applied sciences

MDPI

Article

A Novel Tempogram Generating Algorithm Based on Matching Pursuit

Wenming Gui [1,*], Yao Sun [1,*], Yuting Tao [1], Yanping Li [2], Lun Meng [3] and Jinglan Zhang [4]

1 School of Software Engineering, Jinling Institute of Technology, Nanjing 211169, China; tao_yuting@jit.edu.cn
2 Nanjing University of Posts and Telecommunications, Nanjing 210003, China; liyp@njupt.edu.cn
3 College of Public Administration, Hohai University, Nanjing 210098, China; m_l_01@163.com
4 Science and Engineering Faculty, Queensland University of Technology, Queensland 4001, Australia; jinglan.zhang@qut.edu.au
* Correspondence: gwm@jit.edu.cn (W.G.); suny216@jit.edu.cn (Y.S.); Tel.: +86-25-8618-8709 (W.G.)

Received: 14 February 2018; Accepted: 3 April 2018; Published: 4 April 2018

Featured Application: The proposed tempogram based on matching pursuit may benefit many kinds of applications in the field of music information retrieval. On the one hand, it can be used directly for tempo estimation. On the other hand, it may be helpful to beat tracking, structure analysis, rhythm identification and music classification.

Abstract: Tempogram is one of the most useful representations for tempo, which has many applications, such as music tempo estimation, music structure analysis, music classification, and beat tracking. This paper presents a novel tempogram generating algorithm, which is based on matching pursuit. First, a tempo dictionary is designed in the light of the characteristics of tempo and note onset, then matching pursuit based on the tempo dictionary is executed on the resampled novelty curve, and finally the tempogram is created by assembling the coefficients of matching pursuit. The tempogram created by this algorithm has better resolution, stronger sparsity, and flexibility than those of the traditional algorithms. We demonstrate the properties of the algorithm through experiments and provide an application example for tempo estimation.

Keywords: tempo; tempogram; novelty curve; autocorrelation; Fourier transform; matching pursuit

1. Introduction

In musical terminology, tempo is the speed or pace of a given piece [1]. It is usually measured by beats per minute (bpm). For example, a tempo of 60 bpm signifies one beat per second, while a tempo of 120 bpm is twice as rapid. The note value of a beat will typically be indicated by the meter signature. For instance, in 4/4 the beat will be a crotchet or quarter note. Tempo is one of the most important features of music, which is closely relevant to beat and rhythm. Correspondingly, tempo estimation is one of the most important research fields in music information retrieval (MIR), and moreover, it is the fundamental work for many other applications, such as beat tracking [2–4], music structure analysis [5], rhythm identification [6], and music classification [7].

Tempo estimation is defined as extracting tempo information from musical signals, which are the physical level waveforms and usually exist in the audio file, such as wav and MP3. Tempo estimation is a challenging task, and some special musical elements, such as legato and rest, make it even harder. Tempo always changes when music proceeds. There are two types of change: one is originally designed by the composer of the music, and the other is due to the playing or singing error. The former type of change usually takes place only few times, one or two, and even no one under many circumstances.

However, the latter is inevitable and exists in all parts of the music. Therefore, we should estimate the tempi at any time. We regard all of the tempi components at one time as a vector, called tempo vector. The tempo vectors at all of the proceeding time of the music make up the tempogram.

Following the definition in [8], a tempogram in this paper is a two-dimensional time-tempo representation for a given time-dependent signal, where the time parameter is measured in seconds and the tempo parameter measured in bpm. We could represent the tempo of music using tempogram [9], and then conduct tempo estimation through Dynamic Programming [10], Viterbi [11,12], or other methods [13]. Moreover, with the help of tempogram, other music analysis tasks, such as beat tracking, could also be easier to complete.

In this work, we proposed a novel tempogram generating algorithm that is based on Matching Pursuit (MP) and designed the tempo dictionary to implement the algorithm. This new algorithm outperforms the traditional algorithms with higher resolution, stronger sparsity and flexibility.

The remainder of this paper is organized as follows. Section 2 introduces related work. Section 3 presents the novel tempogram generating algorithm. Section 4 discusses the properties of this novel algorithm. Section 5 sketches an example application. Finally, Section 6 concludes the paper.

2. Related Work

Generally, there are two steps to generate the tempogram. The first step concerns onset detection, in which the musical signal is processed and the output is the novelty curve [8], which is the frame-to-frame difference and roughly describes the temporal context of onsets. To obtain the novelty curve, the musical signal is preprocessed firstly by separating the signal into multiple frequency bands, transient/steady state separation, etc. [14], and then predefined signal features, such as temporal features and spectral features [15–17], are extracted. Probabilistic methods [18–20] can also be employed to generate the novelty curve.

In the second step, periodicities or tempi are derived from the novelty curve, and the tempogram is created based on tempi at all of the musical proceeding time finally. In this paper, we focus on the second step, generating the tempogram, in which extracting periodicity from the novelty curve is the key procedure essentially. There are two main methods in this step [8], autocorrelation function (ACF) and Fourier transform (FT). Besides these two methods, Daniels reviewed some other methods that could implicitly be used to generate tempogram [21]. The approach of comb filtering [22,23] was proposed to estimate periodicity by Scheirer and Klapuri, where the output energy of the filterbank of comb filter indicated resonance at a tempo period. An enhanced ACF approach, called beat histogram [24], was provided by Tzanetakis to use for genre classification, and to estimate tempo in his later work [25]. Also, trying to improve upon traditional ACF, the autocorrelation phase matrix (APM) was introduced by Eck [26], which was aiming at adding beat phase information to the traditional ACF. Unlike the ACF of a novelty curve, Foote [27] proposed the beat spectrum approach to compute ACF over either the frames of rows or columns directly, in which melodic and harmonic patterns were declared to be retained. In addition, an enhanced FT approach was introduced by Rudrich [28], where the phase information of STFT was added, aiming at beat alignment. It is worthwhile to note that there were some approaches whose ideas of matching template is similar to this paper. Laroche [29] introduced the approach of matching the energy flux function, which is a type of novelty curve with a set of metrical templates to derive tempo information, in which he constructed the metrical templates on the assumption of the duple meter. Oliveira [30] used a similar template matching approach, but with no metrical information, to determine the initial beat phase hypotheses. Peeters [31] also proposed the beat template matching method in their beat and downbeat tracking system, however, they learned templates from data with annotated beat times using linear discriminant analysis (LDA), rather than hard-coded templates. Eronen [32] generated templates by the generalized autocorrelation function (GACF), and then employed k-Nearest Neighbor (k-NN) regression to find the matches to periodicity vectors with annotated tempi.

In this paper, we prominently introduce the two main approaches that were explicitly proposed for tempogram generation and also practically used for many research fields of MIR, and we use them for comparison in experiments as well. The ideas of the other approaches that are mentioned above can implicitly be adopted to tempogram generation, but they all were not designed for tempogram explicitly, most of which were used only for beat tracking, and needed to be modified or expanded if being used for tempogram generation.

2.1. Autocorrelation Function

The method of ACF executes the autocorrelation function on the windowed novelty curve firstly, and then derives the periodicities from the lags of the autocorrelation function. Finally, tempi are transformed from the lags to generate the tempogram. The expression of the ACF [8] is:

$$A(t,l) = \sum_{n \in Z} o(n)o(n+l)W(n-t)/(2N+1-l) \tag{1}$$

where t, n is the discrete time, and $l = 1 \ldots N$ is the lag, and $o(n)$ is the novelty curve, and $W(n)$ is a box window, centered at $t = 0$ with support $[-N, N]$. Let f_s denote the sample rate of $o(n)$, then the lag l corresponds to the period l/f_s, and the corresponding frequency is f_s/l, and the corresponding tempo $\tau = 60 \cdot f_s/l$.

2.2. Fourier Transform

The method of FT computes Fourier coefficients in the frequency domain on the windowed novelty curve firstly, and then transform the frequency to the tempo measure. FT is defined as [8]:

$$F(t,\omega) = \sum_{n \in Z} o(n)W(n-t)e^{-2\pi i \omega n} \tag{2}$$

where t, n, $o(n)$ are the same as ACF, and ω is the frequency, and $W(n)$ is a Hann-window centered at $t = 0$. The tempo τ is transformed from ω through the expression $\tau = 60 \cdot \omega$. There are two methods to discretize the parameter ω. The first one follows Discrete Fourier Transform (DFT) [11], in which ω is discretized as $k \cdot f_s/N$, $k = 0 \ldots N - l$. The second chooses the frequently used tempi $\tau \in [30, 480]$, $\tau \in \mathbb{Z}$ bpm to map to ω, where $\omega = \tau/60$ Hz [8,33], and then computes the coefficients following the upper equation.

