Automatic Classification of Orchestral Blend in Multitrack Audio

Julian Vanasse

October 9, 2020

1 Introduction

Timbre blending is a technique at the composer's disposal that is used in orchestration to group voices into blending and contrasting orchestral lines. Instrumental blend has been described in orchestral treatises (Rimsky-Korsakov, 1964; Koechlin, 1954), and more recently has been studied as a perceptual and cognitive phenomenon (Sandell, 1995; Lembke, 2014). Blend can be used to perceptually fuse two or more voices into a single stream, or when contrasted can be used to segregate voices into multiple streams.

This paper introduces a data-driven prediction algorithm that classifies concurrent pairs of voices as either blending or not blending. From monophonic orchestral recordings, a collection of features are extracted from each voice using signal processing and music informatics techniques. These features have been identified from perceptual studies and orchestration treatises as being significant to indicating blend. Feature vectors are then paired with expert orchestral annotations and used to train a Support Vector Machine (SVM) for binary classification. This approach is a marriage of an expert system approach and statistical inference. On the one hand the features used are chosen based on musical qualities identified by orchestrators and music cognition experts, but on the other the degree to which these features effect the classification is decided by patterns extracted from expert annotations.

2 Orchestral Blend

Orchestral blend is a technique used by composers to fuse multiple concurrent voices into a single sonic event. Orchestration treatises, guides to instruct composers on techniques of musical arrangement for orchestra, propose voice pairs that blend. For instance, Rimsky-Korsakov (1964) suggests that woodwind and string instruments blend well.

Blend has also been the subject of perception and cognition research, following the Gestalt frame work developed by Bregman (1990). The law

of common fate suggests grouping of auditory stimulus when change is synchronous and parallel. This is to say that onset synchrony, tonal harmony, synchronous changes in dynamics and simultaneous modulation of frequency can induce the human auditory system to group these sounds into a single event. Spatial position and room acoustics also contribute to grouping (Lembke, 2014).

3 Data

Driving the project is the availability of synthesized multitrack orchestral recordings and expert annotations of orchestral effects. Audio excerpts with expert annotations are 33 in number, spanning the late eighteenth, nineteenth and mid-twentieth centuries and featuring styles from the late Classical period, Romantic period and Impressionism. Excerpts vary in length, the shortest being 8 measures in length and the longest contain dozens of measures. Each excerpt contains multitrack audio for each voice in the score, sometimes as many as fifty instruments.

This section describes the nature of the data used in this project and the care taken to correspond the audio data with the annotation data.

3.1 Multitrack Audio

The multitrack audio used in this project are provided by OrchPlayMusic[1], a firm that specializes in highly realistic orchestral synthesis. These simulations bear the name OrchSim and cover pieces and excerpts from many composers of the Romantic and Impressionist periods. Because the performances are synthesized each monophonic source is completely isolated from the rest of the voices, something which is impossible to accomplish in true orchestral recording. The OrchSim resources have been useful in perceptual studies for this reason, and is used as a pedagogical tool in teaching orchestration using the OrchPlay interface.

The multitrack channels are stored in Logic X[2] project files and must be exported to lossless files to be used. Once extracted, the synthesized orchestras also feature a synchronization track, with a "click" sound on the downbeat of each measure. This synchronization channel is used to divide the musical channels in time, allowing a correspondence between the annotations and the voice audio. Additionally, when these tracks are exported from their Logic X container, audio files are rendered that contain instruments in complete isolation as well as instrument sections. As an example, the two flutes of an excerpt appear in two files in isolation and in a "Flutes.wav" file containing the two summed.

