

AVA: A Graphical User Interface for Automatic Vibrato and Portamento Detection and Analysis

Luwei Yang¹ Khalid Z. Rajab² Elaine Chew¹

¹ Centre for Digital Music, Queen Mary University of London

² Antennas & Electromagnetics Group, Queen Mary University of London

{l.yang, k.rajab, elaine.chew}@qmul.ac.uk

ABSTRACT

Musicians are able to create different expressive performances of the same piece of music by varying expressive features. It is challenging to mathematically model and represent musical expressivity in a general manner. Vibrato and portamento are two important expressive features in singing, as well as in string, woodwind, and brass instrumental playing. We present AVA, an off-line system for automatic vibrato and portamento analysis. The system detects vibratos and extracts their parameters from audio input using a Filter Diagonalization Method, then detects portamenti using a Hidden Markov Model and presents the parameters of the best fit Logistic Model for each portamento. A graphical user interface (GUI), implemented in MATLAB, allows the user to interact with the system, to visualise and hear the detected vibratos and portamenti and their analysis results, and to identify missing vibratos or portamenti and remove spurious detection results. The GUI provides an intuitive way to see vibratos and portamenti in music audio and their characteristics, and has potential for use as a pedagogical and expression analysis tool.

1. INTRODUCTION

Musicians introduce a high degree of acoustic variations in performance, above and beyond the categorical pitches and durations indicated in the musical score [1]. The sources of these acoustic variations include dynamic shaping, tempo variation, vibrato, portamento, staccato, and legato playing. While some expressions have been notated in the score (e.g. tempo and dynamics), musicians sometimes alter the instructions to create their own expressions [2]. We call these devices expressive features as they are usually not denoted in the composition but adopted in performance. These devices result in unique performance styles that differentiate one musician from another.

We focus on two expressive features: vibrato and portamento. Vibrato is a periodic modulation of frequency, amplitude, and even spectrum [3]. Portamento is the note transition that allows musicians to adjust the pitch continuously from one note to the next [4]. Vibrato and portamento characteristics can be used to reveal differences in

performance styles, and performance variation among different musicians [4, 5, 6, 7, 8].

This paper presents an off-line system, AVA, which accepts raw audio and automatically tracks the vibrato and portamento to display their expressive parameters for inspection and further statistical analysis. We employ the Filter Diagonalization Method (FDM) to detect vibrato [9]. The FDM decomposes the local fundamental frequency into sinusoids and returns their frequencies and amplitudes, which the system uses to determine vibrato presence and vibrato parameter values. A fully connected three-state Hidden Markov Model (HMM) is applied to identify portamento. The resulting portamenti are modeled as Logistic Functions which are well suited to displaying the characteristics of a portamento [4]. The AVA system has been implemented in MATLAB and consists of a graphical user interface (GUI) and all relevant functions¹.

The structure of the paper is as follows: Section 2 presents the vibrato and portamento feature detection and analysis modules. Section 3 introduces AVA's MATLAB interface, and Section 4 presents discussions and conclusions.

2. FEATURE DETECTION AND ANALYSIS

The basic architecture of the AVA system is shown in Figure 1. Taking the audio as input, the pitch curve (fundamental frequency) is extracted using the pYIN method [10], a probabilistic version of the original Yin method [11]. The resulting pitch curve is sent to the vibrato detection module, which identifies vibrato existence using an FDM-based method. The detected vibratos are forwarded to the module for vibrato analysis, which outputs the vibrato statistics. To ensure the best possible portamento detection performance, we flatten the detected vibratos using the built-in MATLAB 'smooth' function as the oscillating shape of the vibrato degrades portamento detection. The HMM-based portamento detection module uses this vibrato-free pitch curve to identify potential portamenti. A Logistic Model is fitted to the detected portamentos for quantitative analysis. Moreover, if there are errors in detection, the interface allows the user to indicate missing vibratos or portamenti and remove spurious results.

