Mixing with Mozart

Sumit Basu

Microsoft Research
sumitb@microsoft.com

Abstract

A variety of tools exist in hardware and software for mixing dance music. These work by estimating the "beats-per-minute" count of music with heavy beats. These tools aid a DJ in finding the appropriate speed change and time shift to smoothly combine or transition between two pieces of music. In this work, we present a method for finding these alignments and combining a wider class of songs with dance music (e.g., Mozart with techno). We do this by jointly optimizing the energy alignment of both signals instead of attempting to detect individual beats/tempos. Though computationally intensive if naively computed, we introduce an approximation to greatly speed up the evaluation of the tens of thousands of possible matches. This results in a set of a few top choices for alignment parameters. We also develop a measure of the quality of the match to help assess whether the best alignment is actually a good fit. We show a variety of results demonstrating the use of this method.

1 Introduction

DJing has become a popular and successful art; the DJ and her craft have come to permeate everything from nightclub scenes to clothing commercials. One of the fundamental aspects of this craft is music mixing, blending song A smoothly with song B, typically to transition from one piece to the next, but sometimes only to enhance the sound of both pieces. Given song A, this requires determining five parameters: (1) which song to mix it with, (2) where in song A to do the mixing/transition, (3) where in song B to mix from, (4) the timescale adjustment necessary to align A and B, and (5) the time offset required to align A and B.

Over time, a variety of tools have been developed in hardware and software to help DJs with this process, mostly with respect to the last two parameters. In general, these tools estimate the beats per minute (BPM) of each song. The DJ can then change the speed of the first/second song until they match, and manually find an offset to match up the beats. In more sophisticated software/hardware, the system automatically determines the offset as well by finding the locations of the beat sounds.

In this work, we attempt to extend the range of music DJs can use by doing automatic alignment of music from a variety of genres, for instance mixing Mozart's K. 331 with a piece of heavy techno. Our method looks across the two-dimensional space of parameters (4) and (5) above – for a variety of possible timescalings of song B, it tests a wide range of possible offsets, and returns the top few matches that the DJ can preview and then choose from. It also aids in determining parameters (1) through (3) by returning a suitability score, which describes how strong the given match is. However, the bulk of the responsibility for choosing the next song is still on the DJ, for it is up to her to know what songs will make sense together – we can only help with the assembly. For instance, even though two pieces of music in different keys may line up well, the combination of the melodic aspects of the two may still sound terrible.

The closest prior work in this area is in estimating the beat structure of a single piece of music. Scheirer (1998) approaches this via correlations across filter banks, while Laroche (2001) has a probabilistic approach that allows for variation in the beat. Goto and Muraoko have a long history of work in this area, and have a sophisticated multi-agent method for estimating the rhythm in drumless signals (Goto and Muraoko 1997). All of these methods are quite successful but are still prone to problems. The main difference between our work and these approaches to beat tracking is that instead of individually trying to determine the tempo/phase of each song, we consider *both songs together*. In this way, we are able to compute the best possible alignment against our metric without ever determining the BPM of either song.

We will begin by describing our method for aligning the two pieces, including a fast mechanism for computing the different timescales that allows for real-time application of the algorithm. This method will produce several match candidates. We will then develop a metric for the suitability of each match to help the user select among these. Finally, we will show our results on a variety of types of music and discuss failure modes and future work.

2 The Alignment Method

The method works on two arbitrary pieces of two songs, to be referred to as signals *a* and *b*. The core algorithm is outlined below, followed by a detailed description of each step:

1. Compute the frame-based energy for each song, E_a and E_b
2. For each time-scaling and time shift of E_b within a specified range:
   a. Compute the scaled and shifted version of E_b (this will be referred to as E'_b)
   b. Measure the alignment of E_a and E'_b
3. Find/present the set of best alignments from step 2 and allow the user to choose among them
4. Compute the time-scaled and shifted version of *b* (referred to as *b'*) for the chosen shift/scaling
5. Find the energy scaling for *a* and *b'* and combine the final signals

2.1 Computing Frame-Based Energy

To compute the energy of signal *a*, we first break it up into non-overlapping windows (of *N* samples each). We then compute the energy of each frame, without multiplying by a tapered window as is typical in frame-based energy computations (Rabiner and Schafer 1978):

$$E_a[k] = \left(\sum_{n=kN+1}^{kN+N} a[n]^2 \right)^{1/2}$$

This results in the energy signal E_a. For the results in this paper, we used a sampling rate of 44.1 kHz and a window size *N* of 512 samples, corresponding to 12 ms or about 86 frames per second. The reasons for this unconventional choice of windowing and the lack of overlap are due to the time-scaling operations we will perform on the energy signal as we will show in the next section.

In the figures below, we show the energy signals for a piece of classical music and a piece of dance music. Note that while there is a clear, repetitive beat structure in the dance piece, there is little such information in the classical piece.

