

CS229 Project: Musical Alignment Discovery

Woodley Packard

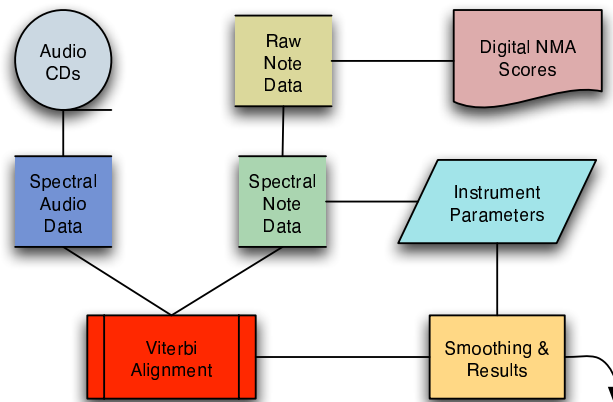
December 16, 2005

Introduction

Logical representations of musical data are widely available in varying forms (for instance, MIDI files are ubiquitous on the internet). Matching acoustic recordings of musical performances are also available from CD stores, online music stores, or other repositories. However, due to variation in tempo and dynamic in performances, the time series alignment between logical and acoustic data is usually nonobvious.

My project aims to automatically discover that alignment. I chose to restrict my explorations to the music of Mozart, because I have access to a large homogeneous library of both logical and acoustic data for that particular composer. One advantage of this restriction is that the set of musical instruments is limited to a few dozen. I take advantage of this by extracting models for the sound of each relevant instrument (overtone coefficients).

This chart shows a high-level outline of the system's architecture:



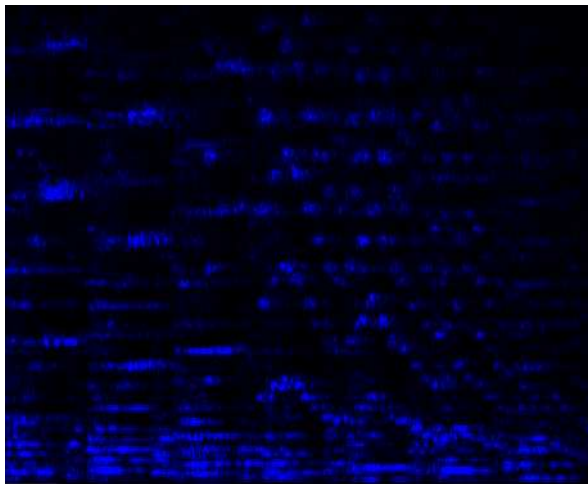
Data Sets

The logical note data input to my project is an accurate textual representation of a physical printed page of musical score, extracted (by a different project) from the comprehensive Neue Mozart Ausgabe (a complete edition of scores to Mozart's music). These data are handled by an external interpreter:

Input Note Data

By modifying the playback facilities of the interpreter, I preprocessed the music into a table of note data containing, for each note in the piece, the *logical start time*, the *pitch*, the *volume*, the *instrument*, the *duration*, and the *measure number*.

The acoustic data came from a commercial compact-disc recording of the symphony. I extracted a spectrogram (see image) from the digital samples using a Gaussian window size of 2048 samples and a step of 512 samples. The data were at 44100 Hz.



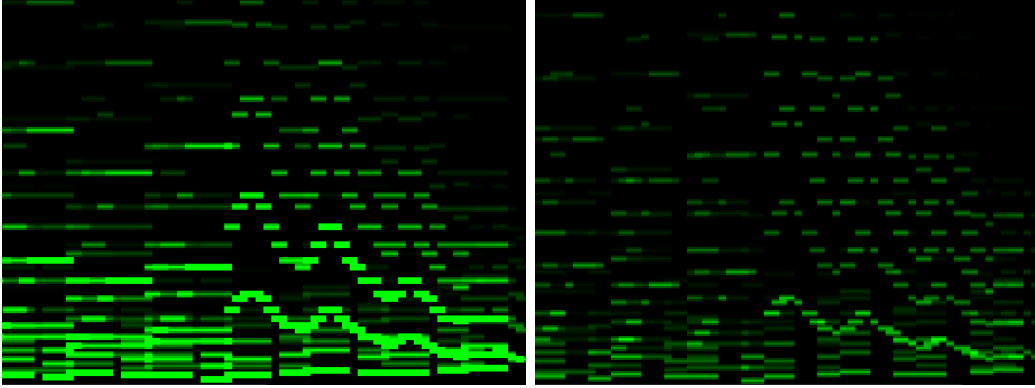
Acoustic Spectrogram

Generative Model

Using the note data table, I constructed an estimation of a "logical" spectrogram by computing the frequency f_m for each note and contributing an exponentially decaying rectangle at each overtone of the fundamental. I used an equal tempered scale, using the following equation for the fundamental frequency [1]:

$$f_m = 440 \cdot 2^{\frac{m - M_{440}}{12}}$$

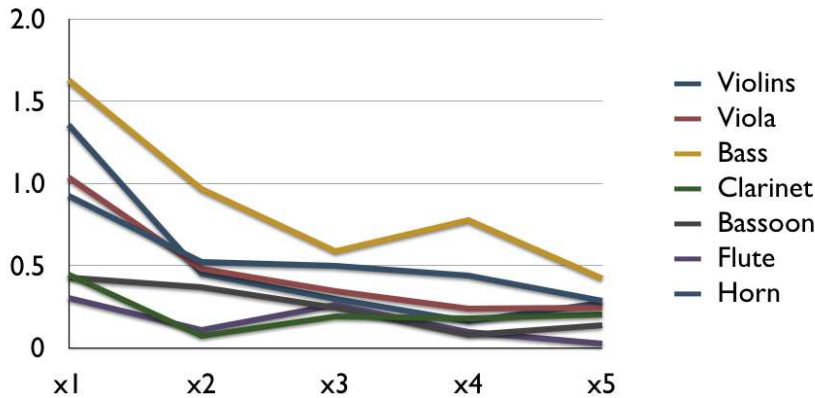
where M_{440} is the MIDI encoding of middle A.



Left: Uniform Predicted Spectrogram; Right: With Learned Instrument Spectra

For the initial alignment, I assumed the spectral properties of all instruments were identical, namely five exponentially decaying evenly spaced overtones. Then, assuming the alignment was correct, I applied a linear regression to find the maximum likelihood spectral characteristics of each instrument given the observed spectrogram. The parameters my system learned were the coefficients of the first five overtones for each instrument. These parameters also implicitly included an overall scaling factor accounting for the difference between the suggested dynamics (loudness) in the score and the average recorded dynamics for each instrument.

After estimating the instruments' spectral properties, I used them to build a superior predicted spectrogram from the logical note data. Then I realigned, producing new training data for the instrument property estimation, etc. By repeating this process (similar to EM), my system was able to learn the properties of the instruments with higher accuracy. The parameters converged after three such iterations.

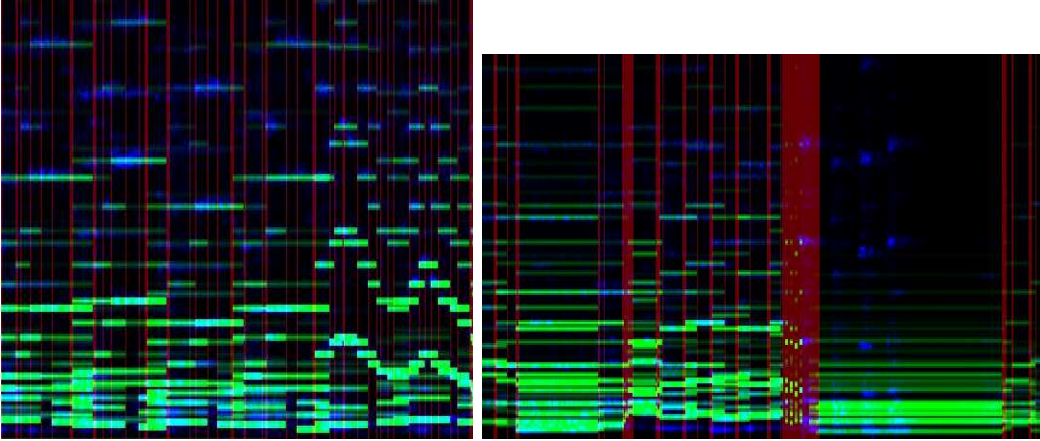


Spectral overtone coefficients learned for several instruments

I assumed the existence of a similarity metric $f(x, y)$ comparing a "logical" spectrogram slice to an acoustic spectrogram slice. I tried several metrics for this purpose, including L^2 distance, a Gaussian kernel, and a simple dot-product (representing the cosine of the angle between the two spectra). The simple dot-product performed noticeably better than the other candidate metrics.