2.1 Vibrato Detection and Analysis

There exist two kinds of vibrato detection methods: note-wise and frame-wise methods. Note-wise methods require

Copyright: ©2016 Luwei Yang et al. This is an open-access article distributed under the terms of the [Creative Commons Attribution License 3.0 Unported](https://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

¹The beta version of AVA is available at luweiyang.com/research/ava-project.

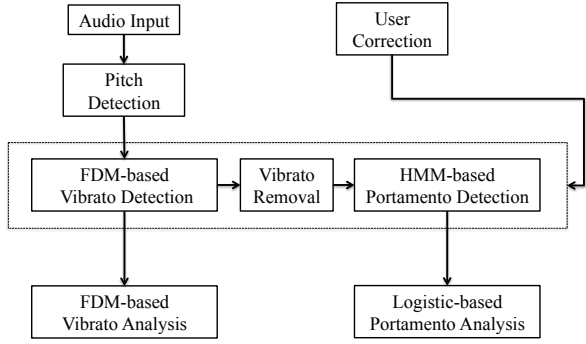


Figure 1. The AVA architecture.

Algorithm 1: The FDM algorithm

Input: Pitch curve (fundamental frequency)
Output: The frequency and amplitude of the sinusoid with the largest amplitude

Set the scan frequency range;
 Filter out any sinusoids whose frequency is out of the frequency range of interest;
 Diagonalize the matrix formed by the pitch curve;

for each iteration do

- Create a matrix using 2D FFT on the pitch curve;
- Diagonalize this matrix;
- Get eigenvalues;
- Check the acceptance of eigenvalues;

end

Calculate the frequencies from the eigenvalues;
 Calculate the amplitudes from the corresponding eigenvectors;
 Return the frequency and amplitude of the sinusoid with the largest amplitude;

a note segmentation pre-processing step before determining if the note contains a vibrato [12, 13]. Frame-wise methods divide the audio stream, or the extracted pitch curve information, into a number of uniform frames. Vibrato existence is then decided based on information in each frame [14, 15, 16, 9].

We employ the Filter Diagonalization Method (FDM) described in [9] to detect vibratos and characterize their properties. The FDM is able to extract the frequency and amplitude of sinusoids for a short time signal, thus making it possible to determine vibrato presence over a short time span. Fundamentally, the FDM assumes that the time signal (pitch curve) of a frame is the sum of exponentially decaying sinusoids,

$$f(t) = \sum_{k=1}^K d_k e^{-in\tau\omega_k}, \text{ for } n = 0, 1, \dots, N, \quad (1)$$

where K is the number of sinusoids required to represent the signal to within some tolerance threshold. ω_k and d_k are fitting parameters which are defined as the complex frequency and complex weight, respectively, of the k -th sinusoid. The aim of the FDM is to find the $2K$ unknowns,

representing all ω_k and d_k . A brief summary of the steps is described in Algorithm 1. Details of the algorithm and implementation are given in [9]. Here, we only consider the frequency and amplitude of the sinusoid having the largest amplitude. A Decision Tree is applied to these two parameters to determine vibrato presence. The window size is set to 0.125 seconds and step size is one quarter of the window. Note pruning (throw away any feature whose duration is less than a threshold) used a threshold of 0.25 seconds.

The vibrato rate and extent fall naturally out of the FDM analysis results. In addition, to characterize the shape of a detected vibrato, we use sinusoid similarity as described in [7]. The sinusoid similarity is a parameter between 0 and 1 that describes the similarity of a vibrato shape to a reference sinusoid using cross correlation.

2.2 Portamento Detection and Analysis

To detect portamentos, we have created a fully connected three-state HMM using the delta pitch curve as input as shown in Figure 2. The three states are down, steady, and up, which correspond to slide down, steady pitch, and slide up gestures. Based on experience, we estimate the transition probabilities to be those shown in Table 1. A Gamma distribution models the probability density distribution of each down and up state's observation. The steady pitch observation's probability density distribution is modeled as a sharp needle around 0 using a Gaussian function. The best likely path is decoded using the Viterbi algorithm. All state changes are considered as boundaries. The 0.09 second note pruning is applied.