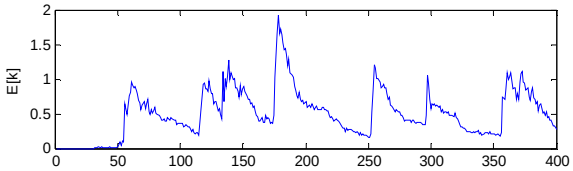

Figure 1: Energy signal for part of a classical piece (Mozart's K. 331, 1st movement)

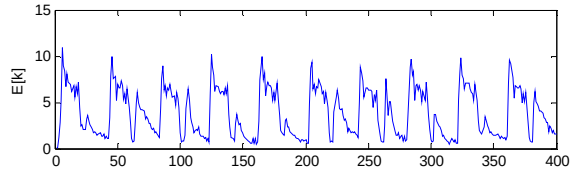

Figure 2: Energy signal for a piece of dance music (The Realm's "Breakdown")

2.2 Iterating Over Scales and Shifts

We then iterate over all scales and shifts of E_b within some specified range, in our case scalings of 0.5 to 2.0 times its current length, with an increment of 0.01 (150 scales) and a correlation range of 100 samples (each sample corresponds to a 12 millisecond energy value, so this is 1.2 seconds). This is a total of 100*150 or 15,000 different modifications of the E_b and must be computed very quickly – in our implementation, this takes only 0.5 seconds. Ideally, we would first compute the scaled version of *b* and then compute its energy, but this would be prohibitively expensive. To accomplish the rescaling in real time, we approximate the energy of the time-scaled signal by timescaling the original energy signal. We do this via a linear resampling of E_b: For each floating point scalefactor *s* in the specified range (i.e., resampling E_b at *s* times its current rate), we approximate the energy of the time-scaled signal at index *n* as follows:

$$f = sn - floor(sn)$$
$$E'_{b,s}[n] = (1-f)E[floor(sn)] + fE[floor(sn)+1]$$

Because we are not windowing the signal, we have the convenient property that the time-scaled version of the energy signal (E'_b) closely approximates the energy of the time-scaled signal ($E_{b'}$). We will now show why this is the case via an example: consider that signal *b* was to be slowed down by a factor of exactly two via linear interpolation to form *b'* (i.e., *s*=0.5). We can then express the precise values for *b'* and for the ideal energy of the time-scaled signal as follows:

$$b'[2n] = b[n]$$
$$b'[2n-1] = \frac{b[n]+b[n-1]}{2}$$
$$E_{b'}[2k] = \left(\sum_{n=2kN+1}^{2kN+N} b'[n]^2 \right)^{1/2} = \left(\sum_{n=kN+1}^{kN+N/2} b'[2n]^2 + \sum_{n=kN+1}^{kN+N/2} b'[2n-1]^2 \right)^{1/2}$$
$$E_{b'}[2k] = \left(\sum_{n=kN+1}^{kN+N/2} b[n]^2 + \sum_{n=kN+1}^{kN+N/2} \left(\frac{b[n]+b[n-1]}{2} \right)^2 \right)^{1/2}$$
$$E_{b'}[2k] = \left(\sum_{n=kN+1}^{kN+N/2} b[n]^2 + \frac{1}{4} \sum_{n=kN+1}^{kN+N/2} b[n]^2 + \frac{1}{4} \sum_{n=kN+1}^{kN+N/2} b[n-1]^2 + \frac{1}{2} \sum_{n=kN+1}^{kN+N/2} b[n]b[n-1] \right)^{1/2}$$

If the signal is not varying too quickly and $b[n] \approx b[n-1]$, we can see that

$$E_{b'}[2k] \approx \left(2 \sum_{n=kN+1}^{kN+N/2} b[n]^2 \right)^{1/2}$$

$$E_{b'}[2k+1] \approx \left(2 \sum_{n=kN+N/2+1}^{kN+N} b[n]^2 \right)^{1/2}$$

$$\left(E_{b'}^2[2k] + E_{b'}^2[2k+1]\right)^{1/2} \approx \left(2 \sum_{n=kN+1}^{kN+N} b[n]^2 \right)^{1/2} = \sqrt{2} E_b[k]$$