[1] `https://www.orchplaymusic.com/`

[2] `https://www.apple.com/logic-pro/`

Composer	**Work**	**Measures**	**Year**
Berlioz	Symphony Fantastique mvt IV	1 - 77	1830
Bizet	Carmen (Overture)	121 - 147	1875
Borodin	In the Steppes of Central Asia	40 - 71	1880
Brahms	Symphony 4 mvt I	1 - 57	1884
Brahms	Symphony 4 mvt IV	1 - 33	1884
D'Indy	Choral Varie Op 55	70 - 78	1903
Debussy	La Mer mvt I	1 - 141	1905
Debussy	La Mer mvt III	31-52, 171-186	1905
Haydn	Symphony 100 mvt II	1-70	1793
Haydn	Symphony 100 mvt III	50-65	1793
Mahler	Symphony 1	1-22, 356-363	1888
Mendelssohn	Symphony 3 mvt II	1-40, 242-273	1841
Mendelssohn	Symphony 3 mvt IV	161-182	1841
Moussorgsky	Pictures at an Exhibition (Baba Yaga)	106-124	1874
Moussorgsky	Pictures at an Exhibition (Catacombae)	1 - 22	1874
Moussorgsky	Pictures at an Exhibition (Gnome)	57-109	1874
Moussorgsky	Pictures at an Exhibition (Promenade 1)	1 - 24	1874
Moussorgsky	Pictures at an Exhibition (Promenade 2)	1 - 12	1874
Moussorgsky	Pictures at an Exhibition (Samuel)	1 - 9	1874
Moussorgsky	Pictures at an Exhibition (Vecchio Castello)	30-52	1874
Mozart	Don Giovanni (Overture)	1-284	1787
Schubert	Symphony 8 mvt I	1-62	1822
Schubert	Symphony 9 mvt II	300-310	1828
Schubert	Symphony 9 mvt III	187-221, 336-359	1828
Schubert	Symphony 9 mvt IV	543-564	1828
Schubert	Symphony 2 mvt II	300-310	1814
Smetana	The Bartered Bride (Overture)	9-59	1866
Smetana	Ma Vlast mvt II	185 - 228	1882
Vaughan Williams	Symphony 8 mvt I	140-161	1956
Vaughan Williams	Symphony 8 mvt II	71-107	1956
Vaughan Williams	Symphony 8 mvt IV	12-25, 54-96	1956
Verdi	Aida (Act II)	41-57	1871
Verdi	La Traviata (Prelude)	17-37	1853

Figure 1: List of annotated excerpts.

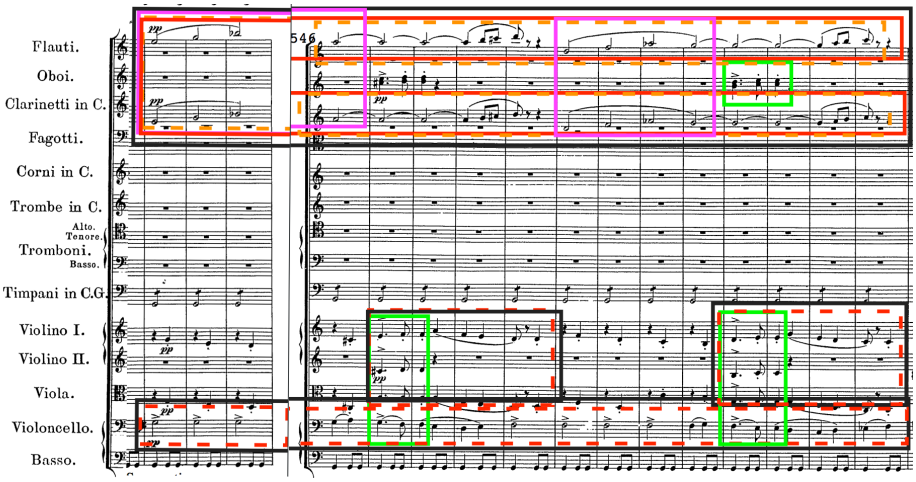

Figure 2: Score annotations from Schuberts Symphony No. 9 (D 944), IV, mm. 543-558, made directly on the score and found in Orch.A.R.D. Blend is indicated in the red boxes enclosing Flute 1 and Clarinet 1.