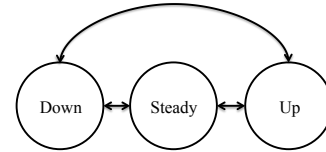


Figure 2. The portamento detection HMM transition network.

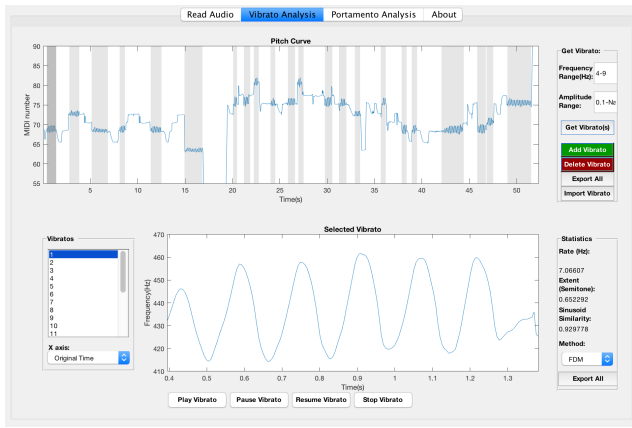
	Down	Steady	Up
Down	0.4	0.4	0.2
Steady	1/3	1/3	1/3
Up	0.2	0.4	0.4

Table 1. Transition probability of HMM-based portamento detection.

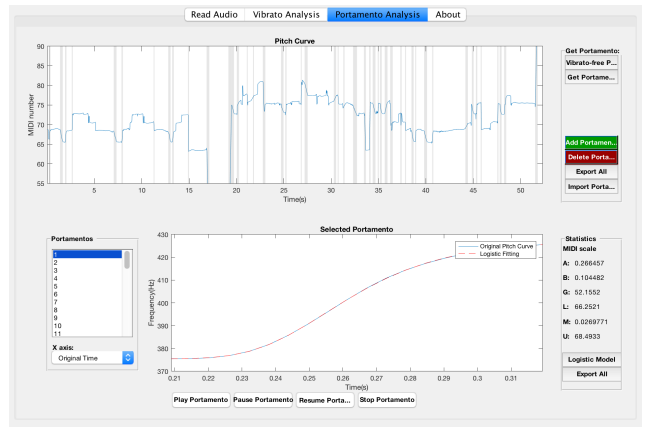
To quantitatively describe the portamento, we apply the Logistic Model in the fashion described in [4]. This model is motivated by the observation that portamenti largely assume S-shapes. An ascending S shape is characterized by an acceleration in the first half and a deceleration in the second half. An inflection point exists between these two processes. The Logistic Model is described as

$$P(t) = L + \frac{(U - L)}{(1 + Ae^{-G(t-M)})^{1/B}}, \quad (2)$$

where L and U are the lower and upper horizontal asymptotes, respectively. Musically speaking, L and U are the



(a) Vibrato Analysis



(b) Portamento Analysis

Figure 3. Screenshots of AVA.

antecedent and subsequent pitches of the transition. A , B , G , and M are constants. G can further be interpreted as the growth rate, indicating the steepness of the transition slope. The time of the inflection point is given by

$$t_R = -\frac{1}{G} \ln \left(\frac{B}{A} \right) + M. \quad (3)$$

The pitch of the inflection point can then be calculated by substituting t_R into Eq. (2).

3. THE AVA INTERFACE

The vibrato and portamento detection and analysis methods described above have been implemented in MATLAB. AVA's GUI consists of three panels accessed through tabs: Read Audio, Vibrato Analysis, and Portamento Analysis. The Read Audio panel allows a user to input or record an audio excerpt and obtain the corresponding pitch curve. The Vibrato Analysis and Portamento Analysis panels provide visualizations of vibrato and portamento detection and analysis results, respectively.