In other words, the energy of the superframe composed from the corresponding frames of $E_{b'}$ (2k and 2k+1) has the same energy as frame k in E_b, modulo a scalefactor of $\sqrt{2}$, since there is now twice as long a frame to contend with. If we use the same framesize in the stretched signal and we have the further property that the energy is not changing rapidly from frame to frame, i.e., $E_{b'}[2k] \approx E_{b'}[2k+1]$, we see that the energy of the time-scaled signal is indeed approximately equal to the energy of the corresponding location in the original signal:

$$\begin{aligned}
&E_{b'}[2k] \approx E_{b'}[2k+1] \\
&\Leftrightarrow \left(2 \sum_{n=kN+1}^{kN+N/2} b[n]^2 \right)^{1/2} \approx \left(2 \sum_{n=kN+N/2+1}^{kN+N} b[n]^2 \right)^{1/2} \\
&\Leftrightarrow \sum_{n=kN+1}^{kN+N/2} b[n]^2 \approx \sum_{n=kN+N/2+1}^{kN+N} b[n]^2 \\
&\Leftrightarrow \sum_{n=kN+1}^{kN+N} b[n]^2 \approx 2 \sum_{n=kN+1}^{kN+N/2} b[n]^2 \\
&\Leftrightarrow \left(\sum_{n=kN+1}^{kN+N} b[n]^2 \right)^{1/2} \approx \left(2 \sum_{n=kN+1}^{kN+N/2} b[n]^2 \right)^{1/2} \\
&\Leftrightarrow E_b[k] \approx E_{b'}[2k]
\end{aligned}$$

It is thus reasonable for us to approximate the energy of the timescaled signal ($E_{b'}$) by the time-scaled energy signal (E'_b). To demonstrate this effect, we show the energy signal from the previous example, rescaled to twice its length (E'_b), compared to the energy signal for the signal *b* itself rescaled to twice its length ($E_{b'}$).

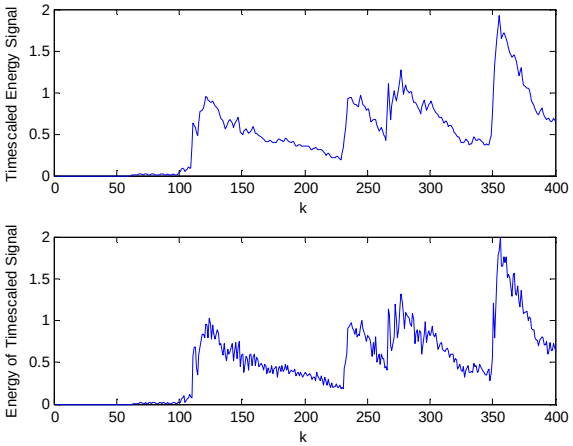

Figure 3: Our approximation, the time-scaled energy signal $E'_b[k]$, vs. the true energy of the time-scaled signal $E_{b'}[k]$ for a scaling of s=0.5. Note the artifacts in the upper signal due to our approximation.

While the resulting signals are very similar, there are visible differences, in that the approximation (top) is a smoothed version of the actual signal, as expected by our development. However, a timescaling of 0.5 is at the very edge of the range we are considering. For a smaller scaling, s=0.9, note that the signals are nearly identical:

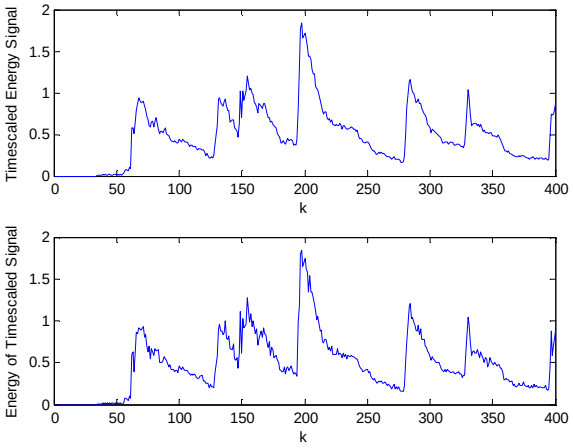

Figure 4: Our approximation, the time-scaled energy signal $E'_b[k]$, vs. the true energy of the time-scaled signal $E_{b'}[k]$ for a scaling of s=0.9. Note that the artifacts are much reduced.

After this, we need to compute the alignment score for this scaled energy signal for all possible shifts in the range specified against E_a. We do this by computing the normalized correlation between the entirety of E_a against the entirety of E'_b for each integer shift in the range of correlations specified (-50 to 50 in our case), over a correlation range of *M* samples (M=1000 in our case).

Within scaling *s* for E_b, For each correlation *k*, we compute the inner product as follows:

$$C_s[k] = \frac{\sum_{i=1}^{M} E_a[i]E'_{b,s}[i+k]}{\left(\sum_{i=1}^{M} E_a[i]^2\right)^{1/2}\left(\sum_{i=1}^{M} E'_{b,s}[i+k]^2\right)^{1/2}}$$

We then choose the maximum score to represent the overall score for each timescale:

$$C[s] = \max_k C_s[k]$$

In the figure below, we show the correlation peaks across timescales for the signals from Figures 1 and 2 (Mozart's K. 331, movement one, and The Realm's "This Is Not a Breakdown.") In this instance, there are two strong peaks, at *s*=0.65 and 0.98, both corresponding to slowdowns of the dance song against the Mozart.