3.2 Annotations

Scores have been annotated, marking orchestral effects, as part of the Analysis Creation and Teaching of Orchestration (ACTOR) project[3]. Originally these annotations were made directly onto scores as in Figure 2. These were transferred to computer-readable Excel format, in tables listing the instruments involved, the temporal (measure) locations of the blend events and the expert rating. These tables are parsed to a format that indicates whether or not a feature vector representing a pair of voices blends, allowing a supervised learning task to train on features extracted from musical excerpts. The transcription to be scraped is illustrated in Table 1.

4 Problem Statement

Because blend is a phenomenon caused by the musical interaction of a minimum of two voices, the classification scheme approaches the problem of classifying pairs of voices as either of the blend category or the non-blend category. The task is thus to take as input two audio vectors from a multitrack audio recording and return a class value $[1, -1]$. But, blend occurs locally in time. It is exceedingly rare for two voices to blend for the duration of a piece. Rather, two voices may blend over a phrase and then cease to blend for the next phrase.

The expert annotations of orchestration effects are compiled in the Orchestration Analysis and Research Database (Orch.A.R.D.). These annota-

[3] `https://www.actorproject.org/`

Schubert - Symphony 9 - IV (mm. 543-564) — Blends

Blend ID	Measures x to y	Measures	Instruments	
BLE2134	543 to 558	543	Flute 1, Clarinet 1	
		544	Flute 1, Clarinet 1	
		545	Flute 1, Clarinet 1	
		546	Flute 1, Clarinet 1	
		547	Flute 1, Clarinet 1	
		548	Flute 1, Clarinet 1	
		549	Flute 1, Clarinet 1	
		550	Flute 1, Clarinet 1	
		551	Flute 1, Clarinet 1	
		552	Flute 1, Clarinet 1	
		553	Flute 1, Clarinet 1	
		554	Flute 1, Clarinet 1	
		555	Flute 1, Clarinet 1	
		556	Flute 1, Clarinet 1	
		557	Flute 1, Clarinet 1	
		558	Flute 1, Clarinet 1	

Table 1: Transcribed annotation from score to table, for the same Schubert excerpt as Figure 2.

tions indicate what measures (temporal locations) blend occurs in a musical excerpt and what voices are involved, with the temporal resolution of about one musical measure. For this reason, the source audio is categorized in small coincident blocks, each one measure in length.

4.1 Parsing Voice Names

A feature of the data that obfuscates the patterns present is the inconsistency of names used for the names of voices in the orchestra. The case of the flute will provide an illustrative example. In the Schubert excerpt observed thus far there is a single flute voice, but appears in the file for that isolated voice and for the file that captures the flute section (see Table 4.1).

The isolated monophonic tracks consistently named with diminutives, such as “Fl1” for Flute 1, “Ob1” for Oboe 1, etc. Annotations are made for individual voices but do not use diminutive forms to name the voices. This has necessitated an effort to create a transcription function to go from diminutive form to the standard form using a custom dictionary. The dictionary was created by hand, as to generate one programmatically would have taken the same amount of time for a medium sized set of names like these. Voices are separated by plural versus singular, but ultimately only singular voices correspond to annotation names.

File Name	Voice
(*Isolated Voices*)	
`Schubert_Symph9_iv(543-564)_003_Fl1.aif`	Flute 1
`Schubert_Symph9_iv(543-564)_008_Ob1.aif`	Oboe 1
`Schubert_Symph9_iv(543-564)_009_Ob2.aif`	Oboe 2
⋮	⋮
(*Sections*)	
`Schubert_Symph9_iv(543-564)_Flutes.aif`	Flute 1
`Schubert_Symph9_iv(543-564)_Oboes.aif`	Oboe 1 + Oboe2

Table 2: Individual and section audio files. The '+' denotes the sum of two individual channels.