The left screenshot in Figure 3 shows the Vibrato Analysis panel analyzing an erhu excerpt. The pitch curve of the entire excerpt is presented in the upper part, with the shaded areas indicating possible vibratos. Vibrato existence is determined using the FDM-based vibrato detection method, which is triggered using the button in the upper right. The interface allows the user to change the default settings for the vibrato frequency and amplitude ranges; these adaptable limits serve as parameters for the Decision Tree vibrato existence detection process.

Shaded boxes highlight the detected vibratos on the pitch curve. Two edit functions, allowing the adding and deleting of shaded windows indicating detected vibratos, are provided for users to correct vibrato detection errors. On the lower left is a box listing the indices of the detected vibratos. The user can click on each shaded area, or choose element in the listing box, or use the left- or right-arrow keys, to navigate between vibratos. The selected vibrato pitch curve is presented in the lower plot with corresponding parameters shown on the right. In this case, with the vibrato frequency range threshold $[4, 9]$ Hz and amplitude

range threshold $[0.1, \infty]$ semitones, the selected vibrato has frequency 7.07 Hz, extent 0.65 semitones, and sinusoid similarity value 0.93. A drop down menu allows the user to choose between the original time axis and a normalized time axis for visualizing each detected vibrato. A playback function assists the user in vibrato selection and inspection.

The right plot in Figure 3, shows the Portamento Analysis panel for the same music excerpt. The pitch curve shown here is that of the Vibrato Analysis panel after flattening the detected vibratos so as to improve portamento detection. Like the Vibrato Analysis panel, the Portamento Analysis panel also provides add and delete functions for the shaded windows indicating the detected portamenti. A click of a button initiates the process to fit Logistic Models to all the portamenti.

The best-fit Logistic model is shown as a red dashed line against the original portamento pitch curve. A panel to the right shows the corresponding Logistic parameters. In the highlighted case, the growth rate is 52.16 and the lower and upper asymptotes are 66.25 and 68.50 (in MIDI number), respectively, which could be interpreted as the antecedent and subsequent pitches. From this, we infer that the transition interval is 2.25 semitones.

Our design principle was to make each panel provide a core functionality while minimizing unnecessary functions having little added value. As vibratos and portamenti relate directly to the pitch curve, each tab shows the entire pitch curve of the excerpt and a selected vibrato or portamento in that pitch curve. To enable user input, we allow the user to create or delete feature highlight windows against the pitch curve. Playback functions allow the user to hear each detected feature so as to inspect and improve detection results. To enable off-line statistical analysis, AVA can export to a text file the vibrato and portamento annotations and the corresponding parameters.

4. DISCUSSIONS AND CONCLUSIONS

In this paper, we have presented an off-line automatic vibrato and portamento detection and analysis system. The system implements an FDM-based vibrato detection method

and an HMM-based portamento detection method. Vibrato parameters is a natural by-product of the FDM process, and a Logistic Model is fitted to each portamento. The system has been implemented in MATLAB, and the GUI provides intuitive visualization of detected vibratos and portamenti and their properties. User feedback allows for the correction of false positive and false negative errors.

The vibrato detection module currently uses a Decision Tree method for determining vibrato existence. The user can set the vibrato frequency and amplitude ranges to affect the output. A more sophisticated Bayesian approach taking advantage of learned vibrato rate and extent distributions is described in [9]. The distributions can be adapted to each instrument or music genre. While this method has been shown to give better results, it requires training data beforehand.

The portamento detection method sometimes misclassifies normal note transitions as portamenti even though a minimum duration threshold is used to prune the results. We observe that the false positives tend to have low intensity (dynamics) values. Future improvements to the HMM-based portamento detection method could take into account intensity features in addition to the delta pitch curve.