3. Tempogram Based on Matching Pursuit

In this section, we will describe our tempogram generating algorithm based on Matching Pursuit (MP). Firstly, we will explain why MP is chosen for tempogram, and then we will describe how to generate tempogram using MP, including creating the tempo dictionary and fully implementing the algorithm.

3.1. Motivation

A tempogram is generated by extracting periodicity from the novelty curve. ACF can be used to obtain the periodicity from the similarity between the novelty curve itself and the lagged curves. FT can also derive the frequency as the periodicity from the novelty curve by computing Fourier coefficients, in which the coefficients represent the similarity between the novelty curve and the orthogonal Fourier basis at certain frequencies. However, these two methods limit the quantity and the forms of the curves to be compared to some extent. ACF confines the compared curves to a series of lagged curves and FT confines them to a series of the orthogonal basis. As these two approaches confine the quantity and the forms of the compared curves, the matching degree are restricted, and the accuracy of the tempogram is affected correspondingly. We require that the compared curves should match well with the characteristics of the tempo, which can be represented by the beat pulse in the time domain.

To some extent, the matching idea was explicitly expressed in the papers [29], however, the quantity and the forms of the compared curves were still limited, because their templates were limited on the assumption in which the duple meter were too simple to be used to describe the realistic metrical templates. Eronen [32] tried to compensate for the limited template set through the periodicity vector resampling, but it seemed that he still cannot systematically solve the problem. It is far more important that the matching algorithm of all the papers [29–32], such as cross-correlation and k-NN regression, still need further improvement.

In this paper, we proposed the tempogram generating algorithm based on MP. The dictionary of MP makes it more flexible to construct the templates or atoms as the similar curves, so that the quantity and the forms of the compared curves can be enriched easily. We constructed a tempo dictionary and designed the atoms in the light of the characteristics of tempo and note onset to ensure higher degree of similarity. Moreover, the matching algorithm of MP is more effective and more systematic to find curves that are similar to the tempo curves. Furthermore, MP makes the tempogram sparser such that it can be applied more effectively. For the tempogram, the fewer the number of big tempo components, in other words, the sparser the coefficients are, the higher the certainty of the tempo will be, and the better the tempogram will be applied certainly. Under the condition of the ideally sparsest temporgram, there would be only one big coefficient attached to the tempo at a given time. It means that there is only one line in the tempogram if the tempo is supposed to be constant.

To implement the proposed algorithm, we first create the tempo dictionary that will be used for the MP algorithm, and then we process the musical signal through MP to obtain the tempogram.

3.2. Tempo Dictionary

The dictionary of MP generally is redundant, in which the flexibility of MP lies. The atoms in the dictionary need to be consistent with the characteristics of tempo and note onsets. We call this dictionary as tempo dictionary. In this algorithm, we choose the frequently used tempi and create the corresponding "mother atom" for every tempo firstly. Then, new atoms are created by shifting the "mother atom". The "mother atom" and the new atoms together make up a set of atoms that are corresponding to the tempo. Finally, our tempo dictionary is composed of all the sets of the tempi. Figure 1 shows the steps and the details are described, as follows.

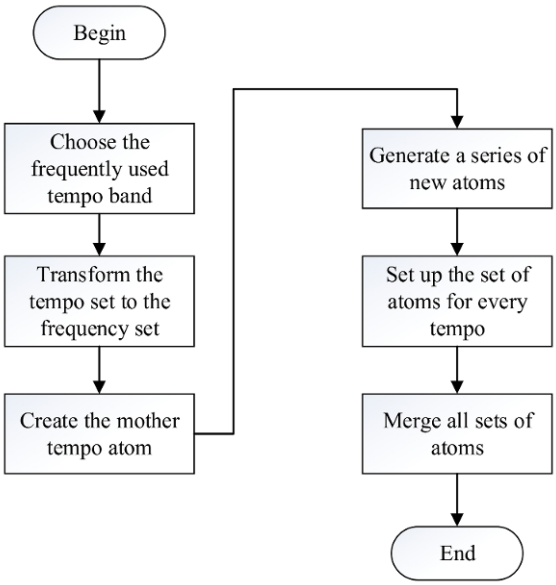

Figure 1. Flow chart for creating tempo dictionary.

3.2.1. Choose Tempo Band and Transform Tempo Set to Frequency Set

We choose the frequently used tempo band $\tau \in [30, 600]$, $\tau \in \mathbb{R}$, and transform the tempo set to the frequency set at a certain tempo resolution. Tempo resolution, in this paper, is defined as the distance between two valid adjacent tempo components in the tempogram. Tempo resolution $\Delta\tau$ is adjustable. We could set it to $\Delta\tau \in \mathbb{N}$, $\Delta\tau = 1, 2, \ldots$, and certainly could set it to $\Delta\tau \in \mathbb{R}$ with some other patterns, for example, a series of $\Delta\tau$ geometrically increasing or decreasing; We could set the same resolution in the whole tempo band, and also could set the different resolution in the different tempo band. As an example, we could set the resolution to 0.25 in the most frequently used band [80, 150] bpm, and set the resolution to 0.5 in the other bands. For comparison to ACF and FT, we set the resolution to 1 in the band $\tau \in [30, 600]$, $\tau \in \mathbb{Z}$, and transform the tempi to the frequencies following the expression: $f_b = \tau/60$, $\tau = [30, 31, \ldots 600]$, where $b = [1 \ldots B]$ are the indices of the tempi or the frequencies.

3.2.2. Create the Mother Tempo Atom for Every Tempo

For every tempo $\tau \in [30, 31, \ldots 600]$, or every corresponding frequency $f_b = \tau/60$, we create the "mother tempo atom" α_b that is based on cosine function with the length of M, which is equal to the frame length of the novelty curve $o(n)$, where $\alpha_b = \cos(2\pi f_b t)$, $t = (0 \ldots M-1)/f_o$ and f_o is the sample rate of $o(n)$. Because the onset is usually at sudden rise of the novelty curve, we only keep the positive part of the cosine function:

$$a_b = \begin{cases} \cos(2\pi f_b t) & \text{if } \cos(2\pi f_b t) \geq 0 \\ 0 & \text{if } \cos(2\pi f_b t) < 0 \end{cases} \tag{3}$$

Because the mother tempo atom is created in the light of both the characteristic of tempo and onset, the similarity between the atom and the novelty curve could be improved greatly.

3.2.3. Shift the Mother Tempo Atom to Generate a Series of New Atoms

Given a certain hop size $d (d \in \mathbb{N}^*)$, we shift right the mother tempo atom α_b for $d \cdot j$ $(j = 1, 2, 3 \ldots)$ steps along the axis of t to obtain a series of new atoms. After shifting to the right, the left part of the mother atom at the positions of $[0 \ldots M - d \cdot j - 1]$ is padded with the corresponding value of $\cos(-2\pi f_b t)$, $t = (M - d \cdot j \ldots 1)/f_o$. Because the mother atom is a periodic function, we limit the max step to the length of one period. Apparently, the limit will reduce the computation complexity of MP.

3.2.4. Set up the Set of Atoms for Every Tempo

For every tempo, we set up a set of atoms $\{d_b\}$, which includes the corresponding mother atom α_b and the series of the new atoms that are generated by shifting the mother atom.

3.2.5. Merge All the Sets of Atoms to Make up a Tempo Dictionary

Finally, the tempo dictionary is build up by merging the sets of atoms of all the tempi, $D = \{d_b, b = [1 \ldots B]\}$.

3.3. Algorithm Implementation

After tempo dictionary is created, we can implement the algorithm to generate the tempogram. At first, we extract the novelty curve from the musical signal. Then, we frame the novelty curve and decompose every frame by MP, using tempo dictionary to obtain the coefficients of the tempi and to generate tempo vectors. Finally, we obtain the tempogram through combining the vectors. The Figure 2 shows the implementation flow, and the detail is described, as follows.