Complicating the issue, the annotation tables are not consistent in their naming. For instance, a blend including the part Violin 1a may be marked in one table as "violins 1a" or "Violin 1". These inconsistencies make corresponding between the audio tracks and annotations frustrating. Ultimately these annotations were edited by hand for consistency.

4.2 Balance or Lack Thereof

A major concern for data driven methods is the class imbalance problem (Ling and Sheng, 2010) (Japkowicz and Stephen, 2002). This problem occurs when classes are not evenly distributed amongst the data, and is especially severe when the minority class represents less than 1 percent of examples. A symptom of this problem is that cost-insensitive classifiers will uniformly predict every sample as the majority class.

Blend occurrence in orchestral music of the eighteenth and nineteenth century is sparse. Consider the Schubert excerpt from Figure 2 and Table 1. Only two voices blend out of 14 voices. Over the duration of this excerpt, only one pair of 91 possible voice pairs blend, a presence of positive examples at only slightly greater than 1 percent. In some excerpts it is less severe, but blend is almost always in the minority.

To mitigate this problem an oversampling technique was chosen to oversample the minority data as suggested in Japkowicz and Stephen (2002). Positive example data are randomly duplicated to balance the set between negative and positive class examples.

4.3 Temporal Divisions

A musical excerpt in multitrack form is first divided in time. The division in our case will necessarily match the resolution of the expert annotations, that is one musical measure in duration. Other divisions are possible, but in choosing a data-driven approach the choice of a finer resolution or a