Alignment

The literature contains many records of experiments of this type using hidden Markov models (HMMs), e.g. [2], [3], [4]. Alternative approaches have been studied with varying degrees of success (e.g. [5] investigates a discriminative learning approach), but most other models require supervised learning. By contrast HMMs can learn to align time series in an unsupervised fashion. Treating the predicted "logical" spectrogram as a linear HMM with output likelihoods given by the similarity metric, I used the Viterbi algorithm to compute the highest likelihood alignment between the "logical" spectrogram and the acoustic spectrogram:



Top: Typical Alignment; Bottom: A Bad Spot

In the example pictured, there were roughly 7 times more spectral slices than logical slices. The aligned image shows the original acoustic spectrogram in blue, with the hypothesized alignment of the logical spectrogram overlayed in green. The red lines on the aligned image indicate transitions between logical spectrogram slices. As can be seen, the alignment found in the case on top is reasonable.

One problem with HMM alignment is that there is no good way to specifying a preferred number of times to stay in a given state. As such, the alignment tends to be locally uneven, even if globally it is a good fit. This can be seen in lower segment above. On average, only $\frac{1}{7}$ of the slices are transitions (red), but in this particularly bad segment, the variance is very high.

To solve this problem, I built a confidence metric to discern which segments of the alignment were smooth and which segments were not. The confidence metric $m(t)$ observes the deviation $d(t)$ between the instantaneous tempo of the alignment and the local average tempo:

$$d(t) = \left| \frac{f(t+n) - f(t-n)}{2n} - f'(t) \right|, \quad m(t) = \exp\left(-\frac{d(t)^2}{2\sigma^2}\right)$$

where f is the Viterbi alignment, n is a parameter controlling the breadth of the local average, and σ^2 is a normalizing constant to put the metric in a useful range. In situations with high variance, the deviation is large and so the confidence metric is near 0, and in situations with low variance the confidence metric is near 1.

The term $f'(t)$ in the above equation for $d(t)$ requires slight explanation, since the available representation of $f(t)$ is not a continuous function. Instead it is a discrete mapping from audio CD time slices to note data time slices. I used a numerical differentiation technique to estimate $f'(t)$ for the underlying latent continuous alignment function.

Smoothing is applied to areas with low confidence, by mixing the aligned values with linear predictions from nearby high-confidence areas.

Results and Applications

The result of the entire process is an alignment function mapping from audio CD time codes to note data time codes. Not having a hand-aligned test set, I am unable to provide a numerical accuracy measurement; [3] and [4] also found it difficult to measure accuracy numerically, although [2] invented a (somewhat discouraging) metric and [5] has a metric based on hand-aligned test data. However, I built a tool to visually inspect the learned alignment between the spectrograms. The visualization made it clear that the alignment is generally quite accurate. More specifically, it is quite good at lining up passages with quickly changing tonalities and moving notes. Slow-changing segments with little tonal (and hence spectral) variation sometimes cause reduced accuracy, but even in such cases the proposed alignment tends to be within a small fraction of a second of the intuitively best alignment.

One possible application of this technique is, as I already explored, automatically learning characteristics of musical instruments. Abstract spectral characteristics of musical instruments could be used in musical synthesis or completely automatic music recognition. Other potential applications include enhancing playback in score studying and browsing, and augmenting a musical listening experience with visual information about the music.

References

- [1] Campbell, Jim: *The Equal Tempered Scale and Peculiarities of Piano Tuning*.
<http://www.precisionstrobe.com/apps/pianotemp/temper.html>
- [2] Raphael, Christopher: *Automatic Transcription of Piano Music*. Proceedings of ISMIR, 2002.
- [3] Orio N., Dechelle F. (2001) *Score Following Using Spectral Analysis and Hidden Markov Models*, Proceedings of the ICMC, 151-154.
- [4] Cano P., Loscos A., Bonada J. (1999) *Score-Performance Matching Using HMMs*, Proceedings of the ICMC, 441-444, 1999.
- [5] Shalev-Shwartz S., Keshet J., Singer Y. (2004) *Learning to Align Polyphonic Music*, Proceedings of ISMIR, 2004.