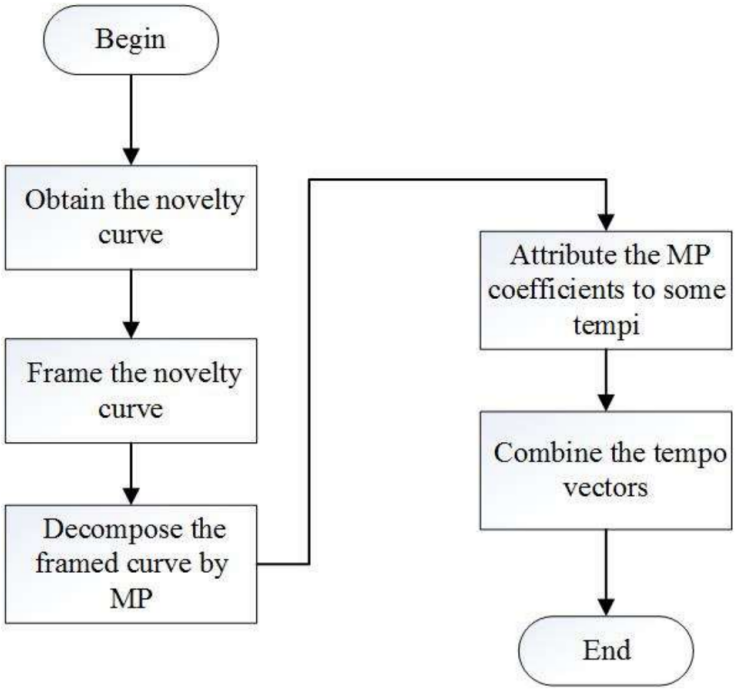

Figure 2. Flow chart for implementation.

3.3.1. Obtain the Novelty Curve from the Musical Signal

The input of our algorithm is an audio file, such as wav and mp3. We extract the musical signals and process them as the paper [8] did to derive the novelty curve $o(n)$.

3.3.2. Frame the Novelty Curve

In this step, we frame the novelty curve. We set the frame length to 6 s, and set the hop size to 0.2 s. Thus, we obtain a matrix $X = X(m, n)$, $m \in [1 \dots M]$, $n \in [1 \dots N]$, where M is the frame length, and N is the number of the frame.

3.3.3. Decompose the Framed Novelty Curve by MP Using Tempo Dictionary

For every frame X_i, $i \in [1 \dots N]$ of the novelty curve $o(n)$, we decompose it using the tempo dictionary D that was created in the previous section. We set the number of iteration K as the termination condition.

1. Initialize the residual signal and the iteration count: $n = 0$, $y_n = X_i$;
2. Compute the inner product between the residual signal y_n and each of the tempo atom $g_j \in D$: $< y_n, g_j >$;
3. Choose the maximum absolute value of the inner products $s_n = |< y_n, g_n >|$, where g_n is the corresponding atom, and save s_n and g_n as the n^{th} iteration result;
4. Compute the residual signal $y_{n+1} = y_n - |< y_n, g_n >| g_n$;
5. If $n < K$ then $n = n + 1$ and go back to step 2, else stop iteration.

The number K can be set according to the requirement of the tempogram, such as K = 10, 20 . . . . After iterations, we can get K MP coefficients s_n, $n = [1 \dots K]$ and K corresponding atoms g_n, $n = [1 \dots K]$.

3.3.4. Attribute the MP Coefficients to Some Tempi

For every MP coefficient s_n, $n = [1 \dots K]$ and its corresponding atom g_n, $n = [1 \dots K]$, we can find the corresponding mother atom from the tempo dictionary to get the tempo index $b (b \in [1 \dots B])$. We regard the coefficient s_n as the coefficient of the tempo τ_b. If there are multiple coefficients corresponding to τ_b, we simply sum up them. Finally, for a frame of the novelty curve, the coefficients of the tempi build up a tempo vector S.

3.3.5. Generate Tempogram

All of the coefficient vectors of the frames $S_n, n \in [1, N]$ are combined by column to generate a tempogram $S = S(b, n)$, $b = [1 \dots B]$, $n = [1 \dots N]$.

4. Discussion

Tempogram generated by our algorithm has better properties: more similar curves, higher resolution, sparser coefficients, and more flexibilities. We will discuss them in this section.

4.1. Tempo Resolution

As mentioned in the section of tempo dictionary, tempo resolution is defined as the distance between the two valid adjacent tempi in the tempogram. To some extent, tempo resolution is just like the frequency resolution, but not the same. The word "valid" here means that the tempo can be computed, but not estimated. Apparently, the bigger the distance is, the worst the resolution is. On the contrary, the smaller the distance is, the better the resolution is. In the tempogram image, the tempo resolution indicates the recognizability or the clarity of the tempo. Under the condition that the tempo resolution is the same, we can judge which is the better algorithm by the image. Generally, the tempo resolution should be smaller than 1. According to the annotation data of MIREX (Music Information Retrieval Evaluation eXchange), the tempo precision is 0.5.

When considering the tempo resolution of ACF, the distance of the adjacent tempi is $\Delta\tau = 60 \cdot f_s \cdot (\frac{1}{n} - \frac{1}{n+1}) = \frac{60 \cdot f_s}{n \cdot (n+1)}$, where n is the lag. It indicates that the resolution is getting worse with the lag decreasing. Following the paper [8], we let $f_s = 1/0.023 = 43.5$. Then, if $l = 51 (\tau = 51.2)$, we can get the resolution $\Delta\tau = 0.98$. But, if $l < 51$, the resolution will be $\Delta\tau > 1$. For example, if $l = 21 (\tau = 124.2)$, then the resolution will be $\Delta\tau = 5.6$. However, the tempi around 120 bpm are frequently used in all kinds of music, and evidently, the tempo resolution of ACF is not enough. If $l < 21$, then the resolution will be even worse. In conclusion, the resolution of ACF is not constant, and if $\tau > 51$ bpm $(l < 51)$, the resolution cannot satisfy with our requirement.

When considering the method of DFT [11], because the Hann-window length is 6 s, the tempo resolution is $\Delta\tau = 60 \cdot f_s / N = 60/6 = 10$ bpm. Apparently, it is not enough for the accuracy of the tempo. Inspecting the method of FT [8], it computed the frequency coefficients corresponding to the tempi by Discrete Time Fourier Transform (DTFT). It looks like it can calculate the coefficient at any frequency, but in fact, it only approximately samples the coefficients on the discretized frequency band. We can judge how the algorithm is by the tempogram image further.

To judge which algorithm produces the best resolution of the image, we chose a clip (train1.wav) that was downloaded from MIREX dataset [34] for demonstration. The musical signal and the corresponding novelty curve are shown in the Figure 3. The sample rate is 22,050 Hz and the ground truth tempo is roughly 129.5 bpm. All of the experiments below were performed on this clip.

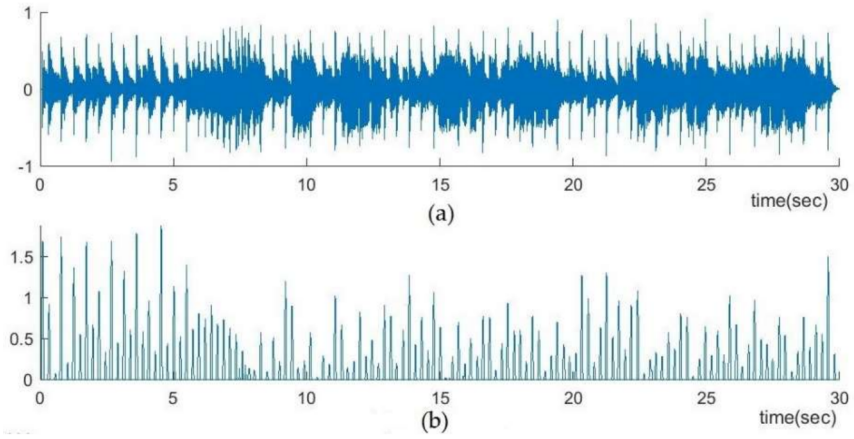

Figure 3. The musical signal (**a**) and the corresponding novelty curve (**b**) of the clip (train1.wav).

To compare the tempo resolution, we show three tempograms of the clip in the Figure 4, in which the tempogram of Figure 4a–c are created by the methods of ACF, FT, and our MP, respectively. For comparative purposes, all of the tempo resolutions are set to 1, that is to say, the coefficients must be computed or estimated at all of the integral tempi of the band $\tau \in [30, 600]$, $\tau \in \mathbb{N}$. Therefore, we can judge which algorithm is the best by the clarity of the images. For FT, we discretized the parameter ω following the papers [8,33], so that the coefficients of the integral tempi could be computed. For ACF, in order to generate 571 coefficients at the integral tempi, we must estimate. Especially for the tempo band $\tau > 51$, we must interpolate. For MP, the number of coefficients is determined by the number of iteration. Therefore, we set the number of iterations of MP to 571, so that the coefficients have the ability to cover all of the integral tempi. Significantly, there are many zero or near-zero coefficients among them. We set the tempo resolution to 1 when transforming the set of tempo to the set of frequency, and set the hop size to 2 when shifting the mother atoms. Consequently, we can see some broad, or narrow horizontal stripes in the Figure 4a,b, but Figure 4c presents the clear points or lines. The stripes make it difficult for us to decide which tempi should be chosen for the candidates. However, the points or lines tell us the truth definitely. Generally, the tempo precision is getting lower and lower with the stripe getting broader and broader. As can be seen, the corresponding stripes or lines in Figure 4c are apparently narrower than Figure 4a,b. It demonstrates that the tempogram of MP has higher tempo resolution than ACF and FT.