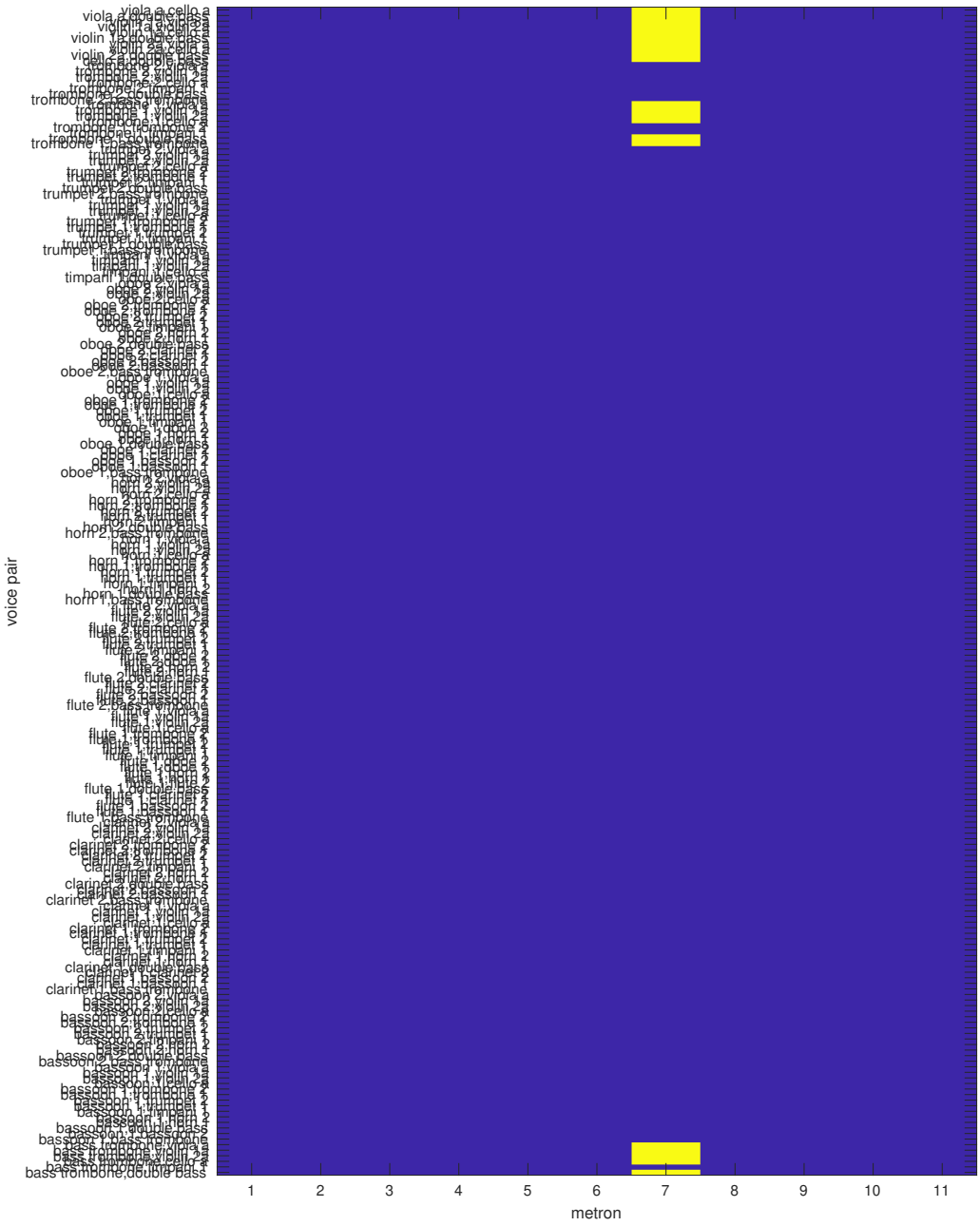

Figure 3: Number of positive examples vs negative examples, to demonstrate sparsity of positive class. This is the ground truth matrix for Schubert Symphony No. 9, II, mm. 300-310. It contains 210 pairs of voices over 11 measures, leading to a matrix containing 2310 values. Only 21 of those values indicate a blend, all of them in measure 7.

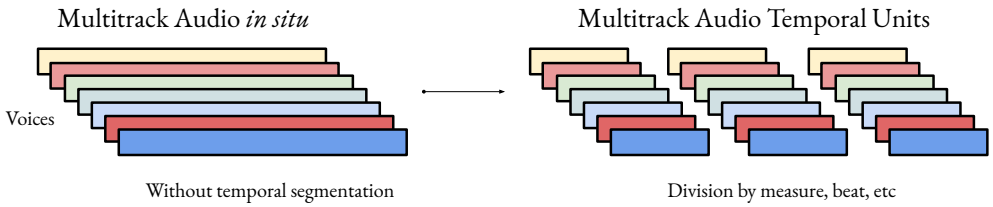

Figure 4: Temporal segmentation process.

resolution of arbitrary length were not explored. This process is visualized in Figure 4.

The multitrack data is accompanied by a metronome, in recording known as a *click* track. This metronome audio is entirely percussive and is the perfect fodder for an onset detection routine. Here onset detection is performed using the MIRToolbox with additional custom thresholding.

4.4 Feature Extraction

Salient information is extracted in two stages: 1) each monophonic audio vector is transformed to a set features describing musical *performance*, then 2) from each pair of features derived from concurrent voices is transformed to a vector that describes musical *interaction*. Here, performance is used to denote the set of features of the voice in isolation, such as pitch, amplitude envelope, spectral centroid, onset location. The interaction features describe how simultaneous voices relate to one another. These features include the harmonic interval relationship between voices, the degree to which their onsets are synchronous, and the degree of spectral overlap. This process is illustrated in Figure 5

Broadly, features can be categorized as either representing symbolic information (*e.g.* onset location) representing non-symbolic information. This dichotomy is important, as both will have an impact on prediction, but the former may also be derived from the score and the latter has no symbolic analogue and can only be heard in the audio domain.

4.4.1 Performance Features

Performance features extracted and stored in a container, an object scripted for such a purpose. Some features, like pitch, will encode local behavior of the signal. Others, like envelope, capture long-term evolution over the measure. Much of the feature extraction in this first performance stage makes use of the MIRToolbox (Lartillot et al., 2008).

