TEMPO EXTRACTION VIA THE PERIODICITY TRANSFORM

William A. Sethares
Dept. Electrical Engineering
1415 Engineering Drive
Universoty of Wisconsin, Madison WI
sethares@ece.wisc.edu

ABSTRACT

This short paper briefly describes an algorithm submitted for the MIREX 2005 tempo extraction contest. Key portions of the method are the feature vectors, a new method of resampling (a step required by the periodicity transform), and a modification to the periodicity transform that allows it to consider only periodicities within some prespecified range. The strengths and weaknesses of the algorithm are highlighted in view of the results of the contest.

Keywords: MIREX, tempo extraction contest, periodicity transform, resampling

1 INTRODUCTION

Our previous work has focused on two aspects of the rhythm-finding problem: beat tracking (3) and the location of various levels of the metrical structure via the Periodicity Transform (PT) (5), both from audio sound sources. Since the problem of tempo finding overlaps both these problems, we have merged parts of our previous algorithms and included modifications to try and match the supplied tempo training data. The result is our entry for the tempo extraction contest.

2 THE ALGORITHM

The algorithm consists of the following steps:

1. Calculate the feature vectors
2. Preliminary analysis to determine proper subsampling rate
3. Resample feature vectors at appropriate rate
4. Apply (modified) periodicity transform to each (resampled) feature vector
5. Pick two largest peaks (these define the two candidate tempos)
6. Find the basis elements corresponding to each of the two tempos. The energy in these basis elements is used to calculate the relative balance between the two tempos. The time at which each basis element achieves its largest value is used to estimate the "phase."

Each of the steps is now discussed more fully.

Step 1: Our early work on finding periodicities (5) used a spectrogram-like set of feature vectors, each representing the energy in a given 1/3-octave frequency band. Since then we have evaluated a large number of different feature vectors for the beat tracking problem (in (3) and (6)) and have selected 14 features that appear to work well in the beat tracking problem. In preparing the algorithm for the contest, we ran the algorithm using both the 1/3-octave frames and the new feature vectors. While the new feature vectors seem to do a good job distilling the data into a single lattice (i.e., a single beat/tempo), they also appear to destroy the multiple levels of hierarchy that are desired when trying to identify two salient tempos as is required by the contest. Accordingly, the submission used the simpler 1/3-octave sub-bands for the tempo extraction algorithm. The frame width was 2^{11} and the initial hop (number of samples between successive frames) was 120 samples. Thus the initial feature vectors were (subsampled) at a rate of $\frac{44100}{120} \approx 367.5$ Hz.

Step 2: One of the quirks of the PT is that it finds only integer-periodic inputs (4). For example, if an input is periodic every 525 ms but the sampling is done only once every 50 ms, then the true periodicity is 10.5 samples. The PT find this periodicity at 21 samples since it does not have a mechanism for reporting fractional periods. To have the true periodicity represented, it is necessary to resample the data. For example, if the data were sampled at a 75 ms rate, then the same periodicity would represented by 7 samples. If resampling at 35 ms, the resampled period would be 15 samples.

Thus it is important to pick a (resampling) rate that will allow the PT to find the desired periodicities (and not their sampling-induced multiples). One way to accomplish this is to take the FFT of part of the data and to find a promiment frequency near the nominal 367.5 Hz. The hop value (which need not be an integer) that will cause this frequency to contain an integer number of samples can then be calculated, and this value is used in step 3.

It is important that the estimated frequency be as accurate as possible. Simply picking the peak of a single FFT has a frequency resolution that is limited to the

$\frac{\text{sampling rate}}{\text{number of samples}}$ Hz. By taking successive FFTs that are offset in time, it is possible to increase this accuracy considerably. For the present estimation, we chose to offset the two FFTs by 500 samples, and to locate the largest peak that appeared in the magnitude of both FFTs that was near the desired value. The phase difference between the successive FFTs is then used as a correction to the frequency estimates, resulting in a considerably more accurate estimation. This strategy is similar to the analysis portion of a phase vocoder.

Step 3: The feature vectors (in this case, the 1/3-octave sub-bands) are recalculated using the new hop value to achieve the desired subsampling rate.