4.2. Similarity

Similarity is very important because we derive the periodicity from the similarity between the novelty curve and the compared curves. Our algorithm can provide more similar curves to compare. To explain this point, we calculated the Spearman correlation coefficients between all of the frames of the novelty curve and their corresponding most similar curves of ACF, FT, and MP respectively. For ACF, the most similar curve corresponding to the original frame of the novelty curve is the curve with the biggest autocorrelation coefficient. We can obtain the most similar curve according to the information of the corresponding lag. For FT, the most similar curve is the curve of the Fourier basis with the biggest Fourier coefficient. For MP, the most similar curve is the curve of the tempo atom with the biggest MP coefficient. Table 1 shows the statistic information of the Spearman correlation coefficients, in which the experiments were performed on the clip train1.wav, as well and the resolutions were set to 0.5 for all of the algorithms. The comparison of the means and variances indicates that the curves that were most similar to the original frames were from MP.

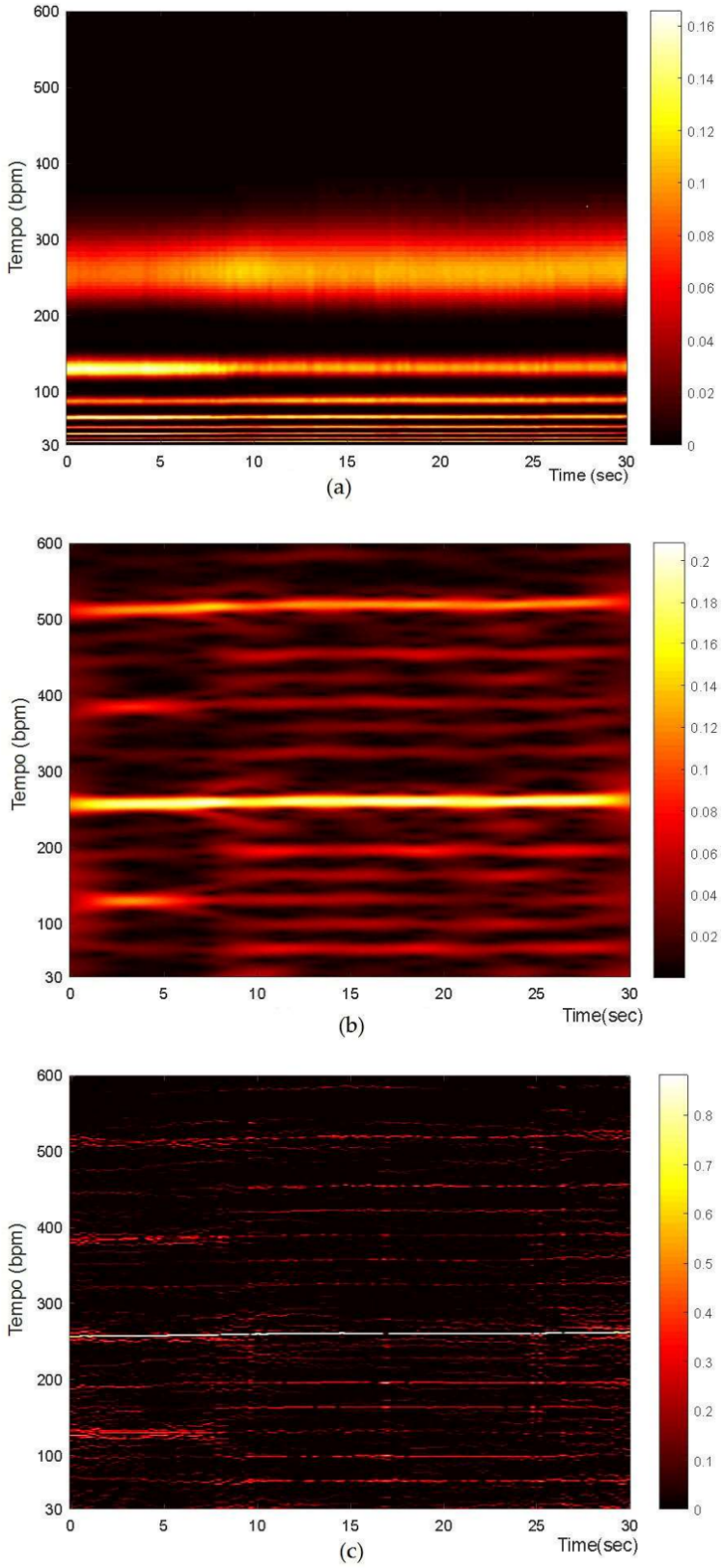

Figure 4. Comparison of tempograms: (**a**) autocorrelation function (ACF) tempogram; (**b**) Fourier transform (FT) tempogram; (**c**) Matching Pursuit (MP) tempogram. All of the tempograms are computed or interpolated to ensure the resolutions to be 1 for comparison.

Figure 5 shows example curves of ACF, FT, and MP most similar to the 20th frame of the novelty curve. For ACF, in theory, the most similar curve is itself, that is to say, $lag = 0$ or $lag\ near\ to\ 0$. However, it is beyond the tempo band we can perceive because the corresponding tempo to $lag = 1$ is 2610 bpm ($f_s = 1/0.023 = 43.5$ [8]). Figure 5b shows the most similar curve that we computed through ACF, in which the corresponding lag is $lag = 47$. Visually, some people may regard the curve from ACF as the most similar curve to the frame of the novelty curve, but actually, many peaks are not incongruous with the peaks of the frame. Figure 5c shows the most similar Fourier basis, where it is notable that the amplitudes of the basis include not only the positive part, but also the negative part. Figure 5d shows the most similar curve that was derived from MP, where the amplitudes only keep the positive part, which is more consistent with the novelty curve. The Spearman correlation coefficients between Figures 5a and 5b–d are 0.73, 0.34, and 0.83. It indicates that the most similar curve to Figure 5a is Figure 5d.

Table 1. Comparison of the Spearman correlation coefficients.

