalue and $\mathbf{p}_i$ is the corresponding eigenvector derived from PCA analysis of $\mathbf{F}$. The transformation from Wacom Pen parameters to gestures $\mathbf{g}(n)$ is shown by equations 1, 2 and 5.

$$\mathbf{F} = [\mathbf{f}(0), \mathbf{f}(1), ..., \mathbf{f}(N-1)]^T \tag{3}$$

$$\mathbf{G} = [\lambda_0\mathbf{p}_0, \lambda_1\mathbf{p}_1, ...\lambda_{M-1}\mathbf{p}_{M-1}]^T \tag{4}$$

$$\mathbf{g}(n) = \mathbf{Gf}(n) \tag{5}$$

3.3 Gestures to Spectral Model Frames

It would be challenging to find a one-size-fits-all mapping from gestures to spectral model frames. Spectral model frames posses a composite structure, made of components that have disparate meanings and values. To overcome this challenge we employ the two-pronged approach outlined in figure 3. One mapping path maps gestures to higher level spectral frame features via linear regression. The other path utilizes a K-Nearest Neighbor (KNN) algorithm where the gesture values describe the coordinates of normalized spectral model frames in a Euclidean space. After the linear regression and KNN models are trained, a new spectral model frame is generated by advancing a gesture through both mapping paths and then combining their output to construct a new spectral model frame. Retaining the normalized frames preserves the complex structure of the spectral model, which in turn contains many of the minute details that differentiate SMS from other forms of synthesis. Concurrently, linearly mapping higher level features allows new sonic spaces to be explored.

3.3.1 Linear Mapping of Higher Level Features

Linearly mapping higher level features expands the capability of the overall mapping algorithm by encapsulating complex features of the spectral model and providing unbounded control of them. Higher level features can capture aspects of the spectral model that happen on a time-scale too small to recreate accurately by drawing, or in a way that maps complexly to gestures. Consider mapping a spectral model's vibrato. The pitch fluctuations happen on a time scale too small to reproduce when pantomiming. By instead mapping a gesture to the depth of a vibrato, the user simply needs to pantomime something that implies more vibrato, not match the fluctuations in pitch directly. Linear mapping also allows the system to generalize parameters beyond the range presented in the original spectral model. For example, if we only used a KNN approach to map pitch, the user would be confined to the pitches existent in the original spectral model. By deriving a linear mapping, the user can go beyond and between the original pitches because the linear map utilizes an unbounded continuous function to convert gestures to pitch.

To create a linear mapping of a higher level feature, first the feature is extracted from the spectral model frames. The function used to extract the feature must normalize the spectral model frame with respect to the feature as well as be invertible so that the spectral model frame could be recreated at a later stage. These higher level features are then mapped linearly by performing linear regression upon the pairs of gestures and their corresponding feature value.

3.3.2 KNN Mapping of Spectral Model Frames

After extracting higher level features from the spectral model frames, the normalized spectral model frames are placed in a Euclidean space where the values of the gesture are utilized as the frame's coordinates.

A KNN algorithm is used to map between gestures and these altered spectral model frames. KNN algorithms work on the assumption that data can be arranged into a metric space, and that a new, unclassified piece of data can best be described by inspecting the K nearest classified data within a training set [4]. This property is extremely attractive in that it allows us to use our gesture vectors as direct predictors of the output spectral model frame, ignoring the complex relationship between the gesture vector and the particular format of the SMS frame.

3.3.3 Spectral Model Frame Reconstruction

After an input gesture has been mapped through the higher level feature linear models and the normalized spectral model frame KNN model, a new spectral model frame is constructed by combining the two. Beginning with the base spectral model frame at the output of the KNN, the higher level features are applied to the frame using the inverse of the function used to extract the higher level feature. This frame is then fed to the synthesizer for audio playback.