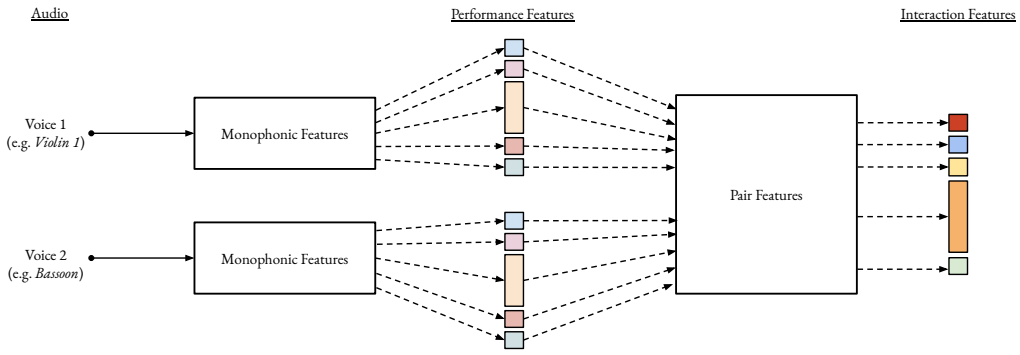

Figure 5: Feature extraction circuit. Feature vectors are extracted from all monophonic sources, transforming from the audio domain to the *Performance Features* domain. Then, *Performance Features* vectors are compared for each pair of coincident voices and a vector in the *Interaction Features* domain.

Envelope The amplitude envelope characterizes the evolution of the signal over the measure (or other division duration). It is also useful in novelty detection algorithms. The envelope is extracted from the monophonic source audio by first computing the analytic signal using an approximate Hilbert Transform, then half-wave rectifying the result. This intermediate signal is then low-pass filtered and downsampled (Lartillot et al., 2008).

Pitch The pitch tracks the fundamental frequency of the voice over the signal duration. Interval relationships and harmonic (partial) overlap play a large role in blending (Lembke, 2014), and the pitch vector is crucial to extracting these relationships. To compute high accuracy pitch vectors for each voice, a data-driven time domain approach is used. The extraction technique, called A Convolutional Representation for Pitch Estimation (CRePE), uses a pretrained network to extract pitch from short time domain segments of audio (Kim et al., 2018). A drawback of the CRePE method is that it is accurate only for audio sampled at 16kHz. Thus, all audio files were downsampled from 44.1kHz and processed.

Pitch tracking is especially important for extracting performance techniques not present in the score. Chiefly, vibrato may be marked in the score, but the rate and intensity are only available from the performance.

Spectral Centroid The evolution of the spectral centroid in time is a useful feature in encoding the timbre and timbral development of a voice. (Lembke, 2014) finds that voices with low spectral centroids blend well with other voices, regardless of their centroid, with low centroid voices blending best with each other.

MFCC and Δ MFCC Mel-frequency cepstrum coefficients (MFCCs) and Delta MFCCs are a crude approximation of timbre used in speech and music processing. This vector of 12 values that forms the MFCC is the average of a transformed critical-band filter bank. The Delta is its first time difference, a spectral flux in the MFCC domain. While not particularly meaningful as a model of perception, their success in other music tasks concerning timbre (particularly instrument classification) encouraged its addition to the feature set.

Modulation Rate Representing musical frequency modulation is still an open question. The approach taken in this project is applied to a narrow class of possible frequency modulation. It can be verified from the audio files provided that pitch is near exact MIDI values. So when vibrato is present, it is relative to the exact MIDI pitch. Thus, after extracting the pitch using CRePe, to compute a vibrato rate the pitch vector is subtracted from a version of itself quantized to MIDI value.

$$\texttt{vibrato} = \texttt{freq2midi}(\texttt{pitch}) - \texttt{round}(\texttt{freq2midi}(\texttt{pitch}))$$

The goal is to remove the "DC" component that the pitch adds and shifting the modulation to be zero mean.