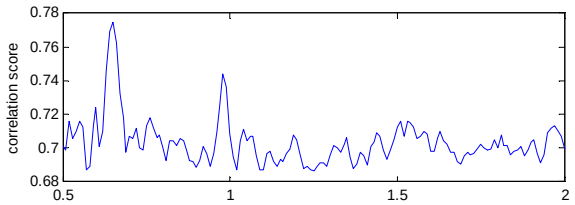

Figure 5: Correlation scores *C[s]* representing the best possible shift for each timescale *k* between 0.5 and 2.0.

The correlation length, *M*, is a critical choice, and represents how long the segments of the song pieces we will do the matching over. The results in this paper for the most part used a correlation length of 1000 frames, which corresponds to about 12 seconds. This is rather a long time, and can become a liability if the tempos of the component songs are changing rapidly, but because song *a* is not heavily beated, this longer window allows us to more confidently find a scaling of *b* against which it is best aligned. The effect of this can be seen by the sharpening of the peaks as *M* goes from 200 to 1000, as shown in the figure below. In this case, the songs are the third movement of Mozart's K. 331, and again The Realm's "Breakdown." Note that with a short window of 200 frames, there are no clear peaks, and in fact the strongest peak of the set is not yet visible. As *M* increases, we see the measure become increasingly confident of the peaks at about 1.2 and 0.6, which are in fact the best matches for this particular pair of signals.

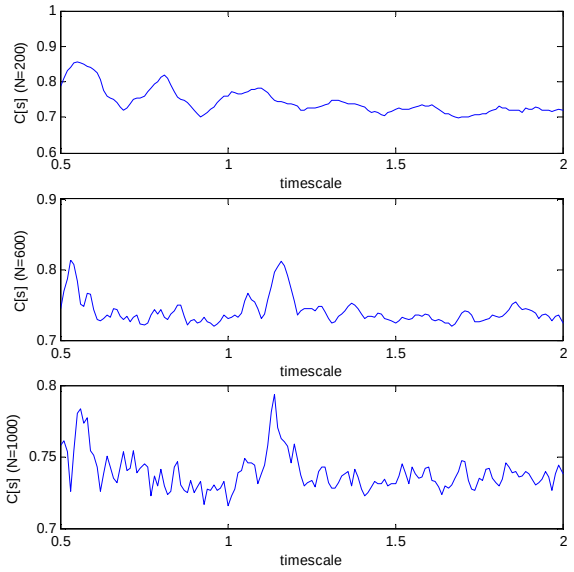

Figure 6: Correlation scores across timescales showing sharper peaks with increasing correlation length *M*.

2.3 Selecting the Best Alignment

We now have a set of possible alignments indexed by *s* along with the corresponding scores. For each scaling *s*, we find the peak locations by choosing all points that are greater than both their left and right neighbor. While this is a simplistic measure, it guarantees that we will cover all possible peaks while avoiding the redundancy resulting from just choosing the top *n* values. If we did the latter, the top values would all be neighbors of the highest peak and not correspond to unique peaks. We choose the top *n* scores from all peaks over all scalings *k*. In the figure below, we show the top 5 identified peak locations:

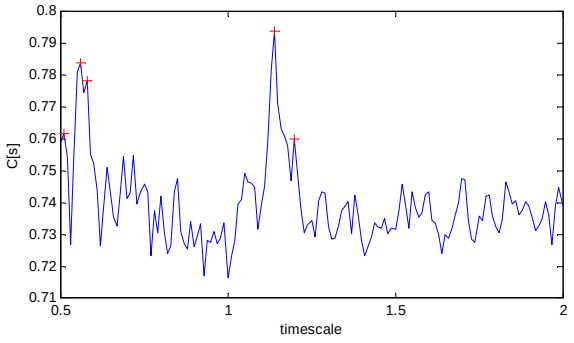

Figure 7: Top 5 peaks identified in *C[s]* using our simple peak picking criteria. Peaks are marked with (+).

In our implementation, we allowed the user to iterate through the top peak values to choose the best match. Note that while this lets us choose among the best matches, it does not tell us whether any of the matches in fact result in a strong mix. In section 3, we will describe a method for evaluating the suitability of the set of matches.

2.4 Computing the Time-Scaled Signal *b'*

Once the candidate scalings/shifts have been determined, the signal *b* needs to scaled and shifted in the same way as E'_b to produce *b'*. There are a variety of ways to do this, and we have implemented two. The first is to do the same linear resampling described in step 2 above. The second is to use a pitch-preserving time-scaling algorithm such as SOLA (Roucos and Wilgas 1985). The former is equivalent to playing the sound faster or slower, resulting in both length and pitch changes but a greater preservation of signal quality, whereas the latter maintains the pitch and changes only the length. We tested both methods, but in our final implementation chose the resampling approach due to its speed.