Algorithm	ACF	FT	MP
mean	0.55	−0.002	0.80
variance	0.11	0.54	0.08

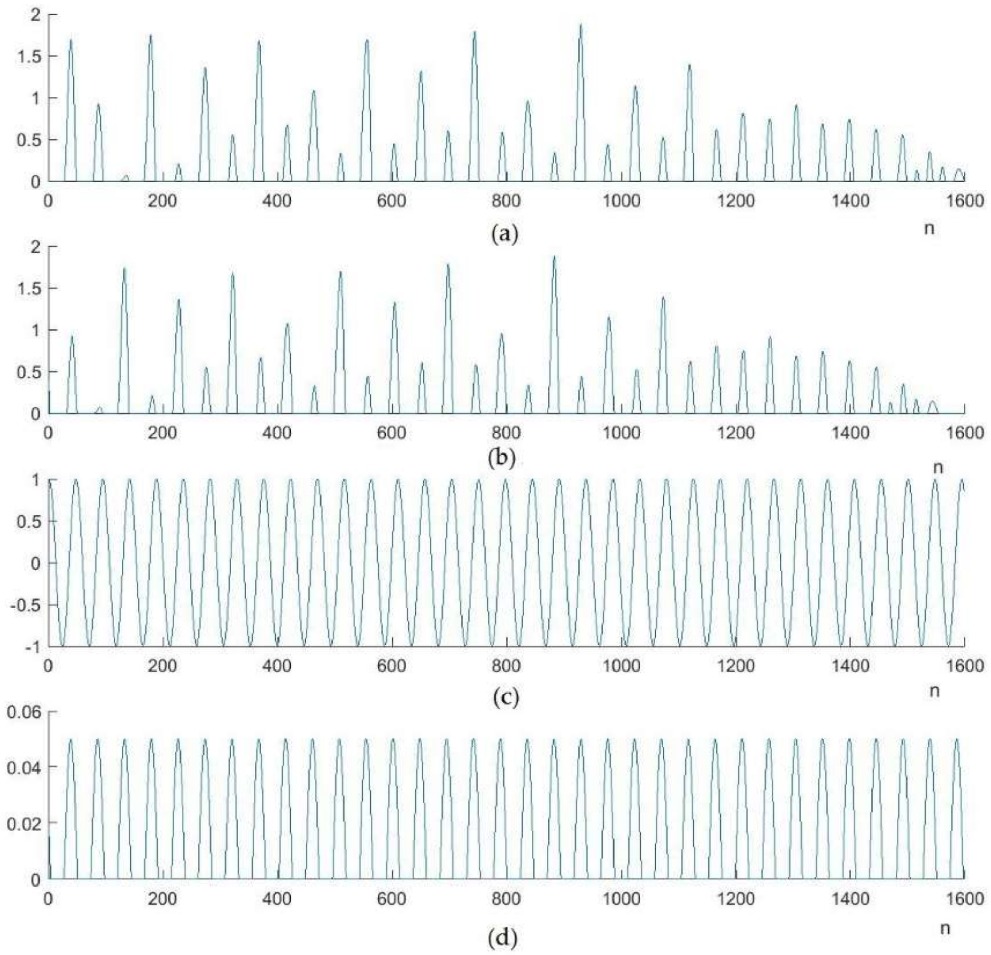

Figure 5. The most similar curves. (**a**) The 20th frame of the novelty curve of train1; (**b**) The most similar curve of ACF; (**c**) The most similar curve of FT; (**d**) The most similar curve of MP.

4.3. Sparsity

The sparsity of tempogram is defined as the number of the none-zero coefficients. The smaller the number is, the sparser the tempogram is. Meanwhile, the sparser tempogram indicates that the energy is better-concentrated, which could help us to distinguish the dominant tempo components. In other words, the sparser tempogram would lead to a more accurate estimation of tempo.

For one novelty curve, the tempo dictionary that was designed with rich atoms makes it easy to find a similar one to match with. Better similarity makes the energy more concentrated on the atom so that the tempogram could be sparser. Otherwise, the energy would scatter over many less similar atoms, which would produce many more none-zero coefficients.

From Figure 4, we can observe from the image that the sparsity of the proposed algorithm is stronger than those of the method of ACF and FT under the same condition. If we estimate the number of the none-zero coefficients as the degree of sparsity, we can get the same results. Firstly, we normalized the image data, and then defined the none-zero coefficient as the coefficient whose value is smaller than 0.1. Table 2 shows the number of the none-zero coefficient of those tempograms in Figure 4. The total number of coefficient of each tempogram is 83,937. It shows that the tempogram of MP is the sparest one.

Table 2. Comparison of the degree of sparsity.

Algorithm	ACF	FT	MP
The number of the none-zero coefficient	27,894	40,746	1914
The percent of the none-zero coefficient	33.23%	48.54%	2.28%

4.4. Flexibility

The proposed algorithm can provide tempogram with not only higher tempo resolution and stronger sparsity, but also better flexibility. It is possible for us to choose different tempo resolution and sparsity for our specific application under some circumstances. As an example, we do not always want highest tempo resolution. Sometimes we want to choose lower tempo resolution to obtain less complexity.

4.4.1. Flexibility of Tempo Resolution

Certainly, we need the algorithm to provide flexible tempo resolution. For instance, we should increase the tempo resolution when we try to estimate the detailed tempo. However, perhaps we may not care about the accuracy of tempo when we try to accomplish music classification task, so we could decrease the tempo resolution. For MP, it could be easy to alter tempo resolution of tempogram by adjusting the tempo resolution when we transform the frequently used tempo that is set to the frequency set at the first step of constructing the tempo dictionary.

Figure 6 shows three tempograms of MP with different tempo resolutions, in which the iteration numbers are set to 20 and the hop sizes are set to 2. The values of tempo resolutions of Figure 6a–c are 0.5, 1 and 2, respectively. As can be seen, the corresponding lines or stripes in the images are getting broader with the values of tempo resolution getting bigger (the resolutions getting lower). It is worthwhile to note that we need more tempo atoms to construct the MP dictionary if we need higher resolution, which will enlarge the size of the dictionary and bring a higher computational cost.

For FT, we could change tempo resolution by discretizing the frequency in different manner as well. But, it only works superficially because it only approximately samples the coefficients on the discretized frequency band and the tempo resolution cannot be changed substantially. For ACF, we cannot alter the tempo resolution at all.

4.4.2. Flexibility of Sparsity

Like tempo resolution, we may want to change sparsity to adapt to a specific application rather than always to keep highest sparsity. As we explained in the motivation Section 3.1, sparsity is mainly determined by the matching degree between the real tempo curve and the compared curve. In fact, sparsity lies in the quantity and quality of the compared curves. The quality cannot be changed for FT and ACF because their compared curves cannot be changed. For FT, though the quantity can be changed just as altering the tempo resolution, sparsity cannot be changed in fact, as we explained above.

However, for MP, we could design the atoms other than the tempo atoms that are provided in this paper, in other words, change the quality of the compared curves, so as to change the sparsity. Furthermore, we could be easy to change the quantity of the atoms. First, we could do just as altering the tempo resolution. Second, we could do that by changing the hop size in the step of shifting the mother atom when creating the tempo dictionary. The bigger the hop size is, the smaller the quantity of the atoms is, and the lower the sparsity is.

Figure 7 above shows three tempograms of MP with different sparsity that is caused by different hop sizes, in which the iteration numbers of MP are set to 20 and the values of tempo resolutions are set to 1. The hop sizes of Figure 7a–c are 2, 5, and 20 respectively. As can be seen, the corresponding none-zero coefficients around the same tempi are getting more and more with the hope size getting bigger and bigger. For example, the stripe around 250 bmp in Figure 7c is apparently broader than the corresponding regions in Figure 7a,b. It shows that the number of none-zero coefficients of Figure 7c is bigger than Figure 7a,b. In other words, it indicates that the sparsity of the tempogram is getting lower and lower with the hop size getting bigger and bigger.

Table 3 shows that the numbers and the percent of the none-zero coefficient of the tempograms in Figure 7. The number and the percent with the hop size of 20 is the biggest, while the smallest number and the percent come from the hop size of 2. It validates the above conclusion of Figure 7 again. It is also worthwhile to note that the computational cost will rise because the quantity of the atoms will grow if we decrease the hop size.

4.4.3. Flexibility of FFMTC

For many applications, under many circumstances, we may want to work out only the First Few Main Tempo Components (FFMTC). For example, in some simple tempo estimation application, we could extract the FFMTC as the target tempi directly. Evidently we cannot compute the FFMTC for ACF and FT until all of the coefficients are worked out. However, through MP, we could obtain the FFMTC easily, moreover, the number of the FFMTC can be customized conveniently by controlling the number of iteration because MP computes the biggest coefficients firstly, and then sequentially computes the second, and so on, which is the nature of MP.

Figure 8 shows the tempograms of MP with different FFMTC, in which the values of tempo resolutions are set to 1 and the hop sizes are set to 2. The numbers of iteration of Figure 8a–c are 2, 5, and 20, respectively. In the Figure 8a, the number of coefficients at a given time is 2 only. We may easily get the first two biggest tempo components through this tempogram. But, generally, if we want to extract the relatively accurate tempi, then we should compute the FFMTC with bigger iteration number to obtain more sample data. However, the number of iteration we need should be much smaller than the number of coefficients of ACF and FT at a given time because of the strong sparsity. As can be seen from Figure 8, the number of lines or stripes is apparently getting more and more with the number of iteration getting bigger and bigger. We should also note that the computational cost will grow with the increase of the number of iteration.

Table 3. Comparison of the sparsity of the tempograms with different hop size.