4.4.2 Interaction Features

Blend in a sense is a measure of similarity between simultaneous musical passages. Monophonic excerpts that have identical onset synchrony and feature octave interval relation are much more likely to blend than two excerpts that are asynchronous and in dissonant relation. Timbral features are also crucial, but interactions are more complex.

Envelope Correlation This metric can be compared with onset synchrony. It is the Pearson's correlation coefficient between the amplitude envelopes of two voices. This feature is meant to capture the degree to which the envelopes are collinear, giving a rough idea that the envelopes are synchronous.

Tonal Relationship This feature is a way to capture the harmonic interval relationship between the pitch features of concurrent performances. Each the elements of each pitch vector are rounded to the nearest MIDI pitch. Next, the two vectors are subtracted. The final feature is a vector of 24 elements where each value represents the percentage of the duration on which the pitch was in a certain interval.

For instance, if the pitch was in octave relation over the entire duration, the first element would have value 1 and the rest would be zero. Transitions

between intervals are ignored. Perhaps a kind of Δ`tonal(...)` can be developed.

Parallelism Musical parallelism is the degree that melodic lines share the same motion. Voices are perfectly parallel when the frequencies they produce are separated by a constant ratio. Sometimes this will be an octave, a ratio of 2:1, should a composer intend to reinforce a phrase. Other ratios, like those of a musical fifth/forth or a musical sixth/third, ratios of 3:2 and 5:3 respectively[4], are common.

To assign a metric to the degree of parallelism, a representation for melodic intervals (the frequency change from one note to another) must be created. First, like the Tonal Relationship representation, the frequency tracker for each voice is mapped to a logarithmic *pitch* domain using the `freq2midi()` function. Then the pitch for each voice is rounded to the nearest whole number to reduce the effects of pitch vibrato.

$$p_i = \text{round}(\texttt{freq2midi}(f_i))$$

The melodic interval is the change in pitch between notes. The representation p_i in this musical context appears to be a piecewise step function. While a note is sounding, the pitch is relatively constant, and change is rapid to the next note. A simple way to represent the melodic interval is to take the first difference of p_i, revealing delta-like distributions at each note change. The height of each distribution is the number of half-steps between the two notes. A positive bump corresponds to an upward melodic change and a negative bump to a descending melodic change.

The method of comparing two interval representations is again Pearson's correlation coefficient. The notion here is should notes change at the same time, the direction of change in pitch is captured in the interval representation. If two voices change at the same time and in the same direction, the correlation of these two vectors should be close to 1, indicating parallel motion. Should these vectors change at the same time and in opposite directions, the correlation coefficient will be closer to -1 indicating contrary motion. A coefficient close to zero indicates notes not changing at the same time (see onset synchrony).

As stated, should notes change at *exactly* the same index in the interval representation, this scheme will exactly encode parallelism. But because the measurement of pitch is prone to noise, additional measures are needed to ensure accurate analysis. The interval vector is sparse, populated by impulse-like distributions. A pair of voices with synchronous changes will have like distributions but due to small errors in the pitch extraction the impulses representing interval change may be offset from one another by a

[4]ratios are given in just intonation for simplicity.