2.5 Combining the Signals

The signals *a* and *b'* are summed together with a scaling factor *r* for *b'*. We choose this scaling factor in a way to make the average energy of *a* and *b'* equal. The scaling factor for *b'* is thus

$$r = \frac{\sum_{k=1}^{M} E_a[k]}{\sum_{k=1}^{M} E'_b[k]}$$

This auto-scaling is quite effective for most samples, but it is important to allow for manual adjustments in order to get the most aesthetic mix of the two signals. Note that in DJing situations, it is typical for a user to modify this parameter dynamically, bringing the mixed-in sound in and out based on the musical context.

3 A Mixing Suitability Metric

Now that we have a variety of matches, we would like to evaluate how good each match is. There is of course the correlation value *C[s]* of each match, but this number is not very informative. It is more the relative shape of the peak with respect to the other matches – we have found that if the match values are fairly uniform across timescale (see figure below), none of the matches are likely to be particularly good. On the other hand, if there are clear, isolated peaks, the matches are significantly better. In the Figures below, we show the difference between bad and good match ranges for *C[s]*:

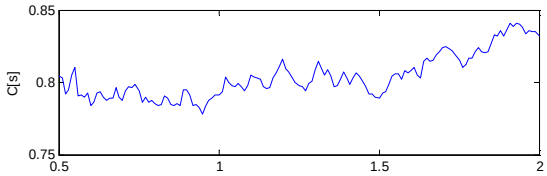

Figure 8: Correlation values *C[s]* for matching two melodic pieces (Mozart K. 331, 3 and Clay Aiken's "Invisible").

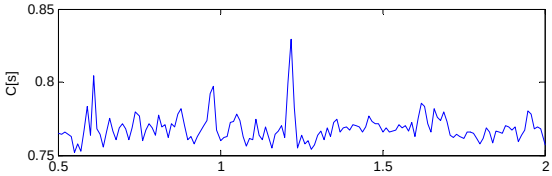

Figure 9: Correlation values *C[s]* for matching a rock song (Outkast's "Hey Ya") and a techno song (Jan Johnston's "Superstar").

From looking at Figures 8 and 9, we can easily see that while the correlation scores for both examples are in the same range, the latter plot has a clear set of peaks while the former has no values that really stand out. Of course, it is probably not practical to show this plot to the DJ; instead, we would like to encapsulate this property of the matches in a single number that can represent that suitability of each match. To do this, we simply take the value of the peak normalized by the mean and variance of the match curve, but with one caveat: we remove the area corresponding to the peak of interest. We do this because we do not want the values from the peak itself to affect the variance; if it is indeed a true outlier, it can significantly affect the mean/variance.

To remove the peak context, we bracket the peak by the valleys to the left and right of the peak, where we define valleys in the same way that we defined peaks, i.e., points that are lower than both their left and right neighbors. In the figure below, we show a zoomed-in view of a peak and its surrounding valleys:

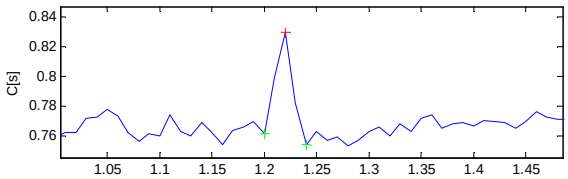

Figure 10: Peak (marked by red +) and surrounding valleys (marked by green +'s).

For a particular peak location at *s**, we compute the peak suitability *p* as follows:

$$p[s^*] = \frac{C[s] - \overline{C}}{\sum_{s \setminus context(s^*)} (C[s] - \overline{C})^2}$$

where $\overline{C}$ is the mean of *C[s]*, again excluding the context of the peak *k**. Using this measure, the suitability for the top peak in Figure 9 is 2.74, whereas for Figure 10 it is 7.88. In general, we found that peaks with suitability values greater than 3.0 tended to result in good matches, while the rest were of variable quality.

4 Choosing the Right Mix

In order to test our method on a wide variety of songs, we developed a simple exploratory interface that would allow a user to apply our methods to arbitrary music files. We show a screenshot in the figure below:

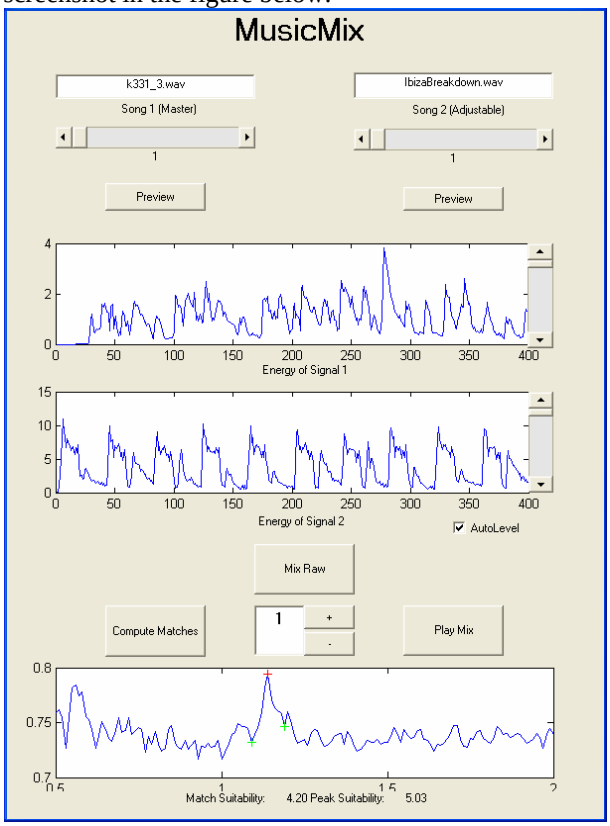

Figure 11: Interface for mixing song pieces. The user can load pieces of songs into the two selection boxes, see their respective energies, compute the best alignments (shown in the lower window), and then choose and play the mixes from the top peaks in the alignment scores.

The user can choose pieces from arbitrary songs, with the first song as the "master," which is preserved with its original timescale, and the second as the "slave," which will be modified and shifted to best fit the first song. The user can then see the energy waveforms of the signals, compute the matches, and then choose from the peaks of *C[s]* and play the resulting mixes. Because we can compute *C[s]* for 12 second segments in only 0.5 seconds, the user can play with different mixes in real time.

This brings us to the important question of which peak to choose, if any. The particular choice depends both on the suitability and on the context. If the suitability is low, it may be better not to mix at all. Even with a strong match, though, there will in general be several choices to pick from. The highest peak will tend to produce the best mix, but if the method is being applied to a DJing context, it is more important to choose a peak with a value of *s* close to 1, so as to require minimal distortion/stretching of the backing signal.

5 Results

To demonstrate the results of our method, we will both graphically show the different alignments chosen by the algorithm and present audio examples of the match results.

5.1 Across Genres

We will first show graphically the different alignments selected by our algorithm. Returning to the songs from Figures 1 and 2, Mozart's K. 331 movement 1 and The Realm's "This Is Not a Breakdown," we show the correlation scores C[s] across timescales for this set:

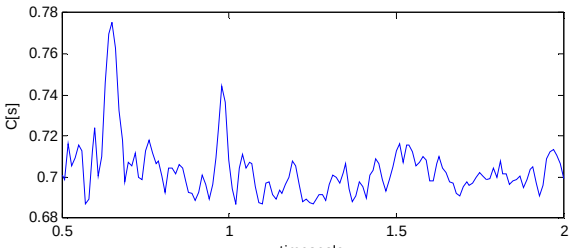

Figure 12: Correlation scores *C[s]* for The Realm's "Breakdown" scaled and shifted against Mozart's K. 331, movement 1.

The two strongest peaks, with suitabilities 7.81 and 3.07 respectively, are at timescalings 0.65 and 0.98. In the figures below, we show the energy waveforms of the scaled and shifted signal, *b'*, along with the energy for the untouched *a*, the master signal.

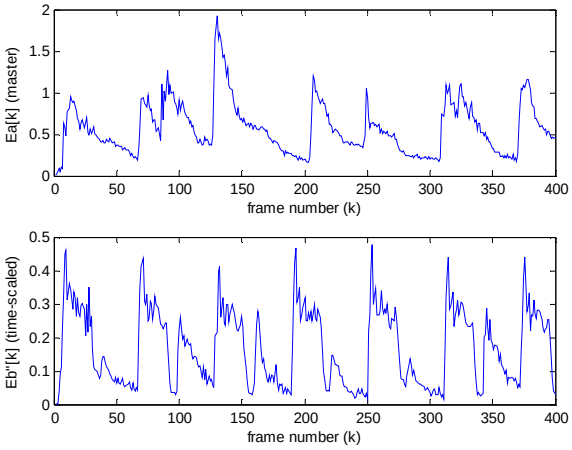

Figure 13: Energy waveforms *E[k]* for the original signal *a* and the timescaled/shifted *b'*, choosing the peak *s*=0.65 (K. 331 mixed with "Breakdown"). Note that the energy is now aligned (compare to Figure 1).

Notice how this best match has resulted in the alignment of the beats from "Breakdown" with the rises and falls in the Mozart piece (compare to the original pair in Figure 1).

While this match corresponds to a significant slowdown of "Breakdown," it is far stronger than the other match. Also, since "Breakdown" is heavily beated, the slowed version still sounds reasonable, and the strong alignment of the signals is pleasing to the ear.