Algorithm	Hop Size = 2	Hop Size = 5	Hop Size = 20
The number of the none-zero coefficient	2003	2245	2409
The percent of the none-zero coefficient	2.39%	2.67%	2.87%

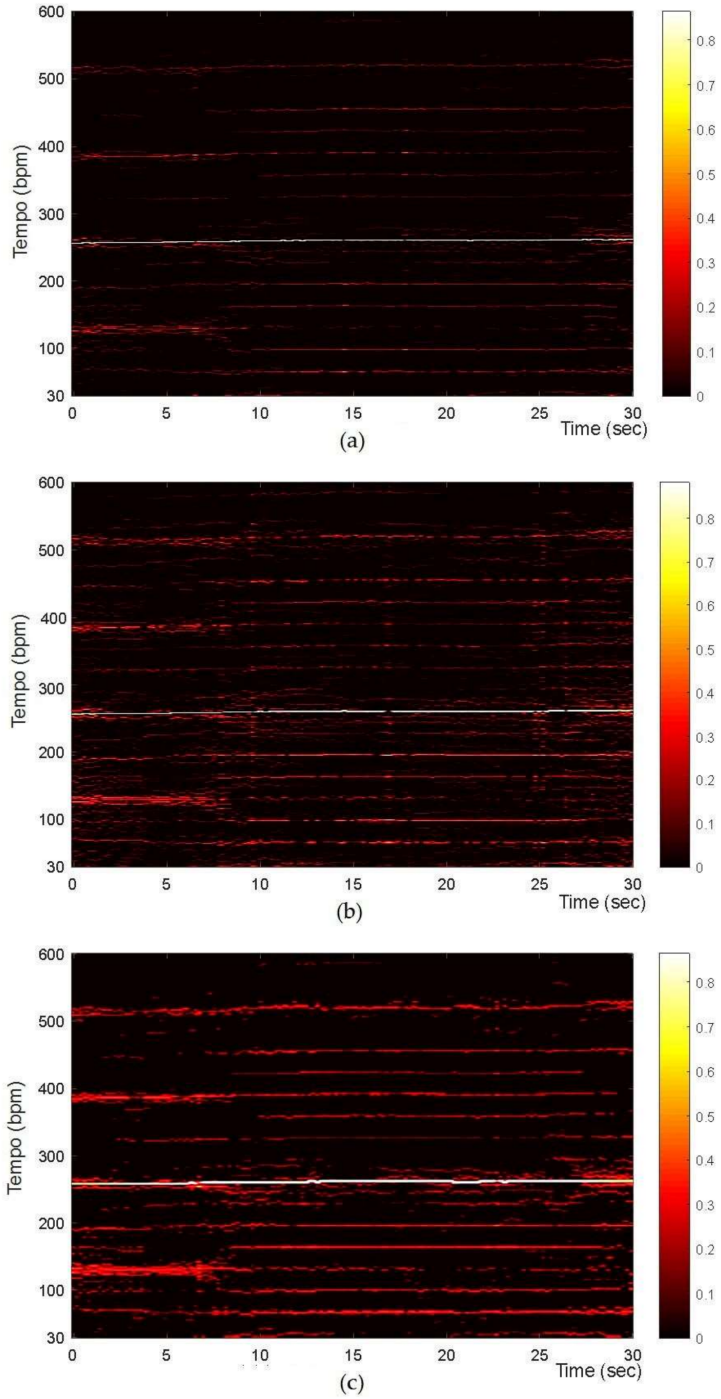

Figure 6. Flexibility of tempo resolution. (**a**) Tempo resolution = 0.5; (**b**) tempo resolution = 1; and, (**c**) tempo resolution = 2.

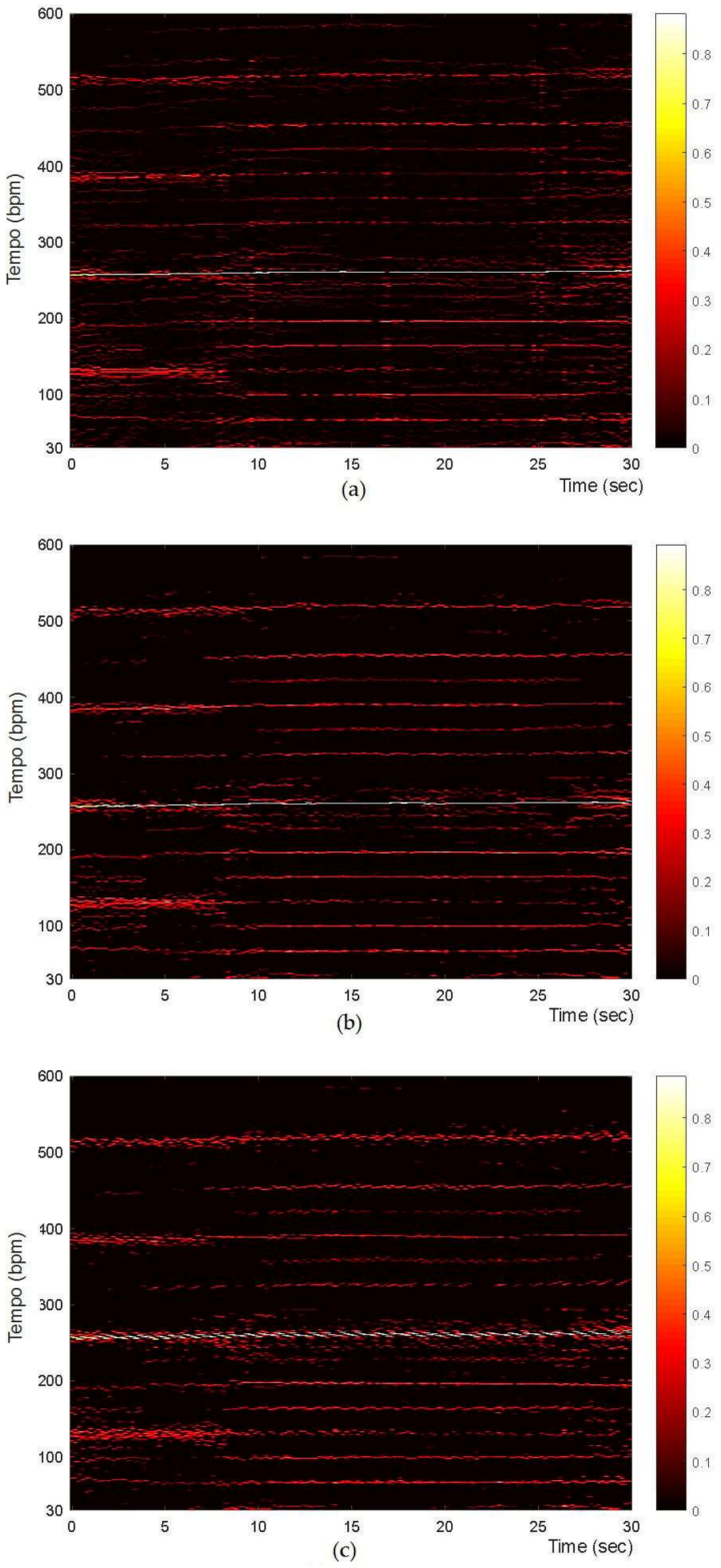

Figure 7. Tempograms with different sparsity. (**a**) Hop size = 2; (**b**) hop size = 5; and, (**c**) hop size = 20.