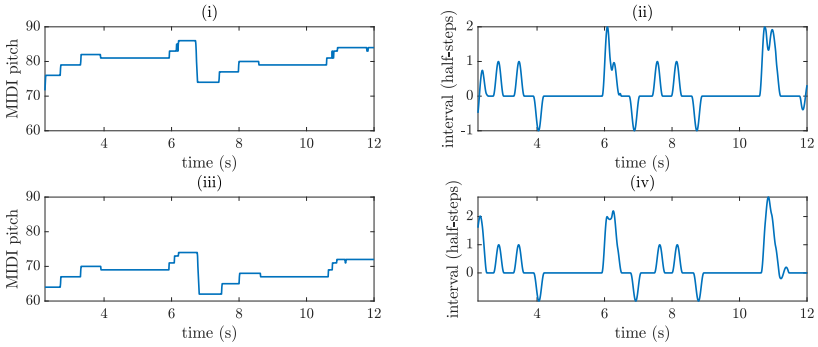

Figure 6: Pitch extraction and melodic interval vectors. (i) and (ii) are the pitch and melodic interval vectors for the Flute 1 from the earlier Schubert excerpt. (iii) and (iv) are the same for Clarinet 1 of the same excerpt. Observable from the score in Figure 2, these two voices are parallel through the excerpt duration. Their Pearson correlation is $r = 0.703$.

few indices. To force interaction between nearby impulses a lowpass filter is used to spread the energy of the impulse into adjacent bins.

5 Pipeline

The representation of annotations and audio described above have been implemented in MATLAB in directories `ANNOTATION_PIPELINE` and `AUDIO_PIPELINE`, respectively. This section outlines the use of the scripts contained within those directories.

5.1 Audio Pipeline

The audio pipeline starts with raw audio vectors and ends with temporally segregated representations of pairs of voices. Several objects have been created and cascade into one another to produce the final result.

Nb. each excerpt is thought of as a matrix with voices as the rows and temporal units as the columns. Each cell in the matrix represents a small amount of time. As the annotations are made on a per-measure basis, the *metron* is chosen to be a musical measure. But this is flexible, and new annotations with a finer or coarser metrical notation will also work in the framework.

Object	Description
`AudioExcerpt`	Takes raw audio as input, divides into *metrons*.
`VoiceFeatureExcerpt`	Provided with an `AudioExcerpt`, for each *metron* a set of performance features is computed.
`PairFeatureExcerpt`	Provided with a `VoiceFeatureExcerpt`, the musical interaction between pairs of voices is computed.

Figure 7: Excerpt description also commented at the top of each class MATLAB file.

5.2 Annotation Pipeline

Much of the work on the annotations was done through careful combing over. To catch inconsistencies names were inventoried and corrected by hand. The result is a simple script that parses the annotations and returns a one-hot matrix. This has one file, `readAnnotations`, that requires the features of the audio to be precomputed.

References

Bregman, A. (1990). *Auditory Scene Analysis: The Perceptual Organization of Sound*. MIT Press.

Japkowicz, N. and Stephen, S. (2002). The Class Imbalance Problem: a Systematic Study. *Intelligent Data Analysis*, 6(5):429–450.

Kim, J. W., Salamon, J., Li, P., and Bello, J. P. (2018). Crepe: A Convolutional Representation for Pitch Estimation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 161–165. ISSN: 2379-190X.

Koechlin, C. (1954). *Trait de l'orchestration: en quatre volumes*. M. Eschig, Paris.

Lartillot, O., Toiviainen, P., and Eerola, T. (2008). A Matlab Toolbox for Music Information Retrieval. In Preisach, C., Burkhardt, H., Schmidt-Thieme, L., and Decker, R., editors, *Data Analysis, Machine Learning and Applications*, pages 261–268. Springer Berlin Heidelberg, Berlin, Heidelberg.

Lembke, S.-A. (2014). When timbre blends musically: perception and acoustics underlying orchestration and performance. page 193.

Ling, C. X. and Sheng, V. S. (2010). Class Imbalance Problem. In Sammut, C. and Webb, G. I., editors, *Encyclopedia of Machine Learning*, pages 171–171. Springer US, Boston, MA.

Rimsky-Korsakov, N. (1964). *Principals of Orchestration*. Dover, New York.

Sandell, G. (1995). Roles of Spectral Centroid and Other Factors in Determining "Blended" Instrument Pairings. *Music Perception*, Winter(2):209–246.