In the next figure, we show the results of using the second match (s=0.98):

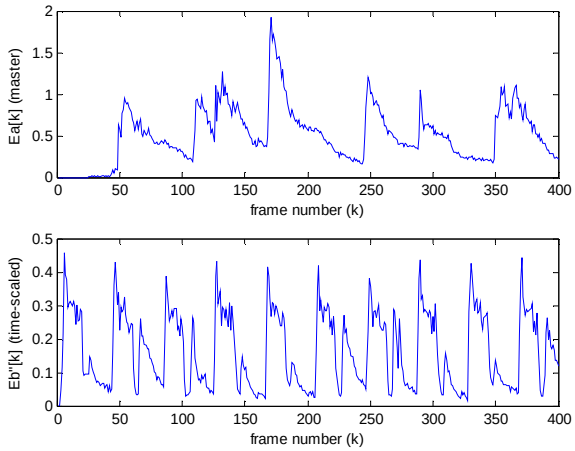

Figure 14: Energy waveforms *E[k]* for the original signal *a* and the timescaled/shifted *b'*, choosing the peak s=0.98 (K. 331 mixed with "Breakdown").

In this case, the techno piece is much closer to its original speed. While it is not as strong a match in terms of its suitability value, it still makes for a good mix.

In this next example, we use a different genre for the master signal – classical Indian music. While there are drums in this music, they are not at all regular, and the time signatures of such pieces can be very complicated indeed. In this case, we are using Ravi Shankar's "Dadra" as the master signal, and Transa's "Enervate" as the signal to mix in.

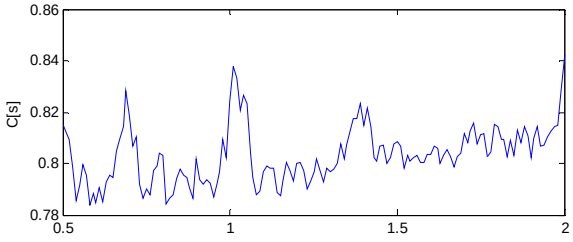

Figure 15: Correlation scores *C[s]* for Transa's "Enervate" scaled and shifted against Ravi Shankar's "Dadra."

Again we see some fairly strong peaks; here the top values are at s=1.01 and 0.69 with suitabilities of 3.4 and 2.33. Though the latter suitability is low, it is clear from how it stands out that it is still a significant peak, and in fact when played results in a good mix. This again shows that our suitability metric tends to be a sufficient but not necessary condition for a good mix; in the end, a visual inspection of the match peaks *C[s]* is still the best way to determine whether a peak is worth investigating.

We now look at the match from the stronger peak, s=3.4. This results in a great mix, and in the figure below it is possible to see how the beats have been timescaled and shifted to match the energy in the Shankar raga.

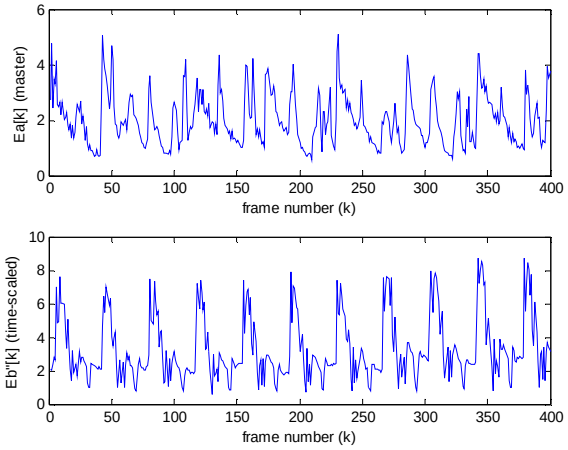

Figure 16: Energy waveforms *E[k]* for the original signal *a* and the timescaled/shifted *b'*, choosing the peak *s*=1.01 ("Dadra" mixed with "Enervate").

We have tried our method on a wide variety of other popular music and found that we can find strong techno/dance matches and good mixes for everything from the accordion solos in the Amelie soundtrack to the rap rhythms of Outkast. On the other hand, our attempts to mix different melodic songs did not prove very successful for two reasons. First, the songs were typically in different keys and had different lyrics, resulting in a cacophony of detuned voices when combined. Second, when neither song had a strong beat, there would be no matches with a high suitability – essentially, all choices of scales and shifts were equally bad.

5.2 Audio Examples

While we have computed and discussed a wide variety of examples with popular music, due to copyright restrictions, we cannot place these samples online. As a result, we have combined segments of our own music to produce a number of audio examples. These are available at the following site: http://www.research.microsoft.com/~sumitb/musicmixexamples.

5.3 Failure Modes

Along with the successes of the method, it is important to point out where it fails. The first and most obvious failure mode is when trying to combine two segments of non-beat oriented music as discussed above.