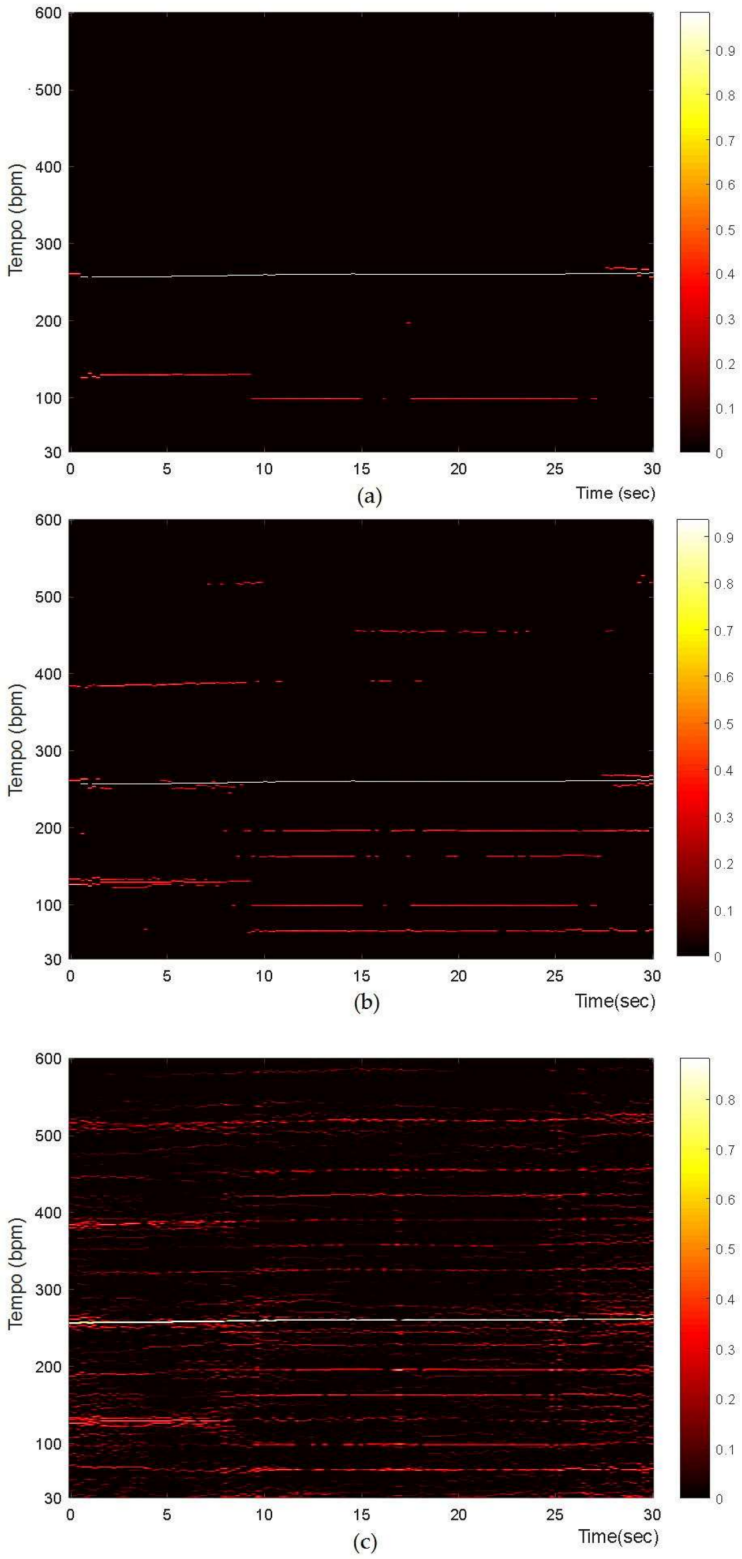

Figure 8. Tempograms with different First Few Main Tempo Components (FFMTC). (**a**) Number of iteration = 2; (**b**) Number of iteration = 5; and, (**c**) Number of iteration = 20.

5. Application Example

To illustrate how to use tempogram based on MP, we give an application example, where the tempogram was used to estimate constant tempo in the proceeding of the music. The experiment was

performed again on the clip train1.wav mentioned above. We created the tempogram based on MP with the number of iteration of 571, the tempo resolution of 1 bpm, and the hop size of 2. Once the tempogram was created, a matrix $S = S(b,n),\ b = [1 \dots B],\ n = [1 \dots N]$ was obtained, where b is the tempo index and n is the frame number of the novelty curve. We could estimate the tempo according to the following steps:

5.1. Compute the Tempo Curve

For every tempo index b, we sum the corresponding coefficients of the entire frame to get the tempo curve vector:

$$T(b) = \sum_{n=1}^{N} S(b,n) \tag{4}$$

For computational convenience, we expand the tempo band from $\tau \in [30,600]\,, \tau \in Z$ to $\tau \in [1,600]\,, \tau \in Z$. So, we have $b \in [1,600]$ and certainly $T(1:29) = 0$. The tempo curve is shown in Figure 9a.

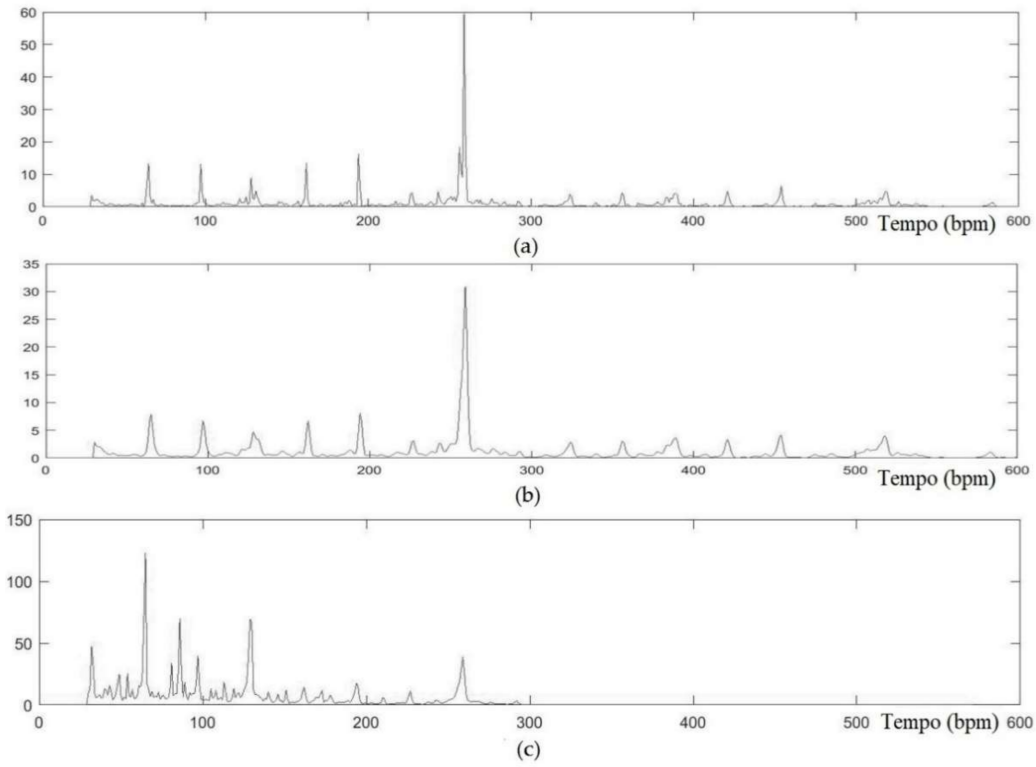

Figure 9. Tempo estimation example using tempogram based on MP. (**a**) tempo curve; (**b**) smoothed tempo curve; and, (**c**) Modified tempo curve by the comb template.

5.2. Smooth the Tempo Curve

The initial tempo curve is not smooth, which will affect us to obtain the dominant tempo components clearly. As an example, there is a branch that we would not want to see on the left of the biggest component in the Figure 9a. We should preprocess the curve by smoothing, in which many smoothing methods could be used. We adopt the Gaussian Kernel [35] method, which took a good effect on note onset detection. The smoothed curve is showed in the Figure 9b, in which the branch is eliminated. We denote the processed curve as $M(b)$.

5.3. Modify the Tempo Curve by the Comb Template

Because of the hierarchy of beat structure, there always exists harmonic structure in the tempo curve, called "harmonic tempi". Generally, the initial biggest component in the curve is inclined to the tempo of the pulse of the smallest interval beat. However, the ground truth of tempo tends to be the smaller and harmonic one. Therefore, we should take some measures to modify the tempo curve. We adopt the weighted comb template similar to the approach used in the paper [36]:

$$\lambda_\tau(l) = \sum_{p=1}^{P} p \cdot \delta(l - \tau \cdot p) \tag{5}$$

where $\tau \in [30, 300]$, $\tau \in Z$ and $l \in [1, 600]$, $l \in Z, P = 4$. Then, we derive the new tempo curve (see Figure 9c) by:

$$C(\tau) = \sum_{b=1}^{B} M(b)\lambda_\tau(b) \tag{6}$$

where $\tau \in [30, 300]$, $\tau \in Z$.

5.4. Choose Two Dominant Tempi as the Estimation Result

Just like MIREX, we choose two dominant tempi as the estimation result and then evaluate the result. The biggest component in the modified tempo curve is selected as the first tempo T_1 directly. Because the ground truth of tempo generally lies in the "harmonic tempi", we delete the points $\tau \in [0.2 \cdot T_1, 1.8 \cdot T_1]$ and choose the biggest component as the second tempo T_2 in the residual curve. Finally, the estimation result is $T_1 = 65, T_2 = 129$. The performance, P, will be given by the following equation, as MIREX defined:

$$P = ST_1 \cdot TT_1 + (1 - ST_1) \cdot TT_2 \tag{7}$$

where ST_1 is the relative perceptual strength of T_1 (given by ground truth data, varies from 0 to 1.0), TT_1 is the ability of the algorithm to identify T_1 to within 8%, and TT_2 is the ability of the algorithm to identify T_2 to within 8%. Because the ground truth is $[64.5, 129.5, 0.12]$, the evaluation result is $P = 1$.