The applications of AVA include music education and expression analysis, and its outputs provide a useful base for expression synthesis and transformation. In instrument learning, AVA can be used to provide visual feedback and quantitative analysis of students' performances, allowing students to inspect their expressive features and adapt accordingly. Furthermore, it is able to integrate other modules, e.g. intonation detection, rhythm detection, to be a completed pedagogical tool. In expression analysis, AVA can be used to quantify musicians' vibrato/portamento playing styles, and the ways in which they use these expressive features. When applied to music of different cultures, it is able to be used to conduct large-scale comparative studies. The analysis results from AVA can also serve as input to expression synthesis engines, or to transform the expressive features in recorded music.

Acknowledgments

This project is supported in part by the China Scholarship Council. The authors would like to thank Siying Wang for discussions of the MATLAB interface.

5. REFERENCES

- [1] C. Palmer and S. Hutchins, *What is musical prosody?* ELSEVIER, 2006, ch. 46, p. 245.
- [2] K. Kosta, O. F. Bandtlow, and E. Chew, "A Change-Point Approach Towards Representing Musical Dynamics," in *Mathematics and Computation in Music*. Springer, 2015, pp. 179–184.
- [3] V. Verfaillie, C. Guastavino, and P. Depalle, "Perceptual Evaluation of Vibrato Models," in *Proceedings of the Conference on Interdisciplinary Musicology (CIM05)*, March 2005.
- [4] L. Yang, E. Chew, and K. Z. Rajab, "Logistic Modeling of Note Transitions," in *Mathematics and Computation in Music*. Springer, 2015, pp. 161–172.
- [5] T. L. Nwe and H. Li, "Exploring Vibrato-Motivated Acoustic Features for Singer Identification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 2, pp. 519–530, 2007.
- [6] T. H. Özaslan, X. Serra, and J. L. Arcos, "Characterization of embellishments in ney performances of makam music in turkey," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2012.
- [7] L. Yang, E. Chew, and K. Z. Rajab, "Vibrato Performance Style: A Case Study Comparing Erhu and Violin," in *Proc. of the 10th International Conference on Computer Music Multidisciplinary Research (CMMR)*, 2013.
- [8] H. Lee, "Violin portamento: An analysis of its use by master violinists in selected nineteenth-century concerti," in *9th International Conference on Music Perception and Cognition, ICMPC9 Proceedings of*, August 2006.
- [9] L. Yang, K. Z. Rajab, and E. Chew, "Filter Diagonalisation Method for Music Signal Analysis: Frame-wise Vibrato Detection and Estimation," *Journal of Mathematics and Music*, 2016, under revision.
- [10] M. Mauch and S. Dixon, "pYIN: A Fundamental Frequency Estimator Using Probabilistic Threshold Distributions," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2014)*, 2014.
- [11] A. de Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *Journal of the Acoustical Society of America*, vol. 111, pp. 1917–1930, 2002.
- [12] S. Rossignol, X. Rodet, P. Depalle, J. Soumagne, and J.-L. Collette, "Vibrato: detection, estimation, extraction, modification," in *Proceedings of the 2nd COST-G6 Workshop on Digital Audio Effects (DAFx)*, December 1999.
- [13] H.-S. Pang and D.-H. Yoon, "Automatic detection of vibrato in monophonic music, Pattern Recognition," *Pattern recognition*, vol. 38, p. Pattern recognition, 2005.
- [14] P. Herrera and J. Bonada, "Vibrato extraction and parameterization in the spectral modeling synthesis framework," in *Proceedings of the Digital Audio Effects Workshop (DAFX98)*, 1998.
- [15] J. Ventura, R. Sousa, and A. Ferreira, "Accurate analysis and visual feedback of vibrato in singing," in *Proceedings of the 5th International Symposium on Communications, Control and Signal Processing*, May 2012, pp. 1–6.
- [16] H. von Coler and A. Roebel, "Vibrato Detection Using Cross Correlation Between Temporal Energy and Fundamental Frequency," in *Audio Engineering Society Convention 131*, October 2011.