The second mode is more subtle and appears in the more typical case of mixing beats with a piece of melodic music.

Though much of the advantage and robustness of our algorithm is due to its avoidance of ever counting or even paying attention to the beat structure, in certain situations this is also its Achilles tendon.

In essence, our algorithm tries to find the best alignment of the energies of the two songs given all scalings and shiftings of the beat track. Songs without beats still often have significant energy alignment with respect to their time signature, and this is what allows our method to work so well. However, since we are making no attempt to examine time signature in either song, there are situations in which fitting three beats of the backing track to a measure in the other song is almost as good as fitting four, though perceptually there is a huge difference. In the following example, we show the results of matching The Realm's "This Is Not a Breakdown" to Beyonce's "Baby Boy:"

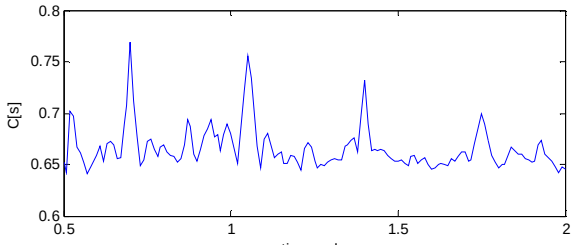

Figure 17: Correlation scores *C[s]* for The Realm's "Breakdown" scaled and shifted against Beyonce's "Baby Boy."

There are a number of very strong peaks in this match. The first peak results in an excellent mix, with two beats of the techno track to every beat of Baby Boy. The second peak, though, at *s*=1.05, is another story entirely:

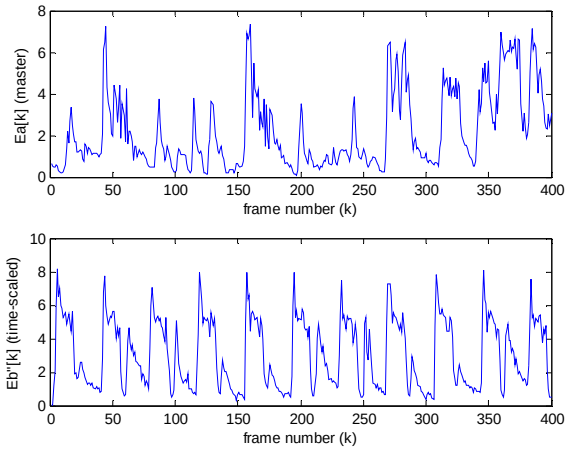

Figure 18: Energy waveforms *E[k]* for the original signal *a* and the timescaled/shifted *b'*, choosing the peak *s*=1.01 ("Baby Boy" mixed with "Breakdown").

Examining the alignment closely, we can see that there are *three* beats of the techno track to every major beat of the first signal – reasonable in terms of energy alignment, but terrible in terms of mix quality. Unfortunately, there is no easy fix to this problem – without examining the beat structure in some way, it is very difficult to determine this match will be perceptually bad. The ability of the DJ to preview the top matches is thus a critical element to applying this work.

6 Future Work

There are a variety of directions in which we wish to take this work. The first is to allow switching of the master and slave signals in the interface, so that a DJ can easily choose to stretch the melodic piece instead of the techno. The algorithm is of course agnostic as to which piece is which; this is merely an interface issue. Next, we would like to examine the use of this method to blend the ends of songs together as a DJ would do in a club – a simple variation on our existing procedure. Finally, we would like to incorporate some aspects of structure understanding to our algorithm in order to avoid the 3/4 mismatches described in the previous section.

7 Conclusions

We have presented an efficient method for robustly aligning beat-oriented music to a wide variety of music genres. We have shown how we can do this via energy alignment and thus without doing explicit beat-counting in either song, and how we can speed up the computation greatly with justifiable approximations. We hope that this method will be useful to expand the range of the DJ's craft, and allow her to now mix in pieces from arbitrary genres live without the need for manual, offline alignments.

8 References

Goto, M. and Y. Muraoka (1997). "Real-Time Rhythm Tracking for Drumless Audio Signals." *Proceedings of IJCAI-97 Workshop on Computational Auditory Scene Analysis*. pp. 135-144.

Laroche, J. (2001). "Estimating Tempo, Swing, and Beat Locations In Audio Recordings." *Proceedings of WASPAA 2001. pp. 135-138.*

Rabiner, L. and R. Schafer. (1978). *Digital Processing of Speech Signals.* Prentice Hall.

Roucos, S. and A. Wilgus (1985). "High Quality Time-Scale Modification of Speech." *Proceedings of ICASSP-85*, pp. 236-239.

Scheirer, E. (1998). "Tempo and Beat Analysis of Acoustic Musical Signals." *Journal Acoust. Soc. Am.* 103(1), 588-601.