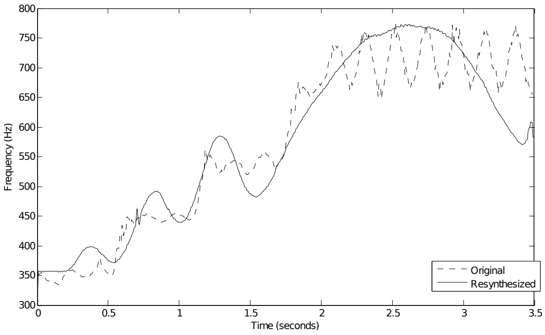

Figure 4: Original and Mapped Fundamental Frequency

4. RESULTS/ANALYSIS

The system as a whole successfully allows intuitive control over a spectral model. It could deduce a mapping at a very satisfactory speed, quick enough for fast experimentation. For a spectral model covering 4 seconds of time, and a pantomime file containing 428 frames it took approximately 2.5 seconds to derive a mapping using a single Intel Core i5 on a Macbook. While the speed of the system does not pose an issue, certain aspects of the mapping algorithm do. First, the user has the option to pantomime several times to the same spectral model and combine all of their pantomime files into a single training run. It was noted that a single pantomime generally produces satisfactory results, but additional pantomime files smoothed out the control over the spectral model, and made it feel more predictable. We hypothesize that this is simply a matter of providing more training data, resulting in more robust calculations in both the PCA and linear regression stages. Additional investigation is needed to shed more light on the appropriate amount of training data needed to derive a control mapping. Also, while many aspects of the original spectral model (loudness, frequency envelope) could be recreated by reproducing the original pantomimes, the resynthesized sound lacked the same authenticity in the original spectral model. Figure 4 shows both the spectral model's original fundamental frequency, and the fundamental frequency determined by mapping the pantomime file through the mapping algorithm. While the general trends of the original fundamental are grossly estimated, many of the finer temporal variations are completely smoothed out. One possible explanation could be that the Wacom Pen's sample rate is too slow (∼50Hz) and control too gross. This makes it incapable of controlling micro-variations of spectral model parameters that change more often than the Wacom Pen is sampled. Second, the linear regression between gestures and the fundamental frequency may be too simple of a model to translate from gestures to the fundamental. Additional investigation is needed to understand exactly which features of the spectral model are retained after the mapping, and which are lost.

5. CONCLUSION AND FUTURE WORK

We have introduced a novel method to extract an intuitive and individualized mapping from a users's performance to control of Spectral Model Synthesis based on capturing tablet parameters as the user pantomimes to synthesized spectral model. A robust machine learning system incorporating time derivative estimation, PCA, KNN, and linear regression produces acceptable results: when the performance gestures imitate the pantomimed training gestures the output sound recognizably approximates the input sound, while related gestures intuitively produce interesting extrapolated sounds.

Several areas of improvement could increase the overall quality of the system. First, a better methodology for pairing Wacom Pen parameters to spectral model frames could improve the overall mapping by reducing the inherent errors of a pantomime. By segmenting both the Wacom Pen parameters and spectral model into sub-note sections (e.g. attack, decay, sustain), we could come to a tightly bound pairing between gestures and sub-note events. Second, additional mapping techniques could be investigated such as artificial neural networks, and logit regression. While these models may require more training and time to create, they have the capability to capture complex inter-feature relationships not captured by the linear regression functions of the current design.

6. REFERENCES

[1] F. Bevilacqua, J. Ridenour, and D. Cuccia. 3D motion capture data: motion analysis and mapping to music. In *Proceedings of the workshop/symposium on sensing and input for media-centric systems*. Citeseer, 2002.

[2] R. Fiebrink, P. Cook, and D. Trueman. Play-along mapping of musical controllers. In *Proc. International Computer Music Conference*. Citeseer, 2009.

[3] R. Godøy, E. Haga, and A. Jensenius. Playing air instruments: Mimicry of sound-producing gestures by novices and experts. *Gesture in Human-Computer Interaction and Simulation*, pages 256–267, 2006.

[4] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer series in statistics. Springer, 2009.

[5] M. Klingbeil. Software for spectral analysis, editing, and synthesis. In *Proceedings of the International Computer Music Conference*, pages 107–110. Citeseer, 2005.

[6] S. Le Groux and P. Verschure. Perceptsynth: mapping perceptual musical features to sound synthesis parameters. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 125–128. IEEE, 2008.

[7] P. Popp. `http://www.phillippopp.com`.

[8] X. Serra and J. Bonada. Sound transformations based on the sms high level attributes. In *Proceedings of the Digital Audio Effects Workshop*. Citeseer, 1998.

[9] X. Serra and J. Smith. A sound decomposition system based on a deterministic plus residual model. *The Journal of the Acoustical Society of America*, 87:S97, 1990.

[10] D. Wessel and M. Wright. Problems and prospects for intimate musical control of computers. *Computer Music Journal*, 26(3):11–22, 2002.

[11] M. Wright, J. Beauchamp, K. Fitz, X. Rodet, A. Röbel, X. Serra, and G. Wakefield. Analysis/synthesis comparison. *Organised Sound*, 5:173–189, 2000/12/01 2000.

[12] M. Wright, D. Wessel, and A. Freed. New musical control structures from standard gestural controllers. In *Proceedings of the ICMC*, 1997.

[13] M. Zbyszynski, M. Wright, A. Momeni, and D. Cullen. Ten years of tablet musical interfaces at CNMAT. In *Proceedings of the 7th international conference on New interfaces for Musical Expression*, pages 100–105. ACM, 2007.