6. Conclusions

Tempogram is one of the most important mid-level features in the field of the music information retrieval, which is widely applied to the studies, such as music tempo estimation, beat tracking, music structure analysis, and rhythm recognition. In this paper, we present a novel tempogram generating algorithm based on matching pursuit. When compared with ACF and FT, the algorithm can produce higher tempo resolution, stronger sparsity, and flexibility. However, its computational cost is higher. The higher the tempo resolution and/or the stronger the sparsity, the higher the cost will be. We suggest that a suitable algorithm should be chosen when using the tempogram. In the future, we aim to use this type of tempogram to improve the performances of tempo estimation, beat tracking, and music structure analysis.

Acknowledgments: We wish to gratefully acknowledge the support for this research provided by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (Grant No. 16KJB520013), the National Social Science Fund of China (ML, Grant No. 16CXW027), the Fundamental Research Funds for the Central Universities (ML, Grant No. B16020303) and the Chinese National Natural Science Foundation (Grant No. 61401227).

Author Contributions: Wenming Gui and Yao Sun conceived and designed the experiments; Yuting Tao, Yanping Li, and Lun Meng performed the experiments; Wenming Gui wrote the paper; Jinglan Zhang fine-tuned the paper and gave some useful advices.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wikipedia. Tempo. Available online: https://en.wikipedia.org/wiki/Tempo (accessed on 10 December 2017).
2. Durand, S.; Bello, J.; David, B.; Richard, G. Robust downbeat tracking using an ensemble of convolutional networks. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2017**, *25*, 76–89. [CrossRef]
3. Elowsson, A. Beat tracking with a cepstroid invariant neural network. In Proceedings of the 17th ISMIR Conference, New York, NY, USA, 7–11 August 2016.
4. Mottaghi, A.; Behdin, K.; Esmaeili, A.; Heydari, M.; Marvasti, F. Obtain: Real-time beat tracking in audio signals. *arXiv* **2017**, arXiv:1704.02216.
5. Tian, M.; Fazekas, G.; Black, D.A.A.; Sandler, M. On the use of the tempogram to describe audio content and its application to music structural segmentation. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, QLD, Australia, 19–24 April 2015.
6. Gkiokas, A.; Katsouros, V.; Carayannis, G. Towards multi-purpose spectral rhythm features: An application to dance style, meter and tempo estimation. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2016**, *24*, 1885–1896. [CrossRef]
7. Kartikay, A.; Ganesan, H.; Ladwani, V.M. Classification of music into moods using musical features. In Proceedings of the International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 26–27 August 2016.
8. Grosche, P.; Müller, M.; Kurth, F. Cyclic tempogram—A mid-level tempo representation for musicsignals. In Proceedings of the 2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), Dallas, TX, USA, 14–19 March 2010; pp. 5522–5525.
9. Wu, F.-H.F. Musical tempo octave error reducing based on the statistics of tempogram. In Proceedings of the 2015 23th Mediterranean Conference on Control and Automation (MED), Torremolinos, Spain, 16–19 June 2015.
10. Ellis, D.P.W. Beat tracking by dynamic programming. *J. New Music Res.* **2007**, *36*, 51–60. [CrossRef]
11. Peeters, G. Time variable tempo detection and beat marking. In Proceedings of the 2015 International Computer Music Conference, Tucson, AZ, USA, 28 June–2 July 2015.
12. Durand, S.; Bello, J.P.; David, B.; Richard, G.L.; Fillon, T.; Joder, C.; Durand, S.; Essid, S. Downbeat tracking with multiple features and deep neural networks a conditional random field system for beat tracking. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, QLD, Australia, 19–24 April 2015.
13. Fillon, T.; Joder, C.; Durand, S.; Essid, S. A conditional random field system for beat tracking. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, QLD, Australia, 19–24 April 2015; pp. 424–428.
14. Bello, J.P.; Daudet, L.; Abdallah, S.; Duxbury, C.; Davies, M.; Sandler, M.B. A tutorial on onset detection in music signals. *IEEE Trans. Speech Audio Process.* **2005**, *13*, 1035–1047. [CrossRef]
15. Holzapfel, A.; Stylianou, Y.; Gedik, A.C.; Bozkurt, B. Three dimensions of pitched instrument onset detection. *IEEE Trans. Audio Speech Lang. Process.* **2010**, *18*, 1517–1527. [CrossRef]
16. Shao, X.; Gui, W.; Xu, C. Note onset detection based on sparse decomposition. *Multimed. Tools Appl.* **2016**, *75*, 2613–2631. [CrossRef]
17. Tan, H.L.; Zhu, Y.; Chaisorn, L.; Rahardja, S. Audio onset detection using energy-based and pitch-based processing. In Proceedings of the 2010 IEEE International Symposium on Circuits and Systems (ISCAS), Paris, France, 30 May–2 June 2010.
18. Abdallah, S.A.; Plumbley, M.D. Probability as metadata: Event detection in music using ICA as a conditional density model. In Proceedings of the 4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003), Nara, Japan, 1–4 April 2003.
19. Schlüter, J.; Böck, S. Musical onset detection with convolutional neural networks. In Proceedings of the 6th International Workshop on Machine Learning and Music, Prague, Czech Republic, 23 September 2013.
20. Stasiak, B.; Mońko, J.; Niewiadomski, A. Note onset detection in musical signals via neural–network–based multi–odf fusion. *Int. J. Appl. Math. Comput. Sci.* **2016**, *26*, 203–213. [CrossRef]
21. Cemgil, A.T.; Kappen, B.; Desain, P.; Honing, H. On tempo tracking: Tempogram representation and kalman filtering. *J. New Music Res.* **2000**, *29*, 259–273. [CrossRef]
22. Scheirer, E.D. Tempo and beat analysis of acoustic musical signals. *J. Acoust. Soc. Am.* **1998**, *103*, 588–601. [CrossRef] [PubMed]

23. Klapuri, A.P.; Eronen, A.J.; Astola, J.T. Analysis of the meter of acoustic musical signals. *IEEE Trans. Audio Speech Lang. Process.* **2006**, *14*, 342–355. [CrossRef]
24. Tzanetakis, G.; Cook, P. Musical genre classification of audio signals. *IEEE Trans. Speech Audio Process.* **2002**, *10*, 293–302. [CrossRef]
25. Tzanetakis, G. Tempo extraction using beat histograms. In Proceedings of the 1st Music Information Retrieval Evaluation eXchange (MIREX 2005), London, UK, 11–15 September 2005.
26. Eck, D. Identifying metrical and temporal structure with an autocorrelation phase matrix. *Music Percept. Interdiscip. J.* **2006**, *24*, 167–176. [CrossRef]
27. Foote, J.; Uchihashi, S. The beat spectrum: A new approach to rhythm analysis. In Proceedings of the 2001 IEEE International Conference on Multimedia and Expo (ICME 2001), Tokyo, Japan, 22–25 August 2001.
28. Rudrich, D.; Sontacchi, A. Beat-aligning guitar looper. In Proceedings of the 20th International Conference on Digital Audio Effects (DAFx-17), Edinburgh, UK, 5–9 September 2017; pp. 451–458.
29. Laroche, J. Efficient tempo and beat tracking in audio recordings. *J. Audio Eng. Soc.* **2003**, *51*, 226–233.
30. Oliveira, J.L.; Davies, M.E.P.; Gouyon, F.; Reis, L.P. Beat tracking for multiple applications: A multi-agent system architecture with state recovery. *IEEE Trans. Audio Speech Lang. Process.* **2012**, *20*, 2696–2706. [CrossRef]
31. Peeters, G.; Papadopoulos, H. Simultaneous beat and downbeat-tracking using a probabilistic framework: Theory and large-scale evaluation. *IEEE Trans. Audio Speech Lang. Process.* **2011**, *19*, 1754–1769. [CrossRef]
32. Eronen, A.J.; Klapuri, A.P. Music tempo estimation with k-NN regression. *IEEE Trans. Audio Speech Lang. Process.* **2010**, *18*, 50–57. [CrossRef]
33. Grosche, P.; Müller, M. Tempogram toolbox: Matlab implementations for tempo and pulse analysis of music recordings. In Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR), Miami, FL, USA, 24–28 October 2011.
34. MIREX. Mirex Tempo Estimation. Available online: http://www.music-ir.org/mirex/wiki/2016:Audio_Tempo_Estimation (accessed on 9 July 2017).
35. Gui, W.; Shao, X.; Ren, C.; Bai, G. Note onset detection based on Gaussian kernel smoothing. *J. Inf. Comput. Sci.* **2011**, *8*, 3401–3409.
36. Davies, M.E.; Plumbley, M.D. Context-dependent beat tracking of musical audio. *IEEE Trans. Audio Speech Lang. Process.* **2007**, *15*, 1009–1020. [CrossRef]