Step 4: One of the things the PT is good at is finding very long-period features of the sound (5). For larger metrical patterns, this is very good. For finding tempos, it is not desirable. Accordingly, we modified the operation of the PT to choose only perodicities within some prespecified range (in this case, the range was the possible range of largest and smallest periodicities found in the training data, plus a small safety factor). In an orthogonal transform like the FFT or the DWT, this would be trivial (truncate the values from the full transform). Because the PT is nonorthogonal, this required some reworking of the operation of the PT. (To be specific, we began with the M-best algorithm (with gamma modification) and disallowed the consideration of periodicities larger or smaller than the prespecified values).

Step 5: The two largest peaks in the output of the PT correspond to the two strongest periodicities in the input. We took all the reported periodicities (up to five from each feature vector) wieghted them by the energy of the corresponding basis functions, and then picked the largest two which were present in the most feature vectors.

Step 6: To calculate the relative strength of the two tempos, let e_i be the energy in the ith basis function. Then the ratio

$$r = \frac{e_1}{e_1 + e_2}$$

gives the relative strengths. The phase of each tempo was calculated by finding the time at which the largest absolute value occurs in the basis functions corresponding to the two tempos.

3 RESULTS AND DISCUSSION

The results of the competition are, overall, encouraging for the emerging field of tempo extraction. The discussion is organized to follow the output of the contest results, which can be found at (1).

3.1 SCORE

The raw scores for the contest ranged from a high of 0.69 to a low of 0.54. The PT method was 0.597 with std of 0.252. Since the standard deviation of the values was larger than 0.23 for all entries, all algorithms are well within a single standard deviation of each other. One obvious reason for this is that the data set is not overly large, which tends to make fine discriminations difficult. But it may also be that the scoring method contributed to this. For example, none of the algorithms was particularly good at locating phases (see below). It is possible that by "rescoring" the competition to remove those parts where the algorithms appear to do little better than chance, it might tighten up the standard deviation so that some significance could be found bewteen the "best" and the "worst" of the algorithms.

3.2 AT LEAST ONE TEMPO CORRECT

The best of the algorithms (the entry by Alonso) correctly identified at least one of the tempos correctly in more than 95% of the cases. Thus, of the 140 pieces, the tempo was correctly found in 133. The PT method achieved 90%, identifying one of the tempos correctly in 123 of the 140 pieces. Eleven of the thirteen entrants correctly identified one tempo in 120 (or more) pieces.

This raises some interesting issues that could be answered by access to more complete reporting of the results of the contest. For example, Alonso's algorithm missed the tempo in 7 of the pieces (others in somewhat more). Did all the algorithms fail on these same 7 pieces, or did some of the algorithms succeed where even the winner failed? Suppose, for example, that the answer is "no," that some algorithm(s) succeeded on each piece. Then there is the chance that by combining the strengths of the current methods tempo extraction can achieve near 100% correct tempo identification. But if the answer is "yes" this means that none of the current methods succeed on these pieces. It would then be up to the community to try and figure out what the pieces have in common (if anything) and to either forbid these pathological pieces from consideration or to devise newer and better methods. In any case, this information ought to be available to the community so that we can understand how well we are really doing.

3.3 BOTH TEMPOS CORRECT

The best of the algorithms correctly identified both tempos correctly in almost 60% of the pieces. The PT succeeded in less than 40%, and this is the main reason that the algorithm did not have a higher overall score.

This is likely caused by the modifications made to the PT in algorithm step 4. As stated earlier, the PT tends to do a good job at finding very long periodicities. When I first began looking at the training data, it would almost always report one of the two tempos, and then another major periodicitiy at (either) three, four, six or eight times that. While this may be desirable from the point of view of identifying hierarchical structure in a piece of music, it is undesirable from the point of view of replicating people's tapping behavior, where the two rates are never separated by more than a factor of three.

In order to try and improve the match, I modified the PT so that periodicities deemed too long or too short (as derived from the training data) were forbidden. While this seemed like a good idea at the time, it can have a distorting effect: several small forbidden periodicities may gather together to create a large (spurious) longer periodicity. Conversely, several forbidden large periodicities may be

reflected down to an allowed (but spurious) smaller one. I suspect that it is these distortions that cause the relatively poor performance of the algorithm in locating the second tempo.

There are other ways to proceed. For example, the PT could be run without modification to find the "best tempo" in the allowable region. The corresponding basis element could then be subtracted out (hence removing that periodicity alone) and then the resulting signal could be parsed again. (This is analogous to the small-to-large PT.) I suspect that the performance could be improved by handling this issue better.

3.4 CORRECT PHASES

The best of the phase-finding algorithms was the fourth entry of Gouyon and Dixon, which matched one of the phases in almost half the cases and matched both phases in slightly more than 10% of the cases. The average number of correct phases over all the algorithms was about 30% and Alonso's winning entry acheived only 25%. The PT was in the middle with 30%.

Somewhat more surprising is the result for both phases correct. Considering that the reported value needed to be within only 15% of the beat period, many of the phase estimates are worse than chance. I am not sure whether this is statistically significant or not, but it may indicate that the majority of the algorithms are very poor phase-finding methods, or it may indicate that the data was not gathered or reported consistently.

3.5 RUNNING TIME

Using a 1.2 GHz powerbook with the algorithm coded in Matlab, it takes between 10 and 20 minutes per training file. The average time is 15 minutes, and the variation occurs because the resampling procedure is data driven (the feature vectors that are the output of the resampling step can be longer or shorter than the original feature vectors). The bulk of the computation occurs in the periodicity transform itself, and so longer (resampled) feature vectors require more computation. This algorithm requires far more time for its operation that any of the other entries in the competition. There are three reasons for this. First, the PT is inherently time consuming (computation is on the order of n^3 flops where n is the length of the data to be transformed.) Second, the code for the PT is written as a Matlab `.m` file and hence is not compiled or optimized (as are the FFTs and autocorrelations used by most of the entrants). Third, the calling code was written hurredly in the time between the arrival of the training data and the start of the competition by an amateur coder (i.e., by me).

3.6 McNEMAR'S TEST

There are two ways in which the percentages in this table may be useful: when they are very large and when they are very small. Large values indicate that the algorithms are alike in some way, and this is easy to believe in the case of Gouyon and Dixon (0) being likened to Gouyon and Dixon (3) at a 50% level. It is also reasonable that Brossier and Tzanetakis might be similar (again near 50%). It is somewhat more surprising that Gouyon and Dixon (0) and (3) should also be similar to both Brossier and Tzanetakis. In a sense, these algorithms have some commonality that is not completely obvious and it would be interesting to see what the authors think might be causing this.

At the other end, the PT is judged the "most different" from the others, never rising above 3% and with only two values above 1%. In contrast, all the other algorithms have an above 30% value somewhere in the table.

I can think of (at least) two possible reasons for this. First is the use of the PT itself, which is quite different from the FFTs and autocorrelations used by others. The second is the subsampling procedure. While this is required by the PT, it is also a unique processing step and could possibly be adopted by other methods. At this point it is unclear how much impact this would have, but it may be that some significant portion of the positive results of the PT method arise not from the PT itself but from the judicious use of the resampling method of step 3.

4 ACKNOWLEDGEMENTS

I would like to thanks the contest organizers for an excellent job, the participants for lively discussions and for all the work, and I wish to congratulate Miguel Alonso for his first place showing!

References

[1] http://www.music-ir.org/evaluation/mirex-results/audio-tempo/index.html

[2] W. A. Sethares and R. D. Morris., "Performance measures for beat tracking," Int. Workshop on Bayesian Data Analysis, Santa Cruz, Aug. 2003.

[3] W. A. Sethares, R. D. Morris and J. C. Sethares, "Beat tracking of audio signals," *IEEE Trans. on Speech and Audio Processing*, Vol. 13, No. 2, March, 2005.

[4] W. A. Sethares and T. Staley, "The periodicity transform," *IEEE Trans. Signal Processing*, Vol. 47, No. 11, Nov. 1999.

[5] W. A. Sethares and T. Staley, "Meter and periodicity in musical performance," *J. New Music Research*, Vol. 30, No. 2, June 2001.

[6] W. A. Sethares and R. D. Morris, "Performance measures for beat tracking," in preparation.